Online proximal gradient for learning graphs from streaming signals

Rasoul Shafipour and Gonzalo Mateos
Dept. of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA

***Abstract*—We leverage proximal gradient iterations to develop an online graph learning algorithm from streaming network data. Our goal is to track the (possibly) time-varying network topology, and effect memory and computational savings by processing the data on-the-fly as they are acquired. The setup entails observations modeled as stationary graph signals generated by local diffusion dynamics on the unknown network. Moreover, we may have a priori information on the presence or absence of a few edges as in the link prediction problem. The stationarity assumption implies that the observations' covariance matrix and the so-called graph shift operator (GSO – a matrix encoding the graph topology) commute under mild requirements. This motivates formulating the topology inference task as an inverse problem, whereby one searches for a (e.g., sparse) GSO that is structurally admissible and approximately commutes with the observations' empirical covariance matrix. For streaming data said covariance can be updated recursively, and we show online proximal gradient iterations can be brought to bear to efficiently track the time-varying solution of the inverse problem with quantifiable guarantees. Specifically, we derive conditions under which the GSO recovery cost is strongly convex and use this property to prove that the online algorithm converges to within a neighborhood of the optimal time-varying batch solution. Preliminary numerical tests illustrate the effectiveness of the proposed graph learning approach in adapting to streaming information and tracking changes in the sought dynamic network.**

***Index Terms*—Network topology inference, graph signal processing, proximal gradient algorithm, online optimization.**

I. Introduction and preliminiarie

Network data supported on the vertices of a graph $\mathcal{G}$ are nowadays ubiquitous across disciplines spanning engineering as well as social and the bio-behavioral sciences; see e.g., [15, Ch. 1]. Such data can be represented as graph signals, namely vectors indexed by the nodes of $\mathcal{G}$. In this context, the goal of graph signal processing (GSP) is to develop information processing algorithms that fruitfully exploit the relational structure of said network data [22]. However, oftentimes $\mathcal{G}$ is not readily available and a first key step is to use nodal observations to identify the underlying network structure (or a useful graph model that facilitates signal representations and downstream learning tasks); see [9], [19] for recent tutorials on graph learning and [15, Ch. 7] for a statistical treatment.

Consider a weighted undirected graph $\mathcal{G}$ consisting of a node set $\mathcal{N}$ of cardinality N, and symmetric adjacency matrix $\mathbf{A}$ with entry $A_{ij} = A_{ji} \neq 0$ denoting the edge weight between node i and node j. We assume that $\mathcal{G}$ contains no self-loops; i.e., $A_{ii} = 0$. One could generically define a *graph-shift operator* (GSO) $\mathbf{S} \in \mathbb{R}^{N \times N}$ as any matrix capturing the same sparsity pattern as $\mathbf{A}$ on its off-diagonal entries. Beyond $\mathbf{A}$, common choices for $\mathbf{S}$ are the combinatorial Laplacian $\mathbf{L} := \text{diag}(\mathbf{A1}) - \mathbf{A}$ as well as their normalized counterparts [22]. Henceforth we focus on $\mathbf{S} = \mathbf{A}$ and aim to recover the adjacency matrix of the unknown graph $\mathcal{G}$. Other GSOs can be accommodated in a similar fashion.

Next, we present an online framework to estimate sparse graphs that explain the structure of a class of streaming random signals. At some time instant, let $\mathbf{y} = [y_1, ..., y_N]^T \in \mathbb{R}^N$ be a zero-mean graph signal in which the ith element y_i denotes the signal value at node i of an *unknown graph* $\mathcal{G}$ with shift operator $\mathbf{S}$. Further consider a zero-mean white signal $\mathbf{x}$. We state that the graph $\mathbf{S}$ represents the structure of the signal $\mathbf{y} \in \mathbb{R}^N$ if there exists a diffusion process in the GSO $\mathbf{S}$ that produces the signal $\mathbf{y}$ from the input signal $\mathbf{x}$ [26], that is

$$\mathbf{y} = \alpha_0 \prod_{l=1}^{\infty} (\mathbf{I} - \alpha_l \mathbf{S})\mathbf{x} = \sum_{l=0}^{\infty} \beta_l \mathbf{S}^l \mathbf{x}. \quad (1)$$

Under the assumption that $\mathbf{C_x} = \mathbf{I}$ (identity matrix), (1) is equivalent to the *stationarity* of $\mathbf{y}$ in $\mathbf{S}$; see e.g., [18, Def. 1], [24], [12]. The justification to say that $\mathbf{S}$ represents the structure of $\mathbf{y}$ is that we can think of the edges of $\mathcal{G}$, i.e. the non-zero entries in $\mathbf{S}$, as direct (one-hop) relations between the elements of the signal. The diffusion in (1) modifies the original correlation by inducing indirect (multi-hop) relations.

