

# Dynamic Anomalography: Tracking Network Anomalies Via Sparsity and Low Rank

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

**Abstract**—In the backbone of large-scale networks, origin-to-destination (OD) traffic flows experience abrupt unusual changes known as *traffic volume anomalies*, which can result in congestion and limit the extent to which end-user quality of service requirements are met. As a means of maintaining seamless end-user experience in dynamic environments, as well as for ensuring network security, this paper deals with a crucial network monitoring task termed *dynamic anomalography*. Given link traffic measurements (noisy superpositions of unobserved OD flows) periodically acquired by backbone routers, the goal is to construct an estimated map of anomalies in real time, and thus summarize the network ‘health state’ along both the flow and time dimensions. Leveraging the low intrinsic-dimensionality of OD flows and the sparse nature of anomalies, a novel online estimator is proposed based on an exponentially-weighted least-squares criterion regularized with the sparsity-promoting  $\ell_1$ -norm of the anomalies, and the nuclear norm of the nominal traffic matrix. After recasting the non-separable nuclear norm into a form amenable to online optimization, a real-time algorithm for dynamic anomalography is developed and its convergence established under simplifying technical assumptions. For operational conditions where computational complexity reductions are at a premium, a lightweight stochastic gradient algorithm based on Nesterov’s acceleration technique is developed as well. Comprehensive numerical tests with both synthetic and real network data corroborate the effectiveness of the proposed online algorithms and their tracking capabilities, and demonstrate that they outperform state-of-the-art approaches developed to diagnose traffic anomalies.

**Index Terms**—Traffic volume anomalies, online optimization, sparsity, network cartography, low rank.

## I. INTRODUCTION

**I**N the backbone of large-scale networks, origin-to-destination (OD) traffic flows experience abrupt unusual changes which can result in congestion, and limit QoS provisioning of the end users. These so-termed *traffic volume anomalies* could be due to unexpected failures in networking equipment, cyberattacks (e.g., denial of service (DoS) attacks), or, intruders which hijack the network services [37]. Unveiling such anomalies in a promptly manner is a crucial monitoring task towards engineering network traffic. This is a challenging task however,

Manuscript received July 31, 2012; revised November 02, 2012; accepted December 01, 2012. Date of publication December 11, 2012; date of current version January 22, 2013. This work was supported by the MURI Grant AFOSR FA9550-10-1-0567. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ery Arias-Castro.

The authors are with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: morteza@umn.edu; mate0058@umn.edu; georgios@umn.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/JSTSP.2012.2233193

since the available data are usually high-dimensional, noisy and possibly incomplete link-load measurements, which are the superposition of *unobservable* OD flows. Several studies have experimentally demonstrated the low intrinsic dimensionality of the nominal traffic subspace, that is, the intuitive *low-rank* property of the traffic matrix in the absence of anomalies, which is mainly due to common temporal patterns across OD flows, and periodic behavior across time [22], [43]. Exploiting the low-rank structure of the anomaly-free traffic matrix, a landmark principal component analysis (PCA)-based method was put forth in [21] to identify network anomalies; see also [28] for a distributed implementation. A limitation of the algorithm in [21] is that it cannot identify multiple anomalous flows. Most importantly, [21] has not exploited the *sparsity* of anomalies across flows and time—anomalous traffic spikes are rare, and tend to last for short periods of time relative to the measurement horizon.

Capitalizing on the low-rank property of the traffic matrix and the sparsity of the anomalies, the fresh look advocated here permeates benefits from rank minimization [9]–[11], and compressive sampling [12], [13], to perform *dynamic anomalography*. The aim is to construct a map of network anomalies in real time, that offers a succinct depiction of the network ‘health state’ across both the flow and time dimensions (Section II). Different from the *batch* centralized and distributed anomalography algorithms in [26] and [25], the focus here is on devising *online* (adaptive) algorithms that are capable of efficiently processing link measurements and track network anomalies ‘on the fly’; see also [4] for a ‘model-free’ approach that relies on the kernel recursive LS (RLS) algorithm. Online monitoring algorithms are attractive for operation in dynamic network environments, since they can cope with traffic nonstationarities arising due to routing changes and missing data. Accordingly, the novel online estimator entails an exponentially-weighted least-squares (LS) cost regularized with the sparsity-promoting  $\ell_1$ -norm of the anomalies, and the nuclear norm of the nominal traffic matrix. After recasting the non-separable nuclear norm into a form amenable to online optimization (Section III.A), a real-time algorithm for dynamic anomalography is developed in Section IV based on alternating minimization. Each time a new datum is acquired, anomaly estimates are formed via the least-absolute shrinkage and selection operator (Lasso), e.g., ([18], p. 68), and the low-rank nominal traffic subspace is refined using RLS [36]. Convergence analysis is provided under simplifying technical assumptions in Section IV.B. For situations where reducing computational complexity is critical, an online stochastic gradient algorithm based on Nesterov’s acceleration technique [6], [30] is developed as well (Section V.A).

The possibility of implementing the anomaly trackers in a distributed fashion is further outlined in Section V.B, where several directions for future research are also delineated.

Extensive numerical tests involving both synthetic and real network data corroborate the effectiveness of the proposed algorithms in unveiling network anomalies, as well as their tracking capabilities when traffic routes are slowly time-varying, and the network monitoring station acquires incomplete link traffic measurements (Section VI). Different from [42] which employs a two-step batch procedure to learn the nominal traffic subspace first, and then unveil anomalies via  $\ell_1$ -norm minimization, the approach here estimates both quantities jointly and attains better performance as illustrated in Section VI.B. Concluding remarks are given in Section VII, while most technical details relevant to the convergence proof in Section IV.C are deferred to the Appendix.

*Notation:* Bold uppercase (lowercase) letters will denote matrices (column vectors), and calligraphic letters will be used for sets. Operators  $(\cdot)'$ ,  $\text{tr}(\cdot)$ ,  $\lambda_{\min}(\cdot)$ ,  $\sigma_{\min}(\cdot)$ ,  $[\cdot]_+$ , and  $\mathbb{E}[\cdot]$ , will denote transposition, matrix trace, minimum eigenvalue, minimum singular value, projection onto the nonnegative orthant, and expectation, 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  $\ell_p$ -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  $\mathbf{M}, \mathbf{U} \in \mathbb{R}^{n \times n}$ ,  $\langle \mathbf{M}, \mathbf{U} \rangle := \text{tr}(\mathbf{M}'\mathbf{U})$  denotes their trace inner product. The Frobenius norm of matrix  $\mathbf{M} = [m_{i,j}] \in \mathbb{R}^{n \times p}$  is  $\|\mathbf{M}\|_F := \sqrt{\text{tr}(\mathbf{M}\mathbf{M}')$ ,  $\|\mathbf{M}\| := \max_{\|\mathbf{x}\|_2=1} \|\mathbf{M}\mathbf{x}\|_2$  is the spectral norm,  $\|\mathbf{M}\|_1 := \sum_{i,j} |m_{i,j}|$  is the  $\ell_1$ -norm, and  $\|\mathbf{M}\|_* := \sum_i \sigma_i(\mathbf{M})$  is the nuclear norm, where  $\sigma_i(\mathbf{M})$  denotes the  $i$ -th singular value of  $\mathbf{M}$ . The  $n \times n$  identity matrix will be represented by  $\mathbf{I}_n$  and its  $i$ -th column by  $\mathbf{e}_i$ , 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 the support set  $\text{supp}(\mathbf{x}) := \{i : x_i \neq 0\}$ .

## II. MODELING PRELIMINARIES AND PROBLEM STATEMENT

Consider a backbone Internet protocol (IP) network naturally modeled as a directed graph  $G(\mathcal{N}, \mathcal{L})$ , where  $\mathcal{N}$  and  $\mathcal{L}$  denote the sets of nodes (routers) and physical links of cardinality  $|\mathcal{N}| = N$  and  $|\mathcal{L}| = L$ , respectively. The operational goal of the network is to transport a set of OD traffic flows  $\mathcal{F}$  (with  $|\mathcal{F}| = F$ ) associated with specific source-destination pairs. For backbone networks, the number of network layer flows is much larger than the number of physical links ( $F \gg L$ ). Single-path routing is adopted here, that is, a given flow's traffic is carried through multiple links connecting the corresponding source-destination pair along a single path. Let  $r_{l,f}$ ,  $l \in \mathcal{L}$ ,  $f \in \mathcal{F}$ , denote the flow to link assignments (routing), which take the value one whenever flow  $f$  is carried over link  $l$ , and zero otherwise. Unless otherwise stated, the routing matrix  $\mathbf{R} := [r_{l,f}] \in \{0, 1\}^{L \times F}$  is assumed fixed and given. Likewise, let  $z_{f,t}$  denote the unknown traffic rate of flow  $f$  at time  $t$ , measured in e.g., Mbps. At any given time instant  $t$ , the traffic carried over link  $l$  is then the superposition of the flow rates routed through link  $l$ , i.e.,  $\sum_{f \in \mathcal{F}} r_{l,f} z_{f,t}$ .

It is not uncommon for some of the flow rates to experience unusual abrupt changes. These so-termed *traffic volume anomalies* are typically due to unexpected network failures, or cyberattacks (e.g., DoS attacks) which aim at compromising the services offered by the network [37]. Let  $a_{f,t}$  denote the unknown traffic volume anomaly of flow  $f$  at time  $t$ . In the presence of anomalous flows, the measured traffic carried by link  $l$  over a time horizon  $t \in [1, T]$  is then given by

$$y_{l,t} = \sum_{f \in \mathcal{F}} r_{l,f} (z_{f,t} + a_{f,t}) + v_{l,t}, \quad t = 1, \dots, T \quad (1)$$

where the noise variables  $v_{l,t}$  account for measurement errors and unmodeled dynamics.

In IP networks, traffic volume can be readily measured on a per-link basis using off-the-shelf tools such as the simple network management protocol (SNMP) supported by most routers. Missing entries in the link-level measurements  $y_{l,t}$  may however skew the network operator's perspective. SNMP packets may be dropped for instance, if some links become congested, rendering link count information for those links more important, as well as less available [33]. To model missing link measurements, collect the tuples  $(l, t)$  associated with the available observations  $y_{l,t}$  in the set  $\Omega \in [1, 2, \dots, L] \times [1, 2, \dots, T]$ . Introducing the matrices  $\mathbf{Y} := [y_{l,t}]$ ,  $\mathbf{V} := [v_{l,t}] \in \mathbb{R}^{L \times T}$ , and  $\mathbf{Z} := [z_{f,t}]$ ,  $\mathbf{A} := [a_{f,t}] \in \mathbb{R}^{F \times T}$ , the (possibly incomplete) set of measurements in (1) can be expressed in compact matrix form as

$$\mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{R}(\mathbf{Z} + \mathbf{A}) + \mathbf{V}) \quad (2)$$

where the sampling operator  $\mathcal{P}_\Omega(\cdot)$  sets the entries of its matrix argument not in  $\Omega$  to zero, and keeps the rest unchanged. Matrix  $\mathbf{Z}$  contains the nominal traffic flows over the time horizon of interest. Common temporal patterns among the traffic flows in addition to their periodic behavior, render most rows (respectively columns) of  $\mathbf{Z}$  linearly dependent, and thus  $\mathbf{Z}$  typically has low rank. This intuitive property has been extensively validated with real network data; see e.g., [22]. Anomalies in  $\mathbf{A}$  are expected to occur sporadically over time, and last shortly relative to the (possibly long) measurement interval  $[1, T]$ . In addition, only a small fraction of the flows is supposed to be anomalous at a any given time instant. This renders the anomaly traffic matrix  $\mathbf{A}$  sparse across both rows (flows) and columns (time).

Given measurements  $\mathcal{P}_\Omega(\mathbf{Y})$  adhering to (2) and the binary-valued routing matrix  $\mathbf{R}$ , the main goal of this paper is to accurately estimate the anomaly matrix  $\mathbf{A}$ , by capitalizing on the sparsity of  $\mathbf{A}$  and the low-rank property of  $\mathbf{Z}$ . Special focus will be placed on devising online (adaptive) algorithms that are capable of efficiently processing link measurements and tracking network anomalies in real time. This critical monitoring task is termed *dynamic anomalography*, and the resultant estimated map  $\hat{\mathbf{A}}$  offers a depiction of the network's 'health state' along both the flow and time dimensions. If  $|\hat{a}_{f,t}| > 0$ , the  $f$ -th flow at time  $t$  is deemed anomalous, otherwise it is healthy. By examining  $\mathbf{R}$  the network operator can immediately determine the links carrying the anomalous flows. Subsequently, planned contingency measures involving traffic-engineering algorithms can be implemented to address network congestion.

### III. UNVEILING ANOMALIES VIA SPARSITY AND LOW RANK

Consider the nominal link-count traffic matrix  $\mathbf{X} := \mathbf{R}\mathbf{Z}$ , which inherits the low-rank property from  $\mathbf{Z}$ . Since the primary goal is to recover  $\mathbf{A}$ , the following observation model

$$\mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{X} + \mathbf{R}\mathbf{A} + \mathbf{V}) \quad (3)$$

can be adopted instead of (2). A natural estimator leveraging the low rank property of  $\mathbf{X}$  and the sparsity of  $\mathbf{A}$  will be sought next. The idea is to fit the incomplete data  $\mathcal{P}_\Omega(\mathbf{Y})$  to the model  $\mathbf{X} + \mathbf{R}\mathbf{A}$  in the least-squares (LS) error sense, as well as minimize the rank of  $\mathbf{X}$ , and the number of nonzero entries of  $\mathbf{A}$  measured by its  $\ell_0$ -(pseudo) norm. Unfortunately, albeit natural both rank and  $\ell_0$ -norm criteria are in general NP-hard to optimize [17], [29]. Typically, the nuclear norm  $\|\mathbf{X}\|_*$  and the  $\ell_1$ -norm  $\|\mathbf{A}\|_1$  are adopted as surrogates, since they are the closest *convex* approximants to  $\text{rank}(\mathbf{X})$  and  $\|\mathbf{A}\|_0$ , respectively [12], [31], [38]. Accordingly, one solves

$$(P1) \quad \min_{\{\mathbf{X}, \mathbf{A}\}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{X} - \mathbf{R}\mathbf{A})\|_F^2 + \lambda_* \|\mathbf{X}\|_* + \lambda_1 \|\mathbf{A}\|_1$$

where  $\lambda_*$ ,  $\lambda_1 \geq 0$  are rank- and sparsity-controlling parameters. When an estimate  $\hat{\sigma}_v^2$  of the noise variance is available, guidelines for selecting  $\lambda_*$  and  $\lambda_1$  have been proposed in [44].

Being convex (P1) is appealing, and it yields reliable performance when full data are available, i.e.,  $\Omega = \emptyset$  [26]. In the presence of missing data, one has to ensure that the sampled subset of links provides sufficient information to identify anomalous flows. Intuitively, for high estimation accuracy each flow must traverse sufficiently many links, whereas network links should not be overloaded by too many flows. These properties typically hold for large-scale networks with distant OD node pairs, and routing paths that are sufficiently ‘spread-out.’ Developing identifiability conditions when link measurements are incomplete is an open problem, and constitutes an interesting future research direction.

Model (3) and its estimator (P1) are quite general, as discussed in the ensuing remark.

*Remark 1 (Subsumed Paradigms):* When there is no missing data and  $\mathbf{X} = \mathbf{0}_{L \times T}$ , one is left with an under-determined sparse signal recovery problem typically encountered with compressive sampling (CS); see e.g., [12] and the tutorial account [13]. The decomposition  $\mathbf{Y} = \mathbf{X} + \mathbf{A}$  corresponds to principal component pursuit (PCP), also referred to as robust principal component analysis (RPCA) [9], [14]. PCP was adopted for network anomaly detection using flow (not link traffic) measurements in [2]. For the idealized noise-free setting ( $\mathbf{V} = \mathbf{0}_{L \times T}$ ), sufficient conditions for exact recovery are available for both of the aforementioned special cases [9], [12], [14]. However, the superposition of a low-rank plus a *compressed* sparse matrix in (3) further challenges identifiability of  $\{\mathbf{X}, \mathbf{A}\}$ ; see [26] for early results. Going back to the CS paradigm, even when  $\mathbf{X}$  is nonzero one could envision a variant where the measurements are corrupted with correlated (low-rank) noise [15]. Last but not least, when  $\mathbf{A} = \mathbf{0}_{F \times T}$  and  $\mathbf{Y}$  is noisy, the recovery of  $\mathbf{X}$  subject to a rank constraint is nothing but PCA—arguably, the workhorse of high-dimensional data analytics. This same

formulation is adopted for low-rank matrix completion, to impute the missing entries of a low-rank matrix observed in noise, i.e.,  $\mathcal{P}_\Omega(\mathbf{Y}) = \mathcal{P}_\Omega(\mathbf{X} + \mathbf{V})$  [10].

Albeit convex, (P1) is a non-smooth optimization problem (both the nuclear and  $\ell_1$ -norms are not differentiable at the origin). In addition, scalable algorithms to unveil anomalies in large-scale networks should effectively overcome the following challenges: (c1) the problem size can easily become quite large, since the number of optimization variables is  $(L + F)T$ ; (c2) existing iterative solvers for (P1) typically rely on costly SVD computations per iteration; see e.g., [26]; and (c3) different from the Frobenius and  $\ell_1$ -norms, (columnwise) nonseparability of the nuclear-norm challenges online processing when new columns of  $\mathcal{P}_\Omega(\mathbf{Y})$  arrive sequentially in time. In the remainder of this section, the ‘big data’ challenges (c1) and (c2) are dealt with to arrive at an efficient batch algorithm for anomalography. Tracking network anomalies is the main subject of Section IV.

To address (c1) and reduce the computational complexity and memory storage requirements of the algorithms sought, it is henceforth assumed that an upper bound  $\rho \geq \text{rank}(\hat{\mathbf{X}})$  is a priori available [ $\hat{\mathbf{X}}$  is the estimate obtained via (P1)]. As argued next, the smaller the value of  $\rho$ , the more efficient the algorithm becomes. Small values of  $\rho$  are well motivated due to the low intrinsic dimensionality of network flows. For instance, experiments with Internet-2 network data [1] show that  $\rho = 5$  suffices [21]; see also [22]. Because  $\text{rank}(\hat{\mathbf{X}}) \leq \rho$ , (P1)’s search space is effectively reduced and one can factorize the decision variable as  $\mathbf{X} = \mathbf{P}\mathbf{Q}'$ , where  $\mathbf{P}$  and  $\mathbf{Q}$  are  $L \times \rho$  and  $T \times \rho$  matrices, respectively. It is possible to interpret the columns of  $\mathbf{X}$  (viewed as points in  $\mathbb{R}^L$ ) as belonging to a low-rank ‘nominal traffic subspace’, spanned by the columns of  $\mathbf{P}$ . The rows of  $\mathbf{Q}$  are thus the projections of the columns of  $\mathbf{X}$  onto the traffic subspace.

