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/ICASSP49357.2023.10094914

DYNAMIC SIGNED GRAPH LEARNING

Abdullah Karaaslanli and Selin Aviyente

Department of Electrical and Computer Engineering, Michigan State University
East Lansing, MI, USA

ABSTRACT

An important problem in graph signal processing (GSP) is to infer the topology of an unknown graph from a set of observations on the nodes of the graph, i.e. graph signals. Recently, graph learning (GL) approaches have been extended to learn dynamic graphs from temporal graph signals. However, existing work primarily focuses on unsigned graphs and cannot learn signed graphs, which are important data structures that can represent the similarity and dissimilarity of the nodes. In this paper, we propose a dynamic signed GL (dynSGL) method based on the assumptions that (i) at each time point signals are smooth with respect to the signed graph, i.e. signal values at two nodes connected with a positive (negative) edge are similar (dissimilar) and (ii) evolution of the graph structures is smooth across time. The performance of dynSGL is evaluated on simulated data and shown to have higher accuracy compared to static signed and dynamic unsigned GL techniques. Application of the proposed method to a financial dataset gives important insights to the time-varying changes to the interactions between stocks.

***Index Terms*—** Signed Graphs, Graph Learning, Dynamic Graphs, Graph Signal Processing

1. INTRODUCTION

In many scientific disciplines, graphs are used to study relational data, where the nodes and edges of the graph represent the objects and their relations, respectively [1]. In most applications, in addition to having the edge weights, one may also have information on the nodes, in the form of node attributes. These node attributes can be viewed as graph signals. Examples of such data include user information on a social network and congestion level in a traffic network. Ubiquity of graph signals in many applications led to the field of graph signal processing (GSP), which aims to process graph signals by extending classical signal processing tools, such as Fourier transform, filtering or sampling to non-Euclidean data [2, 3].

An important problem in GSP is graph learning (GL), where one aims to infer the topology of an unknown graph from a set of observed graph signals [4, 5]. Various methods have been developed for this task by exploiting the graph Fourier domain representation of the signals, such as smoothness [6, 7] or stationarity [8, 9, 10]. However, these methods are limited to learning a single static network, which is restrictive when the connectivity between the nodes of a graph changes with time. In such cases, one needs to learn dynamic graphs, which arise in a variety of applications. For instance, gene regulations change over time with the development of cells or relations between stocks in financial networks vary with time. To this end, static GL approaches have been extended to learn dynamic graphs [11]. These extensions are based on the assumptions made on the evolution of the unknown dynamic graph. Methods proposed in [12, 13, 14] use the assumption that the graph structure varies slowly across time, while [15] assumes the spectrum of the graph changes.

Although aforementioned static and dynamic GL approaches are shown to perform well in some applications [16], they are all restricted to unsigned graphs, where edge weights can only be positive. However, many applications require signed graphs, where the edges can take on positive and negative weights, to model the interactions between the nodes. For instance, amity and enmity in social networks or activating and inhibitory gene regulations are best represented with signed graphs. Recent work has considered the extension of static GL methods to signed graphs. In [17], signed Laplacian [18] is used to define smooth graph signals on signed graphs. Based on this definition, a signal is smooth with respect to a signed graph if the signal values at two nodes connected with a positive (negative) edge are similar (dissimilar). However, dissimilarity is defined as the two nodes' signals having the same magnitude but different signs, which can be restrictive in applications where the signal values are always positive. To overcome this, [19] proposes to decompose a signed graph into two unsigned graphs based on the edge signs. The signed graph is then learned by assuming that signals are smooth and non-smooth with respect to positive and negative parts of the signed graphs, respectively. Similar to [17], signal values at two nodes connected with a positive (negative) edge are similar (dissimilar); but definitions of similarity and dissimilarity are more flexible.

In this work, our aim is to extend the method proposed in [19] to learn dynamic signed graphs, which are signed graphs whose structure changes with time. To this end, we propose a dynamic signed GL (dynSGL) algorithm with the following contributions:

- We define the temporal smoothness of signed graphs such that both the structure and signs of the edges of the signed graph change smoothly with time,
- The temporal smoothness is used to extend [19], leading to a non-convex problem which is solved using alternating direction method of multipliers (ADMM) with convergence guarantees.

