Détection d'événements et inférence de structure pour des vecteurs sur graphes

Batiste Le Bars

université
PARIS-SACLAY

Event detection and structure inference for graph vectors

Thèse de doctorat de l'Université Paris-Saclay

Ecole Doctorale de Mathématique Hadamard (EDMH) n° 574
Spécialité de doctorat : Mathématiques appliquées
Unité de recherche : Centre Borelli (ENS Paris-Saclay), UMR 9010 CNRS
Référent : Ecole normale supérieure de Paris-Saclay

Thèse présentée et soutenue en visioconférence totale le 29 Janvier 2021, par

Batiste LE BARS

Au vu des rapports de :

George Michailidis Professeur, Université de Floride	Rapporteur
Fabrice Rossi Professeur, Université Paris-Dauphine	Rapporteur

Composition du jury :

Charles Bouveyron Professeur, Université Côte d'Azur & INRIA	Président
George Michailidis Professeur, Université de Floride	Rapporteur
Fabrice Rossi Professeur, Université Paris-Dauphine	Rapporteur
Tabea Rebafka Maitresse de conférence, Sorbonne Université	Examinatrice
Gilles Blanchard Professeur, Université Paris-Saclay	Examinateur
Nicolas Vayatis Professeur, Ecole Normale Supérieure Paris-Saclay	Directeur
Argyris Kalogeratos Chercheur, Ecole Normale Supérieure Paris-Saclay	Coencadrant
Olivier Isson Sigfox	Coencadrant

Thèse de doctorat

NNT: 2021UPASM003


école
normale
supérieure
paris–saclay
université
PARIS-SACLAY
CENTRE
BORELLI
Fondation mathématique
FMJH
Jacques Hadamard
sigfox

Contents

Remerciements

Après avoir soutenu ma thèse il y a quelques mois, je me lance dans l'écriture des remerciements. Étonnamment, c'est peut-être la seule partie de mon manuscrit de thèse que j'arrivais déjà à imaginer dès le tout début de ces quatre années. Je n'avais rien trouvé, pas même mon sujet de thèse, que je me voyais rédiger des remerciements bien écrits et originaux : la seule partie à la lecture appréciable pour ma famille et mes amis. Il s'agit finalement du dernier élément que j'appose à mon manuscrit. Et maintenant que tout est fini, la pression redescendue, je rédige des remerciements qui ressembleront en définitive à tous les autres (désolé !).

Je tiens tout d'abord à remercier mon directeur de thèse, Nicolas Vayatis, ainsi que mon co-encadrant, Argyris Kalogeratos, pour leur soutien et leur confiance au cour de ces quatre dernières années. Je remercie également Sigfox et en particulier Olivier Isson qui a su me donner toutes les libertés nécessaires à l'élaboration d'une thèse Cifre. Merci aussi pour ces très bons moments passés autour des meilleures tables de Toulouse.

Je remercie ensuite tous les membres du jury qui ont accepté de participer à ma soutenance de thèse. Merci à Fabrice Rossi et George Michailidis d'avoir tout de suite accepté leur rôle de rapporteurs. Merci au président, Charles Bouveyron et merci également aux deux examinateurs.trices, Tabea Rebafka et Gilles Blanchard.

Mes prochains remerciements vont à l'armée de doctorants du CMLA/Centre Borelli. C'est aussi grâce à chacun de vous que ces années de thèse, parfois difficiles, se sont si bien passées. Merci pour l'entraide, les rires, les ragots. Merci surtout d'avoir supporté ma grosse voix et écouté mes plaintes. Mention spécial à la team des Saints-Pères bien sûr, avec qui j'ai partagé tout cela "puissance dix".

Mon plus gros des "shout out" va bien évidemment à Pierre Humbert, mon frère dans ce labo, avec qui j'ai eu la chance de contribuer et de discuter longuement de tout et de rien. Merci notamment pour nos discussions sur des concepts mathématiques des plus basiques, pendant lesquelles nous n'avons aucune honte à nous poser mutuellement des questions auxquels un étudiant de L1 maths pourrait répondre. Je ne vais pas m'éterniser car on en a déjà beaucoup parlé mais merci pour tout !

Je remercie également mon binôme du côté de Sigfox, Kevin Elgui. On se suit depuis le MVA et j'espère qu'on sera amené à retravailler ensemble un jour ! Merci pour les discussions de maths, les cafés à rallonge et surtout merci pour toutes ces bières et bons repas partagés à Toulouse.

Enfin, merci à mes parents, mon beau-père, mon frère et ma soeur. Merci aussi à tous mes copains et copines dont la liste serait un peu longue à énumérer. Ce n'était pas toujours évident de discuter de ma thèse avec vous, mais finalement tant mieux ! C'est aussi cela qui m'a permis de ne plus y penser, de décompresser et de terminer cette thèse sereinement.

Abstract

Among the diverse vector data that are collected in many different sectors, from medicine to industry or social networks, an important number of them are observed over a network structure. These data, referred as *graph vectors* or *graph signals*, have the particularity to associate each dimension of a vector to a specific node of a graph. This type of data can be observed in particular when the vectors arise directly from networks (e.g. telecommunication, sensor or social networks), but more generally for any vector admitting an underlying graph structure linking its variables.

This thesis addresses different problems around the analysis and the modeling of this type of vectors, using several different mathematical tools and techniques. In particular, we are interested in two tasks. The first one is the problem of *event detection*, i.e. anomaly or change-point detection, in a set of graph signals. The second task concerns the *inference of the graph* structure underlying the observed graph vectors that are contained in a data set.

At first, our work takes an application-oriented aspect in which we propose a method for detecting antenna failures or breakdowns in a telecommunication network. The proposed approach is designed to be effective for communication networks in a broad sense. It relies on the idea that some significant values recorded at a node (e.g. an antenna) are predictable knowing the values observed at the other nodes. Given this intuition, an anomaly will be detected whenever a node's value is far from the prediction that is made for it. With such formulation, we therefore understand that the method implicitly takes into account the underlying graph structure of the data, a node's value being predicted using with the others.

In a second time, the thesis takes a slightly more theoretical aspect. First, a new method for graph structure inference within the framework of *Graph Signal Processing* is investigated. In this problem, notions of both local and global smoothness, with respect to the underlying graph, are imposed to the vectors. These notions are the basic hypotheses of many algorithms treating with graph signals. In a final contribution, the graph learning task is combined with the change-point detection problem. This time, a probabilistic framework is considered to model the vectors, assumed to be distributed from a specific *Markov Random Field*. In the considered modeling, the graph underlying the data is allowed to evolve in time and a change-point is actually detected whenever this graph changes significantly.

Notations

Sets, matrix and vectors

$[n]$	Set of integers $\{1, \dots, n\}$
$\mathcal{S} \setminus \mathcal{S}'$	Set $\mathcal{S}$ minus the subset $\mathcal{S}' \subset \mathcal{S}$
$x^\mathsf{T}, M^\mathsf{T}$	Transpose of vector x, matrix M
$x_{\setminus j}$	Vector x deprived of its j-th coordinate
$x_\mathcal{I}$	Value of the vector x for the subset of indices $\mathcal{I}$
$M_{i,j}$, $M_{i,:}$ and $M_{:,j}$	(i,j)-entry, i-th row and j-th column of a matrix M
I_n	Identity matrix in $\mathbb{R}^{n \times n}$
$\mathbf{0}_n$	Vector of size n containing only zeros
$\mathbf{1}_n$	Vector of size n containing only ones

Graphs

$G = (\mathcal{V}, \mathcal{E})$	A graph with set of nodes $\mathcal{V}$ and edges $\mathcal{E}$
p or N	Number of nodes in the graph
W	Positive weight matrix
L	Laplacian matrix
$\mathcal{N}(u)$	Neighborhood of the node $u \in \mathcal{V}$

Functions and norms

$\lvert x \rvert$ or $\lvert \mathcal{S} \rvert$	Absolute value of the scalar x or the size of the set $\mathcal{S}$
$\mathbb{1}_A(\cdot)$	The indicator function over the set A
$\langle \cdot, \cdot \rangle$	Inner product function
$\lVert x \rVert_0$	The number of non-zero elements of a vector x
$\lVert \cdot \rVert_F$	The Frobenius norm
$\lVert \cdot \rVert_{2,0}$	The number of non-zero rows of a vector M
$\lVert \cdot \rVert_{2,1}$	The $\ell_{2,1}$-norm, with $\lVert M \rVert_{2,1} = \sum_{i=1} \lVert M_{i,:} \rVert_2$

1 Introduction

Contents

1 Context of the thesis

1.1 General context

Over the past decades, the significant increase of the amount of multivariate data available in all sectors, from medicine to industry or social networks, has created urgent needs for data analysis and modeling in order to accomplish various tasks. Among them, the tasks of anomaly, or change-point, detection (called *event detection* when referring to both of them) in massive amounts of data are of major importance. In a nutshell, anomaly detection refers to the task of finding, in a data set, a small amount of vector data that deviates from the *normal* behavior of the vast majority. This task can be applied to any data set and in many real-world applications. For example, it can be used to discover dysfunctional elements in a company's production line. On the other hand, the theoretical basis of change-point detection relies on the vast area of time-series analysis, and it seeks to find time-instances at which there is a change in the regime underlying the data (Figure 1.1). In life science and biology applications, it can for example correspond to the moments when there is a change in the state of the of the monitored system (e.g. an individual sleeping or not via electroencephalogram analysis, the beginning of puberty via the hormone secretion study etc.).

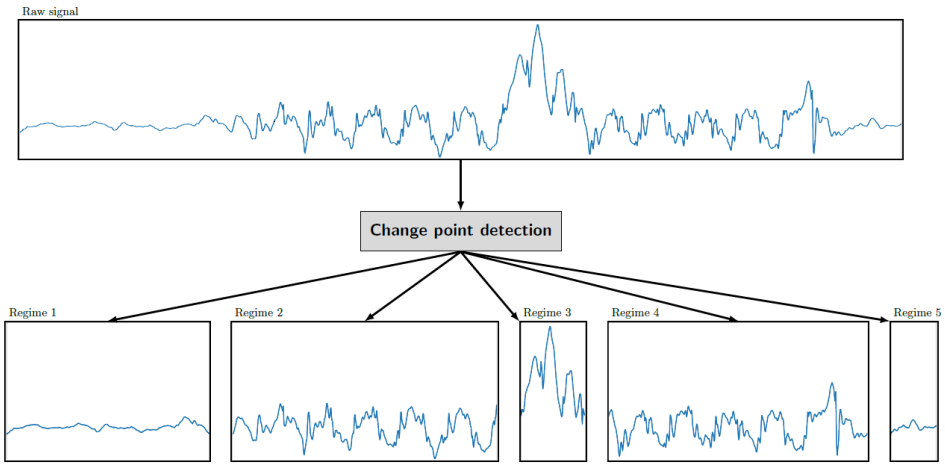


Figure 1.1: Simple illustration of a change-point detection task [157]. In this example, the signal is univariate, 4 change-points are detected, resulting in 5 regimes of different behavior.

New challenges to these problems have been posed by data that appear naturally over a network structure, which interconnects the observations or the variables representing them. This is particularly the case for data coming from social, communication, transport or sensor networks, where the data collection takes place at the level of nodes in a graph. In the social network example, the nodes may refer to users and an edge indicates the social link between two of them; in a sensor network, an edge may simply correspond to the spatial distance between two sensors. Usually, the graph brings knowledge on the process that generates the data (e.g. two linked nodes are highly correlated or has very close values), and being able to build models or learning algorithms – including anomaly or change-point detection methods – from these data, while considering their network structure, is of major concerns to improve learning performances.

This type of collected data are referred as *graph data*, *graph vectors* or *graph signals*. As stated earlier, they simply refer to vector data for which each component is associated to a node in a graph. While in some cases the graph is naturally given and therefore known *a priori* (e.g. the social network or the sensor network example given above), there are many cases where the data admit an underlying graph structure that is not available and needs to be learned from them. This is notably the case in biology, where we are interested in knowing which genes (or proteins) are expressed with each other [62, 94]. More generally, this need for graph inference can appear in any type of data for which one wishes to know which variables are linked with which others, in the sense that they behave in a similar way statistically. This task can have strong impact on the visualization and the understanding of the data being processed, but also, as said before, on the ability to build more efficient learning algorithms.

This thesis work is deeply rooted in the various topics mentioned above. In particular, it focuses on event detection in graph data, whether the graph is known or not. It also focus on the inference of the graph itself, and on the detection of changes in the graph structure underlying the data. Among the notable achievements of this thesis are the methods developed, but also the fact that for the purpose of their development, several mathematical tools and techniques were employed.

1.2 Industrial context

This doctoral thesis was carried out thanks to CIFRE (Convention Individuelle de Formation par la Recherche) and commissioned by the ANRT (Agence Nationale de la Recherche et de la Technologie). It was sponsored by Sigfox, a world-wide telecommunication network operator dedicated to the Internet-Of-Things (IoT). Specifically, Sigfox owns antennas, referred as Base Stations (BSs) that are set up on towers (like a cell phone company), and receives data transmissions from devices like parking sensors, fire detector, water meters, etc. The devices are held by customers, and the role of Sigfox is essentially to collect the (encrypted) data transmitted by the devices, via the BSs, and send them to a cloud to which the customer has access. The particularity of Sigfox is its simple protocol, which is suitable for the devices to send small amounts of data (12 bytes per sent "messages") but received at a long-range and in a very low-power consuming way. Without going into much more detail, let us briefly describe the process of sending a message in the Sigfox protocol.

Let a device that needs to transmit a small amount of data (e.g. temperature, binary information, etc.) to its owner. The information is encoded in a signal that is sent three times, at three different frequency levels, for robustness. The signals are sent without any selection protocol on the Base Stations, they are simply sent "in the air" (broadcast), hoping that they will be received by at least one nearby BS (thanks to a vast coverage, the signals are usually received by many of them). Once at least one BS receives one of the three repeated signals, it is decoded and sent to the cloud using standard internet protocol such as 3G. A scheme of Sigfox network's architecture is provided by figure 1.3. For a more complete description on Sigfox technology and Low-Power Wide-Area Network (LPWAN) in general, we invite the reader to look at [29].

Initially, the principal objective of the collaboration with Sigfox was to propose and develop data-based methods to detect anomalies at the level of a Base Station (e.g. caused by a breakdown). This was a new subject, poorly dealt by the company's researchers and engineers, but the need for it became more and more important due to the significant expansion of the network. Until then, very little use was made of the collected data for this

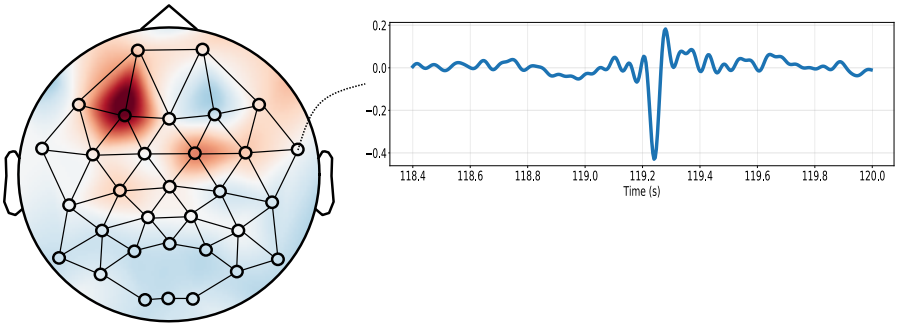


Figure 1.2: Example of signals recorded over a graph structure. (Left) Sensors measuring brain activity over time via electroencephalogram (EEG) are placed on a human head. Their positions over the head induce a graph structure that captures their proximity. A graph signal corresponds to a vector of size equal to the number of sensors (i.e nodes) containing the EEG values at a specific timestamp. In this scenario, it is generally expected that nearby nodes in the graph will have similar values. (Right) Time-series of observations recorded at a single node, pointed by the dotted line.

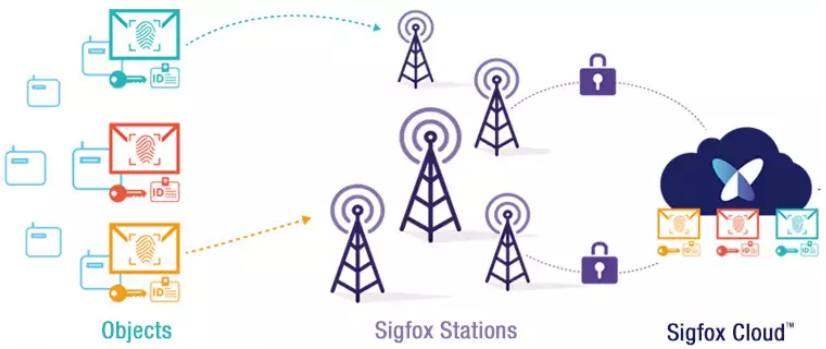


Figure 1.3: Summary of Sigfox's architecture.

purpose and only simple methods, based on threshold exceedances, has been employed. The thresholds were set a priori by field experts and led to a very high false positive rate.

Various data are collected at the level of each BS, from hardware (e.g. temperature of the engine) to software information (e.g. the OS version used). According to the experts, the important features to detect anomalies are those linked with the spectrum of the signals captured by the BSs. Unfortunately, the whole spectrum is not collected but only few statistics summarizing it are computed instead every second. These includes essentially some quantile information on the intensities recorded all over the spectrum. After few weeks of data analysis made at the beginning of this research, it has been finally decided that only the "reception" information about each BS would be kept. In other words, this corresponds the information on the activity of the network: for each message broadcasted, which BS has received it or not. This decision was essentially motivated by the fact that this kind of data are raw (contrarily to the spectrum data that are already processed) and with a priori few errors in it. Moreover, it is intuitively expected that a failure at the level of a BS will directly impact its level of activity, and most probably with a decrease in its total number of received signals.

An interesting property of the reception data is that they appear naturally over a graph structure induced by the spatial distribution of the BSs. Taking for example the vector that indicates, for a fixed sent signal, which BS has received it or not, it is empirically observed that nearby BSs will have more chance to be 'activated' (in the sense of reception) together, and conversely. Another example includes the vector that specifies the number of signals received by each BS over a certain period of time, where close BSs will have highly correlated values (see Figure 1.4).

Under these observations, we made the conclusion that the studied vector data can be considered as graph vectors, making the thesis move towards the task of event detection in a set of graph signals. Nevertheless, quickly, and as explained in detail in the following section, it was realized that the induced spatial graph was not all the time adapted to the data, leading to the second axis of the thesis related to the inference of the network structure itself.

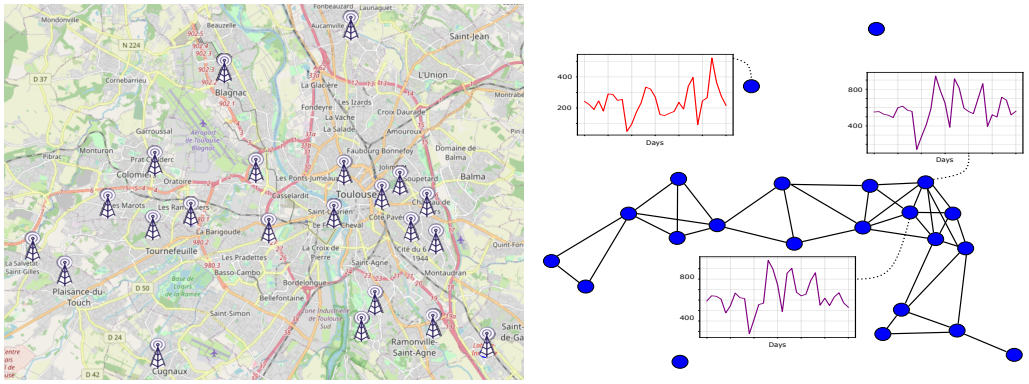


Figure 1.4: (Left) A subset of Base Stations (BSs) located near Toulouse, France. (Right) A naive graph representation that connects two nodes if the BSs are located less than 2km away from each other. The subplots indicates the number of messages received daily over one month by 3 nodes. Clearly, the two linked nodes (purple curves) are also closer to each other in terms of received network traffic, in both scale and correlation, compared to the isolated one (red curve).

2 Objectives and Motivations

In the following, we describe the different objectives that will be at the core of this document.

Detecting behavioral changes or anomalies at node-level in communication networks. This first objective is applied and essentially motivated by the industrial problem raised by the collaboration with Sigfox. However, this detection task can arise in many communication network-related settings (e.g. in social or computer networks), and in order for this work to be more generally applicable, we will seek to propose methods for event detection in a large spectrum of communications networks, including Sigfox. If, in the context of the industrial collaboration, such a task could boil down to the detection of a BS failure, in a sensor network for example [174] it can characterize the detection of an issue at the level of a sensor or the value it quantifies. In a computer or a social network, such detection problem can arise, for example, in network security where the anomaly can come from an attack (e.g. hacking, identity fraud, etc.) [6]. Previous examples illustrate well the importance of such a task: non-detected anomalies can have significant impacts in the performances of the considered network. To sum up, our objectives will be to, first, give a simple definition of a communication network which corresponds to a wide range of real-life deployed networks. Similarly, we will consider simple notions of anomalies that can be observed in various types of networks. Then, we will propose a way to detect anomalies. And in order to remain close to our industrial application, we will restrict ourselves to node-level detection, i.e. the event detection at the level of a single entity (e.g. a base station, a computer, etc.).

Detecting change-points or anomalies in a set of graph signals. This more general task was initially motivated by the conclusion we made in the previous section, namely the fact that the data of interest are graph vectors. But of course, as explained in the previous section, these types of signals are frequently observed in real-world applications,

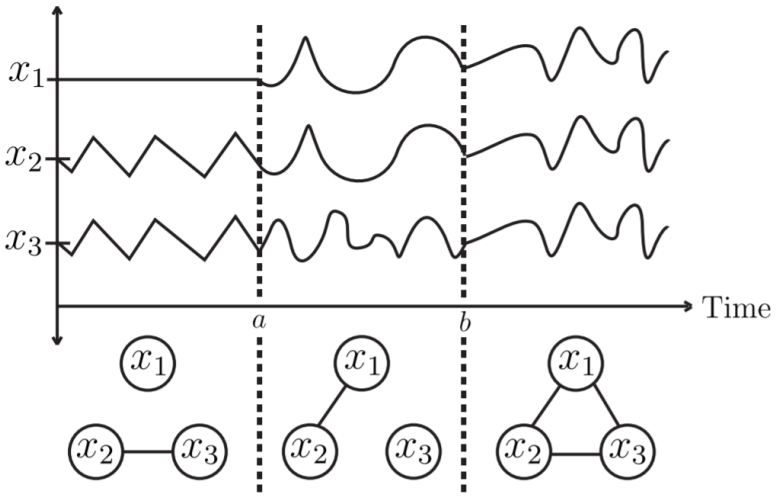


Figure 1.5: Illustration of the change-point detection task combined with graph inference. Each dimension of the time-series corresponds to a node in the corresponding graph [73].

and building new algorithms for detecting anomalies or change-points which are adapted to them is important. When the graph is known, it constitutes a prior knowledge about the model that generates the data. It is therefore conceivable that the performance of the algorithms can be improved by using this additional information [55]. It allows the construction of new features or embeddings based on the information brought by the graph, and thus, allows to discover more complex types of events [31]. Typically, an anomaly detected at the level of one node can come from a value which is abnormal, with respect to istself *and* with respect to its neighbors. When the graph is unknown, no much things can be done. However, knowing that there exists an underlying graph structure tells an important thing: if one wants to detect an anomaly or a change-point at the level of a specific dimension (i.e a specific node), the values observed at the other nodes should be considered as well since they are linked through graph edges [102]. Despite this, an unknown graph suggests a first step that would learn the graph itself, in order to understand better the data and apply more adapted event detection algorithms. This learning problem constitutes our third objective.

Learning the graph structure underlying vector data. This objective is met in many fields and can be applied to any kind of vector data. In an extension of the notion of graph signals previously mentioned, here the objective is to infer the underlying structure of the data. In other words, the goal is to learn notions of relationships between variables, i.e. with which other variables, a variable of interest is more similar or related with (e.g. in term of correlation, conditional independence, scale, values etc.). Such learning procedure is done using a set of vectors assumed to admit the same underlying graph. During the learning process, structural penalties can be imposed, such as sparsity of the graph [50, 61, 132], and in the present work, we will study some of them. Inferring such a graph structure has several applications. First of all, it helps to understand the considered vector data with a simple visualization brought by the graph. Moreover, such a learning task is often linked with a model. This is particularly the case with Markov Random Fields [97], which in many

situations assume a linear relationship between the variables, a relationship determined by the graph itself [79, 132]. With such a modeling, we can imagine that one could try to predict the value of a node as a function of the other node values and the graph weights. This illustrates well the applicability of the learned graph. It can also be used in many machine learning algorithms that require a graph, typically the spectral clustering algorithm [123], semi-supervised algorithms [17], in the framework of the *Graph Signal Processing* [124], etc. One of the most famous real-world application of graph learning is in biology, with genes interaction network emphasizing genes that are most of the time expressed together. At Sigfox, we could think that the graph is directly given by the spatial position of the BSs (Figure 1.4), assuming that nearby BS will receive a lot of messages in common. In practice this is not always true, two BSs located at a small distance from each other can be separated by a wall or be at different altitudes, making them quite different in their ability to receive a same signal. This observation made us conclude that the graph learning task could also be interesting with Sigfox reception data.

Detecting changes in the underlying structure of vector data. This objective can be seen as a combination of graph learning and change-point detection. In fact, contrarily to the vast majority of change-point detection techniques that look for a significant change in the mean of a time-series, the task here is to detect a change in the underlying graph structure itself. Thus the objective is twofold, finding time-instances between which all the observed vector data has the same underlying graph structure as well as learning these graphs. An illustration of such a task is provided by Figure 1.5 for a real-valued time-series with 3 dimensions. In addition to determining instances of time at which the system has undergone some changes, methods that meet this objective also take advantage of the benefits that can be brought by the graph inference (see the previous paragraph), i.e. modeling, applicability of machine learning algorithms, etc. In particular, the visualization aspect mentioned above allows a strong understanding and interpretability of the found change-points [104].

Dealing with binary graph signals. This last objective is essentially motivated by the fact that the considered vector data are binary. Indeed let us recall that in our application setting, a raw data is a vector corresponding to a message broadcasted by a device in Sigfox network. This vector encodes which BS have received the signal of the message or not. Nevertheless, this problem remains important in many other contexts [5], particularly due to the fact that it is often less studied than real-valued vector data or time-series.

3 Background

In this section, we propose to briefly recall some fundamentals on event detection, namely anomaly and change-point detection, on graph theory, and on vectors observed over graphs. The objective is to provide some basic definitions, notions, and algorithms that will be useful in the rest of the manuscript.

3.1 Anomaly detection

In its most classical version, anomaly detection refers to the task of finding in a data set a small amount of vector data that has been generated by a different distribution model

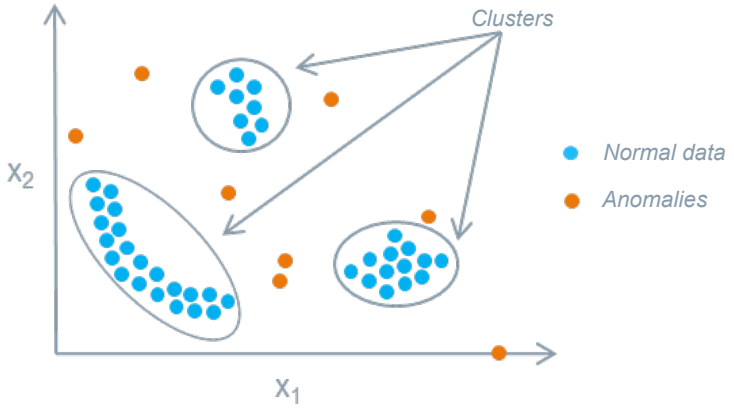


Figure 1.6: An illustrative example of anomalies in a two-dimensional data set.

than the majority of them. This simple formulation has motivated many statistical anomaly detection methods which assume that anomalies stand in regions of low density. Among them, the one of [58] assumes that the *normal* data are generated by a known in advance probability distribution and considers abnormal the data that stands in a region of low probability.

