SAPIENZA
UNIVERSITÀ DI ROMA

Graph-based Learning under Model Perturbations

Facoltà dell'Informazione, Informatica e Statistica

Dottorato di Ricerca in Information and Communication Technologies – XXXII Ciclo

Candidate

Elena Ceci
ID number 1464471

Thesis Advisor

Prof. Sergio Barbarossa

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Information and Communication Technologies

March, 2020

Thesis defended on February 18, 2020
in front of a Board of Examiners composed by:

Prof. Guglielmo D'Inzeo (chairman)
Prof. Luca Facheris
Dr. Andrea Bartolini
Prof. Andrea Detti
Prof. Paolo Baccarelli

Graph-based Learning under Model Perturbations
Ph.D. thesis. Sapienza – University of Rome

© 2019 Elena Ceci. All rights reserved

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: elena.ceci@uniroma1.it

to my family

Abstract

We live in the era of data deluge, where billions of gigabytes of data are generated and collected every day. Such a big amount of data has to be processed with the aim of improving our life in social, economic, scientific, medical aspects and more. Machine learning is a growing field that faces the challenges related to this ever-increasing data amount, like storage and processing, adversaries identification, denoising, time-variability, etc. Among several machine learning tools, graph-based methods are well-appreciated for their ability in capturing relevant information, recognizing patterns in a big amount of data, elaborate high dimensional signals or recover missing data. However, several existing approaches based on graphs are limited to tackle too ideal cases, where topology and/or signal perturbations are not considered.

The present thesis aims at robustifying, against possible perturbations, learning tools used to accomplish several graph-learning tasks. In fact, in many cases, the graph underlying a network presents topology uncertainties/perturbations. A mismatch between the actual graph and the presumed one might be the result of the presence of graph topology inference errors, outliers, unexpected links failure, or model mismatch. One of the goals of this thesis is to analyze some graph signal processing tools taking into account topological perturbations. By incorporating any available prior knowledge on perturbations statistics, small perturbation theory of Laplacian matrices plays a key rule in our study.

Small perturbation theory is instrumental also to accomplish the second goal of this thesis: Given a graph topology, we aim at identifying the edge whose perturbation causes the largest changes in the connectivity of the network.

Then, we address two graph-based learning tasks in the presence of signal and topology perturbations. The first, is the topology identification tasks that may be affected by signal errors, due to outliers, adversaries or observation inaccuracy. To solve this problem, we rely on structural equation models, where signal errors may appear in both the input and output matrices. The second task that we analyze is the signal inference under topology perturbations. In both tasks, we develop total least squares approaches to take into account signal and/or topology perturbations.

Finally, several numerical results show how perturbation-aware methods outperform classical methods that ignore possible perturbations.

Acknowledgements

In the last years, I went through challenging and exciting experiences, during which I have been guided and supported by many people.

First, my Ph.D. advisor, Prof. Sergio Barbarossa, he gave me the opportunity to work with him and pursue the PhD. Thank you for this, and for showing me how to face the research with excitement, curiosity and ambition to chase original results.

I thank Prof. Giannakis, my supervisor at UoM. Thank you for your mentoring and your kind welcome in your group/family in Minneapolis. I also thank all the people of the SPiNCOM group that helped me in my research and studies at UoM and collecting beautiful life experiences.

In the SPCom group at Sapienza, I met two brilliant researchers: Stefania and Paolo. You are good friends, and I admire you for your perseverance and passion for the work. Thank you for your professional and personal support.

Thanks, Francesca and Mattia, my friends and mates at the SPCom group. Sharing the lab with you for the last three years created a strong bond among us, and because of this, I will remember the days at the lab with joy. Thank you to all my old friends especially, Annalisa, Gaia and Eugenio, for the happy moments we enjoyed together and because I know I can always count on you.

Thanks, Vassili, with sweetness and wisdom, you helped me in everything, especially to be strong and motivated. We shared many beautiful and happy moments in the past years and I cannot wait to continue to do so.

Finally, a big thank goes to my family, they always motivated me to do my best and pursue my goals. With love, they guided me in all the experiences of my life. Thank you.

Contents

Acronyms

a.k.a. also known as. 17, 82

AD Alternating Descend. 99

CDF Cumulative Distribution Function. 55, 58

FIER Failures Identification Error Rate. 10, 52, 54

GFT Graph Fourier Transform. 31, 32, 34

GSP Graph Signal Processing. 31–33

ICA independent Component analysis. 32

ID Identification. 82

LoS Line of Sight. 55

LS Least-Squares. 11, 82, 99–102, 104

MIMO Multi Input Multi Output. 11, 57, 58

MMSE Minimum Mean Square Error. 10, 48, 50

MSE Mean Square Error. 11, 48, 99–101, 103

r.v. random variable. 43, 55, 61, 62

RGG Random Geometric Graph. 39, 61, 62

SEM Structural Equation Model. 81–83, 86, 93, 96, 99, 101, 102, 104

SISO Single Input Single Output. 11, 55, 58

SNR Signal to Noise Ratio. 10, 11, 54, 55, 99, 101

STLS Structured Total Least-Squares. 93, 101

SWTLS Structured Weighted Total Least-Squares. 86, 96, 104

TLS Total least-Squares. 11, 81–87, 91, 93–95, 98–102, 104

List of Figures

Chapter 1

Introduction

1.1 Motivation

In the last two decades, we have witnessed a huge increase in the data amount to be analyzed. The increasing collection of data we are experiencing comes not only from books, texts, audio, video, but data arise also from internet-of-things (IoT) devices, activity of cell phone users, in health care for recording digital reports of patients, metabolic or protein interaction networks, social networks, and so on. Such a huge amount of data needs to be processed and new machine learning tools are required to capture relevant information. Among several methodologies, in the last decade, graph-based machine learning has permeated social, biological and technological systems.

A graph is an abstract representation of the interactions between agents in complex systems. In the graph representation, nodes are the agents and the edges describe the interactions, see Fig. 1.1 for a graphical illustration.

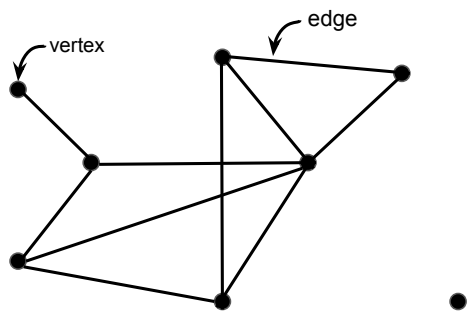


Figure 1.1. Small graph composed by 8 nodes (dots) and 10 edges(lines)

The first benefit of graph representations is that they are important to visualize data in a way that facilitates the extraction of structural features, especially for graphs

with a high degree of modularity. The second benefit of graph representations is to compute several measures and metrics useful to understand what the data are telling us since in most cases the visualization is not useful. Indeed, measures of centrality quantify how important vertices or edges are in a networked system. The degree centrality is one of those measures; it counts the number of edges, i.e. degree, attached to it. It is useful in a social network, for instance, where the degree of an individual is the number of friends this person has within the network. The vertices with high degree are called "hubs", and their identification is critical to reveal central vertices of the network. For example, in social networks, the hubs represent few central individuals with many acquaintances. Both empirical and theoretical results indicate that hubs can have different effects on the networks, playing a central role in particular in-network transport phenomena and resilience, despite being few. A network concept that commonly occurs in real networks and that has practical implications is the so-called *small-world effect* [1]. Defining the geodesic distance between two vertices as the minimum number of edges to traverse, to go from one vertex to another, the effect observed in real networks is that the mean geodesic distance between any vertex pair of the network is very short. Commonly, this effect is also known as the "the six degrees of separation" and it empirically claims that anyone in the world is linked to anyone else via a sequence of no more than five intermediate acquaintances. Such a small-world effect has important implications, for example, in the spread of news in social networks, where news can pass between any pair of individuals of the network with at most six "hops" [2].

Another important network characteristic is the presence of communities. In social networks, there are sub-communities of people strongly interconnected within one another, since groups of people may share most of their friends. In the networks of business relationships between companies, there are groups of companies strongly interconnected between one another within the same sector they operate. The necessity to reveal communities is to capture the level and concept of organization among data that without graph instruments are impossible to see.

Moreover, graphs can model data and complex interactions among them. Each node may have attributes and those are modeled as signals over a graph; for example, the temperature in a given city on a given day in a weather network. Thus, classical signal processing concepts, such as Fourier transform, filtering and frequency response, optimal sampling strategies, have been extended to data residing on graphs, from the rising fields of graph signal processing (GSP) [3].

Similarly to classical signal processing, signals defined over graphs, or graph signals for short, can have properties, such as smoothness, if the signals of neighboring nodes tend to have similar values. The Fourier transform is extended to the case of

graph signals, allowing us to talk for example of frequency and bandlimitedness or graph signal filtering. Operations like sampling and denoising have been accommodated for the graph case, mimicking the classical ones. Moreover, we can learn the underlying structure from data, when the graphs cannot be directly observed [4].

Network examples

Examples of networks may appear in different fields, social, biological, communication science, see figure 1.2 for illustrative examples.

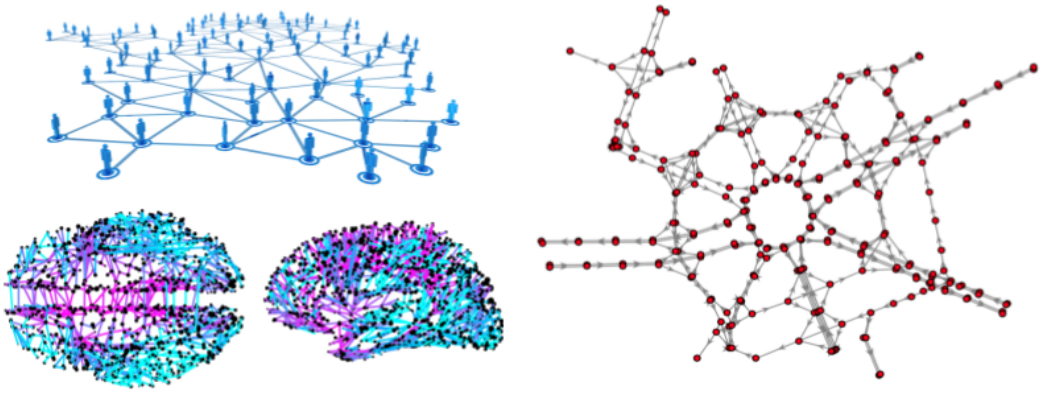


Figure 1.2. Networks examples: social network (top left), brain network (bottom left) and vehicular network (right).

Table 1.1 collects some examples of social networks. In social network analysis, the aim is to discover patterns of interaction between social agents in social networks and the implications of these relationships rather than investigating the social entities themselves. Such analysis is instrumental to extract knowledge from networks and, consequently, in the process of problem solving. For example, some companies operating in the sector of mobile telecommunications, apply graph-based tools to the phone call networks and use them to identify customer's profiles and to recommend personalized mobile phone tariffs, according to these profiles, see [5] and reference therein.

Networks occur in several situations in biology. A first example is brain networks useful in clinical and cognitive neuroscience. Recent studies have discovered that the information processing capabilities of the brain are facilitated by the existence of a complex underlying network. However, such an underlying network is not directly observable and must be inferred from processes measured at different points of the brains (network nodes) [6]. First of all, these processes need to be observed for example using functional magnetic resonance imaging (fMRI), which is a powerful tool to capture varying blood oxygenation patterns modulated by

Table 1.1. Examples of social networks

Examples	Applications
Friendship networks	College/school students, organizations or web (e.g. Facebook)
Follower networks	Twitter, LinkedIn, Pinterest, etc.
Preference similarity networks	Pinterest, Instagram, Twitter, etc.
Interaction networks	Phone calls, Messages, Emails, Whatsapp, Snapchat, etc.
Co-authorship networks	Dblp, Science direct, Wikibooks, other scientific databases, etc.
User–user citation networks	Dblp, Science direct, Wikibooks, other scientific databases, etc.
Spread networks	Epidemics, Information, Rumors, etc.
Co-actor networks	IMDB, etc.

brain activity. Other brain imaging modalities include positron emission tomography (PET), electroencephalography (EEG), and electrocorticography (ECoG), to name just a few. Most state-of-the-art tools for inference of brain connectivity leverage variants of causal and correlational analysis methods applied to time-series obtained from the imaging modalities. Brain networks, such as neural networks, can be also a concrete physical network if the nodes are neurons and the links are the connections between neurons [2].

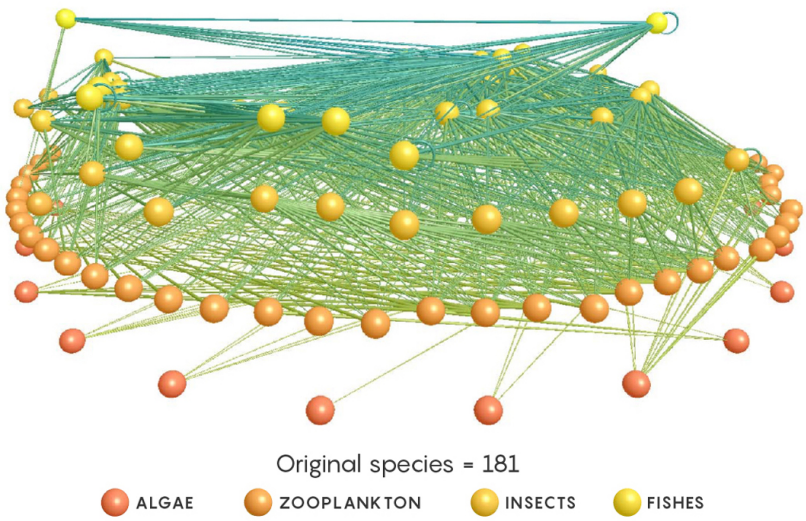


Figure 1.3. The food web of little Rock Lake, Wisconsin.

Fig. 1.3 depicts another type of biological network, the ecological network where vertices are species and links are predator-prey relationships between them. In particular, the presence of the edge says if one species eats the other. The analysis of such a network helps to understand and quantify ecological phenomena, concerning in particular energy and carbon flow in the ecosystem. The edges in this network are asymmetric and are conventionally directed from the prey to the predator, indicating the direction of the flow of energy when the prey is eaten. Other biological networks that we should mention are biochemical networks, such as metabolic networks,

protein-protein interaction networks, and genetic regulatory networks.

Understanding the structure and characteristics of vehicles flow within a road network is useful for planning the construction of connections (roads and highways) linking sites, as well as to forecast the vehicular flow for traffic control. Mathematical modeling and graph-based tools are used to monitor traffic flow and jam problems to recommend solutions for the urban and inter-city vehicular mobility.

Graph-based tools are particularly useful also in wireless sensor networks, present in many applications spanning from environmental monitoring, as a tool to control physical parameters such as temperature, vibration, pressure, or pollutant concentration, to the monitoring of civil infrastructures, such as roads, bridges, buildings, etc.

1.2 State of the Art

Graph-based learning is a wide area of study applied in diverse sectors. One field of study is network science that focuses on graph structure analysis. Graph topology determines a structure of influences among nodes of a network. When processes act on the network, the role of the nodes in these processes is different depending on the position of the nodes. Vertices with very few connections, e.g. pending nodes (1-degree nodes), have limited impact on the dynamics of the network whereas central nodes, e.g. hubs, have a major effect on the behavior of the whole graph. Different works studied methodologies to identify the most important nodes, to explain the network's dynamics, such as the distribution of power in exchange networks [7] or migration in biological networks [8], as well as in designing optimal ways to externally influence the network, e.g., attack vulnerability of networks [9]. Node centrality measures are tools designed to identify such important agents. However, node importance can be interpreted in various ways, giving rise to multiple coexisting centrality measures, the most common ones are degree [10], [11], closeness [12], [13], eigenvector [14], and betweenness [15] centrality. Network science has connections to graph signal processing (GSP) due to graph spectra that GSP builds upon, which is strongly related to the structure of the graph [16, 17]. Spectral graph theory has been demonstrated to be a very powerful tool for data information extraction [16, 18]. The eigenvalues/eigenvectors of the Laplacian matrix of the graph have been used, e.g., to estimate the connectivity of the network [18], to find densely connected clusters of nodes [19], and to search for potential links that would greatly improve the connectivity if they were established [20]. Works in GSP attempt to extend the classical discrete signal processing (DSP) theory from the classical time signals or images to signals defined over the vertices of a graph by introducing the basic

concepts of graph-based filtering [4, 21], graph-based transforms [22], sampling and uncertainty principle [23, 24]. A central role in GSP is played by the graph signal spectral analysis, due to the introduction of the so called Graph Fourier Transform (GFT) [3, 4].

Two major learning tasks of GSP are graph inference and signal recovery. There is a large number of works on network topology identification from nodal observations [6], see recent tutorial works like [25] and [26]. If nodal observations are modeled as random variables or processes, the graph topology typically reflects correlations among nodal signals. However, in many cases looking only at correlations may not capture appropriate causality relations existing among the data. Thus, alternative approaches using the partial correlation [6] or Gaussian graphical models [27], [28] have been studied. GSP has given a strong impulse to find new approaches for the graph topology inference. Some of them make assumptions on the sparsity and/or smoothness of the graph signals [29–31]. Some recent works focus on learning the graph from signals that diffuse over a graph [32–34]. Differently, in [35], authors associate a graph topology to the data to make the observed signals band-limited over the inferred graph. On the other hand, the graph signal inference task emerges in different applications to predict or extrapolate nodal attributes in all the networks given only the attributes of a subset of them. Many preliminary works assume time-invariant nodal signal to reconstruct on a graph , like [3], [36]. Thereafter, reconstruction approaches leverage the notions of graph bandlimitedness [37], [38], sparsity and overcomplete dictionaries [39], smoothness over the graph [40], [36], all of which can be unified as approximations of nonparametric graph functions drawn from a reproducing kernel Hilbert space (RKHS) [41]. In [42], authors jointly reconstruct the signal from partial observation and infer the graph topology, relying on Structural Equation Models.

1.3 Contributions

Most of the literature in GSP and network science consider the graph as perfectly known, except for a few notably works [43–45]. However, there are several situations in which the topology is uncertain, e.g. outages on physical networks or model mismatches in data driven networks. In this thesis, we consider that the graph might be not known perfectly and that, in some cases, the probabilities of links failures are available. We show that graph-learning tasks need to be robustified against the uncertainty of the topology. We analyze some graph-based learning tools under perturbations exploiting in part the small perturbation theory applied to the Laplacian matrix [46]. Following up, we use small perturbation theory to design a

new centrality measure that identifies the most critical edges, i.e. the ones that most influence the way the information flows through the network [47, 48]. Thereafter, to address the signal reconstruction task, we rely on Structural Equation Models (SEMs), and we show the benefit of taking into account possible errors in the topology when this is uncertain. Moreover, the literature on SEM for topology identification does not consider perturbations on the observed signal, but only additive noise. This may happen because of the presence of outliers, adversaries or observation inaccuracy. To fill this gap, in our work we build a total least squares method to cope with errors on the topology and/or on the observed signals [49].

In particular, part of Chapter 5 will solve the topology ID task. It will be clear then how the resulting graph is vulnerable to errors. This is a strong motivation to develop graph learning methodologies, in the Chapter 3, that are robust to graph topology uncertainties, and to identify the critical links of graphs, in Chapter 4.

1.4 Outline

The outline of this thesis is the following.

- **Chapter 1**. The present chapter presents the motivation, outline, and contributions of this thesis.
- **Chapter 2**. In this chapter, we recall basic notions on graph theory, GSP, topology inference and signal reconstruction.
- **Chapter 3**. This chapter presents our work on graph signal processing under topology perturbations. In particular, we expand graph signal processing tools to deal with cases where the graph topology is not perfectly known. supposing that the uncertainty affects only a limited number of edges, we make use of small perturbation analysis to derive closed form expressions instrumental to propose signal processing algorithms that are resilient to imperfect knowledge of the graph topology. Then, we formulate a Bayesian approach to estimate the presence of uncertain edges based only on the observed data and on the data statistics. Finally, we analyze clustering and semi-supervised learning algorithms under topology perturbations. Along with the chapter, numerical tests show the benefits of our perturbation-aware methods.

  The main results of this chapter are also presented in the following papers:

  - E. Ceci, S. Barbarossa, "Graph signal processing with perturbations," IEEE Trans. on Sig. Processing, to appear. [50]

  - E. Ceci, S. Barbarossa, "Robust Graph Signal Processing in the Presence of Uncertainties on Graph Topology," IEEE SPAWC Conf., June 2018.
  - E. Ceci, S. Barbarossa, "Small Perturbation Analysis of Network Topologies," IEEE ICASSP Conf., Apr. 2018.

- **Chapter 4**. In this chapter, we introduce a new centrality measure, named *perturbation topology* measure, to rank the network links according to the impact of their possible failure. By numerical results, we will see that it identifies the inter cluster edges that disconnect or alter critically the connectivity and/or the clustering property of the network.

  The results of this chapter are presented also in part of the following work:

  - S. Barbarossa, S. Sardellitti, E. Ceci, and M. Merluzzi, "The edge cloud: A holistic view of communication, computation and caching," in Cooperative and Graph Signal Processing: Principles and Applications (P.M. Djuric and C. Richard, Eds.), Amsterdam, Netherlands: Elsevier, 2018.

- **Chapter 5**. In this chapter, we present our work on graph-based learning under perturbation via total least-squares. We investigate two major graph-based learning tasks, such as topology identification and inference of signals over graphs relying on SEM. To cope with perturbations, this work introduces a regularized total least-squares (TLS) approach and iterative algorithms with convergence guarantees to solve both tasks. Further generalizations are also considered relying on structured and/or weighted TLS when extra prior information on the perturbation is available. Analyses with simulated and real data corroborate the effectiveness of the novel TLS-based approaches.

  The main results of this chapter are presented also in the following papers:

  - E. Ceci, Y. Shen, and G. B. Giannakis, S. Barbarossa, "Graph-based learning under perturbations via TLS," IEEE Trans. on Sig. Processing, to appear. [51]
  - E. Ceci, Y. Shen, G. B. Giannakis, S. Barbarossa, "Signal and Graph Perturbations via Total Least-Squares," Asilomar Conf. , Oct. 2018.

- **Chapter 6**. Finally, we conclude the thesis summarizing the main obtained results.

Research contributions not presented in this thesis

- P. Di Lorenzo, E. Ceci,"Online recovery of time-varying signals defined over dynamic graphs," IEEE EUSIPCO Conf. , Sept. 2018.
- S. Barbarossa, S. Sardellitti, E. Ceci, "Learning from signals defined over simplicial complexes," IEEE DSW Conf., June 2018.
- S. Barbarossa, E. Ceci, M. Merluzzi,"Overbooking radio/computation resources to meet strict latency constraint in mmW Mobile Edge Computing robust to channel intermittency," EuCNC Conf., June 2017.
- S. Barbarossa, E. Ceci, M. Merluzzi, E. Calvanese-Strinati, "Enabling effective mobile edge computing using millimeter wave links," IEEE ICC Conf., 2017.
- K. Sakaguchi, T. Haustein, S. Barbarossa, E. Calvanese-Strinati, A. Clemente, G. Destino, A. Pärssinen, I. Kim, H. Chung, J. Kim, W. Keusgen, R. J. Weiler, K. Takinami, E. Ceci, A. Sadri, L. Xain, A. Maltsev, G. K. Tran, H. Ogawa, K. Mahler, R. W. Heath Jr, "Where, When and How mmWave is Used in 5G and Beyond", IEICE Transactions on Electronics, Vol. E100.C (2017), Oct. 2017, pp. 790-808.

Notation. Along the thesis, bold lowercase (uppercase) letter will denote column vector, e.g. $\boldsymbol{a}$ (matrix, e.g. $\boldsymbol{A}$), and $a(k)$ (or $[\boldsymbol{A}]_{ij}$) denotes its k-th entry (or the entry at the i-th row and j-th column of matrix $\boldsymbol{A}$). Operators $(\cdot)^\top$, $\mathbb{E}\{\cdot\}$,vec$(\cdot)$, and $\otimes$, will stand for matrix or vector transposition, expectation of random variable (r.v.), column-wise matrix vectorization, and Kronecker product. We will use $|\boldsymbol{a}|$ to denote the absolute value of vector $\boldsymbol{a}$ and $|\boldsymbol{A}|$ as determinant of $\boldsymbol{A}$. The identity matrix of dimension $K \times K$ will denote by $\boldsymbol{I}_K$, and the i-th canonical vector by $\boldsymbol{s}_i$; while diag$(\cdot)$, and bdiag$(\cdot)$ correspondingly represent a diagonal matrix and a block diagonal matrix of its arguments. A set of elements is denoted by a calligraphic letter (e.g., $\mathcal{S}$), while $\bar{\mathcal{S}}$ and $|S|$ represent the complement and the cardinality (i.e., the number of elements of $\mathcal{S}$) of the set $\mathcal{S}$, respectively. Finally, the ℓ_1, ℓ_2, and Frobenius norms will be denoted by $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_F$, respectively.

Chapter 2

Mathematical background

2.1 Graph Theory

In this section, for the reader's convenience we will recall the notation and basic results of graph theory that will be used along the thesis.

Definition 1. *A graph $\mathcal{G}$ is collection of nodes (or vertices) $\mathcal{V}$ along with a set of $\mathcal{E}$ of edges linking pairs of nodes. In particular, a graph composed of N nodes represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, N\}$, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$.*

Definition 2. *A graph is undirected, if the set of relations (edges), $\mathcal{E}$, is symmetric, that is, if $(u, v) \in \mathcal{E}$ implies $(v, u) \in \mathcal{E}$ for all $u, v \in \mathcal{V}$.*

Definition 3. *A directed graph, also called digraph for short, is a network in which each edge has a direction, pointing from one vertex to another.*

There are a number of different ways to mathematically represent a graph. One representation is by using the *adjacency* matrix $\boldsymbol{W} \in \mathbb{R}^{N \times N}$, such that the (i, j)-th element is

$$[\boldsymbol{W}]_{ij} = \begin{cases} w_{ij} & \text{if there is an edge from } j \text{ to } i \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where w_{ij} is the weight of the link between i and j. The graph is said *unweighted* if the weights are equal for every nonzero entry. For a directed graph, for each node is defined an *in degree* $d_i^{in} = \sum_{j=1}^{N} w_{ij}$ and an *out degree* $d_j^{out} = \sum_{i=1}^{N} w_{ij}$. The degree matrix is defined as a $N \times N$ diagonal matrix $\boldsymbol{D} = \mathrm{diag}(d_1^{in}, \ldots, d_N^{in})$. For an undirected graph, the matrix $\boldsymbol{W}$ is symmetric, i.e. $w_{ij} = w_{ji}$, hence $d_j^{out} = d_i^{in}$.

If we label each edge with number from 1 to E, with E the number of edges, an alternative representation of the graph is given by the *incidence* matrix $\boldsymbol{B} \in \mathbb{R}^{N \times E}$

such as

$$[\boldsymbol{B}]_{ij} = \begin{cases} -1 & \text{if the edge } j \text{ leaves node } i; \\ 1 & \text{if the edge } j \text{ enters node } i; \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

Using matrices $\boldsymbol{D}$ and $\boldsymbol{W}$, the Laplacian matrix is defined as $\mathbf{L} = \boldsymbol{D} - \boldsymbol{W}$. Such a matrix is an alternative representation of the graph, commonly used due to its properties. Form the definition of $\mathbf{L}$, it holds that $\mathbf{L}\mathbf{1} = \mathbf{0}$, where $\mathbf{1}$(or $\mathbf{0}$) is the vector with all entries equal to 1(or 0). Then, it is clear that the null space of $\mathbf{L}$ (null($\mathbf{L}$)) contains the all-one vector and the dimension of null($\mathbf{L}$) is at least one. It can be proved that the dimension of the null($\mathbf{L}$) (dim(null($\mathbf{L}$))) is liked to the connectivity of the network [52]. For undirected graphs, $\mathbf{L}$ is symmetric and semidefinite positive, and it can be written as $\mathbf{L} = \boldsymbol{B}\boldsymbol{B}^\top$, thus dim(null($\mathbf{L}$))=dim(null($\boldsymbol{B}$)).

A graph is *connected* if there is a path from every vertex to every other vertex of the graph. In this case, dim(null($\mathbf{L}$))= 1. On the contrary, if the graph is composed of c disconnected components, then dim(null($\mathbf{L}$))= c.

For connected graphs, the second smallest eigenvalue has an important role to measure the connectivity of the graph, for this reason is also called algebraic connectivity [2]. In fact, if $\mathcal{G}_1$ is a subset of a graph $\mathcal{G}$ with the same nodes but a subset of edges then [53]

$$\lambda_2(\mathcal{G}_1) \leq \lambda_2(\mathcal{G}) \tag{2.3}$$

Thus, decreasing the number of edges in a graph, λ_2 also decreases, that is, decreasing the connectivity of $\mathcal{G}$, λ_2 decreases. The algebraic connectivity is also linked to the graph conductance. Such a parameter measures how well knit is the network.

Definition 4. *If $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is a graph of N vertices and $\mathcal{S}$ a subset of $\mathcal{V}$, i.e. $\mathcal{S} \subseteq \mathcal{V}$, let $\partial\mathcal{S}$ denote the set of edges with one end in $\mathcal{S}$ and the other in $\mathcal{V}\backslash\mathcal{S}$. Then, the conductance $\Phi(\mathcal{G})$ of the graph $\mathcal{G}$ is defined as*

$$\Phi(\mathcal{G}) := \min_{|\mathcal{S}| \leq N/2} \frac{|\partial\mathcal{S}|}{|\mathcal{S}|} \tag{2.4}$$

It can be proved that for any $\mathcal{G}$, it holds [52]

$$\Phi(\mathcal{G}) \geq \lambda_2/2. \tag{2.5}$$

The importance of this bound follows from the fact that computing the conductance of a graph is an NP-hard problem, while λ_2 can be computed in polynomial time [52].

2.2 Graph signal processing

A symmetric matrix can always be written as

$$\mathbf{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\top} \tag{2.6}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_N)$ is the diagonal matrix of eigenvalues of $\mathbf{L}$, and $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N]$ the matrix that collects a full set of orthogonal eigenvectors. Since the Laplacian matrix is semidefinite positive, all its eigenvalues are real and non-negative. The null eigenvalue appears with multiplicity equal to the number c of connected components of the graph [16], and thus, considering connected graphs, the graph Laplacian eigenvalues can be ordered as $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \leq \cdots \leq \lambda_N := \lambda_{max}$. The second eigenvalue λ_2 of $\mathbf{L}$ is non-zero if and only if the graph is connected, i.e. it consists of a single component.

A signal or function $\boldsymbol{f} : \mathcal{V} \to \mathbb{R}$ defined over the vertices of a graph may be represented as a vector $\boldsymbol{f} \in \mathbb{R}^N$, where the i-th component of vector $\boldsymbol{f}$ is the signal value at the i-th vertex $v_i \in \mathcal{V}$. Figure 2.1 depicts a graphical example of graph signal.

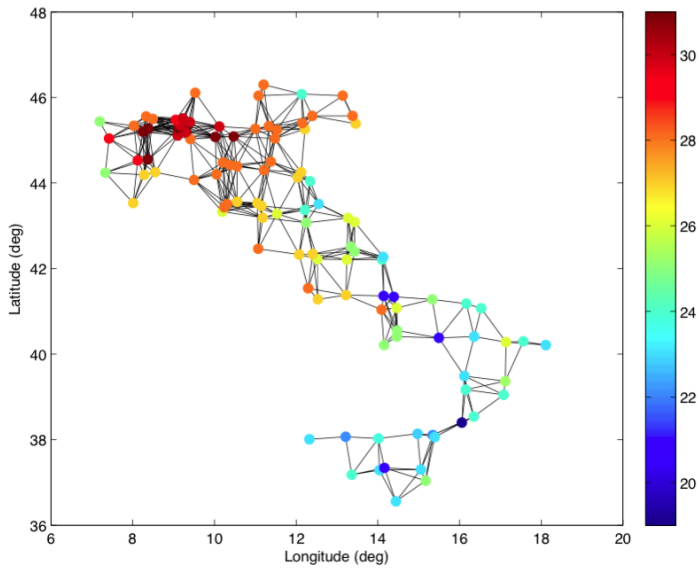


Figure 2.1. Example of graph signal where at each node is associated a temperature value (color of the nodes).

For any signal $\boldsymbol{f} \in \mathbb{R}^N$, we can define its graph Fourier transform (GFT) $\hat{\boldsymbol{f}}$ as

the expansion of $\boldsymbol{f}$ in terms of the eigenvectors of the graph Laplacian:

$$\hat{\boldsymbol{f}} = \boldsymbol{U}^\top \boldsymbol{f} \tag{2.7}$$

where the columns $\{\boldsymbol{u}_i\}_{i=1}^N$ of $\boldsymbol{U}$ are interpreted as the graph Fourier basis and $\{\hat{f}_i\}_{i=1}^N$ are the corresponding graph signal frequency coefficients. The inverse GFT is then defined as

$$\boldsymbol{f} = \boldsymbol{U}\hat{\boldsymbol{f}} = \sum_{i=1}^{N} \hat{f}_i \boldsymbol{u}_i \tag{2.8}$$

In classical Fourier analysis, the eigenvalues carry a specific notion of frequency: the eigenvalues close to zero (low frequencies) are associated to complex exponential eigenfunctions that are smooth, that is, they are slowly oscillating functions, whereas for eigenvalues far from zero (high frequencies), the associated complex exponential eigenfunctions oscillate much more rapidly. For graphs, the Laplacian eigenvalues and eigenvectors have a similar behavior. In particular, for connected graphs, the Laplacian eigenvector $\boldsymbol{u}_1$ associated with $\lambda_1 = 0$ is constant at each vertex. The graph Laplacian eigenvectors associated with low frequencies have values that vary slowly within the clusters present in the graph. The eigenvectors associated with larger eigenvalues oscillate more rapidly and, differently from Fourier analysis of time series or images, they may be highly concentrated, especially when the graph topology departs from a regular topology.

In several cases, the graph signal exhibits clustering features, i.e., it may have similar values within a cluster and vary arbitrarily from one cluster to the other. In such a case, if the columns of $\boldsymbol{U}$ are chosen to represent clusters, the only nonzero (or approximately nonzero) entries of $\hat{\boldsymbol{f}}$ are the ones associated with the clusters and the signal is said band-limited, as we will see later.

For the symmetric, positive semi-definite matrix $\mathbf{L}$ and a signal $\boldsymbol{f} \in \mathbb{R}^N$, it holds the following sum-of-squares property

$$\boldsymbol{f}^\top \mathbf{L} \boldsymbol{f} = \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} (f_i - f_j)^2 \tag{2.9}$$

that is also known as the total variation of the graph signal [3].

A graph signal is smooth if signal samples at neighbouring nodes are similar, and the total variation is a measure of smoothness. Several examples of smooth graph signals include natural images [29], average annual temperatures collected by meteorological stations [31], and product ratings supported over similarity graphs of items or consumers [54], to name a few.

Returning to the graph Laplacian eigenvalue/eigenvector pairs, we can see them

as successive minimizer of the Rayleigh-Ritz ratio, i.e. as solutions of [3]

$$\begin{aligned} \boldsymbol{u}_i = \arg\min_{\boldsymbol{x}} \frac{\boldsymbol{x}^\top \mathbf{L} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \\ \text{s. to } \boldsymbol{x} \perp \boldsymbol{u}_{i'}, i' = 1, \ldots, i-1 \\ \|\boldsymbol{x}\| = 1 \end{aligned} \tag{2.10}$$

and $\lambda_i = \boldsymbol{u}_i^\top \mathbf{L} \boldsymbol{u}_i$, when $\boldsymbol{u}_i$ is of unit length. The total variation in the objective of (2.10) suggests the notion of frequency of the Laplacian matrix mentioned before. In fact, GFT provides an orthogonal basis with increased variation, and such that each additional basis vector minimizes the increase in variation while guaranteeing orthogonality. In summary, the graph connectivity is encoded in $\mathbf{L}$, which is used to define both a graph Fourier transform using $\boldsymbol{U}$ and a notion of signal variation.

We define now a band-limited signal, which is a signal that is smooth within each cluster, while it can assume arbitrary values over different clusters. Fig. 2.2 depicts a real graph signal representing the density of vehicles in a street map of Rome (piazza Mazzini) and its GFT, we can see that in the frequency domain it is approximately band-limited.

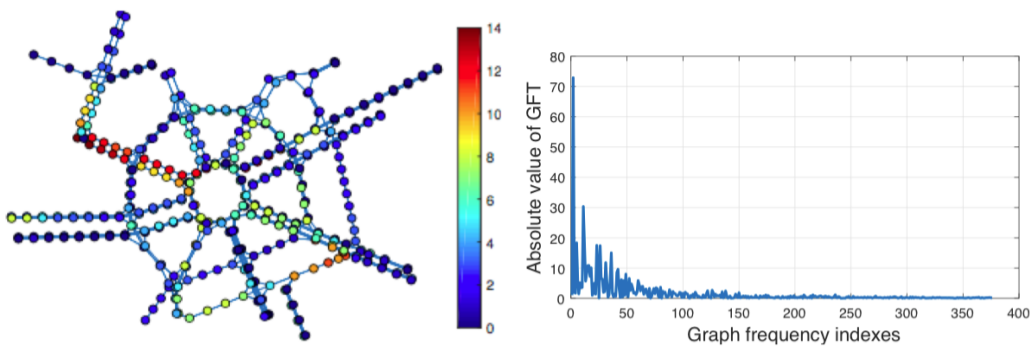


Figure 2.2. Graph signal of vehicles density in a street map of Rome (piazza Mazzini) with its GFT.

Let us define the matrix $\boldsymbol{U}_K \in \mathbb{R}^{N \times K}$ as the collection of the K columns of $\boldsymbol{U}$ associated with a set of indices $\mathcal{K}$. Then, we introduce an $N \times N$ band limiting operator

$$\boldsymbol{P}_{\mathcal{K}} = \boldsymbol{U}_K \boldsymbol{U}_K^\top. \tag{2.11}$$

The matrix $\boldsymbol{P}_{\mathcal{K}}$ projects a vector $\boldsymbol{f}$ onto the subspace spanned by the column of $\boldsymbol{U}_K$.

Therefore, the signal $\boldsymbol{f}$ is perfectly localized in a frequency set of K dimension if

$$\boldsymbol{P}_{\mathcal{K}}\boldsymbol{f} = \boldsymbol{f} = \boldsymbol{U}_K\boldsymbol{s} \tag{2.12}$$

where $\boldsymbol{s}$ is an $K \times 1$ vector such that $[\boldsymbol{s}^\top, \boldsymbol{0}_{N-K}^\top]^\top = \hat{\boldsymbol{f}}$, and $\boldsymbol{0}_k$ is the $k \times 1$ vector of zero entries.

Similarly, if $\mathcal{S}$ is a subset of $\mathcal{V}$, i.e. $\mathcal{S} \subseteq \mathcal{V}$, we define a $N \times 1$ vector $\boldsymbol{d}_{\mathcal{S}}$ where the i-th entry is equal to 1 if $v_i \in \mathcal{S}$ and 0 otherwise, thus the $N \times N$ vertex-limiting operator is defined as

$$\boldsymbol{D}_{\mathcal{S}} = \mathrm{diag}(\boldsymbol{d}_{\mathcal{S}}). \tag{2.13}$$

The signal is perfectly localized on the vertex domain if $\mathcal{D}_{\mathcal{S}}\boldsymbol{f} = \boldsymbol{f}$. Denoting $\boldsymbol{P}_{\mathcal{K}}$ and $\boldsymbol{D}_{\mathcal{S}}$ the set of all $\mathcal{K}$-bandlimited and $\mathcal{S}$-vertex-limited signals, respectively, operators $\boldsymbol{P}_{\mathcal{K}}$ and $\boldsymbol{D}_{\mathcal{S}}$ are self-adjoint and idempotent and they represent orthogonal projectors into $\boldsymbol{P}_{\mathcal{K}}$ and $\boldsymbol{D}_{\mathcal{S}}$, respectively. Differently from the classical continuous-time signals, a graph signal can be localized perfectly both in the vertex domain and the frequency domain under certain conditions [24].

Theorem 1. *A graph $\boldsymbol{f}$ is perfectly localized over both vertex set $\mathcal{S}$ and frequency set $\mathcal{K}$, i.e. $\boldsymbol{f} \in \mathcal{D}_{\mathcal{S}} \cap \mathcal{P}_{\mathcal{K}}$, if and only if the operator $\boldsymbol{P}_{\mathcal{K}}\boldsymbol{D}_{\mathcal{S}}\boldsymbol{P}_{\mathcal{K}}$ has an eigenvalue equal to one; in such a case, $\boldsymbol{f}$ is the eigenvector of $\boldsymbol{P}_{\mathcal{K}}\boldsymbol{D}_{\mathcal{S}}\boldsymbol{P}_{\mathcal{K}}$ associated with the unit eigenvalue [24].*

2.3 Semi-supervised and Unsupervised learning

In this section, we will recall the well-known semi-supervised learning problem of label propagation and the unsupervised learning technique of graph-based clustering, that will be useful later in the thesis.

In supervised learning, the aim is to infer a function from labeled training data that will be used for mapping new examples. Each training data is a pair consisting of an input object $\boldsymbol{x}$ and a desired output value y. Unsupervised learning, on the other hand, aims at finding patterns from data without existing labels. Among the unsupervised learning tasks, clustering is the most relevant one. Semi-supervised learning is somewhere in between unsupervised and supervised. For example semi-supervised classification have the training data consisting of both l labeled and u unlabeled instances, such as to be better than the supervised classifier that train the labeled data alone.

Graph-based semi-supervised learning are based on the construction of a graph from the training data. If the training data are $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, $\{\boldsymbol{x}_j\}_{j=l+1}^{l+u}$, the vertices of the graph are the labeled and the unlabeled instances $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l} \cup \{\boldsymbol{x}_j\}_{j=l+1}^{l+u}$.

Given the graph, the goal is to assign a label to every unlabeled instances exploiting the information on the labeled instances (label propagation). A way to accomplish this task is to solve the following problem

$$\begin{aligned} &\min_{\boldsymbol{f}\in\mathbb{R}^{l+u}} \sum_{i,j=1}^{l+u} w_{ij}(f(\boldsymbol{x}_i)-f(\boldsymbol{x}_j))^2 = \boldsymbol{f}^\top \mathbf{L}\boldsymbol{f} \\ &\text{s. to } f(\boldsymbol{x}_i)=y_i,\ i=1,\dots,l \end{aligned} \tag{2.14}$$

where $\boldsymbol{f} := [f(\boldsymbol{x}_1),\dots,f(\boldsymbol{x}_{u+l})]^\top$. Problem (2.14) is convex and a closed-form unique solution can be found [55]. In particular, writing $\mathbf{L}$ as $\mathbf{L} = \begin{pmatrix} \mathbf{L}_{uu} & \mathbf{L}_{ul} \\ \mathbf{L}_{lu} & \mathbf{L}_{ll} \end{pmatrix}$ the solution is

$$\boldsymbol{f} = -\mathbf{L}_{uu}^{-1}\mathbf{L}_{ul}\boldsymbol{f}_l \tag{2.15}$$

where $\boldsymbol{f}_l$ denotes the vector with the l known labels. Moreover, since $\boldsymbol{f}\in\mathbb{R}^{l+u}$, the values of this function have to be compared with a threshold to produce discrete labels.

Let us recall now the graph-based clustering problem. The goal is to split the vertices of a graph in disjoint sets of nodes well connected within the sets and weakly connected among clusters. Given a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, let $\mathcal{V}_1\subset\mathcal{V}$ and $\mathcal{V}_2\subset\mathcal{V}$ be two disjoint and complementary subsets of vertices of the graph, such that $\mathcal{V}_1\cup\mathcal{V}_2=\mathcal{V}$ and $\mathcal{V}_1\cap\mathcal{V}_2=\emptyset$, where $\emptyset$ is the empty set. Clustering methods aim to find the disjoint sets, (also known as) a.k.a. clusters, $\mathcal{V}_1$ and $\mathcal{V}_2$, such that there are few edges that connect the two sets, while there are many edges joining the vertices within the clusters.

