
# PREDICTION METHODS FOR STATISTICAL INFERENCE IN GRAPH SIGNAL PROCESSING

**Carlos García Ortiz**

A Master's Thesis
Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomuniació de Barcelona
Universitat Politècnica de Catalunya

In partial fulfillment
of the requirements for the degree

MASTER'S DEGREE IN TELECOMMUNICATIONS ENGINEERING

Advisor
**Jordi Borràs Pino**
Co-Advisor
**Prof. Gregori Vázquez Grau**

Barcelona, July 2020

# Abstract

This thesis studies the problem of inferring topology from signal graphs. For this reason, the Master's Thesis is part of the current of thought, growing in recent years, in which the structure of the network is not assumed to be known.

The problem of inferring topology is approached from two angles. The first one, studies how to find the structure of a graph from spectral templates which can be noisy. Thus, from observations of the network, the spectral template of the graph that makes up the network is inferred. In previous works, like my Degree's Thesis, the algorithms for the inference of incomplete spectral templates were studied. In this Master's thesis, we go one step further by demonstrating why the techniques studied do not always work and proposing an algorithm based on *LASSO* to infer the network topology when the spectral templates are noisy. The proposed algorithm is compared with those previously studied obtaining better results in terms of *RMSE* and reliability.

The second point of view addressed in this thesis is the inference of the network from statistical techniques. It is common to find networks whose nodes have some relation. These techniques are based on, from some observations of the network, trying to find the existing relationships between the different nodes of the graph. These techniques can be used in a more generic way than those based on spectral templates. Statistical methods are studied in more depth in this Master's Thesis. Initially, the Pearson correlation coefficient is explained. After studying it, some limitations are found. Thus, a new approach is proposed based on the conditional covariance. Then, it is assumed that the signals follow a Gaussian distribution which brings us to study the Maximum Likelihood estimator while considering the graph's sparsity. Although, the previous approach was improved, we are interested in finding even a better one. Hence, we study an approach based on linear regression. In this last algorithm, we include a term to promote sparsity when finding the solution.

To conclude, the statistical methods studied, are compared by performing some simulations. By performing these simulations, it is observed that the best technique to infer the graph's topology is the one based on linear regression.

# Resumen

Esta tesis estudia el problema de inferir la topología de la red a partir de las señales grafo. Por esta razón, la Tesis Final de Máster se inscribe en la corriente actual de pensamiento, creciente en los últimos años, en la que se supone no conocida la estructura de la red.

El problema de la inferencia de la topología se aborda desde dos ángulos. El primero, estudia cómo encontrar la estructura de un grafo a partir de plantillas espectrales que pueden ser ruidosas o no. Así, a partir de las observaciones, se infiere la plantilla espectral del grafo que compone la red. En trabajos anteriores, como mi Tesis de Final de Grado, se estudiaron los algoritmos para la inferencia de plantillas espectrales incompletas. En esta trabajo, vamos un paso más allá demostrando por qué las técnicas estudiadas no siempre funcionan. Seguimos, proponiendo un algoritmo basado en *LASSO* para inferir la topología de la red cuando las plantillas espectrales son ruidosas. El algoritmo propuesto se compara con los anteriormente estudiados obteniendo mejores resultados en términos de *RMSE* y fiabilidad.

El segundo punto de vista abordado en esta tesis es la inferencia de la red a partir de técnicas estadísticas. Es común encontrar redes cuyos nodos tienen alguna relación entre ellos. Estas técnicas se basan, a partir de algunas observaciones de la red, en tratar de encontrar las relaciones existentes entre los diferentes nodos del grafo. Estas técnicas pueden ser utilizadas de manera más genérica que las basadas en plantillas espectrales. Por ese motivo, los métodos estadísticos se estudian con más profundidad en esta Tesis de Máster. Inicialmente, se explica el coeficiente de correlación de Pearson. Después de estudiarlo, se detecta una limitación. Por ese motivo, se propone un nuevo enfoque basado en la covarianza condicional. Luego, se asume que las señales siguen una distribución Gaussiana, lo que nos lleva a estudiar el estimador de Máxima Verosimilitud, también considerando que la matriz solución es dispersa. Aunque con esta técnica se mejora el enfoque anterior, estamos interesados en encontrar una técnica aún mejor. Por lo tanto, estudiamos un enfoque basado en la regresión lineal. En este último algoritmo, incluimos un término para promover la que la solución sea una matriz dispersa.

Para concluir, los métodos estadísticos estudiados, se comparan realizando algunas simulaciones. Al realizarlas, se observa que la mejor técnica para inferir la topología del grafo es la que se basa en la regresión lineal.

# Resum

Aquesta tesi estudia el problema d'inferir la topologia d'una xarxa a partir dels senyals graf. Per aquesta raó, la Tesi Final de Màster s'inscriu en el corrent actual de pensament, creixent en els darrers anys, en la que se suposa no coneguda l'estructura de la xarxa.

El problema d'inferència de la topologia s'aborda des de dos angles. El primer, estudia com trobar l'estructura d'un graf a partir de plantilles espectrals que poden ser sorolloses o no. Així, a partir de les observacions, s'infereix la plantilla espectral del graf que compon la xarxa. En treballs anteriors, com la meva Tesi de Final de Grau, es van estudiar els algoritmes per a la inferència de plantilles espectrals incompletes. En aquest treball, anem un pas més enllà demostrant per què les tècniques estudiades no sempre funcionen. Seguim, proposant un algoritme basat en *LASSO* per inferir la topologia de la xarxa quan les plantilles espectrals són sorolloses. L'algoritme proposat es compara amb els anteriorment estudiats obtenint millors resultats en termes de *RMSE* i fiabilitat.

El segon punt de vista abordat en aquesta tesi és la inferència de la xarxa a partir de tècniques estadístiques. És comú trobar xarxes on els nodes tenen alguna relació entre ells. Aquestes tècniques es basen, a partir d'algunes observacions de la xarxa, a tractar de trobar les relacions existents entre els diferents nodes del graf. Aquestes tècniques poden ser utilitzades de manera més genèrica que les basades en plantilles espectrals. Per aquest motiu, els mètodes estadístics s'estudien amb més profunditat en aquesta Tesi de Màster. Inicialment, s'explica el coeficient de correlació de Pearson. Després d'estudiar-lo, es detecta una limitació. Per aquest motiu, es proposa un nou enfocament basat en la covariància condicional. Després, s'assumeix que els senyals segueixen una distribució Gaussiana, el que ens porta a estudiar l'estimador de Màxima Versemblança, també considerant que la matriu solució és dispersa. Encara que amb aquesta tècnica es millora l'enfocament anterior, estem interessats a trobar una tècnica encara millor. Per tant, estudiem un enfocament basat en la regressió lineal. En aquest últim algoritme, incloem un terme per promoure que la solució sigui una matriu dispersa.

Per concloure, els mètodes estadístics estudiats, es comparen realitzant algunes simulacions. En realitzar-les, s'observa que la millor tècnica per inferir la topologia del graf és la que es basa en la regressió lineal.

# Acknowledgments

This has been the most strange, difficult, and stressful semester of studies probably for most of us. It started as another normal semester until the pandemic hit.

Despite of this major setback, I am delivering my Master's Thesis. With it, I am finishing my Engineering studies, at least for some time. I think it is time to apply all the knowledge obtained during these past 6 years of studies, and try to improve a bit the world where we live.

Without further ado, I would like to give my most sincere thanks to my tutor Jordi. You have been supporting me, even in weekends, during all this Master's Thesis. I wouldn't want to forget either, all the support and how much I have learned with you during the introduction to research courses. They have been one of the most interesting subjects I have coursed during my Master Degree.

I want to thank a lot to Gregori who has acted as a co-tutor in my Thesis. You have helped me a lot, not only with theoretical concepts but also at the organizational level in a semester as atypical as this one.

I would like to thank a lot also to my friends who have done this Master a bit more interesting and funny. These unforgettable games we played were one of the most funnier things I have done during this last year.

Last but not least, I want to thank to my family and partner for all the support they gave me. This Master's Thesis is dedicated to them.

# List of Acronyms

| | |
|---|---|
| *AWGN* | Additive White Gaussian Noise |
| *BP* | Basis Pursuit |
| *CCA* | Canonical Correlation Analysis |
| *EVD* | Eigendecomposition |
| *FDR* | False Discovery Rate |
| *GFT* | Graph Fourier Transform |
| *GSP* | Graph Signal Processing |
| *LASSO* | Least Absolute Shrinkage and Selection Operator |
| *MAP* | Maximum a posteriori |
| *RMSE* | Root Mean Square Error |
| *RIP* | Restricted Isometry Property |
| *TV* | Total Variation |

# Notation

| | |
|---|---|
| $\boldsymbol{x}$ | A column vector. |
| $\boldsymbol{X}$ | A matrix. |
| $\boldsymbol{I}$ | Identity matrix. |
| $\mathbf{0}$ | All-zeros vector or matrix. |
| $\mathbf{1}$ | All-ones vector or matrix. |
| $\text{diag}(\boldsymbol{X})$ | A column vector containing the diagonal entries of $\boldsymbol{X}$. |
| $\text{trace}(\boldsymbol{X})$ | The trace of matrix $\boldsymbol{X}$. |
| $\text{vec}(\boldsymbol{X})$ | A column vector representing the vectorized form of $\boldsymbol{X}$. |
| $\boldsymbol{X}^{\dagger}$ | The Moore-Penrose pseudoinverse of $\boldsymbol{X}$. |
| $(\cdot)^{*}$ | The conjugate operator. |
| $(\cdot)^{T}$ | The transpose operator. |
| $(\cdot)^{T}$ | The hermitician operator, transpose and conjugate. |
| $\odot$ | Khatri-Rao or column-wise Kronecker product. |
| $\otimes$ | The Kronecker product. |
| $\circ$ | The Hadamard product. |
| $\lvert x \rvert$ | The absolute value of x. |
| $\lVert \boldsymbol{x} \rVert$ | The Euclidian norm of $\boldsymbol{x}$. |
| $\lVert \boldsymbol{X} \rVert_0$ | The $\ell_0$-*norm* of $\boldsymbol{X}$. |
| $\lVert \boldsymbol{X} \rVert_1$ | The $\ell_1$-*norm* of $\boldsymbol{X}$. |
| $\Re$ | The real number's field. |
| $\Im$ | The imaginary number's field. |
| $\mathbb{E}$ | The mathematical expectation operator. |
| $\mathbb{1}\{\cdot\}$ | The classic indicator function |
| $\text{var}(x_i, x_j)$ | Variance between $x_i$ and $x_j$ |
| $\text{cov}(x_i, x_j)$ | Covariance between $x_i$ and $x_j$ |
| $\dim(x)$ | Number of dimensions of vector $\boldsymbol{x}$ |
| $\boldsymbol{e}_i$ | Vector with all entries equal 0 but entry $i$ equal 1 |

# List of Figures

# Contents

# Chapter 1

# Introduction

## 1.1 Scope

The world of graph theory is very extensive as well as its applications. The Graph Signal Processing (*GSP*) implementation in different applications is increasing over the year. There are multiple applications like the study of people movement. Reference [3], studies the taxi movement in New York City in the period between 2010 and 2013. They use *GSP* to discover underlying behaviors in people movements. Another interesting application is rating prediction. In [4], authors present a new approach for recommendation systems by using graph's filters. This use brings us into another interesting application, filter design based on graphs [5, 6]. Applications in the field of medicine could not be missing. The *GSP* is used in the study of cancer [7]. In this field, nodes and edges represent molecular elements and the correlation existing between them. With the arrival of the fifth generation technology *5G* it is increasing the number of sensors. This amount will increase even more with the use of sensors in what are called Smart Cities [8]. The *GSP* is also being applied in the study of data taken from sensors [9]. In this study, data about pollution is taken from different sensors placed around Poland. Much more interesting applications can be found such as: applications to management of energy systems, smart grids, environmental monitoring, or analyzing epidemiological data. An example about this last application can be found this year. The COVID-19 pandemic is being studied by using graph models [10]. In this paper, it is studied a new COVID-19 case prediction that learns from a large spatio-temporal graph.

As introduced in the abstract, when studying this theory or applying it in different fields and working with *GSP*, in general, it is considered that the network that generates the graph signal is known. From this assumption, the analysis of how the network topology influences the graph signal is constructed. However, a trend is growing more and more in which the network is not assumed to be known. Due to the great growth in applications, it is increasingly necessary to establish robust topology inference techniques that can work in optimal and non-optimal conditions in terms of the amount of information available. For this reason, multiple studies have appeared on topology inference on graph networks [11, 12].

Network topology inference can be divided in three different families. As can be seen in figure 1.1 , the three big families are: *Statistical Methods*, *Diffusion Processes* and *Smooth Signals*. The first family, represented as the blue color branch in figure 1.1, includes these

networks that present similarities between signal elements. This is a huge group where it can be include almost all possible networks. In order represent data using a the graph representation, it is necessary to find some relations between pairs of nodes. There are different approaches to decide the weight, or in other words the importance, that these connections must have. The goal when applying techniques over this family of graphs, is to estimate the best undirected graph based on the available data. The only requisite necessary in order to use these techniques, is to have few snapshots in order to compute the relations between nodes.

The second big family of graphs are those produced by diffusion processes. This second family, is represented as the green branch in figure 1.1. This group includes all those graph signals that can be modeled as the output of a graph filter when a driving noise is applied to its input. In order to use the techniques related with this family, it is necessary to do the assumption, or to be certain, that the graph signal can be modeled as the output of a graph filter. Thus, this family includes less graphs than the previous one.

The third big family of graphs are those which present a regularity between the signals in each node. This third family, is represented in yellow branch in figure 1.1. These graph signals have the particularity that, the observations are smooth. This family is a particular case of graphs which is widely found in different scenarios.

In this project, we will review the diffusion processes branch which was studied in the Degree's Thesis represented in green in figure 1.1. We will review the most important concepts, as well as all the notation introduced in the Degree's Thesis, and that will be used in this Master's Thesis. Then, the scope is divided in two parts. The first one is, by using these formulation and knowledge, the study of the topology inference when working with *noisy spectral templates*. We will search for a robust algorithm to infer the graph's topology when facing spectral templates which include a noise component. This will led us to the study of the implementation of *LASSO* to the considered approaches to improve the reliability of the techniques. The second part of the scope is to study the most generic group of techniques which are the *Statistical Methods* represented in blue in figure 1.1. This family is divided in: *Correlation Networks*, *Gaussian graphical models* and *Sparse Linear Regression*. We will study the behavior of these three approaches while comparing them. Then, two scenarios will be simulated to better see the performance of these techniques. The first one will be performed where enough data will be available what will end in a well conditioned problem. The second scenario will be more interesting, the non-well conditioned problem will be studied with all three methods. This will allow us to see the strengths and weaknesses of the techniques.

In chapter 2, we will see that the algorithm that two of the algorithms studied in the Degree's Thesis [1] were not reliable because of the *Restricted Isometry Property (RIP)* can not be guaranteed. Thus, the solution obtained by the convex relaxation using the $\ell_1$-*norm* will not be equivalent to the solution obtained using the $\ell_0$-*norm*. Then, we will see that the *Least Absolute Shrinkage and Selection Operator (LASSO)* approach will provide us better results when applied to noisy spectral templates. By doing some simulation, we will see that this algorithm performs better than the algorithms studied during the Degree's Thesis. Furthermore, we will see that when the number of nodes increases the *RMSE* when applying *LASSO* will tend to 0.

In chapter 3, we will see that statistical methods can obtain really impressive results when estimating the graph's topology. Moreover, two of the techniques, which include a

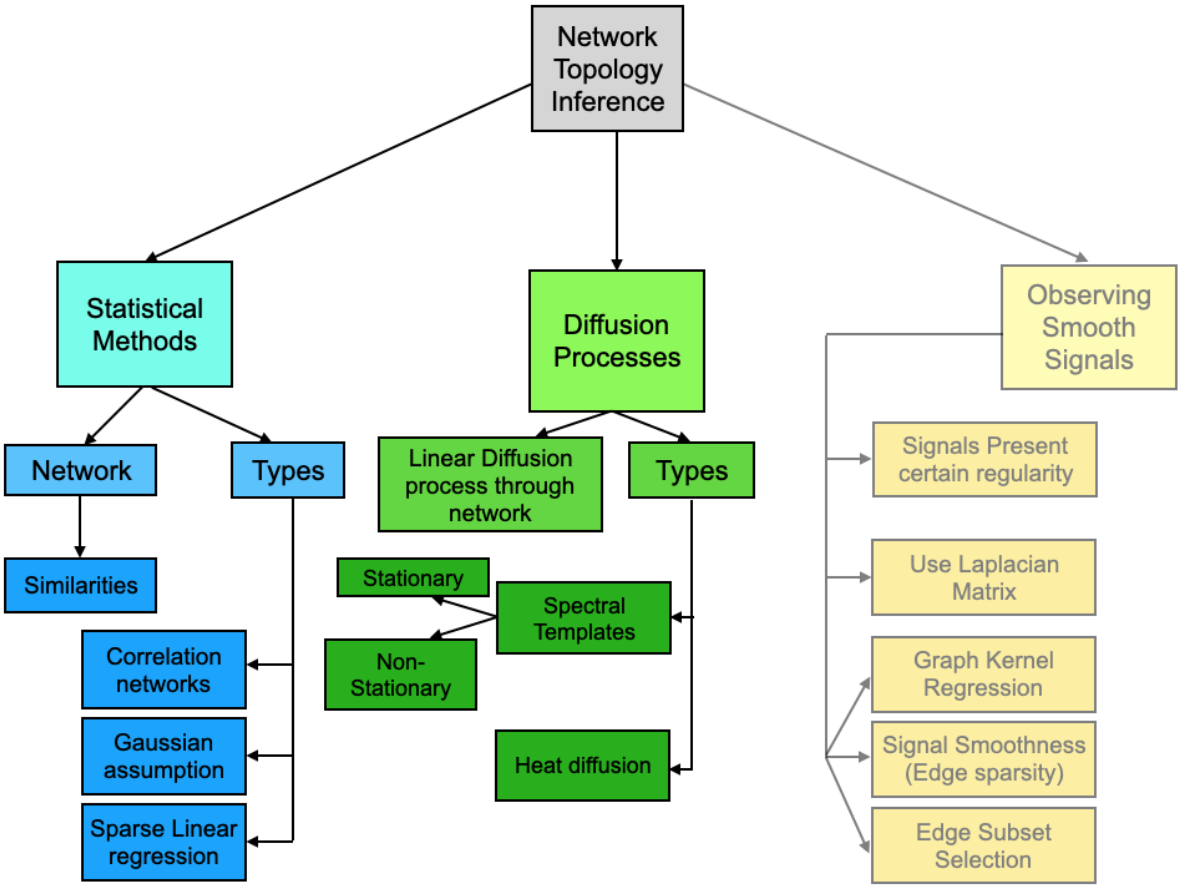


Figure 1.1: General schema of the theoretical part explained in this thesis

sparsity promotion term, will be able to estimate quite well the relations between nodes although working with low information. By comparing the three approaches explained in this chapter, we will see that the *Linear Regression* will be the most optimal one in terms of reliability, robustness, and computational complexity.

In chapter 4, we will perform some simulations to better compare the techniques studied in chapter 3. With these simulations we will perfectly realize about the great performance of the Linear Regression. We will see that, if this technique does not consider the sparsity promotion term it will obtain the same result as the *Conditional Covariance*. We will see that although having less snapshots than nodes in the graph, the Linear Regression will be able to correctly estimate the most important relations between pairs of nodes.

## 1.2 Organization

This thesis is focused in network topology inference with graph signals. In chapter 2 2.1, it is reviewed the most important concepts and formulation introduced during the Degree's Thesis [1]. With this knowledge, it is presented the reason why the algorithms called *Basis Pursuit (BP)* and *Orthogonal Matching Pursuit (OMP)* do not always work. The *RIP* is studied and it is demonstrated that it can not be guaranteed. Thus, the algorithms are not always working as expected. Then, we are focused in studying the topology inference when working with noisy spectral templates. Hence, the *LASSO* algorithm is considered, and it is merged with the existing techniques studied in the Degree's Thesis. To conclude this chapter, some conclusions are extracted were the main concepts studied in this chapter are reviewed. This chapter makes reference to the branch represented in green in figure 1.1.

In chapter 3, statistical methods are studied. This type of networks are analyzed in this chapter by using three different techniques: *Conditional Correlation*, *Gaussian graphical models* and *Sparse Linear Regression*. The *Pearson correlation* is presented as a first approach to assign values to the relations between pairs of nodes. Then, it is improved by considering the conditional correlation. By using this relation, we are able to avoid possible influences of a third node when computing the relation between a pair. Some simple and controlled simulations are performed to better understand some concepts. These tests also allow us to validate the correctly implementation of the method. Then Gaussian distribution assumption is done, taking us to Gaussian graphical models. In this section, the *maximum likelihood estimator (ML)* is derived. In order to take into account the sparsity of the solution, it is incorporated a sparsity promotion term to the *ML* estimator. Then simulations are performed where the graph topology is equal to the one used to validate the conditional correlation implementation. At the end of this chapter, the *Linear Regression* predictor is studied. Then, we include the sparsity promotion term to the linear regression approach. The same simulations as in both previous cases are executed, to validate the correctly implementation of the method. This chapter makes reference to the branch represented in blue in figure 1.1.

In chapter 4, some simulations are performed to compare the techniques studied in this thesis. In these simulations the number of nodes is increased. It is possible to better see the performance, limitations, and reliability of the techniques studied when the number of nodes increases. Also, it is interesting to see the performance of the techniques when working with a low number of snapshots.

Finally, in chapter 5, we present the conclusions to this thesis. The most important ideas and concepts are reviewed. It is also presented the future work where there are explained some of the topics that remain opened after this project.

# Chapter 2

# Topology Inference Facing with Diffusion Processes

## 2.1 Introduction

As presented in 1, there are different approaches to infer the topology of a network. One of them is from spectral templates. It allows us to introduce the notation and some theoretical concepts that are used during this thesis. Some theoretical concepts will be explained and the results will be commented. In this chapter,the problem is approached from another point of view. It is assumed that diffusion processes produce the graph signals [1]. This is the signal graph that can be seen as the output of a filter when the input is an activation noise, see figure 2.1. In section 2.2.1 a more detailed explanation can be found.

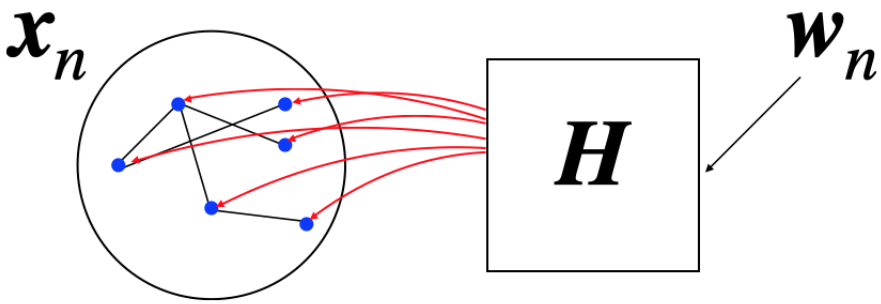


Figure 2.1: Schema of the graph-signal activation by the Filter $\boldsymbol{H}$ and the driving noise, this figure is based on the figure in [1]

As it is reviewed, the information obtained from the observations can be complete, imperfect or incomplete. In this chapter, contrary to chapter 3, the connection between two nodes is established in templates based on connectivity instead of templates based on statistical dependencies. Some assumptions are needed in order to work with spectral templates:

1. Graphs are stationary and undirected.
2. Graphs have no nodes with self-loops.
3. The main focus is data locality and flow simplicity. Therefore, our goal will be to find the sparsest solution to the problem. In different fields it is interesting to find the sparsest solution possible [13].

This type of graphs can be found in different fields such as social networking platforms [14] or vehicular mobility patterns [12, 15].