While being the most generic, the previous formulation actually corresponds to a particular framework of anomaly detection. In fact, depending on the labels available, anomaly detection tasks can be divided in three types. The *supervised* anomaly detection consists in training the algorithm based on a labeled (normal/abnormal) data set including both normal and abnormal observations. This type of detection is therefore highly related to the problem of classifying imbalanced data [151]. The second scenario is the *novelty detection* framework, also referred as one-class classification or semi-supervised anomaly detection. In this setting, only normal data are available for the learning phase. This is the case in applications where normal behaviors are known but e.g. intrusion or attacks induce an unknown behavior and must be detected. This scenario is the one considered in Chapter 2. Finally, the *unsupervised* setup refers to the one presented in the previous paragraph: the learning data set contains both normal and abnormal data but no labels are available.

Usually, most anomaly detection algorithms do not simply map the input vector to a binary value (indicating an anomaly or not). Instead, they return a real-valued function, referred as *scoring function*, that outputs, for a given input vector, a real value as score of abnormality. The advantage of using such a scoring function is that it allows to rank the samples from the less to the most abnormal. This is very important when one has a lot of data to deal with and wants to prioritize some anomalies. Moreover, if one wants a binary output, it suffices to fix a threshold above which the score will be considered abnormal.

There exists a wide variety of anomaly detection algorithms, from those based on density learning [25, 140, 141, 165], like previously explained, to those based on, e.g. decision trees [111]. This great variety of algorithms is accentuated by the kind of available labels, as explained earlier, but also by the type of the analyzed data. Here we focus on classical vector data, but these can be temporal or even text data. It would therefore be impossible to make an exhaustive list of these methods here, and for complete surveys one may refer to [28, 127]. In the following, we present an efficient anomaly detection algorithm, for

standard vector data, that performs well on both the unsupervised and the semi-supervised labeling scenarios.

3.1.1 One-class SVM

The One-class Support Vector Machine (SVM), first introduced in [140], extends the standard SVM for classification with two classes to the problem of novelty detection. Indeed, rather than having access to a labeled data set with both positive and negative labels, it assumes that all the input data belong to class 1 (the normal class). Then, instead of constructing a hyperplane separating two classes, it constructs a hyperplane separating the mapped input points from the origin of the mapped space, treated here as the only point of the second class.

Formally, in its *soft-margin* version, the One-class SVM works as follows.

Let $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$ be n observations and $\Phi : \mathbb{R}^d \to \mathcal{H}$ be a feature map in a Reproducing Kernel Hilbert Space [13] $\mathcal{H}$ with kernel k (usually the Gaussian kernel). To separate the data from the origin, the one-class SVM solves the following quadratic program:

$$\begin{aligned} \min_{\omega \in \mathbb{R}^d, \xi \in \mathbb{R}^n, \rho \in \mathbb{R}} \quad & \frac{1}{2}\|\omega\|^2 + \frac{1}{n}\sum_{i=1}^{n} \xi_i - \nu\rho \\ \text{s.t} \quad & \langle w, \Phi(x^{(i)}) \rangle \geq \rho - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i \geq 0, \end{aligned}$$

where $\nu \in (0, 1)$ is a hyperparameter that prevents overfitting and allows the training data set to contain outliers. In fact, it can be showed that ν corresponds to an upper bound on the fraction of anomalies that are allowed in the learning set. The decision function used to detect anomalies is given by:

$$f(x) = \text{sign}(\langle w, \Phi(x^{(i)}) \rangle - \rho),$$

which will, as required, be positive for most of the learning vectors.

From the previous definition, the One-class SVM can be seen as the estimation of a space having minimum volume while containing almost all the input data. In fact, it is linked with the estimation of a Minimum Volume set [141] with mass $1 - \nu$, i.e. the set of minimum volume with respect to the Lebesgue measure but having a mass of $1 - \nu$ with respect to the probability measure of the normal data.

3.1.2 Quality measures

Like any other learning method, being able to evaluate the quality of anomaly detection algorithms is very important. When labels are available, every quality measure used for binary classifier and scoring functions can also be employed in this case. These include, for example, the analysis of the Receiver Operating Characteristic (ROC) curve and the associated Area Under the Curve (AUC). One of the drawbacks of using such measures is that they are not particularly adapted for imbalanced classes. For this reason, measures related to the normal class, such as the false positive rate, may be preferred.

When few or no labels are available, the question of evaluating the quality of the algorithms remains an open question. Some tracks, seeking to extend the notion of ROC curves to the no label scenario, have notably been studied in [67] and are related to Excess-Mass and Mass-Volume curves. However, these methods will not be considered in the present manuscript and, when necessary, labels will be available.

3.2 Change-point detection

Change-point detection is a particular task of time-series analysis. Its objective is to find time-instances at which significant changes occur in the underlying model of a given time-series. A stated in Section 1.1, this task has many applications, whether it is in speech processing [76], climatology [134], network traffic data analysis [108], etc. An illustration of this segmentation problem is provided in Figure 1.1.

The change-point detection task can be divided in two main categories: *offline* and *online* detection. In the offline scenario, the segmentation is performed after the signal has been observed entirely, it is also referred as *a posteriori* detection. On the contrary, in the online scenario we aim at finding the change-points in real time, while the vector data are being observed. This task is also referred as *sequential* change-point detection. In the present section we focus on the offline scenario, which is considered in Chapter 4. This offline task can again be split in two subcategories: the case where the number of change-points to discover is *known* and the situation when it is *unknown*. Most of the time, the difference between the resolution of the two problems lies in the addition of a term penalizing the number of estimated change-points in the optimization program. Before going further, let us describe the problem more formally.

We consider the statistical framework described in the review of [157]. Let $\{x^{(i)}\}_{i=1}^{n}$ be a time series in $\mathbb{R}^d$ assumed to be piecewise stationary, meaning that there exist instances $\mathcal{T}^\star = \{t_1^\star, \dots, t_{K^\star}^\star\} \subset \{1 \dots, n\}$ at which the model underlying the time-series changes. The objective of change-point detection is to recover the indexes $\mathcal{T}^\star$, and therefore the number of change-points when it is unknown. To do this, most of the methods found in the literature seek a set of indexes $\mathcal{T} = \{t_k\}_{k=1}^{K} \subset \{1 \dots, n\}$, estimating $\mathcal{T}^\star$, that minimizes a function of the form

$$\sum_{k=0}^{K} c(\{x^{(i)}\}_{i=t_k+1}^{t_{k+1}}), \tag{1.1}$$

with $t_0 = 0$, $t_{K+1} = n$ and $c(\cdot)$ is a cost function that evaluates the quality of each learned segment. When the number of change-points is known, $K = K^\star$, otherwise, a term penalizing the size of $\mathcal{T}$ is added.

All change-point detection methods then differ in two main aspects: either on the cost function that is used, usually linked to the model underlying the data (e.g. parametric or not), or on the method used to solve the minimization problem stated above. The problem being combinatorial, many solutions have been proposed (greedy resolution, dynamic programming, heuristics, etc.). The wide variety of cost functions and minimization methods does not allow us to be exhaustive and we invite the reader to see the review of [157] for examples and in-depth discussion. Nevertheless, we give below an example of a model and an associated cost function that we think are important to know.

Example 1.1. *The maximum likelihood approach.*
In this example, the samples of the time-series are assumed independent and identically distributed (iid) piece-wise constantly. In other words, for a given family of parametric distribution densities $\{f(\cdot|\theta)\}$*, we have* $\forall i = 1 \dots, n$*:*

$$x^{(i)} \sim \sum_{k=1}^{K^\star} f(\cdot|\theta_k)\mathbb{1}\{t_k^\star \leq k < t_{k+1}^\star\}.$$

One way to learn the parameters of such model, and thus find the change-point, is to perform maximum likelihood. This is equivalent to taking the following cost function:

$$c(\{x^{(i)}\}_{i=t_k+1}^{t_{k+1}}) = -\sup_\theta \sum_{i=t_k+1}^{t_{k+1}} \log f(x^{(i)}|\theta).$$

The previous example is probably one of the most considered frameworks for change-point detection [60, 101, 144]. The piece-wise i.i.d. framework is actually considered in Chapter 4, but with a slightly different cost function.

3.2.1 Quality measures

Here again, assuming the access to the true change-points, many metrics has been proposed to evaluate the quality of segmentation algorithms. Among them, metrics based on those of binary prediction (change-point or not) such as the F_1-score. However, these do not take into account the temporal aspect of the problem and one should therefore prefer Hausdorff's metric. The latter defines the error $h(\mathcal{T}^\star, \mathcal{T})$ of the set of estimated change-points from the real ones as the greatest temporal distance between a change-point and its prediction:

$$h(\mathcal{T}^\star, \mathcal{T}) \triangleq \max\left\{\max_{t^\star\in\mathcal{T}^\star}\min_{t\in\mathcal{T}}|t-t^\star|, \max_{t\in\mathcal{T}}\min_{t^\star\in\mathcal{T}^\star}|t-t^\star|\right\}.$$

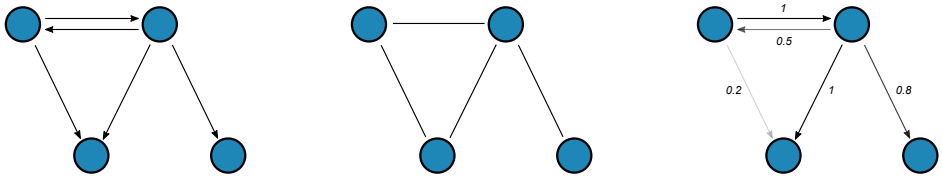
Such metric has the advantage to penalize both over-segmentation and under-segmentation.

While the previous metric evaluates the quality of an algorithm empirically, it is also important to evaluate the quality from a theoretical point of view. This is done with the notion of consistency [157] which states that as soon as the number of samples in each segment tends to infinity, we must have $\mathbb{P}(K = K^\star) \to 0$ and $n^{-1}h(\mathcal{T}^\star, \mathcal{T}) \to 0$ in probability.

3.3 Graph theory and models for graph vectors

3.3.1 Basic definitions

Graphs are mathematical objects describing potentially complex systems via a set of interconnected entities, referred as nodes. They appear in many fields and applications, particularly those involving the notion of networks, such as biological networks, neural networks, sensor networks, computer networks, telecommunication networks, social networks, transportation networks... Graphs are then the most widely used tool to describe and model these networks. In the following, we give basic definitions and concepts of graph theory.



(a) Directed (b) Undirected (c) Weighted

Figure 1.7: Exemples of directed, undirected and weighted graphs with four nodes.

Definition 1.1. (Directed graph.) *A directed graph $G = (\mathcal{V}, \mathcal{E})$ is defined via a finite set of nodes (or vertices) $\mathcal{V} = \{v_1, \dots, v_p\}$ and a set of edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, i.e. pairs of nodes that are considered neighbors. In particular, if $(u, v) \in \mathcal{E}$, we say that u is a parent of v and v is a child of u.*

In the following we do not consider self-edges, meaning that for any node u in $\mathcal{V}$, $(u, u) \notin \mathcal{E}$. Moreover, we may refer to a node $v_i \in \mathcal{V}$ simply by its index i.

Definition 1.2. (Undirected graph.) *An undirected graph $G = (\mathcal{V}, \mathcal{E})$ is a directed graph whose edge set $\mathcal{E}$ is symmetric i.e. $\forall (u, v) \in \mathcal{E}$, $(v, u) \in \mathcal{E}$. In this context, there is no notion of parent nor children and connected nodes are simply referred as neighbors.*

Definition 1.3. (Weighted graph.) *A weighted graph $G = (\mathcal{V}, \mathcal{E})$ is a graph whose edge set $\mathcal{E} = \{(u, v, w_{uv}), u, v \in \mathcal{V}\}$ associates to each edge $(u, v) \in \mathcal{E}$ a weight $w_{uv} \in \mathbb{R}_+$. If the graph is undirected, we have $\forall (u, v) \in \mathcal{E}$, $w_{uv} = w_{vu}$.*

Remark 1.1. *Although largely assumed, the positive assumptions of the weights is not mandatory. When this is not supposed, it will be specified.*

In the following, unless specified, the graphs are assumed undirected.

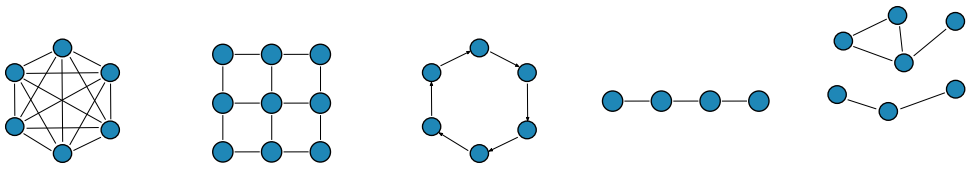
Definition 1.4. (Adjacency matrix.) *Let $G = (\mathcal{V}, \mathcal{E})$ be a graph of size p. Its adjacency matrix $A \in \{0, 1\}^{p \times p}$ is a binary matrix whose entries indicate the presence or absence of edge. $\forall i, j \in \{1, \dots, p\}$:*

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

The adjacency matrix entirely describes its corresponding graph, allowing mathematical manipulations and the formulation of some graph characteristics. For example, the total number of edges of a graph simply corresponds to $||A||_1$ and the *degree* of a node i is $\sum_j A_{ij}$. A is always symmetric for undirected graphs and its generalization to weighted graphs is the *weight matrix* W whose entries corresponds to the edges weights. In the following, the graphs are assumed weighted.

Definition 1.5. (Degree and degree matrix.) *Let $G = (\mathcal{V}, \mathcal{E})$ be a weighted graph of size p with weight matrix W. $\forall i \in \{1, \dots, p\}$, the degree of the node v_i is $d_i = \sum_j W_{ij}$. The degree matrix D of the graph is the diagonal matrix that contains all the nodes degrees.*

Definition 1.6. (Combinatorial graph Laplacian.) *The combinatorial graph Laplacian of a graph G with weight matrix W and degree matrix D is the matrix $L = D - W$.*



(a) Complete (b) Grid (c) Cycle (d) Line (e) Arbitrary

Figure 1.8: Some examples of important graph topologies. The last one (e) admits two connected components.

In the same way as the adjacency matrix, a Laplacian matrix describes entirely its associated graph. In particular, it is known to carry important topological characteristics of the graph and to be linked with spectral graph theory. For example, the number of eigenvalues of L that are equal to zero corresponds to the number of *connected components* of the graph. A connected component being a subset of nodes for which there always exist a path between them, and there exist no path with any other node. An illustration of a graph with two connected components is given in Figure 1.8e.

Definition 1.7. (Graph vector.) *A graph vector, also referred as graph signal or graph function, is a function $x : \mathcal{V} \to \mathbb{R}$ that assigns a real value to all nodes of a graph $G = (\mathcal{V}, \mathcal{E})$. This function can be represented by a simple vector $x \in \mathbb{R}^p$ with x_i the value of x at node v_i.*

This last object allows to define vector data that are observed over a network structure, which is a fundamental aspect of the data that we consider throughout the thesis. Nevertheless, a question remains: what the graph bring to the modeling of this type of vectors. Indeed, taken like that, a graph vector remains a simple vector. In the next sections, we present two different points of view that answer this question. These two frameworks are both considered in the rest of the thesis.

3.3.2 The Graph Signal Processing framework

Graph Signal Processing (GSP) [124, 147] is a relatively recent field. Its aim is to extend most of the tools developed in the field of signal and image processing to graph signals. Thus, notions such as smoothness, sampling or spectral representation of a signal has been extended to cover this type of data. In fact, temporal signal and images are seen as special cases of graph signals where the associated graph corresponds to either a line for a temporal signal or a grid for an image (see Figure 1.8). Within the context of GSP, the graph can now be arbitrary.

In this framework, and similarly to temporal signals or images, the value recorded at a specific node is seen as a shifted version of the values recorded at its neighbors. In the following, we recall some basic notions of GSP and properties assumed to be shared by most of the graph vectors.

Definition 1.8. (Smoothness.) *Let $G = (\mathcal{V}, \mathcal{E})$ be a graph of size p with weight matrix W, L be its Laplacian matrix, and $y \in \mathbb{R}^p$ be a graph signal seen over it. We say that y is s-smooth with respect to the graph G if*

$$y^\top L y = \frac{1}{2} \sum_{i,j \in [p]} w_{ij} \left(y_i - y_j\right)^2 \leq s \ . \tag{1.2}$$

The previous definition provides a notion of smoothness for graph signals. Intuitively, a graph signal y is s-smooth with respect to G if adjacent nodes of the graph carry sufficiently similar signal values. The smaller s is, the smoother the graph signal is. In the GSP framework, graph signals are most of the time assumed smooth with respect to their associated graphs, i.e. with small s values.

Remark 1.2. *The particular case where $s = 0$ implies that all neighboring nodes have the same value.*

Definition 1.9. (Graph Fourier Transform.) *Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph with no self-loops, and $L = X\Lambda X^\mathsf{T}$ be the eigenvalue decomposition of its Laplacian matrix. Then, the Graph Fourier Transform (GFT) of a graph signal $y \in \mathbb{R}^p$ is given by*

$$h = X^\mathsf{T} y,$$

where the components of h are interpreted as Fourier coefficients, the eigenvalues Λ as distinct frequencies, and the eigenvectors X as a decomposition basis.

This definition was initially motivated by the fact that, applied to temporal signals or images, it was recovering the classical Fourier transform. Moreover, it is empirically observed that eigenvectors of X associated to the smallest eigenvalues in Λ are showing less variability across neighboring nodes values than those associated with the biggest, motivating as well the comparison with Fourier analysis.

Definition 1.10. (Spectral sparsity.) *Let $k \in \mathbb{N}^+$, we say that a graph signal y admits a k-sparse spectral representation (equivalently that y is k-bandlimited) with respect to a graph G, if for $h = X^\mathsf{T} y$ we have*

$$\|h\|_0 \leq k \ , \tag{1.3}$$

where $\|h\|_0$ stands for the number of non-zero elements of h.

Regarding this definition, y admits k-sparse spectral representation if the number of non-zero elements in its Fourier coefficients vector is less or equal to k. k-bandlimitedness is the second property assumed to be shared by most graph vectors of the GSP framework. When k is small, this implies that the entire signal can be recovered from a small number of nodes values. Finally, note that if the smoothness property is also assumed, zero-coefficient will have more chance to be associated with large eigenvalues (i.e. high frequencies). This is in accordance with the spectral domain analysis of standard signals where high variability across neighboring nodes values is mostly explained by noise.

3.3.3 A probabilistic framework

In the previous part, graph functions and their corresponding graph was linked together via properties coming from classical signal processing, namely smoothness and bandlimitedness. In the framework presented here, graph vectors are seen as random vectors drawn from a probability distribution known as *Markov Random Fields* (MRF). For this type of probability distribution, the graph encodes a particular dependency structure that is explained in the following.

Definition 1.11. (Conditional independence.) *Let X, Y and Z be three real-valued random variables and denote by $F_{X|Z=z}(x)$ (respectively $F_{Y|Z=z}(y)$) the cumulative distribution function (cdf) of X (respectively Y) knowing $Z = z$. We say that X and Y are conditionally independent on Z, denoted $X \perp\!\!\!\perp Y|Z$ if and only if, $\forall x, y, z$ we have:*

$$F_{X,Y|Z=z}(x, y) = F_{X|Z=z}(x)F_{Y|Z=z}(y).$$

To some extent, the fact that X and Y are conditionally independent with respect to Z tells us that given Z, then knowing X does not bring any information on the likelihood of Y and conversely.

Remark 1.3. *Two conditionally independent variables can be dependent and conversely.*

Definition 1.12. (Markov Random Field.) *Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph and $X = (X_i)_{i\in\mathcal{V}}$ be a random vector whose entries are indexed by the vertices $\mathcal{V}$. We say that X is drawn from a MRF associated to G if the following properties hold:*

(a) $X_u \perp\!\!\!\perp X_v \mid X_{\mathcal{V}\setminus\{u,v\}}$, *for any edge* $(u, v) \notin \mathcal{E}$.

(b) $X_u \perp\!\!\!\perp X_{\mathcal{V}\setminus\mathcal{N}(u)} \mid X_{\mathcal{N}(u)}$, $\forall u \in \mathcal{V}$ *and where* $\mathcal{N}(u) = \{v \in \mathcal{V} : (u, v) \in \mathcal{E}\}$ *is the* neighborhood *of* u.

(c) $X_A \perp\!\!\!\perp X_B \mid X_S$, *for any disjoint subset* $A, B, S \subset \mathcal{V}$ *such that* S *separates* A *and* B *i.e. any path from* A *to* B *(and conversely) passes through* S.

Remark 1.4. *It can be showed [100] that* $(c) \Rightarrow (b) \Rightarrow (a)$. *For certain probability distribution, the reverse is true as well, its notably the case of variables admitting a positive density function.*

Given the previous definition, we understand that building MRFs is not straightforward. For this reason, MRFs are often reduced to the class of probability distribution that *factorizes*, a class of probability distribution that has been shown to validate the three properties required to be an MRF.

Definition 1.13. (Factorization.) *Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph and X be a random vector indexed by V with probability distribution $\mathbb{P}_X(\cdot)$. Recall that a clique is a subset of nodes that are all connected together, we say that $\mathbb{P}_X(\cdot)$ factorizes in G if it is of the form:*

$$\mathbb{P}_X(x) = \frac{1}{Z} \prod_{C\in\mathcal{C}} \psi_C(x_C), \tag{1.4}$$

where $\mathcal{C}$ is the set of all cliques of G, $\psi_C(\cdot)$ are non-negative potential *functions and Z is the normalizing constant.*

The potential functions can be arbitrary and, for a given probability distribution that factorizes, are not necessarily defined uniquely. Famous examples of distributions that factorize, and are therefore MRFs, are the Gaussian Graphical Models and Ising models. The first one is easy to characterize: for any graph G with weight matrix W, the vector $X \sim \mathcal{N}(\cdot, W^{-1})$ factorizes in G. Ising models are considered in Chapter 4 and reminders are made in it.

4 Organization of the thesis

The thesis is organized as follows. Note that each chapter can be read independently.

- Chapter 2: Anomaly detection in communication networks: applications to Sigfox.

  This chapter is essentially dedicated to the resolution of the first objective described in Section 2. The anomaly detection task is taken from the angle of activity monitoring in a communication network. In other words, anomalies are spotted based on e.g. an abnormal number of interactions, amount of exchanged information, number of received signals, etc. This very general framework allows it to be applicable to a large class of communication networks, including Sigfox.

  In the chapter, we first briefly overview the literature on anomaly detection in networks, emphasizing on the graph representation aspects behind it. We also present a simple novelty detection algorithm that aims to detect abnormal levels of communication activity at the level of a node. This algorithm relies on the intuition that the level of activity of a node can be determined or predicted by looking at the level of activity recorded at its neighboring nodes. Thanks to having access to a 'normal' data set and conventional supervised learning methods, the relationship between nodes activity can be learned. Afterwards, an anomaly is detected when the predicted level of activity is far from the real one.

  This method is showed to perform well on both synthetic data and data coming from Sigfox network, which allows us to conclude to the resolution of the first objective. In addition we will see that the presented approach is linked with some other objectives, particularly with the task of graph-based event detection and graph inference that are core-subjects of the thesis and are investigated in the remaining chapters.

- Chapter 3: Structure inference from smooth and bandlimited graph signals.

  In this chapter we consider the problem of learning the underlying structure of a set of graph vectors, i.e. the graph on which they are observed . This chapter is thus linked with the third objective described in Section 2. The graph vectors are assumed to enjoy a sparse representation in the graph spectral domain, a feature which is known to carry information related to the cluster structure of a graph. The signals are also assumed to behave smoothly with respect to the underlying graph structure. For the graph learning problem, we propose a new optimization program to learn the Laplacian of this graph and provide two algorithms to solve it, called IGL-3SR and FGL-3SR. Based on a 3-steps alternating procedure, both algorithms rely on standard minimization methods –such as manifold gradient descent or linear programming– and have lower complexity compared to state-of-the-art algorithms. While IGL-3SR ensures convergence, FGL-3SR acts as a relaxation and is significantly faster since its alternating process relies on multiple closed-form solutions. Both algorithms are evaluated on synthetic and real data.

- Chapter 4: Detecting changes in the graph structure of a varying Ising model.

  This last chapter addresses the demands of the last two objectives defined above. It adopts the probabilistic framework for the modeling of graph vectors and assumes them to be drawn from an Ising model. In particular, the chapter focuses on the

estimation of multiple change-points in a time-varying Ising model that evolves piece-wise constantly. The aim is to identify both the moments at which significant changes occur in the Ising model, as well as the underlying graph structure of each segment of the signal i.e. the part between two change-points. For this purpose, we propose to estimate the neighborhood of each node by maximizing a penalized version of its conditional log-likelihood. The objective of the penalization is twofold: it imposes sparsity in the learned graphs and, thanks to a fused-type penalty, it also enforces them to evolve piece-wise constantly. Using few assumptions, we provide two change-points consistency theorems. Those are the first in the context of unknown number of change-points detection in time-varying Ising model. Finally, experimental results on several synthetic data sets and real-world examples demonstrate the performance of our method.

5 Publications

The work presented in this manuscript has resulted in publications and submissions in international conferences and journals:

- B. Le Bars and A. Kalogeratos, A Probabilistic Framework to Node-level Anomaly Detection in Communication Networks, In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 2188-2196, 2019
- B. Le Bars[1], P. Humbert[1], L. Oudre and A. Kalogeratos, Learning Laplacian Matrix from Bandlimited Graph Signals, In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2937-2941, 2019
- P. Humbert[1] , B. Le Bars[1] , L. Oudre, A. Kalogeratos and N. Vayatis, Learning Laplacian Matrix from Graph Signals with Sparse Spectral Representation, *Submitted to JMLR*, 2020
- B. Le Bars, P. Humbert, A. Kalogeratos and N. Vayatis, Learning the piece-wise constant graph structure of a varying Ising model, In *Proceedings of the* 37^{th} *International Conference on Machine Learning (ICML)*, 2020
- P. Humbert[1], B. Le Bars[1], L. Minvielle[1] and N. Vayatis, Robust Kernel Density Estimation with Median-of-Means principle, *To be submitted, arxiv*, 2020

[1]Authors with equal contribution to the work.

2

Anomaly detection in communication networks: applications to Sigfox

Contents

Abstract

Monitoring the activity in communication networks has become a popular area of research and particular attention has been paid to detection tasks such as spotting anomalies. In this chapter, we briefly overview the literature on this subject, emphasizing on the graph representation aspects behind it. We present a simple supervised learning-based algorithm that aims to detect abnormal level of communication activity in networks like Sigfox. This approach is showed to be linked with the task of graph-based event detection and graph inference, subjects at the core of the thesis in the following chapters.

Associated publication:
A Probabilistic Framework to Node-level Anomaly Detection in Communication Networks [102], Le Bars, Batiste and Kalogeratos, Argyris
Appeared in *IEEE Conference on Computer Communications (INFOCOM), 2019.*

1 Detecting anomalies in networks: a graph perspective

1.1 Introduction

Over the past years, thanks to an increasing availability of data and tools to analyze them, detecting *anomalies* in communication networks has become an important area of research. This task, which looks for events that deviates from the normal network behavior, arises in many network related problems such as monitoring and security [6, 155].

In the present chapter, we refer to *communication network*, or simply *network*, as a set of entities that can interact or exchange information between each other. This simple formulation allows to cover a large spectrum of networks such as computer networks [82, 174], e-mail networks [167], telecommunication network [102] or social networks [77, 87]. However, with such a wide range of networks types, notions of interaction and anomaly may differ a lot. For example, in computer or wireless networks, an interaction refer to bits of information that pass from one node to another (possibly via other nodes) and an anomaly can refer to a network attack. In an e-mail or a social network, the interaction are directly node-to-node and an anomaly can refer to account hacks for example.

The standard way to detect anomalies in networks [6] is to, first, preprocess the available data (data engineering step) and then apply standard anomaly detection techniques, such as Local Outlier Factor [25] or One-Class SVM (see the background in Chapter 1), over those preprocessed data. The way those two steps are performed may depend on the labeling of the data (e.g. with or without labeled anomalies), the kind of available data (e.g. the content of the exchanged information: images, text...), and for example, the ability or not to build a statistical model from them. We can see from this simple procedure that the most important part is the preprocessing step, which obviously depends heavily on the type of considered network, the content of the exchanged information and the type of anomalies we aim to discover. In the next section, we consider a framework to describe and preprocess the data which is adapted to many different situation and which requires data usually available at low cost.

