Universidad
Rey Juan Carlos

TESIS DOCTORAL

Robust Graph Topology Inference

Autor:

Andrei Buciulea Vlas

Director:

Prof. Dr. Antonio García Marqués

Programa de Doctorado Interuniversitario
en Multimedia y Comunicaciones

Escuela Internacional de Doctorado

2024

Resumen

En los últimos años, hemos presenciado una explosión masiva de datos. Esto se debe principalmente a la proliferación de dispositivos de sensores, al uso extendido de las redes sociales y a la creciente digitalización de nuestras actividades diarias. Al mismo tiempo, conforme los sistemas en red contemporáneos crecen en tamaño e importancia, los datos que generan se vuelven más complejos y diversos. Esto impulsa el rápido desarrollo de nuevos métodos y técnicas para procesar conjuntos de datos que se definen sobre dominios irregulares (no euclidianos). Entre los enfoques innovadores que han surgido para abordar los datos contemporáneos, uno especialmente eficaz y prometedor implica modelar la estructura irregular subyacente mediante un *grafo* y luego interpretar los datos como señales definidas en dicho grafo.

Esta perspectiva basada en grafos ha ganado popularidad rápidamente y ha tenido éxito en diversos campos, como las redes sociales, geográficas, energéticas, de comunicación, financieras o biológicas, entre otras. También ha atraído la atención de investigadores de diversos campos, como la estadística, el aprendizaje automático y el procesamiento de señales. Esta forma de interpretar señales con soporte irregular como señales definidas en grafos, y utilizar la estructura subyacente del grafo para procesarlas, constituye el principio básico del procesado de señales definidas en el grafo, también conocido como *graph signal processing* (GSP), un campo que está en rápido desarrollo. El GSP se enfoca en crear nuevos modelos y algoritmos para el tratamiento de señales definidas en el grafo, a menudo adaptando herramientas clásicas diseñadas originalmente para señales definidas en un soporte regular como el tiempo o el espacio.

El GSP se fundamenta en la idea de que existe una estrecha relación entre las propiedades de las señales y la estructura del grafo en el que se definen. Aprovechar eficazmente esta relación es clave para el éxito de GSP. Una parte importante de la investigación en GSP se dedica a comprender cómo las propiedades algebraicas y espectrales del grafo influyen en las propiedades de las señales definidas en él. El *graph-shift operator* (GSO) juega un papel crucial en este análisis. El GSO es una matriz dispersa que codifica la estructura del grafo, lo que lo convierte en un elemento fundamental dentro del marco de GSP. Por ejemplo, el uso del GSO permite definir diversas herramientas espectrales, como la *transformada de Fourier en grafos*. También facilita la creación de operadores de señales definidas en el grafo más generalizados, como los *sistemas de filtros en grafos*, que pueden expresarse como polinomios del GSO.

Antecedentes

GSP abarca una amplia gama de desafíos asociados con los grafos que enfrentamos en la actualidad, lo que requiere abordar múltiples objetivos y hacer diversas suposiciones para resolver los problemas en cuestión. A pesar de la diversidad de los problemas tratados, el concepto subyacente de aprovechar la interacción entre grafos y señales sigue siendo de gran relevancia. Por ejemplo, la investigación en este campo ha explorado ampliamente problemas como la identificación de filtros de grafos, el muestreo de grafos y la reconstrucción de señales definidas en grafos. Estos esfuerzos ilustran la amplitud de los retos que se están abordando en el campo de GSP. Sin embargo, un

problema distinto pero igualmente fundamental para el enfoque de esta tesis se refiere a la inferencia de topología de grafos, también conocido como *network topology inference* o *graph learning*. A diferencia de los problemas del GSP anteriores, el problema de inferencia de topología de red centra la atención en la estructura del grafo que se considera desconocida. En consecuencia, el objetivo principal gira en torno a inferir la estructura del grafo basándose en un conjunto de observaciones en los nodos. Este enfoque particular resalta la complejidad inherente del aprendizaje de grafos y su relevancia en el contexto más amplio de la investigación sobre GSP.

Para estimar la estructura de una red a partir de datos, es crucial hacer ciertas suposiciones sobre los datos, sobre el propio grafo o las posibles relaciones entre los datos y el grafo. En este contexto, los enfoques existentes en la literatura pueden dividirse en tres grupos distintos. En la primera categoría se engloban los modelos que realizan suposiciones sobre las características del grafo. Muchos modelos populares asumen que los grafos considerados son dispersos, lo que refleja la naturaleza típica de los grafos del mundo real, que tienden a tener conexiones limitadas en lugar de una amplia interconectividad. Además, las distintas metodologías exploran diferentes marcos estructurales para los grafos a la hora de evaluar las posibles conexiones entre nodos. Estos marcos incluyen consideraciones sobre la interconectividad de los nodos, como las estructuras comunitarias (similar a las redes sociales), las redes libres de escala (común en las redes de citas) o los grafos jerárquicos (típicos de las estructuras organizativas), entre otros. La segunda categoría está estrechamente relacionada con realizar suposiciones sobre los datos generados dentro de estas redes. En este caso, los enfoques establecidos se centran en las propiedades de las señales, como correlación, independencia condicionada, características espectrales o *smoothness*, por nombrar algunas. Estas suposiciones tratan de modelar las propiedades de las señales del mundo real y se utilizan para formular modelos matemáticos que pretenden servir para estimar la topología del grafo subyacente. La tercera categoría tiende un puente entre el grafo y las señales considerando las propiedades de difusión de las señales a través del grafo. Para entender la relación entre las señales y la estructura del grafo, algunos enfoques emplean modelos de ecuaciones estructurales dispersas o asumen que las señales son estacionarias en el grafo. Considerar estas características de la señal permite explotar la estructura matemática del problema en cuestión obteniendo la estimación de la topología del grafo deseada.

Además de la modelización matemática de la señal, es esencial considerar varios escenarios del mundo real que surgen en el análisis de redes. Uno de los escenarios comunes implica suponer que las señales definidas en el grafo contienen ruido. Por lo tanto, al inferir la estructura de la red a partir de los datos disponibles, es crucial incorporar esta información para obtener una estimación más precisa del gafo subyacente. En muchos casos tener en cuenta que las señales contienen ruido no es suficiente para obtener una estimación adecuada de la red, ya que los escenarios que se nos presentan en el mundo real son mucho más complejos. Uno de estos escenarios que ocurre con frecuencia en el mundo real es el acceso limitado a los datos de los nodos de una red o la falta de acceso a datos de algunos nodos. Esto es común en redes privadas, donde sólo se puede acceder a la información de los nodos que pertenecen a la red, pero los datos asociados a esos nodos pueden verse afectados por nodos no observados. En estos casos, la estimación del grafo obtenida simplemente al tener en cuenta los datos asociados a los nodos observados podría no representar de manera fiable las conexiones realmente existentes entre los nodos debido a la presencia de nodos ocultos. En estos casos surge la necesidad de diseñar métodos de inferencia de topología que sean capaces de modelar este tipo de escenarios para incorporar esta información adicional y obtener una estimación del grafo más acorde a la realidad. Otro escenario muy común que se nos puede presentar es cuando tenemos acceso a datos de redes similares. Un ejemplo típico son los datos asociados a un mismo grupo de individuos en distintas redes sociales. En esta situación, la opción más común es inferir la estructura de cada uno de esas redes por separado. El problema viene cuando la cantidad de datos asociados a cada una de las redes difiere lo que podría hacer que la estimación de los grafos no sea muy buena si el número de señales es reducido. En estos

casos surge la necesidad de diseñar modelos que estimen la estructura de los grafos subyacentes aprovechando las similitudes entre las redes, y de esta manera mejorar la estimación de los grafos que tengan pocos datos asociados.

Objetivos

El propósito fundamental de esta tesis consiste en abordar diversos desafíos relacionados con la estimación de la topología de redes a partir de conjuntos de datos generados en dichas redes. Para llevar a cabo este objetivo, se pretende emplear herramientas pertenecientes al campo de GSP, con el fin de modelar estos problemas de manera más realista. Posteriormente, se propondrán algoritmos para dar solución a los desafíos identificados. En el ámbito amplio de problemas abordados por GSP, este trabajo de tesis se enfocará específicamente en dos de ellos.

Inferencia de Topología de Red Basada en Propiedades de la Señal (P1). Los enfoques existentes en la literatura a menudo asumen propiedades específicas de las señales de grafos, lo que puede limitar su aplicabilidad a datos del mundo real que presentan unas características más complejas. Para mejorar la precisión de las estimaciones de los grafos, es crucial desarrollar modelos de datos más flexibles que puedan capturar una variedad más amplia de características de las señales.

Inferencia de Topología de Red Basada en Escenarios del Mundo Real (P2). Muchos modelos matemáticos utilizados para inferir la estructura del grafo, se construyen sobre escenarios simplificados del mundo real (escenarios con un solo grafo, observación completa de los nodos de la red, etc.) lo que permite proponer soluciones manejables desde el punto de vista teórico y práctico. Sin embargo, estas simplificaciones pueden no reflejar exactamente las observaciones del mundo real. Por ello surge la necesidad de desarrollar modelos de inferencia de grafos que consideren las complejidades inherentes a diversos escenarios del mundo real, como el acceso limitado a información de nodos específicos o la adquisición de datos en redes con características similares.

El objetivo de esta tesis consiste en enfrentar los desafíos delineados anteriormente, abordándolos desde la perspectiva de GSP. A continuación, se exponen cada uno de los objetivos considerados, detallando los métodos propuestos como solución a los problemas tratados.

(O1) Modelado de las señales definidas en el grafo. El primer objetivo se enfoca en abordar el problema de modelado de señales descrito en **(P1)**. Proponemos un método innovador de aprendizaje de grafos para estimar una red a partir de datos de señales observadas. Las contribuciones clave incluyen: 1) la creación de un enfoque de aprendizaje de grafos asumiendo que los datos observados son Gaussianos y estacionarios en el grafo, 2) la formulación de un problema de optimización conjunta que estima el grafo deseado y mejora la estimación de la matriz de precisión del proceso Gaussiano, y 3) el diseño de un algoritmo eficiente para abordar la optimización no convexa en el problema propuesto además de los resultados de convergencia del algoritmo propuesto a un punto estacionario del problema original. Este enfoque es más versátil que los métodos comparativos, ya que considera tanto la Gaussianidad como la estacionariedad, siendo adecuado para una gama más amplia de escenarios. Ofrece ventajas como mejor rendimiento con menos muestras, mayor robustez frente al ruido y compatibilidad con enfoques estadísticos clásicos.

(O2) Modelado conjunto de las señales y del escenario.
El segundo objetivo combina el modelado de señales con el modelado de escenarios. Aquí, abordamos el desafío de inferir la topología del grafo a partir de señales definidas en el grafo que son *smooth* y estacionarias en presencia de nodos ocultos. Los conceptos de suavidad y estacionariedad se refieren al modelado de señales, mientras que la presencia de nodos ocultos se relaciona con el modelado de escenarios. Los conceptos de modelado de señales se han aplicado

en trabajos anteriores por separado cuando todos los nodos son observados, pero su adaptación a variables ocultas no ha sido explorada. Para cerrar esta brecha, investigamos el impacto de variables ocultas (latentes) al asumir señales definidas en el grafo son *smooth* y estacionarias. A continuación, formulamos el problema de inferencia de topología de la red como una optimización con restricciones, considerando explícitamente tanto el modelado de señales como el de escenarios. Para la parte algorítmica, presentamos un método de factorización de matrices de bloques que aprovecha la dispersión del grafo y las características de baja dimensionalidad que surgen de la presencia de nodos ocultos. Por último, para lidiar con la no convexidad del problema original, presentamos varias formulaciones que incluyen relajaciones convexas para manejar los términos de dispersión y baja dimensionalidad.

(O3) Modelado de escenarios del mundo real. En el tercer objetivo nos centramos en el modelado del escenario fijando el modelo de señal a estacionarias. Abordamos el desafío de aprender múltiples grafos relacionados en presencia de variables ocultas. Presentamos un enfoque novedoso que amplía el aprendizaje conjunto de grafos bajo observaciones que son estacionarias en el grafo. Formulamos un problema de optimización convexa para aprender la topología de múltiples grafos relacionados con variables ocultas. Empleamos un método de regularización de grupo inspirado en Lasso para capturar la similitud entre nodos ocultos y observados, y proporcionamos garantías teóricas para la recuperabilidad de los grafos estimados. Demostramos las ventajas de incorporar el modelado de escenarios en nuestro enfoque mediante comparaciones de rendimiento con las alternativas existentes.

Los tres objetivos abordan el desafío de estimar la estructura de grafos a partir de datos definidos sobre esta estructura, considerando diferentes escenarios. El primer objetivo **(O1)** se enfoca en desarrollar un método más general para estimar la topología del grafo en una variedad más amplia de situaciones del mundo real, donde los métodos actuales pueden tener dificultades debido a la complejidad de los datos o la escasez de muestras disponibles. Por otro lado, **(O2)** y **(O3)** continúan explorando el modelado de señales del grafo en entornos más complejos, donde la información completa de los nodos puede no estar disponible o hay múltiples señales disponibles asociadas a redes similares. Considerar estos escenarios más realistas mejora la calidad de las redes estimadas y las hace más útiles para una variedad de aplicaciones futuras que dependen de la estructura de la red.

Metodología

En la elaboración de este trabajo de tesis, se ha seguido un enfoque metódico orientado a la búsqueda de soluciones óptimas. Para cada problema específico, nos hemos dedicado a construir modelos matemáticos que capturen la complejidad inherente del problema. Posteriormente, traducimos estos modelos matemáticos en formulaciones rigurosas de problemas de optimización. Debido a la complejidad inherente de las tareas tratadas, es común que los problemas de optimización resulten ser no convexos. Por consiguiente, nuestra atención se enfoca en proponer relajaciones convexas y/o emplear algoritmos iterativos para alcanzar soluciones (sub)óptimas. Hemos demostrado de manera matemática la convergencia de los algoritmos iterativos hacia puntos estacionarios cuando resulta pertinente.

Una vez que los algoritmos correspondientes han sido desarrollados, resulta imprescindible llevar a cabo una evaluación numérica de su rendimiento y compararlo con diversas alternativas disponibles. Además de la evaluación utilizando datos sintéticos, es crucial aplicar los algoritmos a conjuntos de datos reales para comprender su potencial en escenarios prácticos. Finalmente, para otorgar una mayor visibilidad al trabajo realizado y difundir los resultados obtenidos, el código desarrollado se ha compartido en repositorios en línea en GitHub.

Resultados

El trabajo llevado a cabo en esta tesis se ha plasmado en la redacción de tres artículos para revistas JCR (uno ya publicado, otro aceptado y otro en proceso de revisión), así como en dos publicaciones en conferencias internacionales. También se ha contribuido en la elaboración de otros cuatro artículos de conferencia. Aunque no se consideren contribuciones explícitas de esta tesis, dos de ellos han influido en el desarrollo de los artículos de revista como trabajo previo, y los otros dos se consideran como trabajo futuro. A continuación, se presenta un resumen breve de las publicaciones. Estas publicaciones están organizadas en relación directa con los objetivos de la tesis mencionados previamente.

Las contribuciones relacionadas con el objetivo **(O1)** incluyen [1] y [2]. En trabajos previos de la literatura [3, 4], se ha abordado el problema de inferir la topología de grafos asumiendo propiedades específicas para las señales definidas en el grafo. En [3], se propuso un método que arroja buenos resultados en la estimación de los grafos en escenarios con un número reducido de muestras. Sin embargo, este método es poco expresivo debido a que asume un modelo muy específico para las señales definidas en el grafo. Por otro lado, [4] propuso un modelo de señal más general, pero requiere más muestras para obtener una estimación precisa del grafo. Nuestra propuesta aprovecha las ventajas de ambas alternativas al ser más general que [3] desde el punto de vista del modelado de señal y más robusta que [4] en la estimación del grafo en escenarios con un número reducido de muestras.

La contribución de [5] está relacionada tanto con **(O1)** como con **(O2)** debido a que tiene en cuenta el modelado de la señal y del escenario considerado. En este trabajo proponemos el modelado de la señal asumiendo estacionariedad y también el modelado del escenario teniendo en cuenta la presencia de nodos ocultos en el grafo. Hemos demostrado que tener en cuenta la presencia de nodos ocultos es crítico para la estimación del grafo en muchos escenarios. A continuación, ampliamos el enfoque de inferencia de topología considerando un modelado de señales más intrínseco y cercano a la realidad y, de nuevo, validando los resultados y mostrando la importancia de considerar tanto la presencia de nodos ocultos como el uso de un enfoque de modelado de señales más complejo.

Por último, las contribuciones de [6] y [7] están relacionadas con **(O3)**. En este trabajo, nos enfocamos en la modelización de escenarios y consideramos un modelo general para las señales asumiendo que son estacionarias en el grafo. Nuestro interés radica en modelar escenarios complejos donde tenemos acceso a datos de varias redes relacionadas y queremos estimar la estructura de estas redes, considerando la presencia de nodos ocultos. Esta configuración es más compleja, ya que busca adaptarse a escenarios del mundo real y requiere abordarse mediante enfoques sofisticados, siendo necesario realizar una serie de suposiciones para obtener una estimación de la red acorde a la realidad. El método propuesto utiliza la similitud entre varias redes y también aprovecha esa similitud para los nodos ocultos de la red lo que añade más estructura al problema proporcionando mejores resultados de estimación de los grafos. Este enfoque ha demostrado ventajas en comparación con los algoritmos existentes desde dos perspectivas: una, al considerar la presencia de nodos ocultos, y la otra, al contemplar múltiples redes similares.

Conclusiones

En este trabajo de tesis, se ha contribuido a establecer los fundamentos de un enfoque sólido para dar solución a problemas clásicos de GSP, considerando (i) la naturaleza de los datos definidos en los nodos de la red y ii) el escenario a tener en cuenta a la hora de estimar la topología del grafo. La primera categoría de problemas está relacionada con el objetivo **(O1)** y parte del objetivo **(O2)**. La segunda categoría está relacionada con el objetivo **(O3)** y parte del objetivo **(O2)**.

En primer lugar, en el capítulo 3 se ha presentado un método de estimación de grafos el cual

tiene en cuenta un modelo de señal más complejo que los considerados hasta la fecha en otros métodos de la literatura. Se ha propuesto una relajación convexa al problema original que no lo era, y además se ha diseñado un algoritmo eficiente que permite utilizar el método propuesto en escenarios que presenten redes con un gran número de nodos. El capítulo 4 se ha considerado un entorno en el cual se ha modelado tanto la señal como el escenario. En el caso se la señal, se ha asumido que las señales son *smooth* y estacionarias en el grafo y que el escenario presenta nodos ocultos en los cuales no tenemos acceso a la información. El método propuesto consiste en inferir el grafo teniendo en cuenta el modelo de señal considerado y además la presencia de nodos ocultos. Al tratar de lidiar con un entorno tan complejo, el problema resultante es no convexo, lo que nos impulsó a proponer una relajación convexa que nos permite resolver el problema de manera iterativa y además garantizamos la convergencia del algoritmo propuesto a un punto estacionario del problema original. Por ultimo en el capítulo 5 se ha presentado un problema de inferencia de topología a partir de señales definidas en redes similares teniendo en cuenta la presencia de variables ocultas. El método propuesto consigue estimar la topología de varios grafos asumiendo que son similares y considerando la presencia de nodos ocultos. También hemos demostrado de manera teórica bajo que condiciones se puede recuperar la topología de los grafos considerados.

Adicionalmente, la eficacia y eficiencia de los métodos implementados ha sido testeada de manera extensiva mediante una serie de experimentos sintéticos y también en escenarios con datos reales. El objetivo de los experimentos ha sido demostrar la importancia a la hora de estimar la topología de la red del uso de modelados más complejos tanto para las señales definidas en los grafos de la red como para el propio modelado de los escenarios considerados para que se acerquen lo máximo posible a la realidad.

Agradecimientos

Antes de adentrarnos en la exposición del contenido de la tesis, quiero expresar mi gratitud por el respaldo financiero[1] que ha hecho posible su realización desde un punto de vista económico. Sin embargo, quiero resaltar que, desde una perspectiva aún más vital, deseo dedicar unas líneas de agradecimiento a las personas que han sido fundamentales en este trayecto.

Ante todo, quiero expresar mi más sincero agradecimiento a mi director de tesis por su paciencia conmigo a lo largo de estos años. Reconozco que no ha sido una tarea fácil, y valoro enormemente el esfuerzo y dedicación que ha invertido para guiarme hasta este punto. Gracias por esas repetidas explicaciones en las que yo decía,"Sí, está todo claro" cuando tú realmente sabías que no era así. Gracias a ti, he interiorizado el valor del uso preciso del lenguaje científico y la importancia de una comprensión profunda de los conceptos para poder transmitirlos eficazmente a otros. Me has enseñado no solo a aprender, sino también a enseñar. En resumen, quiero darte las gracias por enseñarme a hacer las cosas bien. También quiero dar las gracias a Elvin y a Geert por recibirme en sus grupos de investigación durante mi estancia en TU Delft. Agradezco sinceramente la oportunidad que me brindaron para aprender de los mejores y por hacerme sentir parte integral del grupo.

En el ámbito familiar, quiero darles las gracias a las personas más importantes de mi vida, mis padres. Gracias por estar siempre a mi lado, por los sacrificios realizados y por todo el amor incondicional que me habéis ofrecido. Gracias por apoyarme en las decisiones que he tomado hasta ahora y por ayudarme a crecer en todos los aspectos. Y como no, darle las gracias a mi queridísima hermana por apoyarme en los momentos difíciles durante todos estos años.

Por último, pero no menos importante, quería dar las gracias a mis compañeros de doctorado por crear un ambiente de trabajo muy agradable, por las curiosidades científicas en las que todos teníamos razón y nunca nos poníamos de acuerdo y también por los planes realizados fuera del entorno laboral. Siendo más específico, quiero darle las gracias a Samuel por ser siempre tan positivo y por no saber decir que no, a Sergio por "llevarnos a Asturias", a Víctor por sacrificarse para acompañarme en "mis momentos de soledad" y a Óscar por ... por ser Óscar.

[1]Este trabajo ha sido parcialmente financiado por la ayuda EU H2020 Tailor Connectivity (No 952215), Unidad ELLIS de Madrid, URJC (PREDOC20-003) y los proyectos SPGraph (PID2019-105032GB-I00), POLIGRAPH(PID2022-136887NB-I00).

Abstract

With the proliferation of large and diverse systems, modern data is becoming more widespread, complex, and characterized by a non-regular structure. The complexity of data generated in these networks poses a challenge to classical processing methods for extracting meaningful information due to its irregular support. A widely used strategy to deal with data generated on an irregular domain is to employ Graph Signal Processing (GSP), which models network structures using graphs and interprets the data generated by the network as graph signals. However, in many cases, the underlying graph structure is unknown and must be estimated from the data in order to gain valuable insights. Existing methods in the literature estimate graph topology by considering various assumptions about both the data and the graph structure. Unfortunately, these assumptions are often not sufficient to deal with the complex nature of real-world data and different scenarios, such as the presence of hidden nodes or multiple networks. To overcome this limitation, this thesis proposes several approaches that consider more sophisticated assumptions for the network structure and the data generated in these networks. The goal is to improve graph estimation by better capturing/modeling the complexity of the data and addressing specific real-world situations related to network structure.

In the first part of this thesis, we delve into the problem of estimating the network structure from data while considering more elaborate approaches for modeling the networked data. Conventional methods for estimating graphs often assume specific/simple signal models such as sparse (direct or partial) correlations, which may not be suitable for a wide range of scenarios due to actual data complexity. To address this issue, we develop methods capable of handling common real-world situations and improve graph estimation by generalizing the signal modeling assumed in existing approaches from the literature. To solve these problems, which are usually non-convex and computationally expensive, we propose the use of convex relaxations and the implementation of efficient algorithms. We show that these adaptations improve graph estimation compared to existing alternatives, reduce time complexity, and enable algorithmic suitability for network structure estimation, especially in scenarios with large numbers of nodes.

In the second part of the thesis, our focus shifts to more complex situations that are prevalent in real-world scenarios. One of the common situations we address is learning the topology of a graph by assuming that there may be *hidden nodes* whose information is not available. Nevertheless, they may influence the observed nodes and the estimated connections between them. We propose several optimization-based approaches that take advantage of the considered assumptions regarding the nature of the graph data and the presence of hidden nodes. Due to the complexity of the considered scenarios, the resulting problems are often ill-conditioned and non-convex. We address this challenge by proposing convex approximations and, in some cases, efficient algorithms, along with convergence results and theoretical guarantees of graph recovery. In particular, we improve the estimation of the underlying graph by exploiting the inherent structure of the problem formulation that results from the considered assumptions (e.g., promoting sparsity by using the reweighted ℓ_1 norm, accounting for the influence of hidden nodes by using low-rank matrix factorization).

Lastly, we extend our work focusing on scenarios where we learn a graph from data generated

in *multiple* similar *networks* by considering the presence of *hidden nodes* in each network. In this case, we propose mathematical models that estimate the multiple graphs by i) assuming that the signals in the graph are stationary, ii) modeling the influence of hidden nodes, and iii) exploiting the similarities between hidden and observed nodes across networks. The different assumptions are used to propose a mathematical formulation that exploits the structure of the problem and thus improves the accuracy of the estimated graphs. On the other hand, all these considerations lead to highly non-convex formulations, for which we propose relaxed approaches together with theoretical guarantees of graph recovery.

In summary, this thesis emphasizes the importance of well-modeling signals and realistic scenarios for accurate graph estimation. Through theoretical developments and experimental validations using synthetic and real-world datasets, we demonstrate the efficacy of our methods. Our findings highlight the significance of nuanced signal and scenario modeling, showcase methodological strengths and weaknesses, and underscore the potential of our approaches in addressing real-world problems.

Contents

Chapter 1

Introduction

To provide context for the work developed in this thesis, we begin by overviewing the general topic of graph learning and graph-based data models, including the practical implications of dealing with graph data in real-world scenarios. By exploring the intricacies of the field, we aim not only to motivate our work but also to highlight the critical need for advancing practical data models that address the complexities of graph-related applications. As we move forward, the chapter outlines the primary objectives that guide this thesis. We then provide a list of the publications associated with the thesis, highlighting their specific contributions and how they address the challenges identified in previous sections. We conclude the chapter with a brief preview of the chapters to come, providing the reader with a roadmap for the dissertation.

1.1 Motivation and context

In recent years, we have witnessed an explosion of data availability, largely driven by the ubiquitous deployment of sensor devices, the extensive utilization of social media networks, and the relentless digitization of our everyday activities. Simultaneously, with the expansion and significance of contemporary networked systems, the data they generate become increasingly complex and heterogeneous. This drives the rapid evolution of new methods and techniques for processing datasets defined over irregular (non-Euclidean) domains [8–11]. One of the innovative methods that has surfaced to tackle modern data involves modeling the underlying irregular structure using a *graph* and subsequently interpreting the data as signals defined on this graph, often termed as *graph signals*. The approach based on graphs has quickly garnered widespread attention and has proven successful in analyzing data from various domains such as social, geographical, financial, energy, communication, and biological networks, among others [12–14]. Researchers in fields as diverse as statistics, machine learning, and signal processing have also become interested in this type of approach.

The core principle of *graph signal processing* (GSP), an evolving field [15–18], involves interpreting signals with irregular support as graph signals and subsequently utilizing the underlying graph structure to process these signals effectively.. GSP focuses on creating new models and algorithms for handling graph signals, often by extending classical tools originally designed for signals with regular support in time or space. GSP operates on the fundamental premise that

signal attributes are closely related to the underlying structure of the graph in which they reside. Effectively exploiting this relationship is key to the success of GSP. A significant amount of GSP research is devoted to understanding how the algebraic and spectral properties of the graph influence the properties of graph signals. The *graph shift operator* (GSO) plays an important role in this analysis. The GSO is a square matrix with a sparsity pattern that encodes the graph structure, making it a fundamental element within the GSP framework [15, 19]. For example, the use of the GSO allows the definition of various spectral tools such as the *graph Fourier transform* [20–22]. It also facilitates the creation of more generalized graph signal operators such as *graph filters*, which can be expressed as polynomials of GSO [19, 23–25].

Many graph-related challenges fall under the umbrella of GSP, reflecting the diversity of goals and assumptions involved in tackling these problems. Despite their diversity, the underlying concept of exploiting the interplay between graphs and signals remains a constant theme. For example, research has extensively explored problems such as graph filter identification [23, 24, 26, 27] and graph sampling and signal reconstruction [28–35]. These efforts illustrate the wide range of challenges that are being addressed in the field of GSP. However, a different but equally fundamental problem that is central to the focus of this thesis concerns network topology inference, also known as graph learning [4, 36–40]. Unlike previous GSP problems, network topology inference focuses attention on the elusive topology of the graph, which in many relevant applications is not known. Consequently, the first step to putting forth a graph-based data methodology is to identify (infer) the structure of the graph using prior information, nodal observations, or a combination of those. This distinct focus underscores the intricate nature of graph learning and its importance within the broader landscape of GSP research.

In order to infer the topology of a graph from data, it is crucial to make basic assumptions about the data, the graph itself, and the potential relationships between them. In this context, existing approaches in the literature can be divided into three different groups. The first category of assumptions revolves primarily around the characteristics of the graph. Many popular models assume that the graphs under consideration are sparse, reflecting the typical nature of real-world networks, which tend to have limited connections rather than dense interconnectivity. In addition, different methodologies explore various structural frameworks for graphs when evaluating potential connections between nodes. These frameworks include considerations of node interconnectivity such as community structure (similar to social networks), scale-free architecture (common in citation networks), or hierarchical graphs (typical of organizational structures). The second category of assumptions is closely related to the data generated within these networks. Here, established approaches focus on signal properties such as correlation, conditional independence, spectral characteristics, or smoothness, to name a few. These properties serve as basic assumptions and are used to formulate mathematical frameworks that aim to reveal the underlying graph topology. The third category of assumptions bridges the gap between the graph and the signals by considering diffusion properties. To define a model that relates the signal to the graph structure, sparse structural equation models [41] assume that the signal at a node can be explained as a linear combination of the signals at its neighbors. A more general approach, proposed in [4], is to assume that the signals in the graph of interest are stationary so that they can be modeled as the output of a graph filter whose input is zero mean white noise. These assumptions about the relationship between the graph and the signals are used to exploit the mathematical structure of the problem and help to better estimate the desired graph topology.

Beyond the mathematical modeling of the signal, it is critical to consider several real-world scenarios that arise in network analysis. One important factor to consider is the limited access to data from all nodes in a network. Often, we only have access to data from a subset of nodes. In such cases, it becomes imperative to incorporate this additional information into our models to

improve the accuracy of graph estimation. Another scenario arises when we have access to data from similar networks. In this situation, we can use this information to build a model that exploits the similarities between networks, thereby improving the accuracy of network estimation.

With this in mind, this thesis addresses several current challenges that are prevalent in real-world networks and that existing approaches struggle to solve optimally. Here is a list of these problems, for which we will present the proposed solutions in the following sections.

- **Problem 1 (P1). Network topology inference using generalized signal models.** The challenge of this problem is that existing approaches in the literature are often tailored to classical statistical properties of graph signals (such as correlation or conditional independence). While these approaches have theoretical support and perform admirably in scenarios where signals conform to predefined models, real-world data is oftentimes more complicated and may not strictly adhere to these models. Consequently, there is a pressing need to develop more generalized data models that, while preserving some of the statistical rigor, have the flexibility to effectively capture a wider range of signal characteristics. By doing so, we can improve the accuracy of graph estimates and provide more robust insights into the underlying network structure.
- **Problem 2 (P2). Network topology inference beyond single graphs with fully observed nodes.** Many of the existing graph-learning works in the literature ignore the limitations of the datasets at hand so that a simplified mathematical model (leading to more tractable solutions) can be used. However, these simplifications can give rise to graph estimates that do not accurately match observations in the real world. To address this, there is a need to design graph inference models that also consider some of the challenges present in various real-world scenarios. Two that we analyze in this dissertation are: limited (or lack of) access to observations at a subset of nodes, and limited observations coming from related (but not identical) networks. Developing inference models that can account for such real-world nuances is essential for producing more reliable and applicable network topology estimates.

1.2 Objectives

Our goal is to address the challenges outlined in the previous section by approaching them from a robust GSP perspective. Here, we present our objectives for dealing with each problem and elaborate on the proposed approaches designed as solutions:

- **Objective 1 (O1).** This objective tackles the signal modeling problem described in **(P1)**. To achieve this goal, we propose a novel graph-learning method for estimating a network from observed signal data. The key contributions encompass: 1) building a novel graph-learning approach based on the assumption that the observed data is both Gaussian and stationary on the graph, 2) formulating a joint optimization problem that not only estimates the desired graph but also enhances the precision matrix estimation of the Gaussian process, and 3) designing an efficient algorithm with convergence guarantees to address the non-convex optimization in the joint problem. The proposed approach is more versatile than the methods we compare to, as it accounts for both Gaussianity and stationarity, making it suitable for a broader range of scenarios. It offers benefits such as improved performance with fewer samples, enhanced robustness to noise, and compatibility with classical statistical approaches.

- **Objective 2 (O2).** The second objective combines signal modeling with scenario modeling. Here, we address the challenge of network topology inference while assuming smooth and stationary graph signals in the presence of *hidden nodes*. The concepts of smoothness and stationarity pertain to signal modeling, while the presence of hidden nodes relates to scenario modeling. Signal modeling concepts have been successfully applied separately when all nodes are observed, but their adaptation to hidden variables has not been explored. To bridge this gap we investigate the impact of hidden (latent) variables when assuming smooth and stationary graph signals. Next, we formulate the network topology inference problem as a constrained optimization, explicitly considering both signal and scenario modeling. For the algorithmic part, we introduce a block matrix factorization method that exploits the sparsity and low-rank characteristics arising from the presence of hidden nodes. Lastly, in order to deal with the non-convexity of the original problem, we present various formulations which include convex relaxations to handle the sparsity and low-rank terms.

- **Objective 3 (O3).** The third objective focuses on dealing with observability limitations in real-world setups, while keeping the signal modeling restricted to stationary signals. Specifically, we address the challenge of learning multiple related graphs in the presence of hidden variables. Existing methods for joint graph learning either assumed complete observations or focused on learning a single graph with hidden nodes, but none handled the scenario of multiple graphs with hidden variables under graph-stationary observations. To address this, we introduce a new approach that jointly estimates similar graphs under the assumption of stationary observations in the presence of hidden variables. Next, we propose an approach to approximate the proposed nonconvex problem by incorporating convex relaxations. Regarding the algorithmic details, we employ a regularization method inspired by group Lasso to capture the similarity between hidden and observed nodes, promoting graph similarity among all nodes. We also provide theoretical guarantees for the recoverability of the estimated graphs in the presence of hidden nodes. Finally, we compare the performance of our algorithm with existing alternatives, demonstrating the advantages of incorporating scenario modeling into our problem formulation.

All three objectives aim to address the challenge of estimating network structure from available data in different contexts. Objective **(O1)** focuses primarily on understanding the assumptions regarding the relationship between signals and the graph. The goal is to develop a more versatile method capable of estimating graph topology in a wider range of real-world scenarios. This is particularly relevant in situations where existing methods struggle due to complicated data structures or limited available samples, preventing meaningful recovery of network structure. Objectives **(O2)** and **(O3)** extend this focus on graph signal modeling to more complex and realistic environments. These scenarios present challenges such as inaccessible information associated with certain nodes or dealing with multiple node data associated with similar networks. By considering these more complicated settings, the goal is to improve the quality of the estimated networks, making them more applicable to various future tasks that rely on or are influenced by the network structure.

Beyond the main objectives, a goal of this thesis is to fill the gap between theoretical methods proposed in the literature for network topology inference and more challenging real-world scenarios. The intention is to improve network estimation by addressing more general and realistic scenarios in the context of network topology inference problems. Another side objective is to assess the performance of the proposed methods both from a theoretical and practical point of view, to further demonstrate the value of the proposed approaches.

1.3 Summary of contributions

This section provides a compilation of publications together with concise summaries. The publications are structured around the findings and contributions that directly relate to the posed problems and objectives outlined in the preceding section.

- The papers related to **(O1)** are [1] and [2]. Previous works [3, 4] addressed the problem of inferring the network topology by assuming certain properties of the graph signals. In [3], the graph learning scheme performs well for small sample scenarios with the drawback of assuming a more confined signal model. On the other hand, [4] proposes a more general signal model, but requires more samples for an accurate graph estimation. Our proposed approach takes advantage of both alternatives and ends up being almost as general as [4], subsuming [3] as a particular case, and requiring far fewer samples than [4]. The publications related to **(O1)** are listed below.

  [1] A. Buciulea and A. G. Marques, "Graph learning from Gaussian and stationary graph signals," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2023, pp. 1–5. DOI:10.1109/ICASSP49357.2023.10096413.

  [2] A. Buciulea, J. Ying, A. G. Marques, and D. P. Palomar, "Polynomial Graphical Lasso: Learning Edges from Gaussian Graph-Stationary Signals," in IEEE Transactions on Signal Processing (under review, available arXiv preprint arXiv:2404.02621).

- The contribution [5] address both **(O1)** and **(O2)**. In this work, we explored three options for signal modeling: 1) smooth, 2) stationary, and 3) smooth and stationary graph signals. In addition, we considered the presence of hidden nodes in the graph. Then, we proposed mathematical models for each of these scenarios and developed (bi-)convex optimization approaches to estimate the topology of the graph based on the properties of the corresponding signals. The designed approaches were accompanied by theoretical guarantees of convergence to the solution of the original problem. We validated the proposed approaches through extensive experiments where we highlighted the importance of considering both the presence of hidden nodes and more sophisticated signal modeling approaches. The related publication is listed below.

  [5] A. Buciulea, S. Rey, and A. G. Marques, "Learning graphs from smooth and graph-stationary signals with hidden variables," IEEE Transactions on Signal and Information Processing over Networks, vol. 8, pp. 273–287, 2022. DOI:10.1109/TSIPN.2022.3161079.

- Lastly, the contributions from [6] and [7] are related to **(O3)**. In these works, we focused on challenging observations setups while considering graph stationary signals. More specifically, our goal is to address scenarios with a limited number of signals per graph but where data from multiple related networks is available. We seek to estimate the structure of these networks jointly, taking into account the potential presence of hidden nodes. This setup presents a higher level of complexity as it aims to capture what happens in various real-world scenarios, necessitating the use of more advanced approaches and multiple assumptions in order to achieve an accurate network topology estimation. A key ingredient in our approach involves exploiting the similarity between the various networks not only for the observed but also for the hidden nodes. The publications related to **(O3)** are listed below.

[6] S. Rey, M. Navarro, A. Buciulea, S. Segarra, and A. G. Marques, "Joint graph learning from Gaussian observations in the presence of hidden nodes," in Asilomar Conference on Signals, Systems, and Computers. IEEE, 2022, pp. 53-57. DOI: 10.1109/IEEECONF56349.2022.10051977

[7] M. Navarro, S. Rey, A. Buciulea, A. G. Marques, and S. Segarra, "Joint network topology inference in the presence of hidden nodes," IEEE Transactions on Signal Processing (accepted April 2024, available arXiv preprint arXiv:2306.17364.)

While not included explicitly in this thesis, additional works in the context of learning graphs, hypergraphs and simplicial complexes from signals carried out by the author during his PhD include

[42] A. Buciulea, S. Rey, C. Cabrera, and A. G. Marques, "Network reconstruction from graph-stationary signals with hidden variables," in Asilomar Conference on Signals, Systems, and Computers. IEEE, 2019, pp. 56–60. DOI:10.1109/IEEECONF44664.2019. 9048913.

[43] S. Rey, A. Buciulea, M. Navarro, S. Segarra, and A. G. Marques, "Joint inference of multiple graphs with hidden variables from stationary graph signals," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022, pp. 5817–5821. DOI:10.1109/ICASSP43922.2022.9747524.

[44] A. Buciulea, E. Isufi, G. Leus, and A. G. Marques, "Learning Graphs and Simplicial Complexes from Data," in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022, pp. 9861-9865. DOI:10.1109/ICASSP48485.2024.10446559

[45] A. Buciulea, E. Isufi, G. Leus, and A. G. Marques, "Learning the Topology of a Simplicial Complex Using Simplicial Signals: A Greedy Approach," in IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM 2024). (Accepted)

Although these papers are not considered as explicit contributions to this thesis, [42, 43] have influenced the development of journal articles as previous work, while [44, 45] are considered future work.

1.4 Outline of the dissertation

The rest of this document is structured as follows: Chapter 2 presents fundamental definitions and concepts of GSP that will be utilized throughout the thesis. Chapters 3, 4 and 5 address the objectives identified in **(O1)**, **(O2)** and **(O3)**, respectively, encompassing the technical contribution of this work. More specifically, Chapter 3 introduces the problem of learning a graph from noisy nodal observations under the assumption that the graph signals at hand are Gaussian distributed and stationary on the sought graph. Chapter 4 focuses on the case where some of the nodes (and, accordingly, their signals) are never observed and, as a result, they are modelled as hidden nodes. In this scenario, the goal is to learn the graph (links) between the observed nodes under the assumptions of graph smoothness or graph stationarity. Chapter 5 goes one step further and considers the case where multiple graphs need to be learned and hidden nodes are present in all of them. Lastly, Chapter 6 offers concluding remarks and outlines potential avenues for future research.

Chapter 2

Fundamentals: Graph Signal Processing and Graph Learning

This chapter briefly reviews the foundations that motivate the research carried out in this thesis, introducing essential concepts and tools in the field of GSP. To begin with, we outline the different notation conventions employed throughout this document. Next, we elaborate on the basic principles of GSP. Then, we introduce some of the specific concepts and tools of GSP that are of particular relevance to this thesis. Finally, we discuss the problem of learning graphs from nodal observations, providing a brief summary of the contributions of GSP to this specific problem.

2.1 Notation

Along this document, we will refer to scalars, vectors, matrices, and sets using low case letters x, low case bold letters $\mathbf{x}$, upper case bold letters $\mathbf{X}$, and upper case calligraphic letters $\mathcal{X}$, respectively. The notation $\mathbf{I}_M$ denotes the identity matrix of size $M \times M$, while $\mathbf{1}_{M\times N}$ and $\mathbf{0}_{M\times N}$ respectively represent matrices of all ones and zeros of size $M \times N$. We will use $\|\cdot\|_0$, $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_F$, and $\|\cdot\|_\infty$ to denote the ℓ_0, ℓ_1, ℓ_2, ℓ_F (Frobenius), and ℓ_∞ norms operators respectively applied to vectors or matrices. For a matrix $\mathbf{Y} \in \mathbb{R}^{M\times N}$, $\mathrm{vec}(\mathbf{Y}) \in \mathbb{R}^{MN}$ denotes the vertical concatenation of the columns of $\mathbf{Y}$. For a vector $\mathbf{x} \in \mathbb{R}^N$, $\mathrm{diag}(\mathbf{x}) \in \mathbb{R}^{N\times N}$ denotes a square diagonal matrix with the values of $\mathbf{x}$ as its diagonal. For a matrix $\mathbf{X} \in \mathbb{R}^{N\times N}$, $\mathrm{diag}(\mathbf{X}) \in \mathbb{R}^N$ denotes a vector whose values corresponds to the diagonal of $\mathbf{X}$. For a matrix $\mathbf{X} \in \mathbb{R}^{N\times N}$, $\mathrm{tr}(\mathbf{X}) \in \mathbb{R}^N$ denotes the sum of the diagonal values of $\mathbf{X}$. The transpose of a matrix $\mathbf{X}$ is defined as $\mathbf{X}^\top$. The operators $\otimes$,$\odot$,$\circ$, and $\mathbb{E}[\cdot]$ as Kronecker product, Khatri-Rao (column-wise Kronecker) product, Hadamard (entry-wise) product, and expectation respectively. We let calligraphic letters denote index sets, where, given any matrix $\mathbf{X} \in \mathbb{R}^{M\times N}$ and any vector $\mathbf{x} \in \mathbb{R}^N$, we let $\mathbf{X}_{\mathcal{C},:}$ and $\mathbf{X}_{:,\mathcal{C}}$ respectively return the rows and columns of $\mathbf{X}$ selected from index set $\mathcal{C}$ and $\mathbf{x}_\mathcal{C}$ returns the entries of $\mathbf{x}$ selected from $\mathcal{C}$. We let $\mathcal{D}$, $\mathcal{L}$, and $\mathcal{U}$ respectively denote the indices of the diagonal, lower triangular, and upper triangular entries of a vectorized square matrix, i.e., for any matrix $\mathbf{Y} \in \mathbb{R}^{M\times M}$ and $\mathbf{y} = \mathrm{vec}(\mathbf{Y})$, we have that $\mathbf{y}_\mathcal{D}$ contains the diagonal entries of $\mathbf{Y}$. We define $\mathbf{y}_\mathcal{L}$ and $\mathbf{y}_\mathcal{U}$ similarly. The notation $O(\cdot)$ and $o(\cdot)$ denote the usual asymptotic meaning, and we say that $f \asymp g$ if $f = O(g)$ and $g = O(f)$.

2.2 Graphs, GSO, and graph signals

A *graph*, formally denoted as $\mathcal{G} := (\mathcal{V}, \mathcal{E})$, consists of two sets: $\mathcal{V}$, containing nodes or vertices, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, containing edges or links that connect these nodes. Throughout this thesis we will mostly use the terms "nodes" and "edges" but sometimes we may refer to these terms as "vertices" and "links", respectively, which are interchangeable. A graph is typically used to encode pairwise relationships between its nodes, represented by their edges. Nodes within $\mathcal{V}$ are typically labeled with integers, such as $\mathcal{V} := \{1, 2, ..., N\}$, where N represents the total number of nodes in the graph. Edges in $\mathcal{E}$ are characterized by pairs of nodes, (i, j), where both i and j belong to $\mathcal{V}$, and $(i, j) \in \mathcal{E}$ indicates a connection between node i and node j. Regarding the direction of the connection, can be distinguished between directed and undirected graphs. In undirected graphs, there is no classification of the nodes in source or destiny and, in that case, the presence of $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$. In directed graphs, this symmetry does not necessarily hold, because there may be a connection between source and destiny, $(i, j) \in \mathcal{E}$, but not the other way around, $(j, i) \notin \mathcal{E}$. Regarding the strength of the connections between nodes, graphs can be categorized as either unweighted or weighted. Unweighted graphs convey only the presence or absence of edges $(i, j) \in \{0, 1\}$, in this case, information about characteristics such as closeness or similarity can not be extracted accurately. On the other hand, weighted graphs incorporate additional information regarding the distance, intensity of connections, or level of influence between nodes by using non-binary values to represent the edges. When such a relation is some type of similarity, it often holds that $(i, j) \in [0, 1]$. Representation of an unweighted directed graph and weighted undirected graph is shown in Fig 2.1a and b, respectively. This distinction is apparent when examining the adjacency matrix, $\mathbf{A}$, a common representation of the graph topology. The adjacency matrix, $\mathbf{A}$, is a sparse $N \times N$ matrix where $A_{ij} \neq 0$ if and only if $(j, i) \in \mathcal{E}$. For unweighted graphs, $\mathbf{A}$ comprises binary entries ($\mathbf{A} \in \{0, 1\}^{N \times N}$), while in weighted graphs, $\mathbf{A} \in \mathbb{R}^{N \times N}$, with non-zero entries indicating the weights of edges. Once again, when the weights capture some notion of node similarity, it often holds that $\mathbf{A} \in [0, w]^{N \times N}$ where w represents the maximum weight (level of similarity) associated with the edges. For undirected graphs, $\mathbf{A}$ exhibits symmetry while in the case of directed graphs, we typically have that $A_{ij} \neq A_{ji}$. Another fundamental concept related to graph connectivity is the neighborhood of a node. For any node i, its neighborhood, denoted as $\mathcal{N}_i := \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$, includes the nodes connected to i. The degree of a node, $d_i := |\mathcal{N}_i| = [\mathbf{A1}]_i$, denotes the number of neighboring nodes, computed by summing the entries of the i-th row of $\mathbf{A}$.

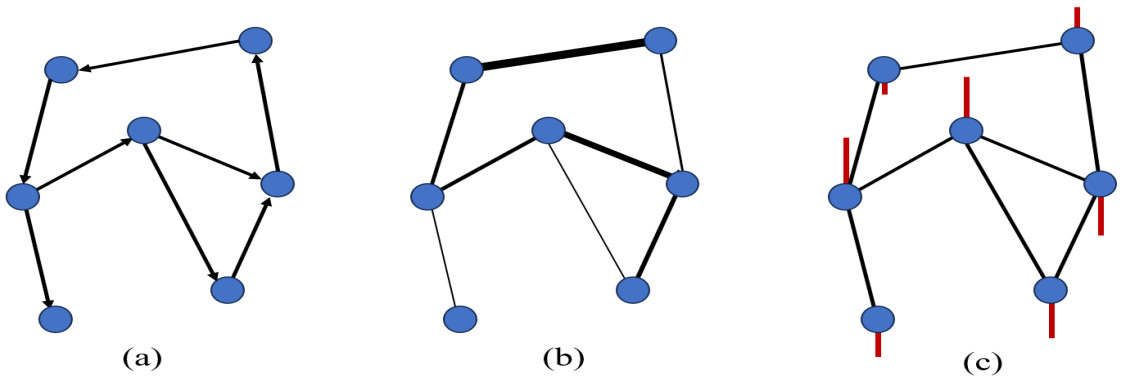

Figure 2.1: Different graph representation. a) Unweighted directed graph, b) weighted undirected graph, and c) unweighted undirected graph with the associated signal on top of each node.

We now proceed to define some elements of the GSP framework. The most prominent actors in GSP are *graph signals*. In essence, a graph signal can be conceptualized as a function that maps the node set onto the real field. This function can be represented as $x : \mathcal{V} \to \mathbb{R}$ or, equivalently, as an N-dimensional real-valued vector $\mathbf{x} = [x_1, ..., x_N]^\top \in \mathbb{R}^N$, where each x_i denotes the value of the signal $\mathbf{x}$ at vertex i (see Fig. 2.1c for an example of graph signal shown in red). For simplicity, we will focus our discussion on scalar, real-valued graph signals. However, it is important to note that in practical applications, node values can be discrete, complex, or even vectors, especially when multiple features per node are observed. Since the graph signal $\mathbf{x}$ operates in the context of $\mathcal{G}$, a fundamental assumption of GSP is that either the values or the properties of $\mathbf{x}$ depend on the topology of $\mathcal{G}$. For example, in a graph representing node similarity, for high values of A_{ij} we would expect the signal x_i and x_j to have similar values. This example illustrates the benefits of considering the topology of the graph when processing graph signals. In the upcoming section 2.4, we will look at different signal models and their relationship to graph topology. In this work, understanding graph signals is essential for modeling different features at each node, such as temperatures recorded at different base stations or stock prices for different companies.