In this context, our goal is to recover $\mathbf{S}$ from a set of *streaming* stationary random signals $\mathcal{Y} := \{\mathbf{y}_1, \ldots, \mathbf{y}_t, \mathbf{y}_{t+1}, \ldots\}$, each of them adhering to the generative model in (1). Unlike [28] but similar to link prediction problems [15, Ch. 7.2], [30], here we rely on a priori knowledge about the presence (or absence) of a few edges; conceivably leading to simpler algorithmic updates and better recovery performance. We may learn about edge status via limited questionnaires and experiments, or, we could perform edge screening prior to topology inference [1]. Stationarity implies that the covariance matrix $\mathbf{C_y}$ of the observations in $\mathcal{Y}$ commutes with $\mathbf{S}$ under mild requirements; see e.g., [18] and Section II. This motivates formulating the topology inference task as an inverse problem, whereby one searches for a (e.g., sparse) $\mathbf{S}$ that is structurally admissible and approximately commutes with the observations' empirical covariance matrix. For streaming data said covariance can be updated recursively, and in Section III we show online proximal gradient iterations can be brought to bear to efficiently track the time-varying solution of the inverse problem with quantifiable (non-asymptotic) guarantees. The algorithm and results of this paper are valid even for dynamic networks, i.e., if the GSO $\mathbf{S}_t$ in (1) is (slowly) time-varying.

Work in this paper was supported by the NSF award CCF-1750428. Author emails: {rshafipo,gmateosb}@ur.rochester.edu.

Relation to prior work. Early topology inference approaches can be traced to the field of (undirected) graphical model selection [15, Ch. 7], [9], [19]. Under Gaussianity assumptions, this line of work has well-documented connections with covariance selection [7] and sparse precision matrix estimation [10], [11], [16], as well as neighborhood-based sparse linear regression [21]. Recent GSP-based network inference frameworks postulate that the network exists as a latent underlying structure, and that observations are generated as a result of a network process defined in such a graph [8], [14], [20], [23], [26], [34]. Different from [6], [8], [14], [25] that infer structure from signals assumed to be smooth over the sought undirected graph, here the measurements are assumed related to the graph via filtering [cf. (1)]. Few works have recently explored this approach by identifying a symmetric GSO given its eigenvectors, either assuming that the input is white [23], [26] – equivalently implying $\mathbf{y}$ is graph stationary [12], [18], [24]; or, colored [31], [32]. Unlike prior *online* algorithms developed based on the aforementioned graph spectral domain design [28], [30], here we estimate the (possibly time-varying) GSO directly and derive quantifiable recovery guarantees. While we assume that the graph signals are stationary, the online scheme in [35] uses observations from a Laplacian-based, continuous-time graph process. Relative to [33] that relies on a single-pole graph filter [13], the filter structure underlying (1) can be arbitrary, but the focus here is on learning undirected graphs. Online proximal gradient methods were adopted for graph inference under dynamic structural equation models [2], but lacking a formal performance analysis. The recovery guarantees in Section III are adapted from the results in [17], obtained therein for online sparse subspace clustering.

II. Graph learning under stationarity

We consider topology inference from stationary signals (1) whereby a small number of edges might be known a priori. To state the problem, we consider the symmetric GSO $\mathbf{S}$ associated with the undirected graph $\mathcal{G}$. Upon defining the vector of coefficients $\mathbf{h} := [h_0, \ldots, h_{L-1}]^T \in \mathbb{R}^L$ and the *symmetric* graph filter $\mathbf{H} := \sum_{l=0}^{L-1} h_l \mathbf{S}^l \in \mathbb{R}^{N \times N}$ [13], [22], [27], the Cayley-Hamilton theorem asserts that the model in (1) boils down to

$$\mathbf{y} = \left(\sum_{l=0}^{L-1} h_l \mathbf{S}^l \right) \mathbf{x} = \mathbf{H}\mathbf{x}, \qquad (2)$$

for some particular $\mathbf{h}$ and $L \leq N$. Note that L specifies the dependency range of the diffusion on the neighhbors.

We first start with the offline setting [26], where the covariance matrix of $\mathbf{y} = \mathbf{H}\mathbf{x}$ is (recall $\mathbf{C}_\mathbf{x} = \mathbf{I}$)

$$\mathbf{C}_\mathbf{y} := \mathbb{E}[\mathbf{y}\mathbf{y}^T] = \mathbb{E}[\mathbf{H}\mathbf{x}(\mathbf{H}\mathbf{x})^T] = \mathbf{H}\mathbb{E}[\mathbf{x}\mathbf{x}^T]\mathbf{H} = \mathbf{H}^2. \quad (3)$$

We used the symmetry of $\mathbf{H}$ to obtain the third equality, as $\mathbf{H}$ is a polynomial in the symmetric GSO $\mathbf{S}$. Using the spectral decomposition of $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ to express the filter as $\mathbf{H} = \sum_{l=0}^{L-1} h_l (\mathbf{V}\mathbf{\Lambda}\mathbf{V}^T)^l = \mathbf{V}(\sum_{l=0}^{L-1} h_l \mathbf{\Lambda}^l)\mathbf{V}^T$, we can diagonalize the covariance matrix in (3) as

$$\mathbf{C}_\mathbf{y} = \mathbf{V} \left(\sum_{l=0}^{L-1} h_l \mathbf{\Lambda}^l \right)^2 \mathbf{V}^T. \qquad (4)$$

