

RESEARCH

Open Access



Time-varying graph learning from smooth and stationary graph signals with hidden nodes

Rong Ye¹ , Xue-Qin Jiang^{1*}, Hui Feng², Jian Wang³, Runhe Qiu¹ and Xinxin Hou¹

*Correspondence:
xqjiang@dhu.edu.cn

¹ College of Information Science and Technology, Donghua University, Shanghai 201620, China

² School of Information Science and Technology, Fudan University, Shanghai 200433, China

³ School of Data Science, Fudan University, Shanghai 200433, China

Abstract

Learning graph structure from observed signals over graph is a crucial task in many graph signal processing (GSP) applications. Existing approaches focus on inferring static graph, typically assuming that all nodes are available. However, these approaches ignore the situation where only a subset of nodes are available from spatiotemporal measurements, and the remaining nodes are never observed due to application-specific constraints, resulting in time-varying graph estimation accuracy declines dramatically. To handle this problem, we propose a framework that consider the presence of hidden nodes to identify time-varying graph. Specifically, we assume that the graph signals are smooth and stationary on the graphs and only a small number of edges are allowed to change between two consecutive graphs. With these assumptions, we present a challenging time-varying graph inference problem, which models the influence of hidden nodes in terms of estimating the graph-shift operator matrices that have a form of graph Laplacian. Moreover, we emphasize similar edge pattern (column-sparsity) between different graphs. Finally, our method is evaluated on both synthetic and real-world data. The experimental results demonstrate the advantage of our method when compared to existing benchmarking methods.

Keywords: Graph signal processing, Graph learning, Time-varying graphs, Hidden nodes, Graph stationary, Column-sparsity

1 Introduction

Recently, there has been a growing prevalence of modern data analysis that involve structured data with non-Euclidean support. In the real world, numerous examples of such data can be found, including weather measurements data in wireless sensor networks, stock price measurements in financial networks and human behaviors in social networks [1–4]. Typically, graphs are used as efficient mathematical tools to describe the latent structures of such data, where nodes act as entities of the graph and edges model the pairwise relationship between the function values at the nodes. Such graph-based data representation leads to the emerging field of graph signal processing (GSP) [5–7].

In some situations, the underlying network topology is known, but in most cases, there are often settings where the graph is not readily available. However, it is feasible to infer the graph structure from a collection of nodal observations in order to capture the relationships between the different entities. This process is known as graph learning

[8–13]. So far, a significant amount of literature has been proposed to learn graph topology, which is summarized in [11–13]. Particularly, the GSP viewpoint provides a new technique for inferring the graph topology from a set of observations. In general, the GSP-based approaches can generally be categorized into three main groups. The first category of approaches makes assumption about the graph by enforcing properties such as smoothness or sparsity of the graph signals [14–16]. Instead of smoothness/sparsity-based approaches, the second category of approaches assumes that the graph signals are generated from a Laplacian constrained Gaussian–Markov random field (GMRF) [17, 18]. The third category of approaches exploits diffusion model [9, 19] to learn graph topology. The model considers that the observed signals are the outcome of a diffusion process on the graph, where each node continuously influences its neighborhoods.

It should be noted that all the aforementioned graph topology inference works focus on the scenario where observations from all the nodes are available. However, in numerous pertinent scenarios, the observed graph signals may correspond only to a subset of the original graph nodes, while the rest graph nodes are hidden. Neglecting these hidden nodes can drastically hinder the performance of graph topology inference methods. Consequently, some recent works have begun to address this related issue, including Gaussian graphical model selection [20–22], linear Bayesian model [23], and nonlinear regression [24]. More recently, the problem of constructing a graph when consider the existence of hidden nodes has been investigated within the context of GSP [25–27]. Notably, two related works have proposed leveraging, respectively, smoothness prior [5] or stationarity prior [28, 29] to infer graph topologies from incomplete data [25, 26]. Another work has focused on estimating multi-layer graphs in the presence of hidden nodes, assuming that the observed graph signals follow a GMRF model [27]. The existing three GSP-based graph learning methods with hidden nodes are limited to learning static graph or multi-layer graphs.

It came to our attention that some scenarios involve the consideration of time-varying generation models to capture the relationships between data variables in the real world. For example, this is observed when estimating the time-varying brain functional connectivity using electroencephalography recordings (EEGs) or resting-state functional magnetic resonance imaging (fMRI) [30]. Additionally, identifying temporal transitions in biological networks, such as protein, RNA, and DNA [31], and inferring relationships between stock trading from financial market data [32] also exhibit the time-varying nature. To address the growing demand for understanding these time-varying graph structures, several approaches have been proposed. These approaches have leveraged prior assumptions about the evolutionary patterns of time-varying graphs to tackle the problem of learning their topology. In a recent study [33], the authors have proposed an efficient method for inferring time-varying topology. They have utilized Tikhonov regularization to ensure smooth temporal changes in edge weights, thereby capturing the evolving nature of the graphs. To apply additional constraints on the sparse changes of graph, the authors in [34] have introduced a l_1 regularization term for graph variation. Additionally, another time-varying graph learning work has been described in [35], where the authors have proposed to extend the graphical Lasso [36] to account for the temporal variability. While these works in [33–35] did not explicitly incorporate scenarios involving hidden nodes, they served as inspiration for our research. We recognize

the significance of collecting observations from related graphs and leveraging information across time-varying graphs to address the challenge of hidden nodes. However, it remains uncertain how existing algorithms can be adapted to measure graph similarity between unobserved nodes. Consequently, modeling the influence of hidden nodes in the context of time-varying graph learning becomes crucial. For a summary of the proposed method and related graph learning methods, please refer to Table 1.

Building on the preceding discussion, the primary objective of this paper is to address the inference problem of time-varying graphs with the presence of hidden nodes. Our two primary contributions are formulating this problem as a convex optimization problem and devising an algorithm to effectively solve it. Our method is predicated on the assumption that the observed signals exhibit simultaneous smoothness and stationarity on the given graphs. While this assumption has proven successful in the field of static graph inference, a robust formula for time-varying graph learning with the hidden nodes has not yet been established. To fill this gap, it is necessary to modify the classic interpretations of smoothness prior and stationarity prior, in order to account for the impact of hidden nodes. We first adopt a block matrix factorization methodology to revise the smooth and stationary assumptions. Then, we exploit the inherent low-rank and sparse patterns within the blocks associated with hidden nodes. The patterns enable the smooth evolution of graph edges, thereby capturing the temporal dynamics across the sequence of graphs. Furthermore, to fully leverage the characteristics of time-varying graphs, it is crucial to capture the similarity among graphs, accounting for both the observed and hidden nodes. This is achieved through utilization of a similar column-sparsity pattern, which emerges from the similarity analysis of each time slot graph. We test the proposed approach on synthetic and real-world data. Experimental results show that the effectiveness of our proposed approach.

The remainder of this paper is structured as follows. Section 2 provides a comprehensive review of fundamental concepts related to signals defined over graphs, as well as an overview of the associated graph learning methods. Section 3 introduces a time-varying graph learning problem with hidden nodes at hand. Section 4 proposes optimization frameworks to solve this problem. Section 5 is dedicated to the evaluation of the performance of our proposed method. Finally, we conclude the paper in Sect. 6.

Notations: The following notations will be used in this paper. All the vectors are denoted by boldface lower case letters, and the matrices by boldface upper case letters. We use calligraphic font capital letters to denote sets. $\mathbb{R}^{N \times N}$ denotes the set of matrices

Table 1 Comparison between proposed method TGSm-St-GL and alternative

Graph type	Method	Approach	Hidden nodes
Static graph	GS-Rw [25]	Signal stationarity model	✓
	GSm-St-GL [26]	Signal smoothness and stationarity model	✓
Time-varying graph	TGSm-St-GL (proposed)	Signal smoothness and stationarity model	✓
	TVGL-TS [33]	Signal smoothness model	–
	TVGL-SV [34]	Signal smoothness model	–
	TVGL [35]	Graphical lasso-based model	–
Multi-layer graphs	PGL [27]	Signal stationarity model	✓

of size $N \times N$ with nonnegative. For vector \mathbf{x} , $\mathbb{E}[\mathbf{x}]$ represents the expected value of \mathbf{x} . For matrix \mathbf{X} , $\|\mathbf{X}\|_F$ represents the Frobenius norm, $\|\mathbf{X}\|_0$ represents the l_0 norm, $\|\mathbf{X}\|_{F,\text{off}}$ is the Frobenius norm of \mathbf{X} that does not include the diagonal elements, $\|\mathbf{X}\|_*$ represents the nuclear norm, $\|\mathbf{X}\|_{2,1}$ represents the $l_{2,1}$ norm and can be understood as a two-step process where one first obtains the l_2 norm of each of the matrix \mathbf{X} , then, the l_1 norm of the resulting row vector is computed. Moreover, $\text{diag}(\cdot)$ is a diagonal matrix with its argument along the main diagonal, $\text{tr}(\cdot)$ is the trace of the matrix, $\mathbf{1}$ stands for all-one vectors and \mathbf{I} stands for the identity matrix. Finally, the minimization operator, the transpose and pseudo-inverse denoted by \min , superscript \top and superscript † , respectively.

2 Preliminaries

In this section, we first outline some basic GSP definitions. Then, we provide a concise overview of two pivotal models of graph signals, namely smooth graph signals and stationary graph signals. Building on these insights, antecedent works of graph learning problem based on these two graph signal models are introduced. Finally, a general framework for learning time-varying graphs is briefly reviewed.