Adopting this reparametrization of  $\mathbf{X}$  in (P1), and defining  $r(\mathbf{P}, \mathbf{Q}, \mathbf{A}) := (1/2) \|\mathcal{P}_\Omega(\mathbf{Y} - \mathbf{P}\mathbf{Q}' - \mathbf{R}\mathbf{A})\|_F^2$ , one arrives at an equivalent optimization problem

$$(P2) \quad \min_{\{\mathbf{P}, \mathbf{Q}, \mathbf{A}\}} r(\mathbf{P}, \mathbf{Q}, \mathbf{A}) + \lambda_* \|\mathbf{P}\mathbf{Q}'\|_* + \lambda_1 \|\mathbf{A}\|_1$$

which is non-convex due to the bilinear terms  $\mathbf{P}\mathbf{Q}'$ . The number of variables is reduced from  $(L + F)T$  in (P1), to  $\rho(L + T) + FT$  in (P2). The savings can be significant when  $\rho$  is small, and both  $L$  and  $T$  are large. Note that the dominant  $FT$ -term in the variable count of (P2) is due to  $\mathbf{A}$ , which is sparse and can be efficiently handled even when both  $F$  and  $T$  are large.

#### A. A Separable Low-Rank Regularization

To address (c2) [along with (c3) as it will become clear in Section IV], consider the following alternative characterization of the nuclear norm [31], [32]

$$\|\mathbf{X}\|_* := \min_{\{\mathbf{P}, \mathbf{Q}\}} \frac{1}{2} \{ \|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2 \}, \quad \text{s. to } \mathbf{X} = \mathbf{P}\mathbf{Q}' \quad (4)$$

The optimization (4) is over all possible bilinear factorizations of  $\mathbf{X}$ , so that the number of columns  $\rho$  of  $\mathbf{P}$  and  $\mathbf{Q}$  is also a variable. Leveraging (4), the following reformulation of (P2)

provides an important first step towards obtaining an online algorithm:

$$(P3) \min_{\{\mathbf{P}, \mathbf{Q}, \mathbf{A}\}} r(\mathbf{P}, \mathbf{Q}, \mathbf{A}) + \frac{\lambda_*}{2} \{ \|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2 \} + \lambda_1 \|\mathbf{A}\|_1.$$

As asserted in [25, Lemma 1], adopting the separable Frobenius-norm regularization in (P3) comes with no loss of optimality relative to (P1), provided  $\rho \geq \text{rank}(\hat{\mathbf{X}})$ . By finding the global minimum of (P3) [which could have considerably less variables than (P1)], one can recover the optimal solution of (P1). However, since (P3) is non-convex, it may have stationary points which need not be globally optimum. Interestingly, the next proposition shows that under relatively mild assumptions on  $\text{rank}(\hat{\mathbf{X}})$  and the noise variance, every stationary point of (P3) is globally optimum for (P1). For a proof, see [25, App. A], .

*Proposition 1:* Let  $\{\bar{\mathbf{P}}, \bar{\mathbf{Q}}, \bar{\mathbf{A}}\}$  be a stationary point of (P3). If  $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{P}}\bar{\mathbf{Q}}' - \mathbf{R}\bar{\mathbf{A}})\| \leq \lambda_*$ , then  $\{\hat{\mathbf{X}} := \bar{\mathbf{P}}\bar{\mathbf{Q}}', \hat{\mathbf{A}} = \bar{\mathbf{A}}\}$  is the globally optimal solution of (P1).

The qualification condition  $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{P}}\bar{\mathbf{Q}}' - \mathbf{R}\bar{\mathbf{A}})\| \leq \lambda_*$  captures tacitly the role of  $\rho$ . In particular, for sufficiently small  $\rho$  the residual  $\|\mathcal{P}_\Omega(\mathbf{Y} - \bar{\mathbf{P}}\bar{\mathbf{Q}}' - \mathbf{R}\bar{\mathbf{A}})\|$  becomes large and consequently the condition is violated [unless  $\lambda_*$  is large enough, in which case a sufficiently low-rank solution to (P1) is expected]. The condition on the residual also implicitly enforces  $\text{rank}(\hat{\mathbf{X}}) \leq \rho$ , which is necessary for the equivalence between (P1) and (P3). Note also that selecting a large value of  $\rho$  does not ensure satisfaction of the condition in Proposition 1. In fact, other factors such as the noise variance and routing matrix structure are critical as well.

### B. Batch Block Coordinate-Descent Algorithm

The block coordinate-descent (BCD) algorithm is adopted here to solve the batch non-convex optimization problem (P3). BCD is an iterative method which has been shown efficient in tackling large-scale optimization problems encountered with various statistical inference tasks, see e.g., [7]. The proposed solver entails an iterative procedure comprising three steps per iteration  $k = 1, 2, \dots$

**[S1] Update the anomaly map:**

$$\mathbf{A}[k+1] = \arg \min_{\mathbf{A}} [r(\mathbf{P}[k], \mathbf{Q}[k], \mathbf{A}) + \lambda_1 \|\mathbf{A}\|_1].$$

**[S2] Update the nominal traffic subspace:**

$$\mathbf{P}[k+1] = \arg \min_{\mathbf{P}} \left[ r(\mathbf{P}, \mathbf{Q}[k], \mathbf{A}[k+1]) + \frac{\lambda_*}{2} \|\mathbf{P}\|_F^2 \right].$$

**[S3] Update the projection coefficients:**

$$\mathbf{Q}[k+1] = \arg \min_{\mathbf{Q}} \left[ r(\mathbf{P}[k+1], \mathbf{Q}, \mathbf{A}[k+1]) + \frac{\lambda_*}{2} \|\mathbf{Q}\|_F^2 \right].$$

To update each of the variable groups, the cost of (P3) is minimized while fixing the rest of the variables to their most

up-to-date values. The minimization in [S1] decomposes over the columns of  $\mathbf{A} := [\mathbf{a}_1, \dots, \mathbf{a}_T]$ . At iteration  $k$ , these columns are updated in parallel via Lasso

$$\mathbf{a}_t[k+1] = \arg \min_{\mathbf{a}} \left[ \frac{1}{2} \|\Omega_t(\mathbf{y}_t - \mathbf{P}[k]\mathbf{q}_t[k] - \mathbf{R}\mathbf{a})\|_2^2 + \lambda_1 \|\mathbf{a}\|_1 \right], \quad t = 1, \dots, T \quad (5)$$

where  $\mathbf{y}_t$  and  $\mathbf{q}_t[k]$  respectively denote the  $t$ -th column of  $\mathbf{Y}$  and  $\mathbf{Q}'[k]$ , while the diagonal matrix  $\Omega_t \in \mathbb{R}^{L \times L}$  contains a one on its  $l$ -th diagonal entry if  $y_{l,t}$  is observed, and a zero otherwise. To keep computational complexity at a minimum, in practice each iteration of the proposed algorithm minimizes (5) inexactly. This is achieved for each  $t = 1, \dots, T$ , by performing a single pass of the cyclic coordinate-descent algorithm in [18, p. 92] over each one of the  $F$  scalar entries in  $\mathbf{a}_t[k+1]$ ; see Algorithm 1 for the resulting iterations, and Appendix A for further details. As shown at the end of this section, this inexact minimization suffices to claim convergence to a stationary point of (P3).

---

#### Algorithm 1: Batch BCD algorithm for unveiling network anomalies

---

**input**  $\mathcal{P}_\Omega(\mathbf{Y}), \Omega, \mathbf{R}, \lambda_*$ , and  $\lambda_1$ .

**initialize**  $\mathbf{P}[1]$  and  $\mathbf{Q}[1]$  at random.

**for**  $k = 1, 2, \dots$  **do**

**[S1] Update the anomaly map:**

**for**  $f = 1, \dots, F$  **do**

$$\tilde{\mathbf{y}}_t^{(-f)}[k+1] = \Omega_t(\mathbf{y}_t - \mathbf{P}[k]\mathbf{q}_t[k] - \sum_{f'=1}^{f-1} \mathbf{r}_{f'} a_{f',t}[k+1]$$

$$\times \sum_{f'=f+1}^F \mathbf{r}_{f'} a_{f',t}[k]), \quad t = 1, \dots, T.$$

$$a_{f,t}[k+1] = \text{sign}(\mathbf{r}_f' \tilde{\mathbf{y}}_t^{(-f)}[k+1]) [|\mathbf{r}_f' \tilde{\mathbf{y}}_t^{(-f)}[k+1] - \lambda_1|_+ \|\Omega_t \mathbf{r}_f\|_2^{-1}], \quad t = 1, \dots, T.$$

**end for**

$$\mathbf{A}[k+1] = [[a_{1,1}[k+1], \dots, a_{F,1}[k+1]]', \dots, [a_{1,T}[k+1], \dots, a_{F,T}[k+1]]']'.$$

**[S2] Update the nominal traffic subspace:**

$$\mathbf{p}_l[k+1] = (\lambda_* \mathbf{I}_\rho + \mathbf{Q}'[k] \Omega_t' \mathbf{Q}[k])^{-1} \mathbf{Q}'[k] \Omega_t' (\mathbf{y}_l^r - \mathbf{A}'[k+1] \mathbf{r}_l^r), \quad l = 1, \dots, L.$$

$$\mathbf{P}[k+1] = [\mathbf{p}_1[k+1], \dots, \mathbf{p}_L[k+1]]'.$$

**[S3] Update the projection coefficients:**

$$\mathbf{q}_t[k+1] = (\lambda_* \mathbf{I}_\rho + \mathbf{P}'[k+1] \Omega_t \mathbf{P}[k+1])^{-1} \times \mathbf{P}'[k+1] \Omega_t (\mathbf{y}_t - \mathbf{R}\mathbf{a}_t[k+1]), \quad t = 1, \dots, T$$

$$\mathbf{Q}[k+1] = [\mathbf{q}_1[k+1], \dots, \mathbf{q}_T[k+1]]'.$$

**end for**

**return**  $\hat{\mathbf{A}} := \mathbf{A}[\infty]$  and  $\hat{\mathbf{X}} := \mathbf{P}[\infty] \mathbf{Q}'[\infty]$ .

---

Similarly, in [S2] and [S3] the minimizations that give rise to  $\mathbf{P}[k+1]$  and  $\mathbf{Q}[k+1]$  are separable over their respective rows. For instance, the  $l$ -th row  $\mathbf{p}_l'$  of the traffic subspace matrix

$\mathbf{P} := [\mathbf{p}_1, \dots, \mathbf{p}_L]'$  is updated as the solution of the following ridge-regression problem

$$\mathbf{p}_l[k+1] = \arg \min_{\mathbf{p}} \left[ \frac{1}{2} \|((\mathbf{y}_l^r)' - \mathbf{p}'\mathbf{Q}'[k] - (\mathbf{r}_l^r)' \mathbf{A}[k+1]) \boldsymbol{\Omega}_l^r\|_2^2 + \frac{\lambda_*}{2} \|\mathbf{p}\|_2^2 \right] \quad (6)$$

where  $(\mathbf{y}_l^r)'$  and  $(\mathbf{r}_l^r)'$  represent the  $l$ -th row of  $\mathbf{Y}$  and  $\mathbf{R}$ , respectively. The  $t$ -th diagonal entry of the diagonal matrix  $\boldsymbol{\Omega}_l^r \in \mathbb{R}^{T \times T}$  is an indicator variable testing whether measurement  $y_{l,t}$  is available. Because (6) is an unconstrained convex quadratic program, the first-order optimality condition yields the closed-form solution tabulated under Algorithm 1. A similar regularized LS problem yields  $\mathbf{q}_t[k+1]$ ,  $t = 1, \dots, T$ ; see Algorithm 1 for the details and a description of the overall BCD solver. The novel batch scheme for unveiling network anomalies is less complex computationally than the accelerated proximal gradient algorithm in [26], since Algorithm 1's iterations are devoid of SVD computations. Different from [26], Algorithm 1 can also accommodate missing link measurements.

Despite being non-convex and non-differentiable, (P3) has favorable structure which facilitates convergence of the iterates generated by Algorithm 1. Specifically, the resulting cost is convex in each block variable when the rest are fixed. The non-smooth  $\ell_1$ -norm is also separable over the entries of its matrix argument. Accordingly, [39, Theorem 5.1] guarantees convergence of the BCD algorithm to a stationary point of (P3). This result together with Proposition 1 establishes the next claim.

*Proposition 2:* If a subsequence  $\{\mathbf{X}[k] := \mathbf{P}[k]\mathbf{Q}'[k], \mathbf{A}[k]\}$  of iterates generated by Algorithm 1 satisfies  $\|\mathcal{P}_{\Omega}(\mathbf{Y} - \mathbf{X}[k] - \mathbf{R}\mathbf{A}[k])\| \leq \lambda_*$ , then it converges to the optimal solution set of (P1) as  $k \rightarrow \infty$ .

In practice, it is desirable to monitor anomalies in real time and accommodate time-varying traffic routes. These reasons motivate devising algorithms for *dynamic* anomalography, the subject dealt with next.

#### IV. DYNAMIC ANOMALOGRAPHY

Monitoring of large-scale IP networks necessitates collecting massive amounts of data which far outweigh the ability of modern computers to store and analyze them in real time. In addition, nonstationarities due to routing changes and missing data further challenge identification of anomalies. In dynamic networks routing tables are constantly readjusted to effect traffic load balancing and avoid congestion caused by e.g., traffic anomalies or network infrastructure failures. To account for slowly time-varying routing tables, let  $\mathbf{R}_t \in \mathbb{R}^{L \times F}$  denote the routing matrix at time  $t$ <sup>1</sup>. In this dynamic setting, the partially observed link counts at time  $t$  adhere to [cf. (3)]

$$\mathcal{P}_{\Omega_t}(\mathbf{y}_t) = \mathcal{P}_{\Omega_t}(\mathbf{x}_t + \mathbf{R}_t \mathbf{a}_t + \mathbf{v}_t), \quad t = 1, 2, \dots \quad (7)$$

<sup>1</sup>Fixed size routing matrices  $\mathbf{R}_t$  are considered here for convenience, where  $L$  and  $F$  correspond to upper bounds on the number of physical links and flows transported by the network, respectively. If at time  $t$  some links are not used at all, or, less than  $F$  flows are present, the corresponding rows and columns of  $\mathbf{R}_t$  will be identically zero.



Fig. 1. Internet-2 network topology graph.

where the link-level traffic  $\mathbf{x}_t := \mathbf{R}_t \mathbf{z}_t$ , for  $\mathbf{z}_t$  from the (low-dimensional) traffic subspace. In general, routing changes may alter a link load considerably by e.g., routing traffic completely away from a specific link. Therefore, even though the network-level traffic vectors  $\{\mathbf{z}_t\}$  live in a low-dimensional subspace, the same may not be true for the link-level traffic  $\{\mathbf{x}_t\}$  when the routing updates are major and frequent. In backbone networks however, routing changes are sporadic relative to the time-scale of data acquisition used for network monitoring tasks. For instance, data collected from the operation of Internet-2 network, shown in Fig. 1, reveals that only a few rows of  $\mathbf{R}_t$  change per week [1]. It is thus safe to assume that  $\{\mathbf{x}_t\}$  still lies in a low-dimensional subspace, and exploit the temporal correlations of the observations to identify the anomalies.

On top of the previous arguments, in practice link measurements 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. The goal is then to recursively estimate  $\{\hat{\mathbf{x}}_t, \hat{\mathbf{a}}_t\}$  at time  $t$  from historical observations  $\{\mathcal{P}_{\Omega_\tau}(\mathbf{y}_\tau), \Omega_\tau\}_{\tau=1}^t$ , naturally placing more importance to recent measurements. To this end, one possible adaptive counterpart to (P3) is the exponentially-weighted LS estimator found by minimizing the empirical cost

$$\min_{\{\mathbf{P}, \mathbf{Q}, \mathbf{A}\}} \sum_{\tau=1}^t \beta^{t-\tau} \left[ \frac{1}{2} \|\mathcal{P}_{\Omega_\tau}(\mathbf{y}_\tau - \mathbf{P}\mathbf{q}_\tau - \mathbf{R}_\tau \mathbf{a}_\tau)\|_2^2 + \frac{\lambda_*}{2 \sum_{u=1}^t \beta^{t-u}} \|\mathbf{P}\|_F^2 + \frac{\lambda_*}{2} \|\mathbf{q}_\tau\|_2^2 + \lambda_1 \|\mathbf{a}_\tau\|_1 \right] \quad (8)$$

in which  $0 < \beta \leq 1$  is the so-termed forgetting factor. When  $\beta < 1$  data in the distant past are exponentially down-weighted, which facilitates tracking network anomalies in nonstationary environments. In the case of static routing ( $\mathbf{R}_t = \mathbf{R}, t = 1, 2, \dots$ ) and infinite memory ( $\beta = 1$ ), the formulation (8) coincides with the batch estimator (P3). This is the reason for the time-varying factor weighting  $\|\mathbf{P}\|_F^2$ .