Such a covariance expression is the requirement for a graph signal to be stationary in $\mathbf{S}$ [18, Def. 2.b]. Remarkably, if $\mathbf{y}$ is graph stationary, or equivalently if $\mathbf{C}_\mathbf{x} = \mathbf{I}$, (4) shows that the *eigenvectors* of the shift $\mathbf{S}$, the filter $\mathbf{H}$, and the covariance $\mathbf{C}_\mathbf{y}$ are *all the same*. Thus given observations $\{\mathbf{y}_t\}_{t=1}^T$, [26] advocates: (i) forming the *sample covariance* $\hat{\mathbf{C}}_\mathbf{y} = \frac{1}{T}\sum_{t=1}^T \mathbf{y}_t \mathbf{y}_t^T$ and extracting its eigenvectors $\hat{\mathbf{V}}$ as spectral templates of $\mathcal{G}$; and recover $\mathbf{S}$ that is optimal in some sense by estimating its eigenvalues $\mathbf{\Lambda}$. Namely, one solves

$$\mathbf{S}^* := \underset{\mathbf{\Lambda}, \mathbf{S} \in \mathcal{S}}{\operatorname{argmin}} \; f(\mathbf{S}), \quad \text{subject to } \; d(\mathbf{S}, \hat{\mathbf{V}}\mathbf{\Lambda}\hat{\mathbf{V}}^T) \leq \epsilon \qquad (5)$$

which is a convex optimization problem provided $f(\mathbf{S})$ and matrix distance $d(\cdot,\cdot)$ are convex, while ϵ is a tuning parameter. The form of the distance $d(\cdot,\cdot)$ depends on the particular application. For instance, if $\|\mathbf{S} - \hat{\mathbf{V}}\mathbf{\Lambda}\hat{\mathbf{V}}^T\|_F$ is chosen the focus is more on the similarities across the entries of the shifts, while $\|\mathbf{S} - \hat{\mathbf{V}}\mathbf{\Lambda}\hat{\mathbf{V}}^T\|_2$ focuses on their spectrum.

In this paper we propose a different formulation from (5). To that end, observe that stationarity of $\mathbf{y}$ implies $\mathbf{C}_\mathbf{y}\mathbf{S} = \mathbf{S}\mathbf{C}_\mathbf{y}$, forcing the covariance $\mathbf{C}_\mathbf{y}$ to be a polynomial in $\mathbf{S}$ as in (4). This commutation identity holds under the pragmatic assumption that all the eigenvalues of $\mathbf{S}$ are simple and $\sum_{l=0}^{L-1} h_l \lambda_i^l \neq 0$, for $i = 1, \cdots, N$. Since one can only estimate $\hat{\mathbf{C}}_\mathbf{y}$ from the available data, our idea to recover the GSO is to solve [cf. (5)]

$$\mathbf{S}^* := \underset{\mathbf{S} \in \mathcal{S}}{\operatorname{argmin}} \; f(\mathbf{S}), \quad \text{subject to } \; d(\mathbf{S}\hat{\mathbf{C}}_\mathbf{y}, \hat{\mathbf{C}}_\mathbf{y}\mathbf{S}) \leq \epsilon. \quad (6)$$

The inverse problem (6) is intuitive. One searches for an admissible $\mathbf{S} \in \mathcal{S}$ that approximately commutes with the observations' empirical covariance matrix $\hat{\mathbf{C}}_\mathbf{y}$, and which is optimal in the sense specified by f. While the feasible set in (5) is lower dimensional (one only goes after N eigenvalues), the novel formulation (6) circumvents computation of eigenvectors. More importantly, as we show in Section III it offers favorable structure to invoke a proximal gradient solver with convergence guarantees, even in an online setting.

In closing, we note that the formulations (5) and (6) entail a general class of network topology inference problems parametrized by the choices described next.

Optimality criteria and admissibility constraints. The selection of cost function $f(\mathbf{S})$ allows to incorporate physical characteristics of the desired graph into the formulation, while being consistent with the estimated covariance $\hat{\mathbf{C}}_\mathbf{y}$. For instance, the matrix (pseudo-)norm $f(\mathbf{S}) = \|\mathbf{S}\|_0$ which counts the number of nonzero entries in $\mathbf{S}$ can be used to minimize the number of edges towards identifying sparse graphs (e.g., of direct relations among signal elements); $f(\mathbf{S}) = \|\mathbf{S}\|_1 = \sum_{ij} |S_{ij}|$ is a convex proxy for the aforementioned edge cardinality function *and henceforth the criterion of choice*. Alternatively, the Frobenius norm $f(\mathbf{S}) = \|\mathbf{S}\|_F = (\sum_{ij} S_{ij}^2)^{1/2}$ can be adopted to minimize the energy of the edges in the graph, or $f(\mathbf{S}) = \|\mathbf{S}\|_\infty = \max_{ij}|S_{ij}|$ can be chosen to obtain shifts $\mathbf{S}$ associated with graphs of uniformly low edge weights. This can be meaningful when identifying graphs subject to capacity constraints.