Let us define $\bar{\mathcal{V}}_1$ the complement set of $\mathcal{V}_1$, and the value $\tilde{W}(\mathcal{V}_1,\bar{\mathcal{V}}_1) := \sum_{i\in\mathcal{V}_1, j\in\bar{\mathcal{V}}_1} w_{ij}$. Hence, for a given number k of subsets, the mincut approach simply consists in choosing a partition $\mathcal{V}_1,\dots,\mathcal{V}_k$ that minimizes [18]

$$\text{ratiocut}(\mathcal{V}_1,\dots,\mathcal{V}_k) = \sum_{i=1}^{k} \frac{\tilde{W}(\mathcal{V}_i,\bar{\mathcal{V}}_i)}{|\mathcal{V}_i|} \tag{2.16}$$

where $|\mathcal{V}_i|$ denotes the cardinality of $\mathcal{V}_i$. Note that, the cut in (2.16) assumes small values when clusters $\mathcal{V}_i$ are not too small. Thus, with this normalization the goal is to find clusters that are "balanced", as measured by the number of vertices, avoiding the trivial solution of splitting one vertex from the rest of the graph. Unfortunately, this problem is NP hard, [56]. Spectral clustering is a way to solve relaxed versions of this problem [18]. Let us consider for simplicity the case $k=2$, such that the

clustering aims at solving the problem

$$\min_{\mathcal{V}_1 \subset \mathcal{V}} \text{ratiocut}(\mathcal{V}_1, \bar{\mathcal{V}}_1). \tag{2.17}$$

Defining

$$h_i = \begin{cases} \sqrt{|\bar{\mathcal{V}}_1|/|\mathcal{V}_1|}, \text{if } v_i \in \mathcal{V}_1 \\ -\sqrt{|\mathcal{V}_1|/|\bar{\mathcal{V}}_1|}, \text{if } v_i \in \bar{\mathcal{V}}_1 \end{cases} \tag{2.18}$$

and collecting the h_i in the $N \times 1$ vector $\mathbf{h} := [h_1, \ldots, h_N]^\top$, it holds

- $\mathbf{h}^\top \mathbf{L}\mathbf{h} = |\mathcal{V}|\text{ratiocut}(\mathcal{V}_1, \bar{\mathcal{V}}_1)$.
- $\mathbf{h}$ is orthogonal to the constant vector $\mathbf{1}$.
- $\|\mathbf{h}\|^2 = N$.

Hence, problem (2.17) can be equivalently rewritten as

$$\begin{aligned} &\min_{\mathcal{V}_1 \subset \mathcal{V}} \mathbf{h}^\top \mathbf{L}\mathbf{h} \\ &\text{s.to } \mathbf{h}^\top \mathbf{1} = 0 \\ &\qquad \|\mathbf{h}\| = \sqrt{N} \\ &\qquad \text{with } h_i \text{ as in eq. (2.18)} \end{aligned} \tag{2.19}$$

Unfortunatly, this problem is NP-hard. To cope with this, we can relax allowing h_i to assume arbitrary values in $\mathbb{R}$: By the Rayleigh-Ritz theorem (cf. (2.10)), it is easy to see that the solution of this problem is given by the vector $\mathbf{h}$ equal to the eigenvector associated with the second smallest eigenvalue of $\mathbf{L}$. However, we need a discrete indicator vector to obtain a partition of the graph that can be choose as

$$\begin{cases} v_i \in \mathcal{V}_1, \text{ if } h_i \geq 0 \\ v_i \in \bar{\mathcal{V}}_1, \text{ if } h_i < 0. \end{cases} \tag{2.20}$$

For $k > 2$ the relaxation of the ratiocut problem follows similar principle as the one above, and it can be proved that its solution is revealed by the Rayleigh-Ritz theorem (cf. (2.10)) giving a matrix which contains the first k eigenvectors of $\mathbf{L}$ as columns. The standard way to reconvert the real valued solution matrix to a discrete partition is to use the k-means algorithms on the rows of eigevector matrix [17].

2.4 Maximum flow

When we know the structure of the network we can calculate measures that capture important features of the network topology. For the later analysis, among

several centrality measures, we recall the *edge betweenness* centrality [57]. Edge betweenness of an edge is defined as the number of shortest paths between pairs of vertices that run along it.

If there is more than one shortest path between a pair of vertices, each path is given equal weight such that the total weight of all of the paths is unity. If a graph is composed of communities or groups that are only weakly connected between one another by a few links, then all shortest paths between different communities necessarily pass through one of these few edges. Clearly, inter communities edges will have an high value of edge betweenness. If we remove all these edges, the graph will be disconnected. Thus, the definition of betweenness is based on the idea of maximum flow. The *max flow/min cut* theorem says that the maximum flow between two vertices is always equal to the size of the minimum cut set. In other words how well the information flows through the network depends on how well connected the network is.

2.5 Graph topology inference

Graph topology inference is a prominent problem in Network Science [6]. The goal is to infer the graph topology to be associated with a set of signal, to capture their correlations. Each signal may be associated with a vertex of a graph and the goal is to find the edges of the graph. The basic idea is that the presence of an edge captures the similarity between the signals associated with the endpoints vertices of that edge. Several topology inference approaches build the graph such that their edge weights correspond to nontrivial correlations or coherence measures between signal profiles at incident nodes.

Consider the network $\mathcal{G}(\mathcal{V}, \mathcal{E})$, described by a matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$, having nonzero (i, j)-th entry, denoted as a_{ij}, only if a directed edge is present from node j to node i. In general $a_{ij} \neq a_{ji}$ for directed networks. Conversely, if the network is undirected, so $a_{ij} = a_{ji}$ is always true. Let us suppose here that the network represents an abstraction of a complex system whose pairwise relations are not directly observable. What is known is a set of signal y_{it} at node i and time t.

The topology identification task consists in finding the matrix $\boldsymbol{A}$ from data and it is solved in Chapter 5 in the case of signal perturbations. The aim of this section is to recall several topology identification methods to give the reader a broader view of the existing literature and introduce in the end the topology ID approach that relies on structural equation models. SEM is the model at the basis of the analysis in Chapter 5 not only for the topology identification, but also for the signal recovery task in presence of graph perturbations.

2.5.1 Statistical methods

The natural starting point when inferring a graph from data is the association of edges with correlations or coherence measures between signal profiles at incident nodes. Such methods rely on ad hoc thresholding of user-defined edgewise score functions. Such thresholds are often tuned manually and are application-dependent. However, under some conditions like e.g. the case of Gaussian graphical models in which the edge weights represent the partial correlation coefficients among data, it can be shown that the graph is correctly inferred with a given probability by selecting an optimal threshold, when the number of data, say N, is large [58].

A common metric quantifying similarity between nodal random variables y_i and y_j, expressed by a_{ij}, is the Pearson correlation coefficient. Given the covariance matrix $\boldsymbol{\Sigma} := \mathbb{E}[(\boldsymbol{y}-\boldsymbol{\mu})(\boldsymbol{y}-\boldsymbol{\mu})^\top]$ of the random graph signal $\boldsymbol{y} = [y_1, \ldots, y_N]$, with mean $\boldsymbol{\mu} := \mathbb{E}[\boldsymbol{y}]$ and be the entries $\boldsymbol{\Sigma} := \sigma_{ij} =cov(y_i, y_j)$, the Pearson correlation coefficient is defined as

$$\rho_{ij} := \frac{\text{cov}(y_i, y_j)}{\sqrt{\text{var}(y_i)\text{var}(y_j)}}. \tag{2.21}$$

Given independent realizations $\mathcal{Y} = \{\boldsymbol{y}_y\}_{t=1}^T$ of $\boldsymbol{y}$, we can compute the empirical correlations $\hat{\rho}_{ij}$ by using the sample covariance matrix $\hat{\boldsymbol{\Sigma}}$. Selecting a threshold T_{fa} to guarantee a prescribed significance level, i.e. a false alarm probability P_{FA}, we compare such coefficient with the threshold $|\hat{\rho}_{ij}| > T_{fa}$. The comparison is used to assert that an edge exist with a strength $a_{ij} = \hat{\rho}_{ij}$ between nodes i and j; see, e.g., [6].

However, $\rho_{ij} = \rho_{ji}$ by definition and it implies that this coefficient cannot reveal the directionality of the edges. In addition, if we consider for instance a three-node toy network $i \to k \to j$, where nodes i and j are mediated through node k, this mediation would imply correlation of variables at nodes i and j based on ρ_{ij}. Hence, correlation-based connectivity can incorrectly declare presence of an (i, j) edge. To cope with this, one can use *partial* correlation coefficients. The partial correlation coefficient between node i and j is defined as

$$\tilde{\rho}_{ij} := \frac{\text{cov}(y_i, y_j|\mathcal{V}\backslash ij)}{\sqrt{\text{var}(y_i|\mathcal{V}\backslash ij)\text{var}(y_j|\mathcal{V}|\backslash ij)}} \tag{2.22}$$

where $\mathcal{V}\backslash ij$ denotes the set of $N-2$ variables $\{y_k\}$ excluding the ones indexed by nodes i and j. Then, given $\mathcal{Y} = \{\boldsymbol{y}_t\}_{t=1}^T$ independent realizations of $\boldsymbol{y}$, one can compare the absolute value of the empirical partial correlation coefficient $\hat{\tilde{\rho}}_{ij}$ with a threshold T_{fa} to infer the nonzero partial correlations. If the signal $\boldsymbol{y}$ is a Gaussian random vector, then $\tilde{\rho}_{ij} = 0$ means that y_i and y_j are conditionally independent given all of the other variables in the set $\mathcal{V}\backslash ij$. The partial correlation network with

edges $\mathcal{E} := \{(i,j) \in \mathcal{V} \times \mathcal{V} : \tilde{\rho}_{ij} \neq 0\}$ is known as Gaussian Markov random field (GMRF).

Moreover, denoting with $\boldsymbol{P} = \boldsymbol{\Sigma}^{-1}$ the inverse of the covariance matrix of $\boldsymbol{y}$, namely the precision matrix, one can infer a GMRF by using the following expression of partial correlation coefficient between i and j [6]

$$\tilde{\rho}_{ij} = \frac{-[\boldsymbol{P}]_{ij}}{\sqrt{[\boldsymbol{P}]_{ii}[\boldsymbol{P}]_{jj}}}. \tag{2.23}$$

This link among linear partial correlation coefficients, conditional uncorrelatedness of nodal variables (or independence in the Gaussian case), and (non)zero entries of $\boldsymbol{P}$ is at the basis of the graphical Lasso approach to topology identification. Consider T realizations $\mathcal{Y} = \{\boldsymbol{y}_t\}_{t=1}^T$ from a multivariate Gaussian distribution with zero mean and positive covariance matrix $\boldsymbol{\Sigma}$. The task of graphical lasso is to estimate the unknown $\boldsymbol{P}$ based of the T samples. The problem is challenging especially when $N \gg T$, when the ordinary maximum likelihood estimate (MLE) does not exist. Even if it does exist, that is, when $T \geq N$, the MLE is often poorly behaved. The graphical lasso method aims at estimating $\boldsymbol{P}$, under the assumption that this matrix is sparse [59]. The graphical lasso problem minimizes a ℓ_1-regularized negative log-likelihood

$$\hat{\boldsymbol{P}} = \arg\max_{\boldsymbol{P} \succeq 0} \{\log\det \boldsymbol{P} - \mathrm{Tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{P}) - \lambda \|\boldsymbol{P}\|_1\} \tag{2.24}$$

where $\hat{\boldsymbol{\Sigma}} = \frac{1}{T}\sum_{t=1}^T \boldsymbol{y}_t \boldsymbol{y}_t^\top$ is the empirical covariance matrix obtained from the data $\mathcal{Y}$, and λ is a tuning parameter controlling the amount of ℓ_1 shrinkage. Although Problem (2.24) is convex, log-determinant problems are usually computationally demanding. A number of approaches to solve (2.24) have been proposed specifically for graphical Lasso, see [27, 60, 61].

An alternative way of building the graph is to learn a neighbourhood for each vertex, that is, to identify the other vertices to which each node is connected. Note that, in the Gaussian setting, where $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{P}^{-1})$, the conditional distribution of $\boldsymbol{y}_i$ given $\boldsymbol{y}_{\setminus i} := [y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_N]^\top$ is also Gaussian. We assume that the observation at a particular vertex may be represented as a function of the observations at the neighbouring nodes. Based on this assumption, the observation at each variable y_i can be approximated as a sparse linear combination of observations at other variables $\boldsymbol{y}_{\setminus i}$ [62]. In fact, the minimum mean square error (MMSE) predictor of y_i based on $\boldsymbol{y}_{\setminus i}$ is $\mathbb{E}[y_i|\boldsymbol{y}_{\setminus i}] = \boldsymbol{y}_{\setminus i}^\top \boldsymbol{\beta}^{(i)}$, where the regression coefficients $\boldsymbol{\beta}^{(i)}$ can be expressed in terms of entries of $\boldsymbol{P}$:

$$\beta_j^{(i)} = -\frac{[\boldsymbol{P}]_{ij}}{[\boldsymbol{P}]_{ii}}. \tag{2.25}$$

Let $\mathcal{Y} = \{\boldsymbol{y}_t\}_{t=1}^T$ be T realizations of a multivariate Gaussian distribution with zero mean and definite positive covariance matrix $\boldsymbol{P}^{-1}$, such that $\boldsymbol{y}_t = [y_{1t}, \dots, y_{Nt}]^\top$. Thus, the neighbourhood-based Lasso regression problem has the form

$$\hat{\boldsymbol{\beta}}^{(i)} = \min_{\boldsymbol{\beta}^{(i)} \in \mathbb{R}^{N-1}} \sum_{t=1}^{T} (y_{it} - \boldsymbol{y}_{t,\setminus i}^\top \boldsymbol{\beta}^{(i)})^2 + \lambda \|\boldsymbol{\beta}^{(i)}\|_1 \tag{2.26}$$

where $\boldsymbol{y}_{t,\setminus i}$ is $\boldsymbol{y}_t$ without y_i. The first term of (2.26) can be interpreted as the negative log-likelihood of $\boldsymbol{\beta}^{(i)}$ and the ℓ_1 penalty enforce sparsity, with λ that balances the two terms. Thus, a connection between two vertices v_i and v_j is established if either $\hat{\beta}_j^{(i)}$ or $\hat{\beta}_i^{(j)}$ is nonzero (OR rule), or both (AND rule).

The neighbourhood-based approach is computationally more appealing with respect to graphical lasso, since the N lasso problems can be solved in parallel. Such a decomposability of neighbourhood-based approach is possible, since in (2.26) the conditional likelihood is per vertex and does not enforce the positive semidefinite constraint $\boldsymbol{P} \succeq 0$.

The concept of undirected conditional independence graphs can be also extended to the multivariate time series. This extension measures the dependence between two time series after removing the linear time invariant effects of a third, or more, series [63, 64]. Suppose $\boldsymbol{y}(t) = [y_1(t), \dots, y_N(t)]^\top$, $t \in \mathbb{Z}$ is a multivariate stationary time series. The graph is built considering that the edge between i and j is missing if $y_i(\cdot)$ and $y_j(\cdot)$ are uncorrelated given the other components of the series. This characterization of the graph can be obtained from the partial spectral coherence. Let $c_{ij}(u) = c_{y_i y_j}(u) = \mathrm{cov}(y_i(t+u), y_j(t))$ be the covariance function of the process. If $\sum_{u=-\infty}^{\infty} |c_{ij}(u)| < \infty$, then the cross-spectrum f_{ij} between $y_i(t)$ and $y_j(t)$ is defined by

$$f_{ij}(\lambda) = \frac{1}{2\pi} \sum_{u=-\infty}^{\infty} c_{ij}(u) e^{-i\lambda u}. \tag{2.27}$$

If the time series $y_i(t+u)$ and $y_j(t)$ are uncorrelated at all lags u, then the cross-spectrum $f_{ij}(\lambda) = 0$, and viceversa. Let us define $x_{\setminus ij}(t) = \{y_z(t) : z \neq i, j\}$. Then, a measure of dependence between $y_i(t)$ and $y_j(t)$ given $y_{\setminus ij}(t)$, is the partial cross-spectrum $f_{y_i y_j | y_{\setminus ij}}(\lambda)$ of $y_i(t)$ and $y_j(t)$ given $y_{\setminus ij}(t)$, whose rescaling leads to the partial spectral coherence

$$R_{y_i y_j | y_{\setminus ij}}(\lambda) = \frac{f_{y_i y_j | y_{\setminus ij}}(\lambda)}{[f_{y_i y_i | y_{\setminus ij}}(\lambda) f_{y_j y_j | y_{\setminus ij}}(\lambda)]^{1/2}}. \tag{2.28}$$

The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is called partial correlation graph when $(i,j) \notin \mathcal{E}$ iff $R_{y_i y_j | y_{\setminus ij}}(\cdot) = 0$ [64].

Interestingly, partial spectral coherence is linked with the inverse of the spectral

matrix [63]. In particular, let $\mathbf{\Theta}(\lambda) := f_{YY}(\lambda)^{-1}$ and

$$Q(\lambda) := \begin{pmatrix} [\mathbf{\Theta}]_{11}^{-1/2} & & 0 \\ & \ddots & \\ 0 & & [\mathbf{\Theta}]_{NN}^{-1/2} \end{pmatrix} \mathbf{\Theta}(\lambda) \begin{pmatrix} [\mathbf{\Theta}]_{11}^{-1/2} & & 0 \\ & \ddots & \\ 0 & & [\mathbf{\Theta}]_{NN}^{-1/2} \end{pmatrix}. \tag{2.29}$$

Hence, if $f_{YY}(\lambda)^{-1}$ is a full rank matrix

$$R_{y_i y_j | y_{\setminus ij}}(\lambda) = -[Q]_{ij}(\lambda), \text{for } i \neq j. \tag{2.30}$$

Its absolute value is then compared with a threshold T_{fa} to find the missing edges (similar to the covariance selection models where the missing edges are characterized by zeros in the inverse covariance matrix). For certain applications it may be interesting to analyze the case where $[Q]_{ij}(\lambda) = 0$ for particular frequency bands, leading to the idea of frequency dependent graph.

2.5.2 Smooth graphs

Several GSP applications aim to associate a graph to the data in such a way that the observed signal admits certain regularity properties, in particular it is smooth over the inferred graph. Given possibly noisy observations $\mathcal{Z} := \{\boldsymbol{z}_t\}_{t=1}^T$, the goal is to infer the graph $\mathcal{G}$ such that the observations are smooth over the graph. Recall that a graph signal is smooth when the values of neighbouring nodes tend to be similar. The smoothness of a graph signal can be quantified making use of the total variation defined as the quadratic form of the Laplacian matrix (cf. (2.9)). Specifically, let $\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{\epsilon}$ be the observed noisy signal of dimension $N \times 1$, with $\boldsymbol{\epsilon}$ the error term. Collecting the T observations in the matrix $\mathbf{Z} = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_T] \in \mathbb{R}^{N \times T}$ and setting $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T] \in \mathbb{R}^{N \times T}$, a topology identification method with graph smoothness prior can be formulated as [30]

$$\begin{aligned} &\min_{\mathbf{L},\mathbf{Y}} \|\mathbf{Z} - \mathbf{Y}\|_F^2 + \alpha \mathrm{Tr}(\mathbf{Y}^\top \mathbf{L} \mathbf{Y}) + \frac{\beta}{2} \|\mathbf{L}\|_F^2 \\ &\text{s.to } \mathrm{Tr}(\mathbf{L}) = N, \ \mathbf{L}\mathbf{1} = \mathbf{0}, \ [\mathbf{L}]_{ij} = [\mathbf{L}]_{ji} \leq 0, \ i \neq j \end{aligned} \tag{2.31}$$

where α and β are tunable regularization parameters. The objective function of (2.31) encourages: (i) data fidelity by using the quadratic loss, penalizing the discrepancies between $\mathbf{Z}$ and $\mathbf{Y}$; (ii) the smoothness of the observed signal over inferred graph through the total variation regularizer, and (iii) $\mathrm{Tr}(\mathbf{L}) = N$ avoids the trivial solution of all-zeros and fixes the ℓ_1 norm of $\mathbf{L}$. Moreover, the Frobenius norm of $\mathbf{L}$ controls the sparsity of the resulting graph. Problem (2.31) is bi-convex and can be solved finding a sub-optimal solution relying on alternating minimization [30].

Denoting with $\odot$ the Hadamard product (entry-wise), an alternative approach of topology ID with smoothness prior is based on the link between smoothness and sparsity revealed by

$$\mathrm{Tr}(\mathbf{Y}^\top \mathbf{L}\mathbf{Y}) = \frac{1}{2}\|\boldsymbol{A} \odot \boldsymbol{H}\|_1 \tag{2.32}$$

where $\boldsymbol{H} \in \mathbb{R}_+^{N\times N}$ is the Euclidean distance matrix with entries $[\boldsymbol{H}]_{ij} := \|\bar{\boldsymbol{y}}_i - \bar{\boldsymbol{y}}_j\|_2^2$, $i, j \in \mathcal{V}$, and $\bar{\boldsymbol{y}}_i \in \mathbb{R}^T$ is the i-th row of $\mathbf{Y}$. In this case, the edge is associated with the pair (i, j) at smaller distance $[\boldsymbol{H}]_{ij}$. Thus, the general learning graph problem formulation become [29]

$$\begin{aligned} \min_{\boldsymbol{A}} \quad & \{\|\boldsymbol{A} \odot \boldsymbol{H}\|_1 + \alpha \mathbf{1}^\top \log(\boldsymbol{A}\mathbf{1}) + \frac{\beta}{2}\|\boldsymbol{A}\|_F^2\} \\ \text{s. to} \quad & \mathrm{diag}(\boldsymbol{A}) = \mathbf{0},\ [\boldsymbol{A}]_{ij} = [\boldsymbol{A}]_{ji} \geq 0,\ i \neq j \end{aligned} \tag{2.33}$$

where α and β are the regularization parameters. The $\log(\boldsymbol{A}\mathbf{1})$ term enforces each vertex to have at least one incident node, while the Frobenius norm regularization over $\boldsymbol{A}$ controls the graph's edge sparsity pattern penalizing larger edge weights. Problem (2.33) is convex and can be solved efficiently with complexity $O(N^2)$.

The topology identification problem of an unweighted graph can be equivalently seen as the identification of the edge set $\mathcal{E}$. Let $L = N(N-1)/2$ be the maximum possible number of edges, and matrix $\boldsymbol{B} := [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_L] \in \mathbb{R}^{N\times L}$ be the incidence matrix of the complete graph on N vertices, where the l-th column $\boldsymbol{b}_l = [b_{l1}, \ldots, b_{lN}]$ is a vector of all zeros but $b_{mi} = 1$ and $b_{mj} = -1$ when i and j connect the link l. Thus, if $\boldsymbol{w} = [w_1, \ldots, w_L]^\top \in \{0, 1\}^L$ is the edge selection vector, the Laplacian matrix can be written as

$$\mathbf{L}(\boldsymbol{\omega}) = \sum_{l=1}^{L} w_l \boldsymbol{b}_l \boldsymbol{b}_l^\top. \tag{2.34}$$

Thus, the problem formulation is [31]

$$\begin{aligned} \min_{\boldsymbol{w}\in\{0,1\}^L} & \mathrm{Tr}(\mathbf{Y}^\top \mathbf{L}(\boldsymbol{w})\mathbf{Y}) \\ \text{s.to } & \|\boldsymbol{w}\|_0 = K \end{aligned} \tag{2.35}$$

where $K \ll N$ is a prescribe number of edges and constraint $\|\boldsymbol{w}\|_0 = K$ forces the sparsity level of $\boldsymbol{w}$ to be equal to K. Such problem is non convex because of the binary nature of $\boldsymbol{w}$, however it can be solved by a simple rank ordering procedure. The solver entails the computation of $\mathrm{Tr}(\mathbf{Y}^\top(\boldsymbol{b}_l\boldsymbol{b}_l^\top)\mathbf{Y})$ for all the candidate edges and setting $w_l = 1$ for the K edges having the smallest scores [31].

2.5.3 Topology ID of network diffusion process structure

Besides global smoothness, alternative signal models also consider signals evolving over the graph, as a result of a diffusion process, which can be represented as a filtering operation over the graph. Such model is appropriate in some real word scenarios to understand information propagation, for instance in geographical spaces, the movement of people in the building or vehicles in the cities, and the shift of the people's interest on social media platform [65, 66].

For this signal model, graph filters and signals may be interpreted as a function $f(\mathcal{G})$ and coefficients collected in a vector $\boldsymbol{c}$, respectively. The function $f(\mathcal{G})$ can be an arbitrary polynomial function of the matrix related to the connectivity of the graph [32, 33], or a diffusion kernel [34]. In this model, the general requirement is that the covariance structure of the observed signal is explained by the unknown network structure. In fact, graph learning algorithms estimate first the eigenvectors of the graph operator (e.g. Laplacian, adjacency) from the sample covariance matrix of the observations, then the eigenvalues are estimated in a second step to obtain the operator [32, 33].

In particular, assuming stationarity and a finite polynomial degree K, the graph signal $\boldsymbol{y}$ can be written as

$$\boldsymbol{y} = \sum_{k=0}^{K} \alpha_k \boldsymbol{S}^k \boldsymbol{c} \tag{2.36}$$

where $\{\alpha_k\}$ is a set of parameter, and $\boldsymbol{S}$ is a general graph operator that encodes the connectivity. Usually, vector $\boldsymbol{c}$ is assumed to be zero-mean with covariance matrix $\boldsymbol{\Sigma}_c = \mathbb{E}[\boldsymbol{c}\boldsymbol{c}^\top]$. If $\boldsymbol{c}$ is also white, then $\boldsymbol{\Sigma}_c = \boldsymbol{I}$ and the model in (2.36) assumes $\boldsymbol{y}$ to be stationary in $\boldsymbol{S}$. In fact, in this case we can write the signal covariance matrix $\boldsymbol{\Sigma}_y$ as

$$\begin{aligned}\boldsymbol{\Sigma}_y = \mathbb{E}[\boldsymbol{y}\boldsymbol{y}^\top] &= \mathbb{E}\left[\sum_{k=0}^{K} \alpha_k \boldsymbol{S}^k \boldsymbol{c} (\sum_{k=0}^{K} \alpha_k \boldsymbol{S}^k \boldsymbol{c})^\top\right] \\ &= \sum_{k=0}^{K} \alpha_k \boldsymbol{S}^k (\sum_{k=0}^{K} \alpha_k \boldsymbol{S}^k)^\top = \boldsymbol{\chi}(\sum_{k=0}^{K} \alpha_k \boldsymbol{\Lambda}^k)^2 \boldsymbol{\chi}^\top \end{aligned} \tag{2.37}$$

where $\boldsymbol{S} = \boldsymbol{\chi}\boldsymbol{\Lambda}\boldsymbol{\chi}^\top$, and the eigenvectors of $\boldsymbol{S}$ are also the eigenvectrs of $\boldsymbol{\Sigma}_y$. Thus, given sufficient number of graph signals, the eigenvectors of $\boldsymbol{S}$ can be approximated with the ones of the sample covariance matrix. In a second step, the operator $\boldsymbol{S}$ is

recovered from its eigenvalues as followed

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{\Psi}} f(\mathbf{S},\mathbf{\Psi}) \\ &\text{s. to } \boldsymbol{S} = \boldsymbol{\chi}\mathbf{\Psi}\boldsymbol{\chi}^\top,\ \boldsymbol{S} \in \mathcal{S} \end{aligned} \tag{2.38}$$

where $f(\cdot)$ is a convex function that imposed desired properties on $\boldsymbol{S}$ like e.g. sparsity; and $\mathcal{S}$ is the constrained set of $\boldsymbol{S}$ being a valid graph operator, e.g. like non-negativity of the edge weights.

In (2.38), it is assumed perfect knowledge of the eigenvectors $\boldsymbol{\chi}$. In practice, it is not true because they are found from a sample covariance, and $\boldsymbol{\chi}$ is as more noisy as the number of data sample is small relative to the number of vertices of the graph. To take this into account, robust network topology methods relax the equality constraint in (2.38) and substitute this with the inequality $d(\boldsymbol{S}, \tilde{\boldsymbol{\chi}}\mathbf{\Lambda}\tilde{\boldsymbol{\chi}}^\top) \leq \epsilon$, with $\tilde{\boldsymbol{\chi}}$ be the noisy covariance eigenvectors, $d(\cdot)$ a convex matrix distance and ϵ is a tuning parameter chosen based on a priori information on the noise level [32].

Alternatively, the graph signal can be viewed as observations at different time instants of a few processes that start at different nodes and diffuse with time. Such a signal can be represented as graph heat kernels or localized graph kernels, see [34,67]. Algorithms used in [34,67] can be seen as a generalization of dictionary learning to graph signals. Dictionary learning is a broad research area where signals are modelled as a linear combination of single components (atoms) in an over-complete basis, and the relevant characteristics of the signal are revealed by its sparse representation. In particular, the concatenation of a set of heat diffusion operators at different time instances defines a dictionary used to model a signal as follows

$$\boldsymbol{y} = [e^{-\tau_1\mathbf{L}}, \ldots, e^{-\tau_S\mathbf{L}}]\boldsymbol{c} \tag{2.39}$$

where $\boldsymbol{\tau} := [\tau_1, \ldots, \tau_S]$ are the heat rates corresponding to each of S diffusion filters $e^{-\tau_s\mathbf{L}}$. The signal $\boldsymbol{y}$ is then the linear combination of different heat diffusion processes evolving on the graph, where $\boldsymbol{c} \in \mathbb{R}^{NS}$ collects the sparse coefficients that combined with the columns of the dictionary approximate the graph signal $\boldsymbol{y}$.

If we collect T observed signals in the matrix $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T] \in \mathbb{R}^{N\times T}$, the graph learning problem can be cast as a structured dictionary learning problem formulated as

$$\begin{aligned} &\min_{\mathbf{L},\mathbf{C},\boldsymbol{\tau}} \quad \|\mathbf{Y} - [e^{-\tau_1\mathbf{L}}, \ldots, e^{-\tau_S\mathbf{L}}]\mathbf{C}\|_F^2 + \alpha\sum_{t=1}^{T}\|\boldsymbol{c}_t\|_1 + \beta\|\mathbf{L}\|_F^2 \\ &\text{s. to } \{\tau_s\}_{s=1}^S \geq 0,\ \mathrm{Tr}(\mathbf{L}) = N,\ \mathbf{L}\mathbf{1} = \mathbf{0},\ [\mathbf{L}]_{ij} = [\mathbf{L}]_{ji},\ i \neq j \end{aligned} \tag{2.40}$$

where $\mathbf{C} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_S]$. The optimization problem in (2.40) is non-convex and it has potentially many local minima. It can be solved with proximal alternating linearized minimization algorithm as in [34], with complexity $O(N^3)$.

2.5.4 Structural equation models based Topology ID

Undirected graphs, like correlation networks, cannot capture causality. We will recall here an alternative topology identification (ID) approach that deal with directionality that entails structural equation models (SEMs). Such model is at the basis of our work on topology identification and signal reconstruction tasks in Chapter 5.

Structural equation model is a statistical modeling method that models causal relationships between variables in a complex systems, and it has been adopted in several fields such as economics, social sciences, genetics, and so on [68]. Linear SEMs postulate that each y_{it} depends on two sets of variables: endogenous y_{jt} and exogenous x_{it}, see Fig. 2.3.

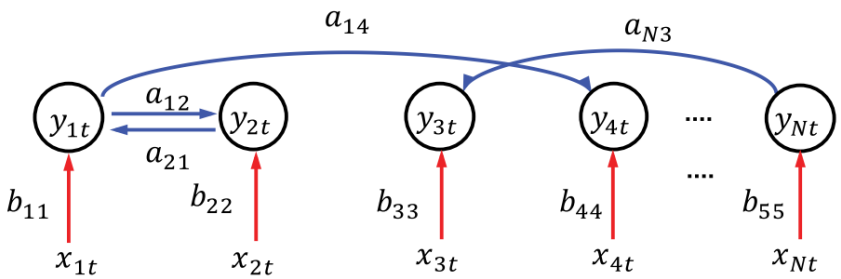


Figure 2.3. Network with N nodes and directed edges (in blue), and (the t-th sample of) exogenous measurements per node (red arrows)

This dependency is regulated by the unknown coefficients $\{a_{ij}, b_{ii}\}$, such that

$$y_{it} = \sum_{j \neq i} a_{ij} y_{jt} + b_{ii} x_{it} + v_{it}, j = 1, \ldots, N \tag{2.41}$$

where v_{it} captures the unmodeled dynamics. The term $\sum_{j \neq i} a_{ij} y_{jt}$ in (2.41) models the network effects, and implies that y_{it} is a linear combination of instantaneous values y_{jt} of the i-th single-hop neighbors $j \in \mathcal{N}_i$. Signal y_{it} also depends on external sources x_{it}, where b_{ii} captures the level of influence of these external sources. In general, $\boldsymbol{y}_t := [y_{1t}, \ldots, y_{Nt}]^\top$ can be seen as an output signal while $\boldsymbol{x}_t := [x_{1t}, \ldots, x_{Nt}]$ is the excitation or a control input. Given samples $\{y_{it}, x_{it}\}$, the topology coefficients a_{ij} can be obtained using least squares (LS) estimation possibly

regularized to effect sparsity. The model can be written in the vector form as

$$\boldsymbol{y}_t = \boldsymbol{A}\boldsymbol{y}_t + \boldsymbol{B}\boldsymbol{x}_t + \boldsymbol{v}_t \tag{2.42}$$

with $\boldsymbol{v}_t := [v_{1,t}, \ldots, v_{N,t}]^\top$ and $\boldsymbol{B} := \text{diag}(b_{11}, \ldots, b_{NN})$, or in the matrix form as

$$\mathbf{Y} = \boldsymbol{A}\mathbf{Y} + \boldsymbol{B}\boldsymbol{X} + \boldsymbol{V} \tag{2.43}$$

where $\mathbf{Y}$, $\boldsymbol{X}$ and $\boldsymbol{V}$ are the matrices collecting the column vectors $\boldsymbol{y}_t$, $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$ for all t, respectively. Thus, the regularized LS problem formulation is the following

$$\{\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}\} = \arg\min_{\boldsymbol{A},\boldsymbol{B}} \|\mathbf{Y} - \boldsymbol{A}\mathbf{Y} - \boldsymbol{B}\boldsymbol{X}\|_F^2 + \lambda\|\boldsymbol{A}\|_1 \tag{2.44a}$$

$$\text{s.to } a_{ii} = 0, \forall i \tag{2.44b}$$

$$b_{ij} = 0, i \neq j \tag{2.44c}$$

where $\|\boldsymbol{A}\|_1$ is a sparsity-promoting regularization and λ controls the sparsity level of $\boldsymbol{A}$. Constraint (2.44b) enforces the absence of a self-loop at each node, while (2.44c) ensures that $\hat{\boldsymbol{B}}$ is diagonal. Both edge sparsity and endogenous inputs have an important role to guarantee the SEM parameters to be uniquely identifiable, see [69] for details. Problem (2.44) is convex but not differentiable. It can be solved by iterative solvers, like the proximal gradient descent [70].

Moreover, the SEM method can be extended to track dynamic network topology [71], to deal with highly correlated data [72] and considering present and past observations, like SVAR leads to the structural vector autoregressive models (SVARMs).

Chapter 3

Graph signal processing in the Presence of Topology Uncertainties

Abstract

In this chapter, we will expand graph signal processing tools in the cases of a not perfectly known graph topology. Under the assumption that the topology uncertainty affects only few edges, we exploit small perturbation theory to derive closed form expressions instrumental to formulate signal processing algorithms that are *resilient to imperfect knowledge of the graph topology.* Moreover, we formulate a Bayesian approach for the estimation of the presence/absence of uncertain edges based only on the observed data and on the statistics of the data. Finally, we make use of our analysis of perturbations to study clustering and semi-supervised learning algorithms. Along the chapter, several numerical tests prove the benefits of our perturbation-aware methods.

3.1 Introduction

In applications like financial, social and biological science, data are often structured and their pairwise relations can be captured through graphs. In general, the graph is either directly observable like in *physical networks*, or it is an abstraction to represent the structure of the observed data as in *data-driven networks.* Examples of physical networks are the Internet network, power grids or wireless communication networks, where the edges have a physical meaning. On the other hand, examples of data-driven networks are brain functional activity networks or financial data networks, where a graph is inferred from the data to capture inner pairwise relations [25, 35, 58]. In most cases, the graph topology is assumed to be either known, as in physical networks, or to be inferred from the observed data. However, there are many important situations where our knowledge of the graph topology is neither perfect nor totally absent. In physical networks, like for example in wireless communication networks, some links may inadvertently drop due to random blocking or fading [73], so that we can only suppose to know a "nominal" graph. Similarly, in data-driven networks, the topology is not directly observable and it is inferred from the data. Thus, the presence of an edge is the result of a decision that depends on the observed data as well as on the inference algorithm. In fact, the data may be corrupted by noise or outliers, and the resulting inferred graph may be imperfect. Therefore, we are interested in the analysis of *uncertain*, or *probabilistic* graphs, i.e. graphs whose edges may be considered to be present only with a certain probability [74]. The family of uncertain graphs are represented by the triplet $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{p})$, where $\mathcal{V}$ is the node set, $\mathcal{E}$ is the edge set, and $\boldsymbol{p}$ is the vector that collects the probabilities of the edges to be present. Generally, all the edges can be uncertain, and in this case the dimension of vector $\boldsymbol{p}$ coincides with

the cardinality of the edge set. Alternatively, there is only a subset of edges affected by uncertainty. In this chapter, we assume that only a small percentage of edges are unknown. This assumption enables us to make use of small perturbation analysis and to derive closed form expressions useful to understand the perturbation effects as well as to derive resilient processing algorithms.

3.1.1 Related works

Uncertain graphs analysis has a long history and it encompasses different research areas [74], [75]. Indeed, in the presence of uncertain graphs, simple measures, like computing shortest path distances, which can be performed in polynomial time using Dijkstra's algorithm, become much more complicated. In particular, the problem of computing the probability that two nodes are reachable over an uncertain graph is known to be NP-complete [75]. One of the key issues in uncertain graphs is the so-called *reliability*, defined as the probability that two given (sets of) nodes are reachable, see e.g., [76, 77]. In particular, there exist a class of measures to evaluate the reliability of networks with respect to edge removal, such as integrity measure, toughness, edge vulnerability etc. [77]. Data mining over uncertain graphs is also a broad area of research [75], which includes clustering over uncertain graphs [78–80], clique discovery over uncertain graphs [81], strong communities identification [82], detection of molecular complexes in large molecular interaction networks [83], nearest-neighbor search in biological databases [84], probabilistic routing in telecommunication networks [85], core decomposition, i.e. discovery of dense subgraphs [82, 86, 87], with applications to task-driven-team-formation [88], sparsification of highly dense probabilistic graphs [89]. Edge misdetection errors propagation into global graph macro-parameters, e.g. the edge count, has also been studied in [90]. Moreover, the authors of [45] analyzed the robustness of different centrality measures under small perturbations of edge weights.

In this chapter, we study the effect of graph topology uncertainties on signal processing algorithms running on graph signals. Graph Signal Processing is wide research area that has received considerably attention in the last few years due to its many potential applications, see, e.g., [4] and the references therein. In GSP, the analysis tools come to depend on the graph topology (cf. Sec. 2.2). For example, for undirected graphs, the GFT has been defined as the projection of the graph signal onto the space spanned by the eigenvectors of the graph Laplacian matrix [3, 22]. For this reason, if the presence of a subset of edges is a random event with a certain probability to occur, the Laplacian matrix and its eigendecomposition, become random as well. This motivates the GFT analysis in probabilistic terms.

Most of the literature on GSP assumes the graph topology perfectly known,

except few noticeable examples. In [43], the authors analyze FIR and ARMA filtering of random graph signals over random graphs. Expressing the filtering operator in terms of powers of the Laplacian matrix and assuming independence between signal and graph coefficients, the authors of [43] derive the statistical properties of the filter output and suggest some denoising operations. In [44], the authors study the effect of graph errors on the performance of independent component analysis (ICA) of graph signals, whereas in [91] a technique to robustify least mean squares methods with respect to graph errors is introduced.

3.1.2 Contributions

The main goal of this chapter is to study the effects of topology uncertainties on some prominent information extraction tools, including graph signal recovery, detection of possible graph perturbation, semi-supervised learning and clustering. Then, building on our small perturbation model, we propose robust strategies that incorporate statistical knowledge about possible graph perturbations.

We suppose to know a *nominal* graph, which may not coincide whit the *real* graph. We also assume that the nominal and real graphs differ only by a small subset of edges, i.e. the number of uncertain edges is small with respect to the overall number of edges. Although representing a simplification, this assumption is well justified in many applications, like in a wireless network where the number of failing links at a given time is of course very small, or in a social network, where as rare case some connections may change over time. On the other hand, such an assumption allows us to derive closed form expressions that, albeit approximate, are useful to shed light on the impact of random edge addition/removal on the spectrum of graph signals and to devise new statistical methods to extract information from signals defined over uncertain graph. More specifically, the main contributions of this chapter are the following:

- **Section 3.2**. In this section, we derive derive: i) exact conditions stating in which cases an eigenvalues/eigenvector pair is altered, or not, from the perturbation, either addition or deletion, of an edge; ii) approximate closed form expressions for the Laplacian matrix eigenvalues/eigenvector pairs that get perturbed by the addition or removal of an edge, valid when the percentage of uncertain edges is small;

- **Section 3.3**. In this section, we evaluate the impact of edge uncertainties on basic GSP tools, such as the GFT;

- **Section 3.4**. In this section, we exploit the derived closed form perturbation

model to propose a robust signal recovery algorithm that incorporates some statistical knowledge about the topology uncertainty;

- **Section 3.5**. In this section, assuming some a priori knowledge of the signal statistical model, we propose a method to detect possible alterations of the nominal graph topology from the observation of a graph signal;
- **Section 3.6**. In this section, we exploit the derived closed form eigenvalue perturbation model to devise a robust power allocation strategy, to be used in wireless ad hoc networks;
- **Section 3.7**. In this section, we analyze the impact of erroneous assumptions about the presence of edges of a graph in semi-supervised and unsupervised algorithms.
- **Section 3.8.** Finally, in this section, we present concluding remarks and future directions.

3.2 Small perturbation analysis of graph Laplacian

In this section, we make use of small perturbation theory to analyze how the eigenpairs of the graph Laplacian are perturbed when a few percentage of edges is perturbed. Small perturbation theory is a well-established theory introduced for the first time in the seminal works of Rayleigh and Schrodinger [92, 93], for a general linear operator, and then applied to matrices [94, 95]. Later, this theory was applied to graphs in [96], [97] and [98]. Borjigin et al. [96] design spectral clustering algorithms for normalized Laplacian matrices. They present a theoretical analysis of the continuity of eigenvalues and eigenspaces that is meaningful in their multiway normalized cut spectral clustering method. In [97], Hata et al. use perturbation theory to analyze the localization properties of the Laplacian eigenvectors of random graphs. Spielman, in [98], recalls the basics of perturbation theory for matrices in order to analyze the behavior of spectral partitioning heuristics on random graphs that are generated to have good partitions. An interesting usage of perturbation theory was suggested by Von Luxburg, in [18], where she presents spectral clustering from the perturbation theory point of view.

Here below, we briefly recall some basic notions about graphs and then we show how the uncertainty on the edges translates onto the eigen-decomposition of the graph Laplacian. For a more specific review of graph theory and GSP see Sec.2.1 and Sec. 2.2. We consider an unweighted and undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, composed of a set of nodes $\mathcal{V} = \{1, \ldots, N\}$, and a set $\mathcal{E}$ of weighted edges, with cardinality

$|\mathcal{E}| = E$. We denote by $\boldsymbol{W}$ the adjacency matrix having either entries w_{ij} equal to one, if $(i,j) \in \mathcal{E}$, $i,j \in \mathcal{V}$, or zero otherwise[1]. We define $\boldsymbol{D}$ as the diagonal matrix of nodes degree, with diagonal entries $[\boldsymbol{D}]_{ii} = \sum_{j=1}^{N} w_{ij}$. Defining the Laplacian matrix as $\mathbf{L} := \boldsymbol{D} - \boldsymbol{W}$, we denote its eigendecomposition as $\mathbf{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, where $\boldsymbol{U}$ is the matrix whose columns are the eigenvectors $\boldsymbol{u}_i, i = 1, \ldots, N$, and $\boldsymbol{\Lambda}$ is the diagonal matrix of eigenvalues $\lambda_i, i = 1, \ldots, N$. Since the graph is undirected, the Laplacian matrix is symmetric and its eigenvalues are nonnegative and real.