1.2 Representing the network activity via dynamic graphs

An intuitive way to represent a communication event involving a set of entities in a network is via a graph. In such a graph, the nodes correspond to all the entities of the network and the edges indicate the structure of the event i.e which node directly interacted with which other node. Taking Sigfox network as an example, a message sent from a device to a set of surrounding Base Station, can be represented by a graph with edges connecting the involved BSs to the device, or simply connecting the BSs together if we omit the device (see figure 2.1). With this formulation, the stream of communication events generates a graph evolving over time, with edges appearing and disappearing every time a communication event begins or finishes [99].

In practice, analyzing such a stream of interactions can be difficult and a standard approach is to analyze the network activity via a time-discretized series of graphs [38]. In other words, the time is discretized (over a daily basis for example) and at each timestamp, a graph aggregating all the interactions that occurred at the corresponding period of time is built. Usually, the weights of the graph edges counts the number of interactions that occurred (or the amount of data shared) between two nodes over the specific timestamp (see Figure 2.2).

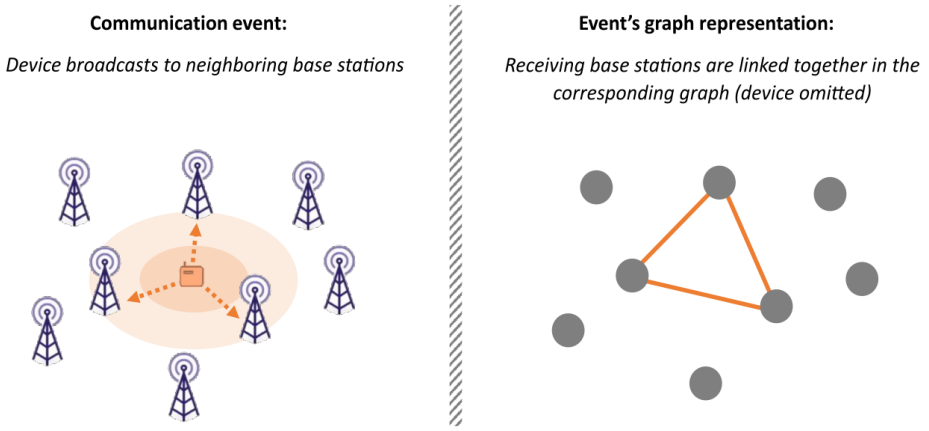


Figure 2.1: A communication event at Sigfox and its graph representation. Only the Base Stations (BSs) are considered in the graph.

In addition to being adapted to many communication networks, the use of this graph-representation is motivated by the fact that, in reality, content-specific features of the exchanged information are usually kept undisclosed so as to preserve privacy (e.g the content of the e-mail). Furthermore, it captures the structure of the network such as clusters or isolated nodes. Consequently, most studies on network anomaly detection only deal with this time-series of graphs representation of their data to build the features [38, 77, 130, 167] and can thus rely to the vast literature on dynamic graph anomaly detection, briefly recalled is the next section. Although adapted to many kinds of networks, this simple data-representation has nevertheless some limitations. First, it does not allow to recover every types of anomalies, but mostly those related to the communication activity and the network structure. Furthermore, the aggregated representation loses information about interactions involving more than two nodes (e.g. multiple receivers) since edges keep in track only node-to-node interactions.

1.3 Recall on the anomaly detection for dynamic graphs

The task of anomaly detection in a time-series of graphs, referred as dynamic graph [131] is closely related to the one of classical anomaly detection for temporal data. Similarly, the goal is to find a subset of graphs that deviates from the majority. The standard approach consists in computing several graph features (e.g. degree of a node, edges weights, centrality measures, shortest path etc.) over each graph, and then apply classical anomaly detection techniques on the derived set of vector data [28, 32, 72]. The methods we can find in the literature differ in the same way as in the classical case. They vary notably according to the availability of the labels i.e supervised, semi-supervised (access to a dataset of *normal* data) [46, 140, 141] or unsupervised (no label available) [25, 82], the type of used method i.e probabilistic model-based [4, 38, 77, 121, 129, 130, 170], distance based [25] etc.

Apart from the graph-based feature engineering, the dynamic graph's anomaly detection also differ from the classical case in the notion of scale of abnormality. In fact, the anomaly

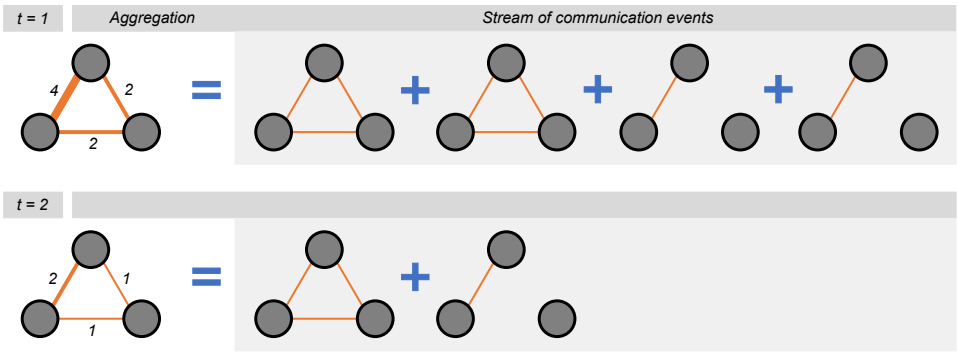


Figure 2.2: Aggregated representation of a stream of communication events over two time intervals.

can be spotted at the level of a node [77, 87], a subgraph [121], or the whole graph-level [38]. For a complete survey on dynamic graph's anomaly detection, the reader should refer to the survey of [131].

2 A regression-based novelty detector

In this section, we describe an intuitive way to detect abnormal level of communication activity recorded at the level of a node. The proposed approach rely on the intuition that networks often has an underlying structure with clusters and neighborhood, meaning that a node usually communicates with a certain subset of other nodes. Based on this intuition, the idea of the presented approach is to consider the communication activity recorded at a node as abnormal by looking at its neighbors activity. More precisely, we assume that it is possible to predict the activity of a node, based on the level of activity of the remaining nodes. Before going into further details, let define our model and the objectives of the detection task.

2.1 Model description

We consider a communication network with N entities, referred as nodes, that can interact between each other. In this network, we assume that a *communication event* can occur at any moment, and can be shared by many nodes at the same time.

Definition 2.1. *(Communication event): A communication event is entirely described by an N-dimensional binary vector $X = (X_1, \dots, X_N) \in \{0,1\}^N$, referred as* fingerprint, *indicating which node took part in the event. $X_j = 1$ if node j is involved in the event, and 0 otherwise.*

With such a definition of communication event, all the involved nodes are assumed to communicate equally between each others, with no notion of roles (e.g. sender or receiver). From a graph point of view, a communication event thus creates a *clique* containing all the

involved nodes.

Since no notion of roles nor protocol are assumed, many examples of communication networks would fit this simple setting. At Sigfox for example, each Base Station corresponds to a node, a message corresponds to a communication event, where the fingerprint X indicates which BS has received the message ($X_j = 1$ means that the j-th BS has received it). We can also think of an e-mail network, where each node corresponds to an e-mail address, an e-mail corresponds to an event, where the corresponding fingerprint indicates which email addresses received it. Last but not least, the network can correspond to a company where each node is an employee and a communication event corresponds to a meeting.

Assumption 2.1. *An event X corresponds to a random vector of size N with probability distribution denoted $\mathbb{P}_X$. Given a set of communication events $\{X^{(1)}, \dots, X^{(n)}\}$, we assume that the communication events are all independent (but not necessarily identically distributed).*

Definition 2.2. *(Conditional probability function): Let $X_{\setminus j}$ be the fingerprint of the event X that indicates the participation of all nodes except node j. Let $x_{\setminus j}$ be a realization of $X_{\setminus j}$ and denote $\eta_j^*(x_{\setminus j})$ the probability that node j participates in the event X, provided the fingerprint $x_{\setminus j}$:*

$$\begin{aligned} \eta_j^*(x_{\setminus j}) &\triangleq \mathbb{P}(X_j = 1 | X_{\setminus j} = x_{\setminus j}) \\ &= \mathbb{E}\left[X_j | X_{\setminus j} = x_{\setminus j}\right]. \end{aligned} \tag{2.1}$$

Knowing the fingerprint over all the other nodes allows us to express the behavior of node j as a Bernoulli random variable:

$$X_j \sim \mathcal{B}\left(\eta_j^*(x_{\setminus j})\right). \tag{2.2}$$

Assumption 2.2. *All the considered communication events have the same conditional probability functions. In other word, let $\{X^{(1)}, \dots, X^{(n)}\}$ be any set of communication event, then, for any node j and any fixed fingerprint $x_{\setminus j} \in \{0,1\}^{N-1}$, we have $\forall i = [n]$, $X_j^{(i)} \overset{iid}{\sim} \mathcal{B}\left(\eta_j^*(x_{\setminus j})\right)$.*

The previous assumption states that even if the events are non identically distributed, the conditional probability function, and thus the dependency structure between the nodes, is the same for each event. The aforementioned assumption can be understood easily for Sigfox. Since the network is constantly evolving, with the appearance of new devices, at different location, the joint probability distribution of an event can change easily. However, the BS are spatially distributed on earth and the fact that a BS receives or not a message, highly depends on the fact that its neighboring BS has received it or not. Since their positions are fixed, one can easily imagine that the dependency structure this spatial proximity implies will not change.

2.2 Objective and ideal scoring function

Let $\mathcal{D}_n = \{X^{(i)}\}_{i=1}^n$ be a set of n communication events. Let $M_j^n = \sum_{i=1}^n X_j^{(i)}$ be the random variable of the number of events recorded at a node $j \in [N]$. Our objective is, for a fixed node $j \in [N]$, to determine if the observed volume of events in which that node participates, denoted $m_j^n = \sum_{i=1}^n x_j^{(i)}$, is abnormal. Assuming the access to all the probability distributions, a way to perform this task is to provide confidence levels for M_j^n and look whenever m_j^n stands in a region of low probability.

In the present approach, we propose to use the knowledge brought by the fingerprints of the other nodes and build these confidence levels knowing them. Thus, an anomaly will be detected if, regarding the communication activity recorded at every node but j, the number of event the node j has participated in is too low are high. Below we define our ideal scoring function, assuming the access to the underlying probabilistic model.

Definition 2.3. *The conditional cumulative distribution function of M_j^n, knowing the fingerprints $\{x_{\setminus j}^{(i)}\}_{i=1}^n$ is denoted by:*

$$F_{M_j^n}(m) = \mathbb{P}\left(M_j^n \leq m \;\middle|\; \forall i = 1, \dots, n, \quad X_{\setminus j}^{(i)} = x_{\setminus j}^{(i)} \right) \tag{2.3}$$

Knowing the fingerprints of the other nodes, M_j^n corresponds to a sum of n Bernoulli random variables, with parameter $p_j^{(i)} \triangleq \eta_j^*(x_{\setminus j}^{(i)})$ for $i \in [n]$. This distribution is called Poisson Binomial [80, 152] and its cumulative distribution function (cdf) has a closed-form solution given by:

$$F_{M_j^n}(m) = \sum_{k=1}^{m} \sum_{A \subset [n], |A|=k} \left(\prod_{i \in A} p_j^{(i)} \prod_{l \in A^{\mathsf{c}}} (1 - p_j^{(l)}) \right) \tag{2.4}$$

This function can be computed efficiently via the method presented in [80], it is based on the computation of the Discrete Fourier Transform of the characteristic function of the Poisson Binomial distribution. We refer by $\texttt{PoiBin}(m; p_j^{(1)}, \dots, p_j^{(n)})$ the computation of such function using their algorithm. Assuming the knowledge of the parameters $p_j^{(i)}$, we can propose the following anomaly scoring function.

Definition 2.4. *(Anomaly scoring function): Under the previous model, we define the anomaly scoring function of the volume of events recorded at node j by:*

$$s(m_j^n) = \max\left\{ F_{M_j^n}(m_j^n), 1 - F_{M_j^n}(m_j^n) \right\} \tag{2.5}$$

Such scoring function is in $[0, 1]$ and the closer to 1 is $s(m_j^n)$, the more m_j^n is considered abnormal. Indeed, when the first term in the $\max$ of equation 2.5 is close to one, it means that m_j^n stands in the right-hand tail of the distribution of M_j^n, which encode an abnormally high number of recorded events at node j. On the opposite, a high value of the second term will encode an abnormally low number of observed communication events.

2.3 A supervised learning framework

In practice, we do not have access to the true conditional probability function $\eta_j^*(\cdot)$ and we propose to estimate it in order to build a proper scoring function. To this end, we suppose we have access to a training set of communication events $\mathcal{D}_{train}$, assumed to have been recorded at times of normal communication behavior. With our definition of $\eta_j^*(\cdot)$ (Definition 2.2), the estimation problem refers to the task of estimating a conditional probability function. Under our specific modeling, $\eta^\star$ corresponds to the conditional expectation of X_j, knowing the value of $X_{\setminus j}$ and thus can be seen as the function regressing X_j using $X_{\setminus j}$. This regression function can be learned using the healthy dataset $\mathcal{D}_{train}$ and any regression algorithm outputting values between 0 and 1 (e.g logistic regression, binary trees, random forest etc.). Such regression algorithm is referred as `Regressor` in algorithm 2.1, it takes as input a set of observed variables $\{x^{(i)}, y^{(i)}\}$ where x stands for the explanatory variable and y for the target one and outputs the corresponding regression function.

Let $\widehat{\eta}_j(\cdot)$ be our estimated regression function of X_j given $X_{\setminus j}$. To build our anomaly scoring function, we propose to simply replace $p_j^{(i)}$ by $\widehat{p}_j^{(i)} \triangleq \widehat{\eta}_j(x_{\setminus j}^{(i)})$ in the conditional cumulative distribution function (cdf) of Eq. 2.4 and replace this new version of conditional cdf in Eq. 2.5. In practice, outputting a score is not enough to spot an anomaly, and one has to fix a threshold s above which the score will be considered abnormal. While this threshold is not always easy to fix in practice, the fact that our scoring function belongs to $[0, 1]$ and has a probabilistic interpretation allows us to fix it intuitively. Indeed, if the conditional cdf is well estimated, our scoring function allows to control the false positive rate: fixing the threshold to $s \in [0, 1]$ will result in a false positive rate of $1 - s$. Thus, relying to the standard threshold of statistical test theory, it can fixed to e.g 0.95 or 0.99.

In practice, the $\eta_j^\star$ function may not be perfectly estimated. Another way to fix the threshold and control the false positive rate is via cross-validation on the training set of communication events. To do so, one has to fix an acceptable false positive rate (e.g 0.05 or 0.01), then via cross-validation find the value of threshold s that generates a false positive rate close to that fixed value.

2.4 Summary

To sum up, we observe an healthy set of communication events $\mathcal{D}_{train}$. For any fixed node j, we learn the probability that this node will take part in an event, given the fingerprint of the other nodes. This is done using $\mathcal{D}_{train}$ and any regression algorithm. We observe a new set of communication events, $\mathcal{D}_n$, for which we want to know if the number of events recorded at node j is abnormally low or high. We evaluate for each event in $\mathcal{D}_n$ the probability that j will take part in the corresponding event and evaluate the anomaly score as described upper. If the score is above a certain threshold, the anomaly is spotted. The overall procedure is summarized in Algorithm 2.1.

Algorithm 2.1 Node-wise network anomaly detection

Input: $\mathcal{D}_{train}$, $\mathcal{D}_n$, node j, threshold s
Regression algorithm: $\texttt{Regressor}(\cdot)$
Output: 1 if anomaly, 0, otherwise

$\hat{\eta} \longleftarrow \texttt{Regressor}\Big(\mathcal{D}_{train} = \{\tilde{x}^{(i)}_{\setminus j}, \tilde{x}^{(i)}_j\}\Big)$
for $i = 1 \ldots, n$ **do**
  $\hat{p}_i \longleftarrow \hat{\eta}(x^{(i)}_{\setminus j})$
end for
$\hat{F} \longleftarrow \texttt{PoiBin}\Big(\sum x^{(i)}_j; \hat{p}^{(1)}_j, \ldots, \hat{p}^{(n)}_j\Big)$ - see Eq. 2.4
$\hat{s} \longleftarrow \max(\hat{F}, 1 - \hat{F})$
if $\hat{s} > s$ **then**
  Output 1: Abnormal node
else
  Output 0: Normal node
end if

3 Applications

In the present section we propose to evaluate the performance of the regression-based approach presented in the previous section. The evaluation is performed over two datasets. The first one is simulated in order to imitate the communication activity recorded in network like Sigfox and the second one is made of real-word communication events recorded at Sigfox. These experiments aim to show the importance of considering the activity recorded at other nodes while performing anomaly detection at a specific node. In addition, the objective is to show the superiority of the regression-based approach compared to the methods presented in the following paragraph.

Comparative methods We compare our regression-based approach with three other methods related to the graph-based algorithms presented in Section 1. All three methods split the training dataset in order to build a set of graphs on which features are engineered. Each graph encodes the communication activity of the network over the different sub-dataset and are build in the same manner: the edges count the number of shared events between two nodes and a node's value captures the total number of events it has participated in. Similarly, a unique graph is build over the dataset for which we want to known if it is abnormal or not. In practice, the datasets used to build the graphs can be of different sizes and the different values in the graphs are normalized by the number of events contained in each dataset.

The comparative methods works the same way. A feature vector aiming to characterize the behavior of the node of interest is build over each graph. Then, a One-class SVM [140] (see chapter 1) builds an hyperplane separating normal feature vectors and abnormal ones using the set of normal vectors build from the training set of graphs. The anomaly score of the graph of interest simply corresponds to the distance of the corresponding feature vector to the separating hyperplane.

The three methods only differ in the feature vector build from each graph. The first one only looks at the value of the node of interest i.e the number of events it has participated in. The second one calculates the weighted degree of the node of interest. The last one

evaluates the difference between the value of the node of interest and the value of its nearest neighboring nodes. The latter is motivated by the intuition that some nodes will have highly correlated variations of activity. It is actually the only multivariate feature vector. The nearest neighbors are selected by looking at the nodes with whom the node of interest shares more events in the training dataset.

Computational remarks All the following experiments has been implemented in Python and performed on a personal laptop computer. The conditional probability functions defined in 2.1 are learned via a Random Forest Regressor [24], using the version implemented in scikit-learn [128]. Same for the One-class SVM algorithm needed in the comparative methods, also implemented in scikit-learn. For both algorithms, the scikit-learn's by default set of hyperparameters are used.

To build the scoring function of equation 2.5 we have to be able to compute the cumulative distribution function of a Poisson Binomial distribution (equation 2.4). This can be done efficiently via the method presented in [80], implemented in the Python package PoiBin[1].

3.1 Simulated experiment

As stated upper, the goal of the following experiments is to apply the different approaches on datasets that simulate the communication activity of a network like Sigfox. We keep the nodes two-dimensional spatial arrangement of Sigfox network and propose the following simulation process.

3.1.1 Simulation process

Sampling the spatial network structure. Draw N node's two-dimensional locations (i.e. analogous to BSs) according to a mixture model $\mathcal{M}$ of K bivariate Gaussian distributions. The mixture model allows to simulate clusters of nodes, which is observed in particular at Sigfox (e.g one city corresponds to a cluster of BS).

Sampling a communication event. First generate an event location ℓ (analogous to a device's location) drawn from $\mathcal{M}$, as in the previous step. Then for each node $v \in [N]$, let its location x_v and draw a Bernoulli with a parameter inversely proportional to the distance $d(x_v, \ell)$. Here, the Bernoulli indicates if the node has participated or not in the event. In our experiments, we set the Bernoulli parameters to be equal to $\exp(-\frac{1}{\sigma_v} d(x_v, \ell))$, where σ_v is a *node-dependent visibility* parameter that controls how much the node v participates in the different events. With such modeling we can see that the closer the location of the event is to a node, the higher is the chance for the node to participate in the event. Furthermore, nearby nodes will have more chance to participates in the same events, which is often observed at Sigfox.

Simulating a normal dataset. Draw n independent fingerprints following the process described in the previous paragraph.

[1] https://github.com/tsakim/poibin

Simulating an abnormal dataset. One way to simulate an anomaly at the level of a node v is to make this node participating in less (or more) events than usual. One way to do it is by changing its visibility parameter σ_v. The more different this parameter is from the normal network behavior, the easier the anomaly will be spotted. Then, the dataset is drawn the same way as a normal one.

3.1.2 Three levels of abnormality

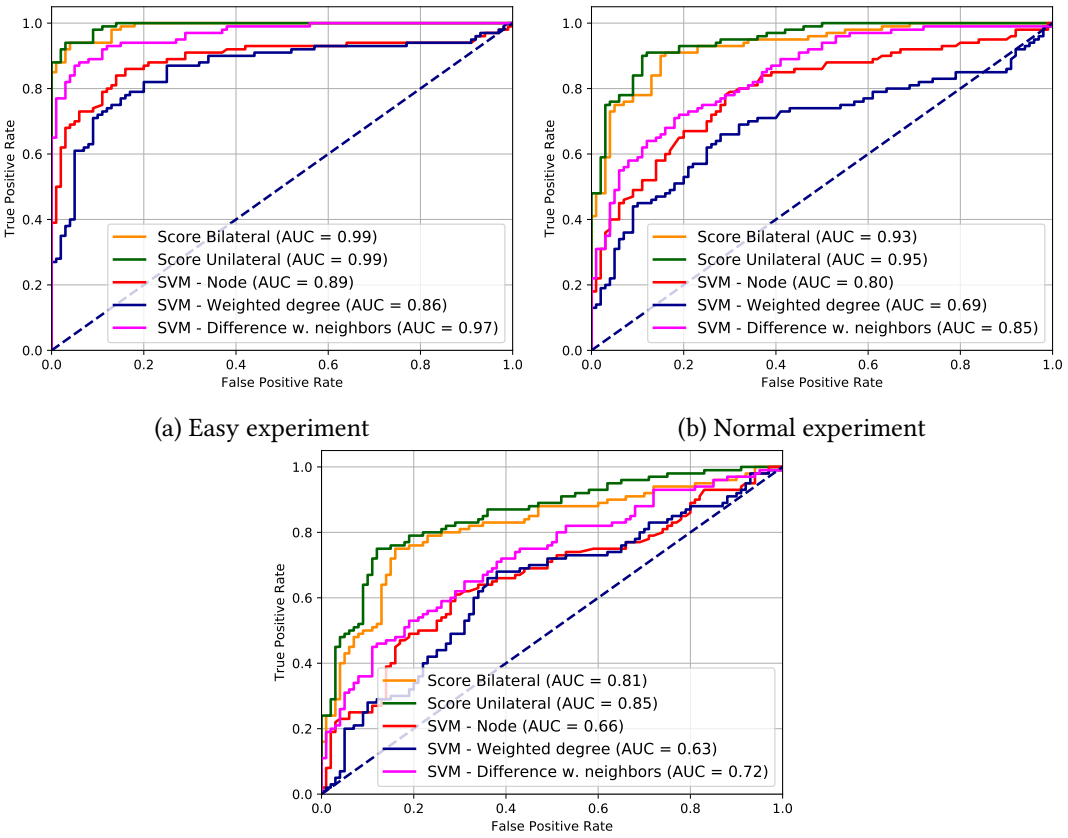
In order to demonstrate the performance of our method, we apply the above generative process in three different situations with different anomaly 'complexity', whereas sharing the following properties:

- $N = 100$ communication nodes are drawn from a mixture of $K = 10$ bivariate Gaussian variables with equal mixture's weights. Each center are drawn uniformly over the square $[-5, 5] \times [-5, 5]$ and the covariance matrices always correspond to the identity matrix of size two.
- The training set contains 20000 communication events. It is equally split in 100 blocks of 200 events for the comparatives methods. The visibility parameters used for each node is the same and is equal to 2.6.
- 200 testing dataset are sampled. Half of them are normal and sampled according to the same process as the training dataset, the other half is abnormal. Each testing dataset have different size where the number of events is drawn from a uniform distribution between 100 and 300.
- A single arbitrary node is chosen to be anomalous and it is the same for each abnormal testing dataset. The anomaly is simulated by a decreasing in the visibility parameter. This is in accordance with the idea that an anomaly at the level of a Base Station implies a decreasing in the number of observed signals.

The three experiments thus only differ in the decreasing of the visibility parameter. The first experiment, referred as *easy* imposes a important decreasing and replace the visibility parameter of the corresponding node by 1.3, the second one, referred as *normal*, replace it by 1.62 and the last one, referred as *hard* replace it by 1.95.

3.1.3 Results

The empirical performances of the different approaches is visualized on the ROC curves of figure 2.3. The regression-based approach is evaluated via two scoring functions: a *Bilateral* and a *Unilateral*. The Bilateral corresponds to the one defined in the previous section (equation 2.5). However, the Unilateral score only evaluates the right-hand element in the maximum of equation 2.5. Thus, it only looks for abnormally low level of the node's events participation: the bigger the score is, the more the number of event the node has participated in stands in the left-hand tail of the Poisson Binomial distribution. The use of the Unilateral score is motivated by the fact that the anomalies we simulate imply a decreasing in the number of event's participation. Using it should improve the rate of false positive. The number of neighbors necessary to compute the third SVM-based algorithm has been selected to be the one with best performances and was fixed at 7.



(a) Easy experiment

(b) Normal experiment

(c) Hard experiment

Figure 2.3: ROC curves and their respective AUC obtained over the simulated experiments. Each figure corresponds to a different level of complexity, from the simplest (a) to the hardest (c).

Several conclusions can be made from the ROC curves. First of all, no matter the complexity of the experiment, the regression-based method always outperforms the One-class SVM-based approaches. Indeed, the ROC curves associated to the firsts are always above the latter, leading to better Area Under the Curve (AUC). Moreover, we must note that as expected, the Unilateral score is slightly better than the Bilateral.

Among the three SVM-based approaches, several observation must also be made. The method looking at the weighted degree appears to be the worst, being even inferior to the simplest approach which is based on the node's value. This illustrates an important point: building naive features, even graph-based, won't solve an anomaly detection problem like this one if they are not adapted to it. On the contrary, the one with the most sophisticate features, which looks at the difference between the node and its nearest neighbors, outperforms the other two. This illustrates the fact that considering the other nodes values is important while performing the detection task. Moreover, it shows that being able to select only a subset of important nodes (here 7), referred as neighbors, improves again the performance.

Finally, it appears that our method is slightly more robust to the increasing complexity of the different experiments. Indeed, we observe a decreasing of 0.04 (respectively 0.14)

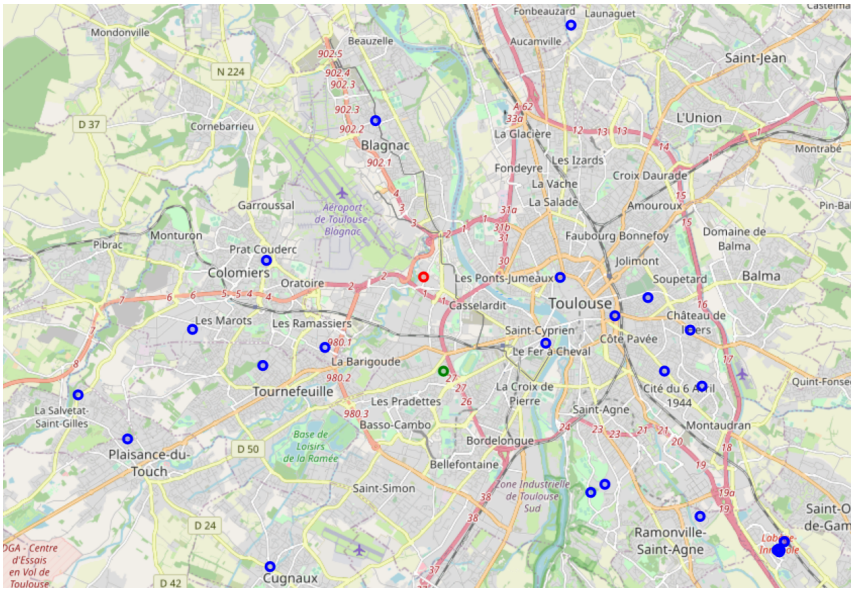


Figure 2.4: Positions of the 34 Base Stations. In red, the failing BS, in green, the one taken as reference and in blue the others. Note that some BS have very close locations, making them appear as a single one over the figure.

between the AUC of the easy and the normal (respectively hard) experiments for the Unilateral score. This decreasing is more important for the "difference with neighbors"'s One-class SVM, with a decreasing of 0.12 (respectively 0.25) between the AUC of the easy and the normal (respectively hard) experiments.

