LEARNING SIGNALS AND GRAPHS FROM TIME-SERIES GRAPH DATA WITH FEW CAUSES

Panagiotis Misiakos, Vedran Mihal, Markus Püschel

Department of Computer Science
ETH Zurich, Switzerland

ABSTRACT

In this paper we port assumptions and techniques from DAG (directed acyclic graph) learning and causal inference to time-series graph data. In particular, we view such data as indexed by a DAG obtained by unrolling the graph in time and generated by a causal linear structural equation model (SEM) from only few causes. For this situation we solve two problems: (1) learning the time series from samples, and (2) learning the graph from time-series data by first learning the entire DAG and then extracting the result. We empirically evaluate our approach targeting the few-causes assumption on both synthetic and real-world data and show significant improvements over prior methods.

Index Terms— Graph signal processing, DAG learning, time-varying graph signals, sampling, Fourier-sparsity

1. INTRODUCTION

Graph signal processing (GSP) has evolved into a framework [1, 2] and set of tools to analyze signals, or data associated with the nodes of an undirected [3] or directed graph [4]. Among the many applications are the problem of learning graph signals from samples [5, 6] and learning the graph from data [7]. Most work here has focused on the undirected case. For example, [8, 9] imposes a smoothness criterion and [10, 11, 12] assumes Fourier-sparsity based on the Laplacian decomposition. The subject of this paper are time series of graph data.

Time-series graph data. Time series on graphs are graph data collected over time, so each node is associated with a time series, or, equivalently, each time step is associated with a graph signal. The joint time-vertex signal processing framework [13] uses Cartesian product decomposition to extend GSP to such data, assuming undirected graphs and [14] generalizes it to directed graphs. Sampling and reconstruction of such signals was considered in [15], and learning the graph from time series in [16].

In this paper we are concerned with both problems, learning the signal from samples and learning the graph from time-series data, however both method and assumptions are different. Namely, we build on the idea of unrolling the graph in time to obtain a directed acyclic graph (DAG) [17]. Then we apply recent methods from causal inference and in particular DAG learning to these problems. In particular, we assume that the data have few causes in the sense of linear structural equation models (SEMs) as explained next.

DAG learning from linear SEMs. DAGs are a common model for data [18] with causal relationships captured by the edges and thus learning the DAG from data is relevant for causal inference, even though proving causality requires additional techniques such as interventions [18]. One common assumption is that the data follow a linear SEM [18, 19, 20] and state-of-the-art DAG learning methods use continuous optimization constraints to enforce acyclicity [21, 22]. Among the continuous optimization techniques some specialize on learning DAGs from time-series data [23, 24].

The aforementioned methods for DAG learning assume that the data are generated from a linear SEM or the analogous linear model for time series, called structural vector autoregression (SVAR) [25]. A linear SEM can be viewed as a linear transform that takes as input a DAG signal of causes to compute the observed signal as output, and has been interpreted as a form of inverse Fourier transform [26]. The common assumption of the input in the above references has been i.i.d. noise since its generality can express any Gaussian distribution as output [27]. In [28] we motivated the situation of an approximately sparse input (i.e., Fourier-sparsity in the sense of [26]), and proposed an associated DAG learning method, to capture the situation that only few nodes are responsible for the observed signal.

Contributions. We port learning techniques from causal inference with DAGs to time-series graph data. In particular we model the data generation of time series on a graph as a linear SEM on the associated time-unrolled DAG. Then we adopt the assumption of few causes with unknown locations in the linear SEM from [28] and solve two problems: (1) learning the entire time series from samples, and (2) learning the graph from the time series by first learning the entire time-unrolled DAG and then extracting the graph from one time step. This way the temporal relations in the time series are exploited. We show superior performance in synthetic experiments compared to prior work that does not exploit the assumption of few causes. Then we show a prototypical real-world experiment that illustrates how our assumptions can be useful in practice.