##### A. Tracking Network Anomalies

Towards deriving a real-time, computationally efficient, and recursive solver of (8), an alternating minimization method is adopted in which iteration  $k$  coincides 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.B. Per time instant  $t$ , a new datum  $\{\mathcal{P}_{\Omega_t}(\mathbf{y}_t), \Omega_t\}$  is drawn and  $\{\mathbf{q}_t, \mathbf{a}_t\}$  are jointly estimated via

$$\{\mathbf{q}[t], \mathbf{a}[t]\} = \arg \min_{\{\mathbf{q}, \mathbf{a}\}} \left[ \frac{1}{2} \|\mathcal{P}_{\Omega_t}(\mathbf{y}_t - \mathbf{P}[t-1]\mathbf{q} - \mathbf{R}_t\mathbf{a})\|_2^2 + \frac{\lambda_*}{2} \|\mathbf{q}\|_2^2 + \lambda_1 \|\mathbf{a}\|_1 \right]. \quad (9)$$

It turns out that (9) can be efficiently solved. Fixing  $\mathbf{a}$  to carry out the minimization with respect to  $\mathbf{q}$  first, one is left with an  $\ell_2$ -norm regularized LS (ridge-regression) problem

$$\begin{aligned} \mathbf{q}[t] &= \arg \min_{\mathbf{q}} \left[ \frac{1}{2} \|\mathcal{P}_{\Omega_t}(\mathbf{y}_t - \mathbf{P}[t-1]\mathbf{q} - \mathbf{R}_t\mathbf{a})\|_2^2 + \frac{\lambda_*}{2} \|\mathbf{q}\|_2^2 \right] \\ &= (\lambda_* \mathbf{I}_\rho + \mathbf{P}'[t-1]\Omega_t\mathbf{P}[t-1])^{-1} \mathbf{P}'[t-1] \\ &\quad \times \mathcal{P}_{\Omega_t}(\mathbf{y}_t - \mathbf{R}_t\mathbf{a}). \end{aligned} \quad (10)$$

Note that  $\mathbf{q}[t]$  is an affine function of  $\mathbf{a}$ , and the update rule for  $\mathbf{q}[t]$  is not well defined until  $\mathbf{a}$  is replaced with  $\mathbf{a}[t]$ . Towards obtaining an expression for  $\mathbf{a}[t]$ , define  $\mathbf{D}[t] := (\lambda_* \mathbf{I}_\rho + \mathbf{P}'[t-1]\Omega_t\mathbf{P}'[t-1])^{-1} \mathbf{P}'[t-1]$  for notational convenience, and substitute (10) back into (9) to arrive at the Lasso estimator

$$\mathbf{a}[t] = \arg \min_{\mathbf{a}} \left[ \frac{1}{2} \|\mathbf{F}[t](\mathbf{y}_t - \mathbf{R}_t\mathbf{a})\|_2^2 + \lambda_1 \|\mathbf{a}\|_1 \right] \quad (11)$$

where  $\mathbf{F}[t] := [\Omega_t - \Omega_t\mathbf{P}[t-1]\mathbf{D}[t]\Omega_t, \sqrt{\lambda_*}\Omega_t\mathbf{D}'[t]]'$ . The diagonal matrix  $\Omega_t$  was defined in Section III.B, see the discussion after (5).

In the second step of the alternating-minimization scheme, the updated subspace matrix  $\mathbf{P}[t]$  is obtained by minimizing (8) with respect to  $\mathbf{P}$ , while the optimization variables  $\{\mathbf{q}_\tau, \mathbf{a}_\tau\}_{\tau=1}^t$  are fixed and take the values  $\{\mathbf{q}[\tau], \mathbf{a}[\tau]\}_{\tau=1}^t$ . This yields

$$\begin{aligned} \mathbf{P}[t] &= \arg \min_{\mathbf{P}} \left[ \sum_{\tau=1}^t \beta^{t-\tau} \frac{1}{2} \|\mathcal{P}_{\Omega_\tau}(\mathbf{y}_\tau - \mathbf{P}\mathbf{q}[\tau] - \mathbf{R}_\tau\mathbf{a}[\tau])\|_2^2 \right. \\ &\quad \left. + \frac{\lambda_*}{2} \|\mathbf{P}\|_F^2 \right]. \end{aligned} \quad (12)$$

Similar to the batch case, (12) decouples over the rows of  $\mathbf{P}$  which are obtained in parallel via

$$\begin{aligned} \mathbf{p}_l[t] &= \arg \min_{\mathbf{p}} \left[ \sum_{\tau=1}^t \beta^{t-\tau} \omega_{l,\tau} \left( y_{l,\tau} - \mathbf{p}'\mathbf{q}[\tau] - (\mathbf{r}_{l,\tau}^r)' \mathbf{a}[\tau] \right)^2 \right. \\ &\quad \left. + \frac{\lambda_*}{2} \|\mathbf{p}\|^2 \right], \quad l = 1, \dots, L \end{aligned} \quad (13)$$

where  $\omega_{l,\tau}$  denotes the  $l$ -th diagonal entry of  $\Omega_\tau$ . For  $\beta = 1$ , subproblems (13) can be efficiently solved using the RLS algorithm [36]. Upon defining  $\mathbf{s}_l[t] := \sum_{\tau=1}^t \beta^{t-\tau} \omega_{l,\tau} (y_{l,\tau} - \mathbf{r}_{l,\tau}^r \mathbf{a}[\tau]) \mathbf{q}[\tau]$ ,  $\mathbf{H}_l[t] := \sum_{\tau=1}^t \beta^{t-\tau} \omega_{l,\tau} \mathbf{q}[\tau] \mathbf{q}'[\tau] + \lambda_* \mathbf{I}_\rho$ , and  $\mathbf{M}_l[t] := \mathbf{H}_l^{-1}[t]$ , with  $\beta = 1$  one simply updates

$$\begin{aligned} \mathbf{s}_l[t] &= \mathbf{s}_l[t-1] + \omega_{l,t} (y_{l,t} - \mathbf{r}_{l,t}^r \mathbf{a}[t]) \mathbf{q}[t] \\ \mathbf{M}_l[t] &= \mathbf{M}_l[t-1] - \omega_{l,t} \frac{\mathbf{M}_l[t-1] \mathbf{q}[t] \mathbf{q}'[t] \mathbf{M}_l[t-1]}{1 + \mathbf{q}'[t] \mathbf{M}_l[t-1] \mathbf{q}[t]} \end{aligned}$$

and forms  $\mathbf{p}_l[t] = \mathbf{M}_l[t] \mathbf{s}_l[t]$ , for  $l = 1, \dots, L$ .

However, for  $0 < \beta < 1$  the regularization term  $(\lambda_*/2) \|\mathbf{p}\|^2$  in (13) makes it impossible to express  $\mathbf{H}_l[t]$  in terms of  $\mathbf{H}_l[t-1]$  plus a rank-one correction. Hence, one cannot resort to the matrix inversion lemma and update  $\mathbf{M}_l[t]$  with quadratic complexity only. Based on direct inversion of  $\mathbf{H}_l[t]$ ,  $l = 1, \dots, L$ , the overall recursive algorithm for tracking network anomalies is tabulated under Algorithm 2. The per iteration cost of the  $L$  inversions (each  $\mathcal{O}(\rho^3)$ , which could be further reduced if one leverages also the symmetry of  $\mathbf{H}_l[t]$ ) is affordable for moderate number of links, because  $\rho$  is small when estimating low-rank traffic matrices. Still, for those settings where computational complexity reductions are at a premium, an online stochastic gradient descent algorithm is described in Section V.A.

---

### Algorithm 2: Online algorithm for tracking network anomalies

---

**input**  $\{\mathcal{P}_{\Omega_t}(\mathbf{y}_t), \Omega_t, \mathbf{R}_t\}_{t=1}^\infty, \beta, \lambda_*$ , and  $\lambda_1$ .

**initialize**  $\mathbf{G}_l[0] = \mathbf{0}_{\rho \times \rho}$ ,  $\mathbf{s}_l[0] = \mathbf{0}_\rho$ ,  $l = 1, \dots, L$ , and  $\mathbf{P}[0]$  at random.

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

$\mathbf{D}[t] = (\lambda_* \mathbf{I}_\rho + \mathbf{P}'[t-1]\Omega_t\mathbf{P}[t-1])^{-1} \mathbf{P}'[t-1]$ .

$\mathbf{F}[t] = [\Omega_t - \Omega_t\mathbf{P}[t-1]\mathbf{D}[t]\Omega_t, \sqrt{\lambda_*}\Omega_t\mathbf{D}'[t]]'$ .

$\mathbf{a}[t] = \arg \min_{\mathbf{a}} [(1/2) \|\mathbf{F}[t](\mathbf{y}_t - \mathbf{R}_t\mathbf{a})\|_2^2 + \lambda_1 \|\mathbf{a}\|_1]$ .

$\mathbf{q}[t] = \mathbf{D}[t]\Omega_t(\mathbf{y}_t - \mathbf{R}_t\mathbf{a}[t])$ .

$\mathbf{G}_l[t] = \beta \mathbf{G}_l[t-1] + \omega_{l,t} \mathbf{q}[t] \mathbf{q}'[t]$ ,  $l = 1, \dots, L$ .

$\mathbf{s}_l[t] = \beta \mathbf{s}_l[t-1] + \omega_{l,t} (y_{l,t} - \mathbf{r}_{l,t}^r \mathbf{a}[t]) \mathbf{q}[t]$ ,  $l = 1, \dots, L$ .

$\mathbf{p}_l[t] = (\mathbf{G}_l[t] + \lambda_* \mathbf{I}_\rho)^{-1} \mathbf{s}_l[t]$ ,  $l = 1, \dots, L$ .

**return**  $\hat{\mathbf{a}}_t := \mathbf{a}[t]$  and  $\hat{\mathbf{x}}_t := \mathbf{P}[t] \mathbf{q}[t]$ .

**end for**

---

*Remark 2 (Robust Subspace Trackers):* Algorithm 2 is closely related to timely robust subspace trackers, which aim at estimating a low-rank subspace  $\mathbf{P}$  from grossly corrupted and possibly incomplete data, namely  $\mathcal{P}_{\Omega_t}(\mathbf{y}_t) = \mathcal{P}_{\Omega_t}(\mathbf{P}\mathbf{q}_t + \mathbf{a}_t + \mathbf{v}_t)$ ,  $t = 1, 2, \dots$ . In the absence of sparse ‘outliers’  $\{\mathbf{a}_t\}_{t=1}^\infty$ , an online algorithm based on incremental gradient descent on the Grassmannian manifold of subspaces was put forth in [5]. The second-order RLS-type algorithm in [16] extends the seminal projection approximation subspace tracking algorithm [41] to handle missing data. When outliers are present, robust counterparts can be found in [15], [19], [27]. Relative to all aforementioned works, the estimation problem here is more challenging due to the presence of the fat (compression) matrix  $\mathbf{R}_t$ ; see [26] for fundamental identifiability issues related to the model (3).

### B. Convergence Analysis

This section studies the convergence of the iterates generated by Algorithm 2, for the infinite memory special case i.e., when  $\beta = 1$ . Upon defining the function

$$g_t(\mathbf{P}, \mathbf{q}, \mathbf{a}) := \frac{1}{2} \|\mathcal{P}_{\Omega_t}(\mathbf{y}_t - \mathbf{P}\mathbf{q} - \mathbf{R}_t\mathbf{a})\|_2^2 + \frac{\lambda_*}{2} \|\mathbf{q}\|_2^2 + \lambda_1 \|\mathbf{a}\|_1 \quad (14)$$

in addition to  $\ell_t(\mathbf{P}) := \min_{\{\mathbf{q}, \mathbf{a}\}} g_t(\mathbf{P}, \mathbf{q}, \mathbf{a})$ , when  $\beta = 1$  Algorithm 2 aims at minimizing the following *average* cost function at time  $t$

$$C_t(\mathbf{P}) := \frac{1}{t} \sum_{\tau=1}^t \ell_{\tau}(\mathbf{P}) + \frac{\lambda_*}{2t} \|\mathbf{P}\|_F^2. \quad (15)$$

Normalization (by  $t$ ) ensures that the cost function does not grow unbounded as time evolves. For fixed routing  $\{\mathbf{R}_{\tau} = \mathbf{R}\}_{\tau=1}^t$ , (15) is essentially identical to the batch estimator in (P3) up to a scaling, which does not affect the value of the minimizers. Note that as time evolves, minimization of  $C_t$  becomes increasingly complex computationally. Even evaluating  $C_t$  is challenging for large  $t$ , since it entails solving  $t$  Lasso problems to minimize all  $g_{\tau}$  and defining the functions  $\ell_{\tau}$ ,  $\tau = 1, \dots, T$ . Hence, at time  $t$  the subspace estimate  $\mathbf{P}[t]$  is obtained by minimizing the *approximate* cost function [cf. (12) when  $\beta = 1$ ]

$$\hat{C}_t(\mathbf{P}) = \frac{1}{t} \sum_{\tau=1}^t g_{\tau}(\mathbf{P}, \mathbf{q}[\tau], \mathbf{a}[\tau]) + \frac{\lambda_*}{2t} \|\mathbf{P}\|_F^2 \quad (16)$$

in which  $\{\mathbf{q}[t], \mathbf{a}[t]\}$  are obtained based on the prior subspace estimate  $\mathbf{P}[t-1]$  after solving [cf. (9)]

$$\{\mathbf{q}[t], \mathbf{a}[t]\} = \arg \min_{\{\mathbf{q}, \mathbf{a}\}} g_t(\mathbf{P}[t-1], \mathbf{q}, \mathbf{a}). \quad (17)$$

Obtaining  $\mathbf{q}[t]$  this way resembles the projection approximation adopted in [41], and can only be evaluated after  $\mathbf{a}[t]$  is obtained [cf. (10)]. Since  $\hat{C}_t(\mathbf{P})$  is a smooth convex function, the minimizer  $\mathbf{P}[t] = \arg \min_{\mathbf{P}} \hat{C}_t(\mathbf{P})$  is the solution of the quadratic equation  $\nabla \hat{C}_t(\mathbf{P}[t]) = \mathbf{0}_{L \times \rho}$ .

So far, it is apparent that the approximate cost function  $\hat{C}_t(\mathbf{P}[t])$  overestimates the target cost  $C_t(\mathbf{P}[t])$ , for  $t = 1, 2, \dots$ . However, it is not clear whether the dictionary iterates  $\{\mathbf{P}[t]\}_{t=1}^{\infty}$  converge, and most importantly, how well can they optimize the target cost function  $C_t$ . The good news is that  $\hat{C}_t(\mathbf{P}[t])$  asymptotically approaches  $C_t(\mathbf{P}[t])$ , and the subspace iterates null  $\nabla C_t(\mathbf{P}[t])$  as well, both as  $t \rightarrow \infty$ . The latter result is summarized in the next proposition, which is proved in the next section.

*Proposition 3:* Assume that: a1)  $\{\Omega_t\}_{t=1}^{\infty}$  and  $\{\mathbf{y}_t\}_{t=1}^{\infty}$  are independent and identically distributed (i.i.d.) random processes; a2)  $\|\mathcal{P}_{\Omega_t}(\mathbf{y}_t)\|_{\infty}$  is uniformly bounded; a3) iterates  $\{\mathbf{P}[t]\}_{t=1}^{\infty}$  are in a compact set  $\mathcal{L} \subset \mathbb{R}^{L \times \rho}$ ; a4)  $\hat{C}_t(\mathbf{P})$  is positive definite, namely  $\lambda_{\min}[\nabla^2 \hat{C}_t(\mathbf{P})] \geq c$  for some  $c > 0$ ; and a5)  $\sigma_{\min}(\mathbf{S}[t]) \geq c_0$ , where the matrix  $\mathbf{S}[t] \in \mathbb{R}^{(L+\rho) \times |\text{supp}(\mathbf{a}[t])|}$  contains the columns of  $\mathbf{F}[t]\mathbf{R}_t$  associated with the elements in  $\text{supp}(\mathbf{a}[t])$ , and  $c_0$  is a positive constant. Then  $\lim_{t \rightarrow \infty} \nabla C_t(\mathbf{P}[t]) = \mathbf{0}_{L \times \rho}$  almost surely (a.s.), which implies that the subspace iterates  $\{\mathbf{P}[t]\}_{t=1}^{\infty}$  asymptotically coincide with the stationary points of (P3) when the routing remains invariant, i.e., when  $\mathbf{R}_t = \mathbf{R}$ ,  $t = 1, 2, \dots$

To clearly delineate the scope of the analysis, it is worth commenting on the assumptions a1)–a5) 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 [35], [36]. While independence is required for tractability, a1) may be grossly violated because OD flows are

correlated across time (cf. the low-rank property of  $\mathbf{Z}$  and  $\mathbf{X}$ ). Still, in accordance with the adaptive filtering folklore e.g., [35, p. 321], as  $\beta \rightarrow 1$  the upshot of the analysis based on i.i.d. data extends accurately to the pragmatic setting whereby the link-counts and missing data patterns exhibit spatiotemporal correlations. Uniform boundedness of  $\mathcal{P}_{\Omega_t}(\mathbf{y}_t)$  [cf. a2)] is satisfied in practice, since the traffic is always limited by the (finite) capacity of the physical links. The bounded subspace requirement in a3) is a technical assumption that simplifies the arguments of the ensuing proof, and has been corroborated via extensive computer simulations including those in Section VI. It is apparent that the sampling set  $\Omega_t$  plays a key role towards ensuring that a4) and a5) are satisfied. Intuitively, if the missing entries tend to be only few and somehow uniformly distributed across links and time, they will not markedly increase coherence of the regression matrices  $\mathbf{F}[t]\mathbf{R}_t$ , and thus compromise the uniqueness of the Lasso solutions. This also increases the likelihood that  $\nabla^2 \hat{C}_t(\mathbf{P}) = (\lambda_*)/(t)\mathbf{I}_{L\rho} + (1/t)\sum_{\tau=1}^t (\mathbf{q}[\tau]\mathbf{q}'[\tau]) \otimes \mathbf{\Omega}_{\tau} \succeq c\mathbf{I}_{L\rho}$  holds. As argued in [24], if needed one could incorporate additional regularization terms in the cost function to enforce a4) and a5). Before moving on to the proof, a remark is in order.

*Remark 3 (Performance Guarantees):* In line with Proposition 2, 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 have been well documented e.g., in [9], [26], [44]. Specifically, given the learned traffic subspace  $\bar{\mathbf{P}}$  and the corresponding  $\bar{\mathbf{Q}}$  and  $\bar{\mathbf{A}}$  [obtained via (9)] over a time window of size  $T$ , is  $\{\hat{\mathbf{X}} := \bar{\mathbf{P}}\bar{\mathbf{Q}}', \hat{\mathbf{A}} := \bar{\mathbf{A}}\}$  an optimal solution of (P1) when  $T \rightarrow \infty$ ? This in turn requires asymptotic analysis of the optimality conditions for (P1), and is left for future research. Nevertheless, empirically the online estimator attains the performance of (P1), as evidenced by the numerical tests in Section VI.

