ESTIMATION OF TIME-VARYING GRAPH TOPOLOGIES FROM GRAPH SIGNALS

Yuhao Liu,⋆ Chen Cui,† Marzieh Ajirak,† and Petar M. Djurić†

⋆Department of Applied Mathematics and Statistics
†Department of Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY 11794

ABSTRACT

In science and engineering, we often deal with signals that are acquired from time-varying systems represented by dynamic graphs. We observe these signals, and the interest is in finding the time-varying topology of the graphs. We propose two Bayesian methods for estimating these topologies without assuming any specific functional relationships among the signals on the graphs. The two methods exploit Gaussian processes, where the first method uses the length scale of the kernel and relies on variational inference for optimization, and the second method is based on derivatives of the functions and Monte Carlo sampling. Both methods estimate the time-varying topologies of the graphs sequentially. We provide numerical tests that show the performance of the methods in two settings.

***Index Terms*—** time-varying graphs, estimation of topology, variational inference, sequential estimation, Monte Carlo sampling

1. INTRODUCTION

In practical data science problems, unveiling the underlying structure of observed data is essential since they provide information about the system that generated the data. Systems are often described by graphs and signals on the graphs. Studying varying patterns of graph signals over time and the dependencies among the signals on the graphs allows for inferring the variability of the graphs over time and, thereby, the evolution of the respective systems that the graphs model. Given the significance of the problem, plenty of research has been conducted in different fields of science and engineering, including biology [1], finance [2], and network science [3]. For example, in neuroscience, the interest could be recovering brain activity while a subject is working on particular tasks. In finance, it is important to reveal interconnections of entities from financial market data over time. In social studies, often the goal is to construct a social network of individuals and groups from data collected from social media platforms.

Much work in estimating graph topologies has been done on static graphs. The work in this space can be grouped into two categories, and they are based on the type of topology of the studied graphs. One is focused on undirected graphs, and the other on directed graphs. Undirected graph inference produces estimates of symmetric adjacency matrices and is somewhat less challenging, but it sacrifices the representation of single-way dependencies on the graphs [4], [5]. To overcome the limitations of undirected graphs, methods like Granger causality [6], vector autoregressive (VAR) models [7], and structural equation models (SEMs) [8] have been introduced to infer the directed dependencies on the graphs. Further, extensions of these methods to nonlinear directed graphs have been proposed in [9]. On the other hand, research on dynamic graphs has been much more sparse, even though real systems modeled by dynamic graphs are very common. Work reported in this area relies on graphical Lasso-based methods that can be applied to both continuous and sudden network changes, but only for undirected graphs, [9]. In [10, 11, 12], the authors extended the SEMs to topology inference in dynamic settings.

In this paper, we propose two novel methods for estimating time-varying topologies of graphs from time-series data observed at the nodes of the graphs. The objective is to infer the causation and variation of the graph in time. We make very mild assumptions about the functional relationships among the signals on the graph, and they also need to be learned. Specifically, we propose to exploit random feature-based Gaussian processes (RFGPs) to model the unknown mapping functions. Compared with exact GPs used in [13], RFGPs reduce the complexity in computation from cubic to the quadratic of the size of observed data. Unlike inducing point-based methods, which are also popular in the GPs literature, the random feature framework does not require any matrix decomposition. Instead, it only needs matrix products, which significantly boosts the speed of computations. We also propose to use variational inference (VI) and sequential Monte Carlo (SMC) to learn the desired functions. Specifically, with VI-RFGPs, we learn the hyperparameters of the pre-defined kernels, which in turn reflect the topology of the graph in an online manner. With SMC-RFGPs, we estimate the weights of the random Fourier features used by the RFGPs by computing the derivatives with respect to the various inputs. Thereby, we can differentiate the time-varying positive and negative relationships on the directed graph.

