ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10096392

DUAL-BASED ONLINE LEARNING OF DYNAMIC NETWORK TOPOLOGIES

Seyed Saman Saboksayr and Gonzalo Mateos

Department of Electrical and Computer Engineering, University of Rochester, USA

ABSTRACT

We investigate online network topology identification from smooth nodal observations acquired in a streaming fashion. Different from non-adaptive batch solutions, our distinctive goal is to track the (possibly) dynamic adjacency matrix with affordable memory and computational costs by processing signal snapshots online. To this end, we leverage and truncate dual-based proximal gradient (DPG) iterations to solve a composite smoothness-regularized, time-varying inverse problem. Numerical tests with synthetic and real electrocorticography data showcase the effectiveness of the novel lightweight iterations when it comes to tracking slowly-varying network connectivity. We also show that the online DPG algorithm converges faster than a primal-based baseline of comparable complexity. Aligned with reproducible research practices, we share the code developed to produce all figures included in this paper.

***Index Terms*—** Dynamic network, topology identification, dual-based proximal gradient, online algorithm, signal smoothness.

1. INTRODUCTION

The intertwined fields of graph signal processing (GSP) [1–3], graph representation learning [4], and machine learning on graphs [5, 6] have recently emerged with the common goal of extracting actionable information from graph-structured (i.e., relational) data describing networks [7, Ch. 1]. In some real-world applications such as network neuroscience [8], said relational structures may not be explicitly available [9, 10]. Therefore, depending on the end goal the first step may be to recover the latent network topology to reveal patterns in the complex system under study; or, to rather learn graph representations that can facilitate downstream tasks such as classification; see e.g, [11]. Recognizing that many of these networks are also *dynamic* and that graph datasets grow every day in volume and complexity, there is a pressing need to develop efficient *online* topology identification algorithms to process network data streams [12].

The term *network topology inference* encompasses a broad class of approaches to identify an underlying graph using data. Inference refers to the process of searching for a graph (represented via some graph shift operator) that is optimal for the task at hand. Data often come in the form of nodal observations (also known as graph signals in the GSP parlance), but partial edge status information is not uncommon [7, Ch. 7]. Optimality notions and constraints are typically driven by statistical priors, physical laws, or explainability goals, all of which translate to models binding the observations to the sought graph. A common probabilistic prior is to model network observations via undirected Gaussian graphical models. In this case, the topology inference problem boils down to graphical model selection [13–15]. Other recent approaches instead assume that graph signals are e.g., stationary and possibly generated by linear network diffusion [16, 17], or, smooth with respect to the graph (i.e., they are sparse in the graph spectral domain) [18–22]. Please refer to [9, 10, 12] for recent surveys of network topology inference advances.

Proposed approach and contributions in context. In this paper, we propose an online algorithm to track the topology of (possibly dynamic) undirected graphs using streaming, smooth signals (Section 2 outlines the required GSP background and formally states the topology inference problem). The rationale behind the adoption of a smoothness prior has been well-documented; see e.g., [19]. Moreover, exploiting this cardinal property of network data is central to graph-based learning tasks including semi-supervised learning and denoising via variation minimization [1, 7], just to name a few. Starting from the widely adopted, but batch, graph learning formulation in [18, 21], in Section 4 we develop a novel online dual-based proximal-gradient (DPG) algorithm that refines graph estimates sequentially-in-time. Capitalizing on favorable *dual domain* structure of the smoothness-regularized inverse problem (Section 3) [23], we leverage and truncate proximal-gradient (PG) iterations in [24] to optimize the resulting time-varying cost function adaptively. Computer simulations in Section 5 showcase the effectiveness of the novel lightweight iterations when it comes to tracking slowly-varying network connectivity. The numerical experiments involve both synthetic and real electrocorticography data [7, 25]. In the interest of reproducible research, the code used to generate the figures in this paper is made publicly available.

Related work. Noteworthy dynamic network topology inference algorithms exploiting a smoothness prior include [20, 26–28]. The online PG method put forth in [26] operates in the primal domain. Yet, because of its quadratic complexity in the number of graph nodes, it is the most natural baseline to assess the proposed method's performance in tracking the optimal solution. Recently, a model-independent framework for learning time-varying graphs from online data was proposed in [28]. Iterations therein can be accelerated by virtue of a prediction-correction strategy, but specific instances may incur cubic complexity. Unlike the novel online DPG algorithm of this paper (as well as [26, 28]), recovery of dynamic network topology in [20, 27] is accomplished via non-recursive batch processing; hence incurring a computational cost and memory footprint that grow linearly with the number of temporal samples. For other models besides signal smoothness; see e.g., [12, 17, 29].