2.1 Basic definitions for GSP

An undirected and weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ with N nodes are considered here, where $\mathcal{V} = \{1, \dots, N\}$ represents the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. The weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a symmetric matrix, each element of the matrix characterizes the strength of the connection. We also assume that there are no self-loops or directed edge in the graph, which implies $\text{diag}(\mathbf{W}) = \mathbf{0}$. The (i, j) -th entry W_{ij} of the adjacency matrix is assigned a nonnegative value if $(i, j) \in \mathcal{E}$, i and j represent two nodes. We utilize a vector $\mathbf{x} = [x_1, \dots, x_N]^\top \in \mathbb{R}^N$ to represent graph signals, where x_i denotes the value measured at the node i .

In graph theory, the graph Laplacian \mathbf{L} is defined as $\mathbf{L} := \mathbf{D} - \mathbf{W}$. The degree matrix \mathbf{D} is a diagonal matrix that contains the degrees of the nodes along diagonal with entries $D_{ii} = \sum_{j=1}^N W_{ij}$ and $D_{ij} = 0$ for $i \neq j$. The matrix \mathbf{L} can be decomposed into $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ due to its symmetry, where $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N] \in \mathbb{C}^{N \times N}$ is a matrix consisting of the eigenvectors of \mathbf{L} , and $\mathbf{\Lambda} \in \mathbb{C}^{N \times N}$ is a diagonal matrix containing the corresponding eigenvalues arranged in increasing order. The graph shift operator (GSO) is a $N \times N$ square matrix \mathbf{S} that captures the underlying topology of graph. The entries of \mathbf{S} , denoted as S_{ij} , can be only nonzero if $i = j$ or there exists an edge $(i, j) \in \mathcal{E}$ in the graph. The adjacency matrix [37] and the Laplacians [15] are selected as popular options for the GSO. Without loss of generality, \mathbf{S} possesses a complete set of orthonormal eigenvectors and associated eigenvalues.

2.2 Graph signal models

2.2.1 Smooth graph signals

In the node domain, the smoothness of graph signals refers to the tendency for the values of graph signals associated with the two end nodes of edges with significant weights in the graph to exhibit similarity. Typically, in the field of GSP, the total variation (TV) of graph signals \mathbf{x} is commonly interpreted as a smoothness measure, quantified through a quadratic form [5]

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

Intuitively, graph signals \mathbf{x} are said to be smooth when the Laplacian quadratic form $\text{TV}(\mathbf{x})$ is small. In particular, the smaller the values of $\text{TV}(\mathbf{x})$, the smoother the graph signals.

When comes to graph learning problem, the smooth property is widely used as a prior information. Considering the matrix $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_K]$ contains K observations, a general graph learning framework is proposed in the works of [14, 15]

$$\min_{\mathbf{L} \in \mathcal{L}} \text{tr}(\mathbf{X}^\top \mathbf{L} \mathbf{X}) + f(\mathbf{L}). \quad (2)$$

The penalty function $f(\mathbf{L}) = \alpha \|\mathbf{L}\|_F^2 - \beta \mathbf{1}^\top \log(\text{diag}(\mathbf{L}))$ is employed to prevent the acquisition of a trivial solution and controls the sparsity of the learned graph. Parameters α and β are constants. The term $\log(\text{diag}(\mathbf{L}))$ is a two-step process. Firstly, the process obtains the diagonal elements of matrix \mathbf{L} using the diag operation, and then the \log operation is applied to the resulting row vector. Therefore, \log is an element-wise operation. The learned Laplacian matrix has to be in the valid combinatorial Laplacians set, by defining $\mathcal{L} := \{L_{ij} \leq 0, i \neq j; \mathbf{L} = \mathbf{L}^\top; \mathbf{L} \mathbf{1} = \mathbf{0}; \mathbf{L} \succ \mathbf{0}\}$. This constraint emphasizes that \mathbf{L} is a symmetric positive semidefinite matrix. The smoothness of all observed signals over the selected graph is quantified by the first term of equation (2).

2.2.2 Stationary graph signals

Given an undirected graph \mathcal{G} , obviously, GSO \mathbf{S} is symmetric matrix. A linear shift-invariant graph filter $\mathbf{H} \in \mathbb{R}^{N \times N}$ can be written as $\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$, where $\mathbf{h} = [h_0, \dots, h_{L-1}]^\top$ is a vector composed of the graph filter coefficients and $L-1$ denotes the filter degree. Since \mathbf{H} is a polynomial of \mathbf{S} , it readily follows that the matrix \mathbf{H} is also symmetric. For a given input signal \mathbf{s} , the output of the graph filter is simply defined as $\mathbf{x} = \mathbf{H} \mathbf{s}$. Assuming that the \mathbf{s} is a white signal follows a normalized i.i.d Gaussian distribution with mean zero, the output of filter \mathbf{H} is stationary on the graph. This is because the following properties are satisfied

$$\begin{aligned} \mathbf{m}_x &= \mathbb{E}[\mathbf{x}] = \mathbf{H} \mathbb{E}[\mathbf{s}] = \mathbf{0}, \\ \mathbf{C} &= \mathbb{E}[\mathbf{x} \mathbf{x}^\top] = \mathbf{H} \mathbb{E}[\mathbf{s} \mathbf{s}^\top] \mathbf{H}^\top = \mathbf{H} \mathbf{H}^\top = \mathbf{H}^2, \end{aligned} \quad (3)$$

where \mathbf{m}_x denotes the expected value and \mathbf{C} represents the covariance matrix of the graph signals \mathbf{x} . Moreover, since \mathcal{G} is undirected, both \mathbf{S} and \mathbf{C} are symmetric. It becomes apparent that the two diagonalizable matrices GSO \mathbf{S} and \mathbf{C} share common eigenvectors \mathbf{U} in the spectral domain [38] from (3). As a result, we also have that the matrices \mathbf{S} and \mathbf{C} commute.

Thus, the task of learning underlying graph from stationary graph signals is equivalent to inferring its shift operator \mathbf{S} . To be more precise, under the general assumption of graph sparsity, the graph learning problem from stationary graph signals can be formulated as [12]

$$\begin{aligned} \min_{\mathbf{S} \in \mathcal{S}} \quad & \|\mathbf{S}\|_0 \\ \text{s.t.} \quad & \mathbf{C}\mathbf{S} = \mathbf{S}\mathbf{C}, \end{aligned} \quad (4)$$

where the set \mathcal{S} enforces the estimated matrix \mathbf{S} to satisfy some specify properties. $\|\cdot\|_0$ promotes sparse solutions of the matrix \mathbf{S} . The equality constraint enforces that commutativity of the Laplacian and the covariance matrix.

2.2.3 Time-varying graph learning

Time-varying graph learning will learn a series of graphs $\mathbf{L}^{(1)}, \dots, \mathbf{L}^{(T)}$ using the graph signals $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(T)}$ collected during T time periods, where $\mathbf{X}^{(t)} = [\mathbf{x}_1^{(t)}, \dots, \mathbf{x}_K^{(t)}] \in \mathbb{R}^{N \times K}$ contains K observations at a time window t . In this case, the selection of slowly changing time-varying graphs can be accomplished by solving [33]

$$\min_{\mathbf{L}^{(t)} \in \mathcal{L}} \sum_{t=1}^T \left[\text{tr}((\mathbf{X}^{(t)})^\top \mathbf{L}^{(t)} \mathbf{X}^{(t)}) + f(\mathbf{L}^{(t)}) \right] + \eta \sum_{t=2}^T f(\mathbf{L}^{(t)} - \mathbf{L}^{(t-1)}), \quad (5)$$

where the term $f(\mathbf{L}^{(t)} - \mathbf{L}^{(t-1)})$ denotes a regularization term that captures the temporal change in graph edges. The parameter η controls the temporal sparseness.

3 Time-varying graph learning with hidden nodes

In this section, we consider situations where the graph signals are observed only from a subset of nodes during the data collection process. Specifically, Sect. 3.1 involves analysis of latent nodal variables and their influence on the time-varying graph. This is accomplished through the utilization of a block matrix factorization methodology to represent the original matrices. Subsequently, we describe the time-varying graph topology inference problem with hidden nodes, as outlined in Sect. 3.2.

3.1 Time-varying graph model with hidden nodes

Formally, we consider a sequence of graph signals that are partitioned into non-overlapping time windows $\{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(T)}\}$. In this paper, we consider an observation model with hidden nodes where the observed graph signals correspond to a subset of $\mathbf{X}^{(t)}$, while the values of graph signal residing on the remaining nodes have never been observed. We partition the set of nodes \mathcal{V} into disjoint subsets \mathcal{O} and \mathcal{H} , where \mathcal{O} is the set of observable nodes and \mathcal{H} is the set of hidden nodes with $\mathcal{H} = \mathcal{V} \setminus \mathcal{O}$. In particular, we set $\mathcal{O} = \{1, \dots, O\}$ with cardinality $|\mathcal{O}| = O$ and $\mathcal{H} = \{O + 1, \dots, N\}$ with cardinality $|\mathcal{H}| = H = N - O$. We represent the graph signal values of observed nodes by the $O \times K$ submatrix $\mathbf{X}_O^{(t)}$ associated with the first O rows of $\mathbf{X}^{(t)}$. As described in Sect. 2, the sample covariance matrices and GSO corresponding to the observed graph signals are given by $\hat{\mathbf{C}}_O^{(t)}$ and $\mathbf{S}_O^{(t)}$, respectively. To this end, for each time slot graph, the matrices $\mathbf{S}^{(t)}$ and $\hat{\mathbf{C}}^{(t)}$ can be described by block structure as

$$\mathbf{S}^{(t)} = \begin{bmatrix} \mathbf{S}_O^{(t)} & \mathbf{S}_{OH}^{(t)} \\ \mathbf{S}_{HO}^{(t)} & \mathbf{S}_H^{(t)} \end{bmatrix}, \hat{\mathbf{C}}^{(t)} = \begin{bmatrix} \hat{\mathbf{C}}_O^{(t)} & \hat{\mathbf{C}}_{OH}^{(t)} \\ \hat{\mathbf{C}}_{HO}^{(t)} & \hat{\mathbf{C}}_H^{(t)} \end{bmatrix}. \quad (6)$$

Here, the submatrices $\mathbf{S}_O^{(t)}, \mathbf{S}_{OH}^{(t)}, \mathbf{S}_H^{(t)}$ specify the dependencies among the observed nodes, between the observed and hidden nodes and among the hidden nodes, respectively.