Furthermore, one can impose constraints to ensure the GSO $\mathbf{S}$ is structurally admissible and incorporate a priori knowledge about $\mathbf{S}$. Namely, if we let $\mathbf{S} = \mathbf{A}$ represent the adjacency matrix of an undirected graph with non-negative weights and no self-loops, we can write $\mathcal{S}=\mathcal{S}_{\mathbf{A}}$ as

$$\mathbf{S}\in\mathcal{S}_{\mathrm{A}}:=\{\mathbf{S}\,|\,S_{ij}\geq 0,\mathbf{S}^T=\mathbf{S},S_{ii}=0,\textstyle\sum_j S_{j1}=1\},\quad(7)$$

where the last condition fixes the scale of the admissible graphs by setting the weighted degree of the first node to 1. This also rules out the trivial solution $\mathbf{S}=\mathbf{0}$. Suppose we have additional prior information on the presence (or absence) of a few edges, or, even their corresponding weights. In that case, we can drop the constraint $\sum_j S_{j1}=1$ and instead add $S_{ij}=s_{ij}$, for vertex pairs (i,j) in the set $\Omega\subset\mathcal{N}\times\mathcal{N}$ of observed edge weights s_{ij}. Accordingly, we can rewrite the set of admissible adjacency matrices as

$$\mathcal{S}_{\mathrm{A}}^{\mathrm{P}}:=\{\mathbf{S}\,|\,S_{ij}\geq 0,\mathbf{S}^T=\mathbf{S},S_{ii}=0,S_{ij}=s_{ij},\ (i,j)\in\Omega\}.\quad(8)$$

Moving forward, we mostly focus on online learning of sparse graphs (thus $f(\mathbf{S})=\|\mathbf{S}\|_1$), which belong to $\mathcal{S}_{\mathrm{A}}$ or $\mathcal{S}_{\mathrm{A}}^{\mathrm{P}}$. The distance $d(\cdot,\cdot)$ in (6) is chosen to be the Frobenius norm. Other scenarios can be accommodated using proper modifications.

III. Online learning via proximalgradient method

Inspired by [17], here we develop an online (first-order) proximal gradient algorithm to estimate $\mathbf{S}$ from streaming data in $\mathcal{Y}:=\{\mathbf{y}_1,\ldots,\mathbf{y}_t,\mathbf{y}_{t+1},\ldots\}$. To make the problem amenable to this optimization method, we dualize the constraint $\|\mathbf{S}\hat{\mathbf{C}}_{\mathbf{y},t}-\hat{\mathbf{C}}_{\mathbf{y},t}\mathbf{S}\|_F^2\leq\epsilon$ in (6) and write the composite, time-varying optimization

$$\mathbf{S}_t^\star\in\operatorname*{argmin}_{\mathbf{S}\in\mathcal{S}}\quad F_t(\mathbf{S}):=\|\mathbf{S}\|_1+\frac{\lambda}{2}\|\mathbf{S}\hat{\mathbf{C}}_{\mathbf{y},t}-\hat{\mathbf{C}}_{\mathbf{y},t}\mathbf{S}\|_F^2 \qquad (\mathcal{P}_t)$$
$$:=f(\mathbf{S})+g_t(\mathbf{S}).$$

In writing $\hat{\mathbf{C}}_{\mathbf{y},t}$ we make explicit that the covariance matrix is estimated with all signals acquired by time t, $g_t(\cdot)$ is convex and L-smooth [i.e., $\nabla g_t(\cdot)$ is L-Lipschitz continuous] and $\lambda>0$ is a tuning parameter.

The solution $\mathbf{S}_t^\star$ of $(\mathcal{P}_t)$ is the batch network estimate at time t. However, solving $(\mathcal{P}_t)$ to optimality might not be feasible within the time interval of signal acquisition. If $\mathcal{G}$ is dynamic it may not be even prudent to obtain $\mathbf{S}_t^\star$ with high precision (hence incurring high delay and computational cost), since at time $t+1$ a new datum arrives and the solution $\mathbf{S}_{t+1}^\star$ may deviate significantly. These reasons motivate devising an efficient online and recursive algorithm to solve the time-varying optimization problem $(\mathcal{P}_t)$. Our approach entails two steps per time instant $t=1,2,\ldots$, where we: (i) recursively update the observations' covariance matrix $\hat{\mathbf{C}}_{\mathbf{y},t}$ in $\mathcal{O}(N^2)$ complexity; and (ii) take a single step of the graph learning algorithm developed in this section to solve $(\mathcal{P}_t)$ efficiently. Step (i) is straightforward, and the sample covariance $\hat{\mathbf{C}}_{\mathbf{y},t}$ is updated once $\mathbf{y}_{t+1}$ becomes available as follows

$$\hat{\mathbf{C}}_{\mathbf{y},t+1}=\frac{1}{t+1}\left(t\hat{\mathbf{C}}_{\mathbf{y},t}+\mathbf{y}_{t+1}\mathbf{y}_{t+1}^T\right).\qquad(9)$$