The rest of the paper is organized as follows. Section 2 gives the necessary background material. dynSGL framework along with the optimization procedure is given in Section 3. Results and concluding remarks are presented in Sections 4 and 5, respectively.

2. BACKGROUND

2.1. Graphs and Graph Signals

A weighted undirected graph is defined as $G = (V, E, \mathbf{W})$ where V with $|V| = n$ is the node set and $E \subseteq V \times V$ is the edge set. $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the adjacency matrix with $W_{ij} = W_{ji}$ equal to

This work was supported in part by the NSF under CCF-2211645 and CCF-2006800.

the edge weight of $(i,j) \in E$. G is an unsigned graph if all edge weights are positive, otherwise it is signed. A signed graph G can be decomposed into two unsigned graphs $G^+ = (V, E^+, \mathbf{W}^+)$ and $G^- = (V, E^-, \mathbf{W}^-)$ based on the sign of its edge weights where $\mathbf{W}^+_{ij} = \mathbf{W}_{ij}$ if $\mathbf{W}_{ij} > 0$ and 0, otherwise and $\mathbf{W}^-_{ij} = |\mathbf{W}_{ij}|$ if $\mathbf{W}_{ij} < 0$ and 0, otherwise.

A graph signal defined on G is a vector $\mathbf{x} \in \mathbb{R}^n$ where x_i is the signal value on the ith node. For an unsigned graph G, the total variation of $\mathbf{x}$ on G is measured by its Dirichlet energy, i.e. $\mathrm{tr}(\mathbf{x}^\top \mathbf{L}\mathbf{x})$ where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian and $\mathbf{D}$ is the diagonal matrix of node degrees [2]. For a signed graph with Laplacian matrices $\mathbf{L}^+$ and $\mathbf{L}^-$ corresponding to G^+ and G^-, respectively, the total variation of $\mathbf{x}$ over the signed graph G can calculated using $\mathrm{tr}(\mathbf{x}^\top \mathbf{L}^+ \mathbf{x})$ and $\mathrm{tr}(\mathbf{x}^\top \mathbf{L}^- \mathbf{x})$ [19].

2.2. Graph Learning

Given a set of graph signals $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^p$ defined on an unknown graph G, the structure of G can be learned using the relation between the signals and the graph. For unsigned graphs, [6] propose to learn G based on the assumption that the signals are smooth with respect to G, i.e. they have small total variation. This assumption results in the following optimization problem where the total variation of signals is minimized with respect to the Laplacian of G:

$$\underset{\mathbf{L}\in\mathbb{L}}{\text{minimize}}\ \mathrm{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) + \alpha\|\mathbf{L}\|_F^2 \ \text{subject to}\, \mathrm{tr}(\mathbf{L}) = 2n, \quad (1)$$

where $\mathbf{X} \in \mathbb{R}^{n\times p}$ is the data matrix whose columns are $\mathbf{x}_i$'s, $\mathbb{L} = \{\mathbf{L} : L_{ij} = L_{ji} \leq 0\ \forall i \neq j,\ \mathbf{L}\mathbf{1} = \mathbf{0}\}$ is the set of valid Laplacian matrices. The Frobenius norm term in the objective function is added to control the density of the learned graph such that larger values of α result in denser graphs. Finally, the constraint is added to prevent the trivial zero solution.

The problem in (1) is extended in [19] to learn an unknown signed graph with the following two assumptions: signals have (i) small total variation with respect to G^+ and (ii) large total variation with respect to G^-. An unknown signed graph G is then learned with the following optimization problem where the total variation of signals is minimized with respect to $\mathbf{L}^+$ and maximized with respect to $\mathbf{L}^-$:

$$\begin{aligned}\underset{\mathbf{L}^+\in\mathbb{L},\mathbf{L}^-\in\mathbb{L}}{\text{minimize}}\ & \sum_{s\in\{+,-\}} \mathrm{tr}(\mathbf{K}^s\mathbf{L}^s) + \alpha_s\|\mathbf{L}^s\|_F^2 \\ \text{subject to}\ & \mathrm{tr}(\mathbf{L}^s) = 2n\ \forall s,\ \text{and}\ (\mathbf{L}^+, \mathbf{L}^-) \in \mathbb{C},\end{aligned} \quad (2)$$

where $\mathbf{K}^+ = \mathbf{X}\mathbf{X}^\top$, $\mathbf{K}^- = -\mathbf{X}\mathbf{X}^\top$ and we used the fact that $\mathrm{tr}(\mathbf{X}^\top\mathbf{L}\mathbf{X}) = \mathrm{tr}(\mathbf{X}\mathbf{X}^\top\mathbf{L})$. $\mathbf{L}^+$ and $\mathbf{L}^-$ are constrained to be in the set $\mathbb{C} = \{(\mathbf{L}^+, \mathbf{L}^-) : L^+_{ij} = 0$ if $L^-_{ij} \neq 0$ and $L^-_{ij} = 0$ if $L^+_{ij} \neq 0, \forall i \neq j\}$ in order to ensure that they are not non-zero for the same indices.

