Accelerated Graph Learning From Smooth Signals

Seyed Saman Saboksayr, *Student Member, IEEE*, and Gonzalo Mateos, *Senior Member, IEEE*

***Abstract*—We consider network topology identification subject to a signal smoothness prior on the nodal observations. A fast dual-based proximal gradient algorithm is developed to efficiently tackle a strongly convex, smoothness-regularized network inverse problem known to yield high-quality graph solutions. Unlike existing solvers, the novel iterations come with global convergence rate guarantees and do not require additional step-size tuning. Reproducible simulated tests demonstrate the effectiveness of the proposed method in accurately recovering random and real-world graphs, markedly faster than state-of-the-art alternatives and without incurring an extra computational burden.**

***Index Terms*—Graph learning, graph signal processing, fast gradient methods, signal smoothness, topology identification.**

I. Introduction

NETWORK-AWARE signal and information processing is having a major impact in technology and the biobehavioral sciences; see e.g, [1, Ch. 1]. In this context, graph signal processing (GSP) builds on a graph-theoretic substrate to effectively model signals with complex relational structures [2]–[4]. However, the required connectivity information is oftentimes not explicitly available. This motivates the prerequisite step of using signals (e.g., brain activity traces, distributed sensor measurements) to unveil latent network structure, or, to construct discriminative graph representations to facilitate downstream learning tasks. As graph data grow in size and complexity, there is an increasing need to develop customized, fast and computationally-efficient graph learning algorithms.

Given nodal measurements (known as graph signals in the GSP parlance), the network topology inference problem is to search for a graph within a model class that is optimal in some application-specific sense, e.g., [1, Ch. 7]. The adopted criterion is naturally tied to the signal model relating the observations to the sought network, which can include constraints motivated by physical laws, statistical priors, or, explainability goals. Workhorse probabilistic graphical models include Gaussian Markov random fields, and topology identification arises with so-termed high-dimensional graphical model selection [5]–[11]. Other recent approaches embrace a signal representation perspective to reveal parsimonious data signatures with respect to the underlying graph. These include stationarity induced via linear network diffusion [12]–[14] and smoothness (i.e., bandlimitedness) [15]–[23]. The interested reader is referred to [24]–[26] for comprehensive tutorial treatments of network topology inference advances.

In this short letter, we develop a fast and scalable algorithm to estimate graphs subject to a smoothness prior (Section II outlines the required background and formally states the problem). Adopting the well-appreciated graph learning framework of [15], [18], in Section III we bring to bear the fast proximal-gradient (PG) iterations in [27] to solve the resulting strongly convex, signal smoothness-regularized optimization problem in the dual domain. There are noteworthy recent scalable solvers for this problem that rely on the primal-dual (PD) method [15], PG [28], or, the linearized alternating-direction method of multipliers (ADMM) [29]. Unlike these algorithms, the novel iterations come with global convergence rate guarantees and do not require additional step-size tuning. Borrowing results from [27], we show that a (possibly infeasible) primal sequence generated from the accelerated graph learning algorithm converges to a globally optimal solution at a rate of $O(1/k)$. To the best of our knowledge, this is the first work that establishes the convergence rate of topology inference algorithms subject to smoothness priors. Computer simulations in Section IV showcase the favorable convergence properties of the proposed approach when recovering a wide variety of graphs. In the interest of reproducible research, the code used to generate all figures in this letter is publicly available. Conclusions are in Section V. Due to page contraints, proofs are deferred to the accompanying Supplementary Material.

II. Graph Learning From Smooth Signals

Let $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ be an undirected graph, where $\mathcal{V}$ are the nodes (or vertices) with $|\mathcal{V}| = N$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ are the edges, and $\mathbf{W} \in \mathbb{R}_+^{N \times N}$ is the symmetric adjacency matrix collecting the edge weights. For $(i, j) \notin \mathcal{E}$ we have $W_{ij} = 0$. We exclude the possibility of self-loops, so $\mathbf{W}$ is hollow meaning $W_{ii} = 0$, for all $i \in \mathcal{V}$. We acquire graph signal observations $\mathbf{x} = [x_1, \ldots, x_N]^\top \in \mathbb{R}^N$, where x_i denotes the signal value at vertex $i \in \mathcal{V}$. More general graphs capturing directionality are important [30], but beyond the scope of this letter.

