Neural Causal Discovery and Graph Signal Processing

Shiuli Subhra Ghosh
Rensselaer Polytechnic Institute

***Abstract*—Graphical models are widely used in data science for capturing useful relations among variables. In this report, we review the theory and applications of graphical models from a signal-processing, and causality perspective. This review is based on two papers, one representing the literature review on Graph Signal Processing (GSP) [1], and the other discussing the neural network approach for causal discovery using a recently proposed continuous constrained optimization technique [2]. First, we give an overview of the topics covered in both papers. Then, focusing on causal discovery, we discuss some recent challenges in the domain with some possible future research directions.**

***Index Terms*—Graphical Models, Causal Discovery, Graph Signal Processing**

I. Introduction

Data is ubiquitous. An enormous amount of data is collected every day at every moment. But the main challenge is to extract useful information from the huge amount of data. In this report, we will focus on data that resides on a network. This network can be of two types, physical network, and logical network (see Table I).

Physical Network	Logical Network
Internet	Social networks
Transportation	The web
Power systems	Complex systems

TABLE I: Example of Different Types of Network

If the underlying network structure is known to us, it helps us to incorporate additional information on the variable interactions for achieving some specific tasks like regression, classification, etc. But if it is unknown, it becomes necessary to learn the network structure for better performance.

A graph is a suitable structure for modelling the system network and complex interactions among the components. Formally, a *graph* ($\mathcal{G} = (\mathbf{V}, \mathcal{E})$) can be defined as a collection of nodes ($\mathbf{V}$) and a set of edges ($\mathcal{E}$) that connect the nodes. A graph is *directed* if all the edges are directed. A *directed path* in a graph is a finite or infinite sequence of directed edges. A *directed cycle* is a directed path whose first and last vertices are the same. If there is a directed path from node x to node y, then we call x an *ancestor* of y, and y is a *descendent* of x. Also, if there are no directed cycles in a directed graph, it is known as a *directed acyclic graph (DAG)*.

In Section II, we discuss the theory and applications of Graph Signal Processing (GSP). In Section III, we focus our discussion on causality for graphical models. In Section IV, we explore the theory of identifiability of causal models from observation data. In Section V, we analyze the continuous constrained optimization method for acyclic causal discovery, and finally, in Section VI and Section VII, we conclude our discussion by mentioning some recent challenges in the domain along with some possible future scopes for research.

II. Signal Processing Approach for graphical Models

Graph signal processing (GSP) is an extended version of the Digital Signal Processing (DSP) approach for modelling graph signals. Given a graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, *graph signal* is defined as a set of values (X) residing on its set of nodes ($\mathbf{V}$). It can also be looked at from the spectral perspective, given, the underlying graph is represented as a matrix. There can be different matrix representations of a graph. Usually, GSP considers frequency representation based on the adjacency matrix, and/or the laplacian matrix, where graph frequencies are defined as the eigenvalues of the matrix representation. Now, we will provide some basic definitions related to GSP.

A. Algebraic representations of graphs

The *adjacency matrix* ($\mathbf{A}$) represents a graph, such that $(\mathbf{A})_{ij}$ is either 0 or 1 based on if there is an edge from node i to node j. When the graph is undirected, the adjacency matrix is symmetric. We also define the *degree matrix* ($\mathbf{D}$) in terms of a diagonal matrix, where

$$(\mathbf{D})_{ii} = \sum_{j=1}^{N} (A)_{ij} \quad \text{and} \quad (\mathbf{D})_{ij} = 0, \forall i \neq j. \tag{1}$$

The *combinatorial graph laplacian* ($\mathbf{L}$) is defined as

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \ . \tag{2}$$

B. Graph Shift Operator

Given a graph $\mathcal{G} = (\mathbf{V}, \mathcal{E})$, the *graph shift* operator can be written as $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, where $(\mathbf{S})_{ij} = 0$ for $i \neq j$ and $(i, j) \notin \mathcal{E}$. For example, the adjacency matrix ($\mathbf{A}$) or the Laplacian matrix ($\mathbf{L}$) can be adopted as the shift for a graph.