## 2.2 Previous Work

In this section, as explained in 2.1, the formulation and notation is reviewed, which is deeply studied in previous works, the Final Degree Thesis [1]. This notation and formulation is fundamental to understand the new method studied in this thesis based of working with noisy spectral templates in section 2.3. Furthermore, the same notation is used in chapter 3 to study the statistical methods which are based on a more general signal model. Firstly, it is reviewed the signal model when working with Diffusion Processes. Then the *Basis Pursuit (BP)* is explained and some results are reviewed. From these results, the need of improving the inference technique appears. This is what sections 2.2.4, 2.2.5 and, 2.2.6 review. To summarize, this section reviews how to infer the graph topology when working with imperfect spectral templates. In this thesis, we are interested in the study of how to infer the topology when working with *noisy spectral templates*. This new approach is explained in section 2.3. To finalize this chapter, there is a conclusions section that introduces the need of studying the next branch of figure 1.1 which are *Statistical Methods*.

### 2.2.1 Signal Model

Let us start by defining a graph $\mathcal{G}$ with a set of $N$ nodes and a set of links $\mathcal{L}$. The adjacency matrix is defined as $\boldsymbol{A}$ which is sparse and its dimensions are $N \times N$. It has entries different from zero $\boldsymbol{A}_{ij}$ if $(i,j) \in \mathcal{L}$. These entries represent the importance of the connection between nodes $i$ and $j$. Another important concept is the graph-shift operator $\boldsymbol{S}$. The pattern of the graph-shift operator $\boldsymbol{S}$ and the adjacency matrix are equivalent in terms of information regarding the graph's structure. Now, the next step is to define the signal model that is used in this chapter. Let $\boldsymbol{x} = [x_1 \; x_2 \; \cdots \; x_N]^T \in \Re^N$ be a graph signal in which each element $i$ denotes the signal at node $i$. The graph filter $\boldsymbol{H}$ and the driving noise $\boldsymbol{w}$ activate the graph's connections. At a given instant $n$, the graph signal (see figure 2.1) is

$$\boldsymbol{x}(n) = [x_1(n) \; x_2(n) \; \cdots \; x_N(n)]^T \tag{2.1}$$

while the driving noise is

$$\boldsymbol{w}(n) = [w_1(n) \; w_2(n) \; \cdots \; w_N(n)]^T. \tag{2.2}$$

Without loss of generality, and in order to simplify the expressions, let us consider that the noise is *AWGN*. Thus,

$$\mathbb{E}\left[\boldsymbol{w}\boldsymbol{w}^T\right] = \boldsymbol{I}. \tag{2.3}$$

The graph signal activated by filter $\boldsymbol{H}$ is defined as

$$\boldsymbol{H} = \sum_{l=0}^{L-1} h_l \boldsymbol{S}^l, \tag{2.4}$$

where $\boldsymbol{S}$ is the shift matrix and $h_l$ the filter coefficients. Let us continue by analyzing the shift matrix with further detail. As we are working with undirected graphs, $\boldsymbol{S}$ must be symmetric and real.

### 2.2.2 First Approach into Graph's Topology Inference

When facing a topology inference problem based on this type of graph signals, generated by a filter when its input is a driving noise, two different scenarios can be found when obtaining the nodes information from the graph. In the first, not common, scenario the technique has enough information, snapshots, to correctly estimate the covariance matrix of the obtained data, $\boldsymbol{C}_x$. In this first part, we are focused in this scenario. It is the simplest one of the both. The algorithm *Basis Pursuit* (*BP*), [16], is used to solve the problem. In a second, usually found, scenario the available information is limited. Thus, the covariance matrix may not be correctly estimated due to lack of information. This case is deeply explained in section 2.2.4 and 2.2.5.

#### Problem Formulation

By performing the eigendecomposition of $\boldsymbol{S}$, the shift matrix can be written in terms of its eigenvectors, $\boldsymbol{V}$, and eigenvalues, $\boldsymbol{\Lambda}$, as

$$\boldsymbol{S} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T. \tag{2.5}$$

Let us study now statistical values of our problem by analyzing the covariance matrix. By taking into account equations (2.3), (2.4), and (2.5), the covariance matrix can be expressed as

$$\boldsymbol{C_x} = \mathbb{E}\left[\boldsymbol{H}\boldsymbol{H}^T\right] = \boldsymbol{V}\sum_{l=0}^{L-1}|h_l|^2\boldsymbol{\Lambda}^{2l}\boldsymbol{V}^T. \tag{2.6}$$

It is important to note that the eigenvectors of the covariance and the shift matrix are equal. The eigenvalues are multiplied by unknown factors, $h_l$. This is a really interesting result as it shows that computing the eigenvectors of the covariance matrix we can learn the eigenvectors of the shift matrix.

We are interested in finding the sparsest solution [13] as described in chapter 3. This is, to minimize the $\ell_0$-*norm* taking into account that the eigenvectors of the solution shift matrix must be equal to the covariance matrix eigenvectors. The diagonal of the solution matrix must be equal to zero, because nodes do not have self-loops. The solution matrix must be symmetric because the graph is undirected. Last but not least, it is important to avoid the trivial solution where the shift matrix is equal to 0. Thus,

$$\begin{aligned} \boldsymbol{S}^* = \arg\min_{\boldsymbol{\lambda}} \|\boldsymbol{S}\|_0 \;\; \text{s.t} \;\; \boldsymbol{S} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T = \sum_{k=1}^{N}\lambda_k \boldsymbol{v}_k\boldsymbol{v}_k^T, \\ \operatorname{diag}(\boldsymbol{S}) = \boldsymbol{0}, \;\; \boldsymbol{S} = \boldsymbol{S}^T, \sum_j \boldsymbol{S}_{j1} = 1. \end{aligned} \tag{2.7}$$

As can be seen, due to the optimization of the $\ell_0$-*norm*, the expression (2.7) is not a convex function, which implies that it is *NP*-hard. In order to solve the problem, a common convex relaxation is applying the $\ell_1$-*norm* [13]. Thus,

$$\hat{\boldsymbol{S}}^* = \arg\min_{\boldsymbol{S},\boldsymbol{\lambda}} \|\boldsymbol{S}\|_1 \;\; \text{s.t} \;\; \boldsymbol{S} = \sum_{k=1}^{N} \lambda_k \boldsymbol{v}_k \boldsymbol{v}_k^T, \; \text{diag}\,(\boldsymbol{S}) \; = \; \boldsymbol{0}, \; \boldsymbol{S} = \boldsymbol{S}^T, \; \sum_j \boldsymbol{S}_{j1} = \boldsymbol{1}. \quad (2.8)$$

To solve the problem by using the well known algorithm *BP* [1] [16], we need to reformulate the optimization problem. In order to do it, it is necessary to carry out some more definitions starting by the Khatri-Rao product [2], $\boldsymbol{W} = \boldsymbol{V} \odot \boldsymbol{V}$ [17, 18]. The shift matrix can be written $\boldsymbol{s} = \text{vec}(\boldsymbol{S})$ as $\boldsymbol{s} = \boldsymbol{W}\boldsymbol{\lambda}$ where $\boldsymbol{\lambda} = \text{diag}\,(\boldsymbol{\Lambda})$. At this point, it is necessary to define set $D$, containing the indices of $\boldsymbol{s}$ corresponding to the diagonal elements of the shift matrix. So, by using this set it is possible to take the corresponding rows of $\boldsymbol{W}$ to form $\boldsymbol{W}_D$. Then it is necessary to define

$$\boldsymbol{M} = \left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\dagger\right)_{D^c} \; \in \; \Re^{N^2 - N \times N^2}, \quad (2.9)$$

where $D^c$ denotes the complement set of $D$. Note that matrix $\boldsymbol{M}$ is the orthogonal projector onto the kernel of $\boldsymbol{W}^T$ constrained to elements in $D^c$. Now, let us define vector $\boldsymbol{e}_1$ to denote the first canonical basis vector and $\boldsymbol{1}_{N-1}$ being a ones vector with $N-1$ entries. With these new definitions, we are able to construct

$$\boldsymbol{R} = [\boldsymbol{M}, \boldsymbol{e}_1 \otimes {}^3\boldsymbol{1}_{N-1}] \; \in \; \Re^{N^2 - N \times N^2 + 1}. \quad (2.10)$$

The optimization problem can be now written as

$$\arg\min_{\boldsymbol{s}} \|\boldsymbol{s}\|_1 \;\; \text{s.t} \;\; \left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\dagger\right)\boldsymbol{s} = \boldsymbol{0}, \; \boldsymbol{s}_D = \boldsymbol{0}, \; (\boldsymbol{e}_1 \otimes \boldsymbol{1}_N)^T \boldsymbol{s} = 1. \quad (2.11)$$

In order to finally obtain the problem in a formulation able to be solved using the *BP* algorithm, it is important to note that the projector $\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\dagger$ is symmetric. This allows us to write the first equality in (2.11) as

$$\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\dagger\right)_D^T \boldsymbol{s}_D + \boldsymbol{M}^T \boldsymbol{s}_{D^c} = \boldsymbol{0}. \quad (2.12)$$

Finally, having $\boldsymbol{b} = [0 \; 0 \; 0 \; \cdots 1 \;]^T$, the problem can be seen as

$$\arg\min_{\boldsymbol{s}_{D^c}} \|\boldsymbol{s}_{D^c}\|_1 \;\; \text{s.t} \;\; \boldsymbol{R}^T \boldsymbol{s}_{D^c} = \boldsymbol{b}. \quad (2.13)$$

### 2.2.3 Results

In order to evaluate the performance of the techniques, a benchmark is used in the representation of the estimated connections. At the right side of the figure 2.2, we have the most important or weighted connections while in the left side we have the unconnected pair of nodes, with weight 0. If the technique estimates a connection that exists in the

[1] Basis Pursuit form: $\arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1$ subject to $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$

[2] Khatri-Rao also known as the column-wise Kronecker product

[3] Kronecker product.

real adjacency matrix, the counter represented with the blue line is increased. On the other hand, in case the technique estimates a false connection, in other words, creates a false connection between a pair of nodes $i$ and $j$, the counter represented by the red line is increased. Both of these counters are accumulative. The black horizontal line represents the number of connections that the graph, this is the adjacency matrix, has. To sum up, the objective is to reach this horizontal line, that represents the number of real connections, with the blue counter, correct estimations, while having 0 errors in the red counter, or in other words errors in the connections estimation, see figure 2.3.

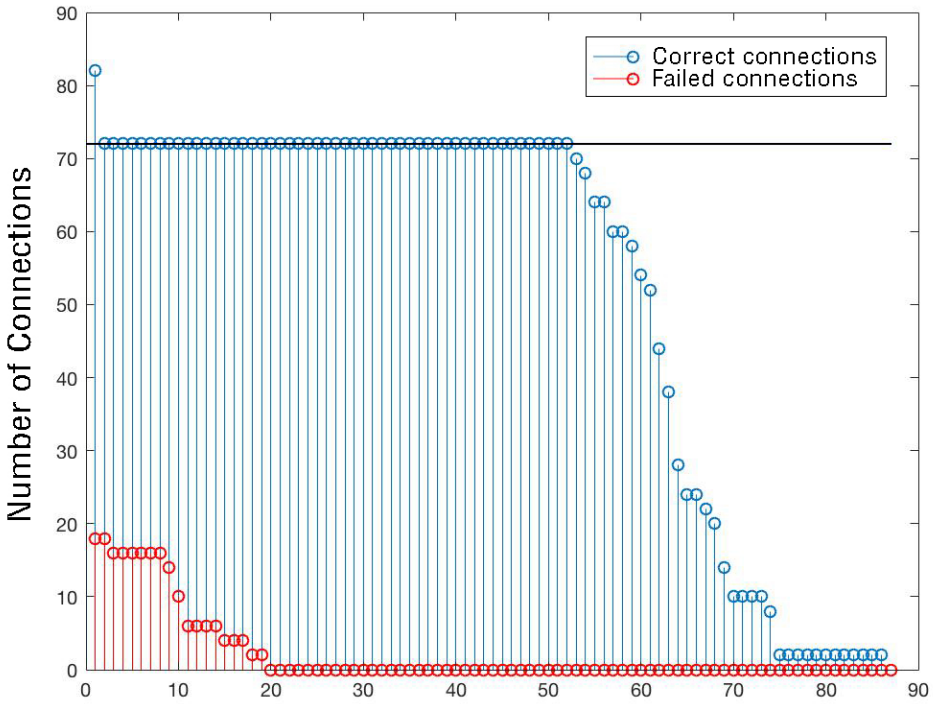


Figure 2.2: Simulation 1, Cumulative correct and failed connections. This simulation can be found in [1]

By monitoring the difference between the estimated eigenvalues, it is possible to see that when two of them, $\lambda_i$ and $\lambda_j$, are really or totally similar there is a flip in the eigenvectors' matrix between columns $i$ and $j$. This flip is the responsible for the errors produced while performing the estimation. By comparing figure 2.2, and figure 2.3 it can be seen that the technique studied is not reliable enough in all cases. It works correctly when there are not two similar eigenvalues. Hence, there is not a flip between two contiguous eigenvectors.

Thus, our next objective is to try to avoid these possible flips between the columns of the eigenvectors matrix to avoid committing estimation errors.

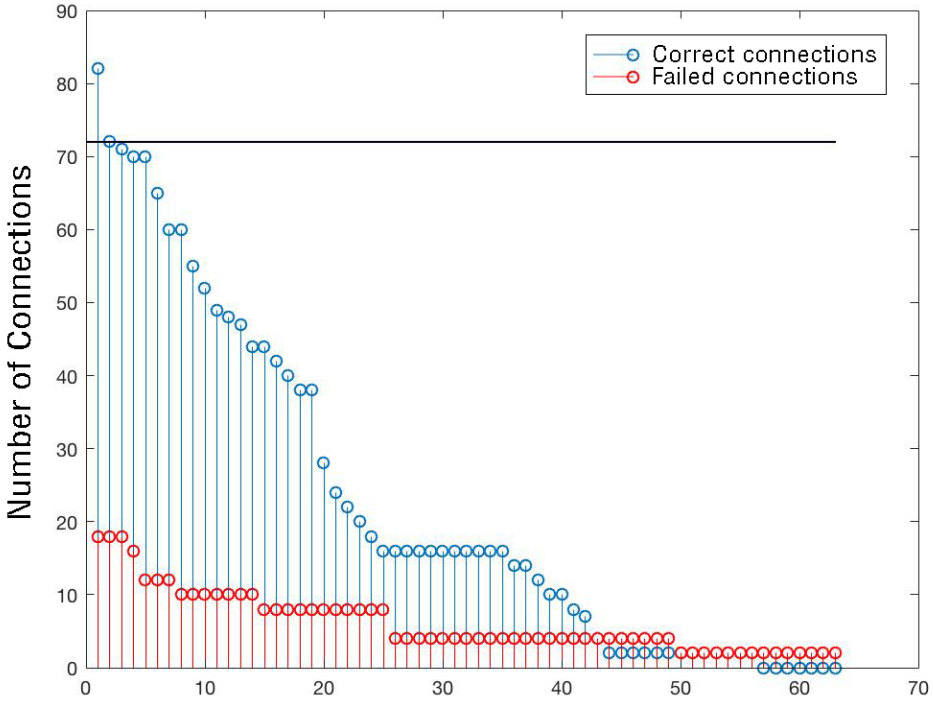


Figure 2.3: Simulation 2, Cumulative correct and failed connections. This simulation can be found in [1]

### 2.2.4 Orthogonal Matching Pursuit

As it was explained in 2.2.2, it is common to find scenarios where the available information is limited. In those cases the algorithm *BP* is not reliable under any condition, see figure 2.2. This is the reason why a relaxation to the *BP* problem is used [19]. This new algorithm is called *Orthogonal Matching Pursuit* (*OMP*).

#### Problem Formulation

In order to estimate the graph spectral template when

- the number of realizations is not high, or
- the knowledge about the signal or the noise subspaces is limited

the *OMP* algorithm can be used. It has more degrees of freedom than the *BP* that allows the algorithm to provide a solution to the problem when the *BP* is incapable. This new algorithm has a form similar to the *BP* but with a relaxation that takes the form of an error

$$\arg\min_{\boldsymbol{s}_{D^c}} \|\boldsymbol{s}_{D^c}\|_1 \quad \text{s.t} \quad \|\hat{\boldsymbol{R}}^T \boldsymbol{s}_{D^c} - \boldsymbol{b}\|_2 \leq \boldsymbol{\epsilon}. \tag{2.14}$$

In order to work, this algorithm needs to know the percentage of connectivity of the adjacency matrix. This is a critical design parameter but, in order to compare both algorithms, *BP* vs. *OMP*, the exact graph's connectivity is used. By indicating a lower connectivity, the algorithm estimates a more sparse solution to the problem. This means

that by controlling this parameter it is possible to manage the importance of the connections that the algorithm estimates. The degree of sparsity is usually an unknown parameter and sparsity estimators are really complex to design.

### Results

This example has a connectivity percentage of the 60% and the available data are $10N$ snapshots. In order to estimate the most important connections, the algorithm is configured with a 40% of connectivity. As can be seen in figure 2.4 the algorithm is not reliable enough because it creates a lot of false connections. As it was explained, the main goal is to estimate all the connections without creating any false one. In the next section, a new algorithm is presented to improve the *BP* and *OMP* algorithms.

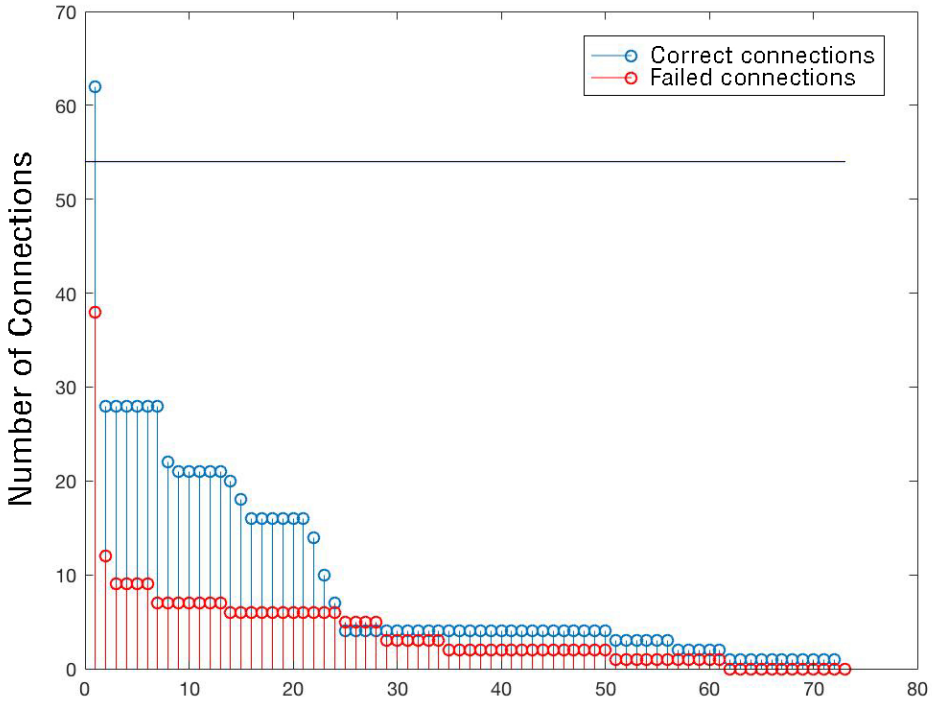


Figure 2.4: Example when using *OMP* algorithm. This simulation can be found in [1]

## 2.2.5 Searching for a Robust Algorithm

As seen in the results section 2.2.3, it is necessary to find a more robust algorithm when trying to estimate the graph's topology. In this section, we attempt to improve the estimation by distinguishing two different subspaces from the taken data. The expected result is that by differentiating the eigenvectors that give us more information from those that do not contribute with reliable information it is possible to improve the performance of the topology inference techniques.

**Problem Formulation**

In this section, the starting conditions are equal as in the previous one. Given a graph signal $\boldsymbol{x}_n = \boldsymbol{H}\boldsymbol{w}_n$, being $\boldsymbol{H}$ the graph filter, it is possible to recover the sparsest shift matrix $\boldsymbol{S}$ by means of *BP* (2.13). The problem is that this algorithm needs a perfect knowledge of covariance matrix eigenvectors. Most of times, this is impossible because not enough data is available. When only $K$ out of $N$ eigenvectors are perfectly known, we face with an *incomplete spectral template*. A new idea is presented in this section, where only a fraction of the total eigenvectors is considered to be perfectly known. Let us formulate our new problem where matrix $\boldsymbol{V}_k$ contains the known eigenvectors' subspace. The complementary subspace, containing the unknown eigenvectors is defined as $\boldsymbol{V}_{\hat{k}}$. In order to decide whether an eigenvector belongs to the known or unknown subspace, the difference between consecutive eigenvalues is used. If this value is less than a fixed threshold, this pair of eigenvalues are considered to be equal. Thus, the eigenvectors associated belong to the unknown subspace. Because of that, it is important to note that eigenvectors must be put in the unknown subspace at least by pairs. Taking these concepts into account, the new conditions can be written as

$$\begin{aligned} \boldsymbol{S}^* = \arg\min_{\boldsymbol{S},\boldsymbol{S}_{\bar{k}},\boldsymbol{\Lambda}} \|\boldsymbol{S}\|_1 \;\text{ s.t }\; \boldsymbol{S} = \boldsymbol{S}_{\bar{k}} + \sum_{k=1}^{K} \lambda_k\, \boldsymbol{v}_k\, \boldsymbol{v}_k^T, \boldsymbol{S}_{\bar{k}}\boldsymbol{V}_k = \boldsymbol{0}, \\ \operatorname{diag}(\boldsymbol{S}) = \boldsymbol{0}, \boldsymbol{S} = \boldsymbol{S}^T,\ \sum_j \boldsymbol{S}_{j1} = \boldsymbol{1}. \end{aligned} \tag{2.15}$$

As before, our objective is to write the problem in order to use the algorithm *BP* but considering the new constraints. Thus, it is necessary to vectorize them. Following the same procedure as in the previous section 2.2.2

$$\boldsymbol{S} = \boldsymbol{S}_{\bar{k}} + \sum_{k=1}^{K} \lambda_k\, \boldsymbol{v}_k\, \boldsymbol{v}_k^T \;\Rightarrow\; \boldsymbol{s} = \boldsymbol{s}_{\bar{k}} + \boldsymbol{W}_k\boldsymbol{\lambda}, \tag{2.16}$$

where, as before, $\boldsymbol{W}_k = \boldsymbol{V}_k \odot \boldsymbol{V}_k$. To ensure the symmetry of the estimated shift matrix, the second constraint is used

$$\boldsymbol{B}\boldsymbol{s} = \boldsymbol{0}, \tag{2.17}$$

where $\boldsymbol{B} \in \Re^{\binom{N}{2}\times N^2}$ indicates which rows are the vectorized forms of $\boldsymbol{B}^{(i,j)}\ \forall\ i,j \in \{1,\cdots,N\}$ and $\boldsymbol{B}^{(i,j)} \in \Re^{N\times N}\ \forall\ i<j$ such that $\boldsymbol{B}_{ij}^{(i,j)} = 1$ and $\boldsymbol{B}_{ji}^{(i,j)} = -1$. The third and forth constraints do not change in comparison to (2.11). Finally,

$$\boldsymbol{S}_{\bar{k}}\boldsymbol{V}_k = \boldsymbol{0} \Rightarrow \left(\boldsymbol{I}\otimes\boldsymbol{V}_k^T\right)\boldsymbol{s}_{\bar{k}} = \boldsymbol{0}. \tag{2.18}$$

It is important to note that in this problem formulation, by contrast to the one in 2.2.2, there are constraints on $\boldsymbol{S}$ and $\boldsymbol{S}_{\bar{k}}$. Some further modifications are needed to allow us to formulate the problem in a *BP* format,

$$\boldsymbol{t}^T = \left[\boldsymbol{s}^T\boldsymbol{s}_{\bar{k}}^T\right], \boldsymbol{\Upsilon} = \left[\boldsymbol{I}_{N^2}\ \boldsymbol{0}_{N^2\times N^2}\right], \boldsymbol{P}^T = \left[\boldsymbol{P}_1^T\ \boldsymbol{P}_2^T\right], \boldsymbol{b}^T = \left[\boldsymbol{0}_z^T\ 1\right], \tag{2.19}$$

where $z = N^2 + N + \binom{N}{2} + NK + 1$. Let us explain how the constraints matrix $\boldsymbol{P}$ is defined.

$$\boldsymbol{P}_1 = \left[\left(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^{\dagger}\right)^T,\ \ \boldsymbol{I}_D^T,\ \ \boldsymbol{B}^T,\ \ \boldsymbol{0}_{NK\times N^2}^T,\ \ \left(\boldsymbol{e}_1\otimes\boldsymbol{1}_N\right)^T\right], \tag{2.20}$$

and

$$\boldsymbol{P}_2 = \left[\left(\boldsymbol{W}\boldsymbol{W}^{\dagger} - \boldsymbol{I}\right)^T, \ \mathbf{0}^T_{N\times N^2}, \ \mathbf{0}^T_{\binom{N}{2}\times N^2}, \ \left(\boldsymbol{I}\otimes \boldsymbol{V}_k^T\right)^T, \ \mathbf{0}^T_{1\times N^2}\right]. \tag{2.21}$$

These constraints are the same as in 2.2.2 but applied to the known, $\boldsymbol{P}_1$, and the unknown, $\boldsymbol{P}_2$, subspace. Thus, at this point it is possible to write the optimization problem considering both subspaces as

$$\arg\min \|\boldsymbol{\Upsilon}\boldsymbol{t}\|_1 \ \text{ s.t } \ \boldsymbol{P}^T\boldsymbol{t} = \boldsymbol{b}. \tag{2.22}$$

### Results

In this simulation, it can be seen that this method is much more reliable than the previous one. In this case, the algorithm is able to estimate the graph's connections without creating any false connection. This allows us to set a threshold to decide when the estimated connection is considered zero or not. As can be seen, the margin to decide this threshold is quite wide because there is a considerable distance between the last values of the connections and the point when the algorithm estimates a false connection.