2. BACKGROUND

Graphs and DAGs. With $\mathcal{G} = (V, \mathbf{B})$ we denote a directed graph (digraph) with vertex set $V = \{1, 2, \ldots, d\}$ and weighted adjacency matrix $\mathbf{B} \in \mathbb{R}^{d\times d}$, where b_{ij} represents the weight of the edge from node j to i (j is then a parent of i) and $b_{ij} = 0$ if there is no such edge. $\mathcal{G}$ may have self-loops. Where needed we will represent undirected graphs by a symmetric $\mathbf{B}$. Especially relevant in this paper are directed acyclic graphs (DAGs) $\mathcal{G} = (V, \mathbf{A})$ that do not have (directed) cycles. In this case we assume V to be topologically sorted, i.e., there is no edge from i to j when $j < i$. This way, $\mathbf{A}$ is lower triangular with zeros on the diagonal and thus $\mathbf{A}^d = \mathbf{0}$.

Graph signals $\mathbf{x} = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$ associate a real value x_i with each vertex i.

Linear SEMs. DAGs are commonly used to capture causal relationships [18] and associated data (or signals) are often modeled with structural equation models (SEMs) [18, 19, 20]. The simplest, but often used are linear SEMs, which assume that the value $\mathbf{x}_i$ at node i is a linear combination of the values at the parent nodes plus an additional contribution c_i from node i. Formally, this means

$$\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{c}, \quad \mathbf{c} = (c_1, \ldots, c_d)^T. \tag{1}$$

Most prior work assumes $\mathbf{c}$ to be i.i.d. noise [19, 21, 22]. If $\mathbf{c}$ is in addition Gaussian, then (1) can model any d-variate Gaussian distribution for $\mathbf{x}$ [27] with a suitable $\mathbf{A}$. But other choices are possible as we assert below.

Equation (1) can be viewed as a recurrence that computes $\mathbf{x}$ starting from the values at the sources (no parents). An equivalent form rewrites (1) (using $\mathbf{A}^d = \mathbf{0}$ [28]) as

$$\begin{aligned}\mathbf{x} &= (\mathbf{I} - \mathbf{A})^{-1}\mathbf{c} = \left(\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + ... + \mathbf{A}^{(d-1)}\right)\mathbf{c} \\ &= \mathbf{W}\mathbf{c}. \end{aligned} \tag{2}$$

Thus a linear SEM is a linear transform that computes the signal $\mathbf{x}$ from the inputs $\mathbf{c}$ at each node. The transform matrix $\mathbf{W}$ captures the influence of a node with contribution c_i to all its descendants (not only the direct children). We call $\mathbf{c}$ the causes of $\mathbf{x}$ in the linear SEM. Interestingly, $\mathbf{c}$ can be viewed as a form of spectrum of the signal $\mathbf{x}$ with a suitable associated notion of shift and convolution [26].

Few causes. In this paper we consider DAG signals $\mathbf{x}$ with *few causes* following [28][1]. This means only a few nodes produce significant input values c_i that percolate through the network following (1) or (2) to produce $\mathbf{x}$. Examples include river pollution data for which only few polluting sites are responsible or gene expression data caused by only few activated genes in the network. In practice, the sparsity assumption will hold only approximately, so we allow for low magnitude noise $\mathbf{n}_c$ in the causes $\mathbf{c}$. Further, the measurement of $\mathbf{x}$ will also be subject to noise $\mathbf{n}_x$, so our model of a DAG signal will be

$$\mathbf{x} = \mathbf{W}(\mathbf{c} + \mathbf{n}_c) + \mathbf{n}_x. \tag{3}$$

[1]Where they are called root causes.

In prior work, we have shown that (1) under mild assumptions, DAGs can be learned from associated signals with few causes [28, 29], and (2) DAG signals with few causes at unknown locations can be learned from samples [26, 30].

3. TIME-SERIES GRAPH DATA AS A DAG

Our goal is to apply the prior models and work on linear SEMs and associated methods of causal inference on DAGs to time-series graph data. To do so we first follow [17] by unrolling in time a graph to obtain a DAG and then we instantiate a linear SEM on it. This can also be viewed as a structural vector autoregression (SVAR) model [25] with time-lag $p = 1$ as the data at time $i + 1$ are obtained from data at time i. Doing so allows us to port DAG learning techniques, and in particular the assumption of few causes, to such data.