C. Graph Filters

Graph filters ($\mathbf{H}$) are the matrix polynomials of the graph shift operator ($\mathbf{S}$).

$$\mathbf{H} := \sum_{\ell=0}^{N-1} h_\ell \mathbf{S}^\ell = \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^{-1} \ , \tag{3}$$

where, $\mathbf{V}$ is the eigenvectors of $\mathbf{H}$. Note that, this is the same as the eigenvectors of the graph shift ($\mathbf{S}$). The difference is in the eigenvalues $\tilde{\mathbf{h}}$. The Laplacian plays a dual role as both graph filter and graph shift. First, it captures the network topology, which is very important, and on the other hand, it is the simplest non-trivial transformation we can apply to the graph signal.

D. Graph Frequencies and Graph Fourier Transform

As graph filters are the polynomial of the graph shift ($\mathbf{S}$), the frequencies are defined by the function of eigenvalues of the shift (3). The eigenvectors corresponding to each eigenvalue are called the *Fourier basis*. If we do the eigendecomposition of the graph filters, the set of orthogonal eigenvectors is called the *Graph Fourier Transform (GFT) matrix*. As an example for graph Laplacian, consider

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T, \tag{4}$$

where, $\mathbf{U}$ is the GFT matrix and $\mathbf{\Lambda}$ is the diagonal matrix with eigenvalues in the diagonal entries. Corresponding to the smallest eigenvalue, which is 0 for the symmetric graph Laplacian, the eigenvector is constant $[\mathbf{1}, \dots \mathbf{1}]^T$. This can be thought of as the DC component for Fourier Transform. As the frequency increase, we can observe more changes in the sign for the corresponding eigenvector. In other words, more frequency induces more oscillation. This is how the intuition behind graph frequency is developed.

E. Application of GSP

Graph signal processing can have multiple applications in engineering domains, where, leveraging the knowledge of network structures can extract useful insights.

1) In **network science**, the graph structure is more important than the data itself. On the other hand, GSP is built upon the graph spectra that can be perceived as a function of the structure of the graph. For example, GSP can answer some questions related to the spectral clustering of the network. Also, the smooth graph signal model helps to detect outliers by using high-pass filtering and thresholding in the graph [3].
2) Given a network, it is also important to model **propagation over networks**. As it is very difficult to model, the network is abstracted using the assumption that all the components in a network affect the other components. But with the increase in network size, the network state space grows exponentially. To study these processes, time and large network asymptotic behaviour is considered [4]. In GSP, these asymptotic behaviours can be seen to depend on the eigenstructure of the underlying graph.
3) **Graph sampling** is another application of GSP that is similar to sampling in DSP. The main idea is to sample certain nodes from the graph, which can be processed to retrieve the information for other nodes. If the graph is smooth, i.e. frequency is low enough, we can sample and reconstruct. Otherwise, if the signal is arbitrary, we cannot reconstruct it. Graph sampling can be used for semi-supervised machine learning applications [5].
4) Another application of GSP is **learning graphs** from large datasets. Usually, graph structure captures the statistical dependence and conditional independence in a dataset. There are some recent works on learning graph topology using GSP results. With an assumption of smoothness in the graph signals, it is possible to formulate the graph discovery problem as an optimization problem that essentially learns the shift (e.g. Laplacian) from data [6].

In the next section, we will discuss the graphical models from the notion of causality.

III. Notion of Causality in Graphical Models

Before defining the notion of causality in Graphical Models, let us have a small introduction to *probabilistic graphical models (PGM)*, specifically, causal models are a special class of PGMs. PGMs are usually modelled by Bayesian Networks.

Definition 1: (Bayesian Network) A Bayesian network (BN) is a DAG that models the dependencies among a set of random variables.

Given a set of random variables $\mathbf{X} = (X_1, \dots, X_n)$ with index set $\mathbf{V} = \{1, \dots, n\}$, we denote their joint distribution by $P(X_1, X_2, \dots, X_n)$. Using the BN chain rule of probability, we can factorize the joint distribution as,

$$P(X_1, X_2, \dots, X_n) = P(X_1) \cdot \prod_i P(X_i | X_{i-1}, \dots X_1). \tag{5}$$

Now, given a probability distribution and a corresponding DAG, we can make *Local Markov Assumption* to formalize the specification of independencies in the graphical model.