A signal $\boldsymbol{x}$ defined over a graph $\mathcal{G}$ is a mapping $\boldsymbol{x} : \mathcal{V} \to \mathbb{R}$, associating each node with a real number. The GFT $\widehat{\boldsymbol{x}}$ of a graph signal $\boldsymbol{x}$ over undirected graphs is defined as the projection of $\boldsymbol{x}$ onto the space spanned by the eigenvectors of the Laplacian matrix [22]

$$\widehat{\boldsymbol{x}} = \boldsymbol{U}^\top \boldsymbol{x}. \tag{3.1}$$

The GFT is of interest particularly when dealing with modular graphs, composed of a set of weakly interconnected clusters, and with signals that are smooth within each cluster, but are free to assume arbitrary values across different clusters [24]. In such a case, in fact, the GFT is (approximately) sparse and this sparsity enables efficient denoising, sampling and recovery algorithms [24]. However, from (3.1) it is clear that, an imperfect knowledge of the graph topology translates into an imperfect knowledge of the Laplacian eigenvectors and then, ultimately, on the GFT of the observed signals. Our purpose is to analyze the impact of an imperfect knowledge of the graph topology on the GFT and to build analytical models that help to mitigate this impact. We consider a mismatch between the *nominal* Laplacian $\mathbf{L}$ and the *actual* Laplacian $\tilde{\mathbf{L}} := \mathbf{L} + \boldsymbol{\Delta}\mathbf{L}$. It is clear that the perturbation of $\mathbf{L}$ induces a perturbation of its eigenvalue decomposition. We denote by

$$\tilde{\mathbf{L}} := \mathbf{L} + \boldsymbol{\Delta}\mathbf{L} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{U}}^\top \tag{3.2}$$

the eigndecomposition of the perturbed matrix, with $\tilde{\boldsymbol{U}} := \boldsymbol{U} + \boldsymbol{\Delta}\boldsymbol{U}$ and $\tilde{\boldsymbol{\Lambda}} := \boldsymbol{\Lambda} + \boldsymbol{\Delta}\boldsymbol{\Lambda}$ denoting the eigenvector and eigenvalue matrices of $\tilde{\mathbf{L}}$, respectively. In general, finding the exact perturbation in closed form is impossible. Nevertheless,in this work we show that, in the case of a single edge perturbation, it is possible: i) to determine which eigenvalue/eigenvector pairs gets perturbed or not by the given edge perturbation, just looking at the eigenvectors of the nominal Laplacian matrix $\mathbf{L}$; ii) to derive approximate closed form expressions for the perturbed eigenvalues/eigenvector pairs, valid under a small perturbation assumption. To simplify the analysis, we assume that

(A0): All the eigenvalues of the nominal Laplacian matrix have multiplicity one.

[1] We consider for simplicity unweighted graphs, but the analysis can be extended to weighted graphs with some straightforward extensions.

Even though *(A0)* is typically true, there are some exceptions where the multiplicity of the eigenvalues of the adjacency matrix reveals some interesting symmetry properties of the graph [99]. In this work, we restrict our analysis to the case where *(A0)* is valid, for the sake of simplicity and of finding closed form expressions. We will add later a remark on the cases in which we can relax *(A0)* to the set of eigenvalues associated with the eigenspace we are interested in.

3.2.1 Single-edge perturbation

Let us start with the simple case where a single edge, said the m-th edge, is perturbed. We introduce the column vector $\boldsymbol{a}_m \in \mathbb{R}^N$, whose entries are all zero except the two entries $a_m(v_{m_1}) = 1$ and $a_m(v_{m_2}) = -1$, where v_{m_1} and v_{m_2} are the endpoints of edge m. Thus, the perturbed Laplacian is

$$\tilde{\mathbf{L}} := \mathbf{L} + \boldsymbol{\Delta}\mathbf{L}^{(m)} := \mathbf{L} + \sigma_m \, \boldsymbol{a}_m \boldsymbol{a}_m^\top, \tag{3.3}$$

where $\sigma_m = 1$, if edge m is added, or $\sigma_m = -1$, if edge m is removed. We consider, without loss of generality, that the original graph is connected and the eigenvalues are listed in increasing order. Since the graph is connected, the smallest eigenvalue is null and it has multiplicity one. Under assumption *(A0)*, we can then write $0 = \lambda_1 < \lambda_2 < \ldots < \lambda_N$. Since the perturbation of a single edge induces a rank-one update of the Laplacian matrix, we can exploit the seminal works [100], [101] and adapt the methodologies introduced therein to our specific case. In particular, we know in advance that, whichever is the perturbed edge, the perturbed matrix will still have the smallest eigenvalue equal to zero and the associated eigenvectors proportional to the vector of all ones, as the original Laplacian, so that there is no perturbation on the first eigenvalue/eigenvector pair, i.e. $\tilde{\lambda}_1 = \lambda_1$ and $\tilde{\boldsymbol{u}}_1 = \boldsymbol{u}_1 = \mathbf{1}$, where $\mathbf{1}$ is the vector of all ones. Since $\lambda_1 = 0$, if we introduce the $N \times (N-1)$ matrix $\boldsymbol{U}_{N-1} := [\boldsymbol{u}_2, \ldots, \boldsymbol{u}_N]$ and the $(N-1) \times (N-1)$ matrix $\boldsymbol{\Lambda}_{N-1} := \mathrm{diag}\{\lambda_2, \ldots, \lambda_N\}$, we can rewrite the original Laplacian matrix as $\mathbf{L} = \boldsymbol{U}_{N-1}\boldsymbol{\Lambda}_{N-1}\boldsymbol{U}_{N-1}^\top$. Let us introduce the column vector of size $N-1$, defined as $\boldsymbol{z}_m := [z_m(2), \ldots, z_m(N)]^\top := \boldsymbol{U}_{N-1}^\top \boldsymbol{a}_m$. Since $\boldsymbol{a}_m$ is orthogonal to $\boldsymbol{u}_1 = \mathbf{1}$, we can represent it as $\boldsymbol{a}_m = \boldsymbol{U}_{N-1}\boldsymbol{z}_m$, so that we can rewrite (3.3) as

$$\tilde{\mathbf{L}} := \boldsymbol{U}_{N-1}(\boldsymbol{\Lambda}_{N-1} + \sigma_m \, \boldsymbol{z}_m \boldsymbol{z}_m^\top) \, \boldsymbol{U}_{N-1}^\top, \tag{3.4}$$

so that we can focus on the eigendecomposition of the diagonal plus rank-one update matrix $\boldsymbol{\Lambda}_{N-1} + \sigma_m \, \boldsymbol{z}_m \boldsymbol{z}_m^\top$. The first result we derive states under what conditions an eigenvalue/eigenvector pair is not altered by the perturbation of an edge, as detailed

below.

Theorem 2. *If $u_i(v_{m_1}) = u_i(v_{m_2})$, then the i-th eigenvalue/eigenvector pair is not altered by the addition/deletion of edge m.*

Proof. Please, see the Appendix. ■

In general, of course, except under the conditions of Theorem 1, the perturbation of an edge causes a perturbation of the eigenvalue/eigenvectors pairs. Nevertheless, since the matrix perturbation is rank-one, this perturbation is limited by the eigenvalue gap, as clarified next. Adapting the methodology of [100] to our case, in case of addition of one edge, the perturbed eigenvalues satisfy the inequalities:

$$\begin{aligned} \lambda_i \leq \quad & \tilde{\lambda}_i \leq \lambda_{i+1}, && i = 2, \ldots, N-1 \\ \lambda_N \leq \quad & \tilde{\lambda}_N \leq \lambda_N + \boldsymbol{z}_m^\top \boldsymbol{z}_m. \end{aligned}$$

Conversely, in case of edge removal, the inequalities become

$$\begin{aligned} \lambda_2 - \boldsymbol{z}_m^\top \boldsymbol{z}_m \quad & \leq \tilde{\lambda}_2 \leq \lambda_2 \\ \lambda_{i-1} \quad & \leq \tilde{\lambda}_i \leq \lambda_i, && i = 3, \ldots, N \end{aligned}$$

Furthermore, using [101], if the eigenvalues λ_i are all distinct and all the elements of $\boldsymbol{z}_m$ are nonzero, then all the above inequalities become strict inequalities. Now, we define the forward eigenvalue gap as $g_i^+ := \lambda_{i+1} - \lambda_i$, for $i = 2, \ldots, N-1$, with $g_N^+ := \boldsymbol{z}_m^\top \boldsymbol{z}_m$, and the backward gaps as $g_i^- := \lambda_i - \lambda_{i-1}$, for $i = 3, \ldots, N$, with $g_2^- := \boldsymbol{z}_m^\top \boldsymbol{z}_m$. Hence, denoting with $\Delta\lambda_{i,m} := \tilde{\lambda}_i - \lambda_i$ the eigenvalue perturbation, we can write

$$\begin{aligned} 0 \leq & \Delta\lambda_{i,m} \leq g_i^+, && \text{in case of edge addition;} \\ -g_i^- \leq & \Delta\lambda_{i,m} \leq 0, && \text{in case of edge removal.} \end{aligned} \tag{3.5}$$

Now, for the sake of finding closed form expressions for the eigenvalue/eigenvector perturbations, we make a small perturbation analysis, assuming that

(A1): $\|\Delta\mathbf{L}\|_F \ll \|\mathbf{L}\|_F$,

In our case, we are interested in the situation in which only a small percentage of edges is altered (either removed or added). This hypothesis is of course consistent with *(A1)*. Under *(A1)*, in the Appendix, we derived approximate closed form expressions for the eigenvalue and eigenvector perturbations, valid under *(A0)*, stating that the perturbed eigenvalue can be approximated as

$$
\begin{aligned}
\tilde{\lambda}_i &\simeq \lambda_i + \boldsymbol{u}_i^\top \boldsymbol{\Delta}\mathbf{L}^{(m)} \boldsymbol{u}_i \\
&= \lambda_i + \sigma_m \boldsymbol{u}_i^\top \boldsymbol{a}_m \boldsymbol{a}_m^\top \boldsymbol{u}_i := \lambda_i + \delta\lambda_{i,m} \\
&= \lambda_i + \sigma_m \left[u_i(v_{m_1}) - u_i(v_{m_2})\right]^2. \qquad (3.6)
\end{aligned}
$$

This formula is accurate when $[u_i(v_{m_1}) - u_i(v_{m_2})]^2 \ll g_i^+$, in case of an edge addition, or $[u_i(v_{m_1}) - u_i(v_{m_2})]^2 \ll g_i^-$, in case of an edge deletion, which means that the eigenvalue perturbation is not only smaller than the eigenvalue gap, but much smaller. The validity of this assumption will be verified numerically later on.

Furthermore, under the above approximations, the perturbed eigenvector is approximately

$$
\begin{aligned}
\tilde{\boldsymbol{u}}_i &\simeq \boldsymbol{u}_i + \sum_{\substack{j=2 \\ j\neq i}}^{N} \frac{\boldsymbol{u}_j^\top \boldsymbol{\Delta}\mathbf{L}^{(m)} \boldsymbol{u}_i}{\lambda_i - \lambda_j} \boldsymbol{u}_j \\
&= \boldsymbol{u}_i + \sigma_m \sum_{\substack{j=2 \\ j\neq i}}^{N} \frac{\boldsymbol{u}_j^\top \boldsymbol{a}_m \boldsymbol{a}_m^\top \boldsymbol{u}_i}{\lambda_i - \lambda_j} \boldsymbol{u}_j := \boldsymbol{u}_i + \delta\boldsymbol{u}_{i,m} \\
&= \boldsymbol{u}_i + \sigma_m \sum_{\substack{j=2 \\ j\neq i}}^{N} \frac{[u_j(v_{m_1}) - u_j(v_{m_2})][u_i(v_{m_1}) - u_i(v_{m_2})]}{\lambda_i - \lambda_j} \boldsymbol{u}_j \\
&:= \boldsymbol{u}_i + \sigma_m \sum_{\substack{j=2 \\ j\neq i}}^{N} b_{ji}^{(m)} \boldsymbol{u}_j, \qquad (3.7)
\end{aligned}
$$

apart a scalar coefficient needed to enforce a unit norm.

3.2.2 Multiple-edge perturbation

Let us consider now the case in which multiple edges are perturbed. The perturbation of the Laplacian matrix can be written as

$$
\Delta\mathbf{L} = \sum_{m\in\mathcal{E}_p} \Delta\mathbf{L}^{(m)} = \sum_{m\in\mathcal{E}_p} \sigma_m \boldsymbol{a}_m \boldsymbol{a}_m^\top, \qquad (3.8)
$$

where $\mathcal{E}_p$ denotes the set of perturbed edges. We can still apply the previous single-edge perturbation analysis, provided that the overall eigenvalue perturbation does not exceed the eigenvalue gap. The new condition becomes then

$$
-g_i^- \ll \sum_{m\in\mathcal{E}_p} \sigma_m [u_i(v_{m_1}) - u_i(v_{m_2})]^2 \ll g_i^+. \qquad (3.9)
$$

Under the above small perturbation condition, the eigenvalue and eigevector perturbation can be expressed as the sum of the perturbations pertaining each edge perturbation, i.e.,

$$\delta\lambda_i = \sum_{m\in\mathcal{E}_p} \sigma_m \delta\lambda_{i,m} \tag{3.10}$$

$$\delta\boldsymbol{u}_i = \sum_{m\in\mathcal{E}_p} \sigma_m \delta\boldsymbol{u}_{i,m} \tag{3.11}$$

with $\delta\lambda_{i,m}$ and $\delta\boldsymbol{u}_{i,m}$ as given in (3.6) and (3.7).

Remark 1: Interestingly enough, the perturbation formulas reported above coincide with the approximation formulas derived in [94] applying small perturbation theory, without necessarily assuming rank-one updates.

Remark 2: Our theoretical analysis has been carried out assuming that all eigenvalues are distinct. Indeed, if we are interested only in a subset of eigenvalues, what is strictly necessary is only that the eigenvalues for which we compute the perturbation are simple. This statement will be later checked numerically. Within their limit of validity, formulas (3.6), and (3.7) shed light on some relevant aspects of perturbation and their relation to graph topology. For example, it is known from spectral graph theory, see e.g. [18], that, if the graph is composed of C clusters, the eigenvectors associated with the C smallest eigenvalues tend to assume approximately the same value within each cluster, while they can vary arbitrarily across different clusters. In such a case, the above perturbation formulas (3.6) and (3.7) give rise to the following interpretations:

1. the edges whose deletion causes the largest perturbation are inter-cluster edges;
2. the perturbation is larger for quantities associated with eigenvalues very similar to each other (recall that formulas (3.6) and (3.7) hold true only for distinct eigenvalues).

To check the validity of the theoretical approximation of the perturbed eigenvalues, as given in (3.6), we generated a random geometric graph (RGG) with $N = 150$ nodes and considered different choices of the coverage radius r_0, in order to enforce different average degrees. Then, for each realization of the RGG, we perturbed its edges by generating a binary random variable (rv) equal to one, with probability $\mathbb{P}$, if the edge is perturbed, or zero otherwise. The perturbations over different edges are generated as statistically independent events. For each resulting graph, we compared the true perturbation $\Delta\lambda_i$ of the i-th eigenvalue with its theoretical approximation $\delta\lambda_i = \sum_{m\in\mathcal{E}} \delta\lambda_{i,m}$, with $\delta\lambda_{i,m}$ given by (3.6). We checked three eigenvalues, namely (λ_2, λ_3 and λ_5), and used as metric the Normalized Perturbation Mismatch (NPM), defined as $\mathrm{NPM}_i := |\Delta\lambda_i - \delta\lambda_i|/\lambda_i$. In Fig. 3.1, we report the normalized error

NPM_i vs. the mean degree, for two different choices of the probability $\mathbb{P}$ that each single edge is perturbed. We can see how, in both cases, the relative accuracy with which the perturbations of the eigenvalues are approximated by our closed form expressions is quite good. We also tested our approximations over scale-free graphs of 250 nodes, generated using the preferential attachment method, using as starting seed an RGG with 6 nodes, considering different choices of the number of new link M_{new} that are added for each new node during the graph building process. In Fig. 3.2, we report the normalized error NPM_i vs. M_{new}. In our simulation, we perturbed only the links incident to the hubs nodes, chosen as the 50 nodes with the largest degree. Finally, we tested our approach over Erdös-Rényi graphs of 150 nodes, considering different edge probabilities p_e. In Fig. 3.3, we show the NPM_i vs. p_e, assuming the same failure probability over all the links. Each curve is averaged over 50 statistically independent graph realizations and, for each graph, $10,000$ realizations of edge random deletion. From Figures 3.2 and 3.3, we can see that, also for the scale-free and the Erdös-Rényi graph cases, the closed form expressions provide a good approximation of the true perturbation. Furthermore, we tested the approximated expressions of the Laplacian eigenvalues of a (connected) modular graph built as a number C of RGGs weakly interconnected through a few links placed at random. Figures 3.4 reports the normalized error NPM_i vs. the number of clusters, for two different choices of the probability $\mathbb{P}$ that each single edge is perturbed. From Figs. 3.4, we can see that the relative accuracy of our perturbation analysis is quite good, also for modular graphs.

To check the validity of the approximation of the perturbed eigenvectors, as given by (3.7), we considered two important applications of the Laplacian eigenvectors: the analysis of band-limited graph signals and clustering. In both cases, we assume a connected graph to be composed of C clusters interconnected among each other by a relatively small number of edges. In the first case, the signal is assumed to be smooth within each cluster, but it can vary arbitrarily from cluster to cluster. Such signals can be well approximated as a linear combinations of the eigenvectors associated with the C smallest eigenvalues of the graph Laplacian. In the clustering case, a popular graph-based method to detect clusters consists, again, in looking at the eigenvectors associated with the C smallest eigenvalues and applying k-means to those vectors [18] (cf. Sec. 2.3). In both cases, the structure of the eigenvectors associated with the C smallest eigenvalues of $\mathbf{L}$ plays a key role. Hence, to test the accuracy of the closed form expressions in (3.7) in view of the above potential applications, we generated random connected graphs composed of C clusters, and each cluster is generated as a RGG, and then we placed few edges at random to connect the clusters. Then, for each synthetic graph, we perturbed the topology by

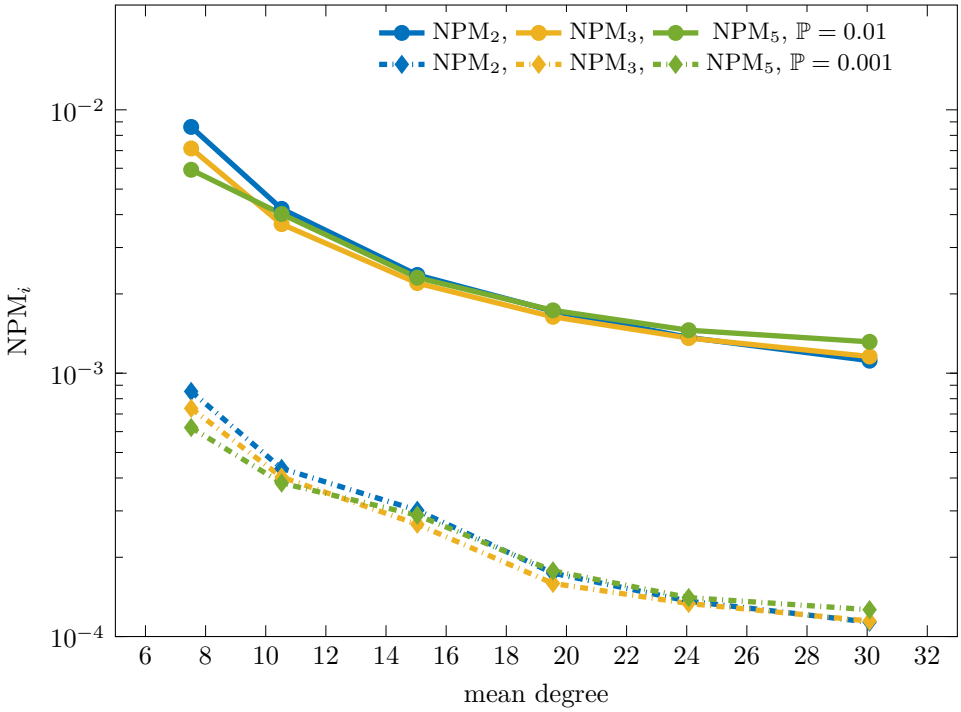


Figure 3.1. Perturbation mismatch vs. mean degree for a RGG, with N=150.

deleting randomly some of its edges, with a probability $\mathbb{P}$ equal for each edge. We chose, as performance metric, the distance (angle in radians) between the subspace spanned by the first C eigenvectors computing the exact eigen-decomposition of the perturbed Laplacian matrix $\tilde{\mathbf{L}}$ and the subspace spanned by the first C eigenvectors of $\tilde{\mathbf{L}}$ computed using (3.7). In Fig. 3.5, we show the subspace distance for different C and N, as a function of failure probability $\mathbb{P}$, constant over all the links. The results in Fig. 3.5 are averaged over 50 independent graph realizations and, for each graph, over $10,000$ realizations of the deletion events. The simulation results show that the approximation improves as the number N of nodes increases. Moreover, the approximation error increases as the number of clusters increases, also because the dimension of the subspace increases and then there are more degrees of freedom.

Remark. Investigating about the spectrum perturbation of the Laplacian matrix arises also the problem of eventual graph symmetries perturbations. The fact that the presence of symmetries in a graph is manifested by the presence of sparse Laplacian eigenvectors is well known [102–104]. More specifically, if two (or more) nodes share the same set of neighbors, they are identified by a sparse eigenvector that has a corresponding integer eigenvalue of the Laplacian matrix. For example, for a 2-sparse eigenvector, it is known the following result [102]:

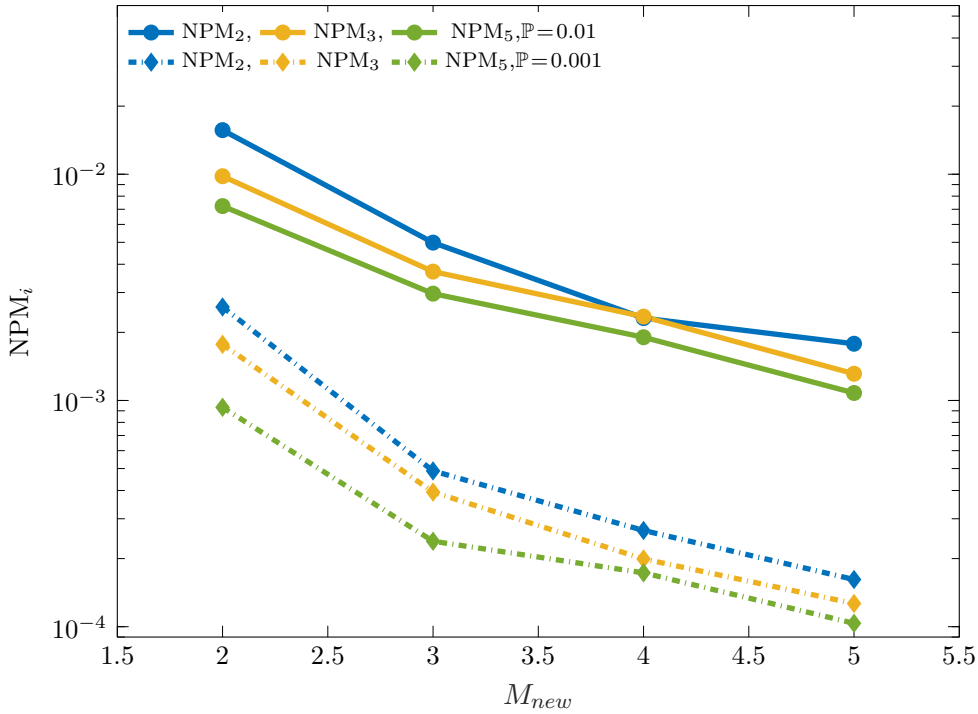


Figure 3.2. Perturbation mismatch vs. M_{new} for a scale free graph, with N=250.

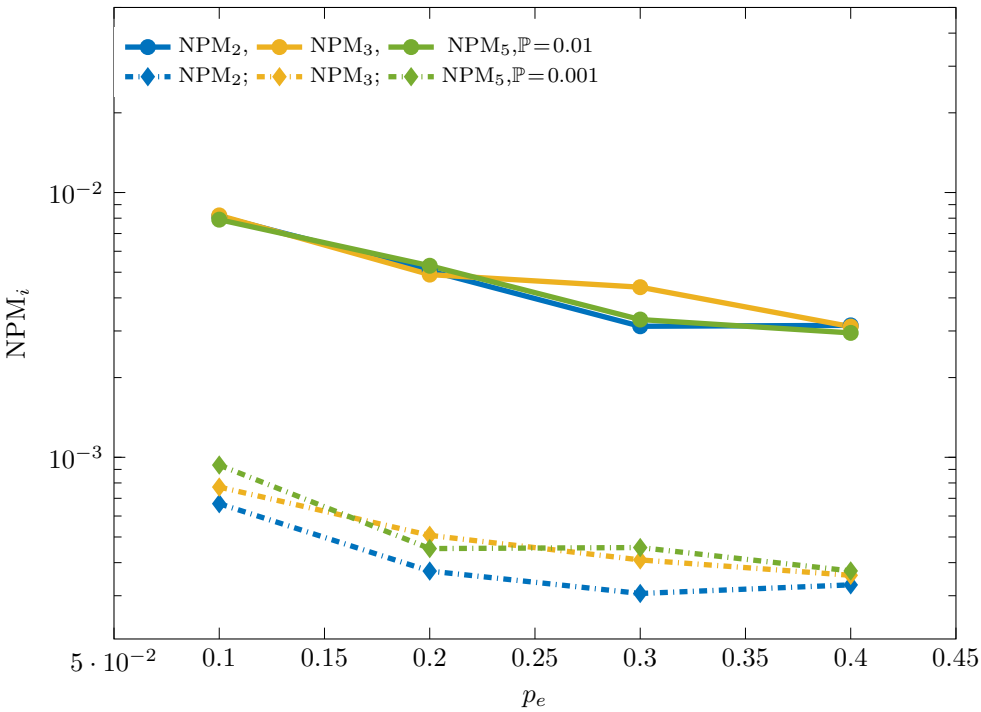


Figure 3.3. Perturbation mismatch vs. edge probability, for an Erdös-Rényi graph, with N=150.

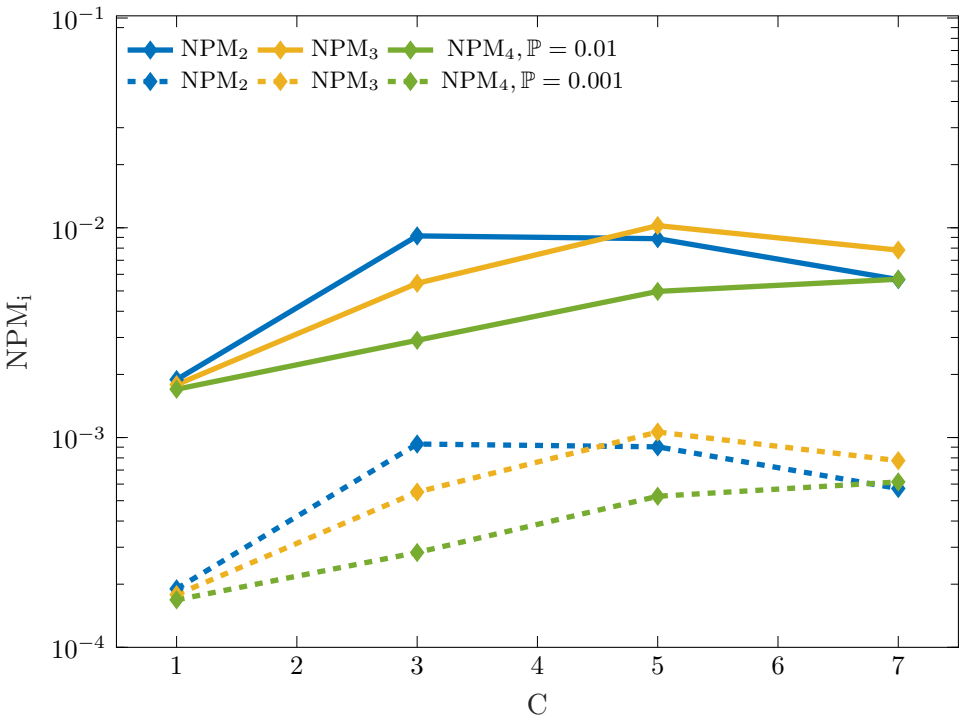


Figure 3.4. NPM$_i$ vs number of clusters C when each cluster is an RGG.

Proposition 1. *Let $\boldsymbol{W}$ be the adjacency matrix of a connected and undirected graph, with entries w_{ij}, and be d_i the degree of the i-th node. Then, there exist two nodes i and j such that*

$$w_{ik} = w_{jk}, \ \forall k \in \{1, \ldots, N\} \backslash \{i, j\} \tag{3.12}$$

if and only if the graph Laplacian $\mathbf{L}$ has a 2-sparse eigenvector $\boldsymbol{u}_k$ with the associated eigenvalue different from zero $\lambda_k = d_i + w_{ij}$.

The 2-sparse unit length eigenvector $\boldsymbol{u}_k$ will have all entries equal to zero but the i-th entry equal $\pm 1/\sqrt{2}$ and the j-th entry equal to $\mp 1/\sqrt{2}$. Thus, from (3.6), it is clear that the largest perturbation of λ_k occurs when the edge w_{ij} is perturbed, i.e. $\delta\lambda_k = 2$. For the binary weights graph, the eigenvalue perturbation computed by (3.6) suggests the same result of Prop. 1. In fact, denoting with $\tilde{d}_i$ the degree of i after the removal of the edge connecting the nodes i and j, the k-th perturbed eigenvalue is exactly $\tilde{\lambda}_k = \tilde{d}_i = \lambda_k - 2$.

3.2.3 Statistical analysis

In this section, we provide a statistical analysis of eigenvalue/eigenvector perturbations. Consider the removal/addition of an edge m as a random event characterized

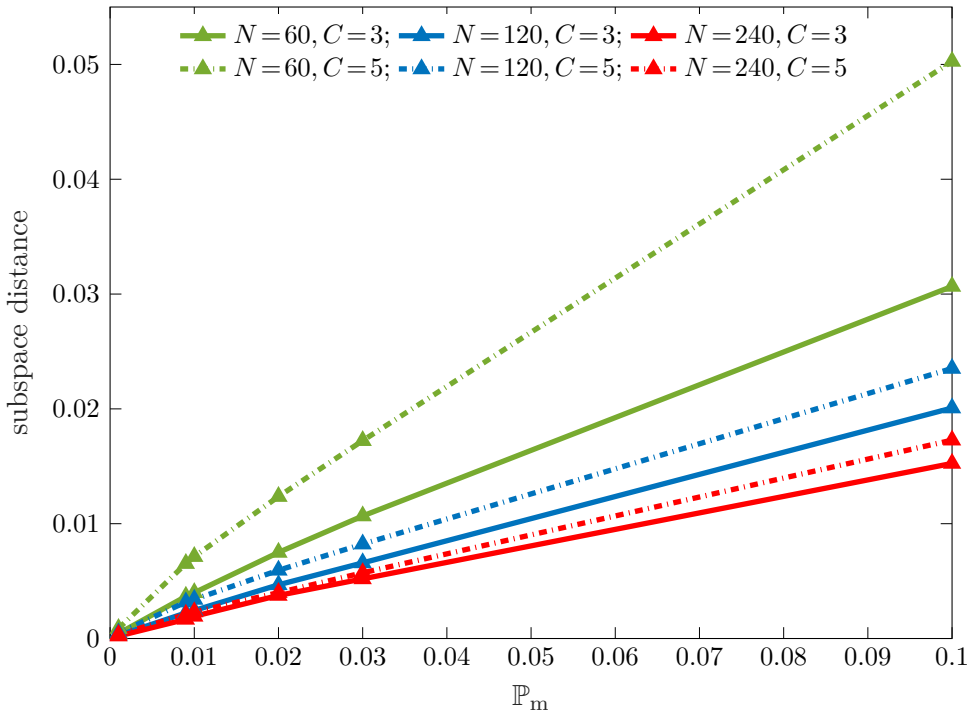


Figure 3.5. Distance vs failure probability.

by a certain probability $\mathbb{P}_m$. In particular, we describe the perturbation of edge m as a binary r.v. Z_m equal to 1, with probability $\mathbb{P}_m$, if edge m is perturbed, and 0 otherwise. Furthermore, we denote by $\mathcal{E}_p \subseteq \mathcal{E}$, the subset of edges present in the *nominal* graph $\mathcal{E}$, which could be missing in the *actual* graph, and by $\bar{\mathcal{E}}_p \subseteq \bar{\mathcal{E}}$ the subset of missing edges in the *nominal* graph $\bar{\mathcal{E}}$, which could be present in the *actual* graph. Within the limits of validity of first order perturbation analysis, the overall perturbation of eigenvalue λ_i and the corresponding eigenvector $\boldsymbol{u}_i$ can be approximated as

$$\delta\lambda_i := \sum_{m \in \mathcal{E}_p \cup \bar{\mathcal{E}}_p} Z_m \delta\lambda_{i,m} \tag{3.13}$$

$$\delta\boldsymbol{u}_i := \sum_{m \in \mathcal{E}_p \cup \bar{\mathcal{E}}_p} Z_m \delta\boldsymbol{u}_{i,m} \tag{3.14}$$

with $\delta\lambda_{i,m}$ and $\delta\boldsymbol{u}_{i,m}$ given in (3.6) and (3.7), respectively. In a small perturbation framework, the validity of (3.13) holds as far as the number of perturbed edges does not lead to a violation of assumptions (A0) and (A1). As a rule of thumb, we assume that, if the probabilities assume a constant value, $\mathbb{P}_m = \mathbb{P}, \forall m \in \mathcal{E}_p \cup \bar{\mathcal{E}}_p$, the product $(|\mathcal{E}_p| + |\bar{\mathcal{E}}_p|)\mathbb{P}$ has to be sufficiently smaller than one.

Assuming statistical independence of the events associated with edge removal/addition,

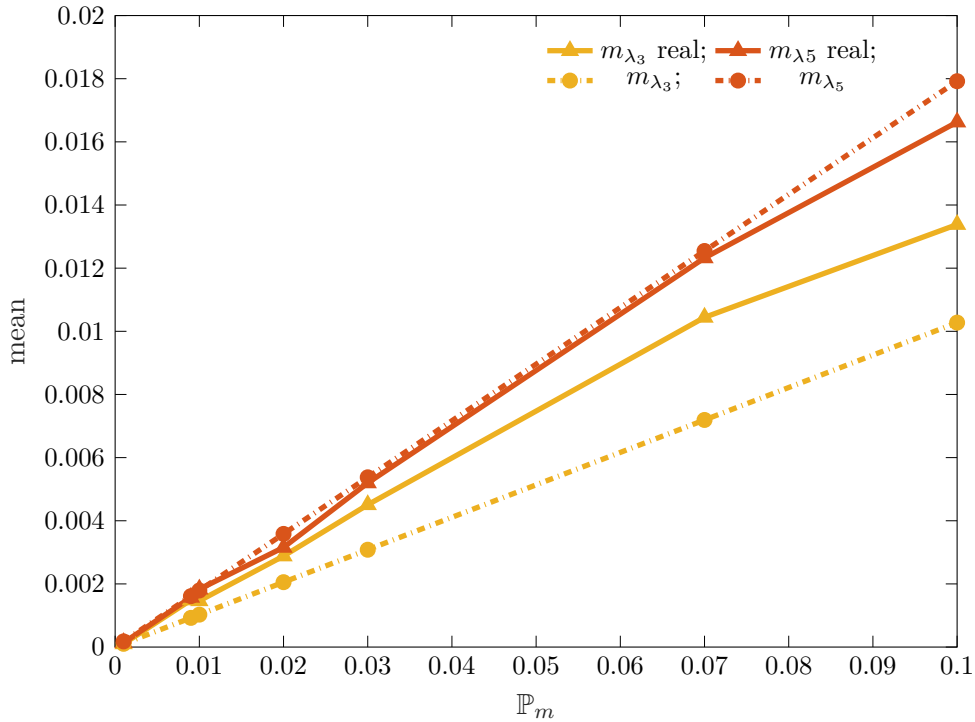


Figure 3.6. Mean value of λ_3 and λ_5 computed numerically (solid line) and with (3.15)

it is easy to derive eigenvectors/eigenvalues statistical properties. In particular, mean value and variance of $\delta\lambda_i$ are

$$m_{\delta\lambda_i} := \mathbb{E}\{\delta\lambda_i\} = \sum_{m\in\mathcal{E}_p} \mathbb{P}_m\, \delta\lambda_{i,m} - \sum_{m\in\bar{\mathcal{E}}_p} \mathbb{P}_m\, \delta\lambda_{i,m}, \tag{3.15}$$

$$\mathrm{var}\{\delta\lambda_{(i)}\} = \sum_{m\in\mathcal{E}_p\cup\bar{\mathcal{E}}_p} (1-\mathbb{P}_m)\mathbb{P}_m\, \delta\lambda_{i,m}^2. \tag{3.16}$$

To check the validity of these expressions, we generated a graph of $N = 240$ nodes and $C = 5$ clusters, with 48 nodes per cluster and few links placed at random to connect the clusters. In Fig. 3.6, we compare the mean value of two eigenvalues computed using expression (3.15) or numerically, by averaging over $1,000$ independent realizations of the failure events. We can see from Fig. 3.6 that the approximation is fairly good. For later purposes, it is useful to evaluate the probability that the random variable $\delta\lambda_i$ or $\delta u_i(\ell)$ does not deviate from their expected values more than a given value t. This probability is not easy to compute, hence we used the

Hoeffding's bound [105]:

$$\mathbb{P}\{\delta u_i(\ell) - m_{\delta u_i(\ell)} \geq t\} \leq \exp\{-2t^2 / \sum_{m \in \mathcal{E}_p \cup \bar{\mathcal{E}}_p} \delta u_{i,m}^2(\ell)\},$$

$$\mathbb{P}\{\delta \lambda_i - m_{\delta \lambda_i} \geq t\} \leq \exp\{-2t^2 / \sum_{m \in \mathcal{E}_p \cup \bar{\mathcal{E}}_p} \delta \lambda_{i,m}^2\}.$$

In particular, the probability that the deviation of the ℓ-th entry of the i-th eigenvector be larger than a certain percentage $\alpha \in [0, 1]$ of the unperturbed value $u_i(\ell)$, is upper bounded by

$$\begin{aligned} &\mathbb{P}\{\delta u_i(\ell) \geq \alpha u_i(\ell)\} \\ &\leq \exp\{-2(\alpha u_i(\ell) - m_{\delta u_i(\ell)})^2 / \sum_{m \in \mathcal{E}_p \cup \bar{\mathcal{E}}_p} \delta u_{i,m}^2(\ell)\}, \end{aligned} \tag{3.17}$$

and the probability that the eigenvalue perturbation be smaller than a certain percentage $\alpha \in [0, 1]$ of the true eigenvalue is:

$$\mathbb{P}\{\delta \lambda_i \geq \alpha \lambda_i\} \leq \exp\{-2(\alpha \lambda_i - m_{\delta \lambda_i})^2 / \sum_{m \in \mathcal{E}_p \cup \bar{\mathcal{E}}_p} \delta \lambda_{i,m}^2\}. \tag{3.18}$$

Note that, the assumption on the statistical independence of the perturbation events is simplistic, but it helps us to derive a perturbation analysis of the eigenvalues/eigenvectors in closed form. This enables us to formulate and solve perturbation-aware graph-based learning problems.

3.3 Impact of graph uncertainties on graph signal spectrum

In this section, we want to assess how the uncertainty on a subset of edges affects the spectrum of a graph signal. In fact, if the graph is only imperfectly known, the uncertainty on the graph topology translates into an uncertainty of the spectrum of the observed signal. Moreover, the results of this section will be used later, to formulate a robust graph signal processing in the presence of uncertainties.

Given an undirected graph $\mathcal{G}$ with Laplacian matrix $\mathbf{L}$, let $\boldsymbol{x}$ be a signal defined over the graph. Its Graph Fourier Transform has been defined as [106], [107]:

$$\widehat{\boldsymbol{x}} = \boldsymbol{U}^\top \boldsymbol{x}, \tag{3.19}$$

where $\boldsymbol{U}$ is the matrix whose columns are the eigenvectors of $\mathbf{L}$. In our setting, $\boldsymbol{U}$ contains the eigenvectors of the *nominal* graph and $\tilde{\boldsymbol{U}} := \boldsymbol{U} + \Delta\boldsymbol{U}$ the eigenvectors associated with the real graph. Clearly, a mismatch between true and nominal eigenvectors induces an error in the spectral analysis. In particular, we consider smooth signals. These signals can be expressed (approximately) as a linear combination of the first eigenvectors of the graph Laplacian. Their spectrum is then typically concentrated around the lowest frequency components. More specifically, if the bandwidth is K, lowpass band-limited signals admit a compact representation (on the true graph):

$$\boldsymbol{x} = \tilde{\boldsymbol{U}}\boldsymbol{s}_0 = \tilde{\boldsymbol{U}}_K\boldsymbol{s} := (\boldsymbol{U}_K + \Delta\boldsymbol{U}_K)\boldsymbol{s}, \tag{3.20}$$

where $\tilde{\boldsymbol{U}}_K \in \mathbb{R}^{N\times K}$ contains the first K eigenevectors of $\tilde{\boldsymbol{U}}$, while $\boldsymbol{s}_0^\top := [\boldsymbol{s}^\top, \mathbf{0}^\top]^\top$, where $\boldsymbol{s}$ is a column vector of size K, which enables a compact representation of the observed signal over the true eigenvector basis $\tilde{\boldsymbol{U}}_K$. The presence of a mismatch $\Delta\boldsymbol{U}_K$ between real and nominal eigenvectors gives rise to erroneous signal components. Using (3.7) and (3.14), and considering, for the sake of simplicity, only failure events, this mismatch can be expressed in closed form as

$$\Delta\boldsymbol{U}_K \simeq \sum_{m\in\mathcal{E}_p} Z_m[\delta\boldsymbol{u}_{1,m}, \ldots, \delta\boldsymbol{u}_{K,m}] = \sum_{m\in\mathcal{E}_p} Z_m\boldsymbol{U}\boldsymbol{B}_m, \tag{3.21}$$

where $\boldsymbol{B}_m \in \mathbb{R}^{N\times K}$, $[\boldsymbol{B}_m]_{ij} = b_{i,j}^{(m)}$ defined in (3.7) for $j \neq 1, i \neq 1$ and $i \neq j$, while we set 1) $b_{i,1}^{(m)} = 0, \forall i = 1, \ldots, N$, 2) $b_{1,j}^{(m)} = 0, \forall j = 1, \ldots, K$, and 3) $b_{i,i}^{(m)} = 0, \forall i = 1, \ldots, K$. These zero entries simply encode the property that 1) the first eigenvector is not perturbed, 2) it does not contribute to perturb any other vector and, trivially, 3) no eigenvector perturbs itself.

Computing the GFT of the observed vector $\boldsymbol{x}$ using the nominal eigenvectors $\boldsymbol{U}$ gives then rise to:

$$\widehat{\boldsymbol{x}} = \boldsymbol{U}^\top\boldsymbol{x} = \boldsymbol{s}_0 + \Delta\widehat{\boldsymbol{x}} \tag{3.22}$$

where, using (3.21), the error can be expressed as

$$\Delta\widehat{\boldsymbol{x}} = \boldsymbol{U}^\top\Delta\boldsymbol{U}_K\boldsymbol{s} \simeq \sum_{m\in\mathcal{E}_p} Z_m\boldsymbol{B}_m\boldsymbol{s}. \tag{3.23}$$

This shows that the spectrum perturbation contains both an out-band and an in-band component, except for the first entry, which is null because the first eigenvector is not perturbed (unless the graph becomes disconnected, in which case the nullspace of $\mathbf{L}$ increases up to an order equal to the number of disconnected components). Clearly, the spectrum error depends on the true signal coefficients as well as on the edge perturbation events. Since $\Delta\widehat{\boldsymbol{x}}$ is a set of random variables,

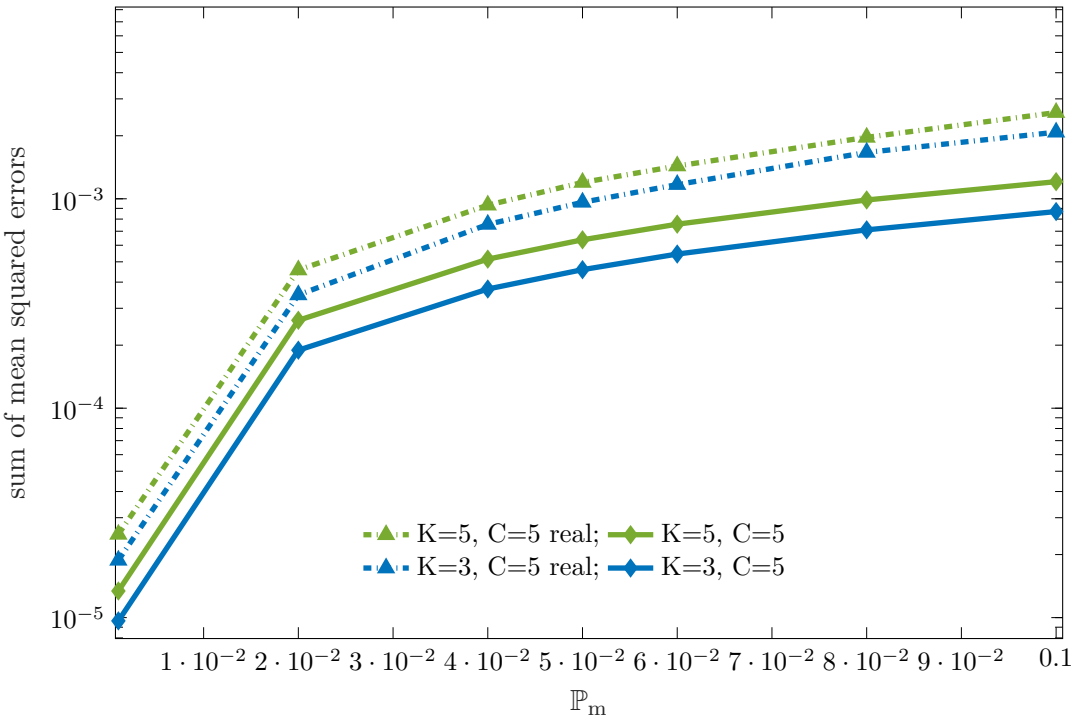


Figure 3.7. Overall mean squared errors vs. edge failure probability.

denoting with $\boldsymbol{q}_m := \boldsymbol{B}_m \boldsymbol{s}$, the expected value and variance of each entry are

$$E\{\Delta\widehat{x}(k)\} = \sum_{m \in \mathcal{E}_p} \mathbb{P}_m q_m(k) \tag{3.24}$$

$$\mathrm{var}\{\Delta\widehat{x}(k)\} = \sum_{m \in \mathcal{E}_p} \mathbb{P}_m (1 - \mathbb{P}_m) q_m^2(k) \tag{3.25}$$

with $k = 2, \ldots, N$.