3. TIME-VARYING SIGNED GRAPH LEARNING

3.1. Problem Formulation

A dynamic signed graph is a sequence of graphs, $\mathcal{G} = \{G^t\}_{t=1}^T$, where $G^t = (V, E^t, \mathbf{W}^t)$ is the signed graph at time point t. Structure of $\mathcal{G}$ can be learned from a given set of time-varying signals $\mathcal{X} = \{\mathbf{X}^t\}_{t=1}^T$ where the columns of $\mathbf{X}^t \in \mathbb{R}^{n\times p^t}$ are the p^t graph signals defined on G^t and $|V| = n$. Let $G^{t,+}$ and $G^{t,-}$ be the decomposition of G^t based on the signs of its edges. In order to learn $\mathcal{G}$, we make the following assumptions:

- **AS1** Signals in $\mathbf{X}^t$ have small and large total variations on $G^{t,+}$ and $G^{t,-}$, respectively.
- **AS2** The topology and signs of the edges of G^t change smoothly over time.

Let $\mathbf{L}^{t,+}$ and $\mathbf{L}^{t,-}$ be the Laplacians of $G^{t,+}$ and $G^{t,-}$, then the above assumptions lead to the following optimization problem:

$$\begin{aligned}\underset{\{\mathbf{L}^{t,+},\mathbf{L}^{t,-}\}_{t=1}^T}{\text{minimize}}\ & \sum_{s\in\{+,-\}}\sum_{t=1}^T \mathrm{tr}(\mathbf{K}^{t,s}\mathbf{L}^{t,s}) + \alpha_s\|\mathbf{L}^{t,s}\|_F^2 \\ & + \sum_{s\in\{+,-\}}\sum_{t=2}^T \beta_s\|\mathbf{L}^{t,s} - \mathbf{L}^{t-1,s}\|^2_{F,off} \\ \text{subject to}\ & \mathrm{tr}(\mathbf{L}^{t,s}) = 2n\ \forall t,s,\ \text{and}\ (\mathbf{L}^{t,+}, \mathbf{L}^{t,-}) \in \mathbb{C},\end{aligned} \quad (3)$$

where $\mathbf{K}^{t,+} = \mathbf{X}^t\mathbf{X}^{t^\top}$, $\mathbf{K}^{t,-} = -\mathbf{X}^t\mathbf{X}^{t^\top}$ and $\|\cdot\|^2_{F,off}$ is the Frobenius norm of the off-diagonal entries. AS1 is satisfied by optimizing the total variation of $\mathbf{X}^t$ with respect to $\mathbf{L}^{t,+}$ and $\mathbf{L}^{t,-}$. AS2 is satisfied by regularizing the distance between $\mathbf{L}^{t,s}$ and $\mathbf{L}^{t-1,s}$, which ensures that the temporal evolution of $\mathcal{G}$ is smooth. Finally, the constraints are the same as those in (2).

3.2. Optimization

In order to optimize (3), we first vectorize the problem where the upper triangular parts of the Laplacians are learned. Define the operators $\mathrm{upper} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n(n-1)/2}$ and $\mathrm{diag} : \mathbb{R}^{n\times n} \to \mathbb{R}^n$, which take a square matrix and return the upper triangular and diagonal parts of the input matrix, respectively. Let $\mathbf{k}^{t,s} = \mathrm{upper}(\mathbf{K}^{t,s})$, $\mathbf{d}^{t,s} = \mathrm{diag}(\mathbf{K}^{t,s})$ and $\boldsymbol{\ell}^{t,s} = \mathrm{upper}(\mathbf{L}^{t,s})$. Define matrix $\mathbf{S} \in \mathbb{R}^{n\times n(n-1)/2}$ such that $\mathbf{S}\boldsymbol{\ell}^{t,s} = -\mathrm{diag}(\mathbf{L}^{t,s})$. Then, (3) can be vectorized as follows:

$$\begin{aligned}\underset{\{\boldsymbol{\ell}^{t,+},\boldsymbol{\ell}^{t,-}\}_{t=1}^T}{\text{minimize}}\ & \sum_{s\in\{+,-\}}\sum_{t=1}^T \{(2\mathbf{k}^{t,s} - \mathbf{S}^\top\mathbf{d}^{t,s})^\top\boldsymbol{\ell}^{t,s} + \alpha_s\|\mathbf{S}\boldsymbol{\ell}^{t,s}\|_2^2 \\ & + 2\alpha_s\|\boldsymbol{\ell}^{t,s}\|_2^2\} + \sum_{s\in\{+,-\}}\sum_{t=2}^T \beta_s\|\boldsymbol{\ell}^{t,s} - \boldsymbol{\ell}^{t-1,s}\|_2^2 \\ \text{subject to}\ & \mathbf{1}^\top\boldsymbol{\ell}^{t,s} = -n,\ \boldsymbol{\ell}^{t,s} \leq 0, \forall t,s,\ \text{and}\ \boldsymbol{\ell}^{t,+} \perp \boldsymbol{\ell}^{t,-}\ \forall t,\end{aligned} \quad (4)$$

where the first term corresponds to the total variation term in (3), the second and third terms correspond to the Frobenius norm controlling the density of the learned graphs and the last term is the regularizer for temporal smoothness. The first constraint prevents the trivial zero solution and $\boldsymbol{\ell}^{t,+} \perp \boldsymbol{\ell}^{t,-}$ along with the non-positivity constraint are called complementarity constraints [20] and correspond to $(\mathbf{L}^{t,+}, \mathbf{L}^{t,-}) \in \mathbb{C}$. This problem is non-convex due to complementarity constraints. ADMM is shown to converge for problems with complementarity constraints under some conditions [21].

To solve (4) with ADMM, we first introduce auxiliary variables $\mathbf{v}^{t,s} = \boldsymbol{\ell}^{t,s},\ \forall t,s$ and then rewrite (4) in ADMM form:

$$\begin{aligned}\underset{\substack{\{\boldsymbol{\ell}^{t,+},\boldsymbol{\ell}^{t,-},\\ \mathbf{v}^{t,+},\mathbf{v}^{t,-}\}_{t=1}^T}}{\text{minimize}}\ & f(\{\boldsymbol{\ell}^{t,+},\boldsymbol{\ell}^{t,-}\}_{t=1}^T) + \sum_{t=1}^T\{\imath_C(\mathbf{v}^{t,+},\mathbf{v}^{t,-}) \\ & + \imath_H(\boldsymbol{\ell}^{t,+}) + \imath_H(\boldsymbol{\ell}^{t,-})\} \\ \text{subject to}\ & \boldsymbol{\ell}^{t,s} = \mathbf{v}^{t,s},\ \forall t,s,\end{aligned} \quad (5)$$

where $f(\cdot)$ is the objective function of (4), $\imath_C$ and $\imath_H$ are the indicator functions for the complementarity constraint and $\mathbf{1}^\top\boldsymbol{\ell}^{t,s} = -n$,

respectively. From the augmented Lagrangian of (5), ADMM steps at kth iteration can be found as follows:

$$\{\widehat{\mathbf{v}}^{t,+},\widehat{\mathbf{v}}^{t,-}\}_{t=1}^{T} = \underset{\mathbf{v}^{t,+},\mathbf{v}^{t,-}}{\operatorname{argmin}} \imath_C(\mathbf{v}^{t,+},\mathbf{v}^{t,-}) + \sum_{s\in\{+,-\}} \frac{\rho}{2}\|\widehat{\boldsymbol{\ell}}^{t,s} - \mathbf{v}^{t,s} + \widehat{\widehat{\mathbf{y}}}^{t,s}\|_2^2, \tag{6}$$