### C. Proof of Proposition 3

The main steps of the proof are inspired by [24], which studies convergence of an online dictionary learning algorithm using the theory of martingale sequences; see e.g., [23]. However, relative to [24] the problem here introduces several distinct elements including: i) missing data with a time-varying pattern  $\Omega_t$ ; ii) a non-convex bilinear term where the tall subspace matrix  $\mathbf{P}$  plays a role similar to the fat dictionary in [24], but the multiplicative projection coefficients here are not sparse; and iii) the additional bilinear terms  $\mathbf{R}_t \mathbf{a}_t$  which entail sparse coding of  $\mathbf{a}_t$  as in [24], but with a known regression (routing) matrix. Hence, convergence analysis becomes more challenging and demands, in part, for a new treatment. Accordingly, in the sequel emphasis will be placed on the novel aspects specific to the problem at hand.

The basic structure of the proof consists of three preliminary lemmata, which are subsequently used to establish that  $\lim_{t \rightarrow \infty} \nabla C_t(\mathbf{P}[t]) = \mathbf{0}_{L \times \rho}$  a.s. through a simple argument. The first lemma deals with regularity properties of functions  $\hat{C}_t$  and  $C_t$ , which will come handy later on; see Appendix B for a proof.

*Lemma 1:* If a2) and a5) hold, then the functions: i)  $\{\mathbf{a}_t(\mathbf{P}), \mathbf{q}_t(\mathbf{P})\} = \arg \min_{\{\mathbf{q}, \mathbf{a}\}} g_t(\mathbf{P}, \mathbf{q}, \mathbf{a})$ , ii)  $g_t(\mathbf{P}, \mathbf{q}[t], \mathbf{a}[t])$ , iii)  $\ell_t(\mathbf{P})$ , and iv)  $\nabla \ell_t(\mathbf{P})$  are Lipschitz continuous for  $\mathbf{P} \in \mathcal{L}$  ( $\mathcal{L}$  is a compact set), with constants independent of  $t$ .

The next lemma (proved in Appendix C) asserts that the distance between two subsequent traffic subspace estimates vanishes as  $t \rightarrow \infty$ , a property that will be instrumental later on when establishing that  $\hat{C}_t(\mathbf{P}[t]) - C_t(\mathbf{P}[t]) \rightarrow 0$  a.s.

*Lemma 2:* If a2)–a5) hold, then  $\|\mathbf{P}[t+1] - \mathbf{P}[t]\|_F = \mathcal{O}(1/t)$ . The previous lemma by no means implies that the subspace iterates converge, which is a much more ambitious objective that may not even hold under the current assumptions. The final lemma however, asserts that the cost sequence indeed converges with probability one; see Appendix D for a proof.

*Lemma 3:* If a1)–a5) hold, then  $\hat{C}_t(\mathbf{P}[t])$  converges a.s. Moreover,  $\hat{C}_t(\mathbf{P}[t]) - C_t(\mathbf{P}[t]) \rightarrow 0$  a.s.

Putting the pieces together, in the sequel it is shown that the sequence  $\{\nabla \hat{C}_t(\mathbf{P}[t]) - \nabla C_t(\mathbf{P}[t])\}_{t=1}^{\infty}$  converges a.s. to zero, and since  $\nabla \hat{C}_t(\mathbf{P}[t]) = \mathbf{0}_{L \times \rho}$  by algorithmic construction, the subspace iterates  $\{\mathbf{P}[t]\}_{t=1}^{\infty}$  coincide with the stationary points of the target cost function  $C_t$ . To this end, it suffices to prove that every convergent *subsequence* nulls the gradient  $\nabla C_t$  asymptotically, which in turn implies that the entire sequence converges to the set of stationary points of the batch problem (P3).

Since  $\mathcal{L}$  is compact by virtue of a3), one can always pick a convergent subsequence  $\{\mathbf{P}[t]\}_{t=1}^{\infty}$  whose limit point is  $\mathbf{P}^*$ , say<sup>2</sup>. Consider the positive-valued decreasing sequence  $\{\alpha_t\}_{t=1}^{\infty}$  that converges to zero slower than  $\hat{C}_t(\mathbf{P}[t]) - C_t(\mathbf{P}[t])$  does, and recall that  $\hat{C}_t(\mathbf{P}[t] + \alpha_t \mathbf{U}) \geq C_t(\mathbf{P}[t] + \alpha_t \mathbf{U})$  for any  $\mathbf{U} \in \mathbb{R}^{L \times \rho}$ . From the mean-value theorem and for arbitrary  $\mathbf{U}$ , expanding both sides of the inequality around the point  $\mathbf{P}[t]$  one arrives at

$$\begin{aligned} & \hat{C}_t(\mathbf{P}[t]) + \alpha_t \text{tr}\{\mathbf{U}' \nabla \hat{C}_t(\mathbf{P}[t])\} + \alpha_t \text{tr}\{\mathbf{U}' (\nabla \hat{C}_t(\Theta_1[t]) \\ & \quad - \nabla \hat{C}_t(\mathbf{P}[t]))\} \geq C_t(\mathbf{P}[t]) + \alpha_t \text{tr}\{\mathbf{U}' \nabla C_t(\mathbf{P}[t])\} \\ & \quad + \alpha_t \text{tr}\{\mathbf{U}' (\nabla C_t(\Theta_2[t]) - \nabla C_t(\mathbf{P}[t]))\} \end{aligned}$$

for some  $\Theta_1[t], \Theta_2[t] \in \mathbb{R}^{L \times \rho}$  on the line segment connecting  $\mathbf{P}[t]$  and  $\mathbf{P}[t] + \alpha_t \mathbf{U}$ . Taking limit as  $t \rightarrow \infty$  and applying Lemma 3 it follows that

$$\begin{aligned} & \lim_{t \rightarrow \infty} \text{tr}\{\mathbf{U}' (\nabla \hat{C}_t(\mathbf{P}[t]) - \nabla C_t(\mathbf{P}[t]))\} \\ & \quad + \lim_{t \rightarrow \infty} \text{tr}\{\mathbf{U}' (\nabla \hat{C}_t(\Theta_1[t]) - \nabla \hat{C}_t(\mathbf{P}[t]))\} \\ & \quad + \lim_{t \rightarrow \infty} \text{tr}\{\mathbf{U}' (\nabla C_t(\mathbf{P}[t]) - \nabla C_t(\Theta_2[t]))\} \geq 0, \text{ a.s.} \quad (18) \end{aligned}$$

For the quadratic function  $\hat{C}_t$ , uniform boundedness of the Hessian  $\nabla^2 \hat{C}_t(\mathbf{P}) = (\lambda_*)/(t) \mathbf{I}_{L\rho} + (1/t) \sum_{\tau=1}^t (\mathbf{q}[\tau] \mathbf{q}'[\tau]) \otimes \mathbf{\Omega}_\tau$  implies that  $\nabla \hat{C}_t$  is Lipschitz. Furthermore, since  $\nabla l_\tau$  is Lipschitz as per Lemma 1,  $\nabla C_t$  is Lipschitz as well. Consequently, according to the Cauchy-Schwarz inequality

$$\begin{aligned} & |\text{tr}\{\mathbf{U}' (\nabla C_t(\mathbf{P}[t]) - \nabla C_t(\Theta_2[t]))\}| \\ & \quad \leq \|\mathbf{U}\|_F \|\nabla C_t(\mathbf{P}[t]) - \nabla C_t(\Theta_2[t])\|_F \\ & \quad \leq c \|\mathbf{U}\|_F \|\mathbf{P}[t] - \Theta_2[t]\|_F \stackrel{(a)}{\leq} c \alpha_t \|\mathbf{U}\|_F^2 \quad (19) \end{aligned}$$

for some constant  $c > 0$ , where (a) holds since  $\Theta_2[t]$  is a convex combination of  $\mathbf{P}[t]$  and  $\mathbf{P}[t] + \alpha_t \mathbf{U}$ . Likewise, one can bound

<sup>2</sup>Formally, the subsequence should be denoted as  $\{\mathbf{P}[t(i)]\}_{i=1}^{\infty}$ , but a slight abuse of notation is allowed for simplicity.

the second term on the left-hand-side of (18). Accordingly, it holds that

$$\begin{aligned} & \lim_{t \rightarrow \infty} \text{tr}\{\mathbf{U}' (\nabla C_t(\mathbf{P}[t]) - \nabla C_t(\Theta_2[t]))\} \\ & \quad = \lim_{t \rightarrow \infty} \text{tr}\{\mathbf{U}' (\nabla \hat{C}_t(\mathbf{P}[t]) - \nabla \hat{C}_t(\Theta_1[t]))\} = 0. \end{aligned}$$

All in all, the second and third terms in (18) vanish and one is left with

$$\lim_{t \rightarrow \infty} \text{tr}\{\mathbf{U}' (\nabla \hat{C}_t(\mathbf{P}_t) - \nabla C_t(\mathbf{P}_t))\} \geq 0. \quad (20)$$

Because  $\mathbf{U} \in \mathbb{R}^{L \times \rho}$  is arbitrary, (20) can only hold if  $\lim_{t \rightarrow \infty} (\nabla \hat{C}_t(\mathbf{P}[t]) - \nabla C_t(\mathbf{P}[t])) = \mathbf{0}_{L \times \rho}$  a.s., which completes the proof. ■

## V. FURTHER ALGORITHMIC ISSUES

For completeness, this section outlines a couple of additional algorithmic aspects relevant to anomaly detection in *large-scale* networks. Firstly, a lightweight first-order algorithm is developed as an alternative to Algorithm 2, which relies on fast Nesterov-type gradient updates for the traffic subspace. Secondly, the possibility of developing distributed algorithms for dynamic anomalography is discussed.

### A. Fast Stochastic-Gradient Algorithm

Reduction of the computational complexity in updating the traffic subspace  $\mathbf{P}$  is the subject of this section. The basic alternating minimization framework in Section IV.A will be retained, and the updates for  $\{\mathbf{q}[t], \mathbf{a}[t]\}$  will be identical to those tabulated under Algorithm 2. However, instead of solving an unconstrained quadratic program per iteration to obtain  $\mathbf{P}[t]$  [cf. (12)], the refinements to the subspace estimate will be given by a (stochastic) gradient algorithm.

As discussed in Section IV.B, in Algorithm 2 the subspace estimate  $\mathbf{P}[t]$  is obtained by minimizing the empirical cost function  $\hat{C}_t(\mathbf{P}) = (1/t) \sum_{\tau=1}^t f_\tau(\mathbf{P})$ , where

$$\begin{aligned} f_t(\mathbf{P}) := & \frac{1}{2} \|\mathbf{\Omega}_t(\mathbf{y}_t - \mathbf{P}\mathbf{q}[t] - \mathbf{R}_t \mathbf{a}[t])\|_2^2 + \frac{\lambda_*}{2t} \|\mathbf{P}\|_F^2 \\ & + \frac{\lambda_*}{2} \|\mathbf{q}[t]\|_2^2 + \lambda_1 \|\mathbf{a}[t]\|_1, \quad t = 1, 2, \dots \quad (21) \end{aligned}$$

By the law of large numbers, if data  $\{\mathcal{P}_{\Omega_t}(\mathbf{y}_t)\}_{t=1}^{\infty}$  are stationary, solving  $\min_{\mathbf{P}} \lim_{t \rightarrow \infty} \hat{C}_t(\mathbf{P})$  yields the desired minimizer of the *expected* cost  $\mathbb{E}[C_t(\mathbf{P})]$ , 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 nominal traffic subspace via a stochastic gradient iteration [36]

$$\begin{aligned} \mathbf{P}[t] & = \arg \min_{\mathbf{P}} Q_{(1/\tilde{\mu}[t]),t}(\mathbf{P}, \mathbf{P}[t-1]) \\ & = \mathbf{P}[t-1] - \tilde{\mu}[t] \nabla f_t(\mathbf{P}[t-1]) \quad (22) \end{aligned}$$

where  $\tilde{\mu}[t]$  is a stepsize,  $Q_{\mu,t}(\mathbf{P}_1, \mathbf{P}_2) := f_t(\mathbf{P}_2) + \langle \mathbf{P}_1 - \mathbf{P}_2, \nabla f_t(\mathbf{P}_2) \rangle + (\mu/2) \|\mathbf{P}_1 - \mathbf{P}_2\|_F^2$ , and  $\nabla f_t(\mathbf{P}) = -\mathbf{\Omega}_t(\mathbf{y}_t -$

$\mathbf{P}\mathbf{q}[t] - \mathbf{R}_t\mathbf{a}[t]\mathbf{q}'[t] + (\lambda_*/t)\mathbf{P}$ . In the context of adaptive filtering, stochastic gradient algorithms such as (21) are known to converge typically slower than RLS. This is expected since RLS can be shown to be an instance of Newton's (second-order) optimization method [36].

Building on the increasingly popular *accelerated* gradient methods for (batch) smooth optimization [6], [30], the idea here is to speed-up the learning rate of the estimated traffic subspace (22), 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  $\mathbf{P}[t] = \tilde{\mathbf{P}}[t] - \tilde{\mu}[t]\nabla f_t(\tilde{\mathbf{P}}[t])$ , which relies on a judicious linear combination  $\tilde{\mathbf{P}}[t-1]$  of the previous pair of iterates  $\{\mathbf{P}[t-1], \mathbf{P}[t-2]\}$ . Specifically, the choice  $\tilde{\mathbf{P}}[t] = \mathbf{P}[t-1] + (k[t]-1)/(k[t])(\mathbf{P}[t-1] - \mathbf{P}[t-2])$ , 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., [6] and references therein. Using this acceleration technique in conjunction with a backtracking stepsize rule [7], a fast online stochastic gradient algorithm for unveiling network anomalies is tabulated under Algorithm 3. Different from Algorithm 2, no matrix inversions are involved in the update of the traffic subspace  $\mathbf{P}[t]$ . Clearly, a standard (non accelerated) stochastic gradient descent algorithm with backtracking stepsize rule is subsumed as a special case, when  $k[t] = 1$ ,  $t = 0, 1, 2, \dots$

---

**Algorithm 3: Online stochastic gradient algorithm for unveiling network anomalies**

---

**input**  $\{\mathbf{y}_t, \mathbf{R}_t, \boldsymbol{\Omega}_t\}_{t=1}^\infty$ ,  $\rho, \lambda_*, \lambda_1, \eta > 1$ .  
**initialize**  $\mathbf{P}[0]$  at random,  $\mu[0] > 0$ ,  $\tilde{\mathbf{P}}[1] := \mathbf{P}[0]$ , and  $k[1] := 1$ .  
**for**  $t = 1, 2, \dots$  **do**  
 $\mathbf{D}[t] = (\lambda_*\mathbf{I}_\rho + \mathbf{P}'[t-1]\boldsymbol{\Omega}_t\mathbf{P}[t-1])^{-1}\mathbf{P}'[t-1]$   
 $\mathbf{F}'[t] := [\boldsymbol{\Omega}_t - \boldsymbol{\Omega}_t\mathbf{P}[t-1]\mathbf{D}[t]\boldsymbol{\Omega}_t, \sqrt{\lambda_*}\boldsymbol{\Omega}_t\mathbf{D}'[t]]$   
 $\mathbf{a}[t] = \arg \min_{\mathbf{a}} [(1/2)\|\mathbf{F}[t](\mathbf{y}_t - \mathbf{R}_t\mathbf{a})\|^2 + \lambda_1\|\mathbf{a}\|_1]$   
 $\mathbf{q}[t] = \mathbf{D}[t]\boldsymbol{\Omega}_t(\mathbf{y}_t - \mathbf{R}_t\mathbf{a}_t)$   
 Find the smallest nonnegative integer  $i[t]$  such that with  $\bar{\mu} := \eta^{i[t]}\mu[t-1]$   
 $f_t(\tilde{\mathbf{P}}[t] - (1/\bar{\mu})\nabla f_t(\tilde{\mathbf{P}}[t])) \leq Q_{\bar{\mu}, t}(\tilde{\mathbf{P}}[t] - (1/\bar{\mu})\nabla f_t(\tilde{\mathbf{P}}[t]), \tilde{\mathbf{P}}[t])$   
 holds, and set  $\mu[t] = \eta^{i[t]}\mu[t-1]$ .  
 $\mathbf{P}[t] = \tilde{\mathbf{P}}[t] - (1/\mu[t])\nabla f_t(\tilde{\mathbf{P}}[t])$ .  
 $k[t+1] = (1 + \sqrt{1 + 4k^2[t]})/(2)$ .  
 $\tilde{\mathbf{P}}[t+1] = \mathbf{P}[t] + ((k[t]-1)/(k[t+1]))(\mathbf{P}[t] - \mathbf{P}[t-1])$ .  
**end for**  
**return**  $\hat{\mathbf{x}}[t] := \mathbf{P}[t]\mathbf{q}[t]$ ,  $\hat{\mathbf{a}}[t] := \mathbf{a}[t]$ .

---

Convergence analysis of Algorithm 3 is beyond the scope of this paper, and will only be corroborated using computer simulations in Section VI. It is worth pointing out that since a non-diminishing stepsize is adopted, asymptotically the iterates generated by Algorithm 3 will hover inside a ball centered at the

minimizer of the expected cost, with radius proportional to the step size.

### B. In-Network Anomaly Trackers

Implementing Algorithms 1–3 presumes that network nodes continuously communicate their local link traffic measurements to a central monitoring station, which uses their aggregation in  $\{\mathcal{P}_{\Omega_t}(\mathbf{y}_t)\}_{t=1}^\infty$  to unveil network anomalies. While for the most part this is the prevailing operational paradigm adopted in current network technologies, it is fair to say there are limitations associated with this architecture. For instance, collecting all this information centrally may lead to excessive protocol overhead, especially when the rate of data acquisition is high at the routers. Moreover, minimizing the exchanges of raw measurements may be desirable to reduce unavoidable communication errors that translate to missing data. Performing the optimization in a centralized fashion raises robustness concerns as well, since the central monitoring station represents an isolated point of failure.

These reasons motivate devising *fully-distributed* iterative algorithms for dynamic anomalography in large-scale networks, embedding the network anomaly detection functionality to the routers. In a nutshell, per iteration nodes carry out simple computational tasks locally, relying on their own link count measurements (a few entries of the network-wide vector  $\mathbf{y}_t$  corresponding to the router links). Subsequently, local estimates are refined after exchanging messages only with directly connected neighbors, which facilitates percolation of local information to the whole network. The end goal is for network nodes to consent on a global map of network anomalies, and attain (or at least come close to) the estimation performance of the centralized counterpart which has all data  $\{\mathcal{P}_{\Omega_t}(\mathbf{y}_t)\}_{t=1}^\infty$  available.