The second key actor in GSP is the *graph shift operator*, GSO [19]. The GSO is a square matrix, denoted as $\mathbf{S} \in \mathbb{R}^{N \times N}$, which plays a dual role. First, it serves as a representation of the underlying graph $\mathcal{G}$ by specifying that its entry S_{ij} can have a non-zero value only if either $i = j$ or $(j, i) \in \mathcal{E}$. Second, the GSO constitutes the simplest graph-aware transformation that can be applied to a graph signal. There are several ways to choose the GSO, with common choices including the adjacency matrix $\mathbf{A}$, the graph combinatorial Laplacian $\mathbf{L} := \text{diag}(\mathbf{A1}) - \mathbf{A}$, and their normalized variations [15, 19]. Using $\mathbf{S}$ as a more abstract representation, rather than committing to a specific GSO choice, proves advantageous as it allows the development of algorithms applicable to a wider range of scenarios. Assuming that $\mathcal{G}$ is undirected, it follows that $\mathbf{S}$ is symmetric and can be diagonalized as $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$. Here, the orthonormal matrix $\mathbf{V} \in \mathbb{R}^{N \times N}$ accumulates the eigenvectors of $\mathbf{S}$, and the diagonal matrix $\mathbf{\Lambda} = \text{diag}(\boldsymbol{\lambda})$ contains the eigenvalues $\boldsymbol{\lambda} \in \mathbb{R}^N$. Conversely, in the case where $\mathcal{G}$ represents a directed graph, we maintain the assumption that $\mathbf{S}$ remains diagonalizable, and its decomposition takes the form $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$. It is important to note that in the directed case, the eigenvalues in $\mathbf{\Lambda}$ tend to be complex numbers. Eigenvectors of the GSO are key to define spectral modes of the graph as well as the graph Fourier transform. Similarly, polynomials of the GSO will be essential to process and model graph signals. This is partially discussed in the next section.

2.3 Graph filters

A versatile tool for modeling the relationship between the signal $\mathbf{x}$ and its underlying graph is the *graph filter*. A graph filter $\mathbf{H} \in \mathbb{R}^{N \times N}$ is a graph-aware linear graph-signal operator that is defined as a polynomial of the GSO $\mathbf{S}$ of the form

$$\mathbf{H} = h_0\mathbf{S}^0 + h_1\mathbf{S}^1 + h_2\mathbf{S}^2 + ... + h_l\mathbf{S}^l = \sum_{l=0}^{L-1} h_l\mathbf{S}^l = \sum_{l=0}^{L-1} h_l\mathbf{V}\mathbf{\Lambda}^l\mathbf{V}^\top = \mathbf{V}\Big(\sum_{l=0}^{L-1} h_l\mathbf{\Lambda}^l\Big)\mathbf{V}^\top, \quad (2.1)$$

where $L-1$ is the filter degree, $\{h_l\}_{l=0}^{L-1}$ are the filter coefficients, and $\mathbf{V}$ and $\mathbf{\Lambda}$ are the eigenvectors and eigenvalues of the GSO, respectively. Since $\mathbf{H}$ is a polynomial of $\mathbf{S}$, it easily follows that both matrices have the same eigenvectors $\mathbf{V}$. Interestingly, it also holds that under the model in (2.1), the product of the graph filter and the GSO commute, i.e., that $\mathbf{HS} = \mathbf{SH}$. This property will be extremely useful in some of our approaches presented in the following chapters. From a graph signal perspective, graph filters can be used to spread an input graph signal $\mathbf{x}$ over a particular

graph represented by $\mathbf{S}$, obtaining as a result $\mathbf{y} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l \mathbf{x} = \mathbf{H}\mathbf{x}$. Then $\mathbf{y}$ can be understood as the diffusion of the graph signal $\mathbf{x}$ over $L-1$ neighborhoods with the associated coefficients h_l modeling (capturing) the importance of the signal from the lth neighborhood.

2.4 Models for graph signals

There exists a variety of (either deterministic or statistical) models that capture different relationships between the graph signals and their underlying graph. In this section, we introduce some commonly used models for graph signals, which will be relevant in the following chapters.

Gaussian graph signals. A Gaussian graph signal is an N-dimensional *random* vector $\mathbf{x}$ whose (vertex-indexed) entries are jointly Gaussian. Thus, if $\mathbf{x} \in \mathbb{R}^N$ is distributed as $\mathcal{N}(\mathbf{0}, \mathbf{C})$, we have that both the covariance matrix $\mathbf{C}$ and the precision matrix $\mathbf{\Theta}$ are $N \times N$ matrices whose entries capture statistical relationships between the node values. In particular, using the Gaussian distribution, we have that

$$f_{\mathbf{\Theta}}(\mathbf{x}) = (2\pi)^{-N/2} \cdot \det(\mathbf{\Theta})^{\frac{1}{2}} \cdot e^{-\frac{1}{2}\mathbf{x}^T\mathbf{\Theta}\mathbf{x}} = (2\pi)^{-N/2} \cdot \det(\mathbf{\Theta})^{\frac{1}{2}} \cdot e^{-\frac{1}{2}\sum_{i=1}^{N}\Theta_{ij}x_i x_j}, \tag{2.2}$$

demonstrating that, in the context of graph signals, the entries of $\mathbf{\Theta} \in \mathbb{R}^{N\times N}$ account for conditional dependence relationships between the (features of the) nodes in the graph [40].

In the case where we have a collection of R Gaussian signals $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_R] \in \mathbb{R}^{N\times R}$ each of them independently drawn from the distribution in (2.2). The log-likelihood associated with $\mathbf{X}$ is

$$L(\mathbf{X}|\mathbf{\Theta}) = \prod_{r=1}^{R} f_{\mathbf{\Theta}}(\mathbf{x}_r), \quad \mathcal{L}(\mathbf{X}|\mathbf{\Theta}) = \sum_{r=1}^{R} \log(f_{\mathbf{\Theta}}(\mathbf{x}_r)). \tag{2.3}$$

This expression will be exploited when formulating our proposed inference approach and establishing links with classical methods in the upcoming chapter.

Smooth graph signals. A graph signal exhibits smoothness on $\mathcal{G}$ if the signal values at connected nodes are relatively close, meaning that the difference between the signal values at neighboring nodes is small. To quantify the smoothness of a graph signal, a common approach is to use the quadratic form

$$\sum_{(i,j)\in\mathcal{E}} A_{ij}(x_i - x_j)^2 = \frac{1}{2}\text{tr}(\mathbf{A}\mathbf{Z}) = \mathbf{x}^\top\mathbf{L}\mathbf{x}, \tag{2.4}$$

where $\mathbf{L}$ is the Laplacian matrix and $\mathbf{Z}$ is the pairwise distance matrix defined as $Z_{ij} = \|x_i - x_j\|^2$ [36]. The middle expression in (2.4) measures how much the signal $\mathbf{x}$ changes with respect to the similarity encoded in the edge weights of $\mathbf{A}$. The right-most expression in (2.4) is often referred to as the local variation (LV) of $\mathbf{x}$. If the goal is to compute the mean LV of R graph signals stored in the $N \times R$ matrix $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_R]$, it can be computed as

$$\frac{1}{R}\sum_{r=1}^{R} \mathbf{x}_r^\top\mathbf{L}\mathbf{x}_r = \frac{1}{R}\sum_{r=1}^{R} \text{tr}(\mathbf{x}_r\mathbf{x}_r^\top\mathbf{L}) = \text{tr}(\hat{\mathbf{C}}_{\mathbf{x}}\mathbf{L}), \tag{2.5}$$

where $\hat{\mathbf{C}}_{\mathbf{x}} := \frac{1}{R}\sum_{r=1}^{R}\mathbf{x}_r\mathbf{x}_r^\top = \frac{1}{R}\mathbf{X}\mathbf{X}^\top$ denotes the sample estimate of the covariance of $\mathbf{X}$. More advanced notions of smoothness can also be defined by considering $\|\mathbf{x} - \mathbf{H}\mathbf{x}\|_2^2$, where $\mathbf{H}$ represents

a predefined low-pass graph filter, and its filter taps or frequency response can be tailored to match the desired notion of smoothness.

Stationary graph signals. The concept of graph stationarity relates the statistical properties of random graph signals to the underlying graph structure. More formally, a zero mean random graph signal is said to be stationary on $\mathcal{G}$ if its covariance matrix $\mathbf{C}_\mathbf{x} = \mathbb{E}[\mathbf{x}\mathbf{x}^\top]$ can be expressed as positive semidefinite (PSD) polynomial of the GSO $\mathbf{S}$ [46]. In cases where $\mathcal{G}$ is an undirected graph, i.e. $\mathbf{V}\mathbf{V}^\top = \mathbf{I}$, a common example of stationary graph signals arises when $\mathbf{x}$ is generated by a linear graph diffusion process as $\mathbf{x} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l \mathbf{w}$, whose input is a zero-mean white signal $\mathbf{w} \in \mathbb{R}^N$. In such a scenario, the covariance of $\mathbf{w}$ is given by $\mathbb{E}[\mathbf{w}\mathbf{w}^\top] = \mathbf{I}$, and $\mathbf{x}$ is related to $\mathbf{w}$ as $\mathbf{x} = \mathbf{H}\mathbf{w}$. In this particular case, the covariance of $\mathbf{x}$ can be expressed as follows

$$\mathbf{C}_\mathbf{x} = \mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \mathbf{H}\mathbb{E}[\mathbf{w}\mathbf{w}^\top]\mathbf{H}^\top = \mathbf{H}\mathbf{H}^\top = \mathbf{H}^2. \tag{2.6}$$

Since the graph filter $\mathbf{H}$ is inherently a polynomial function of the GSO $\mathbf{S}$, it follows that $\mathbf{C}_\mathbf{x}$ is also a polynomial of $\mathbf{S}$. Consequently, in the spectral domain, both $\mathbf{S}$ and $\mathbf{C}_\mathbf{x}$ share the same eigenvectors, and the matrices $\mathbf{S}$ and $\mathbf{C}_\mathbf{x}$ commute, i.e., $\mathbf{C}_\mathbf{x}\mathbf{S} = \mathbf{S}\mathbf{C}_\mathbf{x}$. It is important to note that graph stationarity does not impose a deterministic condition on $\mathbf{x}$ itself, but rather a condition on the covariance structure of the signal. The commutativity expression between $\mathbf{C}_\mathbf{x}$ and $\mathbf{S}$ is a compact and tractable way to account for the graph stationarity of the observed signals and will be used later as a constraint in several of the proposed optimization problems.

2.5 Network topology inference

Network topology inference task poses a significant challenge in network science. In many GSP endeavors, it is assumed that the underlying network is already known. This assumption holds well for certain applications, such as directly observable social and infrastructure networks. However, beyond these scenarios, the methods for constructing graphs are often informal and lack validation. Statistically based network topology inference methods try to close this gap. The goal of network topology inference problems is to estimate the graph structure from node observations. To obtain an accurate and meaningful estimate of the graph topology, assumptions must be made about the signals, the graph, and the relationship between the graph and the signals. Depending on the task and the nature of the available data, several approaches to graph topology estimation have been proposed.

Before we get into the details of the different approaches to tackle the task of estimating the structure of a graph, we first provide a more formal definition of the network topology inference problem. Suppose we have access to a set of R observations associated with the N nodes of a network (graph), and that we collect those observations in the matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_R] \in \mathbb{R}^{N \times R}$. These signals are typically treated as independent realizations of a random network process. Then the goal of the network topology inference problem is to find the optimal graph descriptor $\mathbf{S}$ that best explains the node observations based on the assumed relationship between $\mathbf{X}$ and $\mathcal{G}$. An example is given in Fig. 2.2, where we show a general representation of the network topology inference problem. In this case, finding meaningful connections between nodes depends on the nature of the observations, the relationship between the data and the network, and also on the structure of the network itself (whether it is highly connected or not, whether there are communities, etc.). It is worth mentioning that the estimated graph can be different depending on the assumptions/priors we impose to get a meaningful graph estimation from the set of available observations.

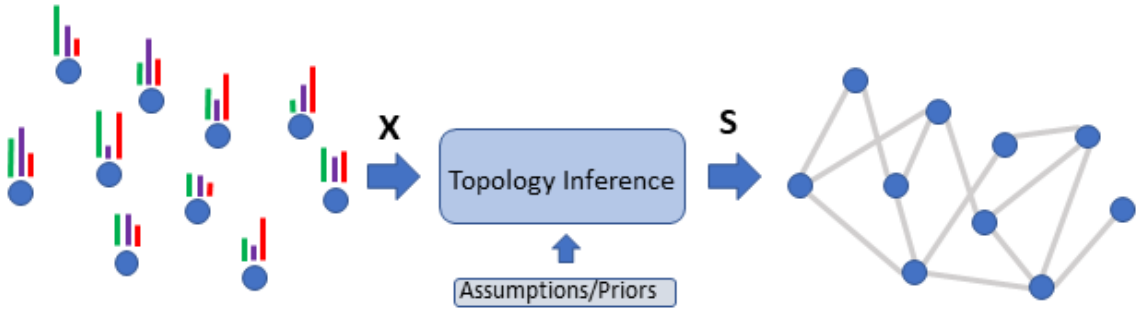

Figure 2.2: General representation of the network topology inference problem. Given a set of nodal observations $\mathbf{X}$ supported on the unknown graph $\mathcal{G}$, find the optimal graph descriptor $\mathbf{S}$ under certain assumptions/relationship between $\mathbf{X}$ and $\mathcal{G}$.

Next, we provide a summary of several network topology inference methods, along with the assumptions they consider for $\mathbf{X}$ and $\mathcal{G}$.

- **Correlation Networks.** Early approaches are based on correlation networks [13]. The node signals are assumed to be random vectors, and the connectivity between nodes is measured in terms of the correlations between the signals at these nodes. In these approaches, the graph is inferred from the Pearson correlations between the nodal signals

$$\rho_{ij} := \frac{\text{cov}(x_i, x_j)}{\sqrt{\text{var}(x_i)\text{var}(x_j)}}. \tag{2.7}$$

  where x_i and x_j denote the nodal random variables at nodes i and j, respectively. In this scenario, the network topology inference problem translates into finding the subset of *nonzero correlations* between the node signals. When the distributions are unknown, and only the signals in $\mathbf{X}$ are available, the Pearson correlation coefficient ρ_{ij} is estimated from the inner products between the rows of $\mathbf{X}$. Therefore, the GSO for correlation networks is often set as the thresholded version of $\hat{\mathbf{C}}_{\mathbf{x}}$, whose nonzero entries identify the support of the graph.

- **Partial Correlation Networks.** An alternative rigorous statistical-based approach to unveil connections between nodes involves considering the influence of third-party nodes. Specifically, there are scenarios where the correlation between two random variables x_i and x_j associated with nodes i and j may be due to their connection with a third node k, rather than a direct link between i and j. To account for such instances of high correlation attributed to latent network effects, partial correlation coefficients are introduced to assess conditional independence between nodes [13]. In particular, we define the partial correlation coefficient as

$$\rho_{ij|\mathcal{V}\setminus ij} := \frac{\text{cov}(x_i, x_j|\mathcal{V}\setminus ij)}{\sqrt{\text{var}(x_i|\mathcal{V}\setminus ij)\text{var}(x_j|\mathcal{V}\setminus ij)}}, \tag{2.8}$$

  where the set $\mathcal{V}\setminus ij$ represents all $N-2$ random variables, excluding those indexed by nodes i and j. Subsequently, a partial correlation network can be derived in a manner analogous to its unconditional correlation network counterpart. The difference being that, in this case, the edge set is defined as $\mathcal{E} := \{(i,j) \in \mathcal{V}\times\mathcal{V} : \rho_{ij|\mathcal{V}\setminus ij} \neq 0\}$. As before, the distributions of the nodal variables are typically unknown and, as a result, one should estimate the partial correlation coefficients from $\mathbf{X}$. This is carried out in two steps, where we first the sample precision matrix is estimated as $\hat{\mathbf{\Theta}}_{\mathbf{x}} = (\hat{\mathbf{C}}_{\mathbf{x}})^{-1}$ and, then, the support of

the graph is estimated by identifying the entries of $\hat{\mathbf{\Theta}}_{\mathbf{x}}$ that exceed a given threshold. The main drawback of this two-step approach is that a large number R of observations is required to guarantee that the inverse is stable. This limitation is addressed by the next method.

- **Graphical Lasso.** A specific instance of the previous setup that is particularly relevant is the scenario where each column of $\mathbf{X}$ is independently sampled from a *Gaussian* distribution. The resulting partial correlation network is commonly known as a Gaussian Markov random field (GMRF) or Gaussian graphical model [47]. The core idea is that the partial correlation coefficients $\mathcal{V} \setminus ij$, which describe the topology of the graph, can be represented as the normalized entries of the precision matrix $\mathbf{\Theta}_{\mathbf{x}} = (\hat{\mathbf{C}}_{\mathbf{x}})^{-1}$. Then, by considering that some of those entries are zero, it readily follows from the Gaussian distribution in (2.2) that the associated nodal variables are conditionally independent [13]. While this still entails that the graph can be inferred from the inverse of the covariance, the expressions in (2.2) and (2.3) can be used to reduce the number of required observations. Exploiting this feature, the well-known graphical Lasso algorithm (GL) [3], which estimates the GSO $\mathbf{S}$ by regularized maximum likelihood estimation, captures the graph structure in the inverse covariance matrix, $\mathbf{S} = \mathbf{\Theta}$, as follows

$$\hat{\mathbf{S}} = \underset{\mathbf{S} \succeq 0}{\operatorname{argmax}} \;\; \log\det(\mathbf{S}) - \mathsf{tr}(\hat{\mathbf{C}}_{\mathbf{x}}\mathbf{S}) - \lambda\|\mathbf{S}\|_1, \tag{2.9}$$

  where the term $\|\mathbf{S}\|_1$ is used to promote sparsity in the estimated precision matrix. Then the recovered graph topology can be seen as a sparse version of the precision matrix which accounts for the indirect relationship between nodes.

- **Network inference from smooth signals.** From another perspective, connectivity between nodes is sometimes measured in terms of the difference in signal values between nodes. For these scenarios, two nodes are considered connected if the difference between their signal values is small. Signals adhering to this model are called smooth on $\mathcal{G}$, and the smoothness level of the signals (how much the signal $\mathbf{x}$ changes with respect to the similarity encoded in the edge weights of the graph representation matrix) is typically measured using the LV metric defined in (2.4) and (2.5). Regarding the problem of estimating a graph from smooth signals, this problem was first addressed in [38] using the following graph learning formulation

$$\begin{aligned} \hat{\mathbf{L}} = \underset{\mathbf{L} \in \mathcal{L}}{\operatorname{argmin}} \;\; & \mathsf{tr}(\mathbf{X}^\top \mathbf{L} \mathbf{X}) + \alpha\|\mathbf{L}\|_F^2, \\ \text{s. t.} \quad & \mathsf{tr}(\mathbf{L}) = s, \end{aligned} \tag{2.10}$$

  where $\mathcal{L}$ represents the set of constraints of a valid graph Laplacian matrix, $s > 0$ is used to control the scale of $\mathbf{L}$ and $\alpha > 0$ controls the sparsity of the estimated graph. The main idea of this formulation is to find a sparse graph whose connections ensure small values of LV for the set of the available graph signals $\mathbf{X}$. Note that the ℓ_1 norm is not penalized in this formulation because it is redundant with the fact of the rows of the Laplacian having zero sum [36].

- **Network inference from stationary signals.** Lastly, there are alternative approaches for estimating the graph topology by establishing a more flexible relationship between the GSO and the observed signals. The one discussed here assumes that the signals are stationary within the sought graph [48–50]. Practically speaking, stationary random graph signals provide an appropriate representation for consensus dynamics, heat diffusion processes, and network processes that occur in structural brain networks [51–53]. From a theoretical point of view, graph stationarity embodies a less restrictive assumption that includes correlation and partial correlation networks as specific instances. Recall from the definition of stationary graph signals in Section 2.4 that the covariance matrix of a graph stationary signal can be

written as a polynomial of the GSO, $\mathbf{C} = (\sum_{l=0}^{L-1} h_l \mathbf{S}^l)^2$. Hence, correlation networks, where $\mathbf{C} = \mathbf{S}$, and GMRFs, where $\mathbf{C} = \mathbf{S}^{-1}$, are very particular cases of polynomials. One possible formulation for estimating a graph from stationary graph signals is [54]:

$$\begin{aligned} \hat{\mathbf{S}} = & \underset{\mathbf{S}\in\mathcal{S}}{\operatorname{argmin}} \ \|\mathbf{S}\|_1 \\ & \text{s. t.} \qquad \|\mathbf{S}\hat{\mathbf{C}}_{\mathbf{x}} - \hat{\mathbf{C}}_{\mathbf{x}}\mathbf{S}\|_F^2 \leq \epsilon, \end{aligned} \tag{2.11}$$

where $\hat{\mathbf{C}}_{\mathbf{x}}$ denotes the sample covariance matrix obtained from the (graph stationary signals in $\mathbf{X}$, and $\epsilon > 0$ ensures commutativity between $\hat{\mathbf{C}}_{\mathbf{x}}$ and $\mathbf{S}$, depending on the accuracy of the estimate $\hat{\mathbf{C}}_{\mathbf{x}}$. The value of ϵ decreases as the approximation of the sample covariance matrix becomes more refined, and approaches $\epsilon = 0$ when the perfect covariance matrix is known. This formulation gives rise to an optimization problem that seeks to identify the sparsest GSO that commutes with $\hat{\mathbf{C}}_{\mathbf{x}}$. The set of valid GSOs encapsulated in $\mathcal{S}$ can be of a specific class, such as adjacency or Laplacian matrices among others. Specifically, the constraints associated with the set of valid adjacency matrices is

$$\mathcal{A} := \{A_{ij} \geq 0;\ \mathbf{A} = \mathbf{A}^\top;\ A_{ii} = 0;\ \mathbf{A}\mathbf{1} \geq \mathbf{1}\}, \tag{2.12}$$

where the GSO is restricted to having positive weights, being symmetric, and not having loops. The role of the last constraint is avoiding the trivial solution $\mathbf{A} = \mathbf{0}_{N\times N}$. The counterpart for the case of a combinatorial Laplacian matrix is

$$\mathcal{L} := \{L_{ij} \leq 0 \text{ for } i \neq j;\ \mathbf{L} = \mathbf{L}^\top;\ \mathbf{L}\mathbf{1} = \mathbf{0};\ \mathbf{L} \succeq 0\}, \tag{2.13}$$

where the valid GSO has off-diagonal negative weights, is a symmetric and PSD matrix, and the sum of the entries of each row is set to be zero.

As a closing comment, we note that, up to this point, this section has primarily addressed a network topology inference scenario known in network science as the "network association" problem [13, Ch. 7.3.1]. While network association is the most widely studied approach in graph learning, two related variants must be mentioned: the link prediction problem and the network tomography problem. "Link prediction" is a simpler task where some of the graph edges are observed in addition to the signals. This additional data can be integrated into the existing framework by adjusting the constraint set $\mathcal{S}$. Conversely, "network tomography" presents a more challenging setup where the available observations are limited to a subset of the nodes. In particular, the focus of chapter 4 is the development of robust algorithms that address the latter problem by exploiting various assumptions from graph signal processing.

Chapter 3

Graph Learning from Gaussian and Stationary Graph Signals

This chapter explores the widespread presence of data defined over non-Euclidean supports, with a focus on the utilization of graphs as a versatile tool to represent these irregular domains. In many scenarios, the graph structure is undefined either due to the absence of a physical network or to the absence of a unique metric to measure relationships between nodes. The main challenge addressed in this chapter is the estimation of the graph topology by modeling the data as graph signals under several assumptions. We propose a new graph-learning method focused on the assumption that the observations are both Gaussian and stationary in the graph. Then, we formulate a joint optimization problem to estimate the graph and the precision matrix of the Gaussian process, presenting an efficient algorithm for this non-convex optimization. Compared to existing methods, the proposed approach offers a more general solution, making it suitable for a broader range of scenarios and outperforming or matching the results achieved by convex counterparts in numerical experiments.

3.1 Introduction

Modern datasets often exhibit irregular non-Euclidean support. In such scenarios, graphs have emerged as a pivotal tool, facilitating the generalization of classical information processing and structured learning techniques to irregular domains. Today, there is a wide range of applications that leverage graphs when processing, learning, and extracting knowledge from their associated datasets (see, e.g., problems in the context of electrical, communication, social, geographic, financial, genetic, and brain networks [13, 14, 55–57], to name a few). When using graphs to process structured non-Euclidean data, it is usually assumed that the underlying network topology is known. Unfortunately, this is not always the case. In many cases, the structure of the graph is not well defined, either because there is no underlying physical network or because the (best) metric to assess the level of association between the nodes is not known.

Since in most cases, the existing relationships are not known beforehand, the standard approach is to infer the structure of the network from a set of available nodal observations/signals/features. To estimate the interactions between the existing nodes, the first step is to formally define the relationship between the topology of the graph and the properties of the signals defined on top of it. Early graph topology inference methods [40, 58] adopted a statistical approach, such as

the correlation network [13, Ch. 7.3.1], partial correlations, or Gaussian Markov random fields (GMRF), with the latter leading to the celebrated graphical Lasso (GL) scheme [13, 59]. Partial correlation methods have been generalized to nonlinear settings [60]. Also in the nonlinear realm, less rigorous approaches simply postulate a similarity scores, with links being drawn if their score exceeds a given threshold. In recent years, GSP-based models [4, 61, 62] have brought new ideas to the field, considering more complex relationships between the signals and the sought graph. These approaches have been generalized to deal with more complex scenarios that often arise in practice, such as the presence of hidden variables [5, 63, 64] or the simultaneous inference of multiple networks [65, 66].

The existing graph-learning methods exhibit different pros and cons, with relevant tradeoffs including computational complexity, expressiveness, model accuracy, or sample complexity, to name a few. For instance, correlation networks need very few samples and can be run in parallel for each pair of nodes, but fail to capture intermediation nodal effects. On the other hand, GL (a maximum likelihood estimator for GMRF) can handle the intermediation effect while still requiring a relatively small number of samples compared to the size of the network. Some disadvantages of GL include assuming a relatively simple signal model (failing to deal with, e.g., linear autoregressive network models) and forcing the learned graph to be a positive definite matrix. To overcome some of these issues, [4] proposed a more general model that assumed that the signals were stationary in the network (GSR) or, equivalently, that the covariance matrix can be defined as a polynomial of the adjacency of the graph [46]. Since GSR is a more general model, it is less restrictive in terms of the signals it can handle. However, it has the disadvantage of requiring a significantly larger number of observations than GL [4].

Our proposal is to combine the advantages of assuming Gaussianity, which implies solving a maximum likelihood problem that requires fewer node observations, with the larger generality of graph-stationary approaches. Our ultimate goal is to generalize the range of scenarios where GL can be used, while keeping the number of observations and computational complexity under control. To be more precise, we introduce Polynomial Graphical Lasso (PGL), a new scheme to learn graphs from signals that works under the assumption that the samples are Gaussian and graph stationary, so that the covariance (precision) matrix of the observations can be written as a polynomial of a sparse graph. These assumptions give rise to a constrained log-likelihood minimization that is jointly optimized over the precision and adjacency matrices, with GL being a particular instance of our problem. The price to pay is that the postulated optimization, even after relaxing the sparsity constraints, is more challenging, leading to a biconvex problem. To mitigate this issue we provide an efficient alternating algorithm with convergence guarantees.

Contributions. To summarize, our main contributions are:

- Introducing PGL, a novel graph-learning scheme that, by assuming that the observations are Gaussian and graph-stationary, generalizes GL and is able to learn a meaningful graph structure in scenarios where the precision/covariance matrices are polynomials of the sparse matrix that represents the graph.
- Formulating the inference problem as a biconvex-constrained optimization, with the variables to optimize being the precision matrix and the graph. While our focus is on learning the graphs, note that this implies that PGL can also be used in the context of covariance estimation.
- Developing an efficient algorithm to solve the proposed optimization, together with conver-

gence guarantees to a stationary point (block coordinatewise minimizer).

- Evaluating the performance of the proposed approach through comparisons with alternatives from the literature on synthetic and real-world datasets.

Outline. The remaining of this chapter is organized as follows. In Section 3.2 the problem of learning (inferring) graphs from signals under different assumptions on the observations is formally stated. Section 3.3 presents a computationally tractable relaxation of the graph-learning problem, along with an efficient algorithm and its associated convergence guarantees. Section 3.4 quantifies and compares the recovery performance of the proposed approach with other methods from the literature using both synthetic and real-data simulations. Section 3.5 closes the chapter with concluding remarks. Additional details regarding the theoretical results are provided in Sections 3.6, 3.7, and 3.8.

3.2 Graph learning problem formulation

This section begins with a formal definition of the graph learning problem, followed by an explanation of some common approaches used in the literature to tackle this problem. Afterwards, we proceed to formalize the learning problem we aim to address and cast it as an optimization problem. We then provide an overview of the key features of our formulation and conduct a comparative analysis with the two closest approaches available in the literature.

To formally state the graph learning problem, let us recall that we assume: i) $\mathcal{G}$ is an undirected graph with N nodes, ii) there is a random process associated with $\mathcal{G}$, and iii) we denote by $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_R] \in \mathbb{R}^{N \times R}$ a collection of R independent realizations of such a process. The goal in graph learning is to use a given set of nodal observations to find the (estimates of the) links/associations between the nodes in the graph (i.e., use $\mathbf{X}$ to estimate the $\mathbf{S}$ associated with $\mathcal{G}$).

This problem has been addressed under different approaches [46, 67, 68]. Differences among these models typically arise from the underlying assumptions that are made regarding 1) the graph, which almost universally entails just considering that the graph is sparse (see, e.g., [69] for a recent exception), 2) the signals, which assume certain properties related to the nature of the signals such as smoothness [68] or Gaussianity, [67] among others, and 3) the relationship between the graph and the signal, such as the stationarity property [46].

The model we propose aims to incorporate several assumptions about both the graph and the graph signals. To that end, we propose an approach for which we assume that: 1) the graph is sparse, 2) the signals are Gaussian and, 3) the signals are stationary in the underlying graph.

Having established these assumptions, we now proceed to formally state our graph learning problem as follows

Problem 1 *Given a set of signals $\mathbf{X} \in \mathbb{R}^{N \times R}$, find the underlying sparse graph structure encoded in $\mathbf{S}$ under the assumptions:*

(AS1): The graph signals $\mathbf{X}$ are i.i.d. realizations of $\mathcal{N}(\mathbf{0}, \mathbf{C})$.

(AS2): The graph signals $\mathbf{X}$ are stationary in $\mathbf{S}$.

Our approach is to recast Problem 1 as the following optimization

$$\begin{aligned} \min_{\boldsymbol{\Theta}\succeq 0, \mathbf{S}\in\mathcal{S}} \quad & -\log(\det(\boldsymbol{\Theta})) + \mathsf{tr}(\hat{\mathbf{C}}\boldsymbol{\Theta}) + \rho\|\mathbf{S}\|_0. \\ \text{s. t.} \quad & \boldsymbol{\Theta}\mathbf{S} = \mathbf{S}\boldsymbol{\Theta}, \end{aligned} \tag{3.1}$$

where $\mathcal{S}$ is a generic set representing additional constraints that $\mathbf{S}$ is known to satisfy (e.g., the GSO being symmetric and its entries being between zero and one). The minimization takes as input the sample covariance matrix $\hat{\mathbf{C}} = \frac{1}{R}\mathbf{X}\mathbf{X}^{\mathsf{T}}$ and generates as output the estimate for $\mathbf{S}$ and, as a byproduct, the estimate for $\boldsymbol{\Theta}$. For the problem in (3.1), we require $\boldsymbol{\Theta}$ to be positive semidefinite. This constraint arises because $\boldsymbol{\Theta}$ is the inverse of $\hat{\mathbf{C}}$ which is symmetric and positive definite by construction. Consequently, $\boldsymbol{\Theta}$ inherits the properties of being symmetric and positive semidefinite.

Next, we explain the motivation for each term in (3.1) with special emphasis on the constraint $\boldsymbol{\Theta}\mathbf{S} = \mathbf{S}\boldsymbol{\Theta}$, which is a fundamental component of our approach.

- The first two terms in the objective function are due to (AS1) and arise from minimizing the negative log-likelihood expression in (2.3). Indeed, it is clear that substituting (2.2) into (2.3) yields

  $$\sum_{r=1}^{R}\left(-\frac{N}{2}\log(2\pi) - \log(\det(\boldsymbol{\Theta})) + \mathsf{tr}(\mathbf{x}_r^T\boldsymbol{\Theta}\mathbf{x}_r)\right).$$

  Since constants are irrelevant for the optimization, we drop the first term and divide the other two by R, yielding

  $$-\log(\det(\boldsymbol{\Theta})) + \frac{1}{R}\sum_{r=1}^{R}\mathsf{tr}(\mathbf{x}_r\mathbf{x}_r^T\boldsymbol{\Theta}) = -\log(\det(\boldsymbol{\Theta})) + \mathsf{tr}(\hat{\mathbf{C}}\boldsymbol{\Theta}). \tag{3.2}$$

- The term $\rho\|\mathbf{S}\|_0$ accounts for the fact of $\mathbf{S}$ being a GSO (hence, sparse), with $\rho > 0$ being a regularization parameter that determines the desired level of sparsity in the graph.

- Finally, the equality constraint serves to embody (AS2). It is important to highlight that the polynomial relationship between $\mathbf{C}$ and $\mathbf{S}$, as implied by (AS2), is typically addressed in estimation and optimization problems through either: i) extracting the eigenvectors of $\mathbf{C}$ and enforcing them to be the eigenvectors of $\mathbf{S}$ [4], or ii) imposing the constraint $\mathbf{C}\mathbf{S} = \mathbf{S}\mathbf{C}$ [5]. In contrast, our approach encodes the polynomial relation implied by (AS2) by enforcing commutativity between $\boldsymbol{\Theta}$ and $\mathbf{S}$. Note that if $\mathbf{C}$ and $\mathbf{S}$ are full-rank matrices and commute, it follows that $\boldsymbol{\Theta}$ and $\mathbf{S}$ also commute. In other words, $\boldsymbol{\Theta} = \mathbf{C}^{-1}$ can be represented as a polynomial in $\mathbf{S}$, which can be verified by the Cayley-Hamilton theorem.

It is important to note that the assumption of stationarity may seem stringent since it implies commutativity between $\mathbf{S}$ and $\boldsymbol{\Theta}$. However, it provides more degrees of freedom than many existing methods. For example, in partial correlation methods, $\mathbf{S}$ is restricted to be $\mathbf{S} = \boldsymbol{\Theta}$, while in our case, assuming stationarity allows $\boldsymbol{\Theta}$ to be *any* polynomial in $\mathbf{S}$. This leads to a more general approach, including partial correlation as a particular case. To further illustrate this point, consider the sparse structural equation model $\mathbf{X} = \mathbf{A}\mathbf{X} + \mathbf{W}$, with $\mathbf{W}$ being white noise. GL will identify $\mathbf{S} = \boldsymbol{\Theta} = (\mathbf{I}_N - \mathbf{A})^2$, while PGL will identify $\mathbf{S} = \mathbf{A}$.

To better understand the features of PGL, let us briefly discuss the main differences relative to its two closest competitors: GSR and GL.

GSR handles Problem 1 without considering the Gaussian assumption in (AS1). As a result, the first two terms in (3.1), which are associated with the log-likelihood function, are not present. This reduces the problem to inferring a sparse graph under the stationarity constraint. The stationarity assumption is incorporated into (3.1) through the expression $\mathbf{\Theta S} = \mathbf{S\Theta}$, which is equivalent to $\mathbf{CS} = \mathbf{SC}$ if $\mathbf{C}$ is a full-rank matrix. This property enables learning the graph by solving the following optimization problem with the commutativity constraint between $\hat{\mathbf{C}}$ and $\mathbf{S}$:

$$\min_{\mathbf{S}\in\mathcal{S}} \quad \rho\|\mathbf{S}\|_0 \quad \text{s. t.} \quad \hat{\mathbf{C}}\mathbf{S} = \mathbf{S}\hat{\mathbf{C}}, \tag{3.3}$$

where the constraint is typically relaxed as $\|\hat{\mathbf{C}}\mathbf{S} - \mathbf{S}\hat{\mathbf{C}}\|_F^2 \leq \epsilon$ to account for the fact that we have $\hat{\mathbf{C}} \approx \mathbf{C}$. By assuming stationarity, the formulation in (3.3) allows the sample covariance to be modeled as any polynomial in $\mathbf{S}$, making it more general than the formulation in (3.1). However, the absence of Gaussianity in (3.3) means that it is no longer a maximum likelihood estimation. As a result, correctly identifying the ground truth $\mathbf{S}$ requires very reliable estimates of $\hat{\mathbf{C}}$, which usually entails having access to a large number of signals to set ϵ close to zero. This is indeed a challenge, especially in setups with a large number of nodes.

For the second scenario, suppose we simplify (AS2) and instead of considering $\mathbf{\Theta}$ as any polynomial in $\mathbf{S}$, we restrict it to a particular case with the following structure: $\mathbf{\Theta} = \mathbf{C}^{-1} = \sigma\mathbf{I} + \delta\mathbf{S}$. Then, up to the diagonal values and scaling issues, the sparse matrix $\mathbf{S}$ to be estimated and $\mathbf{\Theta} = \mathbf{C}^{-1}$ are the same. Consequently, it suffices to optimize over one of them, leading to the well-known GL formulation:

$$\min_{\mathbf{\Theta}\succeq 0, \mathbf{\Theta}\in\mathcal{S}_{\mathbf{\Theta}}} \quad -\log(\det(\mathbf{\Theta})) + \text{tr}(\hat{\mathbf{C}}\mathbf{\Theta}) + \rho\|\mathbf{\Theta}\|_0. \tag{3.4}$$

The main advantages of (3.4) relative to (3.1) are that the number of variables is smaller and the resulting problem (after relaxing the ℓ_0 norm) is convex. The main drawback is that by forcing the support of $\mathbf{S}$ and $\mathbf{\Theta}$ to be the same, the set of feasible graphs (and their topological properties) is more limited. Indeed, GL can only estimate graphs that are positive definite, while the problem in (3.1) can yield any symmetric matrix. Remarkably, when the model assumed in (3.4) holds true (i.e., data is Gaussian and $\mathbf{\Theta}$ is sparse), GL is able to find reliable estimates of $\mathbf{S}$ even when the number of samples R is fairly low. On the other hand, simulations will show that GL does a poor job estimating $\mathbf{S}$ when the relation between the precision matrix and $\mathcal{G}$ is more involved.

In conclusion, from a conceptual point of view, our formulation reaches a favorable balance between GL and graph-stationarity approaches. This leads to the following two main advantages i) a more general model than GL since our approach models $\mathbf{\Theta}$ as any polynomial in $\mathbf{S}$ and ii) a model with more structure than the graph-stationarity approaches due to the incorporation of (AS1). However, it is important to note that the optimization in (3.1), even if the ℓ_0 norm is relaxed, lacks convexity due to the presence of a bilinear constraint that couples the optimization variables $\mathbf{\Theta}$ and $\mathbf{S}$. These challenges will be addressed in the subsequent section.

3.3 Biconvex relation and algorithm design

As explained in the previous section, the problem in (3.1) is not convex and this challenges designing an algorithm to find a good solution. This section reformulates (3.1), develops an

iterative algorithm, referred to as PGL, to estimate $\mathbf{S}$ and $\mathbf{\Theta}$, and characterizes its convergence to a coordinate-wise minimum point. The proposed approach involves several modifications: 1) we replace the ℓ_0-norm with an elastic net regularizer, which is convex [70]; and 2) we relax the commutativity constraint using an inequality instead of an equality. Next, we explain step by step the resulting formulation.

3.3.1 Biconvex relaxation

The first modification to reformulate (3.1) is to relax the constraint that imposes commutativity between $\mathbf{S}$ and $\mathbf{\Theta}$. Such a constraint is stringent and significantly narrows the feasible solution set of (3.1), which may not be practical in real-world scenarios. Furthermore, considering that our access to the covariance (or precision) matrix is limited to its sampled estimates, enforcing exact commutativity is excessively restrictive. To mitigate this, we relax the original constraint by replacing the matrix equality $\mathbf{\Theta S} = \mathbf{S\Theta}$ with the scalar Frobenius norm-based inequality $\|\mathbf{\Theta S} - \mathbf{S\Theta}\|_F^2 \leq \delta$. This modification not only expands the feasible region but also endows the model with a greater degree of robustness.

The second modification addresses the non-convexity of the objective in (3.1), which originates from the use of the ℓ_0-norm. To alleviate this issue, we relax the problem using an elastic net regularizer. Specifically, we replace the $\rho\|\mathbf{S}\|_0$ penalty with $\rho(\|\mathbf{S}\|_1 + \frac{\eta}{2\rho}\|\mathbf{S}\|_F^2)$, where the parameter η controls the trade-off between the ℓ_1-norm and the Frobenius norm components, and is typically set to a very small value. Although elastic net regularizers have demonstrated practical effectiveness, alternative methods for relaxing the ℓ_0-norm exist (see, for example, [71, 72]), each offering distinct trade-offs in computational complexity, convergence speed, and theoretical underpinnings.

With the incorporation of these two modifications, we reformulate the original graph learning problem presented in (3.1) as follows

$$\begin{array}{ll} \min\limits_{\mathbf{\Theta}\succeq 0, \mathbf{S}\in\mathcal{S}} & -\log(\det(\mathbf{\Theta})) + \mathrm{tr}(\hat{\mathbf{C}}\mathbf{\Theta}) + \rho\|\mathbf{S}\|_1 + \frac{\eta}{2}\|\mathbf{S}\|_F^2, \\ \text{s. t.} & \|\mathbf{\Theta S} - \mathbf{S\Theta}\|_F \leq \delta, \end{array} \tag{3.5}$$

In this setup, δ serves as a hyperparameter chosen according to the quality of the estimation of $\hat{\mathbf{C}}$ which affects the estimation of $\mathbf{\Theta}$. A smaller value of δ is appropriate when the quality of $\hat{\mathbf{C}}$ is high, which typically corresponds to having a sample size R that is substantially larger than the number of nodes. While the relaxation of the commutativity constraint enhances the robustness of our formulation to data quality and simplifies the optimization by reducing the number of Lagrange multipliers, the product of $\mathbf{\Theta}$ and $\mathbf{S}$ still introduces nonconvexity into the problem. The way we propose for dealing with the (updated) biconvex constraint is to solve (3.5) using an alternating optimization algorithm. This family of algorithms is widely used to approximate nonconvex problems by dividing the original problem into several convex subproblems and solving them with respect to each of the variables by fixing all the others. In our particular case, this methodology involves alternately solving for $\mathbf{\Theta}$ with $\mathbf{S}$ held fixed, and then updating $\mathbf{S}$ using the newly updated $\mathbf{\Theta}$, at each iteration.

In the subsequent two subsections, we delve into the detailed methodologies employed to solve each of the two subproblems. Following this, we outline the overall algorithm and discuss its convergence properties. To simplify exposition, in the remainder of the section, we will assume that $\mathbf{S}$ represents the *adjacency* matrix of an undirected graph. Consequently, the feasible solution set for $\mathbf{S}$ is defined as:

$$\mathcal{S} := \{\mathbf{S} \in \mathbb{R}^{N\times N} | \mathbf{S} = \mathbf{S}^T;\ \mathbf{S} \geq \mathbf{0};\ \mathrm{diag}(\mathbf{S}) = \mathbf{0};\ \mathbf{S1} \geq \mathbf{1}\}, \tag{3.6}$$

where $\mathbf{S}$ is constrained to be symmetric with zero diagonal entries and non-negative off-diagonal elements. The additional condition $\mathbf{S1} \geq \mathbf{1}$ is imposed to preclude the trivial solution, i.e., $\mathbf{S} = \mathbf{0}$. Nonetheless, the techniques presented next can be readily extended to accommodate different forms of $\mathbf{S}$.

3.3.2 Solving subproblem for $\mathbf{S}$

We begin by addressing the subproblem with respect to $\mathbf{S}$, while holding $\mathbf{\Theta}$ fixed. The subproblem is formulated as:

$$\begin{array}{ll} \min\limits_{\mathbf{S}\in\mathcal{S}} & \rho\|\mathbf{S}\|_1 + \frac{\eta}{2}\|\mathbf{S}\|_F^2, \\ \text{s. t.} & \|\mathbf{\Theta S} - \mathbf{S\Theta}\|_F \leq \delta. \end{array} \tag{3.7}$$

To solve (3.7), we adopt a linearized *alternating direction method of multipliers* (ADMM) approach, which introduces an auxiliary variable $\mathbf{T}$ and leads to the following equivalent formulation:

$$\begin{array}{ll} \min\limits_{\mathbf{S}\in\mathcal{S},\mathbf{T}} & \rho\|\mathbf{S}\|_1 + \frac{\eta}{2}\|\mathbf{S}\|_F^2, \\ \text{s. t.} & \mathbf{\Theta S} - \mathbf{S\Theta} = \mathbf{T}, \|\mathbf{T}\|_F \leq \delta. \end{array} \tag{3.8}$$

The augmented Lagrangian associated with (3.8) is then given by

$$L(\mathbf{S},\mathbf{T},\mathbf{Z}) = \rho\|\mathbf{S}\|_1 + \frac{\eta}{2}\|\mathbf{S}\|_F^2 + \langle \mathbf{Z}, \mathbf{\Theta S} - \mathbf{S\Theta} - \mathbf{T}\rangle + \frac{\beta}{2}\|\mathbf{\Theta S} - \mathbf{S\Theta} - \mathbf{T}\|_F^2, \tag{3.9}$$

where $\mathbf{Z}$ is the Lagrange multiplier.

To update $\mathbf{S}$ at the t-th iteration, we address the following minimization problem:

$$\min_{\mathbf{S}\in\mathcal{S}} \quad \rho\|\mathbf{S}\|_1 + \frac{\eta}{2}\|\mathbf{S}\|_F^2 + \frac{\beta}{2}\|\mathbf{\Theta S} - \mathbf{S\Theta} - \mathbf{T} + \frac{1}{\beta}\mathbf{Z}\|_F^2, \tag{3.10}$$

where, for the sake of simplicity, we omit the iteration subscript from $\mathbf{\Theta}^{(t)}$ and $\mathbf{Z}^{(t)}$. Problem (3.10) does not admit a closed-form solution due to the term $\frac{1}{2}\|\mathbf{\Theta S} - \mathbf{S\Theta} - \mathbf{T} + \frac{1}{\beta}\mathbf{Z}\|_F^2$. To deal conveniently with this problem we resort to the majorization-minimization (MM) algorithm [73]. We denote this term as $g(\mathbf{S})$ and proceed to majorize both $g(\mathbf{S})$ and $\frac{\eta}{2}\|\mathbf{S}\|_F^2$ at the point $\mathbf{S}^{(t)}$, resulting in the following problem:

$$\min_{\mathbf{S}\in\mathcal{S}} \ \langle \rho\mathbf{I}_{N\times N} + \eta\mathbf{S}^{(t)} + \beta\nabla g(\mathbf{S}^{(t)}), \mathbf{S} - \mathbf{S}^{(t)}\rangle + \frac{L_1}{2}\|\mathbf{S} - \mathbf{S}^{(t)}\|_F^2, \tag{3.11}$$

where $\nabla g(\mathbf{S})$ represents the gradient of $g(\mathbf{S})$, detailed in the equation:

$$\nabla g(\mathbf{S}) = \ \mathbf{\Theta\Theta S} + \mathbf{S\Theta\Theta} - 2\mathbf{\Theta S\Theta} + \mathbf{T\Theta} - \mathbf{\Theta T} + \frac{1}{\beta}(\mathbf{\Theta Z} - \mathbf{Z\Theta}). \tag{3.12}$$

Now, Problem (3.11) has a closed-form solution, allowing for the update of $\mathbf{S}^{(t+1)}$ as follows:

$$\mathbf{S}^{(t+1)} = \mathcal{P}_{\mathcal{S}}\Big(\mathbf{S}^{(t)} - \frac{1}{L_1}(\rho\mathbf{I}_N + \eta\mathbf{S}^{(t)} + \beta\nabla g(\mathbf{S}^{(t)}))\Big), \tag{3.13}$$

where $\mathcal{P}_{\mathcal{S}}$ is the projection onto the set $\mathcal{S}$ with respect to the Frobenius norm, which can be computed efficiently by the Dykstra's projection algorithm [74]. More specifically, the set $\mathcal{S}$ can be written as the intersection of two closed convex sets as follows:

$$\mathcal{S} = \mathcal{S}_A \cap \mathcal{S}_B, \tag{3.14}$$

Algorithm 1: Inner loop for $\mathbf{S}$ update.

Input: $\hat{\mathbf{\Theta}}^{(k)}$, $\hat{\mathbf{S}}^{(k)}$, $\hat{\mathbf{T}}^{(k)}$, $\hat{\mathbf{Z}}^{(k)}$, ρ, η, β, δ
Outputs: $\hat{\mathbf{S}}^{(k+1)}$, $\hat{\mathbf{T}}^{(k+1)}$, $\hat{\mathbf{Z}}^{(k+1)}$

Initialize $\mathbf{S}^{(0)}=\hat{\mathbf{S}}^{(k)}$, $\mathbf{T}^{(0)}=\mathbf{T}^{(k)}$, $\mathbf{Z}^{(0)}=\mathbf{Z}^{(k)}$
for $t=0$ **to** $T-1$ **do**
    Update $\mathbf{S}^{(t+1)}$ by (3.13);
    Update $\mathbf{T}^{(t+1)}$ by (3.16);
    Update $\mathbf{Z}^{(t+1)}$ by (3.18);
end
$\hat{\mathbf{S}}^{(k+1)}=\mathbf{S}^{(T)}$, $\hat{\mathbf{T}}^{(k+1)}=\mathbf{T}^{(T)}$, $\hat{\mathbf{Z}}^{(k+1)}=\mathbf{Z}^{(T)}$.

where $\mathcal{S}_A := \left\{\mathbf{S}\in\mathbb{R}^{N\times N} \,|\, \mathbf{S}=\mathbf{S}^T\right\}$ and $\mathcal{S}_B := \left\{\mathbf{S}\in\mathbb{R}^{N\times N} \,|\, \mathbf{S}\geq 0;\, \mathrm{diag}(\mathbf{S})=\mathbf{0};\, \mathbf{S1}\geq\mathbf{1}\right\}$. We employ Dykstra's projection algorithm [74] to compute the nearest point projection of a given point onto the intersection of sets $\mathcal{S}_A$ and $\mathcal{S}_B$. Dykstra's algorithm achieves this by alternately projecting the point onto $\mathcal{S}_A$ and $\mathcal{S}_B$ until the solution is reached. For a more comprehensive understanding of Dykstra's projection algorithm, the reader is directed to [74]. Detailed descriptions of the projection computations onto sets $\mathcal{S}_A$ and $\mathcal{S}_B$ are provided in Appendix 3.6.