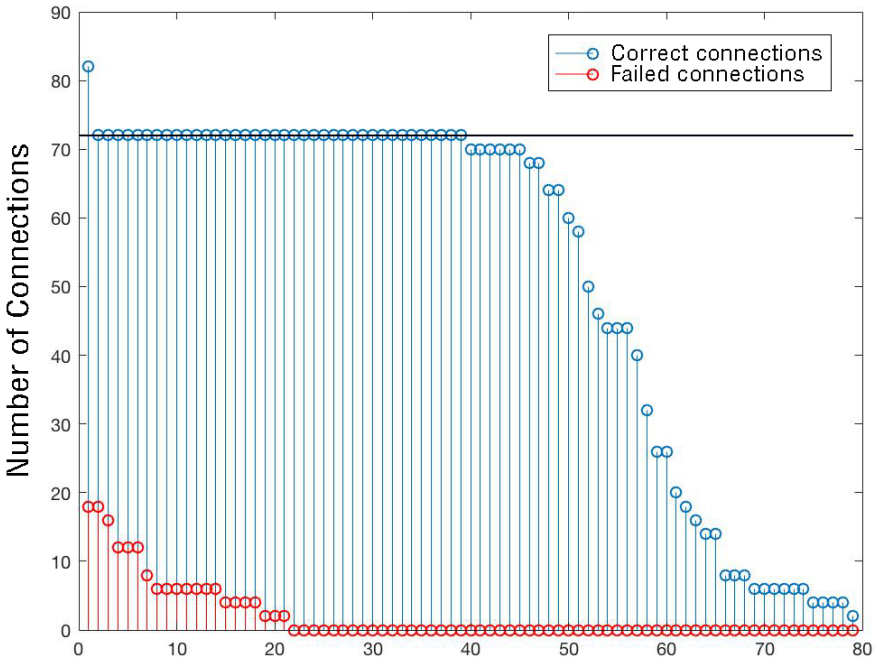


Figure 2.5: Example when using the robust algorithm. This simulation can be found in [1]

## 2.2.6 Robust Orthogonal Matching Pursuit

### Problem Formulation

Let us remember that, in some cases the available information or signal knowledge is not enough to use the *BP*. Furthermore, in section 2.2.4 it was appreciated that the *OMP* algorithm does not provide a reliable solution to the problem. In this section, the *OMP* assembles with the Robust Method to try to improve its performance and reliability. By

following the same notation as in section 2.2.5 and section 2.2.4, it is derived that the optimization problem is

$$\arg\min \|\boldsymbol{\Upsilon t}\|_1 \quad \text{s.t} \quad \|\hat{\boldsymbol{P}}^T \boldsymbol{t} - \boldsymbol{b}\|_2 \leq \boldsymbol{\epsilon}. \tag{2.23}$$

In (2.23) it is seen that the restrictions are applied to both subspaces while a certain error is allowed.

### Results

As studied in 2.2.4, the *OMP* algorithm is not reliable enough. In this section, the results of applying this algorithm jointly with the robust method are shown. This simulation was done with 10 nodes with a graph's percentage of connectivity of 80%. In order to show the potential of this combined techniques to estimate the most important connections without making any errors, a percentage of 40% was indicated in the algorithm.

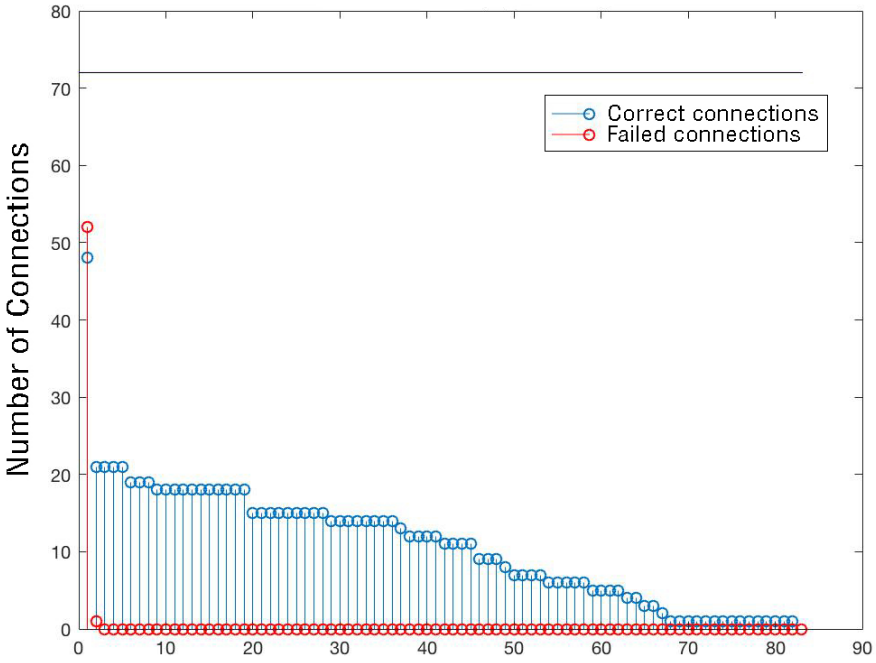


Figure 2.6: Robust *OMP*. This simulation can be found in [1]

In some applications, it might be interesting to know the most important connections instead of trying to estimate all the graph connections. As it was shown, to obtain a reliable result when using this technique it must exist an agreement between the number of connections we are interested in and the number of errors we will allow.

## 2.3 New Robust Enhanced Network Topology Inference with Noisy Spectral Templates

In the previous section, the techniques to infer graphs' topology from imperfect spectral templates were reviewed. In this section, it is studied the case of having *noisy spectral*

*templates*. It starts by explaining why the *OMP* algorithm is not reliable enough. Then, it is proposed a method to work with noisy spectral templates. To finish this section, some simulations are performed to better compare the techniques.

### 2.3.1 Restricted Isometry Property

In the Final Degree thesis it was seen, in a heuristic way, that the *OMP* algorithm was not performing as well as it might be expected. In later studies, it was seen that the *OMP* algorithm does not always work. Let us suppose that, we have an observation such as

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}, \tag{2.24}$$

where $\boldsymbol{x} \in \Re$ is the object, or in our case the graph signal, that we want to recover. The available snapshots are represented as $\boldsymbol{y} \in \Re$, and $\boldsymbol{\Phi}$ is a known matrix. As explained in section 2.2.4, it is not always possible to have all the information needed to correctly estimate the covariance matrix. Thus, the case with fewer equations than unknowns was studied and the *OMP* was proposed as an algorithm to recover the graph signal. Nonetheless, this approach was not always working. Let us study whether it is possible to recover the graph signal $\boldsymbol{x}$. By doing the same assumption as in the previous section 2.2, let us suppose that the solution to the problem is sparse. This is

$$\arg\min_{\hat{\boldsymbol{x}}\in\Re^n} \|\hat{\boldsymbol{x}}\|_1 \text{ s.t } \boldsymbol{\Phi}\hat{\boldsymbol{x}} = \boldsymbol{y}. \tag{2.25}$$

The *BP* or *OMP* can be used to recover exactly $\boldsymbol{x}$ if two conditions are fulfilled. The first one is that the graph signal $\boldsymbol{x}$ is sufficiently sparse, and the second one is that the known matrix $\boldsymbol{\Phi}$ obeys the condition known as the *Restricted Isometry Property (RIP)*. The general expression of this property is

$$(1-\delta)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{\Phi}\boldsymbol{x}\|_2^2 \leq (1+\delta)\|\boldsymbol{x}\|_2^2. \tag{2.26}$$

The *RIP* is a sufficient condition to guarantee that $\boldsymbol{x}$ can be recovered [20]. The most important implication for us is that it is a sufficient condition to use the $\ell_1$-minimization, which is the canonical convex optimization used, to recover the graph signal [21].

Let us formulate the *RIP* with our problem. The object that we want to recover is the graph signal $\boldsymbol{s}_{D^c}$. The known matrix is the matrix formed as

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{M}^T \\ \hat{\boldsymbol{e}}_1^T \end{bmatrix}, \tag{2.27}$$

where $\hat{\boldsymbol{e}}_1^T = \boldsymbol{e}_1^T \otimes \boldsymbol{1}_{N-1}$. In our case, if the *RIP* condition is not obeyed, it is not possible to guarantee that the solution obtained with $\|\cdot\|_0$ is equivalent to the solution obtained with the convex relaxation $\|\cdot\|_1$. It can be demonstrated that, sometimes, the *RIP* is not fulfilled. Thus, the solution obtained with the $\|\cdot\|_0$ is not equivalent to the one obtained by the $\|\cdot\|_1$. Therefore, the *OMP* is not always able to estimate the spectral template without errors (see ref.[22, 23]). The *RIP* condition using our notation is

$$(1-\delta)\|\boldsymbol{s}_{D^c}\|_2 \leq \left\| \begin{bmatrix} \boldsymbol{M}^T \\ \hat{\boldsymbol{e}}_1^T \end{bmatrix} \boldsymbol{s}_{D^c} \right\|_2 \leq (1+\delta)\|\boldsymbol{s}_{D^c}\|_2. \tag{2.28}$$

Developing the expression (2.28),

$$\begin{bmatrix} \boldsymbol{M}^T \\ \hat{\boldsymbol{e}}_1^T \end{bmatrix} \boldsymbol{s}_{D^c} = \begin{bmatrix} \boldsymbol{0} \\ \hat{\boldsymbol{e}}_1^T \boldsymbol{s}_{D^c} \end{bmatrix}. \tag{2.29}$$

Then, by computing the $\ell_2$-*norm* of the term in (2.29), it is obtained that this term is equivalent to $\boldsymbol{s}_{D^c}^T \boldsymbol{e}_1 \boldsymbol{e}_1^T \boldsymbol{s}_{D^c}$. It can be expressed as: $\text{trace}[\boldsymbol{s}_{D^c} \boldsymbol{s}_{D^c}^T \boldsymbol{E}] = \text{trace}[\boldsymbol{P}_{D^c} \boldsymbol{E}]$. Finally, the *RIP*, applied to our problem, is formulated as

$$(1-\delta) \sum_{i=1}^{N^2} [\boldsymbol{P}_{D^c}]_{i,1} \leq \sum_{i=1}^{N-1} [\boldsymbol{P}_{D^c}]_{i,1} \leq (1+\delta) \sum_{i=1}^{N^2} [\boldsymbol{P}_{D^c}]_{i,1}. \tag{2.30}$$

As can be seen in (2.30), the only possible value of $\delta$ to ensure that the lower bound is fulfilled is $\delta = 1$. Otherwise, it is not possible to guarantee that the central term is greater than the lower bound. Thus, the *RIP* is not fulfilled.

Furthermore, the *OMP* needs to decide a target residual. This value is directly related with the sparsity degree of the graph and the *SNR*. In order to obtain this value, a *sparsity estimator* is needed. These estimators are really complex. Moreover, the *OMP* is really dependent to this value, little changes produce high deviations in the *OMP* output [24].

### 2.3.2 Noisy Spectral Templates

In order to use the previous methods, some *a priori* information is needed to subdivide the space in the known and the unknown subspaces. Alternatively, the algorithms are not reliable enough because the *RIP* conditioned is not obeyed 2.3.1. Furthermore, in the previous sections, it was assumed that the signal is $\boldsymbol{x} = \boldsymbol{H}\boldsymbol{w}$, being $\boldsymbol{w}$ the driving noise. Nonetheless, in some cases the signal may contain a noise component

$$\boldsymbol{x} = \boldsymbol{H}\boldsymbol{w} + \boldsymbol{n}, \tag{2.31}$$

where $\boldsymbol{n}$ is, in general, a colored noise. Now, let us apply this new consideration to the equations. In this case, the covariance matrix of the signal is modified including the covariance matrix of the noise

$$\boldsymbol{C}_{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{H}^T + \boldsymbol{C}_{\boldsymbol{n}}. \tag{2.32}$$

Assuming *AWGN*, equation (2.32) can be rewritten as $\boldsymbol{H}\boldsymbol{H}^T + \sigma^2 \boldsymbol{I}$. Now, the *Eigendecomposition* (*EVD*) is performed to obtain the eigenvectors of the covariance matrix. In this case, due to noise, it is obtained

$$\text{EVD}(\boldsymbol{C}_{\boldsymbol{x}}) = \hat{\boldsymbol{V}} \boldsymbol{\Lambda} \hat{\boldsymbol{V}}. \tag{2.33}$$

Note that matrix $\hat{\boldsymbol{V}}$ contains the eigenvectors plus noise. Then all the expressions can be rewritten taking into account the noise. The problem to solve is

$$\hat{\boldsymbol{S}}_1^* = \min_{\boldsymbol{\Lambda}, \boldsymbol{S}} \|\boldsymbol{S}\|_1 \text{ s.t } \left\| \boldsymbol{S} - \hat{\boldsymbol{V}} \boldsymbol{\Lambda} \hat{\boldsymbol{V}} \right\|_F \leq \epsilon. \tag{2.34}$$

Now, let us redefine the matrices by considering the noise term

$$\hat{\boldsymbol{R}} = [\hat{\boldsymbol{M}}, \boldsymbol{e}_1 \otimes \boldsymbol{1}_{N-1}] \in \Re^{N^2 - N \times N^2 + 1}, \tag{2.35}$$

where $\hat{\boldsymbol{M}} = \boldsymbol{I} - \hat{\boldsymbol{W}}\hat{\boldsymbol{W}}^{\dagger}$[4] and $\hat{\boldsymbol{W}} = \hat{\boldsymbol{V}} \odot \hat{\boldsymbol{V}}$. Finally, the problem considering the noise term and a certain fixed error is

$$\min_{\hat{\boldsymbol{s}}_{D^c}} \|\hat{\boldsymbol{s}}_{D^c}\|_1 \text{ s.t } \left\|\hat{\boldsymbol{R}}^T \hat{\boldsymbol{s}}_{D^c} - \boldsymbol{b}\right\|_2 \leq \epsilon. \tag{2.36}$$

At this point, the need to find a robust solution able to work with noisy signals brings us to test the behavior of a well known algorithm to solve these problems, the *Least Absolute Shrinkage and Selection Operator* (*LASSO*) algorithm [25]. The main *a priori* advantages of this algorithm are: it does not need previous knowledge about the signal nor the noise subspaces, it does not require to fix a threshold to divide both subspaces as some other methods, and it does not need to know the graph's sparsity to work (i.e the *OMP*).

The general *LASSO* formulation in the alleged *Lagrangian* form is

$$\arg\min_{\beta \in \Re^P} \left\{ \frac{1}{N} \|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}, \tag{2.37}$$

where $N$ is the number of cases, $\boldsymbol{y}$ is the outcome, $\boldsymbol{\beta}$ the parameter to be optimized, and $\boldsymbol{X}$ the covariance matrix. Note that the regularization parameter $\lambda$ promotes sparsity. Higher this parameter is the more sparse the solution will be. Let us adapt it to the graph notation where the parameter that we want to optimize is the graph signal $\hat{\boldsymbol{s}}_{D^c}$, and the matrix that gives us information is $\hat{\boldsymbol{R}}^T$.

$$\arg\min_{\hat{\boldsymbol{s}}_{D^c}} \left\{ \frac{1}{N} \left\|\hat{\boldsymbol{R}}^T \hat{\boldsymbol{s}}_{D^c} - \boldsymbol{b}\right\|_2^2 + \lambda \|\hat{\boldsymbol{s}}_{D^c}\|_1 \right\}. \tag{2.38}$$

The regularization parameter, $\lambda$, is chosen taking into account the quality of the signal obtained, the *SNR*.

### 2.3.3 Simulations

In order to be able to compare the different algorithms, the following metric was defined

$$RMSE = \frac{1}{N^2} \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} (\boldsymbol{S}_{ij} - \hat{\boldsymbol{S}}_{ij})^2}, \tag{2.39}$$

where $\boldsymbol{S}$ is the graph's shift, and $\hat{\boldsymbol{S}}$ is the estimated shift matrix. *RMSE* value is computed for every simulation and then the mean value of all the simulations is computed to obtain the final value for a given regularization parameter $\lambda$. In order to generate random graphs *Erdös-Rényi's* graphs were used. These graphs consist of $N$ nodes and each pair of nodes might be connected according to a connection probability. This probability is defined as $p/N$ with $0 \leq p \leq N$. As the connections are equiprobable, these graphs have no spatial structure. More detailed information about Erdös-Rényi's graphs can be found in [26, 27]. These graphs are commonly used in different fields [28, 29, 30, 31].

[4] being † the pseudoinverse operator applied to the matrix.

### Comparison of *BP* and *LASSO*

In order to see the improvement achieved when using *LASSO* in front of using *BP* algorithm, some simulations have been done. In figure 2.7 it can be seen that *LASSO* is able to estimate, in average, the graph topology with a lower *RMSE* than the *BP*. As explained, *BP* algorithm has some limitations and these produce an algorithm that does not provide reliable enough results.

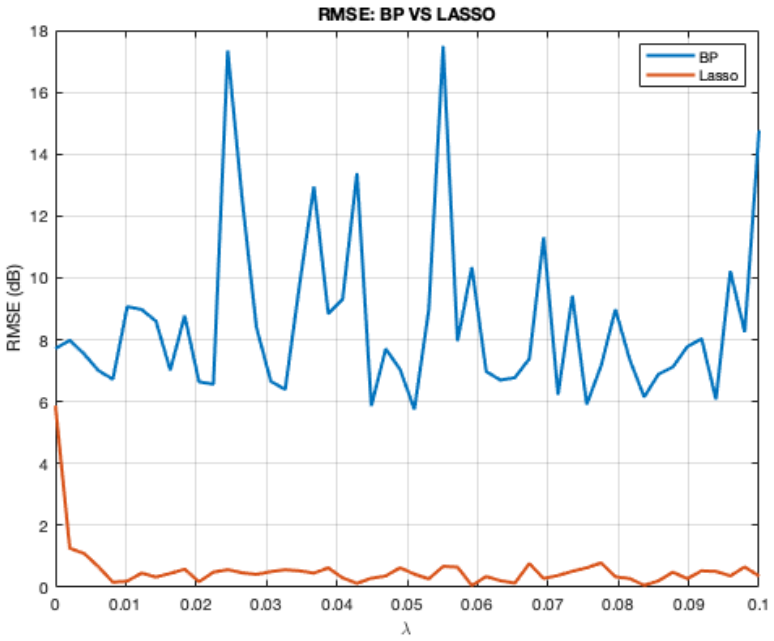


Figure 2.7: *RMSE* comparison *BP* vs. *LASSO* algorithms

In this subsection some simulations are carried out to test how *LASSO* behaves for different regularization parameters $\lambda$, for different number of nodes, and for different graph's percentage of connectivity, as well as to see the dependencies between these magnitudes. Let us start with figure 2.8. In this figure it can be seen the behavior of the *RMSE* function of the number of nodes in the graph and different values of $\lambda$. Firstly, note that as the number of nodes increase the *RMSE* value tends to 0. The second thing that can be observed is that for higher values of $\lambda$ the better *RMSE* is obtained. In order to understand this result, it is important to remember that when using the *BP* algorithm the value of the regularization parameter $\lambda$ tends to infinite, $\lambda \to \infty$. In the other hand, when using *LASSO*, the degree of freedom introduced allows the algorithm to take into account the $\ell_2$-*norm* factor to improve the estimation. The importance of each of both factors is determined by the value of $\lambda$. The higher this value is the more similar the *LASSO* algorithm is to the *BP*. Nonetheless, it is important to note that by allowing a little error (the *norm 2* factor) it is possible to dramatically improve the results of the *BP* algorithm as seen in figure 2.7.

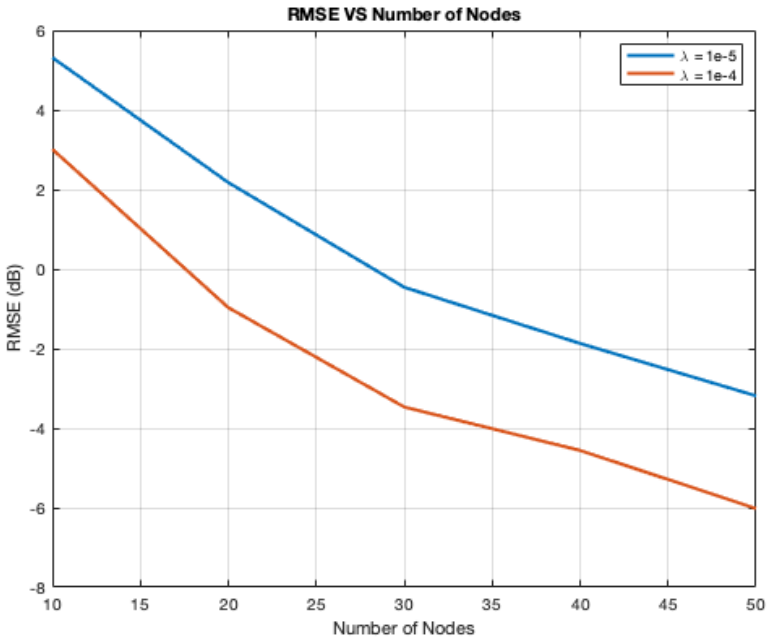


Figure 2.8: *RMSE* vs. Number of nodes for different $\lambda$s

### Percentage of graph connectivity

In this subsection, it will be seen whether exists any relation to improve the *RMSE* (this is to obtain the best graph's topology estimation possible), between the percentage of graph connectivity and regularization parameter $\lambda$. Looking at figure 2.9 it can be seen that, as in the previous case (figure 2.8) for higher values of $\lambda$ the *RMSE* decreases. Nevertheless, it is interesting to note that the percentage of connectivity does not modify the *RMSE* obtained, this is the percentage of connectivity does not vary depending on the value of $\lambda$. To better see this behavior, let us focus in figure 2.10. In this figure the value of $\lambda$ is fixed, and it is plotted the value of the *RMSE* for different number of nodes. After seeing this figure it can be stated that the percentage of connectivity does not depend on $\lambda$. This result is very interesting because this means that it is possible to use *LASSO* without knowing the sparsity degree. Thus, there is not need to use *sparsity estimators*. It is interesting to emphasize that as the number of nodes increases the *RMSE* tends to 0.