To gain insight into the spectrum perturbation, we considered the following example. We considered a graph composed of 240 nodes forming five clusters ($C = 5$) interconnected by a few bridges placed at random. Each edge has a probability to fail $\mathbb{P}_m$, assumed to be all equal for all m, $\mathcal{E}_p = \mathcal{E}$, and $\mathbb{P}_m = 0, \forall m \in \bar{\mathcal{E}}_p$. On top of this graph, we generated two band-limited signals, having bandwidths $K = 3$ and $K = 5$. Fig. 3.7 depicts the sum of mean squared errors on all the spectrum coefficients, computed as $\sum_{k=1}^{N} \mathrm{var}\{\Delta\widehat{x}(k)\} + E\{\Delta\widehat{x}(k)\}^2$, obtained over the two different signals vs. the edge failure probability, obtained by simulation (dashed line) or by using (3.25) (solid line). This figure shows that there is a good agreement between the simulation results and the theoretical findings. The theoretical values tend to slightly underestimate the simulated values, because of the use of first order approximations. As expected, the error increases as the failure probability or the

bandwidth increase.

3.4 Robust signal recovery

In this section, we propose a robust spectrum recovery algorithm, motivated by the analysis of Sec. 3.3. Such algorithm incorporates any available knowledge about the topology perturbations statistics. Consider an observed signal model that includes both the effect of the topology perturbation, modeled as in (3.20) and (3.21), plus noise

$$\boldsymbol{x} = \boldsymbol{U}_K \boldsymbol{s} + \sum_{m \in \mathcal{E}_p} Z_m \boldsymbol{U} \boldsymbol{B}_m \boldsymbol{s} + \boldsymbol{v} \tag{3.26}$$

where $\boldsymbol{s}$ is modeled as a vector of statistically independent random variables with zero mean and covariance matrix $\boldsymbol{R}_{ss} := \sigma_s^2 \boldsymbol{I}$, and $\boldsymbol{v}$ is additive noise, statistically independent of $\boldsymbol{s}$, with zero mean and covariance matrix $\boldsymbol{R}_{vv} := \sigma_n^2 \boldsymbol{I}$.

We consider a linear estimator, so that the estimated vector $\widehat{\boldsymbol{s}}$ is obtained as $\widehat{\boldsymbol{s}} = \boldsymbol{G}^\top \boldsymbol{x}$. The goal of this section is to find the matrix $\boldsymbol{G}$ that yields the minimum mean square error (MMSE) estimate. In the ideal case of unperturbed topology, $\boldsymbol{G}$ is simply proportional to $\boldsymbol{U}_K$. In the presence of unknown perturbations, we minimize the MSE exploiting the structure of the observation model in (3.26) and prior information about the statistics of the graph perturbations, i.e. the probabilities that some of the edges may fail. The overall MSE can be written as

$$\begin{aligned} \mathrm{MSE}(\boldsymbol{G}) &= E\{(\widehat{\boldsymbol{s}} - \boldsymbol{s})(\widehat{\boldsymbol{s}} - \boldsymbol{s})^\top\} \\ &= E\{(\boldsymbol{G}^\top \boldsymbol{x} - \boldsymbol{s})(\boldsymbol{G}^\top \boldsymbol{x} - \boldsymbol{s})^\top\}. \end{aligned} \tag{3.27}$$

Let $\boldsymbol{R}_{zz}$ be the covariance matrices of $\boldsymbol{z} := [Z_1, \ldots, Z_{|\mathcal{E}_p|}]$, whose entries are

$$[\boldsymbol{R}_{zz}]_{m,l} = \begin{cases} \mathbb{P}_m, & \text{if} \quad m = l \\ \mathbb{P}_m \mathbb{P}_l & \text{if} \quad m \neq l. \end{cases} \tag{3.28}$$

In the presence of random perturbations, the MSE is written as follows

$$\begin{aligned} \mathrm{MSE} = \mathrm{Tr}\{&\sigma_s^2 \boldsymbol{I} + \boldsymbol{G}^\top [\sigma_s^2 \boldsymbol{U}_K \boldsymbol{U}_K^\top + \sum_{m \in \mathcal{E}_p} \mathbb{P}_m \sigma_s^2 \boldsymbol{U}_K \boldsymbol{B}_m^\top \boldsymbol{U}^\top + \sum_{m \in \mathcal{E}_p} \mathbb{P}_m \sigma_s^2 \boldsymbol{U} \boldsymbol{B}_m \boldsymbol{U}_K^\top + \\ &\sum_{m \in \mathcal{E}_p} \sum_{l \in \mathcal{E}_p} [\boldsymbol{R}_{zz}]_{m,l} \sigma_s^2 \boldsymbol{U} \boldsymbol{B}_m \boldsymbol{B}_l^\top \boldsymbol{U}^\top + \sigma_n^2 \boldsymbol{I}] \boldsymbol{G} - \boldsymbol{G}^\top [\sigma_s^2 \boldsymbol{U}_K + \sum_{m \in \mathcal{E}_p} \sigma_s^2 \mathbb{P}_m \boldsymbol{U} \boldsymbol{B}_m] \\ &- [\sigma_s^2 \boldsymbol{U}_K^\top + \sum_{m \in \mathcal{E}_p} \sigma_s^2 \mathbb{P}_m \boldsymbol{B}_m^\top \boldsymbol{U}^\top] \boldsymbol{G}\}. \end{aligned} \tag{3.29}$$

Since the MSE is a convex function of $\boldsymbol{G}$, the optimal vector is obtained by equating

the gradient of (3.27) respect to $\boldsymbol{G}$ to zero. The resulting optimal matrix turns out to be

$$\begin{aligned}\boldsymbol{G} = [\boldsymbol{U}_K\boldsymbol{U}_K^\top + \sum_{m\in\mathcal{E}_p}\sum_{l\in\mathcal{E}_p}[\boldsymbol{R}_{zz}]_{m,l}\boldsymbol{U}\boldsymbol{B}_m\boldsymbol{B}_l^\top\boldsymbol{U}^\top + \\ + \frac{\sigma_n^2}{\sigma_s^2}\boldsymbol{I} + \sum_{m\in\mathcal{E}_p}\mathbb{P}_m(\boldsymbol{U}_K\boldsymbol{B}_m^\top\boldsymbol{U}^\top + \boldsymbol{U}\boldsymbol{B}_m\boldsymbol{U}_K^\top)]^{-1} \\ \cdot(\boldsymbol{U}_K + \sum_{m\in\mathcal{E}_p}\mathbb{P}_m\boldsymbol{U}\boldsymbol{B}_m). \end{aligned} \tag{3.30}$$

If there are no perturbations, i.e. $\mathbb{P}_m = 0$, then $\boldsymbol{G} = \frac{1}{1+\frac{\sigma_n^2}{\sigma_s^2}}\boldsymbol{U}_K$, so that $\widehat{\boldsymbol{s}} = \boldsymbol{G}^\top\boldsymbol{x} = \frac{1}{1+\frac{\sigma_n^2}{\sigma_s^2}}\boldsymbol{U}_K^\top\boldsymbol{x}$.

As a numerical example, we consider a stochastic block graph model composed of 120 nodes, arranged in C clusters with inter-cluster probability equal to 10^{-4} and intra-cluster probability equal to 0.5. Each edge has a probability $\mathbb{P}_m$ to fail, assumed to be the same for all the edges. Over these graphs, we generated band-limited signals with bandwidth $K = C$. In Fig. 3.8 we compare the NMSE in the estimation of $\boldsymbol{s}$, computed as $\|\hat{\boldsymbol{s}} - \boldsymbol{s}\|_2^2/\|\hat{\boldsymbol{s}}\|_2^2$, as a function of the failure probability using two possible filters: a) the conventional approach, with $\boldsymbol{G}$ simply proportional to $\boldsymbol{U}_K$; b) the vector $\boldsymbol{G}$ as given by (3.30). The results are averages over 20 realizations of the graphs, and for each graph over $10,000$ realizations of the perturbation events. Form Fig. 3.8, we can see that the incorporation of the perturbation statistics yields a lower MMSE with respect to the conventional approach. As expected, the error is larger as the bandwidth increases, also because we need to estimate a vector with more unknowns. In the next section, we will show how to improve these results.

Finally, in Fig. 3.9, it is shown the MMSE vs $\mathbb{P}_m$ when $K = C = 3$ and the parameter $\mathbb{P}_m$ is underestimated. The assumed inexact parameter is denoted as $\mathbb{P}_m^*$, and we can see slightly larger MMSE values when $\mathbb{P}_m$ is incorrect with respect the optimized case with the true value of $\mathbb{P}_m$.

3.5 Failures detection

In this section, we address the challenging task of edge failure detection, by the observation of a graph signal. Assuming some prior information about the statistics of the signal vector $\boldsymbol{s}$ and of the perturbations, we build a Bayesian approach to decide whether some uncertain edges are present or not. More specifically, the goal is to recover the binary vector $\boldsymbol{z}$, whose dimension is equal to the number of possibly perturbed edges and whose m-th entry is one, if the m-th edge fail, or zero otherwise.

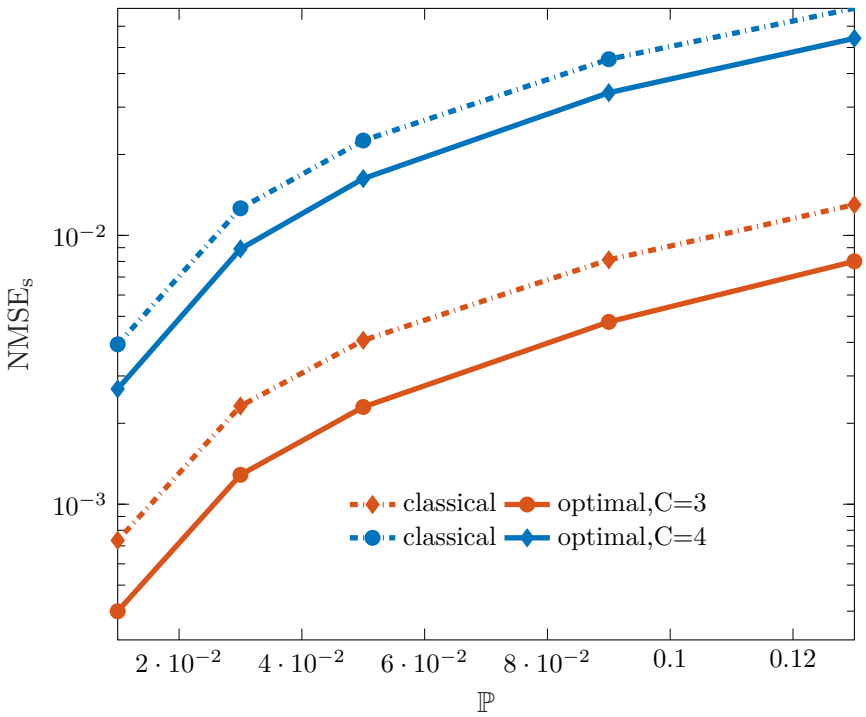


Figure 3.8. MMSE as a function of failure probability of four clusters graph's edges.

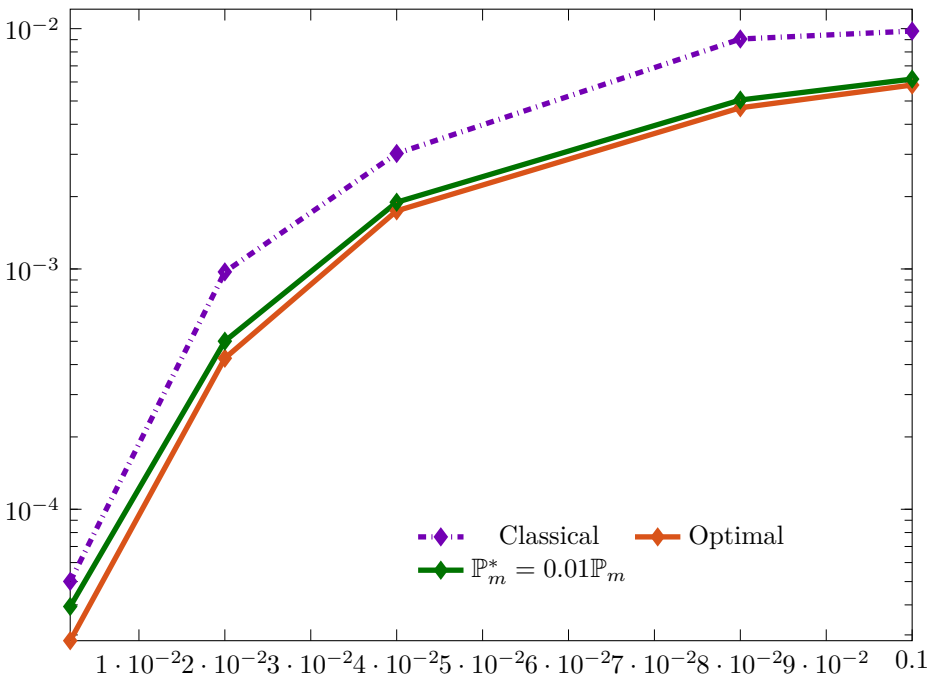


Figure 3.9. MMSE as a function of failure probability in case of uncertainty of the parameter $\mathbb{P}_m$.

For the sake of simplicity, we start considering the simple case where only one link fails, say the m-th link.
Let us consider the model (3.26), where $\boldsymbol{s} \in \mathbb{R}^K$ is now a Gaussian random vector whose joint probability density function is:

$$p(\boldsymbol{s}) = \frac{1}{(2\pi\sigma_s^2)^{K/2}} \exp(-\frac{1}{2\sigma_s^2} \|\boldsymbol{s}\|^2). \tag{3.31}$$

Moreover, we assume the noise vector $\boldsymbol{v}$ in (3.26) to be Gaussian, with zero mean and covariance matrix $\boldsymbol{R}_{vv} = \sigma_n^2 \boldsymbol{I}$. Assuming to have a priori information about the failure probability $\mathbb{P}_m$ of the m-th edge, we decide whether the edge m is present (i.e. $Z_m = 0$) or not (i.e. $Z_m = 1$). The probability of $Z_m = 1$, conditioned to the observed vector $\boldsymbol{x}$, can be written as:

$$\mathbb{P}(Z_m = 1|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|Z_m = 1)\mathbb{P}_m}{p(\boldsymbol{x})}. \tag{3.32}$$

To evaluate (3.32), we compute the marginal distribution $p(\boldsymbol{x}|Z_m = 1) = \int_{-\infty}^{+\infty} p(\boldsymbol{x}, \boldsymbol{s}|Z_m = 1)\, d\boldsymbol{s}$. Since $\boldsymbol{s}$ is statistically independent of Z_m, using $p(\boldsymbol{x}, \boldsymbol{s}|Z_m = 1) = p(\boldsymbol{x}|\boldsymbol{s}, Z_m = 1)p(\boldsymbol{s})$, we get

$$\begin{aligned} p(\boldsymbol{x}|Z_m = 1) = & \frac{1}{(2\pi\sigma_n^2)^{N/2}} \frac{1}{(2\pi\sigma_s^2)^{K/2}} \\ & \int_{-\infty}^{+\infty} \exp\{-\frac{1}{2}(\frac{\|\boldsymbol{x} - \boldsymbol{U}_K\boldsymbol{s} - \Delta\boldsymbol{U}_{K,m}\boldsymbol{s}\|^2}{\sigma_n^2} + \frac{\|\boldsymbol{s}\|^2}{\sigma_s^2})\} d\boldsymbol{s} \end{aligned} \tag{3.33}$$

with $\Delta\boldsymbol{U}_{K,m} := [\Delta\boldsymbol{u}_{1,m}, \ldots, \Delta\boldsymbol{u}_{K,m}]$, where $\Delta\boldsymbol{u}_{i,m}$ denotes the true perturbation of $\boldsymbol{u}_i$ when edge m fails. Defining $\bar{\boldsymbol{U}}_K = \boldsymbol{U}_K + \Delta\boldsymbol{U}_{K,m}$, $c_0 = \sigma_n^2 + \sigma_s^2$, and rewriting the argument of the exponential as $\frac{1}{\sigma_n^2} \|\mathbf{x} - \mathbf{U}_K\mathbf{s} - \Delta\mathbf{U}_{K,m}\mathbf{s}\|^2 + \frac{1}{\sigma_s^2} \|\boldsymbol{s}\|^2 = \frac{c_0}{\sigma_n^2\sigma_s^2} \left\|\boldsymbol{s} - \frac{\sigma_s^2}{c_0}\bar{\boldsymbol{U}}_K^\top\boldsymbol{x}\right\|^2 + \frac{1}{\sigma_n^2}(\boldsymbol{x}^\top\boldsymbol{x} - \frac{\sigma_s^2}{c_0}\boldsymbol{x}^\top\bar{\boldsymbol{U}}_K\bar{\boldsymbol{U}}_K^\top\boldsymbol{x})$ we are able to solve the integral in closed form, so that we have

$$p(\boldsymbol{x}|Z_m = 1) = \frac{\exp\{-\frac{1}{2\sigma_n^2}(\boldsymbol{x}^\top\boldsymbol{x} - \frac{\sigma_s^2}{c_0}\boldsymbol{x}^\top\bar{\boldsymbol{U}}_K\bar{\boldsymbol{U}}_K^\top\boldsymbol{x})\}}{(2\pi)^{\frac{N}{2}}(\sigma_n^2)^{\frac{N-K}{2}}c_0^{\frac{K}{2}}}. \tag{3.34}$$

Similarly, we can find the marginal distribution $p(\boldsymbol{x}|Z_m = 0)$ as

$$p(\boldsymbol{x}|Z_m = 0) = \frac{\exp\{-\frac{1}{2\sigma_n^2}(\boldsymbol{x}^\top\boldsymbol{x} - \frac{\sigma_s^2}{c_0}\boldsymbol{x}^\top\boldsymbol{U}_K\boldsymbol{U}_K^\top\boldsymbol{x})\}}{(2\pi)^{\frac{N}{2}}(\sigma_n^2)^{\frac{N-K}{2}}c_0^{\frac{K}{2}}}. \tag{3.35}$$

Then, using the maximum a posteriori criterion we decide weather the edge m is

present or not. Hence, from (3.32), the decision rule is the following

$$\frac{\mathbb{P}(Z_m = 1|\boldsymbol{x})}{\mathbb{P}(Z_m = 0|\boldsymbol{x})} = \frac{p(\boldsymbol{x}|Z_m = 1)\mathbb{P}_m}{p(\boldsymbol{x}|Z_m = 0)(1 - \mathbb{P}_m)} \underset{\text{present}}{\overset{\text{absent}}{\gtrless}} 1. \tag{3.36}$$

If the ratio is greater than one, we decide that the m-th edge failed, otherwise the topology remains unaltered. After some straightforward algebraic manipulations, the above decision rule becomes

$$\boldsymbol{x}^\top \left(\bar{\boldsymbol{U}}_K \bar{\boldsymbol{U}}_K^\top - \boldsymbol{U}_K \boldsymbol{U}_K^\top\right) \boldsymbol{x} \underset{\text{present}}{\overset{\text{absent}}{\gtrless}} \frac{1}{\alpha} \ln\left(\frac{1 - \mathbb{P}_m}{\mathbb{P}_m}\right) \tag{3.37}$$

with $\alpha := \sigma_s^2/[2\sigma_n^2(\sigma_s^2 + \sigma_n^2)]$.

The above approach can be generalized to the case of multiple uncertain edges; i.e. multiple edges in the nominal graph that may fail. We denote by $\mathcal{E}_p \subseteq \mathcal{E}$ the set of potentially failing edges. Generalizing the previous approach, we need to compute the posterior pdf $p(\mathbf{Z} = \boldsymbol{z}_\ell|\boldsymbol{x})$, where $\boldsymbol{z}_\ell$ is a binary vector of size $|\mathcal{E}_p|$, whose entries are 1 or 0, if the corresponding edge failed or not. In this case, we want to identify the configuration of $\boldsymbol{z}_\ell$ that maximizes the posterior probability

$$p(\mathbf{Z} = \boldsymbol{z}_\ell|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z}_\ell) \prod_{m \in \mathcal{E}_p} \mathbb{P}_m^{z_\ell(m)} (1 - \mathbb{P}_m)^{1 - z_\ell(m)}}{p(\boldsymbol{x})}. \tag{3.38}$$

The recovery method we proposed is tested over the graph of Fig. 3.10, for the case of 2 or 4 failing edges, with $\mathbb{P}_m = 0.5, \forall m \in \mathcal{E}_p$. For any configuration of failing edges, in this experiment, we computed the exact perturbation of the Laplacian eigenvectors numerically. The results, expressed in terms of Failures Identification Error Rate (FIER), measuring the percentage of incorrect identification of failing links, are reported in Fig. 3.11, as a function of the signal to noise ratio $10 \log_{10}(\sigma_s^2/\sigma_n^2)$. We considered separately the cases where the failure involves only intra-cluster edges (solid line) or inter-cluster edges (dashed line). From Fig. 3.11, we can see that, if the signal to noise ratio is sufficiently large, so that the perturbations due to edge failure can be distinguished from the additive noise, the method is able to provide a fairly low error rate. Moreover, the method is more reliable if the signal bandwidth increases. An intuitive explanation is that, as the number of signal coefficients increases, i.e. K, the sensitivity to the perturbation increases as well, thus facilitating the detection of some edge failure. Finally, note that, if we perturb only the inter-cluster edges, the error rate (dashed line) is smaller. In fact, as we know from our perturbation analysis, the perturbation over the inter-cluster edges has a larger impact on the Laplacian eigenevectors so that the mismatch between the observation and the model is easier to detect.

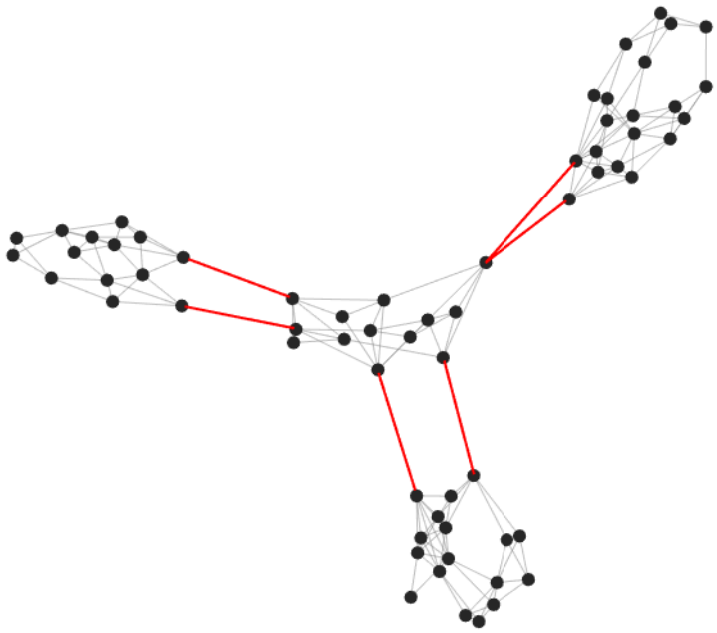


Figure 3.10. Clustered graph.

We also tested our recovery method over the graph of Fig. 3.10, for the case of 4 failing inter-cluster edges, with $\mathbb{P}_m = 0.4, \forall m \in \mathcal{E}_p$. We considered in the decision test (3.37) both the *exact* eigenvector perturbations and the *approximate* expressions, given by (3.7). The results are reported in Fig. 3.12 and Fig. 3.13 , as a function of the signal to noise ratio $10\log_{10}(\sigma_s^2/\sigma_n^2)$. We considered the case where the failure involves only inter-cluster edges, which are indeed the ones having more impact on performance. From Fig. 3.12, we can see that, if the signal to noise ratio is sufficiently large, the Bayesian approach is able to recover a good number of edges. The performance is better for the decision method using the exact eigenvectors, as expected, but that method suffers from a complexity issue, because it needs to pre-compute all possible eigenvector perturbations resulting from each set of altered edges, and thus it has a combinatorial complexity. Conversely, the method based on our approximate expressions (3.7) does not suffer from such a combinatorial issue, but it pays this advantage with a floor on the error rate, as evidenced in Fig. 3.12.

Building on the decision rule (3.37), we devised an algorithm that improves the accuracy of the GFT, in the presence of uncertainties. The algorithm takes a decision about the uncertain edges, using (3.37). Then, it builds the corresponding Laplacian matrix, computes the corresponding eigenvectors and uses them to compute the GFT of the observed signal. The resulting MSE, in the presence of noise, is reported in Fig. 3.13, where we show the MSE obtained using the GFT built using: a) the eigenvectors of the nominal graph; b) the eigenvectors of the Laplacian

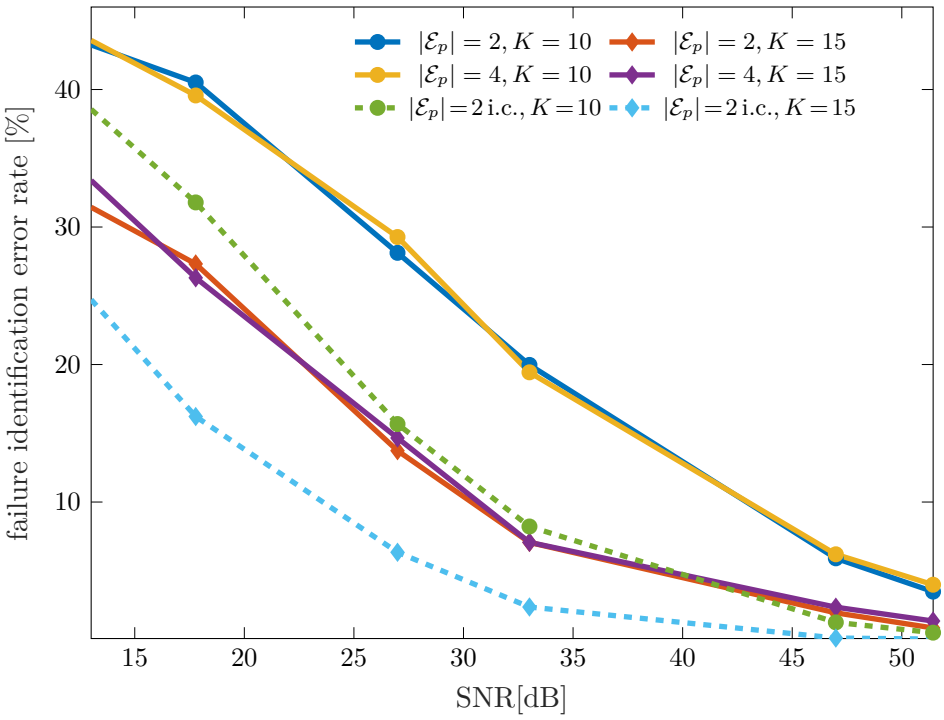


Figure 3.11. FIER (%) vs SNR.

refined through the Bayesian approach using the approximated eigenvectors; c) the eigenvectors built using (3.7), exploiting the edge refinement obtained with the Bayesian approach using the exact eigenvectors. We can check from Fig. 3.13 the advantage of the Bayesian approach in improving the accuracy of the GFT, at high SNR. The results have been obtained by averaging over 10^4 independent realizations of perturbations events, noise and signal $\boldsymbol{s}$.

3.6 Robust information transmission over wireless ad hoc networks

In this section, we apply our statistical model to optimize the resource (power) allocation over an ad hoc wireless network, vulnerable to random link failures. This analysis is motivated by the fact that one of the main issue in wireless ad hoc networks is to guarantee their connectivity in the presence of link failures. The aim is to allocate resources, i.e. transmit power, in order to design a network robust against random link failures, where robustness is assessed in terms of network connectivity. We consider a wireless ad hoc network where links are subject to random failures due to fading. Denoting with h_m the channel coefficient, the capacity over the m-th link assumes the form $C_m = \log_2(1 + |h_m|^2 \rho_m)$ (bits/sec/Hz), where $\rho_m = \frac{P_{T,m}}{\sigma_n^2 r_m^2}$ is the

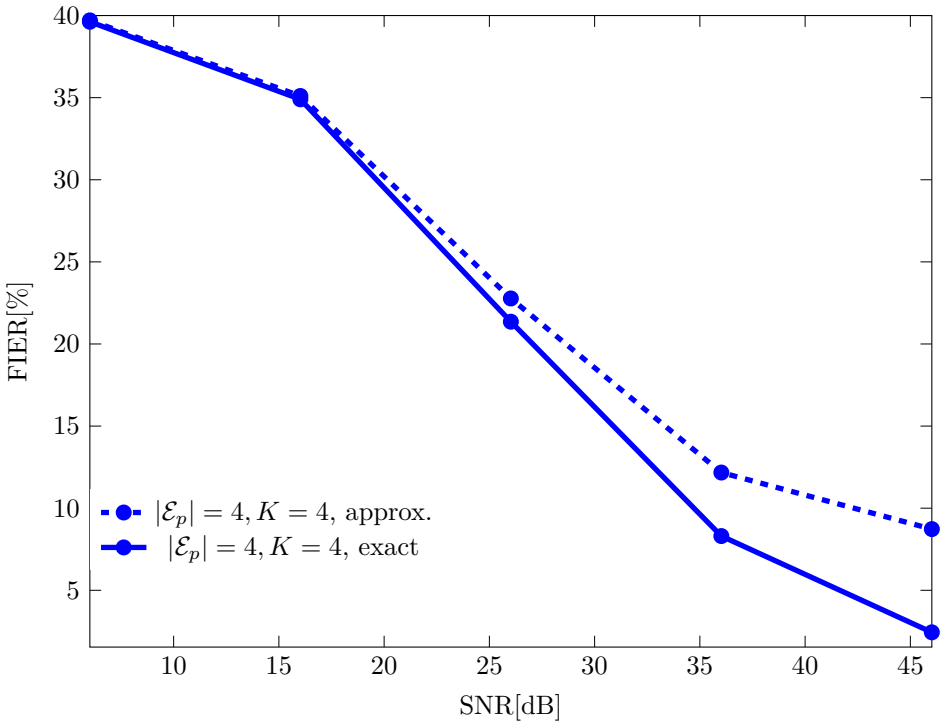


Figure 3.12. FIER (%) vs SNR: exact perturbations (solid) and approximate expressions (dashed).

signal-to-noise ratio (SNR), $P_{T,m}$ is the transmitted power over the m-th link, σ_n^2 is the noise variance, and r_m is the distance covered by link m. We suppose the failure events over different links to be statistically independent. We start considering a single antenna at both transmit and receive sides. In the absence of a Line-of-Sight (LoS) path, this scenario gives rise to a single-input-single-output (SISO) Rayleigh flat fading channel. In such a case, the channel coefficient h_m is a complex circularly symmetric Gaussian r.v. with zero mean. Hence, the random variable $\alpha = |h_m|^2$ has an exponential cumulative distribution function (CDF), which we denote as $F_1(\alpha;\gamma)$, where, more generally, $F_n(x,\gamma)$ denotes the CDF of a gamma r.v. x of order n and parameter γ.

Denoting the the data rate used over the m-th link as R_m, the link is in outage if $R_m > C_m$. The outage probability $\mathbb{P}_m$ over link m can then be computed as

$$\begin{aligned}\mathbb{P}_m &= Pr\{C_m < R_m\} \qquad (3.39)\\ &= Pr\{\log_2(1+|h_m|^2\rho_m) < R_m\} = Pr\left\{|h_m|^2 < \frac{2^{R_m}-1}{\rho_m}\right\}\\ &= \int_0^{\frac{2^{R_m}-1}{\rho_m}} \gamma e^{-\gamma\alpha} d\alpha = F_1\left(\frac{2^{R_m}-1}{\rho_m};\gamma\right) = 1 - e^{-\frac{\gamma}{\rho_m}(2^{R_m}-1)}.\end{aligned}$$

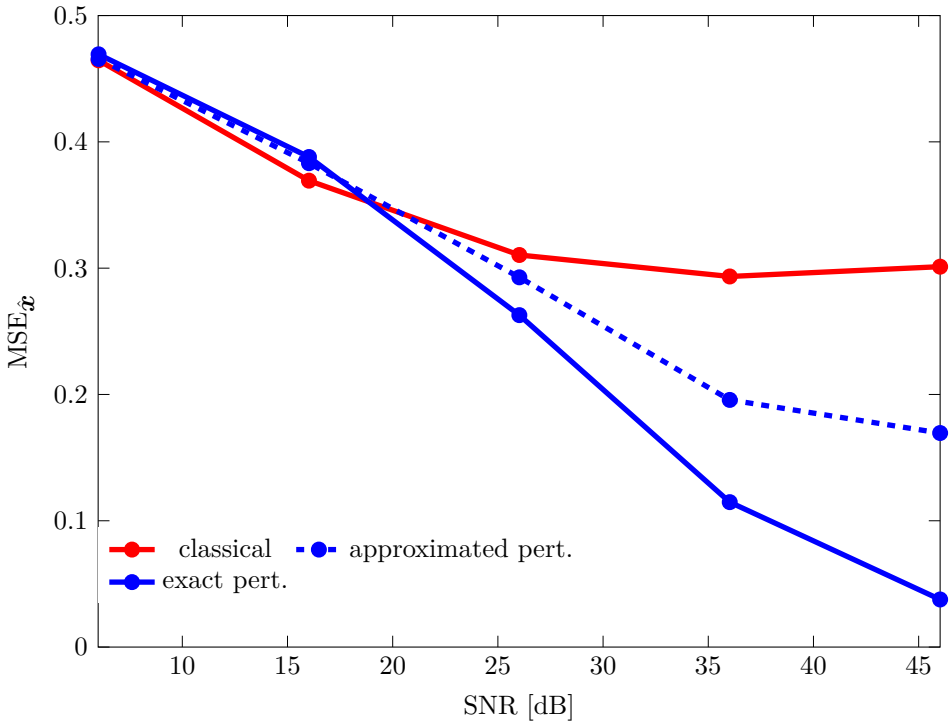


Figure 3.13. MSE of $\hat{\boldsymbol{x}}$ vs SNR: classical GFT (red); Bayesian approach with exact perturbations (, solid) and with approximated expressions (, dashed).

This expression can be inverted to derive the transmit power $P_{T,m}$ as a function of the outage probability over the m-th link

$$P_{T,m} = -\frac{\gamma\,\sigma_n^2\,r_m^2(2^{R_m}-1)}{\ln(1-\mathbb{P}_m)} = \frac{\sigma_n^2\,r_m^2(2^{R_m}-1)}{F_1^{-1}(\mathbb{P}_m;\gamma)}. \tag{3.40}$$

We assess the network robustness as the ability of the network connectivity to be only slightly affected by the failure of a small number of edges. We measure connectivity of the network by using the value of the second smallest eigenvalue of the Laplacian matrix associated with the network. In fact, this parameter, called also *algebraic connectivity*, provides a lower bound for the graph conductance [2]. The algebraic connectivity becomes a random variable, in a network vulnerable to random link failures. We want to determine here the transmit powers $P_{T,m}$, or equivalently the outage probabilities $\mathbb{P}_m$, through (3.40), that minimize the average perturbation of the algebraic connectivity, subject to a cost associated with the total transmit power $P_{T_{max}}$ necessary to establish all the communication links. To formulate the problem, we rely on the small perturbation analysis derived in the previous sections. Hence, the outage probabilities $\mathbb{P}_m$ can be found as the solution of the following

optimization problem

$$\begin{array}{ll} \min\limits_{\{\mathbb{P}_m\}_{m=1}^E} & \sum\limits_{m\in\mathcal{E}} \mathbb{E}\{|\delta\lambda_{2,m}|\} \\ \text{s.to} & \sum\limits_{m\in\mathcal{E}} P_{T,m} \leq P_{T_{max}} \\ & \mathbb{P}_m \in [0,1], \forall m \in \mathcal{E} \end{array} \tag{3.41}$$

where $\delta\lambda_{2,m}$ denotes the approximated expression of the perturbation of the second smallest eigenvalue due to the failure of edge m. Using equations (3.15), (3.6) and (3.40), we can rewrite the optimization problem in terms of the outage probabilities $\mathbb{P}_m$ as

$$\begin{array}{ll} \min\limits_{\{\mathbb{P}_m\}_{m=1}^E} & \sum\limits_{m\in\mathcal{E}} \mathbb{P}_m[u_2(v_{1,m}) - u_2(v_{2,m})]^2 \\ \text{s.to} & \sum\limits_{m\in\mathcal{E}} \frac{r_m^2\left(2^{R_m}-1\right)}{F_1^{-1}(\mathbb{P}_m;\gamma)} \leq \Gamma_{max} \\ & \mathbb{P}_m \in [0,1],\ \forall\, m \in \mathcal{E} \end{array} \tag{3.42}$$

where $\boldsymbol{u}_2$ is the unit length eigenvector associated with the second smallest eigenvalue of the nominal network Laplacian matrix and we set $\Gamma_{max} := P_{T_{max}}/\sigma_n^2$.
Problem (3.42) is non-convex because of the constraint set, which is not convex. If we perform the change of variable $t_m := 1/F_1^{-1}(\mathbb{P}_m;\gamma) = -\gamma/\ln(1-\mathbb{P}_m), m = 1,\ldots,E$, the first constraint set gives rise to a hyperplane, and then it is convex, but the objective function $F_1(\frac{1}{t_m};\gamma)|\delta\lambda_{2,m}| = \sum\limits_{m\in\mathcal{E}}(1-e^{-\frac{\gamma}{t_m}})|\delta\lambda_{2,m}|$ becomes non-convex. However, if we further constrain the unknown variables t_m to satisfy $t_m \geq \gamma/2, \forall\, m$, then the objective function becomes convex. Such a constraint on t_m is not a real limitation, because it is equivalent to require that $\mathbb{P}_m < 0.8647$, but this is perfectly reasonable because in real applications we want the outage probabilities to be well below such a value.

The original problem can be recast as

$$\begin{array}{ll} \min\limits_{\mathbf{t}} & \sum\limits_{m\in\mathcal{E}} (1-e^{-\frac{\gamma}{t_m}})[u_2(v_{1,m}) - u_2(v_{2,m})]^2 \\ \text{s.to} & \sum\limits_{m\in\mathcal{E}} r_m^2\left(2^{R_m}-1\right) t_m \leq \Gamma_{max} \\ & t_m \geq \frac{\gamma}{2}, \quad \forall m \in \mathcal{E}, \end{array} \tag{3.43}$$

where $\boldsymbol{t} := [t_1,\ldots,t_E]^\top$.

In the following, we generalize to a Multi-Input Multi-Output (MIMO) system. In such a case, there are n_T antennas at the transmit side and n_R antennas at the receive side in each link. Denoting by h_{ij}^m the coefficients between the i-th transmit and the j-th receive antenna of the m-th link, and assuming statistically independent Rayleigh fading channels, a full diversity receive scheme gives rise to

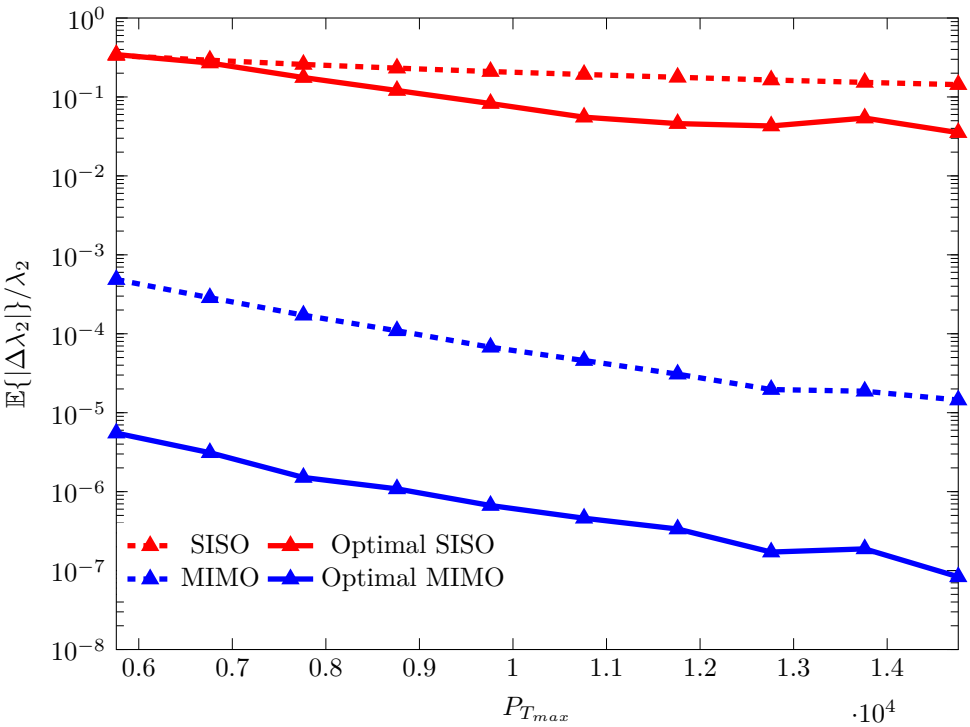


Figure 3.14. Comparisons between SISO (red curves) and MIMO ($n = 4$) systems (curves), with and without optimization.

an equivalent channel coefficient whose square modulus is $\alpha := \sum_{i=1}^{n_T} \sum_{j=1}^{n_R} |h_{ij}^m|^2$. This random variable is characterized by a Gamma CDF of order $n = n_T \times n_R$, i.e. $F_n(\alpha; \gamma)$ [108]. Thanks to a MIMO, instead of SISO, system it is possible to get a diversity gain that makes the communication links more reliable. We wish now to see how this diversity gain can affect the robustness of the ad hoc network, in terms of connectivity. Generalizing the approach followed above in the SISO case, we can formulate the following optimization problem as

$$\min_{\mathbf{t}} \quad \sum_{m \in \mathcal{E}} F_n\left(\frac{1}{t_m}; \gamma\right) |\delta\lambda_{2,m}| \tag{3.44a}$$

$$\text{s.to} \quad \sum_{m \in \mathcal{E}} r_m^2 (2^{R^m} - 1) t_m \leq \Gamma_{max} \tag{3.44b}$$

$$t_m \geq \gamma/(n+1), \quad \forall m \in \mathcal{E}. \tag{3.44c}$$

The condition in (3.44c) is used, again, to make the problem convex. Problem (3.43) is indeed a special case of problem (3.44), with $n = 1$. Note that, the bounding region in (3.44c) increases with the number of independent channels.

As a numerical example, we consider the sensor network deployed in the Intel Berkeley Research lab [109], composed of 54 nodes. The network contains two

clusters, associated with two separated spaces in the laboratory. Given two nodes, we assume there is a link between them if their Eucledian distance is less than $\tau = 10$ meters. Then, we assumed the presence of two inter-cluster edges, to ensure network connectivity. For this simulation, we set $\sigma_n^2 = 10^{-4}$, $R_m = 7$, $n = 4$ and $\gamma = 1$.

Fig. 3.14 shows the expected value of the perturbation of the algebraic connectivity computed numerically, normalized with respect to the nominal value λ_2, as a function of $P_{T_{max}}$. We compare the perturbation using the optimal power allocation, as a solution of (3.44), with the solution where the same power is allocated over all the links, under the same overall power consumption $P_{T_{max}}$. The comparison is performed for both SISO and MIMO cases. The simulations are averaged over 10^6 independent realizations of the perturbation events. From Fig. 3.14, we can observe the gain in terms of the total power necessary to achieve the same expected perturbation of the network algebraic connectivity. We can also see the advantage of using MIMO communications, at least in the case of statistically independent links.