In particular, the sample covariance of the observed graph signals is represented by $\hat{\mathbf{C}}_O^{(t)} = \frac{1}{K} \mathbf{X}_O^{(t)} (\mathbf{X}_O^{(t)})^\top$. The undirected graphs follow that $\mathbf{S}^{(t)} = (\mathbf{S}^{(t)})^\top$ and $\hat{\mathbf{C}}^{(t)} = (\hat{\mathbf{C}}^{(t)})^\top$ due to both matrices $\mathbf{S}^{(t)}$ and $\hat{\mathbf{C}}^{(t)}$ are symmetric. Similarly, the submatrices $\mathbf{S}_{OH}^{(t)}$ and $\hat{\mathbf{C}}_{OH}^{(t)}$ also exhibit the property of symmetry.

As we can see, the block structure of the matrices $\mathbf{S}^{(t)}$ and $\hat{\mathbf{C}}^{(t)}$ in (6) motivates the search for optimal time-varying graphs when consider the existence of hidden nodes. Next, the problem of time-varying graph learning with hidden nodes will be introduced.

3.2 Problem statement

Given the known nodal subset $\mathcal{O} \subset \mathcal{V}$, and the matrices $\{\mathbf{X}_O^{(t)}\}_{t=1}^T$ collect the graph signal values of observed nodes arising from unknown time-varying graphs $\{\mathcal{G}^{(t)}\}_{t=1}^T$. Our objective is to learn the time-varying graph while accounting for the presence of hidden nodes, which is tantamount to learn the GSO sequence $\{\mathbf{S}_O^{(t)}\}_{t=1}^T$ from $\{\mathbf{X}_O^{(t)}\}_{t=1}^T$ if the following assumptions hold

1. The number of hidden nodes far less than the number of observed nodes with cardinality $H \ll O$;
2. The full observations $\{\mathbf{X}^{(t)}\}_{t=1}^T$ satisfy the prior assumption that they are smooth and stationary in $\mathbf{S}^{(t)}$ simultaneously;
3. The number of graph edges permitted to change between consecutive graphs is limited according to a particular function $\psi(\mathbf{S}^{(t)} - \mathbf{S}^{(t-1)})$, a prior that graph edges change smoothly in time.

The task of learning time-varying graphs encoded in the matrices $\{\mathbf{S}_O^{(t)}\}_{t=1}^T$ presents a challenging problem due to the absence of observations from nodes in set \mathcal{H} . To address this problem, the above three assumptions are made to render the problem more tractable. Firstly, the assumption (1) guarantees the availability of information for the majority of nodes. Secondly, the assumption (2) establishes a well-defined relationship between the graph signals and the unknown time-varying graphs. Lastly, the assumption (3) enforces that the graph edges change smoothly over time, providing temporal relations that may exist in time-varying graphs.

4 Proposed optimization framework

In this section, the influence of hidden nodes on smoothness prior and stationarity prior is presented, respectively. Following this, an optimization framework is designed to address the time-varying graph learning problem with hidden nodes, considering the scenario where the observed graph signals are both smooth and stationary.

4.1 Influence of hidden nodes on smoothness prior

The smoothness of signals on time-varying graphs can be computed as $\frac{1}{K} \text{tr}((\mathbf{X}^{(t)})^\top \mathbf{L}^{(t)} \mathbf{X}^{(t)})$. In this part, we focus on $\hat{\mathbf{C}}^{(t)} = \frac{1}{K} \mathbf{X}^{(t)} (\mathbf{X}^{(t)})^\top$, and thus, the TV of graph signals is equal to $\text{tr}(\hat{\mathbf{C}}^{(t)} \mathbf{L}^{(t)})$. However, the existence of hidden nodes restricts our access solely to the observed sampled covariance matrices $\hat{\mathbf{C}}_O^{(t)}$. Regarding the block structure of matrices $\hat{\mathbf{C}}^{(t)}$ and $\mathbf{S}^{(t)}$ defined in (6), the smoothness of signals within the context of time-varying graphs can be rewritten as

$$\begin{aligned}
\text{tr}(\hat{\mathbf{C}}^{(t)}\mathbf{L}^{(t)}) &= \text{tr}(\hat{\mathbf{C}}_O^{(t)}\mathbf{L}_O^{(t)}) + 2\text{tr}(\hat{\mathbf{C}}_{OH}^{(t)}(\mathbf{L}_{OH}^{(t)})^\top) + \text{tr}(\hat{\mathbf{C}}_H^{(t)}\mathbf{L}_H^{(t)}) \\
&= \text{tr}(\hat{\mathbf{C}}_O^{(t)}\mathbf{L}_O^{(t)}) + 2\text{tr}(\mathbf{P}^{(t)}) + \text{tr}(\mathbf{R}^{(t)}) \\
&= \text{tr}(\hat{\mathbf{C}}_O^{(t)}\mathbf{L}_O^{(t)}) + 2\text{tr}(\mathbf{P}^{(t)}) + r^{(t)},
\end{aligned} \tag{7}$$

where the matrices $\mathbf{P}^{(t)} := \hat{\mathbf{C}}_{OH}^{(t)}(\mathbf{L}_{OH}^{(t)})^\top \in \mathbb{R}^{O \times O}$, $\mathbf{R}^{(t)} := \hat{\mathbf{C}}_H^{(t)}\mathbf{L}_H^{(t)} \in \mathbb{R}^{H \times H}$, and $r^{(t)} := \text{tr}(\mathbf{R}^{(t)})$ are nonnegative variables. The first equation in (7) represents the block-wise smoothness of graph signals. However, we do not have knowledge of most of the submatrices related to the hidden nodes. By lifting the matrices $\mathbf{P}^{(t)}$ and $\mathbf{R}^{(t)}$, we circumvent this challenge and solve the time-varying topology inference as a convex problem.

Notice that the matrices $\mathbf{L}_O^{(t)}$ belong to the set $\tilde{\mathcal{L}} := \{L_{ij} \leq 0, i \neq j; \mathbf{L} = \mathbf{L}^\top; \mathbf{L}\mathbf{1} \geq 0; \mathbf{L} > 0\}$, which are different from the set of valid combinatorial Laplacians \mathcal{L} . The only difference between the two set is to replace the condition $\mathbf{L}\mathbf{1} = 0$ with $\mathbf{L}\mathbf{1} \geq 0$, while others remain unchanged. The existence of links between the elements in \mathcal{O} and the elements in \mathcal{H} gives rise to nonzero (negative) entries in $\mathbf{L}_{OH}^{(t)}$ and, as a result, the sum of the off-diagonal elements of can be smaller than the value of the associated diagonal elements (which account for the links in both \mathcal{O} and \mathcal{H}). Therefore, $\mathbf{L}_O^{(t)}$ is not a combinatorial Laplacian.

Indeed, we encounter the challenge that $\mathbf{L}_O^{(t)}$ are not Laplacians themselves, while tackling the time-varying graph topology inference from smooth observations with hidden nodes. In order to circumvent this challenging issue, we turn to estimating $\tilde{\mathbf{L}}_O^{(t)} := \text{diag}(\mathbf{A}_O^{(t)}\mathbf{1}) - \mathbf{A}_O^{(t)}$ rather than $\mathbf{L}_O^{(t)}$, where $\mathbf{A}_O^{(t)}$ represent the adjacency matrices of the observed graph signals in the t th time slot. With this consideration in mind, the matrices $\tilde{\mathbf{L}}_O^{(t)}$ are proper combinatorial Laplacians satisfy the conditions for the valid set of graph Laplacians \mathcal{L} . We formulate the relation between $\mathbf{L}_O^{(t)}$ and $\tilde{\mathbf{L}}_O^{(t)}$ with equation $\tilde{\mathbf{L}}_O^{(t)} = \mathbf{L}_O^{(t)} - \mathbf{D}_{OH}^{(t)}$. We use degree matrices $\mathbf{D}_{OH}^{(t)}$ to represent the edges existing between the observed and hidden nodes, which is defined as $\mathbf{D}_{OH}^{(t)} := \text{diag}(\mathbf{A}_{OH}^{(t)}\mathbf{1}) \in \mathbb{R}^{O \times H}$. By leveraging the matrices $\tilde{\mathbf{L}}_O^{(t)}$, we take the place of the smoothness penalty in (7) as

$$\begin{aligned}
\text{tr}(\hat{\mathbf{C}}^{(t)}\mathbf{L}^{(t)}) &= \text{tr}(\hat{\mathbf{C}}_O^{(t)}\tilde{\mathbf{L}}_O^{(t)}) + \text{tr}(\hat{\mathbf{C}}_O^{(t)}\mathbf{D}_{OH}^{(t)}) + 2\text{tr}(\mathbf{P}^{(t)}) + r^{(t)} \\
&= \text{tr}(\hat{\mathbf{C}}_O^{(t)}\tilde{\mathbf{L}}_O^{(t)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t)}) + r^{(t)},
\end{aligned} \tag{8}$$

where $\tilde{\mathbf{P}}^{(t)} := \hat{\mathbf{C}}_O^{(t)}\mathbf{D}_{OH}^{(t)}/2 + \mathbf{P}^{(t)}$. With the assumption (1), it is obvious that the matrices $\mathbf{P}^{(t)}$ are low-rank matrices with $\text{rank}(\mathbf{P}^{(t)}) \leq H \ll O$. Furthermore, the matrices $\mathbf{D}_{OH}^{(t)}$ are low-rank matrices, if the graphs are sparse. Thus, it can be inferred that $\tilde{\mathbf{P}}^{(t)}$ also exhibits a low-rank structure.