A. Graph Signal Smoothness

For undirected graphs one typically adopts the Laplacian $\mathbf{L} := \text{diag}(\mathbf{d}) - \mathbf{W}$ as descriptor of graph structure, where $\mathbf{d} = \mathbf{W}\mathbf{1}$ collects the vertex degrees. As the central object in spectral graph theory, $\mathbf{L}$ is instrumental in formalizing the notion of smooth (i.e., low-pass bandlimited) signals on graphs [2], [31]. Specifically, the total variation (TV) of the graph signal $\mathbf{x}$ with

Manuscript received September 8, 2021; revised October 20, 2021; accepted October 25, 2021. Date of publication October 27, 2021; date of current version November 23, 2021. This work was supported in part by the NSF Awards under Grants CCF-1750428, CCF-1934962, and ECCS-1809356. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shiwen He. *(Corresponding author: Gonzalo Mateos.)*

The authors are with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, New York, NY 14620 USA (e-mail: ssaboksa@ur.rochester.edu; gmateosb@ece.rochester.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LSP.2021.3123459, provided by the authors.

Digital Object Identifier 10.1109/LSP.2021.3123459

respect to $\mathcal{G}$ is given by the quadratic form

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

We interpret $\text{TV}(\mathbf{x})$ as a smoothness measure for graph signals, which gauges the extent to which $\mathbf{x}$ varies across $\mathcal{G}$. Accordingly, we say a signal is smooth if it has a small total variation. For reference, $0 \leq \text{TV}(\mathbf{x}) \leq \lambda_{\max}$, where $\lambda_{\max}$ is the spectral radius of $\mathbf{L}$. The lower bound is attained by constant signals. The ubiquity of smooth network data has been well-documented, with examples spanning sensor measurements [19], protein function annotations [1], and product ratings [32]. These empirical findings motivate adopting smoothness as the criterion to search for graphs on which measurements exhibit desirable parsimony or regularity.

B. Problem Statement

We study the following graph learning problem.

Problem 1: Given a set $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^P$ of graph signal observations, the goal is to learn an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ such that the observations in $\mathcal{X}$ are smooth on $\mathcal{G}$.

We now briefly review the method proposed in [15], [18] to tackle Problem 1, from which we henceforth build on to develop a fast graph learning algorithm.

Consider the matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N\times P}$, whose columns $\mathbf{x}_p$ are the observations in $\mathcal{X}$. The rows, denoted by $\bar{\mathbf{x}}_i^\top \in \mathbb{R}^{1\times P}$, collect all P measurements at vertex i. Define then the nodal Euclidean-distance matrix $\mathbf{E} \in \mathbb{R}_+^{N\times N}$, where $E_{ij} := \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_2^2$, $i, j \in \mathcal{V}$. Using these notions, the signal smoothness measure over $\mathcal{X}$ can be equivalently written as

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

where $\circ$ denotes element-wise product [15]. Smoothness minimization as criterion in Problem 1 has the following intuitive interpretation: when pairwise nodal distances in $\mathbf{E}$ are sampled from a smooth manifold, the learnt topology $\mathbf{W}$ tends to be sparse, preferentially choosing edges (i, j) whose corresponding E_{ij} are smaller [cf. the weighted ℓ_1-norm in (2)].

Leveraging this neat link between signal smoothness and edge sparsity, a fairly general graph-learning framework was put forth in [15]. The idea therein is to solve the following convex inverse problem

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

where $\alpha, \beta > 0$ are tunable regularization parameters. Different from [16], the logarithmic barrier on the vertex degrees $\mathbf{d} = \mathbf{W}\mathbf{1}$ excludes the possibility of having (often undesirable) isolated vertices in the estimated graph. Through β, the Frobenius-norm penalty offers a handle on the graphs' edge sparsity level. Among the parameterized familiy of solutions to (3), the sparsest graph is obtained when $\beta = 0$.

Arguably, the most important upshot of identity (2) is computational. It facilitates formulating (3) as a search over adjacency matrices, and the resulting constraints (null diagonal, symmetry and non-negativity) are separable across the variables W_{ij}. This does not hold for the Laplacian $\mathbf{L}$. Exploting this favorable structure of (3), efficient solvers were developed based on PD iterations [15], the PG method [28], or the ADMM [29]. However, none of these graph learning methods come with convergence rate guarantees because the objective function of (3) lacks a Lipschitz continuous gradient. To close this gap, next we develop a markedly faster first-order algorithm using an accelerated dual-based PG method [27].

III. Fast Dual Proximal Gradient Algorithm

Because $\mathbf{W}$ is hollow and symmetric, the optimization variables in (3) are effectively the, say, upper-triangular elements $[\mathbf{W}]_{ij}$, $j > i$. Thus, it suffices to retain only those entries in the 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 that edge weights are non-negative, we penalize the cost with the indicator function $\mathbb{I}\{\mathbf{w}\succeq\mathbf{0}\} = 0$ if $\mathbf{w}\succeq\mathbf{0}$, else $\mathbb{I}\{\mathbf{w}\succeq\mathbf{0}\} = \infty$ [15]. This way, we equivalently reformulate (3) as the unconstrained, non-differentiable problem

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

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 (details are in the Supplementary Material), while $g(\mathbf{w}) := -\alpha\mathbf{1}^\top\log(\mathbf{w})$ is a (strictly) convex function for all $\mathbf{w}\succ\mathbf{0}$. Under the aforementioned properties of f and g, the composite problem (4) has a unique optimal solution $\mathbf{w}^\star$; see e.g., [27] and [29].