3.2 Sigfox application

We now propose to verify that the conclusions we made from the previous experiments remain valid in a real-world application. In particular, we apply the different methods to a set of communication events recorded at Sigfox.

3.2.1 Dataset description

In the present experiment, we have access to a set of approximately 232000 Sigfox's messages recorded at the level of $N = 34$ Base Stations over a period of 5 months (from January to June of 2017). Relying to the framework described in Section 2, each BS characterizes a node and each message corresponds to a unique communication event, where the fingerprint indicates which BS has received the message. All the BSs are located near Toulouse, France and their positions can be found in Figure 2.4. Among those BSs, one has been spotted as abnormal by Sigfox's experts (in red on Figure 2.4), approximately from march, while the others has been considered working well.

The objective of the experiment is to learn the normal behavior of the red BS over the month of January (approx. 35000 messages) and then tell if its behavior is abnormal over the remaining period. The same way it would be done in a realistic situation at Sigfox, the prediction is made daily over the remaining 4 months, resulting in 120 test dataset and therefore predictions. Over each day (i.e dataset) of the testing period, approximately

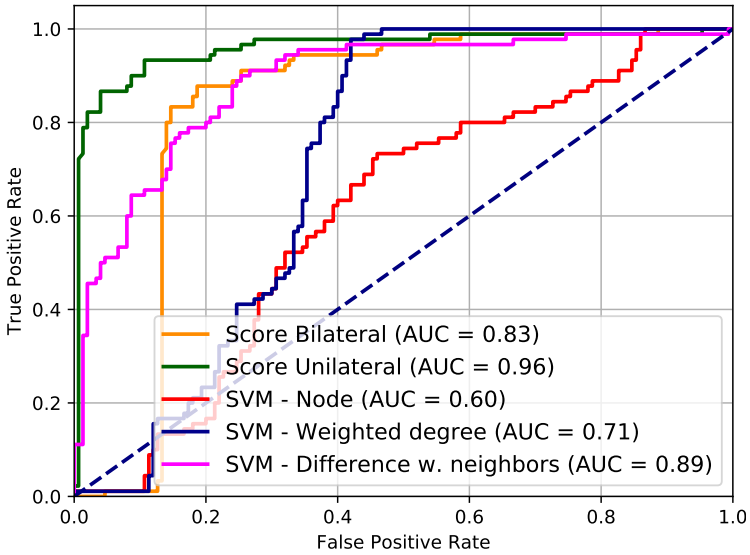


Figure 2.5: ROC curves and their respective AUC obtained over the real-world experiment.

1600 messages are observed in average, but this number can highly vary from one day to another.

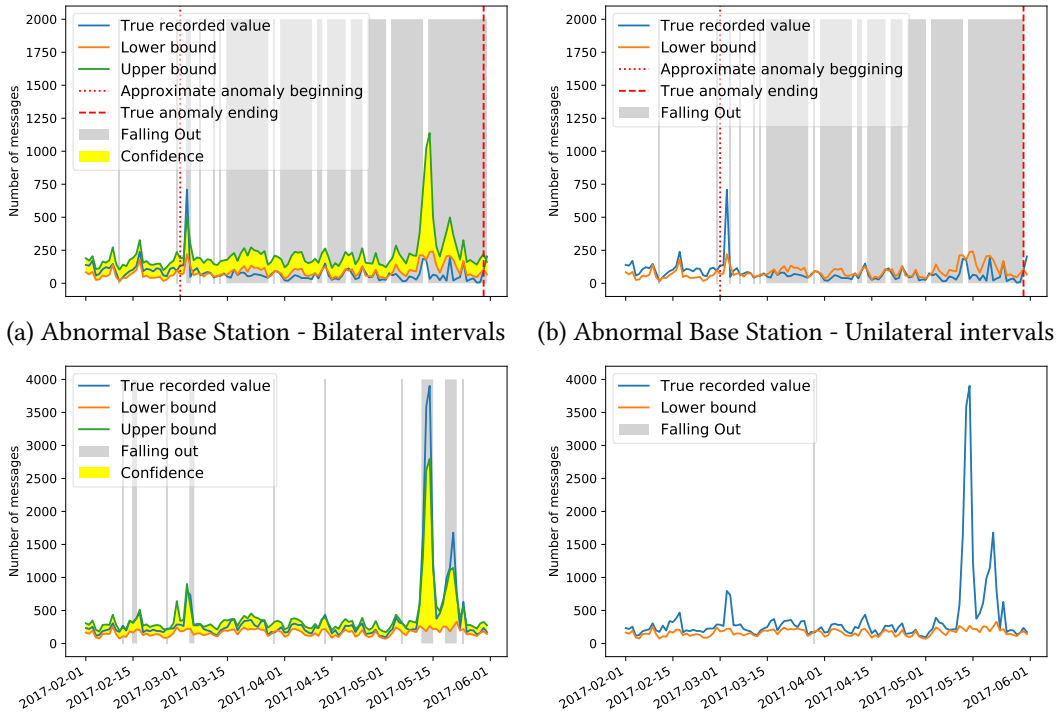
As stated upper, the red BS is considered abnormal by the experts approximately from March. This results in 90 abnormal testing dataset and 30 normal ones. To increase a bit the size of the experiment and make the classes more balanced, we also apply the exact same learning and testing process for a normal BS (in green on Figure 2.4). In the end, 240 test dataset are evaluated as abnormal or not, including 90 days considered abnormal and 150 normal ones.

Finally, to be able to apply the comparative methods, the learning dataset is split according to the daily basis of the testing period. Hence, 31 graphs and consequently 31 feature vectors are build to learn the normal behavior of a node over a day. For each built graph, the nodes carry the number of messages the associated BS has received during the corresponding day and the edges encode the number of messages received in common between two BSs.

3.2.2 Results

First results and comparison between the different methods are evaluated via the ROC curves in Figure 2.5. Several conclusions we made from the experiments on simulated data can be made as well. First, the regression-based method with unilateral score clearly outperforms the others, illustrating its superiority for this type of tasks. Moreover, among the SVM-based approaches, the one that looks at the difference between the node and its nearest neighbors (all the other BS were taken this time), clearly outperforms the other two. We can even observe that this method is slightly better than the *Bilateral* score with an AUC of 0.89 for itself and an AUC of 0.84 for the latter. According to the ROC curves, this seems to come from the fact that the *Bilateral* method gives an important anomaly score to high value of received messages.

To visualize a bit better the performance of the regression-based algorithm, we propose



(a) Abnormal Base Station - Bilateral intervals

(b) Abnormal Base Station - Unilateral intervals

(c) Normal Base Station - Bilateral intervals

(d) Normal Base Station - Unilateral intervals

Figure 2.6: Number of messages received by the corresponding BS, the confidence intervals for a fixed threshold and the timestamps detected to be anomalous.

to fix a threshold above which the score is considered abnormal, plot the "confidence intervals" associated to it and look whenever the number of messages received by the BS of interest lays out of them (Figure 2.6). In other word, for each day of the testing period, we look for the interval of values for which the score is not exceeded. If the number of messages received by the antenna does not belong to these values, the anomaly is detected. The threshold is fix via cross validation on the training dataset, in order to obtain a false positive rate of 0.01 over it.

While the number of messages received by the green BS should never go out the intervals, the red one should have an exceeding score for the abnormal days, indicated between the two red lines of Figure 2.6a and 2.6b. Those figures tells us several things. The high amount of false positive appearing in the *Bilateral* method were essentially coming from high values of received messages by the normal BS and are corrected by using the *Unilateral* approach (Figure 2.6d). We also observe that the total number of messages received by a BS, and therefore the total number of messages sent in the network, can highly variate: this seems to be a reason of the high false positive rate of the Bilateral method and may indicate that the learning phase was not including enough examples. Finally, looking at how regularly the number of messages received by the abnormal BS goes out the intervals, we could ask ourselves if the anomaly began a bit later, around March the 15th.

Overall, we can conclude that the proposed regression-based method is adapted the this kind of problem.

4 Conclusion and discussion

In conclusion, this chapter has enabled us to propose a simple and effective method for detecting anomalies in a communication network. The first objective we had set ourselves has thus been fulfilled, and in particular the objective of the industrial collaboration with Sigfox. Some links can also be made with the other objectives, i.e. the problem of detecting events in a set of graph signals and the graph inference task. Indeed, for the first, characterizing the anomaly of a node based on the values at the other nodes implies a notion of structure underlying the data, implicitly used by the considered approach. Therefore, one could imagine this type of anomaly detection for other types of graph vector data, other than that of a communication network. Also, we have seen that the graph can be used to determine the neighborhood of a node and thus improve the performance of certain methods. This is notably the case of the last approach based on the one-class SVM, but also ours. Indeed, the graph can be used for dimensionality reduction. Therefore, instead of using all the other nodes during the learning phase, one could only use the neighbors of the considered node.

Finally, a link can also be made with the task of graph inference. With an appropriate regression algorithm, the learning step of the presented approach can be seen as the inference of the underlying neighborhood of node j. Indeed, in some sense, we are looking for the nodes that interacts often with j. In particular, a ℓ_1-regularized logistic regression would perfectly match the Ising model structure inference presented by [132]. With such approach, the probability function $\eta_j^\star$ is modeled via a sigmoid of a linear combination of the other nodes fingerprints. Thus, each weights is related to a node and the bigger a weight is, the stronger the interaction between j and the corresponding node is (see chapter 4 for more details). The task of graph inference is properly investigated in the next chapter.

3

Structure inference from smooth and bandlimited graph signals

Contents

Abstract

In this chapter, we consider the problem of learning a graph structure from multivariate signals, known as *graph signals*. Such signals are multivariate observations carrying measurements corresponding to the nodes of an unknown graph, which we desire to infer. They are assumed to enjoy a sparse representation in the graph spectral domain, a feature which is known to carry information related to the cluster structure of a graph. The signals are also assumed to behave smoothly with respect to the underlying graph structure. For the graph learning problem, we propose a new optimization program to learn the Laplacian of this graph and provide two algorithms to solve it, called IGL-3SR and FGL-3SR. Based on a 3-steps alternating procedure, both algorithms rely on standard minimization methods –such as manifold gradient descent or linear programming– and have lower complexity compared to state-of-the-art algorithms. While IGL-3SR ensures convergence, FGL-3SR acts as a relaxation and is significantly faster since its alternating process relies on multiple closed-form solutions. Both algorithms are evaluated on synthetic and real data. They are shown to perform as good or better than their competitors in terms of both numerical performance and scalability. Finally, we present a probabilistic interpretation of the optimization program as a Factor Analysis Model.

Associated publications:
Learning Laplacian Matrix from Bandlimited Graph Signals [103],
Le Bars, Batiste*, Humbert, Pierre*, Oudre, Laurent and Kalogeratos, Argyris
Accepted in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*
* Authors with equal contribution to this work

Learning Laplacian Matrix from Graph Signals with Sparse Spectral Representation [84],
Humbert, Pierre*, Le Bars, Batiste*, Oudre, Laurent, Kalogeratos, Argyris, and Vayatis Nicolas
Submitted to *The Journal of Machine Learning Research (JMLR).*
* Authors with equal contribution to this work.

1 Introduction

Hidden structures in multivariate or multimodal signals can be captured through the notion of *graph*. The availability of such a graph is a core assumption in many computational tasks such as spectral clustering, semi-supervised learning, graph signal processing, etc. However, in most situations, no natural graph can be derived or defined and the underlying graph must be inferred from available data. This task, often referred to as *graph learning*, has received significant attention in fields such as machine learning, signal processing, biology , meteorology, and others [61, 78, 173].

Learning a graph is an ill-posed problem as several graphs can explain the same set of observations. Previous works have been devoted to introducing underlying models or constraints that would narrow down the range of possible solutions. For instance, physical constraints can be imposed to suggest epidemic models or other information propagation and interaction models [52, 69, 137]. From a statistical perspective, the graph learning task is seen as the estimation of the parameters of a certain probability distribution parametrized by the graph itself. Generally, the assumed class of distributions is either a *Bayesian Network* in the case of directed graph, or a *Markov Random Field* for undirected graphs

[97, 153, 171, 176]. Hence, the graph structure encompasses the conditional dependencies between variables. Two variables will be connected in the graph if they are dependent conditionally on all the other variables. In the particular case of Gaussian Random Fields, the graph estimation consists in the estimation of the inverse covariance matrix, known as the *precision matrix* [16, 61]. In the latter reference, the proposed estimation method corresponds to the well-known Graph-Lasso algorithm, which relies on the assumption that the precision matrix is subject to a sparsity constraint.

More recently, *Graph Signal Processing* (GSP) [49, 147], has generalized the standard concepts and tools of signal processing to multivariate signals recorded over graph structures. Notions such as smoothness, sampling, filtering, etc., have been adapted to this framework, opening a new field that paves the way to further developments in graph learning [126, 154]. In this framework, the *smoothness* of observations with respect to the true underlying graph is a common assumption [33, 40, 51, 56, 89], which asks for graphs on which signals have small local variations among adjacent nodes. Another naturally arising property of real-world problems is the sparsity of the observations in the graph spectral basis [139, 160]. In data clustering, for instance, the vector of labels seen as a signal over the vertices of a graph, exhibits a sparse spectral representation. It is smooth within each cluster and varies across different clusters (Figure 3.1). Hence, building such graph is relevant for graph-based clustering approaches, such as spectral clustering. Furthermore, such sparsity assumption is also relevant for the sampling task. Indeed, by making use of this property, it is possible under mild conditions to reconstruct the observations for nodes that have not been sampled [30]. These properties, all borrowed from the GSP field, can be seen as constraints or regularizations for the graph learning task, and offer a new perspective on the topic.

Aim and main contributions. In the present chapter, we introduce an optimization problem to learn a graph from signals that are assumed to be smooth and admitting a sparse representation in the spectral domain of the graph. The main contributions can be summarized as follows:

- The graph learning task problem is cast as the optimization of a smooth nonconvex objective function over a nonconvex set (Section 2). This challenging problem is efficiently solved by introducing a framework that combines barrier methods, alternating minimization, and manifold optimization (Section 3). A relaxed algorithm is also proposed, which allows to scale in time with the graph dimensions (Section 4).

- A factor analysis model for smooth graph signals with sparse spectral representation is introduced (Section 5). This model provides a probabilistic interpretation of our optimization program and links its objective function to a maximum a posteriori estimation.

- The proposed algorithms are tested on several synthetic and real databases, and compared to state-of-the-art approaches (Section 7). Experimental results show that our approach allows to obtain similar or better performance than standard existing methods while significantly lowering the necessary computing resources.

Background and notations. Throughout all the chapter, we consider an undirected and weighted graph G with no self-loops. It is defined as a pair $G = (\mathcal{V}, \mathcal{E})$ with vertices (or

nodes) $\mathcal{V} = \{1, \dots, N\}$, and set edges $\mathcal{E} = \{(i, j, w_{ij}), i, j \in V\}$ with weights $w_{ij} \in \mathbb{R}_+$ arranged in a weight matrix $W \in \mathbb{R}_+^{N \times N}$. More particularly, we focus on its *combinatorial graph* Laplacian matrix which entirely describes it and is given by $L = D - W$, where D is the diagonal degree matrix and W the weight matrix. As G is undirected, L is a symmetric positive semi-definite matrix. Its eigenvalue decomposition can be written as $L = X \Lambda X^\mathsf{T}$, with $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_N)$ a diagonal matrix containing the eigenvalues and $X = (x_1, \dots, x_N)$ a matrix with the eigenvectors as columns. We also consider graph signals (or graph functions) on this graph. A graph signal is defined as a function $y : \mathcal{V} \longrightarrow \mathbb{R}^N$ that assigns a scalar value to each vertex. This function can be represented as a vector $y \in \mathbb{R}^N$, with y_j the function value at the j-th vertex. Also, with $\mathbf{1}_N$ we denote the constant unitary vector of size N, and with $\mathbf{0}_N$ the vector containing only zeros. All remaining notations are given throughout the chapter.

2 Problem Statement

This section describes the graph learning problem for smooth and sparse graph signals.

2.1 Setup and working assumptions

The general task of *graph learning* aims at building a graph G that best explains the structure of n observed graph signals $\{y^{(i)}\}_{i=1}^n$ of size N, composing a matrix $Y = [y^{(1)}, \cdots, y^{(n)}] \in \mathbb{R}^{N \times n}$. The proposed graph learning framework takes as input the matrix Y and outputs the Laplacian matrix L associated to G (note that both notions are equivalent). Our learning process is based on the following assumptions:

Assumption 3.1. (Assumption on the graph G) – *G is undirected, with no self-loops and has a single connected component.*

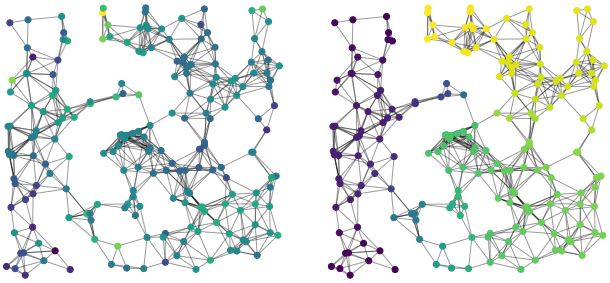
With Assumption 3.1, L is a symmetric positive semi-definite matrix with eigenvalue decomposition $L = X \Lambda X^\mathsf{T}$, where $\lambda_1 = 0$ and $x_1 = \frac{1}{\sqrt{N}} \mathbf{1}_N$ [35].

Assumption 3.2. (Assumption on the signals Y) – *Graph signals Y defined over the true underlying graph G are assumed s-smooth (Def. 1.8) and admit a k-sparse spectral representation (Def. 1.10), with unknown values for s and k.*

On the smoothness assumption. According to the Definition 1.8 of chapter 1, low s values tend to favor smooth signals for which adjacent nodes carry similar signal values. This property has consequently been widely considered for the graph learning task [40, 50].

On the spectral sparsity assumption. This property is known as *bandlimitedness* in the GSP field. In general, it assumes that the null components of h are those associated to the largest eigenvalues (frequencies). Essentially, this additional hypothesis expresses a fundamental principle of signal processing which suggests filtering-out the high-frequency band of a signal, as it carries mainly noise and little or no information.

The bandlimitedness property is very common for graph signals, especially in GSP where it is the main hypothesis of several graph sampling methods [11, 30, 116, 120]. In a word, graph sampling refers to task of recovering a whole graph signal from a subset of



(a) (b)

Figure 3.1: Two graph signals observed on the same graph of 200 nodes. (a) The first signal admits smoothness at the level of adjacent nodes, and a 100-sparse spectral representation. (b) The second signal admits also smoothness, but in this case it extends to larger clusters of connected nodes. As a consequence, this graph signal enjoys a 3-sparse spectral representation.

nodes values. The main property behind the bandlimitedness assumption is that a k-sparse spectral representation implies that only k nodes values are necessary to recover the whole graph signal.

Furthermore, such property is known to carry cluster information on the graph signal: a k-sparse spectral representation implies k clusters. We can visualize this with the following trivial example. Let $y = (1, 1, -1, -1)$ be a graph signal encoding labels of two clusters. If we take the graph with two connected components, the first two nodes being connected together, so as the last two, then the graph exactly matches the cluster structure and the two columns of the spectral basis X associated to the smallest frequencies are $x_1 = 2^{-\frac{1}{2}}(1, 1, 0, 0)$ and $x_2 = 2^{-\frac{1}{2}}(0, 0, 1, 1)$. Thus, y can be written as a linear combination of x_1 and x_2, making it orthogonal with the remaining columns of X and therefore 2-bandlimited..

Figure 3.1 shows visually an example of two graph signals that illustrate the intuition behind our two core assumptions.

2.2 Graph Learning for Smooth and Sparse Spectral Representation

A general graph learning scheme consists in learning the adjacency or the Laplacian matrix. However, since the constraint of Assumption 3.2 (sparsity of the graph signals over the eigen-basis of the Laplacian matrix) is easier to be expressed in the spectral domain, in this chapter we focus on learning the eigendecomposition of the Laplacian matrix $L = X\Lambda X^\mathsf{T}$. The optimization problem incorporates a linear least square regression term depending of Y, X, and H, which controls the distance of the new representation XH to the observations Y. In addition, due to Assumption 3.2, we add two penalization terms: One to control the smoothness of the new representation, depending on Λ and H; the other one to control the sparsity on the spectral domain, which only depends on H. Finally, as we want to learn a Laplacian matrix satisfying Assumption 3.1, equality and inequality constraints relative to

X and Λ are necessary. To that end, we introduce the following optimization problem:

$$\min_{H,X,\Lambda} \|Y - XH\|_F^2 + \alpha\|\Lambda^{1/2}H\|_F^2 + \beta\|H\|_S \;, \tag{3.1}$$

$$\text{s.t.} \quad \begin{cases} X^\mathsf{T}X = I_N,\; x_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N \;, & \text{(a)} \\ (X\Lambda X^\mathsf{T})_{k,\ell} \leq 0 \quad k \neq \ell \;, & \text{(b)} \\ \Lambda = \text{diag}(0, \lambda_2, \ldots, \lambda_N) \succeq 0 \;, & \text{(c)} \\ \text{tr}(\Lambda) = N \in \mathbb{R}_*^+ \;, & \text{(d)} \end{cases}$$

where I_N is the identity matrix of size N, $\text{tr}(\cdot)$ denotes the trace, and $\Lambda \succeq 0$ indicates that the matrix is semi-definite positive.

This problem aims at conjointly learning the Laplacian L (i.e. (X, Λ)) and a smooth bandlimited approximation XH of the observed signals Y. Here, H is the same size as Y and corresponds to the spectral representation of the graph signals through the GFT.

Interpretation of the terms. In the objective function (3.1), the first term corresponds to the quadratic approximation error of Y by XH, where $\|\cdot\|_F$ is the Frobenius norm. The second term is a *smoothness regularization* equally imposed to each column of XH. Indeed, from Equation (1.8), we have $\sum_i y^{(i)\mathsf{T}} L y^{(i)} = \text{tr}(Y^\mathsf{T} L Y) = \|L^{1/2}Y\|_F^2$. Rewriting this for the set of graph signals in XH, we obtain:

$$\|L^{1/2}XH\|_F^2 = \|X\Lambda^{1/2}X^\mathsf{T}XH\|_F^2 = \|\Lambda^{1/2}H\|_F^2 = \sum_{i=1}^{N} \lambda_i \|H_{i,:}\|_2^2 \;,$$

where $H_{i,:}$ is the i-th row of the matrix H. This kind of regularization is very common in graph learning [33, 89]. From its definition, we can see that it tends to be low when high values of $\{\lambda_i\}_{i=1}^N$ are associated to rows of H with low ℓ_2-norm. This corroborates the idea that the $\{\lambda_i\}_{i=1}^N$ can be interpreted as frequencies and the elements of H as Fourier coefficients.

The last term, $\beta\|H\|_S$, is a *sparsity regularization*. In this work, we propose to either use the $\ell_{2,1}$ (sum of the ℓ_2-norm of each row of H) or $\ell_{2,0}$ (number of rows with ℓ_2-norm different than 0) that induces a row-sparse solution $\widehat{H}$.

Remark on the choice of $\|\cdot\|_S$ – In the context of GSP, it is natural to assume that the graph signals are bandlimited at the same dimensions. This property is enforced by $\|\cdot\|_S$ and has two main advantages: it is a key assumption for sampling over a graph and this particular structure is better for inferring graphs with clusters [139]. Therefore, in this chapter, the use of the classical ℓ_0-norm and the ℓ_1-norm have not been investigated since they would impose sparsity at every dimension of the matrix H 'independently', which would consequently break the bandlimitedness assumption.

The hyperparameters, $\alpha, \beta > 0$ are controlling respectively the smoothness of the approximated signals and the sparsity of H. A discussion on the influence of these hyperparameters and an efficient way to fix them is provided in Section 7.3.1. Finally, the first three constraints (3.1a), (3.1b), (3.1c) enforce $X\Lambda X^\mathsf{T}$ to be a Laplacian matrix of a graph with a single connected component (Assumption 3.1). More specifically, by definition, $L = D - W$ with $W \in \mathbb{R}_+^{N\times N}$, thus we necessary have $\forall k \neq \ell, L_{k,\ell} = (X\Lambda X^\mathsf{T})_{k,\ell} \leq 0$ (constraint (3.1b)).

Furthermore, as $X\Lambda X^\mathsf{T}$ is the eigendecomposition of the Laplacian matrix of an undirected graph with a single connected component (Assumption 3.1), $X^\mathsf{T}X = I_N, x_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N$ and $\lambda_1 = 0 < \lambda_2 \leq \ldots \leq \lambda_N$ (constraints (3.1a) and (3.1c)). The last constraint (3.1d) was proposed in Dong et al. [50] to impose structure in the learned graph so that the trivial solution $\widehat{\Lambda} = \mathbf{0}$ is avoided. A discussion about values other than N is made in Kalofolias [89].

The objective function (3.1) is not jointly convex but when $\|\cdot\|_S$ is taken to be the $\ell_{2,1}$ norm, it is convex with respect to each of the block-variables H, X, or Λ, taken independently. A natural approach to solve this problem is therefore to alternate between the three variables, minimizing over one while keeping the others fixed. However, due to the equality constraint $(3.1a)$ and inequalities $(3.1b)$, the feasible set is not convex with respect to X. Hence, this approach raises several difficulties that will be discussed and handled in the following section.

2.3 Reformulation of the problem

As stated in Section 2.2, problem (3.1) is not jointly convex and cannot be solved easily with constraints $(3.1a)$ and $(3.1b)$. In this section, we propose to rewrite constraints $(3.1a)$ and $(3.1b)$, in order to define a new equivalent optimization problem that can be solved with well-known techniques.

2.3.1 Reformulation of the constraint $(3.1a)$

In this section, we show that the constraints (3.1a) can be reformulated as a constraint over the space of orthogonal matrices in $\mathbb{R}^{(N-1)\times(N-1)}$. Although such transformation does not change the convexity of the feasible set, we will see in Section 3.3 that there exist efficient algorithms that perform optimization over such manifold.

Definition 3.1. (Orthogonal group) – *The space of orthogonal matrices in $\mathbb{R}^{N\times N}$, called orthogonal group, is the space:*

$$Orth(N) = \{X \in \mathbb{R}^{N\times N} \mid X^\mathsf{T}X = I_N\} \ .$$

Lemma 3.1. – *Given $X, X_0 \in \mathbb{R}^{N\times N}$ two orthogonal matrices, both having their first column equal to $\frac{1}{\sqrt{N}}\mathbf{1}_N$ (constraint $(3.1a)$), we have the following equality*

$$X = X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & [X_0^\mathsf{T}X]_{2:,2:} \end{bmatrix} \ ,$$

with $[X_0^\mathsf{T}X]_{2:,2:}$ denoting the submatrix of $X_0^\mathsf{T}X$ containing everything but the first row and column of itself. Furthermore, $[X_0^\mathsf{T}X]_{2:,2:}$ is in $Orth(N-1)$.

The above lemma allows us to build an equivalent formulation of Problem (3.1) given by the following proposition.

Proposition 3.1. – *Given $X_0 \in \mathbb{R}^{N\times N}$ an orthogonal matrix with first column being equal to $\frac{1}{\sqrt{N}}\mathbf{1}_N$, an equivalent formulation of optimization problem (3.1) is given by*

$$\min_{H,U,\Lambda} \left\| Y - X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U \end{bmatrix} H \right\|_F^2 + \alpha\|\Lambda^{1/2}H\|_F^2 + \beta\|H\|_S \triangleq f(H,U,\Lambda) \ , \quad (3.2)$$

$$
\text{s.t.} \quad \begin{cases} U^\mathsf{T}U = I_{N-1}\ , & (a') \\ \left(X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U \end{bmatrix} \Lambda \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U^\mathsf{T} \end{bmatrix} X_0^\mathsf{T} \right)_{k,\ell} \leq 0 \quad k \neq \ell\ , & (b') \\ \Lambda = diag(0, \lambda_2, \dots, \lambda_N) \succeq 0\ , & (c) \\ tr(\Lambda) = N \in \mathbb{R}_*^+\ . & (d) \end{cases}
$$

The latter proposition says that since the first column of X is fixed and known, it is sufficient to look for an optimal rotation of a valid matrix X_0 that preserves the first column. Such a rotation matrix is given above and is parametrized by a U in $Orth(N-1)$. Note that in practice, to find a matrix X_0 satisfying (3.1a), we build the Laplacian of any graph with a single connected component and take its eigenvectors.