Time-series graph data as DAG. We consider a digraph (or graph) $\mathcal{G} = (V, \mathbf{B})$ with associated data vectors $\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(T)$ produced at time steps $t = 1, \ldots, T$. We create a DAG $\mathcal{G}' = (V', \mathbf{A})$ where V' consists of T replicas $V(1), \ldots, V(T)$ of V. Then, node j in $V(t)$ is connected to node i in $V(t + 1)$ as determined by b_{ij}. Self-loops become connections between the same node in subsequent time steps. Thus, $\mathcal{G}'$ becomes a DAG with adjacency matrix

$$\mathbf{A} = \mathbf{J} \otimes \mathbf{B} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & ... & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & ... & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{B} & \ddots & \vdots & \\ & \vdots & \ddots & \mathbf{0} & \vdots \\ \mathbf{0} & \mathbf{0} & ... & \mathbf{B} & \mathbf{0} \end{pmatrix}, \tag{4}$$

where $\mathbf{J}$ is the matrix with ones right below the main diagonal and $\otimes$ is the Kronecker product of two matrices: $\mathbf{G} \otimes \mathbf{H} = [g_{i,j}\mathbf{H}]_{i,j}$.

Linear SEMs. Stacking the $\mathbf{x}(i)$ into a vector $\mathbf{x} \in \mathbb{R}^{dT}$, a linear SEM in the recursive form (1) becomes

$$\begin{cases} \mathbf{x}(1) = \mathbf{c}(1), \\ \mathbf{x}(2) = \mathbf{B}\mathbf{x}(1) + \mathbf{c}(2), \\ \qquad \vdots \\ \mathbf{x}(T) = \mathbf{B}\mathbf{x}(T-1) + \mathbf{c}(T). \end{cases} \tag{5}$$

In words, after initializing $\mathbf{x}(1)$, each subsequent $\mathbf{x}(i + 1)$ is obtained from the previous $\mathbf{x}(i)$ as determined by $\mathbf{B}$ and the input $\mathbf{c}(i)$ from the nodes at each time step.

To obtain the closed form in (2) as a transform of causes we need the following lemma.

Lemma 3.1. *The matrix* $\mathbf{W}$ *in* (2) *and* (3) *corresponding to*

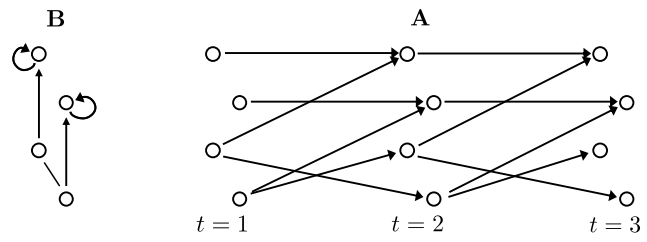

Fig. 1: Unrolling a digraph $\mathbf{B}$ in three time steps into DAG $\mathbf{A}$.

the unrolled DAG $\mathbf{A}$ *of the digraph* $\mathbf{B}$ *is given by*

$$\mathbf{W} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + ... + \mathbf{A}^{(T-1)} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & ... & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{I} & \mathbf{0} & ... & \mathbf{0} & \mathbf{0} \\ \mathbf{B}^2 & \mathbf{B} & \mathbf{I} & \ddots & \vdots & \\ \mathbf{B}^3 & \mathbf{B}^2 & \mathbf{B} & \ddots & \mathbf{0} & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{I} & \mathbf{0} \\ \mathbf{B}^{(T-1)} & \mathbf{B}^{(T-2)} & ... & \mathbf{B}^2 & \mathbf{B} & \mathbf{I} \end{pmatrix} \quad (6)$$