To solve $(\mathcal{P}_t)$ online, we bring to bear the proximal gradient-descent algorithm that is well-suited for ℓ_1-norm minimization problems and provides solid convergence guarantees [4]. To that end, first notice that the gradient of $g_t(\mathbf{S})$ in $(\mathcal{P}_t)$ with respect to $\mathbf{S}$ has the form

$$\nabla g_t(\mathbf{S})=\lambda\big[(\mathbf{S}\hat{\mathbf{C}}_{\mathbf{y},t}-\hat{\mathbf{C}}_{\mathbf{y},t}\mathbf{S})\hat{\mathbf{C}}_{\mathbf{y},t}-\hat{\mathbf{C}}_{\mathbf{y},t}(\mathbf{S}\hat{\mathbf{C}}_{\mathbf{y},t}-\hat{\mathbf{C}}_{\mathbf{y},t}\mathbf{S})\big],\qquad(10)$$

which is Lipschitz continuous with constant $M_t=4\lambda\sigma_{\max}^2(\hat{\mathbf{C}}_t)$, where $\sigma_{\max}(\cdot)$ stands for the largest singular value of its matrix argument. Next, introduce the proximal operator of a function h, convex set $\mathcal{S}$ and matrix $\mathbf{B}\in\mathbb{R}^{N\times N}$ as

$$\mathbf{Z}:=\operatorname{prox}_{\alpha h,\mathcal{S}}(\mathbf{B}):=\operatorname*{argmin}_{\mathbf{X}\in\mathcal{S}}\left[h(\mathbf{X})+\frac{1}{2\alpha}\|\mathbf{X}-\mathbf{B}\|_F^2\right].\qquad(11)$$

With these definitions, the proximal gradient-descent updates with fixed step size $\gamma<\frac{2}{M_t}$ to solve the batch problem $(\mathcal{P}_t)$ at time t are given by ($k=1,2,\ldots$ denote iterations)

$$\mathbf{S}_{k+1}:=\operatorname{prox}_{\gamma\|\cdot\|_1,\mathcal{S}}\big(\mathbf{S}_k-\gamma\nabla g_t(\mathbf{S}_k)\big).\qquad(12)$$

As $k\to\infty$ the sequence of iterates (12) converge to a minimizer $\mathbf{S}_t^\star$ [cf. $(\mathcal{P}_t)$]; see e.g., [3]. Moreover, $F_t(\mathbf{S}_k)-F_t(\mathbf{S}_t^\star)\to 0$ due to the continuity of $F_t(\cdot)$.

For the specific case of sparse graph learning with partial connectivity information, i.e., $\mathcal{S}=\mathcal{S}_A^P$ [cf. (8)] and $h(\cdot)=\|\cdot\|_1$, the proximal operator $\mathbf{Z}$ in (11) has entries given by

$$Z_{ij}=\begin{cases}0, & i=j\\ s_{ij}, & (i,j)\in\Omega\\ \max(0,B_{ij}-\alpha), & \text{otherwise.}\end{cases}\qquad(13)$$

Without a priori information on edge status, i.e., when $\mathcal{S}=\mathcal{S}_A$ [cf. (7)], $\mathbf{Z}$ can be computed in similar efficient fashion modulo an extra projection step onto the $N-1$ dimensional probability simplex to enforce $\sum_j S_{j1}=1$. Said projection can be computed using the method in [5, Algorithm 1].

Building on the insights gained from the batch solver in (12), we let iterations $k=1,2,\ldots$ coincide with the instants t of data acquisition to arrive at an online algorithm. This way, at time t we run a single iteration of (12) to update $\mathbf{S}_t$ before the new datum $\mathbf{y}_{t+1}$ arrives at time $t+1$. Specifically, the online proximal gradient descent algorithm takes the form

$$\mathbf{S}_{t+1}:=\operatorname{prox}_{\gamma_t\|\cdot\|_1,\mathcal{S}}\big(\mathbf{S}_t-\gamma_t\nabla g_t(\mathbf{S}_t)\big),\qquad(14)$$

where the step size γ_t is chosen such that $\gamma_t<\frac{2}{M_t}=\frac{2}{4\lambda.\sigma_{\max}^2(\hat{\mathbf{C}}_t)}$. Recall that the gradient $\nabla g_t(\mathbf{S}_t)$ is given by (10), and it is a function of the updated covariance matrix $\hat{\mathbf{C}}_{\mathbf{y},t+1}$ [cf. (9)]. The proximal operator for the ℓ_1-norm entails the pointwise nonlinearity in (13). If the signals arrive faster, one can create a buffer and perform each iteration of the algorithm on a $\hat{\mathbf{C}}_{t+1}$ updated with a sliding window of all newly observed signals. On the other hand, for a slower arrival rate additional proximal gradient iterations would likely improve recovery performance; see also Remark 1.