2. PRELIMINARIES AND PROBLEM STATEMENT

Consider a network graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$, where $\mathcal{V} = \{1, \ldots, N\}$ is the set of vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the edges. Because $\mathcal{G}$ is assumed to be undirected, elements of $\mathcal{E}$ are unordered pairs of vertices in $\mathcal{V}$. The symmetric adjacency matrix $\mathbf{W} \in \mathbb{R}_+^{N \times N}$ collects the edge weights, and $W_{ij} = 0$ for $(i, j) \notin \mathcal{E}$. Also, $W_{ii} = 0$, $\forall\, i \in \mathcal{V}$, since we exclude self-loops. An equivalent algebraic

Work in this paper was supported by the NSF awards CCF-1750428 and CCF-1934962.

Authorized licensed use limited to: IEEE Xplore. Downloaded on June 04,2023 at 07:22:11 UTC from IEEE Xplore. Restrictions apply.

representation of $\mathcal{G}$'s topology is given by the graph Laplacian $\mathbf{L} := \text{diag}\,(\mathbf{d}) - \mathbf{W}$, where $\mathbf{d} = \mathbf{W}\mathbf{1}$ collects the vertex degree sequence.

Graph signal smoothness. We acquire graph signal observations $\mathbf{x} = [x_1, \ldots, x_N]^\top \in \mathbb{R}^N$, where x_i is the value measured at $i \in \mathcal{V}$. The Dirichlet energy or total variation (TV) of $\mathbf{x}$ with respect to $\mathbf{L}$ is

$$\text{TV}(\mathbf{x}) := \mathbf{x}^\top \mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i \neq j} W_{ij}\,(x_i - x_j)^2\,. \tag{1}$$

Measure $\text{TV}(\mathbf{x}) \in [0, \lambda_{\max}]$ quantifies the smoothness of graph signals supported on $\mathcal{G}$ [1,30], where $\lambda_{\max}$ is the largest eigenvalue of the positive semidefinite matrix $\mathbf{L}$. We say a signal is smooth (or low-pass bandlimited) if it has a small total variation. From (1) it follows the lower bound $\text{TV}(\mathbf{x}) = 0$ is attained by constant signals.

Problem statement. Given a dataset $\mathcal{X} := \{\mathbf{x}_t\}_{t=1}^T$ of network measurements, we want to estimate an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ so that the signals in $\mathcal{X}$ are smooth on $\mathcal{G}$. We also consider tracking dynamic networks with slowly time-varying weight matrix $\mathbf{W}_t$, $t = 1, 2, \ldots$ ($\mathcal{V}$ remains fixed); see Section 4.

2.1. Topology identification from smooth signals

We start by briefly reviewing the *batch* topology inference framework in [18,21], that we build on in the remainder of the paper. Consider arranging the graph signals in $\mathcal{X}$ as columns of the data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$. Let $\bar{\mathbf{x}}_i^\top \in \mathbb{R}^{1 \times T}$ be the ith row of $\mathbf{X}$, which collects all T observations at vertex i. Define the pairwise node dissimilarity matrix $\mathbf{E} \in \mathbb{R}_+^{N \times N}$, where $E_{ij} := \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_2^2$, $i, j \in \mathcal{V}$. With these definitions, it is established in [18] that the aggregate signal smoothness measure over $\mathcal{X}$ can be expressed as

$$\sum_{t=1}^T \text{TV}(\mathbf{x}_t) = \text{trace}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) = \frac{1}{2}\|\mathbf{W} \circ \mathbf{E}\|_1, \tag{2}$$

where $\circ$ stands for the Hadamard product. Notice how TV minimization as criterion for graph topology inference inherently induces sparsity on $\mathcal{E}$. This is because the model preferentially selects edges (i, j) with smaller pairwise nodal dissimilarities E_{ij} [cf. the weighted ℓ_1-norm in (2)]. Exploiting this intuitive relationship between signal smoothness and edge sparsity, a fairly general graph-learning framework was put forth in [18]. The idea therein is to solve the following convex inverse problem