Proof. The power of the Kronecker product of two square matrices is the Kronecker product of their respective powers and thus

$$\mathbf{A} = \mathbf{J} \otimes \mathbf{B} \Rightarrow \mathbf{A}^k = (\mathbf{J} \otimes \mathbf{B})^k = \mathbf{J}^k \otimes \mathbf{B}^k.$$

The assertion now follows from the fact that $\mathbf{J}^k$ has ones on the kth lower diagonal and zeros elsewhere. □

Example. In Fig. 1 we illustrate the unrolling of a graph $(V, \mathbf{B})$ in three time steps into the DAG $(V', \mathbf{A})$. A linear SEM in the recursive form (1) becomes

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \mathbf{x}(3) \end{bmatrix}, \mathbf{x} = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} & \mathbf{0} \end{pmatrix} \mathbf{x} + \mathbf{c}. \quad (7)$$

Time-series graph data are common in many real-world scenarios, where $\mathbf{B}$ may be a graph or digraph and known or not. Examples include daily temperature measurements at fixed locations inside a country or pollution measurements in a river network. We are interested in the situation that the data follow a linear SEM on the unrolled DAG and the assumption of few causes, i.e., $\mathbf{c}$ being (approximately sparse). This means that the data are generated by few events at nodes that propagate through time and space as determined by $\mathbf{B}$ to produce $\mathbf{x}$.

4. EXPERIMENTS

We first perform two synthetic experiments: learning a time-series graph signal from samples and learning the graph from time-series data assuming few causes. The third experiment illustrates a possible real-world application using real data from the Thames river network [31]. We use prior notation, in particular $(V', \mathbf{A})$ is the unrolled DAG for $(V, \mathbf{B})$.

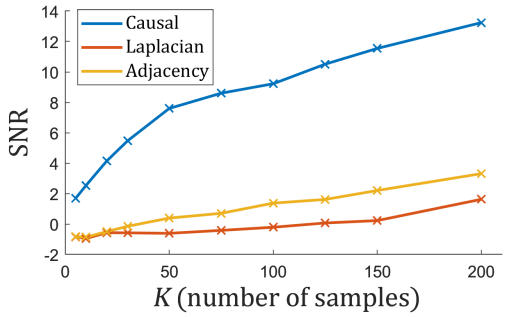

Fig. 2: Average SNR of reconstructed signals. Signals $\mathbf{x}$ are synthetic following (3) with 5% sparsity in the causes $\mathbf{c}$ on an unrolled DAG with $20 \cdot 50 = 1000$ vertices.

Synthetic: Learning signal from samples. We synthetically generate time-series graph data with few causes. First, we construct a random Erdös-Renyi graph with $d = 20$ nodes, assign random edge directions, assume self-loops so the unrolled DAG stays connected, to obtain $3d$ edges. Then we assign uniform random weights from $[-0.9, -0.1] \cup [0.1, 0.9]$ to the edges. We unroll the graph into $T = 50$ time steps to create the final DAG $\mathbf{A}$ of size $dT = 1000$. We then generate the data matrix $\mathbf{X}$ of ten signals as columns with (3) assuming an associated matrix of sparse causes $\mathbf{C}$ with uniform values in $[0, 1]$ with probability 5% and zero otherwise. So the nonzero locations are random and unknown. Both associated spectral and measurement noises $\mathbf{N}_c, \mathbf{N}_x$ are zero-mean Gaussian with standard deviation 0.01.

For each signal we choose K random samples. If $\mathbf{\Phi}_K$ is the corresponding sampling operator we recover the signal using Lasso regression:

$$\hat{\mathbf{x}} = \mathbf{W} \cdot \arg\min_{\mathbf{c}} \frac{1}{2K} \|\mathbf{\Phi}_K \mathbf{x} - \mathbf{\Phi}_K \mathbf{W} \mathbf{c}\|_2^2 + \lambda \|\mathbf{c}\|_1, \quad (8)$$

where $\mathbf{x}$ is the original signal, $\mathbf{\Phi}_K \mathbf{x}$ is the sampled signal, $\hat{\mathbf{x}}$ is the reconstructed signal, $\lambda = 10^{-4}$ is the coefficient of the sparsity regularizer, and $\mathbf{W}$ is as in (2).