2.3.2 Log-barrier method for constraint (3.2*b'*)

In order to deal with constraint (3.2b'), we propose to use a log-barrier method. This barrier function allows us to consider an approximation of problem (3.2) where the inequality constraint (3.2b') is made implicit in the objective function. Denoting by $f(\cdot)$ the objective function of (3.2), we want to solve

$$
\min_{H,U,\Lambda} \quad f(H,U,\Lambda) + \frac{1}{t}\phi(U,\Lambda) \qquad \text{s.t.} \quad (3.2\text{a'}), (3.2\text{c}), (3.2\text{d})\ , \tag{3.3}
$$

where t is a fixed positive constant and $\phi(\cdot)$ is the log-barrier function associated to the constraint (3.2b').

Definition 3.2. (Log-barrier function) – *Let the following matrix in* $\mathbb{R}^{N\times N}$*:*

$$
h(U,\Lambda) = X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U \end{bmatrix} \Lambda \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U \end{bmatrix}^\mathsf{T} X_0^\mathsf{T}\ ,
$$

involved in the constraint (3.2b')*. The associated log-barrier function* $\phi : \mathbb{R}^{(N-1)\times(N-1)} \times \mathbb{R}^{N\times N} \longrightarrow \mathbb{R}$ *is defined by:*

$$
\phi(U,\Lambda) = -\sum_{k=1}^{N-1}\sum_{\ell>k}^{N} \log\Big(-h(U,\Lambda)_{k,\ell} \Big)\ , \tag{3.4}
$$

with $\mathbf{dom}(\phi) = \big\{(U,\Lambda) \in \mathbb{R}^{(N-1)\times(N-1)} \times \mathbb{R}^{N\times N} \mid \forall 1 \leq k < \ell \leq N, h(U,\Lambda)_{k,\ell} < 0\big\}$, i.e. its domain is the set of points that strictly satisfy the inequality constraints (3.2b').

This barrier function allows us to perform block-coordinate descent on three easier to solve subproblems, as we discuss in the next section.

3 Resolution of the problem: IGL-3SR

In this section, we describe our method, the *Iterative Graph Learning for Smooth and Sparse Spectral Representation* (IGL-3SR), and its different steps to solve Problem (3.3). Given a fixed $t > 0$, we propose to use a block-coordinate descent on H, U, and Λ, which permits to split the problem in three partial minimizations that we discuss in this section. One of

the main advantages of IGL-3SR is that each subproblem can be solved efficiently and as the objective function is lower-bounded by 0, this procedure ensures convergence. The summary of the method is presented in Algorithm 3.1.

3.1 Optimization with respect to H

For fixed U and Λ, the minimization Problem (3.3) with respect to H is:

$$\min_{H} \|Y - XH\|_F^2 + \alpha\|\Lambda^{1/2}H\|_F^2 + \beta\|H\|_S \ , \quad \text{where } X = X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U \end{bmatrix} . \tag{3.5}$$

When $\|\cdot\|_S$ is set to $\|\cdot\|_{2,0}$ (resp. $\|\cdot\|_{2,1}$), this problem is a particular case of what is known as Sparsify Transform Learning [133] (resp. is a particular case of the Group Lasso [178] known as Multi-Task Feature Learning [12]). Moreover, as X is orthogonal, we are able to find closed-form solutions (Proposition 3.2).

Proposition 3.2. (Closed-form solution for the $\ell_{2,0}$ and $\ell_{2,1}$-norms) – *The solutions of Problem (3.5) when $\|\cdot\|_S$ is set to $\|\cdot\|_{2,0}$ or $\|\cdot\|_{2,1}$, are given in the following.*

- *Using the $\ell_{2,0}$-norm, the optimal solution of (3.5) is given by the matrix $\widehat{H} \in \mathbb{R}^{N\times n}$ where for $1 \leq i \leq N$,*

$$\widehat{H}_{i,:} = \begin{cases} 0 & \text{if } \frac{1}{1+\alpha\lambda_i}\|(X^\mathsf{T}Y)_{i,:}\|_2^2 \leq \beta \ , \\ \frac{1}{(1+\alpha\lambda_i)}(X^\mathsf{T}Y)_{i,:} & \text{else .} \end{cases} \tag{3.6}$$

- *Using the $\ell_{2,1}$-norm, the optimal solution of (3.5) is given by the matrix $\widehat{H} \in \mathbb{R}^{N\times n}$, where for $1 \leq i \leq N$,*

$$\widehat{H}_{i,:} = \frac{1}{1+\alpha\lambda_i}\Big(1 - \frac{\beta}{2}\frac{1}{\|(X^\mathsf{T}Y)_{i,:}\|_2}\Big)_+ (X^\mathsf{T}Y)_{i,:} \ , \tag{3.7}$$

  where $(t)_+ \triangleq \max\{0, t\}$ is the positive part function.

3.2 Optimization with respect to Λ

For fixed H and U, the optimization Problem (3.3) with respect to Λ is:

$$\min_{\Lambda} \ \alpha \underbrace{\operatorname{tr}(HH^\mathsf{T}\Lambda)}_{\|\Lambda^{1/2}H\|_F^2} + \frac{1}{t}\phi(U,\Lambda) \quad \text{s.t.} \quad \begin{cases} \Lambda = \operatorname{diag}(0, \lambda_2, \ldots, \lambda_N) \succeq 0 \ , & \text{(c)} \\ \operatorname{tr}(\Lambda) = N \in \mathbb{R}_*^+ \ . & \text{(d)} \end{cases} \tag{3.8}$$

This objective function is differentiable and convex with respect to Λ, and the constraints define a Simplex. Thus, several convex optimization solvers can be employed, such as those implemented in CVXPY [47]. Popular algorithms are interior-point methods or projected gradient descent methods [114]. Using one algorithm of the latter type, we compute the gradient of 3.8 and project each iteration onto the Simplex [53].

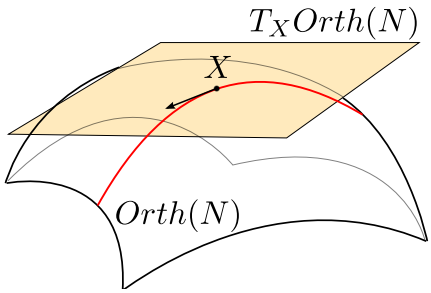


Figure 3.2: The principle of the manifold gradient descend given schematically. $T_X Orth(N)$ is the tangent space of $Orth(N)$ at X. The red line corresponds to a curve in $Orth(N)$ passing through the point X in the direction of the arrow. At each iteration, considering that X is the point of the current solution, a search direction belonging to $T_X Orth(N)$ is first defined, and then a descent along a curve of the manifold is performed (at the direction of the black arrow along the red line).

3.3 Optimization with respect to U

For fixed H and Λ, the optimization Problem (3.3) with respect to U is:

$$\min_{U} \left\| Y - X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^{\mathsf{T}} \\ \mathbf{0}_{N-1} & U \end{bmatrix} H \right\|_F^2 + \frac{1}{t}\phi(U,\Lambda) \qquad \text{s.t.} \qquad U^{\mathsf{T}}U = I_{(N-1)} \ . \quad \text{(a')} \tag{3.9}$$

The objective function is not convex but twice differentiable and the constraint (a') involves the set of orthogonal matrices $Orth(N-1)$ which is not convex. Orthogonality constraint is central to many machine learning optimization problems including Principal Component Analysis (PCA), Sparse PCA, and Independent Component Analysis (ICA) [85, 146, 180]. Unfortunately, optimizing over this constraint is a major challenge since simple updates such as matrix addition usually break orthonormality. One class of algorithms tackles this issue by taking into account that the orthogonal group $Orth(N)$ is a Riemannian submanifold embedded in $\mathbb{R}^{N\times N}$. In this chapter, we focus on manifold adaptation of descent algorithms to solve Problem (3.9).

The generalization of gradient descent methods to a manifold consists in selecting, at each iteration, a search direction belonging to the tangent space of the manifold defined at the current point X, and then performing a descent along a curve of the manifold. Figure 3.2 provides pictures this principle.

Definition 3.3. (Tangent space at a point of $Orth(N)$) – *Let $X \in Orth(N)$. The tangent space of $Orth(N)$ at point X, denoted by $T_X Orth(N)$ is a $\frac{1}{2}N(N-1)$ dimensional vector space defined by:*

$$T_X Orth(N) = \left\{ X\Omega \quad | \quad \Omega \in \mathbb{R}^{N\times N} \textit{ is skew-symmetric} \right\} .$$

When we endow each tangent space with the standard inner product, we are able to define a notion of Riemannian gradient that allows us to find the best direction for the descent. For an objective function $\bar{f} : \mathbb{R}^{N\times N} \to \mathbb{R}$, the Riemannian gradient defined over $Orth(N)$ is given by:

$$\text{grad}\bar{f}(X) = P_X(\nabla_X \bar{f}(X)) \ , \tag{3.10}$$

where P_X is the projection onto the tangent space at X, which is equal to $P_X(\xi) = \frac{1}{2}X(X^\mathsf{T}\xi - \xi^\mathsf{T}X)$, and ∇_X is the standard Euclidean gradient. At each iteration, the manifold gradient descent computes the Riemannian gradient (3.10) that gives a direction in the tangent space. Then the update is given by applying a *retraction* onto this direction, up to a step-size. A retraction consists in an update mapping from the tangent space to the manifold, and there are many possible ways to perform that [3, 14, 54, 118]. From the last equation, we see that in order to solve problem (3.9) with this method, we need the Euclidean gradient of the objective function, namely those of $f(\cdot)$ and $\phi(\cdot)$. These are given in the following proposition.

Proposition 3.3. (Euclidean gradient with respect to U) – *The Euclidean gradient of $f(\cdot)$ and $\phi(\cdot)$ with respect to U are:*

$$\nabla_U f(H, U, \Lambda) = -2\big[(HY^\mathsf{T}X_0)_{2:,2:}\big]^\mathsf{T} + 2U(HH^\mathsf{T})_{2:,2:} \,,$$

$$\nabla_U \phi(U, \Lambda) = -\sum_{k=1}^{N-1}\sum_{\ell>k}^{N} \frac{\Big(B_{k,\ell} + B_{k,\ell}^\mathsf{T}\Big)U\Lambda_{2:,2:}}{h(U,\Lambda)_{k,\ell}} \,,$$

with $\forall 1 \leq k, \ell \leq N, B_{k,\ell} = \big(X_0^\mathsf{T} e_k e_\ell^\mathsf{T} X_0\big)_{2:,2:}$, *and* $h(\cdot)$ *from Definition 3.2.*

3.4 Log-barrier method and initialization

Choice of the t parameter. The quality of the approximation of Problem (3.2) by Problem (3.3) improves as $t > 0$ grows. However, taking a too large t at the beginning may lead to numerical issues. As a solution, we use the path-following method, which computes the solution for a sequence of increasing values of t until the desired accuracy. This method requires an initial value for t, denoted $t^{(0)}$, and a parameter μ such that $t^{(\ell+1)} = \mu t^{(\ell)}$. For an in-depth discussion we refer to Boyd and Vandenberghe [21].

Initialization. At the beginning, our IGL-3SR method requires a feasible solution to initialize the algorithm. One possible choice is to take U as the identity matrix I_{N-1} and to replace (X_0, Λ) by the eigenvalue decomposition of the complete graph with trace equals to N. Indeed, its eigenvalue decomposition will always satisfy the constraints and belong to the domain of the barrier function. The initialization of H is not needed as we start directly with the H-step.

IGL-3SR is summarized in Algorithm 3.1.

3.5 Computational complexity of IGL-3SR

Considering a graph with N nodes and $n > N$ graph signals:

- H-step (non-iterative) – The closed-form solution requires to compute the matrix product $X^\mathsf{T}Y$, which is of complexity $\mathcal{O}(nN^2)$.
- Λ-step (iterative) – When using a projected gradient descent method, the complexity of each iteration is $\mathcal{O}(nN^2)$ to compute the gradient and $\mathcal{O}(N\log(N))$ for the projection [53]. Hence, denoting by τ_Λ the number of iterations in each Λ-step, the complexity is $\mathcal{O}(\tau_\Lambda \cdot nN^2)$

Algorithm 3.1 The IGL-3SR algorithm with $\ell_{2,1}$-norm

Input: $Y \in \mathbb{R}^{N \times n}, \alpha, \beta$
Input of the barrier method: $t^{(0)}, t_{\max}, \mu$ – see Section 3.4
Output: $\widehat{H}, \widehat{X}, \widehat{\Lambda}$
Initialization: L_0 (e.g. with a complete graph) – see Section 3.4

$t \longleftarrow t^{(0)}$
$(X_0, \Lambda) \longleftarrow \text{SVD}(L_0)$
$U \longleftarrow I_{N-1}$
while $t \leq t_{\max}$ **do**
  while not convergence **do**
    ▷ H-`step`: Compute the closed-form solution of Proposition (3.2)
    for $i = 1, \ldots, N$ **do**
      $H_{i,:} \longleftarrow \frac{1}{1+\alpha\lambda_i}\Big(1 - \frac{\beta}{2}\frac{1}{\|(X^\mathsf{T}Y)_{i,:}\|_2}\Big)_+ (X^\mathsf{T}Y)_{i,:}$
    end for
    ▷ Λ-`step`: Solve Problem (3.8)
    $\Lambda \longleftarrow \arg\min_{\Lambda} \alpha \operatorname{tr}(HH^\mathsf{T}\Lambda) + \frac{1}{t}\phi(U,\Lambda) \quad \text{s.t.} \quad \begin{cases} \Lambda = \operatorname{diag}(0, \lambda_2, \ldots, \lambda_N) \succeq 0\,, \\ \operatorname{tr}(\Lambda) = N \in \mathbb{R}_*^+ \end{cases}$
    ▷ U-`step`: Solve Problem (3.9)
    while not convergence **do**
      $U \longleftarrow retraction(U(\big[(HY^\mathsf{T}X_0)_{2:,2:}\big]U - U^\mathsf{T}\big[(HY^\mathsf{T}X_0)_{2:,2:}\big]^\mathsf{T}))$
    end while
  end while
  $t \longleftarrow \mu t$
end while

- X-step (iterative) – The complexity of each iteration is $\mathcal{O}(nN^2)$ to compute the Riemannian gradient and $\mathcal{O}(N^3)$ when we use the QR factorization as retraction [22]. Hence, denoting by τ_X the number of iterations in each X-step, the complexity is $\mathcal{O}(\tau_X \cdot nN^2)$.

Overall – The complexity to go through the big loop of IGL-3SR once (i.e. once through each of the H, Λ, and X steps) is of order $\mathcal{O}(\max(\tau_\Lambda, \tau_X) \cdot nN^2)$. However, recall that τ_Λ and τ_X can be large in practice for reaching a good solution. In the following, we propose a relaxation for a faster resolution that relies on closed-form solutions.

4 A relaxation for a faster resolution: FGL-3SR

In this section, we propose another algorithm called *Fast Graph Learning for Smooth and Sparse Spectral Representation* (FGL-3SR) to approximately solve the initial Problem (3.1). FGL-3SR has a significantly reduced computational complexity due to a well-chosen relaxation. As in the previous section, we use a block-coordinate descent on H, X, and Λ, which permits to decompose the problem in three partial minimizations. FGL-3SR relies on a simplification of the minimization step in X by removing the constraint (3.1b). This simplification allows us to compute a closed-form on this step which greatly accelerates the minimization. However, the constraints (3.1a) and (3.1b) are equally important to obtain

a valid Laplacian matrix at the end, and reducing the problem does not ensure that the constraint (3.1b) will be satisfied. The following proposition explains why we can get rid of constraint (3.1b) at the X-step, while still being able to ensure that the matrix will be a proper Laplacian at the end of the algorithm.

Proposition 3.4. (Feasible eigenvalues) – *Given any $X \in \mathbb{R}^{N\times N}$ being an orthogonal matrix with first column being equal to $\frac{1}{\sqrt{N}}\mathbf{1}_N$ (constraint (3.1a)), there always exists a matrix $\Lambda \in \mathbb{R}^{N\times N}$ such that the following constraints are satisfied:*

$$\begin{cases} (X\Lambda X^\mathsf{T})_{i,j} \leq 0 \quad i \neq j \ , & (3.1b) \\ \Lambda = diag(0, \lambda_2, \dots, \lambda_N) \succeq 0 \ , & (3.1c) \\ tr(\Lambda) = c \in \mathbb{R}_*^+ \ . & (3.1d) \end{cases}$$

In Proposition 3.5 of the next section, we will see that, by ignoring constraint (3.1b) at the X-step, we can compute a closed-form solution to the optimization problem. For this reason, we propose to use the closed-form solution that we derive to learn X, and right after always optimize with respect to Λ. Hence, we are sure that we will obtain a proper Laplacian at the end of the process (Proposition 3.4). The initialization and the optimization with respect to H are not concerned by this relaxation and can therefore be performed as in IGL-3SR (see Sections 3.1 and 3.4).

4.1 Optimization with respect to X

As already explained, during the X-step, we solve the program

$$\min_X \|Y - XH\|_F^2 \qquad \text{s.t.} \quad X^\mathsf{T}X = I_N, \ x_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N \ , \quad (3.1a) \tag{3.11}$$

where the constraint $(3.1b)$ is missing. The closed-form solution is given next.

Proposition 3.5. (Closed-form solution of Problem (3.11)) – *Let X_0 be any matrix that belongs to the constraints set (3.1a), and $M = (X_0^\mathsf{T} Y H^\mathsf{T})_{2:,2:}$ the submatrix containing everything but the input's first row and first column. Finally, let PDQ^T be the SVD of M. Then, the problem admits the following closed form solution:*

$$\widehat{X} = X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & PQ^\mathsf{T} \end{bmatrix} \ . \tag{3.12}$$

In practice, X_0 can be fixed to the current value of X.

4.2 Optimization with respect to Λ

With respect to Λ, the optimization Problem (3.1) becomes:

$$\min_\Lambda \ \alpha \ \underbrace{\operatorname{tr}(HH^\mathsf{T}\Lambda)}_{\|\Lambda^{1/2}H\|_F^2} \qquad \text{s.t.} \quad \begin{cases} (X\Lambda X^\mathsf{T})_{i,j} \leq 0 \quad i \neq j \ , & \text{(b)} \\ \Lambda = \operatorname{diag}(0, \lambda_2, \dots, \lambda_N) \succeq 0 \ , & \text{(c)} \\ \operatorname{tr}(\Lambda) = N \in \mathbb{R}_*^+ \ , & \text{(d)} \end{cases} \tag{3.13}$$

which is a linear program that can be solved efficiently using linear cone programs. Note that this will involve an optimization over N parameters with $\frac{1}{2}N(N-1) + N + 1$ constraints.

FGL-3SR is summarized in Algorithm 3.2.

Algorithm 3.2 The FGL-3SR algorithm with $\ell_{2,1}$-norm

Input : $Y \in \mathbb{R}^{N \times n}$, α, β
Output : $\widehat{H}$, $\widehat{X}$, $\widehat{\Lambda}$
Initialization: L_0 (e.g. with a complete graph) – see Section 3.4

$(X, \Lambda) \longleftarrow \text{SVD}(L_0)$
for $t = 1, 2, \ldots$ **do**

▷ H`-step:` Compute the closed-form solution of Proposition (3.2)
for $i = 1, \ldots, N$ **do**
$$H_{i,:} \longleftarrow \frac{1}{1+\alpha\lambda_i}\Big(1 - \frac{\beta}{2}\frac{1}{\|(X^\mathsf{T}Y)_{i,:}\|_2}\Big)_+ (X^\mathsf{T}Y)_{i,:}$$
end for

▷ X`-step:` Compute the closed-form solution of Proposition (3.5)
$M \longleftarrow (X^\mathsf{T}YH^\mathsf{T})_{2:,2:}$
$(P, D, Q^\mathsf{T}) \longleftarrow \text{SVD}(M)$
$$X \longleftarrow X \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & PQ^\mathsf{T} \end{bmatrix}$$

▷ Λ`-step:` Solve the linear Program (3.13)
$$\Lambda \longleftarrow \arg\min_{\Lambda} \alpha\,\text{tr}(HH^\mathsf{T}\Lambda) \qquad \text{s.t.} \quad \begin{cases} (X\Lambda X^\mathsf{T})_{i,j} \leq 0 \quad i \neq j \\ \Lambda = \text{diag}(0, \lambda_2, \ldots, \lambda_N) \succeq 0 \\ \text{tr}(\Lambda) = N \in \mathbb{R}_*^+ \end{cases}$$

end for

4.3 Computational complexity of FGL-3SR

Considering a graph with N nodes and n graph signals:

- H-step – The closed-form solution requires to compute the matrix product $X^\mathsf{T}Y$, which is of complexity $\mathcal{O}(nN^2)$.
- X-step – The closed-form solution requires to compute the SVD of $(X_0^\mathsf{T}YH^\mathsf{T})_{2:,2:} \in \mathbb{R}^{(N-1)\times(N-1)}$, which is of complexity $\mathcal{O}(N^3)$ [37].
- Λ-step – Solving the LP can be done with interior-point methods or with the ellipsoid method [161]. For accuracy ε, the ellipsoid method yields a complexity of $\mathcal{O}(\max(m, N) \cdot N^3 \log(1/\varepsilon))$, where $m = \frac{1}{2}N(N-1) + N + 1$ is the number of constraints [26].

Overall – As $m > N$, the complexity for FGL-3SR is of order $\mathcal{O}(N^5)$ when using the ellipsoid method. In contrast, the most competitive related algorithm of the literature (ESA-GL [139]) relies on a semi-definite program and is of order at least $\mathcal{O}(N^8)$ (see Section 6). As will be clearly demonstrated in Section 7, in practice the empirical execution time of FGL-3SR is lower than IGL-3SR and ESA-GL.

4.4 Differences between IGL-3SR and FGL-3SR

The two proposed algorithms are based on a modification of the initial optimization Problem (3.1). Indeed, both of them relax the constraint (3.1*b*), $\forall k \neq \ell,\ (X\Lambda X^\mathsf{T})_{k,\ell} \leq 0$, but using two different approaches. IGL-3SR approximates the initial optimization problem through

the use of a log-barrier function. The advantage of the barrier is twofold: first, it allows to overcome the technical constraint (3.1*b*) and solve the program using a block-coordinate descent algorithm; second, the use of the barrier makes the block-variables separable over the constraint set, allowing the convergence of the objective function of IGL-3SR. In addition, IGL-3SR always keeps the set of variables in the initial set of constraints, essential for the matrix $X\Lambda X^\mathsf{T}$ to be a proper Laplacian.

On the other hand, FGL-3SR, instead of using a log-barrier function to relax the constraint (3.1*b*), it removes it from the X-step. Recall that we are perfectly able to do that as we know from Proposition 3.4 that for any X returned by the X-step, there exist a Λ making $X\Lambda X^\mathsf{T}$ a Laplacian. This relaxation speeds-up drastically the X-step while loosing the convergence property and the decreasing over the initial constraints set.

5 A probabilistic interpretation

In this section, we introduce a representation model adapted to smooth graph signals with sparse spectral representation. The goal of this model is to provide a probabilistic interpretation of Problem (3.1) and link its objective function to a maximum a posteriori estimation (Proposition 3.6).

Given a Laplacian matrix $L = X\Lambda X^\mathsf{T}$, we propose the following *Factor Analysis Framework* to model a graph signal y:

$$y = Xh + m_y + \varepsilon \;, \tag{3.14}$$

where $m_y \in \mathbb{R}^N$ is the mean of the graph signal y and ε is a Gaussian noise with zero mean and covariance $\sigma^2 I_N$. Here, the latent variable $h = (h_1, \dots, h_N)$ controls y through the eigenvector matrix X of L. The choice of the representation matrix X is particularly adapted since it reflects the topology of the graph and provides a spectral embedding of its vertices. Moreover, as seen in Section 2, X can be interpreted as a graph Fourier basis, which makes it an intuitive choice for the representation matrix. In a noiseless scenario with $m_y = 0$, h actually corresponds to the GFT of y.

To comply with the spectral sparsity assumption (Assumption 3.2), we now propose a distribution that allows h to admit zero-valued components. To this end, we introduce independent latent Bernoulli variables γ_i with success probability $p_i \in [0, 1]$. Knowing $\gamma_1, \dots, \gamma_N$, the conditional distribution for h is:

$$h|\gamma \sim \mathcal{N}(0, \widetilde{\Lambda}^\dagger) \;, \tag{3.15}$$

where $\widetilde{\Lambda}^\dagger$ is the Moore-Penrose pseudo-inverse of the diagonal matrix containing the values $\{\lambda_i \mathbb{1}\{\gamma_i = 1\}\}_{i=1}^N$. In this model, γ_i controls the sparsity of the i-th element of h. Indeed, if $\gamma_i = 0$, then $h_i = 0$ almost surely. In the other hand, if $\gamma_i = 1$ then h_i follows a Gaussian distribution with zero-mean and variance equal to $1/\lambda_i$. This is adapted to the smoothness hypothesis as for high value of λ_i (high frequency), the distribution of h_i concentrates more around 0, leading to small value of $\lambda_i h_i^2$. The associated probability of success p_i can be chosen *a priori*. One way to chose it is to take p_i inversely proportional to λ_i. Indeed, this would increase the probability to be sparse at dimensions where the associated eigenvalue is high. Note that, since $\lambda_1 = 0$, h_1 follows a centered degenerate Gaussian, i.e h_1 is equal to 0 almost surely. Furthermore, if $p_i = 1$ for all i, our model reduces to the one proposed by Dong et al. [50], which was only focused on the smoothness assumption.

Definition 3.4. (Prior and conditional distributions) – *The following equations summarize the prior and important conditional distributions of our model:*

$$p(h_i|\gamma_i,\lambda_i) \propto \exp(-\lambda_i h_i^2)\mathbb{1}\{\gamma_i = 1\} + \mathbb{1}\{h_i = 0, \gamma_i = 0\} \;, \tag{3.16}$$

$$p(y|h,X) \propto \exp(-\frac{1}{\sigma^2}\|y - Xh - m_y\|_2^2) \;, \tag{3.17}$$

$$p(\gamma_i) \propto p_i^{\gamma_i}(1-p_i)^{1-\gamma_i} \;. \tag{3.18}$$

For simplicity, in the following we consider that $m_y = 0$ and $p_1 = 0$.

Lemma 3.2. – *Assume the proposed Model (3.14). If $p_1 = 0$ and $p_i \in (0,1)$, $\forall i \geq 2$, then:*

$$\begin{aligned} -\log(p(h|y,X,\Lambda)) \;\propto\; & \frac{1}{\sigma^2}\|y - Xh\|_2^2 + \frac{1}{2}h^\mathsf{T}\Lambda h \\ & + \sum_{i=1}^{N} \mathbb{1}\{h_i \neq 0\}\left(p_i \log(\frac{\lambda_i}{\sqrt{2\pi}}) - \log(p_i) - \log(\frac{\lambda_i}{\sqrt{2\pi}})\right) . \end{aligned}$$

Definition 3.5. (Lambert W-Function) – *The Lambert W-Function, denoted by $W(\cdot)$, is the inverse function of $f : W \longmapsto We^W$. In particular, we consider W to be the principal branch of the Lambert function, defined over $[-1/e, \infty)$.*

Proposition 3.6. (A posteriori distribution of h) – *Let $C > 0$, and assume for all $i \geq 2$ that $p_i = e^{-C}$ if $\lambda_i = \sqrt{2\pi}$, whereas $p_i = -W\left(-\frac{e^{-C}\log(\lambda_i/\sqrt{2\pi})}{\lambda_i/\sqrt{2\pi}}\right)\frac{1}{\log(\lambda_i/\sqrt{2\pi})}$ otherwise. Then, $p_i \in (0,1)$ and there exist constants $\alpha, \beta > 0$ such that:*

$$-\log(p(h|y,X,\Lambda)) \;\propto\; \|y - Xh\|_2^2 + \alpha h^\mathsf{T}\Lambda h + \beta\|h\|_0 \;.$$

This proposition tells us that: *for a given Laplacian matrix, the maximum a posteriori estimate of h would corresponds to the minimum of Problem* (3.1).

6 Related work on GSP-based graph learning methods

Here we detail the two state-of-the-art methods for graph learning in the GSP context that are closer to our work and that will be used for our experimental comparison in Section 7.

