[Gonzalo Mateos and Ketan Rajawat]

Dynamic Network Cartography

[Advances in network health monitoring]

Communication networks have evolved from specialized research and tactical transmission systems to large-scale and highly complex interconnections of intelligent devices, increasingly becoming more commercial, consumer oriented, and heterogeneous. Propelled by emergent social networking services and high-definition streaming platforms, network traffic has grown explosively thanks to the advances in processing speed and storage capacity of state-of-the-art communication technologies. As "netizens" demand a seamless networking experience that entails not only higher speeds but also resilience and robustness to failures and malicious cyberattacks, ample opportunities for signal processing (SP) research arise. The vision is for ubiquitous smart network devices to enable data-driven statistical learning algorithms for distributed, robust, and online network operation and management, adaptable to the dynamically evolving network landscape with minimal need for human intervention. This article aims to delineate the analytical background and the relevance of SP tools to dynamic network monitoring, introducing the SP readership to the concept of dynamic network cartography—a framework to construct maps of the dynamic network state in an efficient and scalable manner tailored to large-scale heterogeneous networks.

Digital Object Identifier 10.1109/MSP.2012.2232355
Date of publication: 5 April 2013

INTRODUCTION

The emergence of multimedia-enriched social networking services and Internet-friendly portable devices is multiplying network traffic volume day by day [53]. Wireless connectivity under the envisioned dynamic spectrum paradigm [30] relies on mobile networks of diverse nodes, which are nevertheless united by unparalleled cognition capabilities, adaptability, and decision-making attributes. Moreover, the advent of networks of intelligent devices such as those deployed to monitor the smart power grid, transportation networks, medical information networks, and cognitive radio (CR) networks, will transform the communication infrastructure to an even more complex and

heterogeneous one. Thus, ensuring compliance to service-level agreements and quality-of-service (QoS) guarantees necessitates breakthrough management and monitoring tools providing operators with a comprehensive view of the network landscape. Situational awareness provided by such tools will be the key enabler for effective information dissemination, routing and congestion control, network health management, risk analysis, and security assurance.

But this great promise comes with great challenges. Acquiring network-wide performance and utilization metrics for large networks is no easy task. Suppose, for instance, that traffic volumes are of interest, not only for gauging instantaneous network health but also for more complex network management tasks such as intrusion detection, capacity provisioning, and network planning [56]. While traffic volumes on links (also called link counts) are readily acquired using off-the-shelf tools such as the simple network management protocol (SNMP), missing link-count measurements may still 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 [48], [50]. Classical approaches relying either on simple time-series interpolation or on regularized least-squares (LS) formulations for predicting the missing link counts [51] have not been able to fully capture the complexity of the Internet traffic. This is evidenced by the recent upsurge of efforts toward advanced network tomography [14] and spatiotemporal traffic estimation algorithms for network monitoring [27], [50], [56].

SITUATIONAL AWARENESS PROVIDED BY SUCH TOOLS WILL BE THE KEY ENABLER FOR EFFECTIVE INFORMATION DISSEMINATION, ROUTING AND CONGESTION CONTROL, NETWORK HEALTH MANAGEMENT, RISK ANALYSIS, AND SECURITY ASSURANCE.

Similarly, path metrics such as end-to-end delays are of great interest to service providers because they directly affect the end-user experience. The challenge here is that the number of paths grows very fast as the number of nodes increases. Probing exhaustively all origin-destination (OD) pairs is impractical and wasteful of resources even for moderate-size networks [18], [49]. Accurate prediction of missing delays based on the inherent, e.g., topology-induced correlation or smoothness traits among link and path quantities is therefore crucial for statistical analysis and monitoring tasks [33]. While the prevailing operational paradigm adopted in current networks entails nodes continuously communicating their link measurements to a central monitoring station, in-network distributed cooperation through local interactions is preferred for scalability and robustness considerations [39].

Conventional network monitoring tools entail a couple of additional limitations. First, they are typically resource heavy and tend to overload network operators with crude, unrefined data, without enough processing to separate the "data wheat from the chaff"; see, e.g., [20] and references therein. It is thus of paramount importance to construct parsimonious descriptors of the network state, for the purpose of modeling, monitoring, and management of complex interconnected systems. Due to the diversity of modern networks, the network state can incorporate typical quantities such as traffic volumes and end-to-end delays, as well as latent social metrics such as hierarchy, reputation, and vulnerability. Second, malicious activities intended to undermine network functionality or compromise secrecy of data have grown in sophistication, thus rendering traditional signature-based intrusion detection schemes increasingly obsolete. Intrusion attempts and malicious attacks manifest themselves as abrupt changes in network states [6], and such anomalous patterns are oftentimes hidden within the raw high-dimensional network data [55]. For these reasons, unveiling network anomalies in a reliable and computationally efficient manner is a challenging yet essential goal [34], [39], [55].

All in all, accurate network diagnosis and statistical analysis tools are instrumental for maintaining seamless end-user experience in dynamic environments as well as for ensuring network security and stability. In this direction, this tutorial advocates the concept of dynamic network cartography as a tool for statistical modeling, monitoring, and management of complex networks. Focus will be placed on two complementary aspects of network cartography, specifically, online construction of global network state maps using only a few measurements and the unveiling of network anomalies across network flows and time. The surveyed cartography algorithms leverage recent advances in machine learning and statistical SP methods, including sparsity-cognizant learning, kriged Kalman filtering of dynamical processes over networks, nuclear norm minimization for low-rank matrix completion, semisupervised dictionary learning (DL), and in-network optimization via the alternating-directions method of multipliers. Through a unifying treatment that revolves around network cartography, this article demonstrates how benefits from foundational SP methods can permeate to dynamic network monitoring and collectively enable inference of global network health, thus leading to enhanced network robustness and QoS.

GLOBAL PERFORMANCE PREDICTION VIA DYNAMICAL NETWORK CARTOGRAPHY

This section deals with the problem of mapping the network state from incomplete sets of measurements and touches upon two application domains. A DL algorithm is introduced first to efficiently impute missing link traffic volumes, using measurements from a wide class of (possibly nonstationary) traffic patterns [27]. Subsequently, the problem of tracking and predicting end-to-end network delay is considered, and the dynamic network kriging approach of [46] is described.

SEMISUPERVISED DICTIONARY LEARNING FOR TRAFFIC MAPS

Consider an Internet protocol (IP) network comprising N nodes and L links, carrying the traffic of F OD flows (network connections). Let $x_{l,t}$ denote the traffic volume (in bytes or packets) passing through link $l \in \{1, \ldots, L\}$ over a fixed interval of time $(t, t + \Delta t)$. Link counts across the entire network are collected in the vector $\mathbf{x}_t \in \mathbb{R}^L$, e.g., using the ubiquitous SNMP protocol. Since measured link counts are both unreliable and incomplete due to hardware or software malfunctioning, jitter, and communication errors [56], [48], they are expressed as noisy versions of a subset of $S < L$ links

$$\mathbf{y}_t = \mathbf{S}_t \mathbf{x}_t + \epsilon_t, \quad t = 1, 2, \ldots \tag{1}$$

where $\mathbf{S}_t$ is an $S \times L$ selection matrix with 0–1 entries whose rows correspond to rows of the identity matrix of size L, and ϵ_t is an $S \times 1$ zero-mean noise term with constant variance accounting for measurement and synchronization errors. Given $\mathbf{y}_t$ the aim is to form an estimate $\hat{\mathbf{x}}_t$ of the full vector of link counts $\mathbf{x}_t$, which in this case defines the network state.

A simple approach implemented in measurement-processing software, such as RRDtool [44], is to ignore the noise term and rely on one-dimensional interpolation for the time series $\{x_{l,t}\}$ per link l. The applicability and accuracy of this scheme is, however, limited since it tacitly assumes that the entries of $\mathbf{x}_t$ are uncorrelated; missing entries $x_{l,t}$ are few and do not occur in bursts; and the time series $\{\mathbf{x}_t\}$ is stationary. Nevertheless, none of these assumptions holds true in real networks [48].

The reliance on stationarity and availability of measurements from contiguous time intervals can be foregone if estimation of $\mathbf{x}_t$ is performed for each t individually. In principle, $\hat{\mathbf{x}}_t$ can be obtained if the volumes of OD traffic flows $\mathbf{z}_t \in \mathbb{R}^F$ are available, since they are related through

$$\mathbf{x}_t = \mathbf{R}\mathbf{z}_t, \tag{2}$$

where the so-termed routing matrix $\mathbf{R} := [r_{l,f}] \in \{0, 1\}^{L \times F}$ is such that $r_{l,f} = 1$ if link l carries the flow f, and zero otherwise. However, measuring $\mathbf{z}_t$ is even more difficult and in practice $\mathbf{z}_t$ is itself estimated from $\{\mathbf{x}_t\}$ through tomographic traffic inference [14], [33], where given $\mathbf{R}$ and noisy link counts, the goal is to estimate the OD flows as the solution of a linear inverse problem. Since the inverse problem is highly under-determined $[F = \mathcal{O}(N^2) \gg L = \mathcal{O}(N)]$, early approaches relied on prior knowledge in the form of statistical models for the OD flows (such as the Poisson, Gaussian, logit-choice, or gravity models), that ultimately serve as complexity-controlling (that is regularization) mechanisms [33, Ch. 9]. Among these, the state-of-the-art traffic matrix estimation algorithm uses an entropy-based regularizer and has been shown to be fast, accurate, robust, and flexible [54]. Time-series analysis-based approaches (such as the Kalman filter in [51]) have also been proposed for scenarios where link-count measurements are available over contiguous time slots.