For comparison, we also choose $\mathbf{W}$ as the Laplacian or adjacency Fourier basis (of the underlying undirected graph) as a baseline, which effectively assumes Fourier sparsity in these domains.

Fig. 2 shows the average signal-to-noise ratio SNR of the reconstructed signals (y-axis) for varying number of samples K (x-axis). The method successfully learns signals with few causes, and, as expected, outperforms the Laplacian and the adjacency basis.

Synthetic: Learning the graph from data. We learn the graph $\mathbf{B}$ from time-series data with few causes by applying DAG learning on the unrolled $\mathbf{A}$ to then extract one block $\mathbf{B}$.

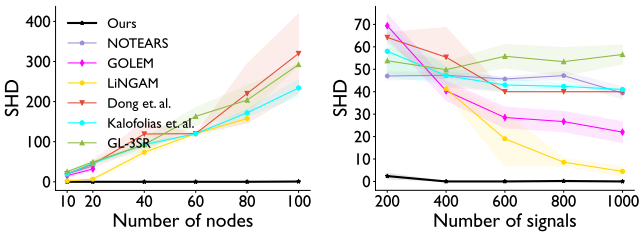

Fig. 3: SHD metric (lower is better) of the reconstructed graph $\mathbf{B}$. Left: varying the number d of nodes for $T = 10$ time steps and $n = 1000$ signals. Right: varying the number n of data vectors for $d = 20$ nodes and $T = 10$ time steps.

We build on the method in [28] but modify it to enforce the special structure of $\mathbf{A}$ in (4). Namely, given the data matrix $\mathbf{X} \in \mathbb{R}^{dT \times n}$ with n signals as columns, we use gradient descent to solve the following optimization problem:

$$\hat{\mathbf{A}} = \underset{\mathbf{A} \in \mathbb{R}^{dT \times dT}}{\arg\min} \|\mathbf{X} - \mathbf{X}\mathbf{A}\|_1 + \lambda \|\mathbf{A} - \mathbf{P}\mathbf{A}\mathbf{Q}\|_2 \quad \text{s.t. } h(\mathbf{A}) = 0. \tag{9}$$

The term $\|\mathbf{X} - \mathbf{X}\mathbf{A}\|_1$ promotes sparsity in A and the regularizer $\|\mathbf{A} - \mathbf{P}\mathbf{A}\mathbf{Q}\|_2$, with coefficient $\lambda = 10^{-3}$, promotes the blocks in the lower block-diagonal to be equal, as in (4). This is done by choosing $\mathbf{P}$ and $\mathbf{Q}$ to perform a cyclic shift on these blocks.[2] The constraint $h(\mathbf{A}) = \operatorname{trace}\left(e^{\mathbf{A} \odot \mathbf{A}}\right) - dT$ ($\odot$ is the entry-wise matrix product) is the continuous acyclicity constraint proposed by [21]. After computing $\hat{\mathbf{A}}$ the approximated undirected graph adjacency matrix $\hat{\mathbf{B}}$ is extracted as the first (top left) block $\mathbf{B}$ in $\hat{\mathbf{A}}$. This choice worked empirically best in our experiments, which makes sense since it affects the entire subsequent DAG data.

We compare against two types of prior work. First, the methods [8], [9] to learn the graph $\mathbf{B}$ directly from the data, and GL-3SR [10, 11]. Second, other methods that, like ours, learn the DAG $\mathbf{A}$, and then we extract $\mathbf{B}$: NOTEARS [21], GOLEM [22] and LiNGAM [19], which are among the state-of-the-art methods for learning DAGs from linear SEM data.

We generate graphs as before but keep them undirected (to apply the above prior graph learning methods), choosing $d = 20$ nodes, average $3d$ edges, $n = 1000$ data vectors and $T = 10$ time steps by default. The metric for comparing is the average structural Hamming distance (SHD) over five repetitions, i.e., the number of edge insertions and deletions needed to convert $\hat{\mathbf{B}}$ to $\mathbf{B}$ (lower is better).