4.2 Influence of hidden nodes on stationarity prior

Upon evaluating the impact of the smoothness assumption on the time-varying graph learning problem involving hidden nodes, we proceed to consider that the graph signals to be stationary over the whole graphs. This graph signals model leads us to the conclusion that the eigenvectors of $\mathbf{C}^{(t)}$ and $\mathbf{S}^{(t)}$ are identical, thereby the equation $\mathbf{C}^{(t)}\mathbf{S}^{(t)} = \mathbf{S}^{(t)}\mathbf{C}^{(t)}$ holds. To this end, we leverage the block structure of matrices $\mathbf{C}^{(t)}$ and $\mathbf{S}^{(t)}$, with a specific focus on the upper left block on both sides of the equation $\mathbf{C}^{(t)}\mathbf{S}^{(t)} = \mathbf{S}^{(t)}\mathbf{C}^{(t)}$, to model the impact of hidden nodes on the stationarity prior

$$\mathbf{C}_O^{(t)} \mathbf{S}_O^{(t)} + \mathbf{C}_{OH}^{(t)} (\mathbf{S}_{OH}^{(t)})^\top = \mathbf{S}_O^{(t)} \mathbf{C}_O^{(t)} + \mathbf{S}_{OH}^{(t)} (\mathbf{C}_{OH}^{(t)})^\top. \quad (9)$$

Equation (9) reveals that we can't simply focus on $\mathbf{C}_O^{(t)} \mathbf{S}_O^{(t)} = \mathbf{S}_O^{(t)} \mathbf{C}_O^{(t)}$ when the hidden nodes are presented, but also need to notice that the associate terms $\mathbf{C}_{OH}^{(t)} (\mathbf{S}_{OH}^{(t)})^\top$ and $\mathbf{S}_{OH}^{(t)} (\mathbf{C}_{OH}^{(t)})^\top$. Furthermore, we set the matrices $\tilde{\mathbf{P}}^{(t)} = \mathbf{C}_{OH}^{(t)} (\mathbf{S}_{OH}^{(t)})^\top$, similar to the definition of $\mathbf{P}^{(t)}$. The key distinction lies in our utilization of the matrices $\mathbf{S}_{OH}^{(t)}$ instead of the Laplacians $\mathbf{L}_{OH}^{(t)}$ to associate the matrices $\tilde{\mathbf{P}}^{(t)}$. Under this setting, equation (9) can be formulated as

$$\mathbf{C}_O^{(t)} \mathbf{S}_O^{(t)} + \tilde{\mathbf{P}}^{(t)} = \mathbf{S}_O^{(t)} \mathbf{C}_O^{(t)} + (\tilde{\mathbf{P}}^{(t)})^\top. \quad (10)$$

Similar to the analysis of $\tilde{\mathbf{P}}^{(t)}$ in section 4.1, the matrices $\tilde{\mathbf{P}}^{(t)}$ are also low-rank matrices. We will exploit the low-rank structure of the matrices $\tilde{\mathbf{P}}^{(t)}$ and $\tilde{\mathbf{P}}^{(t)}$ in our formulation.

4.3 Smoothness prior versus stationarity prior

Supposing that we are given with two datasets, \mathbf{X}_1 and \mathbf{X}_2 , each containing an equal number of graph signals. Specifically, we know that the signals in \mathbf{X}_1 exhibit smoothness characteristics on the graph, and another set of the signals in \mathbf{X}_2 are stationary on the graph. Based on this information, we are able to identify the underlying graph. It is of interest to see which one leads to a better graph topology inference result. Without loss of generality, graph smoothness is a more lenient assumption that limits the TV of the observed values of the graph signal to be small. However, graph stationary outperforms the smoothness-based method, as it has a much better prior assumption with significantly restricts the GSO. In the meantime, there may arise an instance where the observations are both stationary and smooth on the graph. More precisely, it means that the covariance matrix of the observations is diagonalized by eigenvectors with the graph Laplacian and the graph signals is low-frequency-based. In such settings, two graph recovery methods can be combined to enhance recovery performance, which will be explored in the subsequent subsection.

4.4 Topology inference based on smoothness prior and stationarity prior

Taking the assumption (3) into account, the task of learning time-varying graph with the presence of hidden nodes involves acquiring knowledge of a sequence of graphs $\{\mathbf{S}_O^{(t)}\}_{t=1}^T$ from the observed graph signals $\mathbf{X}_O^{(1)}, \dots, \mathbf{X}_O^{(T)}$ collected during T time periods. The task specifically concentrates on the scenario where the GSO is represented by the Laplacian matrix, namely, our ultimate target corresponds to infer $\{\mathbf{L}_O^{(t)}\}_{t=1}^T$. We assume that the observed signals exhibit both smoothness and stationarity characteristics on unknown time-varying graphs. As a result, the smoothness penalty described in (8) and the commutativity constraint accounting for stationary equation in (10) can be jointly considered to approach time-varying graph learning problem. More specifically, this problem can be formulated as the ensuing objective function

$$\begin{aligned}
& \min_{\{\tilde{\mathbf{L}}_O^{(t)}, \tilde{\mathbf{P}}^{(t)}, \tilde{\mathbf{r}}^{(t)}\}_{t=1}^T} \sum_{t=1}^T \left[\text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t)}) + r^{(t)} + \alpha \|\tilde{\mathbf{L}}_O^{(t)}\|_{F,off}^2 \right. \\
& \quad \left. - \beta \mathbf{1}^\top \log(\text{diag}(\mathbf{Q}_O^{(t)})) \right] + \eta \sum_{t=2}^T \psi(\mathbf{Q}_O^{(t)} - \mathbf{Q}_O^{(t-1)}) \\
& \text{s.t.} \quad \text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t)}) + r^{(t)} \geq 0, \\
& \quad \hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)} + \tilde{\mathbf{P}}^{(t)} = \tilde{\mathbf{L}}_O^{(t)} \hat{\mathbf{C}}_O^{(t)} + (\tilde{\mathbf{P}}^{(t)})^\top, \\
& \quad \text{rank}(\tilde{\mathbf{P}}^{(t)}) \leq H, \\
& \quad \text{rank}(\tilde{\mathbf{P}}^{(t)}) \leq H, \\
& \quad \tilde{\mathbf{L}}_O^{(t)} \in \mathcal{L},
\end{aligned} \tag{11}$$

where γ_* and η are tuning parameters. The term $\|\tilde{\mathbf{L}}_O^{(t)}\|_{F,off}^2$ offers a handle on the level of sparsity. The penalty function $\psi(\cdot)$ imposes a constraint that limits the number of edge changes between consecutive graphs to a small value and we set function $\psi(\tilde{\mathbf{L}}_O^{(t)} - \tilde{\mathbf{L}}_O^{(t-1)}) = \|\tilde{\mathbf{L}}_O^{(t)} - \tilde{\mathbf{L}}_O^{(t-1)}\|_F^2$. The first constraint ensures that the TV of graph signals is non-negative. The equality constraint enforces that commutativity of the Laplacians and the covariance matrices when consider the presence of hidden nodes. The two rank constraints capture the fact that the low rankness property of $\tilde{\mathbf{P}}^{(t)}$ and $\tilde{\mathbf{P}}^{(t)}$.