### Robust method VS LASSO

In this subsection, it is compared the robust method and the *LASSO* (section 2.3.2). With simulations, it can be checked that *LASSO* has a better performance, under the *RMSE* criteria defined in this paper, than the robust method. In figure 2.11 and figure 2.12 it is appreciated the improvement achieved by *LASSO*. In this figure, the percentage of connectivity is fixed to 40% which means that the graph is quite sparse. It also can be seen that for the highest $\lambda$ the *LASSO* algorithm works better. The higher $\lambda$ is the more importance is given to the $\ell_1$-*norm* term. In order to justify the improvement, the next step would be to perform the formal mathematical validation of the obtained results.

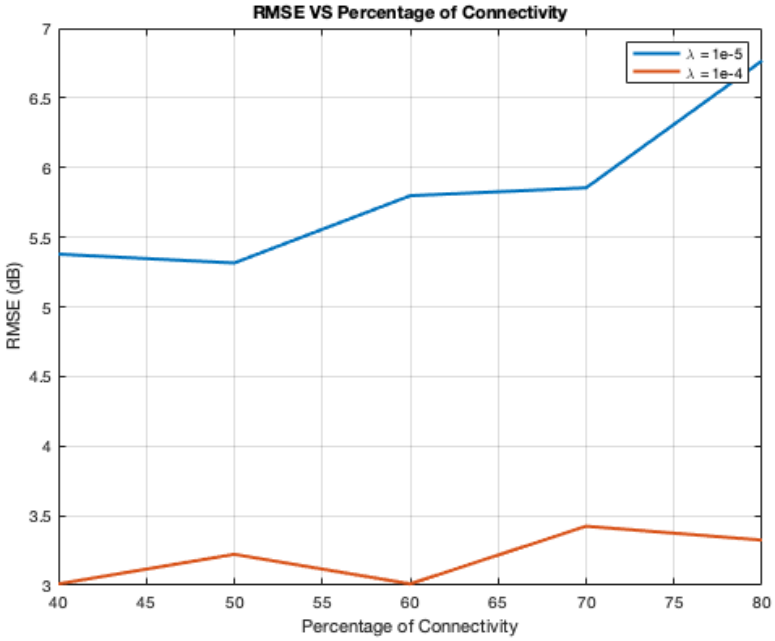


Figure 2.9: *RMSE* vs. Percentage of graph's connectivity for different values of $\lambda$

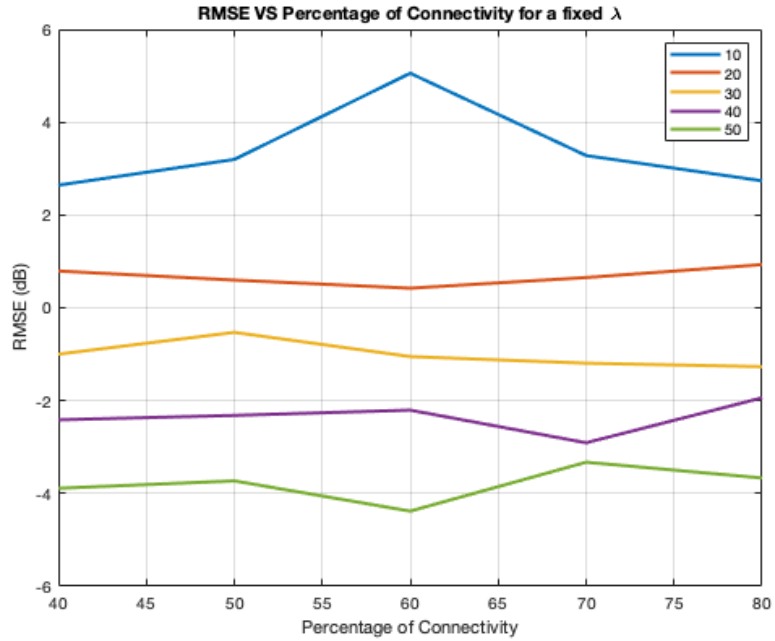


Figure 2.10: *RMSE* vs. Percentage of graph's connectivity for a given regularization parameter $\lambda$ and different number of nodes

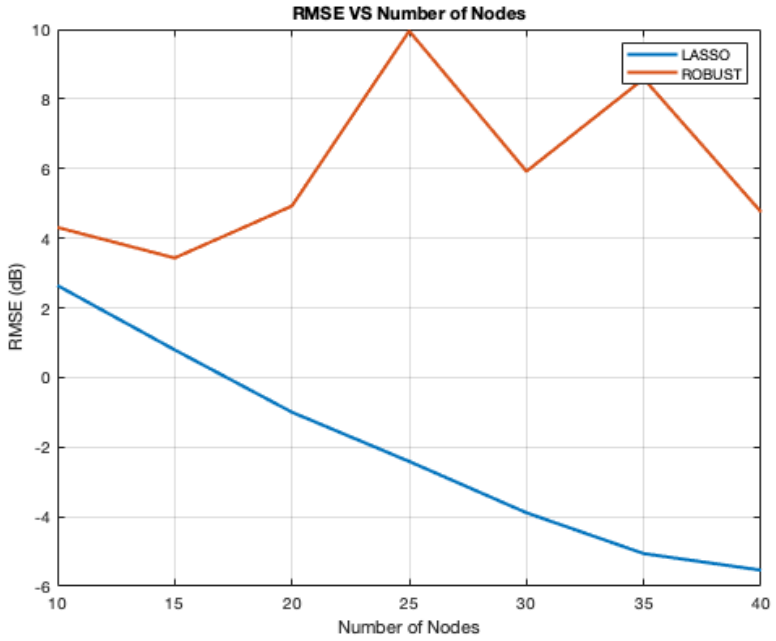


Figure 2.11: Comparison of both methods *RMSE* vs. Number of nodes for a fixed regularization parameter in *LASSO* algorithm of $\lambda = 10^{-4}$

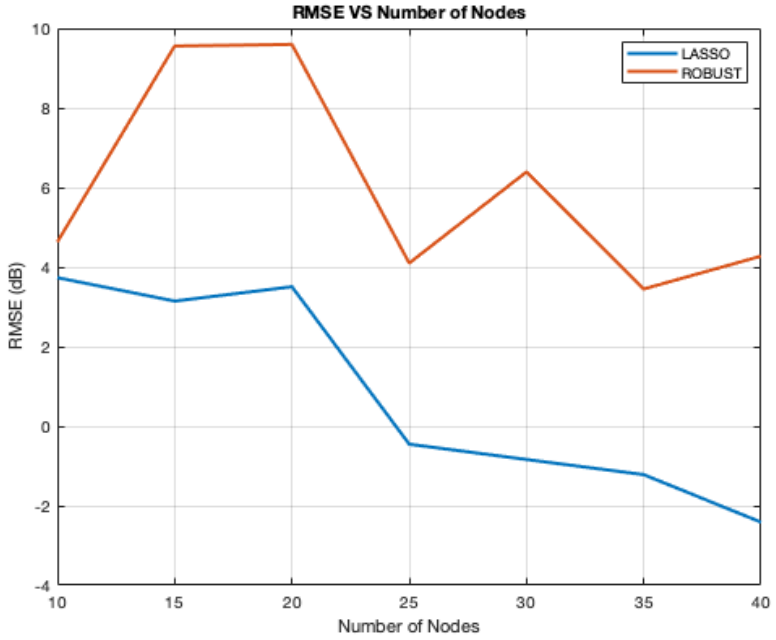


Figure 2.12: Comparison of both methods *RMSE* vs. Number of nodes for a fixed regularization parameter in *LASSO* algorithm of $\lambda = 10^{-5}$

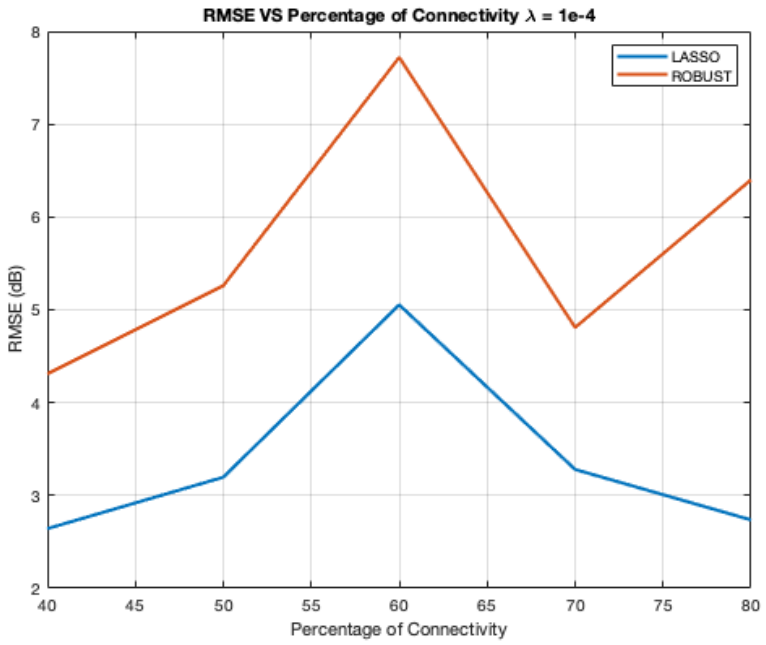


Figure 2.13: Comparison of both methods *RMSE* vs. Percentage of graph's connectivity for a given $\lambda = 10^{-4}$ and different number of nodes

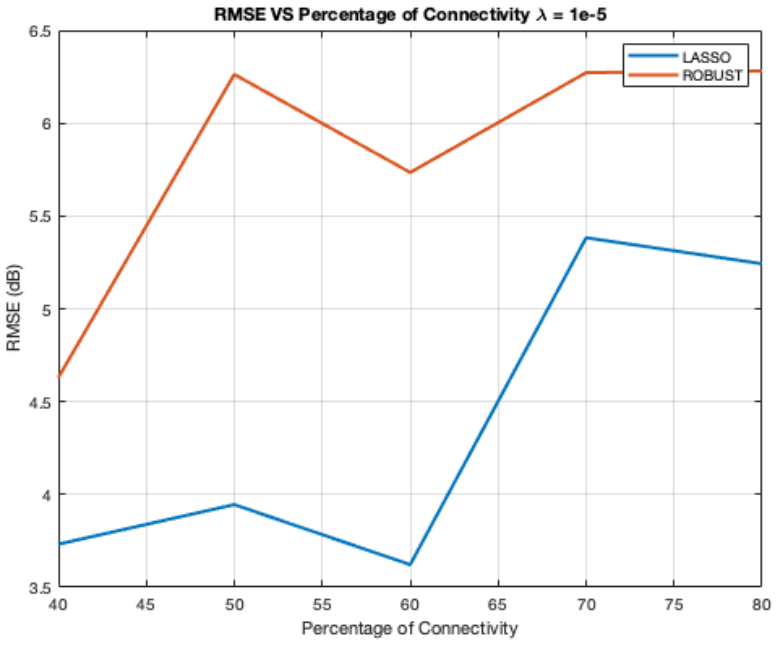


Figure 2.14: Comparison of both methods *RMSE* vs. Percentage of graph's connectivity for a given $\lambda = 10^{-5}$ and different number of nodes

In figure 2.13 and figure 2.14 it is checked the behavior of both algorithms for different sparsity degrees. As it is observed, *LASSO* works better than the robust method for both values of $\lambda$. For higher values of percentage connectivity and lower regularization parameters $\lambda$ the *LASSO* algorithm gets worst. This makes sense because higher is the percentage of connectivity the lower is the graph's sparsity.

## 2.4 Conclusions

In this chapter there were some assumptions about the graph signal which are needed. The first one is that the graph signal is produced as the output of a filter when a driving noise is applied. Although there are multiple cases where the graph signal can be modeled following this criteria, it would be useful to find a technique that allows us to work when the graph signal is not generated by this procedure or when its generation is unknown. Also, as seen, when there are not enough snapshots, this is available data, the algorithms studied are not reliable. Furthermore, in order to use the robust method it is necessary to decide between those eigenvectors that are useful from those which are not. In case of applying the *OMP* algorithm, it is necessary to have prior information about the graph's sparsity or to develop a complex sparsity estimator. Furthermore, by having one or the other the solution obtained is not perfect.

In this last section, it was proposed a robust method to work with noisy spectral templates based on *LASSO*. As showed, the reliability of this algorithm is higher than the obtained with *BP* or *OMP* algorithms. Moreover, with the proposed algorithm, there is no need to know the graph's sparsity degree. This means that no sparsity estimators are needed, which as exposed are complex and do not always work [24]. It was also studied the dependency of the results of *LASSO* function of the regularization parameter. For values around $10^{-4}$ the algorithm works better than for smaller values. It is interesting to note that when the number of nodes tends to infinite the *RMSE* obtained by *LASSO* tends to 0. On the other hand, it was observed that the regularization parameter does not modify the *RMSE* for different percentages of connectivity. To conclude the robust method and *LASSO* algorithm were compared. As stated, the *LASSO* algorithm obtained better results.

In the next chapter another family of techniques is presented where it is possible to work for graphs which are not generated from a filter. Moreover, it is not be necessary to have prior information about the graph's sparsity nor to design an estimator for it.

# Chapter 3

# Statistical Methods

## 3.1 Introduction

It is common to have networks that present similarities between signal elements. So, in order to construct a graph representation of the data it seems a good approach to associate edge weights with nontrivial correlations [32] or with different measurements which are directly related with the signal profiles at incident nodes. Some examples of these measurements and correlations and some applications are the Pearson product-moment correlation used in the quantification of some gene-regulatory interactions, the Jaccard coefficient used for scientific citation networks, the Gaussian radial basis function used to link different measures from a sensor network, or mutual information that allows to detect nonlinear interactions. The goal, when applying these techniques, is to estimate the best undirected graph based on the data available. In these graphs, a connection between two nodes implies that the variables at these nodes are correlated but they might not be physically connected. In figure 3.1 it can be better understood the difference between these two different layers. There are two main possible approaches that can be applied in order to obtain the graph when working with these techniques. On one hand, the use of thresholds that are often manually set in order to obtain the graph that, under a certain criteria, is considered best fits the relational structure of the analyzed data. On the other hand, to define a number $k$ of the top relations and then to retain it for each node. When using these approaches it becomes virtually impossible to measure the accuracy of the obtained graph or somehow to validate the results. In this section, we adopt a methodology in order to study the possible issues and the robustness of these techniques.

This branch of figure 1.1 is the most generic one. It works with no priors nor more information than some snapshots taken from the graph. In this chapter we will use the same graphs to fairly compare the different methods with controlled data. Multiple realizations are performed in order to compute the mean value of the results.

## 3.2 Signal Model

In this section, it is explained the signal model that is used along this chapter. Let us consider $\boldsymbol{x}$ the graph signal. $\boldsymbol{A}$ is not the adjacency matrix but the matrix that indicates

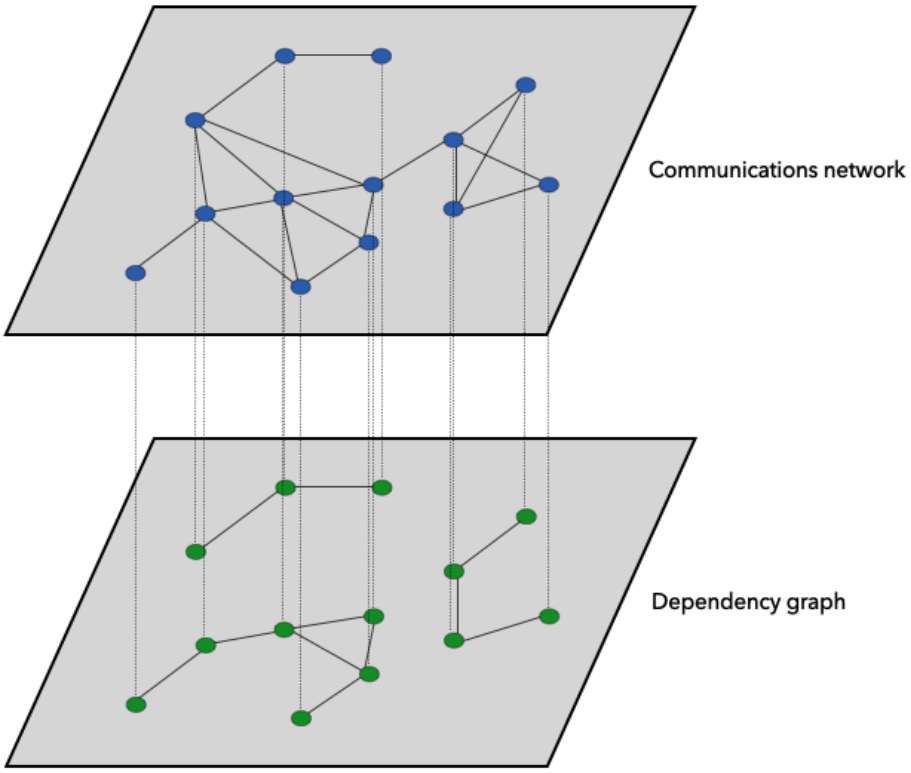


Figure 3.1: Communications network VS dependency graph. Picture based on [2]

the linear combination of graph signals, and $\boldsymbol{\epsilon}$ a random vector

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{\epsilon}. \tag{3.1}$$

In this thesis, the graph signal will be considered to be real, $\boldsymbol{x} \in \Re$. Let us note that matrix $\boldsymbol{A}$ can have values $|a_{ij}| \leq 1$ with the diagonal equal zero. In general it is not symmetric, i.e

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \gamma \\ \alpha & 0 & 0 \\ \beta & \delta & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}. \tag{3.2}$$

As explained, matrix $\boldsymbol{A}$ represent the relation between the signal in node $i$ due to the other nodes. Thus, it is logical to have $\text{diag}(\boldsymbol{A}) = \boldsymbol{0}$. Having this structure 3.1, the graph signal can be written

$$\begin{aligned} (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{x} &= \boldsymbol{\epsilon}; \\ \boldsymbol{x} &= (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{\epsilon}. \end{aligned} \tag{3.3}$$

By defining the graph signal like in (3.3), the definition of the covariance matrix for the case of having a real graph signal as described, is trivial

$$\boldsymbol{\Sigma} = (\boldsymbol{I} - \boldsymbol{A})^{-1}(\boldsymbol{I} - \boldsymbol{A})^{-T}, \tag{3.4}$$

and so it is its inverse also known as *Precision matrix*

$$\boldsymbol{\Theta} = (\boldsymbol{I} - \boldsymbol{A})^T(\boldsymbol{I} - \boldsymbol{A}). \tag{3.5}$$

By assuming this signal model, the identity matrix $\boldsymbol{I}$ is indicating us that all nodes are sources of information and not only receivers.

## 3.3 Correlation Networks

The main goal of this chapter is to study the relations between graph's nodes. As it was exposed, it is interesting to study techniques to infer the relations for different network approaches while measuring the robustness and reliability of the technique. The first approach that is studied is the *Correlation Networks*. Graphs that represent correlation networks are undirected and reflect the coordinated behavior of two or more connected nodes. Some applications of graphs applied to these networks are, in medicine, the dimensional reduction of functional magnetic resonance imaging like in [33] or the study of genotype-dependent metabolomic clusters related to the biochemical pathway [32]. Nowadays, for constructing functional connectivity networks most software use correlation methods [34]. Another important field of application is biology [35]. As can be seen with more detail in [7], this tool is being used in the study of cancer cell metabolism.

### 3.3.1 Problem Formulation and Solution

Firstly, let us define our graph as $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ with vertices $\mathcal{V} := [1, \ldots, N]$ and edge set $\mathcal{E} := (i, j) \in \mathcal{V} \times \mathcal{V} : \rho_{ij} \neq 0$. There are different options to measure the correlation between two random nodes $x_i$, and $x_j$ being i.i.d. Nonetheless, the most adopted linear measure of similarity is the *Pearson Correlation* coefficient which is defined as

$$\rho_{ij} := \frac{\text{cov}(x_i x_j)}{\sqrt{\text{var}(x_i)\text{var}(x_j)}}. \tag{3.6}$$

It can be computed using the entries $\sigma_{ij} := \text{cov}(x_i, x_j)$ in the covariance matrix $\boldsymbol{\Sigma} := \mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T]$ of the random graph signal defined as $\boldsymbol{x} = [x_1, \ldots, x_N]^T$, whose mean vector is $\boldsymbol{\mu} := \mathbb{E}[\boldsymbol{x}]$. In order to assign weights to the connections between $x_i$ and $x_j$ there are three possible options. The correlation strength can directly be set as $W_{ij} = |\rho_{ij}|$ or, the second option would be to assign the unnormalized variant $W_{ij} = |\text{cov}(x_i x_j)|$. Finally, it exists a third option that consists on assigning weights as $W_{ij} = \mathbb{1}\{\rho_{ij} \neq 0\}$[1]. This matrix can be seen as the binarized solution matrix where entries are *0* when two nodes are not correlated or *1* when nodes are related. When working in *GSP* applications, the correlation network is considered to have a graph-shift operator $\boldsymbol{S} := \boldsymbol{\Sigma}$. From this explanation, it can be derived that the problem of identifying the graph's topology can be seen as one of inferring the subset of nonzero correlations between pairs of nodes.

Then, the way to proceed would be to compute all the correlations $\hat{\rho_{ij}}$ by replacing in (3.6) the $\text{cov}(x_i, x_j)$ by the entries $\hat{\sigma_{ij}}$ of the estimated covariance matrix $\hat{\boldsymbol{\Sigma}}$. Then, it

[1] Classic indicator function.

is necessary to define a threshold in order to start assigning nodes to the corresponding largest values of $|\hat{\rho_{ij}}|$. This threshold is usually set manually. Another interesting approach to decide if exists a relation between a pair of nodes is by testing the hypotheses

$$H_0 : \rho_{ij} = 0 \text{ versus } H_1 : \rho_{ij} \neq 0, \tag{3.7}$$

for each of the $\binom{N}{2} = \frac{N(N-1)}{2}$ candidate edges in $\mathcal{G}$. The most convenient choice as test statistic is the Fischer score [36] $z_{ij} := \tanh^{-1}(\hat{\rho_{ij}})$. The reason behind is that $H_0$, approximately, has $z_{ij} \sim \text{Normal}(0, 1/(P-3))$ see [[37], p. 210]. To have a simple form of the null distribution improves the computation of $p-$values or the threshold selection in order to guarantee a certain significance level per test. This procedure of individual testing may not be effective for medium to large sets due to its computational complexity. This problem scales as $\mathcal{O}(N^2)$. Note that for an empty graph, $\mathcal{E} = 0$, a constant probability of false alarm $P_{FA}$ will produce $\mathcal{O}(N^2 P_{FA})$ spurious edges. This number can be considerable for large $N$. In order to reduce the computational complexity and the probability of having spurious edges. A common solution to reduce the complexity is by trying to control the false discovery rate ($FDR$)

$$FDR := E\left[\frac{R_f}{R} \mid R > 0\right] Pr[R\ > 0], \tag{3.8}$$

where $R$ is the number of rejections among all $\mathcal{O}(N^2)$ edgewise tests and $R_f$ is the number of false rejections. This method has a problem when the correlation between a pair of nodes $i$, $j$ is produced by a third node $k$ that is regulating the expression of both $i$ and $j$ and not produced by a direct relation between $i$ and $j$. It is possible to avoid these relations by considering the conditional correlation

$$\rho_{ij} \mid \mathcal{V}\backslash ij := \frac{\text{cov}(x_i, x_j \mid \mathcal{V}\backslash ij)}{\sqrt{\text{var}(x_i \mid \mathcal{V}\backslash ij)\text{var}(x_j \mid \mathcal{V}\backslash ij)}}, \tag{3.9}$$

where $\mathcal{V}\backslash ij$ represents the collection of all nodes excluding those indexed by nodes $i$ and $j$. Thus, the set size is $N-2$. With this procedure it is possible to compute the correlation between $x_i$ and $x_j$ after adjusting the effects of $X_{\mathcal{V}_1}, \cdots X_{\mathcal{V}_m}$ which are common to both. Thus, to study the relation between pairs of nodes when the others are known.