GL-SigRep [50]. This method supposes that the observed graph signals are smooth with respect to the underlying graph, but do not consider the spectral sparsity assumption. To learn the graph, they propose to solve the optimization problem:

$$\min_{L,\tilde{Y}} \|Y - \tilde{Y}\|_F^2 + \alpha\|L^{1/2}\tilde{Y}\|_F^2 + \beta\|L\|_F^2 \quad \text{s.t.} \quad \begin{cases} L_{k,\ell} = L_{\ell,k} \leq 0 \quad k \neq \ell \;, \\ L\mathbf{1} = \mathbf{0} \;, \\ \mathrm{tr}(L) = N \in \mathbb{R}_*^+ \;. \end{cases} \tag{3.19}$$

Remark that since no constraints are imposed on the spectral representation of the signals, the Laplacian matrix is directly learned. The optimization procedure to solve (3.19) consists in an alternating minimization over L and $\tilde{Y}$. With respect to $\tilde{Y}$ the problem has a closed-form solution whereas for L, the authors propose to use a Quadratic Program solver involving $\frac{1}{2}N(N-1)$ parameters and $\frac{1}{2}N(N-1) + N + 1$ constraints.

ESA-GL [139]. This is a two-step algorithm where the signals are supposed to admit a sparse representation with respect to the learned graph. The difference to our method is two-fold. First, ESA-GL does not include the smoothness assumption while learning the Fourier basis X. This brings a different two-step optimization program. Second, the complexity of the ESA-GL algorithm (at least $\mathcal{O}(N^8)$) is much higher than ours ($\mathcal{O}(N^5)$ for FGL-3SR - see Section 4.3), and hence is prohibitive for large graphs. The first step consists in fitting an orthonormal basis such that the observed graph signals Y admit a sparse representation with respect to this basis. They consider the problem:

$$\min_{H,X} \|Y - XH\|_F^2 \qquad \text{s.t.} \quad \begin{cases} X^\top X = I_N,\ x_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N\ , \\ \|H\|_{2,0} \leq K \in \mathbb{N}\ , \end{cases} \tag{3.20}$$

which is solved using an alternating minimization. Once estimates for H and X have been computed, they solve a second optimization problem in order to learn the Laplacian L associated to the learned basis $\widehat{X}$. This is done by minimizing:

$$\min_{L\in\mathbb{R}^{N\times N},\, C_K\in\mathbb{R}^{K\times K}} \operatorname{tr}(\widehat{H}_K^T C_K \widehat{H}_K) + \mu\|L\|_F^2 \quad \text{s.t.} \quad \begin{cases} L_{k,\ell} = L_{\ell,k} \leq 0 \quad k \neq \ell\ , \\ L\mathbf{1}_N = \mathbf{0}_N\ , \\ L\widehat{X}_K = \widehat{X}_K C_K, \quad C_K \succeq 0\ , \\ \operatorname{tr}(L) = N \in \mathbb{R}_*^+\ , \end{cases} \tag{3.21}$$

where $C_K \in \mathbb{R}^{K\times K}$ and $\widehat{X}_K$ corresponds to the columns of $\widehat{X}$ associated to the non-zero rows of $\widehat{H}$ denoted $\widehat{H}_K$. Thus, the second step aims at estimating a Laplacian that enforces the smoothness of the learned signal representation $\widehat{X}\widehat{H}$. This semi-definite program requires the computation of over $\frac{1}{2}N(N-1) + \frac{1}{2}K(K-1)$ parameters that, as we show empirically in the next section, can be difficult to compute for graphs with large number of nodes. For more details on the optimization program and the additional matrix C_K, the readers shall refer to the aforementioned paper.

7 Experimental evaluation

The two proposed algorithms, IGL-3SR and FGL-3SR, are now evaluated and compared with the two state-of-the-art methods presented earlier, GL-SigRep and ESA-GL. The results of our empirical evaluation are organized in three subsections: Section 7.2 and 7.3 use synthetic data for first comparing the different methods and then study the influence of the hyperparameters; Section 7.4 displays several examples on real-world data.

All experiments were conducted on a single personal computer: a personal laptop with with 4-core 2.5GHz Intel CPUs and `Linux/Ubuntu` OS. For the Λ-step of both algorithms, we use the Python's CVXPY package [47]. For the X-step of IGL-3SR, we use the conjugate gradient descent solver combined with an adaptive line search, both provided by `Pymanopt` [156], a Python toolbox for optimization on manifolds. Note that this package only requires the gradients given in Proposition 3.3. The source code of our implementations is available online[1].

[1] https://github.com/pierreHmbt/GL-3SR

7.1 Evaluation metrics

We provide visual and quantitative comparisons of the learned Laplacian $\widehat{L}$ and its weight matrix $\widehat{W}$ using the performance measures: *Recall*, *Precision*, and F_1-*measure*, which are standard for this type of evaluation [126]. The F_1-measure evaluates the quality of the estimated support – the non-zero entries – of the graph and is given by:

$$F_1 = \frac{2 \times precision \times recall}{precision + recall} \ .$$

As in [126], the F_1-measure is computed on a thresholded version of the estimated weight matrix $\widehat{W}$. This threshold is equal to the average value of the off-diagonal entries of $\widehat{W}$ (same process as in [139]).

In addition, we compute the correlation coefficient $\rho(L, \widehat{L})$ between the true Laplacian entries $L_{i,j}$ and their estimates $\widehat{L}_{i,j}$

$$\rho(L, \widehat{L}) = \frac{\sum_{ij}(L_{ij} - L_m)(\widehat{L}_{ij} - \widehat{L}_m)}{\sqrt{\sum_{ij}(L_{ij} - L_m)^2}\sqrt{\sum_{ij}(\widehat{L}_{ij} - \widehat{L}_m)^2}} \ , \tag{3.22}$$

where L_m and $\widehat{L}_m$ are the average values of the entries of the true and estimated Laplacian matrices, respectively. This ρ coefficient evaluates the quality of the weights distribution over the edges.

7.2 Experiments on synthetic data

We now evaluate and compare all algorithms on several types of synthetic data. Details about graphs, associated graph signals, and evaluation protocol used for the experiments, are detailed in the sequel.

Graphs and signals. We carried out experiments on graphs with $20, 50$, and 100 vertices, following: i) a Random Geometric (RG) graph model with a 2-D uniform distribution for the coordinates of the nodes and a truncated Gaussian kernel of width size 0.5 for the edges, where weights smaller than 0.75 were set to 0; ii) an Erdős-Rényi (ER) model with edge probability 0.2.

Given a graph, the sampling process was made according to Model (3.16) that we presented in Section 5. The mean value of each signal was set to 0, the variance of the noise was set to 0.5, and the sparsity was chosen to obtain observations with k-sparse spectral representation, where k is equal to half the number of nodes (i.e $10, 20, 50$).

For each type of graph, we ran 10 experiments with 1000 graph signals generated as explained above. For all the methods, the hyperparameters α and β are set by maximizing the F_1-measure on the thresholded $\widehat{W}$, as explained in Section 7.1.

Choice of $\|\cdot\|_S$. In the following we make all experiments for IGL-3SR and FGL-3SR with the $\ell_{2,1}$-norm. This is motivated by an important fact brought by the closed-form solutions given in Proposition 3.2. Indeed, for $\ell_{2,1}$-norm, the sparsity of $\widehat{H}$ is only controlled by β (Equation (3.7)). On the contrary, when using the $\ell_{2,0}$-norm, the value of α also influences the sparsity (Equation (3.6)). This is an important behavior, as the tuning of β and α becomes *independent* – at least with respect to the H-step – and therefore, as we will see in Section 7.3.1, easier to tune.

N	Metrics	*RG graph model* IGL-3SR	FGL-3SR	ESA-GL	GL-SigRep	*ER graph model* IGL-3SR	FGL-3SR	ESA-GL	GL-SigRep
20	Precision	**0.973** (±**0.042**)	0.952 (±0.042)	0.899 (±0.054)	0.929 (±0.068)	**0.952** (±**0.045**)	0.819 (±0.080)	0.931 (±0.045)	0.704 (±0.125)
	Recall	0.974 (±0.018)	**0.985** (±**0.023**)	0.968 (±0.052)	0.967 (±0.028)	0.927 (±0.046)	0.824 (±0.105)	**0.951** (±**0.041**)	0.899 (±0.075)
	F_1-measure	**0.974** (±**0.028**)	0.968 (±0.027)	0.929 (±0.032)	0.947 (±0.040)	0.938 (±0.028)	0.816 (±0.068)	**0.941** (±**0.038**)	0.779 (±0.071)
	$\rho(L, \hat{L})$	**0.938** (±**0.052**)	0.903 (±0.029)	0.925 (±0.050)	0.786 (±0.037)	**0.917** (±**0.035**)	0.730 (±0.063)	0.897 (±0.045)	0.199 (±0.074)
	Time	< 1min	< 10s	< **5s**	< **5s**	< 1min	< 10s	< **5s**	< **5s**
50	Precision	**0.901** (±**0.022**)	0.817 (±0.041)	0.845 (±0.088)	0.791 (±0.055)	0.820 (±0.027)	0.791 (±0.047)	**0.854** (±**0.038**)	0.476 (±0.037)
	Recall	0.902 (±0.018)	0.807 (±0.036)	**0.910** (±**0.040**)	0.720 (±0.059)	0.812 (±0.042)	0.740 (±0.049)	0.830 (±0.051)	**0.856** (±**0.023**)
	F_1-measure	**0.901** (±**0.014**)	0.812 (±0.017)	0.868 (±0.036)	0.750 (±0.001)	0.815 (±0.021)	0.761 (±0.031)	**0.841** (±**0.021**)	0.610 (±0.026)
	$\rho(L, \hat{L})$	**0.863** (±**0.020**)	0.743 (±0.031)	0.832 (±0.033)	0.549 (±0.022)	0.783 (±0.026)	0.728 (±0.020)	**0.816** (±**0.058**)	0.058 (±0.002)
	Time	< 17mins	< **40s**	< 60s	< **40s**	< 17mins	< **40s**	< 60s	< **40s**
100	Precision	**0.713** (±**0.012**)	0.711 (±0.029)	0.667 (±0.022)	–	**0.677** (±**0.044**)	0.640 (±0.033)	0.654 (±0.038)	–
	Recall	**0.751** (±**0.067**)	0.584 (±0.011)	0.743 (±0.017)	–	0.580 (±0.021)	0.543 (±0.027)	**0.637** (±**0.023**)	–
	F_1-measure	**0.732** (±**0.034**)	0.641 (±0.010)	0.703 (±0.012)	–	**0.623** (±**0.009**)	0.586 (±0.016)	0.589 (±0.019)	–
	$\rho(L, \hat{L})$	**0.612** (±**0.045**)	0.483 (±0.015)	0.596 (±0.033)	–	0.551 (±0.016)	0.512 (±0.0223)	**0.644** (±**0.023**)	–
	Time	< 50mins	< **2mins**	< 4mins	–	< 50mins	< **2mins**	< 4mins	–

Table 3.1: Comparison of the four methods on five quality metrics (avg ± std) for graphs of $N = \{20, 50, 100\}$ nodes, and for fixed number of $n = 1000$ graph signals.

Quantitative results. Average evaluation metrics and their standard deviation are collected in Table 3.1. The results show that the use of the sparsity constraint improves the quality of the learned graphs. Indeed, the two proposed methods IGL-3SR and FGL-3SR, as well as ESA-GL, have better overall performance in all the metrics than GL-SigRep that only considers the smoothness aspect. This had to be expected as our methods match perfectly to the sparse (bandlimited) condition.

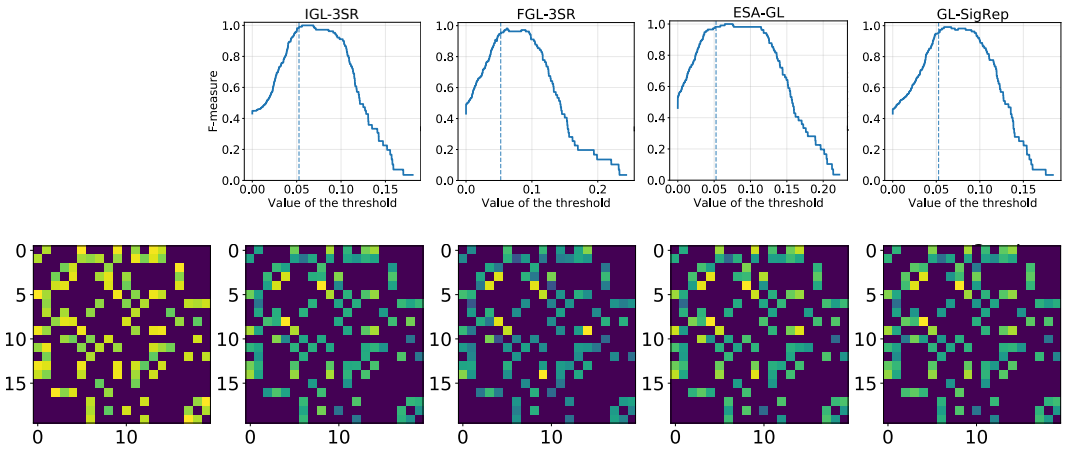
Comparing the results across the different types of synthetic graphs, our methods are robust while being more efficient on RG graphs.

In general, IGL-3SR, and FGL-3SR present similar performance to ESA-GL. However IGL-3SR seems preferable in the case of RG graphs. For 100 nodes, the computational resources necessary for GL-SigRep was already too demanding, therefore only the results for the rest three methods are reported. We can see that, while IGL-3SR has better results than FGL-3SR, the time necessary to estimate the graph is much longer. In addition, examples of learned graphs are displayed in Figure 3.3 with the ground-truth on the left and the learned weighted adjacency matrices (after thresholding). The evolution of the F_1-measure regarding the value of the threshold is also displayed and shows that a large range of threshold could have been used to obtain similar performance. All these results, combined with those of Table 3.1, indicate that in this sampling process the proposed FGL-3SR method managed to infer accurate graphs despite the relaxation.

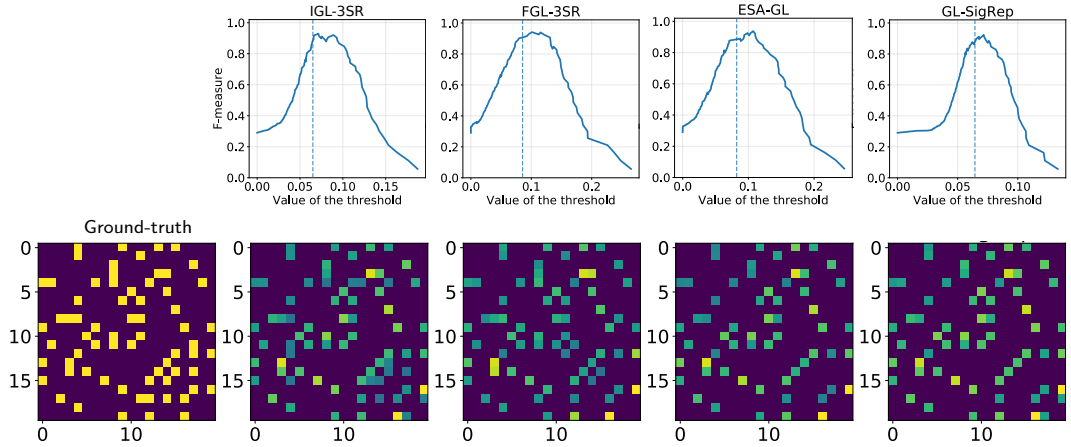
Speed performance. Figure 3.4 displays the evolution of the empirical computation time as the number of nodes increases. FGL-3SR appears to be much faster than the other methods. Furthermore, we observe that our methods are scalable over a wider range of graph sizes than the competitors. Indeed, even quite small graphs of 100 and 150 nodes, respectively, were already too 'large' for the two competitors to be able to produce results, and they even led to memory allocation errors.

7.3 Influence of the hyperparameters

We now study how hyperparameters of IGL-3SR and FGL-3SR influence their overall performance, with respect to the F_1-measure. This study is made on a RG graph with

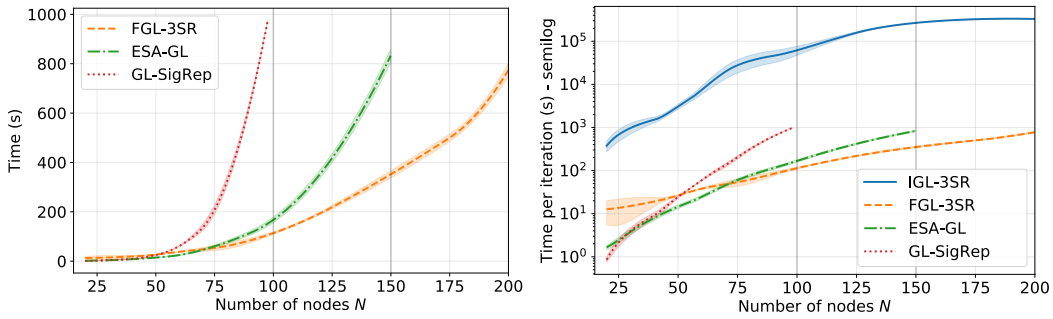


(a) Graph learning on RG synthetic graphs.



(b) Graph learning on ER synthetic graphs.

Figure 3.3: Graph learning results on random synthetic graphs of 20 nodes: (a) for a RG graph, and (b) for an ER graph. Each of the two subfigures presents: (top row) the evolution of the F_1-measure with respect to different threshold values and the dashed line indicates the chosen threshold value; (bottom row) shows as leftmost the ground truth adjacency matrix, followed by the respective learned adjacency matrices (thresholded) by the compared methods.



(a) Standard scale.

(b) Semi-log scale.

Figure 3.4: Average and standard deviation of the computation time over 10 trials for IGL-3SR, FGL-3SR, ESA-GL, and GL-SigRep, as the number of nodes increases. GL-SigRep and ESA-GL failed to produce a result for graphs with more than 100 and 150 nodes, respectively. (a) The total computation times, and (b) the time needed for a single iteration of each algorithm. For IGL-3SR and FLG-3SR, a single iteration means the computation of the 3 steps one time.

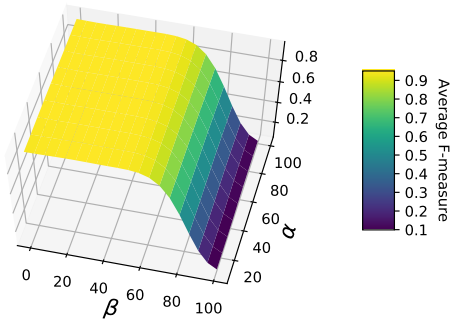
$N = 20$ nodes and 10-bandlimited signals Y in $\mathbb{R}^{20\times 1000}$.

7.3.1 Influence of α and β

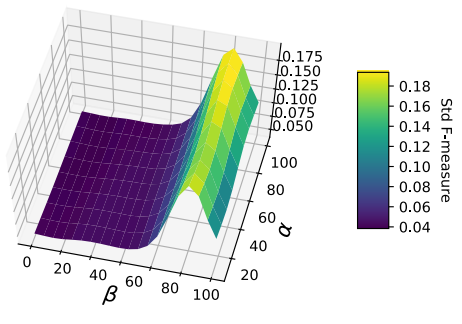
We first highlight the influence of α and β on FGL-3SR. We run and collect the F_1-measure for 20 values of α (resp. β) in $[10^{-5}, 100]$ (resp. $[10^{-5}, 60]$). The resulting heatmaps are displayed in Figure 3.5. The most important observation is that the value of α does not seem to impact the quality of the resulted graphs. Indeed, for a fixed value of β, the F_1-measure is stable when α varies. However, it is interesting that the convergence curve of FGL-3SR (Figure 3.6) is directly impacted by α: large values for α tend to produce oscillations on the convergence curves. Thus, setting to a small value $\alpha > 0$ is suggested. Contrary to α, tuning the parameter β is critical since high β values cause a drastic decrease in F_1-measure. This sharp decrease appears when the chosen β imposes too much sparsity for the learned $\widehat{H}$. One may note that the best β corresponds to the value just before the sharp decrease, and this is the value that should be chosen. Although the previous analysis has been done on FGL-3SR, during our experimental studies, α and β influenced the F_1-measure similarly when using IGL-3SR.

7.3.2 Influence of t

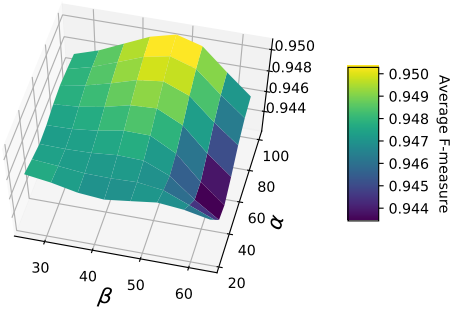
We now highlight the influence of t on IGL-3SR. Figure 3.7 shows the learned graphs for several values of $t \in [10,\ 10^4]$. This experiment brings two main messages: first, when t is too low, the learned graph is very close to the complete graph, whereas when t increases the learned graph becomes more structured and tends to be sparse. This result was expected since a larger t brings the barrier closer to the true constraint, i.e. we allow elements of the resulting Laplacian matrix to be closer to 0. Second, it appears that α also influences the final results in a similar way to t. Again, this was expected as the minimization of the



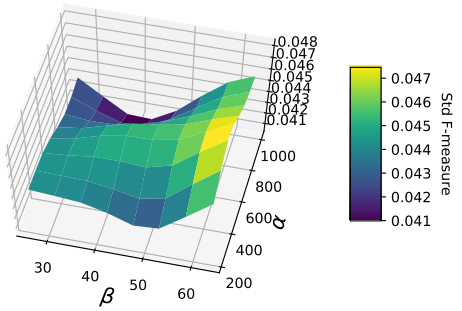
(a) Average of the F_1-measure.



(b) Standard deviation of the F_1-measure.

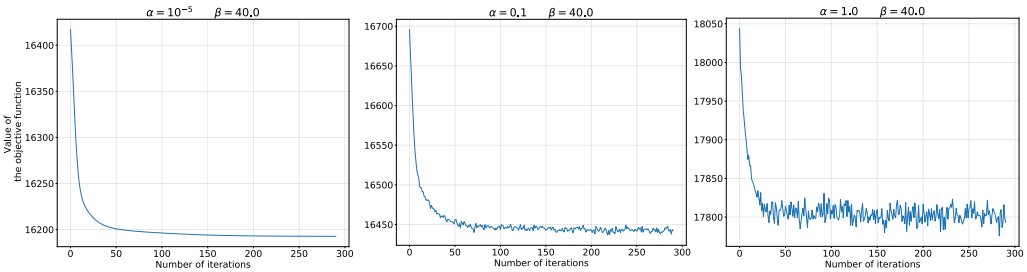


(c) Average of the F_1-measure (focus).



(d) Standard deviation of the F_1-measure (focus).

Figure 3.5: Evolution of the average (a)(c) and standard deviation (b)(d) of the F_1-measure over 10 runs of FGL-3SR on RG graphs with 20 nodes. At the top figure row $\beta \in [0,\ 100]$, and at the bottom row $\beta \in [20,\ 70]$.



(a) Low α value.

(b) Medium α value.

(c) High α value.

Figure 3.6: Convergence curves of the objective function as the number of iterations increases, using FGL-3SR with (a) $\alpha = 10^{-5}$, (b) $\alpha = 10^{-1}$, (c) $\alpha = 1$.

objective function during the Λ-step of Problem (3.3) is equivalent to the minimization of $\mathrm{tr}(HH^\mathsf{T}\Lambda) + \frac{1}{\alpha t}\phi(U, \Lambda)$.

For a discussion on the initial value of t, $t^{(0)}$, and the step size μ such that $t^{(\ell+1)} = \mu t^{(\ell)}$, both relative to the barrier method, we refer the reader to [21]. However, recall that t is not a hyperparameter to tune in practice, and should be taken as large as possible. The mere goal is to prevent numerical issues. Fortunately, a wide range of values for $t^{(0)}$ and μ achieves that goal [21].

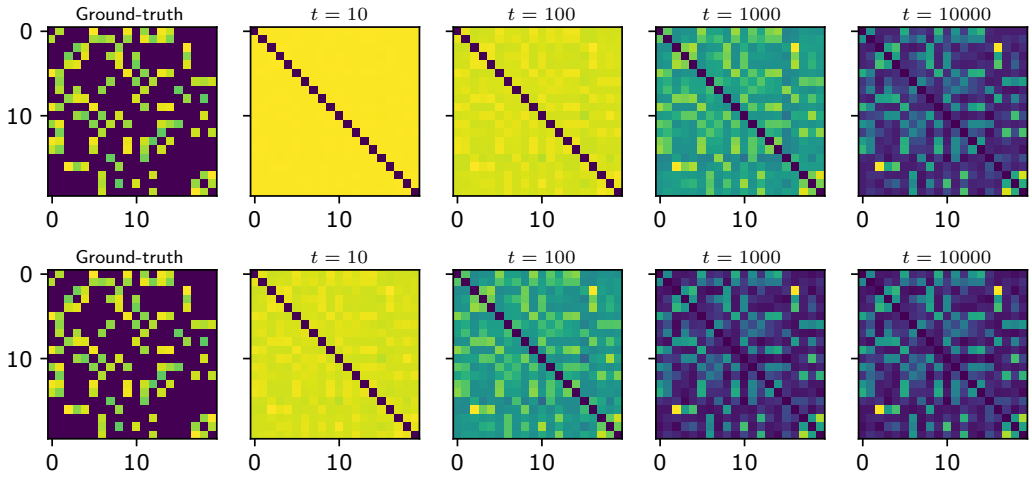


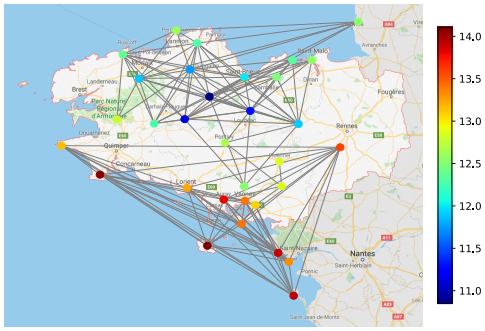
Figure 3.7: Learned graphs with increasing t values: (top row) $\alpha = 10^{-4}$, (bottom row) $\alpha = 10^{-3}$.

Tuning the hyperparameters. The hyperparameter α does not seems to have a substantial impact on the F_1-measure. However, a low value of it may be preferred in FGL-3SR for convergence purpose (Figure 3.6). The parameter t always needs to be maximal provided that it does not cause numerical issues. Classical heuristics and methods, like the one presented in Section 3.4, can be used to tune t [21]. Hence, according to our experiments, it remains only β as a critical hyperparameter to tune for both these methods. Based on Figure 3.5, one way to fix it is to find the largest β value that leads to satisfying results in terms of signal reconstruction. Alternatively, if we have an idea about the number of clusters k that resides on the graph, we could select a β value that produces a k-sparse spectral representation. Bearing in mind that other related works require the tuning of two hyperparameters, our approach turns out to be of higher value for practical application on real data where these parameters are unknown and must be tuned.

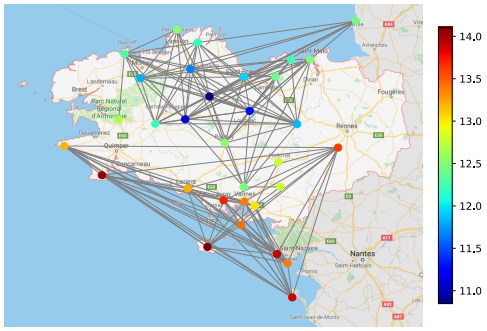
7.4 Temperature data

We used hourly temperature (C°) measurements on 32 weather stations in Brittany, France, during a period of 31 days [33]. The dataset contains $24 \times 31 = 744$ multivariate observations, i.e. $Y \in \mathbb{R}^{32\times744}$, that are assumed to correspond to an unknown graph, which is our objective to infer. For our two algorithms, we set $\alpha = 10^{-4}$, and β is chosen so that we obtain a 2-sparse spectral representation, which this last assumes that there are two clusters of weather stations.