In most instances, it is not feasible to obtain the entire covariance $\mathbf{C}_O^{(t)}$. Therefore, we resort to relying on the sample covariance matrices $\hat{\mathbf{C}}_O^{(t)}$ and relax the stationary constrain to guarantee that the left and right terms of the original equation (10) are roughly equivalent, though not entirely so. It is worth noting that under this more lenient condition, $\tilde{\mathbf{P}}^{(t)}$ and $\mathbf{P}^{(t)}$ are equivalent. In such circumstances, we focus on $\text{rank}(\tilde{\mathbf{P}}^{(t)})$ only and exploit the nuclear norm to capture the low-rank structure of matrices $\tilde{\mathbf{P}}^{(t)}$. To this end, we reformulate the optimization objective function (11) as

$$\begin{aligned}
& \min_{\{\tilde{\mathbf{L}}_O^{(t)}, \tilde{\mathbf{P}}^{(t)}, \tilde{\mathbf{r}}^{(t)}\}_{t=1}^T} \sum_{t=1}^T \left[\text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t)}) + r^{(t)} + \alpha \|\tilde{\mathbf{L}}_O^{(t)}\|_{F,off}^2 \right. \\
& \quad \left. - \beta \mathbf{1}^\top \log(\text{diag}(\mathbf{Q}_O^{(t)})) + \gamma_* \|\tilde{\mathbf{P}}^{(t)}\|_* \right] + \eta \sum_{t=2}^T \|\mathbf{Q}_O^{(t)} - \mathbf{Q}_O^{(t-1)}\|_F^2 \\
& \text{s.t.} \quad \text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t)}) + r^{(t)} \geq 0, \\
& \quad \|\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)} + \tilde{\mathbf{P}}^{(t)} - \tilde{\mathbf{L}}_O^{(t)} \hat{\mathbf{C}}_O^{(t)} - (\tilde{\mathbf{P}}^{(t)})^\top\|_F^2 \leq \epsilon, \\
& \quad \tilde{\mathbf{L}}_O^{(t)} \in \mathcal{L},
\end{aligned} \tag{12}$$

where the nuclear norm penalty $\|\tilde{\mathbf{P}}^{(t)}\|_*$ is employed to encourage low-rank solutions by favoring matrices with sparse singular values. The non-negative constant ϵ is an essential parameter that characterizes the accuracy of the sample covariance. The value of the parameter under consideration is inherently related to the amount of noise and the total number of samples K . This value is used as an indicator of the accuracy and faithfulness of the estimated covariance.

Based on previous analysis, the matrices $\tilde{\mathbf{P}}^{(t)}$ are not only inseparable from the product of the matrices $\hat{\mathbf{C}}_O^{(t)}$ and $\mathbf{D}_{OH}^{(t)}$ but also related to the matrices $\mathbf{P}^{(t)}$. Recalling that the diagonal of $\mathbf{D}_{OH}^{(t)}$ are sparse, it is obviously that $\hat{\mathbf{C}}_O^{(t)} \mathbf{D}_{OH}^{(t)}$ are the matrices with

several zero columns. More precisely, the assumption (1) reveals the presence of a column-sparse structure in the matrices $\tilde{\mathbf{P}}^{(t)}$. However, the rank constraint fails to preserve the desired columns sparsity characteristic. Following the classical approach in the literature, an efficient way to circumvent this issue is to replace the nuclear norm with the group Lasso penalty. This penalty not only reduces the number of nonzero columns but also promotes solutions with a low rankness.

To further improve the performance of graph topology inference, we consider leveraging the aforementioned column-sparsity regularization. Taking this consideration into account, a convex optimization problem for solving the time-varying graph learning with hidden nodes is proposed

$$\begin{aligned}
 \min_{\{\tilde{\mathbf{L}}_O^{(t)}, \tilde{\mathbf{P}}^{(t)}, r^{(t)}\}_{t=1}^T} & \sum_{t=1}^T \left[\text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t)}) + r^{(t)} + \alpha \|\tilde{\mathbf{L}}_O^{(t)}\|_{F,off}^2 \right. \\
 & \left. - \beta \mathbf{1}^\top \log(\text{diag}(\mathbf{Q}_O^{(t)})) + \gamma_{2,1} \|\tilde{\mathbf{P}}^{(t)}\|_{2,1} \right] + \theta \sum_{t=2}^T \left\| \begin{bmatrix} \tilde{\mathbf{P}}^{(t)} \\ \tilde{\mathbf{P}}^{(t-1)} \end{bmatrix} \right\|_{2,1} \\
 & + \eta \sum_{t=2}^T \|\tilde{\mathbf{L}}_O^{(t)} - \tilde{\mathbf{L}}_O^{(t-1)}\|_F^2 \\
 & + \rho \sum_{t=1}^T \|\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)} + \tilde{\mathbf{P}}^{(t)} - \tilde{\mathbf{L}}_O^{(t)} \hat{\mathbf{C}}_O^{(t)} - (\tilde{\mathbf{P}}^{(t)})^\top\|_F^2 \\
 \text{s.t.} & \text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t)}) + r^{(t)} \geq 0, \\
 & \tilde{\mathbf{L}}_O^{(t)} \in \mathcal{L}.
 \end{aligned} \tag{13}$$

The assumption (3) guarantees the similarity of temporally adjacent graphs. To exploit this property, we construct a tall matrix consisting of the matrices $\tilde{\mathbf{P}}^{(t)}$ and $\tilde{\mathbf{P}}^{(t-1)}$. By applying the $l_{2,1}$ norm to the tall matrix, we are able to capture and preserve the desired column-sparsity characteristic. This approach allows us to effectively leverage the temporal similarity between adjacent graphs, ensuring that columns with nonzero entries are likely to be consistently positioned across the varying matrices $\tilde{\mathbf{P}}^{(t)}$. This is particularly important to consider this additional structure, which is helpful to improve the estimation of $\tilde{\mathbf{P}}^{(t)}$ and result in a better recovery performance of $\tilde{\mathbf{L}}_O^{(t)}$. The effectiveness of the formulation (13) in promoting the desired column-sparsity pattern is demonstrated through the experimental results in Sect. 5.3.

We solve the optimization problem (13) by adopting an alternating minimization scheme. To find a numerically efficient solution, we decouple (13) into three simpler optimization problems. Specifically, with $m = 0, \dots, M - 1$ being the iteration index, we initialize two variables $\tilde{\mathbf{P}}^{(t),(m)}$ and $r^{(t),(m)}$. At the first step, for the given $\tilde{\mathbf{P}}^{(t),(m)}$ and $r^{(t),(m)}$, we solve the following optimization problem with respect to $\tilde{\mathbf{L}}_O^{(t)}$

$$\begin{aligned}
\tilde{\mathbf{L}}_O^{(t),(m+1)} := & \min_{\tilde{\mathbf{L}}_O^{(t)} \in \mathcal{L}} \sum_{t=1}^T \left[\text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)}) + \alpha \|\tilde{\mathbf{L}}_O^{(t)}\|_{F,off}^2 - \beta \mathbf{1}^\top \log(\text{diag}(\mathbf{Q}_O^{(t)})) \right] \\
& + \rho \sum_{t=1}^T \|\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)} + \tilde{\mathbf{P}}^{(t),(m)} - \tilde{\mathbf{L}}_O^{(t)} \hat{\mathbf{C}}_O^{(t)} - (\tilde{\mathbf{P}}^{(t),(m)})^\top\|_F^2 \\
& + \eta \sum_{t=2}^T \|\tilde{\mathbf{L}}_O^{(t)} - \tilde{\mathbf{L}}_O^{(t-1)}\|_F^2 \\
\text{s.t. } & \text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t),(m)}) + r^{(t),(m)} \geq 0.
\end{aligned} \tag{14}$$

At the second step, we fix $r^{(t),(m)}$ and leverage the estimate $\tilde{\mathbf{L}}_O^{(t),(m+1)}$ from the previous step to optimize the objective function with respect to $\tilde{\mathbf{P}}^{(t),(m+1)}$, which leads to the following optimization problem

$$\begin{aligned}
\tilde{\mathbf{P}}^{(t),(m+1)} := & \min_{\tilde{\mathbf{P}}^{(t)}} \sum_{t=1}^T \left[2\text{tr}(\tilde{\mathbf{P}}^{(t)}) + \gamma_{2,1} \|\tilde{\mathbf{P}}^{(t)}\|_{2,1} \right] + \theta \sum_{t=2}^T \left\| \begin{bmatrix} \tilde{\mathbf{P}}^{(t)} \\ \tilde{\mathbf{P}}^{(t-1)} \end{bmatrix} \right\|_{2,1} \\
& + \rho \sum_{t=1}^T \|\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t),(m+1)} + \tilde{\mathbf{P}}^{(t)} - \tilde{\mathbf{L}}_O^{(t),(m+1)} \hat{\mathbf{C}}_O^{(t)} - (\tilde{\mathbf{P}}^{(t)})^\top\|_F^2 \\
\text{s.t. } & \text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t),(m+1)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t)}) + r^{(t),(m)} \geq 0.
\end{aligned} \tag{15}$$

Algorithm 1 Time-varying graph learning method with hidden nodes

-
- Input** : Parameters $\alpha, \beta, \gamma_{2,1}, \theta, \eta, \rho$, original graph signals $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(T)} \in \mathbb{R}^{N \times K}$, time slots T , number of iterations M , number of hidden nodes H ,
Output : Estimate $\hat{\mathbf{L}}_O^{(1)}, \dots, \hat{\mathbf{L}}_O^{(T)}$
Initialize: $\tilde{\mathbf{P}}^{(t),(0)}$ and $r^{(t),(0)}, t = 1, \dots, T$
- 1 Select the set of hidden nodes $\mathcal{H} = \{1, \dots, H\}$ by random selection
 Select the remaining $O = N - H$ rows of original graph signals as observed graph signals $\mathbf{X}_O^{(1)}, \dots, \mathbf{X}_O^{(T)}$
 Compute the sampled covariance matrix $\hat{\mathbf{C}}_O^{(t)}$
 - 2 **for** $m = 0$ **to** $M - 1$ **do**
 - Update $\tilde{\mathbf{L}}_O^{(t),(m+1)}$ by solving (14)
 - Update $\tilde{\mathbf{P}}^{(t),(m+1)}$ by solving (15)
 - Update $r^{(t),(m+1)}$ by solving (16)
 - 3 **end**
 - 4 $\tilde{\mathbf{L}}_O^{(t)} = \tilde{\mathbf{L}}_O^{(t),(M)}, \tilde{\mathbf{P}}^{(t)} = \tilde{\mathbf{P}}^{(t),(M)}, r^t = r^{(t),(M)}$
-

At the last step, according to the $\tilde{\mathbf{L}}_O^{(t),(m+1)}$ and $\tilde{\mathbf{P}}^{(t),(m+1)}$ obtained in the first two steps, we solve the following convex optimization problem with respect to $r^{(t),(m+1)}$

$$\begin{aligned}
r^{(t),(m+1)} := & \min_{r^{(t)}} \sum_{t=1}^T r^{(t)} \\
\text{s.t. } & \text{tr}(\hat{\mathbf{C}}_O^{(t)} \tilde{\mathbf{L}}_O^{(t),(m+1)}) + 2\text{tr}(\tilde{\mathbf{P}}^{(t),(m+1)}) + r^{(t)} \geq 0.
\end{aligned} \tag{16}$$

We alternate between the three steps outlined in (14), (15) and (16) to obtain the final solution for the optimization problem described in (13). We generally observe convergence within a few iterations. The algorithm is summarized in Algorithm 1.