The paper is organized as follows. In Section II, we provide a brief overview of the background, and in Section III, we explain the addressed model. We present our proposed solutions in Section IV. In Section V, we illustrate the performance of the proposed methods, and in the last section, we conclude the paper with some final remarks.

2. BACKGROUND

2.1. Graphs and Graph Signals

Consider a graph denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ where $\mathcal{V}$ is a set of N nodes, $\mathcal{E}$ is a set of edges, and $\mathbf{W}$ is the graph's weighted adjacency matrix. One way to describe the topology of a graph $\mathcal{G}$ is through the adjacency matrix $\mathbf{W}$, which can be symmetric or asymmetric and, thus, implying if the graph is undirected or directed, respectively. The (n, m)th entry of $\mathbf{W}$ is $w_{nm} \in [0, 1]$. For an undirected graph, $w_{nm} = w_{mn}$, and if its value is non-zero, there is an edge *between*

This work has been supported by NSF under Award 2212506.

node m and node n. Similarly, for a directed graph, if w_{nm} is non-zero, there is an edge *pointing from* node m *to* node n. The entries $w_{n,m} \in \mathbb{R}$ represent the strength of the coupling of the mth and nth nodes.

On a given graph, its signals are defined as follows. Consider an unordered set of data $\mathcal{S} = \{s_{\alpha_0}, ..., s_{\alpha_N}\}$, which are associated with $\mathcal{G}$. We assume each datum in $\mathcal{S}$ is assigned to a single specific node in $\mathcal{G}$. Then the data $\mathcal{S}$ are ordered by the nodes in $\mathcal{G}$ and are given by an N-tuple $\mathbf{s} = \{s_1, ...s_n, ..., s_N\}$. We can think of $\mathbf{s}$ as a graph signal over $\mathcal{G}$ [14]. The nth element s_n in $\mathbf{s}$ is indexed by the node n of $\mathcal{G}$.

2.2. Gaussian Processes

GPs are a class of stochastic processes that are used in machine learning for modeling functions [15]. More specifically, let $(\mathbf{x}_t, y_t)$, $t = 1, 2, \ldots, T$, be T input-output values, where $\mathbf{y} = [y_1\ y_2 \ldots y_T]^\top$, and $\mathbf{y} = \mathbf{f}(\mathbf{X})$, with $\mathbf{f} \in \mathbb{R}^{T\times 1}$ and $\mathbf{X} \in \mathbb{R}^{T\times d_x}$ being a matrix whose rows represent the inputs to the function $\mathbf{f}$, that is,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_T^\top \end{bmatrix}, \qquad \mathbf{y} = \mathbf{f}(\mathbf{X}) = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_T) \end{bmatrix}. \tag{1}$$

The idea behind GPs is that function samples are jointly drawn from a Gaussian distribution. Mathematically, we have $\mathbf{f} \sim \mathcal{GP}\left(\mathbf{m}(\mathbf{X}), \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})\right)$, where $\mathbf{m}(\mathbf{X})$ is the mean function, $\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})$ is the covariance (kernel) function of the process, and $\boldsymbol{\theta}$, is a vector of hyperparameters of the GP, i.e.,

$$\begin{aligned} \mathbf{m}(\mathbf{X}) &= \mathbb{E}[\mathbf{f}(\mathbf{X})], \\ [\mathbf{K}_{\boldsymbol{\theta}}(\mathbf{X})]_{ij} &= \mathbb{E}[(f(\mathbf{x}_i) - m(\mathbf{x}_i))(f(\mathbf{x}_j) - m(\mathbf{x}_j))]. \end{aligned} \tag{2}$$

In practice, without loss of generality, we let the mean function to be set to $\mathbf{0}$. By definition, the kernel has to be positive definite [15].