Relying on the alternating-directions method of multipliers as the basic tool to carry out distributed optimization, a general framework for in-network sparsity-regularized rank minimization was put forth in a companion paper [25]. In the context of network anomaly detection, results therein are encouraging yet there is ample room for improvement and immediate venues for future research open up. For instance, the distributed algorithms of [25] can only tackle the batch formulation (P3), so extensions to a dynamic network setting, e.g., building on the ideas here to devise distributed anomaly trackers seems natural. To obtain desirable tradeoffs in terms of computational complexity and speed of convergence, developing and studying algorithms for distributed optimization based on Nesterov's acceleration techniques emerges as an exciting and rather pristine research direction; see [20] for early work dealing with separable batch optimization.

## VI. PERFORMANCE TESTS

Performance of the proposed batch and online estimators is assessed in this section via computer simulations using both synthetic and real network data.

*Selection of Tuning Parameters:* In the batch case,  $\lambda_1$  and  $\lambda_*$  are tuned to optimize the relative error  $\|\hat{\mathbf{A}} - \mathbf{A}_0\|_F / \|\mathbf{A}_0\|_F$ , with  $\mathbf{A}_0$  and  $\hat{\mathbf{A}}$  denoting the true and estimated anomaly matrices, respectively. In particular, one needs to perform a grid search over the bounded two-dimensional region  $\mathcal{R} := \{(\lambda_1, \lambda_*) : \lambda_1 \in (0, \|\mathbf{R}'\mathcal{P}_{\Omega}(\mathbf{Y})\|_\infty), \lambda_* \in (0, \|\mathcal{P}_{\Omega}(\mathbf{Y})\|)\}$ .

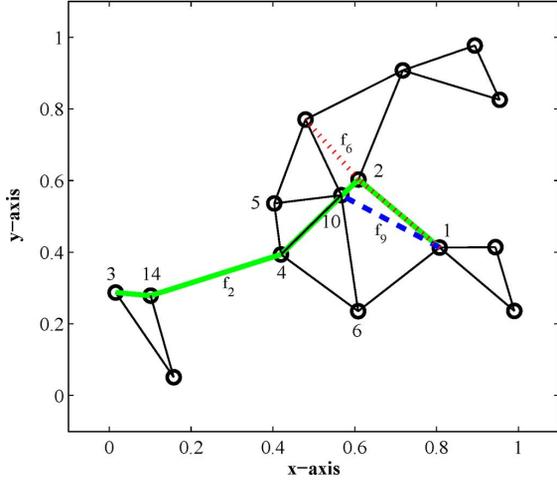


Fig. 2. Synthetic network topology graph, and the paths used for routing three flows.

The corresponding bounds are derived from the optimality conditions for (P1), which indicate that for  $(\lambda_1, \lambda_*) \in \mathcal{R}^c$  the optimal solution is  $\{\mathbf{0}_{L \times T}, \mathbf{0}_{F \times T}\}$ . Practical rules that do not require knowledge of  $\mathbf{A}_0$  can be devised along the lines of [3] and [10]. Supposing that the true values are zero, choosing  $\lambda_1 > \|\mathbf{R}'\mathcal{P}_\Omega(\mathbf{V})\|_\infty$  and  $\lambda_* > \|\mathcal{P}_\Omega(\mathbf{V})\|$  the estimator (P1) outputs  $\{\hat{\mathbf{X}} = \mathbf{0}_{L \times T}, \hat{\mathbf{A}} = \mathbf{0}_{F \times T}\}$ . This mitigates noise, but it may overshrink the true values. To avoid overshrinking, these parameters can be chosen close to their corresponding lower bounds, e.g., pick  $\lambda_* = \|\mathcal{P}_\Omega(\mathbf{V})\|$  and  $\lambda_1 = \|\mathbf{R}'\mathcal{P}_\Omega(\mathbf{V})\|_\infty$ . One can further simplify the candidate parameters by making the following reasonable assumptions: i) Gaussian noise  $v_{l,t} \sim \mathcal{N}(0, \sigma^2)$ , ii) uniform sampling with each entry of  $\Omega$  chosen independently with probability  $\pi$ , and iii) large dimensions  $F, T \rightarrow \infty$ . It is then known that  $(\sqrt{F} + \sqrt{T})^{-1} \|\mathcal{P}_\Omega(\mathbf{V})\| \rightarrow \sqrt{\pi}\sigma$ , almost surely, see e.g., [10], and thus one can pick  $\lambda_* = (\sqrt{F} + \sqrt{T})\sqrt{\pi}\sigma$ . Also, large-deviation tail bounding implies that  $\|\mathbf{R}'\mathcal{P}_\Omega(\mathbf{V})\|_\infty \leq 4\sigma \max_i \|\mathbf{R}\mathbf{e}_i\|_2 \log(FT)$  with high probability, which suggests selecting  $\lambda_1 = \sigma \max_i \|\mathbf{R}\mathbf{e}_i\|_2 \log(FT)$ . The said regularization parameters can also be used for online processing (upon setting  $T = t$ ), where they naturally increase as time evolves.

### A. Synthetic Network Data Tests

*Synthetic Network Example:* A network of  $N = 15$  nodes is considered as a realization of the random geometric graph model with agents randomly placed on the unit square, and two agents link if their Euclidean distance is less than a prescribed communication range of  $d_c = 0.35$ ; see Fig. 2. The network graph is bidirectional and comprises  $L = 52$  links, and  $F = N(N - 1) = 210$  OD flows. For each candidate OD pair, minimum hop count routing is considered to form the routing matrix  $\mathbf{R}$ . Entries of  $\mathbf{v}_t$  are i.i.d., zero-mean, Gaussian with variance  $\sigma^2$ ; i.e.,  $v_{l,t} \sim \mathcal{N}(0, \sigma^2)$ . Flow-traffic vectors  $\mathbf{z}_t$  are generated from the low-dimensional subspace  $\mathbf{U} \in \mathbb{R}^{F \times r}$  with i.i.d. entries  $u_{f,i} \sim \mathcal{N}(0, 1/F)$ , and projection coefficients  $w_{i,t} \sim \mathcal{N}(0, 1)$  such that  $\mathbf{z}_t = \mathbf{U}\mathbf{w}_t$ . Every entry of  $\mathbf{a}_t$  is randomly drawn from the set  $\{-1, 0, 1\}$ , with  $\Pr(a_{f,t} = -1) = \Pr(a_{f,t} = 1) = p/2$ . Entries of  $\mathbf{Y}$  are sampled uniformly at

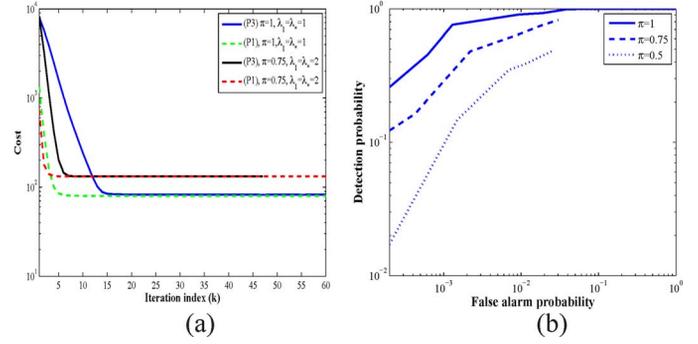


Fig. 3. Performance of the batch estimator (P3) for  $p = 0.005$  and different amounts of missing data. (a) Cost of the estimators (P1) and (P3) versus iteration index when  $\sigma = 10^{-2}$ . (b) ROC curves when  $\sigma = 10^{-1}$ .

random with probability  $\pi$  to form the diagonal sampling matrix  $\Omega_t$ . The observations at time instant  $t$  are generated according to  $\mathcal{P}_{\Omega_t}(\mathbf{y}_t) = \Omega_t(\mathbf{R}\mathbf{z}_t + \mathbf{R}\mathbf{a}_t + \mathbf{v}_t)$ . Unless otherwise stated,  $r = 2$ ,  $\rho = 5$ , and  $\beta = 0.99$  are used throughout. Different values of  $\sigma$ ,  $p$  and  $\pi$  are tested.

*Performance of the Batch Estimator:* To demonstrate the merits of the batch BCD algorithm for unveiling network anomalies (Algorithm 1), simulated data are generated for a time interval of size  $T = 100$ . For validation purposes, the benchmark estimator (P1) is iteratively solved by alternating minimization over  $\mathbf{A}$  (which corresponds to Lasso) and  $\mathbf{X}$ . The minimizations with respect to  $\mathbf{X}$  can be carried out using the iterative singular-value thresholding (SVT) algorithm [8]. Note that with full data, SVT requires only a single SVD computation. In the presence of missing data however, the SVT algorithm may require several SVD computations until convergence, rendering the said algorithm prohibitively complex for large-scale problems. In contrast, Algorithm 1 only requires simple  $\rho \times \rho$  inversions. Fig. 3(a) depicts the convergence of the respective algorithms used to solve (P1) and (P3), for different amounts of missing data (controlled by  $\pi$ ). It is apparent that both estimators attain identical performance after a few tens of iterations, as asserted by Proposition 1. To corroborate the effectiveness of Algorithm 1 in unveiling network anomalies across flows and time, the ROC curves are plotted for various percentages of missing link observations in Fig. 3(b) when  $\sigma = 10^{-1}$ . To discard spurious estimates, the hypothesis test  $\hat{a}_{f,t} \stackrel{\mathcal{H}_1}{\geq} 0.1 \stackrel{\mathcal{H}_0}{\leq}$  is considered, with anomalous and anomaly-free hypotheses  $\mathcal{H}_1$  and  $\mathcal{H}_0$ , respectively. Apparently, an inferior detection performance is expected as the percentage of missing data increases. Note that when link observations are missing ( $\pi < 1$ ), some flows may not be identifiable because they may traverse none of the observed links. For such flows, the anomalous traffic is assumed zero. Hence, as it is seen in Fig. 3(b), the maximum achievable detection probability equals the fraction of (partially) observed flows. For the instances of  $(P_{FA} = 0.021, P_D = 0.96)$  and  $(P_{FA} = 0.016, P_D = 0.69)$  corresponding to  $\pi = 1$  and  $\pi = 0.75$ , respectively, Fig. 4 depicts the magnitude of the true and estimated anomalies.

*Performance of the Online Algorithms:* To confirm the convergence and effectiveness of the online Algorithms 2 and 3, simulation tests are carried out for infinite memory  $\beta = 1$  and invariant routing matrix  $\mathbf{R}$ . Fig. 5(a) depicts the evolutions of

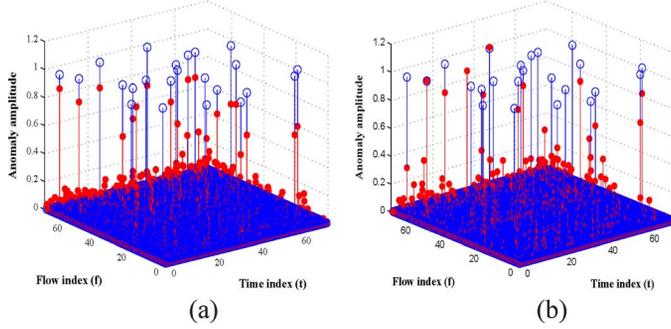


Fig. 4. Amplitude of the true (blue) and estimated (red) anomalies for  $\sigma = 10^{-1}$ . (a)  $\pi = 1$  (no missing data),  $P_{FA} = 0.021$  and  $P_D = 0.96$ . (b)  $\pi = 0.75$ ,  $P_{FA} = 0.016$  and  $P_D = 0.69$ .

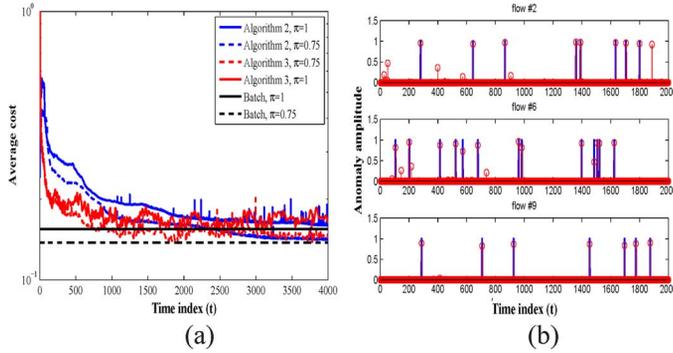


Fig. 5. Performance of the online estimator for  $\sigma = 10^{-2}$ ,  $p = 0.005$ ,  $\lambda_1 = 0.11$ , and  $\lambda_* = 0.36$ . (a) Evolution of the average cost  $C_t(\mathbf{P}_t)$  of the online algorithms versus the batch counterpart (P3). (b) Amplitude of true (solid) and estimated (circle markers) anomalies via the online Algorithm 2, for three representative flows when  $\pi = 1$  (no missing data).

the average cost  $C_t(\mathbf{P}_t)$  in (15) for different amounts of missing data  $\pi = 0.75, 1$  when the noise level is  $\sigma = 10^{-2}$ . It is evident that for both online algorithms the average cost converges (possibly within a ball) to its batch counterpart in (P3) normalized by the window size  $T = t$ . Impressively, this observation together with the one in Fig. 3(a) corroborate that the online estimators can attain the performance of the benchmark estimator, whose stable/exact recovery performance is well documented e.g., in [11], [26], [44]. It is further observed that the more data are missing, the more time it takes to learn the low-rank nominal traffic subspace, which in turn slows down convergence.

To examine the tracking capability of the online estimators, Fig. 5(b) depicts the estimated versus true anomalies over time as Algorithm 2 evolves for three representative flows indicated on Fig. 2, namely  $f_2, f_6, f_9$  corresponding to the  $i = 2, 6, 9$ -th rows of  $\mathbf{A}_0$ . Setting the detection threshold to the value 0.1 as before, for the flows  $f_2, f_6, f_9$  Algorithm 2 attains detection rate  $P_D = 0.83, 1, 1$  at false alarm rate  $P_{FA} = 0.0171, 0.0040, 0.0081$ , respectively. As expected, more false alarms are declared at early iterations as the low-rank subspace has not been learnt accurately. Upon learning the subspace performance improves and almost all anomalies are identified. Careful inspection of Fig. 5(b) reveals that the anomalies for  $f_9$  are better identified visually than those for  $f_2$ . As shown in Fig. 2,  $f_2$  is carried over links (1, 2), (2, 4), (4, 14), (14, 3) each one carrying 33, 31, 35, 22 additional flows, respectively, whereas  $f_9$  is aggregated over link (1, 3) with only

2 additional flows. Hence, identifying  $f_2$ 's anomalies from the highly-superimposed load of links (1, 2), (2, 4), (4, 14), (14, 3) is a more challenging task relative to link (1, 3). This simple example manifests the fact that the detection performance strongly depends on the network topology and the routing policy implemented, which determine the routing matrix. In accordance with [26], the coherence of sparse column subsets of the routing matrix plays an important role in identifying the anomalies. In essence, the more incoherent the column subsets of  $\mathbf{R}$  are, the better recovery performance one can attain. An intriguing question left here to address in future research pertains to desirable network topologies giving rise to incoherent routing matrices.

*Tracking Routing Changes:* The measurement model in (7) has two time-varying attributes which challenge the identification of anomalies. The first one is missing measurement data arising from e.g., packet losses during the data collection process, and the second one pertains to routing changes due to e.g., network congestion or link failures. It is thus important to test whether the proposed online algorithm succeeds in tracking these changes. As discussed earlier, missing data are sampled uniformly at random. To assess the impact of routing changes on the recovery performance, a simple probabilistic model is adopted where each time instant a single link fails, or returns to the operational state. Let  $\Phi$  denote the adjacency matrix of the network graph  $G$ , where  $[\Phi]_{i,j} = 1$  if there exists a physical link joining nodes  $i$  and  $j$ , and zero otherwise. Similarly, the active links involved in routing the data at time  $t$  are represented by the effective adjacency matrix  $\Phi_t^{\text{eff}}$ . At time instant  $t + 1$ , a biased coin is tossed with small success probability  $\alpha$ , and one of the links, say  $(i, j) \in \Phi_t^{\text{eff}}$ , is chosen uniformly at random and removed from  $G$  while ensuring that the network remains connected. Likewise, an edge  $(\ell, k) \in \Phi \setminus \Phi_t^{\text{eff}}$  is added with the same probability  $\alpha$ . The resulting adjacency matrix is then  $\Phi_{t+1}^{\text{eff}} = \Phi_t^{\text{eff}} + \mathbb{1}_{\{b_{1,t}\}} e_\ell e_k' - \mathbb{1}_{\{b_{1,t}\}} e_i e_j'$ , where the indicator function  $\mathbb{1}_{\{x \in \mathcal{X}\}}$  equals one when  $x \in \mathcal{X}$ , and zero otherwise; and  $b_{1,t}, b_{2,t} \sim \text{Ber}(\alpha)$  are i.i.d. Bernoulli random variables. The minimum hop-count algorithm is then applied to  $\Phi_{t+1}^{\text{eff}}$ , to update the routing matrix  $\mathbf{R}_{t+1}$ . Note that  $\mathbf{R}_{t+1} = \mathbf{R}_t$  with probability  $(1 - \alpha)^2$ .