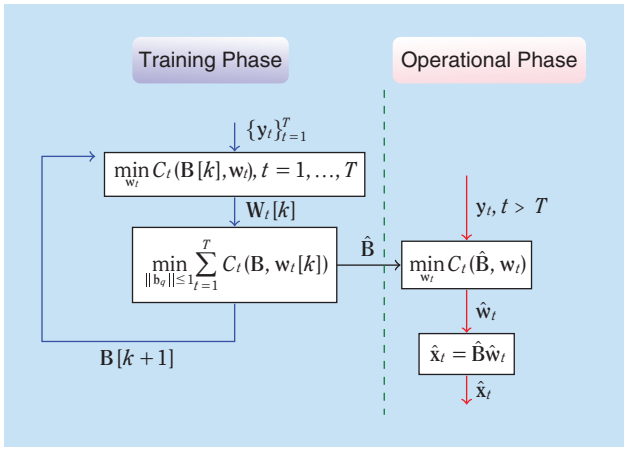
Recently, a link-count prediction algorithm was put forth in [27], where missing entries of $\mathbf{x}_t$ are estimated from historical measurements in $\mathcal{T}_S := \{\mathbf{y}_t\}_{t=1}^T$ by leveraging the structural regularity of $\mathbf{R}$ through a semisupervised DL approach. Under the DL framework, data-driven dictionaries for sparse signal representation are adopted as a versatile means of capturing parsimonious signal structures; see, e.g., [52] for a tutorial treatment. Propelled by the success of compressive sampling (CS) [24], sparse signal modeling has led to major advances in several machine learning, audio, and image processing tasks [52], [28]. Motivated by these ideas, it is postulated in [27] that link counts can be represented as a linear combination $\mathbf{x}_t = \mathbf{B}\mathbf{w}_t$ of a few ($\ll Q$) columns of an overcomplete dictionary (basis) matrix $\mathbf{B} := [\mathbf{b}_1, \ldots, \mathbf{b}_Q] \in \mathbb{R}^{L \times Q}$, where $\mathbf{w}_t \in \mathbb{R}^Q$ is a sparse vector of expansion coefficients. Many signals including speech and natural images admit sparse representations even under generic predefined dictionaries, such as those based on the Fourier and the wavelet bases, respectively [52]. Like audio and natural images, link counts can exhibit strong correlations as evidenced from the structure of $\mathbf{R}$ [cf. (2)]. For instance, the traffic volumes on links i and j are highly correlated if they both carry common flows. DL schemes are attractive due to their flexibility, since they utilize training data to *learn* an appropriate overcomplete basis customized for the data at hand. However, the use of DL for modeling network data is well motivated but so far relatively unexplored.

ACCURATE NETWORK DIAGNOSIS AND STATISTICAL ANALYSIS TOOLS ARE INSTRUMENTAL FOR MAINTAINING SEAMLESS END-USER EXPERIENCE IN DYNAMIC ENVIRONMENTS AS WELL AS FOR ENSURING NETWORK SECURITY AND STABILITY.

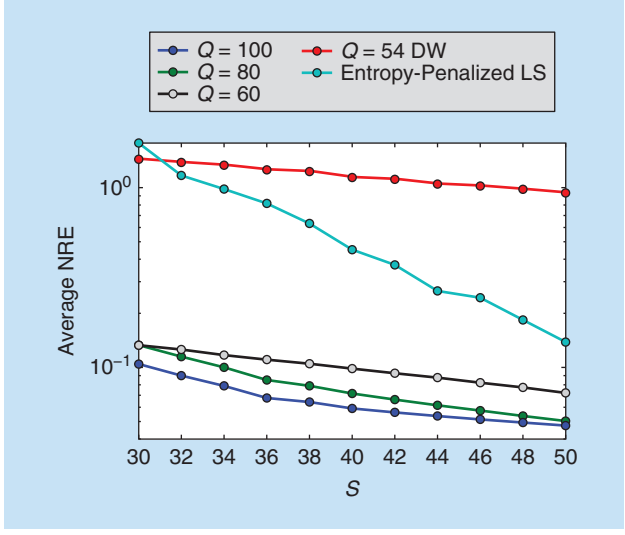
PREDICTION OF LINK COUNTS

Suppose for now that either a learned, or, a suitable prespecified dictionary $\mathbf{B}$ is available and consider predicting the missing link counts. Data-driven learning of dictionaries from historical data will be addressed in the ensuing subsection. Given $\mathbf{R}$ and the link count measurements $\mathbf{y}_t$, contemporary tools developed in the area of CS and semisupervised learning can be used to form $\hat{\mathbf{x}}_t$, which includes estimates for the missing $L - S$ link counts [9], [28], [24]. The spatial regularity of the link counts is captured through the auxiliary weighted graph $\mathcal{G}$ with L vertices, one for each link in the network. The edge weights for all edges in $\mathcal{G}$ are subsumed by the off-diagonal entries of the Gram matrix $\mathbf{G} = [g_{i,j}] := \mathbf{R}\mathbf{R}' \in \mathbb{R}^{L \times L}$, where $(\cdot)'$ denotes transposition. The off-diagonal entries $g_{i,j}$ count the number of OD flows that are common to both links i and j. Main diagonal entries of $\mathbf{G}$ count the number of OD flows that use the corresponding links.

Given a snapshot of incomplete link counts $\mathbf{y}_t$ during the operational phase (where a suitable basis $\mathbf{B}$ is



[FIG1] Training and operational phases of the semisupervised DL approach for link-traffic cartography in [27], where $C_t(B,w)$ denotes the tth summand from the cost in (4) and $k = 1, 2, \ldots$ indicate iterations of the BCD solver.



[FIG2] Link-traffic cartography of Internet-2 data [1]. Comparison of NRE for different values of S [27]. (Figure used with permission from [27].)

available), the sparse basis expansion coefficient vector $\mathbf{w}_t$ is estimated as

$$\hat{\mathbf{w}}_t := \arg\min_{\mathbf{w}_t} \|\mathbf{y}_t - \mathbf{S}_t\mathbf{B}\mathbf{w}_t\|_2^2 + \lambda_w\|\mathbf{w}_t\|_1 + \lambda_g\mathbf{w}_t'\mathbf{B}'\mathbf{L}\mathbf{B}\mathbf{w}_t, \quad (3)$$

where $\mathbf{L} := \mathrm{diag}(\mathbf{G}\mathbf{1}_L) - \mathbf{G}$ denotes the Laplacian matrix of $\mathcal{G}$; $\lambda_w, \lambda_g > 0$ are tunable regularization parameters; and $\mathbf{1}_L$ is the $L \times 1$ vector of all ones. The criterion in (3) consists of an LS error between the observed and postulated link counts, along with two regularizers. The ℓ_1-norm $\|\mathbf{w}_t\|_1$ encourages sparsity in the coefficient vector $\hat{\mathbf{w}}_t$ [24], [28]. With $\mathbf{x}_t := [x_{1,t}, \ldots, x_{L,t}]'$ given by $\mathbf{x}_t = \mathbf{B}\mathbf{w}_t$, the Laplacian regularization can be explicitly written as $\mathbf{w}_t'\mathbf{B}'\mathbf{L}\mathbf{B}\mathbf{w}_t = (1/2)\sum_{i=1}^L \sum_{j=1}^L g_{i,j}(x_{i,t} - x_{j,t})^2$. It is thus apparent that $\mathbf{w}_t'\mathbf{B}'\mathbf{L}\mathbf{B}\mathbf{w}_t$ encourages the link counts to be close if their corresponding vertices are connected in $\mathcal{G}$. Each summand is weighted according to the number of OD flows common to links i and j. Typically adopted for semisupervised learning, such a regularization term encourages $\mathbf{B}\mathbf{w}_t$ to lie on a smooth manifold approximated by $\mathcal{G}$, which constrains how the measured link counts relate to $\mathbf{x}_t$ [9], [45]. It is also common to use normalized variants of the Laplacian instead of $\mathbf{L}$ [33, p. 46].

The cost in (3) is convex but nonsmooth, and customized solvers developed for ℓ_1-norm regularized optimization can be employed here as well, e.g., [28, p. 92]. Once $\hat{\mathbf{w}}_t$ is available, an estimate of the full vector of link counts is readily obtained as $\hat{\mathbf{x}}_t := \mathbf{B}\hat{\mathbf{w}}_t$. It is apparent that the quality of the imputation depends on the chosen $\mathbf{B}$, and DL from historical network data in $\mathcal{T}_S$ is described next.