2.3. Random Feature-Based Gaussian Processes

The computation complexity of GPs explodes with the increase of the data size T because of the need to invert a $T \times T$ matrix. Several approximation methods have been proposed to address this problem. One widely used approximation is RFGPs, which construct the GPs with features that come from a feature space. One possibility is to use Fourier features, which are obtained from the power spectral density of the adopted kernel. We can express them by

$$\boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x}) = \frac{1}{\sqrt{J}}[\sin(\mathbf{x}^\top \mathbf{v}^1), \cos(\mathbf{x}^\top \mathbf{v}^1), \ldots, \sin(\mathbf{x}^\top \mathbf{v}^J), \cos(\mathbf{x}^\top \mathbf{v}^J)], \tag{3}$$

where J is the number of frequencies sampled from the power spectral density of the kernel, $\mathbf{v}^{(1:J)} = \{\mathbf{v}^j\}_{j=1}^J =: \mathbf{V}$ [16]. If the kernel is shift-invariant, then the GPs can be approximated as follows:

$$f \approx \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{x})\mathbf{w}, \tag{4}$$

where $\mathbf{w}$ is a vector of unknown linear coefficients.

3. MODEL DESCRIPTION

Let $\mathbf{y}(t) \in \mathbb{R}^{N\times 1}$ represent a vector of graph signals collected from the graph $\mathcal{G}$ from all its nodes at time t, where $y_n(t)$ denotes the graph signal of node n at time t. Further, we assume that the graph signal $y_n(t)$ is a function of the previous data of all the nodes (or some of the nodes). Specifically, consider the data model

$$\mathbf{y}(t) = [f_{1t}(\mathbf{Y}^M(t)), \ldots, f_{Nt}(\mathbf{Y}^M(t))]^\top + \boldsymbol{\epsilon}(t), \tag{5}$$

where $\mathbf{Y}^M(t) := [\mathbf{y}(t-M)^\top, \ldots, \mathbf{y}(t-m)^\top, \ldots, \mathbf{y}(t-1)^\top] \in \mathbb{R}^{1\times MN}$, M is a discrete delay time, and $\mathbf{y}(t-m)$ are the graph signals over all the nodes at time $t-m$. The model noise is $\boldsymbol{\epsilon}(t) \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon^2\mathbf{I})$, and f_{nt} is the function of node n that maps all the inputs of that node to its observed output (the actual observation at that node, $y_n(t)$). We also assume an independent relationship between the functions of the nodes. Thus, for the nth entry of $\mathbf{y}(t)$ we can write

$$y_n(t) = f_{nt}(\mathbf{Y}^M(t)) + \epsilon_n(t). \tag{6}$$

Further, we assume the function f_{nt} is a GP with time-varying hyperparameters, i.e., $f_{nt}(\cdot) \sim \mathcal{GP}(m_t(\cdot), k_n(\cdot,\cdot|\boldsymbol{\theta}_t))$. The work in [13] assumes that the function is time-invariant, but in the present work, the function f_{nt} is time-variant. This means that the relationships among the nodes vary over time, reflecting a changing topology of the graph. For tracking the changes in the topology on the graph, we propose to exploit RFGPs. In the sequel, we only consider the function of node n and, thus, ignore the subscript n for simplification in notation. Then the model shown in (6) can be approximated by

$$f_t(\mathbf{Y}^M(t)) \approx \boldsymbol{\phi}_{\mathbf{v}}(\mathbf{Y}^M(t))\mathbf{w}(t). \tag{7}$$

It is important to note that in (7), the coefficients of the model $\mathbf{w}(t)$ vary with time.

4. PROPOSED SOLUTIONS

In this paper, we propose two different online Bayesian approaches to measure the importance of the edges of the graphs. One method uses the length scale of the kernel and exploits VI while the other method relies on derivatives of the functions computed by SMC.