Assumption 1: (Local Markov Assumption) In a DAG, a node X is independent of all its non-descendants given its parents.

This assumption helps us to factorize the (5) as,

$$P(X_1, X_2, \dots, X_n) = P(X_1) \cdot \prod_i P(X_i | \text{Pa}(X_i)). \tag{6}$$

where, X_1 is the root node and $\text{Pa}(X_i)$ is the parents of X_i in the graph.

Now, as we have defined the statistical notion of independence in terms of local Markov property, we would like to augment these models with some causal assumptions.

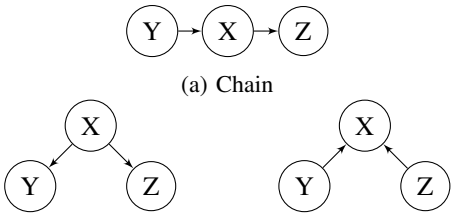
Definition 2: (Cause) A variable X is said to be the cause of a variable Y if Y can change in response to changes in X.

For transforming a directed graph to a causal graph we need the following assumption.

Assumption 2: (Strict Causal Edge Assumption) In a directed graph, every parent is a direct cause of all its children.

This leads us to define the basic building blocks of a graph, i.e. chain, fork, and collider (see Fig 1).

According to the local Markov assumption, we can encode the independencies from the graph structures. Fig 1b and Fig 1a encodes $Y \perp\!\!\!\perp Z \mid X$. In Fig 1b, as X is the common cause of Y and Z, we call X a confounder. On the other hand, Fig 1c encodes $Y \perp\!\!\!\perp Z$ (not conditioning on X, Y is independent of Z). Hence, X is called the collider or the common effect. Graphical models are based on these building blocks, from which we can define a notion of "flow of association", which means whether any two nodes in a graph are associated or not.



(a) Chain

(b) Fork (c) Collider

Fig. 1: Basic graph building blocks

So, we define a blocked path between nodes X and Y by conditioning set Z if, along the path, there is a chain of the form $\cdots \rightarrow W \rightarrow \ldots$ or a fork of the form $\cdots \leftarrow W \rightarrow \ldots$, where $W \in Z$ is conditioned on. Otherwise, there is a collider W or any descendants of the collider W are not conditioned on. The notion of a blocked path helps us to define *d-separation* and the corresponding global Markov assumption.

Definition 3: (d-separation) Two (sets of) nodes X and Y are d-separated by a set of nodes Z if all of the paths between (any node in) X and (any node in) Y are blocked by Z.

Assumption 3: (Global Markov assumption) Given that P is Markov with respect to G (satisfies the local Markov assumption 1), if X and Y are d-separated in G conditioned on Z, then X and Y are independent in P conditioned on Z.

$$X \perp\!\!\!\perp_G Y \mid Z \implies X \perp\!\!\!\perp_P Y \mid Z \tag{7}$$

The notion of association in the causal graph is the same as in the Bayesian network. The difference lies in the Assumption 2, which gives the edges a causal meaning. This assumption introduces the notion of causality where X is a cause of Y, which does not imply the converse is true (asymmetric relation). But the statistical notion of association is a symmetric relation. Likewise, the probabilistic graphical models can be extended to the causal models using the acyclic Bayesian network, which has a causal meaning associated with the edges.

IV. Learning Causal Models from Data

Previously while defining the notion of causality in graphical models, we assumed that we know the graph structure beforehand. The main assumption we have seen is the global Markov assumption which essentially encodes the independencies in the graph structure consistent with the probability distribution. Now, we want to solve the inverse problem i.e., learning the graph structure based on observational distribution.

A. Constraint Based Causal Discovery

One possibility is that, if we can read the independencies from the empirical distribution, we can use the information to form a causal graph. But the global Markov assumption does not help us to solve this inverse problem. For this, we need a different assumption called *faithfulness*.

Assumption 4: (faithfulness) A graph G is said to be faithful according to a distribution P if,

$$X \perp\!\!\!\perp_P Y \mid Z \implies X \perp\!\!\!\perp_G Y \mid Z. \tag{8}$$

Also, some methods assume another assumption of causal sufficiency where no unobserved confounders are present in the distribution. In other words, all the common causes are observed.