DATA-DRIVEN DL

In its canonical form, DL seeks a (typically fat) dictionary $\mathbf{B}$ so that training data $\mathcal{T}_L := \{\mathbf{x}_t\}_{t=1}^T$ are well approximated as $\mathbf{x}_t \approx \mathbf{B}\mathbf{w}_t$, $t = 1, \ldots, T$, for some sparse vectors $\mathbf{w}_t$ of expansion coefficients [52]. Standard DL algorithms cannot, however, be directly applied to learn $\mathbf{B}$ since they rely on the entire vector $\mathbf{x}_t$. To learn the dictionary in the training phase using incomplete link counts $\mathcal{T}_S$ instead of $\mathcal{T}_L$, the idea is to capitalize on the structure in $\mathbf{x}_t$, of which $\mathcal{G}$ is an abstraction [27]. To this end, one can adopt a similar cost function as in the operational phase [cf. (3)], yielding the data-driven basis and the corresponding sparse representation

$$\{\hat{\mathbf{W}}, \hat{\mathbf{B}}\} := \arg\min_{\mathbf{W}, \mathbf{B}:\{\|\mathbf{b}_q\|_2 \le 1\}_{q=1}^Q} \sum_{t=1}^T [\|\mathbf{y}_t - \mathbf{S}_t\mathbf{B}\mathbf{w}_t\|_2^2 + \lambda_w\|\mathbf{w}_t\|_1 + \lambda_g\mathbf{w}_t'\mathbf{B}'\mathbf{L}\mathbf{B}\mathbf{w}_t], \quad (4)$$

where $\hat{\mathbf{W}} := [\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_T] \in \mathbb{R}^{Q \times T}$. The constraints $\{\|\mathbf{b}_q\|_2 \le 1\}_{q=1}^Q$ remove the scaling ambiguity in the products $\mathbf{B}\mathbf{w}_t$ and prevent the entries in $\mathbf{B}$ from growing unbounded. Again, the combined regularization terms in (4) promote both sparsity in $\mathbf{w}_t$ through the ℓ_1-norm, and smoothness across the entries of $\mathbf{B}\mathbf{w}_t$ via the Laplacian $\mathbf{L}$. The regularization parameters λ_w and λ_g are typically cross-validated [28, Ch. 7]. Although (4) is nonconvex, a block coordinate-descent (BCD) solver still guarantees convergence to a stationary point [10]. The BCD updates involve solving for $\mathbf{B}$ and $\mathbf{W}$ in an alternating fashion, both doable efficiently via convex programming [27]. Alternatively, the online DL algorithm in [37] offers enhanced scalability by sequentially processing the data in $\mathcal{T}_S$. The training and operational (prediction) phases are summarized in Figure 1, where $C_t(\mathbf{B}, \mathbf{w})$ denotes the tth summand from the cost in (4), and $k = 1, 2, \ldots$ indicate iterations of the BCD solver employed during the training phase.

The explicit need for Laplacian regularization is apparent from (4). Indeed, if measurements from a certain link are not present in $\mathcal{T}_S$, the corresponding row of $\mathbf{B}$ may still be estimated with reasonable accuracy because of the third term in $C_t(\mathbf{B}, \mathbf{w})$. On top of that, it is because of Laplacian regularization that the prediction performance degrades gracefully as the number of

missing entries in $\mathbf{y}_t$ increases; see also Figure 2. It is worth stressing that the time series $\{\mathbf{y}_t\}$ need not be stationary or even contiguous in time. The link-traffic cartography approach described so far can also be adapted to accommodate time-varying network topologies or routing matrices, using a time-dependent Laplacian $\mathbf{L}_t$. A word of caution is due, however, since drastic changes in either $\mathbf{L}_t$ or in the statistical properties of the underlying OD flows $\mathbf{z}_t$, will necessitate retraining $\mathbf{B}$ to attain satisfactory performance. Finally, note that DL techniques incur a complexity at least cubic in the size of the network and are better suited for monitoring of backbone wide-area networks which are typically not very large.

A PROMISING APPROACH IN THIS CONTEXT HAS BEEN THE APPLICATION OF KRIGING, A TOOL FOR SPATIAL PREDICTION POPULAR IN GEOSTATISTICS AND ENVIRONMENTAL SCIENCES.

Next, a numerical test on link count data from the Internet-2 measurement archive [1] is outlined. The data consists of link counts, sampled at five-minute intervals, collected over several weeks. For the purposes of comparison, the training phase consisted of 2,000 time slots, with a random subset of 50 links measured (out of $L = 54$ per time slot). The performance of the learned dictionary is then assessed over the next $T_0 = 2{,}000$ time slots. Each test vector $\mathbf{y}_t$ is constructed by randomly selecting S entries of the full link count vector $\mathbf{x}_t$. The tuning parameters are chosen via cross-validation ($\lambda_s = 0.1$ and $\lambda_g = 10^{-5}$). Figure 2 shows the normalized reconstruction error (NRE), evaluated as $(LT_0)^{-1}\sum_{t=1}^{T_0}\|\mathbf{y}_t - \hat{\mathbf{x}}_t\|^2$ for different values of Q and S. For comparison, the prediction performance with a fixed diffusion wavelet matrix [19] (instead of the data-trained dictionary), as well as that of the entropy-penalized LS method [54] is also shown. The latter approach solves a LS problem augmented with a specific entropy-based regularizer that encourages the traffic volumes at the source/destination pairs to be stochastically independent. The DL-based method markedly outperforms the competing approaches, especially for low values of S. Furthermore, note how performance degrades gracefully as S decreases. Remarkably, the predictions are close to the actual traffic even when using only 30 link counts during the prediction phase.

DELAY CARTOGRAPHY VIA DYNAMIC NETWORK KRIGING

Instead of link counts, consider now the problem of monitoring delays $d_{p,t}$ on a set of multihop paths $p \in \mathcal{P}$, that connect $P := |\mathcal{P}|$ source-destination pairs in an IP network. Path delays are important metrics required by network operators for assessment, planning, and fault diagnosis [18], [33], [46]. However, monitoring path metrics is challenging primarily because P generally grows as the square of the number of nodes in the network. Therefore, at any time t delays can only be measured on a subset of paths $\mathcal{S}_t \subset \mathcal{P}$, collected in the vector $\mathbf{d}_t^s$. Based on the partial current and past measurements $\mathcal{H}_t := \{\mathbf{d}_\tau^s\}_{\tau=1}^t$, delay cartography amounts to predicting the remaining path delays $\mathbf{d}_t^{\bar{s}} := \{d_{p,t}\}_{p\in\mathcal{P}\setminus\mathcal{S}}$.

A promising approach in this context has been the application of kriging, a tool for spatial prediction popular in geostatistics and environmental sciences [22]. A network kriging scheme was developed in [18], which advocates prediction of network-wide path delays using measurements on a fixed subset of paths. The class of linear predictors introduced therein leverages network topology information to model the covariance among path delays. Building on these ideas, a dynamic network kriging approach capable of real-time spatiotemporal delay predictions was put forth in [46]. Specifically, a kriged Kalman filter (KKF) is employed to explicitly capture temporal variations due to queuing delays, while retaining the topology-based spatial kriging predictor. The per-path delay $d_{p,t}$ comprises several independent components due to contributions from each intermediate link and router and is modeled in [46] as

$$d_{p,t} = \chi_{p,t} + \nu_{p,t} + \varepsilon_{p,t}. \tag{5}$$

The queuing delay $\chi_{p,t}$ (collected in $\boldsymbol{\chi}_t \in \mathbb{R}^P$) depends on the traffic and exhibits spatiotemporal correlation, periodic behavior as well as occasional bursts, prompting the following random walk model

$$\boldsymbol{\chi}_t = \boldsymbol{\chi}_{t-1} + \boldsymbol{\eta}_t, \tag{6}$$

where the driving noise $\boldsymbol{\eta}_t$ has zero mean and covariance matrix $\mathbf{C}_\eta$. The second term in (5), collected in the vector $\boldsymbol{\nu}_t$, combines the processing, transmission, and propagation delays and is temporally white but spatially correlated, owing to the overlap between paths. Similar to [18], the correlation between two paths is modeled as being proportional to the number of links they share, so that the covariance matrix $\mathbf{C}_\nu = \alpha\mathbf{U}\mathbf{U}'$, where α is a parameter to be estimated from training path-delay data; while $u_{p,l} = 1$ if path p contains link l, and $u_{p,l} = 0$ otherwise. Finally, the noise term $\epsilon_{p,t}$ is zero mean independent and identically distributed (i.i.d.) with known variance σ^2. Defining the $S \times P$ path selection matrix as in the section "Semisupervised Dictionary Learning for Traffic Maps," the measurement equation can be written as (introduce $\boldsymbol{\nu}_t^s := \mathbf{S}_t\boldsymbol{\nu}_t$ and likewise $\boldsymbol{\epsilon}_t^s$)

$$\mathbf{d}_t^s = \mathbf{S}_t\boldsymbol{\chi}_t + \boldsymbol{\nu}_t^s + \boldsymbol{\epsilon}_t^s. \tag{7}$$