3.7 Impact of uncertainties on semi-supervised and supervised learning

In this section, we asses topology uncertainty effects on graph-based learning tasks such as semi-supervised learning and clustering method to be used in unsupervised learning.

3.7.1 Semi-supervised learning over uncertain graphs

A well-known approach in graph-based semi-supervised learning methods is to propagate the known labels throughout a graph representing similarities among data points of a point cloud [55]. However, if the graph presents uncertainties on some edges, these uncertainties reflect into the final label assignment to the unlabeled nodes. This motivates the analysis of the present section, where we show how the uncertainty on some edges affects the graph-based semi-supervised learning process and then the final labels. We assume that only a small percentage of edges is perturbed. For the sake of simplicity, we consider the case where the perturbation consists only in possible failing links, but our framework can of course be extended including also the appearance of a small percentage of edges. Given a graph, we suppose that N_l nodes have been labeled, whereas N_u are unlabeled. We denote by $\boldsymbol{y} \in \mathbb{R}^{N_l}$ the vector collecting the given labels. The goal of semi-supervised learning is to label the unlabeled nodes exploiting the knowledge on the graph structure and

the labeled nodes. One possible way consists is solving the following problem

$$\min_{\boldsymbol{f}} \quad \boldsymbol{f}^\top \mathbf{L} \boldsymbol{f} \tag{3.45a}$$

$$\text{s.to} \quad f(x_i) = y_i, \quad \forall i = 1, \ldots, N_l, \tag{3.45b}$$

where $\boldsymbol{f} = [f(x_1), \ldots, f(x_{N_u+N_l})]^\top$ is the vector of labels associated with all the vertices. Assuming, without loss of generality, that the vertices are ordered in such a way that the first N_u vertices correspond to the unlabeled nodes, we can partition the Laplacian matrix as follows

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{uu} & \mathbf{L}_{ul} \\ \mathbf{L}_{ul}^\top & \mathbf{L}_{ll} \end{bmatrix} \tag{3.46}$$

where $\mathbf{L}_{uu}$ ($\mathbf{L}_{ll}$) is the matrix composed of N_u (N_l) rows and N_u (N_l) columns of $\mathbf{L}$ related to unlabeled nodes (labeled nodes), and $\mathbf{L}_{ul}$ is the matrix composed of N_u rows of $\mathbf{L}$ related to unlabeled nodes, and N_l columns of $\mathbf{L}$ related to labeled nodes. Vector $\boldsymbol{f}$ correspondingly is ordered as $\boldsymbol{f} = [\boldsymbol{f}_u, \boldsymbol{f}_l]^\top$, where $\boldsymbol{f}_u$ contains the unlabeled nodes and $\boldsymbol{f}_l$ the labeled ones. Problem (3.45) has a known closed-form solution, also known as the harmonic solution, [55, 110]:

$$\boldsymbol{f}_u = -\mathbf{L}_{uu}^{-1} \mathbf{L}_{ul} \, \boldsymbol{y}_l. \tag{3.47}$$

Consider now that the given graph topology is affected by uncertainties on some edges, i.e. some existing links may not be present. Since a topological perturbation yields a perturbed Laplacian matrix $\mathbf{L} + \Delta\mathbf{L}$, with $\Delta\mathbf{L} = \sum_{m \in \mathcal{E}_p} Z_m \Delta\mathbf{L}^{(m)}$, it is clear from (3.47) that a perturbation has effects also on the final labels. If the submatrices $\mathbf{L}_{uu}$ and $\mathbf{L}_{ul}$ are perturbed by terms $\Delta\mathbf{L}_{uu}$ and $\Delta\mathbf{L}_{ul}$, respectively, the vector $\boldsymbol{f}_u$ gets perturbed as follows

$$\begin{aligned} \widehat{\boldsymbol{f}}_u &= -(\mathbf{L}_{uu} + \Delta\mathbf{L}_{uu})^{-1} (\mathbf{L}_{ul} + \Delta\mathbf{L}_{ul}) \, \boldsymbol{y}_l \\ &\approx -(\mathbf{L}_{uu}^{-1} - \mathbf{L}_{uu}^{-1} \Delta\mathbf{L}_{uu} \mathbf{L}_{uu}^{-1})(\mathbf{L}_{ul} + \Delta\mathbf{L}_{ul}) \, \boldsymbol{y}_l \end{aligned} \tag{3.48}$$

where we retained only the first order approximation terms. Therefore, vector $\boldsymbol{f}_u$ gets perturbed by a quantity that is approximately equal to

$$\delta \boldsymbol{f}_u \approx \mathbf{L}_{uu}^{-1} \left[\sum_{m \in \mathcal{E}_p} Z_m \Delta\mathbf{L}_{uu}^{(m)} \mathbf{L}_{uu}^{-1} \mathbf{L}_{ul} - \sum_{m \in \mathcal{E}_p} Z_m \Delta\mathbf{L}_{ul}^{(m)} \right] \boldsymbol{y}_l. \tag{3.49}$$

Since the perturbation variables Z_m are random, the error is random as well. Thus, within the limits of validity of first order approximations, the mean value of the

error is

$$\mathbb{E}\{\delta \boldsymbol{f}_u\} = \mathbf{L}_{uu}^{-1}\left[\sum_{m\in\mathcal{E}_p} \mathbb{P}_m \Delta\mathbf{L}_{uu}^{(m)}\mathbf{L}_{uu}^{-1}\mathbf{L}_{ul} - \sum_{m\in\mathcal{E}_p} \mathbb{P}_m \Delta\mathbf{L}_{ul}^{(m)}\right]\boldsymbol{y}_l. \tag{3.50}$$

To write compactly the error covariance matrix, it is useful to introduce some matrices. Let $\boldsymbol{G} = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_{|\mathcal{E}_p|}]$, where $\boldsymbol{g}_m := \mathbf{L}_{uu}^{-1}\Delta\mathbf{L}_{uu}^{(m)}\mathbf{L}_{uu}^{-1}\mathbf{L}_{ul}\boldsymbol{y}_l$ and $\boldsymbol{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_{|\mathcal{E}_p|}]$, where $\mathbf{h}_m := \mathbf{L}_{uu}^{-1}\Delta\mathbf{L}_{ul}^{(m)}\boldsymbol{y}_l$. The covariance matrix of $\delta\boldsymbol{f}_u$ can then be written as

$$\mathrm{Cov}\{\delta\boldsymbol{f}_u\} = (\boldsymbol{G} - \boldsymbol{H})\mathbf{C}_{zz}(\boldsymbol{G} - \boldsymbol{H})^\top, \tag{3.51}$$

where $\mathbf{C}_{zz} = \mathrm{diag}(\mathbb{P}_1(1-\mathbb{P}_1), \ldots, \mathbb{P}_{|\mathcal{E}_p|}(1-\mathbb{P}_{|\mathcal{E}_p|}))$.

As a numerical example to assess the impact of edge failures on the label propagation, we generated a graph of $N = 50$ nodes, composed of two RGGs of 25 nodes each, having the same mean degree, inter-connected with four inter-cluster edges randomly placed. We assume two classes associated with two different RGGs, labeled either 1 or -1, as representative of the two classes. We assigned a label 1 to some points at random within one cluster and the other label -1 to some other points at random within the other cluster. Then, we perturbed the graph by assigning a binary r.v. to each edge is a subset $\mathcal{E}_p \in \mathcal{E}$, each r.v. is equal to one, with probability $\mathbb{P}_m$, or zero otherwise. For simplicity, we assumed $\mathbb{P}_m = \mathbb{P}, \forall m \in \mathcal{E}_p$. We applied (3.47) over several independent realizations of the graph topology and then computed the sum of the error variances obtained by simulation or by using the formula in (3.51). We considered only the error variances, since by numerical simulations we observed that the bias is typically negligible with respect to the standard deviation. Fig. 3.15 depicts the comparison of the sum of the variances on the final labels obtained by numerical simulations (dashed lines) and by our theoretical derivations (solid line), as a function of the mean degree of each cluster. The simulations are implemented assuming fixed number $|\mathcal{E}_p| = 200$ of uncertain edges, for every value of mean degree. The results in Fig. 3.15 have been obtained by averaging over $10,000$ independent realizations of the perturbation events. Fig. 3.15 shows, as expected, that the theoretical results become closer and closer to the simulation results as the mean degree increases and as the number of initial labels increases.

Building on our perturbation model, we address now the following question: If we have some degrees of freedom in the selection of the vertices to which we assign the initial labels, is there a way to optimize the location of these labels? Previous analysis is instrumental to address this question. We propose the following approach: We start with an initial set $\mathcal{M}$ of 2 labeled nodes, one per class. Then, we follow a greedy iterative strategy that adds, at each iteration, a new node as the node

that yields the minimum error variance, according to (3.51). Then, we repeat this procedure iteratively until the cardinality of $\mathcal{M}$ is equal to M. The aim of this greedy algorithm is to select the optimal labels positions that are robust to graph uncertainties. As numerical example to test the effectiveness of our greedy strategy, we generated a graph of $N = 50$ nodes and composed by two RGGs connected with four inter-cluster edges placed at random. Each RGG is associated with a different label (e.g., 1 or -1). We perturbed the graph assigning to each edge a binary r.v., equal to one, with probability $\mathbb{P}_m$ (failure probability), and zero otherwise. For sake of simplicity, we set $\mathbb{P}_m := \mathbb{P}, \forall m$. Then, we find the labels over the unlabeled nodes with (3.47). The first two label are assigned, in each cluster, to the node with the highest degree. Then, the successive label positions are chosen according to two different strategies: a) our perturbation-aware greedy strategy proposed before; b) a random strategy. In Fig. 3.16, we show the sum of the error variances computed numerically as a function of the number of labeled nodes M for the two strategies. The results have been obtained by averaging over $10,000$ independent realizations. Note from Fig. 3.16 that our proposed strategy yields a significant reduction of the overall variance with respect to the random approach, thus leading to a more robust allocation of the labels. This analysis suggests that, whenever possible, it makes perfect sense to optimize the location of the labels within each cluster.

3.7.2 Analysis of clustering over uncertain graph

In this section, we assess the effect of random topological perturbations on clustering, exploiting our small perturbation framework. As recalled in Sec. 2.3, spectral clustering methods exploit the structure of the Laplacian eigenvectors to detect clusters [18]. In particular, when the graph presents C clusters, clustering is based on the eigenvectors associated with C smallest eigenvalues of the Laplacian matrix [18]. However, if $\mathbf{L}$ is uncertain, this uncertainty reflects into its eigenvectors' structure and then on clustering. Modeling the presence of an edge as a binary r.v., we can compute the probability whether a node belongs to a cluster or not [18]. For the sake of simplicity and with the aim of finding, albeit approximate, closed form expressions, we consider the case of $C = 2$, i.e. connected graph composed of two clusters. In such a case, we only need to consider the perturbation $\delta\boldsymbol{u}_2$ of the eigenvector associated with the second smallest eigenvalue of $\mathbf{L}$. Then, we want to exploit our perturbation analysis to derive, although in approximate form, the probability of erroneous clustering due to random perturbation of a few edges. When the graph is composed of only two clusters, a possible way to cluster the graph entails looking at the sign of the eigenvector $\boldsymbol{u}_2$ associated with the second smallest eigenvalue of $\mathbf{L}$ (cf. Sec. 2.3). The sign of the i-th entry of $\boldsymbol{u}_2$ is used to decide

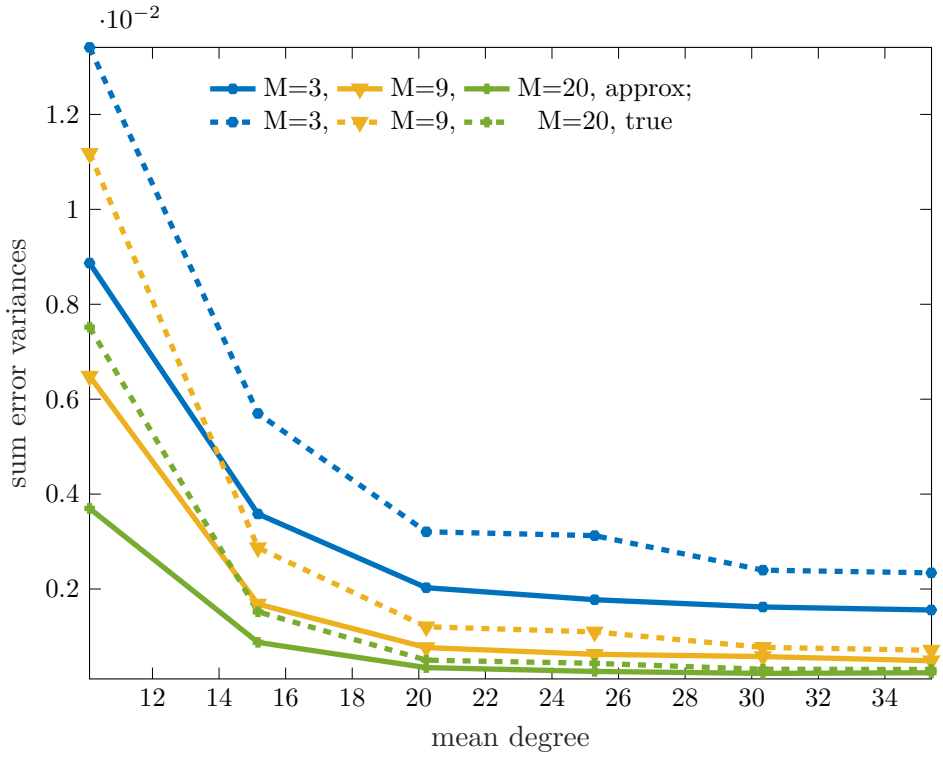


Figure 3.15. Trace of the covariance matrix as a function of cluster mean degree, for $\mathbb{P} = 0.01$, and for different number of labeled nodes.

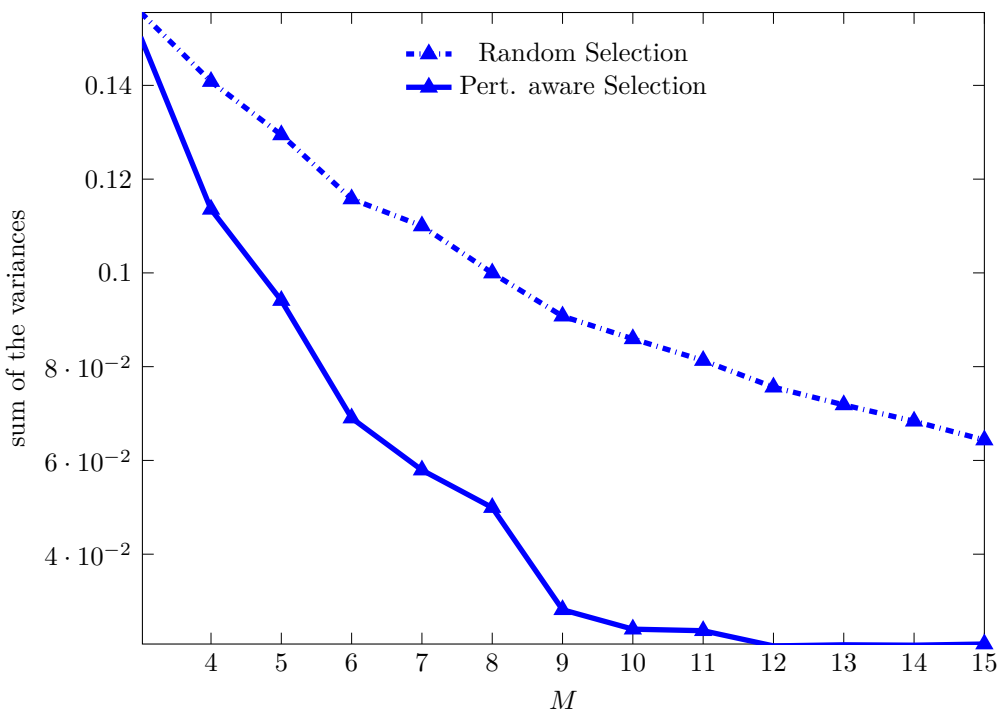


Figure 3.16. Trace of the covariance matrix as a function of number of labeled nodes M, with $\mathbb{P} = 0.09$.

the assignment of the i-th node to either cluster 1 or 2. In particular, let v_i be a given node, v_i is assigned to cluster 1 if $\boldsymbol{u}_2(v_i) \leq 0$, or to cluster 2, if $\boldsymbol{u}_2(v_i) \geq 0$. A vertex v_i is erroneously labeled when the perturbation $\Delta u_{2,m}(v_i)$ induced by the random drop (or appearance) of edge m causes a perturbation on $u_2(v_i)$ that makes its sign to flip. This is possible when $\Delta u_{2,m}(v_i)$ and $u_2(v_i)$ have opposite sign and $|\Delta u_{2,m}(v_i)| > |u_2(v_i)|$. In formulas, a mislabeling occurs if

$$\frac{\Delta u_{2,m}(v_i)}{u_2(v_i)} \leq -1. \tag{3.52}$$

If the perturbation may occur over multiple links (either dropping or appearing), within the first order of approximation, the overall perturbation is approximately equal to the sum of the perturbations caused by each single link. From the approximations developed in Sec. 3.2, we can say that the overall perturbation on the eigenvectors is approximately equal to the sum of the eigevectors perturbation for each m. This allows us to write the following probability for the occurrence of a labelling error

$$Pr\Big\{\sum_{m\in\mathcal{E}_p} \frac{Z_m \delta u_{2,m}(v_i)}{u_2(v_i)} + \sum_{m\in\bar{\mathcal{E}}_p} \frac{Z_m \delta u_{2,m}(v_i)}{u_2(v_i)} \leq -1\Big\}. \tag{3.53}$$

where we used the approximated expression $\delta u_{2,m}$, instead of $\Delta u_{2,m}$. However, it is not simple to derive this probability. Hence, we can use the Hoeffding's bound derived in Sec. 3.2.3, see e.g., (3.17), to derive an upper bound of the above probability. As a numerical result, we consider the graph of the IEEE 118 Bus Test Case, which represents a portion of the Electric Power System in the Midwestern United States [111]. The nodes of the graph are 118 and represent the buses, while the links are the transmission lines connecting the buses. We assume that the graph topology is subject to failures of the 5% of the links, i.e. $|\mathcal{E}_p| = 9$ over 179 links and $|\bar{\mathcal{E}}_p| = 0$, and that such perturbation can occur in random locations. The color of each node in Fig. 3.17 encodes a value associated with the error probability: the darker is the node, the higher is the probability that the node is incorrectly associated with the wrong cluster. In particular, the top side figure represents the Hoeffding's bound and, as a benchmark, the bottom side figure represents the error probability as measured by numerical simulations. Note that, for the given perturbation probabilities, most of the nodes in two clusters are correctly identified (see, e.g. the white nodes). There are however a few nodes, the few darker ones, that are much more sensitive to edge perturbations. Despite all the approximations and bounding, the comparison of the two figures shows that our theoretical analysis is able to predict quite well which are the nodes with the highest clustering error

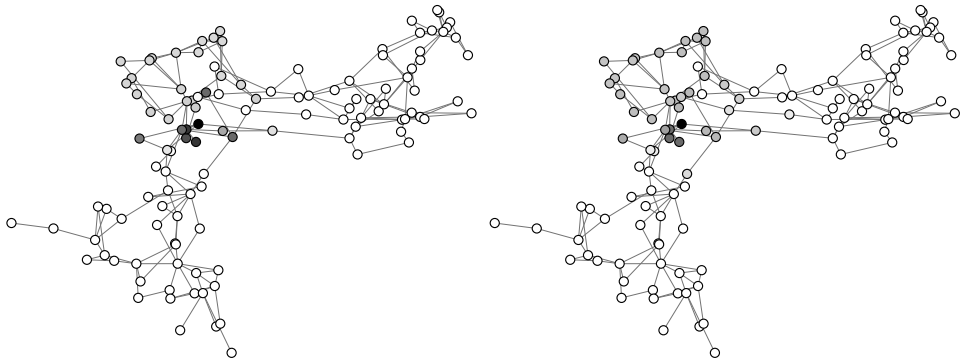


Figure 3.17. Clustering error sensibility, for C=2, computed with the bound (top) and numerically (bottom).

probability.

3.8 Conclusion

In summary, in this chapter, we have examined the impact of uncertainties on the graph topology. Most specifically, building on approximate closed form expressions of the graph Laplacian eigen-decomposition perturbation, we have proposed new graph signal recovery strategies incorporating our imperfect knowledge of the graph. We have also suggested methods to detect possible graph topology alterations with respect to a *nominal* graph, based on the observation of a graph signal assumed to be a random signal with assigned pdf. Moreover, we have assessed the impact that an imperfect knowledge of the graph may have on semi-supervised or on unsupervised learning. Despite the approximation of derived closed form expression for the eigenvalue/eigenvector perturbation due to alteration (suppression/appearance) of some edges, such expressions have been useful to devise new learning robust to the perturbation. In fact, these strategies have been shown to outperform conventional approaches that are unaware of possible topology mismatching between the assumed nominal graph and the real graph.

Several extensions are possible. First of all, we have considered binary graphs, however, our setting could be extended for weighted graphs, where small perturbations could be a small variation of edge weights. In our analysis, we focused, for simplicity, on the case where the Laplacian eigenvalues are all distinct, but it would be useful to extend the analysis to graphs whose eigenvalues have multiplicity greater than one, because these multiplicities sometimes reveal interesting symmetry properties. Furthermore, in our analysis of the impact of perturbations on cluster analysis, we considered, for simplicity, only the two-cluster case, but it would be useful to extend that approach to the multi-cluster case. Furthermore, the graph

perturbation model can also be useful to analyze the case where we observe a dynamic signal over a time-varying graph, as we starting exploring in [112].

3.9 Appendix

In this Section, we prove first Theorem 1, then we provide a small perturbation analysis to derive closed form expressions for the eigenvalues/eigenvector pairs of the perturbed Laplacian.

Proof. We now prove Theorem 1. From (3.4), the eigenvalues of $\tilde{\mathbf{L}}$ must be the solution of the following equation

$$|\mathbf{\Lambda}_{N-1} - \tilde{\lambda}\boldsymbol{I} + \sigma_m \boldsymbol{z}_m \boldsymbol{z}_m^\top| = 0. \tag{3.54}$$

If $u_i(v_{m_1}) = u_i(v_{m_2})$, then $z_m(i) = 0$. As a consequence, if we set $\tilde{\lambda} = \lambda_i$ in (3.54), the matrix in the argument of the above determinant has the i-th column and row equal to zero. Hence, the determinant is null, which proves that λ_i is an eigenvalue of $\mathbf{\Lambda}_{N-1} + \sigma_m \boldsymbol{z}_m \boldsymbol{z}_m^\top$ and then of $\tilde{\mathbf{L}}$. This proves that, if $u_i(v_{m_1}) = u_i(v_{m_2})$, the i-th eigenvalue is not perturbed. Let us now prove that the corresponding eigenvector is not perturbed as well. Using the fact that $\tilde{\lambda}_i = \lambda_i$, the perturbed eigenvector $\tilde{\boldsymbol{u}}_i$ must be a solution of

$$\boldsymbol{U}_{N-1}\left(\mathbf{\Lambda}_{N-1} + \sigma_m\, \boldsymbol{z}_m \boldsymbol{z}_m^\top\right) \boldsymbol{U}_{N-1}^\top \tilde{\boldsymbol{u}}_i = \lambda_i \tilde{\boldsymbol{u}}_i. \tag{3.55}$$

If we set $\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_i$ in (3.55), we get

$$\boldsymbol{U}_{N-1}\mathbf{\Lambda}_{N-1}\mathbf{e}_i + \sigma_m\, \boldsymbol{U}_{N-1}\boldsymbol{z}_m \boldsymbol{z}_m^\top \mathbf{e}_i = \lambda_i \boldsymbol{u}_i, \tag{3.56}$$

with $i = 2, \ldots, N$, where $\mathbf{e}_i$ is defined as a vector of size $N-1$, with all zeros, except the $(i-1)$-th entry, which is equal to 1. Now, if $u_i(v_{m_1}) = u_i(v_{m_2})$, $z_m(i)$ is zero, which implies $\boldsymbol{z}_m^\top \mathbf{e}_i = 0$. Furthermore, $\boldsymbol{U}_{N-1}\mathbf{\Lambda}_{N-1}\,\mathbf{e}_i = \lambda_i \boldsymbol{u}_i$. Hence, equation (3.55) is satisfied with $\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_i$. This concludes the proof. ■

Theorem 1 handles all the cases where some entries of vector $\boldsymbol{z}_m$ are equal to zero. Let us denote by $\mathcal{S} := \{i : z_m(i) \neq 0\}$ the set of indices for which $z_m(i) \neq 0$ and by $\bar{\mathcal{S}}$ the complement set. From Theorem 1, we know that $\tilde{\lambda}_i = \lambda_i$, for $i \in \bar{\mathcal{S}}$. Let us now derive the remaining $N - 1 - |\mathcal{S}|$ eigenvalues, which must be the zeros of the

following characteristic polynomial, for $\tilde{\lambda} \neq \lambda_i$ (see also [100], [101])

$$\begin{aligned}
&|\mathbf{\Lambda}_{N-1} - \tilde{\lambda}\boldsymbol{I} + \sigma_m \boldsymbol{z}_m \boldsymbol{z}_m^\top| \\
&= |\mathbf{\Lambda}_{N-1} - \tilde{\lambda}\boldsymbol{I}| \cdot |\boldsymbol{I} + \sigma_m (\mathbf{\Lambda}_{N-1} - \tilde{\lambda}\boldsymbol{I})^{-1} \boldsymbol{z}_m \boldsymbol{z}_m^\top| \\
&= |\mathbf{\Lambda}_{N-1} - \tilde{\lambda}\boldsymbol{I}| \left(1 + \sigma_m \boldsymbol{z}_m^\top (\mathbf{\Lambda}_{N-1} - \tilde{\lambda}\boldsymbol{I})^{-1} \boldsymbol{z}_m\right) \\
&= \prod_{i=2}^{N} (\lambda_i - \tilde{\lambda}) \cdot \left(1 + \sigma_m \sum_{\ell \in \mathcal{S}} \frac{z_m^2(\ell)}{\lambda_\ell - \tilde{\lambda}}\right) \\
&= \prod_{i \in \bar{\mathcal{S}}} (\lambda_i - \tilde{\lambda}) \cdot \left[\prod_{i \in \mathcal{S}} (\lambda_i - \tilde{\lambda}) + \sigma_m \sum_{\ell \in \mathcal{S}} z_m^2(\ell) \prod_{\substack{j \in \mathcal{S} \\ j \neq \ell}} (\lambda_j - \tilde{\lambda})\right].
\end{aligned} \tag{3.57}$$

Note that the matrix inversion used above is possible, because we are considering the case $\tilde{\lambda} \neq \lambda_i$. In particular, since $\tilde{\lambda} \neq \lambda_i, i \in \mathcal{S}$, the left-hand determinant in the last line of (3.57) is different from zero and then the eigenvalues $\tilde{\lambda}_i$, with $i \in \mathcal{S}$, must be the roots of the following equation

$$\prod_{i \in \mathcal{S}} (\lambda_i - \tilde{\lambda}) + \sigma_m \sum_{\ell \in \mathcal{S}} z_m^2(\ell) \prod_{\substack{j \in \mathcal{S} \\ j \neq \ell}} (\lambda_j - \tilde{\lambda}) = 0. \tag{3.58}$$

Finding the roots of this polynomial entails the use of a numerical algorithm. However, under the assumption that the perturbation is very small, we can derive approximate expressions for the perturbed eigenvalues. We know from (3.5) that the eigenvalue perturbations are bounded by the eigenvalue gaps. Now, we further assume that the perturbations are much smaller than the gaps, i.e. $\Delta\lambda_{i,m} \ll g_i^+$, in case of edge addition, or $|\Delta\lambda_{i,m}| \ll g_i^-$, in case of edge deletion. We will verify later on that this assumption is verified when $z_m^2(i) = [u_i(v_{m_1}) - u_i(v_{m_2})]^2 \ll g_i^+$, in case of edge addition and $z_m^2(i) = [u_i(v_{m_1}) - u_i(v_{m_2})]^2 \ll g_i^-$, in case of edge deletion. Under this assumption, to be verified later on, we can analyze the polynomial in (3.58) in the neighborhood of a generic eigenvalue λ_ℓ of the nominal graph, for $\ell \in \mathcal{S}$, setting $\tilde{\lambda} := \lambda_\ell + \varepsilon_\ell$ in (3.58). Taking the first order approximation of the polynomial in (3.58), with respect to the variables ε_ℓ and $z_m(\ell), \ell \in \mathcal{S}$, we get

$$-\varepsilon_\ell \prod_{\substack{i \in \mathcal{S} \\ i \neq \ell}} (\lambda_i - \tilde{\lambda}) + \sigma_m z_m^2(\ell) \prod_{\substack{i \in \mathcal{S} \\ i \neq \ell}} (\lambda_i - \tilde{\lambda}) = 0, \tag{3.59}$$

which yields the simple solution

$$\varepsilon_\ell = \sigma_m z_m^2(\ell) = \sigma_m [u_\ell(v_{m_1}) - u_\ell(v_{m_2})]^2. \tag{3.60}$$

Indeed, we can verify that $z_m^2(\ell)$ is of the same order of magnitude as the modulus of ε_ℓ, so that, if $z_m^2(\ell)$ is much smaller than the eigenvalue gap, also the eigenvalue perturbation is much smaller than the gap.

Let us consider now the perturbation of the eigenvectors. Starting again from (3.4), we wish to find $\tilde{\boldsymbol{u}}_i$ as a solution of

$$(\boldsymbol{U}_{N-1}\boldsymbol{\Lambda}_{N-1}\boldsymbol{U}_{N-1}^\top + \sigma_m \boldsymbol{a}_m \boldsymbol{a}_m^\top)\, \tilde{\boldsymbol{u}}_i = \tilde{\lambda}_i \tilde{\boldsymbol{u}}_i \tag{3.61}$$

Since the eigenvector $\tilde{\boldsymbol{u}}_i$ must be orthogonal to the vector $\mathbf{1}$ of all ones associated with the null eigenvalue of $\mathbf{L}$, $\tilde{\boldsymbol{u}}_i$ must be in the span of the columns of $\boldsymbol{U}_{N-1}$, i.e. there must exist a nontrivial vector $\mathbf{b}$, such that $\tilde{\boldsymbol{u}}_i = \boldsymbol{U}_{N-1}\mathbf{b}$. At the same time, if we define the vector $\boldsymbol{v}_i = \boldsymbol{U}_{N-1}^\top \tilde{\boldsymbol{u}}_i$, we get $\boldsymbol{v}_i = \mathbf{b}$. Now, multiplying both sides of (3.61) by $\boldsymbol{U}_{N-1}^\top$ from the left side, and using the vector $\boldsymbol{v}_i := \boldsymbol{U}_{N-1}^\top \tilde{\boldsymbol{u}}_i$, we can rewrite (3.61) as

$$(\boldsymbol{\Lambda}_{N-1} - \tilde{\lambda}_i \boldsymbol{I} + \sigma_m \boldsymbol{z}_m \boldsymbol{z}_m^\top)\, \boldsymbol{v}_i = \mathbf{0}. \tag{3.62}$$

Let us consider first the case in which there are some elements $z_m(i) = 0$, for $i \in \bar{\mathcal{S}}$, in which case we know from Theorem 1 that the corresponding eigenvector $\boldsymbol{u}_i$ is not perturbed. We can then focus on the cases where $z_m(i) \neq 0$, where $\tilde{\lambda}_i \neq \lambda_i$. The solution of equation (3.62), valid when $\tilde{\lambda}_i \neq \lambda_i$, is known from [101] to be:

$$\boldsymbol{v}_i = -(\boldsymbol{\Lambda}_{N-1} - \tilde{\lambda}_i \boldsymbol{I})^{-1}\, \boldsymbol{z}_m. \tag{3.63}$$

Under the conditions specified before, the invertibility of the matrix is guaranteed. Given $\boldsymbol{v}_i$, the perturbed eigenevector can then be found as $\tilde{\boldsymbol{u}}_i = \boldsymbol{w}_i / \|\boldsymbol{w}_i\|$, $i = 2, \ldots, N$, where

$$\boldsymbol{w}_i = \boldsymbol{U}_{N-1}\boldsymbol{v}_i = -\boldsymbol{U}_{N-1}(\boldsymbol{\Lambda}_{N-1} - \tilde{\lambda}_i \boldsymbol{I})^{-1}\, \boldsymbol{z}_m, \tag{3.64}$$

or, more explicitly,

$$\boldsymbol{w}_i = -\sum_{j=2}^{N} \frac{1}{\lambda_j - \tilde{\lambda}_i}\, [u_j(v_{m_1}) - u_j(v_{m_2})]\, \boldsymbol{u}_j. \tag{3.65}$$

Let us consider now again a small perturbation analysis, valid when the eigenvalue perturbation is much smaller than the eigenvalue gap, i.e. $\varepsilon_\ell = \sigma_m z_m^2(\ell) = \sigma_m [u_\ell(v_{m_1}) - u_\ell(v_{m_2})]^2 \ll g_\ell^+$, in case of edge addition (or g_ℓ^-, in case of deletion).

In such a case, the first order approximation becomes

$$\boldsymbol{w}_i = \frac{[u_i(v_{m_1}) - u_i(v_{m_2})]}{\varepsilon_i}\,\boldsymbol{u}_i + \sum_{\substack{j=2\\ j\neq i}}^{N} \frac{[u_j(v_{m_1}) - u_j(v_{m_2})]}{\lambda_i - \lambda_j}\,\boldsymbol{u}_j. \tag{3.66}$$

Denoting by α the normalization coefficient needed to enforce the unit norm, and using (3.60), we can write

$$\tilde{\boldsymbol{u}}_i = \frac{1}{\alpha}\left(\boldsymbol{u}_i + \sigma_m \sum_{\substack{j=2\\ j\neq i}}^{N} \frac{[u_j(v_{m_1}) - u_j(v_{m_2})][u_i(v_{m_1}) - u_i(v_{m_2})]}{\lambda_i - \lambda_j}\boldsymbol{u}_j\right). \tag{3.67}$$

Chapter 4

Perturbation Centrality

Abstract

Graph-based learning tools are efficient methods to extract information from large amount of data. In fact, studying how agents of a network interact between one another is instrumental to understand, control and compare complex systems. In particular, if we consider physical or social networks where the links are directly observable, we can identify the role of the nodes and the links and rank them according to the impact of their possible failure. In this chapter, we will address the problem of ranking the edges of a network, identifying the inter cluster links whose removal would disconnect or alter critically the connectivity and/or the clustering property of the network.

4.1 Introduction

Network science has the role of understanding and explaining phenomena happening in complex systems. Such systems, are made of arbitrary entities, e.g. in a social system they can be persons or organizations, and these entities are linked by one or more types of relations. Essential network analysis tools are centrality indices defined over vertices [15] or edges [52] of the graph. Some of the well-known indices for computing centrality are closeness [12], stress [113] and betweenness [15]. They are designed to rank vertices or edges of the network according to their position in the network. Many centrality indices are based on shortest paths linking pairs of vertices, measuring, e.g., the average distance from other vertices, or the number of shortest paths traversing the edges.

Betweenness and edge betweenness centrality assume an important role among the centrality measures when the goal of the analysis is to identify which vertices and/or edges are more traversed by information flows between any pair of vertices of the network. Applications of betweenness include lethality in biological networks [114, 115], identifying key actors in terrorist networks [116, 117], and supply chain management processes [118]. Betweenness is also used as the primary routine in popular algorithms for clustering and community identification [119] in real-world networks. For instance, the Girvan-Newman [57] algorithm iteratively partitions a network by identifying edges with high betweenness scores, removing them and recomputing centrality scores. Betweenness centrality measure is based on the enumeration of shortest paths. Thus, such a measure is linked with the network conductance, and by extension with the graph connectivity. Network conductance measures how well knit the graph is, and a well known result is that the conductance parameter is bounded by the algebraic connectivity (cf. Ch.2). In this chapter, we will propose an alternative centrality measure that, like the betweenness, it may be

used to identify links connecting the communities. This new centrality measure is based on the connectivity and clustering properties of the network. In particular, such a new centrality measure, denominated *perturbation centrality*, is computed evaluating the impact of each edge removal on the connectivity and clustering characteristic of the graph. Small perturbation theory plays a role in computing the impact of the edge removal, making this computation fast and simple.

The rest of the chapter is organized as follows:

- **Section 4.2.** In this section, we introduce the perturbation centrality measure.
- **Section 4.3.** In this section, we present a comparison between perturbation centrality and edge betweenness.
- **Section 4.4.** In this section, we apply the *perturbation centrality* measure at some real world networks.
- **Section 4.5.** Finally, in this section, we outline concluding remarks.

4.2 Topology perturbation centrality

Small perturbation analysis of network topology, developed in Ch. 3, can be an effective way to assess the impact of an edge perturbation on the connectivity properties of the overall graph. In this section, we introduce an edge centrality parameter that measures, for each edge, how much its removal affects the connectivity of the whole graph. The connectivity of a graph can be measured by the conductance parameter $\mathbf{\Phi}$ [120]. However, computing $\mathbf{\Phi}$ is known to be an NP-hard problem. Nevertheless, a well-known result of algebraic graph theory states that the conductance is lower bounded by the second smallest eigenvalue of its Laplacian matrix λ_2, also known as *algebraic connectivity*, see [2] and Section 2.1. In fact, it is also known that if we add a new edge in a graph, λ_2 can only increase or remain constant [18]. Hence, if we want to assess the change of connectivity (of connected graphs) resulting from the deletion of an edge, we can use the perturbation of the algebraic connectivity. When the m-th edge fails, the approximated expression of second smallest eigenvalue perturbation allows us to write (cf. Sec. **??**):

$$|\delta\lambda_{2,m}| = [u_2(v_{1,m}) - u_2(v_{2,m})]^2 \tag{4.1}$$

where $\mathbf{u}_2$ is the eigenvector of unit length associated with the second smallest eigenvalue, and $v_{1,m}, v_{2,m}$ are the nodes incident to the m-th edge. This expression is particularly relevant if the connected graph is composed of two clusters, suggesting that the perturbation is higher for edges connecting vertices belonging to different

clusters. Generalizing these arguments to graphs composed of $C > 2$ clusters, we should account for the first C eigenvalues: When the graphs are modular (i.e. composed of clusters), it is well known from spectral clustering theory [18] that the smallest eigenvalues of the Laplacian carry information about the number of clusters in a graph. Denoting with $\delta\lambda_{i,m}$ the approximated expression of the i-th eigenvalue perturbation due to the deletion of edge m, we define the *topology perturbation centrality* (TPC) of edge m as follows

$$\mathsf{p}(m) := \sum_{i=2}^{C} |\delta\lambda_{i,m}|. \tag{4.2}$$

The summation starts from $i = 2$ simply because, from (3.6), the perturbation induced by the deletion of every edge on the smallest eigenvalue is null, since the corresponding eigenvector is constant. Parameter $\mathsf{p}(m)$ assigns to each edge a number that quantifies how much the deletion of that particular edge affects the set of the C smallest eigenvalues of the Laplacian matrix. An example of application of this parameter is reported in Fig. 4.1a, where we encode on each edge the value assumed by the TPC (the darker the edge, the largest is the TPC value). We can see that, as expected, the darkest colors correspond to the edges whose removal would cause the split of the network.

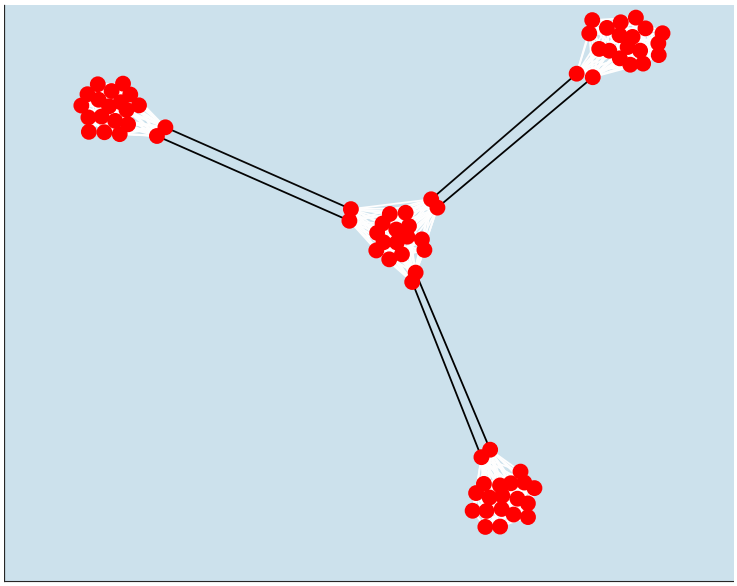
Note that, as the betweenness centrality, our new centrality measure can be also used as the primary routine in clustering and community identification algorithms in clustered real-world networks, similarly to the Girvan-Newman algorithm [57].

4.3 Comparison with edge betweenness

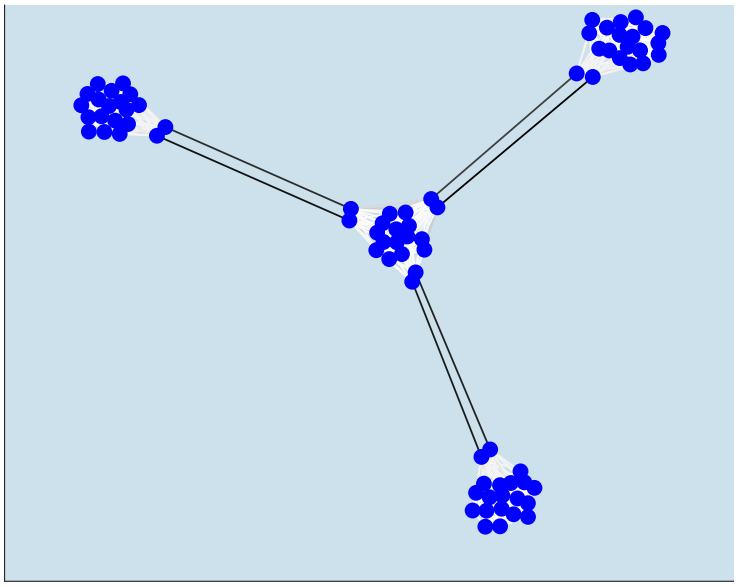
In many cases, the TPC value turns out to behave similarly to the edge betweeness centrality (EBC).

To test this similarity, in Fig. 4.1, we report an example of modular graph, obtained by connecting four clusters, generated as random geometric graphs, through a few edges placed at random. The perturbation centrality and edge betweenness are encoded by the color intensity of each edge: the darkest colors correspond to the edges with the largest centrality values.

This comparison is performed in order to investigate when the two measures, even if different in their definitions, give a common interpretation of some edges. The above considerations and Fig. 4.1 suggest that, for modular graphs, our perturbation centrality is closely related to the *edge betweenness* of the network [2]. Denoting with $\mathsf{B}(m)$ the *edge betweenness*, we recall that it measures the extent of a given edge to be part of the shortest path between any pair of vertices of the network [2].



(a)



(b)

Figure 4.1. Example of (a) *Perturbation centrality* and (b) *edge betweenness* of a modular graph with 4 clusters.

This parameter captures then the centrality of edges, which are central in the sense that most of the paths connecting other vertices pass through them. Conversely, our perturbation centrality measures how much the deletion of an edge tends to make the network disconnected, or close to become disconnected. To investigate the similarity between the two measures, we estimate the correlation coefficient ρ between them, defined as:

$$\rho = \frac{\sum_{m=1}^{M_r} \mathsf{B}(m)\mathsf{p}(m)}{\sqrt{\sum_{m=1}^{M_r} \mathsf{B}^2(m) \sum_{m=1}^{M_r} \mathsf{p}^2(m)}} \tag{4.3}$$

where $M_r \leq M$ is the number of relevant links. The choice of M_r depends on the topology and it affects the computational complexity of finding ρ. It is related to the number of critical links (bridge edges) and it can be estimated as $p_b N_1 N_2$, where N_1 and N_2 are the number of nodes of cluster 1 and 2, respectively, and the bridge probability p_b, which measures the probability that vertices belonging to different clusters (RGG) are linked to each other.