The performance is tested here for fast and slowly varying routing corresponding to  $\alpha = 0.1$  and  $\alpha = 0.01$ , respectively, when  $\beta = 0.9$ . A metric of interest is the average square error in estimating the anomalies, namely  $e_t^a := (1/t) \sum_{i=1}^t \|\hat{\mathbf{a}}_i - \mathbf{a}_i\|_2^2$ , and the link traffic, namely  $e_t^x := (1/t) \sum_{i=1}^t \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2$ . Fig. 6(a) plots the average estimation error for various noise variances and amounts of missing data. The estimation error decreases quickly and after learning the subspace it becomes almost invariant. To evaluate the support recovery performance of the online estimator, define the average detection and false alarm rate

$$P_D := \frac{\sum_{\tau=1}^t \sum_{f=1}^F \mathbb{1}_{\{\hat{a}_{f,\tau} \geq 0.1, a_{f,\tau} \geq 0.1\}}}{\sum_{\tau=1}^t \sum_{f=1}^F \mathbb{1}_{\{a_{f,\tau} \geq 0.1\}}},$$

$$P_{FA} := \frac{\sum_{\tau=1}^t \sum_{f=1}^F \mathbb{1}_{\{\hat{a}_{f,\tau} \geq 0.1, a_{f,\tau} \leq 0.1\}}}{\sum_{\tau=1}^t \sum_{f=1}^F \mathbb{1}_{\{a_{f,\tau} \leq 0.1\}}}.$$

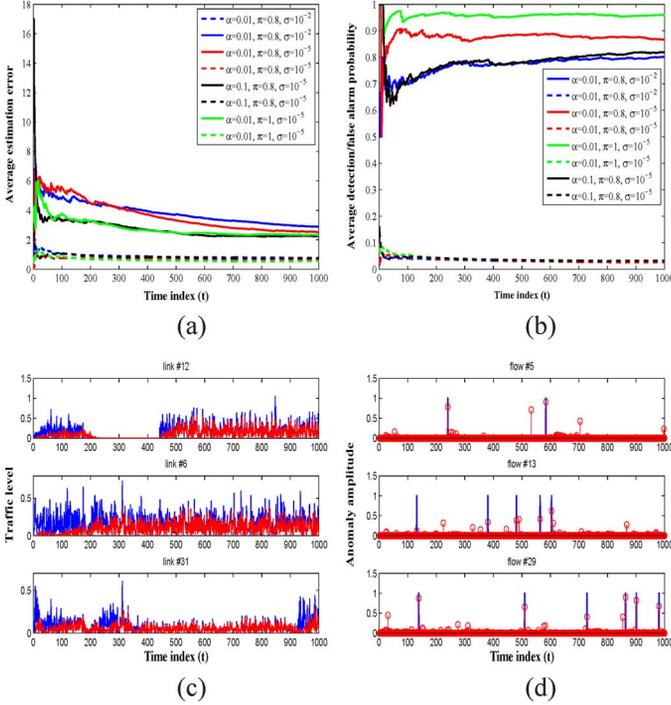


Fig. 6. Tracking routing changes for  $p = 0.005$ . (a) Evolution of average anomaly (dotted) and traffic (solid) estimation errors. (b) Evolution of average detection (solid) and false alarm (dotted) rates. (c) Estimated (red) versus true (blue) link traffic for three representative links. (d) Estimated (circle markers) versus true (solid) anomalies for three representative flows when  $\pi = 0.8$ ,  $\sigma = 10^{-5}$ , and  $\alpha = 0.01$ .

Inspecting Fig. 6(b) one observes that for  $\alpha = 0.01$  and  $\pi = 0.8$ , increasing the noise variance from  $10^{-5}$  to  $10^{-2}$  lowers the detection probability by 10%. Moreover, when  $\sigma = 10^{-5}$  and  $\alpha = 0.01$ , dropping 20% of the observations renders the estimator misdetect 11% more anomalies. The routing changes from  $\alpha = 0.01$  to  $\alpha = 0.1$  when  $\sigma = 10^{-5}$  and  $\pi = 0.8$  comes with an adverse effect of about 6% detection-rate decrease. For a few representative network links and flows Fig. 6(c) and (d) illustrate how Algorithm 2 tracks the anomalies and link-level traffic. Note that in Fig. 6(c) link 12 is dropped for the time period  $t \in [220, 420]$ , and thus the traffic level becomes zero. The flows being carried over link 31 are also varying due to routing changes, which occur at time instants  $t = 220, 940$  when the traffic is not tracked accurately.

### B. Real Network Data Tests

*Internet-2 Network Example:* Real data including OD flow traffic levels are collected from the operation of the Internet-2 network (Internet backbone network across USA) [1], shown in Fig. 1. Flow traffic levels are recorded every 5-minute intervals, for a three-week operational period of Internet-2 during Dec. 8–28, 2008 [1]. Internet-2 comprises  $N = 11$  nodes,  $L = 41$  links, and  $F = 121$  flows. Given the OD flow traffic measurements, the link loads in  $\mathbf{Y}$  are obtained through multiplication with the Internet-2 routing matrix, which in this case remains invariant during the three weeks of data acquisition [1]. Even though  $\mathbf{Y}$  is ‘constructed’ here from flow measurements, link loads can be typically acquired from SNMP traces [37].

The available OD flows are incomplete due to problems in the data collection process. In addition, flows can be modeled

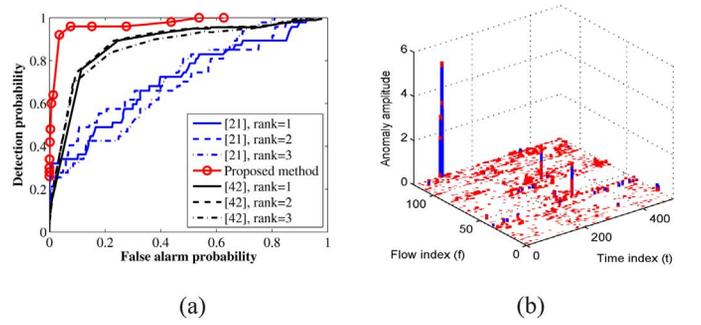


Fig. 7. Performance of the batch estimator for Internet-2 network data. (a) ROC curves of the proposed versus the PCA-based methods. (b) Amplitude of the true (blue) and estimated (red) anomalies for  $P_{FA} = 0.04$  and  $P_D = 0.93$ .

as the superposition of ‘clean’ plus anomalous traffic, i.e., the sum of some unknown ‘ground-truth’ low-rank and sparse matrices  $\mathcal{P}_\Omega(\mathbf{X}_0 + \mathbf{A}_0)$ . Therefore, setting  $\mathbf{R} = \mathbf{I}_F$  in (P1) one can first run the batch Algorithm 1 to estimate the ‘ground-truth’ components  $\{\mathbf{X}_0, \mathbf{A}_0\}$ . The estimated  $\mathbf{X}_0$  exhibits three dominant singular values, confirming the low-rank property of the nominal traffic matrix. To be on the conservative side, only important spikes with magnitude greater than the threshold level  $50\|\mathbf{Y}\|_F/LT$  are retained as benchmark anomalies (nonzero entries in  $\mathbf{A}_0$ ).

*Comparison With PCA-Based Batch Estimators [21], [42]:* To highlight the merits of the batch estimator (P3), its performance is compared with the spatial PCA-based schemes reported in [21] and [42]. These methods capitalize on the fact that the anomaly-free traffic matrix has low-rank, while the presence of anomalies considerably increases the rank of  $\mathbf{Y}$ . Both algorithms rely on a two-step estimation procedure: (s1) perform PCA on the data  $\mathbf{Y}$  to extract the (low-rank) anomaly-free link traffic matrix  $\tilde{\mathbf{X}}$ ; and (s2) declare anomalies based on the residual traffic  $\tilde{\mathbf{Y}} := \mathbf{Y} - \tilde{\mathbf{X}}$ . The algorithms in [42] and [21] differ in the way (s2) is performed. On its operational phase, the algorithm in [21] declares the presence of an anomaly at time  $t$ , when the projection of  $\mathbf{y}_t$  onto the anomalous subspace exceeds a prescribed threshold. It is clear that the aforementioned method is unable to identify anomalous flows. On the other hand, the network anomography approach of [42] capitalizes on the sparsity of anomalies, and recovers the anomaly matrix by minimizing  $\|\tilde{\mathbf{A}}\|_1$ , subject to the linear constraints  $\tilde{\mathbf{Y}} = \mathbf{R}\tilde{\mathbf{A}}$ .

The aforementioned methods require a priori knowledge on the rank of the anomaly-free traffic matrix, and assume there is no missing data. To carry out performance comparisons, the detection rate will be adopted as figure of merit, which measures the algorithm’s success in identifying anomalies across both flows and time instants. ROC curves are depicted in Fig. 7(a), for different values of the rank required to run the PCA-based methods. It is apparent that the estimator (P3) obtained via Algorithm 1 markedly outperforms both PCA-based methods in terms of detection performance. This is somehow expected, since (P3) advocates joint estimation of the anomalies and the nominal traffic matrix. For an instance of  $P_{FA} = 0.04$  and  $P_D = 0.93$ , Fig. 7(b) illustrates the effectiveness of the proposed algorithm in terms of unveiling the anomalous flows and time instants.

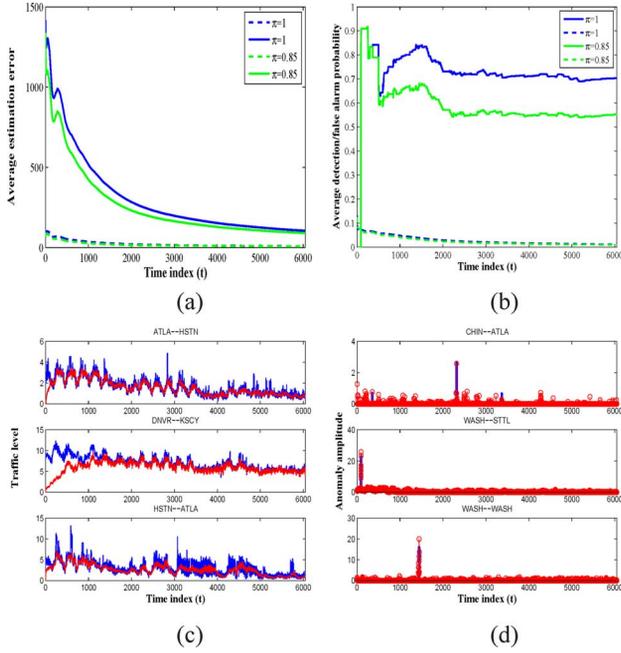


Fig. 8. Performance of the online estimator for Internet-2 network data. (a) Evolution of average anomaly (dotted) and traffic (solid) estimation errors. (b) Evolution of average detection (solid) and false alarm (dotted) rates. (c) Estimated (red) versus true (blue) link traffic for three representative links. (d) Estimated (circle markers) versus true (solid) anomalies for three representative flows when  $\pi = 0.85$ .

*Online Operation:* Algorithm 2 is tested here with the Internet-2 network data under two scenarios: with and without missing data. For the incomplete data case, a randomly chosen subset of link counts with cardinality  $0.15 \times LT$  is discarded. The penalty parameters are tuned as  $\lambda_1 = 0.7$  and  $\lambda_* = 1.4$ . The evolution of the average anomaly and traffic estimation errors, and average detection and false alarm rates are depicted in Fig. 8(a), (b), respectively. Note how in the case of full-data, after about a week the traffic subspace is accurately learned and the detection (false alarm) rates approach the values 0.72 (0.011). It is further observed that even with 15% missing data, the detection performance degrades gracefully. Finally, Fig. 8(c) [(d)] depicts how three representative link traffic levels [OD flow anomalies] are accurately tracked over time.

## VII. CONCLUDING REMARKS

An online algorithm is developed in this paper to perform a critical network monitoring task termed *dynamic anomaly-graphy*, meaning to unveil traffic volume anomalies in backbone networks adaptively. Given link-level traffic measurements (noisy superpositions of OD flows) acquired sequentially in time, the goal is to construct a *map* of anomalies in *real time*, that summarizes the network ‘health state’ along both the flow and time dimensions. Online algorithms enable tracking of anomalies in nonstationary environments, typically arising due to e.g., routing changes and missing data. The resultant online schemes offer an attractive alternative to batch algorithms, since they scale gracefully as the number of flows in the network grows, or, the time window of data acquisition increases. Comprehensive numerical tests with both synthetic and real network data corroborate the effectiveness of the proposed algorithms

and their tracking capabilities, and show that they outperform existing workhorse approaches for network anomaly detection.

## APPENDIX

### A. Update of the Anomaly Map in Algorithm 1

As argued in Section III.B, the matrix Lasso problem under [S1] decomposes over the columns of  $\mathbf{A} := [\mathbf{a}_1, \dots, \mathbf{a}_T]$ . Hence, it suffices to focus on the update of a single column, say  $\mathbf{a}_t := [a_{1,t}, \dots, a_{F,t}]'$ , which boils down to solving [cf. (5)]

$$\begin{aligned} \mathbf{a}_t[k+1] &= \arg \min_{\mathbf{a}} \left[ \frac{1}{2} \left\| \boldsymbol{\Omega}_t \left( \mathbf{y}_t - \mathbf{P}[k] \mathbf{q}_t[k] - \sum_{f=1}^F \mathbf{r}_f a_{f,t} \right) \right\|_2^2 \right. \\ &\quad \left. + \lambda_1 \sum_{f=1}^F |a_{f,t}| \right] \end{aligned} \quad (23)$$

where  $\mathbf{r}_f$  denotes the  $f$ -th column of  $\mathbf{R}$ .

Let  $n = 0, 1, \dots$ , denote the (inner) iteration index for the cyclic coordinate descent algorithm adopted to solve (23) [18, p. 92]. For the minimization at step  $k$  of the (outer) BCD iterations in Algorithm 1, the sequence of iterates  $\mathbf{a}_t[k; n]$  are initialized as  $\mathbf{a}_t[k; 0] := \mathbf{a}_t[k]$ . At each step  $n$ , the scalar coordinates  $a_{f,t}$  of vector  $\mathbf{a}_t$  are updated cyclically, by solving sequentially for  $f = 1, 2, \dots, F$

$$\begin{aligned} a_{f,t}[k; n+1] &= \arg \min_a \left[ \frac{1}{2} \left\| \tilde{\mathbf{y}}_t^{(-f)}[k; n+1] - \boldsymbol{\Omega}_t \mathbf{r}_f a \right\|_2^2 + \lambda_1 |a| \right] \end{aligned} \quad (24)$$

$$\begin{aligned} \tilde{\mathbf{y}}_t^{(-f)}[k; n+1] &:= \boldsymbol{\Omega}_t \left( \mathbf{y}_t - \mathbf{P}[k] \mathbf{q}_t[k] - \sum_{f'=1}^{f-1} \mathbf{r}_{f'} a_{f',t}[k; n+1] \right. \\ &\quad \left. - \sum_{f'=f+1}^F \mathbf{r}_{f'} a_{f',t}[k; n] \right). \end{aligned} \quad (25)$$

Vector  $\tilde{\mathbf{y}}_t^{(-f)}$  corresponds to the partial residual error without considering the contribution of the predictor  $\boldsymbol{\Omega}_t \mathbf{r}_f$ . The usefulness of a coordinate descent approach stems from the fact that the coordinate updates (24) amount to scalar Lasso-type optimizations. Skipping details that can be found in, e.g., [18, p. 93], the solutions are thus expressible in the closed form

$$\begin{aligned} a_{f,t}[k; n+1] &= \text{sign} \left( \mathbf{r}_f' \tilde{\mathbf{y}}_t^{(-f)}[k; n+1] \right) \\ &\quad \times \left[ \left| \mathbf{r}_f' \tilde{\mathbf{y}}_t^{(-f)}[k; n+1] \right| - \lambda_1 \right]_+ / \|\boldsymbol{\Omega}_t \mathbf{r}_f\|_2 \end{aligned} \quad (26)$$

which is oftentimes referred to as soft-thresholding of the partial residual  $\tilde{\mathbf{y}}_t^{(-f)}$ . Separability of the nondifferentiable  $\ell_1$ -norm term in (23) is sufficient to guarantee the convergence of (26) to a minimizer of (23), as  $n \rightarrow \infty$  [39]. Hence, the update  $\mathbf{a}_t[k+1] := \lim_{n \rightarrow \infty} [a_{1,t}[k; n], \dots, a_{F,t}[k; n]]'$  is well defined, and identical to the one in (23).

The rationale behind the actual anomaly map updates in Algorithm 1 hinges upon the fact that the solution of (23) does not need to be super accurate, since it is just an intermediate step in the outer loop defined by the BCD solver. In the relaxation pursued here, the inner iteration is halted after a single step (i.e., when  $n = 1$ ) to yield an inexact minimizer of (23). In this case, the index  $n$  can be dropped and (25)–(27) simplify to the sequential updates for  $f = 1, 2, \dots, F$

$$\begin{aligned} & \tilde{\mathbf{y}}_t^{(-f)}[k+1] \\ & := \mathbf{\Omega}_t \left( \mathbf{y}_t - \mathbf{P}[k] \mathbf{q}_t[k] - \sum_{f'=1}^{f-1} \mathbf{r}_{f'} a_{f',t}[k+1] \right. \\ & \quad \left. - \sum_{f'=f+1}^F \mathbf{r}_{f'} a_{f',t}[k] \right) \end{aligned} \quad (27)$$

$$\begin{aligned} & a_{f,t}[k+1] \\ & = \text{sign} \left( \mathbf{r}'_f \tilde{\mathbf{y}}_t^{(-f)}[k+1] \right) \left[ \left| \mathbf{r}'_f \tilde{\mathbf{y}}_t^{(-f)}[k+1] \right| - \lambda_1 \right]_+ \\ & \quad \times \|\mathbf{\Omega}_t \mathbf{r}_f\|_2^{-1} \end{aligned} \quad (28)$$

as tabulated under Algorithm 1.