In the absence of $\mathbf{S}_t$, the spatiotemporal model in (6) and (7) is widely employed in geostatistics, where $\boldsymbol{\chi}_t$ is generally referred to as trend, and $\boldsymbol{\nu}_t$ captures the random fluctuations around $\boldsymbol{\chi}_t$; see, e.g. [41]. Similar models have been employed in [31] to describe the dynamics of wireless propagation channels, and in [21] for spatiotemporal random field estimation. For a static selection matrix, i.e., $\mathbf{S}_t := \mathbf{S}$ for all t, the network kriging approach [18] entails the following two-step procedure: Step 1) treat $\boldsymbol{\nu}_t^s$ as noise, and estimate $\boldsymbol{\chi}_t$ using the generalized LS criterion; and Step 2) use the aforesaid estimate to find the linear

minimum mean-square error (LMMSE) estimator (denoted by $\mathbb{E}^*$) for $\boldsymbol{\nu}_t^s$, specifically

$$\mathbb{E}^*[\boldsymbol{\nu}_t^s \,|\, \boldsymbol{\chi}_t] = \mathbf{S}\mathbf{C}_\nu \mathbf{S}'(\mathbf{S}\mathbf{C}_\nu \mathbf{S}' + \sigma^2 \mathbf{I}_S)^{-1}[\mathbf{d}_t^s - \mathbf{S}_t \boldsymbol{\chi}_t]. \tag{8}$$

Recently, a CS-based approach has also been reported for predicting network-wide performance metrics [19]. For instance, diffusion wavelets were utilized in [19] to obtain a compressible representation of the delays and account for spatial and temporal correlations. Although this allows for enhanced prediction accuracy relative to [18], it requires batch processing of measurements, which does not scale well to large networks for real-time operation. Pictorially, the performance of different algorithms can be assessed through the delay maps shown in Figure 3.

The spatiotemporal model set forth earlier can provide a better estimate of $\boldsymbol{\chi}_t$ by efficiently processing both present and past measurements jointly. Towards this end, a Kalman filter is employed in [46], which at time t yields the following update equations:

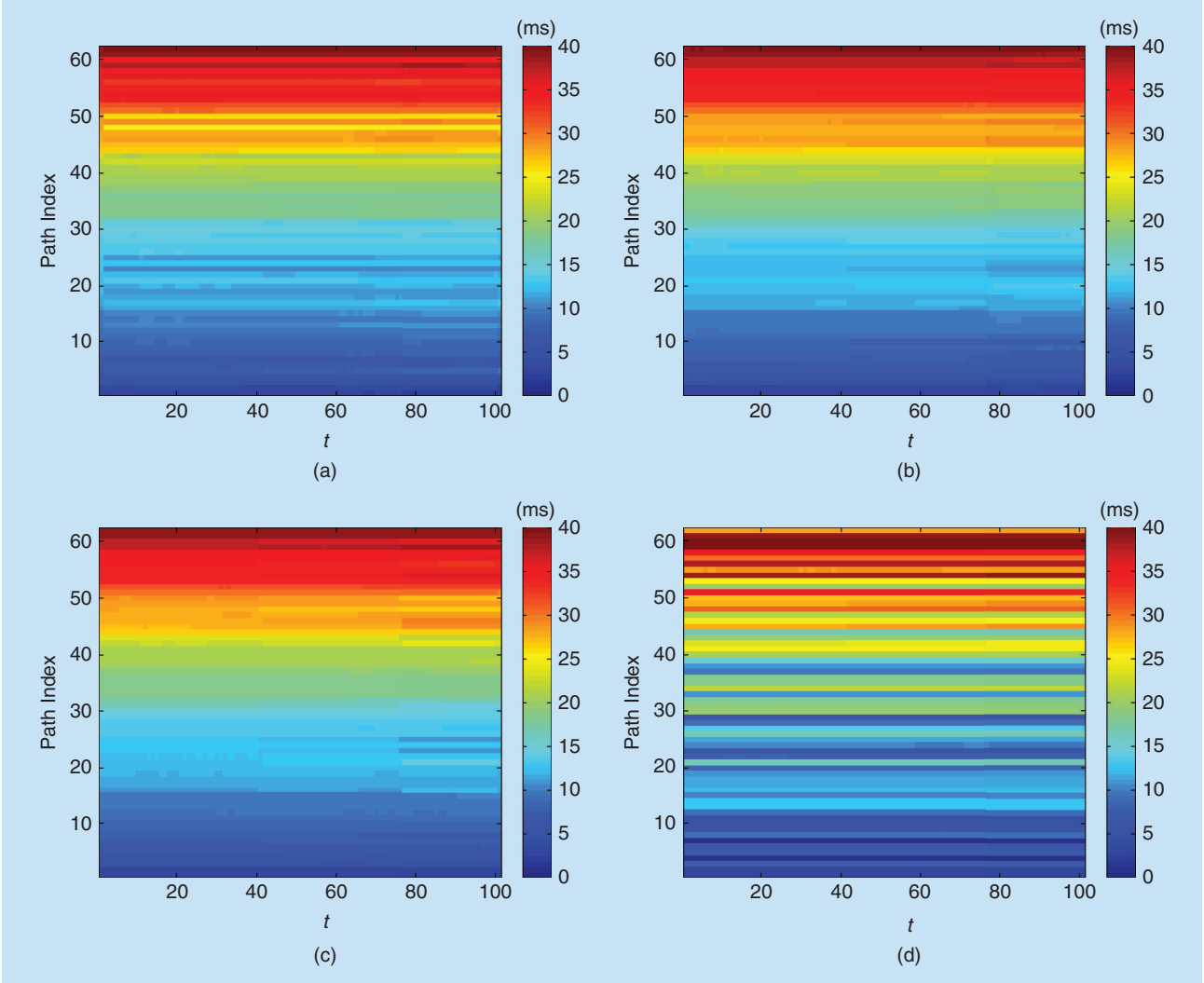
$$\hat{\boldsymbol{\chi}}_t := \mathbb{E}^*[\boldsymbol{\chi}_t | \mathcal{H}_t] = \hat{\boldsymbol{\chi}}_{t-1} + \mathbf{K}_t(\mathbf{d}_t^s - \mathbf{S}_t \hat{\boldsymbol{\chi}}_{t-1})$$
$$\mathbf{M}_t := \mathbb{E}[(\boldsymbol{\chi}_t - \hat{\boldsymbol{\chi}}_t)(\boldsymbol{\chi}_t - \hat{\boldsymbol{\chi}}_t)'] = (\mathbf{I}_P - \mathbf{K}_t \mathbf{S}_t)(\mathbf{M}_{t-1} + \mathbf{C}_\nu),$$

where $\mathbf{K}_t := (\mathbf{M}_{t-1} + \mathbf{C}_\nu)\mathbf{S}_t'[\mathbf{S}_t(\mathbf{C}_\nu + \mathbf{C}_\eta + \mathbf{M}_{t-1})\mathbf{S}_t' + \sigma^2 \mathbf{I}_S]^{-1}$ is the so-termed Kalman gain. The final predictor, referred also as the KKF, is given by

$$\hat{\mathbf{d}}_t^{\bar{s}} := \bar{\mathbf{S}}_t \hat{\boldsymbol{\chi}}_t + \bar{\mathbf{S}}_t \mathbf{C}_\nu \mathbf{S}_t'(\mathbf{S}_t \mathbf{C}_\nu \mathbf{S}_t' + \sigma^2 \mathbf{I}_S)^{-1}[\mathbf{d}_t^s - \mathbf{S}_t \hat{\boldsymbol{\chi}}_t]$$

and the prediction error covariance matrix is

$$\mathbf{M}_t^{\bar{s}} := \mathbb{E}[(\mathbf{d}_t^{\bar{s}} - \hat{\mathbf{d}}_t^{\bar{s}})(\mathbf{d}_t^{\bar{s}} - \hat{\mathbf{d}}_t^{\bar{s}})']$$
$$= \sigma^2 \mathbf{I}_S + \bar{\mathbf{S}}_t \left[(\mathbf{M}_{t-1} + \mathbf{C}_\nu + \mathbf{C}_\eta)^{-1} + \frac{1}{\sigma^2}\mathbf{S}_t'\mathbf{S}_t\right]^{-1} \bar{\mathbf{S}}_t'.$$



[FIG3] True and predicted delay map for 62 paths in the Internet-2 data set [1] over an interval of 100 min. (a) True delays. (b) Network kriging [18]. (c) Difussion wavelets [19]. (d) KKF [46]. Delays of several paths change slightly around $t = 80$, but this change is only discernible from the delay predictions offered by KKF. Delay maps summarize the network state and are useful tools aiding operational decision in network monitoring and control stations [46]. (Figure used with permission from [46].)