The key difference between the batch algorithm (12) and its online counterpart (14) is the variability of g_t per iteration in the latter. Ideally, we would like the online algorithm (14) to closely track the sequence of minimizers $\{\mathbf{S}_t^*\}$ for large enough t, something which we corroborate numerically in Section IV. Following closely the analysis in [17], we derive recovery (i.e., tracking error) bounds $\|\mathbf{S}_t-\mathbf{S}_t^*\|_F$ under the pragmatic assumption that g_t is strongly convex and $\mathbf{S}_t^*$ is the unique minimizer of $(\mathcal{P}_t)$, for each t. Before stating the main result in Theorem 1, the following proposition offers a condition for strong convexity of g_t[1].

Proposition 1 *Let set $\mathcal{D}$ contain the indices of vec$(\mathbf{S})$ corresponding to the diagonal entries of $\mathbf{S}$; i.e., $\mathcal{D}:=\{N(i-1)+i \mid i \in \{1,\cdots,N\}\}$, and $\mathcal{D}^c$ be the complement of $\mathcal{D}$. Define $\mathbf{\Psi}_t := \hat{\mathbf{C}}_{\mathbf{y},t} \otimes \mathbf{I}_N - \mathbf{I}_N \otimes \hat{\mathbf{C}}_{\mathbf{y},t}$, where $\otimes$ denotes the Khatri-Rao product and $\mathbf{I}_N$ is the $N \times N$ identity matrix. If $\mathbf{\Psi}_{t,\mathcal{D}^c}$ (submatrix of $\mathbf{\Psi}_t$ that contains columns indexed by the set $\mathcal{D}^c$) is full column rank, then $g_t(\mathbf{S})$ in $(\mathcal{P}_t)$ is strongly convex with constant $m_t > 0$ being the smallest (nonzero) singular value of $\mathbf{\Psi}_{t,\mathcal{D}^c}$.*

In extensive simulations involving several real-world graphs, we have observed that $\mathbf{\Psi}_{t,\mathcal{D}^c}$ is typically full column rank and thus g_t is strongly convex. Under the strong convexity assumption, we have the following (non-asymptotic) performance guarantee.

Theorem 1 *Let $\mu_t = \left\|\mathbf{S}_{t+1}^* - \mathbf{S}_t^*\right\|_F$ capture the variability of the underlying graph. If g_t in $(\mathcal{P}_t)$ is strongly convex with constant m_t, then for all $t \geq 1$ we have*

$$\|\mathbf{S}_t - \mathbf{S}_t^*\|_F \leq \tilde{L}_{t-1}\left(\|\mathbf{S}_0 - \mathbf{S}_0^*\|_F + \sum_{\tau=0}^{t-1} \frac{\mu_\tau}{\tilde{L}_\tau}\right), \qquad (15)$$

where $L_t = \max\{|1-\gamma_t m_t|, |1-\gamma_t M_t|\}, \tilde{L}_t = \prod_{\tau=0}^{t} L_\tau$.

As expected, Theorem 1 asserts that the higher the variability in the underlying graph, the higher the recovery performance penalty. Even if the graph $\mathcal{G}$ (and hence the GSO) is time-invariant, then μ_t will be non-zero especially for small t since the solution $\mathbf{S}_t^*$ may fluctuate due to lack of data. In the next section, we use computer simulations to corroborate the performance of the online graph learning algorithm. But before moving on, a couple remarks are in order.

Remark 1 In case of taking n_t algorithmic iterations instead of one per time step, we can simply modify $\tilde{L}_t = \prod_{\tau=0}^{t} L_\tau^{n_\tau}$ in Theorem 1 to use the result stated in (15). The alternative would be to redefine g_t as $g_{\lfloor t/n_t \rfloor}$.

Remark 2 If g_t is not strongly convex, we can derive dynamic regret bounds similar to the one in [17]. Alternatively, we can add a Tikhonov regularization term $\frac{\lambda_r}{2}\|\mathbf{S}\|_F^2$ to make $(\mathcal{P}_t)$ strongly convex and still benefit from the result in Theorem 1. However, this would incur an error of the form $\|\mathbf{S}_{r,t} - \mathbf{S}_t^*\| \leq \left\|\mathbf{S}_{r,t} - \mathbf{S}_{r,t}^*\right\| + \left\|\mathbf{S}_{r,t}^* - \mathbf{S}_t^*\right\|$ in the minimizer, where $\mathbf{S}_{r,t}$ and $\mathbf{S}_{r,t}^*$ are the regularized tracking sequence and regularized minimizer sequence, respectively.