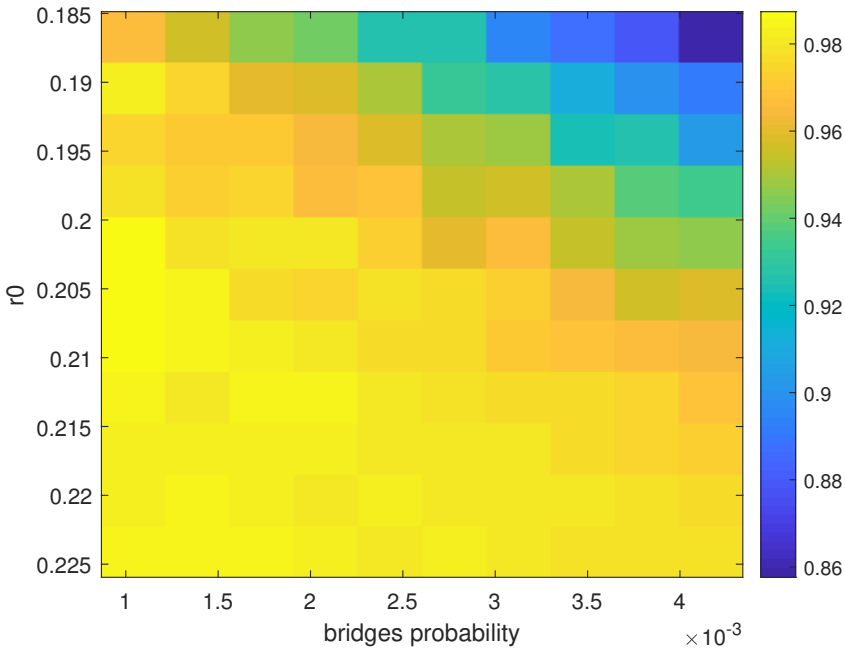


Figure 4.2. Average correlation coefficient ρ as a function of bridge probabilities and coverage radius r_0.

We estimated the correlation coefficient over a class of random graphs having different modularities. More specifically, we generated a graph composed of two separated random geometric graphs (RGG), linked to each other through a random number of edges. Two parameters control the modularity of this family of random graphs: r_0 is the coverage radius of each RGG and it controls the intra-cluster

connectivity; and the bridge probability p_b, that measures the inter-cluster connectivity. In Fig. 4.2, we report the correlation coefficient between $\mathsf{B}(m)$ and $\mathsf{p}(m)$, for different pairs (r_0, p_b). We can observe that for small values of p_b and high values of r_0, that is when the graph is highly modular, $\rho(r_0, p_b)$ is indeed very close to one, meaning that for highly modular graphs the two measures of centrality capture similar properties. Otherwise, the two measures tend to represent different aspects.

4.4 Real data tests

In this section, we will analyze some real world clustered networks. We will show how also in real networks the proposed centrality plays a key role in identifying the edges between clusters.

4.4.1 Dolphins network

Consider the undirected social network of frequent associations between 62 bottlenose dolphins in a community living off Doubtful Sound, New Zealand [121]. The network edges are 159, and they represent associations between dolphin pairs observed to co-occur more often than expected occasionally.

For this network we apply our measure of perturbation centrality. The color of the nodes indicates two different clusters observed on the network in [122] and [123]. Fig. 4.3 shows the dolphin social network for $K = 2$, where the color of each edge encodes the value of the perturbation centrality normalized with the maximum value of centrality of all nodes. We can see that the perturbation centrality ranks the edges identifying the inter-clusters links.

4.4.2 Books on US politics network

We consider now a network representing a set of books about US politics [124]. The 105 graph's nodes are books sold by *Amazon.com*, categorized by Mark Newman as liberal, neutral, or conservative. The edges represent frequent co-purchasing of books by the same buyers, that is, if two books are bought by the same customer then an edge exists. Thus, we perform our measure for $K = 2$ and $K = 3$ as shown in Fig. 4.4. From this figures, we can see how two clusters are evident, i.e. conservatives and liberals. In fact, the perturbation centrality ranks the edges identifying the links between this two clusters. Since the third cluster is not evident the case of $K = 3$ is less informative.

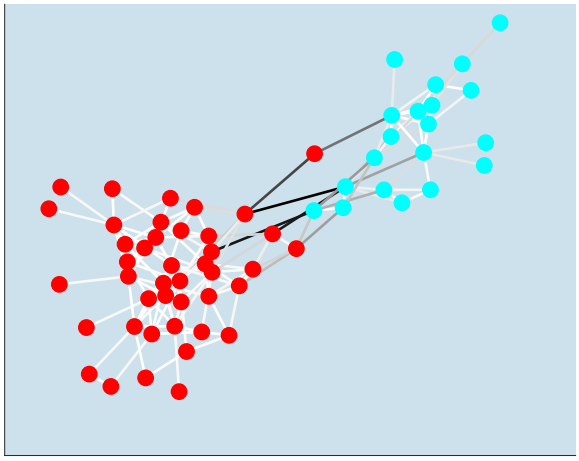


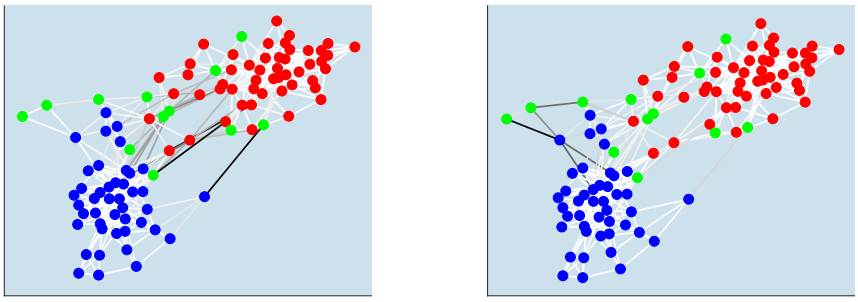
Figure 4.3. Perturbation centrality for dolphin social network for K=2.

4.4.3 American football network

This American football network has been obtain from the schedule of games played between between Division IA colleges during regular season Fall 2000 [57]. The graph's nodes are the 115 teams belonging to the 12 conferences, while an edge exists if the two teams (nodes) played a game between each other. Fig. 4.5 represents the value of normalized perturbation centrality for each link, encoded by the edges color, for different K. In the graphs of Fig. 4.5 each node has a color depending on the conference to which it belongs. The colors define the clusters, even if some clusters are overlapped. From the figures, we can see how the case of $K = 12$ is the one that identifies the edges inter-clusters signed with dark gray colors while the intra-cluster edges are white colored meaning that they have a perturbation centrality value around zero. For the other value of K, the perturbation centrality values are incomplete since we are considering less clusters than the actual number and we are not able to identify all the inter-clusters links.

4.5 Conclusions

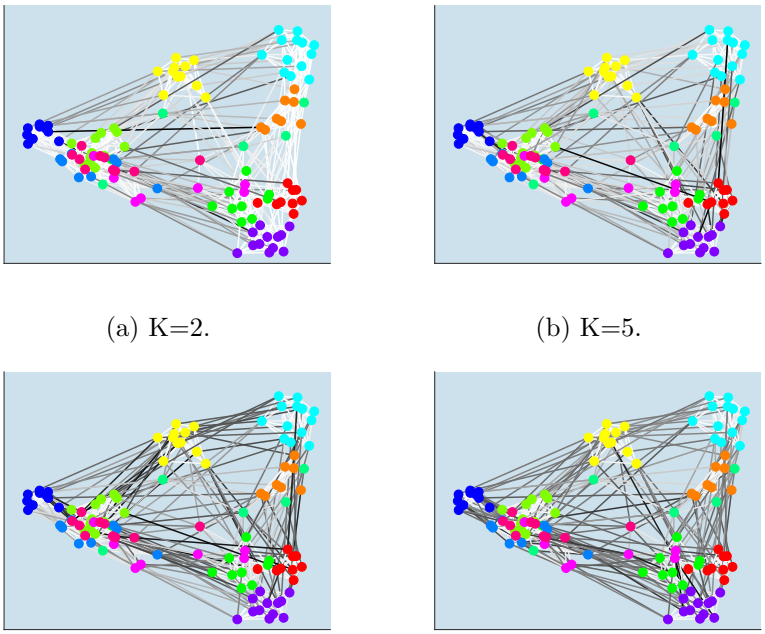
In this chapter, we introduced a new centrality measure, named Topology Perturbation Centrality (TPC), able to identify those edges whose removal is going to affect the graph connectivity. The computation of TPC is based on the perturbation of the Laplacian eigenvalues, whose approximation is easily found using the small perturbation theory. For modular graphs, such a measure is able to identify the edges whose failure cause the largest eiganvalues perturbations, and thus, the ones



(a) K=2 (b) K=3

Figure 4.4. Perturbation centrality for books political orientation network, for different values of K. In green the neutral, in red the conservative, and in blue the liberal.

that alter more the clustering properties of the graph when they fail [18]. Numerical results proved the ability of this measure to find inter-clusters edges, and they showed that, for a certain class of modular graphs, the perturbation centrality provides similar results to the edge betweenness centrality. Further investigations include its application in primary routine of community identification algorithms or in network science.



(a) K=2. (b) K=5.

(c) K=8. (d) K=12.

Figure 4.5. Perturbation centrality for the American football network.

Chapter 5

Graph-based learning under perturbations via TLS

Abstract

In this chapter, we solve two major graph-based learning tasks, such as graph topology identification and inference of graph signals over graphs. Among the possible models to explain data interdependences, we rely on structural equation models (SEMs) that are know to accommodate several applications involving topology identification. However, obtaining conventional SEMs requires measurements across nodes. On the other hand, typical signal inference methods blindly trust a given topology. In practice however, signal or topology perturbations may occur in both tasks, due to model mismatch, outliers, outages or adversarial behavior. To cope with perturbations, we introduce a regularized total least-squares (TLS) approach and iterative algorithms with convergence guarantees. Generalizations are also considered relying on structured and/or weighted TLS when prior information on the perturbation is available. Finally, simulated and real data tests corroborate the effectiveness of the novel TLS-based approaches.

5.1 Introduction

Graphs are pervasive in many applications to analyze complex systems. In financial, biological or social sciences, data-driven graphs are used to model undirected as well as directed data dependencies. In physical networks, graphs are adopted to represent physical or engineered links between vertices of, for instance, vehicular, power or communication networks; in particular, they are useful in tasks such as devising resource allocation strategies or recovering missing data. However, perturbations may afflict links or vertices in both data-driven and physical networks, compromising the performance of graph-based learning tasks. In brain networks, for instance, the topology that is inferred may be imperfect because of e.g., noise in the data or model mismatch; while in a communication network, graph topology perturbations may arise due to link or node outages.

5.1.1 Related works

The vulnerability of networked systems to failures, anomalies, or model mismatch has been investigated in [43, 47, 78, 90, 112, 125, 126]. In the context of statistical analysis of network data, error propagation in network characteristics (e.g. count of subgraphs) has been studied in [90] and [125]. In order to account for topological perturbation, probabilistic or uncertain graphs have been considered for clustering [78], graph filtering [43], and consensus [127]. In other works, we developed tools based on small perturbation analysis of the Laplacian matrix [94], as described in

Ch. 3. In particular, such analysis is used to handle graph perturbations for robust resource allocation [47], graph signal inference [126], and tracking of time-varying graph signals [112].

In this chapter, we analyze signal and graph perturbations for the tasks of topology identification (ID) and graph signal recovery based on total least-squares. TLS is the generalization of least-squares (LS) tailored to account for error mismatch (a.k.a. noise) present in both the input and the output matrices [17]. TLS is used in a plethora of applications including system identification [128], information retrieval [129], forecasting of financial data and reconstruction of medical images [130]. Building upon TLS, weighted TLS [131], structured TLS [132], and sparse TLS [133] have also been used to incorporate different prior information.

In this chapter, we rely on SEM [68] as model to explain the relationships between data. Such a model has been widely adopted because differently from other approaches, it captures causal relationships among nodes, as recalled in Ch. 2, where we summarized some of the most important topology ID methods.

In particular, SEMs have been employed in diverse fields for network topology identification [25, 70, 134–137], most of which require availability of measurements across nodes. Topology identification with partially observed nodal processes has also been studied recently [42, 138]. Leveraging piecewise stationarity, SEMs-based topology inference was pursued in [138] when only (partial) statistics of nodal measurements are given, while a joint inference algorithm was developed in [42] to identify the topology as well as interpolating graph signals based on partial observations of the nodal signals. However, neither of them accounts for signal perturbations.

There are also approaches for topology identification that rely on Graphical LASSO with its generalizations [27, 139], and graphical model selection for stationary [140] and non-stationary [141, 142] processes; see also [25, 143]. Different from these approaches, the methods here do not rely on any probabilistic assumptions for the network model, further account for perturbations in the topology or the nodal observations.

Most existing works on graph signals reconstruction assume the graph topology known and they rely on the concept of signal smoothness on the graph, that is, neighboring nodes have similar signal values. In particular, parametric [37, 144, 145] and non-parametric [36, 146, 147], approaches have been devised to deal with this task. However, differently from our work, they assumed that the network topology is perfectly known or that the nodal signal lies in a graph-related subspace.

5.1.2 Contributions

In the present chapter, we present solutions for the two following graph learning tasks:

T1. Topology identification based on perturbed nodal signal observations; and,

T2. graph signal inference given partial nodal observations and perturbed topologies.

An example of T1 is the inference of a gene regulatory network, that might be influenced by eventual errors on the available data, e.g., during the data collection process. Moreover, task T2 is for example the case of signal recovery over a wireless network with faiding links.

The specific contributions of this work are as follows. Task T1 is solved by developing two algorithms with complementary strengths: the first one yields an ε-optimal solution, while the second achieves a sub-optimal yet more efficient solution. In particular, the problem is regarded in its fractional form, and solved with a bisection-based algorithm that achieves near-optimum solution. The second solver reaches more efficiently a local minimum by employing an alternating descent approach. The topology ID task is also investigated when having prior information on the structure of error matrix. Then, the signal recovery task is solved by an alternating descent method. Task T2 studies also the case of prior information on topology perturbations and noise variance. Finally, the identifiability of the proposed model is analyzed.

The rest of the chapter is organized as follows:

- **Section 5.2.** In this section, we introduce the context and the TLS formulation with its weighted and structured variants.
- **Section 5.3.** In this section, we investigate the topology ID problem T1.
- **Section 5.4.** In this section, we address the graph signal inference task T2.
- **Section 5.5.** In this section, synthetic and real data tests are carried out to illustrate the merits of the proposed TLS-based approaches.
- **Section 5.6.** Finally, in this section, concluding remarks and future directions are outlined.

5.2 Preliminaries

In the present section, we review linear SEMs and TLS, along with structured and weighted TLS variants.

5.2.1 Structural Equation Models

Consider a directed network of N nodes, whose topology is captured by the generally asymmetric adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ with entries $a_{ij} := [\mathbf{A}]_{ij}$, and $a_{ij} \neq 0$ if a directed edge from node j to node i is present. Assume that the network represents a complex system, where y_{it} is the measurement at node i at instant t. The output measurement y_{it} in SEMs depends on its single-hop neighbor measurements, and an exogenous input signal x_{it}, that is

$$y_{it} = \sum_{j \neq i} a_{ij} y_{jt} + b_{ii} x_{it}, \qquad t = 1, \ldots, T \tag{5.1}$$

where $b_{ii} > 0$ weighs the exogenous input. Concatenating nodal measurements in vectors $\boldsymbol{y}_t := [y_{1t}, \ldots, y_{Nt}]^\top$, and $\boldsymbol{x}_t := [x_{1t}, \ldots, x_{Nt}]^\top$ per slot t, the matrix-vector version of (5.1) can be compactly written as $\boldsymbol{y}_t = \mathbf{A}\boldsymbol{y}_t + \mathbf{B}\boldsymbol{x}_t,\ t = 1, \ldots, T$, where $a_{ii} = 0$ and $\mathbf{B} := \text{diag}(b_{11}, \ldots, b_{NN})$.

Collecting inputs and outputs[1] across T slots, $N \times T$ matrices $\mathbf{X} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T]$ and $\mathbf{Y} := [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T]$ can be formed, to obtain the linear matrix model

$$\mathbf{Y} = \mathbf{A}\mathbf{Y} + \mathbf{B}\mathbf{X}\,. \tag{5.2}$$

Prior works on SEMs treat perturbations as additive observation noise, $\mathbf{Y} = \mathbf{A}\mathbf{Y} + \mathbf{B}\mathbf{X} + \mathbf{V}$, where $\mathbf{V} \in \mathbb{R}^{N \times T}$ is the error matrix. Generally, these works aim to estimate $\mathbf{A}$ (and possibly $\mathbf{B}$), when measurements $\mathbf{Y}$ and $\mathbf{X}$ are given, using LS or regularized LS [69, 134]. On the other hand, if matrices $\mathbf{A}$, $\mathbf{B}\mathbf{X}$ (e.g. obtained by historical data) and a subset of entries of $\mathbf{Y}$ are given, it is also possible to recover the unobserved nodal signals using LS-based methods [42]. However, existing approaches do not consider possible errors in $\mathbf{A}$ or $\mathbf{Y}$, and we are motivated to adopt TLS methods to cope with graph signal and topology perturbations that can be possibly present in SEMs. In particular, if $\mathbf{Y}$ is corrupted by noise, such that the model is $\mathbf{Z} - \mathbf{E} = \mathbf{A}(\mathbf{Z} - \mathbf{E}) + \mathbf{B}\mathbf{X}$, with $\mathbf{Z} = \mathbf{Y} + \mathbf{E}$, we infer $\mathbf{A}$ by using a TLS-based approach that takes into account errors both in the output and the input. On the other hand, given a perturbed $\mathbf{A}$ and partial noisy nodal observations, we recover the graph signal by using a TLS-based approach, minimizing both errors in $\mathbf{A}$ and in the observations. Before introducing the formulation of the two aforementioned tasks, we review the classical TLS and its weighed and structured variants, in the following subsection.

[1] Causes-effect per node do not have to happen instantaneously, since causes $\{y_{jt}, x_{it}\}$ can occur at the beginning and effect y_{it} at the end of slot t.

5.2.2 Weighted and structured TLS

Total least-squares assumes the perturbed linear system of equations $\boldsymbol{F} = (\boldsymbol{H} + \boldsymbol{P})\boldsymbol{\Theta} - \boldsymbol{\Sigma}$, where $\boldsymbol{F} \in \mathbb{R}^{M\times T}$ represents the output matrix with $M < T$, $\boldsymbol{H} \in \mathbb{R}^{M\times N}$ the input (or regression) matrix, $\boldsymbol{\Theta} \in \mathbb{R}^{N\times T}$ an unknown matrix of parameters, while $\boldsymbol{\Sigma} \in \mathbb{R}^{M\times T}$ and $\boldsymbol{P} \in \mathbb{R}^{M\times N}$ denote the error matrices. See Fig. 5.1 for graphical representation of the perturbed linear system. Different from

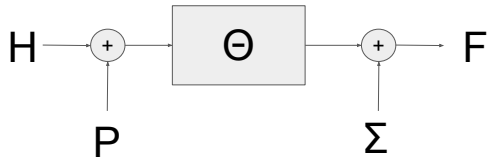


Figure 5.1. Graphical representation of total least-squares model.

classical LS where $\mathbf{P} = \mathbf{0}$, TLS treats symmetrically the input and the output in the sense that both $\boldsymbol{H}$ and $\boldsymbol{F}$ may have errors due to model mismatch, noise, or outliers. Thus, TLS solves the following problem

$$\min_{\boldsymbol{\Theta},\boldsymbol{P},\boldsymbol{\Sigma}} \quad \|[\boldsymbol{P},\boldsymbol{\Sigma}]\|_F^2 \tag{5.3a}$$

$$\text{s. to} \quad \boldsymbol{F} = (\boldsymbol{H} + \boldsymbol{P})\boldsymbol{\Theta} - \boldsymbol{\Sigma}. \tag{5.3b}$$

In the structured version of TLS, one can exploit the structure of input and output matrices, as well as noise statistics, to achieve improved estimation performance. The structure of a matrix in the TLS context is defined as follows [130, 133].

Definition 1. Given a parameter vector $\boldsymbol{\omega} \in \mathbb{R}^{n_\omega}$, the $M \times (N + T)$ data matrix $[\boldsymbol{H}, \boldsymbol{F}](\boldsymbol{\omega})$ has a structure $\mathscr{S}(\boldsymbol{\omega})$ characterized by $\boldsymbol{\omega}$, if and only if there is a mapping such that $\boldsymbol{\omega} \in \mathbb{R}^{n_\omega} \to [\boldsymbol{H}, \boldsymbol{F}](\boldsymbol{\omega}) := \mathscr{S}(\boldsymbol{\omega}) \in \mathbb{R}^{M\times(N+T)}$.

Note that Definition 1 reduces trivially to the unstructured case when $\boldsymbol{\omega} := \text{vec}([\boldsymbol{H}, \boldsymbol{F}])$ with dimension $M(N + T)$. However, when $\boldsymbol{\omega}$ provides a parsimonious representation of the data matrix with $n_\omega \ll M(N + T)$, we can take advantage of the matrices' structure. Such a structure can be present in many applications. For example, system identification, deconvolution, and linear prediction may involve Toeplitz and Hankel matrices, other examples are Vandermonde and circulant matrices that show up in e.g., spatio-temporal harmonic retrieval problems [130]. By introducing the parameter vector $\boldsymbol{\omega}$ and the noise parameter vector $\boldsymbol{\nu} \in \mathbb{R}^{n_\omega}$, such

that $\mathscr{S}(\boldsymbol{\omega}+\boldsymbol{\nu}) := [\boldsymbol{H}+\mathbf{P}, \boldsymbol{F}+\boldsymbol{\Sigma}](\boldsymbol{\omega}+\boldsymbol{\nu})$, the Frobenius norm $\|[\boldsymbol{P}, \boldsymbol{\Sigma}]\|_F^2$ becomes $\|\boldsymbol{\nu}\|_2^2$. The weighted TLS is obtained if prior knowledge about the $\boldsymbol{\nu}$ is incorporated by weighting the norm $\|\boldsymbol{\nu}\|_2^2$ through the $n_\omega \times n_\omega$ positive definite matrix $\mathbf{W}$. Hence, the structured and weighted TLS (SWTLS) cost is expressed as $\boldsymbol{\nu}^\top \mathbf{W} \boldsymbol{\nu}$. Note that, when $\mathbf{W} = \mathbf{I}$, the SWTLS boils down to a structured-only form. In our work, the classical SWTLS approach has been revised to accomplish the signal recovery task. In particular, Def. 1 is used to capture the nonzero patterns of $\mathbf{A}$, when we know a priori that the perturbations occur only on the existing links, and the weight matrices are employed to incorporate possible a priori information on link failure probabilities and observation error variances (see Sec. 5.4.1).

5.3 Topology ID with signal perturbations

Defects and outliers in the measuring process lead to perturbed nodal signals. This perturbation may compromise the performance of the topology ID task. Let us rewrite matrix $\mathbf{Y}$ in (5.2) as $\mathbf{Z} - \mathbf{E}$, where $\mathbf{E}$ is a perturbation matrix. Considering this perturbation, given $\mathbf{Z}$ and $\mathbf{BX}$, the aim of this section is to find $\mathbf{A}$ from the "measurement-perturbed" SEM

$$\mathbf{Z} - \mathbf{E} = \mathbf{A}(\mathbf{Z} - \mathbf{E}) + \mathbf{BX}. \tag{5.4}$$

The presence of the perturbation that appears in both sides motivates a formulation inspired by TLS method recalled in (5.3), noting that the peculiarity of our model is that the perturbation of the input and output matrix is exactly the same, i.e. $\mathbf{E} \in \mathbb{R}^{N \times T}$. In most real-world networks, such as social, transportation, and biological networks, the nodes exhibit a few interconnections and the corresponding adjacency matrix is sparse. Thus, accounting for the latter through a sparsity-promoting regularization term, we formulate a regularized TLS-based approach for "measurement-perturbed" SEM (5.4) (TLS-SEM) given by

$$\{\hat{\mathbf{A}}, \hat{\mathbf{E}}\} = \arg\min_{\mathbf{A},\mathbf{E}} \quad \|\mathbf{E}\|_F^2 + \lambda \|\mathbf{A}\|_1 \tag{5.5a}$$

$$\text{s.to} \quad \mathbf{Z} = \mathbf{A}(\mathbf{Z} - \mathbf{E}) + \mathbf{BX} + \mathbf{E} \tag{5.5b}$$

$$a_{ii} = 0, \quad i = 1, \ldots, N \tag{5.5c}$$

where λ is a non-negative regularization parameter, and constraint (5.5c) enforces the absence of self-loops in $\mathbf{A}$. However, the optimization problem in (5.5) is nonconvex, and will be solved in the following subsections developing two solvers with complementary merits.

5.3.1 Bisection-based algorithm

In this subsection, we first recast problem (5.5) into a fractional form that can be solved using a bisection-based (BB) iteration, which is convergent to an ε-optimal solution in a finite number of iterations, even though (5.5) is nonconvex [148]. The following lemma shows how to reformulate (5.5) in a fractional form.

Lemma 3. *With $\mathbf{\Phi} := \mathbf{Z} - \mathbf{BX}$, and $\boldsymbol{\varphi}_i^\top$ denoting its i-th row, the TLS problem in* (5.5) *is equivalent to the fractional problem*

$$\hat{\mathbf{A}} = \arg\min_{\{\boldsymbol{a}_{-i}\}_{i=1}^N} \quad \sum_{i=1}^N \Big[\frac{\left\|\boldsymbol{\varphi}_i - (\mathbf{Z}_{-i})^\top \boldsymbol{a}_{-i}\right\|_2^2}{1 + N\left\|\boldsymbol{a}_{-i}\right\|_2^2} + \lambda \left\|\boldsymbol{a}_{-i}\right\|_1\Big] \tag{5.6}$$

where $\boldsymbol{a}_{-i}^\top$ is the i-th row of $\mathbf{A}$ without the i-th entry, and $\mathbf{Z}_{-i}$ the $(N-1)\times T$ submatrix of $\mathbf{Z}$ after removing its i-th row.

Proof. Clearly, (5.5) can be rewritten as

$$\arg\min_{\{\boldsymbol{a}_i, \boldsymbol{\epsilon}_i\}_{i=1}^N} \quad \sum_{i=1}^N \Big(\frac{1}{N}\left\|[\mathbf{E}^\top, \sqrt{N}\boldsymbol{\epsilon}_i]\right\|_F^2 + \lambda \left\|\boldsymbol{a}_i\right\|_1\Big) \tag{5.7a}$$

$$\text{s. to} \quad \boldsymbol{z}_i = (\mathbf{Z}^\top - \mathbf{E}^\top)\boldsymbol{a}_i + b_{ii}\boldsymbol{x}_i + \boldsymbol{\epsilon}_i\,, \forall i \tag{5.7b}$$

$$a_{ii} = 0, \forall i \tag{5.7c}$$

where $\boldsymbol{a}_i^\top$, $\boldsymbol{z}_i^\top$, $\boldsymbol{x}_i^\top$, $\boldsymbol{\epsilon}_i^\top$ are the i-th rows of $\mathbf{A}$, $\mathbf{Z}$, $\mathbf{X}$, $\mathbf{E}$, respectively, and b_{ii} is the i-th diagonal entry of $\mathbf{B}$. Thus, the constraint (5.7b) becomes

$$\boldsymbol{\varphi}_i = (\mathbf{Z}^\top - \mathbf{E}^\top)\boldsymbol{a}_i + \boldsymbol{\epsilon}_i\,. \tag{5.8}$$

Next, with $\boldsymbol{v}_i := \text{vec}([\mathbf{E}^\top, \sqrt{N}\boldsymbol{\epsilon}_i])$, we have $\left\|[\mathbf{E}^\top, \sqrt{N}\boldsymbol{\epsilon}_i]\right\|_F^2 = \|\boldsymbol{v}_i\|_2^2$; and upon defining $\mathbf{G}(\boldsymbol{a}_i) := ([-\boldsymbol{a}_i^\top, \frac{1}{\sqrt{N}}] \otimes \mathbf{I}_T)$, constraint (5.7b) is re-expressed as

$$\boldsymbol{\varphi}_i - \mathbf{Z}^\top \boldsymbol{a}_i = \mathbf{G}(\boldsymbol{a}_i)\boldsymbol{v}_i, \forall i. \tag{5.9}$$

Note that, with $\mathbf{A}$ fixed, (5.7) becomes $\min_{\boldsymbol{v}_i} \|\boldsymbol{v}_i\|_2^2$ subject to (5.9), which admits a closed-form solution

$$\begin{aligned}\boldsymbol{v}_i &= \mathbf{G}^\top(\boldsymbol{a}_i)[\mathbf{G}(\boldsymbol{a}_i)\mathbf{G}^\top(\boldsymbol{a}_i)]^{-1}(\boldsymbol{\varphi}_i - \mathbf{Z}^\top\boldsymbol{a}_i)\\ &= (\|\boldsymbol{a}_i\|_2^2 + \frac{1}{N})^{-1}\mathbf{G}^\top(\boldsymbol{a}_i)(\boldsymbol{\varphi}_i - \mathbf{Z}^\top\boldsymbol{a}_i)\end{aligned} \tag{5.10}$$

where the second equality holds because $\mathbf{G}(\boldsymbol{a}_i)\mathbf{G}^\top(\boldsymbol{a}_i) = ([-\boldsymbol{a}_i^\top, \frac{1}{\sqrt{N}}]\otimes\mathbf{I}_T)([-\boldsymbol{a}_i^\top, \frac{1}{\sqrt{N}}]^\top \otimes \mathbf{I}_T) = (\|\boldsymbol{a}_i\|_2^2 + \frac{1}{N})\mathbf{I}_T$. Substituting (5.10) into (5.7a), and incorporating the con-

straint (5.7c), yields (5.6). ■

The fractional problem (5.6) is separable across rows of $\mathbf{A}$ as

$$\hat{\boldsymbol{a}}_{-i} = \arg\min_{\boldsymbol{a}_{-i}} \frac{\left\|\boldsymbol{\varphi}_i - (\mathbf{Z}_{-i})^\top \boldsymbol{a}_{-i}\right\|_2^2}{1 + N\left\|\boldsymbol{a}_{-i}\right\|_2^2} + \lambda \left\|\boldsymbol{a}_{-i}\right\|_1 \tag{5.11}$$

which can be viewed as a Lagrangian function. Considering the solution $\hat{\boldsymbol{a}}_{-i}$ for a given multiplier $\lambda > 0$ and letting $\mu := \|\hat{\boldsymbol{a}}_{-i}\|_1$, (5.11) is equivalent to

$$\begin{aligned} \hat{\boldsymbol{a}}_{-i} &= \arg \min_{\boldsymbol{a}_{-i} \in \boldsymbol{\chi}(\mu)} f(\boldsymbol{a}_{-i}) \\ f(\boldsymbol{a}_{-i}) &:= \frac{\left\|\boldsymbol{\varphi}_i - (\mathbf{Z}_{-i})^\top \boldsymbol{a}_{-i}\right\|_2^2}{1 + N\left\|\boldsymbol{a}_{-i}\right\|_2^2} \end{aligned} \tag{5.12}$$

where $\boldsymbol{\chi}(\mu) := \{\boldsymbol{a}_{-i} \in \mathbb{R}^{(N-1)} : \|\boldsymbol{a}_{-i}\|_1 \leq \mu\}$, and the relationship between μ and λ is data dependent.

The fractional problem in (5.6) remains nonconvex, and will be solved using an iterative solver. The solver consists of an outer loop based on bisection [149], and an inner loop using a variant of a branch-and-bound method [150]. The outer loop squeezes per i the minimum cost in (5.12) between a lower and an upper bounds. These bounds are obtained through the inner iteration, where a surrogate quadratic function is minimized. The surrogate quadratic function has non-fractional form, whose optimization is more convenient than directly optimizing $f(\boldsymbol{a}_{-i})$. Specifically, with q denoting a given upper bound of the cost in (5.12), we have

$$0 \leq q^* := \min_{\boldsymbol{a}_{-i} \in \boldsymbol{\chi}(\mu)} f(\boldsymbol{a}_{-i}) \leq q. \tag{5.13}$$

Then, we define

$$g^*(q) := \min_{\boldsymbol{a}_{-i} \in \boldsymbol{\chi}(\mu)} g(\boldsymbol{a}_{-i}, q) \tag{5.14}$$

with $g(\boldsymbol{a}_{-i}, q) := \|\boldsymbol{\varphi}_i - (\mathbf{Z}_{-i})^\top \boldsymbol{a}_{-i}\|_2^2 - q(1 + N\|\boldsymbol{a}_{-i}\|_2^2)$. Due to (5.13) and (5.14), it holds that

$$g^*(q) \leq 0. \tag{5.15}$$

Let q^* belong to a known interval $\mathcal{I}_i := [l_i, u_i]$ after the i-th outer iteration. Such an interval decreases at every step of the outer loop, and l_i, u_i are chosen depending on the sign of $g(\boldsymbol{a}_{-i}, q)$ (cf. Alg. 1). In particular, suppose that $g^*(q)$ is obtained at the middle point of $\mathcal{I}_i$, namely $q_m = (u_i + l_i)/2$. The sign of $g(q_m)$ indicates whether

(5.13) holds or not. If $g(q_m) > 0$, then we deduce from (5.13) that $q^* > q_m > l_i$, and $q^* \in \mathcal{I}_{i+1} := [q_m, u_i]$. On the other hand, $g(q_m) < 0$ implies $q^* \in \mathcal{I}_{i+1} := [l_i, q_m]$. In both cases, the interval at iteration $i+1$ shrinks through bisection.

algorithm 1 Bisection-based (BB) scheme

Input : $\boldsymbol{\Phi}$, $\boldsymbol{Z}$, ε, and δ

Output : $\mathbf{A}^*_{\varepsilon,i}$ (ε-optimal solution)

for $i = 1, \ldots, N$ **do**

  Set $l_0 = 0$, $u_0 = \|\boldsymbol{\varphi}_i\|_2^2$, iteration index $j = 0$, achievable cost $f_m = u_0$, and $(\boldsymbol{a}^*_{\varepsilon,i})^\top = 0$ be the i-th row of $\mathbf{A}^*_{\varepsilon,i}$

  while $u_j - l_j > \varepsilon$ **do**

    Let $q = \frac{u_j + l_j}{2}$

    Call Algorithm 2 and obtain $\boldsymbol{a}^*_{\delta,i}$

    $f_g = f(\boldsymbol{a}^*_{\delta,i})$ and $j = j + 1$

    $f_m = f_g$ and $\boldsymbol{a}^*_{\varepsilon,i} = \boldsymbol{a}^*_{\delta,i}$ **if** $f_g < f_m$

    Upper and lower bounds updates:

    $u_j = q, l_j = l_{j-1}$ **if** $g(\boldsymbol{a}^*_{\delta,i}, q) \leq 0$

    $l_j = q, u_j = u_{j-1}$ **if** $g(\boldsymbol{a}^*_{\delta,i}, q)\text{ł} \geq \delta$

    $l_j = q - \delta, u_j = u_{j-1}$ **if** $0 < g(\boldsymbol{a}^*_{\delta,i}, q) < \delta$

    Set $u_j = \min(u_j, f_g)$

  end

end

Note that, the Hessian of $g(\boldsymbol{a}_{-i}, q)$ is $\boldsymbol{H} := 2(\mathbf{Z}_{\_i}(\mathbf{Z}_{\_i})^\top - qN\mathbf{I})$, and since qN is positive, $\boldsymbol{H}$ is not guaranteed to be positive or negative definite. Thus, $g(\boldsymbol{a}_{-i}, q)$ is an indefinite quadratic.

Hence, in the inner loop, a branch-and-bound algorithm is used to find a feasible and δ-optimal solution $\boldsymbol{a}^*_{\delta,i}$ of (5.14), such that $g^*(q) \leq g(\boldsymbol{a}^*_{\delta,i}, q) \leq g^*(q) + \delta$, where δ denotes a specified margin. The branch-and-bound scheme, summarized in Alg. 2, searches for the upper and lower bounds of the function

$$g_{box}(\boldsymbol{a}_{-i}) = \min_{\boldsymbol{a}_{-i} \in \boldsymbol{\chi}(\mu), \boldsymbol{a}_L \leq \boldsymbol{a}_{-i} \leq \boldsymbol{a}_U} g(\boldsymbol{a}_{-i}, q) \tag{5.16}$$

where the constraint $\boldsymbol{a}_L \leq \boldsymbol{a}_{-i} \leq \boldsymbol{a}_U$ represents a box that shrinks as iterations progress. The upper bound U of $g_{box}(\boldsymbol{a}_{-i})$ can be obtained by a sub-optimal yet efficient solver for (5.16), see e.g., [151, 152]. While the lower bound L can be found by minimizing a function that under-approximates $g_{box}(\boldsymbol{a}_{-i})$, over the interval

$\boldsymbol{a}_L \leq \boldsymbol{a} \leq \boldsymbol{a}_U$. This convex approximate function is the following

$$g_L(\boldsymbol{a}_{-i}, q) = g(\boldsymbol{a}_{-i}, q) + (\boldsymbol{a}_{-i} - \boldsymbol{a}_L)^\top \mathbf{D}(\boldsymbol{a}_{-i} - \boldsymbol{a}_U) \tag{5.17}$$

where $\mathbf{D}$ is a diagonal positive semi-definite matrix chosen to ensure the convexity of $g_L(\boldsymbol{a}_{-i}, q)$, as the solution of the following semi-definite program

$$\min_{\mathbf{D}} \quad (\boldsymbol{a}_U - \boldsymbol{a}_L)^\top \mathbf{D}(\boldsymbol{a}_U - \boldsymbol{a}_L) \tag{5.18a}$$

$$\text{s. to} \quad \mathbf{H} + 2\mathbf{D} \succeq 0 \tag{5.18b}$$

where (5.18b) is used to assure (5.17) to remain convex. At each iteration of the inner loop, the initial box constraint of (5.16) is split depending on how $U - L$ compares with the preselected δ. This splitting process leads to a smaller U and a tighter L. The detailed inner loop is listed in Alg. 2.

In summary, Alg. 2 is called by Alg. 1 to find the δ-optimal solution and evaluate the sign of $g^*(q)$. However, since $\boldsymbol{a}^*_{\delta,i}$ is δ-optimal, meaning $g^*(q) \geq g(\boldsymbol{a}^*_{\delta,i}, q) - \delta$, if $g(\boldsymbol{a}^*_{\delta,i}, q) > \delta$, we set the lower bound l_{i+1} to q_m; otherwise, if $0 < g(\boldsymbol{a}^*_{\delta,i}, q) < \delta$ we set $l_{i+1} = q_m - \delta$. As far as convergence is concerned, the following can be established.

algorithm 2 Branch-and-Bound scheme

Input : $\boldsymbol{\psi}_i$, $\boldsymbol{Z}_i$, q, and δ.

Output : $\boldsymbol{a}^*_{\delta,i}$ (δ-optimal solution)

Initialize U, L, $\boldsymbol{a}_L$, $\boldsymbol{a}_U$, and set $\mathcal{K} = \{\boldsymbol{a}_L, \boldsymbol{a}_U, L\}$

while $\mathcal{K} \neq \emptyset$ **do**

  Solve (5.16) to obtain $\hat{\boldsymbol{a}}^*_{-i}$

  if $g(\hat{\boldsymbol{a}}^*_{-i}, q) < U$ **then**

    $U = g(\hat{\boldsymbol{a}}^*_{-i}, q)$ and $\boldsymbol{a}^*_{\delta,i} = \hat{\boldsymbol{a}}^*_{-i}$

  end

  Find $\mathbf{D}$ via (5.18), and find $\check{\boldsymbol{a}}^*_{-i}$ and $L = g_L(\check{\boldsymbol{a}}^*_{-i})$ minimizing (5.17).

  if $U - L > \delta$ *{split}* **then**

    Find $k = \max_n([\boldsymbol{a}_U]_n - [\boldsymbol{a}_L]_n)$

    Set $\boldsymbol{a}_{L,1} = \boldsymbol{a}_L(\boldsymbol{a}_{U,1} = \boldsymbol{a}_U)$ and $\boldsymbol{a}_{L,2} = \boldsymbol{a}_L(\boldsymbol{a}_{U,2} = \boldsymbol{a}_U)$ except the k-th entry:

    $[\boldsymbol{a}_{L,1}]_k = [\boldsymbol{a}_L]_k$ and $[\boldsymbol{a}_{U,1}]_k = \frac{[\boldsymbol{a}_U]_k + [\boldsymbol{a}_L]_k}{2}$

    $[\boldsymbol{a}_{U,2}]_k = [\boldsymbol{a}_U]_k$ and $[\boldsymbol{a}_{L,2}]_k = \frac{[\boldsymbol{a}_U]_k + [\boldsymbol{a}_L]_k}{2}$.

    Compute $\mathbf{D}_1$, $\mathbf{D}_2$ and L_1, L_2 for each new boxes and

    $\mathcal{K} = (\boldsymbol{a}_{L,1}, \boldsymbol{a}_{U,1}, L_1) \cup (\boldsymbol{a}_{L,2}, \boldsymbol{a}_{U,2}, L_2)$

    Compare L_1 and L_2 with U:

    $\mathcal{K} = \mathcal{K} \setminus (\boldsymbol{a}_{L,1}, \boldsymbol{a}_{U,1}, L_1)$, **if** $L_1 > U$

    $\mathcal{K} = \mathcal{K} \setminus (\boldsymbol{a}_{L,2}, \boldsymbol{a}_{U,2}, L_2)$, **if** $L_2 > U$

    $\mathcal{K} = \mathcal{K} \setminus (\boldsymbol{a}_{L,m}, \boldsymbol{a}_{U,m}, L_m)$, $m = \arg\min(L_m)$, **otherwise**.

  else

    $\mathcal{K} = \mathcal{K} \setminus (\boldsymbol{a}_L, \boldsymbol{a}_U, L)$

  end

end

Proposition 2. *After at most* $\left\lceil \ln(\frac{\|\boldsymbol{\varphi}_i\|_2^2}{\varepsilon - 2\delta}) / \ln(2) \right\rceil$ *iterations, with* $\varepsilon > 2\delta$, *an* ε-*optimal solution* $\boldsymbol{a}^*_{\varepsilon,i}$ *to* (5.13) *is reached, satisfying*

$$\boldsymbol{a}^*_{\varepsilon,i} \in \boldsymbol{\chi}(\mu), \text{ and } q^* \leq f(\boldsymbol{a}^*_{\varepsilon,i}) \leq q^* + \varepsilon, \; i = 1, \ldots, N\,. \tag{5.19}$$

Proof. See [133]. ■

5.3.2 Alternating descent algorithm

We now propose an efficient alternative at the solver proposed in the previous analysis, with guaranteed convergence at least to a stationary point. In fact, the bisection-based solver developed in the previous subsection can approach the global optimum of the fractional TLS, but it is computationally demanding. We reformulate (5.5), substituting (5.5b) into (5.5a), and we add $\|\mathbf{E}\|_F^2$ to the cost

function to constraint the error norm to be small, obtaining

$$\{\hat{\mathbf{A}}, \hat{\mathbf{E}}\} = \arg\min_{\mathbf{A},\mathbf{E}} \|\mathbf{E}\|_F^2 + \|\mathbf{Z} - \mathbf{A}(\mathbf{Z} - \mathbf{E}) - \mathbf{B}\mathbf{X}\|_F^2 + \lambda \|\mathbf{A}\|_1 \tag{5.20a}$$

$$\text{s. to} \quad a_{ii} = 0, \; i = 1, \ldots, N. \tag{5.20b}$$

Note that the minimization of (5.20) does not guarantee that (5.5b) is still satisfied. Problem (5.20) is convex with respect to (wrt) each block (matrix) variable $\mathbf{A}$ and $\mathbf{E}$. This motivates an alternating descent iteration to find a sub-optimal yet efficient solution. At iteration $k+1$, given $\hat{\mathbf{A}}[k]$, the error matrix can be estimated as

$$\hat{\mathbf{E}}[k+1] = \arg\min_{\mathbf{E}} \|\mathbf{Z} - \hat{\mathbf{A}}[k](\mathbf{Z} - \mathbf{E}) - \mathbf{B}\mathbf{X}\|_F^2 + \|\mathbf{E}\|_F^2$$

which admits the closed-form solution

$$\hat{\mathbf{E}}[k+1] = (\hat{\mathbf{A}}^\top[k]\hat{\mathbf{A}}[k] + \mathbf{I}_N)^{-1}\hat{\mathbf{A}}^\top[k](\hat{\mathbf{A}}[k]\mathbf{Z} + \mathbf{B}\mathbf{X} - \mathbf{Z}). \tag{5.21}$$

Likewise, given $\hat{\mathbf{E}}[k+1]$, the adjacency matrix is updated as

$$\hat{\mathbf{A}}[k+1] = \arg\min_{\mathbf{A}} \|\mathbf{Z} - \mathbf{A}(\mathbf{Z} - \hat{\mathbf{E}}[k+1]) - \mathbf{B}\mathbf{X}\|_F^2 + \lambda \|\mathbf{A}\|_1 \tag{5.22}$$

which is strongly convex and can be solved via proximal gradient iterations reaching the global optimum. The derivation of the algorithm is omitted here, see [70] for details.