Robust PCA as Bilinear Decomposition With Outlier-Sparsity Regularization

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

***Abstract*—Principal component analysis (PCA) is widely used for dimensionality reduction, with well-documented merits in various applications involving high-dimensional data, including computer vision, preference measurement, and bioinformatics. In this context, the fresh look advocated here permeates benefits from variable selection and compressive sampling, to robustify PCA against outliers. A least-trimmed squares estimator of a low-rank bilinear factor analysis model is shown closely related to that obtained from an ℓ_0-(pseudo)norm-regularized criterion encouraging *sparsity* in a matrix explicitly modeling the outliers. This connection suggests robust PCA schemes based on convex relaxation, which lead naturally to a family of robust estimators encompassing Huber's optimal M-class as a special case. Outliers are identified by tuning a regularization parameter, which amounts to controlling sparsity of the outlier matrix along the whole *robustification* path of (group) least-absolute shrinkage and selection operator (Lasso) solutions. Beyond its ties to robust statistics, the developed outlier-aware PCA framework is versatile to accommodate novel and scalable algorithms to: i) track the low-rank signal subspace robustly, as new data are acquired in real time; and ii) determine principal components robustly in (possibly) infinite-dimensional feature spaces. Synthetic and real data tests corroborate the effectiveness of the proposed robust PCA schemes, when used to identify aberrant responses in personality assessment surveys, as well as unveil communities in social networks, and intruders from video surveillance data.**

***Index Terms*—(Group) Lasso, outlier rejection, principal component analysis, robust statistics, sparsity.**

I. Introduction

PRINCIPAL component analysis (PCA) is the workhorse of high-dimensional data analysis and dimensionality reduction, with numerous applications in statistics, engineering, and the biobehavioral sciences; see, e.g., [22]. Nowadays ubiquitous e-commerce sites, the Web, and urban traffic surveillance systems generate massive volumes of data. As a result, the problem of extracting the most informative, yet low-dimensional structure from high-dimensional datasets is of paramount importance [17]. To this end, PCA provides least-squares (LS) optimal linear approximants in $\mathbb{R}^q$ to a data set in $\mathbb{R}^p$, for $q \leq p$. The desired linear subspace is obtained from the q dominant eigenvectors of the sample data covariance matrix [22].

Data obeying postulated low-rank models include also outliers, which are samples not adhering to those nominal models. Unfortunately, LS is known to be very sensitive to outliers [19], [32], and this undesirable property is inherited by PCA as well [22]. Early efforts to robustify PCA have relied on robust estimates of the data covariance matrix; see, e.g., [4]. A fast algorithm for computer vision applications was put forth in [35]. Related approaches are driven from statistical physics [41], and also from M-estimators [8]. Recently, polynomial-time algorithms with remarkable performance guarantees have emerged for low-rank matrix recovery in the presence of sparse—but otherwise arbitrarily large—errors [5], [7]. This pertains to an "idealized robust" PCA setup, since those entries not affected by outliers are assumed error free. Stability in reconstructing the low-rank and sparse matrix components in the presence of "dense" noise have been reported in [40], [44]. A hierarchical Bayesian model was proposed to tackle the aforementioned low-rank plus sparse matrix decomposition problem in [9].

In the present paper, a robust PCA approach is pursued requiring minimal assumptions on the outlier model. A natural least-trimmed squares (LTS) PCA estimator is first shown closely related to an estimator obtained from an ℓ_0-(pseudo)norm-regularized criterion, adopted to fit a low-rank bilinear factor analysis model that explicitly incorporates an unknown *sparse* vector of outliers per datum (Section II). As in compressive sampling [37], efficient (approximate) solvers are obtained in Section III, by surrogating the ℓ_0-norm of the outlier matrix with its closest convex approximant. This leads naturally to an M-type PCA estimator, which subsumes Huber's optimal choice as a special case [13]. Unlike Huber's formulation though, results here are not confined to an outlier contamination model. A tunable parameter controls the sparsity of the estimated matrix, and the number of outliers as a byproduct. Hence, effective data-driven methods to select this parameter are of paramount importance, and systematic approaches are pursued by efficiently exploring the entire *robustifaction* (a.k.a. homotopy) path of (group-) Lasso solutions [17], [43]. In this sense, the method here capitalizes on but *is not limited to* sparse settings where outliers are sporadic, since one can examine all sparsity levels along the robustification path. The outlier-aware generative data model and its sparsity-controlling estimator are quite general, since minor modifications discussed in Section III-D enable robustifiying linear regression [14], dictionary learning [24], [36], and K-means clustering as well [12], [17]. Section IV deals with further modifications for bias

Manuscript received November 07, 2011; revised April 12, 2012; accepted June 08, 2012. Date of publication June 15, 2012; date of current version September 11, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ignacio Santamaria. This work was supported by MURI (AFOSR FA9550-10-1-0567) grant. Part of the paper appeared in the *Proceedings of the Fourty-Fourth Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 7–10, 2010.

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: 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/TSP.2012.2204986

reduction through nonconvex regularization, and automatic determination of the reduced dimension q.

Beyond its ties to robust statistics, the developed outlier-aware PCA framework is versatile to accommodate scalable *robust* algorithms to: i) track the low-rank signal subspace, as new data are acquired in real time (Section V); and ii) determine principal components in (possibly) infinite-dimensional feature spaces, thus robustifying kernel PCA [34], and spectral clustering as well [17, p. 544] (Section VI). The vast literature on *non-robust* subspace tracking algorithms includes [24], [42], and [2]; see also [18] for a first-order algorithm that is robust to outliers and incomplete data. Relative to [18], the online robust (OR-) PCA algorithm of this paper is a second-order method, which minimizes an outlier-aware exponentially-weighted LS estimator of the low-rank factor analysis model. Since the outlier and subspace estimation tasks decouple nicely in OR-PCA, one can readily devise a first-order counterpart when minimal computational loads are at a premium. In terms of performance, online algorithms are known to be markedly faster than their batch alternatives [2], [18], e.g., in the timely context of low-rank matrix completion [29], [30]. While the focus here is not on incomplete data records, extensions to account for missing data are immediate and will be reported elsewhere.

In Section VII, numerical tests with synthetic and real data corroborate the effectiveness of the proposed robust PCA schemes, when used to identify aberrant responses from a questionnaire designed to measure the Big-Five dimensions of personality traits [21], as well as unveil communities in a (social) network of college football teams [15], and intruders from video surveillance data [8]. Concluding remarks are given in Section VIII.

Notation: Bold uppercase (lowercase) letters will denote matrices (column vectors). Operators $(\cdot)'$, $\mathrm{tr}(\cdot)$, $\mathrm{med}(\cdot)$, and $\odot$ will denote transposition, matrix trace, median, and Hadamard product, respectively. Vector $\mathrm{diag}(\mathbf{M})$ collects the diagonal entries of $\mathbf{M}$, whereas the diagonal matrix $\mathrm{diag}(\mathbf{v})$ has the entries of $\mathbf{v}$ on its diagonal. The ℓ_p-norm of $\mathbf{x} \in \mathbb{R}^n$ is $\|\mathbf{x}\|_p := (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ for $p \geq 1$; and $\|\mathbf{M}\|_F := \sqrt{\mathrm{tr}\,(\mathbf{M}\mathbf{M}')}$ is the matrix Frobenius norm. The $n \times n$ identity matrix will be represented by $\mathbf{I}_n$, while $\mathbf{0}_n$ will denote the $n \times 1$ vector of all zeros, and $\mathbf{0}_{n\times m} := \mathbf{0}_n\mathbf{0}_m'$. Similar notation will be adopted for vectors (matrices) of all ones. The i-th vector of the canonical basis in $\mathbb{R}^n$ will be denoted by $\mathbf{b}_{n,i},\ i = 1, \ldots, n$.