A fast dual-based PG algorithm was developed in [27] to solve the non-smooth, strictly convex optimization problem $\min_{\mathbf{w}}\{f(\mathbf{w}) + g(\mathbf{S}\mathbf{w})\}$ of which (4) is a particular instance. In the remainder of this section we will bring to bear this optimization framework to develop a novel graph learning algorithm with global rate of convergence guarantees.

A. The Dual Problem

The structure of (4) lends itself naturally to variable-splitting via the equivalent linearly-constrained form

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

Attaching Lagrange multipliers $\boldsymbol{\lambda}\in\mathbb{R}^N$ to the equality constraints and minimizing the Lagrangian function $\mathcal{L}(\mathbf{w},\mathbf{d},\boldsymbol{\lambda}) = f(\mathbf{w}) + g(\mathbf{d}) - \langle\boldsymbol{\lambda}, \mathbf{S}\mathbf{w} - \mathbf{d}\rangle$ w.r.t. the primal variables $\{\mathbf{w},\mathbf{d}\}$, one arrives at the (minimization form) dual problem [27]

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

where

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

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

Interestingly, the strong convexity of f induces useful smoothness properties for F (namely, the composition of $\mathbf{S}\mathbf{w}$ with the

Fenchel conjugate of f), that we summarize next. The result is adapted from [27, Lemma 3.1] and the additional proof arguments can be found in the Supplementary Material.

Lemma 1: Function $F(\boldsymbol{\lambda})$ in (7) is smooth, and the gradient $\nabla F(\boldsymbol{\lambda})$ is Lipschitz continuous with constant $L := \frac{N-1}{\beta}$.

This additional structure of (6) makes it feasible to apply accelerated PG algorithms [33] (such as FISTA [34]), to solve the dual problem.

B. Accelerated Dual Proximal Gradient Algorithm

The FISTA algorithm applied to the dual problem (6) yields the following iterations (initialized as $\boldsymbol{\omega}_1 = \boldsymbol{\lambda}_0 \in \mathbb{R}^N$ and $t_1 = 1$, henceforth $k = 1, 2, \ldots$ denotes the iteration index)

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

$$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \tag{10}$$

$$\boldsymbol{\omega}_{k+1} = \boldsymbol{\lambda}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)[\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{k-1}], \tag{11}$$

where the proximal operator of a proper, lower semi-continuous convex function h is (see e.g., [35])

$$\mathbf{prox}_h(\mathbf{x}) = \text{argmin}_{\mathbf{u}}\left\{h(\mathbf{u}) + \frac{1}{2}\|\mathbf{u}-\mathbf{x}\|_2^2\right\}. \tag{12}$$

An adaptation of the result in [27, Lemma 3.2] – stated as Proposition 1 below – yields the novel graph learning iterations tabulated under Algorithm 1. Again, due to page constraints the proof details are deferred to the Supplementary Material.

Proposition 1: The dual variable update iteration in (9) can be equivalently rewritten as $\boldsymbol{\lambda}_k = \boldsymbol{\omega}_k - L^{-1}(\mathbf{S}\bar{\mathbf{w}}_k - \mathbf{u}_k)$, with

$$\bar{\mathbf{w}}_k = \max\left(\mathbf{0}, \frac{\mathbf{S}^\top\boldsymbol{\omega}_k - 2\mathbf{e}}{2\beta}\right), \tag{13}$$

$$\mathbf{u}_k = \frac{\mathbf{S}\bar{\mathbf{w}}_k - L\boldsymbol{\omega}_k + \sqrt{(\mathbf{S}\bar{\mathbf{w}}_k - L\boldsymbol{\omega}_k)^2 + 4\alpha L\mathbf{1}}}{2}, \tag{14}$$

where $\max(\cdot,\cdot)$ in (13) as well as both $(\cdot)^2$ and $\sqrt{(\cdot)}$ in (14) are element-wise operations on their vector arguments.

The updates in Proposition 1 are fully expressible in terms of parameters from the original graph learning problem, namely N, α, β, $\mathbf{S}$ and the data in $\mathbf{e}$. This is to be contrasted with (9), which necessitates the conjugate functions F and G.