IV. Preliminary Numerical Test

In this section we assess the performance of the online graph learning using the proposed proximal gradient descent iterations (14). To that end, we consider the social network of Zachary's karate club [36] represented by a graph $\mathcal{G}$ consisting of $N = 34$ nodes or members of the club and 78 undirected edges symbolizing friendships among them; see Fig. 1-(a). We seek to infer this graph from the observation of diffusion processes that are synthetically generated via graph filtering as in (2). For the graph shift $\mathbf{S} = \mathbf{A}$, we consider a second-order filter $\mathbf{H} = \sum_{l=0}^{2} h_l \mathbf{S}^l$, where the coefficients $\{h_l\}$ are drawn uniformly from $[0, 1]$. We assume that we (randomly) know one of the 78 edges as a priori information and aim to infer the rest of the edges. At each time step, 10 synthetic signals $\{\mathbf{y}^{(p)}\}$ are generated through diffusion process $\mathbf{H}$ where the entries of the inputs $\{\mathbf{x}^{(p)}\}$ are drawn independently from the normal Gaussian distribution to make the observations stationary. In the online case, upon sensing 10 signals at each time step, we first update the sample covariance $\hat{\mathbf{C}}_{\mathbf{y},t}$ and then carry out 10 iterations of the proximal gradient descent. Also, to examine the tracking capability of the online estimator, after 5000 time steps, we remove 10% of the existing edges and add the same number of edges elsewhere. This would affect the graph filter $\mathbf{H}$ accordingly.

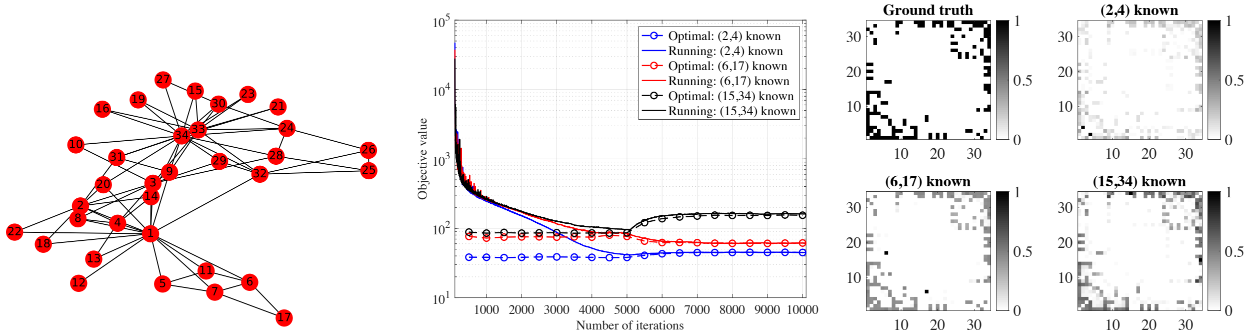
To corroborate the assumption in Theorem 1, it is worth mentioning that throughout the process we observed that $\mathbf{\Psi}_{t,\mathcal{D}^c}$ was full column rank and thus the cost in $(\mathcal{P}_t)$ was strongly convex; see Proposition 1. Fig. 1-(b) depicts the running objective value $F_t(\mathbf{S}_t)$ [cf. $(\mathcal{P}_t)$] averaged over 10 experiments as a function of the time steps and the a priori knowledge – 3 randomly picked edges. We also superimpose Fig. 1-(b) with the optimal objective value $F_t(\mathbf{S}_t^\star)$ at each time step. First, we notice that the objective value trajectory converges to a region above the optimal trajectory. Also, we observe that after 5000 iterations, the performance deteriorates at first due to the sudden change of the network structure, but after observing large enough number of new samples, the online algorithm can adapt and track the batch estimator as well. This demonstrates the effectiveness of the developed online algorithm when it comes to adapting to network perturbations.

Finally, we study the quality of the online learned graph $\mathbf{S}_t$ at iteration 5000. Fig. 1-(c) depicts the heat maps of the ground-truth and inferred adjacency matrices for different a priori information. Although the procedure results in a slight gap between $F_t(\mathbf{S}_t^\star)$ and $F_t(\mathbf{S}_t)$, it still reveals the underlying support of $\mathbf{A}$ with reasonable accuracy. Interestingly, we notice that an edge with lower betweenness centrality [e.g., $(6, 17)$ and $(15, 34)$ compared to $(2, 4)$] is better to know as a priori knowledge in the topology inference; see also [30].

V. Conclusion

We studied the problem of identifying the topology of an undirected network from streaming observations of stationary signals diffused on the unobservable graph. The stationarity assumption implies that the observations' covariance matrix and the GSO commute under mild requirements. This motivates formulating the topology inference task as an inverse problem, whereby one searches for a (e.g., sparse) GSO that is structurally admissible and approximately commutes with the observations' empirical covariance matrix. For streaming data said covariance can be updated recursively, and we show online proximal gradient iterations can be brought to bear to efficiently track the time-varying solution of the inverse problem with quantifiable recovery guarantees.