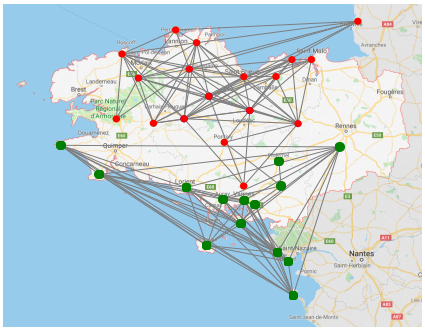
The graphs obtained with each of the method are displayed in Figure 3.8 (a-b). They are in accordance with the one found in [33] on the same dataset. Both the proposed methods provide similar results, which shows that the relaxation used in FGL-3SR has a moderate influence in practice in this real-world problem. Although ground-truth is not available for this use-case, the quality of the learned graph can be assessed when using it as input in standard tasks such as graph clustering or sampling. For instance, when applying spectral clustering [123] with two clusters on the resulting Laplacian matrices, it can be seen that both methods split the learned graph in two parts corresponding to the north and the south



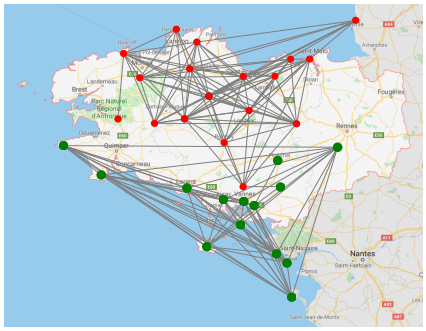
(a) Learned graph by IGL-3SR.



(b) Learned graph by FGL-3SR.



(c) Graph clustering on IGL-3SR's result.



(d) Graph clustering on FGL-3SR's result.

Figure 3.8: (Top row) Learned graph with (a) IGL-3SR and (b) FGL-3SR. The node color corresponds to the average temperature in C° during all the period of observation. (Bottom row) Graph segmentation in two parts (red vs. green nodes) with spectral clustering using the Laplacian matrix learned by (c) IGL-3SR and (d) FGL-3SR.

of the region of Britanny (Figure 3.8 (c-d)), which is an expected natural segmentation.

The learned graphs can be also employed in the graph sampling task. Indeed, due to the constraints used in the optimization problem, the graph signals are bandlimited with respect to learned graphs. For instance, in this example the graph signals are 2-bandlimited. This property means that it is possible to select only 2 nodes and to reconstruct the graph signal values of the 30 remaining nodes using linear interpolation. Figure 3.9 displays an example of such reconstruction: thanks to the learned graph structure, the use of only 2 nodes allows to reconstruct sufficiently well the whole data matrix with a mean absolute error of 0.614. Again, this is a very interesting result that indirectly shows the quality of the learned graph.

7.5 Results on the ADHD dataset

In this third experiment, we consider the Attention Deficit Hyperactivity Disorder (ADHD) dataset [18] composed of functional Magnetic Resonance Imaging (fMRI) data. ADHD is a mental pathophysiology characterized by an excessive activity [23]. We study the resting-state fMRI of 20 subjects with ADHD and 20 healthy subjects available from `Nilearn` [1]. Each fMRI consists in a series of images measuring the brain activity. These images are processed as follows. First we split the brain into 39 Regions Of Interest (ROIs) with the Multi-Subject Dictionary Learning atlas [164] (see Figure 3.10a). Each ROI defines a node

(a) (b)

Figure 3.9: (a) The 2 nodes kept for the signal interpolation are shown in black. (b) The true signal at the target node (in red) shown on the left and its reconstruction using only the 2 selected nodes shown on the left (in black).

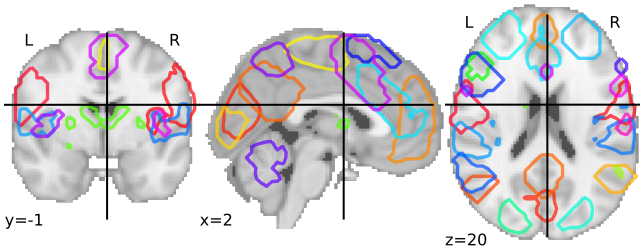
of our graph and the signal value at a certain node is an aggregation of the fMRI values over the associated ROI. For each subject, we therefore obtain a matrix in $\mathbb{R}^{n\times 39}$, where n stands for the number of images in the fMRI for the subject.

We then estimate the graph of each subject using such preprocessed data. Examples of learned graphs with FGL-3SR for an ADHD subject and a healthy subject are displayed in Figure 3.10. Visually, they reveal strong symmetric links between the right and left hemisphere of the brain. This phenomenon is common in resting-state fMRI where one hemisphere tends to correlate highly with the homologous anatomical location in the opposite hemisphere [41, 149]. Pointing out differences, though, the graph from the ADHD subject seems less structured and contains several spurious links (diagonal and north-south connections).

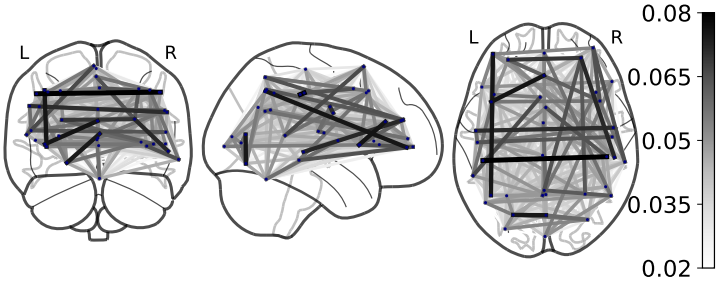
Aiming to better highlight the potential value of quality learned graphs for such studies, we proceed and use the Laplacian matrices of the brain graphs to classify the subjects, as proposed in several resting-state fMRI studies [2, 39]. First, we subtract the average graph for all subjects, which in fact removes the symmetrical connections common to all subjects), and then we use a 3-Nearest Neighbors classification algorithm. We use the correlation coefficient of Equation (3.22) as distance metric between Laplacian matrices, and a leave-one-out cross-validation strategy. The classification accuracy of the described approach reaches 65%. This level shall be compared with the performance obtained using simple correlation graphs [2] that, on these 40 subjects, leads to an accuracy of 52.5%. It appears that in this context the use of a more sophisticated graph learning process allows a subject characterization that goes beyond considering basic statistical correlation effects. Interestingly, this score is also comparable with state-of-the-art results reported in Sen et al. [145] for the same task, but on a larger database (67.3% of accuracy), using more sophisticated and specially-tailored processing steps, as well as carefully chosen classifiers.

7.6 Sigfox application

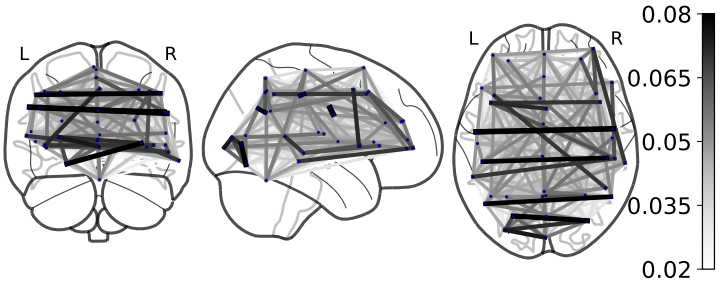
For the sake of completeness, we finally propose to apply our graph-learning method to the Sigfox training dataset used in Section 3.2 of the previous chapter. Recall that it consists in a set of approximately 35000 Sigfox messages recorded over one month at the level of 34 BSs and in this framework, one observation consists in a binary vector indicating which BS has received the message. To improve the speed of convergence, only 5000 messages



(a) Indicative Regions of Interest (ROIs) from Varoquaux et al. [164].



(b) ADHD subject.



(c) Healthy subject.

Figure 3.10: (a) Indicative ROIs from the Multi-Subject Dictionary Learning atlas extracted in Varoquaux et al. [164] with sparse dictionary learning. Results: Graphs returned by FGL-3SR, separately for (b) an ADHD patient and (c) a healthy subject, where darker edges indicate larger weights of connection.

are sampled randomly from this dataset. Before presenting the results, it should be taken into consideration that the proposed method has not been designed for this type of binary data. In particular, there is nothing here to suggest that smoothness can be characteristic of the underlying graph. In fact, many BS may have the same value 0 (did not receive the message), without being necessarily linked to each other. The learned graph is presented in figure 3.11.

We can observe that the spatial structure is globally preserved by the inferred graph, a potential indicator of a well-learned graph. Only the group of BSs located in the lower right corner (about 10 BSs at the same position) do not seem to respect this structure by establishing connections with BSs far away from them. Although it is difficult to explain the reason of such an observation, it should be noted that this set of BSs is located at Sigfox headquarters, a place where a very large number of messages are sent, and therefore received, in order to perform tests. It is therefore conceivable that such bias could have led to the observed results. Nevertheless, it remains very likely that the method presented

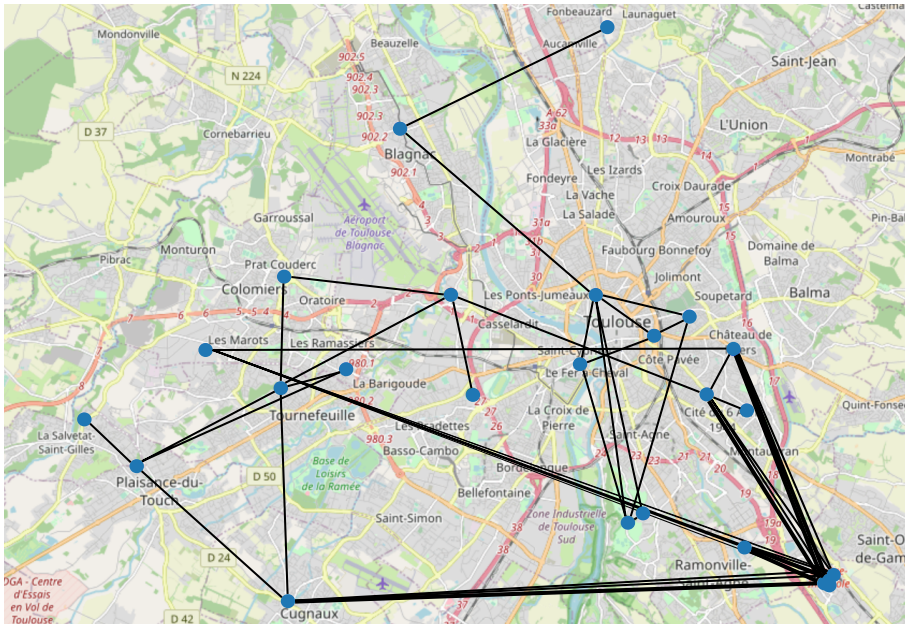


Figure 3.11: Graph obtained with FGL-3SR over the Sigfox training dataset.

here is not adapted to this type of data, reason why the work presented in the following chapter has been investigated.

8 Conclusions

This chapter presented a data-driven graph learning approach by employing a combination of two assumptions that are well known in the Graph Signal Processing framework. The first is the most standard in the literature of GSP and concerns the *smoothness* of the graph signals with respect to the underlying graph structure. The second is the *spectral sparsity* assumption, also referred as bandlimited property, which is notably known to carry the presence of clusters in real-world graphs. We proposed two algorithms to solve the corresponding optimization problem. The first one, IGL-3SR, effectively minimizes the objective function and has the advantage to decrease at each iteration. To address its low speed of convergence, we propose FGL-3SR that is a fast and scalable alternative. The findings of our empirical evaluation on synthetic data showed that the proposed approaches are as good or better performing than the reference state-of-the-art algorithms in term of reconstructing the unknown underlying graph and of computational cost (running time). Experiments on real-world benchmark use-cases suggest that our algorithms learn graphs that are useful and promising for any graph-based machine learning methodology, such as graph clustering, subsampling, etc.

The objective that was set in the introduction regarding the structural inference from graph signals has thus been well addressed here. Nevertheless, several objectives remain. We have seen that the proposed method was not necessarily well adapted to Sigfox data and therefore learning a graph that is suited to binary data seems important. Moreover, the task of event detection has been omitted in this chapter. For these reasons, the next chapter will try to combine the two tasks by proposing a method to detect change-points

in the underlying structure of binary graph vectors.

9 Technical proofs

This section provides the technical proofs of the different propositions exposed in the chapter.
Recall that lower case variables refer to vectors/scalars while upper case variables denote matrices. The table below provides the main notations used in the technical discussion that follows.

$x^\mathsf{T}, M^\mathsf{T}$	Transpose of vector x, matrix M.
$\mathrm{tr}(M)$	Trace of matrix M.
$\mathrm{diag}(x)$	Diagonal matrix containing the vector x.
$M_{k,l}$	(k,l)-th element of the matrix M.
$M_{k,:}$	k-th row of M.
$M_{:,l}$	l-th column of M.
$M_{k:,l:}$	Submatrix containing the elements of M from the k-th row to the last row, and from the l-th column to the last column.
$M \succeq 0$	M is a positive semi-definite matrix.
$M^\dagger$	The Moore-Penrose pseudoinverse of M.
e_k	Vector containing zeros except a 1 at position k.
I_n	Identity matrix of size n.
$\mathbf{0}_n$	Vector of size n containing only zeros.
$\mathbf{1}_n$	Vector of size n containing only ones.
$\mathbb{1}_A(\cdot)$	The indicator function over the set A.
$\|x\|_0$	The number of non-zero elements of a vector x.
$\|\cdot\|_F$	The Frobenius norm.
$\|\cdot\|_{2,0}$	The $\ell_{2,0}$-norm, with $\|M\|_{2,0} = \sum_{i=1} \mathbb{1}_{\{\|M_{i,:}\|_2 \neq 0\}}$.
$\|\cdot\|_{2,1}$	The $\ell_{2,1}$-norm, with $\|M\|_{2,1} = \sum_{i=1} \|M_{i,:}\|_2$.
∇f	Gradient of the function f.
$\langle \cdot, \cdot \rangle$	Inner product function.
$Orth(N)$	The set of all orthogonal matrices of size $N \times N$.
$\mathcal{O}(\cdot)$	Order of magnitude (e.g. of computational complexity).
τ	Number of iterations needed for an optimization procedure.

Table 3.2: Table of notations used throughout the chapter.

Lemma 3.1 – *Given $X, X_0 \in \mathbb{R}^{N \times N}$ two orthogonal matrices with first column equals to $\frac{1}{\sqrt{N}}\mathbf{1}_N$ (constraint (3.1a)), we have the following equality:*

$$X = X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & [X_0^\mathsf{T} X]_{2:,2:} \end{bmatrix},$$

with $[X_0^\mathsf{T} X]_{2:,2:}$ denoting the submatrix of $X_0^\mathsf{T} X$ containing everything but the first row and column of itself. Furthermore, remark that $[X_0^\mathsf{T} X]_{2:,2:}$ is in $Orth(N-1)$.

Proof. Let consider $X, X_0 \in \mathbb{R}^{N\times N}$ two orthogonal matrix with first column equals to $\frac{1}{\sqrt{N}}\mathbf{1}_N$. We have the following equalities:

$$X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & [X_0^\mathsf{T} X]_{2:,2:} \end{bmatrix} = \begin{bmatrix} \vdots & \vdots \\ X_{0(:,1)} & X_{0(:,2:)}[X_0^\mathsf{T} X]_{2:,2:} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots \\ \frac{1}{\sqrt{N}}\mathbf{1}_N & X_{:,2:} \\ \vdots & \vdots \end{bmatrix} = X \ .$$

Furthermore, thanks to the orthogonality of X and X_0, we have

$$[X_0^\mathsf{T} X]_{2:,2:} \left[[X_0^\mathsf{T} X]_{2:,2:}\right]^\mathsf{T} = X_{0,(2:,:)}^\mathsf{T} X_{:,2:} \left[X_{0,(2:,:)}^\mathsf{T} X_{:,2:}\right]^\mathsf{T} = X_{0,(2:,:)}^\mathsf{T} X_{:,2:} X_{:,2:}^\mathsf{T} [X_{0,(2:,:)}^\mathsf{T}]^\mathsf{T} = I_{N-1}.$$

By symmetry we conclude that $[X_0^\mathsf{T} X]_{2:,2:} \in Orth(N-1)$. □

Proposition 3.1 – *Given $X_0 \in \mathbb{R}^{N\times N}$ an orthogonal matrix with first column equals to $\frac{1}{\sqrt{N}}\mathbf{1}_N$, an equivalent formulation of optimization problem (3.1) is given by:*

$$\min_{H,U,\Lambda} \left\| Y - X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U \end{bmatrix} H \right\|_F^2 + \alpha \|\Lambda^{1/2} H\|_F^2 + \beta \|H\|_S \triangleq f(H, U, \Lambda) \ ,$$

$$\text{s.t.} \quad \begin{cases} U^\mathsf{T} U = I_{N-1} \ , & (a') \\ \left(X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U \end{bmatrix} \Lambda \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U^\mathsf{T} \end{bmatrix} X_0^\mathsf{T} \right)_{k,\ell} \leq 0 \quad k \neq \ell \ , & (b') \\ \Lambda = \mathit{diag}(0, \lambda_2, \ldots, \lambda_N) \succeq 0 \ , & (c) \\ \mathit{tr}(\Lambda) = N \in \mathbb{R}_*^+ \ . & (d) \end{cases}$$

Proof. From the previous lemma, we know that X can be decompose into two orthogonal matrices X_0 and $U = [X_0^\mathsf{T} X]_{2:,2:}$. Hence, we can optimize with respect to U instead of X and the second part of the constraint $(3.1a)$ is automatically satisfied. To make the equivalence, we just replace X from the main optimization problem to $X_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\mathsf{T} \\ \mathbf{0}_{N-1} & U \end{bmatrix}$ where U is now imposed to be orthogonal. □

Proposition 3.2 (Closed-form solution for the $\ell_{2,0}$ and $\ell_{2,1}$-norms) – *The solutions of problem (3.5) when $\|\cdot\|_S$ is set to $\|\cdot\|_{2,0}$ or $\|\cdot\|_{2,1}$ are given in the following.*

- *Using the $\ell_{2,0}$-norm, the optimal solution of (3.5) is given by the matrix $\widehat{H} \in \mathbb{R}^{N\times n}$ where for $1 \leq i \leq N$,*

$$\widehat{H}_{i,:} = \begin{cases} 0 & \text{if } \|(X^\mathsf{T} Y)_{i,:}\|_2^2/(1+\alpha\lambda_i) \leq \beta \ , \\ (X^\mathsf{T} Y)_{i,:}/(1+\alpha\lambda_i) & \text{else .} \end{cases}$$

- *Using the $\ell_{2,1}$-norm, the optimal solution of (3.5) is given by the matrix $\widehat{H} \in \mathbb{R}^{N\times n}$ where for $1 \leq i \leq N$,*

$$\widehat{H}_{i,:} = \frac{1}{1+\alpha\lambda_i}\Big(1 - \frac{\beta}{2}\frac{1}{\|(X^\mathsf{T} Y)_{i,:}\|_2}\Big)_+ (X^\mathsf{T} Y)_{i,:} \ ,$$

where $(t)_+ \triangleq \max\{0, t\}$ is the positive part function.

Proof. In the following, we suppose that $Y \neq 0$ since in this trivial case, the solution is simply given by $\widehat{H} = 0$.

Closed-form solution for the $\ell_{2,0}$. Recall that $\|H\|_{2,0} = \sum_{i=1} \mathbb{1}_{\{\|H_{i,:}\|_2 \neq 0\}}$, the objective function can be written as:

$$\begin{aligned}
f(X, \Lambda, H) &= \|X^\mathsf{T} Y - H\|_F^2 + \alpha \|\Lambda^{1/2} H\|_F^2 + \beta \|H\|_{2,0} \\
&= \|Y\|_F^2 + \sum_{i=1}^{N} \left(\sum_{j=1}^{n} \left(H_{i,j}^2 - 2(X^\mathsf{T} Y)_{i,j} H_{i,j} + \alpha \lambda_i H_{i,j}^2 \right) + \beta \mathbb{1}_{\{\|H_{i,:}\|_2 \neq 0\}} \right) \\
&= \|Y\|_F^2 + \sum_{i=1}^{N} \left(\|H_{i,:}\|_2^2 - 2\langle (X^\mathsf{T} Y)_{i,:}, H_{i,:} \rangle + \alpha \lambda_i \|H_{i,:}\|_2^2 + \beta \mathbb{1}_{\{\|H_{i,:}\|_2 \neq 0\}} \right) \\
&= \|Y\|_F^2 + \sum_{i=1}^{N} \left((1 + \alpha \lambda_i) \|H_{i,:}\|_2^2 - 2\langle (X^\mathsf{T} Y)_{i,:}, H_{i,:} \rangle + \beta \mathbb{1}_{\{\|H_{i,:}\|_2 \neq 0\}} \right) \\
&= \|Y\|_F^2 + \sum_{i=1}^{N} \tilde{f}_i(X, \Lambda, H_{i,:}) \ .
\end{aligned}$$

Our objective function is written as a sum of independent objective functions, each associated with a different $H_{i,:}$. Hence, we can optimize the problem for each i. Our problem for a given i is:

$$\min_{H_{i,:} \in \mathbb{R}^n} \quad (1 + \alpha \lambda_i) \|H_{i,:}\|_2^2 - 2\langle (X^\mathsf{T} Y)_{i,:}, H_{i,:} \rangle + \beta \mathbb{1}_{\{\|H_{i,:}\|_2 \neq 0\}} \ .$$

When we restrict the minimization to $\|H_{i,:}\|_2 = 0$, the unique solution is $\widehat{H}_{i,:} = \mathbf{0}_n$ and $\tilde{f}_i(X, \Lambda, \widehat{H}_{i,:}) = 0$.

When $\|H_{i,:}\|_2 \neq 0$, the objective function is convex and differentiable, thus it suffice to take the following derivative equal to 0:

$$\begin{aligned}
\frac{\partial}{\partial H_{i,:}} \tilde{f}_i(H_{i,:}) = 2(1 + \alpha \lambda_i) H_{i,:} - 2(X^\mathsf{T} Y)_{i,:} = 0 \ , \\
\widehat{H}_{i,:} = (X^\mathsf{T} Y)_{i,:} / (1 + \alpha \lambda_i) \ .
\end{aligned}$$

With this solution, the objective function $\tilde{f}_i$ is equal to:

$$\begin{aligned}
\tilde{f}(X, \Lambda, \widehat{H}_{i,:}) &= (1 + \alpha \lambda_i) \|(X^\mathsf{T} Y)_{i,:} / (1 + \alpha \lambda_i)\|_2^2 - 2\langle (X^\mathsf{T} Y)_{i,:}, (X^\mathsf{T} Y)_{i,:} / (1 + \alpha \lambda_i) \rangle + \beta \\
&= \frac{1}{1 + \alpha \lambda_i} \|(X^\mathsf{T} Y)_{i,:}\|_2^2 - \frac{2}{1 + \alpha \lambda_i} \|(X^\mathsf{T} Y)_{i,:}\|_2^2 + \beta \\
&= \beta - \frac{1}{1 + \alpha \lambda_i} \|(X^\mathsf{T} Y)_{i,:}\|_2^2 \ .
\end{aligned}$$

Hence, whenever $\dfrac{1}{1 + \alpha \lambda_i} \|(X^\mathsf{T} Y)_{i,:}\|_2^2 \leq \beta$, the objective function is positive, making $\widehat{H}_{i,:} = 0$ a better choice for the minimization and conversely. In conclusion, for all

$1 \leq i \leq N$, the solution is:

$$\widehat{H}_{i,:} = \begin{cases} 0 & \text{if } \|(X^\mathsf{T}Y)_{i,:}\|_2^2/(1+\alpha\lambda_i) \leq \beta \ , \\ (X^\mathsf{T}Y)_{i,:}/(1+\alpha\lambda_i) & \text{else .} \end{cases}$$

Closed-form solution for the $\ell_{2,1}$. Similarly to the $\ell_{2,0}$ case, the objective function can be decomposed by a sum of independent objectives functions.

$$\begin{aligned} f(X,\Lambda,H) &= \|X^\mathsf{T}Y - H\|_F^2 + \alpha\|\Lambda^{1/2}H\|_F^2 + \beta\|H\|_{2,1} \\ &= \|Y\|_F^2 + \sum_{i=1}^{N}\left(\sum_{j=1}^{n}\left(H_{i,j}^2 - 2(X^\mathsf{T}Y)_{i,j}H_{i,j} + \alpha\lambda_i H_{i,j}^2\right) + \beta\sqrt{\sum_{j=1} H_{i,j}^2}\right) \\ &= \|Y\|_F^2 + \sum_{i=1}^{N}\left(\|H_{i,:}\|_2^2 - 2\langle(X^\mathsf{T}Y)_{i,:}, H_{i,:}\rangle + \alpha\lambda_i\|H_{i,:}\|_2^2 + \beta\|H_{i,:}\|_2\right) \\ &= \|Y\|_F^2 + \sum_{i=1}^{N}\left((1+\alpha\lambda_i)\|H_{i,:}\|_2^2 - 2\langle(X^\mathsf{T}Y)_{i,:}, H_{i,:}\rangle + \beta\|H_{i,:}\|_2\right) \\ &= \|Y\|_F^2 + \sum_{i=1}^{N}\tilde{f}_i(X,\Lambda,H_{i,:}) \ . \end{aligned}$$

Again, we can optimize the problem for each row i of H independently. Our problem for a given i is:

$$\min_{H_{i,:}\in\mathbb{R}^n} \quad (1+\alpha\lambda_i)\|H_{i,:}\|_2^2 - 2\langle(X^\mathsf{T}Y)_{i,:}, H_{i,:}\rangle + \beta\|H_{i,:}\|_2 \ . \tag{3.23}$$

Although non-differentiable at $H_{i,:} = \mathbf{0}_n$, this function is convex and we need to find $H_{i,:}$ such that the vector $\mathbf{0}_n$ belongs to the subdifferential of $\tilde{f}_i$ denoted by $\partial\tilde{f}_i(H_{i,:})$ and is equal to:

$$\partial\tilde{f}_i(H_{i,:}) = \begin{cases} \mathcal{B}_2\big(-2(X^TY)_{i,:}, \beta\big) & \text{if } H_{i,:} = \mathbf{0}_n \ , \\ 2\big(1+\alpha\lambda_i + \dfrac{\beta}{2}\dfrac{1}{\|H_{i,:}\|_2}\big)H_{i,:} - 2(X^\mathsf{T}Y)_{i,:} & \text{otherwise .} \end{cases}$$

Where $\mathcal{B}_2$ stand for the ℓ_2-norm bowl.

Remark that when $\|(X^\mathsf{T}Y)_{i,:}\|_2 \leq \frac{\beta}{2}$, $\mathbf{0}_n \in \mathcal{B}_2\big(-2(X^TY)_{i,:}, \beta\big)$ and thus in this case $\widehat{H}_{i,:} = \mathbf{0}_n$.