Algorithm 1's overall computational complexity is dominated by the update (13), which incurs a per iteration cost of $\mathcal{O}(N^2)$. The remaining updates are 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 $\bar{\mathbf{w}}_k$]. The overall complexity of $O(N^2)$ is in par with state-of-the-art PD and linearized ADMM algorithms [29], which have been shown to scale well to large networks with N in the order of thousands. The computational cost can be further reduced by constraining a priori the space of possible edges; see [18] for examples where this approach is warranted. For a given problem instance, there are no step-size parameters to tune here (on top of α and β) since we can explicitly compute the Lipschitz constant L in Lemma 1. On the other hand, the linearized ADMM algorithm in [29] necessitates tuning two step-sizes and the penalty parameter defining the augmented Lagrangian.

Algorithm 1: Topology inference via fast dual PG (FDPG).

Input parameters α, β, data $\mathbf{e}$, set $L = \frac{N-1}{\beta}$.
Initialize $t_1 = 1$ and $\boldsymbol{\omega}_1 = \boldsymbol{\lambda}_0$ at random.
for $k = 1, 2, \ldots,$ **do**
$\bar{\mathbf{w}}_k = \max(\mathbf{0}, \frac{\mathbf{S}^\top\boldsymbol{\omega}_k - 2\mathbf{e}}{2\beta})$
$\mathbf{u}_k = \frac{\mathbf{S}\bar{\mathbf{w}}_k - L\boldsymbol{\omega}_k + \sqrt{(\mathbf{S}\bar{\mathbf{w}}_k - L\boldsymbol{\omega}_k)^2 + 4\alpha L\mathbf{1}}}{2}$
$\boldsymbol{\lambda}_k = \boldsymbol{\omega}_k - L^{-1}(\mathbf{S}\bar{\mathbf{w}}_k - \mathbf{u}_k)$
$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
$\boldsymbol{\omega}_{k+1} = \boldsymbol{\lambda}_k + (\frac{t_k-1}{t_{k+1}})[\boldsymbol{\lambda}_k - \boldsymbol{\lambda}_{k-1}]$
end
Output graph estimate $\hat{\mathbf{w}}_k = \max(\mathbf{0}, \frac{\mathbf{S}^\top\boldsymbol{\lambda}_k - 2\mathbf{e}}{2\beta})$

The distinctive feature of the proposed accelerated dual PG algorithm is that it comes with global convergence rate guarantees. These results are outlined in the ensuing section.

C. Convergence Rate Analysis

Moving on to convergence properties, when $k \to \infty$ the iterates $\boldsymbol{\lambda}_k$ generated by Algorithm 1 provably approach a dual optimal solution $\boldsymbol{\lambda}^\star$ that minimizes $\varphi(\boldsymbol{\lambda}) := F(\boldsymbol{\lambda}) + G(\boldsymbol{\lambda})$ in (6); see e.g., [34]. The celebrated FISTA rate of convergence for the dual cost function is stated next.

Theorem 1: [34, Theorem 4.4] For all $k \geq 1$, dual iterates $\boldsymbol{\lambda}_k$ stemming from Algorithm 1 are such that

$$\varphi(\boldsymbol{\lambda}_k) - \varphi(\boldsymbol{\lambda}^\star) \leq \frac{2(N-1)\|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^\star\|_2^2}{\beta k^2}. \tag{15}$$

This well-documented $O(1/k^2)$ global convergence rate of accelerated PG algorithms implies an $\mathcal{O}(1/\sqrt{\epsilon})$ iteration complexity to return an ϵ-optimal dual solution measured in terms of φ values; see also [36], [37] for potential transient speedups.

We now consider a primal sequence generated from the iterates of Algorithm 1, and borrow the results from [27] to show the sequence is globally convergent to $\mathbf{w}^\star$ at a rate of $O(1/k)$. To this end, suppose that for all $k \geq 1$ we are given dual updates $\boldsymbol{\lambda}_k$ generated from the accelerated dual PG algorithm. We can construct a primal sequence as $\hat{\mathbf{w}}_k = \text{argmin}_{\mathbf{w}}\mathcal{L}(\mathbf{w}, \mathbf{d}, \boldsymbol{\lambda}_k)$, namely [cf. (7)]

$$\begin{aligned}\hat{\mathbf{w}}_k &= \text{argmax}_{\mathbf{w}}\left\{\langle \mathbf{S}^\top\boldsymbol{\lambda}_k, \mathbf{w}\rangle - f(\mathbf{w})\right\} \\ &= \max\left(\mathbf{0}, \frac{\mathbf{S}^\top\boldsymbol{\lambda}_k - 2\mathbf{e}}{2\beta}\right).\end{aligned} \tag{16}$$

As noted in [29], this primal sequence may be infeasible in the sense that resulting nodal degrees $\hat{\mathbf{d}}_k := \mathbf{S}\hat{\mathbf{w}}_k$ are not guaranteed to lie in the domain of g. The promised $O(1/k)$ rate of converge result for $\hat{\mathbf{w}}_k$ is stated next.