With all these causal faithfulness and causal sufficiency assumptions, we can expect to retrieve the actual causal graph. But unfortunately, these assumptions only help us to learn the causal graph up to a Markov equivalence class [7]. For example, when discussing chains and forks (Fig 1), we mentioned that they share the same conditional independencies. If we reverse the direction of the arrows in the chain graph, it will also encode the same conditional independencies. So the presence or absence of conditional independencies in the data is not enough to distinguish the chains and forks. But using these, we can learn the skeleton and the immoralities in the actual graph structure, which we call *CPDAG (Complete Partially Directed Acyclic Graph)*. One example of such an algorithm for causal discovery is the PC algorithm [8].

Another possibility is to learn the structural equation models (SEM) instead of learning independencies from data. We can encode the causal interaction by introducing some structural equations $Y := f_y(X)$, where ":=" denotes one directional causal relationship. We incorporate the probabilistic notion in the SEM by adding a noise term to the effect, which can be seen as the unknown causes of variable Y. So, Y can be written as a function of X and the unobserved noise variables N_Y.

$$Y := f_y(X, N_Y), X \perp\!\!\!\perp N_Y \tag{9}$$

In this case, the functional class is too large, and we do not have structural identifiability [9].

B. Restricted Models for Causal Discovery

1) Linear Models: Instead of considering a large functional class, if we restrict our model to a linear non-Gaussian noise setting, it turns out that we can identify the causal graph from data [10], [11].

Assumption 5: (Linear Non-Gaussian Noise) All structural equations are of the following form

$$Y := f(X) + U, \quad U \perp\!\!\!\perp X. \tag{10}$$

where f is a linear function, $X \perp\!\!\!\perp U$, and U is distributed as a non-Gaussian exogenous random variable.

2) Non-Linear Models: We can also make the causal models to be identifiable if we consider the restricted functional class in a non-linear additive noise setting [9]. This is a less restrictive class of Structural Equation Models (SEM).

Assumption 6: (Nonlinear Additive Noise) All causal mechanisms are nonlinear, where the noise enters additively.

$$X_i := f(\text{Pa}(X_i)) + U_i, \quad U_i \perp\!\!\!\perp \text{Pa}(X_i), \tag{11}$$

where f is a nonlinear function.

C. Score-Based Causal Discovery

Score-based methods are also used for causal discovery. Given the independent and identically distributed (i.i.d) samples ($\mathcal{D}$) from data, the idea is to design a score function $\mathcal{S}(\mathcal{D}, \mathcal{G})$ for each graph $\mathcal{G}$ and search over the space of DAGs to select the best one which maximizes the score, i.e.,

$$\hat{\mathcal{G}} = \arg\max_{\mathcal{G}} \mathcal{S}(\mathcal{D}, \mathcal{G}). \tag{12}$$

. There are many suggestions regarding designing the score function, such as using prior knowledge of the graph and looking into the problem from the MAP estimator perspective [12]. Another famous approach is to use the parametric model and define a score function based on the maximum likelihood of the parameter (BIC score) [13].

D. Scalability Problem for the existing methods

All the methods mentioned above have a common problem of scalability since the search space of all DAGs grows super exponentially with the number of nodes. Some greedy approaches exist in the literature, but they can get stuck in the local minima. In the next section, we will discuss how the combinatorial optimization problem is converted to a continuous optimization problem to offer better scalability in the causal discovery domain.

V. Continuous Optimization for Causal Discovery

Continuous optimization for causal discovery was first proposed by Zheng et. al. [14] in 2018. They proposed a score-based approach for learning DAG called NOTEARS, which constructs a smooth score function that encodes the acyclicity constraint. To start with, they have considered a data matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ of n i.i.d. observations of the random vector $X = (X_1, \ldots, X_d)$. They also considered that their data distribution follows a linear structural equation model of the form,

$$X_j = w_j^T X + z_j. \tag{13}$$

where, $W = [w_1, \ldots w_n]$ is the weighted adjacency matrix of the causal graph. z_j is the random noise vector. X_j can be modelled, by the expectation of the conditional distribution $\mathbb{E}(X_j \mid X_{\text{Pa}(X_j)}) = f(w_j^T X)$ via logistic regression. The score function they considered is the simplest least square, penalized with a sparsity constraint.