Returning to the augmented Lagrangian in (3.9), we update $\mathbf{T}$ at the t-th iteration by solving the following problem:

$$\begin{array}{ll}\min\limits_{\mathbf{T}} & \frac{\beta}{2}\|\mathbf{T}-\mathbf{\Theta S}+\mathbf{S\Theta}-\frac{1}{\beta}\mathbf{Z}\|_F^2,\\ \text{s. t.} & \|\mathbf{T}\|_F\leq\delta,\end{array} \tag{3.15}$$

where we have simplified the notation by omitting the iteration subscripts from $\mathbf{S}^{(t+1)}$ and $\mathbf{Z}^{(t)}$. Problem (3.15) has a closed-form solution. As a result, $\mathbf{T}^{(t+1)}$ can be updated by

$$\mathbf{T}^{(t+1)}=\mathcal{P}_\delta\left(\mathbf{\Theta S}-\mathbf{S\Theta}+\frac{1}{\beta}\mathbf{Z}\right), \tag{3.16}$$

where $\mathcal{P}_\delta$ denotes the projection defined by:

$$\mathcal{P}_\delta(\mathbf{A})=\begin{cases}\frac{\delta}{\|\mathbf{A}\|_F}\mathbf{A} & \text{if } \|\mathbf{A}\|_F>\delta\\ \mathbf{A} & \text{otherwise.}\end{cases} \tag{3.17}$$

Finally, the dual variable $\mathbf{Z}$ is updated according to:

$$\mathbf{Z}^{(t+1)}=\mathbf{Z}^{(t)}+\beta(\mathbf{\Theta S}-\mathbf{S\Theta}-\mathbf{T}), \tag{3.18}$$

where the iteration subscripts from $\mathbf{S}^{(t+1)}$ and $\mathbf{T}^{(t+1)}$ have been omitted for simplicity. A pseudocode of the steps to be performed for the update of $\mathbf{S}$ is summarized in Algorithm 1.

If the parameter L_1 in (3.11) is larger than the Lipschitz constant of the gradient of $\beta g(\mathbf{S})+\frac{\eta}{2}\|\mathbf{S}\|_F^2$, then the sequence $\left\{(\mathbf{S}^{(t)},\mathbf{T}^{(t)})\right\}$ converges to the optimal solution of Problem (3.8), and $\left\{\mathbf{Z}^{(t)}\right\}$ converges to the optimal solution of the dual of problem (3.8), which follows from the existing convergence result of majorized ADMM [75]. To enhance empirical convergence rates, adopting a more proactive strategy for selecting the parameter L_1 is beneficial. For example, utilizing a backtracking line search to determine the stepsize in (3.13) can help to accelerate convergence.

We note that the choice of the penalty parameter β can affect the convergence speed of the ADMM algorithm. A poorly chosen β may lead to very slow convergence in practice. Adaptive

schemes that adjust β have been shown to often result in better practical performance. For example, we can adopt the adaptive update rule presented in [76]:

$$\beta^{(t+1)} = \begin{cases} \tau^{\text{inc}}\beta^{(t)}, & \text{if } \|\mathbf{r}^{(t)}\|_F > \mu\|\mathbf{s}^{(t)}\|_F, \\ \beta^{(t)}/\tau^{\text{dec}}, & \text{if } \|\mathbf{s}^{(t)}\|_F > \mu\|\mathbf{r}^{(t)}\|_F, \\ \beta^{(t)}, & \text{otherwise,} \end{cases} \tag{3.19}$$

where $\mu > 1$, $\tau^{\text{inc}} > 1$, and $\tau^{\text{dec}} > 1$ are predefined parameters. Here, $\mathbf{r}^{(t)}$ and $\mathbf{s}^{(t)}$ represent the primal and dual residuals at iteration t, respectively. They are defined as

$$\mathbf{r}^{(t)} = \boldsymbol{\Theta}\mathbf{S}^{(t)} - \mathbf{S}^{(t)}\boldsymbol{\Theta} - \mathbf{P}^{(t)},$$

and

$$\mathbf{s}^{(t)} = \beta^{(t)}\boldsymbol{\Theta}(\mathbf{P}^{(t)} - \mathbf{P}^{(t-1)}) - \beta^{(t)}(\mathbf{P}^{(t)} - \mathbf{P}^{(t-1)})\boldsymbol{\Theta}.$$

Although it can be challenging to prove the convergence of ADMM when β varies by iteration, the convergence theory established for a fixed β remains applicable if one assumes that β becomes constant after a finite number of iterations.

3.3.3 Solving subproblem for $\boldsymbol{\Theta}$

Using the formulation from (3.5), we now turn our attention to the subproblem for $\boldsymbol{\Theta}$

$$\begin{array}{ll} \min\limits_{\boldsymbol{\Theta}\succeq 0} & -\log(\det(\boldsymbol{\Theta})) + \text{tr}(\hat{\mathbf{C}}\boldsymbol{\Theta}), \\ \text{s. t.} & \|\mathbf{S}\boldsymbol{\Theta} - \boldsymbol{\Theta}\mathbf{S}\|_F \leq \delta. \end{array} \tag{3.20}$$

Similarly to the approach taken for the $\mathbf{S}$ subproblem, we reformulate the subproblem (3.20) for $\boldsymbol{\Theta}$ as follows

$$\begin{array}{ll} \min\limits_{\boldsymbol{\Theta}\succeq 0,\, \mathbf{Q}} & -\log(\det(\boldsymbol{\Theta})) + \text{tr}(\hat{\mathbf{C}}\boldsymbol{\Theta}), \\ \text{s. t.} & \mathbf{S}\boldsymbol{\Theta} - \boldsymbol{\Theta}\mathbf{S} = \mathbf{Q},\ \|\mathbf{Q}\|_F \leq \delta. \end{array} \tag{3.21}$$

The augmented Lagrangian associated with (3.21) is given by

$$L(\boldsymbol{\Theta}, \mathbf{Q}, \mathbf{Y}) = -\log(\det(\boldsymbol{\Theta})) + \text{tr}(\hat{\mathbf{C}}\boldsymbol{\Theta}) + \frac{\beta}{2}\|\mathbf{S}\boldsymbol{\Theta} - \boldsymbol{\Theta}\mathbf{S} - \mathbf{Q} + \frac{1}{\beta}\mathbf{Y}\|_F^2. \tag{3.22}$$

To update $\boldsymbol{\Theta}$ at the t-th iteration, we solve the following optimization problem

$$\min_{\boldsymbol{\Theta}\succeq 0} -\log(\det(\boldsymbol{\Theta})) + \text{tr}(\hat{\mathbf{C}}\boldsymbol{\Theta}) + \frac{\beta}{2}\|\mathbf{S}\boldsymbol{\Theta} - \boldsymbol{\Theta}\mathbf{S} - \mathbf{Q} + \frac{1}{\beta}\mathbf{Y}\|_F^2, \tag{3.23}$$

where we have omitted the iteration subscripts from $\mathbf{Q}^{(t)}$ and $\mathbf{Y}^{(t)}$ for simplicity.

Let $f(\boldsymbol{\Theta}) = \frac{1}{2}\|\mathbf{S}\boldsymbol{\Theta} - \boldsymbol{\Theta}\mathbf{S} - \mathbf{Q} + \frac{1}{\beta}\mathbf{Y}\|_F^2$. We then construct the majorizer of the objective function in (3.23) at the point $\boldsymbol{\Theta}^{(t)}$ and obtain

$$\min_{\boldsymbol{\Theta}\succeq 0} -\log(\det(\boldsymbol{\Theta})) + \langle \beta\nabla f(\boldsymbol{\Theta}^{(t)}) + \hat{\mathbf{C}}, \boldsymbol{\Theta} - \boldsymbol{\Theta}^{(t)}\rangle + \frac{L_2}{2}\|\boldsymbol{\Theta} - \boldsymbol{\Theta}^{(t)}\|_F^2, \tag{3.24}$$

where $\nabla f(\boldsymbol{\Theta})$ denotes the gradient of $f(\boldsymbol{\Theta})$

$$\nabla f(\boldsymbol{\Theta}) = \mathbf{S}\mathbf{S}\boldsymbol{\Theta} + \boldsymbol{\Theta}\mathbf{S}\mathbf{S} - 2\mathbf{S}\boldsymbol{\Theta}\mathbf{S} + \mathbf{Q}\mathbf{S} - \mathbf{S}\mathbf{Q} + \frac{1}{\beta}(\mathbf{S}\mathbf{Y} - \mathbf{Y}\mathbf{S}). \tag{3.25}$$

Algorithm 2: Inner loop for $\boldsymbol{\Theta}$ update.

Input: $\mathbf{C}$, $\hat{\boldsymbol{\Theta}}^{(k)}$, $\hat{\mathbf{S}}^{(k+1)}$, $\hat{\mathbf{Q}}^{(k)}$, $\hat{\mathbf{Y}}^{(k)}$, β, δ

Outputs: $\hat{\boldsymbol{\Theta}}^{(k+1)}$, $\hat{\mathbf{Q}}^{(k+1)}$, $\hat{\mathbf{Y}}^{(k+1)}$

Initialize $\boldsymbol{\Theta}^{(0)} = \hat{\boldsymbol{\Theta}}^{(k)}$, $\mathbf{Q}^{(0)} = \hat{\mathbf{Q}}^{(k)}$, $\mathbf{Y}^{(0)} = \hat{\mathbf{Y}}^{(k)}$

for $t = 0$ **to** $T - 1$ **do**

  Update $\boldsymbol{\Theta}^{(t+1)}$ by (3.26);

  Update $\mathbf{Q}^{(t+1)}$ by (3.28);

  Update $\mathbf{Y}^{(t+1)}$ by (3.29);

end

$\hat{\boldsymbol{\Theta}}^{(k+1)} = \boldsymbol{\Theta}^{(T)}$, $\hat{\mathbf{Q}}^{(k+1)} = \mathbf{Q}^{(T)}$, $\hat{\mathbf{Y}}^{(k+1)} = \mathbf{T}^{(T)}$.

Lemma 1. *The optimal solution of problem* (3.23) *is [77]*

$$\boldsymbol{\Theta}^{(t+1)} = \mathbf{U}\left(\frac{\boldsymbol{\Lambda} + \sqrt{\boldsymbol{\Lambda}^2 + \frac{4}{L_2}}}{2}\right)\mathbf{U}^\top, \tag{3.26}$$

where $\boldsymbol{\Lambda}$ and $\mathbf{U}$ contain the eigenvalues and eigenvectors of $\boldsymbol{\Theta}^{(t)} - \frac{1}{L_2}\left(\beta\nabla f(\boldsymbol{\Theta}^{(t)}) + \hat{\mathbf{C}}\right)$, respectively.

Then, $\mathbf{Q}$ is updated by addressing the following problem

$$\begin{array}{ll}\min\limits_{\mathbf{Q}} & \frac{\beta}{2}\|\mathbf{Q} - \mathbf{S}\boldsymbol{\Theta} + \boldsymbol{\Theta}\mathbf{S} - \frac{1}{\beta}\mathbf{Y}\|_F^2, \\ \text{s. t.} & \|\mathbf{Q}\|_F \leq \delta,\end{array} \tag{3.27}$$

where the iteration subscripts from $\boldsymbol{\Theta}^{(t+1)}$ and $\mathbf{Y}^{(t)}$ have been omitted for simplicity. Similar to the case of updating $\mathbf{T}$, we update $\mathbf{Q}$ as

$$\mathbf{Q}^{(t+1)} = \mathcal{P}_\delta\left(\mathbf{S}\boldsymbol{\Theta} - \boldsymbol{\Theta}\mathbf{S} + \frac{1}{\beta}\mathbf{Y}\right). \tag{3.28}$$

Finally, the Lagrange multiplier $\mathbf{Y}$ is updated as follows

$$\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \beta(\mathbf{S}\boldsymbol{\Theta} - \boldsymbol{\Theta}\mathbf{S} - \mathbf{Q}), \tag{3.29}$$

where the iteration subscripts from $\boldsymbol{\Theta}^{(t+1)}$ and $\mathbf{Q}^{(t+1)}$ have been similarly omitted for clarity.

We can also use the adaptive strategy in (3.19) to adjust β during iterations, with $\mathbf{r}^{(t)}$ and $\mathbf{s}^{(t)}$ defined as

$$\mathbf{r}^{(t)} = \mathbf{S}\boldsymbol{\Theta}^{(t)} - \boldsymbol{\Theta}^{(t)}\mathbf{S} - \mathbf{Q}^{(t)},$$

and

$$\mathbf{s}^{(t)} = \beta^{(t)}\mathbf{S}(\mathbf{Q}^{(t)} - \mathbf{Q}^{(t-1)}) - \beta^{(t)}(\mathbf{Q}^{(t)} - \mathbf{Q}^{(t-1)})\mathbf{S}.$$

A pseudocode of the steps to be performed for the update of $\boldsymbol{\Theta}$ is summarized in Algorithm 2.

3.3.4 Graph-learning algorithm and convergence analysis

Leveraging the results presented in the previous two subsections, the steps to run our iterative scheme to find a solution $(\hat{\boldsymbol{\Theta}}, \hat{\mathbf{S}})$ to (3.5) are summarized in Algorithm 3.

Before presenting the associated theoretical analysis, several comments regarding the implementation of Algorithm 3 are in order:

- For simplicity, the algorithm considers a fixed number of iterations, but a prudent approach is to monitor the cost reduction at each iteration and implement an early exit approach if no meaningful improvement is achieved.
- The value of the hyperparameters ρ, η and δ is an input to the algorithm. We note that η is typically set to a small value to guarantee that the (sparsity promoting) ℓ_1 norm plays a more prominent role. Moreover, the value of δ should be chosen based on the quality of the estimate of the sample covariance matrix $\hat{\mathbf{C}}$. The higher the number of observations R (hence, the better the quality of $\hat{\mathbf{C}}$), the smaller the value of δ. Similarly, if the number of nodes N is high, the value of δ should be re-scaled accordingly, so that the constraint does not become too restrictive.
- The update of $\mathbf{S}$ poses the primary computational challenge, mainly due to the complex nature of its estimation. In contrast to $\mathbf{\Theta}$, which is primarily estimated from the data matrix $\hat{\mathbf{C}}$, the estimation of $\mathbf{S}$ relies on its interplay with $\mathbf{\Theta}$ through the relaxed commutativity constraint. This indirect relationship adds to the complexity of the estimation, as it does not directly benefit from data-driven insights, often necessitating greater precision in solving the corresponding subproblem. Furthermore, the constraints imposed on $\mathbf{S}$ are substantially more complex than those on $\mathbf{\Theta}$, thereby increasing the computational load to obtain a solution that lies within the feasible set. To alleviate the computational demand, we may employ a relatively loose stopping criterion for the $\mathbf{\Theta}$ subproblem, which can expedite convergence without significantly affecting the quality of the solution.

To establish the theoretical convergence of Algorithm 3, we begin by introducing several definitions and (mild) assumptions pertinent to Problem (3.5).

Let $f(\mathbf{\Theta}, \mathbf{S})$ denote the objective function and $\mathcal{X}$ the feasible set of Problem (3.5), respectively. We define $(\bar{\mathbf{\Theta}}, \bar{\mathbf{S}})$ as a *block coordinatewise minimizer* of Problem (3.5) if:

$$f(\bar{\mathbf{\Theta}}, \bar{\mathbf{S}}) \leq f(\mathbf{\Theta}, \bar{\mathbf{S}}), \quad \text{for all } \mathbf{\Theta} \in \bar{\mathcal{X}}_{\mathbf{\Theta}}, \tag{3.30}$$

and

$$f(\bar{\mathbf{\Theta}}, \bar{\mathbf{S}}) \leq f(\bar{\mathbf{\Theta}}, \mathbf{S}), \quad \text{for all } \mathbf{S} \in \bar{\mathcal{X}}_{\mathbf{S}}, \tag{3.31}$$

where $\bar{\mathcal{X}}_{\mathbf{\Theta}} = \{\mathbf{\Theta} \in \mathbb{R}^{N \times N} \,|\, (\mathbf{\Theta}, \bar{\mathbf{S}}) \in \mathcal{X}\}$, and $\bar{\mathcal{X}}_{\mathbf{S}} = \{\mathbf{S} \in \mathbb{R}^{N \times N} \,|\, (\bar{\mathbf{\Theta}}, \mathbf{S}) \in \mathcal{X}\}$.

Furthermore, we introduce the following assumptions for our analysis:

Assumption 3.1. *The parameter δ in Problem* (3.5) *is sufficiently large to ensure that the feasible set of the subproblem* (3.7) *is nonempty at every iteration.*

We require Assumption 3.1 because the intersection $\mathcal{S} \cap \{\mathbf{S} \in \mathbb{R}^{N \times N} |, \|\mathbf{\Theta}\mathbf{S} - \mathbf{S}\mathbf{\Theta}\|_F \leq \delta\}$ may otherwise be empty, implying that the feasible set of subproblem (3.7) could be nonexistent. However, Assumption 3.1 is relatively mild, as we can always choose a sufficiently large δ to ensure that the feasible set of subproblem (3.7) remains nonempty at every iteration. Given Assumption 3.1, our algorithm is guaranteed to find a minimizer of subproblem (3.7) throughout its iterations. Additionally, the feasibility of subproblem (3.20) is inherently assured.

Assumption 3.2. *The matrix $\hat{\mathbf{C}}$ in Problem* (3.5) *is positive definite.*

Algorithm 3: Polynomial Graphical Lasso (PGL)

Input: $\hat{\mathbf{C}}$, ρ, η, β, and δ
Outputs: $\hat{\mathbf{\Theta}}$ and $\hat{\mathbf{S}}$

Initialize $\hat{\mathbf{\Theta}}^{(0)} = \hat{\mathbf{C}}^{-1}$
Initialize $\hat{\mathbf{S}}^{(0)}$ by solving (3.3)
Initialize $\mathbf{T}^{(0)}$, $\mathbf{Q}^{(0)}$, $\mathbf{Y}^{(0)}$, and $\mathbf{Z}^{(0)}$ to zero
for $k = 0$ **to** $K - 1$ **do**
    Update $\hat{\mathbf{S}}^{(k+1)}$ by running Algorithm 1;
    Update $\hat{\mathbf{\Theta}}^{(k+1)}$ by running Algorithm 2;
end
$\hat{\mathbf{\Theta}} = \hat{\mathbf{\Theta}}^{(K)}$, $\hat{\mathbf{S}} = \hat{\mathbf{S}}^{(K)}$.

Assumption 3.2, which requires all the eigenvalues to be nonzero, guarantees that subproblem (3.20) is well defined. Without this assumption, the objective function in (3.20) may fail to achieve a finite minimum value. In cases where Assumption 3.2 may not hold, incorporating an norm regularizer for $\mathbf{\Theta}$ would bound the solution, thereby ensuring the existence of a minimizer.

Theorem 3.1. *Let* $\{(\hat{\mathbf{\Theta}}^{(k)}, \hat{\mathbf{S}}^{(k)})\}_{k\in\mathbb{N}}$ *be a sequence generated by Algorithm 3. Under Assumptions 3.1 and 3.2, every limit point of* $\{(\hat{\mathbf{\Theta}}^{(k)}, \hat{\mathbf{S}}^{(k)})\}_{k\in\mathbb{N}}$ *is a block coordinatewise minimizer of Problem* (3.5).

The proof of Theorem 3.1 is deferred to Appendix 3.8. This theorem asserts the subsequence convergence of our algorithm to a *block coordinatewise minimizer* of Problem (3.5). When the *block coordinatewise minimizer* lies within the interior of the feasible set, it becomes a stationary point. The theorem's assertions are significant both theoretically and practically. As discussed earlier in the chapter, GMRFs are a particular case of Gaussian graph-stationary processes. Taking this into account, we can always initialize our algorithm using the solution estimated by GL (which is optimal for GMRF) and then, run iteratively the updates over $\mathbf{\Theta}$ and $\mathbf{S}$ in Algorithm 3, to get an (enhanced) coordinatewise minimum estimate.

3.4 Numerical experiments

This section evaluates quantitatively the performance of PGL. Since PGL can be understood as a generalization of the widely adopted GL (with $\mathbf{\Theta}$ being any polynomial in $\mathbf{S}$), in most test cases, we will test both algorithms. Similarly, we also test the learning performance of GSR [4], which assumes stationarity but not Gaussianity. The performance results obtained from both synthetic and real-data experiments are summarized in Figs. 3.1-3.6. Unless otherwise stated, to assess the quality of the estimated GSO, we use the normalized mean error between the estimated and true $\mathbf{S}$. Mathematically, this entails computing[1]

$$\mathrm{nme}(\mathbf{S}^*, \hat{\mathbf{S}}) = \frac{\|\mathbf{S}^* - \hat{\mathbf{S}}\|_F^2}{\|\mathbf{S}^*\|_F^2}, \tag{3.32}$$

[1]Results for other recovery metrics (including accuracy and F1 score) as well as additional simulations can be found both in our conference precursor [1] and in our online repository `https://github.com/andreibuciulea/GaussSt_TopoID`.

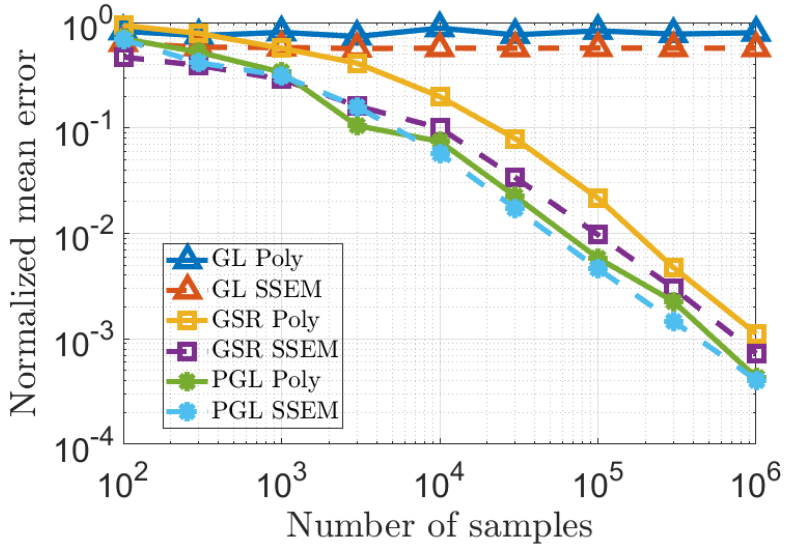

Figure 3.1: Graph estimation error $\mathrm{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$ vs. number of samples R for different graph learning approaches (PGL, GL, and GSR). The six lines reported in each subplot correspond to the combination of 3 different graph learning methods and 2 different covariance setups (SSEM and Poly) for a noise-free scenario.

where $\hat{\mathbf{S}}$ and $\mathbf{S}^*$ represent the estimated and true $\mathbf{S}$ respectively. Moreover, for the synthetic experiments we test the graph learning algorithms over 100 realizations of random graphs $\{\mathcal{G}_i\}_{i=1}^{100}$ and report the average normalized mean error $\frac{1}{100}\sum_{i=1}^{100}\mathrm{nme}(\mathbf{S}_i^*, \hat{\mathbf{S}}_i)$.

If not specified otherwise, in our synthetic experiments, we consider graphs with $N = 20$ nodes generated using the Erdös-Rényi (ER) model with a link probability of $p = 0.1$. Regarding the generation of the graph signals, three different setups for the covariance matrices have been studied. For the setup referred to as "Poly", the covariance matrix $\mathbf{C}$ is generated as a random polynomial of the GSO of the form $\mathbf{C} = (\sum_{l=0}^{L-1} h_l \mathbf{S}^l)^2$, where h_l are random coefficients drawn from a normalized zero-mean Gaussian distribution, and the square operator guarantees matrix $\mathbf{C}$ to be positive definite. The setup referred to as "SSEM" constructs the covariance matrices following the sparse structural equation model [41] for graph signal generation as $\mathbf{C} = (\mathbf{I} - \mathbf{S})^2$, where $\mathbf{S}$ is selected to ensure that $\mathbf{C}$ is positive definite. The setup referred to as "MRF" constructs the covariance matrices following the assumptions made by GL as $\mathbf{C} = (\mu\mathbf{I} + \nu\mathbf{S})^{-1}$, where μ is some positive number large enough to assure that $\mathbf{C}^{-1}$ is positive definite and ν is some positive random number.

3.4.1 Test case 1: Estimation error vs. number of samples for multiple synthetic scenarios.

In this first test case, we employ synthetic scenarios for testing the performance of our approach in terms of $\mathrm{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$ vs. R and also compare the results with other methods from the literature. The different scenarios considered are detailed below.

Error vs. number of samples for different graph learning methods and signal models. The results of the experiment depicted in Fig. 3.1a, compare the $\mathrm{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$ (y-axis) of various algorithms with respect to the number of available samples R (x-axis). Moreover, we consider

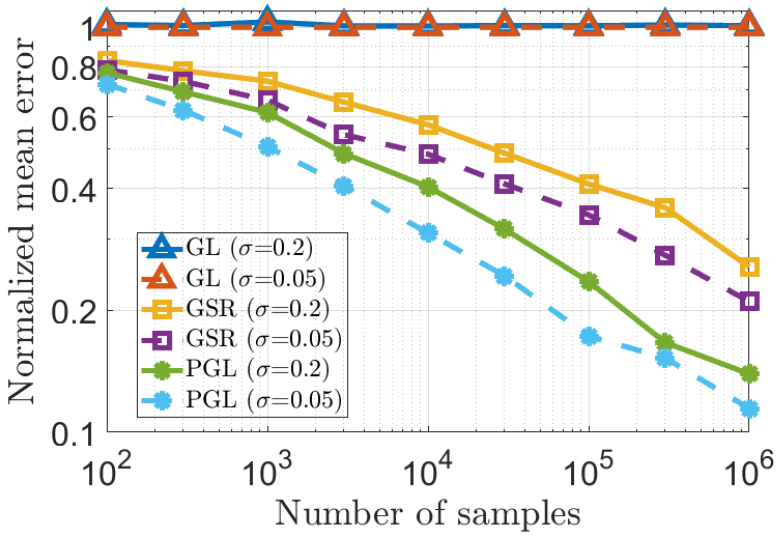

Figure 3.2: Graph estimation error $\text{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$ vs. number of samples R for different graph learning approaches (PGL, GL, and GSR). The six lines reported in each subplot correspond to the combination of 3 different graph learning methods and 2 noise levels $\sigma \in \{0.05, 0.2\}$ for a Poly setup.

SSEM and Poly setups for data generation. The results shown in Fig. 3.1a reveal that: i) PGL outperforms its competitors, ii) the error decreases as R increases, and iii) estimation is more accurate for SSEM than for Poly. Next, we discuss these three main findings in greater detail. All algorithms do a better job estimating the graph for the SSEM model. Since $\mathbf{\Theta}$ for SSEM is a specific second-order polynomial in $\mathbf{S}$ it can be seen as a particular case of Poly, and consequently, a scenario from which the graph structure is easier to estimate. Indeed, while GL fails to estimate the graph for the Poly model, it is able to estimate some of the links for the SSEM. However, the estimation error of GL is quite large and does not change with the number of samples R, demonstrating that the poor performance is due to a model mismatch. This will be further confirmed in Section 3.4.2, where we simulate a GMRF data generation setup that aligns perfectly with the assumptions made by GL. Regarding PGL and GSR, we observe that the error decreases almost linearly with the number of samples R. Perhaps more importantly, we also observe that, as R increases, the gap between PGL and GSR diminishes. For example, while for the Poly case GSR needs 10 times more samples than PGL to achieve an error of 10^{-1}, GSR only needs 3 times more samples than PGL to achieve an error of 10^{-3}. This illustrates that, as expected, the gains associated with assuming Gaussianity are stronger when the number of observations R is small, vanishing as R grows very large.

Error vs. number of samples for noisy observations. Next, we assess the performance of the graph-learning algorithms in the presence of additive white Gaussian observation noise. The results are shown in Fig. 3.1b. As in the previous test case, we report $\text{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$ vs R for PGL, GL and GSR. The difference here is that we consider only the more intricate signal generation model (Poly) and two normalized noise levels ($\sigma = 0.05$, $\sigma = 0.2$). The main observations in this case are: i) PGL outperforms GSR and GL, ii) the error for PGL and GSR decreases as R increases, while the one for GL is flat and close to 1, iii) the estimation performance for PGL and GSR worsens as the noise level σ increases, deteriorating noticeably with respect to the setup in Fig. 3.1a, and iv) the gap between PGL and GSR grows. The findings in i) and ii) are consistent with those found in the previous test case. Finding iii) is expected and common in all graph-learning approaches. Finally, the larger gap in finding iv) is due to the fact that this is a more challenging scenario (high-order

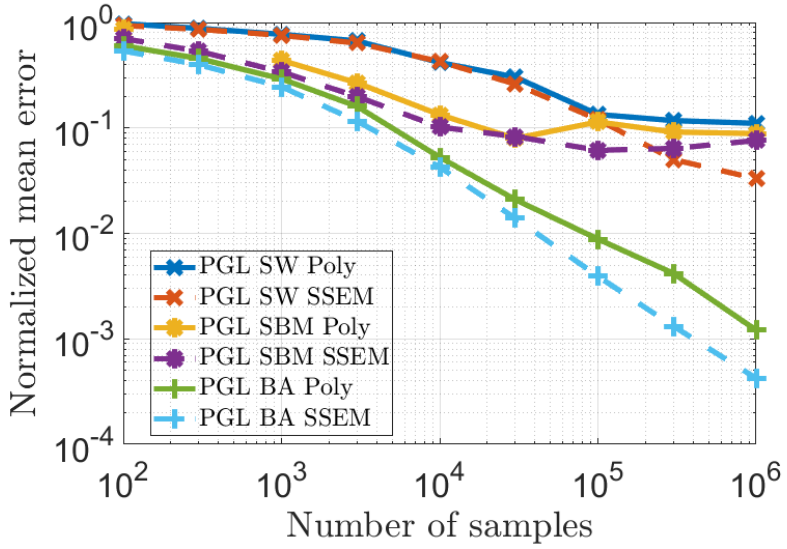

Figure 3.3: Graph estimation error $\mathrm{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$ vs. number of samples R for different graph generation models (SW, SBM, and BA). The six lines reported in each subplot correspond to the combination of 3 graph generation models and 2 covariance models (SSEM and Poly) for the PGL algorithm in a noise-free scenario.

polynomial covariances and observation noise), and, as a result, the higher level of sophistication of PGL relative to GSR translates into more noticeable gains.

Error vs. number of samples for different graph models. This test case considers network models other than ER. In particular, three different types of graphs are considered: 1) Small World (SW) graphs with mean node degree 4 and rewiring probability 0.15; 2) Stochastic block model (SBM) graphs with 4 clusters, and intra and inter-cluster edge probability of 0.8 and 0.05, respectively; and 3) Barabási-Albert (BA) graphs with 2 edges to attach at every step. As in the first test case, we consider two types of signal generation models: SSEM and Poly. Fig. 3.1c reports the error vs. the number of samples for the six combinations considered (3 types of graphs and 2 types of signal generation models). The results show that there is a significant difference in performance between SW, SBM, and BA, which is part due to the sparsity level present in each graph. One of the assumptions codified in our model is that the graph is sparse, and, as a result, our algorithm does a better job estimating BA (the one with the lowest average degree) than SBM and SW (the one with the highest average degree). Finally, we also note that the estimation error achieved with SSEM consistently outperforms that of the Poly setup. These findings align with the results presented in Fig. 3.1a for ER graphs and are in accordance with the theoretical discussion that postulated SSEM as a specific instance of Poly.

3.4.2 Test case 2: Noisy GMRF graph signals.

In this test case, the goal is to assess the behavior of PGL for GMRF observations, which is the setup that motivated the development of the GL algorithm.

Estimation error considering noisy GMFR signals. In this experiment, we replicate the scenario from Fig.3.1b, utilizing a GMFR model to generate the signals. The performance of PGL, GL and GSR for two different noise levels, $\sigma \in \{0.05, 0.2\}$, is depicted in Fig. 3.4. The main observations

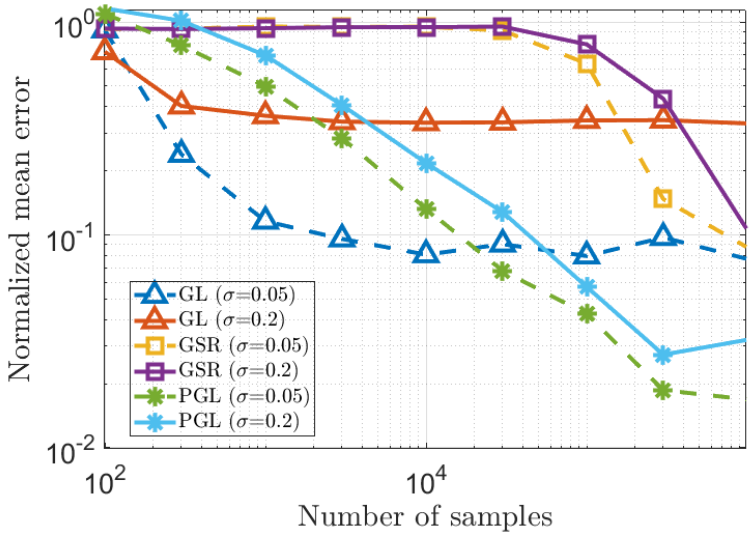

Figure 3.4: Graph estimation error $\mathrm{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$ vs. number of samples R for data generated according to a GMRF. We consider three different graph learning schemes (PGL, GL, and GSR) and two levels of additive white noise ($\sigma \in \{0.05, 0.2\}$), giving rise to the six lines in the figure.

are: i) across all considered approaches, increasing the noise level σ leads to a deterioration in terms of $\mathrm{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$; ii) GSR always performs worse than PGL, providing very poor results when $R < 10^4$; and iii) GL outperforms PGL when the number of observations R is small. Findings i) and ii) are unsurprising and consistent with the behavior observed in the previous experiments, showcasing the benefits of considering the log-likelihood regularization in the optimization run by PGL. Regarding iii), GL outperforming PGL is expected, since the latter needs to "learn" the particular polynomial between $\mathbf{\Theta}$ and $\mathbf{S}$.

However, as the value of R increases, the error in PGL gradually decreases, while the error in GL remains relatively constant, leading to PGL outperforming GL for large values of R. This behavior is more surprising and can be attributed to the fact that GL focuses on learning the precision matrix $\mathbf{\Theta}$, while PGL balances the accuracy in terms of both the precision $\mathbf{\Theta}$ and the graph $\mathbf{S}$. Since the reported error focuses on the estimation $\mathbf{S}$, the values of the diagonal of $\mathbf{\Theta}$ are not relevant for $\mathrm{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$ and this can explain that GL, which is a maximum likelihood estimate for $\mathbf{\Theta}$, does not yield the minimum $\mathrm{nme}(\mathbf{S}^*, \hat{\mathbf{S}})$.

3.4.3 Test case 3: Computational complexity.

Here, we compare the runtime obtained by the efficient implementation of our PGL scheme provided in Algorithm 3 and a generic block-coordinate alternating minimization algorithm that uses a generic software for disciplined convex programming (CVX) [78], the most common off-the-shelf solver for convex problems.

Computational complexity and estimation error. The objective of this experiment is to evaluate the running time and estimation error for different versions of our algorithm as the number of nodes increases. In particular, the experiment focuses on the Poly setup, utilizing $R = 10^6$ graph signals, and averages the results over 50 graph realizations. Table 3.1 lists the elapsed time required to obtain the graph and precision estimates for problems with different numbers of nodes N using

Alg. $\setminus N$	20	30	40	50	60	Metric
PGL-CVX	$2.37 \cdot 10^{1}$	$3.88 \cdot 10^{1}$	$1.29 \cdot 10^{2}$	$1.11 \cdot 10^{3}$	$3.13 \cdot 10^{3}$	**Time** (s)
PGL-Alg.3	$2.49 \cdot 10^{0}$	$2.72 \cdot 10^{0}$	$2.88 \cdot 10^{0}$	$3.81 \cdot 10^{0}$	$4.45 \cdot 10^{0}$	
PGL-CVX	$7.98 \cdot 10^{-4}$	$3.12 \cdot 10^{-3}$	$1.35 \cdot 10^{-2}$	$2.59 \cdot 10^{-2}$	$9.16 \cdot 10^{-2}$	nme$(\mathbf{S}^*, \hat{\mathbf{S}})$
PGL-Alg.3	$4.66 \cdot 10^{-4}$	$1.30 \cdot 10^{-3}$	$2.18 \cdot 10^{-3}$	$5.07 \cdot 10^{-3}$	$1.59 \cdot 10^{-2}$	

Table 3.1: Test the impact in the nme$(\mathbf{S}^*, \hat{\mathbf{S}})$ and time complexity comparing two different implementations of the proposed approach using i) an off-the-shell convex solver (PGL-CVX) and ii) the method in Algorithm 3 (PGL-Alg.3) for different graph sizes. PGL-Alg.3 performs better than PGL-CVX in terms of both time complexity and nme$(\mathbf{S}^*, \hat{\mathbf{S}})$.

two algorithms: 1) solving the optimization in (3.5) with a block coordinate approach where the minimization over $\mathbf{\Theta}$ given $\mathbf{S}$ and the minimization over $\mathbf{S}$ given $\mathbf{\Theta}$ are run using CVX (this algorithm is labeled as PGL-CVX) and 2) employing the efficient scheme outlined in Algorithm 3 (this choice is labeled as PGL-Alg.3). To guarantee that the results are comparable, the nme$(\mathbf{S}^*, \hat{\mathbf{S}})$ is also reported. Examining the listed running times, we observe that as the number of nodes increases, both solvers require more time to estimate the graph (note that the number of variables scales with N^2). More importantly, there exists a noticeable difference between PGL-CVX and PGL-Alg.3. Not only the latter is faster for small graphs, but the gains grow significantly as N increases. Note that the results for graphs with more than $N = 60$ nodes are not reported for PGL-CVX, since the computation time exceeded two hours. In terms of nme$(\mathbf{S}^*, \hat{\mathbf{S}})$, our approach achieves similar (slightly better) results than PGL-CVX. In conclusion, the experiment demonstrates that the efficient implementation described in Section 3.3 is more efficient than readily available convex solvers, rendering it particularly well-suited for large graphs while yielding comparable nme$(\mathbf{S}^*, \hat{\mathbf{S}})$.

3.4.4 Test case 4: Real data scenarios.

Finally, we compare different graph-learning algorithms (including PGL) in the context of two different graph-aware applications. The details and results are provided next.

Stock graph-based clustering from returns. For this real-data experiment, we tested our graph learning algorithm on financial data and further performed a clustering task. Specifically, 40 companies from 4 different sectors of the S&P 500 were selected (10 companies from each sector) and the market data (log returns) of each company in the period 2010-2015 was retrieved. This gives rise to a data matrix of size $\mathbf{X} \in \mathbb{R}^{40 \times 1510}$. In this application, as in many others dealing with graph learning, we do not have access to the ground truth graph. Hence, we cannot quantify the quality of the estimated network directly by using a metric like nme or F1 score. As a result, we need to assess the quality of the estimated graph indirectly, using the graph as input for an ulterior task. In this experiment, we use the graph to estimate the community each company belongs to in an unsupervised manner. More specifically, we implement the following pipeline: 1) estimate several graphs from a subset of the available graph signals, 2) use spectral clustering to group the nodes into 4 communities (as many as sectors were selected), and 3) compute the ratio of incorrectly clustered nodes. To obtain more reliable results, we averaged the clustering errors over 50 realizations in which the subset of graph signals was chosen uniformly at random. The vertical axis of Fig.3.5 represents the normalized clustering error $\frac{1}{50}\sum_{i=1}^{50}\frac{N_w^{(i)}}{N}$ which is computed as the average of the fraction between the number of wrongly clustered nodes N_w and the total number of nodes N over 50 graph realizations. The horizontal axis of Fig. 3.5

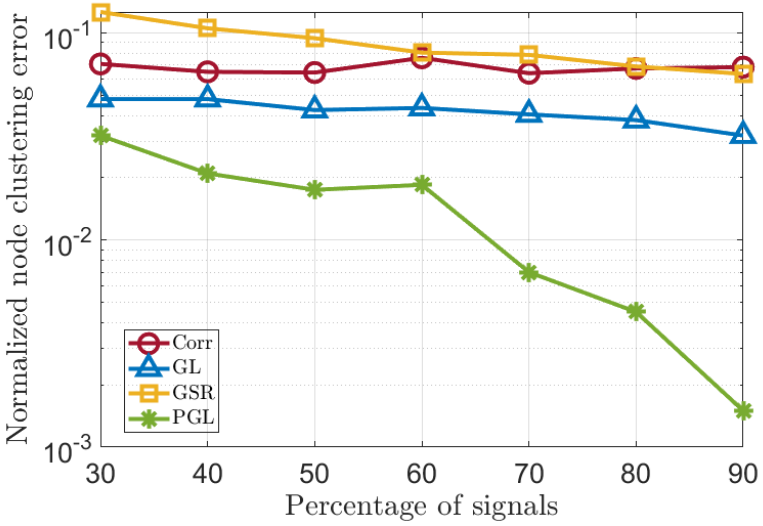

Figure 3.5: Test the impact in terms of normalized node clustering error while increasing the percentage of available signals considering 40 companies of the S&P 500 in the period 2010-2015.

represents the percentages of graph signals used to estimate $\hat{\mathbf{S}}$. Results are provided for 4 graph-learning approaches: PGL, GL, GSR, and "Corr", which estimates the graph as a thresholded version of the correlation matrix. The idea is that companies from the same sector have stronger ties among them and, as a result, when running a graph-based clustering method, the 4 sectors should arise. Based on the results presented in Fig.3.5, it can be observed that PGL outperforms the other alternatives and additionally, as the number of available signals increases, the clustering error for PGL drops significantly. The superiority of PGL may indicate that considering more complex relationships among stocks (beyond the simple correlations considered in "Corr" or the partial correlations considered in GL) is a better model to understand the dependencies between log-returns in the stock market. On the other hand, GSR shows high clustering error with a limited number of samples, but it improves as the number of available samples increases. This observation aligns with our earlier discussion in Section 3.2, where we highlighted that this particular model offers greater generality but necessitates a substantial number of samples to accurately estimate the graph.

Learning sequential graphs for investing. This experiment deals with a different real-world problem and dataset. We still look at stocks, but consider now the close price of the 7 FAAMUNG[2] companies from Jul 2019 to May 2020. The goal is to design an investment strategy to maximize the benefits using as input a graph describing the relationships among the companies. Inspired by the approach in [79], we first build a graph, analyze its connectivity and then, invest (or not) in a stock according to the graph connectivity. To be more specific, we use the close price to estimate multiple 7×7 adjacency matrices. We estimate a total of 200 matrices, where each adjacency matrix (graph) is estimated in a rolling window fashion. The window consists of 30 consecutive days and for each graph estimation, we shift the window one day at a time. These graph estimations help to visualize how the graph topology changes during this time period. The finding in [79] is that big changes in the graph connectivity indicate opportunities to invest. To that end, we keep track of the algebraic connectivity value, which is the second smallest eigenvalue (λ_2) of the estimated Laplacian matrix. The lower the value, the less connected the graph is and, as a result, the easier breaking the graph into multiple components is. Fig.3.6 shows the value

[2]Facebook, Amazon, Apple, Microsoft, Uber, Netflix, and Google

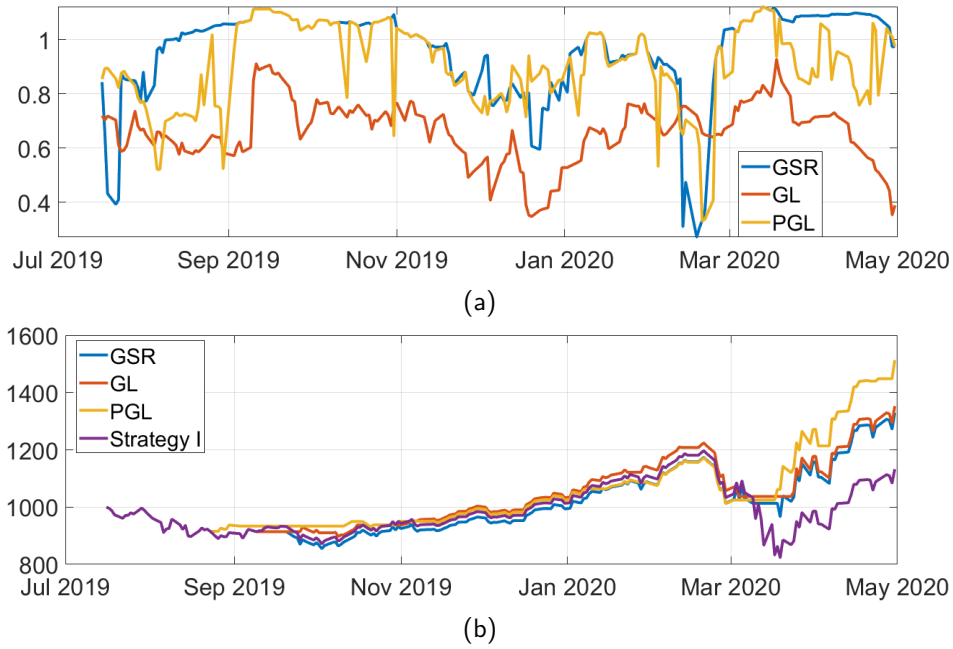

Figure 3.6: (a) Value of the algebraic connectivity indicator (λ_2) associated with each one of the 200 estimated graphs from July 2019 to May 2020 using three different graph learning approaches. (b) Benefits obtained from applying four different investment strategies based on the estimated algebraic connectivity indicator for each of the considered approaches.

of λ_2 for each of the 200 considered graphs (each associated with a 30-day period). Then, using the approach in [79] we invest only if λ_2 is below a fixed threshold. We learn the graph using 3 methods (GL, GSR, and PGL) and, for each of them, we select the best possible threshold (the one that maximizes the benefits). Correlation was not used here due to its poor performance. The results of applying the graph-connectivity-based investment strategy to the graphs estimated with each of the algorithms are shown in Fig. 3.6. The blue line labeled as "Strategy I" represents the benefits of investing every day the available amount and is used as a baseline. By analyzing the results obtained, we can observe that: i) the graph-based strategies outperform (gain more money than) the baseline; ii) the strategies based on GL and GSR provide similar gains; and iii) the strategy based on PGL yields the highest gains. This provides additional validation for the graph-learning methodology proposed in this chapter.

3.5 Conclusions

This chapter has introduced PGL, a novel scheme for learning a graph from nodal signals, with our key contribution being the modeling of the signals as Gaussian and stationary on the graph. This approach opens the door to a graph-learning formulation that leverages the advantages of GL (needing a relatively small number of signals to get a good estimation of the graph structure) while encompassing a more comprehensive model (because it handles cases where the precision matrix can be any polynomial form of the sought graph). Given the increased complexity and nonconvex nature of the resulting optimization problem, we have developed a low-complexity algorithm that alternates between estimating the graph and precision matrices and have characterized its convergence to a block coordinatewise minimum. To assess its efficacy, we have conducted numerical simulations comparing PGL with various alternatives, using both synthetic and real data. The results have showcased the benefits of our approach, motivating the adoption and further investigation of the

proposed graph-learning methodology.

3.6 Appendix: Computations of projections

We present the details about how to compute the projections $\mathcal{P}_{\mathcal{S}_A}$ and $\mathcal{P}_{\mathcal{S}_B}$.

The computation of $\mathcal{P}_{\mathcal{S}_A}$ is straightforward as follows:

$$\mathcal{P}_{\mathcal{S}_A}(\mathbf{A}) = (\mathbf{A} + \mathbf{A}^\top)/2. \tag{3.33}$$

The projection $\mathcal{P}_{\mathcal{S}_B}$ is defined as the minimizer of the optimization problem as follows:

$$\mathcal{P}_{\mathcal{S}_B}(\mathbf{A}) := \underset{\mathbf{X}\in\mathcal{S}_B}{\arg\min}\, \frac{1}{2}\|\mathbf{X} - \mathbf{A}\|_F^2. \tag{3.34}$$

To compute the projection $\mathcal{P}_{\mathcal{S}_B}$, we solve Problem (3.34) row by row. For the j-th row, we solve the following problem,

$$\begin{array}{ll} \underset{\mathbf{x}\in\mathbb{R}^N}{\min} & \frac{1}{2}\|\mathbf{x} - \mathbf{a}\|^2, \\ \text{s. t.} & \mathbf{x}^\top\mathbf{1} \geq 1,\, x_j = 0,\, \mathbf{x}_{\setminus j} \geq \mathbf{0}, \end{array} \tag{3.35}$$

where $\mathbf{a} \in \mathbb{R}^N$ contains all entries of the j-th row of $\mathbf{A}$, x_j denotes the j-th entry of $\mathbf{x}$, and $\mathbf{x}_{\setminus j} \in \mathbb{R}^{N-1}$ contains all entries of $\mathbf{x}$ except the j-th one.

Let $\hat{\mathbf{x}}$ denote the optimal solution of Problem (3.35). Proposition 3.1 below, proved in Appendix 3.7, presents the optimal solution of Problem (3.35).

Proposition 3.1. *The optimal solution* $\hat{\mathbf{x}}$ *to Problem* (3.35) *can be obtained as follows:*

- *If* $\sum_{i\neq j} \max(a_i, 0) \geq 1$, *then* $\hat{x}_j = 0$, *and* $\hat{x}_i = \max(a_i, 0)$, *for* $i \neq j$.
- *If* $\sum_{i\neq j} \max(a_i, 0) < 1$, *then* $\hat{x}_j = 0$, *and* $\hat{x}_i = \max(a_i + \phi, 0)$, *for* $i \neq j$, *where* ϕ *satisfies* $\sum_{i\neq j} \max(a_i + \phi, 0) = 1$.

Several efficient approaches have been developed to tackle the piecewise linear equation $\sum_{i\neq j} \max(a_i + \phi, 0) = 1$. Among these, the sorting-based method described in [80] is noteworthy. Central to this method is the sorting of the vector $\mathbf{a}$, which constitutes the most computationally intensive step, generally necessitating $\mathcal{O}(N \log N)$ operations.