5 Numerical experiments

In this section, we present some numerical results validating the effectiveness of the proposed time-varying graph learning method for both synthetic and real-world data. The proposed method (hereinafter called TGSm-St-GL) is compared with benchmarking methods, including static graph learning from smooth and stationary graph signals with hidden nodes (GSm-St-GL) [26], time-varying graph learning method based on temporal smoothness (TVGL-TS) [33] and time-varying graph learning method based on sparse variation (TVGL-SV) [34]. We commence with an introduction of the general experimental settings. Next, we assess the efficacy of our method using synthetic data and conduct a comparative analysis of our method against established classical methods. Finally, we introduce the simulation performed over one real-world data. In our experiments, we solve optimization problems using CVX [39], which is a package for solving convex programs.

5.1 Experimental settings

5.1.1 Evaluation metrics

We employ five evaluation metrics to assess the performance of our proposed method. The first evaluation metric is Mean Error that measures the estimation accuracy of recovered graphs. The Mean Error is defined as $\|\hat{\mathbf{L}}_O^{(t)} - (\mathbf{L}_O^{(t)})^*\|_F^2 / \|(\mathbf{L}_O^{(t)})^*\|_F^2$, where $\hat{\mathbf{L}}_O^{(t)}$ are the estimated Laplacian matrices and $(\mathbf{L}_O^{(t)})^*$ are the ground truth. Additionally, we utilize three metrics, namely Precision, Recall and Fscore, to evaluate how effectively the true edge structure of the graph is captured. More precisely, the Fscore provides a measure of the accuracy in estimating the graph topology, which is closely related to the metrics Precision and Recall. The Fscore ranges between 0 and 1, with higher values indicating better performance in capturing the graph topology. The mutual dependence between the obtained edge set and the ground-truth graph is measured by the last evaluation metric Normalized Mutual Information (NMI).

5.1.2 Baseline methods

We discuss various related graph learning strategies to compare with our proposed method. The first is GSm-St-GL, stands for the static graph learning method. This method considers the existence of hidden nodes and assumes that the entire graph signals exhibit both smoothness and stationarity simultaneously. The second is TGSm-St-GL-nh, a time-varying graph inference method that aims to address the same problem as TGSm-St-GL but ignores the presence of hidden nodes. The other two baseline time-varying graph learning schemes are TVGL-TS and TVGL-SV. These two schemes are applicable when all nodes are available and the graph signals satisfy the stationarity assumption.

5.2 Synthetic data

We create a type of synthetic graph signals generated from a time-varying Erdős–Rényi graph (abbreviated as TV-ER graph). The process of constructing the data involves two steps: firstly, the creation of the TV-ER graph; secondly, the generation of time-varying graph signals by utilizing probability distributions based on the graph Laplacians of the aforementioned TV-ER graph.

5.2.1 Time-varying graph construction

In general, the generation of a TV-ER graph involves two steps. At the first step, an initial static ER graph $\mathcal{G}^{(1)}$ is constructed. The graph $\mathcal{G}^{(1)}$ consists of $N = 20$ nodes, and we set the edge connection probability $s = 0.3$. At the second step, we change the connections of edges in the original ER graph over time to construct the TV-ER graph. The t th graph $\mathcal{G}^{(t)}$ is obtained by resampling 10% of edges from the previous graph $\mathcal{G}^{(t-1)}$. In this way, we construct a set of graphs, such that only a few edges switch at a time while most of the edges remain unchanged, i.e., the set of graphs follow the assumption (3). The edge weights of graphs belong to the set $\{0, 1\}$.

5.2.2 Generating synthetic graph signals

We generate time-varying graph signals by utilizing distributions derived from the graph Laplacians of the TV-ER graph that we construct. Let $\mathbf{L}^{(t)}$ represents the graph Laplacian of a graph at a certain time slot t , we can write its eigendecomposition as $\mathbf{L}^{(t)} = \mathbf{U}^{(t)} \Lambda^{(t)} (\mathbf{U}^{(t)})^\top$. We create the smooth graph signals as $\mathbf{X}^{(t)} = \mathbf{U}^{(t)} \mathbf{Z}^{(t)}$ with $K = 50$, where $\mathbf{Z}^{(t)} \sim \mathcal{N}(\mathbf{0}, (\Lambda^{(t)})^\dagger)$. It is worth mentioning that the covariance of $\mathbf{X}^{(t)}$ is represented as $\mathbf{C}^{(t)} = ((\mathbf{L}^{(t)})^\dagger)^2$. Hence, the graph signals generated from this model satisfy the assumption of being both smooth and stationary on the time-varying graphs.

5.3 Results on synthetic data

We conduct several experiments to investigate the behavior of our proposed method and the other baseline methods on synthetic data. Different settings are considered in these experiments, including the number of hidden nodes, the noise level, and the column-sparse structure of matrices $\tilde{\mathbf{P}}^{(t)}$.

5.3.1 Number of hidden nodes

We assess the performance of each method by varying the number of hidden nodes and set $\mathcal{H} = \{1, 2, 3, 4, 5\}$. We select the hidden nodes from all nodes in the graph by random selection.

The results in Fig. 1 show that the performance comparisons for different number of hidden nodes based on the TV-ER graph. The Mean Error of recovered graphs and variation of the Fscore are reported in Fig. 1a, b, respectively. It can be seen that the Mean Error increases with the growing H and the Fscore decreases with the growing H for TGSm-St-GL. This observation highlights the significant influence of hidden nodes on time-varying graph recovery. The comparison depicted in Fig. 1 further supports the conclusion that the proposed method outperforms TGSm-St-GL-nh. This

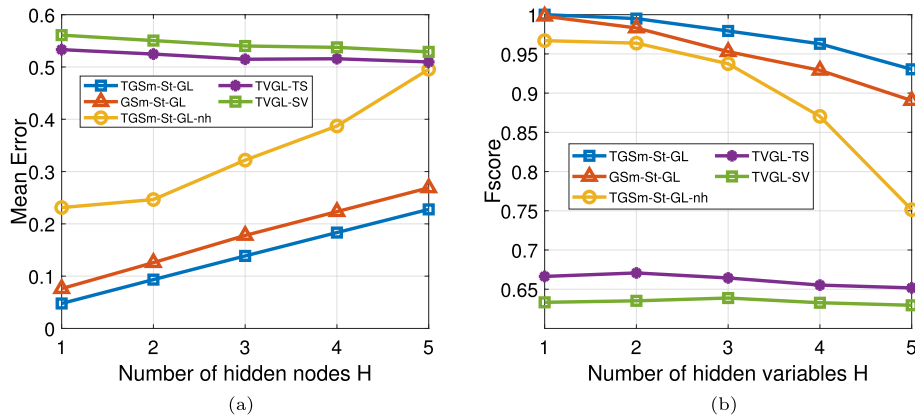


Fig. 1 Numerical validation of the proposed algorithm. **a** Mean Error of the recovered graphs for several algorithms as the number of hidden nodes increases. **b** Fscore of TV-ER graphs as the number of hidden nodes increases for different algorithms

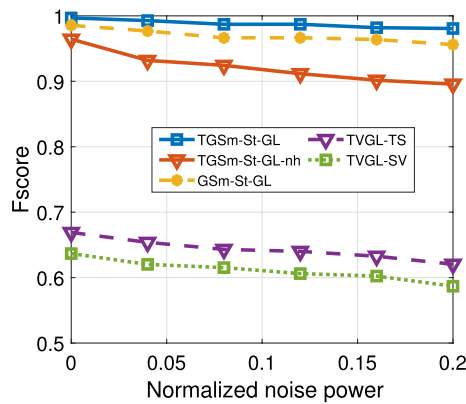


Fig. 2 Fscore of inferred graphs from observed graph signals against the noise level

is because TGSm-St-GL-nh ignores the presence of hidden nodes. As the same time, TVGL-TS and TVGL-SV present the worst performance since that these two methods not only ignore the presence of hidden nodes but also only account for smoothness assumption for the graph signals. TGSm-St-GL outperforms GSm-St-GL because the latter lacks consideration of the temporal relationship between graphs.