### B. Proof of Lemma 1

With  $\mathbf{P}_1, \mathbf{P}_2 \in \mathcal{L}$  consider the function

$$\begin{aligned} u_t(\mathbf{a}, \mathbf{P}_1, \mathbf{P}_2) & := \frac{1}{2} \|\mathbf{F}_t(\mathbf{P}_1)(\mathbf{y}_t - \mathbf{R}_t \mathbf{a})\|_2^2 \\ & \quad - \frac{1}{2} \|\mathbf{F}_t(\mathbf{P}_2)(\mathbf{y}_t - \mathbf{R}_t \mathbf{a})\|_2^2 \end{aligned} \quad (29)$$

where  $\mathbf{F}_t(\mathbf{P}) := [\mathbf{\Omega}_t[\mathbf{I}_L - \mathbf{P}\mathbf{D}_t(\mathbf{P})]\mathbf{\Omega}_t, \sqrt{\lambda_*} \mathbf{\Omega}_t \mathbf{D}'_t(\mathbf{P})]'$ , and  $\mathbf{D}_t(\mathbf{P}) := (\lambda_* \mathbf{I}_\rho + \mathbf{P}' \mathbf{\Omega}_t \mathbf{P})^{-1} \mathbf{P}'$ . From the convexity of the Lasso problem in (11) together with the mean-value theorem and a5), it can be readily inferred that

$$\begin{aligned} & u_t(\mathbf{a}_t(\mathbf{P}_2), \mathbf{P}_1, \mathbf{P}_2) - u_t(\mathbf{a}_t(\mathbf{P}_1), \mathbf{P}_1, \mathbf{P}_2) \\ & \geq c_0 \|\mathbf{a}_t(\mathbf{P}_2) - \mathbf{a}_t(\mathbf{P}_1)\|_2^2 \end{aligned} \quad (30)$$

for some positive constant  $c_0$ . The rest of the proof deals with Lipschitz continuity of  $u_t(\cdot, \mathbf{P}_1, \mathbf{P}_2)$ . For  $\mathbf{a}_1$  and  $\mathbf{a}_2$  from a compact set  $\mathcal{A}$ , consider

$$\begin{aligned} & 2|u_t(\mathbf{a}_1, \mathbf{P}_1, \mathbf{P}_2) - u_t(\mathbf{a}_2, \mathbf{P}_1, \mathbf{P}_2)| \\ & = 2\langle \mathbf{R}'_t[\mathbf{F}'_t(\mathbf{P}_2)\mathbf{F}_t(\mathbf{P}_2) - \mathbf{F}'_t(\mathbf{P}_1)\mathbf{F}_t(\mathbf{P}_1)], (\mathbf{a}_2 - \mathbf{a}_1)\mathbf{y}'_t \rangle \\ & \quad + (\|\mathbf{F}_t(\mathbf{P}_1)\mathbf{R}_t \mathbf{a}_1\|_2^2 - \|\mathbf{F}_t(\mathbf{P}_1)\mathbf{R}_t \mathbf{a}_2\|_2^2) \\ & \quad - (\|\mathbf{F}_t(\mathbf{P}_2)\mathbf{R}_t \mathbf{a}_1\|_2^2 - \|\mathbf{F}_t(\mathbf{P}_2)\mathbf{R}_t \mathbf{a}_2\|_2^2). \end{aligned} \quad (31)$$

Introducing the auxiliary variable  $\mathbf{\Delta}_a := \mathbf{a}_2 - \mathbf{a}_1$ , the last two summands in (31) can be bounded as

$$\begin{aligned} & \|\mathbf{F}_t(\mathbf{P}_1)\mathbf{R}_t \mathbf{a}_1\|_2^2 - \|\mathbf{F}_t(\mathbf{P}_1)\mathbf{R}_t \mathbf{a}_2\|_2^2 \\ & - \|\mathbf{F}_t(\mathbf{P}_2)\mathbf{R}_t \mathbf{a}_1\|_2^2 + \|\mathbf{F}_t(\mathbf{P}_2)\mathbf{R}_t \mathbf{a}_2\|_2^2 \\ & = (\|\mathbf{F}_t(\mathbf{P}_1)\mathbf{R}_t \mathbf{\Delta}_a\|_2^2 - \|\mathbf{F}_t(\mathbf{P}_2)\mathbf{R}_t \mathbf{\Delta}_a\|_2^2) \\ & \quad + 2\langle \mathbf{R}'_t[\mathbf{F}'_t(\mathbf{P}_2)\mathbf{F}_t(\mathbf{P}_2) - \mathbf{F}'_t(\mathbf{P}_1)\mathbf{F}_t(\mathbf{P}_1)], \mathbf{a}_2 \mathbf{\Delta}'_a \rangle \end{aligned}$$

$$\begin{aligned} & \leq c_1 \|\mathbf{F}_t(\mathbf{P}_2) - \mathbf{F}_t(\mathbf{P}_1)\| \|\mathbf{\Delta}_a\|_2^2 \\ & \quad + c_2 \|\mathbf{F}'_t(\mathbf{P}_2)\mathbf{F}_t(\mathbf{P}_2) - \mathbf{F}'_t(\mathbf{P}_1)\mathbf{F}_t(\mathbf{P}_1)\| \|\mathbf{\Delta}_a\|_2 \\ & \leq c_3 \|\mathbf{F}_t(\mathbf{P}_2) - \mathbf{F}_t(\mathbf{P}_1)\| \|\mathbf{\Delta}_a\|_2 \end{aligned} \quad (32)$$

for some constants  $c_1, c_2, c_3 > 0$ , since  $\|\mathbf{F}_t(\mathbf{P})\|$  for  $\mathbf{P} \in \mathcal{L}$ ,  $\|\mathbf{\Delta}_a\|_2$ ,  $\|\mathbf{a}_2\|_2$  for  $\mathbf{a}_1, \mathbf{a}_2 \in \mathcal{A}$ , and  $\|\mathbf{R}_t\|$  are all uniformly bounded. The first summand on the right-hand side of (31) is similarly bounded (details omitted here). Next, to establish that  $\mathbf{F}_t(\mathbf{P})$  is Lipschitz one can derive the following bound ( $\mathbf{\Delta}_P := \mathbf{P}_2 - \mathbf{P}_1$ )

$$\begin{aligned} & \|\mathbf{F}_t(\mathbf{P}_2) - \mathbf{F}_t(\mathbf{P}_1)\| \\ & \leq \|\mathbf{\Omega}_t[\mathbf{P}_2 \mathbf{D}_t(\mathbf{P}_2) - \mathbf{P}_1 \mathbf{D}_t(\mathbf{P}_1)]\mathbf{\Omega}_t\| \\ & \quad + \sqrt{\lambda_*} \|\mathbf{\Omega}_t(\mathbf{D}'_t(\mathbf{P}_2) - \mathbf{D}'_t(\mathbf{P}_1))\| \\ & \leq \|\mathbf{P}_1\| (\|\mathbf{P}_1\| + \sqrt{\lambda_*}) \|(\lambda_* \mathbf{I}_\rho + \mathbf{P}'_2 \mathbf{\Omega}_t \mathbf{P}_2)^{-1} \\ & \quad - (\lambda_* \mathbf{I}_\rho + \mathbf{P}'_1 \mathbf{\Omega}_t \mathbf{P}_1)^{-1}\| \\ & \quad + \|\mathbf{\Delta}_P\| (\|\mathbf{P}_1\| + \|\mathbf{P}_2\| \\ & \quad + \sqrt{\lambda_*}) \|(\lambda_* \mathbf{I}_\rho + \mathbf{P}'_2 \mathbf{\Omega}_t \mathbf{P}_2)^{-1}\|. \end{aligned} \quad (33)$$

Define  $\mathbf{G}_t := \mathbf{\Delta}'_P \mathbf{\Omega}_t \mathbf{P}_1 + \mathbf{\Delta}'_P \mathbf{\Omega}_t \mathbf{\Delta}_P + \mathbf{P}'_1 \mathbf{\Omega}_t \mathbf{\Delta}_P$  and  $\mathbf{H}_{t,i} := \lambda_* \mathbf{I}_\rho + \mathbf{P}_i \mathbf{\Omega}_t \mathbf{P}'_i$ ,  $i = 1, 2$ , and consider the following identity

$$\mathbf{H}_{t,1}^{-1} = (\mathbf{H}_{t,1} + \mathbf{G}_t)^{-1} + \mathbf{H}_{t,1}^{-1} \mathbf{G}_t (\mathbf{H}_{t,1} + \mathbf{G}_t)^{-1}.$$

The first term in the right-hand of (33) is then bounded as follows

$$\begin{aligned} & \|(\lambda_* \mathbf{I}_\rho + \mathbf{P}'_2 \mathbf{\Omega}_t \mathbf{P}_2)^{-1} - (\lambda_* \mathbf{I}_\rho + \mathbf{P}'_1 \mathbf{\Omega}_t \mathbf{P}_1)^{-1}\| \\ & = \|(\mathbf{H}_{t,1} + \mathbf{G}_t)^{-1} - \mathbf{H}_{t,1}^{-1}\| \\ & \leq \|\mathbf{H}_{t,1}^{-1}\| \|\mathbf{G}_t\| \|(\mathbf{H}_{t,1} + \mathbf{G}_t)^{-1}\| \\ & \leq \left(\frac{1}{\lambda_*}\right)^2 \|\mathbf{G}_t\| \leq c_4 \|\mathbf{\Delta}_P\|. \end{aligned} \quad (34)$$

Putting the pieces together  $\mathbf{F}_t(\cdot)$  is found to be Lipschitz and subsequently (31) is bounded by a constant factor of  $\|\mathbf{\Delta}_P\| \|\mathbf{\Delta}_a\|_2$ . Substituting  $\mathbf{a}_1 = \mathbf{a}_t(\mathbf{P}_1)$  and  $\mathbf{a}_2 = \mathbf{a}_t(\mathbf{P}_2)$  along with the bound in (30) yields the desired result  $\|\mathbf{a}_t(\mathbf{P}_2) - \mathbf{a}_t(\mathbf{P}_1)\|_2 \leq c_5 \|\mathbf{P}_2 - \mathbf{P}_1\|$ . Furthermore, from the relationship  $\mathbf{q}_t = \mathbf{D}_t(\mathbf{P}) \mathbf{\Omega}_t (\mathbf{y}_t - \mathbf{R}_t \mathbf{a}_t)$ , Lipschitz continuity of  $\mathbf{q}_t(\mathbf{P})$  readily follows.

Moreover,  $g_t(\mathbf{P}, \mathbf{q}[t], \mathbf{a}[t])$  is a quadratic function on a compact set, and thus clearly Lipschitz continuous. To prove Lipschitz continuity of  $\ell_t(\mathbf{P})$ , recall the definition  $\{\mathbf{q}_t(\mathbf{P}), \mathbf{a}_t(\mathbf{P})\} = \arg \min_{\{\mathbf{q}, \mathbf{a}\}} g_t(\mathbf{P}, \mathbf{q}, \mathbf{a})$  to obtain after some algebra

$$\begin{aligned} & \ell_t(\mathbf{P}_2) - \ell_t(\mathbf{P}_1) = \frac{1}{2} \|\mathcal{P}_{\Omega_t}(\mathbf{P}_2 \mathbf{q}_t(\mathbf{P}_2) + \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_2))\|_2^2 \\ & \quad - \|\mathcal{P}_{\Omega_t}(\mathbf{P}_1 \mathbf{q}_t(\mathbf{P}_1) + \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_1))\|_2^2 \\ & \quad - \langle \mathcal{P}_{\Omega_t}(\mathbf{y}_t), \mathbf{P}_2 \mathbf{q}_t(\mathbf{P}_2) + \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_2) \\ & \quad - \mathbf{P}_1 \mathbf{q}_t(\mathbf{P}_1) - \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_1) \rangle \\ & \quad + \frac{\lambda_*}{2} (\|\mathbf{q}_t(\mathbf{P}_2)\|_2^2 - \|\mathbf{q}_t(\mathbf{P}_1)\|_2^2) \\ & \quad + \lambda_1 (\|\mathbf{a}_t(\mathbf{P}_2)\|_1 - \|\mathbf{a}_t(\mathbf{P}_1)\|_1). \end{aligned} \quad (35)$$

The first term in the right-hand side of (35) is bounded as

$$\begin{aligned}
& \|\mathcal{P}_{\Omega_t}(\mathbf{P}_2 \mathbf{q}_t(\mathbf{P}_2) + \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_2))\|_2^2 \\
& - \|\mathcal{P}_{\Omega_t}(\mathbf{P}_1 \mathbf{q}_t(\mathbf{P}_1) + \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_1))\|_2^2 \\
& \leq (\|\mathcal{P}_{\Omega_t}(\mathbf{P}_2 \mathbf{q}_t(\mathbf{P}_2) - \mathbf{P}_1 \mathbf{q}_t(\mathbf{P}_1))\|_2 \\
& \quad + \|\mathcal{P}_{\Omega_t}(\mathbf{R}_t \mathbf{a}_t(\mathbf{P}_2) - \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_1))\|_2) \\
& \quad \times (\|\mathcal{P}_{\Omega_t}(\mathbf{P}_2 \mathbf{q}_t(\mathbf{P}_2) + \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_2))\|_2 \\
& \quad + \|\mathcal{P}_{\Omega_t}(\mathbf{P}_1 \mathbf{q}_t(\mathbf{P}_1) + \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_1))\|_2) \\
& \leq c_6 (\|\mathbf{P}_2 - \mathbf{P}_1\| \|\mathbf{q}_t(\mathbf{P}_2)\|_2 \\
& \quad + \|\mathbf{P}_1\| \|\mathbf{q}_t(\mathbf{P}_2) - \mathbf{q}_t(\mathbf{P}_1)\|_2 \\
& \quad + \|\mathbf{R}_t\| \|\mathbf{a}_t(\mathbf{P}_2) - \mathbf{a}_t(\mathbf{P}_1)\|_2) \quad (36)
\end{aligned}$$

for some constant  $c_6 > 0$ . The second one is bounded as

$$\begin{aligned}
& \langle \mathcal{P}_{\Omega_t}(\mathbf{y}_t), \mathbf{P}_2 \mathbf{q}_t(\mathbf{P}_2) + \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_2) - \mathbf{P}_1 \mathbf{q}_t(\mathbf{P}_1) - \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_1) \rangle \\
& \leq \|\mathcal{P}_{\Omega_t}(\mathbf{y}_t)\|_2 (\|\mathcal{P}_{\Omega_t}(\mathbf{P}_2 \mathbf{q}_t(\mathbf{P}_2) - \mathbf{P}_1 \mathbf{q}_t(\mathbf{P}_1))\|_2 \\
& \quad + \|\mathcal{P}_{\Omega_t}(\mathbf{R}_t \mathbf{a}_t(\mathbf{P}_2) - \mathbf{R}_t \mathbf{a}_t(\mathbf{P}_1))\|_2) \\
& \leq \|\mathcal{P}_{\Omega_t}(\mathbf{y}_t)\|_2 (\|\mathbf{P}_2 - \mathbf{P}_1\| \|\mathbf{q}_t(\mathbf{P}_2)\|_2 + \|\mathbf{P}_1\| \\
& \quad \times \|\mathbf{q}_t(\mathbf{P}_2) - \mathbf{q}_t(\mathbf{P}_1)\|_2 + \|\mathbf{R}_t\| \|\mathbf{a}_t(\mathbf{P}_2) - \mathbf{a}_t(\mathbf{P}_1)\|_2). \quad (37)
\end{aligned}$$

Finally, one can bound the third term in (35) as

$$\begin{aligned}
& \frac{\lambda_*}{2} (\|\mathbf{q}_t(\mathbf{P}_2)\|_2^2 - \|\mathbf{q}_t(\mathbf{P}_1)\|_2^2) + \lambda_1 (\|\mathbf{a}_t(\mathbf{P}_2)\|_1 - \|\mathbf{a}_t(\mathbf{P}_1)\|_1) \\
& \leq \frac{\lambda_*}{2} \|\mathbf{q}_t(\mathbf{P}_2) - \mathbf{q}_t(\mathbf{P}_1)\|_2 (\|\mathbf{q}_t(\mathbf{P}_2)\|_2 + \|\mathbf{q}_t(\mathbf{P}_1)\|_2) \\
& \quad + \lambda_1 \sqrt{F} \|\mathbf{a}_t(\mathbf{P}_2) - \mathbf{a}_t(\mathbf{P}_1)\|_2. \quad (38)
\end{aligned}$$

Since  $\mathbf{q}_t(\mathbf{P})$  and  $\mathbf{a}_t(\mathbf{P})$  are Lipschitz as proved earlier, and  $\mathbf{P}_1, \mathbf{P}_2 \in \mathcal{L}$  are uniformly bounded, the expressions in the right-hand side of (36)–(38) are upper bounded by a constant factor of  $\|\mathbf{P}_2 - \mathbf{P}_1\|$ , and so is  $|\ell_t(\mathbf{P}_2) - \ell_t(\mathbf{P}_1)|$  after applying the triangle inequality to (35).

Regarding  $\nabla \ell_t(\mathbf{P})$ , notice first that since  $\{\mathbf{q}_t(\mathbf{P}), \mathbf{a}_t(\mathbf{P})\}$  is the unique minimizer of  $g_t(\mathbf{P}, \mathbf{q}, \mathbf{a})$  [cf. a5)], Danskin's theorem [7, Prop. B.25(a)] implies that  $\nabla \ell_t(\mathbf{P}) = \mathcal{P}_{\Omega_t}(\mathbf{y}_t - \mathbf{P} \mathbf{q}_t(\mathbf{P}) - \mathbf{R}_t \mathbf{a}_t(\mathbf{P})) \mathbf{q}'_t(\mathbf{P})$ . In the sequel, the triangle inequality will be used to split the norm in the right-hand side of

$$\begin{aligned}
& \|\nabla \ell_t(\mathbf{P}_2) - \nabla \ell_t(\mathbf{P}_1)\|_F \\
& = \|\mathcal{P}_{\Omega_t}(\mathbf{y}_t)[\mathbf{q}_t(\mathbf{P}_2) - \mathbf{q}_t(\mathbf{P}_1)]' - [\mathcal{P}_{\Omega_t}(\mathbf{P}_2 \mathbf{q}_t(\mathbf{P}_2)) \mathbf{q}'_t(\mathbf{P}_2) \\
& \quad - \mathcal{P}_{\Omega_t}(\mathbf{P}_1 \mathbf{q}_t(\mathbf{P}_1)) \mathbf{q}'_t(\mathbf{P}_1)] - [\mathcal{P}_{\Omega_t}(\mathbf{R}_t \mathbf{a}_t(\mathbf{P}_2)) \mathbf{q}'_t(\mathbf{P}_2) \\
& \quad - \mathcal{P}_{\Omega_t}(\mathbf{R}_t \mathbf{a}_t(\mathbf{P}_1)) \mathbf{q}'_t(\mathbf{P}_1)]\|_F. \quad (39)
\end{aligned}$$

The first term inside the norm is bounded as

$$\begin{aligned}
& \|\mathcal{P}_{\Omega_t}(\mathbf{y}_t)[\mathbf{q}_t(\mathbf{P}_2) - \mathbf{q}_t(\mathbf{P}_1)]'\|_F \\
& \leq \|\mathcal{P}_{\Omega_t}(\mathbf{y}_t)\|_2 \|\mathbf{q}_t(\mathbf{P}_2) - \mathbf{q}_t(\mathbf{P}_1)\|_2. \quad (40)
\end{aligned}$$