3.7 Appendix: Proof of Proposition 3.1

Proof. The Lagrangian of the optimization in (3.35) is

$$L(\mathbf{x}, u, \mathbf{v}) = \frac{1}{2}\|\mathbf{x} - \mathbf{a}\|^2 - u(\mathbf{x}^\top\mathbf{1} - 1) - \langle \mathbf{v}_{\setminus j}, \mathbf{x}_{\setminus j}\rangle + v_j x_j,$$

where $u \in \mathbb{R}$ and $\mathbf{v} \in \mathbb{R}^N$ are Karush-Kuhn-Tucker (KKT) multipliers. Let $(\hat{\mathbf{x}}, \hat{u}, \hat{\mathbf{v}})$ be the primal and dual optimal point. Then $(\hat{\mathbf{x}}, \hat{u}, \hat{\mathbf{v}})$ must satisfy the KKT system:

$$\hat{x}_i - a_i - \hat{u} - \hat{v}_i = 0, \quad \text{for } i \neq j; \tag{3.36}$$

$$\hat{x}_j - a_j - \hat{u} + \hat{v}_j = 0; \tag{3.37}$$

$$\hat{x}_i \geq 0,\ \hat{v}_i \geq 0,\ \hat{v}_i \hat{x}_i = 0, \quad \text{for } i \neq j; \tag{3.38}$$

$$\hat{x}_j = 0,\ \hat{u} \geq 0,\ \hat{\mathbf{x}}^T \mathbf{1} \geq 1; \tag{3.39}$$

$$\hat{u}(\hat{\mathbf{x}}^T \mathbf{1} - 1) = 0; \tag{3.40}$$

Therefore, for any $i \neq j$, it holds that $\hat{x}_i = a_i + \hat{u} + \hat{v}_i$. Then we obtain the following results:

- If $a_i + \hat{u} < 0$, then $\hat{v}_i = -a_i + \hat{u}$ and $\hat{x}_i = 0$, following from the fact that $\hat{v}_i \hat{x}_i = 0$ and $\hat{x}_i \leq 0$.
- If $a_i + \hat{u}_r \geq 0$, then $\hat{v}_i = 0$. This can be obtained as follows: if $\hat{x}_i = 0$, then $\hat{v}_i = -(a_i + \hat{u}) \leq 0$. Since $\hat{v}_i \geq 0$, one has $\hat{v}_i = 0$; On the other hand, if $\hat{x}_i \neq 0$, then $\hat{v}_i = 0$, following from the fact that $\hat{v}_i \hat{x}_i = 0$. As a result, we get $\hat{x}_i = a_i + \hat{u}$.

Overall, we obtain that

$$\hat{x}_j = 0, \quad \text{and} \quad \hat{x}_i = \max(a_i + \hat{u}, 0), \quad \text{for all } i \neq j. \tag{3.41}$$

On the other hand, $\hat{\mathbf{x}}$ and $\hat{u}$ satisfy that $\hat{\mathbf{x}}^T \mathbf{1} \geq 1$, $\hat{u} \geq 0$, and $\hat{u}(\hat{\mathbf{x}}^T \mathbf{1} - 1) = 0$. To this end, we can obtain the following results:

- If $\sum_{i \neq j} \max(a_i, 0) \geq 1$, then $\hat{u} = 0$, indicating that $\hat{x}_i = \max(a_i, 0)$, for any $i \neq j$.
- If $\sum_{i \neq j} \max(a_i, 0) < 1$, then $\hat{u} \neq 0$. This is because $\hat{u} = 0$ will result in $\hat{\mathbf{x}}^\top \mathbf{1} < 1$, which does not satisfy the KKT system. Together with the KKT condition that $\hat{u}(\hat{\mathbf{x}}^T \mathbf{1} - 1) = 0$, one has $\hat{\mathbf{x}}^\top \mathbf{1} = 1$. Therefore, one obtains that, for any $i \neq j$, $\hat{x}_i = \max(a_i + \hat{u}, 0)$, where $\hat{u}$ is chosen such that $\sum_{i \neq j} \max(a_i + \hat{u}, 0) = 1$.

We note that the ϕ in Proposition 3.1 is exactly the dual optimal point $\hat{u}$. □

3.8 Appendix: Proof of Theorem 3.1

Proof. The convergence result stated in Theorem 3.1 is based on the framework established by Theorem 2.3 in [81]. To demonstrate the validity of Theorem 3.1, it suffices to establish that the conditions and assumptions of Theorem 2.3 are satisfied in our context. Our approach to block updates aligns with the procedure delineated in equation (1.3a) of [81].

We first verify the conditions the requisite conditions of Theorem 2.3 in [81] are met. Specifically, Theorem 2.3 stipulates that the objective function, along with the feasible set of the optimization problem, should exhibit *block multiconvexity*. For Problem (3.5), the objective function f is convex with respect to each of the blocks $\boldsymbol{\Theta}$ and $\mathbf{S}$ when the other block is fixed, a property that defines *block multiconvexity* as per [81]. Moreover, the function f is strongly convex with respect to both $\boldsymbol{\Theta}$ and $\mathbf{S}$.

The feasibility constraints of Problem (3.5) form a set $\mathcal{X}$ that satisfies the criteria for *block multiconvexity* as defined in [81]. This is due to the convexity of the individual set maps $\mathcal{X}_{\mathbf{\Theta}}$ and $\mathcal{X}_{\mathbf{S}}$. The set map $\mathcal{X}_{\mathbf{\Theta}}$ is defined as

$$\mathcal{X}_{\mathbf{\Theta}} = \{\mathbf{\Theta} \in \mathbb{R}^{N\times N} \,|\mathbf{\Theta} \succeq \mathbf{0}, \| \mathbf{\Theta}\mathbf{S} - \mathbf{S}\mathbf{\Theta}\|_F \leq \delta\} \tag{3.42}$$

for some given $\mathbf{S}$, and similarly, the set map $\mathcal{X}_{\mathbf{S}}$ is defined as

$$\mathcal{X}_{\mathbf{S}} = \{\mathbf{S} \in \mathbb{R}^{N\times N} \,|\, \mathbf{S} \in \mathcal{S}, \|\mathbf{\Theta}\mathbf{S} - \mathbf{S}\mathbf{\Theta}\|_F \leq \delta\} \tag{3.43}$$

for some given $\mathbf{\Theta}$. Consequently, the optimization subproblems with respect to $\mathbf{\Theta}$ and $\mathbf{S}$ in Problem (3.5) are convex.

We now validate the assumptions required by Theorem 2.3 in [81] within the context of our setting. Specifically, it is required that the objective function f is bounded below over the feasible set $\mathcal{X}$, that is, $\inf_{(\mathbf{\Theta},\mathbf{S})\in\mathcal{X}} f(\mathbf{\Theta},\mathbf{S}) > -\infty$. This is indeed the case here, because the term $\rho\|\mathbf{S}\|_1 + \frac{\eta}{2}\|\mathbf{S}\|_F^2$ is nonnegative. Additionally, the function $-\log(\det(\mathbf{\Theta})) + \mathrm{tr}(\hat{\mathbf{C}}\mathbf{\Theta})$ attains a finite infimum when $\hat{\mathbf{C}}$ is positive definite under Assumption 3.2. To see this, first observe that $\det(\mathbf{\Theta}) \leq \|\mathbf{\Theta}\|_2^N$, where $\|\mathbf{\Theta}\|_2$ is the largest eigenvalue of $\mathbf{\Theta}$. Consequently, $-\log\det(\mathbf{\Theta}) \geq -N\log(\|\mathbf{\Theta}\|_2)$. Further, since $\mathrm{tr}(\hat{\mathbf{C}}\mathbf{\Theta}) \geq \gamma\mathrm{tr}(\mathbf{\Theta}) \geq \gamma\|\mathbf{\Theta}\|_2$, with $\gamma > 0$ being the smallest eigenvalue of $\hat{\mathbf{C}}$, we obtain

$$-\log\det(\mathbf{\Theta}) + \mathrm{tr}(\hat{\mathbf{C}}\mathbf{\Theta}) \geq -N\log(\|\mathbf{\Theta}\|_2) + \gamma\|\mathbf{\Theta}\|_2, \tag{3.44}$$

which indeed has a finite minimum. Thus, we conclude that $\inf_{(\mathbf{\Theta},\mathbf{S})\in\mathcal{X}} -\log\det(\mathbf{\Theta}) + \mathrm{tr}(\hat{\mathbf{C}}\mathbf{\Theta}) > -\infty$.

Furthermore, the existence of *block coordinatewise minimizers* is assured by the compactness of the feasible set. Moreover, Theorem 2.3 in [81] stipulates that set maps change continuously during iterations. Referring to (3.42) and (3.43), it is clear that the only constraint that changes through iterations is $\|\mathbf{\Theta}\mathbf{S} - \mathbf{S}\mathbf{\Theta}\|_F \leq \delta$, while the other constraints, $\mathbf{\Theta} \succeq \mathbf{0}$ and $\mathbf{S} \in \mathcal{S}$, remain constant. Given that $\|\mathbf{\Theta}\mathbf{S} - \mathbf{S}\mathbf{\Theta}\|_F$ is a continuous function with respect to both $\mathbf{\Theta}$ and $\mathbf{S}$, the set maps indeed change continuously, satisfying the theorem's conditions.

These verifications above have demonstrated that the conditions and assumptions of Theorem 2.3 are satisfied in our context, completing the proof. □

Chapter 4

Graph Learning from Smooth and Stationary Graph Signals with Hidden Nodes

The main goal of graph learning approaches is to estimate the graph structure from data of the observed nodes. This is crucial in various fields, but a significant challenge arises when some nodes are hidden or unobserved. This chapter introduces a novel approach that considers the presence of hidden variables in a network topology inference problem while assuming smoothness and/or stationarity for the graph signals. The approach involves constrained optimization, block matrix factorization, and relaxation techniques to adapt to the presence of hidden nodes. Overall, this chapter addresses the challenge of hidden variables in network topology inference within the GSP framework, offering potential insights and directions for further research.

4.1 Introduction

As we discussed in the previous chapter, datasets with non-Euclidean support have become prominent, leading to the adoption of graph-based techniques to address various challenges in information processing. This approach has found success in diverse applications attracting interest from researchers across statistics, machine learning, and signal processing domains. Although networks may exist as physical entities, oftentimes they are abstract mathematical representations with nodes describing variables and links describing pairwise relationships between them. More importantly for this chapter, such relationships may not always be known a priori. In such cases, the graph needs to be learned from a set of node observations under the fundamental assumption that there is a relationship between the properties of the observed signals and the topology of the sought graph. The described task represents the network topology inference problem [4, 37, 39, 40, 58, 82], discussed in Section 2.5. Noteworthy approaches include correlation networks [13], partial correlations and (Gaussian) Markov random fields [13, 59, 67, 83], sparse structural equation models [41, 84], GSP-based approaches [4, 37, 39, 61, 85], as well as their non-linear generalizations [60, 86], to name a few.

The standard network-inference approach in the aforementioned works is to assume that observations from all the nodes of the graph are available. In certain environments, however, only observations from a subset of nodes are available, with the remaining nodes being unobserved or *hidden*. The existence of hidden/latent nodes constitutes a relevant and challenging problem since

closely related values from two observed nodes may be explained not only by an edge between the two nodes but by a third latent node connected to both of them. Moreover, because there are no observations from the hidden nodes, modeling their influence renders the network inference problem substantially more challenging and ill-posed. Except for direct pairwise methods, which can be trivially generalized to the setup at hand, most of the existing approaches require important modifications to deal with hidden nodes. Network-inference works that have looked at the problem of hidden variables include examples in the context of Gaussian graphical model selection [63, 87], inference of linear Bayesian networks [88], nonlinear regression [89], or brain connectivity [64] to name a few. Nonetheless, there are still a number of effective network-inference methods (including most in the context of GSP) that have not considered the presence of latent unobserved nodes.

Motivated by the previous discussion, in this chapter, we approach the problem of network topology inference with hidden variables by leveraging two fundamental concepts of the GSP framework: smoothness [55] and stationarity [48, 90]. Recall that, as mentioned in Section 2.4, a signal being smooth on a graph implies that the signal values at two neighboring nodes are close so that the signal varies slowly across the graph. This fairly general assumption has been successfully exploited to infer the topology of the graph when values from all nodes are observed [85, 91, 92]. From a different perspective, assuming that a random process is stationary on a graph is tantamount to assuming that the covariance matrix of the random process is a polynomial of the GSO, as discussed in Section 2.4. This property has been leveraged in the context of network inference to develop new algorithms and establish important links between graph stationarity and classical correlation and partial-correlation approaches [4, 54, 93]. Although the assumptions of smoothness and stationarity have been successfully adopted in the context of the network-topology inference problem, a formulation robust to the presence of hidden variables is still missing. To fill this gap, this chapter builds on our previous work [42] and investigates how the presence of the hidden variables impacts the classical definitions of graph smoothness and stationarity. Then, it formulates the network-recovery problem as a constrained optimization that accounts explicitly for the modified definitions. A key in our formulation is the consideration of a block matrix factorization approach and exploitation of the low rankness and the sparsity pattern present in the blocks related to hidden variables. A range of formulations are presented and suitable (convex and non-convex) relaxations to deal with the sparsity and low-rank terms are considered. While our focus is to learn the connections among observed nodes, some of our approaches also reveal information related to links involving hidden nodes. A further investigation of this matter is left as future work.

Related work and contributions. Early methods in graph topology inference, accounting for hidden nodes, were initially introduced in [63], assuming observations follow a GMRF model. Other related works that came later include Bayesian network inference [88], nonlinear regression [89], and brain connectivity studies [64]. However, many effective network inference methods, particularly in the context of GSP, have overlooked the presence of latent unobserved nodes. In this chapter, we extend our prior work from [42] by (i) proposing different approaches for network topology inference with hidden variables in scenarios where the available signals exhibit smoothness and/or stationarity within the sought graph and (ii) providing convex algorithms for addressing the proposed problems, and (iii) showing theoretical convergence guarantees of the proposed method to a stationary point. Furthermore, the expressive capabilities of the GMRF model are constrained when observations possess a more intrinsic structure. Addressing these challenges, our goal is to devise broader methods for topology inference with hidden variables. This involves closing the gap in the context of GSP and understanding the influence of hidden nodes in scenarios where graph signals exhibit intrinsic characteristics such as smoothness, stationarity, or both, which are investigated in this work.

To summarize, our main contributions are:

- We analyze the influence of hidden variables on graph smoothness and graph stationarity.
- We propose several optimization approaches to solve the topology inference problem with hidden variables when the observed signals are smooth, stationary or both in the sought graph.
- We propose an iterative algorithm for estimating the network structure and also theoretical guarantees of convergence to a stationary point.
- We present an extensive evaluation of the proposed models through both synthetic and real experiments showing the benefits of the proposed approaches compared to the state-of-the-art alternatives.

Outline. The remainder of this chapter is organized as follows. We introduce in Section 4.2 the task of inferring the graph structure in the presence of hidden nodes. In Section 4.3, Section 4.4, and Section 4.5 we present our proposed optimization problem that accounts for the presence of hidden nodes when the available graph signals are smooth, stationary, and both smooth and stationary, respectively. The performance of the proposed methods is validated and compared with other alternatives from the literature by several synthetic and real-world experiments in Section 4.6. In Section 4.7 a concluding discussion is provided about the work presented in this chapter. Finally, we provide theoretical convergence guarantees for the solution obtained by our method to a stationary point in Section 4.8.

4.2 Influence of hidden variables in the network topology inference problem

The current section is devoted to formally posing the network topology inference problem when only observations from a subset of nodes of the graph are available. We present a general formulation and highlight the influence of the hidden variables.

Denote as $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_R] \in \mathbb{R}^{N \times R}$ the collection of R signals defined on top of the *unknown* graph $\mathcal{G}$ with N nodes. Then, we consider that we only observe the values of $\mathbf{X}$ from a subset of nodes $\mathcal{O} \subset \mathcal{V}$ with cardinality $O < N$. In contrast, the values corresponding to the remaining $H = N - O$ nodes in the subset $\mathcal{H} = \mathcal{V} \setminus \mathcal{O}$ stay hidden[1]. For simplicity and without loss of generality, let the observed nodes correspond to the first O nodes of the graph, so the values of the given signals at $\mathcal{O}$ are collected in the submatrix $\mathbf{X}_\mathcal{O} \in \mathbb{R}^{O \times R}$, which is formed by the first O rows of the matrix $\mathbf{X}$. As explained in the previous section, these observations can be used to form the sample covariance matrix. When doing so, it is important to notice the matrices $\mathbf{S} \in \mathbb{R}^{N \times N}$ and $\hat{\mathbf{C}} \in \mathbb{R}^{N \times N}$, which respectively represent the GSO and the sample covariance matrix associated with the full graph $\mathcal{G}$, and the signals $\mathbf{X}$, present the following block structure

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_\mathcal{O} \\ \mathbf{X}_\mathcal{H} \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} \mathbf{S}_\mathcal{O} & \mathbf{S}_{\mathcal{O}\mathcal{H}} \\ \mathbf{S}_{\mathcal{H}\mathcal{O}} & \mathbf{S}_\mathcal{H} \end{bmatrix}, \ \hat{\mathbf{C}} = \begin{bmatrix} \hat{\mathbf{C}}_\mathcal{O} & \hat{\mathbf{C}}_{\mathcal{O}\mathcal{H}} \\ \hat{\mathbf{C}}_{\mathcal{H}\mathcal{O}} & \hat{\mathbf{C}}_\mathcal{H} \end{bmatrix}. \tag{4.1}$$

The $O \times O$ matrix $\mathbf{S}_\mathcal{O}$ denotes the GSO describing the connections between the observed nodes, while the remaining blocks model the edges involving hidden nodes. Similarly, $\hat{\mathbf{C}}_\mathcal{O} = \frac{1}{R}\mathbf{X}_\mathcal{O}\mathbf{X}_\mathcal{O}^\top$

[1]With a slight abuse of notation, we use H to denote the number of hidden nodes and the square matrix $\mathbf{H}$ to denote a generic graph filter.

denotes the sample covariance of the observed signals, and the other blocks denote the submatrices of $\hat{\mathbf{C}}$ involving signal values from the hidden nodes. Since $\mathcal{G}$ is undirected, both $\mathbf{S}$ and $\hat{\mathbf{C}}$ are symmetric, and thus, $\mathbf{S}_{\mathcal{H}\mathcal{O}} = \mathbf{S}_{\mathcal{O}\mathcal{H}}^\top$ and $\hat{\mathbf{C}}_{\mathcal{H}\mathcal{O}} = \hat{\mathbf{C}}_{\mathcal{O}\mathcal{H}}^\top$.

With the previous definitions in place, the problem of network topology inference/graph learning in the presence of hidden variables is formally introduced next.

Problem 4.1. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with N nodes and GSO $\mathbf{S} \in \mathbb{R}^{N\times N}$, and suppose that $\{\mathcal{V}, \mathcal{E}, N, \mathbf{S}\}$ are all unknown. Given the nodal subset $\mathcal{O} \subset \mathcal{V}$ with cardinality $|\mathcal{O}| = O$, and the observations $\mathbf{X}_\mathcal{O} \in \mathbb{R}^{O\times R}$ corresponding to the values of R graph signals observed at the nodes in $\mathcal{O}$, find the underlying graph structure encoded in $\mathbf{S}_\mathcal{O} \in \mathbb{R}^{O\times O}$ under the assumptions that:*
(AS1) The number of hidden variables (nodes) is substantially smaller than the number of observed nodes, i.e., $O \lessapprox N$; and
(AS2) There exists a (known) property relating the full graph signals $\mathbf{X} \in \mathbb{R}^{N\times R}$ to the GSO $\mathbf{S}$.

Despite having observations from O nodes, there are still $H = N - O$ nodes that remain unseen and influence the observed signals $\mathbf{X}_\mathcal{O}$, rendering the inference problem challenging and severely ill-conditioned. To make the problem more tractable, (AS1) ensures that the number of hidden variables is small. Assumption (AS2) is more generic and establishes that there is a known relationship between the graph signals $\mathbf{X}$ and the full graph $\mathbf{S}$. The particular relationship is further developed in the following sections, where we assume that $\mathbf{X}$ is either smooth (Section 4.3) or stationary (Section 4.4) on $\mathbf{S}$. The key issue to address is how (AS2), which involves the full signals and GSO, translates to the submatrices $\mathbf{X}_\mathcal{O}$, $\mathbf{S}_\mathcal{O}$, and $\mathbf{C}_\mathcal{O}$ in (4.1).

Given the above considerations, a general formulation to solve Problem 4.1 is as follows

$$\begin{aligned} \hat{\mathbf{S}}_\mathcal{O} = \operatorname{argmin}_{\mathbf{S}_\mathcal{O}} \; & f(\mathbf{S}_\mathcal{O}) \\ \text{s. t.} \quad & \mathbf{X}_\mathcal{O} \in \mathcal{X}(\mathbf{S}), \\ & \mathbf{S}_\mathcal{O} \in \mathcal{S}, \end{aligned} \tag{4.2}$$

where $f(\cdot)$ is a (preferably convex) function that promotes desirable properties on the sought graph. Typical examples include the ℓ_1 norm, the Frobenius norm, the spectral radius, or linear combinations of those [40]. Note that the first constraint in (4.2) (referred to as observation constraint) takes into account that $\mathcal{X}$ involves the full matrices $\mathbf{X}$ and $\mathbf{S}$ but only $\mathbf{X}_\mathcal{O}$ is observed. It is also important to remark that, as will be apparent in the following sections, for observations that are either smooth or stationarity in the graph, the constraint $\mathbf{X}_\mathcal{O} \in \mathcal{X}(\mathbf{S})$ can be reformulated in terms of the (sample) covariance matrices $\hat{\mathbf{C}}_\mathcal{O} = \frac{1}{R}\mathbf{X}_\mathcal{O}\mathbf{X}_\mathcal{O}^\top$ and $\mathbf{C}_\mathcal{O} = \mathbb{E}\left[\mathbf{x}_\mathcal{O}\mathbf{x}_\mathcal{O}^\top\right]$. Regarding the second constraint in (4.2), the set $\mathcal{S}$ collects the requirements for $\mathbf{S}$ to be a specific type of GSO. As mentioned in Section 2.5, a typical example of $\mathbf{S}$ being a specific type of GSO is belonging to the set of adjacency matrices

$$\mathcal{A} := \{A_{ij} \geq 0;\ \mathbf{A} = \mathbf{A}^\top;\ A_{ii} = 0;\ \mathbf{A}\mathbf{1} \geq \mathbf{1}\}, \tag{4.3}$$

where we require the GSO to have non-negative weights, be symmetric and have no self-loops, and the last constraint rules out the trivial 0 solution by imposing that every node has at least one neighbor. Analogously, the set of combinatorial Laplacian matrices is

$$\mathcal{L} := \{L_{ij} \leq 0 \text{ for } i \neq j;\ \mathbf{L} = \mathbf{L}^\top;\ \mathbf{L}\mathbf{1} = \mathbf{0};\ \mathbf{L} \succeq 0\}, \tag{4.4}$$

where we require the GSO to be a PSD matrix, have non-positive off-diagonal values, have positive entries on its diagonal, and have the constant vector as an eigenvector (i.e, the sum of the entries

of each row to be zero). Lastly, we want to stress that the objective $f(\mathbf{S}_{\mathcal{O}})$ and the constraint $\mathbf{S}_{\mathcal{O}} \in \mathcal{S}$ can be alternatively formulated based on the full GSO $\mathbf{S}$, provided that we know that the structural properties (for instance sparsity in the objective and positive entries in the constraints) hold also for the non-observed parts of $\mathbf{S}$. Such an approach is suitable when the interest goes beyond $\mathbf{S}_{\mathcal{O}}$ and spans the estimation of the links involving the nodes in $\mathcal{H}$.

Hidden variables in correlation and partial-correlation networks: Before discussing our specific solutions to Problem 4.1, a relevant question is how classical network topology inference approaches (namely correlation and partial-correlation networks) handle the problem of latent nodal variables. The so-called direct methods consider that a link between nodes i and j exists based only on a pairwise similarity metric between the signals observed at i and j. Within this class of methods, correlation networks set the similarity metric to the correlation and, as a result, $\mathbf{S}$ corresponds to a (thresholded) version of $\mathbf{C}$. Given their simplicity, the generalization of direct methods to setups where hidden variables are present is straightforward and simply given by $\mathbf{S}_{\mathcal{O}} = \hat{\mathbf{C}}_{\mathcal{O}}$. Nevertheless, a high correlation between two nodes can be due to global network effects rather than to the direct influence among pairs of neighbors, calling for more involved network topology inference methods. To that end, partial-correlation methods, including the celebrated GL algorithm [3], propose estimating the graph as a matrix of partial correlation coefficients, which boils down to assuming that the connectivity patterns can be identified as $\mathbf{S} = \mathbf{C}^{-1}$, with $\mathbf{C}^{-1}$ being known as the precision matrix. When hidden variables are present, the submatrix of the precision matrix is given by $\mathbf{C}_{\mathcal{O}}^{-1} = \mathbf{S}_{\mathcal{O}} - \mathbf{B}$, with $\mathbf{B} = \mathbf{S}_{\mathcal{O}\mathcal{H}}\mathbf{S}_{\mathcal{H}}^{-1}\mathbf{S}_{\mathcal{H}\mathcal{O}}$ being a low-rank matrix since $H \ll O$. Leveraging this structure, the authors in [63] modified the GL algorithm to deal with hidden variables via a maximum-likelihood estimator augmented with a nuclear-norm regularizer to promote low rankness in $\mathbf{B}$. The resulting algorithm is known as latent variable graphical Lasso (LVGL) and is given by

$$\begin{aligned}\max_{\mathbf{S}_{\mathcal{O}}-\mathbf{B}\succeq\mathbf{0},\,\mathbf{B}\succeq\mathbf{0}} \log\det(\mathbf{S}_{\mathcal{O}}-\mathbf{B}) - \operatorname{trace}(\hat{\mathbf{C}}_{\mathcal{O}}(\mathbf{S}_{\mathcal{O}}-\mathbf{B})) \\ -\lambda_1\|\mathbf{S}_{\mathcal{O}}\|_1 - \lambda_2\|\mathbf{B}\|_*,\end{aligned} \tag{4.5}$$

where $\hat{\mathbf{C}}_{\mathcal{O}}$ represents the sample covariance of the observed data and λ_1 and λ_2 are regularization constants [63].

Rather than assuming that the relation between $\mathbf{X}$ and $\mathbf{S}$ postulated in (AS2) is given by either correlations or partial-correlations, this chapter looks at setups where the operational assumption is that the observed signals are: i) smooth on the graph; ii) stationary on the graph; and iii) both smooth and stationary. Sections 4.3-4.5 deal with each of those three setups. Section 4.6 evaluates numerically the performance of the developed algorithms and compares it with that of classical correlation and LVGL schemes.

4.3 Network topology inference from smooth signals with hidden vairables

In this section, we address Problem 4.1 by particularizing (4.2) to the case of the signals $\mathbf{X}$ being smooth on $\mathcal{G}$.

As explained in Section 2.4, a natural way of measuring the smoothness of (a set of) graph signals is to leverage the graph Laplacian and compute their LV as $\frac{1}{R}\text{tr}(\mathbf{X}\mathbf{X}^\top\mathbf{L})$ [cf. (2.5)]. As a result, in this section, we set $\mathbf{S} = \mathbf{L}$ and focus on $\hat{\mathbf{C}} = \frac{1}{R}\mathbf{X}\mathbf{X}^\top$. Recall that, due to the existence of hidden variables, the whole covariance matrix is not observed. To account for this and leveraging the block definition of $\hat{\mathbf{C}}$ and $\mathbf{S}$ introduced in (4.1), we can rewrite the LV of our dataset as

$$\text{tr}(\hat{\mathbf{C}}\mathbf{L}) = \text{tr}(\hat{\mathbf{C}}_{\mathcal{O}}\mathbf{L}_{\mathcal{O}}) + 2\text{tr}(\hat{\mathbf{C}}_{\mathcal{O}\mathcal{H}}\mathbf{L}_{\mathcal{O}\mathcal{H}}^\top) + \text{tr}(\hat{\mathbf{C}}_{\mathcal{H}}\mathbf{L}_{\mathcal{H}}), \tag{4.6}$$

where only $\hat{\mathbf{C}}_{\mathcal{O}} = \frac{1}{R}\mathbf{X}_{\mathcal{O}}\mathbf{X}_{\mathcal{O}}^{\top}$ is assumed to be known and the influence of the hidden variables in the LV has been made explicit.

Although the block-wise smoothness presented in (4.6) could be directly employed to approach the network-topology inference as an optimization problem, most of the submatrices are not known and need to be estimated. Incorporating the terms $\mathbf{C}_{\mathcal{O}\mathcal{H}}\mathbf{L}_{\mathcal{O}\mathcal{H}}^{\top}$ and $\mathbf{C}_{\mathcal{H}}\mathbf{L}_{\mathcal{H}}$ would directly render the problem non-convex. To circumvent this issue, we lift the problem by defining the matrix $\mathbf{K} := \mathbf{C}_{\mathcal{O}\mathcal{H}}\mathbf{L}_{\mathcal{O}\mathcal{H}}^{\top} \in \mathbb{R}^{O\times O}$. Since (AS1) guarantees that $\text{rank}(\mathbf{K}) \leq H \ll O$, we exploit the low-rank structure of the matrix $\mathbf{K}$ in our formulation. Correspondingly, we also define the matrix $\mathbf{R} := \mathbf{C}_{\mathcal{H}}\mathbf{L}_{\mathcal{H}} \in \mathbb{R}^{H\times H}$ and note that, since $\mathbf{R}$ is the product of two PSD matrices, it has positive eigenvalues and, as a result, it holds that $\text{tr}(\mathbf{R}) \geq 0$.

With these considerations in mind, the network topology inference problem from smooth signals is formulated as

$$\begin{aligned}\min_{\mathbf{L}_{\mathcal{O}},\mathbf{K},\mathbf{R}} \quad & \text{tr}(\mathbf{C}_{\mathcal{O}}\mathbf{L}_{\mathcal{O}})+2\text{tr}(\mathbf{K})+\text{tr}(\mathbf{R})+\alpha\|\mathbf{L}_{\mathcal{O}}\|_{F,off}^2 \\ & -\beta\log(\text{diag}(\mathbf{L}_{\mathcal{O}})) + \gamma\|\mathbf{K}\|_* \\ \text{s. t.} \quad & \text{tr}(\mathbf{C}_{\mathcal{O}}\mathbf{L}_{\mathcal{O}}) + 2\text{tr}(\mathbf{K}) + \text{tr}(\mathbf{R}) \geq 0, \\ & \text{tr}(\mathbf{R}) \geq 0, \\ & \mathbf{L}_{\mathcal{O}} \in \bar{\mathcal{L}},\end{aligned} \tag{4.7}$$

where $\|\cdot\|_{F,off}^2$ denotes the Frobenius norm excluding the elements of the diagonal. This term, together with $\log(\text{diag}(\mathbf{L}_{\mathcal{O}}))$, serves to control the sparsity of $\mathbf{L}_{\mathcal{O}}$. Furthermore, the logarithmic barrier rules out the trivial solution of $\mathbf{L}_{\mathcal{O}} = \mathbf{0}$. The nuclear norm $\|\cdot\|_*$ is a convex regularizer that promotes low-rank solutions for the matrix $\mathbf{K}$ and it is typically employed as a surrogate of the (non-convex) rank constraint. The adoption of the nuclear norm, together with the consideration of the matrices $\mathbf{K}$ and $\mathbf{R}$, ensure the convexity of (4.7) so a globally optimum solution can be efficiently found. The weights $\alpha, \beta, \gamma \geq 0$ control the trade-off between the regularizers, the first constraint ensures that the LV is non-negative, and the second constraint captures that fact of matrix[2] $\mathbf{R}$ being PSD.

The last point to discuss in detail is the form of $\bar{\mathcal{L}}$. Mathematically, the set $\bar{\mathcal{L}}$ is equivalent to the set of combinatorial Laplacians $\mathcal{L}$, but replacing the condition $\mathbf{L1} = \mathbf{0}$ with $\mathbf{L1} \geq \mathbf{0}$, i.e., $\bar{\mathcal{L}} := \{L_{ij} \leq 0 \text{ for } i \neq j;\ \mathbf{L} = \mathbf{L}^{\top};\ \mathbf{L1} \geq \mathbf{0};\ \mathbf{L} \succeq 0\}$. The modification is required because, strictly speaking, $\mathbf{L}_{\mathcal{O}}$ is not a combinatorial Laplacian. The existence of links between the elements in $\mathcal{O}$ and the hidden nodes in $\mathcal{H}$ give rise to non-zero (negative) entries in $\mathbf{L}_{\mathcal{O}\mathcal{H}}$ and, as a result, the sum of the off-diagonal elements of $\mathbf{L}_{\mathcal{O}}$ can be smaller than the value of the associated diagonal elements (which account for the links in both $\mathcal{O}$ and $\mathcal{H}$). Intuitively, the more relaxed condition $\mathbf{L}_{\mathcal{O}}\mathbf{1} \geq \mathbf{0}$ enlarges the set of feasible solutions rendering the inference process harder to solve, an issue that has been observed when running the numerical experiments. Moreover, when estimating the diagonal of $\mathbf{L}_{\mathcal{O}}$ we are indirectly estimating the number of edges between the observed and the hidden nodes. This could be potentially leveraged to estimate links with non-observed nodes, but this entails a more challenging problem that goes beyond the scope of this chapter. An approach to bypass some of these issues is analyzed next.

[2]From an algorithmic point of view, it is worth noticing that the matrix $\mathbf{R}$ always appears as $\text{tr}(\mathbf{R})$ in (4.7). As a result, if convenient to reduce the numerical burden, one can replace $\text{tr}(\mathbf{R})$ with r and optimize over r in lieu of $\mathbf{R}$. See the related formulation in (4.9) for details.

4.3.1 Exploiting the Laplacian of the observed adjacency matrix

The Laplacian $\mathbf{L}$ offers a neat way to measure the smoothness of graph signals [cf. (2.5)]. However, when addressing the problem of estimating the Laplacian from smooth signals under the presence of hidden nodes, we must face the challenges associated with the fact of the submatrix $\mathbf{L}_{\mathcal{O}}$ not being a Laplacian itself. As discussed in the preceding paragraphs, this requires dropping some of the Laplacian constraints from the optimization, leading to a looser recovery framework. To circumvent these issues, rather than estimating $\mathbf{L}_{\mathcal{O}}$, this section looks at the problem of estimating $\tilde{\mathbf{L}}_{\mathcal{O}} := \text{diag}(\mathbf{A}_{\mathcal{O}}\mathbf{1}) - \mathbf{A}_{\mathcal{O}}$, the Laplacian associated with the observed adjacency matrix $\mathbf{A}_{\mathcal{O}} \in \mathbb{R}^{O\times O}$. In contrast to $\mathbf{L}_{\mathcal{O}}$, the matrix $\tilde{\mathbf{L}}_{\mathcal{O}}$ is a proper combinatorial Laplacian ($\tilde{\mathbf{L}}_{\mathcal{O}} \in \mathcal{L}$) and, hence, the original Laplacian constraints can be restored. The remaining of this section is devoted to reformulating (4.7) in terms of $\tilde{\mathbf{L}}_{\mathcal{O}}$.

Upon defining the $O \times O$ diagonal matrices $\mathbf{D}_{\mathcal{O}} := \text{diag}(\mathbf{A}_{\mathcal{O}}\mathbf{1})$ and $\mathbf{D}_{\mathcal{O}\mathcal{H}} := \text{diag}(\mathbf{A}_{\mathcal{O}\mathcal{H}}\mathbf{1})$, which count the number of observed and hidden neighbors for the nodes in $\mathcal{O}$, the matrix $\mathbf{L}_{\mathcal{O}}$ is expressed as $\mathbf{L}_{\mathcal{O}} = \mathbf{D}_{\mathcal{O}} + \mathbf{D}_{\mathcal{O}\mathcal{H}} - \mathbf{A}_{\mathcal{O}} = \tilde{\mathbf{L}}_{\mathcal{O}} + \mathbf{D}_{\mathcal{O}\mathcal{H}}$. With this equivalence, the smoothness penalty in (4.7) is rewritten as

$$\begin{aligned}\text{tr}(\mathbf{C}\mathbf{L}) &= \text{tr}(\mathbf{C}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}) + \text{tr}(\mathbf{C}_{\mathcal{O}}\mathbf{D}_{\mathcal{O}\mathcal{H}}) + 2\text{tr}(\mathbf{K}) + \text{tr}(\mathbf{R}) \\ &= \text{tr}(\mathbf{C}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}) + 2\text{tr}(\tilde{\mathbf{K}}) + \text{tr}(\mathbf{R}),\end{aligned} \tag{4.8}$$

where $\tilde{\mathbf{K}} := \mathbf{C}_{\mathcal{O}}\mathbf{D}_{\mathcal{O}\mathcal{H}}/2 + \mathbf{K}$. Because the entries of $\mathbf{D}_{\mathcal{O}\mathcal{H}}$ depend on the presence of edges between the observed and the hidden nodes, if the graph is sparse, the matrix $\mathbf{D}_{\mathcal{O}\mathcal{H}}$ will be a low-rank matrix. Furthermore, since the sparsity pattern of the diagonal of $\mathbf{D}_{\mathcal{O}\mathcal{H}}$ depends on the matrix $\mathbf{A}_{\mathcal{O}\mathcal{H}} = -\mathbf{L}_{\mathcal{O}\mathcal{H}}$, it follows that the column sparsity pattern of $\mathbf{C}_{\mathcal{O}}\mathbf{D}_{\mathcal{O}\mathcal{H}}$ matches that of $\mathbf{K}$, and thus, $\tilde{\mathbf{K}}$ is also low rank.

With these considerations in mind, we reformulate the optimization in (4.7) replacing $\mathbf{L}_{\mathcal{O}}$ with $\tilde{\mathbf{L}}_{\mathcal{O}}$, resulting in the following convex optimization problem

$$\begin{aligned}\min_{\tilde{\mathbf{L}}_{\mathcal{O}},\tilde{\mathbf{K}},r} \quad & \text{tr}(\mathbf{C}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}) + 2\text{tr}(\tilde{\mathbf{K}}) + r + \alpha\|\tilde{\mathbf{L}}_{\mathcal{O}}\|_{F,off}^2 \\ & - \beta\log(\text{diag}(\tilde{\mathbf{L}}_{\mathcal{O}})) + \gamma_*\|\tilde{\mathbf{K}}\|_* + \gamma_{2,1}\|\tilde{\mathbf{K}}\|_{2,1} \\ \text{s. t.} \quad & \text{tr}(\mathbf{C}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}) + 2\text{tr}(\tilde{\mathbf{K}}) + r \geq 0, \\ & r \geq 0 \\ & \tilde{\mathbf{L}}_{\mathcal{O}} \in \mathcal{L},\end{aligned} \tag{4.9}$$

where $\bar{\mathcal{L}}$ in (4.7) has been replaced with $\mathcal{L}$ in (4.9), which is the set of all valid combinatorial Laplacian matrices defined in (4.4). Moreover, knowing that the matrix $\mathbf{R}$ only appears as $\text{tr}(\mathbf{R})$ we replace it with the nonnegative variable r to alleviate the numerical burden. Note that, although we replaced $\mathbf{K}$ with $\tilde{\mathbf{K}}$, the terms previously associated with $\mathbf{K}$ in (4.7) remain unchanged in (4.9). Nonetheless, while the original matrix $\mathbf{K} \in \mathbb{R}^{O\times O}$ is low rank because it is the product of a tall $O \times H$ matrix and a fat $H \times O$ matrix, the low-rankness of $\tilde{\mathbf{K}}$ is a byproduct of the sparsity of the graph. More precisely, the matrix $\tilde{\mathbf{K}}$ involves the product of the square (full rank) matrix $\mathbf{C}_{\mathcal{O}}$ and the diagonal matrix $\mathbf{D}_{\mathcal{O}\mathcal{H}}$. Since the diagonal of $\mathbf{D}_{\mathcal{O}\mathcal{H}}$ is sparse, such a product gives rise to a matrix with several zero columns, with the rank of the resultant matrix coinciding with the number of non-zero columns. We exploit this structure by further regularizing the matrix $\tilde{\mathbf{K}}$ with the $\ell_{2,1}$ norm.

Indeed, two different configurations of (4.9) can be obtained depending on the values of the regularization constants. Setting $\gamma_{2,1} = 0$ we promote a solution with a low rank on $\tilde{\mathbf{K}}$ by applying

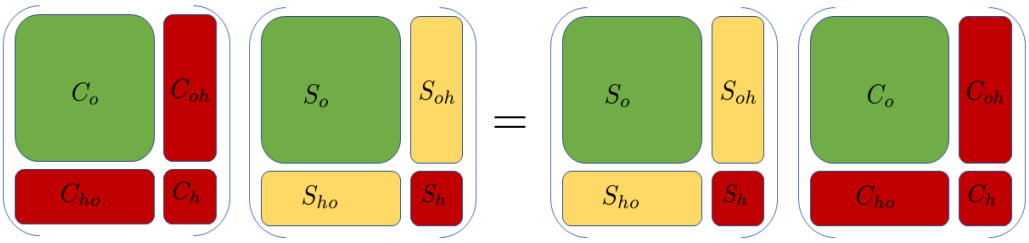

Figure 4.1: Visual representation of the commutativity between $\mathbf{C}$ and $\mathbf{S}$ incorporating the presence of hidden nodes by considering block matrix structure for both, $\mathbf{C}$ and $\mathbf{S}$. Blocks in green are associated with the observed nodes, and blocks in yellow and red are associated with the hidden nodes. Blocks in yellow correspond to the part of the network that could be estimated by assuming stationarity.

the nuclear norm regularization. Since the nuclear norm minimization does not ensure the desired column-sparsity of $\tilde{\mathbf{K}}$, an alternative is to set $\gamma_* = 0$ and rely on the penalty $\|\tilde{\mathbf{K}}\|_{2,1}$. The computation of $\|\tilde{\mathbf{K}}\|_{2,1}$ can be understood as a two-step process where one first obtains the ℓ_2 norm of each of the columns of $\tilde{\mathbf{K}}$ and, then, the ℓ_1 norm of the resulting row vector is computed. This regularization is commonly known as the group Lasso penalty [94, 95] and has been used in a number of sparse-recovery problems. The results in Section 4.6 will illustrate that the formulation in (4.9) succeeds in promoting the desired column-sparsity pattern when using the appropriate values for the hyperparameters γ_* and $\gamma_{2,1}$. Note also that, by looking at the non-zero columns of $\tilde{\mathbf{K}}$, the nodes in $\mathcal{O}$ with connections to hidden nodes can be identified.

4.4 Network topology inference from stationary signals with hidden variables

In this section, instead of relying on the smoothness of the signals $\mathbf{X}$, we approach Problem 4.1 by modifying (AS2) and considering that the data is stationary on the sought graph. The assumption of $\mathbf{X}$ being stationary on $\mathcal{G}$ is tantamount to the matrices $\mathbf{C}$ and $\mathbf{S}$ sharing the same eigenvectors $\mathbf{V}$ [48]. As a result, the approach for the fully observable case is to use the observations to estimate the sample covariance $\hat{\mathbf{C}}$ and then rely on the sample covariance to estimate the eigenvectors $\mathbf{V}$ [4]. However, when dealing with hidden variables, there is no obvious way to obtain $\mathbf{V}_\mathcal{O}$, the submatrix of the eigenvectors of the full covariance, using as input the submatrix $\hat{\mathbf{C}}_\mathcal{O}$. To bypass this problem, instead of requiring the eigenvectors of $\mathbf{C}$ and $\mathbf{S}$ being the same, our approach is to require that $\mathbf{C}$ and $\mathbf{S}$ commute, i.e., that the equation $\mathbf{CS} = \mathbf{SC}$ must hold [96]. To see why this condition leads to a more tractable formulation, let us leverage the block structure of $\mathbf{C}$ and $\mathbf{S}$ described in (4.1). It follows readily that the upper left submatrix of size $O \times O$ in both sides of the equality $\mathbf{CS} = \mathbf{SC}$ is given by

$$\mathbf{C}_\mathcal{O}\mathbf{S}_\mathcal{O} + \mathbf{C}_{\mathcal{OH}}\mathbf{S}_{\mathcal{OH}}^\top = \mathbf{S}_\mathcal{O}\mathbf{C}_\mathcal{O} + \mathbf{S}_{\mathcal{OH}}\mathbf{C}_{\mathcal{OH}}^\top. \tag{4.10}$$

The above expression succeeds in relating the sought $\mathbf{S}_\mathcal{O}$ with $\mathbf{C}_\mathcal{O}$, which can be efficiently estimated using $\mathbf{X}_\mathcal{O}$. Furthermore, (4.10) reveals that when hidden variables are present, we cannot simply ask $\mathbf{S}_\mathcal{O}$ and $\mathbf{C}_\mathcal{O}$ to commute, but we also need to account for the associated terms $\mathbf{C}_{\mathcal{OH}}\mathbf{S}_{\mathcal{OH}}^\top$ and $\mathbf{S}_{\mathcal{OH}}\mathbf{C}_{\mathcal{OH}}^\top$. To better understand the equation in 4.10, Fig. 4.1 shows a visual representation of the commutativity between $\mathbf{C}$ and $\mathbf{S}$ assuming a block structure for both of them.

Implementing steps similar to those in Section 4.3, we can lift the problem defining the matrix $\mathbf{K} = \mathbf{C}_{\mathcal{OH}}\mathbf{S}_{\mathcal{OH}}^\top \in \mathbb{R}^{O\times O}$ and leverage the fact that $\text{rank}(\mathbf{K}) \leq H \ll O$, due to (AS1). Note that the matrix $\mathbf{K}$ is equivalent to the one defined in Section 4.3 with the only difference that now we

use a block from the generic GSO $\mathbf{S}_{\mathcal{OH}}$ instead of the Laplacian $\mathbf{L}_{\mathcal{OH}}$. Moreover, since both $\mathbf{C}$ and $\mathbf{S}$ are symmetric matrices, we have that $\mathbf{K}^\top = \mathbf{S}_{\mathcal{OH}}\mathbf{C}_{\mathcal{OH}}^\top$. Then, under the general assumption that graphs are typically sparse, we can approach Problem 4.1 with stationary observations by solving

$$\begin{aligned} \min_{\mathbf{S}_\mathcal{O},\mathbf{K}} \ & \|\mathbf{S}_\mathcal{O}\|_0 \\ \text{s. t.} \quad & \mathbf{C}_\mathcal{O}\mathbf{S}_\mathcal{O} + \mathbf{K} = \mathbf{S}_\mathcal{O}\mathbf{C}_\mathcal{O} + \mathbf{K}^\top, \\ & \text{rank}(\mathbf{K}) \leq H, \\ & \mathbf{S}_\mathcal{O} \in \mathcal{S}, \end{aligned} \tag{4.11}$$

where the ℓ_0 norm promotes sparse solutions, the equality constraint ensures commutativity of the GSO and the covariance matrix while accounting for latent nodes, and the rank constraint captures the low rank of $\mathbf{K}$ due to (AS1).

Regarding the specific choice of the GSO, when the interest is in the Laplacian matrix we set $\mathbf{S}_\mathcal{O} = \tilde{\mathbf{L}}_\mathcal{O}$, with $\tilde{\mathbf{L}}_\mathcal{O}$ denoting the Laplacian of the observed adjacency matrix. Then, the matrix $\mathbf{K}$ is replaced with $\tilde{\mathbf{K}} = \mathbf{C}_\mathcal{O}\mathbf{D}_{\mathcal{OH}} + \mathbf{K}$, which accounts for the fact of using $\tilde{\mathbf{L}}_\mathcal{O}$ instead of $\mathbf{L}_\mathcal{O}$ in (4.10). This was further motivated in Section 4.3.1, and the discussion provided there also applies here.

The presence of the rank constraint and the ℓ_0 norm renders (4.11) non-convex and computationally hard to solve. Furthermore, the first constraint assumes perfect knowledge of $\mathbf{C}_\mathcal{O}$, which may not always represent a practical setup. These issues are addressed in the next section.

4.4.1 Convex and robust stationary network topology inference method

A natural approach to deal with (4.11) is to relax the non-convex terms, replacing the ℓ_0 norm with the ℓ_1 norm and the rank constraint with the nuclear norm, their closest convex surrogates. Furthermore, in most practical scenarios the ensemble covariance $\mathbf{C}_\mathcal{O}$ is not known and one must rely on its sampled counterpart $\hat{\mathbf{C}}_\mathcal{O}$. This requires relaxing the equality constraint $\mathbf{C}_\mathcal{O}\mathbf{S}_\mathcal{O} + \mathbf{K} = \mathbf{S}_\mathcal{O}\mathbf{C}_\mathcal{O} + \mathbf{K}^\top$ and replacing it with a constraint which guarantees that the terms on the left-hand side and right-hand side are similar but not necessarily the same. Taking all these considerations into account, the relaxed convex topology-inference problem is

$$\begin{aligned} \min_{\mathbf{S}_\mathcal{O},\mathbf{K}} \ & \|\mathbf{S}_\mathcal{O}\|_1 + \eta\|\mathbf{K}\|_* \\ \text{s. t.} \ & \|\hat{\mathbf{C}}_\mathcal{O}\mathbf{S}_\mathcal{O} + \mathbf{K} - \mathbf{S}_\mathcal{O}\hat{\mathbf{C}}_\mathcal{O} - \mathbf{K}^\top\|_F^2 \leq \epsilon, \\ & \mathbf{S}_\mathcal{O} \in \mathcal{S}, \end{aligned} \tag{4.12}$$

where $\eta \geq 0$ controls the low rankness of $\mathbf{K}$. Regarding the (relaxed) stationarity constraint, the squared Frobenius norm has been adopted to measure the similarity between the matrices at hand, but other (convex) distances could be alternatively used. It is also important to note that the value of the non-negative constant ϵ should be selected based on prior knowledge on the noise level present in the observations and, more importantly, the number of samples R used to estimate the covariance. Clearly, if $R < O$, the matrix is not full rank, increasing notably the size of the feasible set. On the other hand, if $R \to \infty$, one can set $\epsilon = 0$. This reduces drastically the degrees of freedom of the formulation and, as a result, renders more likely the solution to (4.12) to coincide with the actual GSO.

Remark 1 (Reweighted algorithm): The formulation in (4.12) is convex and robust. However, while replacing the original ℓ_0 norm with the convex ℓ_1 norm constitutes a common approach, it is

well-known that non-convex surrogates can lead to sparser solutions. Indeed, a more sophisticated alternative in the context of sparse recovery is to define δ as a small positive number and replace the ℓ_0 norm with a (non-convex) logarithmic penalty $\|\mathbf{S}_{\mathcal{O}}\|_0 \approx \sum_{i,j=1}^{O} \log(|[\mathbf{S}_{\mathcal{O}}]_{ij}|+\delta)$ [72]. An efficient way to handle the non-convexity of the logarithmic penalty is to rely on an MM approach [97], which considers an iterative linear approximation to the concave objective and leads to an *iterative* re-weighted ℓ_1 minimization. To be specific, with $t = 1, ..., T$ being the iteration index, adopting such an approach for the problem in (4.12) results in

$$\begin{aligned} \mathbf{S}_{\mathcal{O}}^{(t+1)} := \underset{\mathbf{S}_{\mathcal{O}},\mathbf{K}}{\operatorname{argmin}} \; & \sum_{i,j=1}^{O} [\mathbf{W}^{(t)}]_{ij} |[\mathbf{S}_{\mathcal{O}}]_{ij}| + \eta \|\mathbf{K}\|_* \\ \text{s. t.} \quad & \mathbf{C}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \mathbf{K} = \mathbf{S}_{\mathcal{O}}\mathbf{C}_{\mathcal{O}} + \mathbf{K}^\top, \\ & \mathbf{S}_{\mathcal{O}} \in \mathcal{S}, \end{aligned} \tag{4.13}$$

with $\mathbf{W}^{(t)}$ being defined as $[\mathbf{W}^{(t)}]_{ij} = \left(\left|[\mathbf{S}_{\mathcal{O}}^{(t-1)}]_{ij}\right| + \delta\right)^{-1}$. Since the iterative algorithm penalizes (assigns a larger weight to) entries of $\mathbf{S}_{\mathcal{O}}$ that are close to zero, the obtained solution is typically sparser at the expense of a higher computational cost. Finally, note that the absolute values can be removed whenever the constraint $[\mathbf{S}_{\mathcal{O}}]_{ij} \geq 0$ is enforced.