$$\min_{\mathbf{W}} \left\{ \|\mathbf{W} \circ \mathbf{E}\|_1 - \alpha \mathbf{1}^\top \log(\mathbf{W}\mathbf{1}) + \frac{\beta}{2}\|\mathbf{W}\|_F^2 \right\}, \tag{3}$$
$$\text{s. to} \quad \text{diag}(\mathbf{W}) = \mathbf{0},\ W_{ij} = W_{ji} \geq 0,\ i \neq j$$

where $\alpha, \beta > 0$ are tunable regularization parameters. In most applications, it is undesirable to have isolated (null degree) vertices in the learned graph. Hence, the logarithmic barrier imposed over the nodal degree sequence $\mathbf{d} = \mathbf{W}\mathbf{1}$. The Frobenius-norm regularization on adjacency matrix $\mathbf{W}$ offers a handle on the level of sparsity (through β). Indeed, the sparsest solution of (3) will be attained by setting $\beta = 0$.

Formulating (3) as a search over adjacency matrices offers noteworthy computational complexity benefits. Unlike inverse problems whose optimization variable is a graph Laplacian $\mathbf{L}$ [19], the constraints in (3) (i.e. null diagonal, symmetry and non-negativity) are all separable across the entries W_{ij}. This favorable structure has enabled a host of efficient *batch solvers* derived based on primal-dual (PD) iterations [18], the PG method [26], and the linearized alternating-directions method of multipliers (ADMM) [31]. Next, we present a dual-based algorithm we proposed in [23], which comes with convergence rate guarantees in the batch setting while it is amenable to an online scheme in streaming scenarios.

3. PROXIMAL GRADIENT IN THE DUAL DOMAIN

Because $\mathbf{W}$ is symmetric and has a null diagonal, the free decision variables in (3) are effectively the, say, lower-triangular elements $[\mathbf{W}]_{ij}$, $j < i$. Thus, we henceforth work with the compact vector $\mathbf{w} := \text{vec}[\text{triu}[\mathbf{W}]] \in \mathbb{R}_+^{N(N-1)/2}$, were we have adopted convenient Matlab notation. To impose the non-negativity constraints over edge weights, we augment the cost with a penalty function $\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} = 0$ if $\mathbf{w} \succeq \mathbf{0}$, else $\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} = \infty$ [18]. Given these definitions, one can rewrite the objective in (3) as the unconstrained, non-differentiable problem

$$\min_{\mathbf{w}} \Big\{ \underbrace{\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} + 2\mathbf{w}^\top \mathbf{e} + \beta\|\mathbf{w}\|_2^2}_{:= f(\mathbf{w})} - \underbrace{\alpha \mathbf{1}^\top \log(\mathbf{S}\mathbf{w})}_{:= -g(\mathbf{S}\mathbf{w})} \Big\}, \tag{4}$$

where $\mathbf{e} := \text{vec}[\text{triu}[\mathbf{E}]]$ and $\mathbf{S} \in \{0, 1\}^{N \times N(N-1)/2}$ maps edge weights to nodal degrees, i.e., $\mathbf{d} = \mathbf{S}\mathbf{w}$. The non-smooth function $f(\mathbf{w}) := \mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} + 2\mathbf{w}^\top \mathbf{e} + \beta\|\mathbf{w}\|_2^2$ is strongly convex with strong convexity parameter 2β, while $g(\mathbf{w}) := -\alpha \mathbf{1}^\top \log(\mathbf{w})$ is a (strictly) convex function for all $\mathbf{w} \succ \mathbf{0}$. Given the aforementioned properties of f and g, one can establish that the composite problem (4) has a unique optimal solution $\mathbf{w}^\star$; see e.g., [24, 31]. To tackle (4) efficiently we can adopt a dual-based PG algorithm introduced in [24], which is capable of solving general non-smooth, strictly convex optimization problems of the form $\min_{\mathbf{w}} \{f(\mathbf{w}) + g(\mathbf{S}\mathbf{w})\}$. In the remainder of this section we briefly review the DPG-based graph learning framework proposed in [23].