4.1. Length Scales with Variational Inference

From [13], the length scales of the ARD RBF kernel can act as important indicators of whether the edges are important or not, where the ARD RBF kernel has the form

$$k\left(\mathbf{x}, \mathbf{x}'\right) = \sigma_\theta^2 \exp\left(-\frac{1}{2}\sum_{j=1}^{d_x} \frac{\left(x_j - x_j'\right)^2}{l_j^2}\right), \tag{8}$$

where $l = \{l_j\}_{j=1}^{d_x}$ are length scales and represent hyperparameters. A larger length scale means that the learned GP varies less in that dimension, and hence less importance is given to that dimension. Although there is no explicit expression of the length scale l in (4), the random features are, in fact, sampled from the power spectral density of the kernel. We note that the hyperparameters of the kernel are optimized simultaneously.

The VI aims at finding variational distributions $q(\mathbf{w})$ and $q(\mathbf{V})$ that approximate the true posterior distributions $p(\mathbf{w}|\mathbf{X}, \mathbf{Y})$ and $p(\mathbf{V}|\mathbf{X}, \mathbf{Y})$, respectively. Defining the marginalized log-likelihood

$L = \sum_{n=1}^{N} \log p(y_n|\mathbf{x})$ and $L' = \sum_{n=1}^{N} \mathbb{E}_{q(\mathbf{w})} \log p(y_n|\mathbf{x}, \mathbf{w})$, we obtain

$$L \geq L' - \mathrm{KL}\left[q(\mathbf{w})||p(\mathbf{w})\right] - \mathrm{KL}_\theta\left[q(\mathbf{V})||p(\mathbf{V})\right], \quad (9)$$

where KL stands for Kullback-Leibler divergence, $p(\mathbf{w})$ is the prior distribution of the hidden variables $\mathbf{w}$, and $p(\mathbf{V})$ is the power spectral density of the kernel that is related to the hyperparameters of the chosen kernel. Note that the KL divergence regularizes $\mathbf{w}$ and $\mathbf{V}$ automatically, which avoids overfitting when $||\mathbf{w}||$ or $||\mathbf{V}||$ are too large. In practice, the real marginalized log-likelihood L is usually intractable. Therefore, we utilize the right side of the inequality, named evidence lower bound (ELBO), which is an approximation to L and tractable to handle. Finally, we can follow the gradient descent-based algorithm to find the optimal hyperparameters, including the length scales. The property of the gradient descent-based algorithm naturally leads to online learning of the parameters and consequently adapts to the time-varying graphs.

4.2. Derivatives with Monte Carlo Sampling

There are two major disadvantages to using length scales. First, the length scale can only provide relative importance for other edges. Second, the length scale must be positive, and then we cannot judge whether the edge has a positive or negative contribution. To handle these two limitations, we propose using the function's derivatives to measure the importance of edges. The derivatives of the random feature-based function

$$f = \phi(\mathbf{x})\mathbf{w} = \mathbf{w}^\top[\cos(\mathbf{V}^\top\mathbf{x}), \sin(\mathbf{V}^\top\mathbf{x})]/\sqrt{J} \quad (10)$$

with respect to $\mathbf{x}$ is

$$f' = \mathbf{w}^\top[\mathrm{diag}\left(-\sin(\mathbf{V}^\top\mathbf{x})\right), \mathrm{diag}\left(\cos(\mathbf{V}^\top\mathbf{x})\right)]\mathbf{V}^\top/\sqrt{J}. \quad (11)$$

We observe that the derivatives reflect how sensitive the output variables are to input variables [17]. The higher the absolute value of the sensitivity to a particular input variable is, the more critical that variable is to the output variable.

The VI can obtain the derivatives when the local parameterization trick is not applied. However, this would lead to a dramatic decline in the accuracy of the VI. To obtain steady and precise estimates of the derivatives, we propose to use the approach from [16].