4.4.2 Exploiting structure through alternating optimization

In the previous section, the product of the unknown matrices $\mathbf{C}_{\mathcal{OH}}$ and $\mathbf{S}_{\mathcal{OH}}^\top$ was absorbed into matrix $\mathbf{K}$. Since such a matrix is low rank, the convex nuclear norm was used to promote low-rank solutions while achieving convexity. However, when implementing this approach, there were other properties (such as $\mathbf{S}_{\mathcal{OH}}$ being sparse) that were ignored. A reasonable question is, hence, if the judicious incorporation of the additional information outperforms the potential loss of convexity. In this section, we propose an efficient alternating *non-convex* algorithm that accounts for the additional structure present in our setup. Its associated recovery performance (along with comparisons to its convex counterparts) will be tested in Section 4.6.

A well-established approach in low-rank optimization is to factorize the matrix of interest as the product of a tall and fat matrix, which boils down to replacing $\mathbf{K}$ with the original submatrices $\mathbf{C}_{\mathcal{OH}}$ and $\mathbf{S}_{\mathcal{OH}}^\top$. Moreover, when the value of H is unknown, which determines the size of $\mathbf{C}_{\mathcal{OH}}$ and $\mathbf{S}_{\mathcal{OH}}^\top$, a principled approach is to rely on an upper bound on H and add the Frobenius terms $\|\mathbf{C}_{\mathcal{OH}}\|_F$ and $\|\mathbf{S}_{\mathcal{OH}}\|_F$ to the objective function (see, e.g., [98] for a formal derivation of this approach). In our particular setup, this factorization has the additional benefit of $\mathbf{S}_{\mathcal{OH}}$ being sparse. Then, the resulting non-convex optimization problem is given by

$$\begin{aligned} \min_{\mathbf{S}_{\mathcal{O}},\mathbf{C}_{\mathcal{OH}},\mathbf{S}_{\mathcal{OH}}} \; & \sum_{i,j=1}^{O} \log(|[\mathbf{S}_{\mathcal{O}}]_{ij}| + \delta) + \eta \|\mathbf{S}_{\mathcal{OH}}\|_F^2 \\ & + \nu \sum_{i,j=1}^{O,H} \log(|[\mathbf{S}_{\mathcal{OH}}]_{ij}| + \delta) + \eta \|\mathbf{C}_{\mathcal{OH}}\|_F^2 \\ & + \rho \|\hat{\mathbf{C}}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \mathbf{C}_{\mathcal{OH}}\mathbf{S}_{\mathcal{OH}}^\top - \mathbf{S}_{\mathcal{O}}\hat{\mathbf{C}}_{\mathcal{O}} - \mathbf{S}_{\mathcal{OH}}\mathbf{C}_{\mathcal{OH}}^\top\|_F^2 \\ \text{s. t.} \quad & \mathbf{S}_{\mathcal{O}} \in \mathcal{S}, \;\; \mathbf{S}_{\mathcal{OH}} \in \mathcal{S}_{\mathcal{OH}}, \end{aligned} \tag{4.14}$$

Clearly, problem (4.14) guarantees that the rank of the matrix $\mathbf{S}_{\mathcal{OH}}\mathbf{C}_{\mathcal{OH}}^\top$ is upper bounded by the size of its composing factors $\mathbf{S}_{\mathcal{OH}}$ and $\mathbf{C}_{\mathcal{OH}}$. In this case, the sparse solutions for $\mathbf{S}_{\mathcal{O}}$ and $\mathbf{S}_{\mathcal{OH}}$ are promoted by means of the (concave) logarithmic penalty, introduced on Remark 1. The

robust commutativity constraint is placed on the objective function as a penalty term, and the set $\mathcal{S}_{\mathcal{OH}}$ captures the fact that $\mathbf{S}_{\mathcal{OH}}$ is a block from the GSO. In its simplest form, we have that $\mathcal{S}_{\mathcal{OH}} := \{S_{ij} \geq 0\}$ if the GSO is the adjacency matrix, and $\mathcal{S}_{\mathcal{OH}} := \{S_{ij} \leq 0\}$ if it is set to the Laplacian matrix.

The main drawback associated with the formulation in (4.14) is that the presence of the bilinear term $\mathbf{C}_{\mathcal{OH}}\mathbf{S}_{\mathcal{OH}}^\top$ and the logarithmic penalty render the problem non-convex. To address this issue, we implement a block successive upper bound minimization (BSUM) algorithm [99], an iterative approach that blends techniques from MM and alternating optimization. Then, we find a solution to (4.14) by iterating between the following thee steps.

Step 1. Given the estimates $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}$, we substitute $\mathbf{C}_{\mathcal{OH}} = \hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ and $\mathbf{S}_{\mathcal{OH}} = \hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}$ into (4.14) and solve it to estimate $\mathbf{S}_{\mathcal{O}}$. This yields

$$\begin{aligned}\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)} := \underset{\mathbf{S}_{\mathcal{O}} \in \mathcal{S}}{\operatorname{argmin}} \sum_{i,j=1}^{O} [\mathbf{W}_{\mathcal{O}}^{(t)}]_{ij} |[\mathbf{S}_{\mathcal{O}}]_{ij}| \\ + \rho \|\hat{\mathbf{C}}_{\mathcal{O}}\mathbf{S}_{\mathcal{O}} + \hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}[\hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}]^\top - \mathbf{S}_{\mathcal{O}}\hat{\mathbf{C}}_{\mathcal{O}} - \hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}[\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}]^\top\|_F^2,\end{aligned} \tag{4.15}$$

where the logarithmic penalty is approximated by the re-weighted ℓ_1 norm as detailed after (4.13).

Step 2. Given the estimate $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ from the previous iteration, and leveraging the estimate $\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}$ from the last step, we estimate the matrix $\mathbf{S}_{\mathcal{OH}}$ by solving

$$\begin{aligned}\hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)} := \underset{\mathbf{S}_{\mathcal{OH}} \in \mathcal{S}_{\mathcal{OH}}}{\operatorname{argmin}} \sum_{i,j=1}^{O,H} [\mathbf{W}_{\mathcal{OH}}^{(t)}]_{ij} |[\mathbf{S}_{\mathcal{OH}}]_{ij}| + \eta \|\mathbf{S}_{\mathcal{OH}}\|_F^2 \\ + \rho \|\hat{\mathbf{C}}_{\mathcal{O}}\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)} + \hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}\mathbf{S}_{\mathcal{OH}}^\top - \hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}\hat{\mathbf{C}}_{\mathcal{O}} - \mathbf{S}_{\mathcal{OH}}[\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}]^\top\|_F^2.\end{aligned} \tag{4.16}$$

Step 3. With the estimates from the previous steps in place, the last step involves estimating the matrix $\mathbf{C}_{\mathcal{OH}}$ by solving

$$\begin{aligned}\hat{\mathbf{C}}_{\mathcal{OH}}^{(t+1)} := \underset{\mathbf{C}_{\mathcal{OH}}}{\operatorname{argmin}} \quad \eta \|\mathbf{C}_{\mathcal{OH}}\|_F^2 \\ \|\hat{\mathbf{C}}_{\mathcal{O}}\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)} + \mathbf{C}_{\mathcal{OH}}[\hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)}]^\top - \hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}\hat{\mathbf{C}}_{\mathcal{O}} - \hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)}\mathbf{C}_{\mathcal{OH}}^\top\|_F^2.\end{aligned} \tag{4.17}$$

The alternating algorithm is initialized by solving (4.12) to obtain $\hat{\mathbf{K}}$ and setting $\hat{\mathbf{C}}_{\mathcal{OH}}^{(0)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(0)}$ as

$$\hat{\mathbf{C}}_{\mathcal{OH}}^{(0)} = \mathbf{F}_{\mathcal{OH}}\mathbf{\Sigma}_{\mathcal{H}}^{\frac{1}{2}} \text{ and } \hat{\mathbf{S}}_{\mathcal{OH}}^{(0)} = \mathbf{G}_{\mathcal{OH}}\mathbf{\Sigma}_{\mathcal{H}}^{\frac{1}{2}}, \tag{4.18}$$

where $\mathbf{F}_{\mathcal{OH}}$ and $\mathbf{G}_{\mathcal{OH}}$ are the left and right singular vectors associated with the top H singular values, $\mathbf{\Sigma}_{\mathcal{H}}$, obtained from the singular value decomposition $\hat{\mathbf{K}} = \mathbf{F}\mathbf{\Sigma}\mathbf{G}^\top$. A summary of the proposed iterative algorithm is presented in Algorithm 4.

The three steps proposed in (4.15)-(4.17) are iterated until convergence to a stationary point is achieved, a result that is formally stated next.

Proposition 1. *Denote with f the objective function in* (4.14)*. Let $\mathcal{Y}^*$ be the set of stationary points of* (4.14)*, and let $\mathbf{y}^{(t)} = [\text{vec}(\mathbf{S}_{\mathcal{O}}^{(t)})^\top, \text{vec}(\mathbf{S}_{\mathcal{OH}}^{(t)})^\top, \text{vec}(\mathbf{C}_{\mathcal{OH}}^{(t)})^\top]^\top$ be the solution generated after running the 3 steps in* (4.15)-(4.17) *t times. Then, the solution generated by the iterative algorithm* (4.15)-(4.17) *converges to a stationary point of f as t goes to infinity, i.e.,*

$$\lim_{t\to\infty} d(\mathbf{y}^{(t)}, \mathcal{Y}^*) = 0,$$

Algorithm 4: BSUM network topology inference method for stationary signals with hidden variables (BSUM-GSHV)

Input: $\hat{\mathbf{C}}_{\mathcal{O}}$

Outputs: $\hat{\mathbf{S}}_{\mathcal{O}}$, $\hat{\mathbf{C}}_{\mathcal{OH}}$, and $\hat{\mathbf{S}}_{\mathcal{OH}}$

Initialize $\hat{\mathbf{S}}_{\mathcal{O}}^{(0)}$ and $\hat{\mathbf{K}}$ by solving (4.12)

Initialize $\hat{\mathbf{C}}_{\mathcal{OH}}^{(0)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(0)}$ following (4.18)

for $t = 0$ **to** $T - 1$ **do**

  Update $\mathbf{W}_{\mathcal{O}}^{(t)} = \left(\left|\hat{\mathbf{S}}_{\mathcal{O}}^{(t)}\right| + \delta\right)^{-1}$ and $\mathbf{W}_{\mathcal{OH}}^{(t)} = \left(\left|\hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}\right| + \delta\right)^{-1}$

  Update $\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}$ by solving (4.15) using $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(t)}$

  Update $\hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)}$ by solving (4.16) using $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t)}$ and $\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}$

  Update $\hat{\mathbf{C}}_{\mathcal{OH}}^{(t+1)}$ by solving (4.17) using $\hat{\mathbf{S}}_{\mathcal{O}}^{(t+1)}$ and $\hat{\mathbf{S}}_{\mathcal{OH}}^{(t+1)}$

end

$\hat{\mathbf{S}}_{\mathcal{O}} = \hat{\mathbf{S}}_{\mathcal{O}}^{(T)}$, $\hat{\mathbf{S}}_{\mathcal{OH}} = \hat{\mathbf{S}}_{\mathcal{OH}}^{(T)}$, $\hat{\mathbf{C}}_{\mathcal{OH}} = \hat{\mathbf{C}}_{\mathcal{OH}}^{(T)}$

with $d(\mathbf{y}, \mathcal{Y}^*) := \min_{\mathbf{y}^* \in \mathcal{Y}^*} \|\mathbf{y} - \mathbf{y}^*\|_2$.

Note that convergence was not obvious since at least one of the steps does not have a unique minimizer, and the first and second steps employ an approximation of the objective function in (4.14). The details of the proof, which relies on convergence results for BSUM schemes [99, Th. 1b], are provided in Section 4.8.

While incurring additional computational costs (see Remark 2 for more details), the numerical tests in Section 4.6 confirm that the supplemental structure incorporated by replacing $\mathbf{K}$ with $\mathbf{S}_{\mathcal{OH}}$ and $\mathbf{C}_{\mathcal{OH}}$ together with the re-weighted ℓ_1 approach for encouraging sparsity give rise to a better network reconstruction, provided that the iterative optimization is initialized with the solution to the convex formulation in (4.12). Last but not least, notice that an additional benefit of the formulation in (4.14) is that, by analyzing $\hat{\mathbf{S}}_{\mathcal{OH}}$, information of the potential links between nodes in $\mathcal{O}$ and the hidden nodes in $\mathcal{H}$ is obtained. While network-tomography schemes [13] go beyond the scope of this chapter, the results in this section can be used as a first step towards that goal.

Remark 2 (Computational complexity): The computational complexity required to solve the optimization problems proposed in this chapter scales polynomially with the size of the graph. More specifically, since (AS1) guarantees that $H \ll O$, for the convex formulations, the complexity scales as $\mathrm{O}(O^7)$, which is an order similar to that of the "plain vanilla" LVGL in (4.5), but considerably larger than the order $\mathrm{O}(RO^2)$ for correlation networks. Regarding the complexity of solving the non-convex formulation in (4.14) using Algorithm 4, each of the steps (4.15)-(4.17) entails solving a convex problem, so the complexity scales as $\mathrm{O}(TO^7)$, with T denoting the number of iterations. In practice, our simulations show that the number of iterations required to converge in all tested scenarios is fairly low (with T taking values between 3-6), which is a behavior also observed in other applications of the BSUM algorithm to sparsity-promoting biconvex problems. As a result, the complexity of solving the non-convex problem in (4.14) is expected to scale similarly to that required to solve the non-iterative convex formulations presented in the previous sections. While this complexity is associated with the fact of considering challenging operating conditions, the aforementioned levels hinder the application of the proposed algorithms to large graphs. An approach to mitigate this issue is to exploit the structure of the problems at hand, developing tailored block-coordinate algorithms that solve for each variable separately and exploit the sparsity

of the GSO. Indeed, some algorithmic alternatives such as projected gradient and ADMM could be applied to implement more scalable and efficient models [40, 57, 84]. Although interesting, the development of efficient algorithms is beyond the scope of this chapter so it is left as future work.

Remark 3 (Graph stationary vis-à-vis graph smoothness): Suppose that we are given two datasets $\mathbf{X}_{\mathcal{O}}$ and $\mathbf{X}'_{\mathcal{O}}$, both with the same number of signals. Moreover, suppose that we also know that the observed signals $\mathbf{X}_{\mathcal{O}}$ are smooth on an unknown graph, that $\mathbf{X}'_{\mathcal{O}}$ are stationary on an unknown graph, and that our goal is to identify the underlying graphs. Based on that information, we run the algorithms in Section 4.3 for the dataset $\mathbf{X}_{\mathcal{O}}$ and those in this section for the dataset $\mathbf{X}'_{\mathcal{O}}$. An interesting question is which one yields a better recovery result. While the exact answer depends on all the particularities of each of the setups, from a general point of view stationary schemes are expected to achieve better results. The reason is that stationarity strongly limits the degrees of freedom of the GSO, while smoothness is a more lenient assumption, an intuition that will be validated in Section 4.6. Equally relevant, there can be situations where the data is both stationary and smooth. That is the case, for example, if the covariance matrix shares the eigenvectors with the graph Laplacian and its power spectral density is low pass. In such a setup, one could combine both network-recovery approaches, leading to a better recovery performance. This is precisely the subject of the ensuing section.

4.5 Network topology inference from stationary and smooth graph signals with hidden variables

In this section, we address Problem 4.1 by assuming that the graph signals $\mathbf{X}$ are both smooth and stationary on the unknown graph $\mathcal{G}$. These two assumptions can be jointly considered to design optimization problems with additional structure to enhance the estimation of $\mathbf{S}_{\mathcal{O}}$. To that end, we consider the smoothness-based inference problem described in (4.9) and incorporate the robust commutativity constraint accounting for stationarity [cf. (4.10)], resulting in

$$\begin{aligned}\min_{\tilde{\mathbf{L}}_{\mathcal{O}},\tilde{\mathbf{K}},r}\ & \mathrm{tr}(\hat{\mathbf{C}}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}})+2\mathrm{tr}(\tilde{\mathbf{K}})+r+\alpha\|\tilde{\mathbf{L}}_{\mathcal{O}}\|_{F,off}^2 \\ & \quad -\beta\log(\mathrm{diag}(\tilde{\mathbf{L}}_{\mathcal{O}}))+\gamma_*\|\tilde{\mathbf{K}}\|_*+\gamma_{2,1}\|\tilde{\mathbf{K}}\|_{2,1} \\ \text{s. t.}\ & \mathrm{tr}(\hat{\mathbf{C}}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}})+2\mathrm{tr}(\tilde{\mathbf{K}})+r\geq 0, \\ & \tilde{\mathbf{L}}_{\mathcal{O}}\in\mathcal{L}, \\ & \|\hat{\mathbf{C}}_{\mathcal{O}}\tilde{\mathbf{L}}_{\mathcal{O}}+\tilde{\mathbf{K}}-\tilde{\mathbf{L}}_{\mathcal{O}}\hat{\mathbf{C}}_{\mathcal{O}}-\tilde{\mathbf{K}}^\top\|_F^2\leq\epsilon.\end{aligned} \tag{4.19}$$

Since the smooth formulation involves the Laplacian matrix, note that we adopted the Laplacian of the observed adjacency matrix as the GSO. Regarding the stationarity constraint, as discussed for (4.12), the value of ϵ should be selected based on the number of available signals R and the observation noise. It is also worth noting that the matrix $\tilde{\mathbf{K}}$ is inconspicuously absorbing the error derived from the presence of the hidden variables and from using $\tilde{\mathbf{L}}_{\mathcal{O}}$ instead of $\mathbf{L}_{\mathcal{O}}$ in both the smoothness penalty and the commutativity constraint. Regarding matrix $\tilde{\mathbf{K}}$, two different regularizers are considered: the nuclear norm (to promote solutions with a low rank) and the $\ell_{2,1}$ norm (to promote column sparsity). Since having solutions with columns that are zero also reduces the rank, it is prudent to tune the value of the hyperparameters γ_* and $\gamma_{2,1}$ jointly, so that the (joint) dependence between the rank and the column sparsity is kept under control.

We close the section by noting that the formulation in (4.19) is convex so that its globally optimal solution can be found efficiently. However, non-convex versions of (4.19) that leverage the

re-weighted $\ell_{2,1}$ norm to promote column sparsity and factorization approaches for the low-rank penalty (similar to those used in Section 4.4) could be developed here as well.

4.6 Numerical experiments

This section runs numerical experiments to gain insights on the proposed schemes and evaluate their recovery performance. First, we test the smooth-based approaches with synthetic data and compare the results with existing algorithms from the literature. Secondly, we assess the performance of the stationary-based schemes proposed in Section 4.4, comparing them with those in Section 4.5 and the classical LVGL. Lastly, we apply the proposed algorithms to two real-world datasets and compare the obtained results with those of existing alternatives.[3]

4.6.1 Synthetic experiments based on smooth signals

We start by defining the default setup for the experiments in this section. With $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$ denoting the eigendecomposition of the graph Laplacian, the smooth signals $\mathbf{X}$ are generated as $\mathbf{X} = \mathbf{V}\mathbf{J}$, where the columns of $\mathbf{J} \in \mathbb{R}^{N\times R}$ are independent realizations of a multivariate Gaussian distribution $\mathbf{J} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^\dagger)$. Note that this model, which is oftentimes referred to as factor analysis [85, 100, 101], assigns more energy to the low-frequency components, promoting smoothness on the generated graph signals. Unless otherwise stated, the number of signals is set to $R = 100$ and the number of nodes to $N = 20$. Moreover, to measure the recovery performance of the algorithms, in this section we focus on unweighted graphs and employ the F*score*, which is defined as

$$F\textit{score} = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{4.20}$$

where $precision$ indicates the percentage of estimated edges that are edges of the ground-truth graph and $recall$ is the percentage of existing edges that were correctly estimated.

Influence of hidden nodes. The results in Fig. 4.2 show the variation of the F*score*, as the number of hidden variables H increases, for different recovery algorithms. Graphs are randomly generated using the model in [85], where nodes are placed in the unit square uniformly at random and edges are computed with a Gaussian radial basis function (RBF) as $A_{ij} = \exp(-d(i,j)^2/2\sigma^2)$, with $d(i,j)$ being the euclidean distance between two vertices and $\sigma = 0.5$. Edges with weights smaller than 0.75 are removed and the surviving ones are set to 1. The hidden nodes are chosen uniformly at random among all the nodes in the graph. The algorithms considered in this experiment are the following: (i) GL-SigRep refers to the algorithm presented in [85]; (ii) GSm is a modified version of GL-SigRep that incorporates the logarithmic penalty and relies on the sample covariance matrix $\hat{\mathbf{C}}$ for the smoothness term in the objective function; (iii) GSm-LR represents the low-rank regularized algorithm proposed in (4.9), with $\gamma_{2,1} = 0$; and (iv) GSm-GL denotes the algorithm described in (4.9), with $\gamma_* = 0$, where column-sparsity is promoted in $\tilde{\mathbf{K}}$ via group Lasso. Comparing GL-SigRep with GSm allows us to quantify the improvement obtained exclusively from including hidden variables in the formulation, providing a fairer analysis of the proposed algorithms. The results in Fig. 4.2 indicate that, although the performance of all the algorithms deteriorates when the number of hidden variables increases, the algorithms GSm-LR and GSm-GL which account for the presence of hidden variables, outperform the alternatives. Moreover, their performance drops

[3] The MATLAB scripts for running all the numerical experiments presented in this section as well as additional related test cases can be found in `https://github.com/andreibuciulea/topoIDhidden`

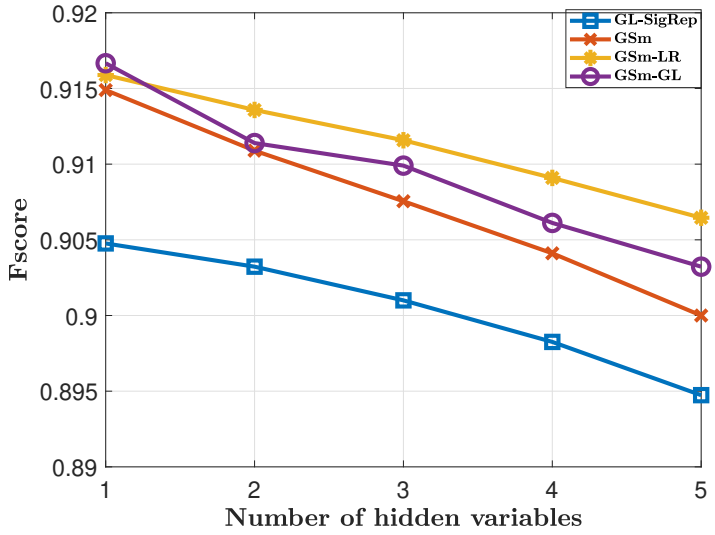

Figure 4.2: Median F*score* for the algorithms based on smooth graph signals with $N = 20$ and $R = 100$. Obtained results regarding the impact of varying the number of hidden variables H for different algorithms when using RBF graphs.

slowly as H increases, demonstrating the importance of taking into account the presence of hidden variables. The overall decay was expected since a higher number of hidden variables renders the network topology inference problem more challenging and ill-posed, confirming the importance of (AS1). Comparing GSm-LR with GSm-GL, we observe that their performance is similar since the generated graphs are sufficiently sparse. It is also worth mentioning that the GSm scheme clearly outperforms GL-SigRep, illustrating the benefits of replacing the formulation introduced in [85] with the one presented in this chapter, which relies on the matrix $\hat{\mathbf{C}}$ and the logarithmic barrier.

Noisy smooth observations. The second experiment assumes that the observations $\mathbf{X}_{\mathcal{O}}$ correspond to the ground-truth signals corrupted by additive white Gaussian noise (AWGN). For that setup, we evaluate the link-identification performance upon evaluating the F*score* achieved by GSm-LR and GSm-GL schemes, comparing them with GSm, as the power of the AWGN increases, for graphs with different sparsity levels. In the experiments, we use ER graphs with edge probability values of $p = \{0.1, 0.3, 0.5\}$ and set the number of hidden variables to $H = 1$. The results, shown in Fig. 4.3, reveal that the performance of the algorithms deteriorates not only when the noise increases but also for higher values of p. This behavior is consistent with the discussion provided in Section 4.3, since the formulation assumes that sparsity exists and, as a result, promotes solutions where several of the columns of $\tilde{\mathbf{K}}$ are zero. Furthermore, we observe that GSm-LR and GSm-GL have similar performance for lower values of p, but when the graphs become denser GSm-GL outperforms GSm-LR. This illustrates the fact that the low-rank regularization $\|\tilde{\mathbf{K}}\|_*$ is more sensitive to the sparsity of the graph than the group Lasso penalty $\|\tilde{\mathbf{K}}\|_{2,1}$. It is also worth noting that, even though the proposed schemes were not designed to specifically account for noisy observations, the rate at which F*score* decays is smaller than the rate at which the noise power increases, showcasing the "natural" robustness to noise of the proposed schemes. Finally, note that GSm-LR and GSm-GL outperform GSm for the different values of p, which reinforces the importance of considering the presence of hidden variables.

Influence of the LV level. Next, we assess the relevance of the smoothness prior to the performance of the GSm-LR scheme. To that end, Fig. 4.4 depicts the F*score* obtained with this

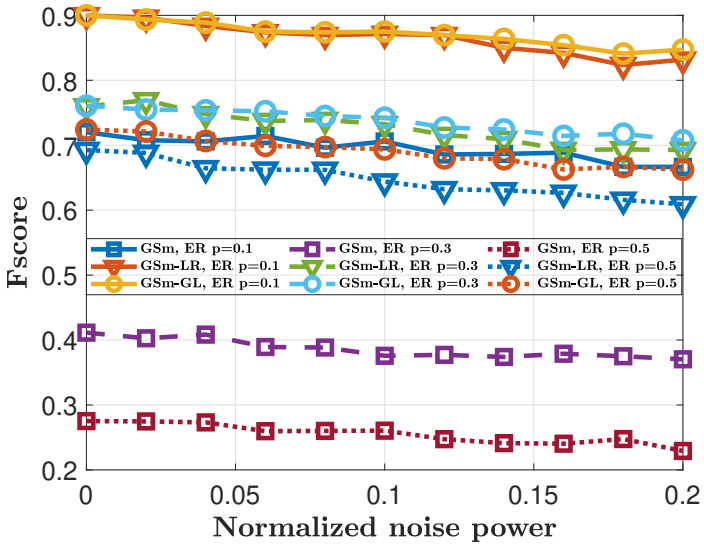

Figure 4.3: Median $Fscore$ for the algorithms based on smooth graph signals with $N = 20$ and $R = 100$. Obtained results regarding the impact of varying the noise level present in the observations $\mathbf{X}$ when using Erdős Rényi graphs with different link probabilities $p = \{0.1, 0.3, 0.5\}$.

scheme for different values of LV. Note that as we move to the right on the x-axis, the observed signals exhibit a larger variation (higher frequency) and, as a result, are less smooth. To control the LV level, the signals are generated combining K successive eigenvectors as $\mathbf{X} = \mathbf{V}_K\mathbf{J}$, with $\mathbf{V} \in \mathbb{R}^{O\times K}$ and $\mathbf{J} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \in \mathbb{R}^{K\times R}$. The smoothest signals are obtained by selecting the first K eigenvectors since they are associated with the low-frequency components. In contrast, activating the K last eigenvectors maximizes the LV of the graph signals. For this experiment, we set $H = 1$, $K = 5$, and $N = 30$. The first generated signal is associated with eigenvectors

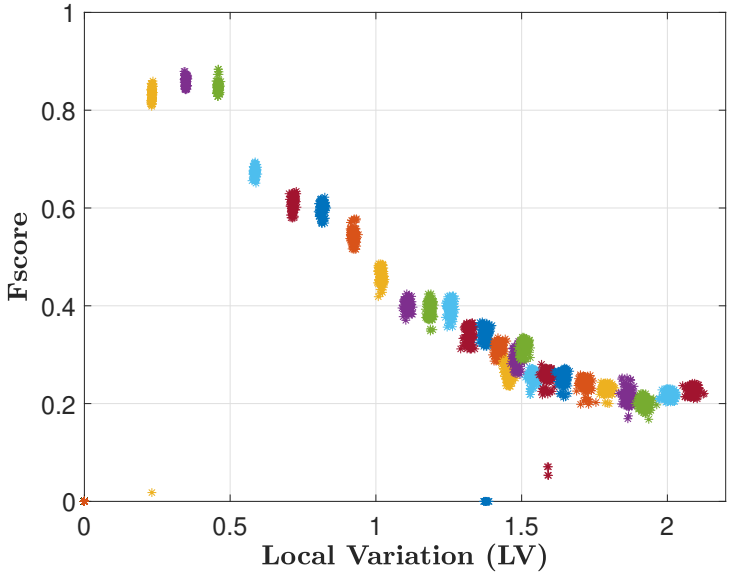

Figure 4.4: Median $Fscore$ for the GSm-LR algorithm with $N = 20$ and $R = 100$. Obtained results regarding the impact of varying the average level of LV of the observations $\mathbf{X}$ for a GSm-LR algorithm when using RBF graphs.

$k = 1, ..., 5$, the second one with eigenvectors $k = 2, ..., 6$, and the last (26th) one with eigenvectors $k = 26, ..., 30$. The link-identification performance for those 26 types of signals is shown in Fig. 4.4, where the vertical axis represents the F*score* and the horizontal axis the average LV as $\text{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X})/R$. Each color represents a different set of active frequencies and, for each set, 128 realizations of $\mathbf{J}$ have been generated (corresponding to the cloud of points shown in the figure). The results highlight the importance of the low values of LV when assuming smooth signals on the graph since the link identification performance decays noticeably as the signal becomes high-pass.

4.6.2 Synthetic experiments based on stationary signals

In these experiments, we focus on signals that are stationary on the sought GSO $\mathbf{S}$. To facilitate comparisons with GL, two different signal models are considered: (i) $\mathbf{C}_{poly}$ and (ii) $\mathbf{C}_{MRF}$. For the first one, the covariance of the observed signals is generated as a random polynomial of the GSO of the form $\mathbf{C}_{poly} = \mathbf{H}^2$ with $\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{S}^l$, where h_l are random coefficients following a normalized zero-mean Gaussian distribution. Note that this generative model guarantees that the covariance is PSD and a polynomial (of degree $2L$) of the GSO. In the second model, the covariance is generated as $\mathbf{C}_{MRF} = (\sigma \mathbf{I} + \delta \mathbf{S})^{-1}$, where σ is some positive number large enough to guarantee that $\mathbf{C}_{MRF}^{-1}$ is PSD and δ is some positive random number. As in the previous case, this generation guarantees the covariance matrix to be PSD and a polynomial of the GSO. Moreover, it also guarantees that the sparsity pattern of $\mathbf{C}_{MRF}^{-1}$ coincides with that of the GSO $\mathbf{S}$, which is the model assumed by GL. Regarding the metric used to evaluate the performance, rather than using the F*score*, we will generate multiple graphs and report the ratio of graphs that have been perfectly recovered (i.e., those graphs for which *all* the entries of the associated $\mathbf{S}_{\mathcal{O}}$ are estimated correctly). The reason for using this metric is that the incorporation of the stationary constraints boosts the ability of the algorithm to identify the topology, so that the value of F*score* will be very close to one for all tested schemes, rendering the comparison more difficult. Differently, reporting the ratio of graphs perfectly recovered helps us to better assess the differences between the tested algorithms.

Leveraging the structure of $\mathbf{K}$. While the ultimate goal of this work is to recover $\mathbf{S}_{\mathcal{O}}$, the properties of matrix $\mathbf{K}$ played a key role in developing several of our network topology inference algorithms. For that reason, the goal of this experiment is to illustrate the recovered (estimated) $\hat{\mathbf{S}}_{\mathcal{O}}$ and $\hat{\mathbf{K}}$, so that we can gain insights on the effectiveness of the different approaches considered in this chapter and their influence in recovering the graph. The results are shown in Fig. 4.5, where the first row represents the GSOs and the second row the matrices $\mathbf{K}$. The first column corresponds to the ground-truth values, and the second, third and fourth columns present the estimates obtained with the low-rank scheme GSt [cf. (4.12)], the group Lasso scheme GSm-St-GL [cf. (4.19) with $\gamma_* = 0$], and the factorized scheme GSt-Rw-Fact [cf. (4.15)-(4.17)], respectively. First, focusing on $\hat{\mathbf{K}}$, it is apparent that for the depicted example the low-rank scheme GSt is not capable of recovering the column-sparse structure of the original matrix $\mathbf{K}$. Differently, when using either the group Lasso regularization (Fig. 4.5g) or the factorized approach (Fig. 4.5h), the estimated $\hat{\mathbf{K}}$ exhibits a row-sparsity pattern that is close to that of the ground truth. More importantly, when looking at the estimated $\hat{\mathbf{S}}_{\mathcal{O}}$ we observe that, as desired, the more accurate estimation of $\mathbf{K}$ translates into a superior estimation of the network topology, with GSm-St-GL yielding better estimates than GSt and GSt-Rw-Fact outperforming GSm-St-GL due to the replacement of the ℓ_1 norm with the linearized version of the logarithmic penalty. Overall, we believe that this simple experiment provides further intuition and strengthens the discussion about the different regularizers presented in Sections 4.3 and 4.4. The next step is to test the stationary-based schemes in a more systematic way, which is the goal of the following subsections.

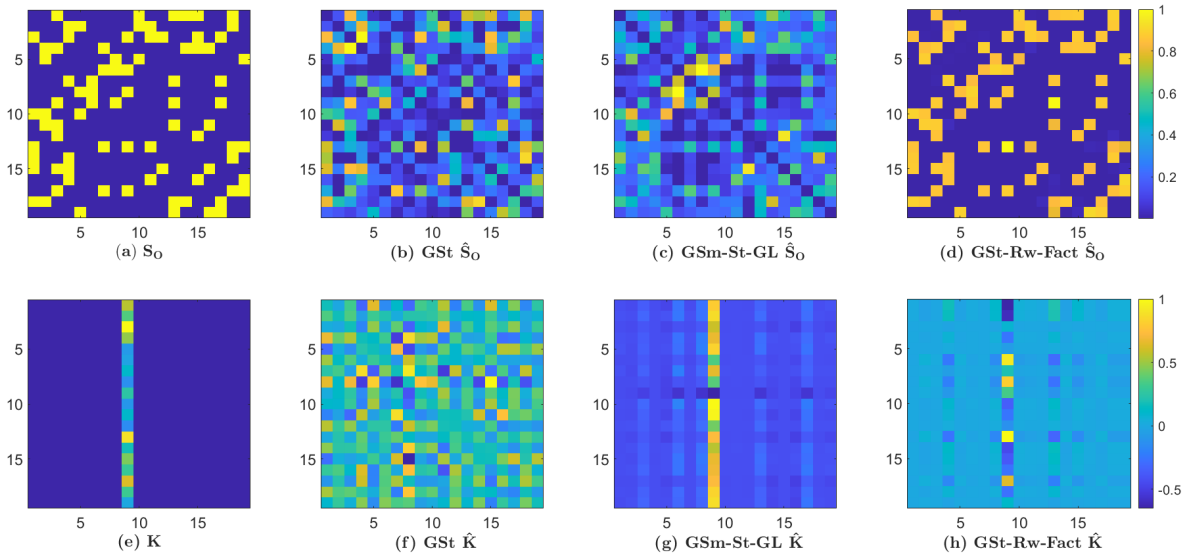

Figure 4.5: Graphical representation of the estimates of matrices $\mathbf{S}_{\mathcal{O}}$ (top row) and $\mathbf{K} = \mathbf{C}_{\mathcal{OH}}\mathbf{S}_{\mathcal{OH}}^{\top}$ (bottom row) for different algorithms that assume the observed signals to be stationary on the graph, with $N = 20$ and $H = 1$. The ground-truth matrices $\mathbf{S}_{\mathcal{O}}$ and $\mathbf{K}$ are represented in the first column [cf. panels (a) and (e)]. Analogously, the estimates $\hat{\mathbf{S}}_{\mathcal{O}}$ and $\hat{\mathbf{K}}$ generated by GSt are represented in panels (b) and (f), those generated by GSm-St-GL in panels (c) and (g), and those generated by GSt-Rw-Fact in (d) and (h).

Number of hidden variables. This experiment investigates the effect of the hidden nodes on the ability of our approach to recover the true graph topology. To that end, we vary the number of hidden variables H. We consider both the $\mathbf{C}_{poly}$ and $\mathbf{C}_{MRF}$ models for the observations, assume that the covariance matrices can be perfectly estimated, and select the set of hidden nodes as those with the minimum degree. The results are shown in Fig. 4.6, where the x-axis represents the number of hidden variables and the y-axis the proportion of graphs successfully recovered. The results in Fig. 4.6 confirm that larger values of H render the inference problem more challenging,

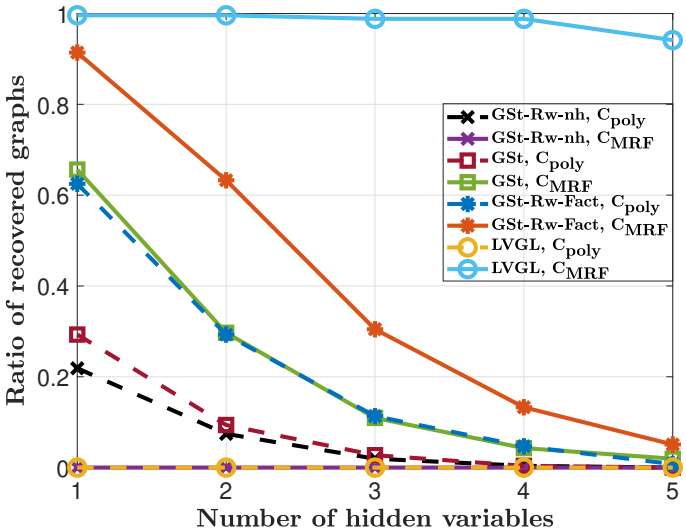

Figure 4.6: The ratio of recovered graphs averaged over 200 realizations of random graphs with $N = 20$ and stationary observations. Obtained results regarding the impact of increasing the number of hidden variables H for a scenario with perfectly known covariance matrices.

leading to a worse ratio of recovered graphs. We also observe that for the $\mathbf{C}_{MRF}$ model, LVGL achieves the best performance, especially when H increases. This is not surprising since the LVGL is tailored for this specific type of signal generation. On the other hand, LVGL fails to recover any graph when the observed signals follow the more general $\mathbf{C}_{poly}$ model. This contrasts with the GSt and GSt-Rw-Fact methods proposed in this chapter, which recover the graphs in both settings. For both $\mathbf{C}_{MRF}$ and $\mathbf{C}_{poly}$ models, the proposed algorithms outperform GSt-Rw-nh, which solves the same problem as GSt-Rw-Fact but ignores the presence of hidden variables. This behavior of GSt-Rw-nh was expected since, as the number of hidden variables increases, their influence is more significant and the stationarity constraint becomes less accurate. It is also worth noting that the results obtained in Fig. 4.6 outperform those presented in Fig. 4.2. This is due to the fact that graph stationarity imposes more structure on the observed signals than graph smoothness, at the expense of needing more observations to accurately estimate the covariance matrices.

Sample covariance matrix. The next step is to assess the effect of replacing the true covariance matrix with its sampled estimate $\hat{\mathbf{C}}_{\mathcal{O}} = \frac{1}{R}\mathbf{X}_{\mathcal{O}}\mathbf{X}_{\mathcal{O}}^{\top}$. The number of hidden variables is set to $H = 1$, both $\mathbf{C}_{MRF}$ and $\mathbf{C}_{poly}$ generative models are tested, the signals are assumed to be Gaussian and zero mean, and all other parameters are set as in the default test-case scenario. Fig. 4.7 illustrates the ratio of recovered graphs as the number of samples R varies. Clearly, the larger the value of R the better the estimate of $\hat{\mathbf{C}}_{\mathcal{O}}$. Analyzing the results in Fig. 4.7, we observe that, when using $\mathbf{C}_{MRF}$, LVGL obtains the best performance and needs the least number of samples to achieve its best ratio of recovered graphs. As noted in the previous experiment, we also observe that LVGL is incapable of recovering graphs when the observations are generated using the $\mathbf{C}_{poly}$ model. On the other hand, GSt and GSt-Rw-Fact achieve good performance for both covariance models, even though they need a higher number of samples.

If we focus on the covariance model $\mathbf{C}_{MRF}$, GSt-Rw-Fact achieves a performance close to that of LVGL. This behavior is consistent with the one observed in scenarios where all nodes were observed, and latent variables did not exist [4]. Upon comparing the results achieved by GSt and GSt-Rw-Fact, the experiments reveal that GSt: i) needs a higher value of R than GSt-Rw-Fact to

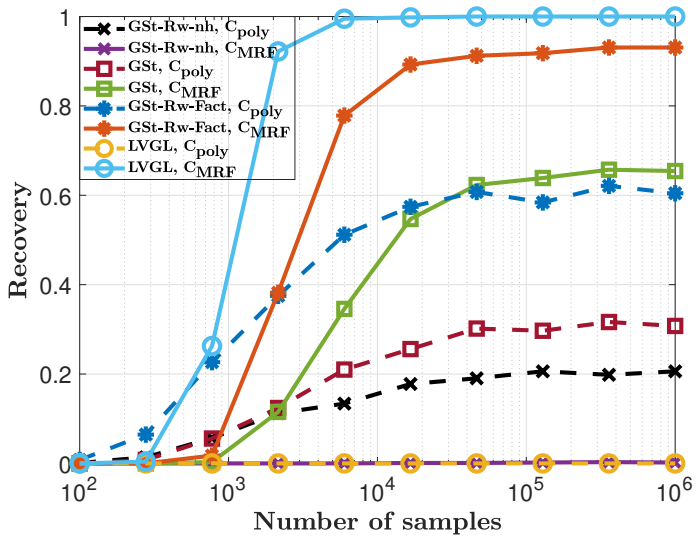

Figure 4.7: The ratio of recovered graphs averaged over 200 realizations of random graphs with $N = 20$ and stationary observations. Obtained results regarding the impact of increasing the number of signal observations R when using the sample covariance matrix.

achieve the same performance; and ii) converges to a worse ratio of recovered graphs. To conclude, as mentioned in Fig. 4.6, ignoring the presence of hidden variables fails to capture the true structure of the inference problem, impacting the recovery ratio of GSt-Rw-nh for both covariance models. This is consistent with the results shown in previous experiments and, once again, illustrates the benefits of incorporating additional structure and using more sophisticated regularizers.

4.6.3 Synthetic experiments based on smooth and stationary signals

To close the experiments based on synthetic data, we consider here the case where the observed signals are simultaneously smooth and stationary on the unknown graph and evaluate the schemes proposed in Section 4.5. As done in the smooth-based experiments, we create the graph signals as $\mathbf{X} = \mathbf{V}\mathbf{J}$, with $\mathbf{J}$ sampled from $\mathcal{N}(\mathbf{0}, \mathbf{\Lambda}^{\dagger})$. We note that the covariance of $\mathbf{X}$ is given by $\mathbf{C} = (\mathbf{L}^{\dagger})^2$, which is certainly a polynomial of the GSO provided that we set $\mathbf{S} = \mathbf{L}$. In other words, while the signals generated in Section 4.6.1 were already stationary on the graph, none of the algorithms leveraged that existing structure.

Leveraging graph stationarity and smoothness. Hence, the goal of this last synthetic experiment is to assess the benefits of considering the mentioned structure in the proposed approaches. To that end, we compare the schemes GSm-LR and GSm-GL, which only assume that the signals are smooth on the graph, with GSm-St-LR, which corresponds to (4.19) with $\gamma_{2,1} = 0$, and GSm-St-GL, which corresponds to the (4.19) with $\gamma_* = 0$. Additionally, we compare the aforementioned algorithms with two schemes that do not consider the presence of hidden variables, GSm and GSm-St-nh, with the latter assuming that the signals are both smooth and stationary on the graph. Note that GSm-St-LR and GSm-St-GL are, respectively, versions of GSm-LR and GSm-GL that account for the stationarity of the signals. Fig. 4.8 shows the ratio of recovered graphs as the number of hidden variables increases for the different algorithms. The advantages of including the stationarity assumption are clear, since, even for $H = 3$, the stationary-aware

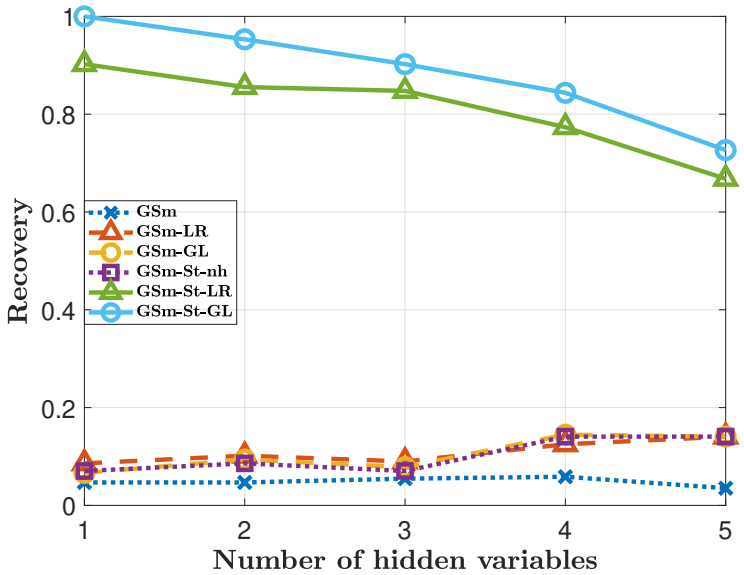

Figure 4.8: The ratio of recovered graphs averaged over 200 realizations of random graphs with $N = 20$ and stationary observations. Obtained results regarding the impact of increasing the number of hidden variables H when the inputs are not only stationary but also smooth signals.

algorithms are able to perfectly recover more than 60% of the generated graphs. In contrast, the algorithms that ignore stationarity and account only for smoothness recover correctly less than 20% of the graphs. As expected, including additional information about the observed signals endows the optimization problem with more structure and results in better estimates. Focusing on the importance of considering the presence of hidden variables, we observe that i) GSm-St-LR and GSm-St-GL outperform GSm-St-nh; and ii) GSm-LR and GSm-GL outperform GSm. As expected, not considering the presence of hidden variables leads to an inaccurate estimation of the GSO, decreasing the percentage of recovered graphs by GSm and GSm-St-nh. If, as in Section 4.6.1, the recovery performance is measured using the F*score* associated with individual links, then the differences narrow, with GSm-LR and GSm-GL achieving a (median) F*score* of around 0.95 and GSm-St-LR and GSm-St-GL a F*score* that is basically 1.

4.6.4 Infering graph structure from real datasets

We close this section by evaluating our proposed approaches and comparing their recovery performance with existing alternatives in the literature using three real-world datasets.

Infering meteorological graph from temperature data. We start by considering the average monthly temperature collected at 88 measuring stations in Switzerland during the period between 1981 and 2010 [102]. This leads to a set of signals $\mathbf{X} \in \mathbb{R}^{88\times 12}$, with 12 signals that represent the monthly average temperatures measured at the 88 weather stations. The goal of the experiment is to use these observations to infer a graph where stations with similar temperature patterns across the year are connected. While using the geographical graph based on physical distances between the stations can be a more natural (non-data-based) solution to the problem at hand, one must note that Switzerland is a steep terrain. As a result, two nearby stations do not necessarily record similar temperatures across the year, since, for instance, their difference in altitude is large. Motivated by this and, as also done in [85], we build the "ground-truth" graph upon considering the similarity between stations in terms of their altitude. More specifically, in this experiment, we consider that two stations are connected with a unitary weight if their altitude difference is smaller than 300 meters. As we want to infer the best-represented graph from the available smooth signals and also take into account the presence of hidden variables, we are going to assume that $\mathcal{O} = \{1, ..., 20\}$, so that only the 20 first stations are observed, with our goal being inferring the connections between those stations.

We leverage the schemes developed in Section 4.3 (GSm-LR and GSm-GL) and Section 4.5 (GSm-St-LR and GSm-St-GL) to learn the graph associated with the observed nodes from the temperature measurements. To facilitate comparisons, the evaluation metrics used here are the same as those in [85], namely F*score*, $precision$, $recall$, and normalized mutual information (NMI);

Table 4.1: Performance achieved by the schemes GL-SigRep ([85]), GSm-LR (Section 4.3), GSm-GL (Section 4.3), GSm-St-LR (Section 4.5) and GSm-St-GL (Section 4.5) when infering a meteorological graph.

Algorithms	F*score*	Precision	Recall	NMI
GL-SigRep	0.8800	0.9016	0.8594	0.5746
GSm-GL	0.9118	0.8611	0.9688	0.6647
GSm-LR	0.9130	0.8514	0.9844	0.6806
GSm-St-LR	0.9130	0.8514	0.9844	0.6806
GSm-St-GL	0.9130	0.8514	0.9844	0.6806

in addition, the GL-SigRep algorithm from [85] is used as a baseline. The results achieved by the optimal setting of the regularization constants for each of the algorithms are listed in Table 4.1. The main observation is that the explicit consideration of hidden variables when inferring the graph structure leads to better performance. Furthermore, we also observe that GSm-LR outperforms both GL-Sig-Rep and GSm-GL. It is also worth noticing that GSm-St-LR and GSm-St-GL obtain the same performance as GSm-LR, revealing that assuming stationarity for this dataset does not seem to further enhance the recovery results. Although this contrasts with the results from the synthetic experiments, it is not surprising since the number of available samples ($R = 12$) is smaller than the number of nodes, which leads to a rank-deficient $\hat{\mathbf{C}}_o$ and renders the commutativity constrain inefficient. Indeed, the fact of the covariance being rank-deficient was the reason for not testing the algorithms developed in Section 4.5 in this experiment.