## 3.4 Conditional Correlation

As stated, the previous method is not able to distinguish when a relation between nodes is due to a common origin or not. To better understand it, let us see figure 3.2. Note that in this figure the connection between nodes are not physical, as stated in 3.1 but correlation connections as in figure 3.1. In figure 3.2 nodes 2 and 3 are not directly correlated but it exists a relation between them due to both depends on node 1. Thus, there is information from 1 that is shared with 2 and 3. These relations due to a common origin, can be avoided by considering the partial correlations between nodes also known as conditional correlation (3.9).

Let us formulate and derive the problem to study the conditional covariance matrix, and why the conditional correlation is useful to estimate the precision matrix $\boldsymbol{\Theta}$ [38]. Let

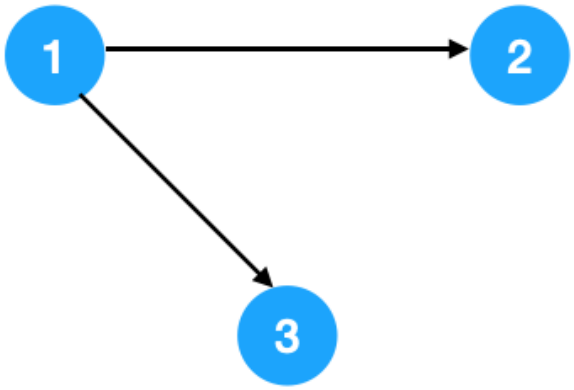


Figure 3.2: Relation between nodes 2 and 3 due to 1

us start by considering a pair of jointly Gaussian vectors $\boldsymbol{x}$, and $\boldsymbol{y}$ being i.i.d. Vector $\boldsymbol{y}$ is considered to be known. Let us construct, as composition of both, vector $\boldsymbol{z} = \left[\boldsymbol{x}^T : \boldsymbol{y}^T\right]^T$, which is also Gaussian with mean $\boldsymbol{m} = \begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix}$ and covariance matrix defined as $\boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_{xx} & \boldsymbol{\Sigma}_{xy}\\ \boldsymbol{\Sigma}_{yx} & \boldsymbol{\Sigma}_{yy}\end{bmatrix}$. Under these assumptions, $\boldsymbol{x}$ is conditionally Gaussian, this is $P_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y})$ follows a Gaussian distribution. The conditional correlation, in probability terms is written as

$$P_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{P_{\boldsymbol{xy}(\boldsymbol{x},\boldsymbol{y})}}{P_{\boldsymbol{y}(\boldsymbol{y})}}. \tag{3.10}$$

Where the numerator is

$$P_{\boldsymbol{xy}}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{(2\pi)^{\frac{l}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}}\exp\left[\frac{-1}{2}(\boldsymbol{\Sigma}_{\boldsymbol{xy}})^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xy}}\right], \tag{3.11}$$

where $\ell = \dim(\boldsymbol{x}) + \dim(\boldsymbol{y})$. Equivalently, the denominator is

$$P_{\boldsymbol{y}}(\boldsymbol{y}) = \frac{1}{(2\pi)^{\frac{k}{2}}|\boldsymbol{\Sigma}_{\boldsymbol{yy}}|^{\frac{1}{2}}}\exp\left[\frac{-1}{2}(\boldsymbol{\Sigma}_{\boldsymbol{yy}})^T\boldsymbol{\Sigma}_{yy}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{yy}}\right] \tag{3.12}$$

where $k = \dim(\boldsymbol{y})$. To analyze the expression, let us put together these equations while rewriting the covariance matrices. Thus,

$$P_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{P_{\boldsymbol{xy}(\boldsymbol{x},\boldsymbol{y})}}{P_{\boldsymbol{y}(\boldsymbol{y})}} = \frac{|\boldsymbol{\Sigma}_{yy}|^{\frac{1}{2}}\exp\left[\frac{-1}{2}\left[(\boldsymbol{x}-\hat{\boldsymbol{x}})(\boldsymbol{y}-\hat{\boldsymbol{y}})^T\right]\boldsymbol{\Sigma}^{-1}\begin{bmatrix}\boldsymbol{x}-\hat{\boldsymbol{x}}\\ \boldsymbol{y}-\hat{\boldsymbol{y}}\end{bmatrix}\right]}{(2\pi)^{\frac{N}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}\exp\left[\frac{-1}{2}(\boldsymbol{y}-\hat{\boldsymbol{y}})^T\boldsymbol{\Sigma}_{yy}^{-1}(\boldsymbol{y}-\hat{\boldsymbol{y}})\right]} \tag{3.13}$$

where $N = l - k = \dim(\boldsymbol{x})$. In order to compute it, let us start by computing the determinant of the covariance matrix by using the following relation with direct demonstration

$$\begin{bmatrix}\boldsymbol{I} & -\boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Sigma}_{\boldsymbol{yy}}^{-1}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\boldsymbol{\Sigma}\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ -\boldsymbol{\Sigma}_{\boldsymbol{xy}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{yy}}^{T} & \boldsymbol{I}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{xx}} - \boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Sigma}_{\boldsymbol{yy}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{yx}} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Sigma}_{\boldsymbol{yy}}\end{bmatrix}. \tag{3.14}$$

Finally, the determinant is easily computed. The terms of the matrix are square matrices with the same size. Thus, the determinant of the product is the product of determinants

$$\det(\boldsymbol{\Sigma}) = \det(\boldsymbol{\Sigma_{xx}} - \boldsymbol{\Sigma_{xy}}\boldsymbol{\Sigma_{yy}^{-1}}\boldsymbol{\Sigma_{yx}})\det(\boldsymbol{\Sigma_{yy}}). \tag{3.15}$$

The second step is to compute the exponential

$$\begin{aligned} \left[\boldsymbol{x} - \hat{\boldsymbol{x}}^T - \boldsymbol{y} - \hat{\boldsymbol{y}}^T\right] \boldsymbol{\Sigma}^{-1} \begin{bmatrix} \boldsymbol{x} - \hat{\boldsymbol{x}} \\ \boldsymbol{y} - \hat{\boldsymbol{y}} \end{bmatrix} = \\ (\boldsymbol{x} - \bar{\boldsymbol{x}})^T(\boldsymbol{\Sigma_{xx}} - \boldsymbol{\Sigma_{xy}}\boldsymbol{\Sigma_{yy}^{-1}}\boldsymbol{\Sigma_{yx}})^{-1}(\boldsymbol{x} - \bar{\boldsymbol{x}}) + (\boldsymbol{y} - \hat{\boldsymbol{y}})^T\boldsymbol{\Sigma_{yy}^{-1}}(\boldsymbol{y} - \hat{\boldsymbol{y}}), \end{aligned} \tag{3.16}$$

where $\bar{\boldsymbol{x}} = \hat{\boldsymbol{x}} + \boldsymbol{\Sigma_{xy}}\boldsymbol{\Sigma_{yy}^{-1}}(\boldsymbol{y} - \hat{\boldsymbol{y}})$. Finally, using this notation, we can rewrite the conditional probability as

$$P_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y}) = \frac{\exp\left[\frac{-1}{2}(\boldsymbol{x} - \bar{\boldsymbol{x}})^T(\boldsymbol{\Sigma_{xx}} - \boldsymbol{\Sigma_{xy}}\boldsymbol{\Sigma_{yy}^{-1}}\boldsymbol{\Sigma_{yx}})^{-1}(\boldsymbol{x} - \bar{\boldsymbol{x}})\right]}{(2\pi)^{\frac{N}{2}}|\boldsymbol{\Sigma_{xx}} - \boldsymbol{\Sigma_{xy}}\boldsymbol{\Sigma_{yy}^{-1}}\boldsymbol{\Sigma_{yx}}|^{\frac{1}{2}}}. \tag{3.17}$$

Thus, we have that the conditional covariance matrix is

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma_{xx}} - \boldsymbol{\Sigma_{xy}}\boldsymbol{\Sigma_{yy}^{-1}}\boldsymbol{\Sigma_{yx}} \tag{3.18}$$

In (3.18) it is seen why this concept is so important. The dependency that others nodes can produce is counteracted. Note that the conditional covariance is independent of $y$ [p.25 [38]]. Applied to our problem, the observation of the pair of nodes we want to calculate the correlation conform vector $\boldsymbol{x}$. Thus, the other nodes of the set conform vector $\boldsymbol{y}$. This result is really interesting because it means that we can calculate the correlation between pairs of nodes independently to the rest. Note, that element $(i,j)^{th}$ of the inverse of the covariance matrix, $\boldsymbol{\Theta}_{ij}$ is equal to the value of the conditional correlation between nodes $i$ and $j$.

The *canonical correlation analysis (CCA)* is another correlation measurement commonly used. The *CCA* is similar to the Pearson correlation (3.6) [39, 40]. Its objective is, given two vectors $\boldsymbol{x} = [x_1, \cdots, x_n]$, and $\boldsymbol{y} = [y_1, \cdots, y_n]$, to find the space of linear combinations that maximize the correlation. It is based on the design of projectors to modify the base in which each of the vectors is represented into a canonical base. When comparing two data sets, it is important that both are expressed in the same basis. Otherwise, some correlations may go unnoticed and others may be detected although they do not exist. The difference with the Pearson correlation is in the matrix projection, because the *CCA* is based on the direction vectors. Given the two defined vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$ let us define the linear combinations as $x = \boldsymbol{x}^T\boldsymbol{u}_i$, and $y = \boldsymbol{y}^T\boldsymbol{v}_i$. Thus,

$$\begin{aligned} \text{corr}(x,y)_i &= \frac{\text{cov}(x,y)}{\sqrt{\text{var}(x)\text{var}(y)}}, \\ \text{corr}(x,y)_i &= \frac{\boldsymbol{u}_i^T\text{cov}(\boldsymbol{x},\boldsymbol{y})\boldsymbol{v}_i}{\sqrt{\boldsymbol{u}_i^T\text{var}(\boldsymbol{x})\boldsymbol{u}_i\boldsymbol{v}_i^T\text{var}(\boldsymbol{y})\boldsymbol{v}_i}}. \end{aligned} \tag{3.19}$$

As can be observed, this equation is really similar to (3.6). The maximum canonical correlation is the $\text{corr}(x,y)|_{max}$ with respect to the direction vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ [41, 42].

Nonetheless, it was seen that this correlation is no able to detect those relations produced due to a dependency with a third vector. The analysis can be modified to include the conditional covariance, in order to consider the possible effect produced by a third vector. Let us consider the conditional covariance applied in expression (3.19) [43]. Hence,

$$\text{corr}(x, y) = \frac{\boldsymbol{u}_i \text{cov}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{z}) \boldsymbol{v}_i}{\sqrt{\boldsymbol{u}_i^T \text{var}(\boldsymbol{x} \mid \boldsymbol{z}) \boldsymbol{u}_i \boldsymbol{v}_i^T \text{var}(\boldsymbol{y} \mid \boldsymbol{z}) \boldsymbol{v}_i}}. \tag{3.20}$$

where $\boldsymbol{z}$ contains the set of vertices $\mathcal{V} \backslash ij$. As it can be seen, this last expression is equivalent to (3.9), but the projection is based on the linear transformation $\boldsymbol{u}_i$, and $\boldsymbol{v}_i$. Thus, there is a near relation between the *CCA* and the conditional correlation method.

Let us continue the study, assuming that the graph signal $\boldsymbol{x} = [x_1 \ \cdots \ x_N]^T$ is a Gaussian real random vector [44]. Under this assumption, it is possible to reformulate (3.9) to consider that the graph signals are Gaussian. Let us define $\boldsymbol{W}_1 = (x_i, x_j)^T$ containing the graph signal of the pair of nodes $i$ and $j$ under study. Also, let us define $\boldsymbol{W}_2 = \boldsymbol{x}_{\mathcal{V} \backslash ij}$ containing the information of the rest of nodes that compose the vertices set. Then the covariance matrix can be expressed as

$$\text{cov} \begin{bmatrix} \boldsymbol{W}_1 \\ \boldsymbol{W}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix}, \tag{3.21}$$

Considering that $(x_i, x_j, x_{\mathcal{V}_1}, ..., x_{\mathcal{V}_m})^T$ follows a multivariate Gaussian distribution, then $\rho_{ij|\mathcal{V} \backslash ij} = 0$, if and only if, $x_i$ and $x_j$ are independent given all the other variables in $\mathcal{V} \backslash ij$. Note that for the particular case of $m = 0$, the partial correlation reduces to the Pearson correlation. From this, it can be seen that the conditional correlation specifies conditional independence relations between the entries as studied in 3.4. This is known as *Undirected Gaussian Graphical Models* or *Gaussian Markov Random Field (GMRF)* [45, 46, 47]. The partial correlation coefficients from (3.9) can be expressed as

$$\rho_{ij|\mathcal{V} \backslash ij} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii} \theta_{jj}}}, \tag{3.22}$$

where the element $\theta_{ij}$ is the $(i, j)^{th}$ entry of the precision matrix $\boldsymbol{\Theta} := \boldsymbol{\Sigma}^{-1}$. Thus, each element $\theta_{ij}$ from the precision matrix that equals 0 indicates that the pair of nodes $(x_i, x_j)$ are independent given all the other graph's nodes. This means that those nodes $(x_l, x_m)$ that are related due to a third node, $x_n$, are represented in the precision matrix, also, as $\theta_{lm} = 0$, as seen in 3.4. This is a really powerful idea that will allow us to decide whether two nodes have a relation or not only looking to the precision matrix. This problem of identifying the conditional independence relations when having identical distributed realizations from a multivariate Gaussian distribution is really known and a lot of literature can be found, [48, 49, 50, 51, 52]. It is called *Covariance Selection* [48].

### 3.4.1 Beyond Equations

In order to validate the algorithms and the software, some controlled simulations are performed. These examples are done by generating a fix $\boldsymbol{A}$. Multiple simulations are performed in order to compute the mean value between all of them to better see how the

algorithms behave. In each iteration a random value between $[-0.25, 0.25]$ is added to the relations in matrix $\boldsymbol{A}$. This means that better and worse scenarios are considered and the maximum possible value for a relation, as said in 3.2, is $a_{ij} \leq 1$.

The relations between nodes, in the first simulation case, is the following

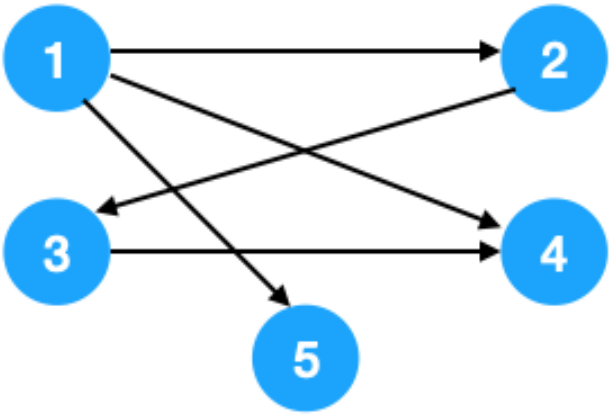


Figure 3.3: Information flow, first graph example

where the relations have the following central values

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 & 0 \\ 1 & 0 & 0.5 & 0 & 0 \\ 0.75 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.23}$$

This first simulation is performed by using $10N$ snapshots. This is a stable case where enough snapshots are available to correctly estimate the covariance matrix. Our goal is to obtain a matrix where pairs of nodes which are related are clearly distinguishable from those which are not. The expected relations, in this first scenario, are

$$\hat{\boldsymbol{\Theta}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.24}$$

where the *1*s represent an existing relation between nodes and *0*s are non-related pairs of nodes. In this expected $\hat{\boldsymbol{\Theta}}$, values which are different from *0* in the $\boldsymbol{A}$ matrix are *1*s. Also, $\text{diag}(\hat{\boldsymbol{\Theta}}) = \mathbf{1}$ because each node has a relation with itself and it is represented in the precision matrix as a *1*. In this particular case, we also expect that the position *(1,3)* and, knowing that the precision matrix is the inverse of the covariance matrix, its symmetric to be *1*. As explained in section 3.3, although it does not exist a direct relation between nodes 1 and 3, they actually have a relation due to the fact that node 4 is not able to know

from which of those two nodes its information comes from. This produces that nodes 1 and 3 are related.

After applying this technique, results can be seen in figure 3.4. The values are represented in dB to have a better contrast between the values which represent a relation and those which are zero. By doing the simulation it can be seen that the expected result was correct. All values which are different from *0* in matrix $\boldsymbol{A}$ are represented as a relation. Also, the relation between nodes 1 and 3 can be seen in figure 3.4. In this case we can see

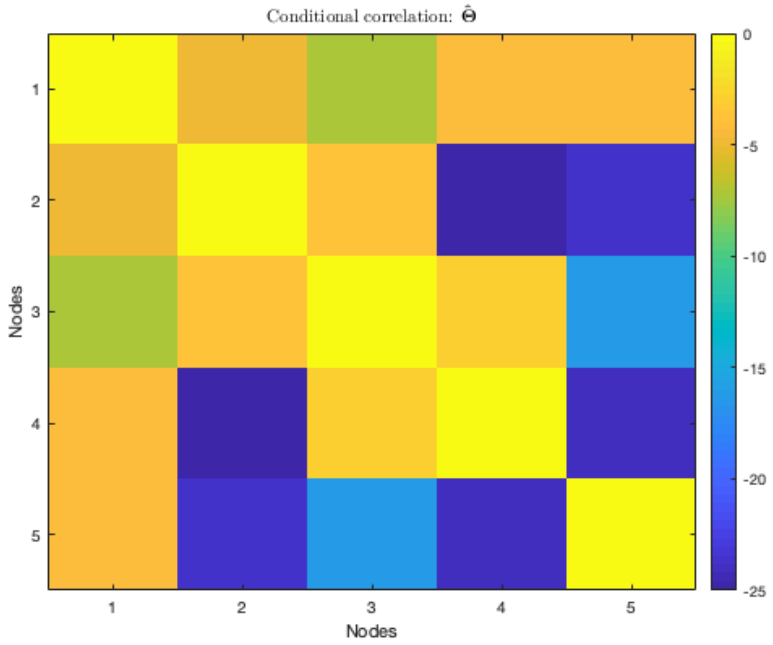


Figure 3.4: Estimated $\hat{\boldsymbol{\Theta}}$ by using the conditional correlation method for $10N$ snapshots

that the technique worked well. All values are correctly estimated, those which represent a relation and those which are no-relations. This is not a critical scenario because enough snapshots were available to correctly estimate the covariance matrix. Next algorithms are also tested for the non-stable case of having $N-1$ snapshots. It makes no sense to test this method under these conditions because the covariance matrix is not well conditioned, thus is a singular matrix.

Let us know study another interesting example. In this second simulation the graph's relations are represented in figure 3.5. The relations have the following central values

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0.5 & -1 & 0 \\ -1 & 0 & 0.75 & 0 & 0 \\ -1 & 0.75 & 0 & 0 & 0 \\ 0 & 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{3.25}$$

This simulation is performed, as in the first example, by using $10N$ snapshots. This is a stable case because enough information is available to estimate the covariance matrix.

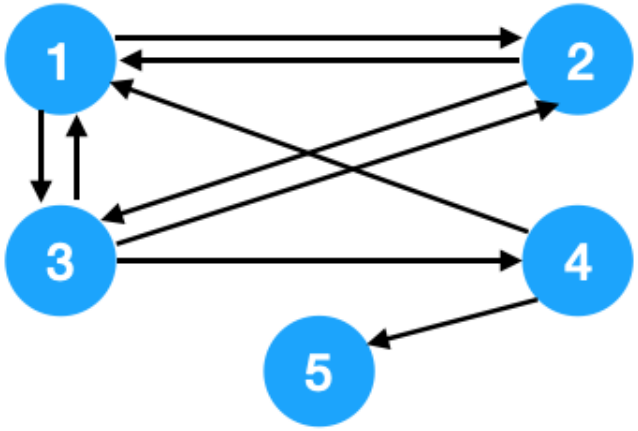


Figure 3.5: Information flow, second graph

As in the previous case, let us analyze which are the expected relations

$$\hat{\Theta} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{3.26}$$

With this simulation we want to see if in case of having a relation between 1 and 2, so node 1 knows the information from node 2, 1 is able to distinguish between nodes 2 and 4. The expected result, as can be seen in (3.26) is that although 1 has this knowledge it exists a relation between 2 and 4 because 1 will not be able to distinguish from whom the information comes from. The results can be seen in figure 3.6. As can be seen, the zero values are clearly represented in figure 3.6. By doing this simulation, our objective was to better understand the nodes' relations, as well as, to verify that the algorithm is properly implemented. This simulation has produced the expected result where the pairs *(5,1),(5,2),(5,3)* are not dependent between them. A strong dependency between nodes 4 and 5 can be observed. For sure, this relation is due to the fact that node number 5 is only influenced by node 4. Thus, 5 has a total dependency on 4 because all its information comes from this node.

## 3.5 Maximum Likelihood Estimator with Sparse Regularization

When talking about graph signals, it is common that they follow a Gaussian distribution. In this section, we study the case of having a graph signal which is a Gaussian random vector. This type of signals can be found in different fields such as model predictive control and analysis systems [53], image analysis and synthesis [54] or in speech processing [55]. Gaussian processes are also commonly found in communications systems [56]. With this type of signals found in so many fields, it is interesting to study this particular case The

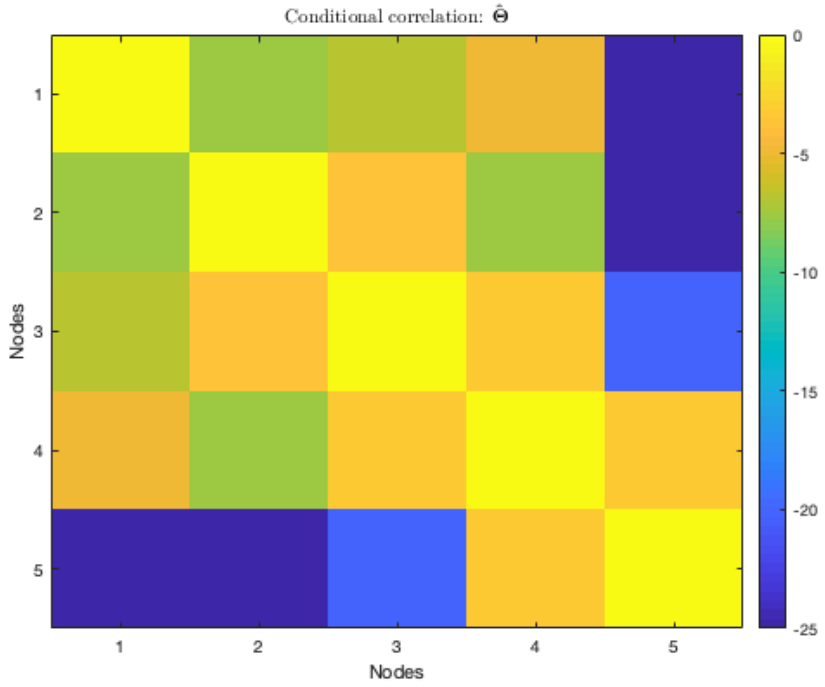


Figure 3.6: Estimated $\hat{\boldsymbol{\Theta}}$ by using the conditional correlation method for $10N$ snapshots

study of the conditional correlation when working with these signals was explained in 3.4 section.