We first apply the VI to a small set of data to gain a proper initialization of the distribution of $\mathbf{V}$, i.e., the posterior of the power density of the kernel. Then we work with an ensemble of different sets of $\mathbf{V}$, which are randomly sampled from the posterior. If we denote with $w_t^s = p(s|y_{1:t}, \mathbf{x}_{1:t})$ the posterior weight of the sth feature at time t, it can be updated by the Bayes' formula

$$w_t^s \propto w_{t-1}^s p(y_t|s, y_{1:t-1}, \mathbf{x}_{1:t-1}). \quad (12)$$

The estimate of f_t is obtained from the ensemble of GPs, i.e.,

$$\widehat{f}_t = \sum_{s=1}^{S} w_t^s \widehat{f}_t^s. \quad (13)$$

Therefore, the estimated derivatives are naturally

$$\widehat{f'}_t = \sum_{s=1}^{S} w_t^s \widehat{f'}_t^s. \quad (14)$$

Both the estimated functions and their derivatives are time-varying, and they adjust to the time-varying graphs.

5. NUMERICAL RESULTS

5.1. Length Scale with Large Scale Data

To test the ability of the proposed method on a relatively large network, we applied it on a 100 dimensional VAR model with a sudden change, i.e., $\mathbf{y}(t) = \mathbf{W}_s\mathbf{y}(t-1), s = 1, 2$. The weighted adjacency matrices $\mathbf{W}$ are ER random graphs with edge probability equal to 0.05, and $\mathbf{W}_1$ and $\mathbf{W}_2$ are independently drawn. We drew the initial values of the data and the noise from a multivariate Gaussian, i.e., $\mathbf{y}(0) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and $\sigma_\epsilon^2 = 0.5$.

We randomly chose node 66 as the target node, and nodes 1 to 100, including node 66 itself, as the input nodes. The first graph topology (T1) lasts from 0 to 50,00 time units, while the second graph topology (T2) is from 5,000 to 10,000 time units. The important nodes and their weights are listed below for both topologies:

T1:	nodes:	[30,	46,	67,	69,	81,	85]
	weights:	[0.37,	0.17,	0.33,	0.48,	0.13,	0.01]
T2:	nodes:	[29,	67,	85,	93]		
	weights:	[0.14,	0.09,	0.25,	0.46]		

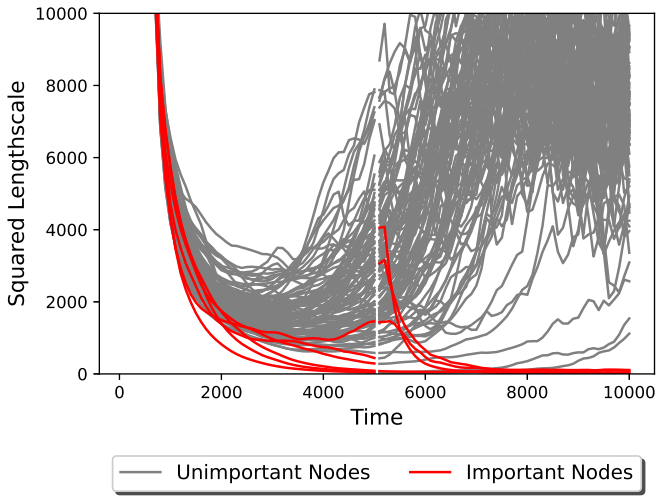

Fig. 1. The estimated squared length scales vary with time for all 100 dimensions.

The estimated squared length scales for every input node are shown in Fig. 1. The red lines represent the important nodes with non-zero weights, while the grey lines are the unimportant nodes with zero weights. For the duration of the first topology, the important nodes are grouped into three levels. Node 85 has a non-significant weight of 0.01 near zero, nodes 46 and 81 have weak significant weights of around 0.1, and nodes 30, 67, and 69 have significant weights of around 0.4. Recall that larger length scales refer to smaller node importance. From the figure, our method can recognize the important nodes quickly and also the levels of importance in terms of the values of squared length scales. Node 85, with non-significant weight, cannot be recognized with other unimportant nodes because of its low weight. However, nodes 46 and 81 can be recognized even with weak significant weights. Nodes 30, 67, and 69 have much less value of length scales compared with other nodes since they have a strong contribution to the target node. After the

change point at 5,000, the newly important nodes 29 and 93 have their estimated length scales decrease quickly, and the newly unimportant nodes 30, 46, 69, and 81 behave conversely. Note that node 85 had insignificant weight in T1 (0.01) and a significant value in T2 (0.25).