Infering structural properties of proteins. In this case, our goal is to identify the structural properties of proteins from a mutual information graph of the co-variation of amino-acid residues simulating the presence of hidden variables. We have access to the mutual information matrix of protein BPT1 BOVIN and also to the binary ground-truth contact network built by medical experts, see [104] and [103] for details. The original dimension of both matrices is 53×53, but in our hidden-variable setup, we consider that we can only observe a submatrix of size 41×41 and leave the other 12 nodes as hidden. The y-axis in Fig. 4.9 represents the fraction of the real contact edges recovered for several schemes and the x-axis represents the number of top-edge predictions. This way, a fraction of recovered edges of 0.6 indicates that if we consider the estimated 100 links with the highest weight, 60 of them match the ground-truth links. Five different algorithms are considered: GSt-Rw-Fact (Section 4.5); GSt no hidden (which approaches the topology-identification problem with stationarity assumptions but ignoring the presence of hidden variables [48]); LVGL; network deconvolution [104]; and mutual information, with the last two being baselines that have been advocated for this particular dataset. The best performance is achieved by the scheme GSt-Rw-Fact that is accounting for the presence of hidden variables, showcasing the benefits of a more robust formulation. Interestingly, we also observe that even though LVGL accounts for hidden variables, it leads to the worst recovery performance, illustrating the relevance of using topology-inference algorithms that go beyond classical graphical models

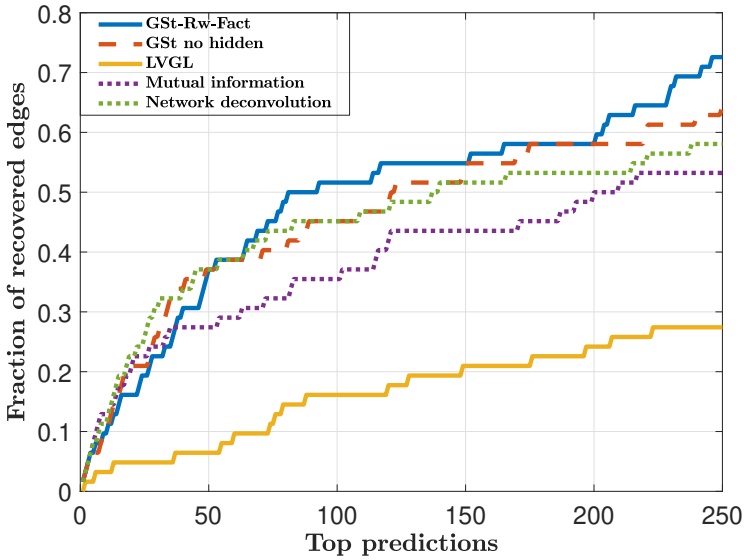

Figure 4.9: Fraction of the real contact edges between amino-acids [103] recovered for each method as a function of the number of edges considered.

when dealing with real datasets.

Infering graph from voting data. In this final real-data experiment, our goal is to learn a political graph from voting data [105]. More specifically, the 26 cantons of Switzerland are considered as nodes and the percentage of votes of each canton for 37 related initiatives (submitted to the voters between 2008 and 2012) are considered as graph signals. To validate the estimated graphs, we require a ground truth that reflects the level of association between the political preferences of the cantons. Since defining such a ground-truth graph is not evident, our first experiment is to compare the graph estimated by GSm-St-LR with the one estimated by GL-SigRep, with the latter being equivalent to the solution implemented in [85]. Once the two graphs are estimated, we apply spectral clustering to obtain 3 clusters that group the 26 cantons according to their voting patterns.

Figs. 4.10.a and 4.10.b show the graphs estimated by GL-SigRep and GSm-St-LR respectively, along with the 3 clusters. To identify the cluster each node belongs to, we used 3 colors (blue, red, and yellow). Figs. 4.10.a and 4.10.b reveal that, while GSm-St-LR estimates a sparser graph

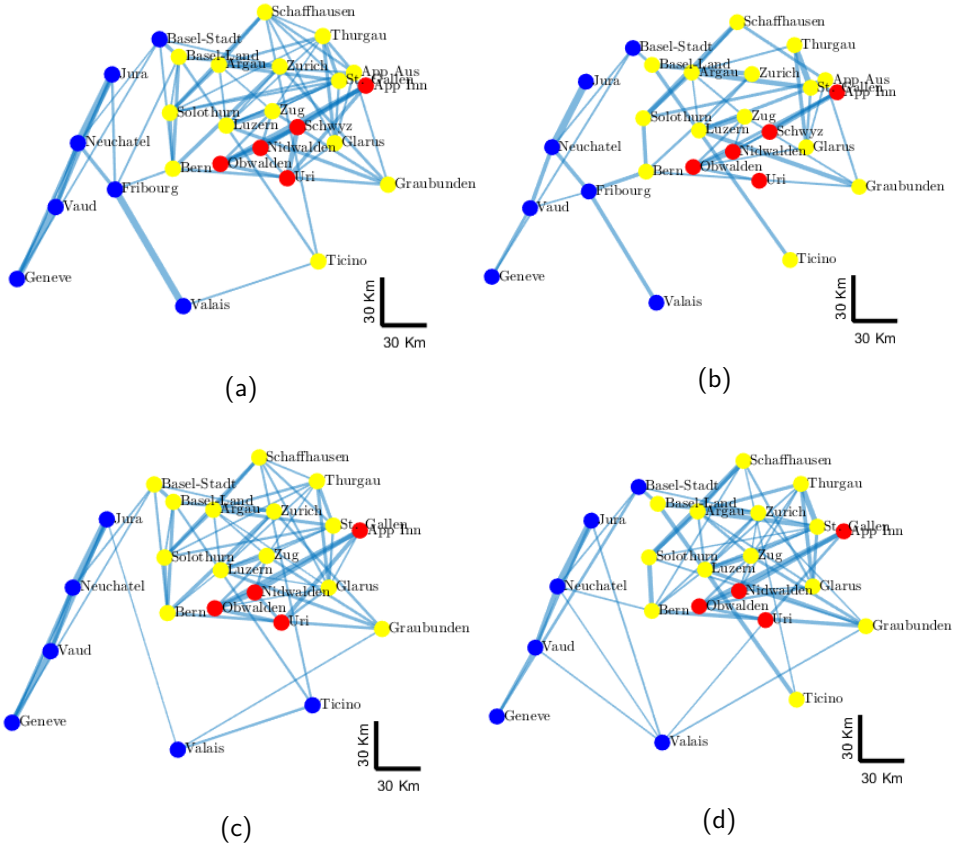

Figure 4.10: Political associations among the 26 cantons of Switzerland estimated from electoral data. The colors blue, yellow, and red denote the 3 main clusters identified using spectral clustering and represent if a canton is against, supports, or strongly supports the initiatives, respectively. The two left-most graphs correspond to the political association networks estimated by (a) GL-SigRep and (b) GSm-St-LR when the voting data of all 26 cantons is considered. While the graphs are slightly different, the way in which cantons are clustered is the same. The two right-most graphs, which have 23 nodes each, represent the association networks estimated by (c) GL-SigRep and (d) GSm-St-LR when the voting data of 3 cantons (one belonging to each of the clusters) is removed. We observe that, in this case, the clusters in (c) and (d) are not the same and that (c) is less robust to the presence of hidden data.

(in general less edges for each node) than GL-SigRep, the 3-cluster partition is the same for both graphs. Equally important, the clusters convey meaningful information about the electoral preferences of the cantons, implicitly validating the obtained graphs. More specifically: i) the cantons that are against the initiatives correspond to the blue cluster; ii) the cantons that strongly support the initiatives correspond to the red cluster; and iii) the cantons that moderately support the initiatives correspond to the yellow cluster.

The next step is to assess the influence of (and robustness to) hidden variables. To that end, we randomly remove the voting data associated with one canton from each cluster and re-estimate the two graphs. The estimation results for GL-SigRep and GSm-St-LR in the presence of 3 hidden variables are reported in Figs. 4.10.c and 4.10.d, respectively, with the particular realization shown corresponding to the removal of Fribourg, Appenzell Ausserrhoden, and Schwyz. Focusing first on the GSm-St-LR algorithm, the comparison of Figs. 4.10.b and 4.10.d reveals that, while the graphs change slightly (new weak links appear in Fig. 4.10.d to account for 2-hop relations that were broken after dropping the hidden nodes), the assignment of cantons to clusters does not change. On the other hand, for the GL-SigRep-based graphs (Figs. 4.10.a and 4.10.c), we observe that while the links barely change, two of the nodes (Basel-Stadt and Ticino) are assigned to a different cluster.

In other words, the results of Figs. 4.10.c and 4.10.d confirm that GSm-St-LR is more robust than GL-SigRep to the presence of hidden variables since it is able to maintain the same clustering pattern even when hidden nodes are present.[4]

4.7 Conclusions

This chapter analyzed the problem of infering the graph of a network from nodal signal observations in the presence of hidden (latent) nodes. To approach this ill-conditioned network topology inference task, we considered that the observed signals were (i) smooth on the sought graph; (ii) stationary on the graph; and (iii) a combination of the two previous assumptions. To render the problem tractable, we further assumed that the number of hidden variables was much smaller than the number of observed nodes and formulated constrained optimization problems that accounted for the topological and signal constraints. The key to handle the presence of hidden nodes was to consider block-matrix factorization approaches that led to sparse and low-rank constrained optimizations. Since several of the resulting formulations were non-convex, novel judicious convex relaxations were designed. The performance of the developed algorithms was evaluated in several synthetic and real-world datasets and the results were compared with alternatives from the literature.

4.8 Appendix: Proof of Proposition 1

Key to our proof are the results from [99], which guarantee the convergence of BSUM algorithms to a stationary point.

We aim to show that our proposed algorithm satisfies the conditions specified in [99, Th. 1b].

[4]For conciseness, only one experiment with three missing nodes is presented, but the difference in terms of robustness is also observed if the hidden nodes at hand change or if the value of H is either 1 or 2. We refer interested readers to the repository provided in footnote 3, which allows them to run the experiments for any desired configuration.

To that end, let $f(\mathbf{y})$ represent the objective function in (4.14), with $\mathbf{y} := [\mathbf{y}_1^\top, \mathbf{y}_2^\top, \mathbf{y}_3^\top]^\top$ and $\mathbf{y}_1 := \mathrm{vec}(\mathbf{S}_\mathcal{O})$, $\mathbf{y}_2 := \mathrm{vec}(\mathbf{S}_{\mathcal{OH}})$, $\mathbf{y}_3 := \mathrm{vec}(\mathbf{C}_{\mathcal{OH}})$ denoting the 3 blocks of variables considered in our algorithm. For each of the $B = 3$ block of variables $\mathbf{y}_b$, we approximate $f(\mathbf{y})$ by defining the functions $u_1(\mathbf{y}_1)$, $u_2(\mathbf{y}_2)$, and $u_3(\mathbf{y}_3)$, corresponding to the objective functions in (4.15), (4.16) and (4.17), respectively. Also, recall that $\mathcal{Y}^*$ denotes the set of stationary points of $f(\mathbf{y})$ and that $\mathbf{y}^{(t)} := [(\mathbf{y}_1^{(t)})^\top, (\mathbf{y}_2^{(t)})^\top, (\mathbf{y}_3^{(t)})^\top]^\top$ is the solution obtained after running t iterations of our algorithm.

With the previous definitions in place, the assumptions required to ensure the convergence of our algorithm are the following.

(AS A) The approximation functions $u_b(\mathbf{y}_b)$ must be a global upper bound of $f(\mathbf{y})$ and the first order behavior of $u_b(\mathbf{y}_b)$ and $f(\mathbf{y})$ must be the same.

(AS B) *The function $f(\mathbf{y})$ must be regular (cf. [99]) at every point in $\mathcal{Y}^*$.*

(AS C) *The level set $\mathcal{Y}^{(0)} = \{\mathbf{y} \mid f(\mathbf{y}) \leq f(\mathbf{y}^{(0)})\}$ is compact.*

(AS D) *The problems in* (4.15)-(4.17) *must have a unique solution for any point $\mathbf{y}^{(t)} \in \mathcal{Y}^*$ for at least two of the blocks.*

We address each of the four assumptions separately, proving that our approach satisfies all of them.

Assumption (***AS A***) requires the surrogate functions $u_b(\mathbf{y}_b)$ to be global upper bounds of $f(\mathbf{y})$. For the first block ($b = 1$), we approximate $f(\mathbf{y})$ with the Taylor series of order 1 of the logarithmic penalty, given by

$$\begin{aligned}\tilde{u}_1(\mathbf{y}_1) = &\sum_{i=1}^{O^2} \log\left(|[\mathbf{y}_1^{(t)}]_i| + \delta\right) \\ &+ \sum_{i=1}^{O^2} \frac{\mathsf{sign}([\mathbf{y}_1^{(t)}]_i)}{|[\mathbf{y}_1^{(t)}]_i| + \delta}\left([\mathbf{y}_1]_i - [\mathbf{y}_1^{(t)}]_i\right) + \rho f_c(\mathbf{y}_1),\end{aligned} \tag{4.21}$$

where f_c denotes the commutativity penalty in (4.15). Since the entries of $\mathbf{y}_1^{(t)}$ are always either positive or negative [cf. (4.3) and (4.4)], we have that $\mathsf{sign}([\mathbf{y}_1^{(t)}]_i)[\mathbf{y}_1]_i = |[\mathbf{y}_1]_i|$. After dropping the constant terms, we obtain

$$u_1(\mathbf{y}_1) = \sum_{i=1}^{O^2} \frac{|[\mathbf{y}_1]_i|}{|[\mathbf{y}_1^{(t)}]_i| + \delta} + \rho f_c(\mathbf{y}_1), \tag{4.22}$$

which is the objective function in (4.15). Because the $\log$ is a concave differentiable function it follows that its Taylor series of order one constitutes a global upper bound. Therefore, u_1 satisfies (***AS A***). The proof for u_2 is equivalent to the proof for u_1 so it is omitted for brevity. Lastly, $u_3(\mathbf{y}_3) = f(\mathbf{y})$ when the blocks $\mathbf{y}_1$ and $\mathbf{y}_2$ remain constant, so it also satisfies the requirements, and hence, (***AS A***) is fulfilled.

To proof (***AS B***), according to the definition of regular functions presented in [99], it suffices to show that the non-smooth parts of $f(\mathbf{y})$ are separable across the different blocks of variables. To that end, we recall that $\mathbf{y}_1 := \mathrm{vec}(\mathbf{S}_\mathcal{O})$, $\mathbf{y}_2 := \mathrm{vec}(\mathbf{S}_{\mathcal{OH}})$ and $\mathbf{y}_3 := \mathrm{vec}(\mathbf{C}_{\mathcal{OH}})$, and decompose f as $f = g_A + g_B + g_C$, with functions g_A, g_B and g_C being defined as

- $g_A(\mathbf{S}_\mathcal{O}, \mathbf{S}_{\mathcal{OH}}, \mathbf{C}_{\mathcal{OH}}) = \eta\|\mathbf{S}_{\mathcal{OH}}\|_F^2 + \eta\|\mathbf{C}_{\mathcal{OH}}\|_F^2 + \rho\|\hat{\mathbf{C}}_\mathcal{O}\mathbf{S}_\mathcal{O} + \mathbf{C}_{\mathcal{OH}}\mathbf{S}_{\mathcal{OH}}^\top - \mathbf{S}_\mathcal{O}\hat{\mathbf{C}}_\mathcal{O} - \mathbf{S}_{\mathcal{OH}}\mathbf{C}_{\mathcal{OH}}^\top\|_F^2$, where g_A is a smooth function,

- $g_B(\mathbf{S}_{\mathcal{O}}) = \sum_{i,j=1}^{O} \log(|[\mathbf{S}_{\mathcal{O}}]_{ij}| + \delta)$, where g_B is a non-smooth function,
- $g_C(\mathbf{S}_{\mathcal{OH}}) = \sum_{i,j=1}^{O,H} \log(|[\mathbf{S}_{\mathcal{OH}}]_{ij}| + \delta)$, where g_C is a non-smooth function.

Since the non-smooth terms appear in $g_B(\mathbf{S}_{\mathcal{O}})$, which only involves variables of the first block $\mathbf{y}_1 = \mathrm{vec}(\mathbf{S}_{\mathcal{O}})$, and $g_C(\mathbf{S}_{\mathcal{OH}})$, which only involves variables of the second block $\mathbf{y}_2 = \mathrm{vec}(\mathbf{S}_{\mathcal{OH}})$, it follows that the function $f(\mathbf{y})$ is regular for all feasible points.

Next, we show that the level set $\mathcal{Y}^{(0)} = \{\mathbf{y} \mid f(\mathbf{y}) \leq f(\mathbf{y}^{(0)})\}$ is compact as required by (***AS C***). First, note that the entries of $\mathbf{S}_{\mathcal{O}}$ and $\mathbf{S}_{\mathcal{OH}}$ are continuous subsets of $\mathbb{R}$ (e.g., $[\mathbf{S}_{\mathcal{O}}]_{ij}, [\mathbf{S}_{\mathcal{OH}}]_{ij} \in \mathbb{R}_+$ when $\mathcal{S} = \mathcal{A}$), and that $\mathbf{C}_{\mathcal{OH}} \in \mathbb{R}^{O \times H}$, so $f(\mathbf{y})$ is continuous. Moreover, since we have that $f(\mathbf{y}) \leq f(\mathbf{y}^{(0)})$, this implies that the continuous functions $\log(|[\mathbf{S}_{\mathcal{O}}]_{ij}| + \delta)$, $\log(|[\mathbf{S}_{\mathcal{OH}}]_{ij}| + \delta)$, and $\|\mathbf{C}_{\mathcal{OH}}\|_F^2$ are all bounded, rendering the domain of $f(\mathbf{y})$ bounded. Therefore, it follows that the level set $\mathcal{Y}^{(0)}$ is compact.

Finally, since the optimization problems in (4.16) and (4.17) are strictly convex, two of the three problems have unique solutions, satisfying (***AS D***) and concluding the proof.

Chapter 5

Joint Graph Inference from Stationary Graph Signals with Hidden Nodes

In this chapter, we explore the field of network topology inference by considering scenarios involving multiple interconnected networks and the presence of hidden nodes. It is evident from the previous chapter that ignoring the presence of hidden nodes can significantly limit the effectiveness of graph learning tasks. Consequently, in this chapter, we propose an approach that jointly estimates the graph in the context of multiple interconnected networks by considering the presence of hidden nodes. For the proposed method we assume that observed signals are stationary on the graph and formalize the relationship between observed and hidden nodes, which is given by the assumed signal model. We also exploit graph similarities, both between observed and hidden nodes, and for doing so we employ a regularization inspired by the group Lasso penalty. This approach has the potential to improve the graph learning performance of existing methods by i) exploiting the inherent relationships between these connected graphs and ii) modeling the effect of the hidden nodes. In order to show the benefits of the proposed approach we present mathematical and numerical analyses, including conditions for the recovery of sparse solutions and the associated error bounds. We conclude this chapter by evaluating the performance of the proposed method through simulations on synthetic and real data, emphasizing the impact of hidden variables on multiple network topology inference.

5.1 Introduction

As mentioned in the previous chapters, the task of *network topology inference*, has emerged as a vibrant research area with GSP [15, 16, 19, 106]. A crucial assumption for learning the graph topology is the statistical relationship between the signals and the unknown topology. Different assumptions lead to different methods, with noteworthy examples including correlation networks and GMRF [3,13,59], smooth (local total variation) models [36,38,107], GSP-based approaches [4, 62, 93], and models with more elaborate graph priors [108, 109]. A common feature of the previous works is that they focus on learning a single graph. However, many contemporary setups involve *multiple related networks*, each with a subset of signals. Some examples include brain analytics, where observations from different *patients* are used to estimate their brain functional networks; social networks, where the same set of users may present different types of *interactions*; or multi-hop communication networks in dynamic environments, where a network needs to be inferred for each

time instant. Intuitively, in situations where several closely related networks exist, approaching the problem in a joint fashion can boost the performance of network topology inference by harnessing the relationships among graphs [65, 66, 110–113].

Despite the clear benefits, joint network topology inference approaches usually assume that observations from every node are available, which is often not the case. In many relevant scenarios, the observed signals correspond only to a subset of the nodes in the whole graph, while the remaining nodes stay unobserved or *hidden*. Ignoring the presence of the hidden nodes can drastically hinder the performance of the graph learning algorithms. Nevertheless, accounting for their influence is not a trivial endeavor since the inference task becomes ill-posed. For *single network* inference, some works dealing with this challenging setting include graphical models [63, 64], inference of linear Bayesian networks [88], nonlinear regression [89], and stationary-based algorithms [5, 42]. However, the presence of hidden nodes is yet to be addressed for several unknown graphs. Since the key to joint topology inference is exploiting the similarity of the graphs, it is crucial to model the influence of the hidden nodes to measure the graph similarity between nodes that remain unobserved.

To this end, we propose a topology inference method that simultaneously performs *joint estimation of multiple graphs* and *accounts for the presence of hidden variables*. Under the assumption that the observed signals are realizations of a random process that is *stationary* on the graph [16, 46], we formalize the relationship between the nodal observations and the unknown networks under the influence of the hidden nodes. The joint formulation necessitates exploiting graph similarities, not only with respect to observed nodes but also to hidden ones. To accomplish this, we carefully model the structure associated with latent variables and exploit it with a regularization inspired by the group Lasso penalty [95]. Finally, we conduct thorough mathematical and numerical analyses of the proposed approach, where we show the conditions under which it recovers the sparsest solution and bounds the error of the estimated graphs, and we evaluate its performance and the influenceof the hidden variables through simulations with synthetic and real-world data.

Related work and contributions. Early methods for joint graph learning were introduced in [65] assuming that observations follow a GMRF and, later on, in [66] followed by a joint inference method for graph stationary signals. However, both works assumed that observations from the whole graphs were available. At the same time, the influence of hidden nodes when learning a single graph was studied in [63] and [5] assuming that the observations adhered respectively to a GMRF or a graph-stationary model. On the other hand, the relevant task of learning several graphs in the presence of hidden nodes has only been considered under GMRF assumptions in the preliminary results from [6]. In contrast, in this chapter, we (i) build upon our previous work from [43] for joint graph learning with hidden variables under the more lenient assumption of stationary observations; and (ii) develop a theoretical analysis to characterize how the hidden nodes influence the quality of the estimated graphs. Finally, note that GMRF and graph stationarity are intrinsically different models for the observations, resulting in materially different inference algorithms and, even more relevant for the problem at hand, requiring different methods to encourage graph similarities with respect to both observed and hidden nodes.

To summarize, our main contributions are:

- We design a convex optimization problem to jointly learn the topology of several related graphs in the presence of hidden variables under graph-stationary observations.
- We rely on a regularization inspired by group Lasso to model the similarity between both hidden and observed nodes.

- We derive theoretical recoverability guarantees and bound the error of the estimated graphs when considering the presence of hidden nodes.
- We evaluate the performance of the proposed approach and compare it with state-of-the-art alternatives in synthetic and real-world datasets.

Outline. The remainder of this chapter is organized as follows. We introduce in Section 5.2 the task of learning graphs in the presence of hidden nodes. In Section 5.3 we present our proposed optimization problem that accounts for hidden nodes, along with its convex relaxation. We provide theoretical guarantees for the viability and performance of our method in Section 5.4, which are validated by several synthetic and real-world experiments in Section 5.5. In Section 5.6 a concluding discussion is provided about the work presented in this chapter. Finally, in Section 5.7, Section 5.8, and Section 5.9 we present the detailed proof for the results associated with the theoretical recoverability guarantees and the error bounds of the estimated graphs stated in Section 5.4.

5.2 Inference of multilayered graphs with latent variables

Let there be a set of K undirected networks $\{\mathcal{G}^{(k)}\}_{k=1}^K$ on the same set $\mathcal{V}$ of N nodes with GSOs denoted as $\{\mathbf{S}^{*(k)}\}_{k=1}^K$. We assume that for each graph there exists a set with R_k realizations of a *stationary* graph signal collected in data matrices $\mathbf{X}^{(k)} \in \mathbb{R}^{N\times R_k}$, where the R_k columns contain the nodal observations on the k-th graph. For a signal $\mathbf{x}^{(k)}$ on the k-th graph, its covariance matrix is denoted by $\mathbf{C}^{(k)} = \mathbb{E}[\mathbf{x}^{(k)}(\mathbf{x}^{(k)})^\top]$. We further assume that for every graph we do not know the entire data matrix $\mathbf{X}^{(k)}$ but only observe signal values on a subset $\mathcal{O} \subset \mathcal{V}$ of O nodes, where $\mathcal{H} := \mathcal{V}\backslash\mathcal{O}$ denotes the set of H hidden nodes. Our goal is to *estimate the subnetwork of each network $\mathcal{G}^{(k)}$ induced by $\mathcal{O}$ from partially observed graph signals*.

Under this setting, we can now formalize the task of estimating the network structure at the node subset $\mathcal{O}$ that is encoded in the GSOs $\{\mathbf{S}^{*(k)}\}_{k=1}^K$. Without loss of generality, we partition the GSO and the covariance matrix of each network as

$$\mathbf{S}^{*(k)} = \begin{bmatrix} \mathbf{S}^{*(k)}_{\mathcal{O}} & \mathbf{S}^{*(k)}_{\mathcal{O}\mathcal{H}} \\ \mathbf{S}^{*(k)}_{\mathcal{H}\mathcal{O}} & \mathbf{S}^{*(k)}_{\mathcal{H}} \end{bmatrix}, \quad \mathbf{C}^{(k)} = \begin{bmatrix} \mathbf{C}^{(k)}_{\mathcal{O}} & \mathbf{C}^{(k)}_{\mathcal{O}\mathcal{H}} \\ \mathbf{C}^{(k)}_{\mathcal{H}\mathcal{O}} & \mathbf{C}^{(k)}_{\mathcal{H}} \end{bmatrix}, \tag{5.1}$$

where $\mathbf{S}^{*(k)}_{\mathcal{O}\mathcal{H}} = (\mathbf{S}^{*(k)}_{\mathcal{H}\mathcal{O}})^\top$ and $\mathbf{C}^{(k)}_{\mathcal{O}\mathcal{H}} = (\mathbf{C}^{(k)}_{\mathcal{H}\mathcal{O}})^\top$ by the symmetry of $\mathbf{S}^{*(k)}$ and $\mathbf{C}^{(k)}$. The submatrices $\mathbf{S}^{*(k)}_{\mathcal{O}} \in \mathbb{R}^{O\times O}$ and $\mathbf{S}^{*(k)}_{\mathcal{H}} \in \mathbb{R}^{H\times H}$ encode the connectivity of the subnetworks of $\mathcal{G}^{(k)}$ induced by $\mathcal{O}$ and $\mathcal{H}$, respectively, while $\mathbf{S}^{*(k)}_{\mathcal{O}\mathcal{H}} \in \mathbb{R}^{O\times H}$ represents the edges connecting observed nodes to hidden nodes. We similarly define $\mathbf{C}^{(k)}_{\mathcal{O}}$, $\mathbf{C}^{(k)}_{\mathcal{H}}$, and $\mathbf{C}^{(k)}_{\mathcal{O}\mathcal{H}}$. Given the partitions in (5.1), we aim to estimate the subnetworks encoded in $\{\mathbf{S}^{*(k)}_{\mathcal{O}}\}_{k=1}^K$.

We also partition each $\mathbf{X}^{(k)}$ to be conformal with $\mathbf{S}^{*(k)}$ and $\mathbf{C}^{(k)}$ as $\mathbf{X}^{(k)} = [\mathbf{X}^{(k)\top}_{\mathcal{O}}, \mathbf{X}^{(k)\top}_{\mathcal{H}}]^\top$, where $\mathbf{X}^{(k)}_{\mathcal{O}} \in \mathbb{R}^{O\times R_k}$ is the data matrix containing the partially observed graph signals and $\mathbf{X}^{(k)}_{\mathcal{H}} \in \mathbb{R}^{H\times R_k}$ remains unknown. We can thus apply the partially observed *stationary* graph signals $\mathbf{X}^{(k)}_{\mathcal{O}}$ and the commutative relationship $\mathbf{C}^{(k)}\mathbf{S}^{*(k)} = \mathbf{S}^{*(k)}\mathbf{C}^{(k)}$ as described in Section 2.4 to recover the structure in $\mathbf{S}^{*(k)}_{\mathcal{O}}$. Given the problem setting, we can now formalize our joint topology inference problem in the presence of hidden nodes as follows.

Problem 1 *Given the sets $\{\mathbf{X}^{(k)}_{\mathcal{O}}\}_{k=1}^K$ of graph signal values at the observed nodes for each of the K graphs, recover $\{\mathbf{S}^{*(k)}_{\mathcal{O}}\}_{k=1}^K$ under the following assumptions:*

(AS1) the number of hidden nodes H is much smaller than the number of observed nodes, that is, $H \ll O$;
(AS2) the signals in $\mathbf{X}^{(k)}$ are realizations of a process that is stationary in $\mathbf{S}^{*(k)}$; and
(AS3) the GSOs $\mathbf{S}^{*(k)}$ and $\mathbf{S}^{*(k')}$ are sparse and have similar sparsity patterns.

We elaborate on the implications of the assumptions. The first assumption *(AS1)* ensures the tractability of the problem. When most of the nodes in the graph are observed, the covariance submatrix $\mathbf{C}_{\mathcal{O}}^{(k)}$ sufficiently characterizes the structure of $\mathbf{S}_{\mathcal{O}}^{*(k)}$. Importantly, under $H \ll O$, the matrix product $\mathbf{C}_{\mathcal{OH}}^{(k)}\mathbf{S}_{\mathcal{HO}}^{*(k)}$ is low-rank, a crucial result for inferring $\mathbf{S}_{\mathcal{O}}^{*(k)}$, which is also assumed in different single graph topology inference approaches. Assumption *(AS2)* establishes a global relationship between the graph signals $\mathbf{X}^{(k)}$ and the unknown graph structure $\mathbf{S}^{*(k)}$, including both observed and hidden nodes. This assumption enables us to specify how the hidden nodes affect $\mathbf{X}^{(k)}$ by considering the connectivity between observed and hidden nodes encoded in $\mathbf{S}_{\mathcal{OH}}^{*(k)}$ from (5.1) and the commutative relationship $\mathbf{C}^{(k)}\mathbf{S}^{*(k)} = \mathbf{S}^{*(k)}\mathbf{C}^{(k)}$. The final assumption *(AS3)* guarantees that all K graphs have similar edge connectivity patterns across all the shared node set $\mathcal{V}$. Not only can we then benefit from jointly inferring the observed subnetworks, but we may also share hidden node information across all K graphs during inference. We naturally expect that the support of $\mathbf{S}_{\mathcal{O}}^{*(k)}$ will be similar across all K graphs [6, 65, 66]; however, it is important to also exploit the edgewise similarity for $\mathbf{S}_{\mathcal{OH}}^{*(k)}$ to account for connections between observed and hidden nodes.

Notice that for the simpler case where the set $\mathcal{H}$ of hidden nodes differs across graphs, *(AS3)* would allow us to exploit nodal observations from graph k that are hidden for graph k' to account for hidden nodes. However, in this work, we address the more challenging scenario in Problem 1, where there is a subset of nodes for which there are no direct observations for *any* graph. We rely on the statistical relationship between the graph signals and the graph topology to formulate a suitable optimization problem for jointly inferring the subnetworks in $\mathbf{S}_{\mathcal{O}}^{*(k)}$.

5.3 Joint graph inference with latent variables as a convex optimization problem

Network topology inference with stationary graph signals commonly exploits the commutativity of the graph signal covariance matrices and the GSOs. We also adopt this approach; however, unlike previous works, we cannot directly apply the commutative relationship due to the presence of hidden nodes. We must revisit the commutativity of $\mathbf{C}^{(k)}$ and $\mathbf{S}^{*(k)}$ with the partitions in (5.1) before introducing our inference problem with stationary graph signals. From stationarity *(AS2)*, we know that $\mathbf{S}^{*(k)}\mathbf{C}^{(k)} = \mathbf{C}^{(k)}\mathbf{S}^{*(k)}$ for all $k = 1, \dots, K$. From (5.1) it then follows that

$$\mathbf{C}_{\mathcal{O}}^{(k)}\mathbf{S}_{\mathcal{O}}^{*(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}\mathbf{C}_{\mathcal{O}}^{(k)} = (\mathbf{P}^{*(k)})^{\top} - \mathbf{P}^{*(k)} \tag{5.2}$$

for all $k = 1, \dots, K$, where $\mathbf{P}^{*(k)} := \mathbf{C}_{\mathcal{OH}}^{(k)}\mathbf{S}_{\mathcal{HO}}^{*(k)}$. The right-hand side of (5.2) fully accounts for the influence of hidden nodes. When $\mathbf{P}^{*(k)}$ is known, estimating $\mathbf{S}_{\mathcal{O}}^{*(k)}$ relies solely on the commutator on the left-hand side. This is similar to traditional network inference with stationary graph signals, where we also know the value of the commutator $\mathbf{C}^{(k)}\mathbf{S}^{*(k)} - \mathbf{S}^{*(k)}\mathbf{C}^{(k)} = \mathbf{0}_{N\times N}$.

With the prior structural information in place, we can approach estimating the subnetworks from sample covariance submatrices $\hat{\mathbf{C}}_{\mathcal{O}}^{(k)} = \frac{1}{R_k}\mathbf{X}_{\mathcal{O}}^{(k)}(\mathbf{X}_{\mathcal{O}}^{(k)})^{\top}$ by the following nonconvex optimization

problem

$$\begin{aligned}\min_{\{\mathbf{S}_{\mathcal{O}}^{(k)},\mathbf{P}^{(k)}\}_{k=1}^{K}} \quad & \sum_{k=1}^{K}\alpha_k\|\mathbf{S}_{\mathcal{O}}^{(k)}\|_0+\sum_{k<k'}\beta_{k,k'}\|\mathbf{S}_{\mathcal{O}}^{(k)}-\mathbf{S}_{\mathcal{O}}^{(k')}\|_0 \\ & +\sum_{k=1}^{K}\gamma_k\|\mathbf{P}^{(k)}\|_{2,1}+\sum_{k<k'}\eta_{k,k'}\left\|\begin{bmatrix}\mathbf{P}^{(k)}\\ \mathbf{P}^{(k')}\end{bmatrix}\right\|_{2,1} \\ \text{s.t. } & \textstyle\sum_{k=1}^{K}\|\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\mathbf{S}_{\mathcal{O}}^{(k)}-\mathbf{S}_{\mathcal{O}}^{(k)}\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}+\mathbf{P}^{(k)}-(\mathbf{P}^{(k)})^{\top}\|_F^2\le\epsilon^2, \\ & \mathbf{S}_{\mathcal{O}}^{(k)}\in\mathcal{S}, \end{aligned} \tag{5.3}$$

where we have introduced auxiliary matrices $\{\mathbf{P}^{(k)}\}_{k=1}^{K}$ to account for the right hand side of (5.2). We first discuss (5.3) as it relates to $\{\mathbf{S}_{\mathcal{O}}^{(k)}\}_{k=1}^{K}$. The first two terms in the objective of (5.3) encourage sparse subnetworks with similar sparsity patterns as in *(AS3)*. The second constraint encourages valid GSOs for $\mathbf{S}_{\mathcal{O}}^{(k)}$. In this work, we let the GSOs denote adjacency matrices, so we define

$$\mathcal{S}:=\left\{\mathbf{S}\ :\ \mathbf{S}=\mathbf{S}^{\top},\ \operatorname{diag}(\mathbf{S})=\mathbf{0},\ \textstyle\sum_j\mathbf{S}_{j1}=1\right\}, \tag{5.4}$$

where $\{\mathbf{S}_{\mathcal{O}}^{(k)}\}_{k=1}^{K}$ denote valid submatrices of nontrivial adjacency matrices, that is, $\mathbf{S}_{\mathcal{O}}^{(k)}\neq\mathbf{0}_{O\times O}$. While we select adjacency matrices as GSOs, problem (5.3) accommodates other GSOs, such as the graph Laplacian [4], under minor modifications.

We next discuss the auxiliary matrices $\{\mathbf{P}^{(k)}\}_{k=1}^{K}$. The first constraint encourages the commutativity in (5.2) with $\mathbf{P}^{(k)}$ as an approximation of $\mathbf{P}^{*(k)}=\mathbf{C}_{\mathcal{OH}}^{(k)}\mathbf{S}_{\mathcal{HO}}^{*(k)}$ to avoid a bilinear formulation. As will be discussed in Section 5.4, the upper bound ϵ accounts for both the sample covariance submatrix error and the difference between $\mathbf{P}^{(k)}$ and $\mathbf{P}^{*(k)}$. Thus, similarly to [6], we introduce the low-rank matrices $\mathbf{P}^{(k)}$ to replace entities that depend on hidden nodes. However, instead of using the standard convex surrogate for low-rankness given by the nuclear norm, we rely on the $\ell_{2,1}$ to impose additional structure on $\mathbf{P}^{(k)}$ based on the assumptions in Problem 1.

Precisely, the last two terms in the objective apply a group Lasso penalty via the $\ell_{2,1}$ norm [95], which evaluates the ℓ_1 norm of the vector containing the ℓ_2 norm of each column of the input matrix, that is, $\|\mathbf{P}^{(k)}\|_{2,1}=\sum_{i=1}^{O}\|\mathbf{P}_{\cdot,i}^{(k)}\|_2$. Recall that since $H\ll O$ by *(AS1)*, the matrix $\mathbf{P}^{*(k)}$ is not only low-rank but has sparse columns, hence the third term in the objective applying the $\ell_{2,1}$ norm to encourage column-sparsity in $\mathbf{P}^{(k)}$. While low-rank constraints are commonly implemented with the convex nuclear norm penalty [5], where solutions with sparse singular values are sought, we simultaneously promote low-rankness while encouraging column sparsity by the group Lasso penalty. Additionally, since the networks are assumed to have similar sparsity patterns by *(AS3)*, we expect that the column sparsity patterns of $\mathbf{P}^{*(k)}$ across networks will be similar, hence the fourth term in the objective.

As is common with optimization problems for sparse network inference, we introduce a convex relaxation of (5.3) that enjoys efficient solvability and theoretical guarantees. Our convex

formulation is

$$\begin{aligned}\min_{\{\mathbf{S}_{\mathcal{O}}^{(k)},\mathbf{P}^{(k)}\}_{k=1}^{K}} \quad & \sum_{k=1}^{K}\alpha_k\|\mathbf{S}_{\mathcal{O}}^{(k)}\|_1+\sum_{k<k'}\beta_{k,k'}\|\mathbf{S}_{\mathcal{O}}^{(k)}-\mathbf{S}_{\mathcal{O}}^{(k')}\|_1 \\ & +\sum_{k=1}^{K}\gamma_k\|\mathbf{P}^{(k)}\|_{2,1}+\sum_{k<k'}\eta_{k,k'}\left\|\begin{bmatrix}\mathbf{P}^{(k)}\\ \mathbf{P}^{(k')}\end{bmatrix}\right\|_{2,1} \\ \text{s. t. } & \textstyle\sum_{k=1}^{K}\|\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\mathbf{S}_{\mathcal{O}}^{(k)}-\mathbf{S}_{\mathcal{O}}^{(k)}\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}+\mathbf{P}^{(k)}-(\mathbf{P}^{(k)})^{\top}\|_F^2\leq\epsilon^2, \\ & \mathbf{S}_{\mathcal{O}}^{(k)}=(\mathbf{S}_{\mathcal{O}}^{(k)})^{\top},\ \mathrm{diag}(\mathbf{S}_{\mathcal{O}}^{(k)})=\mathbf{0},\ \text{for all}k=1,\ldots,K, \\ & \textstyle\sum_j[\mathbf{S}_{\mathcal{O}}^{(1)}]_{j1}=1,\end{aligned} \tag{5.5}$$

where we have removed the nonconvexities in (5.3) by substituting the ℓ_0 norms in the objective with convex ℓ_1 norms. We further specified the constraints according to (5.4) for valid adjacency submatrices. While the last constraint is valid to preclude trivial adjacency submatrices, it would not be viable for graph Laplacians as GSOs. However, the theoretical results in Section 5.4 still hold for graph Laplacian GSOs by replacing the last constraint in (5.4) to enforce valid graph Laplacian submatrices.

5.4 Theoretical results

We formalize the viability of the convex relaxation in (5.5) by presenting conditions under which the solutions to (5.3) and (5.5) are equivalent. We also compute an upper bound on the error of the solution to (5.5) and apply the bound to evaluate the effectiveness of (5.5) at accounting for hidden nodes.

5.4.1 Sparsity of the convex relaxation

We first introduce the following definitions to rewrite the optimization problems in (5.3) and (5.5) in vector form. Let the vectors $\boldsymbol{\alpha}\in\mathbb{R}^K$ and $\boldsymbol{\beta}\in\mathbb{R}^{K(K-1)/2}$ collect values of α_k and $\beta_{k,k'}$, respectively. Let $\mathcal{L}':=\mathcal{L}^{(1)}\cup\cdots\cup\mathcal{L}^{(K)}$, where $\mathcal{L}^{(k)}:=\{i=j+(k-1)O^2:j\in\mathcal{L}\}$ for $\mathcal{L}$ containing indices for a O^2-length vector (corresponding to the vector form of an $O\times O$ matrix) as described in Section 5.2. We define the directed difference matrix $\mathbf{E}:=[\mathbf{1}_K^{\top}\otimes-\mathbf{I}_K]_{\cdot,\mathcal{L}}+[\mathbf{I}_K\otimes\mathbf{1}_K^{\top}]_{\cdot,\mathcal{L}}$, where $\mathcal{L}$ contains indices for a K^2-length vector. We can then introduce the matrix $\boldsymbol{\Psi}:=2[\boldsymbol{\Psi}_0]_{\cdot,\mathcal{L}'}$ associated with the objectives of (5.3) and (5.5), where

$$\boldsymbol{\Psi}_0:=\begin{bmatrix}\mathrm{diag}(\boldsymbol{\alpha})\otimes\mathbf{I}_{O^2}\\ \mathrm{diag}(\boldsymbol{\beta})\mathbf{E}^{\top}\otimes\mathbf{I}_{O^2}\end{bmatrix}.$$

For the first constraint of (5.3) and (5.5), we introduce $\boldsymbol{\Sigma}:=\mathrm{blockdiag}(\boldsymbol{\Sigma}^{(1)},\ldots,\boldsymbol{\Sigma}^{(K)})$, where $\boldsymbol{\Sigma}^{(k)}:=[\boldsymbol{\Sigma}_0^{(k)}]_{\cdot,\mathcal{L}}+[\boldsymbol{\Sigma}_0^{(k)}]_{\cdot,\mathcal{U}}$ and $\boldsymbol{\Sigma}_0^{(k)}=(-\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\oplus\hat{\mathbf{C}}_{\mathcal{O}}^{(k)})$ for all $k=1,\ldots,K$, and $\mathcal{L}$ and $\mathcal{U}$ for $\boldsymbol{\Sigma}^{(k)}$ return entries of a vector of length O^2. Furthermore, let $\mathbf{Q}$ be a commutation matrix such that for any square matrix $\mathbf{Y}$, we have that $\mathrm{vec}(\mathbf{Y}^{\top})=\mathbf{Q}\mathrm{vec}(\mathbf{Y})$, and let $\mathbf{M}=\mathrm{blockdiag}(\mathbf{I}_{O^2}-\mathbf{Q},\ldots,\mathbf{I}_{O^2}-\mathbf{Q})$ with K diagonal blocks. Let $\mathcal{E}^{(k,i)}=\{(k-1)O^2+(i-1)O+j\}_{j=1}^{O}$ be index sets for all $k=1,\ldots,K$ and $i=1,\ldots,O$. Based on this, define $\mathcal{E}^{(k,k',i)}=\mathcal{E}^{(k,i)}\cup\mathcal{E}^{(k',i)}$ for every $k,k'=1,\ldots,K$ with $k<k'$, where $\mathcal{E}^{(k,i)}$ corresponds to the indices of the i-th column in the vectorized version of the matrix $\mathbf{P}^{(k)}$ and $\mathcal{E}^{(k,k',i)}$ to the indices of the i-th columns of the vectorized versions of $\mathbf{P}^{(k)}$ and $\mathbf{P}^{(k')}$.

With the following vectorizations,

$$\mathbf{s} = [\mathrm{vec}(\mathbf{S}_{\mathcal{O}}^{(1)})_{\mathcal{L}}^{\top},\ \cdots,\ \mathrm{vec}(\mathbf{S}_{\mathcal{O}}^{(K)})_{\mathcal{L}}^{\top}]^{\top} \in \mathbb{R}^{KO(O-1)/2}, \tag{5.6}$$

$$\mathbf{p} = [\mathrm{vec}(\mathbf{P}^{(1)})^{\top},\ \cdots,\ \mathrm{vec}(\mathbf{P}^{(K)})^{\top}]^{\top} \in \mathbb{R}^{KO^2}, \tag{5.7}$$

we may rewrite the optimization problem (5.3) as

$$\begin{aligned}\{\mathbf{s}', \mathbf{p}'\} = \underset{\{\mathbf{s},\mathbf{p}\}}{\mathrm{argmin}} \quad & \|\mathbf{\Psi s}\|_0 + \sum_{k=1}^{K}\sum_{i=1}^{O}\gamma_k \|\mathbf{p}_{\mathcal{E}^{(k,i)}}\|_2 \\ & + \sum_{k<k'}\sum_{i=1}^{O}\eta_{k,k'}\|\mathbf{p}_{\mathcal{E}^{(k,k',i)}}\|_2 \\ \text{s.t.} \quad & \|\mathbf{\Sigma s} + \mathbf{M p}\|_2 \le \epsilon,\ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^{\top}\mathbf{s} = 1 \end{aligned} \tag{5.3'}$$

and (5.5) as

$$\begin{aligned}\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\} = \underset{\{\mathbf{s},\mathbf{p}\}}{\mathrm{argmin}} \quad & \|\mathbf{\Psi s}\|_1 + \sum_{k=1}^{K}\sum_{i=1}^{O}\gamma_k \|\mathbf{p}_{\mathcal{E}^{(k,i)}}\|_2 \\ & + \sum_{k<k'}\sum_{i=1}^{O}\eta_{k,k'}\|\mathbf{p}_{\mathcal{E}^{(k,k',i)}}\|_2 \\ \text{s.t.} \quad & \|\mathbf{\Sigma s} + \mathbf{M p}\|_2 \le \epsilon,\ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^{\top}\mathbf{s} = 1. \end{aligned} \tag{5.5'}$$

We further denote $\mathcal{J}$ as $\mathrm{supp}(\mathbf{\Psi s}')$ and $\mathcal{I}$ as $\mathrm{supp}(\mathbf{s}')$, where $\mathrm{supp}(\mathbf{y})$ denotes the support of the vector $\mathbf{y}$. With the above definitions in place, we have the following result.

Theorem 1. *Assume that problem* (5.5') *is feasible. The solution* $\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\}$ *of* (5.5') *is equivalent to the solution* $\{\mathbf{s}', \mathbf{p}'\}$ *of* (5.3') *if the following two conditions are satisfied:*

1) $\mathbf{\Sigma}_{\cdot,\mathcal{I}}$ *is full column rank; and*

2) There exist constants $\psi, C_s > 0$ *such that*

$$\|\mathbf{\Psi}_{\mathcal{J}^c,\cdot}(\mathbf{T}_1 - \mathbf{T}_2)\mathbf{\Psi}_{\mathcal{J},\cdot}^{\top}\|_{\infty} < 1,$$

where

$$\begin{aligned}\mathbf{T}_1 &:= (\psi^{-2}(\mathbf{\Sigma}^{\top}\mathbf{\Sigma} + 2\epsilon^2 C_s^{-2}\mathbf{I}_{KO(O-1)/2}) \\ &\qquad + \mathbf{\Psi}_{\mathcal{J}^c,\cdot}^{\top}\mathbf{\Psi}_{\mathcal{J}^c,\cdot})^{-1}, \\ \mathbf{T}_2 &:= \frac{\mathbf{T}_1(\mathbf{e}_1 \otimes \mathbf{1}_{O-1})(\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^{\top}\mathbf{T}_1}{(\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^{\top}\mathbf{T}_1(\mathbf{e}_1 \otimes \mathbf{1}_{O-1})}. \end{aligned}$$

The proof of Theorem 1 can be found in Section 5.7, but we also provide a summary here. To decouple the joint optimization of $\mathbf{s}$ and $\mathbf{p}$, we consider an alternating minimization algorithm, permitting separate analysis of $\mathbf{s}$-subproblems and $\mathbf{p}$-subproblems at each iteration. Proximal alternating minimization [114], an iterative optimization algorithm, applied to (5.3') and (5.5') can be shown to converge to the original solutions $\{\mathbf{s}', \mathbf{p}'\}$ and $\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\}$, respectively. We then can show that the $\mathbf{p}$-subproblems for (5.3') and (5.5') are equivalent for every iteration, and therefore $\mathbf{p}' = \hat{\mathbf{p}}$. When the iterations grow sufficiently large for convergence, the $\mathbf{s}$-subproblems for (5.3') and (5.5') are equivalent under the conditions of Theorem 1, so $\mathbf{s}' = \hat{\mathbf{s}}$.

Under the sufficient conditions of Theorem 1, the convex relaxation in (5.5) enjoys recovery of the sparsest solution of (5.3) even in the presence of hidden nodes. Note that this result differs significantly from that of Theorem 1 in [66] due to the presence of another variable $\mathbf{p}$ that is not associated with an entrywise sparsity penalty. Condition *1)* of Theorem 1 guarantees that the solution to (5.5) is unique, and condition *2)* permits the existence of a dual certificate that ensures that the solutions to (5.5) and (5.3) are equivalent [66, 115]. Thus, under the conditions of Theorem 1, the ℓ_1 norm does not introduce any estimation error for obtaining the sparsest GSO submatrix estimates, and we need only consider the distortion from the sample covariance submatrices $\{\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\}_{k=1}^K$ and auxiliary matrices $\{\hat{\mathbf{P}}^{(k)}\}_{k=1}^K$ obtained from (5.5).

5.4.2 Robust recovery under hidden nodes

By Theorem 1, we can guarantee under mild conditions when the solution to (5.5) is equivalent to the sparsest solution from (5.3). Therefore, to evaluate the efficacy of our method in estimating the true GSO submatrices $\{\mathbf{S}_{\mathcal{O}}^{*(k)}\}_{k=1}^K$, we need only consider the estimation error of (5.5). In the sequel, we derive an upper bound on the distortion between the true GSO submatrices $\{\mathbf{S}_{\mathcal{O}}^{*(k)}\}_{k=1}^K$ and the estimated ones $\{\hat{\mathbf{S}}_{\mathcal{O}}^{(k)}\}_{k=1}^K$ obtained from (5.5). Let $\mathbf{s}^*$ be the vectorization of the true GSO submatrices $\{\mathbf{S}_{\mathcal{O}}^{*(k)}\}_{k=1}^K$ as in (5.6). We define $\mathcal{K}$ as $\operatorname{supp}(\mathbf{\Psi}\mathbf{s}^*)$, and we let $R := \sum_{k=1}^K R_k$ and $\omega := \max_{k=1,\dots,K} \omega_k$, where $\omega_k := \max\{\max_i [\mathbf{C}_{\mathcal{O}}^{(k)}]_{ii}, \max_i [\mathbf{S}_{\mathcal{O}}^{*(k)} \mathbf{C}_{\mathcal{O}}^{(k)} \mathbf{S}_{\mathcal{O}}^{*(k)}]_{ii}\}$. We present our main result on the performance of our proposed method.