$$\{\widehat{\boldsymbol{\ell}}^{t,+},\widehat{\boldsymbol{\ell}}^{t,-}\}_{t=1}^{T} = \underset{\{\boldsymbol{\ell}^{t,+},\boldsymbol{\ell}^{t,-}\}_{t=1}^{T}}{\operatorname{argmin}} f(\{\boldsymbol{\ell}^{t,+},\boldsymbol{\ell}^{t,-}\}_{t=1}^{T}) + \sum_{s\in\{+,-\}}\sum_{t=1}^{T} \imath_H(\boldsymbol{\ell}^{t,s}) + \frac{\rho}{2}\|\boldsymbol{\ell}^{t,s} - \widehat{\mathbf{v}}^{t,s} + \widehat{\widehat{\mathbf{y}}}^{t,s}\|_2^2, \tag{7}$$

$$\widehat{\mathbf{y}}^{t,s} = \widehat{\widehat{\mathbf{y}}}^{t,s} + \widehat{\boldsymbol{\ell}}^{t,s} - \widehat{\mathbf{v}}^{t,s},\ \forall t,s, \tag{8}$$

where $\widehat{\ }$ and $\widehat{\widehat{\ }}$ represent the values of the variables at the kth and $(k-1)$th iterations, respectively. $\mathbf{y}^{t,s}$ is the Lagrangian multiplier related to $\mathbf{v}^{t,s} = \boldsymbol{\ell}^{t,s}$, and ρ is the parameter of the augmented Lagrangian. The problem in (6) can be solved for each $(\mathbf{v}^{t,+},\mathbf{v}^{t,-})$ separately, where each subproblem is the projection onto the complementarity set. The problem in (7) is separable across $\{\boldsymbol{\ell}^{t,+}\}_{t=1}^{T}$ and $\{\boldsymbol{\ell}^{t,-}\}_{t=1}^{T}$, where both subproblems can be solved with block coordinate descent [22].

4. RESULTS

4.1. Simulated Data

In this section, the performance of dynSGL[1] is assessed on simulated datasets. The results are compared to signed graph learning (SGL, see the optimization problem in (2)) and dynamic unsigned graph learning (dynGL, the optimization problem in (3) without the $\mathbf{L}^{t,-}$ terms). SGL and dynSGL require the selection of α_s parameters, which control the density of the positive and negative edges of the learned signed graphs. We selected these parameters such that the density of positive and negative edges are around 0.1. Similarly, for dynGL, α is selected such that the learned unsigned graph has an edge density of 0.1. dynSGL requires the selection of β_s, which determines the amount of temporal smoothness. We set it to a value such that the correlation between G^t and G^{t-1} is around 0.75 $\forall t$. For dynGL, β is set the same way. For the performance metric, we used the multiclass F1 score, which is the average of F1+ and F1-, which are calculated by comparing the positive and negative edges of the learned graph to the positive and negative edges of the ground truth, respectively. For dynGL, which returns unsigned graphs, learned edges are compared to the positive and negative edges separately to calculate F1+ and F1-.

Data Generation: Given a signed graph G with n nodes, a synthetic graph signal $\mathbf{x}$ can be generated based on AS1. Let $\mathbf{L}^+$ and $\mathbf{L}^-$ be the Laplacians of G and let $\mathbf{L}^+ = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ and $\mathbf{L}^- = \mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^\top$ be their eigendecompositions. AS1 states that $\mathbf{x}$ is the linear combination of the eigenvectors of $\mathbf{L}^+$ and $\mathbf{L}^-$. In particular, a graph signal $\mathbf{x}$ that follows AS1 can be generated by $\mathbf{x} = (\mathbf{U}h_1(\boldsymbol{\Lambda})\mathbf{U}^\top + \mathbf{V}h_2(\boldsymbol{\Sigma})\mathbf{V}^\top)\mathbf{x}_0 + \boldsymbol{\epsilon}$ where $h_1(\boldsymbol{\Lambda}) = \boldsymbol{\Lambda}^\dagger/\|\boldsymbol{\Lambda}^\dagger\|_F$ is a low-pass graph filter, $h_2(\boldsymbol{\Sigma}) = \boldsymbol{\Sigma}/\|\boldsymbol{\Sigma}\|_F$ is a high-pass graph filter, † is the pseudo-inverse operator, $\mathbf{x}_0 \sim \mathcal{N}(0,\mathbf{I})$ and $\boldsymbol{\epsilon}$ is additive white Gaussian noise. Using this synthetic graph signal generation process, we generate data for dynamic signed GL problem as follows. We first generate G^1 from either an Erdős–Rényi (ER) model ($n = 100$, $p_{ER} = 0.2$) or Barabási–Albert (BA) model ($n = 100$, $m_{BA} = 8$). Half of the edges of G^1 are selected randomly and set to be negative edges, while the remaining ones are set as positive edges. $\mathcal{G} = \{G^t\}_{t=1}^{T}$ with $T = 10$ is then constructed where each G^t is generated by perturbing r fraction of edges of G^{t-1}. Finally, p graph signals are generated from each G^t following the graph signal generation procedure described above to construct the data matrices $\mathbf{X}^t$. White Gaussian noise with variance equal to 10% of the signal power is also added to the data matrices. Each experiment is repeated 20 times and average F1 score of these runs are reported.