Using a standard variable-splitting technique we recast (4) as

$$\min_{\mathbf{w}, \mathbf{d}} \{f(\mathbf{w}) + g(\mathbf{d})\}, \quad \text{s. to } \mathbf{d} = \mathbf{S}\mathbf{w}. \tag{5}$$

Attaching Lagrange multipliers $\boldsymbol{\lambda} \in \mathbb{R}^N$ to the equality constraints and minimizing the Lagrangian function $\mathcal{L}(\mathbf{w}, \mathbf{d}, \boldsymbol{\lambda}) = f(\mathbf{w}) + g(\mathbf{d}) - \langle \boldsymbol{\lambda}, \mathbf{S}\mathbf{w} - \mathbf{d} \rangle$ with respect to the primal variables $\{\mathbf{w}, \mathbf{d}\}$, we obtain the (minimization form) dual problem

$$\min_{\boldsymbol{\lambda}} \{F(\boldsymbol{\lambda}) + G(\boldsymbol{\lambda})\}, \text{where} \tag{6}$$

$$F(\boldsymbol{\lambda}) := \max_{\mathbf{w}} \left\{ \langle \mathbf{S}^\top \boldsymbol{\lambda}, \mathbf{w} \rangle - f(\mathbf{w}) \right\}, \tag{7}$$

$$G(\boldsymbol{\lambda}) := \max_{\mathbf{d}} \{\langle -\boldsymbol{\lambda}, \mathbf{d} \rangle - g(\mathbf{d})\}. \tag{8}$$

Because f is strongly convex, one can derive useful smoothness properties for its Fenchel conjugate F. Indeed, it follows that the gradient $\nabla F(\boldsymbol{\lambda})$ is Lipschitz continuous with constant $L := \frac{N-1}{\beta}$; see [23, Lemma 1]. Recognizing this additional structure of (6), the PG algorithm [24] (say of the ISTA type) becomes an attractive choice to solve the the dual problem. Accordingly, when applied to (6) the PG method yields the following iterations (initialized as $\boldsymbol{\lambda}_0 \in \mathbb{R}^N$, henceforth $k = 1, 2, \ldots$ denotes the iteration index)

$$\boldsymbol{\lambda}_k = \textbf{prox}_{L^{-1}G}\left(\boldsymbol{\lambda}_{k-1} - \frac{1}{L}\nabla F(\boldsymbol{\lambda}_{k-1}) \right), \tag{9}$$

where the proximal operator of a proper, lower semi-continuous convex function h is (see e.g., [32])

$$\textbf{prox}_h(\mathbf{x}) = \underset{\mathbf{u}}{\text{argmin}} \left\{ h(\mathbf{u}) + \frac{1}{2}\|\mathbf{u} - \mathbf{x}\|_2^2 \right\}. \tag{10}$$

Authorized licensed use limited to: IEEE Xplore. Downloaded on June 04,2023 at 07:22:11 UTC from IEEE Xplore. Restrictions apply.

4. ONLINE DUAL PROXIMAL GRADIENT ALGORITHM

We switch gears to online estimation of $\mathbf{W}$ (or even tracking $\mathbf{W}_t$ in a dynamic setting) from streaming signals $\{\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots\}$. A natural approach would be to solve the time-varying optimization problem at each time instant $t = 1, 2, \ldots$ [cf. (4)]

$$\mathbf{w}_t^\star \in \operatorname*{argmin}_{\mathbf{w}} \overbrace{\Big\{\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} + 2\mathbf{w}^\top \mathbf{e}_{1:t} + \beta\|\mathbf{w}\|_2^2 \underbrace{-\alpha \mathbf{1}^\top \log(\mathbf{S}\mathbf{w})}_{:=-g(\mathbf{S}\mathbf{w})}\Big\}}^{:=f_t(\mathbf{w})}, \quad (11)$$

where the vectorized dissimilarity matrix $\mathbf{e}_{1:t}$ is formed using all signals acquired by time t. As data come in, the edge-wise ℓ_1-norm weights will fluctuate explaining the time dependence of (11) through its non-smooth component f_t.

One can naively solve (11) by sequentially running the batch DPG algorithm (cf. Section 3) every time a new datum arrives. However, this approach falls short when it comes to (pseudo) real-time operation, particularly for delay-sensitive tasks that preclude running multiple $k_t = 1, 2, \ldots$ inner DPG iterations per time interval $[t, t+1)$ (in order to attain convergence to $\mathbf{w}_t^\star$). What is more, for dynamic networks it may not be even worth obtaining a high-precision solution $\mathbf{w}_t^\star$ (thus incurring high delay and computational burden), because at time $t+1$ a new datum arrives and the solution $\mathbf{w}_{t+1}^\star$ may be substantially off the prior estimate; see also Section 5.1. All these considerations motivate well the pursuit of an recursive algorithm that can track the solution of the time-varying optimization (11).