This seminal work solved the scalability issue in general. But for the linear case, if we consider the Gaussian model, the causal structure is still identifiable upto the Markov equivalence class. That is why, as mentioned in Section IV-B2, the requirement of a non-linear additive noise model is important. There is another work [2] in the literature that extends the NOTEARS algorithm to more generic settings. From the following section onwards, we will discuss that algorithm for causal discovery.

A. Gradient-Based Neural DAG Learning (GraN-DAG)

The problem setting is similar to [14]. The main difference is that the SEM can take non-linear functions.

$$X_j = f_j(\text{Pa}(X_j)) + z_j \tag{14}$$

The causal structure is identifiable if f_j is a non-linear function and the noise variables z_j are mutually independent. For theoretical proof, we refer to Theorem 19 in [9].

1) Modelling Joint Density using Neural Network : As the functions are non-linear, logistic regression cannot be applied to model the conditional distribution. Hence, in [2], the authors suggested modelling the conditional distribution by fitting a multi-layer neural network for each variable X_j. X_{-j} is the vector containing X, where the j^{th} entry is masked to 0. Each neural network takes X_{-j} as an input and outputs $\theta_{(j)} \in \mathbb{R}^m$, the m dimensional parameter vector for the desired distribution family for the j^{th} neural network. Here m can be different for different conditional densities. Let, $\phi_{(j)}$ represents the parameter of the j^{th} neural network. When $\phi_{(j)}$ is unconstrained, each neural net can learn $P_j(x_j \mid x_{-j}, \phi_{(j)})$. We want the product of all the conditional densities, integrate to 1. To achieve this, $\phi_{(j)}$ can be constrained by defining a new *weighted adjacency matrix* A_ϕ, which can enforce acyclicity as described below.

2) Weighted Adjacency Matrix: To learn the causal model, we are first interested in retrieving the parents of a variable in a network. So, by using the neural network, we want to essentially get the joint distribution $P_j(x_j \mid \text{Pa}(x_j), \phi_{(j)})$. Hence, there will be some variables $x_i \notin \text{Pa}(x_j)$, and, in the neural network, the sum of all path products from that node i to the output k will be zero. We define all the path products for x_i, as $(C_{(j)})_{ki}, k \in \{1, \ldots, m\}$. The weighted adjacency matrix A_ϕ can be defined as,

$$(A_\phi)_{i,j} = \begin{cases} \sum_{k=1}^{m} (C_{(j)})_{ki}, & j \neq i \\ 0 & j = i \end{cases} \tag{15}$$

Using this, the acyclicity constraint can be written as,

$$h(\phi) = \text{Tr}(\exp(A_\phi)) - d = 0. \tag{16}$$

3) A Differentiable Score and Optimization: In [2], the authors proposed solving the MLE problem,

$$\max_{\phi} \ \mathbb{E}_{X \sim P_X} \sum_{j=1}^{d} \log P_j(x_j \mid \text{Pa}(x_j), \phi_{(j)}). \tag{17}$$

such that, $h(\phi) = 0$. This is solved by the *Augmented Lagrangian method*, which optimizes a sequence of sub-problems that are known to converge to a stationary point of the constrained problem.

$$\max_{\phi} \mathcal{L}(\phi, \lambda_t, \mu_t) = \mathbb{E}_{X \sim P_X} \sum_{j=1}^{d} \log P_j(x_j \mid \text{Pa}(x_j), \phi_{(j)}) - \lambda_t h(\phi) - \frac{\mu_t}{2} h(\phi)^2 \tag{18}$$

where, λ_t and μ_t are Lagrangian and penalty coefficients of the t^{th} sub-problem respectively. This is how GraN-DAG incorporates the continuous optimization method for solving causal discovery problems with combinatorial constraints.