Theorem 2. *Let $\{\hat{\mathbf{S}}_{\mathcal{O}}^{(k)}\}_{k=1}^K$ be the estimated subnetworks obtained from* (5.5) *with $\epsilon = \epsilon_R + \alpha$ for*

$$\alpha^2 = \sum_{k=1}^K \left\| (\hat{\mathbf{P}}^{(k)} - (\hat{\mathbf{P}}^{(k)})^\top) - (\mathbf{P}^{*(k)} - (\mathbf{P}^{*(k)})^\top) \right\|_F^2$$

and $\epsilon_R \geq C_1 O \omega \sqrt{(K \log O)/R}$ for some constant $C_1 > 0$. Under the following four conditions,

1) $K = o(\log O)$;

2) $R_1 \asymp R_2 \asymp \cdots \asymp R_K$;

3) $\log O = o(\min\{R/(K^7 (\log R)^2), (R/K^7)^{1/3}\})$; *and*

4) $\mathbf{\Sigma}$ *is full column rank;*

with probability at least $1 - e^{-C_2 \log O}$ for some constant C_2 we have that

$$\sum_{k=1}^K \|\hat{\mathbf{S}}_{\mathcal{O}}^{(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}\|_1 \leq \tau(\epsilon_R + \alpha),$$
$$\text{where } \tau = \frac{4\sqrt{|\mathcal{K}|}\sigma_{\max}(\mathbf{\Psi})\|\mathbf{\Psi}^\dagger\|_1}{\sigma_{\min}(\mathbf{\Sigma})}(2 + \sqrt{|\mathcal{K}|}). \tag{5.8}$$

The proof of Theorem 2 can be found in Section 5.8. In brief, we first apply the commutative relationship described in Section 2.4 to show that $\{\mathbf{s}^*, \hat{\mathbf{p}}\}$ is a feasible solution to (5.5'). We can then bound the ℓ_1-norm difference between the vectorization of the true GSOs $\mathbf{s}^*$ and the estimated one $\hat{\mathbf{s}}$ based on the commutativity constraint, $\epsilon = \epsilon_R + \alpha$.

Theorem 2 presents an upper bound on the estimation error of (5.5). If K and O are fixed, then as the number of observed graph signals R increases, the sample covariance submatrices $\{\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\}_{k=1}^{K}$ approach the true covariance submatrices, and the first term $\tau\epsilon_R$ in the upper bound in (5.8) becomes negligible. With enough observed graph signals, the error primarily depends on the second term $\tau\alpha$, which denotes the approximation error of $\{\hat{\mathbf{P}}^{(k)}\}_{k=1}^{K}$, the crux of our proposed method. If (5.5) is effective at enforcing $\mathbf{P}^{(k)}$ to share structural characteristics of $\mathbf{C}_{\mathcal{OH}}^{(k)}\mathbf{S}_{\mathcal{HO}}^{*(k)}$ such that they are close, then the estimation of the GSO submatrices $\mathbf{S}_{\mathcal{O}}^{*(k)}$ becomes easier according to (5.8). Furthermore, as $\mathbf{P}^{(k)}$ becomes a more accurate approximation of $\mathbf{P}^{*(k)}$, the estimation accuracy of $\hat{\mathbf{S}}_{\mathcal{O}}^{(k)}$ improves increasingly when compared to estimating $\mathbf{S}_{\mathcal{O}}^{*(k)}$ while ignoring the presence of hidden nodes. We formalize this statement in the following result that characterizes the effectiveness of our proposed formulation with respect to the auxiliary matrices $\{\mathbf{P}^{(k)}\}_{k=1}^{K}$.

Corollary 1. *Let the naive subnetwork estimates considering only observed nodes be denoted as* $\{\tilde{\mathbf{S}}_{\mathcal{O}}^{(k)}\}_{k=1}^{K}$ *[66], which we define as the solution to* (5.5) *while fixing* $\mathbf{P}^{(k)} = \mathbf{0}_{O\times O}$ *for every* $k = 1, 2, \ldots, K$*, and we let* $\epsilon = \epsilon_R$*, where* $\epsilon_R \geq C_1 O\omega\sqrt{(K\log O)/R}$ *for some constant* $C_1 > 0$*, and* $\gamma_k = 0$*,* $\eta_{k,k'} = 0$ *for every* $k, k' = 1, 2, \ldots, K$ *and* $k < k'$*. Additionally, let* $\tilde{\mathbf{s}}$ *be the vectorization as in* (5.6) *of* $\{\tilde{\mathbf{S}}_{\mathcal{O}}^{(k)}\}_{k=1}^{K}$ *and define* δ *as*

$$\delta^2 = \sum_{k=1}^{K} \|\mathbf{P}^{*(k)} - (\mathbf{P}^{*(k)})^\top\|_F^2.$$

Then, we have that

$$\begin{aligned}
&\sum_{k=1}^{K} \|\tilde{\mathbf{S}}_{\mathcal{O}}^{(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}\|_1 \leq (\tau + \tau')(\epsilon_R + \tfrac{1}{2}\delta),\\
&where\ \tau = \frac{4\sqrt{|\mathcal{K}|}\sigma_{\max}(\mathbf{\Psi})\|\mathbf{\Psi}^\dagger\|_1}{\sigma_{\min}(\mathbf{\Sigma})}(2 + \sqrt{|\mathcal{K}|})\\
&and\ \tau' = \frac{2\rho KO(O-1)(1+\sqrt{|\mathcal{K}|})\sigma_{\max}(\mathbf{\Psi})\|\mathbf{\Psi}^\dagger\|_1}{\sigma_{\min}(\mathbf{\Sigma})}
\end{aligned} \tag{5.9}$$

for some $\rho \in [0, 1]$*. Furthermore, we have that if*

$$\begin{aligned}
&\sum_{k=1}^{K} \left\| (\hat{\mathbf{P}}^{(k)} - (\hat{\mathbf{P}}^{(k)})^\top) - (\mathbf{P}^{*(k)} - (\mathbf{P}^{*(k)})^\top) \right\|_F^2\\
&\qquad \leq \left(\frac{\tau'}{\tau}\right)^2 \epsilon_R^2 + \left(\frac{\tau+\tau'}{2\tau}\right)^2 \sum_{k=1}^{K} \left\| \mathbf{P}^{*(k)} - (\mathbf{P}^{*(k)})^\top \right\|_F^2,
\end{aligned} \tag{5.10}$$

then the error bound in (5.8) *is lower than the error bound in* (5.9)*.*

The proof of Corollary 1 can be found in Section 5.9, which follows a similar procedure to the proof of Theorem 2. Corollary 1 demonstrates the criticality of accounting for hidden nodes. We describe these implications more intuitively here. First, as discussed following Theorem 2, we note that as $\hat{\mathbf{P}}^{(k)}$ approximates $\mathbf{P}^{*(k)}$ more accurately, we achieve greater improvement over $\{\tilde{\mathbf{S}}_{\mathcal{O}}^{(k)}\}_{k=1}^{K}$ from our proposed inference problem (5.5). Indeed, as the matrix difference $(\hat{\mathbf{P}}^{(k)})^\top - \hat{\mathbf{P}}^{(k)}$ approaches the right-hand side of (5.2), we remove the influence of the hidden nodes on the estimation of the observed submatrices. Second, note that the second term in the upper bound of (5.10) is proportional to δ, which measures the influence of the hidden nodes on the observed nodes in the stationary graph signal regime. When δ is negligible, the hidden nodes have little

effect on the observed nodes, and the inclusion of $\{\mathbf{P}^{(k)}\}_{k=1}^K$ in the inference process may affect performance detrimentally. However, as δ increases, the need to account for the right-hand side of (5.2) becomes crucial. We verify this comparison of (5.5) and the naive solution $\{\tilde{\mathbf{S}}_{\mathcal{O}}^{(k)}\}_{k=1}^K$ with synthetic simulations in Section 5.5.

5.5 Numerical evaluation

We introduce several experiments to assess the performance of the proposed network topology inference method. The experiments employ synthetic and real-world data and compare the quality of the graphs estimated by different algorithms. For the k-th graph, we compute the normalized error between the target $\mathbf{S}_{\mathcal{O}}^{*(k)}$ and the estimated $\hat{\mathbf{S}}_{\mathcal{O}}^{(k)}$ as

$$\operatorname{nerr}(\mathbf{S}_{\mathcal{O}}^{*(k)}, \hat{\mathbf{S}}_{\mathcal{O}}^{(k)}) = \frac{\|\mathbf{S}_{\mathcal{O}}^{*(k)} - \hat{\mathbf{S}}_{\mathcal{O}}^{(k)}\|_F^2}{\|\mathbf{S}_{\mathcal{O}}^{*(k)}\|_F^2}, \tag{5.11}$$

and then report the average across the K graphs being estimated, i.e., $\frac{1}{K}\sum_{k=1}^K \operatorname{nerr}(\mathbf{S}_{\mathcal{O}}^{*(k)}, \hat{\mathbf{S}}_{\mathcal{O}}^{(k)})$. The code for the proposed method and the experiments is available on GitHub[1].

5.5.1 Synthetic experiments

We rely on synthetic graphs and signals to assess how different elements impact the performance of the proposed approach. Unless specified otherwise, in the following experiments we consider $K = 3$ graphs with $N = 20$ nodes from which $O = 19$ are observed. The graph $\mathcal{G}^{(1)}$ is sampled from an ER random graph model with a link probability of $p = 0.2$, and the related graphs are created by randomly rewiring a fixed number of edges. We ensure that sampled graphs are connected to preclude any isolated nodes. Stationary graph signals are generated by diffusing a white input signal across the graph, that is, $\mathbf{x} = \mathbf{H}\mathbf{w}$, where the coefficients of $\mathbf{H}$ are drawn from a uniform distribution and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Under this model, the covariance of $\mathbf{x}$ is a polynomial of $\mathbf{S}$, which constitutes a more general setting than, for example, graph signals sampled from a GMRF.

Varying the effect of hidden nodes. We start by illustrating the result in (5.10) that expresses when it is beneficial to incorporate $\mathbf{P}^{(k)}$ for hidden nodes. To this end, we estimate $K = 3$ networks from perfectly known covariance submatrices $\mathbf{C}_{\mathcal{O}}^{(k)}$ so $\epsilon_R = 0$ [cf. (5.10)], to assess only the effects of $\mathbf{P}^{(k)}$ and the hidden nodes $\mathcal{H}$, characterized respectively by α from Theorem 2 and δ from Corollary 1. We compare two network inference methods: (i) JH-GSR, which denotes the method in (5.5) that accounts for hidden nodes, and (ii) J-GSR, which denotes the method described in Corollary 1 that ignores hidden variables [66]. Fig. 5.1 shows the network estimation error as the edge weights connecting observed nodes and hidden nodes increase, that is, as nonzero entries in $\mathbf{S}_{\mathcal{O}\mathcal{H}}^{*(k)}$ grow larger. While the GSO sparsity patterns do not change, the hidden node influence δ increases with the edge weights in $\mathbf{S}_{\mathcal{O}\mathcal{H}}^{*(k)}$. To measure performance that is consistent with Corollary 1, we report the average error across all K graphs as the normalized ℓ_1-norm difference, equivalent to computing (5.11) with the ℓ_1 norm replacing the squared Frobenius norm. We let $\epsilon = 10^{-8}$ for the first constraint in (5.5); however, the solution to the naive problem with $\mathbf{P}^{(k)} = \mathbf{0}_{O\times O}$ may not be feasible. Indeed, when ϵ is small enough, it may be impossible to obtain

[1] `https://github.com/reysam93/hidden_joint_inference`

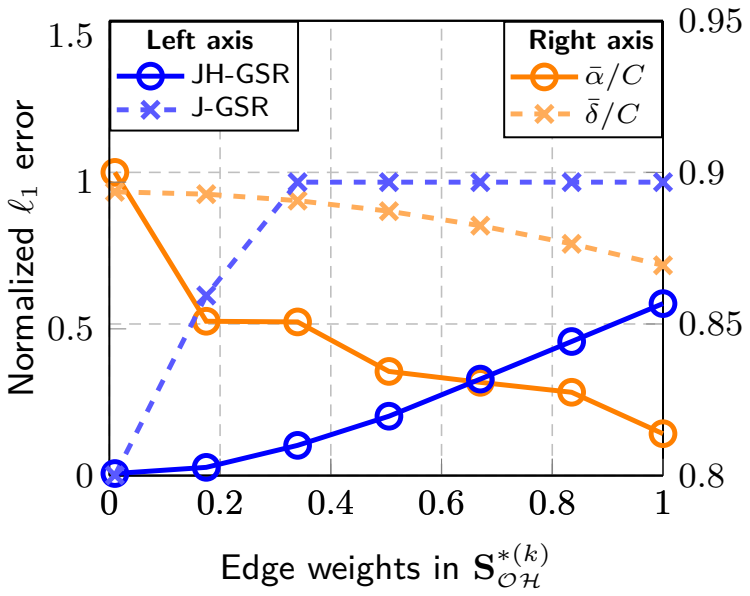

Figure 5.1: Evaluation of the performance of graph inference accounting for hidden nodes via (5.5) and graph inference ignoring hidden nodes as described in Corollary 1 as the weights of edges between observed and hidden nodes increase. The experiment considers different network topology inference alternatives and the reported results are the average error of 100 independent realizations.

a feasible solution $\{\tilde{\mathbf{S}}_{\mathcal{O}}^{(k)}\}_{k=1}^{K}$ such that all constraints hold. In such a case where the solution is infeasible, we let its error be 1. Along with network estimation error, we compare in Fig. 5.1 normalized values of α and δ to evaluate when the result in (5.10) holds. In particular, we let $\bar{\alpha} := \sum_k \mathrm{nerr}(\mathbf{P}^{*(k)}, (\mathbf{P}^{*(k)})^\top + \hat{\mathbf{P}}^{(k)} - (\hat{\mathbf{P}}^{(k)})^\top)/K$ and $\bar{\delta} := \sum_k \mathrm{nerr}(\mathbf{P}^{*(k)}, (\mathbf{P}^{*(k)})^\top)/K$. Since we need only consider which value is greater, we plot $\bar{\alpha}/C$ and $\bar{\delta}/C$ for some constant $C > 0$ such that the values are between 0 and 1.

When the edge weight is 0, the hidden nodes are decoupled from the network and thus have no effect on the observed nodes, and indeed J-GSR perfectly recovers the target networks. For zero-valued edge weights in $\mathbf{S}_{\mathcal{OH}}^{*(k)}$, we observe $\alpha \geq \delta$, where JH-GSR is comparable but not superior to J-GSR. As the edge weight increases and becomes nonnegligible, the effect of the hidden nodes increases, and we observe in Fig. 5.1 that $\alpha < \delta$ for all nonzero edge weights and JH-GSR consistently outperforms J-GSR as expected from (5.10). We thus validate the necessity of our proposed method, where as the influence of hidden nodes increases, we must account for their presence to maintain a satisfactory estimation error.

Varying the number of graphs. We next assess the benefits of considering a joint network topology inference approach when several graphs need to be learned. To that end, Fig. 5.2 illustrates the normalized error computed according to (5.11) as the number of graphs K being estimated increases. The performance of JH-GSR is compared with (i) S-GSR, the network topology inference method from stationary observations [4] where graphs are learned individually and the presence of hidden variables is ignored; SH-GSR, a generalization of (i) that takes into account the influence of hidden variables [5]; and (iii) J-GSR as in Fig. 5.1. Looking at the results, we observe that JH-GSR outperforms the alternatives, showcasing the benefits of harnessing the graph similarity while accounting for the influence of the hidden nodes. We also observed that the joint approaches achieve a lower error when more than one graph is being estimated, and furthermore, that the benefits of the joint approaches increase with K. Lastly, Fig. 5.2 also shows that for the setup

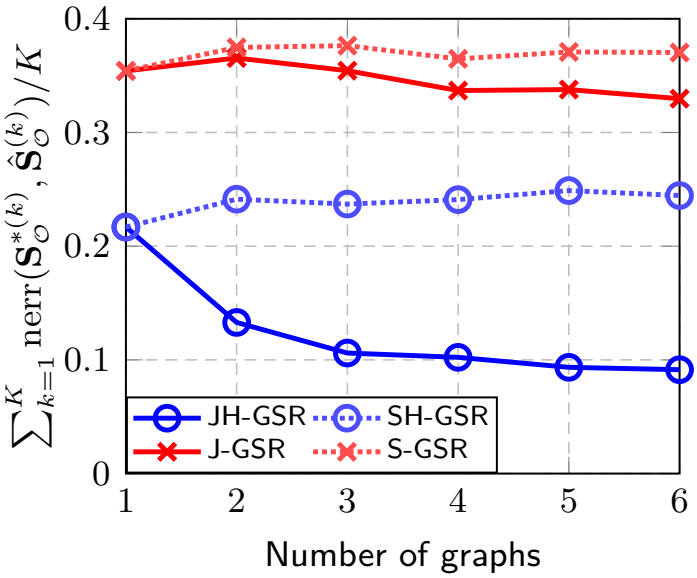

Figure 5.2: Evaluation of the influence of increasing the number of graphs being estimated. The experiment considers different graph topology inference alternatives and the reported results are the average error of 100 independent realizations.

at hand, ignoring the influence of hidden nodes results in a worse performance than ignoring the relation across networks, which is studied in more detail in the following experiment.

Varying the number of hidden nodes. The results in Fig. 5.3 investigate the detrimental influence of the presence of hidden nodes in the network topology inference task. We examine fixed-size graphs with $N = 20$ nodes and increase the number of hidden nodes H as shown in the x-axis. We evaluate the performance of (i) our proposed method, JH-GSR, (ii) an alternative implementation of our method replacing the group Lasso penalty by the nuclear norm, NN, and

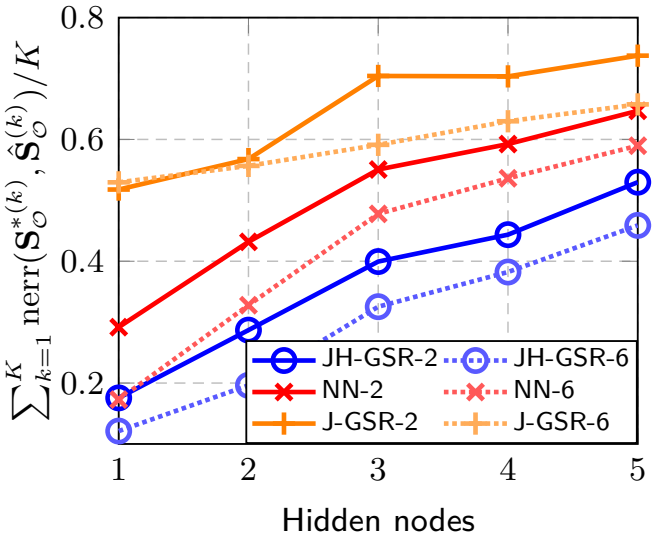

Figure 5.3: Evaluation of the detrimental effects of increasing the number of hidden nodes. The experiments consider different graph topology inference alternatives and the reported results are the average error of 100 independent realizations.

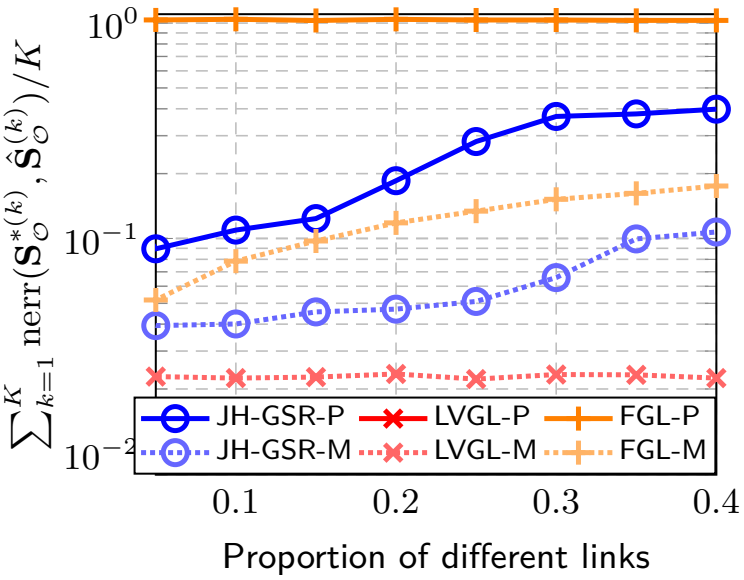

Figure 5.4: Evaluation of the impact of the graph similarity in joint network topology inference methods in different graph topology inference alternatives. The results reported are the average error of 100 independent realizations.

(iii) the joint network topology inference ignoring the presence of hidden nodes, J-GSR [66]. Then, for each baseline, we consider the estimation of either 2 or 6 graphs. First, from Fig. 5.3, it can be seen that increasing the number of hidden nodes renders the inference problem more challenging and, moreover, that ignoring the presence of hidden nodes results in poor performance. Second, the superior performance of JH-GSR over NN supports our initial intuition that the group Lasso penalty is better suited to capture the structure of the problem at hand. Furthermore, we also observe that estimating 6 graphs leads to a better performance than estimating 2, a behavior aligned with the previous experiment.

Varying graph similarity. Next, we evaluate the impact of *(AS3)*, a critical assumption in joint graph topology inference methods. More precisely, we consider estimating $K = 3$ graphs as the proportion of different edges increases, i.e., as the graphs become more dissimilar. The errors of the estimated graphs are depicted in Fig. 5.4, where we compare the performance of JH-GSR with (i) LVGL, a GL algorithm modeling the presence of hidden nodes [63]; and (ii) FGL, a joint GL algorithm [65]. Moreover, since GL algorithms assume that the observations are drawn from a GMRF, we consider two different types of signals. Signals sampled from a GMRF are denoted as "M", and signals generated as the diffusion of a white input via a polynomial of the GSO are denoted as "P". As expected from *(AS3)*, Fig. 5.4 shows that the performance of joint methods, JH-GSR and FGL, deteriorates as we consider a higher number of different links. For the two signal models, we observe that JH-GSR-M is superior to JH-GSR-P since the GMRF model is a simpler special case of graph stationarity that is less sensitive to hidden nodes. Interestingly, JH-GSR-M also outperforms GFL-M, although the latter is a method tailored for GMRF observations, showcasing the more general nature of the stationary model and the importance of accounting for the presence of hidden nodes. In contrast, we observe that graphical models are incapable of estimating graphs from stationary observations, and we note that LVGL-P is not included in the figure due to its high error.

Varying graph sparsity. In the last experiment based on synthetic data, we assess the performance of the proposed method in terms of the recovery of the support and how the weight of the

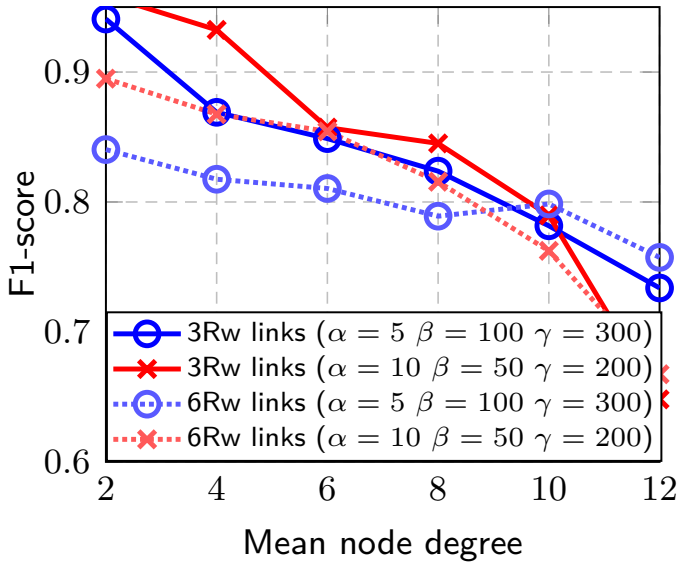

Figure 5.5: Evaluation of the impact of the graph sparsity in the support recovery for different hyperparameter selections. The experiment considers two settings for graph similarity by rewiring 3 and 6 links. The results reported are the average error of 100 independent realizations.

regularizers influences the results. To that end, Fig. 5.5 depicts the evolution of the F*score* as the mean node degree increases for different configurations of the hyperparameters. Graph $\mathcal{G}^{(1)}$ is drawn from a small world random graph model with a rewiring probability of 0.1, and similar graphs are generated by rewiring either 3 or 6 links (respectively "3Rw" or "6Rw" in the legend). The results illustrate how higher values of α obtain the best performance when the graph is sparse but deteriorates as the graph becomes denser. Similarly, a high value of β harnesses the similar support of the graphs but, when graphs are less alike, it may deteriorate the performance. Last but not least, Fig. 5.5 illustrates how the support of the graphs is almost perfectly recovered when graphs are sparse, but the performance deteriorates as the density of edges increases.

5.5.2 Application to real-world graphs

In addition to the synthetic data where we know the model relating the networks and the observed graph signals, we assess our proposed method with real-world data to demonstrate its efficacy in several scenarios, including those where the stationarity assumption is not explicitly enforced.

Students dataset. The following experiment combines real-world graphs with synthetic signals. This mixed approach allows us to investigate the applicability of the proposed method to real-world graphs while ensuring that the observed signals are stationary. We employed three graphs defined on a common set of 32 nodes, where nodes represent students from the University of Ljubljana, and the different graphs encode various types of interactions among the students[2]. The results are displayed in Fig. 5.6, where we observe the error of the recovered graphs as the number of samples increases. The error reported is the average of 50 realizations of random stationary graph signals, with only one hidden node considered. For each of the three graphs, we evaluate the performance of both the joint and the separate estimation methods, JH-GSR and SH-GSR. From the results, it is evident that the recovery of all three graphs significantly improves with a joint approach,

[2]Original data available at `http://vladowiki.fmf.uni-lj.si/doku.php?id=pajek:data:pajek:students`

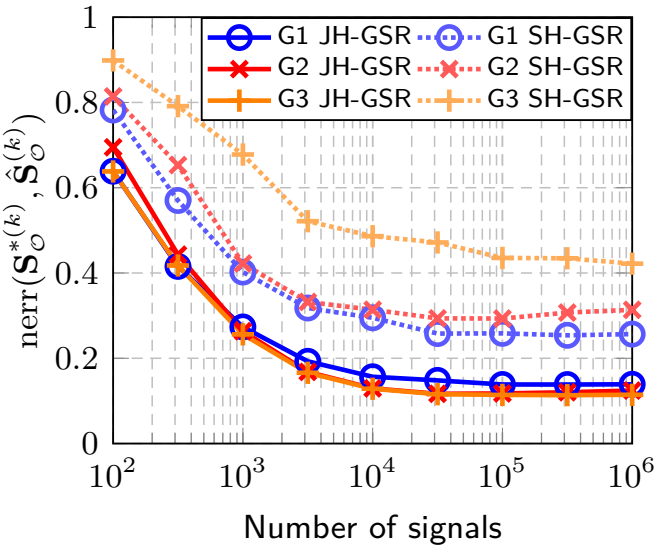

Figure 5.6: Evaluating the performance of the proposed network topology inference in real-world scenarios. Error estimating three graphs considering either a joint or a separate method. Graphs are obtained from the students of the University of Ljubljana dataset.

demonstrating the benefits of leveraging the existing relationship between the networks.

Infering multiple observed graphs from voting data. Finally, we close with an experiment aimed at learning two related political graphs from voting data[3]. More specifically, we consider 25 cantons of Switzerland as the nodes of the graph and the percentage of votes in favor of 185 initiatives submitted between 2000 and 2020 as the signals. In this setting, links reflect social influence between cantons (for example, if a canton has a great influence over others its degree will be larger), and hidden nodes correspond to cantons whose votes are never observed. Our goal then is to infer the political graph of Switzerland for two consecutive periods of time. Intuitively, although political representation may evolve with time, this process is typically slow and, hence, the two graphs are expected to be closely related. We validate the estimations via ground truth graphs whose links reflect the political preferences of the cantons, which are obtained by performing separate inference of both graphs with all available signals. We consider two setups with $H = 2$ and $H = 4$ hidden nodes, respectively illustrated in Fig. 5.7a and Fig. 5.7b. The figures present the normalized error of the estimated graphs as the percentage of available signals ranges from 70% to 90% of all available signals. We compare the proposed algorithm, JH-GSR, with three alternative methods: J-GSR, SH-GSR, and J-LVGL from [6].

First, we focus on the estimation performance of the four methods when $H = 2$ hidden nodes are considered as shown in Fig. 5.7a. Since the number of available signals for the second graph is considerably smaller than the signals available for the first graph, we observe a much larger estimation error for the second graph when the separate approach SH-GSR is employed. In contrast, for the joint estimation method J-GSR, we observe that errors are similar for both graphs and inferior on average compared to SH-GSR. This behavior illustrates that harnessing the similarity of the graphs results in an improvement in performance since it allows sharing common learned structures across graphs. Moreover, we observe that JH-GSR outperforms both SH-GSR and J-GSR since, in addition to being a joint approach, it takes into account the influence of the hidden nodes. We also compare JH-GSR with J-LVGL, both of which perform joint network inference

[3]Original data available at `https://swissvotes.ch/page/home`

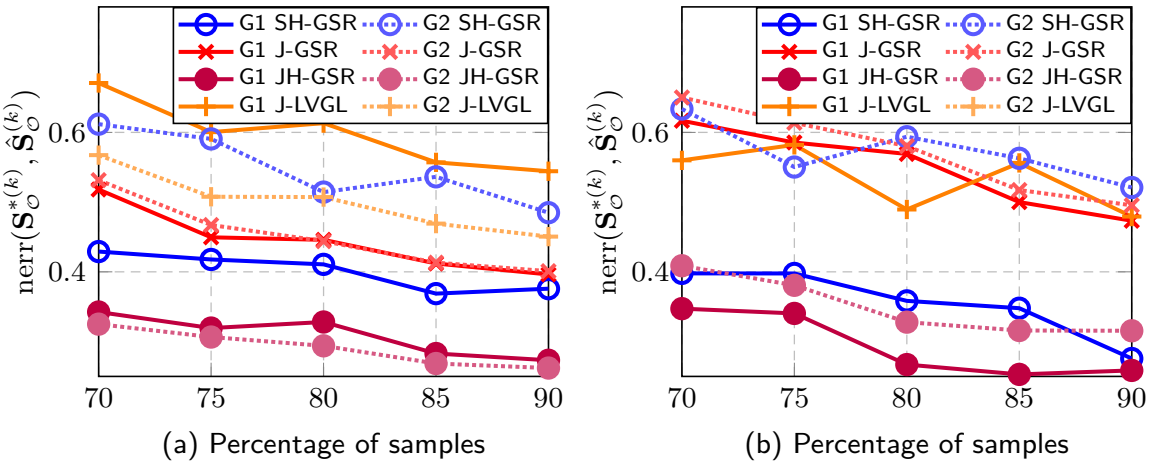

Figure 5.7: Evaluating the performance of the proposed network topology inference approach considering voting data. (a) Error estimating two graphs from voting signals considering different approaches for $H = 2$. (b) Error estimating two graphs from voting signals considering different approaches for $H = 4$.

while accounting for hidden nodes. However, we find that JH-GSR is drastically superior due to complexities in the data structure that J-LVGL cannot capture accurately. Indeed, the stationary model subsumes the GMRF model while allowing for more complex statistical relationships between the graph topology and the signals.

Moving to the results of Fig. 5.7b, we observe that increasing the number of hidden variables renders the problem more challenging, hence leading to a drop in the performance of all the algorithms. It is worth mentioning that the error corresponding to "G2 J-LVGL" was too high, so it is not included in the figure. Also note that the fraction of hidden nodes is $4/25$, which is relatively large. Nevertheless, we observe that methods accounting for the presence of hidden nodes are more resilient to this challenging setting, while the performance of the non-robust alternatives deteriorates significantly. Moreover, the proposed method JH-GSR continues to outperform the alternatives, achieving a lower error in both recovered graphs.

To summarize, it is not only crucial to account for the presence of hidden nodes but, when several related graphs are involved, it is also important to exploit the similarity between both observed and hidden nodes. This becomes particularly relevant when data is limited to a subset of the graphs, as demonstrated in the improved estimation of the second graph when considering joint network inference methods.

5.6 Conclusions

In this chapter, we presented a method to infer multiple networks on the same node set in the presence of hidden nodes. To characterize the effect of the hidden nodes, we assumed that graph signals were stationary on their respective networks. By the inherent block structure of the covariance matrix $\mathbf{C}^{(k)}$ and the GSO $\mathbf{S}^{*(k)}$ of the k-th network, we introduced a set of auxiliary matrices $\mathbf{P}^{(k)}$ to account for the effect of hidden nodes in the relationship $\mathbf{C}^{(k)}\mathbf{S}^{*(k)} = \mathbf{S}^{*(k)}\mathbf{C}^{(k)}$ stemming from the stationarity assumption. By prior assumptions on structure and stationarity, we derive characteristics of $\mathbf{P}^{(k)}$ that permit us to form an optimization problem that performs network inference while accounting for the presence of hidden nodes. Moreover, we verified that the estimation of the sparsest networks is equivalent to a computationally feasible convex relaxation

under mild conditions. We further demonstrated a bound on the error of our proposed method dependent on the error due to the sample covariance matrices and $\mathbf{P}^{(k)}$. The performance of our method was evaluated in multiple synthetic and real-world datasets in comparison with other baseline methods, and we also verified the improvement in estimation due to the incorporation of $\mathbf{P}^{(k)}$.

5.7 Appendix: Proof of Theorem 1

We first combine the last two terms in the objective functions of (5.3') and (5.5') by defining the combined index set $\mathcal{E} := \bigcup_{i=1}^{O}\{\mathcal{E}^{(k,i)}\}_{k=1}^{K} \cup \{\mathcal{E}^{(k,k',i)}\}_{k<k'}$ and parameters $\{\eta'_g\}_{g\in\mathcal{E}}$ such that $\eta'_{\mathcal{E}^{(k,i)}} = \gamma_k$ and $\eta'_{\mathcal{E}^{(k,k',i)}} = \eta_{k,k'}$ for every $k, k' = 1, \ldots, K$ such that $k < k'$ and $i = 1, \ldots, O$.

Let us consider solving (5.3') by proximal alternating minimization [114] with

$$\begin{aligned}
\mathbf{p}'^{(t)} &= \operatorname*{argmin}_{\mathbf{p}} \sum_{g\in\mathcal{E}} \eta'_g \|\mathbf{p}_g\|_2 + \frac{1}{2\lambda'_t}\|\mathbf{p} - \mathbf{p}'^{(t-1)}\|_2^2 \\
&\qquad \text{s.t. } \|\boldsymbol{\Sigma}\mathbf{s}'^{(t-1)} + \mathbf{M}\mathbf{p}\|_2 \le \epsilon, \qquad (5.12a)\\
\mathbf{s}'^{(t)} &\in \operatorname*{argmin}_{\mathbf{s}} \|\boldsymbol{\Psi}\mathbf{s}\|_0 + \frac{1}{2\mu'_t}\|\mathbf{s} - \mathbf{s}'^{(t-1)}\|_2^2 \\
&\qquad \text{s.t. } \|\boldsymbol{\Sigma}\mathbf{s} + \mathbf{M}\mathbf{p}'^{(t)}\|_2 \le \epsilon,\ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top \mathbf{s} = 1, \qquad (5.12b)
\end{aligned}$$

and (5.5') with

$$\begin{aligned}
\hat{\mathbf{p}}^{(t)} &= \operatorname*{argmin}_{\mathbf{p}} \sum_{g\in\mathcal{E}} \eta'_g \|\mathbf{p}_g\|_2 + \frac{1}{2\hat{\lambda}_t}\|\mathbf{p} - \hat{\mathbf{p}}^{(t-1)}\|_2^2 \\
&\qquad \text{s.t. } \|\boldsymbol{\Sigma}\hat{\mathbf{s}}^{(t-1)} + \mathbf{M}\mathbf{p}\|_2 \le \epsilon, \qquad (5.13a)\\
\hat{\mathbf{s}}^{(t)} &= \operatorname*{argmin}_{\mathbf{s}} \|\boldsymbol{\Psi}\mathbf{s}\|_1 + \frac{1}{2\hat{\mu}_t}\|\mathbf{s} - \hat{\mathbf{s}}^{(t-1)}\|_2^2 \\
&\qquad \text{s.t. } \|\boldsymbol{\Sigma}\mathbf{s} + \mathbf{M}\hat{\mathbf{p}}^{(t)}\|_2 \le \epsilon,\ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top \mathbf{s} = 1, \qquad (5.13b)
\end{aligned}$$

for $t \in \mathbb{N}$, where the parameters λ'_t, μ'_t, $\hat{\lambda}_t$, and $\hat{\mu}_t$ are bounded above and below by positive real numbers. By the proximal terms in (5.12) and (5.13), the subproblems (5.12a), (5.13a), and (5.13b) are strongly convex, and each iteration of these has a unique solution. Furthermore, for every $t \in \mathbb{N}$ and any given pair of constants $C_t^s, C_t^p \ge 0$, we may select positive values λ'_t, μ'_t, $\hat{\lambda}_t$, and $\hat{\mu}_t$ such that the solutions to (5.12) and (5.13) are equivalent to

$$\begin{aligned}
\mathbf{p}'^{(t)} &= \operatorname*{argmin}_{\mathbf{p}} \sum_{g\in\mathcal{E}} \eta'_g \|\mathbf{p}_g\|_2 \\
&\qquad \text{s.t. } \|\boldsymbol{\Sigma}\mathbf{s}'^{(t-1)} + \mathbf{M}\mathbf{p}\|_2 \le \epsilon,\ \|\mathbf{p} - \mathbf{p}'^{(t-1)}\|_2 \le C_t^p, \qquad (5.14a)\\
\mathbf{s}'^{(t)} &\in \operatorname*{argmin}_{\mathbf{s}} \|\boldsymbol{\Psi}\mathbf{s}\|_0 \\
&\qquad \text{s.t. } \|\boldsymbol{\Sigma}\mathbf{s} + \mathbf{M}\mathbf{p}'^{(t)}\|_2 \le \epsilon,\ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top \mathbf{s} = 1 \\
&\qquad\quad \|\mathbf{s} - \mathbf{s}'^{(t-1)}\|_2 \le C_t^s, \qquad (5.14b)
\end{aligned}$$

and

$$\hat{\mathbf{p}}^{(t)} = \underset{\mathbf{p}}{\operatorname{argmin}} \sum_{g\in\mathcal{E}} \eta'_g \|\mathbf{p}_g\|_2$$
$$\text{s.t. } \|\boldsymbol{\Sigma}\hat{\mathbf{s}}^{(t-1)} + \mathbf{M}\mathbf{p}\|_2 \le \epsilon,\ \|\mathbf{p} - \hat{\mathbf{p}}^{(t-1)}\|_2 \le C_t^p, \tag{5.15a}$$
$$\hat{\mathbf{s}}^{(t)} = \underset{\mathbf{s}}{\operatorname{argmin}} \|\boldsymbol{\Psi}\mathbf{s}\|_1$$
$$\text{s.t. } \|\boldsymbol{\Sigma}\mathbf{s} + \mathbf{M}\hat{\mathbf{p}}^{(t)}\|_2 \le \epsilon,\ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top \mathbf{s} = 1$$
$$\|\mathbf{s} - \hat{\mathbf{s}}^{(t-1)}\|_2 \le C_t^s. \tag{5.15b}$$

Let us initialize the proximal alternating minimization steps for (5.14) and (5.15) with $\mathbf{p}_0 := \mathbf{p}'^{(0)} = \hat{\mathbf{p}}^{(0)}$ and $\mathbf{s}_0 := \mathbf{s}'^{(0)} = \hat{\mathbf{s}}^{(0)}$. Note that the objective functions of (5.3') and (5.5') are semi-algebraic functions [116] and thus have the Kurdyka-Łojasiewicz property [114]. By [114, Theorem 3.3], there exist constants $r', s' > 0$ such that when we let $\|\mathbf{p}' - \mathbf{p}_0\|_2 + \|\mathbf{s}' - \mathbf{s}_0\|_2 < r'$ and

$$\begin{aligned} \|\boldsymbol{\Psi}\mathbf{s}'\|_0 + \sum_{g\in\mathcal{E}} \eta'_g \|\mathbf{p}'_g\|_2 &\le \|\boldsymbol{\Psi}\mathbf{s}_0\|_0 + \sum_{g\in\mathcal{E}} \eta'_g \|[\mathbf{p}_0]_g\|_2 \\ &< \|\boldsymbol{\Psi}\mathbf{s}'\|_0 + \sum_{g\in\mathcal{E}} \eta'_g \|\mathbf{p}'_g\|_2 + s', \end{aligned} \tag{5.16}$$

where the first inequality is due to the optimality of $\{\mathbf{s}', \mathbf{p}'\}$ for feasible $\{\mathbf{s}_0, \mathbf{p}_0\}$, then we have that the sequence $\{\mathbf{s}'^{(t)}, \mathbf{p}'^{(t)}\}$ converges to $\{\mathbf{s}', \mathbf{p}'\}$ in finitely many steps. Similarly, there exist constants $\hat{r}, \hat{s} > 0$ such that we can guarantee that the sequence $\{\hat{\mathbf{s}}^{(t)}, \hat{\mathbf{p}}^{(t)}\}$ converges to $\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\}$ in finitely many steps. More specifically, there exist positive integers T_1, T_2 such that $\{\mathbf{s}', \mathbf{p}'\} = \{\mathbf{s}'^{(t)}, \mathbf{p}'^{(t)}\}$ for every $t \ge T_1$ and $\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\} = \{\hat{\mathbf{s}}^{(t)}, \hat{\mathbf{p}}^{(t)}\}$ for every $t \ge T_2$.

Note that $\hat{r} > 0$ may take any arbitrarily large finite number [114], so we may select $\{\mathbf{s}_0, \mathbf{p}_0\}$ such that $\|\mathbf{p}' - \mathbf{p}_0\|_2 + \|\mathbf{s}' - \mathbf{s}_0\|_2 < r'$ and (5.16) are satisfied. Then, we let $\hat{r} \ge r' + \|\hat{\mathbf{s}} - \mathbf{s}'\|_2$. Such a finite $\hat{r}$ exists since problems (5.3') and (5.5') have coercive objective functions and we assume feasibility of both, that is, $\|\hat{\mathbf{s}} - \mathbf{s}'\|_2 \le \|\hat{\mathbf{s}}\|_2 + \|\mathbf{s}'\|_2 < +\infty$. Similarly, we may select a finite upper bound $C_s = C_t^s \ge \|\hat{\mathbf{s}} - \mathbf{s}'\|_2$ for every $t \ge T$ for the last constraint in subproblems (5.14b) and (5.15b).

We select feasible initial points $\{\mathbf{s}_0, \mathbf{p}_0\}$ to guarantee convergence of (5.12) and (5.13). Recall that we define the set $\mathcal{M} = \{O, O+1, \ldots, KO(O-1)/2\}$, and let $\mathbf{a}' := [\mathbf{s}'^\top_{\mathcal{M}}, \mathbf{p}'^\top]^\top$ and $\mathbf{a}_0 := [[\mathbf{s}_0]^\top_{\mathcal{M}}, \mathbf{p}_0^\top]^\top$. Consider the optimization problem

$$\min_{\mathbf{a}_0} \|\mathbf{a}_0\|_2^2 \ \text{s.t. } \|\mathbf{a}' - \mathbf{a}_0\|_2 \le r,$$

whose optimal solution is $\mathbf{a}_0 = C\mathbf{a}'$ where $C = (\|\mathbf{a}'\|_2 - r)/\|\mathbf{a}'\|_2$. Then, our optimal initial point is $[\mathbf{s}_0]_{\mathcal{M}^c} = \mathbf{s}'_{\mathcal{M}^c}$, $[\mathbf{s}_0]_{\mathcal{M}} = C\mathbf{s}'_{\mathcal{M}}$, and $\mathbf{p}_0 = C\mathbf{p}'$. By the inequality $(a+b)^2 \le 2a^2 + 2b^2$ and our assumption that $r < 2^{-1/2}(\|\mathbf{s}'_{\mathcal{M}}\|_2 + \|\mathbf{p}'\|_2) \le \|\mathbf{a}'\|_2$, we have that $C \in [0, 1)$. Moreover, the solution $\{\mathbf{s}_0, \mathbf{p}_0\}$ satisfies $\|\mathbf{s}' - \mathbf{s}_0\|_2 + \|\mathbf{p}' - \mathbf{p}_0\|_2 \le \sqrt{2}\|\mathbf{a}' - \mathbf{a}_0\|_2 \le \sqrt{2}r < r'$. By our condition on ϵ, we have that

$$\begin{aligned} \epsilon &\ge \sigma_{\max}(\boldsymbol{\Sigma})r' + 2\hat{r} \\ &\quad + \sqrt{2}(\sigma_{\max}(\boldsymbol{\Sigma}) + 2)(\|\mathbf{s}'\|_2 + \|\mathbf{p}'\|_2 - r) \\ &\ge \sigma_{\max}(\boldsymbol{\Sigma})r' + 2\hat{r} \\ &\quad + (\sigma_{\max}(\boldsymbol{\Sigma}) + 2)(\|\mathbf{s}'_{\mathcal{M}^c}\|_2 + C\sqrt{2}\|\mathbf{a}'\|_2) \\ &\ge \sigma_{\max}(\boldsymbol{\Sigma})r' + 2\hat{r} \\ &\quad + (\sigma_{\max}(\boldsymbol{\Sigma}) + 2)(\|\mathbf{s}_0\|_2 + \|\mathbf{p}_0\|_2). \end{aligned}$$

Then, since $\sigma_{\max}(\mathbf{M}) = 2$,

$$\begin{aligned}\|\boldsymbol{\Sigma}\mathbf{s}' + \mathbf{M}\hat{\mathbf{p}}\|_2 &\leq \|\boldsymbol{\Sigma}(\mathbf{s}' - \mathbf{s}_0)\|_2 + \|\mathbf{M}(\hat{\mathbf{p}} - \mathbf{p}_0)\|_2 \\ &\quad + \|\boldsymbol{\Sigma}\mathbf{s}_0 + \mathbf{M}\mathbf{p}_0\|_2 \\ &\leq \sigma_{\max}(\boldsymbol{\Sigma})r' + 2\hat{r} \\ &\quad + (\sigma_{\max}(\boldsymbol{\Sigma}) + 2)(\|\mathbf{s}_0\|_2 + \|\mathbf{p}_0\|_2) \\ &\leq \epsilon. \end{aligned} \tag{5.17}$$

By the finite convergence of (5.14) and (5.15), we have that $\mathbf{s}' = \mathbf{s}'^{(t)}$ and $\hat{\mathbf{s}} = \hat{\mathbf{s}}^{(t)}$ for every $t \geq T$. We may rewrite (5.14b) and (5.15b) at iteration $T + 1$ as

$$\begin{aligned}\mathbf{s}' = \underset{\mathbf{s}}{\operatorname{argmin}} \|\boldsymbol{\Psi}\mathbf{s}\|_0 \\ \text{s.t.} \|\boldsymbol{\Sigma}\mathbf{s} + \mathbf{M}\mathbf{p}'\|_2 \leq \epsilon,\ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top \mathbf{s} = 1, \\ \|\mathbf{s} - \mathbf{s}'\|_2 \leq C_s, \end{aligned} \tag{5.18}$$

$$\begin{aligned}\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \|\boldsymbol{\Psi}\mathbf{s}\|_1 \\ \text{s.t.} \|\boldsymbol{\Sigma}\mathbf{s} + \mathbf{M}\hat{\mathbf{p}}\|_2 \leq \epsilon,\ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top \mathbf{s} = 1, \\ \|\mathbf{s} - \hat{\mathbf{s}}\|_2 \leq C_s. \end{aligned} \tag{5.19}$$

Thus, the convergence of proximal alternating minimization allows us to consider minimization with respect to $\mathbf{s}$ for both (5.3') and (5.5').

We next consider when the solutions to (5.18) and (5.19) are equivalent. We introduce a modification to (5.19) without the last constraint

$$\begin{aligned}\bar{\mathbf{s}} \in \underset{\mathbf{s}}{\operatorname{argmin}} \|\boldsymbol{\Psi}\mathbf{s}\|_1 \\ \text{s.t.} \|\boldsymbol{\Sigma}\mathbf{s} + \mathbf{M}\hat{\mathbf{p}}\|_2 \leq \epsilon,\ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top \mathbf{s} = 1, \end{aligned} \tag{5.20}$$

which may not have a unique solution. By (5.17), $\mathbf{s}'$ is a feasible solution to (5.20).

By the proof of Theorem 1 in [66] and Theorem 1 of [115], if $\boldsymbol{\Sigma}_{\cdot,\mathcal{I}}$ is full column rank and there exists a constant $\psi > 0$ such that

$$\|\boldsymbol{\Psi}_{\mathcal{J}^c,\cdot}(\psi^{-2}\mathbf{T} + \boldsymbol{\Psi}_{\mathcal{J}^c,\cdot}^\top \boldsymbol{\Psi}_{\mathcal{J}^c,\cdot})^{-1}\boldsymbol{\Psi}_{\mathcal{J},\cdot}^\top\|_\infty < 1, \tag{5.21}$$

then we not only have that $\mathbf{s}' = \bar{\mathbf{s}}$, but $\mathbf{s}'$ is also the unique solution to (5.20). These are exactly conditions *1)* and *2)* in the statement of Theorem 1. Thus, we need only show that $\bar{\mathbf{s}} = \hat{\mathbf{s}}$.

Since (5.19) and (5.20) share the first two constraints and $\|\hat{\mathbf{s}} - \mathbf{s}'\|_2 = \|\hat{\mathbf{s}} - \bar{\mathbf{s}}\|_2 \leq C_s$, $\hat{\mathbf{s}}$ and $\bar{\mathbf{s}}$ are both feasible solutions for (5.19) and (5.20). Moreover, both problems have unique solutions, so $\hat{\mathbf{s}} = \bar{\mathbf{s}} = \mathbf{s}'$, as desired.

5.8 Appendix: Proof of Theorem 2

To establish an upper bound on the estimation error of (5.5), we first provide the following lemma necessary to determine an upper bound on the error of (5.5).