Our online approach entails two steps per time instant $t = 1, 2, \ldots$ of data acquisition. First, we recursively update the upper-triangular entries $\mathbf{e}_{1:t}$ of the Euclidean-distance matrix once $\mathbf{e}_t$ becomes available. In stationary settings where the graph is static, it is prudent to adopt an infinite-memory scheme

$$\mathbf{e}_{1:t} = \frac{1}{t}\sum_{\tau=1}^{t} \mathbf{e}_\tau = \frac{\mathbf{e}_t + (t-1)\mathbf{e}_{1:t-1}}{t}. \quad (12)$$

On the other hand, in non-stationary environments arising with dynamic networks it is preferable to update $\mathbf{e}_{1:t}$ via a moving average in order to track the topology fluctuations. To that end, we employ an exponentially-weighted moving average (EWMA)

$$\mathbf{e}_{1:t} = (1-\gamma)\mathbf{e}_{1:t-1} + \gamma \mathbf{e}_t, \quad (13)$$

where the constant $\gamma \in (0, 1)$ is a discount (or forgetting) factor. In other words, higher γ downweighs older observations faster. Second, we run a single iteration of the batch graph learning algorithm developed in Section 3 to update $\mathbf{w}_{t+1}$, namely

$$\boldsymbol{\lambda}_t = \mathbf{prox}_{L^{-1}G}\left(\boldsymbol{\lambda}_{t-1} - \frac{1}{L}\nabla F_t(\boldsymbol{\lambda}_{t-1})\right), \quad (14)$$

The dual variable update iteration in (14) can be equivalently rewritten as $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} - L^{-1}(\mathbf{S}\mathbf{v}_t - \mathbf{u}_t)$, with

$$\mathbf{v}_t = \max\left(\mathbf{0}, \frac{\mathbf{S}^\top \boldsymbol{\lambda}_{t-1} - 2\mathbf{e}_{1:t}}{2\beta}\right), \quad (15)$$

$$\mathbf{u}_t = \frac{\mathbf{S}\mathbf{v}_t - L\boldsymbol{\lambda}_{t-1} + \sqrt{(\mathbf{S}\mathbf{v}_t - L\boldsymbol{\lambda}_{t-1})^2 + 4\alpha L \mathbf{1}}}{2}, \quad (16)$$

Algorithm 1: Online DPG for dynamic topology inference

Input parameters α, β, stream $\mathbf{e}_1, \mathbf{e}_2, \ldots$, set $L = \frac{N-1}{\beta}$.
Initialize $\boldsymbol{\lambda}_0$ at random.
for $t = 1, 2, \ldots,$ **do**
  Update $\mathbf{e}_{1:t}$ via either (12) or (13)
  $\mathbf{v}_t = \max\left(\mathbf{0}, \frac{\mathbf{S}^\top \boldsymbol{\lambda}_{t-1} - 2\mathbf{e}_{1:t}}{2\beta}\right)$
  $\mathbf{u}_t = \frac{\mathbf{S}\mathbf{v}_t - L\boldsymbol{\lambda}_{t-1} + \sqrt{(\mathbf{S}\mathbf{v}_t - L\boldsymbol{\lambda}_{t-1})^2 + 4\alpha L \mathbf{1}}}{2}$
  $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} - L^{-1}(\mathbf{S}\mathbf{v}_t - \mathbf{u}_t)$
end
Output topology estimate $\hat{\mathbf{w}}_t = \max\left(\mathbf{0}, \frac{\mathbf{S}^\top \boldsymbol{\lambda}_t - 2\mathbf{e}_{1:t}}{2\beta}\right)$

where $\max(\cdot,\cdot)$, $(\cdot)^2$, and $\sqrt{(\cdot)}$ in (15) are all element-wise operations of their vector arguments, see [23, Proposition 1] for all details. The novel online DPG iterations are tabulated under Algorithm 1.

Computational complexity. Update (15) in Algorithm 1 incurs a per iteration cost of $\mathcal{O}(N^2)$, on par with the online PG algorithm in [26]. The auxiliary variable $\mathbf{u}_t$ is also given in closed form, through simple operations of vectors living in the dual N-dimensional domain of nodal degrees [cf. the $N(N-1)/2$-dimensional primal variables $\bar{\mathbf{w}}_k$]. If additional prior knowledge about the set of possible edges is available, one can reduce the complexity further [21]. There are no step-size parameters to tune here (on top of α and β) since we explicitly know the Lipschitz constant L [23, Lemma 1].