Theorem 2: [27, Theorem 4.1] For all $k \geq 1$, the primal sequence (16) defined in terms of dual iterates $\boldsymbol{\lambda}_k$ generated by Algorithm 1 satisfies

$$\|\hat{\mathbf{w}}_k - \mathbf{w}^\star\|_2 \leq \frac{\sqrt{2(N-1)}\|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^\star\|_2}{\beta k}. \tag{17}$$

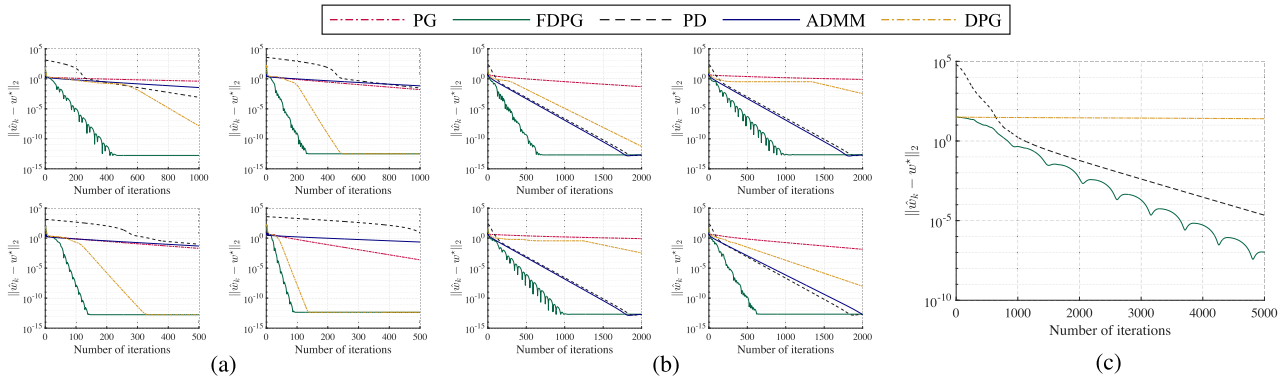

Fig. 1. Convergence performance in terms of primal variable error $\|\hat{\mathbf{w}}_k - \mathbf{w}^\star\|_2$ when recovering different synthetic and real graphs. (a) ER graphs with $N = 200$ (top-left) and $N = 400$ nodes (top-right); SBM graphs with $N = 200$ (bottom-left), and $N = 400$ nodes (bottom-right). (b) Four representative structural brain graphs with $N = 66$ ROIs; Subject 1 (top-left), Subject 2 (top-right), Subject 4 (bottom-left), and Subject 6 (bottom-right). (c) Minnesota road network with $N = 2642$ intersections. In all cases, the proposed FDPG method converges faster to $\mathbf{w}^\star$ than state-of-the-art graph learning algorithms.

IV. Numerical Results

Here we test the proposed fast dual PG (FDPG) algorithm for learning random and real-world graphs from simulated signals. The merits of the formulation (3) in terms of recovering high-quality graphs have been well documented; see e.g., [15], [18], [24], [25] and references therein. For this reason, the numerical experiments that follow will exclusively focus on algorithmic performance, with no examination of the quality of the optimal solution $\mathbf{w}^\star$ that defines the learnt graph. In all ensuing test cases, we search for the best regularization parameters α, β in terms of graph recovery performance, adopting the edge-detection F-measure as criterion. We compare Algorithm 1 to other state-of-the-art methods such as PD [15], PG [28], and linearized ADMM [29]. We also consider the non-accelerated dual PG (DPG) method that is obtained from Algorithm 1 when $t_k \equiv 1$ for all $k \geq 1$. For FDPG we implemented customary fixed-interval restarts of the momentum term in Algorithm 1; see also [38] for adaptive restart rules. Moreover, the ADMM parameters and PD step-size are tuned to yield the best possible convergence rate. Implementation details can be found in the publicly available code[1], which can be used to generate all plots in Fig. 1.

A. Random Graphs

We generate ground-truth graphs as draws from the Erdős-Rényi (ER) model (edge formation probability $p = 0.1$) with $N = 200$ and 400 nodes, as well as from the 2-block Stochastic Block Model (SBM) with the same number of nodes, and connection probability $p_1 = 0.3$ for nodes in the same community and $p_2 = 0.05$ for nodes in different blocks. We simulate $P = 1000$ i.i.d. graph signals $\mathbf{x}_p \sim \mathcal{N}(\mathbf{0}, \mathbf{L}^\dagger + \sigma_e^2 \mathbf{I}_N)$, where $\sigma_e = 0.1$ represents the noise level and $\mathbf{L}$ is the Laplacian of the ground-truth random graph. For a graph-based factor analysis model justifying this approach to smooth signal generation, see e.g., [16]. We compare the convergence performance of the aforementioned methods through the evolution of the primal variable error $\|\hat{\mathbf{w}}_k - \mathbf{w}^\star\|_2$. To obtain $\mathbf{w}^\star$ for the chosen α and β, we ran the PD method for 50000 iterations. The results of these comparisons are illustrated in Fig. 1(a). Apparently, the proposed FDPG algorithm markedly outperforms all other methods in terms of convergence rate, uniformly across graph model classes and number of nodes. Here, convergence to the largest graphs takes less iterations than for $N = 200$.