II. Robustifying PCA

Consider the standard PCA formulation, in which a set of data $\mathcal{T}_y := \{\mathbf{y}_n\}_{n=1}^N$ in the p-dimensional Euclidean *input* space is given, and the goal is to find the best q-rank ($q \leq p$) linear approximation to the data in $\mathcal{T}_y$; see e.g., [22]. Unless otherwise stated, it is assumed throughout that the value of q is given. One approach to solving this problem, is to adopt a low-rank bilinear (factor analysis) model

$$\mathbf{y}_n = \mathbf{m} + \mathbf{U}\mathbf{s}_n + \mathbf{e}_n, \quad n = 1, \ldots, N \tag{1}$$

where $\mathbf{m} \in \mathbb{R}^p$ is a location (mean) vector; matrix $\mathbf{U} \in \mathbb{R}^{p\times q}$ has orthonormal columns spanning the signal subspace; $\{\mathbf{s}_n\}_{n=1}^N$ are the so-termed *principal components*, and $\{\mathbf{e}_n\}_{n=1}^N$ are zero-mean i.i.d. random errors. The unknown variables in (1) can be collected in $\mathcal{V} := \{\mathbf{m}, \mathbf{U}, \{\mathbf{s}_n\}_{n=1}^N\}$, and they are estimated using the LS criterion as

$$\min_{\mathcal{V}} \sum_{n=1}^N \|\mathbf{y}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n\|_2^2, \quad \text{s. to} \quad \mathbf{U}'\mathbf{U} = \mathbf{I}_q. \tag{2}$$

PCA in (2) is a nonconvex optimization problem due to the bilinear terms $\mathbf{U}\mathbf{s}_n$, yet a global optimum $\hat{\mathcal{V}}$ can be shown to exist; see e.g., [42]. The resulting estimates are $\hat{\mathbf{m}} = \sum_{n=1}^N \frac{\mathbf{y}_n}{N}$ and $\hat{\mathbf{s}}_n = \hat{\mathbf{U}}'(\mathbf{y}_n - \hat{\mathbf{m}}),\ n = 1, \ldots, N$; while $\hat{\mathbf{U}}$ is formed with columns equal to the q-dominant right singular vectors of the $N \times p$ data matrix $\mathbf{Y} := [\mathbf{y}_1, \ldots, \mathbf{y}_N]'$ [17, p. 535]. The principal components (entries of) $\mathbf{s}_n$ are the projections of the centered data points $\{\mathbf{y}_n - \hat{\mathbf{m}}\}_{n=1}^N$ onto the signal subspace. Equivalently, PCA can be formulated based on maximum variance, or, minimum reconstruction error criteria; see e.g., [22].

A. Least-Trimmed Squares PCA

Given training data $\mathcal{T}_x := \{\mathbf{x}_n\}_{n=1}^N$ possibly contaminated with outliers, the goal here is to develop a robust estimator of $\mathcal{V}$ that requires minimal assumptions on the outlier model. Note that there is an explicit notational differentiation between: i) the data in $\mathcal{T}_y$ which adhere to the nominal model (1); and ii) the given data in $\mathcal{T}_x$ that may also contain outliers, i.e., those $\mathbf{x}_n$ not adhering to (1). Building on LTS regression [32], the desired robust estimate $\hat{\mathcal{V}}_{LTS} := \{\hat{\mathbf{m}}, \hat{\mathbf{U}}, \{\hat{\mathbf{s}}_n\}_{n=1}^N\}$ for a prescribed $\nu < N$ can be obtained via the following LTS PCA estimator [cf. (2)]

$$\hat{\mathcal{V}}_{LTS} := \arg\min_{\mathcal{V}} \sum_{n=1}^{\nu} r_{[n]}^2(\mathcal{V}), \quad \text{s. to} \quad \mathbf{U}'\mathbf{U} = \mathbf{I}_q \tag{3}$$

where $r_{[n]}^2(\mathcal{V})$ is the n-th order statistic among the squared residual norms $r_1^2(\mathcal{V}), \ldots, r_N^2(\mathcal{V})$, and $r_n(\mathcal{V}) := \|\mathbf{x}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n\|_2$. The so-termed *coverage* ν determines the breakdown point of the LTS PCA estimator [32], since the $N - \nu$ largest residuals are absent from the estimation criterion in (3). Beyond this universal outlier-rejection property, the LTS-based estimation offers an attractive alternative to robust linear regression due to its high breakdown point and desirable analytical properties, namely $\sqrt{N}$-consistency and asymptotic normality under mild assumptions [32].

Remark 1 (Robust Estimation of the Mean): In most applications of PCA, data in $\mathcal{T}_y$ are typically assumed zero mean. This is without loss of generality, since nonzero-mean training data can always be rendered zero mean, by subtracting the sample mean $\sum_{n=1}^N \frac{\mathbf{y}_n}{N}$ from each $\mathbf{y}_n$. In modeling zero-mean data, the known vector $\mathbf{m}$ in (1) can obviously be neglected. When outliers are present however, data in $\mathcal{T}_x$ are not necessarily zero mean, and it is unwise to center them using the non-robust sample mean estimator which has a breakdown point equal to zero [32]. Towards robustifying PCA, a more sensible approach is to estimate $\mathbf{m}$ robustly, and jointly with $\mathbf{U}$ and the principal components $\{\mathbf{s}_n\}_{n=1}^N$.

Because (3) is a nonconvex optimization problem, a nontrivial issue pertains to the existence of the proposed LTS PCA

estimator, i.e., whether or not (3) attains a minimum. Fortunately, the answer is in the affirmative as asserted next.

Property 1: The LTS PCA estimator is well defined, since (3) has (at least) one solution.

Existence of $\hat{\mathcal{V}}_{LTS}$ can be readily established as follows: i) for each subset of $\mathcal{T}$ with cardinality ν (there are $\binom{N}{\nu}$ such subsets), solve the corresponding PCA problem to obtain a unique candidate estimator per subset; and ii) pick $\hat{\mathcal{V}}_{LTS}$ as the one among all $\binom{N}{\nu}$ candidates with the minimum cost.

Albeit conceptually simple, the solution procedure outlined under Property 1 is combinatorially complex, and thus intractable except for small sample sizes N. Algorithms to obtain approximate LTS solutions in large-scale linear regression problems are available; see e.g., [32].

B. ℓ_0-Norm Regularization for Robustness

Instead of discarding large residuals, the alternative approach here explicitly accounts for outliers in the low-rank data model (1). This becomes possible through the vector variables $\{\mathbf{o}_n\}_{n=1}^N$ one per training datum $\mathbf{x}_n$, which take the value $\mathbf{o}_n \neq \mathbf{0}_p$ whenever datum n is an outlier, and $\mathbf{o}_n = \mathbf{0}_p$ otherwise. Thus, the novel outlier-aware factor analysis model is

$$\mathbf{x}_n = \mathbf{y}_n + \mathbf{o}_n = \mathbf{m} + \mathbf{U}\mathbf{s}_n + \mathbf{e}_n + \mathbf{o}_n, \quad n = 1, \ldots, N \quad (4)$$

where $\mathbf{o}_n$ can be deterministic or random with unspecified distribution. In the *under-determined* linear system of equations (4), both $\mathcal{V}$ as well as the $N \times p$ matrix $\mathbf{O} := [\mathbf{o}_1, \ldots, \mathbf{o}_N]'$ are unknown. The percentage of outliers dictates the degree of *sparsity* (number of zero rows) in $\mathbf{O}$. Sparsity control will prove instrumental in efficiently estimating $\mathbf{O}$, rejecting outliers as a byproduct, and consequently arriving at a *robust* estimator of $\mathcal{V}$. To this end, a natural criterion for controlling outlier sparsity is to seek the estimator [cf. (2)]

$$\{\hat{\mathcal{V}}, \hat{\mathbf{O}}\} = \arg\min_{\mathcal{V},\mathbf{O}} \|\mathbf{X} - \mathbf{1}_N\mathbf{m}' - \mathbf{S}\mathbf{U}' - \mathbf{O}\|_F^2 + \lambda_0\|\mathbf{O}\|_0, \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q \quad (5)$$

where $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]' \in \mathbb{R}^{N\times p}$, $\mathbf{S} := [\mathbf{s}_1, \ldots, \mathbf{s}_N]' \in \mathbb{R}^{N\times q}$, and $\|\mathbf{O}\|_0$ denotes the nonconvex ℓ_0-norm that is equal to the number of nonzero rows of $\mathbf{O}$. Vector (group) sparsity in the rows $\hat{\mathbf{o}}_n$ of $\hat{\mathbf{O}}$ can be directly controlled by tuning the parameter $\lambda_0 \geq 0$.

As with compressive sampling and sparse modeling schemes that rely on the ℓ_0-norm [37], the robust PCA problem (5) is NP-hard [26]. In addition, the sparsity-controlling estimator (5) is intimately related to LTS PCA, as asserted next.

Proposition 1: If $\{\hat{\mathcal{V}}, \hat{\mathbf{O}}\}$ minimizes (5) with λ_0 chosen such that $\|\hat{\mathbf{O}}\|_0 = N - \nu$, then $\hat{\mathcal{V}}_{LTS} = \hat{\mathcal{V}}$.