### 3.5.1 Problem Formulation and Solution

The covariance selection term was firstly used by Demptser in the early 1970s [48] and he proposed an approach to solve this problem. He presented the possibility of estimating the entries of $\boldsymbol{\Theta}$ via a recursive, likelihood-based threshold procedure. It worked well for the type of networks that existed in this period. Nonetheless, this algorithm does not scale well when applied to nowadays large-scale networks. The reason behind is that when working with high-dimensional regimes the estimation of the $\hat{\boldsymbol{\Sigma}}$ is rank deficient. A good way to solve the problem related with the $\hat{\boldsymbol{\Sigma}}$ rank is by means of a regularization. It is really common to consider that the network has a high degree of sparsity. In [57] it is explained why the sparsest solution to a problem is useful in different fields. In order to take into account the sparsity, the $\ell_0$-*norm* must be used. Nonetheless, due to its computational complexity as the search space is exponentially large, [58], it is used the $\ell_1$-*norm* which is the convex function closest to the $\ell_0$-*norm* [13]. A most extensive explanation about this topic can be found in [59]. In our particular case, the regularization is the $\ell_1$-*norm*. In order to simplify the equations and without loss of generality, we assume zero-mean $\boldsymbol{x} \sim \text{Normal}(\mathbf{0}, \boldsymbol{\Sigma})$ since the objective is estimating graph's structure encoded in the entries of the precision matrix $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$. Under these assumptions the maximum-likelihood ($ML$) estimator of the precision matrix is given by

$$\hat{\boldsymbol{\Theta}}_{ML} = \arg\max_{\boldsymbol{\theta} \geq \mathbf{0}} \{\log\det \boldsymbol{\Theta} - \text{trace}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})\}, \tag{3.27}$$

where $\boldsymbol{\theta} \geq \mathbf{0}$ is a requirement to ensure that the matrix is positive semi-definite and $\hat{\boldsymbol{\Sigma}} = \frac{1}{P}\sum_{p=1}^{P} \boldsymbol{x}_p \boldsymbol{x}_p^T$ is the empirical covariance matrix. It can be seen, that when $N >> P$ matrix $\hat{\boldsymbol{\Sigma}}$ is singular, expression (3.27) is not the *ML* estimator and, in fact, it does not exist. As explained, the regularization called *Graphical Lasso* [60] is applied. Nonetheless, we are interested in a sparse solution. Thus, in order to promote sparsity the $\ell_1$-*norm* is used [61], obtaining

$$\hat{\boldsymbol{\Theta}}_{ML} = \max_{\boldsymbol{\theta}\geq 0} \{\log\det \boldsymbol{\Theta} - \text{trace}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta}) - \lambda \parallel \boldsymbol{\Theta} \parallel_1\}. \tag{3.28}$$

The computational complexity of these methods, called *Interior-Point Methods* (also known as *barrier methods* or *IBM* , is $O(N^6)$. Furthermore, interior-point methods require computing and storing, in each of the iterations, a Hessian matrix of size $O(N^2)$. These methods are algorithms that are used to solve linear and nonlinear convex optimization problems and there are multiple different algorithms. A further and more detailed explanation can be found in [62, 63]. Some examples where these methods are applied are in power generation [64, 65] and in designing recursive maximum likelihood methods to estimate some parameters for hypersonic vehicles [66].

This means that for large networks the computational complexity becomes an issue in terms of time and storage. Nonetheless, this method works much better than the conditional correlation because it is able to work with much less snapshots. By using the appropriate value of the sparsity promoting parameter $\lambda$ it is possible to improve a lot the matrix estimation.

### 3.5.2 Beyond Equations

The relations between nodes in the first simulation case are the same as in figure3.3 with the values represented in equation (3.23) and the expected result, as in the other cases is 3.24.

This first simulation, where enough snapshots are available, is used as test to ensure that the algorithm works as expected. Once the algorithm is validated, another simulation is performed. In this second case, only $N-1$ snapshots are available. This is a more critical and extreme scenario that allows us to see the potential of this technique. As can be seen in figure 3.8, although the difference between the related and the non-related pairs of nodes is less significant than in figure 3.7, it is still possible to distinguish those nodes which are not related. This means that with this algorithm it is possible to recover the graph's relations although the scenario is not totally well conditioned, or in other words not enough snapshots are available. Nonetheless, it is important to note that this technique is much more computationally complex than the next algorithm studied the *Sparse Linear Regression*. Then, the next case is also studied with this second algorithm, figure 3.9. As can be seen all values can be perfectly recovered with this second technique. By improving the value of the normalization parameter $\lambda$ it is possible to improve the estimated result. With this simulation, it can be seen that the algorithm works for different scenarios without problems. By performing this simulation it is possible to validate that the algorithm is correctly implemented. Note that those pairs of nodes which are directly related in 3.26 have a stronger relation than those which are related because a third node is not able to distinguish from whom its information comes from. An example of a direct relation is pair *(5,4)* and an example of those lower relations is *(4,1)*.

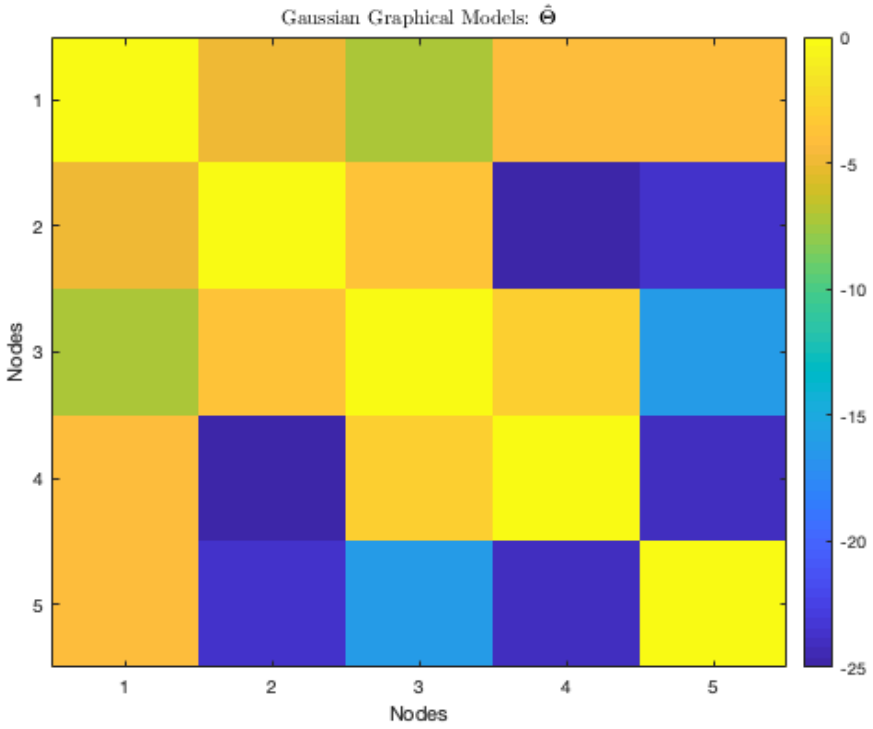


Figure 3.7: Estimated $\hat{\mathbf{\Theta}}$ by using the *ML* estimator promoting sparsity for $10N$ snapshots

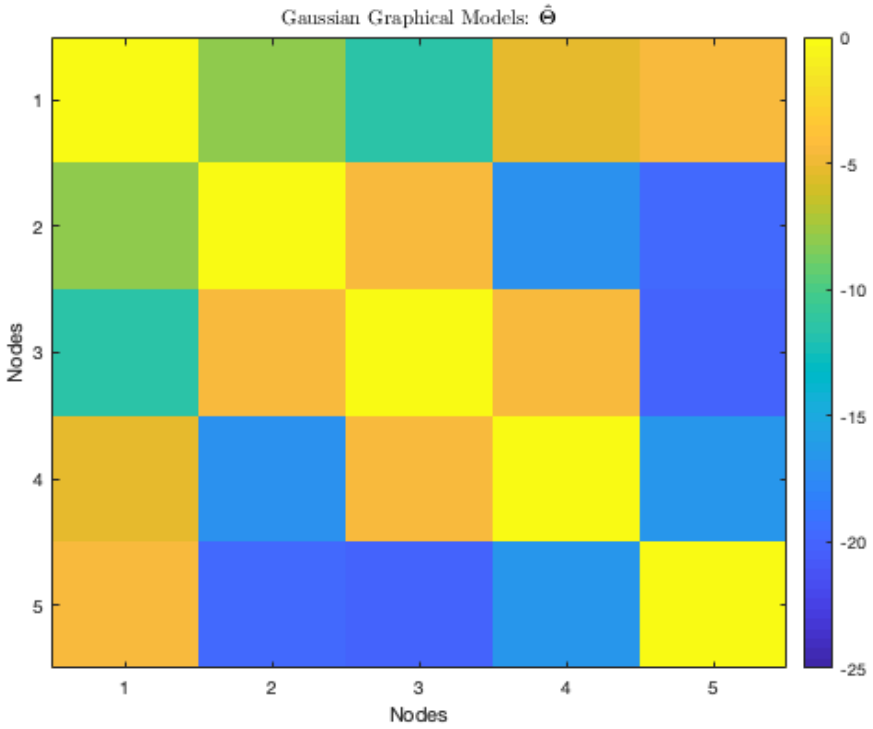


Figure 3.8: Estimated $\hat{\mathbf{\Theta}}$ by using the *ML* estimator promoting sparsity for $N-1$

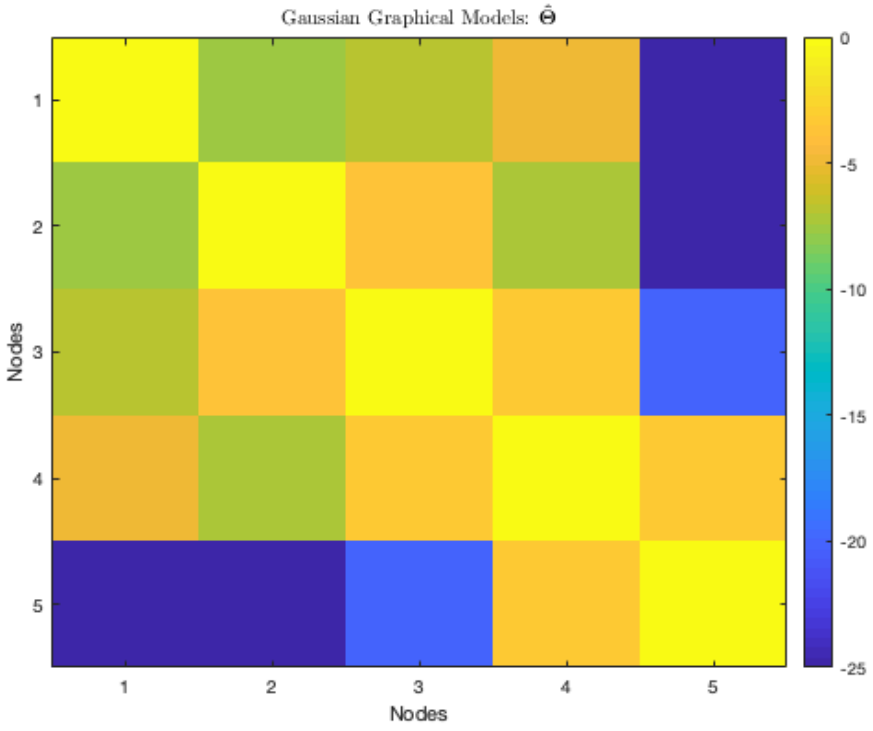


Figure 3.9: Estimated $\hat{\mathbf{\Theta}}$ by using the *ML* estimator promoting sparsity for $10N$ snapshots

## 3.6 Sparse Linear Regression

As seen in previous section 3.5.1 the $\ell_1$-*norm* regularization is really useful but it would be interesting to reduce the computational complexity. It is also necessary to find an algorithm that works for lower number of snapshots, this is when the covariance matrix is rank deficient as explained in the last section 3.5.1. In those scenarios it is common to follow the approach of finding the minimum mean-square error.

### 3.6.1 Problem Formulation and Solution

Another way to set weights to the edges, is to find the minimum mean-square error predictor of the snapshot in node $i$. Firstly, let us start defining the observations of the signal graph as $\boldsymbol{x}_p$. Using this information, let us found the minimum mean-square error predictor for every node $i$. The predictor for node $i$ is defined as $\boldsymbol{\gamma}_i$. The error at node $i$ is defined as $\epsilon_i = \boldsymbol{\gamma}_i^T \boldsymbol{x}$. The objective is to minimize

$$\boldsymbol{\gamma}_i = \arg\min_{\epsilon_i} \sum_{p=1}^{P} |\epsilon_i|^2 \text{ s.t } \boldsymbol{\gamma}_i^T \boldsymbol{e_i} = 1, \tag{3.29}$$

where the restriction is forcing the element $i$, of the predictor of node $i$, to be 1. Vector $\boldsymbol{e}_i$ has all its entries set to 0 but entry $i$ set to 1. At this point, it is important to remember that, during all the thesis we are interested in searching the sparsest solution to the problem. Thus, it is necessary to include the regularization parameter based on the $\ell_1$-*norm*. Let us define the final predictor with the sparsity promotion

$$\boldsymbol{\gamma}_i = \arg\min_{\gamma_i} \sum_{p=1}^{P} |\boldsymbol{\gamma}_i^T \boldsymbol{x}_p|^2 + \lambda_i \|\boldsymbol{\gamma}_i\|_1 \text{ s.t } \boldsymbol{\gamma}_i^T \boldsymbol{e}_i = 1, \tag{3.30}$$

where $\boldsymbol{x}_p$ is the graph signal in all nodes at snapshot $p$. The design factor $\lambda_i$ is really critical, higher it is the more sparse the solution will be because the algorithm gives more importance to the $\ell_1$-*norm*. Note that for each predictor, this is for each node, $\lambda_i$ is not necessarily the same. It is important to note that the predictor is computing the values in each node, this is, predicting column $i$ of the conditional covariance matrix. An important remark is that if the sparsity promotion parameter $\lambda_i$, is set to 0, the optimal $\boldsymbol{\gamma}_i|_{opt}$ would be

$$\boldsymbol{\gamma}_i|_{opt} = \frac{\boldsymbol{\Sigma}^{-1}\boldsymbol{e}_i}{\boldsymbol{e}_i^T \boldsymbol{\Sigma}^{-1} \boldsymbol{e}_i}. \tag{3.31}$$

Note that the predictor will estimate the inverse of column $i$ of the covariance matrix multiplied by a scaling factor. This result is the same that the conditional correlation provides. Thus, if the design factor $\lambda_i = 0$ the solution obtained with this method and the conditional correlation are equivalent. In chapter 4 there is an example that illustrates this concept.

As exposed, each of these $\boldsymbol{\gamma}_i$ has its own scaling factor. Once all the matrix is conformed by each of the columns, it is necessary to decide for each entry *(i,j)* whether it is a zero or not. In other words, if it exists a relation between nodes $i$ and $j$. In order to fairly compare values, it is necessary to take into account the scaling factor that each column has. Columns need to be multiplied by the scaling factor that is the square error.

Noting that, the error vector is defined as $\epsilon_i = \boldsymbol{\gamma}_i^T \boldsymbol{x}$. Let us define the covariance matrix as $\boldsymbol{\Sigma} := \mathbb{E}\left[\boldsymbol{x}(n)\boldsymbol{x}(n)^T\right]$ and $\boldsymbol{\gamma}_i|_{opt}$ as (3.31). Thus, the minimum square error is defined as

$$\xi_{min}^{2^{(i)}} = \mathbb{E}\left[|\boldsymbol{\epsilon}_i(n)|^2\right]\Big|_{\boldsymbol{\gamma}_i=\boldsymbol{\gamma}_i|_{opt}}. \tag{3.32}$$

Let us develop the expression

$$\xi_{min}^{2^{(i)}} = \mathbb{E}\left[\boldsymbol{\gamma}_i^T \boldsymbol{x}(n)\boldsymbol{x}(n)^T \boldsymbol{\gamma}_i\right]\Big|_{\boldsymbol{\gamma}_i=\boldsymbol{\gamma}_i|_{opt}}. \tag{3.33}$$

The predictor is deterministic. Thus, it is extracted from the mathematical expectation operator. Then, it is possible to write the minimum square error in terms of the covariance matrix and the optimal $\boldsymbol{\gamma}_i|_{opt}$

$$\xi_{min}^{2^{(i)}} = \boldsymbol{\gamma}_i^T \mathbb{E}\left[\boldsymbol{x}(n)\boldsymbol{x}(n)^T\right]\boldsymbol{\gamma}_i\Big|_{\boldsymbol{\gamma}_i=\boldsymbol{\gamma}_i|_{opt}} = (\boldsymbol{\gamma}_i^T \boldsymbol{\Sigma}\boldsymbol{\gamma}_i)\Big|_{\boldsymbol{\gamma}_i|_{opt}}. \tag{3.34}$$

Finally, substituting the value of $\boldsymbol{\gamma}_i|_{opt}$ by (3.31) the value obtained is

$$\xi_{min}^{2^{(i)}} = \frac{\boldsymbol{e}_i^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1}\boldsymbol{e}_i}{(\boldsymbol{e}_i^T \boldsymbol{\Sigma}^{-1}\boldsymbol{e}_i)^2} = \frac{1}{(\boldsymbol{e}_i^T \boldsymbol{\Sigma}^{-1}\boldsymbol{e}_i)} = \boldsymbol{\Theta}_{ii}^{-1}. \tag{3.35}$$

Then, in order to normalize the predictor by multiplying the corresponding scaling factor it is necessary to multiply by the solution obtained in (3.35). Thus,

$$\boldsymbol{\beta}_i = \boldsymbol{\gamma}_i|_{opt}\, \xi_{min}^2, \tag{3.36}$$

where $\xi_{min}^2 = \frac{1}{(\boldsymbol{e}_i^T \boldsymbol{\Sigma}^{-1}\boldsymbol{e}_i)} = \boldsymbol{\Theta}_{ii}^{-1}$. To finish, once the normalization is done for all predictors, the estimated precision matrix $\hat{\boldsymbol{\Theta}}$ is build with column $i$ equal to predictor $\boldsymbol{\beta}_i$.

### 3.6.2 Beyond Equations

The relations between nodes in the first simulation case are the same as in figure 3.3 with the values represented in equation (3.23) and the expected result, as in the other cases will be (3.24).

By selecting an appropriate value for $\lambda$, in this case $\lambda = 10^{-3}$, and by applying the columns normalization explained in this section, it is possible to recover perfectly the nodes' relations. It is important to note that the selection of the regularization parameter $\lambda$ is critical to obtain a reliable result. In order to find the optimal value in this thesis, the mean square error produced was monitored. For a given sparsity, 30% of non-zero values, the optimum value for $\lambda$ has been found for different number of snapshots. Then a linear regression has been performed in order to find the expressions that best allows us to find the optimal parameter $\lambda$, see figure 3.10. This method is not based on a close expression to find the optimal value for any case, but this is beyond the scope of this project. In figure 3.12 it can be seen that this method is quite reliable even in the worst scenarios. By doing the second simulation figure 3.13, we can ensure that the algorithm is correctly implemented. The result in this case is really similar to the obtained with the other algorithms because this example is performed in all cases with a well conditioned case were the covariance matrix is perfectly estimated. Thus, the $\ell_1$-*norm* is not significantly important.

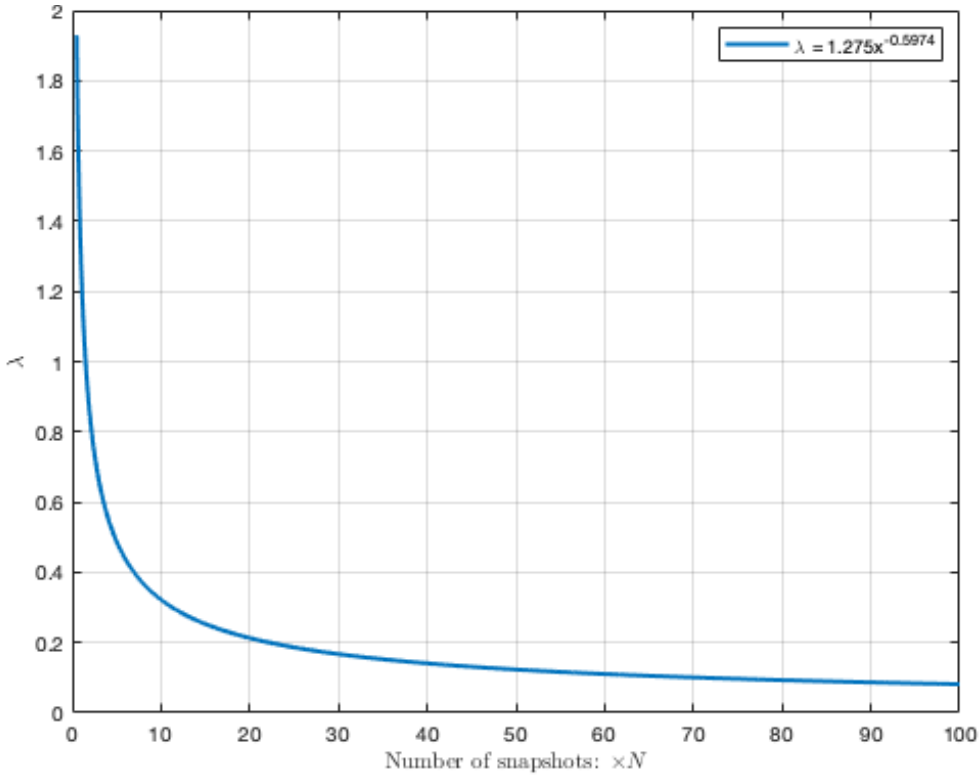


Figure 3.10: $\lambda_{opt}$ for a 30% of non-zero values and different number of snapshots

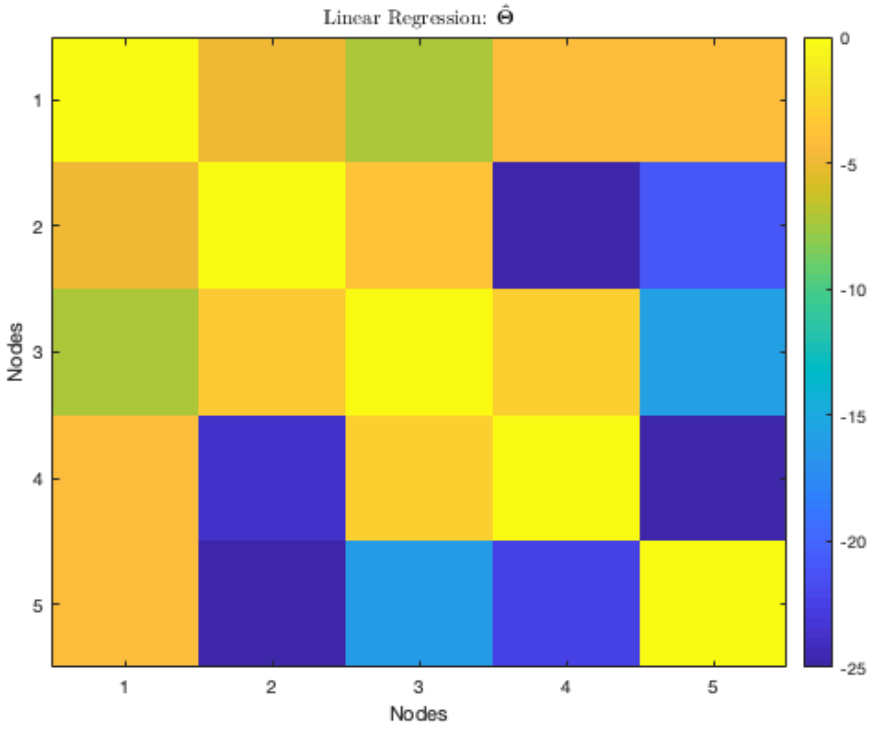


Figure 3.11: Estimated $\hat{\Theta}$ by using the linear regression method for $10N$ snapshots