[1] http://www.ece.rochester.edu/~gmateosb/code/FDPG.zip.

B. Brain and Road Networks

We first focus on recovering the topology of 6 unweighted structural brain graphs [39], all with $N = 66$ regions of interest (ROIs) and whose edges connect ROIs with non-trivial density of axonal bundles; see also [40] for additional details. For a larger-scale experiment, we adopt the Minnesota road network which is an unweighted and undirected graph with $N = 2642$ intersections [41]. In both cases, we generated synthetic smooth signals over the real topologies using the generative model in Section IV-A. The high value of N renders the ADMM's 3-D parameter search a significantly time consuming operation. Hence, for the Minnesota road network experiment, we only focus on the proposed (F)DPG methods and the PD algorithm in [15].

Fig. 1(b) depicts the convergence results for the structural brain networks of 4 representative subjects. Once more, in all cases the FDPG method is faster, but for these smaller graphs the performance gap appears to narrow. The gains can also be quantified in terms of wall-clock time. For instance, for Subject 6 the time in seconds for the algorithms to reach a suboptimality of 10^{-8} are: 0.021 s for FDPG, 0.092 s for PD, 0.071 s for ADMM and 0.081 s for DPG. Results for the Minnesota road network are depicted in Fig. 1(c), where the superiority of the proposed method is also apparent.

V. Conclusion

We developed a fast and scalable algorithm to learn the graph structure of signals subject to a smoothness prior. Leveraging this cardinal property of network data is central to various statistical learning tasks, such as graph smoothing and semi-supervised node classification. We brought to bear a fast dual-based PG method to derive lightweight graph-learning iterations that come with global convergence rate guarantees. The merits of the proposed algorithm are showcased via experiments using several random and real-world graphs.

REFERENCES

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

[2] 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, May 2018.

[3] 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, May 2013.

[4] 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.

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

[6] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

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

[8] B. M. Lake and J. B. Tenenbaum, "Discovering structure by learning sparse graphs," in *Proc. Annu. Cogn. Sci. Conf.*, 2010, pp. 778–783.

[9] 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.

[10] 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.

[11] 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.

[12] 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, Sep. 2017.

[13] 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, Sep. 2018.

[14] 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.

[15] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. Artif. Intell. Statist.*, 2016, pp. 920–929.

[16] 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, Dec. 2016.

[17] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, "Learning time varying graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 2826–2830.

[18] V. Kalofolias and N. Perraudin, "Large scale graph learning from smooth signals," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–12.

[19] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, "Learning sparse graphs under smoothness prior," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 6508–6512.

[20] M. G. Rabbat, "Inferring sparse graphs from smooth signals with theoretical guarantees," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 6533–6537.

[21] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "Graph topology inference based on sparsifying transform learning," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1712–1727, Apr. 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] B. Le Bars, P. Humbert, L. Oudre, and A. Kalogeratos, "Learning Laplacian matrix from bandlimited graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 2937–2941.

[24] 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, May 2019.

[25] 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, May 2019.

[26] 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, May 2018.

[27] A. Beck and M. Teboulle, "A fast dual proximal gradient algorithm for convex minimization and applications," *Oper. Res. Lett.*, vol. 42, no. 1, pp. 1–6, 2014.

[28] S. S. Saboksayr, G. Mateos, and M. Cetin, "Online graph learning under smoothness priors," in *Proc. Eur. Signal Process. Conf.*, Dublin, Ireland, 2021, pp. 1820–1824.

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

[30] A. G. Marques, S. Segarra, and G. Mateos, "Signal processing on directed graphs: The role of edge directionality when processing and learning from network data," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 99–116, Nov. 2020.

[31] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *Proc. Int. Conf. Mach. Learn.*, 2004, pp. 132–137.

[32] W. Huang, A. G. Marques, and A. R. Ribeiro, "Rating prediction via graph signal processing," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5066–5081, Oct. 2018.

[33] A. Beck, *First-order Methods in Optimization*. Philadelphia, PA, USA: SIAM, 2018.

[34] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.

[35] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[36] G. Silva and P. Rodriguez, "FISTA: Achieving a rate of convergence proportional to k^{-3} for small/medium values of k," in *Proc. Eur. Signal Process. Conf.*, 2019, pp. 1–5.