Experiment 1: We first study how the different methods are affected from varying the number of signals. In particular, at each time point we generate p graph signals where p ranges from 20 to 200. Perturbation fraction r is set to 0.05. In Fig. 1, F1 scores of the graphs learned by the different methods are plotted as a function of p. All methods perform better with increasing number of signals as expected. dynGL is the worst performing method, since it can only learn unsigned temporal graphs. dynSGL performs better than SGL for all values of p, i.e. there is up to 20% increase in F1 score when graphs are learned with dynSGL. This result indicates that imposing temporal smoothness leads to better learning. Finally, these observations hold for both ER and BA models.

Experiment 2: Next, we study how the performance of the different methods changes when the value of r, i.e. amount of temporal variation, is increased. We set the number of signals p to 100 and the value of r varies from 0.05 to 0.5. r controls the amount of the temporal smoothness in the ground truth graph, that is for larger values of r, $\mathcal{G}$ is temporally less smooth. Results for the different methods are given in Fig. 2. Since SGL learns a graph at each time individually, its performance is not affected from the change in r. However, F1 scores of both dynSGL and dynGL drop, since the ground truth graphs do not fit AS2 anymore as r increases. Similar to the first experiment, dynGL is the worst performing method. dynSGL performs better than SGL up to $r = 0.3$, then its performance gets worse, which is expected as imposing temporal smoothness would degrade the quality of the learned graphs when AS2 is not valid. As above, the observations hold for both random graph models.

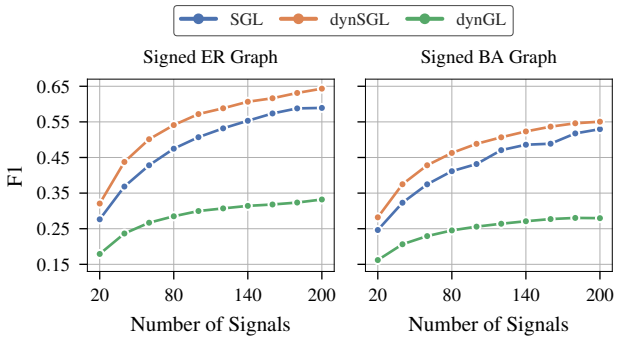

Fig. 1. Performance of methods on simulated dynamic signed graphs as a function of the number of signals at each time point. Left panel shows the performance for the model and the right one shows the performance for BA model.

4.2. Real Data

dynSGL is applied to a financial dataset to learn the time-varying interactions between 89 stocks from NASDAQ 100 index. Adjusted

[1]Codes can be found at https://github.com/SPLab-aviyente/dynSGL

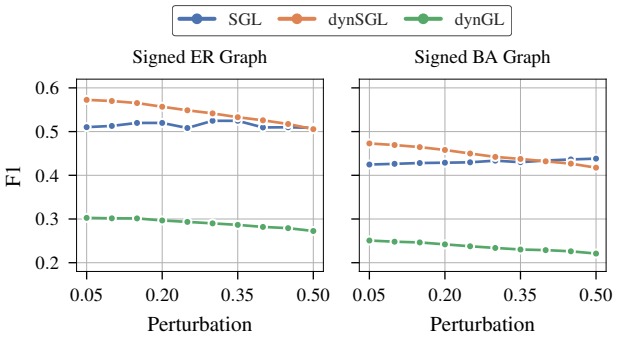

Fig. 2. Performance of methods on simulated dynamic signed graphs as a function of perturbation amount over time. Left panel shows the performance for ER model and the right panel shows the performance for BA model.