After some algebraic manipulations, the second term is also bounded as

$$\begin{aligned}
& \|\mathcal{P}_{\Omega_t}(\mathbf{P}_2 \mathbf{q}_t(\mathbf{P}_2)) \mathbf{q}'_t(\mathbf{P}_2) - \mathcal{P}_{\Omega_t}(\mathbf{P}_1 \mathbf{q}_t(\mathbf{P}_1)) \mathbf{q}'_t(\mathbf{P}_1)\|_F \\
& \leq \|\mathbf{P}_2 - \mathbf{P}_1\|_F \|\mathbf{q}_t(\mathbf{P}_2)\|_2^2 + \|\mathbf{q}_t(\mathbf{P}_2) - \mathbf{q}_t(\mathbf{P}_1)\|_2 \\
& \quad \times (\|\mathbf{q}_t(\mathbf{P}_2)\|_2 + \|\mathbf{q}_t(\mathbf{P}_1)\|_2) \quad (41)
\end{aligned}$$

and finally one can simply bound the third term as

$$\begin{aligned}
& \|\mathcal{P}_{\Omega_t}(\mathbf{R}_t \mathbf{a}_t(\mathbf{P}_2)) \mathbf{q}'_t(\mathbf{P}_2) - \mathcal{P}_{\Omega_t}(\mathbf{R}_t \mathbf{a}_t(\mathbf{P}_1)) \mathbf{q}'_t(\mathbf{P}_1)\|_F \\
& \leq \|\mathbf{R}_t\| (\|\mathbf{a}_t(\mathbf{P}_2) - \mathbf{a}_t(\mathbf{P}_1)\|_2 \|\mathbf{q}_t(\mathbf{P}_1)\|_2 \\
& \quad + \|\mathbf{q}_t(\mathbf{P}_2) - \mathbf{q}_t(\mathbf{P}_1)\|_2 \|\mathbf{a}_t(\mathbf{P}_1)\|_2). \quad (42)
\end{aligned}$$

Since  $\mathbf{a}_t(\mathbf{P})$  and  $\mathbf{q}_t(\mathbf{P})$  are Lipschitz and uniformly bounded, from (40)–(42) one can easily deduce that  $\nabla \ell_t(\cdot)$  is indeed Lipschitz continuous. ■

### C. Proof of Lemma 2

Exploiting that  $\nabla \hat{C}_t(\mathbf{P}[t]) = \nabla \hat{C}_{t+1}(\mathbf{P}[t+1]) = \mathbf{0}_{L \times \rho}$  by algorithmic construction and the strong convexity assumption on  $\hat{C}_t$  [cf. a4)], application of the mean-value theorem readily yields

$$\hat{C}_t(\mathbf{P}[t+1]) \geq \hat{C}_t(\mathbf{P}[t]) + \frac{c}{2} \|\mathbf{P}[t+1] - \mathbf{P}[t]\|_F^2$$

$$\hat{C}_{t+1}(\mathbf{P}[t]) \geq \hat{C}_{t+1}(\mathbf{P}[t+1]) + \frac{c}{2} \|\mathbf{P}[t+1] - \mathbf{P}[t]\|_F^2.$$

Upon defining the function  $h_t(\mathbf{P}) := \hat{C}_t(\mathbf{P}) - \hat{C}_{t+1}(\mathbf{P})$  one arrives at

$$c \|\mathbf{P}[t+1] - \mathbf{P}[t]\|_F^2 \leq h_t(\mathbf{P}[t+1]) - h_t(\mathbf{P}[t]). \quad (43)$$

To complete the proof, it suffices to show that  $h_t$  is Lipschitz with constant  $\mathcal{O}(1/t)$ , and upper bound the right-hand side of (43) accordingly. Since [cf. (16)]

$$\begin{aligned}
h_t(\mathbf{P}) &= \frac{1}{t(t+1)} \sum_{\tau=1}^t g_\tau(\mathbf{P}, \mathbf{q}[\tau], \mathbf{a}[\tau]) \\
& \quad - \frac{1}{t+1} g_{t+1}(\mathbf{P}, \mathbf{q}[t+1], \mathbf{a}[t+1]) \\
& \quad + \frac{\lambda_*}{2t(t+1)} \|\mathbf{P}\|_F^2 \quad (44)
\end{aligned}$$

and  $g_i(\mathbf{P})$  is Lipschitz according to Lemma 1, it follows that  $h_t$  is Lipschitz with constant  $\mathcal{O}(1/t)$ . ■

### D. Proof of Lemma 3

The first step of the proof is to show that  $\{\hat{C}_t(\mathbf{P}[t])\}_{t=1}^\infty$  is a quasi-martingale sequence, and hence convergent a.s. [23]. Building on the variations of  $\hat{C}_t(\mathbf{P}[t])$ , one can write

$$\begin{aligned}
& \hat{C}_{t+1}(\mathbf{P}[t+1]) - \hat{C}_t(\mathbf{P}[t]) = \hat{C}_{t+1}(\mathbf{P}[t+1]) - \hat{C}_{t+1}(\mathbf{P}[t]) \\
& \quad + \hat{C}_{t+1}(\mathbf{P}[t]) - \hat{C}_t(\mathbf{P}[t]) \\
& \stackrel{(a)}{\leq} \hat{C}_{t+1}(\mathbf{P}[t]) - \hat{C}_t(\mathbf{P}[t]) \\
& = \frac{1}{t+1} \left[ g_{t+1}(\mathbf{P}[t], \mathbf{q}[t+1], \mathbf{a}[t+1]) \right. \\
& \quad \left. - \frac{1}{t} \sum_{\tau=1}^t g_\tau(\mathbf{P}[\tau], \mathbf{q}[\tau], \mathbf{a}[\tau]) \right] \\
& \stackrel{(b)}{\leq} \frac{1}{t+1} \left[ g_{t+1}(\mathbf{P}[t], \mathbf{q}[t+1], \mathbf{a}[t+1]) \right. \\
& \quad \left. - \frac{1}{t} \sum_{\tau=1}^t \ell_\tau(\mathbf{P}[t]) \right] \quad (45)
\end{aligned}$$

where (a) uses that  $\hat{C}_{t+1}(\mathbf{P}[t+1]) \leq \hat{C}_{t+1}(\mathbf{P}[t])$ , and (b) follows from  $C_t(\mathbf{P}[t]) \leq \hat{C}_t(\mathbf{P}[t])$ .

Collect all past data in  $\mathcal{F}_t = \{(\Omega_\tau, \mathbf{y}_\tau) : \tau \leq t\}$ , and recall that under a1) the random processes  $\{\Omega_t, \mathbf{y}_t\}$  are i.i.d. over time. Then, the expected variations of the approximate cost function are bounded as

$$\begin{aligned} & \mathbb{E}[\hat{C}_{t+1}(\mathbf{P}[t+1]) - \hat{C}_t(\mathbf{P}[t]) | \mathcal{F}_t] \\ & \leq \frac{1}{t+1} \left( \mathbb{E}[g_{t+1}(\mathbf{P}[t], \mathbf{q}[t+1], \mathbf{a}[t+1]) | \mathcal{F}_t] \right. \\ & \quad \left. - \frac{1}{t} \sum_{\tau=1}^t \ell_\tau(\mathbf{P}[t]) \right) \\ & \stackrel{(a)}{=} \frac{1}{t+1} \left( \mathbb{E}[\ell_1(\mathbf{P}[t])] - \frac{1}{t} \sum_{\tau=1}^t \ell_\tau(\mathbf{P}[t]) \right) \\ & \leq \frac{1}{t+1} \sup_{\mathbf{P}[t] \in \mathcal{C}} \left( \mathbb{E}[\ell_1(\mathbf{P}[t])] - \frac{1}{t} \sum_{\tau=1}^t \ell_\tau(\mathbf{P}[t]) \right) \end{aligned} \quad (46)$$

where (a) follows from a1). Using the fact that  $\ell_i(\mathbf{P}_t)$  is Lipschitz from Lemma 1, and uniformly bounded due to a2), Donsker's Theorem [40, Ch. 19.2] yields

$$\mathbb{E} \left[ \sup_{\mathbf{P}[t]} \left| \mathbb{E}[\ell_1(\mathbf{P}[t])] - \frac{1}{t} \sum_{\tau=1}^t \ell_\tau(\mathbf{P}[t]) \right| \right] = \mathcal{O}(1/\sqrt{t}). \quad (47)$$

From (46) and (47) the expected non-negative variations can be readily bounded as

$$\mathbb{E}[\mathbb{E}[\hat{C}_{t+1}(\mathbf{P}[t+1]) - \hat{C}_t(\mathbf{P}[t]) | \mathcal{F}_t]_+] = \mathcal{O}(1/t^{3/2}) \quad (48)$$

and consequently

$$\sum_{t=1}^{\infty} \mathbb{E}[\mathbb{E}[\hat{C}_{t+1}(\mathbf{P}[t+1]) - \hat{C}_t(\mathbf{P}[t]) | \mathcal{F}_t]_+] < \infty \quad (49)$$

which indeed proves that  $\{\hat{C}_t(\mathbf{P}[t])\}_{t=1}^{\infty}$  is a quasi-martingale sequence.

To prove the second part, define first  $U_t(\mathbf{P}[t]) := C_t(\mathbf{P}[t]) - (\lambda_*)/(2t)\|\mathbf{P}[t]\|_F^2$  and  $\hat{U}_t(\mathbf{P}[t]) := \hat{C}_t(\mathbf{P}[t]) - (\lambda_*)/(2t)\|\mathbf{P}[t]\|_F^2$  for which  $U_t(\mathbf{P}[t]) - \hat{U}_t(\mathbf{P}[t]) = C_t(\mathbf{P}[t]) - \hat{C}_t(\mathbf{P}[t])$  holds. Following similar arguments as with  $\hat{C}_t(\mathbf{P}[t])$ , one can show that (49) holds for  $\hat{U}_t(\mathbf{P}[t])$  as well. It is also useful to expand the variations

$$\begin{aligned} \hat{U}_{t+1}(\mathbf{P}[t+1]) - \hat{U}_t(\mathbf{P}[t]) &= \hat{U}_{t+1}(\mathbf{P}[t+1]) - \hat{U}_{t+1}(\mathbf{P}[t]) \\ & \quad + \frac{\ell_{t+1}(\mathbf{P}[t]) - U_t(\mathbf{P}[t])}{t+1} + \frac{U_t(\mathbf{P}[t]) - \hat{U}_t(\mathbf{P}[t])}{t+1} \end{aligned}$$

and bound their expectation conditioned on  $\mathcal{F}_t$ , to arrive at

$$\begin{aligned} & \frac{U_t(\mathbf{P}[t]) - \hat{U}_t(\mathbf{P}[t])}{t+1} \\ & \leq |\mathbb{E}[\hat{U}_{t+1}(\mathbf{P}[t+1]) - \hat{U}_{t+1}(\mathbf{P}[t]) | \mathcal{F}_t]| \\ & \quad + |\mathbb{E}[\hat{U}_{t+1}(\mathbf{P}[t+1]) - \hat{U}_t(\mathbf{P}[t]) | \mathcal{F}_t]| \\ & \quad + \frac{1}{t+1} \left| \mathbb{E}[\ell_1(\mathbf{P}[t])] - \frac{1}{t} \sum_{\tau=1}^t \ell_\tau(\mathbf{P}[t]) \right|. \end{aligned} \quad (50)$$

Focusing on the right-hand side of (50), the second and third terms are both  $\mathcal{O}(1/t^{3/2})$  since counterparts of (47) and (48) also hold for  $\hat{U}_t(\mathbf{P}[t])$ . With regards to the first term, using the fact that  $\hat{C}_{t+1}(\mathbf{P}[t+1]) < \hat{C}_{t+1}(\mathbf{P}[t])$ , from Lemma 1 and a4), it follows that  $\hat{U}_{t+1}(\mathbf{P}[t+1]) - \hat{U}_{t+1}(\mathbf{P}[t]) = o(1/t)$ . All in all,

$$\sum_{t=1}^{\infty} \frac{\hat{U}_t(\mathbf{P}[t]) - U_t(\mathbf{P}[t])}{t+1} < \infty \quad \text{a.s.} \quad (51)$$

Defining  $d_t(\mathbf{P}[t]) := \hat{U}_t(\mathbf{P}[t]) - U_t(\mathbf{P}[t])$ , due to Lipschitz continuity of  $\ell_t$  and  $g_t$  (cf. Lemma 1), and uniform boundedness of  $\{\mathbf{P}_t\}_{t=1}^{\infty}$  [cf. a3)], invoking Lemma 2 one can establish that  $d_{t+1}(\mathbf{P}[t+1]) - d_t(\mathbf{P}[t]) = \mathcal{O}(1/t)$ . Hence, Dirichlet's theorem [34] applied to the sum (51) asserts that  $\lim_{t \rightarrow \infty} d_t(\mathbf{P}[t]) = 0$  a.s., and consequently  $\lim_{t \rightarrow \infty} (\hat{C}_t(\mathbf{P}[t]) - C_t(\mathbf{P}[t])) = 0$  a.s. ■

## REFERENCES

- [1] [Online]. Available: <http://internet2.edu/observatory/archive/datacollections.html>[Online]. Available:
- [2] A. Abdelkafi, Y. Jiang, W. Wang, A. Aslebo, and O. Kvittem, "Robust traffic anomaly detection with principal component pursuit," in *Proc. ACM CoNEXT Student Workshop*, Philadelphia, PA, Nov. 2010.
- [3] A. Agarwal, S. Negahban, and M. J. Wainwright, "Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions," *Ann. Statist.*, vol. 40, pp. 1171–1197, 2012.
- [4] T. Ahmed, M. Coates, and A. Lakhina, "Multivariate online anomaly detection using kernel recursive least squares," in *Proc. IEEE/ACM Int. Conf. Comput. Commun.*, Anchorage, AK, May 2007.
- [5] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Jun. 2010.
- [6] 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.
- [7] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Nashua, NH: Athena-Scientific, 1999.
- [8] J. F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2008.
- [9] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 1, pp. 1–37, 2011.
- [10] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2009.
- [11] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–722, 2009.
- [12] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
- [13] E. J. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
- [14] V. Chandrasekaran, S. Sanghavi, P. R. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.
- [15] Q. Chenlu and N. Vaswani, "Recursive sparse recovery in large but correlated noise," in *Proc. 49th Allerton Conf. Commun., Control, Comput.*, Sep. 2011, pp. 752–759.
- [16] Y. Chi, Y. C. Eldar, and R. Calderbank, "PETRELS: Subspace estimation and tracking from partial observations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 3301–3304.
- [17] A. Chistov and D. Grigorev, "Complexity of quantifier elimination in the theory of algebraically closed fields," in *Math. Found. of Computer Science*, ser. Lecture Notes in Computer Science. Berlin/Heidelberg, Germany: Springer, 1984, vol. 176, pp. 17–31.
- [18] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. New York: Springer, 2009.

- [19] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Visi. Pattern Recogn.*, Providence, RI, Jun. 2012.
- [20] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast Distributed Gradient Methods," arXiv:1112.2972v1 [cs.IT].
- [21] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. ACM SIGCOMM*, Portland, OR, Aug. 2004.
- [22] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," in *Proc. ACM SIGMETRICS*, New York, Jul. 2004.
- [23] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identification*, 2nd ed. Cambridge, MA: MIT Press, 1983.
- [24] J. Mairal, J. 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.
- [25] M. Mardani, G. Mateos, and G. B. Giannakis, "In-network sparsity regularized rank minimization: Applications and algorithms," *IEEE Trans. Signal Process.*, 2013, see also arXiv:1203.1570v1 [cs.MA], submitted for publication.
- [26] M. Mardani, G. Mateos, and G. B. Giannakis, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *IEEE Trans. Inf. Theory.*, 2013, see also arXiv:1204.6537v1 [cs.IT], submitted for publication.
- [27] G. Mateos and G. B. Giannakis, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5176–5190, Oct. 2012.
- [28] Z. Meng, A. Wiesel, and A. Hero, "Distributed principal component analysis on networks via directed graphical models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 2877–2880.
- [29] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, pp. 227–234, 1995.
- [30] 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.
- [31] 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, 2010.
- [32] B. Recht and C. Re, "Parallel stochastic gradient algorithms for large-scale matrix completion," *Math. Programm. Comput.*, 2011 [Online]. Available: [http://www.optimization-online.org/DB\\_HTML/2011/04/3012.html](http://www.optimization-online.org/DB_HTML/2011/04/3012.html), Preprint, submitted for publication
- [33] M. Roughan, "A case study of the accuracy of SNMP measurements," *J. Electr. Comput. Eng.*, Dec. 2010, article ID 812979.
- [34] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York: McGraw-Hill, 1976.
- [35] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.
- [36] V. Solo and X. Kong, *Adaptive Signal Processing Algorithms: Stability and Performance*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [37] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2191–2204, Aug. 2003.
- [38] J. Tropp, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 1030–1051, Mar. 2006.
- [39] P. Tseng, "Convergence of a block coordinate descent method for non-differentiable minimization," *J. Optimiz. Theory Applicat.*, vol. 109, pp. 475–494, 2001.
- [40] A. W. Van Der Vaart, *Asymptotic Statistics*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [41] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal. Process.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.
- [42] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan, "Network anomography," in *Proc. ACM SIGCOM Conf. Internet Meas.*, Berekly, CA, Oct. 2005.
- [43] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *Proc. ACM SIGCOM Conf. Data Commun.*, New York, Oct. 2009.
- [44] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma, "Stable principal component pursuit," in *Proc. Intl. Symp. Inf. Theory*, Austin, TX, Jun. 2010, pp. 1518–1522.



**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 inference and optimization, sparse and low rank recovery, and cross-layer design of wireless networks.

Mr. Mardani is the recipient of the Best Student Paper Award from the 13th IEEE Workshop on Signal Processing Advances in Wireless Communications in June 2012. He also received the ADC Fellowship Award from the Digital Technology Center at the University of Minnesota for two academic years 2009–2010 and 2010–2011.



**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 and computer engineering from the University of Minnesota, Minneapolis, in 2009 and 2011.

Since 2012, he has been a post doctoral associate with the Department of Electrical and Computer Engineering and the Digital Technology Center, University of Minnesota. Since 2003, he is 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 communication theory, signal processing and networking. His current research focuses on distributed signal processing, sparse linear regression, and statistical learning for social data analysis and network health monitoring.



**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 325 journal papers, 525 conference papers, 20 book chapters, two edited books and two research monographs. Current research focuses on compressive sensing, cognitive radios, cross-layer designs, wireless sensors, social and power grid networks. He is the (co-) inventor of 21 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, and the G. W. Taylor Award for Distinguished Research from the University of Minnesota. 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.