4) Results: The results are compared with the existing methods in terms of Structural Hamming Distance (SHD) and Structural Interventional Distance (SID).

a) SHD: Structural Hamming distance is the measure that counts the number of false positives, missing, and reversed edges. For causal models, SHD might not always be a good measure. It can provide the distance between the actual and the learned graph but doesn't reflect the capacity for causal inference.

b) SID: The causal models are useful because when one particular node in a causal graph is forced to take a value, it can answer some questions related to the effect on its children. This is called *intervention*. The idea of distance should follow the causal interpretation of a graph that enables the prediction of the result of interventions. SID puts more importance on reversing an edge than adding an edge. Given the results in [2],

- By solving the (17) to optimality with the synthetic data, GraN-DAG can learn the true underlying causal model.
- Overall performance of GraN-DAG is the best for synthetic data. But for real data, we usually do not know about the data distribution. For example, in linear Gaussian structural equation models, the identifiability is still up to the Markov equivalence class, and if the real data follows linear Gaussian SEM, then the performance of GraN-DAG will be worse.
- For more details about the performance of GraN-DAG we refer to Section 4 in [2].

VI. Discussion

So far, we have discussed the signal processing approach and the notion of causality for graphical models. We also focused on learning causal graphs from data. In this section, we would like to cover a few aspects of graph learning from a signal processing perspective and try to analyze the differences with the causal discovery approach.

In the GSP literature, structure learning by leveraging the graph properties like smoothness and stationarity has been an active research topic for a long time [15]. A lot of effort has been made to learn undirected graphs in terms of graph laplacian from the observational data [16]. In fact, a continuous optimization technique for graph learning was proposed in 2007 [17], [18]. But they considered only the undirected Gaussian graphical models, which are not suitable for causal discovery as mentioned in Proposition 22 in [9]. The causal discovery problem is a more challenging task because causal models can be represented in terms of DAGs. To reduce the combinatorial search space for DAGs, [2], [14] absorbed the continuous constrained optimization methods, resulting in a huge improvement in the literature. Nevertheless, incorporating the causal notion for graph signal processing also motivated researchers to explore more in this domain [19].

Sometimes, the notion of causality in a physical system is inevitably required to be modelled by directed cyclic graphical models. These systems are called *dynamical systems*, which can be modelled by either assuming time series-based dependencies or by considering that the system is in equilibrium [20]. Learning causal dynamical models from time-series data is a really challenging task because capturing data with high temporal resolution is often not possible. Hence, cyclic causal discovery based on equilibrium data points gained a lot of interest [20], [21]. But the literature based on cyclic causal discovery is still sparse.

VII. Future Research Direction

Now we summarize a few possible future research directions.

1) In [20], the authors tried to formulate the cyclic causal discovery problem using MAP estimate. They didnot consider the notion of stability while optimizing over all possible functions in the search space. We can use the neural network approach mentioned in [2] for modelling the non-linear interactions and optimizing the score by incorporating stability constraints in a continuous optimization framework. This might incorporate a notion of stability for non-linear cyclic causal models.
2) The spectral representation of graphical models (e.g. the weighted adjacency matrix) can be used for better causal discovery. For equilibrium cyclic models, if we consider the stable linear systems, the eigenvalues of the adjacency matrix are strictly less than 1. As Graph Signal Processing deals with graph frequencies, spectral representation for the weighted adjacency matrix might be useful for cyclic causal systems.

VIII. Conclusion

Graphical models provide a powerful framework for understanding complex systems by representing the relationships between variables through a graphical structure, and this report reviews the theory and applications of graphical models from a signal-processing and causality perspective. The literature review on GSP highlights the importance of understanding the underlying structure of a graph to identify patterns and relationships within the data. The neural network approach for causal discovery using a continuously constrained optimization technique offers some important improvements for inferring causal relationships between variables, such as improved scalability and the ability to capture complex nonlinear relationships. Overall, this report demonstrates the value of integrating different perspectives and techniques for incorporating graphical models for data science problems and proposes some possible directions for further research in the field of graphical models and causal discovery.