5.2. Derivatives with Smoothly Varying Graph

This example illustrates that the proposed method based on derivatives can recover the network topology with smooth changes. We apply this method on a ten dimensional VAR model, with the entries of the adjacency matrix changing with time according to

$$\begin{cases} w_{1,3}(t) = -0.00005t, & t < 10000, \\ w_{1,3}(t) = -0.5, & t \geq 10000, \end{cases} \tag{15}$$

$$\begin{cases} w_{1,5}(t) = -0.25 + 0.00005 * t, & t < 10000, \\ w_{1,5}(t) = 0.25, & t \geq 10000, \end{cases} \tag{16}$$

$$\begin{cases} w_{1,7}(t) = 0.5, & t < 5000, \\ w_{1,7}(t) = -0.00005 * t + 0.75, & 5000 \leq t < 15000, \\ w_{1,7}(t) = 0, & t \geq 15000. \end{cases} \tag{17}$$

Except for node one, the evolutions of the other nodes only depend on themselves (they have self-loops). The initial value of the data and the noise are randomly drawn from a multivariate Gaussian, i.e., $\mathbf{y}(0) \sim \mathcal{N}(\mathbf{0}, \sigma_\epsilon \mathbf{I})$, with $\sigma_\epsilon^2 = 0.5$.

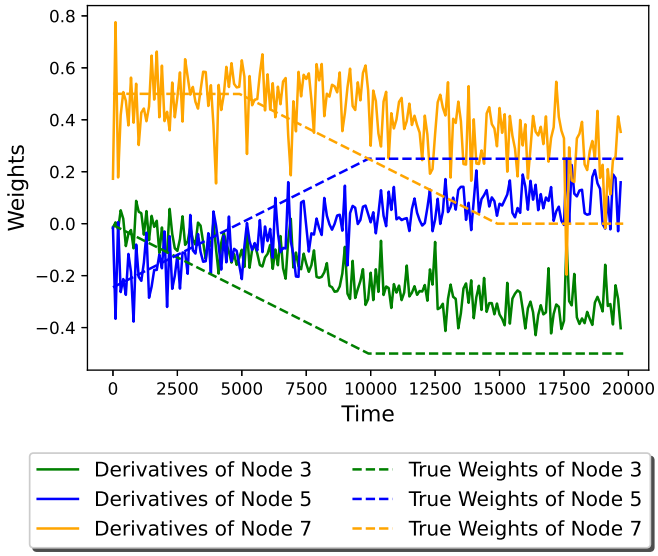

Fig. 2. The estimated derivatives vary with time for all three important nodes.

The estimated derivatives for the important nodes 3, 5, and 7 are shown in Fig. 2, and the remaining unimportant nodes are presented in Fig. 3. The solid lines represent the derivatives of important nodes, while the dashed lines are the true weights used in the data generating process. Specifically, the green, blue, and orange lines are for nodes 3, 5, and 7, respectively. From the figure, the estimated derivatives capture the varying weights correctly in the signs, values, and patterns.