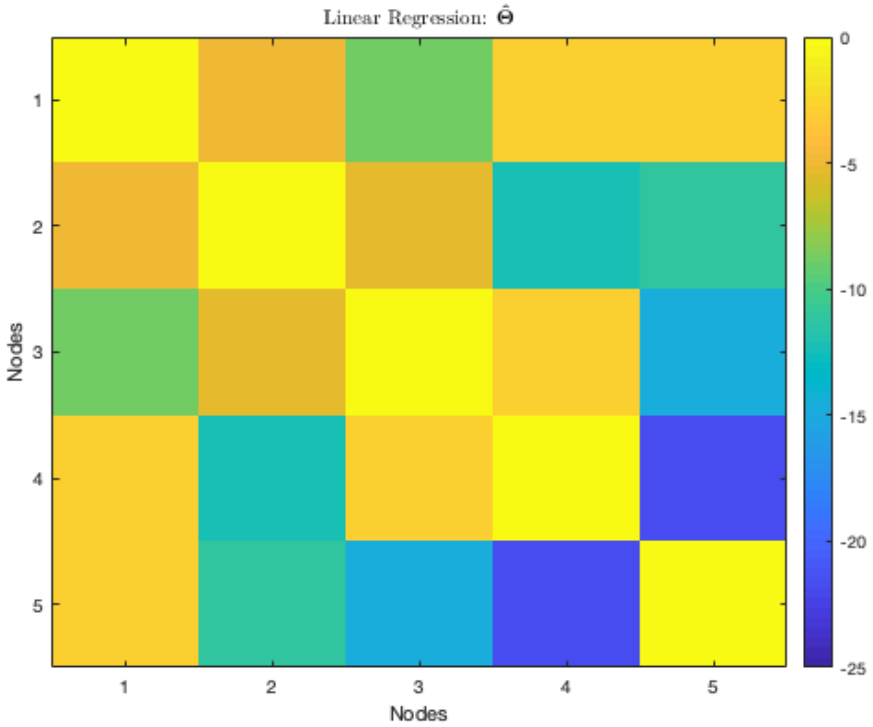


Figure 3.12: Estimated $\hat{\mathbf{\Theta}}$ by using the linear regression method for $N-1$ snapshots

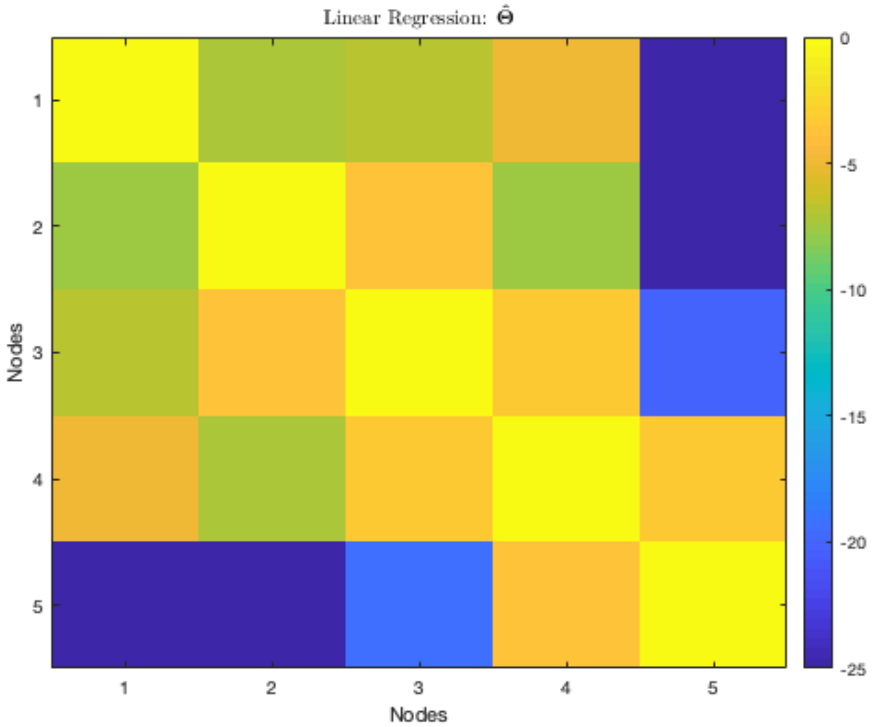


Figure 3.13: Estimated $\hat{\mathbf{\Theta}}$ by using the linear regression method for $10N$ snapshots

# Chapter 4

# Comparison of Methods

## 4.1 Introduction

In the previous chapter, three different methods to infer the graphs' topology were studied. These techniques were based on searching the precision matrix $\mathbf{\Theta}$. In this chapter, the objective is to compare these three approaches by generating random graphs. As it was explained during the thesis, the differences between them are more significant when they try to estimate a sparse matrix, and when the number of nodes increases. The conditional correlation method does not take into account the sparsity of the solution while the other two methods consider it. In order to easily analyze the results while increasing the number of nodes, in this chapter algorithms work with $N = 10$. This number is selected because it allows us to easily compare the solutions obtained while increasing the number of nodes. The same figures as in the previous chapter are used to compare the methods performance. In order to be in control of the matrix's sparsity, the precision matrix, $\mathbf{\Theta}$, is synthetically generated. In these simulations, it is not possible to represent the graph because different graph structures can produce the same precision matrix, see figure 4.1. This is a simple example, but when the number of graph's nodes increases, the number of possible topology combinations that can be represented with the same precision matrix increments. Thus, considering that these techniques try to estimate the precision matrix, in order to compare them, the generated precision matrix is represented.

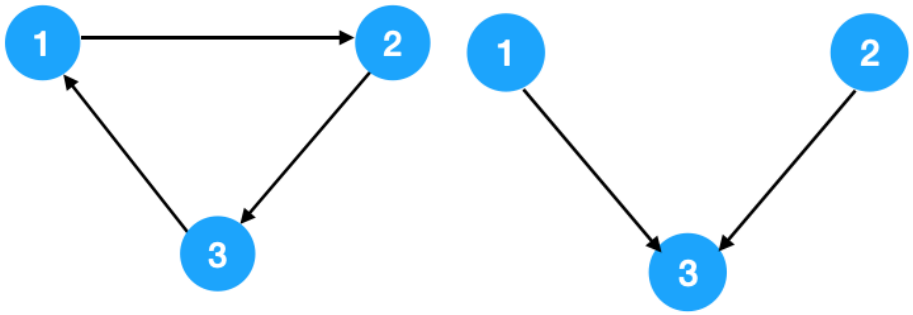


Figure 4.1: Different graph structures that generates the same precision matrix $\mathbf{\Theta}$

## 4.2 Simulation Results

Firstly, we start by analyzing the well conditioned case. For $10N$ snapshots and a precision matrix $\hat{\mathbf{\Theta}}$ with a percentage of values different to 0 of the 30%. The precision matrix generated can be seen in figure 4.2. In this precision matrix, there are three different

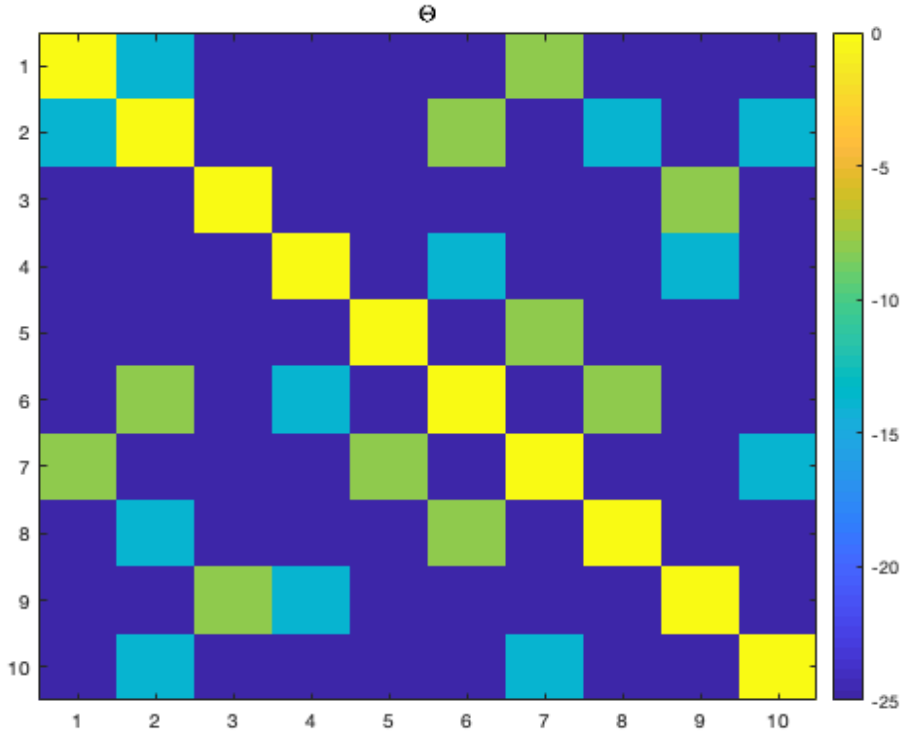


Figure 4.2: $\mathbf{\Theta}$ generated with a percentage of values different to 0 of 30%

values. The matrix is generated where $\text{diag}(\mathbf{\Theta}) = \mathbf{1}$. Outside the diagonal values are 0 or a fixed high value (i.e *0.8*), represented as a green color in figure 4.2 or a fixed low value (i.e *0.2*) which is represented in blue color. Matrix in figure 4.2 is the one that the algorithm should estimate. Now, let us focus in the performance of the three different methods starting by the conditional correlation in figure 4.3. On one hand, as can be seen the high values are correctly estimated. On the other hand, the lower values are also correctly estimated. Nonetheless, the difference between some of these values and the non-relations, with weight *0*, is not clear enough. This is the case for the pair *(5,7)*, which is a not relation and is really close to the value of the pair *(5,6)*, which is a relation. By using the *ML* estimator or the Linear Regression methods which take into account the optimization of the $\ell_1$-*norm* it is expected to obtain a more sparse result.

As can be seen in figure 4.4 and figure 4.5, respectively, a more sparse matrix is obtained. When using the *ML* estimator method the high values are perfectly estimate. Low values are mostly represented but some are missing. If the normalization parameter $\lambda$ is reduced, some false relations might appear. If more data is available, it can be used to improve the estimation. Thus, low values can be correctly estimated but more snapshots

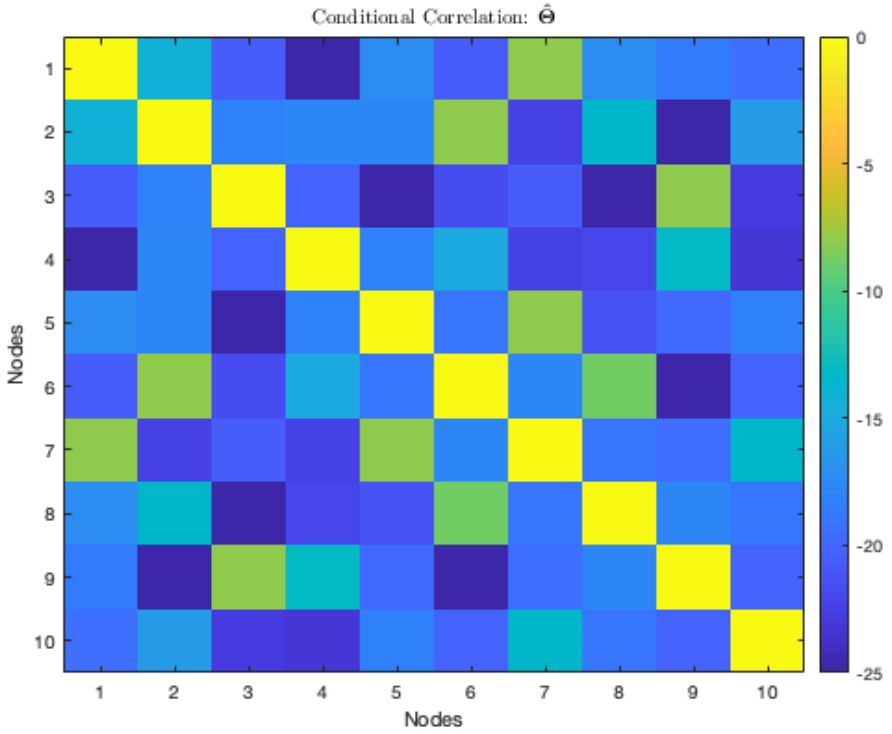


Figure 4.3: Estimated $\hat{\mathbf{\Theta}}$ by using the conditional correlation method

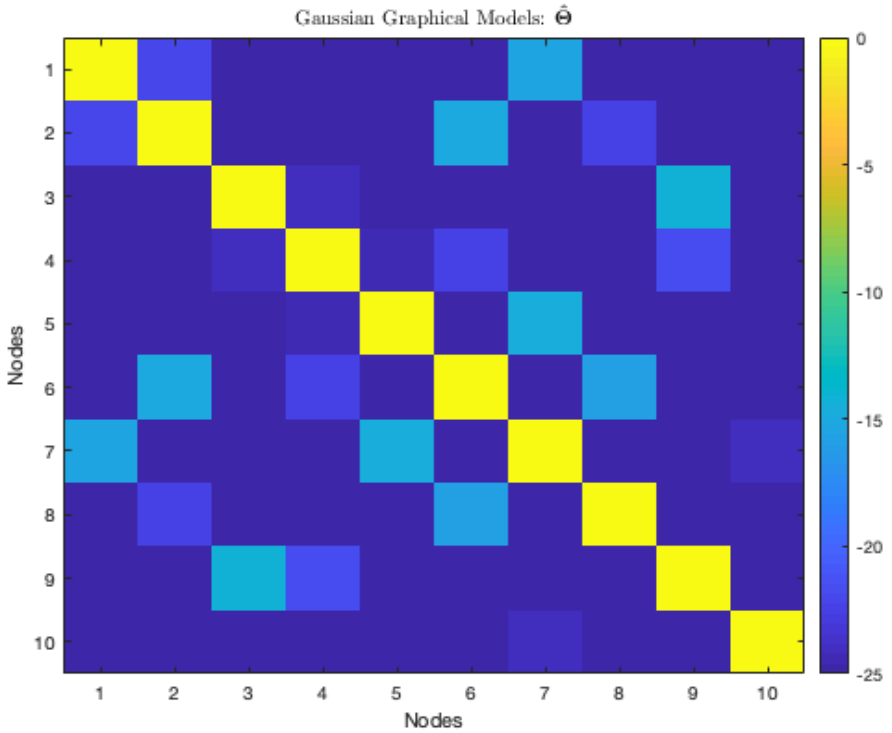


Figure 4.4: Estimated $\hat{\mathbf{\Theta}}$ by using the *ML* estimator

are needed. Using the Linear Regression approach, the result obtained in this case is really similar to figure 4.4. Nonetheless, it is interesting to note that it worked slightly better than the *ML* estimator approach and the computational time is around 10× faster in the case of the Linear Regression method.

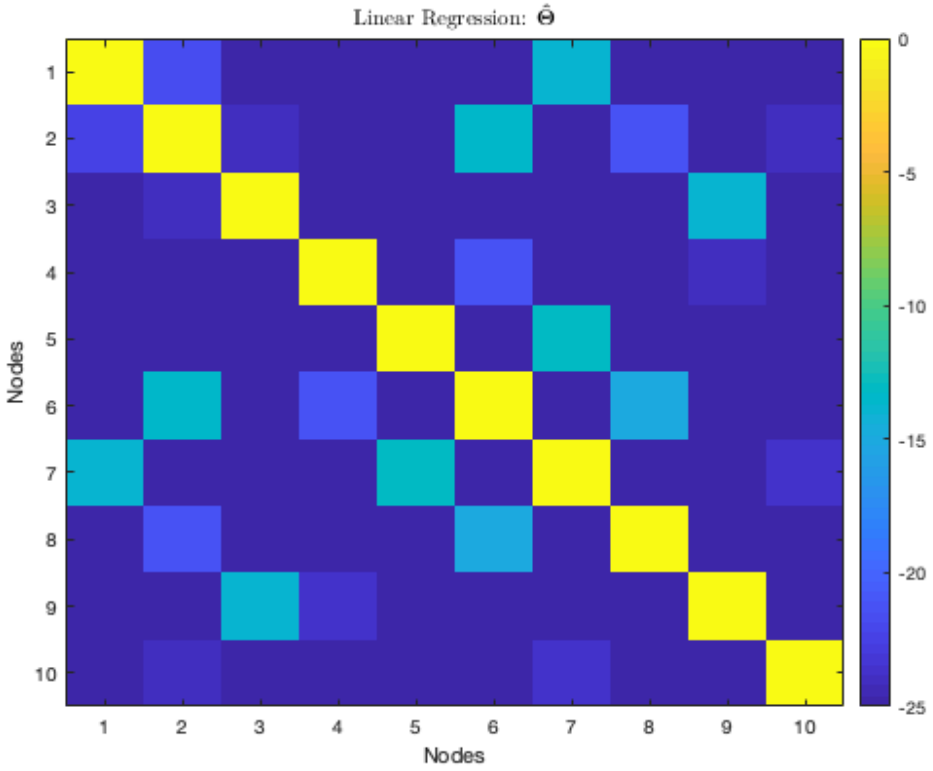


Figure 4.5: Estimated $\hat{\Theta}$ by using the Linear Regression method

To conclude this first simulation, let us analyze figure 4.6. It can be seen the result of applying the Linear Regression method with a normalization parameter equal to 0. As expected, the result obtained by the Linear Regression method when no normalization parameter is taken into account is identical to the conditional correlation solution, as explained in 3.6. By comparing figure 4.6 with figure 4.5 it is possible to see the benefits of using the optimization of the $\ell_1$-*norm*. It represents more clearly the non-relations. To conclude, we can say that by providing this extra information, the algorithm is able to better differentiate those values which are zero from those which are relations between pairs of nodes.

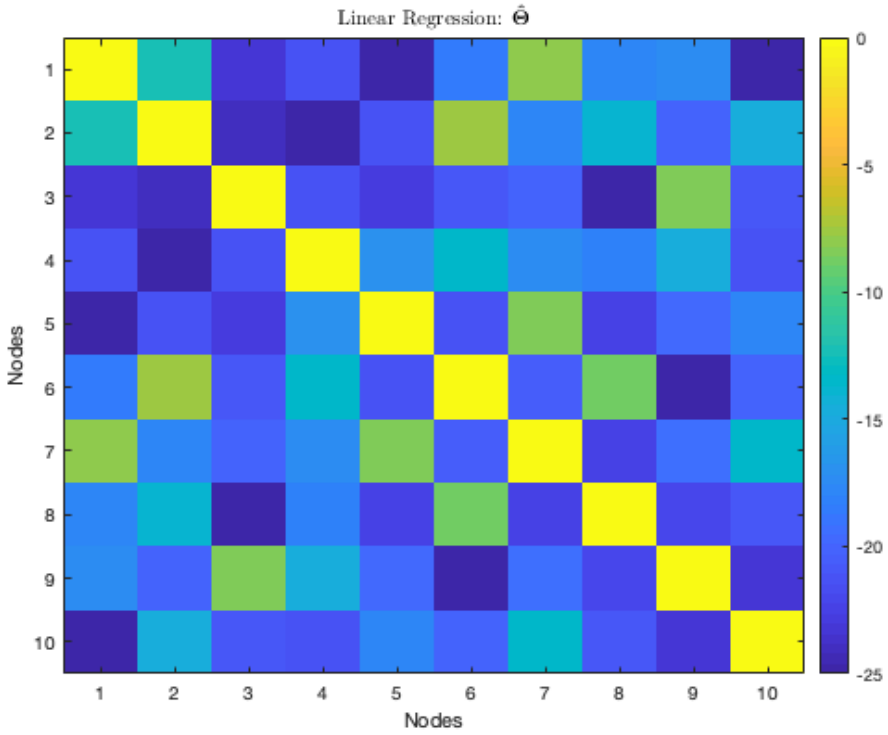


Figure 4.6: Estimated $\hat{\boldsymbol{\Theta}}$ by using the Linear Regression method with no normalization parameter

In this next example figure 4.7 a new precision matrix is generated. Nonetheless, in this case algorithms only have $N-1$ snapshots to estimate it. In this scenario, as explained in 3.3 it makes no sense to try to use the conditional correlation method because the covariance matrix is rank deficient. Thus, it is not possible to compute the precision matrix as the inverse of the covariance matrix.

The precision matrix, $\boldsymbol{\Theta}$ that is used in this simulation can be seen in figure 4.7. As can be seen, nodes 9 and 10 are isolated, this is they do not depend on any other node. Let us now focus in the results obtained by the two algorithms that incorporates the $\ell_1$-*norm* optimization, the sparsity promotion parameter $\lambda$. By looking into figure 4.8 it can be seen that some of the most important relations are estimated but any of the lightest relations is represented. A similar result can be seen in the estimation obtained by the linear regression method in figure 4.9.

This result means that for bad scenarios where the snapshots available are not enough to correctly estimate the covariance matrix, it is still possible to recover the most important relations between pairs of nodes. By increasing the number of snapshots taken it is possible to estimate the lowest weighted relations.

After doing all these simulations it is interesting to note that the time needed for the three different algorithms is significantly different. To compute the conditional correlation is approximately $10\times$ faster than to solve the optimization problem by using the Linear Regression method. The slowest solution is the *ML* estimator because it required approximately $100\times$ more time than the Linear Regression method.

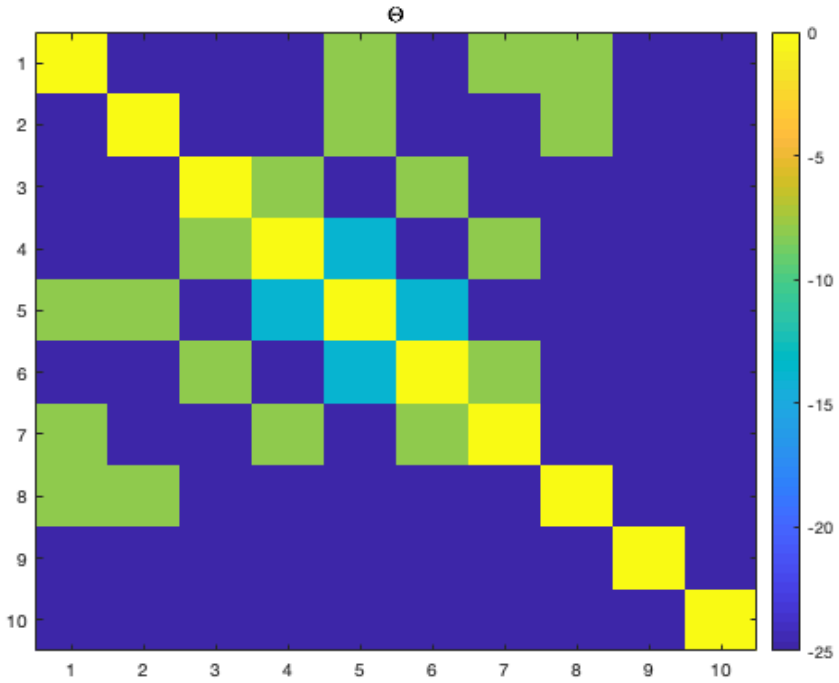


Figure 4.7: $\boldsymbol{\Theta}$ generated with a percentage of values different to 0 of 30%

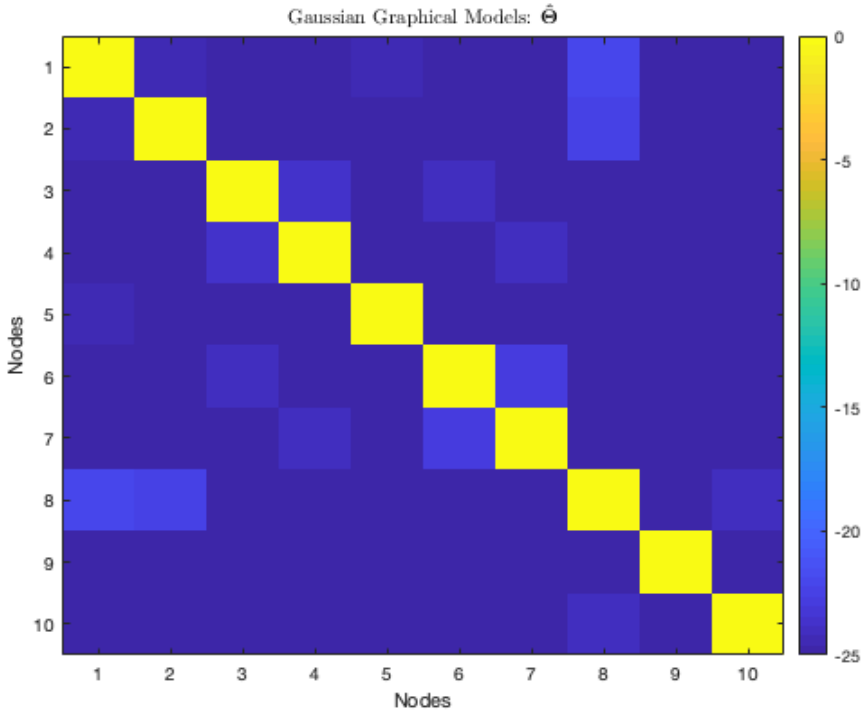


Figure 4.8: $\hat{\boldsymbol{\Theta}}$ estimated by using the Gaussian Graphical Models, *ML* estimator

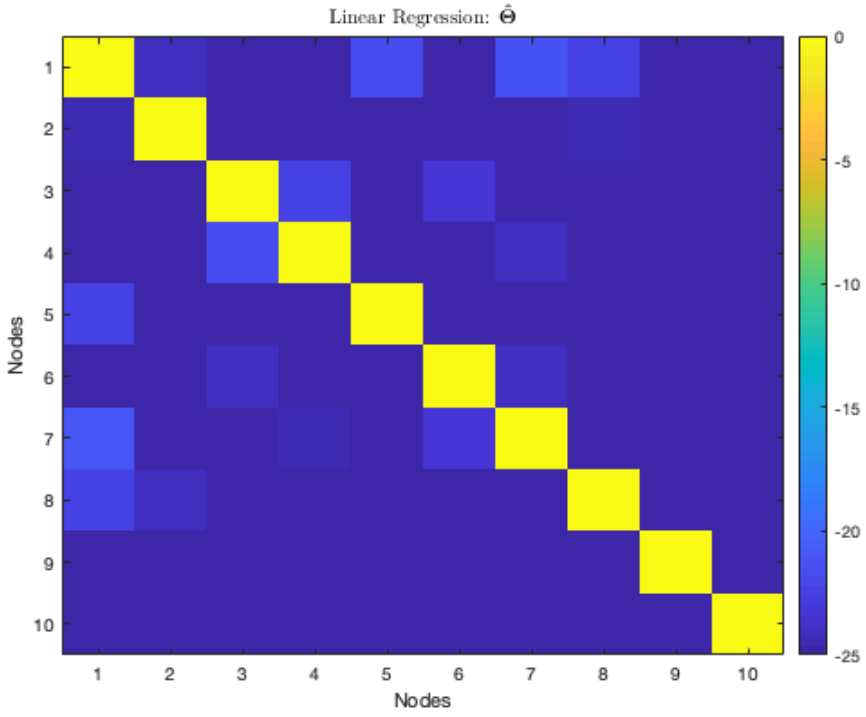


Figure 4.9: $\hat{\boldsymbol{\Theta}}$ estimated by using the Linear Regression method

# Chapter 5

# Conclusions and Future Work

## 5.1 Conclusions

As we observed, this Master's Thesis is divided in three different parts. The first one is focused in network topology inference when the signal graph can be modeled as the output of a graph filter when a driving noise is applied. The second part is centered in a more general branch, the study of statistical methods. Finally, the third part is devoted to simulations and comparisons between the statistical methods.