Fig. 3 shows the results for varying number d of nodes and varying number of signals n. We conclude that our method performs significantly better than the baselines which are oblivious to the property of few causes.

[2]More precisely: $\mathbf{P} = (\mathbf{0}_1 \oplus \mathbf{C}_{T-1}) \otimes \mathbf{I}_d$, $\mathbf{Q} = (\mathbf{C}_{T-1}^T \oplus \mathbf{0}_1) \otimes \mathbf{I}_d$. $\mathbf{C}$ is the cyclic shift matrix and $\oplus$ the block-diagonal composition.

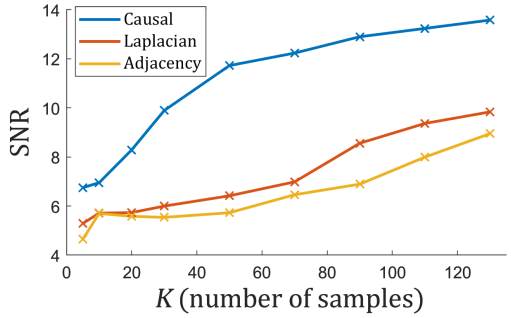

Fig. 4: Average SNR of reconstructed Thames signals. The network has 13 sites unrolled into 49 time steps (weeks) to obtain a DAG with $13 \cdot 49 = 637$ vertices.

4.1. Real-world experiment: Thames river network

We perform an experiment on data from the Thames river network [31] to illustrate how our method could be applied in a real-world problem. The Thames river network consists of thirteen different sites along the river connected by directed edges through the water flow. For seven years, roughly weekly (49/year) measurements are available. We group these into seven signals with $d = 13$, each unrolled with $T = 49$. The signals measure the dissolved reactive silicon concentration in [mg/L]. The graph $\mathcal{G}$ is known, but not the edge weights in $\mathbf{B}$.

Seven is too few to learn $\mathbf{B}$ from the data. However, we can learn the edge weights, and then use these weights to learn the signals from samples, thus testing whether the assumption of few causes may hold.

We learn the weights by modifying (9) to restrict $\mathbf{A}$ to only contain weights in the entries corresponding to the true edges. We do this by masking the weighted variable $\mathbf{A}$ with a mask containing the edge locations. Once $\hat{\mathbf{A}}$ is obtained, we learn the signals from random samples as before in (8). Fig. 4 shows the average (over seven signals) SNR of the reconstructed signals (y-axis) for varying number of samples K (x-axis).

Our method performs significantly better, especially for small K, which implies that the data can be viewed as having few causes in a linear SEM model.

5. CONCLUSION

We presented a novel way of analyzing time-series data on graphs. The basic idea was to unroll the graph in time and apply DAG learning techniques to the obtained DAG, whose acyclicity is guaranteed due to the nature of time. In particular, we focused on the case that the data are generated by few causes, i.e., few inputs from nodes, in the sense of a linear SEM model. In this setting we showed both how to learn signals from samples and the graph from data, and gave a prototypical application example with real-world data.

6. REFERENCES

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

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

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

[4] 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, 2020.

[5] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510–6523, 2015.

[6] Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung, "Sampling signals on graphs: From theory to applications," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 14–30, 2020.

[7] F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, and H. Liu, "Graph learning: A survey," *IEEE Trans. on Artificial Intelligence*, vol. 2, no. 2, pp. 109–127, 2021.

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

[9] V. Kalofolias, "How to learn a graph from smooth signals," in *Artificial Intelligence and Statistics*, pp. 920–929, Proceedings of Machine Learning Research, 2016.

[10] B. Le Bars, P. Humbert, L. Oudre, and A. Kalogeratos, "Learning Laplacian matrix from bandlimited graph signals," in *Proc. Int. Conf. Acoust., Speech, and Signal Process. (ICASSP)*, pp. 2937–2941, IEEE, 2019.