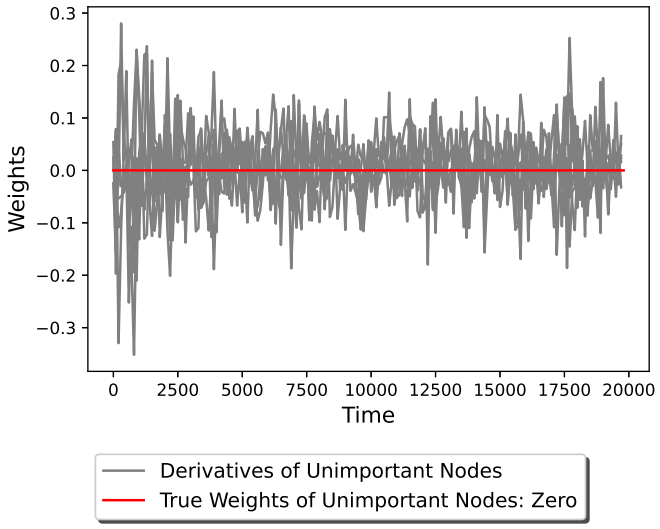

Fig. 3. The estimated derivatives vary with time for all the unimportant nodes.

6. CONCLUSION

In this paper, we proposed two Bayesian methods for estimating time-varying graphs from graph signals. The methods are based on Gaussian processes, where the first method exploits RBF kernels and the automated relevance determination principle, and the second method uses derivatives of the functions modeled by the GPs to determine the importance of the edges in the graph. We implemented the optimizations in the first method by variational inference, whereas in the second method, we worked with derivatives and sequential Monte Carlo sampling. We presented two examples that show the accuracy of the proposed methods. The results suggest that both methods are capable of tracking the changes in the graph topologies.

7. REFERENCES

[1] Vidit Nanda and Radmila Sazdanović, "Simplicial models and topological inference in biological systems," in *Discrete and Topological Models in Molecular Biology*, pp. 109–141. Springer, 2014.

[2] Dror Y Kenett, Tobias Preis, Gitit Gur-Gershgoren, and Eshel Ben-Jacob, "Dependency network and node influence: Application to the study of financial markets," *International Journal of Bifurcation and Chaos*, vol. 22, no. 07, pp. 1250181, 2012.

[3] Yanbing Mao and Emrah Akyol, "On network topology inference of social networks," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 804–809.

[4] 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.

[5] Hilmi E Egilmez, Eduardo Pavez, and Antonio Ortega, "Graph learning from data under Laplacian and structural constraints,"

IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 6, pp. 825–841, 2017.

[6] Anil Seth, "Granger causality," *Scholarpedia*, vol. 2, no. 7, pp. 1667, 2007.

[7] Helmut Lütkepohl, "Vector autoregressive models," in *Handbook of Research Methods and Applications in Empirical Macroeconomics*, pp. 139–164. Edward Elgar Publishing, 2013.

[8] William T Bielby and Robert M Hauser, "Structural equation models," *Annual Review of Sociology*, pp. 137–161, 1977.

[9] 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.

[10] Brian Baingana and Georgios B Giannakis, "Tracking switched dynamic network topologies from information cascades," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 985–997, 2016.

[11] Brian Baingana, Gonzalo Mateos, and Georgios B Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 563–575, 2014.

[12] Emily Fox, Erik Sudderth, Michael Jordan, and Alan Willsky, "Nonparametric Bayesian learning of switching linear dynamical systems," *Advances in Neural Information Processing Systems*, vol. 21, 2008.

[13] Chen Cui, Paolo Banelli, and Petar M Djurić, "Gaussian processes for topology inference of directed graphs," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 2156–2160.

[14] Petar Djuric and Cédric Richard, *Cooperative and Graph Signal Processing: Principles and Applications*, Academic Press, 2018.

[15] Christopher KI Williams and Carl Edward Rasmussen, *Gaussian Processes for Machine Learning*, vol. 2, MIT press Cambridge, MA, 2006.

[16] Yuhao Liu, Marzieh Ajirak, and Petar M Djurić, "Inference with deep Gaussian process state space models," in *2022 30th European Signal Processing Conference (EUSIPCO)*. IEEE, 2022, pp. 792–796.

[17] Kurt Butler, Guanchao Feng, and Petar M Djurić, "A differential measure of the strength of causation," *IEEE Signal Processing Letters*, 2022.