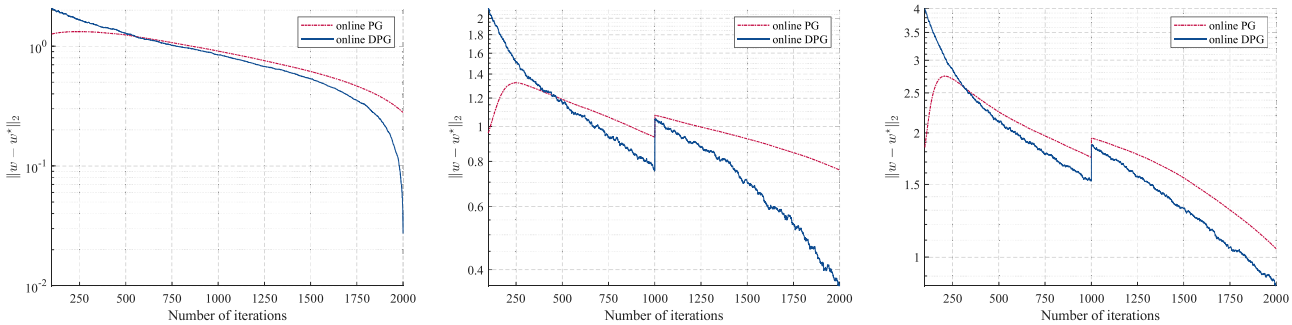
5. NUMERICAL EXPERIMENTS

We perform numerical experiments to assess how well the online DPG algorithm learns random and real-world graphs, in both stationary and dynamic environments. Our main objective is to evaluate Algorithm 1's effectiveness in tracking $\mathbf{w}_t^\star$ (11). This is different from explicitly monitoring the quality of the minimizer in terms of recovering the ground-truth graph used to generate the data; see e.g., the experiments in [18, 21] for a study on the latter. This clarification notwithstanding, for the synthetic experiments we perform a rough grid search to tune α, β, the criterion being to maximize the F-measure of the recovered edge set. Unless otherwise stated, the discount factor is set to $\gamma = 0.002$ for all ensuing experiments. We compare Algorithm 1 to online PG [26], since they both incur the same computational complexity per iteration. The code to generate all figures in this section can be downloaded from `http://hajim.rochester.edu/ece/sites/gmateos/code/ODPG.zip`.

5.1. Random graphs

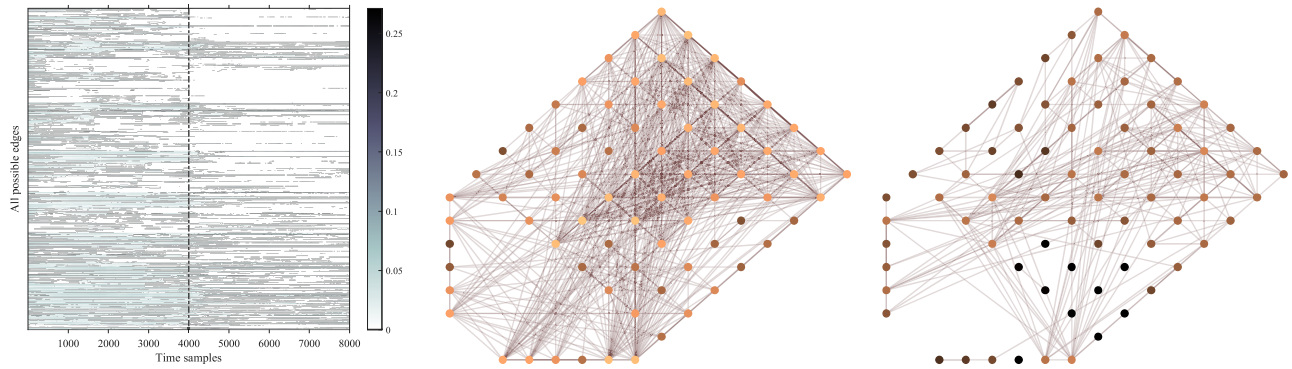
Here we examine a pair of test cases. First, we assume the underlying graph is time invariant and use (12) in Algorithm 1. We draw a $N = 100$-node graph realization from the Erdős-Rényi (ER) model (edge probability $p = 0.2$). Second, we consider dynamic graph learning where the underlying network topology changes midway through the trial. We simulate two settings. A piecewise-constant sequence of two: i) random graphs is drawn from the ER model ($p = 0.2$) with $N = 50$ nodes; as well as from ii) a 2-block Stochastic Block Model (SBM) with $N = 100$ vertices (even split of nodes across blocks), and connection probability $p_1 = 0.3$ for nodes in the same community and $p_2 = 0.05$ for vertices in different blocks. The initial graph switches after $t = 1000$, leading