[1] The proofs of Proposition 1 and Theorem 1 are omitted here due to lack of space, but can be found in [29].



(a) (b) (c)

Fig. 1. (a) Zachary's karate club graph with $N = 34$ nodes. (b) Evolution of the objective values for the online and batch estimators in inferring a karate club. We perform 10 steps of the proposed online algorithm upon sensing each 10 new signals. (c) True adjacency matrix and corresponding estimates with different a priori information on the connectivities attained after 5000 time steps.

References

[1] T. Ahmed and W. U. Bajwa, "Correlation-based ultrahigh-dimensional variable screening," in *IEEE Intl. Wrksp. Computat. Advances Multi-Sensor Adaptive Process. (CAMSAP)*, Dec 2017, pp. 1–5.

[2] B. Baingana, G. Mateos, and G. B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *IEEE J. Sel. Topics Signal Process.*, vol. 8, pp. 563–575, Aug. 2014.

[3] H. H. Bauschke, P. L. Combettes *et al.*, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2011, vol. 408.

[4] A. Beck, *First-order methods in optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2018.

[5] Y. Chen and X. Ye, "Projection onto a simplex," *arXiv preprint arXiv:1101.6081*, 2011.

[6] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, "Learning sparse graphs under smoothness prior," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, New Orleans, LA, Mar. 5-9, 2017, pp. 6508–6512.

[7] A. P. Dempster, "Covariance selection," *Biometrics*, vol. 28, no. 1, pp. 157–175, 1972.

[8] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, Aug. 2016.

[9] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, 2019.

[10] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, Sep. 2017.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[12] B. Girault, "Stationary graph signals using an isometric graph translation," in *European Signal Process. Conf. (EUSIPCO)*, Aug 2015, pp. 1516–1520.

[13] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274–288, Jan. 2017.

[14] V. Kalofolias, "How to learn a graph from smooth signals," in *Intl. Conf. Artif. Intel. Stat. (AISTATS)*, 2016, pp. 920–929.

[15] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. New York, NY: Springer, 2009.

[16] B. M. Lake and J. B. Tenenbaum, "Discovering structure by learning sparse graph," in *Annual Cognitive Sc. Conf.*, 2010, pp. 778 – 783.

[17] L. Madden, S. Becker, and E. Dall'Anese, "Online sparse subspace clustering," in *2019 IEEE Data Science Workshop (DSW)*, June 2019, pp. 248–252.

[18] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5911–5926, Aug. 2017.

[19] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, 2019.

[20] J. Mei and J. M. F. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 2077–2092, Apr. 2017.

[21] N. Meinshausen and P. Buhlmann, "High-dimensional graphs and variable selection with the lasso," *Ann. Stat.*, vol. 34, pp. 1436–1462, 2006.

[22] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[23] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 3, pp. 481–496, 2018.

[24] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, Jul. 2017.

[25] M. G. Rabbat, "Inferring sparse graphs from smooth signals with theoretical guarantees," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, New Orleans, LA, Mar. 5-9, 2017, pp. 6533–6537.

[26] S. Segarra, A. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, Aug. 2017.

[27] S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind identification of graph filters with sparse inputs," in *IEEE Intl. Wrksp. Computat. Advances Multi-Sensor Adaptive Process. (CAMSAP)*, 2015, pp. 449–452.

[28] R. Shafipour, A. Hashemi, G. Mateos, and H. Vikalo, "Online topology inference from streaming stationary graph signals," in *IEEE Data Science Wrkshp. (DSW)*, June 2019, pp. 140–144.

[29] R. Shafipour and G. Mateos, "Online topology inference from streaming stationary graph signals with partial connectivity information," *Algorithms*, 2020; see also arXiv:2007.03653 [eess.SP].

[30] R. Shafipour and G. Mateos, "Online network topology inference with partial connectivity information," in *IEEE Intl. Wrksp. Computat. Advances Multi-Sensor Adaptive Process. (CAMSAP)*, Dec 2019, pp. 226–230.

[31] R. Shafipour, S. Segarra, A. G. Marques, and G. Mateos, "Network topology inference from non-stationary graph signals," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, New Orleans, LA, Mar. 5-9, 2017.

[32] ——, "Identifying the topology of undirected networks from diffused non-stationary graph signals," *IEEE Open J. Signal Process.*, 2020, (submitted; see also arXiv:1801.03862 [eess.SP]).

[33] Y. Shen, B. Baingana, and G. B. Giannakis, "Tensor decompositions for identifying directed graph topologies and tracking dynamic networks," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3675–3687, Jul. 2017.

[34] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 484–499, Sept 2017.

[35] S. Vlaski, H. P. Maretić, R. Nassif, P. Frossard, and A. H. Sayed, "Online graph learning from sequential data," in *IEEE Data Science Wrkshp. (DSW)*, 2018, pp. 190–194.

[36] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.