[37] P. Rodriguez, "Improving FISTA's speed of convergence via a novel inertial sequence," in *Proc. Eur. Signal Process. Conf.*, 2019, pp. 1–5.

[38] B. O'Donoghue and E. Candes, "Adaptive restart for accelerated gradient schemes," *Found. Comput. Math.*, vol. 15, pp. 715–732, 2015.

[39] P. Hagmann *et al.*, "Mapping the structural core of human cerebral cortex," *PLoS Biol*, vol. 6, no. 7, 2008, Art. no. e 159.

[40] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology identification from spectral templates," in *Proc. IEEE Stat. Signal Process. Workshop*, 2016, pp. 1–5.

[41] T. A. Davis and Y. Hu, "The university of Florida sparse matrix collection," *ACM Trans. Math. Softw.*, vol. 38, no. 1, pp. 1–25, Dec. 2011.

SUPPLEMENTARY MATERIAL

Proof of Lemma 1

A couple preliminary calculations are required to derive an explicit expression for the Lipschitz constant of $F(\boldsymbol{\lambda})$.

Lemma 2 *The 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 constant* $\sigma := 2\beta > 0$.

Proof: The strong convexity of f with parameter $\sigma = 2\beta > 0$ follows because

$$f(\mathbf{w}) - \frac{\sigma}{2}\|\mathbf{w}\|^2 = \mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} + 2\mathbf{w}^\top \mathbf{e}$$

is a convex function. ■

Lemma 3 *Going back to* (4)*, recall* $\mathbf{S} \in \{0,1\}^{N \times N(N-1)/2}$ *defined so that* $\mathbf{d} = \mathbf{W}\mathbf{1} = \mathbf{S}\mathbf{w}$. *Then,* $\|\mathbf{S}\|_2 = \sqrt{2(N-1)}$.

Proof: Because $\mathbf{S}$ maps the upper-triangular adjacency matrix entries in $\mathbf{w}$ to the degree sequence $\mathbf{d}$, then $\mathbf{S}$ has $N-1$ ones in each row while all other entries are zero. Hence, the diagonal entries of $\mathbf{S}\mathbf{S}^\top$ are all $N-1$ and the off-diagonal entries are equal to 1. The eigenvalues λ of $\mathbf{S}\mathbf{S}^\top = (N-2)\mathbf{I} + \mathbf{1}\mathbf{1}^\top$ are the roots of the characteristic polynomial

$$\begin{aligned}
\det\left(\mathbf{S}\mathbf{S}^\top - \lambda\mathbf{I}\right) &= \det\left((N-2)\mathbf{I} + \mathbf{1}\mathbf{1}^\top - \lambda\mathbf{I}\right) \\
&= \det(\underbrace{(N-2-\lambda)\mathbf{I}}_{:=\mathbf{Q}} + \mathbf{1}\mathbf{1}^\top) \\
&= \det(\mathbf{Q}) + \mathbf{1}^\top \mathrm{adj}(\mathbf{Q})\mathbf{1} \\
&= \prod_{i=1}^{N} Q_{ii} + \sum_{j=1}^{N}\prod_{i\neq j} Q_{ii} \\
&= (N-2-\lambda)^N + N(N-2-\lambda)^{N-1} \\
&= (2N-2-\lambda)(N-2-\lambda)^{N-1} = 0.
\end{aligned}$$

To obtain the third equality we leveraged the Sherman-Morrison formula, where $\mathrm{adj}(\mathbf{Q})$ stands for the adjugate matrix of $\mathbf{Q}$. From the final factorization of the polynomial, the eigenvalues are $2(N-1) = \lambda_1 > \lambda_2 = \cdots = \lambda_N = N-2$. Because $\|\mathbf{S}\|_2 = \sqrt{\lambda_1}$, the result follows. ■

Since f is strongly convex (with constant σ), by virtue of [27, Lemma 3.1] the function $F(\boldsymbol{\lambda}) := \max_{\mathbf{w}} \left\{\langle \mathbf{S}^\top\boldsymbol{\lambda}, \mathbf{w}\rangle - f(\mathbf{w})\right\}$ is continuously differentiable and it has a Lipschitz continuous gradient with constant $L := \frac{\|\mathbf{S}\|_2^2}{\sigma}$. From the expressions for σ and $\|\mathbf{S}\|_2$ in Lemmata 2 and 3, the result follows. □