5.3.2 Noisy observations

The effect of different noise levels is evaluated in the second experiment. We use TV-ER model with edge probability values of $s = 0.3$ to generate random time-varying graphs and set $H = 2$. Assuming that the ground-truth graph signals $\mathbf{X}_O^{(t)}$ are corrupted by a multivariate Gaussian distribution noise with mean zero and covariance $\sigma^2 \mathbf{I}$, resulting in the observed noise graph signals $\tilde{\mathbf{X}}_O^{(t)}$. As depicted in Fig. 2, the Fscore of the learned graphs is plotted on the y-axis, while the power of noise is represented on the x-axis. Notably, TGSm-St-GL demonstrates superior performance compared to GSm-St-GL. This finding is consistent with the previous experimental results. Besides, compared to TGSm-St-GL, we observe that the performance of TGSm-St-nh, TVGL-TS and TVGL-SV deteriorates significantly with increase in noise power,

further emphasizing the necessity of considering the existence of hidden nodes. Furthermore, the result of TGSm-St-GL in terms of Fscore decays slightly when the noise power increases, demonstrating the proposed method is robust to noise.

5.3.3 Structure properties of $\tilde{\mathbf{P}}^{(t)}$

Although the primary objective of this study is to achieve the recovery of $\{\hat{\mathbf{L}}_O^{(t)}\}_{t=1}^T$, the structure properties of $\{\tilde{\mathbf{P}}^{(t)}\}_{t=1}^T$ make a significantly contribution to our proposed method at the same time. Consequently, illustrate the recovered $\{\hat{\mathbf{L}}_O^{(t)}\}_{t=1}^T$ and $\{\hat{\mathbf{P}}^{(t)}\}_{t=1}^T$ is the purpose of this experiment. In this way, we can gain a clearer understanding of the impact of different methods on graph structure recovery. The outcomes are depicted in Figs. 3 and 4.

In Fig. 3, the first column corresponds to ground-truth graph topology and the corresponding covariance matrix separately. The second column corresponds to the ground-truth values of $\tilde{\mathbf{L}}_O^{(1)}$ and $\tilde{\mathbf{P}}^{(1)}$. The last two columns present the estimates obtained by the group Lasso scheme TGSm-St-GL [cf. (13)] and the low-rank scheme TGSm-St [cf. (12)], respectively. It is apparent that for the depicted example, the low-rank scheme TGSm-St is not capable of recovering the column-sparse structure of the original matrix $\tilde{\mathbf{P}}^{(1)}$. On the contrary, the estimated matrix $\hat{\mathbf{P}}^{(1)}$ in Fig. 3g exhibits similar column sparsity as the ground truth $\tilde{\mathbf{P}}^{(1)}$. Significantly, from the perspective of the estimated $\hat{\mathbf{L}}_O^{(1)}$, it becomes evident that the more precise estimation of $\hat{\mathbf{P}}^{(1)}$ leads to a superior inference of the graph topology. Thus, the group Lasso scheme TGSm-St-GL yielding better estimates than the low-rank scheme TGSm-St.

In Fig. 4, we show that the learned matrices $\hat{\mathbf{P}}^{(t)}$ of time-varying graph, where the value of t range from 1 to 4. It is apparent that the columns with nonzero entries maintain consistent positions across adjacent time slots for the learned matrices

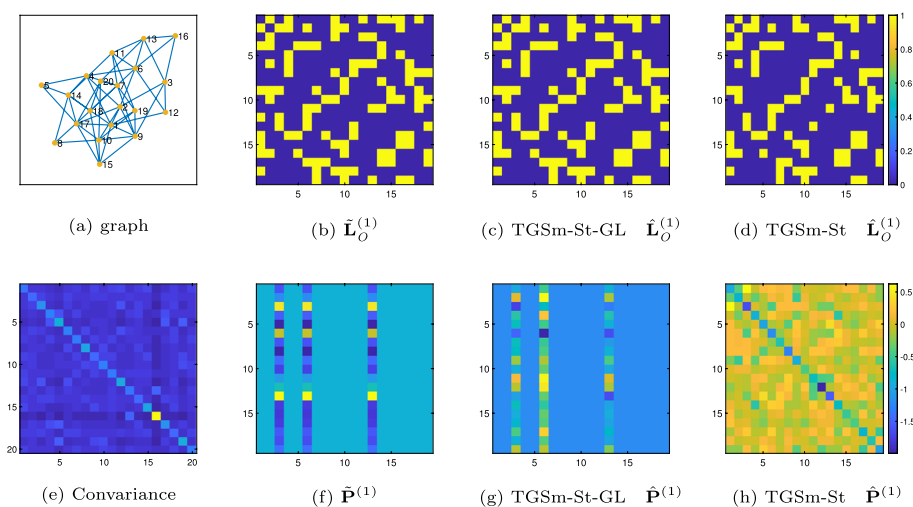
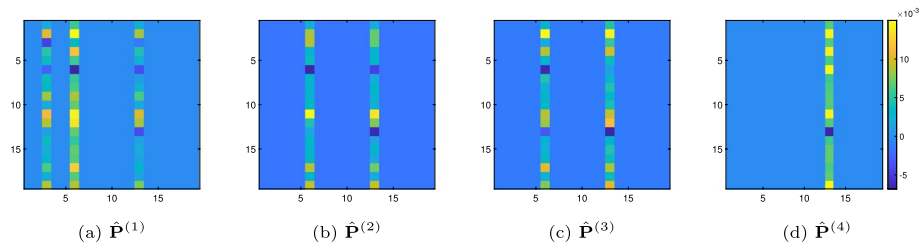


Fig. 3 Graphical representation of the ground-truth graph and the estimated graph at $t=1$ time slot with $N = 20$ and $H = 1$. **a** represents the ground-truth graph and the corresponding covariance matrix in **(e)**. The ground-truth matrices $\tilde{\mathbf{L}}_O^{(1)}$ and $\tilde{\mathbf{P}}^{(1)}$ are represented in the second column [cf. panels **(b)** and **(f)**]. Analogously, the estimates $\hat{\mathbf{L}}_O^{(1)}$ and $\hat{\mathbf{P}}^{(1)}$ generated by (13) are represented in panels **(c)** and **(g)**, those generated by (12) in panels **(d)** and **(h)**

Table 2 The performance achieved by the schemes TGSm-St-GL, GSm-St-GL, TVGL-TS and TVGL-SV while learning time-varying graphs

Datasets	Algorithm	Fscore	Precision	Recall	NMI
PM 2.5	TGSm-St-GL	0.5011	0.5597	0.4605	0.1249
	GSm-St-GL	0.3672	0.2268	0.9721	3.049e−5
	TVGL-TS	0.2268	0.5055	0.1628	0.0409
	TVGL-SV	0.2295	0.5257	0.1488	0.0430
COVID-19	TGSm-St-GL	0.2629	0.1775	0.7890	0.0603
	GSm-St-GL	0.2546	0.1496	0.8923	0.0410
	TVGL-TS	0.0938	0.1494	0.2967	0.0590
	TVGL-SV	0.1577	0.1494	0.5627	0.0290

**Fig. 4** The visualization of the matrices $\hat{\mathbf{P}}^{(t)}$ in time-varying graph learned from the synthetic data based on TGSm-St-GL method

$\hat{\mathbf{P}}^{(t-1)}$. In other words, the scheme TGSm-St-GL captures the similar column-sparsity pattern of $\hat{\mathbf{P}}^{(t)}$ resulted from the temporal similarity of time-varying graph.

5.4 Experiments on real-world data

In this section, we evaluate our algorithms using two real-world data and compare their recovery performance with existing alternatives in the literature.

5.4.1 Application to PM 2.5 data

We start by considering the daily mean PM 2.5 concentration data from California provided by the US Environmental Protection Agency [40]. The dataset contains daily measurements collected from 93 sensors in California over the initial 304 days of 2015. According to the longitude and latitude coordinates of these 93 sensors, we build an initial graph. To infer best-represented time-varying graphs from incomplete graph signals, we make the assumption that only the 15 first sensors are observed. In this case, the goal is to infer the connections between those 15 sensors. Moreover, we divide 304 days into 10 time slots in equal proportion, and thus, each sensor includes data from 30 days, i.e., $\mathbf{X}^{(t)} \in \mathbb{R}^{15 \times 30}$.

The comparative outcomes between the proposed approach and other relevant alternatives are shown in Table 2. We notice that the proposed TGSm-St-GL obtains the higher Fscore 0.5011 and the higher NMI than the other methods.

5.4.2 Application to COVID-19 data

Finally, we use the global COVID-19 dataset provided by the Johns Hopkins University [41]. The dataset contains the cumulative number of COVID-19 cases for each day and each locality between January 22 and April 6, as well as the geographical localization of 259 places including some regions of the world, i.e., overall the dataset has 7 time slots with $N = 259$ and $M = 10$. As we want to take into account the presence of hidden variables, we are going to assume that $\mathcal{O} = 1, \dots, 30$, so that only the 30 first stations are observed, with our goal being inferring the connections between those stations.

The results are listed in Table 2. We observe that the proposed TGSm-St-GL outperforms GSm-St-GL, TVGL-TS and TGSm-St-SV. In particular, TGSm-St-GL and GSm-St-GL have comparable performance for the Fscore. This is not surprising, since the value of time slot T is relatively small. On the other hand, TVGL-TS and TGSm-St-SV get worse performance than TGSm-St-GL. The result once again clearly reflects that the explicit consideration of hidden variables when inferring the graph structure leads to better performance.