Lemma 1. *Under the following four conditions,*

1) $K = o(\log O)$*;*

2) $R_1 \asymp R_2 \asymp \cdots \asymp R_K$*;*

3) $\log O = o(\min\{R/(K^7(\log R)^2), (R/K^7)^{1/3}\})$*; and*

4) $\epsilon_R \geq CO\omega\sqrt{(K \log O)/R}$ *for some constant* $C > 0$*;*

with probability at least $1 - e^{-C_1 \log O}$ *for some constant* C_1 *we have that*

$$\sum_{k=1}^{K} \left\| (\hat{\mathbf{C}}_{\mathcal{O}}^{(k)} - \mathbf{C}_{\mathcal{O}}^{(k)})\mathbf{S}_{\mathcal{O}}^{*(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}(\hat{\mathbf{C}}_{\mathcal{O}}^{(k)} - \mathbf{C}_{\mathcal{O}}^{(k)}) \right\|_F^2 \leq \epsilon_R^2.$$

Proof. The proof of Lemma 1 follows from the proof of Claim 2 in [66]. □

Recall that $\mathbf{s}^*$ is the vectorization of the target GSO submatrices $\{\mathbf{S}_{\mathcal{O}}^{*(k)}\}_{k=1}^K$ as in (5.6). We show that $\{\mathbf{s}^*, \hat{\mathbf{p}}\}$ is a feasible solution to (5.5'). We demonstrate an upper bound on the commutativity of sample covariance submatrices and target subnetworks as

$$\begin{aligned}
&\left| \sum_{k=1}^{K} \|\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\mathbf{S}_{\mathcal{O}}^{*(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}\hat{\mathbf{C}}_{\mathcal{O}}^{(k)} + \hat{\mathbf{P}}^{(k)} - (\hat{\mathbf{P}}^{(k)})^\top\|_F^2 \right|^{\frac{1}{2}} \\
&\leq \left| \sum_{k=1}^{K} \left\| (\hat{\mathbf{C}}_{\mathcal{O}}^{(k)} - \mathbf{C}_{\mathcal{O}}^{(k)})\mathbf{S}_{\mathcal{O}}^{*(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}(\hat{\mathbf{C}}_{\mathcal{O}}^{(k)} - \mathbf{C}_{\mathcal{O}}^{(k)}) \right\|_F^2 \right|^{\frac{1}{2}} \\
&\quad + \left| \sum_{k=1}^{K} \left\| (\hat{\mathbf{P}}^{(k)} - (\hat{\mathbf{P}}^{(k)})^\top) - (\mathbf{P}^{*(k)} - (\mathbf{P}^{*(k)})^\top) \right\|_F^2 \right|^{\frac{1}{2}} \\
&\leq \epsilon_R + \alpha,
\end{aligned} \tag{5.22}$$

where we have used Lemma 1, the definition of α, and the relationship in (5.2). Because $\sum_{j=1}^{O}[\mathbf{S}_{\mathcal{O}}^{*(k)}]_{j1} = 1$ by definition, (5.22) is equivalent to

$$\|\boldsymbol{\Sigma}\mathbf{s}^* + \mathbf{M}\hat{\mathbf{p}}\|_2 \leq \epsilon_R + \alpha = \epsilon, \tag{5.23}$$

so $\{\mathbf{s}^*, \hat{\mathbf{p}}\}$ is a feasible solution to (5.5').

We introduce a modification of (5.5') to combine the constraints into one inequality. Consider the following modified optimization problem that is parameterized by $r > 0$

$$\begin{aligned}
\{\hat{\mathbf{s}}_r, \hat{\mathbf{p}}_r\} = \underset{\{\mathbf{s},\mathbf{p}\}}{\operatorname{argmin}} \quad & \|\boldsymbol{\Psi}\mathbf{s}\|_1 + \sum_{k=1}^{K}\sum_{i=1}^{O} \gamma_k \|\mathbf{p}_{\mathcal{E}^{(k,i)}}\|_2 \\
& + \sum_{k<k'}\sum_{i=1}^{O} \eta_{k,k'} \|\mathbf{p}_{\mathcal{E}^{(k,k',i)}}\|_2 \\
\text{s.t.} \quad & \|\bar{\boldsymbol{\Phi}}_r\mathbf{s} + \bar{\mathbf{R}}\mathbf{p} - \bar{\mathbf{b}}_r\|_2 \leq \epsilon,
\end{aligned} \tag{5.24}$$

where $\bar{\boldsymbol{\Phi}}_r = [\boldsymbol{\Sigma}^\top, r(\mathbf{e}_1 \otimes \mathbf{1}_{O-1})]^\top$, $\bar{\mathbf{R}} = [\mathbf{M}^\top, \mathbf{0}_{KO^2}]^\top$, and $\bar{\mathbf{b}}_r = [\mathbf{0}_{KO(O-1)/2}^\top, r]^\top$. The parameter r determines the strictness of the second constraint in (5.5') such that when $r \to \infty$, we have that $\hat{\mathbf{s}}_r \to \hat{\mathbf{s}}$. Note that since $(\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top\hat{\mathbf{s}} = 1$ and $(\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top\mathbf{s}^* = 1$, then by (5.23) and the definition of $\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\}$, we have that $\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\}$ and $\{\mathbf{s}^*, \hat{\mathbf{p}}\}$ are feasible solutions of (5.24) for every $r > 0$.

We next provide an upper bound on the difference between $\hat{\mathbf{s}}$ and $\mathbf{s}^*$ following the proof of Claim 1 in [66]. Recall that we define $\mathcal{K}$ as $\mathrm{supp}(\mathbf{\Psi}\mathbf{s}^*)$. First, note that as in the proof of Claim 1 of [66], we have that when $\mathbf{\Sigma}$ is full column rank, then so is $\bar{\mathbf{\Phi}}_r$, which guarantees the existence of a dual certificate $\mathbf{y} = \mathbf{I}_{\mathcal{K},\cdot}^\top \mathrm{sign}(\mathbf{\Psi}_{\mathcal{K},\cdot}\mathbf{s}^*)$, where $\mathbf{\Psi}^\top\mathbf{y} = \bar{\mathbf{\Phi}}_r^\top\bar{\mathbf{\Phi}}_r(\bar{\mathbf{\Phi}}_r^\top\bar{\mathbf{\Phi}}_r)^{-1}\mathbf{\Psi}^\top\mathbf{I}_{\mathcal{K},\cdot}^\top\mathrm{sign}(\mathbf{\Psi}_{\mathcal{K},\cdot}\mathbf{s}^*) \in \mathrm{Im}(\bar{\mathbf{\Phi}}_r^\top)$, $\mathbf{y}_{\mathcal{K}} = \mathrm{sign}(\mathbf{\Psi}_{\mathcal{K},\cdot}\mathbf{s}^*)$, $\|\mathbf{y}_{\mathcal{K}^c}\|_\infty < 1$, and $\|\mathbf{\Psi}\mathbf{s}^*\|_1 = \mathbf{y}^\top\mathbf{\Psi}\mathbf{s}^*$.

Consider the following inequality

$$\|\mathbf{\Psi}\mathbf{s}^* - \mathbf{\Psi}\hat{\mathbf{s}}\|_1 \le \|\mathbf{\Psi}\hat{\mathbf{s}} - \mathbf{u}\|_1 + \|\mathbf{\Psi}\mathbf{s}^* - \mathbf{u}\|_1, \tag{5.25}$$

where $\mathbf{u} \in \mathbb{R}^{KO(O-1)/2}$ such that $\mathrm{supp}(\mathbf{u}) \subseteq \mathcal{K}$. We derive an upper bound for the second term on the right-hand side of (5.25) as

$$\begin{aligned}
\|\mathbf{\Psi}\mathbf{s}^* - \mathbf{u}\|_1 \le& \sqrt{|\mathcal{K}|}\|\mathbf{\Psi}\mathbf{s}^* - \mathbf{u}\|_2 \\
\le& \sqrt{|\mathcal{K}|}\|\mathbf{\Psi}\mathbf{s}^* - \mathbf{\Psi}\hat{\mathbf{s}}\|_2 + \sqrt{|\mathcal{K}|}\|\mathbf{\Psi}\hat{\mathbf{s}} - \mathbf{u}\|_1 \\
\le& \sqrt{|\mathcal{K}|}\sigma_{\max}(\mathbf{\Psi})\|\mathbf{s}^* - \hat{\mathbf{s}}\|_2 \\
&+ \sqrt{|\mathcal{K}|}\|\mathbf{\Psi}\hat{\mathbf{s}} - \mathbf{u}\|_1 \\
\le& \frac{\sqrt{|\mathcal{K}|}\sigma_{\max}(\mathbf{\Psi})}{\sigma_{\min}(\bar{\mathbf{\Phi}}_r)}\|\bar{\mathbf{\Phi}}_r(\mathbf{s}^* - \hat{\mathbf{s}})\|_2 \\
&+ \sqrt{|\mathcal{K}|}\|\mathbf{\Psi}\hat{\mathbf{s}} - \mathbf{u}\|_1.
\end{aligned} \tag{5.26}$$

For the first term on the right-hand side of (5.25), we have that

$$\begin{aligned}
\xi :=& \min_{\mathbf{u}:\mathrm{supp}(\mathbf{u})\subseteq\mathcal{K}} \|\mathbf{\Psi}\hat{\mathbf{s}} - \mathbf{u}\|_1 \\
=& \max_{\mathbf{v}} \min_{\mathbf{u}} \|\mathbf{\Psi}\hat{\mathbf{s}} - \mathbf{u}\|_1 \\
&+ \mathbf{v}^\top\mathbf{I}_{\mathcal{K}^c,\cdot}(\mathbf{u} - \mathbf{\Psi}\hat{\mathbf{s}}) + \mathbf{v}^\top\mathbf{I}_{\mathcal{K}^c,\cdot}\mathbf{\Psi}\hat{\mathbf{s}} \\
=& \max_{\mathbf{w}:\mathrm{supp}(\mathbf{w})\subseteq\mathcal{K}^c} \min_{\mathbf{u}} \|\mathbf{\Psi}\hat{\mathbf{s}} - \mathbf{u}\|_1 \\
&+ \mathbf{w}^\top(\mathbf{u} - \mathbf{\Psi}\hat{\mathbf{s}}) + \mathbf{w}^\top\mathbf{\Psi}\hat{\mathbf{s}},
\end{aligned} \tag{5.27}$$

where (5.27) results from the Lagrangian of ξ and duality theory. Given the dual certificate $\mathbf{y}$, we have that

$$\begin{aligned}
\xi =& \max_{\substack{\mathbf{w}:\mathrm{supp}(\mathbf{w})\subseteq\mathcal{K}^c, \\ \|\mathbf{w}\|_\infty\le 1}} (\mathbf{y} + \mathbf{w})^\top\mathbf{\Psi}\hat{\mathbf{s}} - \mathbf{y}^\top\mathbf{\Psi}\hat{\mathbf{s}} \\
\le& \|\mathbf{\Psi}\hat{\mathbf{s}}\|_1 - \mathbf{y}^\top\mathbf{\Psi}\hat{\mathbf{s}} + \mathbf{y}^\top\mathbf{\Psi}\mathbf{s}^* - \|\mathbf{\Psi}\mathbf{s}^*\|_1 \\
\le& \mathbf{y}^\top\mathbf{\Psi}(\mathbf{s}^* - \hat{\mathbf{s}}),
\end{aligned} \tag{5.28}$$

where the final inequality is due to the optimality of $\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\}$ and the feasibility of $\{\mathbf{s}^*, \hat{\mathbf{p}}\}$ for (5.5'). Lastly, since $\mathbf{\Psi}^\top\mathbf{y} = \bar{\mathbf{\Phi}}_r^\top\bar{\mathbf{\Phi}}_r(\bar{\mathbf{\Phi}}_r^\top\bar{\mathbf{\Phi}}_r)^{-1}\mathbf{\Psi}^\top\mathbf{I}_{\mathcal{K},\cdot}^\top\mathrm{sign}(\mathbf{\Psi}_{\mathcal{K},\cdot}\mathbf{s}^*)$, we have that

$$\begin{aligned}
&\mathbf{y}^\top\mathbf{\Psi}(\mathbf{s}^* - \hat{\mathbf{s}}) \\
&\le \mathrm{sign}(\mathbf{\Psi}_{\mathcal{K},\cdot}\mathbf{s}^*)^\top\mathbf{I}_{\mathcal{K},\cdot}\mathbf{\Psi}(\bar{\mathbf{\Phi}}_r^\top\bar{\mathbf{\Phi}}_r)^{-1}\bar{\mathbf{\Phi}}_r^\top\bar{\mathbf{\Phi}}_r(\mathbf{s}^* - \hat{\mathbf{s}}) \\
&\le \frac{\sqrt{|\mathcal{K}|}\sigma_{\max}(\mathbf{\Psi})}{\sigma_{\min}(\bar{\mathbf{\Phi}}_r)}\|\bar{\mathbf{\Phi}}_r(\mathbf{s}^* - \hat{\mathbf{s}})\|_2,
\end{aligned} \tag{5.29}$$

where the second inequality results from the fact that every positive scalar and its ℓ_2 norm are equal. We may substitute (5.26) and (5.29) into (5.25) and the fact that $\mathbf{\Psi}$ is full column rank to obtain

$$\|\mathbf{s}^* - \hat{\mathbf{s}}\|_1 \leq \tau_r \|\bar{\mathbf{\Phi}}_r(\mathbf{s}^* - \hat{\mathbf{s}})\|_2,$$

where

$$\tau_r = \frac{\sqrt{|\mathcal{K}|}\sigma_{\max}(\mathbf{\Psi})\|\mathbf{\Psi}^\dagger\|_1}{\sigma_{\min}(\bar{\mathbf{\Phi}}_r)}(2 + \sqrt{|\mathcal{K}|}). \tag{5.30}$$

As $r \to \infty$, we have that

$$\begin{aligned}\|\mathbf{s}^* - \hat{\mathbf{s}}\|_1 &\leq \lim_{r\to\infty} \tau_r \|\bar{\mathbf{\Phi}}_r(\mathbf{s}^* - \hat{\mathbf{s}})\|_2 \\ &\leq 2 \lim_{r\to\infty} \tau_r(\epsilon_R + \alpha),\end{aligned}$$

where by the feasibility of $\{\hat{\mathbf{s}}, \hat{\mathbf{p}}\}$ and $\{\mathbf{s}^*, \hat{\mathbf{p}}\}$ for every $r > 0$, we have that

$$\begin{aligned}\|\bar{\mathbf{\Phi}}_r(\mathbf{s}^* - \hat{\mathbf{s}})\|_2 &\leq \|\bar{\mathbf{\Phi}}_r\mathbf{s}^* + \bar{\mathbf{R}}\hat{\mathbf{p}} - \bar{\mathbf{b}}_r\|_2 \\ &\quad + \|\bar{\mathbf{\Phi}}_r\hat{\mathbf{s}} + \bar{\mathbf{R}}\hat{\mathbf{p}} - \bar{\mathbf{b}}_r\|_2 \\ &\leq 2(\epsilon_R + \alpha).\end{aligned} \tag{5.31}$$

Finally, we return to the equivalent matrix formulation as

$$\sum_{k=1}^{K} \|\hat{\mathbf{S}}_{\mathcal{O}}^{(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}\|_1 \leq 4\tau_r(\epsilon_R + \alpha). \tag{5.32}$$

By the end of the proof of Theorem 2 in [66], we have that $\lim_{r\to\infty} 4\tau_r \leq \tau$, as desired.

5.9 Appendix: Proof of Corollary 1

Consider the following optimization problem

$$\begin{aligned}\min_{\{\mathbf{S}_{\mathcal{O}}^{(k)}\}_{k=1}^K} \quad & \sum_{k=1}^{K} \alpha_k \|\mathbf{S}_{\mathcal{O}}^{(k)}\|_1 + \sum_{k<k'} \beta_{k,k'} \|\mathbf{S}_{\mathcal{O}}^{(k)} - \mathbf{S}_{\mathcal{O}}^{(k')}\|_1 \\ \text{s.t. } & \textstyle\sum_{k=1}^K \|\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\mathbf{S}_{\mathcal{O}}^{(k)} - \mathbf{S}_{\mathcal{O}}^{(k)}\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\|_F^2 \leq \epsilon_R^2, \\ & \mathbf{S}_{\mathcal{O}}^{(k)} = (\mathbf{S}_{\mathcal{O}}^{(k)})^\top, \ \mathrm{diag}(\mathbf{S}_{\mathcal{O}}^{(k)}) = \mathbf{0}, \ \text{for all} k = 1, \ldots, K, \\ & \textstyle\sum_j [\mathbf{S}_{\mathcal{O}}^{(1)}]_{j1} = 1,\end{aligned} \tag{5.33}$$

whose solution is equivalent to the naive solution $\{\tilde{\mathbf{S}}_{\mathcal{O}}^{(k)}\}_{k=1}^K$ described in the statement of Corollary 1. Similarly to (5.5), we can define a vectorized version of (5.33) as

$$\tilde{\mathbf{s}} = \operatorname*{argmin}_{\mathbf{s}} \|\mathbf{\Psi}\mathbf{s}\|_1 \ \text{ s.t. } \ \|\mathbf{\Sigma}\mathbf{s}\|_2 \leq \epsilon_R, \ (\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^\top \mathbf{s} = 1, \tag{5.34}$$

and a version parameterized by $r > 0$ as

$$\tilde{\mathbf{s}}_r = \operatorname*{argmin}_{\mathbf{s}} \|\mathbf{\Psi}\mathbf{s}\|_1 \ \text{ s.t. } \ \|\bar{\mathbf{\Phi}}_r\mathbf{s} - \bar{\mathbf{b}}_r\|_2 \leq \epsilon_R, \tag{5.35}$$

where $\bar{\mathbf{\Phi}}_r$ and $\bar{\mathbf{b}}_r$ are defined as for (5.24) and $\lim_{r\to\infty}\tilde{\mathbf{s}}_r = \tilde{\mathbf{s}}$.

We provide the following upper bound via (5.2)

$$\begin{aligned}
&\Big|\sum_{k=1}^{K}\|\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\mathbf{S}_{\mathcal{O}}^{*(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}\hat{\mathbf{C}}_{\mathcal{O}}^{(k)}\|_F^2\Big|^{\frac{1}{2}}\\
&\leq \Big|\sum_{k=1}^{K}\Big\|(\hat{\mathbf{C}}_{\mathcal{O}}^{(k)} - \mathbf{C}_{\mathcal{O}}^{(k)})\mathbf{S}_{\mathcal{O}}^{*(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}(\hat{\mathbf{C}}_{\mathcal{O}}^{(k)} - \mathbf{C}_{\mathcal{O}}^{(k)})\Big\|_F^2\Big|^{\frac{1}{2}}\\
&+ \Big|\sum_{k=1}^{K}\Big\|\mathbf{P}^{*(k)} - (\mathbf{P}^{*(k)})^{\top}\Big\|_F^2\Big|^{\frac{1}{2}}\\
&\leq \epsilon_R + \delta,
\end{aligned}$$

and similarly to Theorem 2, we apply Lemma 1 to get

$$\|\bar{\mathbf{\Phi}}_r\mathbf{s}^* - \bar{\mathbf{b}}_r\|_2 \leq \epsilon_R + \delta,$$

where $\mathbf{s}^*$ may not be a feasible solution to (5.35). However, by the triangle inequality and the optimality of $\tilde{\mathbf{s}}_r$, there exists $\rho \in [0,1]$ such that

$$\|\mathbf{\Psi}\tilde{\mathbf{s}}_r\|_1 - \|\mathbf{\Psi}\mathbf{s}^*\|_1 \leq \rho\|\mathbf{\Psi}\tilde{\mathbf{s}}_r - \mathbf{\Psi}\mathbf{s}^*\|_1. \tag{5.36}$$

In particular, let $\rho = \max\{0, (\|\mathbf{\Psi}\tilde{\mathbf{s}}_r\|_1 - \|\mathbf{\Psi}\mathbf{s}^*\|_1)/\|\mathbf{\Psi}\tilde{\mathbf{s}}_r - \mathbf{\Psi}\mathbf{s}^*\|_1\}$, where $\rho = 0$ when $\mathbf{s}^*$ is a feasible solution to (5.35), but otherwise, it may be possible that $\rho \in (0,1]$. Furthermore, since $(\mathbf{e}_1 \otimes \mathbf{1}_{O-1})^{\top}\tilde{\mathbf{s}} = 1$, then $\tilde{\mathbf{s}}$ is a feasible solution to (5.35) for every $r > 0$.

We then can introduce a similar inequality to (5.25) as

$$\|\mathbf{\Psi}\mathbf{s}^* - \mathbf{\Psi}\tilde{\mathbf{s}}\|_1 \leq \|\mathbf{\Psi}\tilde{\mathbf{s}} - \tilde{\mathbf{u}}\|_1 + \|\mathbf{\Psi}\mathbf{s}^* - \tilde{\mathbf{u}}\|_1, \tag{5.37}$$

where $\tilde{\mathbf{u}} \in \mathbb{R}^{KO(O-1)/2}$ such that $\mathrm{supp}(\tilde{\mathbf{u}}) \subseteq \mathcal{K}$. The upper bound for the second term of the right-hand side of (5.37) can be found analogously to (5.26), where we have

$$\begin{aligned}
\|\mathbf{\Psi}\mathbf{s}^* - \tilde{\mathbf{u}}\|_1 \leq &\frac{\sqrt{|\mathcal{K}|}\sigma_{\max}(\mathbf{\Psi})}{\sigma_{\min}(\bar{\mathbf{\Phi}}_r)}\|\bar{\mathbf{\Phi}}_r(\mathbf{s}^* - \tilde{\mathbf{s}}_r)\|_2\\
&+ \sqrt{|\mathcal{K}|}\|\mathbf{\Psi}\tilde{\mathbf{s}}_r - \tilde{\mathbf{u}}\|_1.
\end{aligned} \tag{5.38}$$

Similarly to (5.28) in the proof of Theorem 2, we can upper bound the first term as

$$\begin{aligned}
\tilde{\xi} \;:=\; &\min_{\tilde{\mathbf{u}}:\mathrm{supp}(\tilde{\mathbf{u}})\subseteq\mathcal{K}}\|\mathbf{\Psi}\tilde{\mathbf{s}} - \tilde{\mathbf{u}}\|_1\\
\leq\; &\|\mathbf{\Psi}\tilde{\mathbf{s}}\|_1 - \mathbf{y}^{\top}\mathbf{\Psi}\tilde{\mathbf{s}} + \mathbf{y}^{\top}\mathbf{\Psi}\mathbf{s}^* - \|\mathbf{\Psi}\mathbf{s}^*\|_1\\
\leq\; &\mathbf{y}^{\top}\mathbf{\Psi}(\mathbf{s}^* - \tilde{\mathbf{s}}) + \rho\|\mathbf{\Psi}(\mathbf{s}^* - \tilde{\mathbf{s}})\|_1,
\end{aligned} \tag{5.39}$$

where we account for the possible infeasibility of $\mathbf{s}^*$ with (5.36). We may combine (5.39), and (5.38) to obtain

$$\|\tilde{\mathbf{s}} - \mathbf{s}^*\|_1 \leq (\tau_r + \tau_r')(2\epsilon_R + \delta), \tag{5.40}$$

where τ_r is defined in (5.30) and we let

$$\tau_r' := \frac{\rho KO(O-1)(1+\sqrt{|\mathcal{K}|})\sigma_{\max}(\mathbf{\Psi})\|\mathbf{\Psi}^{\dagger}\|_1}{2\sigma_{\min}(\bar{\mathbf{\Phi}}_r)}.$$

As with the proof of Theorem 2, we have that for $r \to \infty$,

$$\sum_{k=1}^{K} \|\tilde{\mathbf{S}}_{\mathcal{O}}^{(k)} - \mathbf{S}_{\mathcal{O}}^{*(k)}\|_1 \leq (\tau + \tau')(\epsilon_R + \tfrac{1}{2}\delta), \tag{5.41}$$

as desired.

Finally, the bound (5.10) is equivalent to the following inequality

$$\alpha^2 \leq \left(\frac{\tau'}{\tau}\right)^2 \epsilon_R^2 + \left(\frac{\tau + \tau'}{2\tau}\right)^2 \delta^2,$$

which is a sufficient condition for the upper bound in (5.8) to be less than the upper bound in (5.9).

Chapter 6

Concluding Remarks

The main goal of this thesis was to address the network topology inference problem from a Graph Signal Processing (GSP) perspective, considering several scenarios involving: i) different models for the graph signals, ii) the presence of hidden nodes, and iii) multilayer networks. To achieve this goal, we focused on two types of setups: 1) implementing more general/elaborated approaches for graph signal modeling; and 2) implementing approaches by considering more realistic scenarios for the network topology inference problem, including the presence of hidden nodes and/or scenarios involving multiple related graphs. This chapter serves a twofold purpose. First, we review our main contributions and place them in context relative to the objectives defined in Chapter 1. Secondly, we identify and discuss different avenues for future research.

This thesis started by tackling the challenge of learning a graph from complex data generated in networks. Existing methods may not accurately estimate the underlying graph due to the intricate nature of the data, the simplicity of the approaches used to model graph data, or the limited number of available samples. Our proposed solution involved generalizing current methods by modeling the graph signals as both Gaussian and stationary. For the designed method, we proposed convex relaxations and an efficient algorithm to handle problems where the number of nodes in the network is large. We also provided theoretical convergence guarantees for the proposed algorithm to stationary points of the original problem. We tested the proposed algorithm using synthetic and real data experiments where the results demonstrated its ability to accurately estimate graphs from complex data in scenarios where existing methods fell short. This work, summarized in Chapter 3, addressed the problem described in **(P1)** and aligned well with the goals in **(O1)**.

Next, in addition to considering generalizations for signal modeling, we incorporated the presence of hidden nodes into the network topology inference problem. We assumed that the graph signals were smooth, stationary, or a combination of both, and that the number of hidden nodes was much smaller than the observed ones. Then, we formulated constrained optimization problems that took into account the different signal models and the topological structure. Several of the proposed formulations were non-convex, for which we designed convex iterative algorithms and presented convergence results to a stationary point of the original problem. The performance of the proposed approaches has been compared to existing alternatives through synthetic and real data experiments in scenarios where the information of some nodes was not accessible, showing the benefits of considering their influence. This work was presented in Chapter 4, addressed part

of the challenges outlined in **(P1)** and **(P2)**, and aligned with the objectives described in **(O2)**.

Lastly, we focused on learning graphs in environments where we had access to data generated in multiple similar networks in the presence of hidden nodes, a scenario that has not been previously addressed in the literature. We formulated a method that modeled the presence of hidden variables in the graph and also took advantage of the similarity between the different graphs for both observed and hidden nodes to improve the joint estimation of the considered graphs. We ended up with a non-convex problem formulation, which we had to relax using convex approximations, and then solved the problem using an iterative algorithm. We demonstrated theoretical guarantees for recoverability and bounded the error of the estimated graphs in terms of the number of samples and the influence of the hidden nodes. The performance of the proposed approach was tested in several synthetic and real data scenarios showing the capability of our algorithm to jointly exploit the structure of the multiple networks, the presence of hidden nodes, and the signal model. This work was described in Chapter 5, addressed the problems described in **(P3)**, and was aligned with the research goals outlined in **(O3)**.

6.1 Future lines of research

We conclude this thesis by presenting several avenues for further research. The proposed directions range from exploring feasible generalizations of the schemes described in the previous chapters to broader and more challenging research paths. Directions in the first category are typically well-defined and feasible for short- to medium-term exploration, while those in the second category represent a medium- to long-term research roadmap.

6.1.1 Generalizations of the previous works

Joint multilayer graph topology inference for GMRF with hidden variables. As mentioned in the previous chapter, the use of multilayer graphs is gaining traction to accurately describe real-world datasets containing observations from similar networks. However, progress in this area when hidden nodes are present is almost non-existent. To address this issue, we will propose an approach for jointly estimating similar graph topologies in the presence of hidden nodes, assuming the data associated with each graph can be modeled as a GMRF. In general, alternative metrics for measuring the impact of hidden nodes and the similarity between observed and hidden nodes can be explored based on prior information or depending on the specific characteristics of the application at hand. Another avenue of interest, which is very common but also more complex, involves scenarios where the set of nodes is not shared between the considered networks. Additionally, the goal is not only to design an effective algorithm but also to analyze its performance. To that end, previous works in [54, 117] offer promising starting points.

Online network topology inference with hidden variables. The increasing use of streaming data in contemporary environments motivates the development of network topology inference methods that can operate in an online setting. A critical aspect of this research is understanding the relationship between the network and the data, and being able to model their evolution over time. Given the nature of streaming data, time complexity plays a crucial role in the implementation of algorithms for online settings. In some cases, sacrificing optimality or accuracy should be considered to satisfy the requirements of the specific application at hand. Existing work in online settings focuses on signal modeling by assuming either smooth or stationary graph signals [93, 118]. While our setup is more complicated, those two foundational works can serve as a starting point for

incorporating the presence of hidden nodes. Overall, the goal is to generalize existing methods to meet the requirements of online environments, paying particular attention to network size and time complexity. By adapting the methodological design to the unique requirements of real-time processing, future research efforts can yield innovative approaches that facilitate *dynamic graph topology estimation* in online settings.

Design of reduced-complexity algorithms. Datasets associated with networks often contain a large number of nodes, requiring network topology inference methods to have low computational complexity for practical deployment. Robust and rigorously designed efficient algorithms are crucial for this purpose, particularly when dealing with multiple networks and hidden variables. Hence, we plan to redesign and enhance our current algorithms to accommodate larger-size networks, enabling their application to a broader set of scenarios. To reach this objective, we will explore the use of iterative convex approximation functions and stochastic approximation to simplify complex computations and provide theoretical convergence guarantees. This will allow us not only to reduce computational complexity but also to improve adaptability and scalability, which are critical in real-world applications.

Application of the proposed methods in financial engineering. Large datasets of financial assets are a rich source of information that can be used to shed light on the (hidden) relationships between the considered assets. Hence, modeling the assets as nodes, the relationships as edges, and using asset prices to learn one or multiple graphs can be a critical step in financial applications such as portfolio optimization. The main issue with these types of datasets is that unveiling direct relationships between financial assets is highly non-trivial. Our goal is to adapt and customize the methods proposed in this thesis to the specificities of the financial datasets at hand. Firstly, we aim to create signal models that accurately capture the intricate structure of financial data. Secondly, we seek to establish a model for the connections between nodes, incorporating prior information about external factors such as financial news or events. The resulting models could be used in various tasks such as portfolio optimization, asset clustering, covariance estimation, and others. The overall goal is to uncover a meaningful graph that captures the relationships between the considered nodes (e.g., financial assets) and provides valuable insights.

6.1.2 New research paths in GSP

Consideration of more intricate graph signal models. Many existing models for graph signals from the literature assume linearity and real-valued observations. However, there are situations where the data exhibits greater complexity, such as non-linear dependencies or categorical data. As a result, there is a research opportunity to develop specific methods that address these complexities in the data structure.

One way to better capture the nonlinear or categorical aspects of the data is by employing nonlinear signal models. In this context, our objective is to propose approaches for inferring the network structure based on signal modeling, specifically considering nonlinear or categorical graph signals.

To achieve this goal, a careful examination of the implications associated with these novel graph signal models is essential. Additionally, it is crucial to define the necessary GSP tools tailored for handling these complexities. Recent works in the literature [119, 120] can serve as a starting point to address the problem at hand.

From learning graphs to learning higher-order interactions. The fundamental approach in

GSP is to use graphs to process, learn, and extract knowledge from node signals. As a result, most network-topology inference works focus on unveiling pairwise interactions among the nodes. However, there is a growing interest in processing data using higher-order structures such as simplicial complexes [121,122] and hypergraphs [123,124]. Motivated by this, some recent studies have started to look at the problem of inferring higher-order interactions between nodes, assuming knowledge of the underlying graph topology or having access to signals defined on the edges of the graph. Hence, one of our future lines of work is to use signals and higher-order signal models to infer the presence of higher-order interactions. We aim to use not only node signals but also edge signals when available and to address scenarios with missing values or partial observability in the data. A key step in this task will be to define new signal models that relate nodal and edge signals to the topology of the higher-order graph. Then, the next step will be the postulation of tractable optimization problems that allow identifying the pairwise and higher-order interactions. As in the rest of this thesis, we will also aim at incorporating assumptions that are consistent with the complexity inherent in real-world scenarios, including missing information, outliers, hidden nodes, and limited observations, to name a few. We have just started exploring this avenue, see [44, 45] for preliminary results.

Incorporation of (uncertainty graph models). The widespread assumption when learning graphs from data is that edges are sparse, with most methods failing to leverage more complex prior information about the graph topology. Clearly, graph estimation can be improved by incorporating known (desired) graph structure, provided that there is an efficient way to incorporate the prior information into the inference and optimization schemes. Joint multilayer graph learning works [7, 65] achieve this by introducing regularizers promoting that the multiple estimated graphs are similar, primarily by measuring similarity via their edge support. Other recent efforts integrate prior graph information by incorporating structural conditions, such as imposing spectral constraints [125], assuming that the estimated graph is drawn from a known graphon [108], or assuming similarity in terms of motif density with a known reference graph [126]. However, these approaches often require access to the graph distribution or a similar reference graph, making them unsuitable for many scenarios and potentially yielding suboptimal solutions. Our goal is to develop different topological regularizers to encourage the estimated graph to exhibit the desired structure. Specifically, we will generalize previous methods by: i) incorporating more informative graph priors that are present in many real-world networks (e.g., community structure and modularity in social networks) and ii) focusing on learning/designing the score function (i.e., the gradient of the regularizer) rather than the regularizer itself.

Acronyms

ADMM Alternating Direction Method of Multipliers

AWGN Additive White Gaussian Noise

BA Barabási-Albert

BSUM Block Successive Upper Bound Minimization

CVX Convex Optimization Toobox

ER Erdős-Rényi

GL Graphical Lasso

GMRF Gaussian Markov Random Field

GSO Graph-Shift Operator

GSP Graph Signal Processing

KKT Karush-Kuhn-Tucker

LV Local Variation

LVGL Latent Variable Graphical Lasso

MM Majorization-Minimization

NMI Normalized Mutual Information

PSD Positive Semi Definite

RBF Radial Basis Function

SBM Stochastic Block Model

SW Small-World

Bibliography

[1] A. Buciulea and A. G. Marques, "Graph learning from gaussian and stationary graph signals," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2023, pp. 1–5.

[2] A. Buciulea, J. Ying, A. G. Marques, and D. Palomar, "Polynomial graphical Lasso: Learning edges from Gaussian graph-stationary signals," *arXiv preprint arXiv:2404.02621*, 2024.

[3] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[4] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 467–483, 2017.

[5] A. Buciulea, S. Rey, and A. G. Marques, "Learning graphs from smooth and graph-stationary signals with hidden variables," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 8, pp. 273–287, 2022.

[6] S. Rey, M. Navarro, A. Buciulea, S. Segarra, and A. G. Marques, "Joint graph learning from Gaussian observations in the presence of hidden nodes," in *Conf. Signals, Syst., Computers (Asilomar)*. IEEE, 2022, pp. 53–57.

[7] M. Navarro, S. Rey, A. Buciulea, A. G. Marques, and S. Segarra, "Joint network topology inference in the presence of hidden nodes," *arXiv preprint arXiv:2306.17364*, 2023.

[8] D. Easley and J. Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[9] M. E. J. Newman, "The structure and function of complex networks," *SIAM rev.*, vol. 45, no. 2, pp. 167–256, 2003.

[10] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Statistical Science*, vol. 19, no. 3, pp. 499–517, 2004.

[11] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive data sets*. Cambridge University Press, 2020.

[12] K. Nodop, R. Connolly, and F. Girardi, "The field campaigns of the european tracer experiment (etex): Overview and results," *Atmospheric Environment*, vol. 32, no. 24, pp. 4095–4108, 1998.

[13] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. New York, NY: Springer, 2009.

[14] O. Sporns, *Discovering the Human Connectome*. Boston, MA: MIT Press, 2012.

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

[16] P. Djuric and C. Richard, *Cooperative and Graph Signal Processing: Principles and Applications*. Academic Press, 2018.

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

[18] A. G. Marques, N. Kiyavash, J. M. F. Moura, D. V. D. Ville, and R. Willett, "Graph signal processing: Foundations and emerging directions (editorial)," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 11–13, Nov. 2020.

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

[20] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, 2014.

[21] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274–288, 2016.

[22] S. Sardellitti, S. Barbarossa, and P. D. Lorenzo, "On the graph Fourier transform for directed graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 796–811, 2017.

[23] S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind identification of graph filters," *IEEE Trans. Signal Process.*, vol. 65, no. 5, pp. 1146–1159, 2017.

[24] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4117–4131, 2017.

[25] F. J. Iglesias, S. Segarra, S. Rey, A. G. Marques, and D. Ramírez, "Demixing and blind deconvolution of graph-diffused sparse signals," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2018, pp. 4189–4193.

[26] J. Liu, E. Isufi, and G. Leus, "Filter design for autoregressive moving average graph filters," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 47–60, 2018.

[27] N. Tremblay, P. Gonçalves, and P. Borgnat, "Design of graph filters and filterbanks," in *Cooperative Graph Signal Process.* Elsevier, 2018, pp. 299–324.

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

[29] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2014, pp. 3864–3868.

[30] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1832–1843, 2015.

[31] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Appl. Comput. Harmonic Anal.*, vol. 44, no. 2, pp. 446–475, 2018.

[32] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačevic, "Signal denoising on graphs via graph filtering," in *Global Conf. Signal Info. Process. (GlobalSIP)*. IEEE, 2014, pp. 872–876.

[33] S. Chen, A. Sandryhaila, G. Lederman, Z. Wang, J. M. F. Moura, P. Rizzo, J. Bielak, J. H. Garrett, and J. Kovačevic, "Signal inpainting on graphs via total variation minimization," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2014, pp. 8267–8271.

[34] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, "Graph signal denoising via trilateral filter on graph spectral domain," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 137–148, 2016.

[35] S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević, "Signal recovery on graphs: Variation minimization," *IEEE Trans. Signal Process.*, vol. 63, no. 17, pp. 4609–4624, 2015.

[36] V. Kalofolias, "How to learn a graph from smooth signals," in *Intl. Conf. Artif. Intel. Statist. (AISTATS)*. J. Mach. Learn. Res., 2016, pp. 920–929.

[37] E. Pavez and A. Ortega, "Generalized Laplacian precision matrix estimation for graph signal processing," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*, 2016, pp. 6350–6354.

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

[39] S. Segarra, A. G. Marques, M. Goyal, and S. Rey, "Network topology inference from input-output diffusion pairs," in *IEEE Wrkshp. Statistical Signal Process. (SSP)*. IEEE, 2018, pp. 508–512.

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

[41] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Sparse structural equation modeling for inference of gene regulatory networks exploiting genetic perturbations," *PLoS, Comput. Biology*, vol. 9, no. 5, pp. 1–13, May 2013.

[42] A. Buciulea, S. Rey, C. Cabrera, and A. G. Marques, "Network reconstruction from graph-stationary signals with hidden variables," in *Conf. Signals, Syst., Computers (Asilomar)*. IEEE, 2019, pp. 56–60.

[43] S. Rey, A. Buciulea, M. Navarro, S. Segarra, and A. G. Marques, "Joint inference of multiple graphs with hidden variables from stationary graph signals," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2022, pp. 5817–5821.

[44] A. Buciulea, E. Isufi, G. Leus, and A. G. Marques, "Learning graphs and simplicial complexes from data," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*, 2024, pp. 9861–9865.

[45] A. Buciulea, E. Isufi, G. Leus, and A. G. Marques, "Learning the topology of a simplicial complex using simplicial signals: A greedy approach," in *IEEE Sensor Array and Multichannel Signal Process. Wrksp. (SAM 2024)*, 2024.

[46] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5911–5926, 2017.

[47] M. Yuan and Y. Lin, "Model selection and estimation in the Gaussian graphical model," *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[48] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Trans. Signal Process.*, vol. 65, no. 22, pp. 5911–5926, Nov. 2017.

[49] B. Girault, "Stationary graph signals using an isometric graph translation," in *European Signal Process. Conf. (EUSIPCO)*, Aug 2015, pp. 1516–1520.

[50] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, 2017.

[51] Y. Zhu, M. T. Schaub, A. Jadbabaie, and S. Segarra, "Network inference from consensus dynamics with unknown parameters," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 300–315, 2020.

[52] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 3, pp. 484–499, 2017.

[53] Y. Li and G. Mateos, "Identifying structural brain networks from functional connectivity: A network deconvolution approach," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*, 2019, pp. 1135–1139.

[54] M. Navarro, Y. Wang, A. G. Marques, C. Uhler, and S. Segarra, "Joint inference of multiple graphs from matrix polynomials," *J. Mach. Learn. Res.*, 2020.

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

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

[57] J. V. D. M. Cardoso, J. Ying, and D. P. Palomar, "Algorithms for learning graphs in financial markets," *arXiv preprint arXiv:2012.15410*, 2020.

[58] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "Graph topology inference based on sparsifying transform learning," *IEEE Trans. Signal Process.*, vol. 67, no. 7, pp. 1712–1727, 2019.

[59] N. Meinshausen and P. Buhlmann, "High-dimensional graphs and variable selection with the lasso," *Ann. Statist.*, vol. 34, pp. 1436–1462, 2006.

[60] G. V. Karanikolas, G. B. Giannakis, K. Slavakis, and R. M. Leahy, "Multi-kernel based nonlinear models for connectivity identification of brain networks," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2016, pp. 6315–6319.

[61] J. Mei and J. Moura, "Signal processing on graphs: Estimating the structure of a graph," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*, 2015, pp. 5495–5499.

[62] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, 2017.

[63] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, "Latent variable graphical model selection via convex optimization," *Annu. Allerton Conf. Commun., Control, Comput.*, vol. 40, no. 4, pp. 1935–1967, 2012.

[64] A. Chang, T. Yao, and G. I. Allen, "Graphical models and dynamic latent factors for modeling functional brain connectivity," *2019 IEEE Data Science Wrksp. (DSW)*, pp. 57–63, 2019.

[65] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *J. Roy. Statistical Soc.: Ser. B (Statistical Methodology)*, vol. 76, no. 2, pp. 373–397, 2014.

[66] M. Navarro, Y. Wang, A. G. Marques, C. Uhler, and S. Segarra, "Joint inference of multiple graphs from matrix polynomials," *J. Mach. Learn. Res.*, vol. 23, no. 76, pp. 1–35, 2022.

[67] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[68] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, 2019.

[69] S. S. M. Sevilla, A. G. Marques, "Estimation of partially known gaussian graphical models with score-based structural priors," in *27th Intl. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2024.

[70] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statistical Soc.: Ser. B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[71] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statistical Soc.: Ser. B (Statistical Methodology)*, vol. 58, no. 1, pp. 267–288, 1996.

[72] E. J. Candes, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, 2008.

[73] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. on Signal Processing*, vol. 65, no. 3, pp. 794–816, 2017.

[74] J. P. Boyle and R. L. Dykstra, "A method for finding projections onto the intersection of convex sets in Hilbert spaces," in *Advances in order restricted statistical inference*. Springer, 1986, pp. 28–47.

[75] M. Li, D. Sun, and K.-C. Toh, "A majorized ADMM with indefinite proximal terms for linearly constrained convex composite optimization," *SIAM J. on Optimization*, vol. 26, no. 2, pp. 922–950, 2016.

[76] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[77] J. V. d. M. Cardoso, J. Ying, and D. Palomar, "Learning bipartite graphs: Heavy tails and multiple components," *Adv. in Neural Inf. Process. Syst.*, vol. 35, pp. 14 044–14 057, 2022.

[78] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," https://cvxr.com/cvx, Mar. 2014.

[79] J. V. d. M. Cardoso and D. Palomar, "Learning undirected graphs in financial markets," in *Conf. Signals, Syst., Computers (Asilomar)*, 2020, pp. 741–745.

[80] L. Condat, "Fast projection onto the simplex and the l_1 ball," *Mathematical Programming*, vol. 158, no. 1-2, pp. 575–585, 2016.

[81] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[82] V. Kalofolias, "How to learn a graph from smooth signals," in *Intl. Conf. Artif. Intel. Statist. (AISTATS)*. J Mach. Learn. Res., 2016, pp. 920–929.

[83] B. M. Lake and J. B. Tenenbaum, "Discovering structure by learning sparse graph," *Annu. Cognitive Sc. Conf.*, pp. 778 – 783, 2010.

[84] B. Baingana, G. Mateos, and G. B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *IEEE J. Sel. Topics Signal Process.*, vol. 8, pp. 563–575, Aug. 2014.

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

[86] Y. Shen, B. Baingana, and G. B. Giannakis, "Kernel-based structural equation models for topology identification of directed networks," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2503–2516, May 2017.

[87] X. Yang, M. Sheng, Y. Yuan, and T. Q. S. Quek, "Network topology inference from heterogeneous incomplete graph signals," *IEEE Trans. Signal Process.*, vol. 69, pp. 314–327, 2020.

[88] A. Anandkumar, D. Hsu, A. Javanmard, and S. Kakade, "Learning linear Bayesian networks with latent variables," in *Int. Conf. Mach. Learn. (ICML)*, 2013, pp. 249–257.

[89] J. Mei and J. M. F. Moura, "SILVar: Single index latent variable models," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2790–2803, 2018.

[90] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, Jul. 2017.

[91] V. Kalofolias and N. Perraudin, "Large scale graph learning from smooth signals," in *Int. Conf. Learn. Representations (ICLR)*, 2018.

[92] X. Wang, C. Yao, H. Lei, and A. M.-C. So, "An efficient alternating direction method for graph learning from smooth signals," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2021, pp. 5380–5384.

[93] R. Shafipour and G. Mateos, "Online topology inference from streaming stationary graph signals with partial connectivity information," *Algorithms*, vol. 13, no. 9, p. 228, 2020.

[94] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statistical Soc.: Ser. B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[95] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. Graphical Statist.*, vol. 22, no. 2, pp. 231–245, 2013.

[96] S. Segarra, Y. Wang, C. Uhler, and A. G. Marques, "Joint inference of networks from stationary graph signals," in *Conf. Signals, Syst., Computers (Asilomar)*. IEEE, 2017, pp. 975–979.

[97] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, 2016.

[98] N. Srebro and A. Shraibman, "Rank, trace-norm and max-norm," in *Intl. Conf. Comp. Learning Theory*. Springer, 2005, pp. 545–560.

[99] M. Hong, M. Razaviyayn, Z. Luo, and J. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 57–77, 2016.

[100] A. Basilevsky, *Statistical factor analysis and related methods*. Wiley, 1994.

[101] D. J. Bartholomew, M. Knott, and I. Moustaki, *Latent variable models and factor analysis: A unified approach*. Wiley, 2011.

[102] Online, "Meteoswiss, historic measured meteorological data," *https://www.meteoswiss.admin.ch/home/measurement-and-forecasting-systems/datenmanagement/historic-measured-meteorological-data.html*, 2010.

[103] D. S. Marks, L. J. Colwell, R. Sheridan, T. A. Hopf, A. Pagnani, R. Zecchina, and C. Sander, "Protein 3d structure computed from evolutionary sequence variation," *PLoS ONE*, vol. 6, no. 12, pp. 1–20, 2011.

[104] S. Feizi, D. Marbach, M. Medard, and M. Kellis, "Network deconvolution as a general method to distinguish direct dependencies in networks," *Nat Biotech*, vol. 31, no. 8, pp. 726–733, 2013.

[105] Online, "Database on swiss popular votes," *[Online]. http://www.swissvotes.ch*, 2012.

[106] S. Rey, V. M. Tenorio, and A. G. Marques, "Robust graph filter identification and graph denoising from signal observations," *IEEE Trans. Signal Process.*, vol. 71, pp. 3651–3666, 2023.

[107] S. S. Saboksayr and G. Mateos, "Accelerated graph learning from smooth signals," *IEEE Signal Process. Lett.*, vol. 28, pp. 2192–2196, 2021.

[108] T. M. Roddenberry, M. Navarro, and S. Segarra, "Network topology inference with graphon spectral penalties," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2021, pp. 5390–5394.

[109] S. Rey, T. M. Roddenberry, S. Segarra, and A. G. Marques, "Enhanced graph-learning schemes driven by similar distributions of motifs," *IEEE Trans. Signal Process.*, vol. 71, pp. 3014–3027, 2023.

[110] Y. Murase, J. Török, H. H. Jo, K. Kaski, and J. Kertész, "Multilayer weighted social network model," *Physical Review E*, vol. 90, no. 5, p. 052810, 2014.

[111] J. Arroyo, A. Athreya, J. Cape, G. Chen, C. E. Priebe, and J. T. Vogelstein, "Inference for multiple heterogeneous networks with a common invariant subspace," *J. Mach. Learn. Res.*, vol. 22, no. 142, pp. 1–49, 2021.

[112] M. Navarro and S. Segarra, "Joint network topology inference via a shared graphon model," *IEEE Trans. Signal Process.*, 2022.

[113] Y. Wang, S. Segarra, and C. Uhler, "High-dimensional joint estimation of multiple directed Gaussian graphical models," *ejs*, vol. 14, no. 1, pp. 2439–2483, 2020.

[114] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality," *Math. Operations Res.*, vol. 35, no. 2, pp. 438–457, 2010.

[115] H. Zhang, M. Yan, and W. Yin, "One condition for solution uniqueness and robustness of both ℓ1-synthesis and ℓ1-analysis minimizations," *Advances in Computat. Math.*, vol. 42, no. 6, pp. 1381–1399, 2016.

[116] S. Friedland and M. Stawiska, "Some approximation problems in semi-algebraic geometry," *Banach Center Publications*, vol. 107, pp. 133–147, 2015.

[117] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu, "High-dimensional covariance estimation by minimizing ℓ1-penalized log-determinant divergence," *Electron. J. Statistics*, vol. 5, pp. 935–980, 2011.

[118] S. S. Saboksayr, G. Mateos, and M. Cetin, "Online graph learning under smoothness priors," in *European Signal Process. Conf. (EUSIPCO)*. IEEE, 2021, pp. 1820–1824.

[119] G. Leus, M. Yang, M. Coutino, and E. Isufi, "Topological volterra filters," in *IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*. IEEE, 2021, pp. 5385–5399.

[120] Q. Yang, M. Coutino, G. Leus, and G. B. Giannakis, "Autoregressive graph volterra models and applications," *J. on Advances in Signal Process.*, vol. 2023, no. 1, pp. 1–21, 2023.

[121] S. Barbarossa and S. Sardellitti, "Topological signal processing over simplicial complexes," *IEEE Trans. Signal Process.*, vol. 68, pp. 2992–3007, 2020.

[122] M. T. Schaub, Y. Zhu, J.-B. Seby, T. M. Roddenberry, and S. Segarra, "Signal processing on higher-order networks: Livin'on the edge... and beyond," *Signal Process.*, vol. 187, p. 108149, 2021.

[123] C. Berge, *Hypergraphs: combinatorics of finite sets*. Elsevier, 1984, vol. 45.

[124] J. G. Young, G. Petri, and T. P. Peixoto, "Hypergraph reconstruction from network data," *Comm. Physics*, vol. 4, no. 1, p. 135, 2021.

[125] S. Kumar, J. Ying, J. V. de Miranda Cardoso, and D. Palomar, "Structured graph learning via laplacian spectral constraints," *Advances Neural Inf. Process. Syst.*, vol. 32, 2019.

[126] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.