Proof of Proposition 1

Leveraging the result in [27, Lemma 3.2], it follows that for all $k \geq 1$, the dual variable update iteration in (9) can be equivalently rewritten as $\boldsymbol{\lambda}_k = \boldsymbol{\omega}_k - L^{-1}(\mathbf{S}\bar{\mathbf{w}}_k - \mathbf{u}_k)$, with

$$\bar{\mathbf{w}}_k = \operatorname*{argmax}_{\mathbf{w}} \left\{\langle \mathbf{S}^\top\boldsymbol{\omega}_k, \mathbf{w}\rangle - f(\mathbf{w})\right\}, \tag{18}$$

$$\mathbf{u}_k = \mathbf{prox}_{Lg}\left(\mathbf{S}\bar{\mathbf{w}}_k - L\boldsymbol{\omega}_k\right). \tag{19}$$

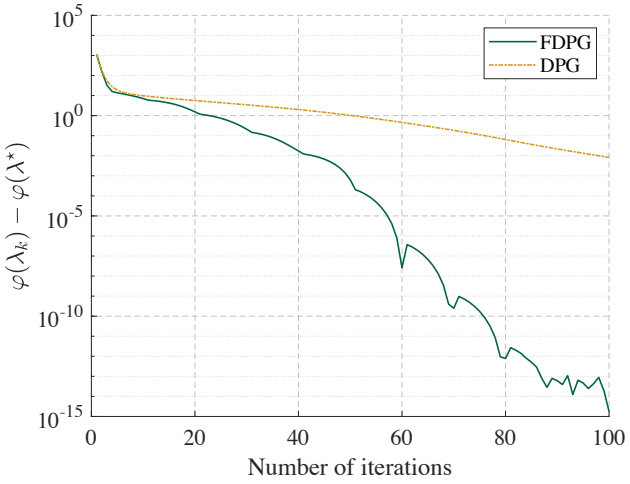

Fig. 2. Convergence performance in terms of dual suboptimality $\varphi(\boldsymbol{\lambda}_k) - \varphi(\boldsymbol{\lambda}^\star)$, when recovering the SBM graph with $N = 200$ nodes described in Section IV-A. As expected, the FDPG graph learning algorithm converges markedly faster than its non-accelerated counterpart.

Starting with (18), we have from the definition of f that

$$\begin{aligned}
\bar{\mathbf{w}}_k &= \operatorname*{argmin}_{\mathbf{w}} \left\{\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} + \beta\|\mathbf{w}\|_2^2 - \langle \mathbf{S}^\top\boldsymbol{\omega}_k - 2\mathbf{e}, \mathbf{w}\rangle\right\} \\
&= \operatorname*{argmin}_{\mathbf{w}} \left\{\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} + \frac{1}{2}\left\|\mathbf{w} - \frac{\mathbf{S}^\top\boldsymbol{\omega}_k - 2\mathbf{e}}{2\beta}\right\|_2^2\right\} \\
&= \mathbf{prox}_{\mathbb{I}\{\mathbf{w}\succeq\mathbf{0}\}}\left(\frac{\mathbf{S}^\top\boldsymbol{\omega}_k - 2\mathbf{e}}{2\beta}\right) \\
&= \max\left(\mathbf{0}, \frac{\mathbf{S}^\top\boldsymbol{\omega}_k - 2\mathbf{e}}{2\beta}\right)
\end{aligned}$$

as desired [cf. (13)]. The last equality follows from the fact that the proximal operator of $\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\}$ is the projection onto the non-negative orthant $\mathbf{w} \succeq \mathbf{0}$.

To arrive at the update of $\mathbf{u}_k$ in (14), it suffices to start from (19) and recall that the proximal operator of $Lg(\mathbf{w}) = -L\alpha\mathbf{1}^\top \log(\mathbf{w})$ is given by (see e.g., [15] and [29, Proposition 2])

$$\mathbf{prox}_{Lg}(\mathbf{w}) = \frac{\mathbf{w} + \sqrt{\mathbf{w}^2 + 4\alpha L\mathbf{1}}}{2},$$

where the square and square root are understood to be taken element-wise. Evaluating the proximal operator at $\mathbf{S}\bar{\mathbf{w}}_k - L\boldsymbol{\omega}_k$, the result follows. □

Dual suboptimality

Recall one of the test cases in Section IV-A, where the goal was to recover a 2-block Stochastic Block Model (SBM) with $N = 200$ nodes from $P = 1000$ synthetically-generated smooth signals. In Fig. 2 we depict the evolution of the dual suboptimality $\varphi(\boldsymbol{\lambda}_k) - \varphi(\boldsymbol{\lambda}^\star)$ for the FDPG (Algorithm 1) and DPG iterations. As expected, the proposed FDPG solver markedly outperforms its non-accelerated counterpart in terms of convergence rate and wall-clock time; 0.049s for FDPG and 0.113s for DPG to attain a suboptimality gap of 10^{-8}. Similar behavior can be observed for graphs drawn from the Erdős-Rényi (ER) model and for different values of N and P. The corresponding plots are not included due to lack of space.