Authorized licensed use limited to: IEEE Xplore. Downloaded on June 04,2023 at 07:22:11 UTC from IEEE Xplore. Restrictions apply.



(a) (b) (c)

Fig. 1. Convergence behavior illustrated via the evolution of $\|\hat{\mathbf{w}}_t - \mathbf{w}_t^\star\|_2$, for various random graph models. (a) Stationary ER graph with $N = 100$, (b) dynamic ER graph with $N = 50$, and (c) dynamic SBM graph with $N = 100$. For the dynamic network, the topology changes at $t = 1000$. In all settings, the proposed online DPG method converges faster to $\mathbf{w}_t^\star$ than the baseline algorithm in [26].



(a) (b) (c)

Fig. 2. Dynamic graph learning using ECoG data. (a) Evolution of edge weights from pre-ictal to ictal stage, where the vertical line indicates seizure onset. Recovered brain graphs (b) 2.5s prior to seizure; and (c) 2.5s after. An edge-thinning pattern is apparent on seizure onset.

to a different topology obtained after resampling 10% of the edges. For $t = 1, \ldots, 2000$, we generate i.i.d. Gaussian-distributed smooth signals $\mathbf{x}_t \sim \mathcal{N}\left(\mathbf{0}, \mathbf{L}_t^\dagger + \sigma_e^2 \mathbf{I}_N\right)$, where $\sigma_e = 0.01$ and $\mathbf{L}_t$ is the Laplacian of the ground-truth graph; see e.g., [19].

We examine the convergence behavior of the aforementioned methods by monitoring the evolution of the error metric $\|\hat{\mathbf{w}}_t - \mathbf{w}_t^\star\|_2$. For given α, β, we compute the solution $\mathbf{w}_t^\star$ by running the batch algorithm in [18] until convergence (we use all signals available over the interval where the graph remains fixed). The results of these tests are presented Fig. 1. The plots clearly show that Algorithm 1 converges faster to $\mathbf{w}_t^\star$ than the baseline [26]; in both the stationary and dynamic settings, uniformly across model classes and graph sizes.

5.2. Dynamic network-based analysis of epileptic seizures

We test Algorithm 1 on an inherently dynamic task, namely a network-based study of epileptic seizures [25]. To that end, we resort to a publicly available dataset that contains electrocorticogram (ECoG) signals of 8 seizure instances acquired from human patients with epilepsy [7]. ECoG data are captured by $N = 76$ electrodes, where 64 of them form an 8×8 grid located at the cortical brain's surface. The other 12 electrodes are placed deeper in the left suborbital frontal lobe and over the left hippocampal region. Signals are recorded with sampling rate of 400 Hz.

We apply online DPG to the ECoG signals including 10 seconds before an epileptic seizure (pre-ictal) and 10 seconds upon seizure onset (ictal). In general, we notice a significant reduction in the overall level of brain connectivity during the seizures. In Fig. 2(a) we plot the temporal variation of edge weights as we update the recovered dynamic brain network. The vertical line indicates the moment of seizure onset, where the drop in edge density is considerable. Moreover, in Figs. 2(b) and (c) we depict two snapshots of the learned graph at 2.5 seconds prior to seizure and 2.5 seconds after; respectively. Vertex colors in Fig. 2(b) are proxies of closeness centrality values, where darker shades correspond to lower values. The closeness centrality scores are calculated as the inverse sum of the distances from a node to all other nodes [7, Ch.4]. We observe in Fig. 2(b) and (c) that edge thinning is more prominent in the bottom corner of the grid and along the two strips. This result is well aligned with the findings in [25].