Proof: Given λ_0 such that $\|\hat{\mathbf{O}}\|_0 = N - \nu$, the goal is to characterize $\hat{\mathcal{V}}$ as well as the positions and values of the nonzero rows of $\hat{\mathbf{O}}$. Because $\|\hat{\mathbf{O}}\|_0 = N - \nu$, the last term in the cost of (5) is constant, hence inconsequential to the minimization. Upon defining $\hat{\mathbf{r}}_n := \mathbf{x}_n - \hat{\mathbf{m}} - \hat{\mathbf{U}}\hat{\mathbf{s}}_n$, the rows of $\hat{\mathbf{O}}$ satisfy

$$\hat{\mathbf{o}}_n = \begin{cases} \mathbf{0}_p, & \|\hat{\mathbf{r}}_n\|_2 \leq \sqrt{\lambda_0} \\ \hat{\mathbf{r}}_n, & \|\hat{\mathbf{r}}_n\|_2 > \sqrt{\lambda_0} \end{cases}, \quad n = 1, \ldots, N. \quad (6)$$

This follows by noting first that (5) is separable across the rows of $\mathbf{O}$. For each $n = 1, \ldots, N$, if $\hat{\mathbf{o}}_n = \mathbf{0}_p$ then the optimal cost becomes $\|\hat{\mathbf{r}}_n - \hat{\mathbf{o}}_n\|_2^2 + \lambda_0\|\hat{\mathbf{o}}_n\|_0 = \|\hat{\mathbf{r}}_n\|_2^2$. If on the other hand $\hat{\mathbf{o}}_n \neq \mathbf{0}_p$, the optimality condition for $\mathbf{o}_n$ yields $\hat{\mathbf{o}}_n = \hat{\mathbf{r}}_n$, and thus the cost reduces to λ_0. In conclusion, for the chosen value of λ_0 it holds that $N - \nu$ squared residuals effectively do not contribute to the cost in (5).

To determine $\hat{\mathcal{V}}$ and the row support of $\hat{\mathbf{O}}$, one alternative is to exhaustively test all $\binom{N}{N-\nu} = \binom{N}{\nu}$ admissible row-support combinations. For each one of these combinations (indexed by j), let $\mathcal{S}_j \subset \{1, \ldots, N\}$ be the index set describing the row support of $\hat{\mathbf{O}}^{(j)}$, i.e., $\hat{\mathbf{o}}_n^{(j)} \neq \mathbf{0}_p$ if and only if $n \in \mathcal{S}_j$; and $|\mathcal{S}_j| = N - \nu$. By virtue of (6), the corresponding candidate $\hat{\mathcal{V}}^{(j)}$ solves $\min_{\mathcal{V}} \sum_{n\in\mathcal{S}_j} r_n^2(\mathcal{V})$ subject to $\mathbf{U}'\mathbf{U} = \mathbf{I}_q$, while $\hat{\mathcal{V}}$ is the one among all $\{\hat{\mathcal{V}}^{(j)}\}$ that yields the least cost. Recognizing the aforementioned solution procedure as the one for LTS PCA outlined under Property 1, it follows that $\hat{\mathcal{V}}_{LTS} = \hat{\mathcal{V}}$. ■

The importance of Proposition 1 is threefold. First, it formally justifies model (4) and its estimator (5) for robust PCA, in light of the well documented merits of LTS [32]. Second, it further solidifies the connection between sparsity-aware learning and robust estimation. Third, problem (5) lends itself naturally to efficient (approximate) solvers based on convex relaxation, the subject dealt with next.

III. Sparsity-Controlling Outlier Rejection

Recall that the row-wise ℓ_2-norm sum $\|\mathbf{B}\|_{2,r} := \sum_{n=1}^N \|\mathbf{b}_n\|_2$ of matrix $\mathbf{B} := [\mathbf{b}_1, \ldots, \mathbf{b}_N]' \in \mathbb{R}^{N\times p}$ is the closest convex approximation of $\|\mathbf{B}\|_0$[37]. This property motivates relaxing problem (5) to

$$\min_{\mathcal{V},\mathbf{O}} \|\mathbf{X} - \mathbf{1}_N\mathbf{m}' - \mathbf{S}\mathbf{U}' - \mathbf{O}\|_F^2 + \lambda_2\|\mathbf{O}\|_{2,r}, \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q. \quad (7)$$

The nondifferentiable ℓ_2-norm regularization term encourages row-wise (vector) sparsity on the estimator of $\mathbf{O}$, a property that has been exploited in diverse problems in engineering, statistics, and machine learning [17]. A noteworthy representative is the group Lasso [43], a popular tool for joint estimation and selection of grouped variables in linear regression. Note that (7) is only nondifferentiable at the origin, which is a minimal restriction.

It is pertinent to ponder on whether problem (7) still has the potential of providing robust estimates $\hat{\mathcal{V}}$ in the presence of outliers. The answer is positive, since (7) is equivalent to an M-type estimator

$$\min_{\mathcal{V}} \sum_{n=1}^N \rho_v(\mathbf{x}_n - \mathbf{m} - \mathbf{U}\mathbf{s}_n), \quad \text{s. to } \mathbf{U}'\mathbf{U} = \mathbf{I}_q \quad (8)$$

where $\rho_v : \mathbb{R}^p \to \mathbb{R}$ is a vector extension to Huber's convex loss function [19]; see also [23], and

$$\rho_v(\mathbf{r}) := \begin{cases} \|\mathbf{r}\|_2^2, & \|\mathbf{r}\|_2 \leq \frac{\lambda_2}{2} \\ \lambda_2\|\mathbf{r}\|_2 - \frac{\lambda_2^2}{4}, & \|\mathbf{r}\|_2 > \frac{\lambda_2}{2}. \end{cases} \quad (9)$$

Towards establishing the equivalence between problems (7) and (8), consider the pair $\{\hat{\mathcal{V}}, \hat{\mathbf{O}}\}$ that solves (7). Assume that $\hat{\mathcal{V}}$ is

Dynamic Structural Equation Models for Tracking Cascades Over Social Networks

Brian Baingana, Gonzalo Mateos and Georgios B. Giannakis
Department of Electrical and Computer Engineering and Digital Technology Center
University of Minnesota, Minneapolis, MN 55455
{baing011,mate0058,georgios}@umn.edu

Abstract

Many real-world processes evolve in cascades over networks, whose topologies are often unobservable and change over time. However, the so-termed adoption times when for instance blogs mention popular news items are typically known, and are implicitly dependent on the underlying network. To infer the network topology, a *dynamic* structural equation model is adopted to capture the relationship between observed adoption times and the unknown edge weights. Assuming a slowly time-varying topology and leveraging the sparse connectivity inherent to social networks, edge weights are estimated by minimizing a sparsity-regularized exponentially-weighted least-squares criterion. To this end, a solver is developed by leveraging (pseudo) real-time sparsity-promoting proximal gradient iterations. Numerical tests with synthetic data and real cascades of online media demonstrate the effectiveness of the novel algorithm in unveiling sparse dynamically-evolving topologies, while accounting for external influences in the adoption times.

1 Introduction

Networks arising in natural and man-made settings provide the backbone for the propagation of *contagions* such as the spread of popular news stories, the adoption of buying trends among consumers, and the spread of infectious diseases [28, 8]. For example, a terrorist attack may be reported within minutes on mainstream news websites. An information cascade emerges because these websites' readership typically includes bloggers who write about the attack as well, influencing their own readers in turn to do the same. Although the times when "nodes" get infected are often observable, the underlying network topologies over which cascades propagate are typically unknown and dynamic. Knowledge of the topology plays a crucial role for several reasons e.g., when social media advertisers select a small set of initiators so that an online campaign can go viral, or when healthcare initiatives wish to infer hidden needle-sharing networks of injecting drug users.

The propagation of a contagion is tantamount to *causal* effects or interactions being exerted among entities such as news portals and blogs, consumers, or people susceptible to being infected with a contagious disease. Acknowledging this viewpoint, *structural equation models* (SEMs) provide a general statistical modeling technique to estimate causal relationships among traits; see e.g., [12, 24]. These directional effects are often not revealed by standard linear models involving symmetric associations between random variables, such as those represented by covariances or correlations, [20], [9], [14]. SEMs are attractive because of their simplicity and ability to capture edge directionalities. They have been widely adopted in many fields, such as economics, psychometrics [22], social sciences [10], and recently in genetics for *static* gene regulatory network inference; see e.g., [5, 18] and references therein. However, SEMs have not been utilized to track the dynamics of causal effects among interacting nodes. In this context, the present paper proposes a *dynamic* SEM to account for time-varying directed networks over which contagions propagate, and describes how node infection times depend on both topological and external influences. Accounting for ex-

ternal influences is well motivated by drawing upon examples from online media, where established news websites depend more on on-site reporting than blog references. External influence data is also useful for model identifiability, and has been shown necessary to resolve directional ambiguities [3].

Supposing the network varies slowly with time, parameters in the proposed dynamic SEM are estimated adaptively by minimizing a sparsity-promoting exponentially-weighted least-squares (LS) criterion (Section 3). To account for the inherently sparse connectivity of social networks, an ℓ_1-norm regularization term that promotes sparsity on the entries of the network adjacency matrix is incorporated in the cost function; see also [6, 15, 1]. A novel algorithm to jointly track the network's adjacency matrix and the weights capturing the level of external influences is developed in Section 3.1, which minimizes the resulting non-differentiable cost function via a proximal-gradient (PG) solver; see e.g., [23, 7, 4]. The resulting dynamic iterative shrinkage-thresholding algorithm (ISTA) is provably convergent, and offers parallel, closed-form, and sparsity-promoting updates per iteration. Numerical tests on synthetic network data demonstrate the superior error performance of the developed algorithms, and highlight their merits when compared to the sparsity-agnostic approach in [27]. Experiments in Section 4 involve real temporal traces of popular global events that propagated on news websites and blogs in 2011 [17]. Interestingly, topologies inferred from cascades associated to the meme "Kim Jong-un" exhibit an abrupt increase in the number of edges following the appointment of the new North Korean ruler.