[11] P. Humbert, B. Le Bars, L. Oudre, A. Kalogeratos, and N. Vayatis, "Learning Laplacian matrix from graph signals with sparse spectral representation," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 8766–8812, 2021.

[12] 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, 2019.

[13] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 817–829, 2018.

[14] C. Cheng, Y. Chen, Y. J. Lee, and Q. Sun, "SVD-based graph Fourier transforms on directed product graphs," *IEEE Trans. Signal and Information Process. over Networks*, vol. 9, pp. 531–541, 2023.

[15] Z. Xiao, H. Fang, S. Tomasin, G. Mateos, and X. Wang, "Joint sampling and reconstruction of time-varying signals over directed graphs," *IEEE Trans. Signal Process.*, vol. 71, pp. 2204–2219, 2023.

[16] A. Einizade and S. H. Sardouie, "Learning product graphs from spectral templates," *IEEE Trans. Signal and Information Process. over Networks*, vol. 9, pp. 357–372, 2023.

[17] H. Kim and R. Anderson, "Temporal node centrality in complex networks," *Physical Review E*, vol. 85, no. 2, p. 026107, 2012.

[18] J. Peters, D. Janzing, and B. Schölkopf, *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[19] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen, "A linear non-Gaussian acyclic model for causal discovery," *Journal of Machine Learning Research*, vol. 7, no. 72, pp. 2003–2030, 2006.

[20] A. Ghoshal and J. Honorio, "Learning linear structural equation models in polynomial time and sample complexity," in *Int. Conf. Artificial Intelligence and Statistics*, pp. 1466–1475, Proceedings of Machine Learning Research, 2018.

[21] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, "DAGs with NO TEARS: Continuous optimization for structure learning," *Advances in Neural Information Processing Systems*, vol. 31, pp. 9472—9483, 2018.

[22] I. Ng, A. Ghassami, and K. Zhang, "On the role of sparsity and DAG constraints for learning linear dags," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17943–17954, 2020.

[23] R. Pamfil, N. Sriwattanaworachai, S. Desai, P. Pilgerstorfer, K. Georgatzis, P. Beaumont, and B. Aragam, "Dynotears: Structure learning from time-series data," in *Int. Conf. Artificial Intelligence and Statistics*, pp. 1595–1605, Proceedings of Machine Learning Research, 2020.

[24] S. Fan, S. Zhang, X. Wang, and C. Shi, "Directed acyclic graph structure learning from dynamic graphs," *Proc. AAAI Conf. on Artificial Intelligence*, vol. 37, no. 6, pp. 7512–7521, 2023.

[25] L. Kilian, "Structural vector autoregressions," *Centre for Economic Policy Research Discussion Paper No. DP8515*, 2011.

[26] B. Seifert, C. Wendler, and M. Püschel, "Causal Fourier analysis on directed acyclic graphs and posets," *IEEE Trans. Signal Process.*, vol. 71, pp. 3805–3820, 2023.

[27] B. Aragam and Q. Zhou, "Concave penalized estimation of sparse Gaussian Bayesian networks," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2273–2328, 2015.

[28] P. Misiakos, C. Wendler, and M. Püschel, "Learning DAGs from data with few root causes," *Advances in Neural Information Processing Systems*, vol. 36, 2023. To appear.

[29] M. Chevalley *et al.*, "The CausalBench challenge: A machine learning contest for gene network inference from single-cell perturbation data," *arXiv preprint arXiv:2308.15395*, 2023.

[30] B. Seifert, C. Wendler, and M. Püschel, "Learning Fourier-sparse functions on DAGs," in *Int. Conf. Learning Representations. Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022.

[31] M. J. Bowes, L. K. Armstrong, S. A. Harman, D. J. E. Nicholls, H. D. Wickham, P. M. Scarlett, and M. D. Juergens, "Weekly water quality data from the river Thames and its major tributaries (2009-2017)," 2020. `https://doi.org/10.5285/cf10ea9a-a249-4074-ac0c-e0c3079e5e45`.