6. CONCLUDING SUMMARY

We proposed an online algorithm to track the topology of slowly-varying undirected graphs from streaming signals. Capitalizing on favorable dual domain structure of a smoothness-regularized inverse problem with well-documented merits, we derive and truncate proximal gradient iterations to minimize a time-varying cost in an online fashion. The novel algorithm is devoid of (often hard to tune) step-sizes, it is lightweight and demands constant memory storage regardless of the number of measurements. Numerical tests with synthetic and real brain activity signals demonstrate the effectiveness of the dynamic network topology tracker, and that it compares favorably against a state-of-the-art baseline of comparable complexity.

Authorized licensed use limited to: IEEE Xplore. Downloaded on June 04,2023 at 07:22:11 UTC from IEEE Xplore. Restrictions apply.

7. REFERENCES

[1] 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.

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[3] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[4] W. L. Hamilton, "Graph representation learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 14, no. 3, pp. 1–159, 2020.

[5] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy, "Machine learning on graphs: A model and comprehensive taxonomy," *J. Mach. Learn. Res.*, vol. 23, no. 89, pp. 1–64, 2022.

[6] X. Dong, D. Thanou, L. Toni, M. Bronstein, and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 117–127, 2020.

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

[8] O. Sporns, *Networks of the Brain*. MIT Press, 2010.

[9] 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.

[10] 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.

[11] S. S. Saboksayr, G. Mateos, and M. Cetin, "Online discriminative graph learning from multi-class smooth signals," *Signal Process.*, vol. 186, p. 108101, 2021.

[12] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. IEEE*, vol. 106, no. 5, pp. 787–807, 2018.

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

[14] E. Pavez, H. E. Egilmez, and A. Ortega, "Learning graphs with monotone topology properties and multiple connected components," *IEEE Trans. Signal Process.*, vol. 66, no. 9, pp. 2399–2413, May 2018.

[15] S. Kumar, J. Ying, J. V. de M. Cardoso, and D. P. Palomar, "A unified framework for structured graph learning via spectral constraints," *J. Mach. Learn. Res.*, vol. 21, no. 22, pp. 1–60, 2020.

[16] 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.

[17] R. Shafipour and G. Mateos, "Online topology inference from streaming stationary graph signals with partial connectivity information," *Algorithms*, vol. 13, no. 9, pp. 1–19, Sep. 2020.

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

[19] 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, 2016.

[20] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, 2017, pp. 2826–2830.

[21] V. Kalofolias and N. Perraudin, "Large scale graph learning from smooth signals," in *Int. Conf. Learning Representations (ICLR)*, 2019.

[22] P. Berger, G. Hannak, and G. Matz, "Efficient graph learning from noisy and incomplete data," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 105–119, 2020.

[23] S. S. Saboksayr and G. Mateos, "Accelerated graph learning from smooth signals," *IEEE Signal Process. Lett.*, vol. 28, pp. 2192–2196, 2021.

[24] A. Beck and M. Teboulle, "A fast dual proximal gradient algorithm for convex minimization and applications," *Operations Research Letters*, vol. 42, no. 1, pp. 1–6, 2014.

[25] M. A. Kramer, E. D. Kolaczyk, and H. E. Kirsch, "Emergent network topology at seizure onset in humans," *Epilepsy Res.*, vol. 79, no. 2, pp. 173–186, 2008.

[26] S. S. Saboksayr, G. Mateos, and M. Cetin, "Online graph learning under smoothness priors," in *European Signal Process. Conf. (EUSIPCO)*, Dublin, Ireland, 2021, pp. 1820–1824.

[27] J. V. d. M. Cardoso and D. P. Palomar, "Learning undirected graphs in financial markets," in *Asilomar Conf. Signals, Systems, and Computers*, 2020, pp. 741–745.

[28] A. Natali, E. Isufi, M. Coutino, and G. Leus, "Learning time-varying graphs from online data," *IEEE Open J. Signal Process.*, vol. 3, pp. 212–228, 2022.

[29] S. Vlaski, H. P. Maretić, R. Nassif, P. Frossard, and A. H. Sayed, "Online graph learning from sequential data," in *IEEE Data Sci. Wrksp. (DSW)*, 2018.

[30] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *Int. Conf. Mach. Learning (ICML)*, 2004.

[31] X. Wang, C. Yao, H. Lei, and A. M.-C. So, "An efficient alternating direction method for graph learning from smooth signals," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Toronto, Canada, 2021, pp. 5380–5384.

[32] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, p. 127–239, 2014.

Authorized licensed use limited to: IEEE Xplore. Downloaded on June 04,2023 at 07:22:11 UTC from IEEE Xplore. Restrictions apply.