Related work. Inference of networks using temporal traces of infection events has recently become an active area of research. According to the taxonomy in [13, Ch. 7], this can be viewed as a problem involving inference of *association* networks. Several prior approaches postulate probabilistic models and rely on maximum likelihood estimation (MLE) to infer edge weights as pairwise transmission rates between nodes [26], [21]. However, these methods assume that the network does not change over time. A dynamic algorithm has been recently proposed to infer time-varying diffusion networks by solving an MLE problem via stochastic gradient descent iterations [27]. Although successful experiments on large-scale web data reliably uncover information pathways, the estimator in [27] does not explicitly account for edge sparsity prevalent in social and information networks. Moreover, most prior approaches only attribute node infection events to the network topology, and do not account for the influence of external sources such as a ground crew for a mainstream media website.

Notation. Bold uppercase (lowercase) letters will denote matrices (column vectors), while operators $(\cdot)^\top$, $\lambda_{\max}(\cdot)$, and $\mathrm{diag}(\cdot)$ will stand for matrix transposition, spectral radius, and diagonal matrix, respectively. The $N \times N$ identity matrix will be represented by $\mathbf{I}_N$, while $\mathbf{0}_N$ will denote the $N \times 1$ vector of all zeros, and $\mathbf{0}_{N\times P} := \mathbf{0}_N\mathbf{0}_P^\top$. The ℓ_p and Frobenius norms will be denoted by $\|\cdot\|_p$, and $\|\cdot\|_F$, respectively.

2 Network Model and Problem Statement

Consider a dynamic network with N nodes observed over time intervals $t = 1, \ldots, T$, whose abstraction is a graph with topology described by an unknown, time-varying, and weighted adjacency matrix $\mathbf{A}^t \in \mathbb{R}^{N\times N}$. Entry (i, j) of $\mathbf{A}^t$ (henceforth denoted by a_{ij}^t) is nonzero only if a directed edge connects nodes i and j (pointing from j to i) during the time interval t. As a result, one in general has $a_{ij}^t \neq a_{ji}^t$, i.e., matrix $\mathbf{A}^t$ is generally non-symmetric, which is suitable to model directed networks. Note that the model tacitly assumes that the network topology remains fixed during any given time interval t, but can change across time intervals.

Suppose C contagions propagate over the network, and the infection time of node i by contagion c is denoted by y_{ic}^t. In online media, y_{ic}^t can be obtained by recording the time when website i mentions news item c. For uninfected nodes at slot t, y_{ic}^t is set to an arbitrarily large number. Assume that the susceptibility x_{ic} of node i to external (non-topological) infection by contagion c is known and time invariant over the observation interval. In the web context, x_{ic} can be set to the search engine rank of website i with respect to (w.r.t.) keywords associated with c.

The infection time of node i during interval t is modeled according to the following *dynamic* structural equation model (SEM)

$$y_{ic}^t = \sum_{j\neq i} a_{ij}^t y_{jc}^t + b_{ii}^t x_{ic} + e_{ic}^t \tag{1}$$

where b_{ii}^t captures the time-varying level of influence of external sources, and e_{ic}^t accounts for measurement errors and unmodeled dynamics. It follows from (1) that if $a_{ij}^t \neq 0$, then y_{ic}^t is affected by the value of y_{jc}^t. With $\mathbf{B}^t := \text{diag}(b_{11}, \ldots, b_{NN})$, collecting observations for the entire network and all C contagions yields the dynamic matrix SEM

$$\mathbf{Y}^t = \mathbf{A}^t\mathbf{Y}^t + \mathbf{B}^t\mathbf{X} + \mathbf{E}^t \tag{2}$$

where $\mathbf{Y}^t := [y_{ic}^t]$, $\mathbf{X} := [x_{ic}]$, and $\mathbf{E}^t := [e_{ic}^t]$ are all $N \times C$ matrices. A single network topology $\mathbf{A}^t$ is adopted for all contagions, which is suitable e.g., when information cascades are formed around a common meme or trending (news) topic in the Internet; see also the data in Section 4.

Given $\{\mathbf{Y}^t\}_{t=1}^T$ and $\mathbf{X}$ adhering to (2), the goal is to track the underlying network topology $\{\mathbf{A}^t\}_{t=1}^T$ and the effect of external influences $\{\mathbf{B}^t\}_{t=1}^T$. To this end, the algorithm developed in the next section assumes slow time variation of the network topology and leverages the inherent sparsity of edges that is typical of social networks.

3 Topology Tracking Algorithm

To estimate $\{\mathbf{A}^t, \mathbf{B}^t\}$ in a static setting, one can solve the following regularized LS problem

$$\begin{aligned} \{\hat{\mathbf{A}}, \hat{\mathbf{B}}\} = \arg\min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}\sum_{t=1}^{T} \|\mathbf{Y}^t - \mathbf{A}\mathbf{Y}^t - \mathbf{B}\mathbf{X}\|_F^2 + \lambda\|\mathbf{A}\|_1 \\ \text{s. to} \quad & a_{ii} = 0,\ b_{ij} = 0,\ \forall i \neq j \end{aligned} \tag{3}$$

where $\|\mathbf{A}\|_1 := \sum_{i,j} |a_{ij}|$ is a sparsity-promoting regularization, and $\lambda > 0$ controls the sparsity level of $\hat{\mathbf{A}}$. Absence of a self-loop at node i is enforced by the constraint $a_{ii} = 0$, while having $b_{ij} = 0,\ \forall i \neq j$, ensures that $\hat{\mathbf{B}}$ is diagonal as in (2). The batch estimator (3) yields single estimates $\{\hat{\mathbf{A}}, \hat{\mathbf{B}}\}$ that best fit the data $\{\mathbf{Y}^t\}_{t=1}^T$ and $\mathbf{X}$ over the whole measurement horizon $t = 1, \ldots, T$, and as such (3) neglects potential network variations across time intervals.

Exponentially-weighted LS estimator. In practice, measurements are typically acquired in a sequential manner and the sheer scale of social networks calls for estimation algorithms with minimum storage requirements. Recursive solvers enabling sequential inference of the underlying dynamic network topology are thus preferred.

For $t = 1, \ldots, T$, consider the sparsity-regularized exponentially-weighted LS estimator (EWLSE)

$$\begin{aligned} \{\hat{\mathbf{A}}^t, \hat{\mathbf{B}}^t\} = \arg\min_{\mathbf{A},\mathbf{B}} \quad & \frac{1}{2}\sum_{\tau=1}^{t} \beta^{t-\tau}\|\mathbf{Y}^\tau - \mathbf{A}\mathbf{Y}^\tau - \mathbf{B}\mathbf{X}\|_F^2 + \lambda_t\|\mathbf{A}\|_1 \\ \text{s. to} \quad & a_{ii} = 0,\ b_{ij} = 0,\ \forall i \neq j \end{aligned} \tag{4}$$

where $\beta \in (0, 1]$ is the forgetting factor that forms estimates $\{\hat{\mathbf{A}}^t, \hat{\mathbf{B}}^t\}$ using all measurements acquired until time t. Whenever $\beta < 1$, past data are exponentially discarded thus enabling tracking of dynamic network topologies. The first summand in the cost corresponds to an exponentially-weighted moving average (EWMA) of the squared model residuals norms. The EWMA can be seen as an average modulated by a sliding window of equivalent length $1/(1-\beta)$, which clearly grows as $\beta \to 1$. In the infinite-memory setting whereby $\beta = 1$, (4) boils down to the batch estimator (3).

Selection of the (possibly time-varying) tuning parameter λ_t is an important aspect of regularization methods such as (4), because λ_t controls the sparsity level of the inferred network and how its structure may change over time. For sufficiently large values of λ_t one obtains the trivial solution $\hat{\mathbf{A}}^t = \mathbf{O}_{N\times N}$, while increasingly more dense graphs are obtained as $\lambda_t \to 0$. An increasing λ_t will be required for accurate estimation over extended time-horizons, since for $\beta \approx 1$ the norm of the LS term in (4) grows due to noise accumulation. This way the effect of the regularization term will be downweighted unless one increases λ_t at a suitable rate, for instance proportional to $\sqrt{\sigma^2 t}$ as suggested by large deviation tail bounds when the errors are assumed $e_{ic}^t \sim \mathcal{N}(0, \sigma^2)$, and the problem dimensions N, C, T are sufficiently large [20, 19, 1]. In the topology tracking experiments of Section 4, a time-invariant value of λ is adopted and typically chosen via trial and error to optimize the performance. This is justified since smaller values of β are selected for tracking