In chapter 2, we reviewed the concepts already studied in the Degree's Thesis that were needed to understand section 2.3. This part corresponds to the green branch in figure 5.1. In the previous work section 2.2, it was seen how to infer the topology when working with imperfect spectral templates. At this point, a natural question arises: how do we infer the topology when working with noisy spectral templates? This question is studied in this Master's Thesis in section 2.3. It was examined why the *OMP* algorithm was not working due to the *RIP* condition, which could not be guaranteed. Then, the study is centered in noisy spectral templates, a new robust method based on *LASSO* is proposed. In section 2.3.3, we compared the results obtained using the *BP*, *OMP*, and Robust Method, with the results obtained using *LASSO*. As it was previously showed, the reliability of this new technique is higher than the obtained with the *BP* or *OMP* algorithms. It is important to clear up that, *LASSO* does not require to have knowledge about the graph's sparsity degree. It was studied the algorithm's dependency of the regularization parameter. Which was showed that the regularization parameter did not modify the *RMSE* for different sparsity degrees. The most important conclusion is that for a number of nodes $N$ tending to infinite, the *RMSE* obtained with the *LASSO* approach tends to 0.

In chapter 3, we studied statistical methods. This part corresponds to the blue branch in figure 5.1. As exposed, these techniques were more generic than those explained in chapter 2. These approaches did not require any other information but some snapshots taken from the graph. To begin with, we presented the signal model used along this chapter. Then, in then next section the correlation networks were presented. Three approaches to estimate the graph's topology were introduced in this chapter: the conditional correlation, the maximum likelihood estimator with sparse regularization, and the linear regression. The simplest one was the conditional correlation method but, it was not possible to use when having a low number of snapshots. Some simulations were performed to

explain some concepts regarding the precision matrix $\mathbf{\Theta}$, as well as, to see the method's performance. It was then exposed, that it is typical to assume a Gaussian distribution. Gaussian graphical models were presented and the *ML* estimator was derived based on Gaussian assumption. The sparsity was also considered in the *ML* estimator, to finally obtain the *ML* estimator with sparse regularization in 3.5.1. Some simulations were performed to validate the method. At the end of this chapter, the linear regression technique was explained. It was formulated and the sparse regularization was taken into account. The predictor was estimating for each node $i$ the mean square error predictor. Then, a normalization was needed because each of these predictors had its own scaling factor. In order to conclude this section, the precision matrix was build where each of its columns $i$ was the normalized predictor $\boldsymbol{\beta}_i$. Some simulations were carried out to show the potential of this technique when working with low information.

In chapter 4, more simulations were performed to better compare the three approaches studied in chapter 3. As explained, the differences between these techniques and their potential can be better seen, when the number of nodes, and the sparsity degree increase. This is the reason why, in this chapter, we increased the number of nodes up to $N = 10$. It was explained that the number of nodes was not higher because it was already possible to see the differences between the techniques, and to easily analyze the results. With a higher number of nodes, the analysis would be much more difficult. Two different precision matrices were generated to test the methods. In the first case, it was seen that the methods which were considering the sparsity promotion obtained better results. Also, it was showed that if the linear regression method does not consider the sparsity promotion term, its result is equal to the one obtained with the conditional correlation. In the second example, we tried to estimate the precision matrix with only $N - 1$ snapshots. As it was explained, it made no sense to test the conditional correlation method due to the lack of information to correctly estimate it, the covariance matrix is singular. It was seen that, although the techniques had some difficulties estimating the precision matrix's lowest values, they were able to correctly estimate most of the highest values. Also, it was noted that the linear regression method obtained a better result than the *ML* estimator. Furthermore, the linear regression algorithm was much faster than the *ML* method.

## 5.2 Future Work

To better complete the study of topology inference in graph signal processing it remains to study of *Smooth Signals*. This part corresponds to the yellow branch in figure 5.1. There are multiple applications were it is interesting to construct a graph where data may have a certain regularity. As an example, it exists the possibility to learn the graph's topology by observing smooth signals [11, 67, 68, 69]. A signal graph is said to be smooth when the graph's weights tend to be similar. An objective measure of the signal smoothness can be quantified by the total variation (TV) [11, 67, 70] with respect to the Laplacian. These signals are interesting because they admit low-pass and band-limited representations using the Graph Fourier Transform basis. An example of this band-limited representations are sparse signals. Furthermore, smoothness is a very important property used in multiple graph-based statistical learning tasks such as nearest-neighbor prediction, denoising or spectral clustering. There are multiple examples of smooth signals in real

world. Those examples include natural images [67], types of practice in a network of lawyer collaborations and product ratings supported over similarity graphs of items or consumers [4], the environmental measures taken over a large area to the find relations between them [9] or the attack detection over the electrical grid [71].

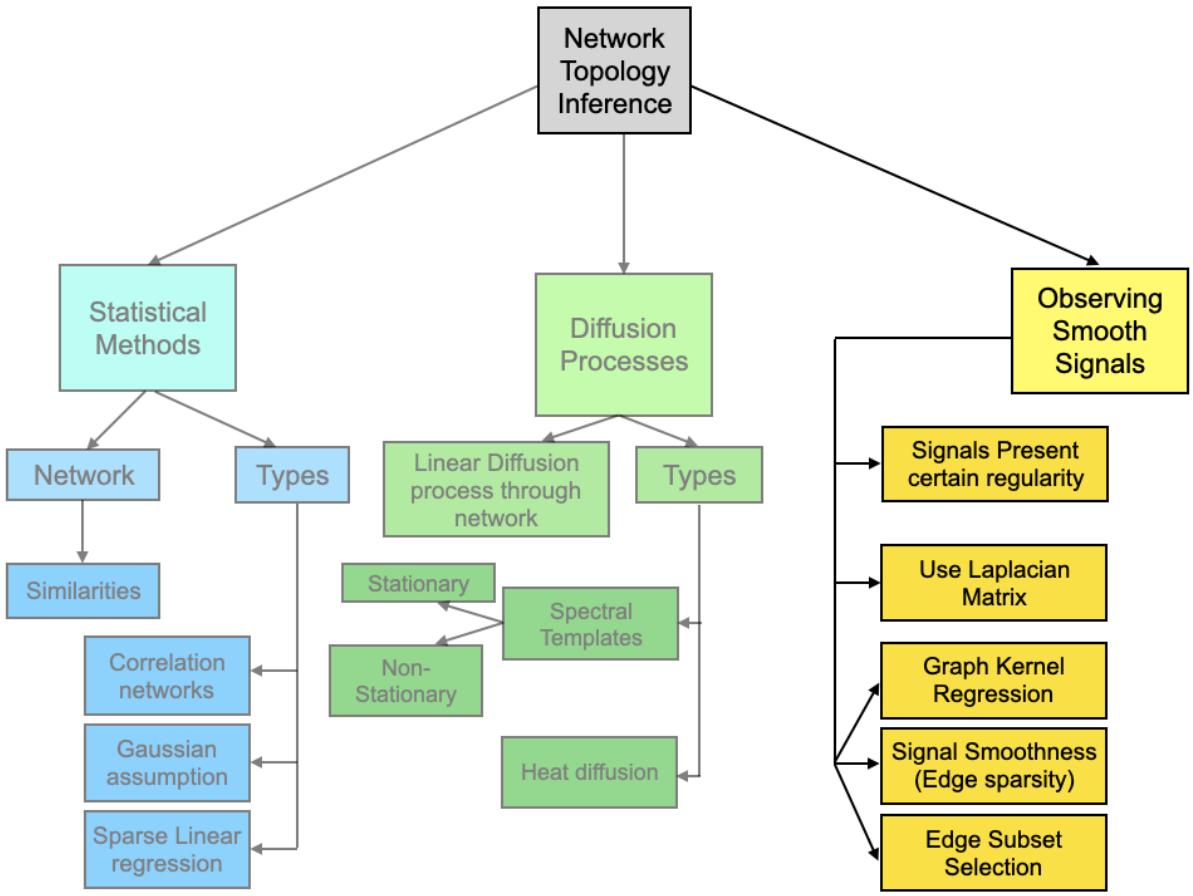


Figure 5.1: General schema of the future work of the theoretical part

# Bibliography


[1] C. García, "Network topolgy with imperfect spectral templates on graphs (final degree thesis)," June 2018.

[2] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Chapter 7 - distributed detection and estimation in wireless sensor networks**this work has been partially supported by the european project tropic, nr. ict-318784." in Academic Press Library in Signal Processing: Volume 2, ser. Academic Press Library in Signal Processing, N. D. Sidiropoulos, F. Gini, R. Chellappa, and S. Theodoridis, Eds. Elsevier, 2014, vol. 2, pp. 329 – 408. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123965004000077

[3] J. A. Deri and J. M. F. Moura, "New york city taxi analysis with graph signal processing," in 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2016, pp. 1275–1279.

[4] W. Huang, A. G. Marques, and A. R. Ribeiro, "Rating prediction via graph signal processing," IEEE Transactions on Signal Processing, vol. 66, no. 19, pp. 5066–5081, 2018.

[5] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph filters," in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6163–6166.

[6] S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed implementation of linear network operators using graph filters," in 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2015, pp. 1406–1413.

[7] S. Kanaya, A. Batushansky, D. Toubiana, and A. Fait, "Correlation-based network generation, visualization, and analysis as a powerful tool in biological studies: A case study in cancer cell metabolism."

[8] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," IEEE Internet of Things Journal, vol. 1, no. 1, pp. 22–32, 2014.

[9] I. Jabłoński, "Graph signal processing in applications to sensor networks, smart grids, and smart cities," IEEE Sensors Journal, vol. 17, no. 23, pp. 7659–7666, 2017.

[10] A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes, M. Blais, and S. O'Banion, "Examining covid-19 forecasting using spatio-temporal graph neural networks," 2020.

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

[12] R. Shafipour, S. Segarra, A. G. Marques, and G. Mateos, "Identifying the topology of undirected networks from diffused non-stationary graph signals," 2018.

[13] J. A. Tropp, "Just relax: convex programming methods for identifying sparse signals in noise," IEEE Transactions on Information Theory, vol. 52, no. 3, pp. 1030–1051, 2006.

[14] B. Baingana, G. Mateos, and G. B. Giannakis, "Proximal-gradient algorithms for tracking cascades over social networks," IEEE Journal of Selected Topics in Signal Processing, vol. 8, no. 4, pp. 563–575, 2014.

[15] D. Thanou, X. Dong, D. Kressner, and P. Frossard, "Learning heat diffusion graphs," IEEE Transactions on Signal and Information Processing over Networks, vol. 3, no. 3, pp. 484–499, 2017.

[16] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," SIAM Review, vol. 43, no. 1, pp. 129–159, 2001. [Online]. Available: https://doi.org/10.1137/S003614450037906X

[17] C. F. Loan, "The ubiquitous kronecker product," Journal of Computational and Applied Mathematics, vol. 123, no. 1, pp. 85 – 100, 2000, numerical Analysis 2000. Vol. III: Linear Algebra. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377042700003939

[18] S. Liu and O. TRENKLER, "Hadamard, khatri-rao, kronecker and other matrix products," International Journal of Information & Systems Sciences, vol. 4, 01 2008.

[19] S. Kunis and H. Rauhut, "Random sampling of sparse trigonometric polynomials, ii. orthogonal matching pursuit versus basis pursuit," Foundations of Computational Mathematics, vol. 8, no. 6, pp. 737–763, 2008. [Online]. Available: https://doi.org/10.1007/s10208-007-9005-x

[20] E. J. Candes and T. Tao, "Decoding by linear programming," IEEE Transactions on Information Theory, vol. 51, no. 12, pp. 4203–4215, 2005.

[21] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," Comptes Rendus Mathematique, vol. 346, no. 9, pp. 589 – 592, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1631073X08000964

[22] T. Zhang, "Sparse recovery with orthogonal matching pursuit under rip," IEEE Transactions on Information Theory, vol. 57, no. 9, pp. 6215–6221, 2011.

[23] M. A. Davenport and M. B. Wakin, "Analysis of orthogonal matching pursuit using the restricted isometry property," IEEE Transactions on Information Theory, vol. 56, no. 9, pp. 4395–4401, Sep. 2010.

[24] N. Hurley and S. Rickard, "Comparing measures of sparsity," IEEE Transactions on Information Theory, vol. 55, no. 10, pp. 4723–4741, Oct 2009.

[25] P. A. Traganitis, Y. Shen, and G. B. Giannakis, "Network topology inference via elastic net structural equation models," in 2017 25th European Signal Processing Conference (EUSIPCO), 2017, pp. 146–150.

[26] P. Erdős and A. Rényi, "On the evolution of random graphs," in PUBLICATION OF THE MATHEMATICAL INSTITUTE OF THE HUNGARIAN ACADEMY OF SCIENCES, 1960, pp. 17–61.

[27] E. Ben-Naim, P. L. Krapivsky, and S. Redner, "Complex networks."

[28] T. L. Mifflin, C. Boner, G. A. Godfrey, and J. Skokan, "A random graph model for terrorist transactions," in 2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720), vol. 5, 2004, pp. 3258–3264 Vol.5.

[29] K. Kandhway and J. Kuri, "Campaigning in heterogeneous social networks: Optimal control of si information epidemics," IEEE/ACM Transactions on Networking, vol. 24, no. 1, pp. 383–396, 2016.

[30] S. Garg and S. Kumar, "Modeling and analyzing information diffusion behaviour of social networks," in 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014, pp. 566–572.

[31] A. Chatterjee, D. Das, M. K. Naskar, N. Pal, and A. Mukherjee, "Heuristic for maximum matching in directed complex networks," in 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 1146–1151.

[32] A. Fukushima, M. Kusano, H. Redestig, M. Arita, and K. Saito, "Metabolomic correlation-network modules in arabidopsis based on a graph-clustering approach," BMC Systems Biology, vol. 5, no. 1, p. 1, 2011. [Online]. Available: https://doi.org/10.1186/1752-0509-5-1

[33] M. Ménoret, N. Farrugia, B. Pasdeloup, and V. Gripon, "Evaluating graph signal processing for neuroimaging through classification and dimensionality reduction," in 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2017, pp. 618–622.

[34] A. P. Georgopoulos, "Networks of the brain by olaf sporns," The Quarterly Review of Biology, vol. 87, no. 2, pp. 157–158, 2012. [Online]. Available: https://doi.org/10.1086/665450

[35] N. M. Morel, J. M. Holland, J. [van der Greef], E. W. Marple, C. Clish, J. Loscalzo, and S. Naylor, "Primer on medical genomics part xiv: Introduction to systems biology—a new approach to understanding disease and treatment," Mayo Clinic Proceedings, vol. 79, no. 5, pp. 651 – 658, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0025619611622878

[36] Q. Gu, Z. Li, and J. Han, "Generalized fisher score for feature selection," 2012.

[37] E. D. Kolaczyk, Statistical Analysis of Network Data, ser. Springer Series in Statistics.

[38] B. Anderson and J. Moore, Optimal Filtering. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[39] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," 06 2013.

[40] G. Lee, A. Singanamalli, H. Wang, M. D. Feldman, S. R. Master, N. N. C. Shih, E. Spangler, T. Rebbeck, J. E. Tomaszewski, and A. Madabhushi, "Supervised multi-view canonical correlation analysis (smvcca): Integrating histologic and proteomic features for predicting recurrent prostate cancer," IEEE Transactions on Medical Imaging, vol. 34, no. 1, pp. 284–297, 2015.

[41] V. Uurtio, J. M. Monteiro, J. Kandola, J. Shawe-Taylor, D. Fernandez-Reyes, and J. Rousu, "A tutorial on canonical correlation methods," ACM Computing Surveys, vol. 50, no. 6, p. 1–33, Jan 2018. [Online]. Available: http://dx.doi.org/10.1145/3136624

[42] M. Borga, "Canonical correlation a tutorial," 1999.

[43] J. Cai and H. Sun, "Kernel-based conditional canonical correlation analysis via modified tikhonov regularization," Applied and Computational Harmonic Analysis, vol. 41, 04 2015.

[44] E. P. Xing, X. Ruan, and K. Kandasamy, Gaussian graphical models and Ising models: modeling networks, ser. 708. Carniege Mellon University, vol. 10.

[45] E. P. Xing, Undirected Graphical Models, ser. 708. Carniege Mellon University, vol. 10.

[46] E. Xing, "Lecture 3: Undirected graphical models," Jan 2019. [Online]. Available: https://sailinglab.github.io/pgm-spring-2019/notes/lecture-03/

[47] P. Sidén and F. Lindsten, "Deep gaussian markov random fields," 2020.

[48] A. P. Dempster, "Covariance selection," Biometrics, vol. 28, no. 1, pp. 157–175, 1972. [Online]. Available: http://www.jstor.org/stable/2528966

[49] G. R. PRICE, "Extension of covariance selection mathematics," Annals of Human Genetics, vol. 35, no. 4, pp. 485–490, 1972. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1469-1809.1957.tb01874.x

[50] F. Wong, C. K. Carter, and R. Kohn, "Efficient estimation of covariance selection models," Biometrika, vol. 90, no. 4, pp. 809–830, 12 2003. [Online]. Available: https://doi.org/10.1093/biomet/90.4.809

[51] S. Zhou, P. Rutimann, M. Xu, and P. Buhlmann, "High-dimensional covariance estimation based on gaussian graphical models," 2010.

[52] S. Sullivant, K. Talaska, and J. Draisma, "Trek separation for gaussian graphical models," The Annals of Statistics, vol. 38, no. 3, p. 1665–1685, Jun 2010. [Online]. Available: http://dx.doi.org/10.1214/09-AOS760

[53] M. Stepančič and J. Kocijan, "On-line identification with regularised evolving gaussian process," in 2017 Evolving and Adaptive Intelligent Systems (EAIS), 2017, pp. 1–7.

[54] H. Wang, X. Gao, K. Zhang, and J. Li, "Single image super-resolution using gaussian process regression with dictionary-based sampling and student- $t$ likelihood," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3556–3568, 2017.

[55] D. Moungsri, T. Koriyama, and T. Kobayashi, "Duration prediction using multiple gaussian process experts for gpr-based speech synthesis," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 5495–5499.

[56] K. G. Gard, H. M. Gutierrez, and M. B. Steer, "Characterization of spectral regrowth in microwave amplifiers based on the nonlinear transformation of a complex gaussian process," IEEE Transactions on Microwave Theory and Techniques, vol. 47, no. 7, pp. 1059–1069, 1999.

[57] J. A. Tropp, "Just relax: convex programming methods for identifying sparse signals in noise," IEEE Transactions on Information Theory, vol. 52, no. 3, pp. 1030–1051, 2006.

[58] B. K. Natarajan, "Sparse approximate solutions to linear systems," SIAM Journal on Computing, vol. 24, no. 2, pp. 227–234, 1995. [Online]. Available: https://doi.org/10.1137/S0097539792240406

[59] J. F. Claerbout and F. Muir, "Robust modeling with erratic data," GEOPHYSICS, vol. 38, no. 5, pp. 826–844, 1973. [Online]. Available: https://doi.org/10.1190/1.1440378

[60] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," Biostatistics, vol. 9, no. 3, pp. 432–441, 12 2007. [Online]. Available: https://doi.org/10.1093/biostatistics/kxm045

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

[62] M. H. Wright, "The interior-point revolution in optimization: History, recent developments, and lasting consequences," BULLETIN (New Series) OF THE AMERICAN MATHEMATICAL SOCIETY, vol. 42, p. 39–56, 2005. [Online]. Available: https://www.ams.org/journals/bull/2005-42-01/S0273-0979-04-01040-7/S0273-0979-04-01040-7.pdf

[63] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press, 2004.

[64] L. M. Ramos Carvalho and A. R. Leite Oliveira, "Primal-dual interior point method applied to the short term hydroelectric scheduling including a perturbing parameter," IEEE Latin America Transactions, vol. 7, no. 5, pp. 533–538, 2009.

[65] Y. Tan and R. Ma, "Stochastic optimal power flow with wind generator based on stochastic response surface method (srsm) and interior point methods," in 2015 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies (DRPT), 2015, pp. 2079–2083.

[66] C. Hu and Q. Liu, "Online identification for hypersonic vehicle using recursive maximum likelihood method based on interior-point algorithm," in 2013 25th Chinese Control and Decision Conference (CCDC), 2013, pp. 1862–1867.

[67] V. Kalofolias, "How to learn a graph from smooth signals," 2016.

[68] M. G. Rabbat, "Inferring sparse graphs from smooth signals with theoretical guarantees," in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 6533–6537.

[69] E. Pavez and A. Ortega, "Generalized laplacian precision matrix estimation for graph signal processing," in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 6350–6354.

[70] A. Gholami and S. M. Hosseini, "A balanced combination of tikhonov and total variation regularizations for reconstruction of piecewise-smooth signals," Signal Processing, vol. 93, no. 7, pp. 1945 – 1960, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168412004501

[71] E. Drayer and T. Routtenberg, "Detection of false data injection attacks in smart grids based on graph signal processing," 2018.