Note that, the operation in (5.21) will cost $\mathrm{O}(N^2T)$, for $T \geq N$, in terms of time complexity, while the minimum of (5.22) will be reached in (worst-case) $\mathrm{O}(1/\varepsilon)$ iterations (or $O(1/\sqrt{\varepsilon})$ using fast iterative shrinkage-thresholding algorithms) if ε is the precision of the solution, and each row of $\mathbf{A}$ can be computed in parallel, see [70]. In particular, for each row of $\mathbf{A}$, the proximal gradient algorithm will entail matrix-vector multiplication and soft thresholding operations, that when the number of iterations needed for the proximal gradient algorithm to converge is relatively smaller than N, they are negligible wrt $\mathrm{O}(N^2T)$ of (5.21).[2]

Note that, if $\mathbf{B}$ is also a variable to be estimated, problem (5.20) is still per-block convex, and $\mathbf{B}$ can be readily found as in [70]. Under regularity conditions the alternating minimization method is guaranteed to converge at least to a stationary point, as asserted in the following proposition.

Proposition 3. *The iterates in* (5.21) *and* (5.22) *converge monotonically at least*

[2] In the numerical test we observe that the soft thresholding algorithms typically needs only a few iterations for convergence.

to a stationary point of problem (5.20).

Proof. See [153]. ■

5.3.3 Topology ID with sparse signal perturbations

In the previous sections, we have analyzed the cases where perturbations affect all nodal measurements. However, in several settings, only a small subset of nodes can be influenced. For example, in a heterogeneous network, some devices, e.g. sensors, may be less reliable than others. In this case, sparsity of the signal perturbations is well motivated. Introducing a sparse regularizer yields the sparse TLS (sparseTLS) SEM

$$\{\hat{\mathbf{A}}, \hat{\mathbf{E}}\} = \arg\min_{\mathbf{A},\mathbf{E}} \|\mathbf{Z} - \mathbf{A}(\mathbf{Z} - \mathbf{E}) - \mathbf{B}\mathbf{X}\|_F^2 + \lambda_E \|\mathbf{E}\|_1 + \lambda_A \|\mathbf{A}\|_1 \tag{5.23a}$$

$$\text{s. to} \quad a_{ii} = 0, \; i = 1, \ldots, N \tag{5.23b}$$

where $\lambda_A > 0$ and $\lambda_E > 0$ are sparsity promoting scalars.

In some applications such as sensor networks, we may even know which nodes are the more sensitive or vulnerable, which prompts us to leverage additional structure, namely the nonzero pattern of the error matrix. Hence, we write $\mathbf{E}$ as

$$\mathbf{E} = \mathscr{S}(\boldsymbol{\upsilon}) = \sum_{e=1}^{N_E} \upsilon_e (\boldsymbol{n}_e \cdot \boldsymbol{t}_e^\top) \tag{5.24}$$

where $\boldsymbol{\upsilon} := [\upsilon_1, \ldots, \upsilon_{N_E}]^\top$ is the collection of the nonzero values of $\text{vec}(\mathbf{E}^\top)$; the $N \times 1$ vector $\boldsymbol{n}_e$ has all zero entries except one that equals unity in the node affected by the e-th error value; and, $\boldsymbol{t}_e$ is the $T \times 1$ vector of all zeros except one that equals unity in the observation instant of the e-th error value. The structured error (S)TLS-SEM is then formulated as

$$\{\hat{\mathbf{A}}, \hat{\boldsymbol{\upsilon}}\} = \arg\min_{\mathbf{A},\boldsymbol{\upsilon}} \quad \|\mathbf{Z} - \mathbf{A}(\mathbf{Z} - \sum_{e=1}^{N_E} \upsilon_e (\boldsymbol{n}_e \cdot \boldsymbol{t}_e^\top)) - \mathbf{B}\mathbf{X}\|_F^2 + \lambda_E \|\boldsymbol{\upsilon}\|_2^2 + \lambda_A \|\mathbf{A}\|_1$$

$$\text{s. to} \qquad a_{ii} = 0, \forall i \tag{5.25}$$

where $\lambda_E > 0$ and $\lambda_A > 0$. The STLS-SEM problem is still only per-block convex, but can be solved by alternating minimization, as in the previous subsection.

5.4 Signal inference with topology perturbations

In this section, we study another problem that oftentimes arises in graph-related applications, that is, graph signal inference. In fact, in many cases, signals over all the nodes may not be available, due to, e.g., energy-saving or privacy reasons. This prompts to reconstruct signals over the unobserved nodes, when the graph topologies is given. However, such given graph topology may be perturbed, due to, e.g., link outages, in communication or power networks. This motivates the goal of this section to recover $\mathbf{Y}$, given a possibly perturbed adjacency matrix and the signal observed over a subset of nodes, indexed by $\mathcal{S}_t$ at each instant t. The observation model can then be written as

$$\boldsymbol{\psi}_t = \mathbf{D}_{\mathcal{S}_t}(\boldsymbol{y}_t + \boldsymbol{\varepsilon}_t)\,, \quad t = 1, \ldots, T \tag{5.26}$$

where $\mathbf{D}_{\mathcal{S}_t} := \text{diag}(d_{11}^{(t)}, \ldots, d_{NN}^{(t)})$, and $d_{ii}^{(t)} = 1$ if $i \in \mathcal{S}_t$, and zero otherwise; $\boldsymbol{\varepsilon}_t \in \mathbb{R}^N$ denotes the observation error; and, $\boldsymbol{\psi}_t \in \mathbb{R}^N$ represents the observation at time t, with $|\mathcal{S}_t| := M < N$ nonzero entries. For simplicity in exposition, M is considered fixed over time, but it can be generalized as time-varying.

Denoting with $\mathbf{A}_0$ the given nominal adjacency matrix, and $\boldsymbol{\Delta} \in \mathbb{R}^{N\times N}$ the topology perturbation matrix, the linear SEM in (5.2) becomes

$$\mathbf{Y} = (\mathbf{A}_0 - \boldsymbol{\Delta})\mathbf{Y} + \mathbf{B}\mathbf{X} \tag{5.27}$$

where $\mathbf{A}_0 - \boldsymbol{\Delta}$ is the perturbed adjacency matrix. As in the previous section, we consider $\mathbf{B}\mathbf{X}$ given, e.g. acquired from historical data or $\mathbf{B}\mathbf{X} = 0$ when $\mathbf{X}$ is not present, since the focus of the present section is to identify $\boldsymbol{\Delta}$ and $\{\boldsymbol{y}_t\}_{t=1}^T$. Exploiting the TLS method to account for topology perturbations, the topology perturbation aware TLS-SEM can be written as (cf. (5.26) and (5.27))

$$\{\hat{\boldsymbol{\Delta}}, \hat{\mathbf{Y}}\} = \arg\min_{\boldsymbol{\Delta},\mathbf{Y}} \quad \lambda_1\|\boldsymbol{\Delta}\|_1 + \lambda_2 \sum_{t=1}^{T} \|\boldsymbol{\psi}_t - \mathbf{D}_{\mathcal{S}_t}\boldsymbol{y}_t\|_2^2 + \|\mathbf{Y} - (\mathbf{A}_0 - \boldsymbol{\Delta})\mathbf{Y} - \mathbf{B}\mathbf{X}\|_F^2 \tag{5.28a}$$

$$\text{s.to} \quad [\boldsymbol{\Delta}]_{ii} = 0, \quad i = 1, \ldots, N \tag{5.28b}$$

where the ℓ_1-norm promotes sparsity of the perturbed links. In addition to sparsity, it has been shown that the elastic net regularizer [154] leads to improved recovery when the network weights are highly correlated [72]. Motivated by this, the elastic

norm regularized TLS (elTLS) approach to signal recovery yields

$$\{\hat{\mathbf{\Delta}}, \hat{\mathbf{Y}}\} = \arg\min_{\mathbf{\Delta},\mathbf{Y}} \sum_{t=1}^{T} \|\boldsymbol{\psi}_t - \mathbf{D}_{\mathcal{S}_t}\boldsymbol{y}_t\|_2^2 + \lambda_{1\Delta}\|\mathbf{\Delta}\|_1 + \lambda_{2\Delta}\|\mathbf{\Delta}\|_F^2 + \lambda_Y \|\mathbf{Y} - (\mathbf{A}_0 - \mathbf{\Delta})\mathbf{Y} - \mathbf{B}\mathbf{X}\|_F^2 \quad \text{s. to } [\mathbf{\Delta}]_{ii} = 0, \quad i = 1, \ldots, N \tag{5.29}$$

where $\lambda_{1\Delta} > 0$, $\lambda_{2\Delta} > 0$, and $\lambda_Y > 0$.

The costs in (5.28) and (5.29) are both per-block convex, and can be solved iteratively via alternating minimization with guaranteed convergence to at least a stationary point, as argued in Prop. 3.

5.4.1 Structured and weighted TLS under topology perturbations

In this subsection, we solve the signal recovery problem having additional information on the structure of the nominal adjacency matrix along with prior information on the perturbations. In this case, our aim is to formulate a structured and weighted TLS problem (cf. Sec. 5.2.2) for the signal inference task under topology perturbations. Denoting with L the number of (directed) links of the nominal graph and $\boldsymbol{\omega} := [\omega_1, \ldots, \omega_L]^\top$ the vector collecting the nonzero edge weights, the nominal adjacency matrix can be represented as (cf. Definition 1)

$$\mathbf{A}_0 = \mathscr{S}(\boldsymbol{\omega}) := \sum_{l=1}^{L} \omega_l (\boldsymbol{s}_{u_l}\boldsymbol{s}_{v_l}^\top) \tag{5.30}$$

where (u_l, v_l) are the incident nodes of link l, and $\boldsymbol{s}_i$ the $N \times 1$ i-th canonical vector. The structure $\mathscr{S}(\boldsymbol{\omega})$ accounts for the L nonzero entries of $\mathbf{A}_0$. Assuming that perturbations occur only on the existing links, it will also allow us to reduce the number of unknown perturbations from N^2 to L.

According to Sec. 5.2.2 and Eqn. (5.30), we will parameterize $\mathbf{A}_0$ using $\boldsymbol{\omega}$, and correspondingly $\mathbf{\Delta}$ via $\boldsymbol{\nu} := [\nu_1, \ldots, \nu_L]^\top$, whose nonzero entries represent a failure or error in the edge weight. Thus, the perturbed adjacency matrix is given by

$$\mathbf{A}_0 - \mathbf{\Delta} = \mathscr{S}(\boldsymbol{\omega} - \boldsymbol{\nu}) := \sum_{l=1}^{L} (\omega_l - \nu_l)(\boldsymbol{s}_{u_l}\boldsymbol{s}_{v_l}^\top). \tag{5.31}$$

In certain cases, extra information such as the link failure probabilities $\{\pi_l\}_{l=1}^L$ and the observation noise variance $\{\sigma_i^2\}_{i=1}^N$ can be available across nodes. Such prior information can be collected after observing the network over time and recording the occurrence of failures, as well as the statistics of the measurement errors.

Let $\mathbf{W}_A := \text{diag}(r(\pi_1)\dots r(\pi_L))$ denote the topology reliability weight matrix, where $r(\pi_l)$ is a known function of π_l, e.g. $r(\pi_l) = \pi_l^{-1}$, and likewise $\mathbf{W}_\Psi := [\text{diag}(\sigma_1^2 \dots \sigma_N^2)]^{-1}$ for the measurement errors. In order to use an SWTLS cost (cf. Sec. 5.2.2), we replace the first two terms in (5.28a) with the weighted ℓ_1-norm of the topology error vector $\|\mathbf{W}_A \boldsymbol{\nu}\|_1$, and the sum of the weighted ℓ_2-norm of the observation errors $\sum_{t=1}^T \|\boldsymbol{\psi}_t - \mathbf{D}_{\mathcal{S}_t}\boldsymbol{y}_t\|_{\mathbf{W}_\Psi}^2$. Combining with (5.31), the regularized SWTLS-based SEMs can be written as

$$\{\hat{\boldsymbol{\nu}}, \hat{\mathbf{Y}}\} = \arg\min_{\boldsymbol{\nu},\mathbf{Y}} \quad \lambda_1 \|\mathbf{W}_A\boldsymbol{\nu}\|_1 + \lambda_2 \sum_{t=1}^T \|\boldsymbol{\psi}_t - \mathbf{D}_{\mathcal{S}_t}\boldsymbol{y}_t\|_{\mathbf{W}_\Psi}^2 + \|\mathbf{Y} - \sum_{l=1}^L (\omega_l - \nu_l)(\boldsymbol{s}_{u_l}\boldsymbol{s}_{v_l}^\top)\mathbf{Y} - \mathbf{B}\mathbf{X}\|_F^2 \tag{5.32}$$

which can be solved via alternating minimization. Given $\hat{\boldsymbol{\nu}}[k]$ from iteration k, and exploiting the separability across columns of $\mathbf{Y}$, the graph signal at $k+1$ is reconstructed per slot t as

$$\hat{\boldsymbol{y}}_t[k+1] = \arg\min_{\boldsymbol{y}_t} \quad \lambda_2 \|\boldsymbol{\psi}_t - \mathbf{D}_{\mathcal{S}_t}\boldsymbol{y}_t\|_{\mathbf{W}_\Psi}^2 + \|\boldsymbol{y}_t - \sum_{l=1}^L (\omega_l - \hat{\nu}_l[k]) y_{v_l,t}\boldsymbol{s}_{u_l} - \mathbf{B}\boldsymbol{x}_t\|_2^2 \tag{5.33}$$

where $\boldsymbol{s}_{v_l}^\top \boldsymbol{y}_t = y_{v_l,t}$ because $\boldsymbol{s}_{v_l}$ is defined as canonical vector.

The minimization in (5.33) leads to the closed-form update

$$\hat{\boldsymbol{y}}_t[k+1] = (\mathbf{C}^\top[k]\mathbf{C}[k] + \lambda_2 \mathbf{D}_{\mathcal{S}_t}^\top \mathbf{W}_\Psi \mathbf{D}_{\mathcal{S}_t})^{-1}(\mathbf{C}^\top[k]\mathbf{B}\boldsymbol{x}_t + \lambda_2 \mathbf{D}_{\mathcal{S}_t}^\top \mathbf{W}_\Psi \boldsymbol{\psi}_t), \quad t = 1, \dots, T \tag{5.34}$$

with $\mathbf{C}[k] := (\mathbf{I}_N - \sum_{l=1}^L (\omega_l - \hat{\nu}_l[k])\boldsymbol{s}_{u_l}\boldsymbol{s}_{v_l}^\top)$.

Given $\hat{\mathbf{Y}}[k+1] = [\hat{\boldsymbol{y}}_1[k+1], \dots, \hat{\boldsymbol{y}}_t[k+1]]$, we can exploit in (5.32) the separability across rows of $\mathbf{Y}$. Let L_n denote the number of neighbors of node n, and $\boldsymbol{\omega}_n := [\omega_1^{(n)}, \dots, \omega_{L_n}^{(n)}]^\top$ and $\boldsymbol{\nu}_n := [\nu_1^{(n)}, \dots, \nu_{L_n}^{(n)}]^\top$ the vectors collecting edge and error weights in the neighborhood of n. Similarly, let the diagonal matrix $\mathbf{W}_A^n$ be the n-th block of the block diagonal matrix $\mathbf{W}_A$. With $\boldsymbol{\gamma}_n^\top$ and $\boldsymbol{x}_n^\top$ representing the n-th row of $\mathbf{Y}$ and $\mathbf{X}$, respectively, $\hat{\boldsymbol{\nu}}_n[k+1]$ can be updated as

$$\hat{\boldsymbol{\nu}}_n[k+1] = \arg\min_{\boldsymbol{\nu}_n} \quad \lambda_1 \|\mathbf{W}_A^n \boldsymbol{\nu}_n\|_1 + \|\hat{\boldsymbol{\gamma}}_n[k+1] - (\hat{\mathbf{Y}}_n[k+1])^\top(\boldsymbol{\omega}_n - \boldsymbol{\nu}_n) - b_{nn}\boldsymbol{x}_n\|_2^2 \tag{5.35}$$

where $\mathbf{Y}_n$ is a submatrix of $\mathbf{Y}$ formed by the rows corresponding to the neighboring

nodes of n in the nominal topology. Sub-problem (5.35) is again convex, but not differentiable, which suggests an iterative proximal gradient solver.

The time complexity of (5.34) is $\mathrm{O}(N^3)$ and each $\boldsymbol{y}_t, t = 1, \ldots, T$ can be estimated in parallel. The minimum of (5.35) can be reach in (worst-case) $\mathrm{O}(1/\varepsilon)$ iterations (or $O(1/\sqrt{\varepsilon})$ using fast iterative shrinkage-thresholding algorithms) if ε is the precision of the solution [70] and each $\boldsymbol{\nu}_n$ can be computed in parallel. In particular, the proximal gradient solver employed to solve (5.35) will entail matrix-vector multiplication and soft thresholding operations. When $\{L_n\}_n$ are smaller than N, the complexity of (5.35) is negligible wrt $\mathrm{O}(N^3)$.

5.4.2 Identifiability of topology perturbations

The goal of this subsection is to investigate the conditions that ensure uniqueness in identifying the perturbation vector $\boldsymbol{\nu}$ in the noise-free[3] structured topology perturbation model in Sec. 5.4.1 (cf. (5.27) and (5.31)). To this end, consider the n-th row of the $N \times T$ matrix $\mathbf{Y}$ in (5.27), which can be expressed as

$$\boldsymbol{y}_n^\top = (\boldsymbol{a}_n^\top - \boldsymbol{\delta}_n^\top)\mathbf{Y} + b_{nn}\boldsymbol{x}_n^\top \tag{5.36}$$

with $\boldsymbol{a}_n^\top$ and $\boldsymbol{\delta}_n^\top$ likewise denoting the nth rows of $\mathbf{A}_0$ and $\boldsymbol{\Delta}$, respectively. With L_n being the number of neighbors of node n, we define the $1 \times L_n$ vector $\boldsymbol{\omega}_n^\top$ formed after removing the zero entries of $\boldsymbol{a}_n^\top$ per node n; and similarly the $1 \times L_n$ vector $\boldsymbol{\nu}_n^\top$ after removing the corresponding entries of $\boldsymbol{\delta}_n^\top$. Using these definitions, (5.36) can be simplified to

$$\boldsymbol{y}_n^\top = (\boldsymbol{\omega}_n^\top - \boldsymbol{\nu}_n^\top)\mathbf{Y}_n + b_{nn}\boldsymbol{x}_n^\top \tag{5.37}$$

where $\mathbf{Y}_n$ is an $L_n \times T$ submatrix obtained after removing the rows of $\mathbf{Y}$ corresponding to the zero entries of $\boldsymbol{a}_n^\top$.

To take into account the number of samples T_n per node n, we further introduce the $T_n \times T$ matrix $\boldsymbol{\mathcal{D}}_n$ obtained after removing the all-zero rows of the $T \times T$ diagonal matrix $\mathrm{diag}\{d_{nn}^{(1)} \ldots d_{nn}^{(T)}\}$, where $d_{nn}^{(t)} = 1$ if node is sampled at slot t, and $d_{nn}^{(t)} = 0$ otherwise. Multiplying $\boldsymbol{\mathcal{D}}_n$ from the right with a matrix, selects T_n (out of T) rows corresponding to the time-slot indices that node n is sampled. We rely on $\boldsymbol{\mathcal{D}}_n$ to form the $T_n \times 1$ vector $\boldsymbol{\phi}_n := \boldsymbol{\mathcal{D}}_n\boldsymbol{y}_n$, which after employing the transposed version of (5.37) can be expressed as

$$\boldsymbol{\phi}_n = \boldsymbol{\mathcal{D}}_n[\mathbf{Y}_n^\top(\boldsymbol{\omega}_n - \boldsymbol{\nu}_n) + b_{nn}\boldsymbol{x}_n] \,. \tag{5.38}$$

Motivated by the fact that e.g., adversaries can compromise only a few links

[3] Absence of noise ($\boldsymbol{\epsilon}_t \equiv \mathbf{0}\ \forall t$) is typically assumed in identifiability studies, in order to isolate (non) uniqueness issues from estimation errors.

per node n, it is reasonable to explore identifiability conditions when the sought perturbation vector $\boldsymbol{\nu}_n$ is sparse with p_n ($< L_n$) nonzero entries.

Arguing by contradiction to establish that $\boldsymbol{\nu}_n$ can be uniquely identified from (5.38), we will suppose that there exists another $L_n \times 1$ vector $\boldsymbol{\xi}_n \neq \boldsymbol{\nu}_n$ with p_n nonzero entries satisfying $\boldsymbol{\phi}_n = \boldsymbol{\mathcal{D}}_n[\mathbf{Y}_n^\top(\boldsymbol{\omega}_n - \boldsymbol{\xi}_n) + b_{nn}\boldsymbol{x}_n]$. Subtracting the latter from (5.38), yields

$$\mathbf{0} = \boldsymbol{\mathcal{D}}_n \mathbf{Y}_n^\top (\boldsymbol{\nu}_n - \boldsymbol{\xi}_n) \; . \tag{5.39}$$

Clearly, the difference $\boldsymbol{\nu}_n - \boldsymbol{\xi}_n$ of the two p_n-sparse vectors $\boldsymbol{\nu}_n$ and $\boldsymbol{\xi}_n$, has at most $2p_n$ nonzero entries; and with $p_{\max} := \max_{n=1,\ldots,N} p_n$, we have that the differences $\{\boldsymbol{\nu}_n - \boldsymbol{\xi}_n\}$ across all nodes can have at most $2p_{\max}$ nonzero entries.

To proceed with specifying identifiability conditions of our sparse vector differences, we will need the following definition of the Kruskal rank of a matrix.

Definition 2 [155]. *The Kruskal rank of a matrix* $\mathbf{M}$*, denoted as* $\mathrm{kr}(\mathbf{M})$*, is defined as the maximum number* ρ *such that any combination of* ρ *columns of* $\mathbf{M}$ *constitutes a full-rank submatrix.*

Since the $2p_{\max}$ nonzero entries of $\boldsymbol{\nu}_n - \boldsymbol{\xi}_n$ can occur in *any* subset of this vector difference, we deduce that having $\mathrm{kr}(\boldsymbol{\mathcal{D}}_n \mathbf{Y}_n^\top) \geq 2p_{\max}$, guarantees that any $2p_{\max}$ columns of $\boldsymbol{\mathcal{D}}_n \mathbf{Y}_n^\top$ submatrix will be full rank. Under this condition, we find from (5.39) that $\boldsymbol{\nu}_n = \boldsymbol{\xi}_n$, which leads to contradiction. Summarizing, we have established the following result.

Proposition 4. *If* $\mathrm{kr}(\boldsymbol{\mathcal{D}}_n \mathbf{Y}_n^\top) \geq 2\max_n p_n$*, the* p_n*-sparse perturbation vector* $\boldsymbol{\nu}_n$ *is identifiable from* (5.38)*, for* $n = 1, \ldots, N$*.*

Intuitively, Proposition 4 asserts that sparsity in the perturbation renders the bound on the Kruskal rank easier to satisfy, and thus ensure identifiability. As a word of caution, it is worth mentioning that finding the Kruskal rank of a matrix is combinatorially complex in its dimensions [155]. In addition, this condition may be hard to check since matrix $(\boldsymbol{\mathcal{D}}_n \mathbf{Y}_n^\top)$ is not always observed in practice.

5.5 Numerical tests

In this section, we test with a number of synthetic and real data the proposed TLS-based algorithms, both for topology ID under signal perturbations, and graph signal inference under topology perturbations. The regularization parameters are selected by grid search cross-validation for all the algorithms.

5.5.1 Synthetic tests for topology ID under signal perturbations

Bisection-based versus alternating descent iterations

In this test, the adjacency matrix $\mathbf{A}^{(0)}$ is simulated as a 6×6 matrix of binary 0-1 entries with 2 nonzero entries per row, and $\mathbf{Z} = \mathbf{Y} + \mathbf{E}$, with $[\mathbf{E}]_{ij} \sim \mathcal{N}(0, 1 \times 10^{-2})$, while the observation $\mathbf{Y} = (\mathbf{I}_N - \mathbf{A}^{(0)})^{-1}\mathbf{B}\mathbf{X}$, with $\mathbf{B} = \mathbf{I}_N$ and $[\mathbf{X}]_{ij} \sim \mathcal{U}(0, 1.5)$. Alg. 1 is tested with $\mu = 5$, $\boldsymbol{a}_L = \mathbf{0}$, and $\boldsymbol{a}_U = \mathbf{1}$.

Fig. 5.2 illustrates the performance reached by the alternating descent (AD) iterations in (5.21) and (5.22), the conventional least-squares (LS) SEM [70, 134], where perturbations $\mathbf{E}$ on the observations are not considered, and the BB scheme of Sec. 5.3.1, all in terms of $\text{MSE}_A = \sum_{ij}(\hat{a}_{ij} - a_{ij})^2/N^2$, for different values of ε. The ε-optimal BB solver performs better as ε decreases, while the solutions of the AD and LS-SEM schemes do not depend on ε, and hence are constant $\forall\varepsilon$. For $\varepsilon < 10^{-2}$, both perturbation-aware methods outperform the LS-SEM method, and note that the BB method slightly outperforms the AD one. However, the BB algorithm is computationally demanding.

Fig. 5.3 depicts the runtime of the three competing algorithms in seconds,[4] when $\varepsilon = 10^{-3}$, and demonstrates that the AD method is computationally more efficient than the BB scheme. For this reason, the following tests will include only the AD iteration, which will be henceforth abbreviated as TLS-SEM. The numerical results of this subsection were obtained by averaging over 100 Monte Carlo realizations of $\mathbf{X}$ and $\mathbf{E}$.

Topology ID under signal perturbations

In this subsection, we test the performance of the AD solver (5.20) for simulated data, and compare it with LS-SEM. We generated a Kronecker graph with $N = 64$ as in [156], and $\mathbf{B} = \mathbf{I}_N$ was assumed given. We generated random matrices with uniformly distributed entries $[\mathbf{X}]_{it} \sim \mathcal{U}[0, 1.5]$, and Gaussian distributed entries $[\mathbf{E}]_{it} \sim \mathcal{N}(0, \sigma_E^2)$. Matrices $\mathbf{Y}$ and $\mathbf{Z}$ were then constructed according to (5.2) and (5.4), with $T = 120$, while λ was selected via cross-validation. Fig. 5.4 shows the MSE_A performance of LS-SEM and TLS-SEM for different SNR(dB):= $10\log_{10}(\|\bar{\boldsymbol{y}}\|_2^2/(N\sigma_E^2))$ and $\bar{\boldsymbol{y}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{y}_t$. It can be observed that TLS-SEM outperforms LS-SEM. Fig. 5.5 shows the performance versus different number of observations T, with fixed $\sigma_E = 0.2$. Evidently, TLS-SEM outperforms LS-SEM even when the number of observations is small. These numerical results were obtained by averaging over 100 Monte Carlo realizations of $\mathbf{X}$ and $\mathbf{E}$.

[4] This experiment was run on a machine with i5-6200U @2.30 GHz CPU, and 8GB of RAM.

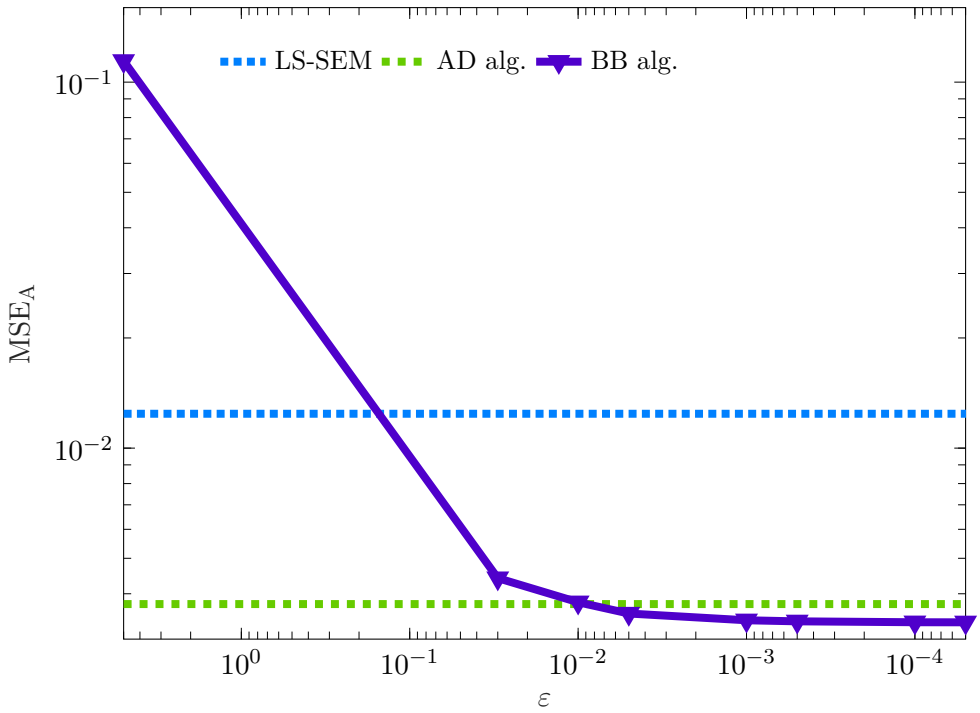


Figure 5.2. MSE_A across ε, obtained by the ε-optimal algorithm. This result is compared with the LS-SEM, and with the AD (TLS-SEM) iteration.

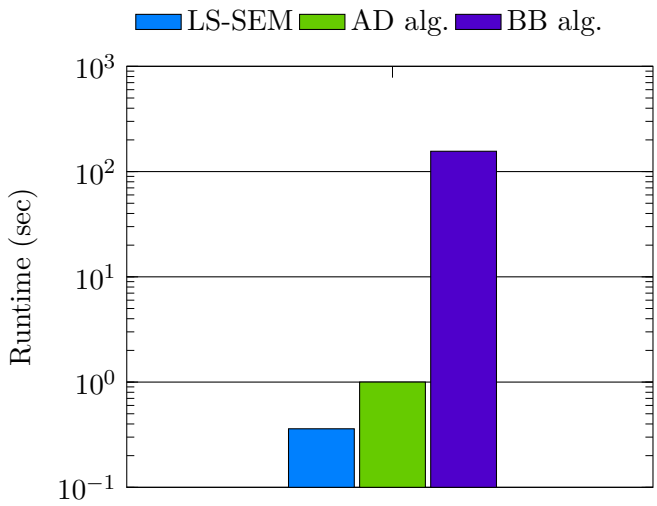


Figure 5.3. Runtime in seconds

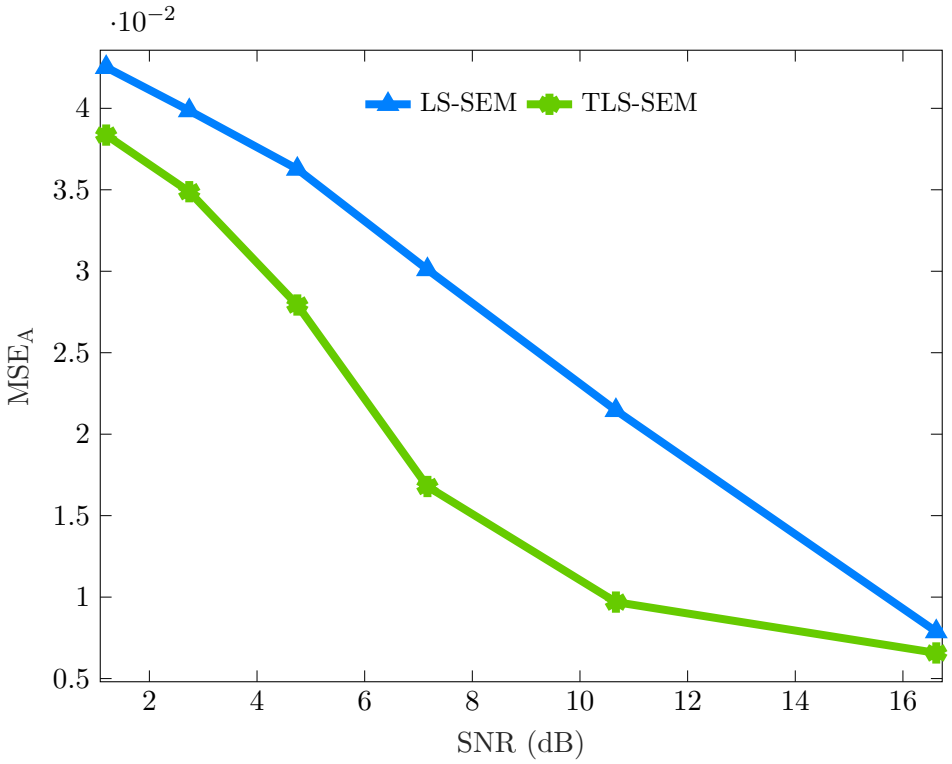


Figure 5.4. MSE$_A$ versus SNR.

Sparse signal perturbation

In this experiment, we tested the performance of sparse TLS in (5.23) and (5.25). An adjacency matrix is generated as a Kronecker graph with $N = 64$ with binary entries. Entries of $\mathbf{X}$ were generated as uniform i.i.d. random variables, that is $[\mathbf{X}]_{ij} \sim \mathcal{U}[0, 1.5]$, and $\mathbf{B} = \mathbf{I}_N$. Furthermore, we set $\mathbf{Z} = \mathbf{Y} + \mathbf{E}$, where $\mathbf{Y} = (\mathbf{I}_N - \mathbf{A})^{-1}\mathbf{B}\mathbf{X}$, and the sparse $\mathbf{E}$ was generated such that $\mathbf{E}$ has zero entries on $N_0 = N - 8$ selected rows, while the nonzero entries on the selected 8 rows were drawn from a uniform distribution over $[0, 0.3]$. Results were averaged over 100 realizations of $\mathbf{X}$ and $\mathbf{E}$.

Fig. 5.6 shows the performance of LS-SEM, TLS-SEM in (5.20), sparseTLS in (5.23) and STLS-SEM in (5.25), in terms of MSE$_A$ for different T. The TLS-SEM outperforms LS-SEM, and the performance gain increases as more data observations are collected. The result is obtained by averaging over 100 different realizations of $\mathbf{X}$ and $\mathbf{E}$.

5.5.2 Real data tests for topology ID with signal perturbations

Further, we present experiments on gene expression data to identify the underlying gene regulatory network. The real data were collected from 69 unrelated Nigerian

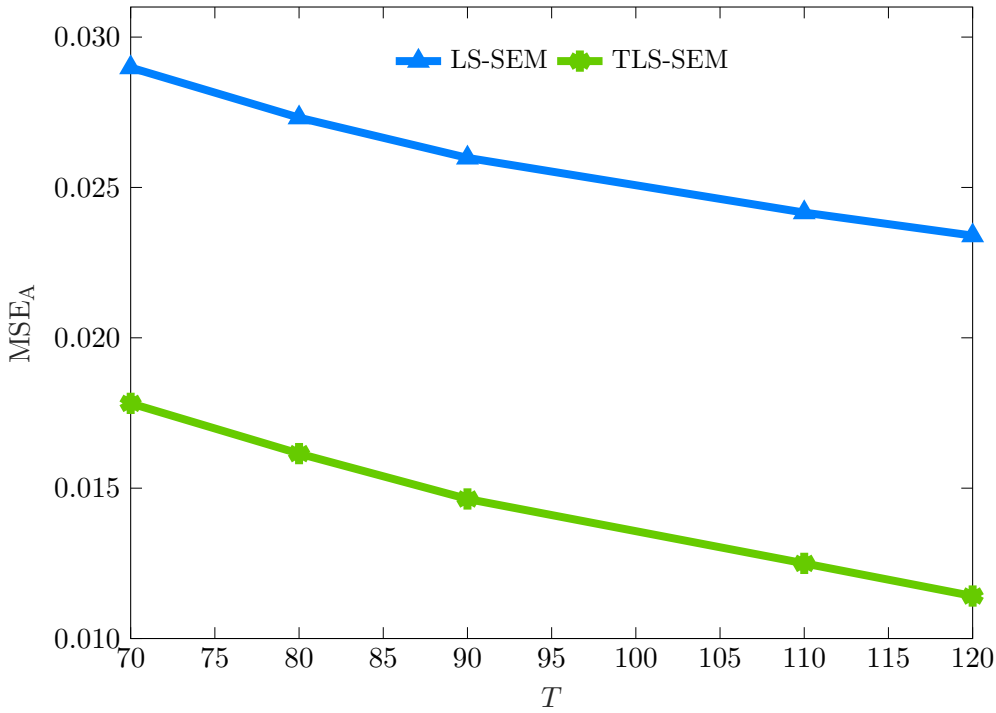


Figure 5.5. MSE$_A$ versus T.

individuals, under the International HapMap project [157]. From the 929 identified genes, expression levels and the genotypes of the expression quantitative trait loci (eQTLs) of 39 immune-related genes were selected and normalized; see [134] and [158] for further details. Genotypes of eQTLs were adopted as known exogenous inputs $\mathbf{X}$, and gene expression levels were treated as the endogenous variables $\mathbf{Y}$. The underlying network as well as the matrix $\mathbf{B}$, were inferred by adopting TLS-SEM, sparseTLS, and LS-SEM methods. Fig. 5.7 illustrates the fitting loss divided by the norm of the data $\mathbf{Z}$, as $\|\mathbf{Z} - \mathbf{AZ} - \mathbf{BX}\|_F^2 / \|\mathbf{Z}\|_F^2$ for LS-SEM, and $\|\mathbf{Z} - \mathbf{A}(\mathbf{Z} - \mathbf{E}) - \mathbf{BX}\|_F^2 / \|\mathbf{Z}\|_F^2$ for TLS-SEM, as a function of parameter λ_A. For all values of λ_A, i.e. the regularization parameter promoting the adjacency matrix sparsity, TLS-SEM and sparseTLS-SEM outperform the LS-SEM, which implies that the inferred matrix $\mathbf{A}$ fits the model better when the signal perturbations are taken into account. When λ_A reaches very large values, all approaches perform similarly since the regularization term $\lambda_A \|\mathbf{A}\|_1$ prevails on all the other terms of the cost functions and $\mathbf{A}$ becomes an all zero matrix. Furthermore, Fig. 5.8 illustrates the performance in terms of fitting error $\|\mathbf{Y} - \mathbf{AY} - \mathbf{BX}\|_F^2$, with $\mathbf{Y} = \mathbf{Z} - \mathbf{E}$ for TLS-SEM and sparseTLS, and $\mathbf{Y} = \mathbf{Z}$ for LS-SEM across values of λ_A. Again, TLS-SEM and sparseTLS-SEM outperform LS-SEM.

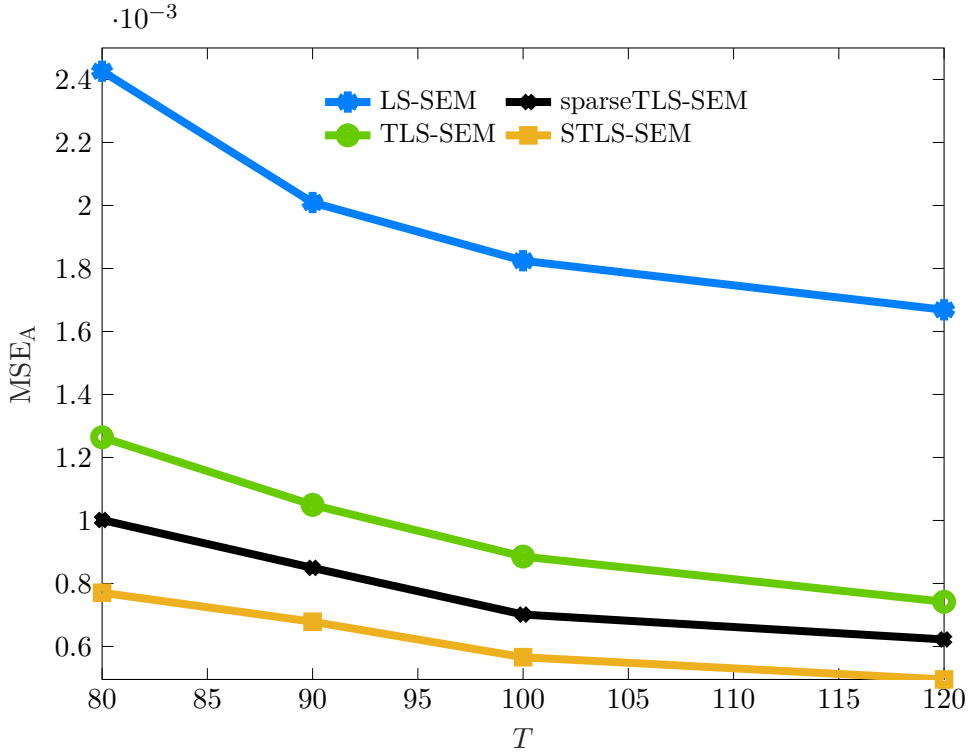


Figure 5.6. MSE_A as a function of T.

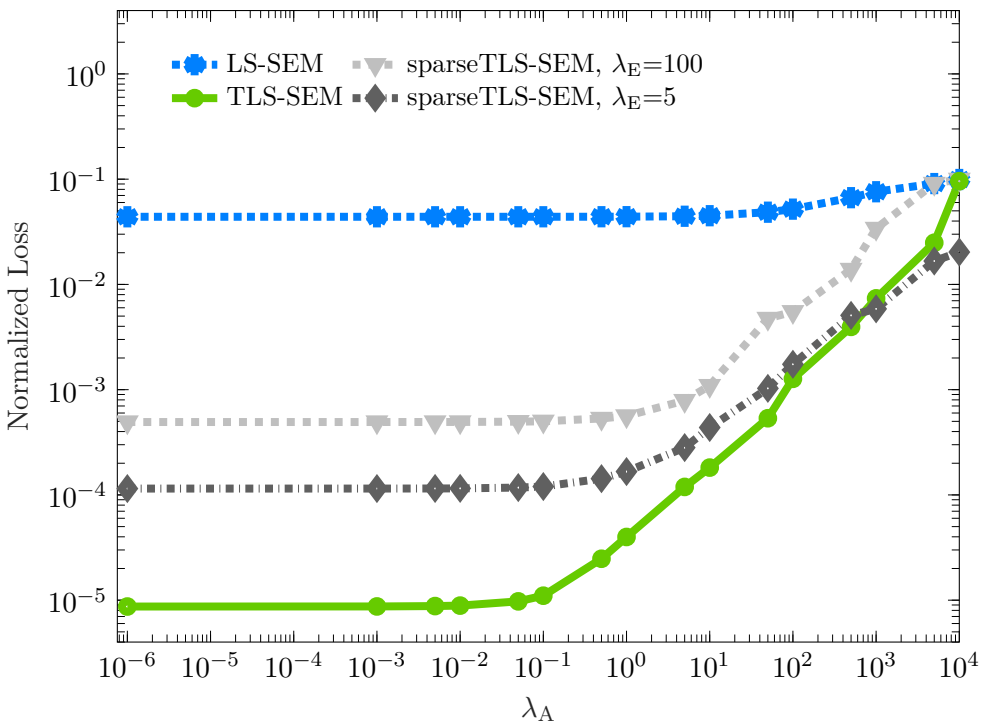


Figure 5.7. Loss function vs. λ_A for Gene regulatory network.

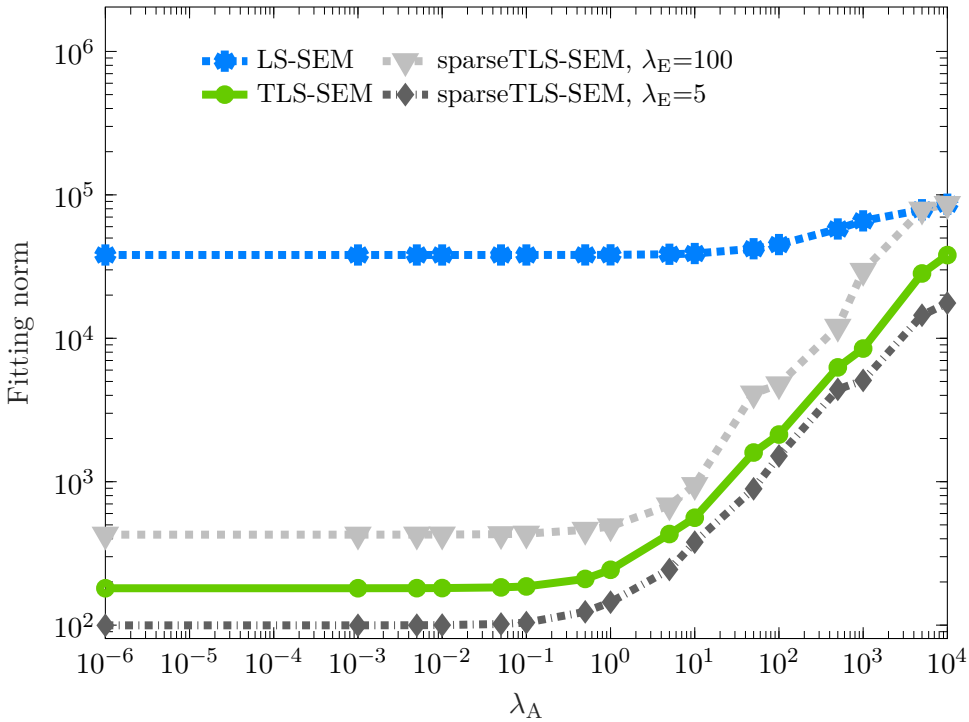


Figure 5.8. Fitting norm vs. λ_A for Gene regulatory network.