REFERENCES

1 Ortega, A., Frossard, P., Kovačević, J., Moura, J. M. F., and Vandergheynst, P., "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

2 Lachapelle, S., Brouillard, P., Deleu, T., and Lacoste-Julien, S., "Gradient-based neural DAG learning," *CoRR*, vol. abs/1906.02226, 2019. [Online]. Available: http://arxiv.org/abs/1906.02226

3 Egilmez, H. E. and Ortega, A., "Spectral anomaly detection using graph-based filtering for wireless sensor networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1085–1089.

4 Zhang, J. and Moura, J. M. F., "Diffusion in social networks as sis epidemics: Beyond full mixing and complete graphs," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 537–551, 2014.

5 Anis, A., Gadde, A., and Ortega, A., "Towards a sampling theorem for signals on arbitrary graphs," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3864–3868.

6 Dong, X., Thanou, D., Frossard, P., and Vandergheynst, P., "Learning laplacian matrix in smooth graph signal representations," 2014. [Online]. Available: https://arxiv.org/abs/1406.7842

7 Verma, T. S. and Pearl, J., "On the equivalence of causal models," 2013. [Online]. Available: https://arxiv.org/abs/1304.1108

8 Spirtes, P., Glymour, C., and Scheines, R., *Causation, Prediction, and Search*, 2nd ed. MIT press, 2000.

9 Peters, J., Mooij, J. M., Janzing, D., and Schölkopf, B., "Causal discovery with continuous additive noise models," *Journal of Machine Learning Research*, vol. 15, no. 58, pp. 2009–2053, 2014. [Online]. Available: http://jmlr.org/papers/v15/peters14a.html

10 Shimizu, S., Hoyer, P. O., Hyvrinen, A., and Kerminen, A., "A linear non-gaussian acyclic model for causal discovery," *Journal of Machine Learning Research*, vol. 7, no. 72, pp. 2003–2030, 2006. [Online]. Available: http://jmlr.org/papers/v7/shimizu06a.html

11 Shimizu, S., Inazumi, T., Sogawa, Y., Hyvarinen, A., Kawahara, Y., Washio, T., Hoyer, P. O., and Bollen, K., "Directlingam: A direct method for learning a linear non-gaussian structural equation model," 2011. [Online]. Available: https://arxiv.org/abs/1101.2489

12 Koller, D. and Friedman, N., *Probabilistic Graphical Models: Principles and Techniques*, ser. Adaptive computation and machine learning. MIT Press, 2009. [Online]. Available: https://books.google.co.in/books?id=7dzpHCHzNQ4C

13 Chickering, D. M., "Optimal structure identification with greedy search," *Journal of machine learning research*, vol. 3, no. Nov, pp. 507–554, 2002.

14 Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P., "Dags with no tears: Continuous optimization for structure learning," 2018. [Online]. Available: https://arxiv.org/abs/1803.01422

15 Mateos, G., Segarra, S., Marques, A. G., and Ribeiro, A., "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal Processing Magazine*, vol. 36, no. 3, pp. 16–43, 2019.

16 Dong, X., Thanou, D., Frossard, P., and Vandergheynst, P., "Learning laplacian matrix in smooth graph signal representations," 2014. [Online]. Available: https://arxiv.org/abs/1406.7842

17 Yuan, M. and Lin, Y., "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 03 2007. [Online]. Available: https://doi.org/10.1093/biomet/asm018

18 Friedman, J., Hastie, T., and Tibshirani, R., "Sparse inverse covariance estimation with the lasso," 2007. [Online]. Available: https://arxiv.org/abs/0708.3517

19 Marques, A. G., Segarra, S., and Mateos, G., "Signal processing on directed graphs," 2020. [Online]. Available: https://arxiv.org/abs/2008.00586

20 Mooij, J. M., Janzing, D., Heskes, T., and Schölkopf, B., "On causal discovery with cyclic additive noise models," in *Advances in Neural Information Processing Systems*, Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., Eds., vol. 24. Curran Associates, Inc., 2011. [Online]. Available: https://proceedings.neurips.cc/paper/2011/file/d61e4bbd6393c9111e6526ea173a7c8b-Paper.pdf

21 Lacerda, G., Spirtes, P. L., Ramsey, J., and Hoyer, P. O., "Discovering cyclic causal models by independent components analysis," 2012. [Online]. Available: https://arxiv.org/abs/1206.3273