day close prices of stocks for the year 2020[2] are used to calculate the daily log-returns, which are considered as graph signals. We split the dataset into sliding overlapping time windows, where graph signals in each window are used to construct $\mathcal{X} = \{\mathbf{X}^t\}_{t=1}^{T}$'s with $T = 49$. Each window includes data from 10 days, i.e. $\mathbf{X}^t \in \mathbb{R}^{89\times 10}$, and the overlap between consecutive windows is 5 days. dynSGL is applied to $\mathcal{X}$ to find the temporal signed graph $\mathcal{G} = \{G^t\}_{t=1}^{49}$ where α_s's are selected such that the positive and negative edge densities of G^t are 0.1 and β_s's are selected such that the correlation between G^t and G^{t-1} is 0.9. For comparison, SGL is also applied to each $\mathbf{X}^t$ separately to find G^t where α_s's are selected the same way as for dynSGL.

Since there is no ground truth temporal graph, we validate the learned $\mathcal{G}$ by analyzing its properties. In particular, sectoral division[3] of stocks is considered as the ground truth community structure and we calculate the signed modularity [24] with respect to this division at each time point. High values of signed modularity indicates that the graph is modular, i.e. positive edges are mostly within communities, while negative edges are mostly between communities. In Fig. 3, the signed modularity values for dynSGL and SGL are plotted across time. For dynSGL, a significant drop in modularity value is observed from February to March and another smaller drop happens from September to October. The former corresponds to the start of COVID-19 pandemic, and the latter is before US presidential election. Drop in modularity indicates that the modular structure of the financial network disappears, meaning that the prices of most stocks move in the same direction. This result is inline with [25], which finds increasing graph connectivity in financial data at the time of uncertainty using unsigned temporal graph learning. Comparing SGL and dynSGL, we observe that SGL's modularity values are noisier than those of dynSGL. Thus, significant drops in modularity are more visible for dynSGL than SGL. For instance, SGL's modularity values are very noisy after July, making drop due to US presidential election less visible. Finally, dynSGL learns more modular graphs than SGL, since it can preserve graph structure through temporal smoothness.

[2]Data is obtained from Yahoo! Finance using [23].

[3]Sector information is obtained from tradingview.com

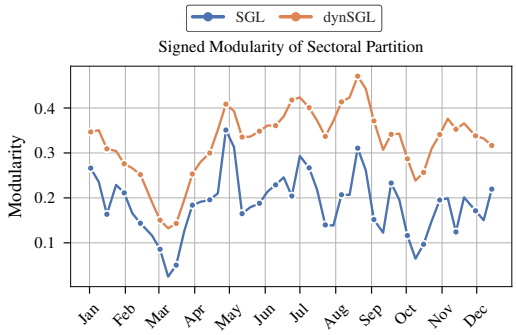

Fig. 3. Signed modularity values when learned graphs are partitioned based on the sectors of stocks.

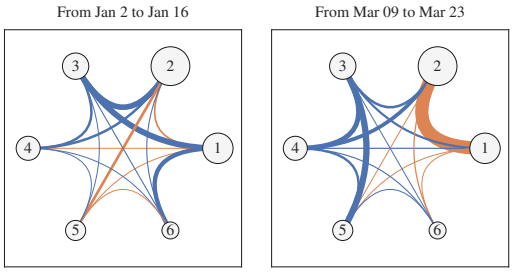

Fig. 4. The signed graphs induced by the sector division. Only sectors that has at least 5 stocks are shown. Blue and orange correspond to negative and positive edges, respectively. Node sizes are proportional to the number of stocks in the sectors. Sector names: 1) Electronic Technology, 2) Technology Services, 3) Health Technology, 4) Retail Trade, 5) Consumer Services, 6) Consumer Non-Durables.

To get a better insight into the graph structures across time, Fig. 4 shows the inter-sectoral edges of the two learned graphs, where each node corresponds to a sector. In January, most edges are negative, indicating strong modular structure. While in March, the strengths of negative edges between sectors 1 and 3 and 2 and 3 decrease and a very strong positive edge appears between sectors 1-2. This implies that sectors 1 and 2 become a single community, while the separation between sectors 1-3 and 2-3 reduces.

5. CONCLUSIONS

This paper introduced a new algorithm for learning temporal signed graphs. Temporal GL is important in a variety of applications where the interactions between the nodes of a graph change with time such as in biological, social and financial networks. While there has been some work on temporal GL for unsigned graphs, this paper is the first to extend this line of work to signed graphs that model similarity and dissimilarity of nodes. In particular, we extended the concept of graph signal smoothness and temporal smoothness to signed graphs and formulated the graph learning problem as an optimization problem. The results from both simulated and real data illustrate the effectiveness of the proposed method.