5.5.3 Signal inference under topology perturbations

In this subsection, we test the performance of the TLS algorithms in Sec. 5.4, and compare them with the conventional LS-SEM. In this setting, the topology is perturbed and the goal is to identify $\mathbf{Y}$ from a subset of observations. For this numerical test, a Kronecker graph with $N = 27$ is generated as before. With $T = 50$ and $\mathbf{B} = \mathbf{I}_N$, the entries of $\mathbf{X}$ and $\{\boldsymbol{\varepsilon}_t\}_{t=1}^T$ are again randomly drawn as $[\mathbf{X}]_{ij} \sim \mathcal{U}[0,3]$ and $\varepsilon_{ti} \sim \mathcal{N}(0, \sigma_i^2)$, $\forall t$. Moreover, for the perturbation matrix $\boldsymbol{\Delta}$, we model its entries $[\boldsymbol{\Delta}]_{ij}$ as Bernoulli$(\pi_l) \times [\mathbf{A}]_{ij}$, with $l := (v_i, v_j)$, which means that perturbations occur when one or more weighted links fail. In particular, $\pi_1 = \pi_2 = 0.9$, and $\pi_l \in [0.001, 0.02], l = 3, \ldots, L$, and we choose $r(\pi_l) = \frac{1}{\pi_l}$. Matrices $\mathbf{Y}$ and $\boldsymbol{\Psi} := [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_T]$ are then constructed according to (5.27) and (5.26), while λ_1 and λ_2 are selected via cross validation. Fig. 5.9 illustrates the performance of LS-SEM, TLS-SEM, structured TLS under topology perturbations (topSTLS-SEM), and SWTLS-SEM in terms of normalized mean-square error

$$\text{NMSE} = \frac{\|\hat{\mathbf{Y}} - \mathbf{Y}\|_F^2}{\|\mathbf{Y}\|_F^2}. \tag{5.40}$$

The results are obtained by averaging over 500 Monte Carlo realizations of $\mathbf{X}, \boldsymbol{\varepsilon}_t, \boldsymbol{\Delta}$, and $\mathbf{D}_{\mathcal{S}}$. Fig. 5.9 shows the performance as a function of the number of sampled

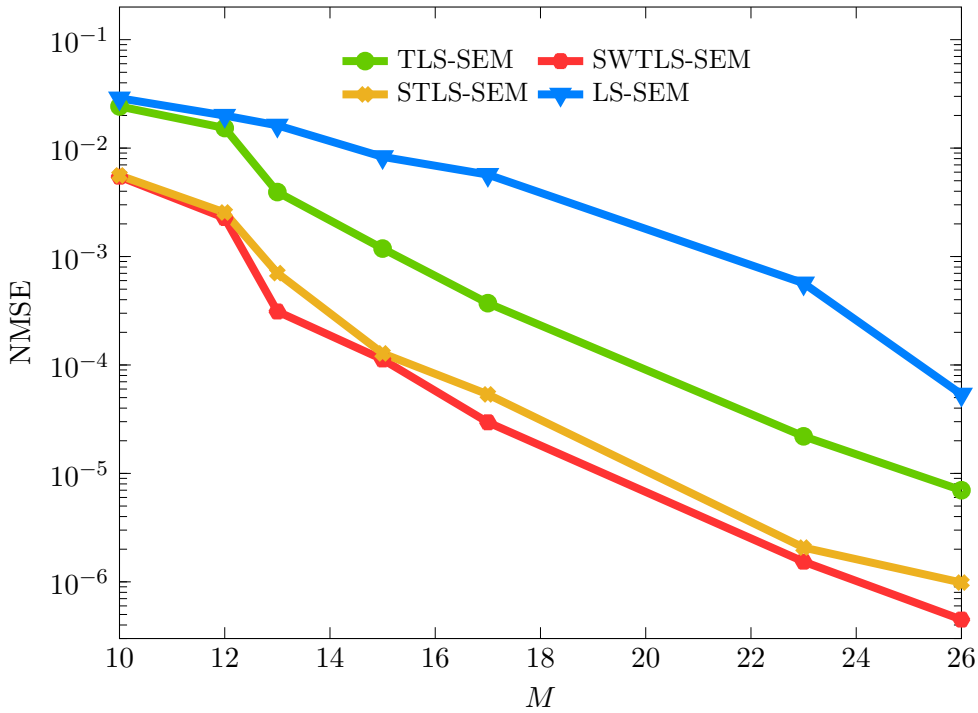


Figure 5.9. NMSE versus M

nodes M. Clearly, estimation performance improves as extra prior information is accounted for.

5.5.4 Real tests for signal inference with topology perturbations

Finally, we test here the proposed elTLS-based approach in (5.29) to recover the signal from a subset of noisy observations and a perturbed graph topology.

The real dataset consists of path delay measurements on the Internet2 backbone [159]. The network has 9 nodes and 26 directed links. The delays are available for $N = 70$ paths per minute. Set $\{y_{nt}\}$ contains a subset of delays in milliseconds per path n and minute slot t. The topologies are obtained based on the following three possible models.

M1. The paths connect origin-destination nodes by a series of links described by the path-link routing matrix $\Pi \in \{0,1\}^{N\times 26}$, whose (n,l) entry is $\Pi_{n,l} = 1$ if path n traverses link l, and 0 otherwise. A graph is constructed with each vertex corresponding to one of these paths, and with the time-invariant adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ given by

$$\mathbf{A}_{n,n'} = \frac{\sum_{l=1}^{26} \Pi_{n,l}\Pi_{n',l}}{\sum_{l=1}^{26} \Pi_{n,l} + \sum_{l=1}^{26} \Pi_{n',l} - \sum_{l=1}^{26} \Pi_{n,l}\Pi_{n',l}} \tag{5.41}$$

for $n, n' = 1, \ldots, N$ and $n \neq n'$. The edge weight model in (5.41) assigns greater weights to edges connecting vertices whose associated paths share more links. This is reasonable because paths with common links usually experience similar delays [160].
M2. For this model, a training phase is introduced based on a subset of the signal observations, collected in the matrix $\mathbf{Y}_{train}$, to estimate the adjacency as the solution of

$$\begin{aligned} \min_{\mathbf{A}} \ & \|\mathbf{Y}_{train} - \mathbf{A}\mathbf{Y}_{train}\|_F^2 \\ \text{s. to} \ & a_{ii} = 0, \ i = 1, \ldots, N \end{aligned} \tag{5.42}$$

where $\mathbf{Y}_{train} \in \mathbb{R}^{N \times T_{train}}$, with $T_{train} = 20$.
M3. The third topology is found as in (5.42), but the training signals used for training are corrupted by noise, that is, $\bar{\mathbf{Y}}_{train} := \mathbf{Y}_{train} + \mathbf{\Xi}$, with $[\mathbf{\Xi}]_{ij} \sim \mathcal{N}(0, \sigma_\xi^2)$; while σ_ξ^2 is chosen such that $10 \log_{10}(\|\bar{\boldsymbol{y}}_{train}\|_2^2 / (N\sigma_\xi^2)) = -8$ dB, where $\bar{\boldsymbol{y}}_{train} \in \mathbb{R}^N$ is the average of the columns of $\mathbf{Y}_{train}$. Solving problem (5.42) with $\bar{\mathbf{Y}}_{train}$ instead of $\mathbf{Y}_{train}$ gives rise to an alternative topology with an inherent model mismatch. The observation error in (5.26) is generated using $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, \sigma_\varepsilon^2 \mathbf{I}), \forall t$, and we define the SNR:= $10 \log_{10}(\|\bar{\boldsymbol{y}}\|_2^2 / (N\sigma_\varepsilon^2))$, with $\bar{\boldsymbol{y}} \in \mathbb{R}^N$ is the average of the columns of $\mathbf{Y}$.

Fig. 5.10 shows the NMSE versus the number of sampled nodes M when the topology is obtained from M1. It illustrates that the novel perturbation-aware elTLS-SEM outperforms the LS-SEM by accounting for the possible model mismatch.

Figs. 5.11 and 5.12 depict the NMSE as a function of observations T with adjacency matrices obtained via M2 and M3, respectively. Once again, perturbation-aware elTLS-SEM outperforms LS-SEM. The performance gain of elTLS-SEM in Fig. 5.11 is less evident than that in Figures 5.10 and 5.12 because the adjacency matrix is obtained exactly following the SEM. Results are averaged over 100 realizations.

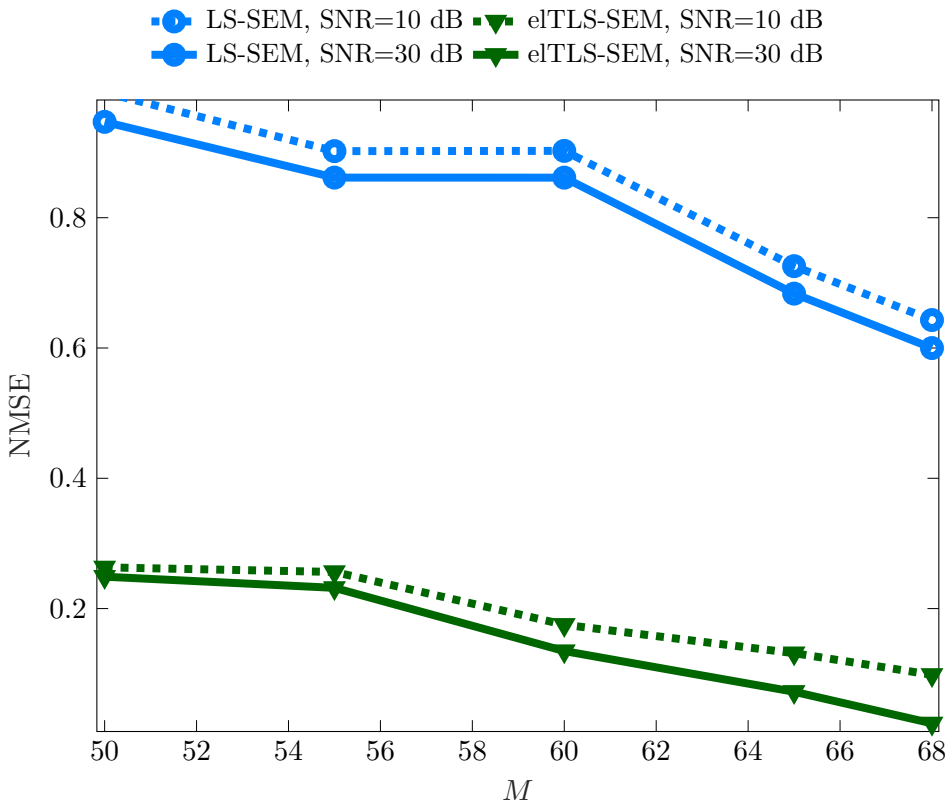


Figure 5.10. NMSE versus M, with $\mathbf{A}$ obtained via M1 and $T = 100$.

5.6 Conclusions

In summary, in this chapter, we have examined two challenging tasks over graphs, namely topology ID under signal perturbations, and signal inference under topology perturbations. To address the associated challenges, a spectrum of approaches based on total least-squares and structural equation models were developed. Moreover, structured and weighted variants of TLS-SEM were introduced to flexibly account for extra prior information. Numerical tests on both synthetic and real data demonstrated the efficacy of the proposed algorithms.

Future research directions include distributed implementation of TLS-SEM to accommodate large-scale graphs, and generalizations of perturbed SEMs to account for nonlinear or dynamic inter-dependencies.

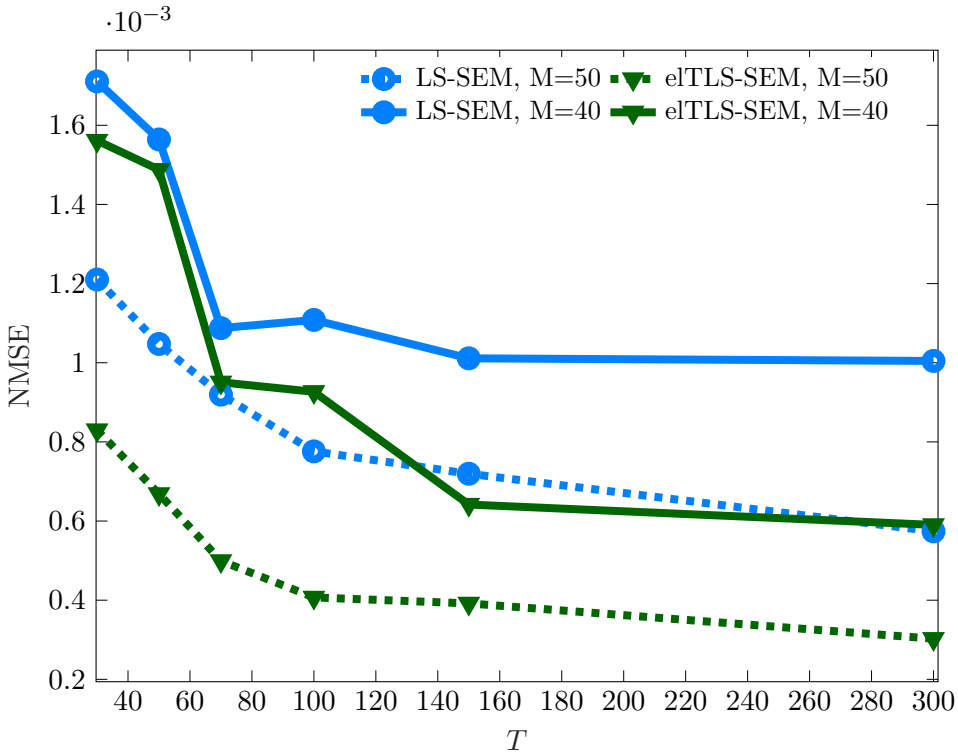


Figure 5.11. NMSE verus T, with $\mathbf{A}$ obtained via M2 and SNR= 30 dB.

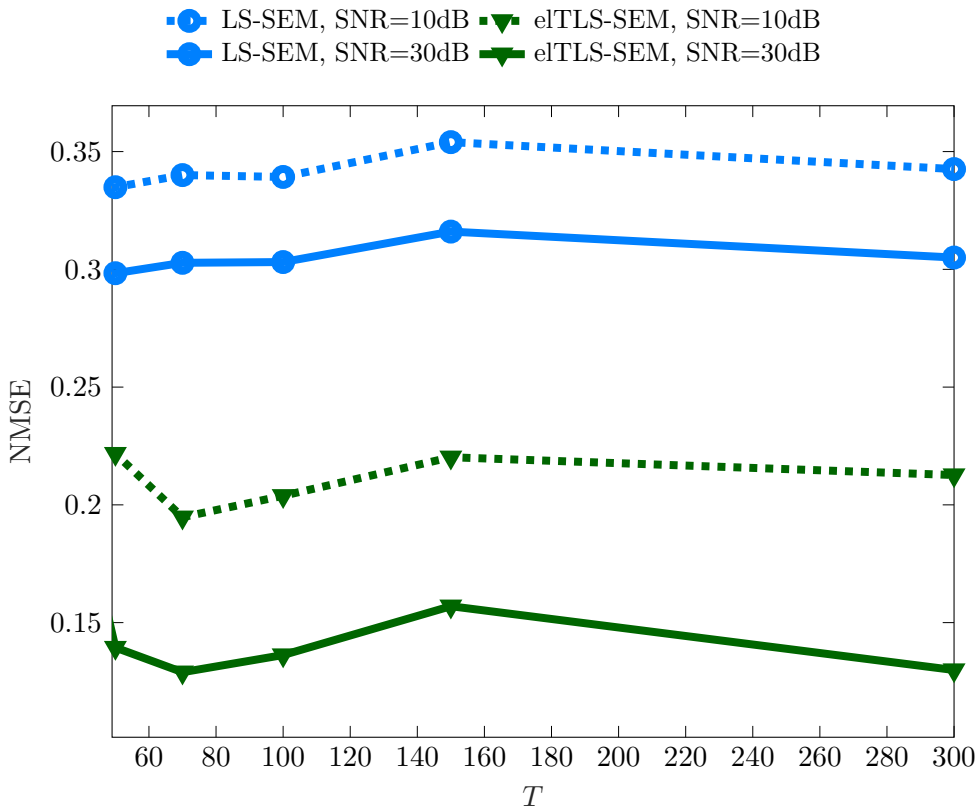


Figure 5.12. NMSE versus T, with $\mathbf{A}$ obtained M3 and $M = 50$.

Chapter 6

Conclusion

Graph-based learning techniques have become fundamental tools to facilitate a comprehensive understanding of complex data systems and prediction of network behavior. There is a vast literature on graph-based techniques to extract relevant information from data. However, in most of the literature, the graph is either assumed to be known a priori or it is inferred from data. In both cases, there might be a mismatch between the graph topology we assume to be true and the real topology. The major goal of the thesis has been to analyze the impact of imperfect knowledge about the graph or on the signal defined over the graph, on various algorithms aimed to extract information from data.

Topology perturbations are present in several situations due to model mismatch or link failure, while signal perturbations may occur because of outliers, adversaries or observation inaccuracy. This prompted us to propose tools that consider possible perturbations making use of small perturbation theory and total least-squares approaches, considering also, in some cases, prior information of the perturbation events. We also observed the effect of the perturbation of a graph topology to assess the role of the edges in the network and to rank them starting from the most critical ones.

First of all, in Ch. 1, we have illustrated the reasons why graphs represent a powerful tool to analyze structured complex networks. Chapter 2 has recalled the theoretical results associated with graphs, GSP, and SEMs, which are relevant to the subjects investigated in this thesis.

In Chapter 3, a small perturbation analysis of Laplacian matrices has been instrumental to extend classical GSP tools to cope with cases in which the topology of the graph is not perfectly known. Our main assumption was that the considered perturbation is small, i.e., the percentage of perturbed edges is small. This assumption allowed us to use the small perturbation theory of matrices to obtain approximated closed form expressions, instrumental to propose signal processing

algorithms that are resilient to imperfect knowledge of the graph topology.

Small perturbation theory has been used to evaluate the effect of edge removal in a network and to identify the edges that have the most critical effects when they fail. Such an analysis allowed us to propose, in Chapter 4, a new centrality measure, named *topology perturbation* centrality.

In Chapter 5, we analyzed two graph-based learning tasks, such as topology ID and signal recovery in presence of perturbations. Relying on structural equation models, we have introduced regularized total least-squares methods that can deal with possible errors in the given graph signal or/and in the given graph topology.

Finally, we have shown the benefits of perturbation-aware network-based learning methods by several numerical results.

In summary, the goal of this thesis has been to cover a gap present in the available literature on graph signal processing and graph-based learning, which typically assumes a perfect knowledge of the graph. To be able to find closed form expressions, very useful to understand the impact of each edge failure on the connectivity of the networks, we have used a small perturbation analysis and we have considered the case where the Laplacian eigenvalues have multiplicity one. A direct extension of this work should incorporate the case of eigenvalues of multiplicity greater than one. An important area worth of further investigation is that of unsupervised learning and semi-supervised learning. In the first case, we considered only the case of two clusters, but it would be important to generalize the approach to multiple clusters. Also in the semi-supervised learning framework, we studied the impact of erroneous assumption about the topology on label propagation, considering a two-class example. This choice was made also to derive close form expressions that have been useful to understand the impact of errors, but it would be advisable to extend the approach to multiple classes. Finally, the employment of total least-squares methods allowed as to illustrate the benefit of considering, signal and/or topological errors. However, when considering total least-squares method, other signal models can be assumed to incorporate possible dynamic or non-linear models. A further effort could be made accommodating the analysis of graph and signal perturbations over large-scale graphs.

Bibliography

[1] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[2] M. Newman, *Networks: An Introduction.* New York, NY, USA: Oxford University Press, Inc., 2010.

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

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

[5] S. Tabassum, F. S. F. Pereira, S. Fernandes, and J. Gama, "Social network analysis: An overview," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 5, p. e1256, 2018.

[6] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[7] K. Cook, R. Emerson, M. Gillmore, and T. Yamagushi, "The distribution of power in exchange networks: Theory and experimental results," 1983.

[8] C. J. Garroway, J. Bowman, D. Carr, and P. J. Wilson, "Applications of graph theory to landscape genetics," *Evolutionary Applications*, vol. 1, no. 4, pp. 620–630, 2008.

[9] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Phys. Rev. E*, vol. 65, no. 5, p. 056109, May 2002.

[10] M. E. Shaw, "Group structure and the behavior of individuals in small groups," *The Journal of Psychology*, vol. 38, no. 1, pp. 139–149, 1954.

[11] U. Nieminen, "On the centrality in a directed graph," *Social Science Research*, vol. 2, no. 4, pp. 371 – 378, 1973.

[12] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, December 1966.

[13] M. A. Beauchamp, "An improved index of centrality," *Behavioral Science*, vol. 10, no. 2, pp. 161–163, 1965.

[14] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of mathematical sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[15] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977.

[16] F. R. K. Chung, *Spectral Graph Theory.* American Mathematical Society, 1997.

[17] S. Van Huffel and J. Vandewalle, *The Total Least Squares Problem: Computational Aspects and Analysis*, ser. Frontier in Applied Mathematics. Philadelphia, PA: SIAM, 1991, vol. 9.

[18] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[19] B. Mohar, "Laplace eigenvalues of graphs—a survey," *Discrete Mathematics*, vol. 109, no. 1, pp. 171 – 183, 1992.

[20] A. Ghosh and S. Boyd, "Growing well-connected graphs," in *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec 2006, pp. 6605–6611.

[21] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter-banks for graph structured data," *CoRR*, vol. abs/1106.3693, 2011.

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

[23] I. Pesenson, "Sampling in paley-wiener spaces on combinatorial graphs," *Transactions of The American Mathematical Society - TRANS AMER MATH SOC*, vol. 361, 11 2011.

[24] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Transactions on Signal Processing*, vol. 64, no. 18, pp. 4845–4860, Sep. 2016.

[25] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. of the IEEE*, vol. 106, no. 5, pp. 787–807, May 2018.

[26] G. Mateos, S. Segarra, and A. Marques, *Inference of Graph Topology*, Jan. 2018, pp. 349–374.

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

[28] B. M. Lake and J. B. Tenenbaum, "Discovering structure by learning sparse graph," in *Proc. of Cognitive Science Conf.*, 2010.

[29] V. Kalofolias, "How to learn a graph from smooth signals," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, ser. Proc. Mach. Learn. Research, A. Gretton and C. C. Robert, Eds., vol. 51, Cadiz, Spain, May. 2016, pp. 920–929.

[30] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning laplacian matrix in smooth graph signal representations," *Trans. Sig. Proc.*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016.

[31] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero, "Learning sparse graphs under smoothness prior," *2017 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6508–6512, 2016.

[32] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal and Information Processing over Networks*, vol. 3, no. 3, pp. 467–483, 2017.

[33] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. Rabbat, "Characterization and inference of graph diffusion processes from observations of stationary signals," *IEEE Trans. on Sig. and Inf. Proc. over Networks*, vol. PP, pp. 1–1, 08 2017.

[34] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," *IEEE Trans. on Sig.l and Inf. Proc. over Networks*, vol. 3, pp. 484–499, 2017.

[35] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "Graph topology inference based on sparsifying transform learning," *IEEE Trans. on Sig.Proc.*, vol. 67, pp. 1712–1727, Apr. 2019.

[36] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning Theory and Kernel Machines*, B. Schölkopf and M. K. Warmuth, Eds. Springer Berlin Heidelberg, 2003, pp. 144–158.

[37] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies." *IEEE Trans. Signal Processing*, vol. 64, no. 14, pp. 3775–3789, 2016.

[38] S. Segarra, A. Marques, G. Leus, and A. Ribeiro, "Reconstruction of graph signals through percolation from seeding nodes," *IEEE Transactions on Signal Processing*, vol. 64, 07 2015.

[39] D. Thanou, D. I. Shuman, and P. Frossard, "Learning parametric dictionaries for signals on graphs," *IEEE Transactions on Signal Processing*, vol. 62, no. 15, pp. 3849–3862, Aug 2014.

[40] R. I. Kondor and J. Lafferty, "Diffusion kernels on graphs and other discrete structures," in *In Proceedings of the ICML*, 2002, pp. 315–322.

[41] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Transactions on Signal Processing*, vol. 65, no. 3, pp. 764–778, Feb 2017.

[42] V. N. Ioannidis, Y. Shen, and G. B. Giannakis, "Semi-blind inference of topologies and dynamical processes over dynamic graphs," *IEEE Trans. Sig. Process.*, vol. 67, no. 9, pp. 2263–2274, May 2019.

[43] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Filtering random graph processes over random time-varying graphs," *IEEE Trans. Sig. Process.*, vol. 65, no. 16, pp. 4406–4421, 2017.

[44] J. Miettinen, S. A. Vorobyov, and E. Ollila, "Graph error effect in graph signal processing," *Proc. IEEE Int. Conf. Acous., Speech, and Sig. Process.*, pp. 4164–4168, Calgary, Canada, Apr. 2018.

[45] S. Segarra and A. Ribeiro, "Stability and continuity of centrality measures in weighted graphs," *IEEE Trans. Sig. Process.*, vol. 64, no. 3, pp. 543–555, 2016.

[46] E. Ceci and S. Barbarossa, "Graph signal processing in the presence of topology uncertainties," *submitted to IEEE Transactions on Signal Processing*, 2019.

[47] ——, "Small perturbation analysis of network topologies," in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Process.*, April 2018, Calgary, Canada, pp. 4194–4198.

[48] S. Barbarossa, S. Sardellitti, E. Ceci, and M. Merluzzi, "The edge cloud: A holistic view of communication, computation and caching," *arXiv preprint arXiv:1802.00700*, 2018.

[49] E. Ceci, Y. Shen, G. Giannakis, and S. Barbarossa, "Graph-based learning under perturbations via total least squares," *submitted to IEEE Transactions on Signal Processing*, 2019.

[50] E. Ceci and S. Barbarossa, "Graph signal processing in the presence of topology uncertainties," *IEEE Transactions on Signal Processing*, pp. 1–1, 2020.

[51] E. Ceci, Y. Shen, B. G. Giannakis, and S. Barbarossa, "Graph-based learning under perturbations via tls," *IEEE Transactions on Signal Processing*, pp. 1–1, 2020.

[52] C. Godsil and G. F. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. Springer, 2001, no. Book 207.

[53] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[54] W. Huang, A. Marques, and A. Ribeiro, "Rating prediction via graph signal processing," *IEEE Trans. Signal Processing*, vol. 66, no. 19, pp. 5066–5081, Oct. 2018.

[55] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich, *Introduction to Semi-Supervised Learning.* Morgan and Claypool Publishers, 2009.

[56] D. Wagner and F. Wagner, "Between min cut and graph bisection," in *International Symposium on Mathematical Foundations of Computer Science.* Springer, 1993, pp. 744–750.

[57] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *PNAS*, vol. 99, no. 12, pp. 7821–7826, 2002.

[58] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models.* Springer Series Stat., 2009.

[59] M. Yuan and Y. Lin, "Model selection and estimation in the gaussian graphical model," pp. 19–35, 2007.

[60] C. jui Hsieh, I. S. Dhillon, P. K. Ravikumar, and M. A. Sustik, "Sparse inverse covariance matrix estimation using quadratic approximation," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. S. Zemel, P. L.

Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2011, pp. 2330–2338.

[61] O. Banerjee, L. El Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data," *J. Mach. Learn. Res.*, vol. 9, pp. 485–516, Jun. 2008.

[62] N. Meinshausen and P. Bühlmann, "High dimensional graphs and variable selection with the lasso," *Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, 2006.

[63] R. Dahlhaus, "Graphical interaction models for multivariate time series1," *Metrika*, vol. 51, no. 2, pp. 157–172, Aug 2000.

[64] D. Brillinger, *Time Series: Data Analysis and Theory*, ser. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 2001.

[65] F. Chung, "The heat kernel as the pagerank of a graph," in *Proc. of the National Academy of Sciences*, vol. 104, no. 50, 2007, pp. 19 735–19 740.

[66] H. Ma, H. Yang, M. R. Lyu, and I. King, "Mining social networks using heat diffusion processes for marketing candidates selection," in *Proc. of the 17th ACM Conference on Information and Knowledge Management*, ser. CIKM '08. ACM, 2008, pp. 233–242.

[67] H. Petric Maretic, D. Thanou, and P. Frossard, "Graph learning under sparsity priors," in *Proc. Int. Conf. on Acoustic, Speech and Sig. Proc.*, 03 2017, pp. 6523–6527.

[68] D. Kaplan, *Structural Equation Modeling: Foundations and Extensions.* Sage, 2009, vol. 2nd ed.

[69] J. A. Bazerque, B. Baingana, and G. B. Giannakis, "Identifiability of sparse structural equation models for directed and cyclic networks," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Austin, TX, Dec. 2013.

[70] B. Baingana, G. Mateos, and G. B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," *IEEE J. Sel Topics Sig. Process.*, vol. 8, no. 4, pp. 563–575, Aug. 2014.

[71] B. Baingana and G. B. Giannakis, "Tracking switched dynamic network topologies from information cascades," *IEEE Trans. Signal Processing*, vol. 65, no. 4, pp. 985–997, 2017.

[72] P. A. Traganitis, Y. Shen, and G. B. Giannakis, "Network topology inference via elastic net structural equation models," in *Proc. of European Signal Processing Conference*, Aug. 2017, pp. 146–150.

[73] J. Ghosh, H. Q. Ngo, S. Yoon, and C. Qiao, "On a routing problem within probabilistic graphs and its application to intermittently connected networks," in *Proc. IEEE Int.Conf. on Computer Communications*, May 2007, pp. 1721–1729.

[74] A. Khan, Y. Ye, and L. Chen, *On Uncertain Graphs*, ser. Synthesis Lectures on Data Management. Morgan & Claypool, 2018, vol. 10.

[75] V. Kassiano, A. Gounaris, A. Papadopoulos, and K. Tsichlas, "Mining uncertain graphs: An overview," 04 2017, pp. 87–116.

[76] G. Rubino, "Network reliability evaluation," in *Network Performance Modeling and Simulation*, J. Walrand, K. Bagchi, and G. W. Zobrist, Eds. Gordon and Breach Science Publishers, Inc., 1999, pp. 275–302.

[77] F. Beichelt and P. Tittmann, *Reliability and Maintenance: Networks and Systems.* Taylor & Francis, 2012.

[78] G. Kollios, M. Potamias, and E. Terzi, "Clustering large probabilistic graphs," *IEEE Trans. on Knowledge and Data Engineering*, vol. 25, no. 2, pp. 325–336, 2013.

[79] L. Liu, R. Jin, C. Aggarwal, and Y. Shen, "Reliable clustering on uncertain graphs," in *Proc. IEEE Int. Conf. on Data Mining*, 2012, pp. 459–468.

[80] M. Ceccarello, C. Fantozzi, A. Pietracaprina, G. Pucci, and F. Vandin, "Clustering uncertain graphs," *Proc. VLDB Endowment*, vol. 11, no. 4, pp. 472–484, Dec. 2017.

[81] Z. Zou, J. Li, H. Gao, and S. Zhang, "Finding top-k maximal cliques in an uncertain graph," pp. 649–652, 2010.

[82] R. Jin, L. Liu, and C. C. Aggarwal, "Discovering highly reliable subgraphs in uncertain graphs," in *Proc. of Int. Conf. on Knowledge discovery and data mining.* ACM, August 2011, San Diego, California, pp. 992–1000.

[83] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC bioinformatics*, vol. 4, no. 1, p. 2, 2003.

[84] P. Sevon, L. Eronen, P. Hintsanen, K. Kulovesi, and H. Toivonen, "Link discovery in graphs derived from biological databases," in *International Workshop on Data Integration in the Life Sciences.* Springer, 2006, pp. 35–49.

[85] J. Ghosh, H. Q. Ngo, S. Yoon, and C. Qiao, "On a routing problem within probabilistic graphs and its application to intermittently connected networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE.* IEEE, 2007, pp. 1721–1729.

[86] D. Eppstein, M. Löffler, and D. Strash, "Listing all maximal cliques in sparse graphs in near-optimal time," in *International Symposium on Algorithms and Computation.* Springer, 2010, pp. 403–414.

[87] B. Zhao, J. Wang, M. Li, F.-X. Wu, and Y. Pan, "Detecting protein complexes based on uncertain graph model," *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, vol. 11, no. 3, pp. 486–497, May 2014.

[88] X. Huang, H. Cheng, and J. X. Yu, "Attributed community analysis: Global and ego-centric views." *IEEE Data Eng. Bull.*, vol. 39, no. 3, pp. 29–40, 2016.

[89] P. Parchas, N. Papailiou, D. Papadias, and F. Bonchi, "Uncertain graph sparsification," *IEEE Trans.on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2435–2449, March 2018.

[90] P. Balachandran, E. D. Kolaczyk, and W. D. Viles, "On the propagation of low-rate measurement error to subgraph counts in large networks," *J. Mach. Lear. Res.*, vol. 18, no. 1, pp. 2025–2057, 2017.

[91] J. Miettinen, S. A. Vorobyov, and E. Ollila, "Robust least mean squares estimation of graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech and Sig. Process.*, May 2019, pp. 5416–5420.

[92] J. Rayleigh, *The Theory of Sound*, ser. The Theory of Sound. Macmillan, 1894, no. v. 1.

[93] E. Schrödinger, *Collected papers on wave mechanics.* American Mathematical Soc., 2003, vol. 302.

[94] J. H. Wilkinson, *The Algebraic Eigenvalue Problem.* New York, NY, USA: Oxford University Press, Inc., 1988.

[95] G. Stewart, *Introduction to matrix computations*, ser. Computer science and applied mathematics. Academic Press, 1973.

[96] S. Borjigin and C. Guo, "Perturbation analysis for the normalized laplacian matrices in the multiway spectral clustering method," *Science China. Information Sciences*, vol. 57, no. 11, pp. 1–17, Nov 2014.

[97] S. Hata and H. Nakao, "Localization of laplacian eigenvectors on random networks," *Scientific Reports*, vol. 7, no. 1, pp. 1121–1121, April 2017.

[98] D. A. Spielman, "Spectral graph theory and its applications," in *Proc. Annual IEEE Symposium on Foundations of Computer Science*, Oct. 2007, pp. 29–38.

[99] N. Biggs, N. L. Biggs, and B. Norman, *Algebraic graph theory.* Cambridge university press, 1993, vol. 67.

[100] G. H. Golub, "Some modified matrix eigenvalue problems," *SIAM Review*, vol. 15, no. 2, pp. 318–334, 1973.

[101] J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, "Rank-one modification of the symmetric eigenproblem," *Numerische Mathematik*, vol. 31, no. 1, pp. 31–48, 1978.

[102] O. Teke and P. P. Vaidyanathan, "Uncertainty principles and sparse eigenvectors of graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 20, pp. 5406–5420, Oct 2017.

[103] A. Banerjee and J. Jost, "On the spectrum of the normalized graph laplacian," *Linear algebra and its applications*, vol. 428, no. 11-12, pp. 3015–3022, 2008.

[104] B. D. MacArthur, R. J. Sánchez-García, and J. W. Anderson, "Symmetry in complex networks," *Discrete Applied Mathematics*, vol. 156, no. 18, pp. 3525–3531, 2008.

[105] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. American Stat. Association*, vol. 58, no. 301, pp. 13–30, 1963.

[106] I. Pesenson, "Sampling in Paley-Wiener spaces on combinatorial graphs," *Trans. of the American Math. Society*, vol. 360, no. 10, pp. 5603–5627, Oct. 2008.

[107] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, pp. 129–150, Dec. 2011.

[108] S. Barbarossa, *Multiantenna Wireless Communication Systems*, ser. Mobile Communications Series. Artech House, 2003.

[109] P. Bodik *et al.* (2004) Intel berkeley research lab [Online].

[110] Y. Bengio, O. Delalleau, and N. Le Roux, "Label propagation and quadratic criterion," in *Semi-Supervised Learning.* MIT Press, Jan. 2006, pp. 193–216.

[111] Kai Sun, Da-Zhong Zheng, and Qiang Lu, "A simulation study of OBDD-based proper splitting strategies for power systems under consideration of transient stability," *IEEE Trans. on Power Syst.*, vol. 20, no. 1, pp. 389–399, Feb 2005.

[112] P. Di Lorenzo and E. Ceci, "Online recovery of time-varying signals defined over dynamic graphs," in *Proc. European Signal Process. Conf.*, Sept. 2018, Rome, Italy, pp. 131–135.

[113] A. Shimbel, "Structural parameters of communication networks," *The bulletin of mathematical biophysics*, vol. 15, no. 4, pp. 501–507, Dec 1953.

[114] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, no. 6833, pp. 41–42, May 2001.

[115] A. del Sol, H. Fujihashi, and P. O'Meara, "Topology of small-world networks of protein–protein complex structures," *Bioinformatics*, vol. 21, no. 8, pp. 1311–1315, 01 2005.

[116] T. Coffman, S. Greenblatt, and S. Marcus, "Graph-based technologies for intelligence analysis," *Commun. ACM*, vol. 47, no. 3, pp. 45–47, Mar. 2004.

[117] V. E. Krebs, "Mapping networks of terrorist cells," *Connections*, vol. 24, no. 3, pp. 43–52, 2001.

[118] D. Cisic, B. Kesic, and L. Jakomin, "Research of the power in the supply chain. international trade," *Economics Working Paper Archive EconWPA*, 01 2000.

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

[120] J. Cheeger, "A lower bound for the smallest eigenvalue of the laplacian," in *Proc. Princeton conf. in honor of Prof. S. Bochner*, 1969, pp. 195–199.

[121] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, , and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, no. 54, pp. 396–405, 2003.

[122] M. Newman, "Spectral methods for community detection and graph partitioning," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 88, pp. 042 822–1–042 822–10, 10 2013.

[123] J. Cheng, L. Li, M. Leng, W. Lu, Y. Yao, and X. Chen, "A divisive spectral method for network community detection," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2016, no. 3, p. 033403, Mar. 2016.

[124] V. Krebs. Books about us politics.

[125] J. Chang, E. D. Kolaczyk, and Q. Yao, "Estimation of edge density in noisy networks," *arXiv preprint arXiv:1803.02488*, Mar. 2018.

[126] E. Ceci and S. Barbarossa, "Robust graph signal processing in the presence of uncertainties on graph topology," in *Proc. IEEE-SP Workshop on Sig. Process. Advances in Wireless Comm.*, June 2018, Kalamata, Greece, pp. 1–5.

[127] H. Zhu, A. Cano, and G. B. Giannakis, "Distributed consensus-based demodulation: Algorithms and error analysis," *IEEE Trans. on Wireless Communications*, vol. 9, no. 6, pp. 2044–2054, June 2010.

[128] P. Sprent, "Models in regression and related topics," *Methuen's Monographs on Applied Probability and Statistics*, Oct. 1969.

[129] E. P. Jiang and M. W. Berry, "Solving total least-squares problems in information retrieval," *Linear Algebra and its Applications*, vol. 316, no. 1-3, pp. 137–156, Jan 2000.

[130] I. Markovsky and S. Van Huffel, "Overview of total least-squares methods," *Sig. Proc.*, vol. 87, no. 10, pp. 2283–2302, Apr. 2007.

[131] M. Aoki and P. Yue, "On a priori error estimates of some identification methods," *IEEE Trans. on Automatic Control*, vol. 15, no. 5, pp. 541–548, Oct. 1970.

[132] B. De Moor, "Structured total least-squares and L2 approximation problems," *Linear Algebra and its Applications*, vol. 188, no. 5, pp. 163–205, Aug. 1993.

[133] H. Zhu, G. Leus, and G. B. Giannakis, "Sparsity-cognizant total least-squares for perturbed compressive sampling," *IEEE Trans. Signal Processing*, vol. 59, no. 5, pp. 2002–2016, May 2011.

[134] X. Cai, J. A. Bazerque, and G. B. Giannakis, "Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations," *PLoS Comp. Biol.*, vol. 9, no. 5, p. Art. no. e1003068, May 2013.

[135] X. Ning and G. Karypis, "Slim: Sparse linear methods for top-n recommender systems," in *Proc. of Intl. Conf. on Data Mining*, Vancouver, Canada, Dec. 2011, pp. 497–506.

[136] B. Muthén, "A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators," *Psychometrika*, vol. 49, no. 1, pp. 115–132, Mar. 1984.

[137] A. S. Goldberger, "Structural equation methods in the social sciences," *Econometrica*, vol. 40, no. 6, pp. 979–1001, Nov. 1972.

[138] Y. Shen, B. Baingana, and G. B. Giannakis, "Tensor decompositions for identifying directed graph topologies and tracking dynamic networks," *IEEE Trans. Signal Processing*, vol. 65, no. 14, pp. 3675–3687, July 2017.

[139] A. Jung, G. Hannak, and N. Goertz, "Graphical lasso based model selection for time series," *IEEE Sig. Process.Letters*, vol. 22, no. 10, pp. 1781–1785, Oct 2015.

[140] A. Jung, "Learning the conditional independence structure of stationary time series: A multitask learning approach," *IEEE Trans. on Sig. Process.*, vol. 63, no. 21, pp. 5677–5690, Nov 2015.

[141] N. T. Quang and A. Jung, "Learning conditional independence structure for high-dimensional uncorrelated vector processes," in *Proc. Intl. Conf. Acoust., Speech and Signal Processing*, March 2017, pp. 5920–5924.

[142] M. Kolar and E. P. Xing, "On time varying undirected graphs," in *Proc. Intl. Conf. Art. Intell. Stat.*, July, 2011.

[143] G. Mateos, S. Segarra, and A. Marques, "Inference of graph topology," in *Cooperative and Graph Signal Processing*, P. M. Djurić and C. Richard, Eds. Academic Press, 2018, pp. 349 – 374.

[144] S. K. Narang, A. Gadde, E. Sanou, and A. Ortega, "Localized iterative methods for interpolation in graph structured data." IEEE, pp. 491–494.

[145] P. Di Lorenzo, S. Barbarossa, and P. Banelli, "Sampling and recovery of graph signals," in *Cooperative and Graph Signal Processing*, P. M. Djurić and C. Richard, Eds. Academic Press, 2018.

[146] V. N. Ioannidis, M. Ma, A. N. Nikolakopoulos, G. B. Giannakis, and D. Romero, "Kernel-based inference of functions over graphs," in *Adaptive Learning Methods for Nonlinear System Modeling.* Elsevier, 2018, pp. 173–198.

[147] V. N. Ioannidis, D. Romero, and G. B. Giannakis, "Inference of spatio-temporal functions over graphs via multikernel kriged kalman filtering," *IEEE Transactions on Signal Processing*, vol. 66, no. 12, pp. 3228–3239, 2018.

[148] A. Beck, A. Ben-tal, and M. Teboulle, "Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares," *SIAM J. Matrix Anal. Appl*, vol. 28, no. 2, pp. 425–445, 2006.

[149] W. Dinkelbach, "On nonlinear fractional programming," *Manag. Sci.*, vol. 13, no. 7, pp. 492–498, Mar. 1967.

[150] I. G. Akrotirianakis and C. A. Floudas, "Computational experience with a new class of convex underestimators: Box-constrained nlp problems," *J. Global Optimization*, vol. 29, no. 3, pp. 249–264, Jul. 2004.

[151] D. P. Bertsekas, "Nonlinear programming," *J. Oper. Res. Soc.*, vol. 48, no. 3, pp. 334–334, 1997.

[152] G. Wang, V. Kekatos, A. J. Conejo, and G. B. Giannakis, "Ergodic energy management leveraging resource variability in distribution grids," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4765–4775, Nov. 2016.

[153] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

[154] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. of the Royal Stat. Soc.: Series B*, vol. 67, no. 2, pp. 301–320, 2005.

[155] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra Appl.*, vol. 18, no. 2, pp. 95–138, 1977.

[156] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *J. Mach. Learn. Res.*, vol. 11, pp. 985–1042, Mar. 2010.

[157] K. A. Frazer *et al.*, "A second generation human haplotype map of over 3.1 million SNPs," *Nature*, vol. 449, no. 7164, p. 851, 2007.

[158] J. K. Pickrell *et al.*, "Understanding mechanisms underlying human gene expression variation with RNA sequencing," *Nature*, vol. 464, no. 7289, p. 768, 2010.

[159] "One-way ping internet2," [Online], Available: http://software.internet2.

[160] D. B. Chua, E. D. Kolaczyk, and M. Crovella, "Network kriging," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 12, pp. 2263–2272, Dec. 2006.