6 Conclusion

In this paper, we introduced an optimization framework aimed at addressing the issue of time-varying graph learning with hidden nodes. The framework relied on the assumption that the observed signals are both smooth and stationary on learned graphs and identified graph topologies by leveraging the similarity in time-varying graphs. Specially, the key was to leverage the block structure of matrix to handle the presence of hidden nodes, and an optimization framework based on the graph topologies and the graph signals constraints is proposed. Moreover, in order to capture the characteristics of the learned graphs precisely, we augmented the objective function with a column-sparsity constrain and considered the connection of the similarity of different time slots graphs on column-sparsity. The experimental results from both simulated and real-world data verified the effectiveness and superiority of our method. Our future work includes considering the inference problem of time-varying graphs under different evolutionary modes in the presence of hidden nodes.

Acknowledgements

Not applicable.

Author contributions

Not applicable.

Funding

This work was supported by the Innovation Program for Quantum Science and Technology under Grants No. 2021ZD0300703, the National Natural Science Foundation of China under Grants No. 61971146, and the University-Industry Collaborative Education Program under Grants No. 230805384035416.

Data availability

All data generated or analyzed during this study are included in this published article

Declarations

Competing interests

The author's declared that they have no competing interests.

Received: 23 October 2023 Accepted: 1 March 2024

Published online: 13 March 2024

References

1. E.D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models* (Springer, New York, 2009)
2. O. Sporns, *Discovering the Human Connectome* (MIT Press, Boston, 2012)
3. S. Myers, J. Leskovec, On the convexity of latent social network inference. *NIPS*. **23** (2010)
4. A. Namaki, A. Shirazi, R. Raei, G. Jafari, Network analysis of a financial market based on genuine correlation and threshold method. *Phys. A* **390**(21), 3835–3841 (2011)
5. D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Sign. Process. Mag.* **30**(3), 83–98 (2013). <https://doi.org/10.1109/MSP.2012.2235192>
6. A. Sandryhaila, J.M.F. Moura, Big data analysis with signal processing on graphs: representation and processing of massive data sets with irregular structure. *IEEE Sign. Process. Mag.* **31**(5), 80–90 (2014). <https://doi.org/10.1109/MSP.2014.2329213>
7. A.G. Marques, N. Kiyavash, J.M.F. Moura, D. Van De Ville, R. Willett, Graph signal processing: foundations and emerging directions [from the guest editors]. *IEEE Sign. Process. Mag.* **37**(6), 11–13 (2020). <https://doi.org/10.1109/MSP.2020.3020715>
8. E. Pavez, A. Ortega, Generalized Laplacian precision matrix estimation for graph signal processing. in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6350–6354 (2016). <https://doi.org/10.1109/ICASSP2016.7472899>
9. S. Segarra, A.G. Marques, G. Mateos, A. Ribeiro, Network topology inference from spectral templates. *IEEE Trans. Sign. Inf. Process. Netw.* **3**(3), 467–483 (2017). <https://doi.org/10.1109/TSIPN.2017.2731051>
10. S. Segarra, A.G. Marques, M. Goyal, S. Rey, Network topology inference from input-output diffusion pairs. in *2018 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 508–512 (2018). <https://doi.org/10.1109/SSP2018.8450838>
11. X. Dong, D. Thanou, M. Rabbat, P. Frossard, Learning graphs from data: a signal representation perspective. *IEEE Sign. Process. Mag.* **36**(3), 44–63 (2019). <https://doi.org/10.1109/MSP.2018.2887284>
12. G. Mateos, S. Segarra, A.G. Marques, A. Ribeiro, Connecting the dots: identifying network structure via graph signal processing. *IEEE Sign. Process. Mag.* **36**(3), 16–43 (2019). <https://doi.org/10.1109/MSP.2018.2890143>
13. F. Xia, K. Sun, S. Yu, A. Aziz, L. Wan, S. Pan, H. Liu, Graph learning: a survey. *IEEE Trans. Artif. Intell.* **2**(2), 109–127 (2021). <https://doi.org/10.1109/TAI.2021.3076021>
14. V. Kalofolias, How to learn a graph from smooth signals. in *Proceedings of International Conference on Artificial Intelligence and Statistics*, vol. 51, pp. 920–929 (2016)
15. X. Dong, D. Thanou, P. Frossard, P. Vandergheynst, Learning Laplacian matrix in smooth graph signal representations. *IEEE Trans. Sign. Process.* **64**(23), 6160–6173 (2016). <https://doi.org/10.1109/TSP.2016.2602809>
16. S.P. Chepuri, S. Liu, G. Leus, A.O. Hero, Learning sparse graphs under smoothness prior. in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6508–6512 (2017). <https://doi.org/10.1109/ICASSP.2017.7953410>
17. H.E. Egilmez, E. Pavez, A. Ortega, Graph learning from data under Laplacian and structural constraints. *IEEE J. Sel. Top. Sign. Process.* **11**(6), 825–841 (2017). <https://doi.org/10.1109/JSTSP.2017.2726975>
18. S. Kumar, J. Ying, J.V. Miranda Cardoso, D. Palomar, Structured graph learning via Laplacian spectral constraints. *Adv. Neural Inf. Process. Syst.* **32**, 11647–11658 (2019)
19. D. Thanou, X. Dong, D. Kressner, P. Frossard, Learning heat diffusion graphs. *IEEE Trans. Sign. Inf. Process. Netw.* **3**(3), 484–499 (2017). <https://doi.org/10.1109/TSIPN.2017.2731164>
20. V. Chandrasekaran, P.A. Parrilo, A.S. Willsky, Latent variable graphical model selection via convex optimization. in *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 1610–1613 (2010). <https://doi.org/10.1109/ALLERTON.2010.5707106>
21. A. Chang, T. Yao, G.I. Allen, Graphical models and dynamic latent factors for modeling functional brain connectivity. in *2019 IEEE Data Science Workshop (DSW)*, pp. 57–63 (2019). <https://doi.org/10.1109/DSW.2019.8755783>
22. X. Yang, M. Sheng, Y. Yuan, T.Q.S. Quek, Network topology inference from heterogeneous incomplete graph signals. *IEEE Trans. Sign. Process.* **69**, 314–327 (2021). <https://doi.org/10.1109/TSP.2020.3039880>
23. A. Anandkumar, D. Hsu, S. A. Javanmard, Kakade, Learning linear bayesian networks with latent variables. in *International Conference on Machine Learning*, pp. 249–257 (2013)
24. J. Mei, M.F. Moura, Silvar: single index latent variable models. *IEEE Trans. Sign. Process.* **66**(11), 2790–2803 (2018). <https://doi.org/10.1109/TSP.2018.2818075>
25. A. Buciualea, S. Rey, C. Cabrera, A.G. Marques, Network reconstruction from graph-stationary signals with hidden variables. in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pp. 56–60 (2019). <https://doi.org/10.1109/IEEECONF44664.2019.9048913>
26. A. Buciualea, S. Rey, A.G. Marques, Learning graphs from smooth and graph-stationary signals with hidden variables. *IEEE Trans. Sign. Inf. Process. Netw.* **8**, 273–287 (2022). <https://doi.org/10.1109/TSIPN.2022.3161079>
27. S. Rey, A. Buciualea, M. Navarro, S. Segarra, A.G. Marques, Joint inference of multiple graphs with hidden variables from stationary graph signals. in *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5817–5821 (2022). <https://doi.org/10.1109/ICASSP43922.2022.9747524>
28. A.G. Marques, S. Segarra, G. Leus, A. Ribeiro, Stationary graph processes and spectral estimation. *IEEE Trans. Sign. Process.* **65**(22), 5911–5926 (2017). <https://doi.org/10.1109/TSP.2017.2739099>
29. N. Perraudin, P. Vandergheynst, Stationary signal processing on graphs. *IEEE Trans. Sign. Process.* **65**(13), 3462–3477 (2017). <https://doi.org/10.1109/TSP.2017.2690388>

30. M.G. Preti, T.A. Bolton, D. Van De Ville, The dynamic functional connectome: state-of-the-art and perspectives. *Neuroimage* **160**, 41–54 (2017)
31. Y. Kim, S. Han, S. Choi, D. Hwang, Inference of dynamic networks using time-course data. *Brief. Bioinform.* **15**(2), 212–228 (2014)
32. R.N. Mantegna, Hierarchical structure in financial markets. *Eur. Phys. J. B- Condens. Matter Complex Syst.* **11**(1), 193–197 (1999)
33. V. Kalofolias, A. Loukas, D. Thanou, P. Frossard, Learning time varying graphs. in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2826–2830 (2017). <https://doi.org/10.1109/ICASSP.2017.7952672>
34. K. Yamada, Y. Tanaka, A. Ortega, Time-varying graph learning with constraints on graph temporal variation (2020). Preprint at <https://arxiv.org/abs/2001.03346>
35. D. Hallac, Y. Park, S. Boyd, J. Leskovec, Network inference via the time-varying graphical lasso. in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 205–213 (2017)
36. J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**(3), 432–441 (2008)
37. A. Sandryhaila, J. Moura, Discrete signal processing on graphs. *IEEE Trans. Sign. Process.* **61**(7), 1644–1656 (2013)
38. B. Girault, Stationary graph signals using an isometric graph translation. in *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 1516–1520 (2015)
39. M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1 beta. <http://cvxr.com/cvx> (2013)
40. Air data: Air quality data collected at outdoor monitors across the US. <https://www.epa.gov/outdoor-air-quality-data>
41. E. Dong, H. Du, L. Gardner, An interactive web-based dashboard to track Covid-19 in real time. *Lancet Infect.* **20**(5), 533–534 (2020)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.