Future work will consider different extensions of the proposed formulation. First in the current formulation, we assumed that the number of nodes is constant across time. However, in many real applications nodes may appear or disappear with time. Second, current formulation uses squared Frobenius norm to regularize temporal variation of graphs, future work will consider other distances, such as L1-norm, which imposes sparse temporal variation.

6. REFERENCES

[1] Mark Newman, *Networks*, Oxford university press, 2018.

[2] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[3] Antonio Ortega, Pascal Frossard, Jelena Kovačević, José MF Moura, and Pierre Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[4] Xiaowen Dong, Dorina Thanou, Michael Rabbat, and Pascal Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 44–63, 2019.

[5] Georgios B Giannakis, Yanning Shen, and Georgios Vasileios Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 787–807, 2018.

[6] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6160–6173, 2016.

[7] Vassilis Kalofolias, "How to learn a graph from smooth signals," in *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 920–929.

[8] Bastien Pasdeloup, Vincent Gripon, Grégoire Mercier, Dominique Pastor, and Michael G Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE transactions on Signal and Information Processing over Networks*, vol. 4, no. 3, pp. 481–496, 2017.

[9] Santiago Segarra, Antonio G Marques, Gonzalo Mateos, and Alejandro Ribeiro, "Network topology inference from spectral templates," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 467–483, 2017.

[10] Michele Cirillo, Vincenzo Matta, and Ali H Sayed, "Estimating the topology of preferential attachment graphs under partial observability," *IEEE Transactions on Information Theory*, 2022.

[11] Alberto Natali, Elvin Isufi, Mario Coutino, and Geert Leus, "Learning time-varying graphs from online data," *IEEE Open Journal of Signal Processing*, vol. 3, pp. 212–228, 2022.

[12] Vassilis Kalofolias, Andreas Loukas, Dorina Thanou, and Pascal Frossard, "Learning time varying graphs," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Ieee, 2017, pp. 2826–2830.

[13] Koki Yamada, Yuichi Tanaka, and Antonio Ortega, "Time-varying graph learning based on sparseness of temporal variation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5411–5415.

[14] Madeline Navarro, Yuhao Wang, Antonio G Marques, Caroline Uhler, and Santiago Segarra, "Joint inference of multiple graphs from matrix polynomials," *J. Machine Learning Research*, 2020.

[15] Stefania Sardellitti, Sergio Barbarossa, and Paolo Di Lorenzo, "Online learning of time-varying signals and graphs," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5230–5234.

[16] Gonzalo Mateos, Santiago Segarra, Antonio G Marques, and Alejandro Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 16–43, 2019.

[17] Gerald Matz and Thomas Dittrich, "Learning signed graphs from data," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5570–5574.

[18] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak, "Spectral analysis of signed graphs for clustering, prediction and visualization," in *Proceedings of the 2010 SIAM international conference on data mining*. SIAM, 2010, pp. 559–570.

[19] Abdullah Karaaslanli, Satabdi Saha, Selin Aviyente, and Tapabrata Maiti, "scsgl: kernelized signed graph learning for single-cell gene regulatory network inference," *Bioinformatics*, vol. 38, no. 11, pp. 3011–3019, 2022.

[20] Holger Scheel and Stefan Scholtes, "Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity," *Mathematics of Operations Research*, vol. 25, no. 1, pp. 1–22, 2000.

[21] Yu Wang, Wotao Yin, and Jinshan Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.

[22] Hao-Jun Michael Shi, Shenyinying Tu, Yangyang Xu, and Wotao Yin, "A primer on coordinate descent algorithms," *arXiv preprint arXiv:1610.00040*, 2016.

[23] Ran Aroussi, "yfinance," `https://github.com/ranaroussi/yfinance`.

[24] Sergio Gómez, Pablo Jensen, and Alex Arenas, "Analysis of community structure in networks of correlated data," *Physical Review E*, vol. 80, no. 1, pp. 016114, 2009.

[25] Seyed Saman Saboksayr, Gonzalo Mateos, and Mujdat Cetin, "Online graph learning under smoothness priors," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1820–1824.