On the contrary, when $\|(X^\mathsf{T}Y)_{i,:}\|_2 > \frac{\beta}{2}$, we must find $H_{i,:}$ such that:

$$\big(1+\alpha\lambda_i + \frac{\beta}{2}\frac{1}{\|H_{i,:}\|_2}\big)H_{i,:} = (X^\mathsf{T}Y)_{i,:} \ .$$

By tacking the norm of the previous equation, we obtain

$$\begin{aligned}
&\big(1+\alpha\lambda_i+\frac{\beta}{2}\frac{1}{\|H_{i,:}\|_2}\big)\|H_{i,:}\|_2 = \|(X^\mathsf{T}Y)_{i,:}\|_2 \\
\iff &(1+\alpha\lambda_i)\|H_{i,:}\|_2+\frac{\beta}{2} = \|(X^\mathsf{T}Y)_{i,:}\|_2 \\
\iff &\|H_{i,:}\|_2 = \big(\|(X^\mathsf{T}Y)_{i,:}\|_2-\frac{\beta}{2}\big)/(1+\alpha\lambda_i) > 0\ .
\end{aligned}$$

We can now replace $\|H_{i,:}\|_2$ in the initial equation and get $H_{i,:}$.

$$\begin{aligned}
&\big(1+\alpha\lambda_i+\frac{\beta(1+\alpha\lambda_i)}{2\|(X^\mathsf{T}Y)_{i,:}\|_2-\beta}\big)H_{i,:} = \frac{(1+\alpha\lambda_i)\|(X^\mathsf{T}Y)_{i,:}\|_2}{\|(X^\mathsf{T}Y)_{i,:}\|_2-\beta/2}H_{i,:} = (X^\mathsf{T}Y)_{i,:} \\
\iff &H_{i,:} = \frac{\|(X^\mathsf{T}Y)_{i,:}\|_2-\beta/2}{(1+\alpha\lambda_i)\|(X^\mathsf{T}Y)_{i,:}\|_2}(X^\mathsf{T}Y)_{i,:} = \frac{1}{1+\alpha\lambda_i}\Big(1-\frac{\beta}{2}\frac{1}{\|(X^\mathsf{T}Y)_{i,:}\|_2}\Big)(X^\mathsf{T}Y)_{i,:}\ ,
\end{aligned}$$

which concludes the proof. □

Proposition 3.3 (Euclidean gradient with respect to U) – *The Euclidean gradient of f and ϕ with respect to U are*

$$\nabla_U f(H,U,\Lambda) = -2\big[(HY^\mathsf{T}X_0)_{2:,2:}\big]^\mathsf{T} + 2U(HH^\mathsf{T})_{2:,2:}\ ,$$

$$\nabla_U \phi(U,\Lambda) = -\sum_{k=1}^{N-1}\sum_{\ell>k}^{N}\frac{\Big(B_{k,\ell}+B_{k,\ell}^\mathsf{T}\Big)U\Lambda_{2:,2:}}{h(U,\Lambda)_{k,\ell}}\ .$$

with $\forall 1\le k,\ell\le N, B_{k,\ell} = \big(X_0^\mathsf{T}e_k e_\ell^\mathsf{T}X_0\big)_{2:,2:}$, *and* $h(\cdot)$ *from Definition 3.2.*

Proof. We begin by computing the gradient of the main objective, with respect to U. Recall the objective function with respect to U:

$$f(H,U,\Lambda) = -2\mathrm{tr}(Y^\mathsf{T}X_0\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U\end{bmatrix}H) + \mathrm{tr}(H^\mathsf{T}\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U^\mathsf{T}U\end{bmatrix}H)\ .$$

The corresponding gradient is the following.

$$\begin{aligned}
\nabla_U f(H,U,\Lambda) &= -2\nabla_U\mathrm{tr}(Y^\mathsf{T}X_0\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U\end{bmatrix}H) + \nabla_U\mathrm{tr}(H^\mathsf{T}\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U^\mathsf{T}U\end{bmatrix}H) \\
&= -2\nabla_U\mathrm{tr}\left(HY^\mathsf{T}X_0\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U\end{bmatrix}\right) + \nabla_U\mathrm{tr}\left(HH^\mathsf{T}\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U^\mathsf{T}U\end{bmatrix}\right) \\
&= -2\nabla_U\left((HY^\mathsf{T}X_0)_{1,1}\cdot 1 + \mathrm{tr}\Big((HY^\mathsf{T}X_0)_{2:,2:}U\Big)\right) \\
&\quad + \nabla_U\left((HH^\mathsf{T})_{1,1}\cdot 1 + \mathrm{tr}\big((HH^\mathsf{T})_{2:,2:}U^\mathsf{T}U\big)\right) \\
&= -2\big[(HY^\mathsf{T}X_0)_{2:,2:}\big]^\mathsf{T} + 2U(HH^\mathsf{T})_{2:,2:}\ .
\end{aligned}$$

We now derive the gradient of the barrier function $\phi(U,\Lambda)$ with respect to U:

$$\begin{aligned}\nabla_U \phi(U,\Lambda) &= -\sum_{k=1}^{N-1}\sum_{\ell>k}^{N} \nabla_U \log\Big(-h(U,\Lambda)_{k,\ell}\Big) \\ &= -\sum_{k=1}^{N-1}\sum_{\ell>k}^{N} \frac{1}{h(U,\Lambda)_{k,\ell}} \nabla_U h(U,\Lambda)_{k,\ell}\ .\end{aligned}$$

We can write the h function as:

$$\begin{aligned}h(U,\Lambda)_{k,\ell} &= \big\langle e_k e_\ell^\mathsf{T}, h(U,\Lambda)\big\rangle = \Big\langle X_0^\mathsf{T} e_k e_\ell^\mathsf{T} X_0, \begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U\end{bmatrix}\Lambda\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U\end{bmatrix}^\mathsf{T}\Big\rangle \\ &= \Big\langle X_0^\mathsf{T} e_k e_\ell^\mathsf{T} X_0, \begin{bmatrix}\lambda_1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U\Lambda_{2:,2:}U^\mathsf{T}\end{bmatrix}\Big\rangle = \mathrm{tr}\Big(X_0^\mathsf{T} e_\ell e_k^\mathsf{T} X_0 \begin{bmatrix}0 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & U\Lambda_{2:,2:}U^\mathsf{T}\end{bmatrix}\Big) \\ &= \big(X_0^\mathsf{T} e_\ell e_k^\mathsf{T} X_0\big)_{1,1}\cdot 0 + \mathrm{tr}\Big(\big(X_0^\mathsf{T} e_\ell e_k^\mathsf{T} X_0\big)_{2:,2:} U\Lambda_{2:,2:}U^\mathsf{T}\Big) \\ &= \mathrm{tr}\Big(B_{k,\ell}^\mathsf{T} U\Lambda_{2:,2:}U^\mathsf{T}\Big)\ .\end{aligned}$$

In conclusion we have $\nabla_U h(U,\Lambda)_{k,\ell} = \Big(B_{k,\ell}+B_{k,\ell}^\mathsf{T}\Big)U\Lambda_{2:,2:}$, which finishes the proof. □

Proposition 3.4 (Feasible eigenvalues) – *Given any $X \in \mathbb{R}^{N\times N}$ being an orthogonal matrix with first column equals to $1/\sqrt{N}$ (constraint (3.1a)), there always exist a matrix $\Lambda \in \mathbb{R}^{N\times N}$ such that the following constraints are satisfied:*

$$\begin{cases} (X\Lambda X^\mathsf{T})_{i,j} \le 0 \quad i\neq j\ , & (3b)\\ \Lambda = \mathit{diag}(0,\lambda_2,\dots,\lambda_N) \succeq 0\ , & (3c)\\ \mathit{tr}(\Lambda) = c \in \mathbb{R}_*^+\ . & (3d)\end{cases}$$

Proof. Let us consider a positive real value $c>0$. Taking $\Lambda = \mathrm{diag}(0,c,\dots,c)/(N-1)$ leads to $\mathrm{tr}(\Lambda)=c$ and $\forall i\neq j, (X\Lambda X^\mathsf{T})_{i,j} = -c/N < 0$. However, this solution with constant eigenvalues actually corresponds to the complete graph. For our purpose, it is the worst case scenario as it contains no structural information between the nodes. □

Proposition 3.5 (Closed-form solution of problem (3.11)) – *Consider the optimization problem (3.11). Let X_0 be any matrix that belongs to the constraints set (a), and $M = (X_0^\mathsf{T} Y H^\mathsf{T})_{2:,2:}$ the submatrix containing everything but the input's first row and first column. Finally, let PDQ^T be the SVD of M. Then, the problem admits the following closed form solution*

$$\widehat{X} = X_0\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & PQ^\mathsf{T}\end{bmatrix}\ .$$

Proof. One can observe that the relaxed optimization problem is equivalent to finding:

$$\widehat{G} = \underset{G}{\mathrm{argmin}}\Big\| Y - X_0 \underbrace{\begin{bmatrix}1 & \mathbf{0}_{N-1}^\mathsf{T}\\ \mathbf{0}_{N-1} & G\end{bmatrix}}_{\triangleq\ \tilde{G}} H\Big\|_F^2\ , \tag{3.24}$$

s.t. $G^\mathsf{T} G = I_{N-1}$. This is obtained by replacing X with $X_0 \tilde{G}$.

Solving the above Equation (3.24) is equivalent to finding:

$$\widehat{G} = \arg\max_G \operatorname{tr}\left(HY^\mathsf{T} X_0 \tilde{G}\right) = \arg\max_G \operatorname{tr}\left(M^\mathsf{T} G\right),$$

s.t. $G^\mathsf{T}\widehat{G} = I_{N-1}$. Then, as proved in [180], we finally have $G^* = PQ^\mathsf{T}$, which completes the proof. □

Lemma 3.2 – *Assume the proposed Model (3.14). If $p_1 = 0$ and $p_i \in (0,1), \forall i \geq 2$, then,*

$$\begin{aligned} -\log(p(h|y,X,\Lambda)) \propto \frac{1}{\sigma^2}\|y - Xh\|_2^2 + \frac{1}{2}h^\mathsf{T}\Lambda h \\ + \sum_{i=1}^{N} \mathbb{1}_{\{h_i \neq 0\}} \left(p_i \log(\frac{\lambda_i}{\sqrt{2\pi}}) - \log(p_i) - \log(\frac{\lambda_i}{\sqrt{2\pi}}) \right) . \end{aligned}$$

Proof. Based on the Factor Analysis model and the independence of h_i's,

$$\begin{aligned} \log(p(h|y,X,\Lambda)) &\propto \log\left(p(y|h,X,\Lambda)\right) + \log\left(p(h|X,\Lambda)\right) \\ &\propto -\frac{1}{2\sigma^2}\|y - Xh\|_2^2 + \sum_{i=1}^{N} \log\left(p(h_i|\lambda_i)\right) . \end{aligned} \tag{3.25}$$

Let us now focus on $\log\left(p(h_i|\lambda_i)\right)$, for which we have:

$$\begin{aligned} \log\left(p(h_i|\lambda_i)\right) &= \log\left(\sum_{\gamma_i=\{0,1\}} p(h_i,\gamma_i|\lambda_i) \right) \\ &= \log\left(\sum_{\gamma_i=\{0,1\}} p(h_i,\gamma_i|\lambda_i) \frac{p(\gamma_i|h_i,\lambda_i)}{p(\gamma_i|h_i,\lambda_i)} \right) \\ &\underset{(=)}{\geq} \sum_{\gamma_i=\{0,1\}} p(\gamma_i|h_i,\lambda_i) \log\left(\frac{p(h_i,\gamma_i|\lambda_i)}{p(\gamma_i|h_i,\lambda_i)} \right) . \end{aligned}$$

The last equality is obtain using the concavity of the logarithm and Jensen inequality. For this particular case, it correspond to an equality. Then we have:

$$\begin{aligned} \log\left(p(h_i|\lambda_i)\right) = &\sum_{\gamma_i=\{0,1\}} p(\gamma_i|h_i,\lambda_i) \log\left(p(h_i,\gamma_i|\lambda_i)\right) \quad (\star) \\ &- \sum_{\gamma_i=\{0,1\}} p(\gamma_i|h_i,\lambda_i) \log\left(p(\gamma_i|h_i,\lambda_i)\right) . \quad (\star\star) \end{aligned}$$

Before computing the previous two sums, we need to observe that:

$$p(\gamma_i = 1|h_i) = \begin{cases} 1 & \text{if} \quad h_i \neq 0 , \\ p_i & \text{if} \quad h_i = 0 . \end{cases}$$

We can now compute $(\star)$ and $(\star\star)$ as follows:

$$\begin{aligned}(\star) &= \sum_{\gamma_i=\{0,1\}} p(\gamma_i|h_i,\lambda_i)\left[\log\left(p(h_i|\gamma_i,\lambda_i)\right)+\log\left(p(\gamma_i|\lambda_i)\right)\right]\\ &= \left(\mathbb{1}_{\{h_i\neq 0\}}+p_i\mathbb{1}_{\{h_i=0\}}\right)\left[\log\left(\frac{\lambda_i}{\sqrt{2\pi}}\right)-\frac{1}{2}\lambda_i h_i^2+\log\left(p_i\right)\right]\\ &\qquad+\left((1-p_i)\mathbb{1}_{\{h_i=0\}}\right)\left[\log\left(\mathbb{1}_{\{h_i=0\}}\right)+\log\left(1-p_i\right)\right]\\ (\star\star) &= \left[p_i\log(p_i)+(1-p_i)\log(1-p_i)\right]\mathbb{1}_{\{h_i=0\}}\ .\end{aligned}$$

Finally we can compute $\log\left(p(h_i|\lambda_i)\right)$:

$$\begin{aligned}\log\left(p(h_i|\lambda_i)\right) &= (\star)-(\star\star)\\ &= \mathbb{1}_{\{h_i\neq 0\}}\left(\log\left(\frac{\lambda_i}{\sqrt{2\pi}}\right)-\frac{1}{2}\lambda_i h_i^2+\log\left(p_i\right)\right)+p_i\log\left(\frac{\lambda_i}{\sqrt{2\pi}}\right)\mathbb{1}_{\{h_i=0\}}\\ &= \mathbb{1}_{\{h_i\neq 0\}}\left(\log\left(\frac{\lambda_i}{\sqrt{2\pi}}\right)+\log\left(p_i\right)-p_i\log\left(\frac{\lambda_i}{\sqrt{2\pi}}\right)\right)+p_i\log\left(\frac{\lambda_i}{\sqrt{2\pi}}\right)\\ &\qquad\qquad+-\frac{1}{2}\lambda_i h_i^2\\ &\propto \mathbb{1}_{\{h_i\neq 0\}}\left(\log\left(\frac{\lambda_i}{\sqrt{2\pi}}\right)+\log\left(p_i\right)-p_i\log\left(\frac{\lambda_i}{\sqrt{2\pi}}\right)\right)-\frac{1}{2}\lambda_i h_i^2\ .\end{aligned}$$

Note that with our parametrization, the particular case $i=1$ leads to $\log\left(p(h_1|\lambda_1)\right)=0$. Now plugging our result in equation (3.25) and multiplying on both side by -1, we get our final result. □

Proposition 3.6 (A posteriori distribution of h) – *Let $C>0$, and assume for all $i\geq 2$ that $p_i=e^{-C}$ if $\lambda_i=\sqrt{2\pi}$ and $p_i=-W\left(-\frac{e^{-C}\log(\lambda_i/\sqrt{2\pi})}{\lambda_i/\sqrt{2\pi}}\right)/\log(\lambda_i/\sqrt{2\pi})$ if not. Then, $p_i\in(0,1)$ and there exist constants $\alpha,\beta>0$ such that:*

$$-\log(p(h|y,X,\Lambda))\propto\|y-Xh\|_2^2+\alpha h^{\mathsf{T}}\Lambda h+\beta\|h\|_0\ .$$

Proof. To show that the p_i's are well-defined and belongs to $(0,1)$, it suffices to apply Lemma 3.3 with $x=\lambda_i/\sqrt{2\pi}$.

We now proof the main result of the proposition. If $\lambda_i=\sqrt{2\pi}$, then $p_i=e^{-C}<1$ and

$$p_i\log(\frac{\lambda_i}{\sqrt{2\pi}})-\log(p_i)-\log(\frac{\lambda_i}{\sqrt{2\pi}})=-\log(p_i)=C\ .$$

If $\lambda_i \neq \sqrt{2\pi}$, then $-p_i \log(\lambda_i/\sqrt{2\pi}) = W\left(-\frac{e^{-C}\log(\lambda_i/\sqrt{2\pi})}{\lambda_i/\sqrt{2\pi}}\right)$. Since W corresponds to the inverse function of $f(W) = We^W$, we have:

$$
\begin{aligned}
& -p_i \log(\lambda_i/\sqrt{2\pi}) e^{-p_i \log(\lambda_i/\sqrt{2\pi})} = -\frac{e^{-C}\log(\lambda_i/\sqrt{2\pi})}{\lambda_i/\sqrt{2\pi}} \\
\iff & \left|-p_i \log(\lambda_i/\sqrt{2\pi}) e^{-p_i \log(\lambda_i/\sqrt{2\pi})}\right| = \left|-\frac{e^{-C}\log(\lambda_i/\sqrt{2\pi})}{\lambda_i/\sqrt{2\pi}}\right| \\
\iff & \log\left(p_i \left|\log(\lambda_i/\sqrt{2\pi})\right| e^{-p_i \log(\lambda_i/\sqrt{2\pi})}\right) = \log\left(\frac{e^{-C}\left|\log(\lambda_i/\sqrt{2\pi})\right|}{\lambda_i/\sqrt{2\pi}}\right) \\
\iff & \log(p_i) + \log\left(\left|\log(\lambda_i/\sqrt{2\pi})\right|\right) - p_i \log(\lambda_i/\sqrt{2\pi}) \\
& \quad = -C + \log\left(\left|\log(\lambda_i/\sqrt{2\pi})\right|\right) - \log(\lambda_i/\sqrt{2\pi}) \; .
\end{aligned}
$$

Same as the case where $\lambda_i = \sqrt{2\pi}$, the final equality gives us:

$$
p_i \log(\frac{\lambda_i}{\sqrt{2\pi}}) - \log(p_i) - \log(\frac{\lambda_i}{\sqrt{2\pi}}) = C \; . \tag{3.26}
$$

Plugging the equation (3.26) into the final result of proposition 1, we obtain:

$$
\begin{aligned}
-\log(p(h|y, X, \Lambda)) &\propto \frac{1}{2\sigma^2}\|y - Xh\|_2^2 + \frac{1}{2}h^\mathsf{T}\Lambda h + C\|h\|_0 \\
&\propto \|y - Xh\|_2^2 + \alpha h^\mathsf{T}\Lambda h + \beta\|h\|_0 \; ,
\end{aligned}
$$

taking $\alpha = \sigma^2$ and $\beta = 2C\sigma^2$. This concludes the proof. □

Lemma 3.3. *Let $C > 0$. For any $x > 0$,*

$$
0 \leq -W\left(-\frac{e^{-C}\log(x)}{x}\right) / \log(x) \leq 1 \; . \tag{3.27}
$$

Proof. First, we show that this function is decreasing for $x > 0$. The derivative of the function is given by

$$
\frac{\partial}{\partial x} - W\left(-\frac{e^{-C}\log(x)}{x}\right) / \log(x) = \frac{W\left(-\frac{e^{-C}\log(x)}{x}\right)\left(W\left(-\frac{e^{-C}\log(x)}{x}\right) + \log(x)\right)}{x\log^2(x)\left(W\left(-\frac{e^{-C}\log(x)}{x}\right) + 1\right)} \; . \tag{3.28}
$$

For $x > 0$ and $C > 0$,

$$
-1/e < -e^{-(C+1)} = \min_{x>0} -\frac{e^{-C}\log(x)}{x} \leq -\frac{e^{-C}\log(x)}{x} \; . \tag{3.29}
$$

As $W(\cdot)$ is strictly increasing for $x > -1/e$, we have $W\left(-\frac{e^{-C}\log(x)}{x}\right) > W(-1/e) = -1$. Hence, the bottom part of the previous equation is always positive.

For $0 < x \leq 1$, $W\left(-\frac{e^{-C}\log(x)}{x}\right)$ is positive. Furthermore,

$$-e^{-C}\frac{\log(x)}{x} < -\frac{\log(x)}{x} \iff W\left(-e^{-C}\frac{\log(x)}{x}\right) < W\left(\frac{-\log(x)}{x}\right) = -\log(x) \tag{3.30}$$

$$\iff W\left(-e^{-C}\frac{\log(x)}{x}\right) + \log(x) < 0\ . \tag{3.31}$$

Hence, when $0 < x \leq 1$, the upper part of the previous equation is negative.

For $1 < x \leq e$, $W\left(-\frac{e^{-C}\log(x)}{x}\right)$ is negative. Furthermore,

$$-\frac{1}{e} \leq -\frac{\log(x)}{x} < -e^{-C}\frac{\log(x)}{x} \iff W\left(-\frac{\log(x)}{x}\right) = -\log(x) < W\left(-e^{-C}\frac{\log(x)}{x}\right) \tag{3.32}$$

$$\iff W\left(-e^{-C}\frac{\log(x)}{x}\right) + \log(x) > 0\ . \tag{3.33}$$

Hence, when $1 < x \leq e$, the upper part of the previous equation is negative again.

For $x > e$, $W\left(-\frac{e^{-C}\log(x)}{x}\right)$ is negative. Furthermore, $W\left(-\frac{e^{-C}\log(x)}{x}\right) > -1$ and $\log(x) > 1$. Hence, the addition is positive and the upper part of the previous equation is negative again.

We just have shown that the derivative is negative for $x > 0$. Hence, the initial function is decreasing on this interval. We now go back to the initial inequality (3.27). The left part of the inequality is straightforward as for x large enough, the function corresponds to the product of two positive functions. The function being decreasing, the lower bound follows. For the upper bound, let us remind that for $y > e$, we have the inequality $W(y) < \log(y)$ [81]. Let $f(x) = -\frac{e^{-C}\log(x)}{x}$, for x small enough we have:

$$\begin{aligned} W\left(f(x)\right) < \log(f(x)) \ &\Leftrightarrow\ -W\left(f(x)\right) > -\log(f(x)) \\ &\Leftrightarrow -W\left(f(x)\right)/\log(x) < -\log(f(x))/\log(x)\ . \end{aligned}$$

Tacking the limit when $x \longrightarrow 0_+$ conclude the proof,

$$\begin{aligned} \lim_{x\to 0_+} -\log(f(x))/\log(x) &= \lim_{x\to 0_+} -\log(-\frac{e^{-C}\log(x)}{x})/\log(x) \\ &= \lim_{x\to 0_+} -\Big(\log(e^{-C}) + \log(-\log(x)) - \log(x)\Big)/\log(x) \\ &= \lim_{x\to 0_+} \frac{C}{\log(x)} + \frac{\log(\log(1/x))}{\log(1/x)} + 1 = 1\ . \end{aligned}$$

□

4

Detecting changes in the graph structure of a varying Ising model

Contents

Abstract

This chapter is dedicated to the estimation of changes in the underlying structure of a series of binary graph signals, assumed to be drawn from the probability distribution known as Ising model. In particular, the present work focuses on the estimation of multiple change-points in a time-varying Ising model that evolves piece-wise constantly. The aim is to identify both the moments at which significant changes occur in the Ising model, as well as the underlying graph structures. For this purpose, we propose to estimate the neighborhood of each node by maximizing a penalized version of its conditional log-likelihood. The objective of the penalization is twofold: it imposes sparsity in the learned graphs and, thanks to a fused-type penalty, it also enforces them to evolve piece-wise constantly. Using few assumptions, we provide two change-points consistency theorems. Those are the first in the context of unknown number

of change-points detection in time-varying Ising model. Finally, experimental results on several synthetic datasets and a real-world dataset demonstrate the performance of our method.

Associated publication:
Learning the piece-wise constant graph structure of a varying Ising model [104],
Le Bars, Batiste, Humbert Pierre, Kalogeratos, Argyris and Vayatis, Nicolas
Accepted in *Proceedings of the* 37^{th} *International Conference on Machine Learning (ICML).*

1 Introduction

Graphs are fundamental tools to model and study static or varying relationships between variables of potentially high-dimensional vector data. They have many applications in physics, computer vision and statistics [34, 115]. In the static scenario, learning relationships between variables is referred to as *graph inference* and emerges in many fields such as in graph signal processing [50, 103], in probabilistic modeling, or in physics and biology [52, 137]. In this work, we consider a probabilistic framework where the observed data are drawn from an *Ising model*, a discrete *Markov Random Field* (MRF) with $\{-1, 1\}$-outputs. MRF are undirected probabilistic graphical models [97] where a set of random variables is represented as different nodes of a graph. An edge between two nodes in this graph indicates the conditional dependency between the two corresponding random variables, given the other variables.

Learning the structure of an MRF using a set of observations has been widely investigated [16, 117]. In particular for Gaussian graphical models [135, 179] with the well-known graphical lasso [61]. The Ising model inference task has also been addressed in the past [66, 79, 132, 166, 175]. However, previous methods do not consider the case where the underlying structure is evolving through time.

Over the past years, there has been a burst of interest in learning the structure of time-varying MRF [73, 177]. This task combined with the *change-point detection*, which is the detection of the moments in time at which significant changes in the graph structure occur, is of particular interest. Those have been widely investigated for piece-wise constant Gaussian graphical models [63, 113, 168], in all types of the change-point detection objectives: single change-point [27], multiple change-points [64], offline detection [63, 95], online detection [90], etc.

The advancements related to the time-varying Ising model are though limited. Especially, the combination of multiple change-points detection and structure inference has not been studied properly in the past. In [5, 96], the authors learn the parameters of a time-varying Ising model without looking for change-points since the network is allowed to change at each timestamp. In [59], the authors assume that the change-point location is known and only focus on the inference of the structural changes between Ising models. More recently, the problem of detecting a single change-point has been studied in [138].

Contribution. This work focuses on the estimation of multiple change-points in a time-varying Ising model that evolves piece-wise constantly. The aim is to identify both the moments at which significant changes occur in the Ising model, as well as the underlying graph structure of the model among consecutive change-points. Our work extends the

work in [63, 95] on Gaussian graphical models, to the case of an Ising model. We also derive two change-points consistency theorems that, to our knowledge, we are the first to demonstrate. More specifically, our method follows a "node-wise regression" approach [132] and estimates the neighborhood of each node by maximizing a penalized version of its conditional log-likelihood. The penalization allows us to efficiently recover sparse graphs and, thanks to the use of a group-fused penalty [19, 75, 95], as well to recover the change-points. The proposed method is referred as TVI-FL, which stands for Time-Varying Ising model identified with Fused and Lasso penalties.

Organization. The chapter is organized as follows. First, we briefly recall important properties of the static Ising model and describe its piece-wise constant version. Second, we present our methodology to infer the piece-wise constant graph structure over time and the moments in time at which significant changes occur. Next, we present our main theoretical results which consist in two change-point consistency theorems. Finally, we empirically demonstrate, on multiple synthetic datasets and a real-world problem, that our method is the best suited to recover both structure and change-points.

2 The time-varying Ising model

The *static Ising model* is a discrete MRF with $\{-1, 1\}$-outputs. This model is defined by a graph $G = (\mathcal{V}, \mathcal{E})$ where an edge between two nodes indicates that the two corresponding random variables are dependent given the other ones. We associate this graph to a symmetric weight matrix $\Omega \in \mathbb{R}^{p\times p}$ whose non-zero elements correspond to the set of edges $\mathcal{E}$. Formally, we have $\omega_{ab} \neq 0$ iff $(a, b) \in \mathcal{E}$ where ω_{ab} stands for the (a, b)-th element of Ω. An Ising model is thus entirely described by its associated weight matrix Ω. Let $X \sim \mathcal{I}(\Omega)$ be a random vector following an Ising model with weight matrix Ω. Let $x \in \{-1, 1\}^p$ be a realization and x_a, x_b respectively its a-th and b-th coordinates. Then, its probability function is given by:

$$\mathbb{P}_\Omega(X = x) = \frac{1}{Z(\Omega)} \exp\left\{\sum_{a<b} x_a x_b \omega_{ab}\right\}, \tag{4.1}$$

where $Z(\Omega) = \sum_{x\in\{-1,1\}^p} \exp\left\{\sum_{a<b} x_a x_b \omega_{ab}\right\}$ is the normalizing constant. For clarity in the following we denote $\mathbb{P}_\Omega(X = x) = \mathbb{P}_\Omega(x)$.

A *time-varying Ising model* is defined by a set of n graphs $G^{(i)} = (\mathcal{V}, \mathcal{E}^{(i)})$, $i \in \{1, \ldots, n\}$ over a fixed set of nodes V through a time-varying set of edges $\{\mathcal{E}^{(i)}\}_{i=1}^n$. Similarly to the static case, each $G^{(i)}$ is associated to a symmetric weight matrix $\Omega^{(i)} \in \mathbb{R}^{p\times p}$ and a distribution $\mathbb{P}_{\Omega^{(i)}}$ given by Eq. (4.1). A random variable associated to this model is a set of n independent random vectors $X^{(i)} \sim \mathcal{I}(\Omega^{(i)})$. A single realization is therefore a set of n vectors, each denoted by $x^{(i)} \in \{-1, 1\}^p$.

In the sequel, we assume in addition that the model is *piece-wise constant*, i.e. there exist a collection of D timestamps $\mathcal{D} \triangleq \{T_1, \ldots, T_D\} \subset \{2, \ldots, n\}$, sorted in ascending order, and a set of symmetric matrices $\{\Theta^{(j)}\}_{j=1}^{D+1}$ such that $\forall i \in \{1, \ldots, n\}$:

$$\Omega^{(i)} = \sum_{j=0}^{D} \Theta^{(j+1)} \mathbb{1}\{T_j \leq i < T_{j+1}\}, \tag{4.2}$$

where $T_0 = 1, T_{D+1} = n+1$. $\mathcal{D}$ thus corresponds to the set of change-points. According to Eq. (4.2), for a fixed $j \in \{0, \dots, D\}$, the set $\left\{x^{(i)} : T_j \leq i < T_{j+1}\right\}$ contains i.i.d. vectors drown from $\mathbb{P}_{\Theta^{(j+1)}}$.

3 Learning Methodology

Assuming the observation of a single realization $\{x^{(i)}\}_{i=1}^n$ of the described time-varying model at each timestamp, our objective is twofold. We want to recover the set of change-points $\mathcal{D}$, as well as the graph structure underlying the observed data vectors, i.e. which edges are activated at each timestamp. In practice, we may observe multiple data vectors at each timestamp. However, since this does not change our analysis, we leave the related discussion for the experimental section. Next, we now describe our methodology to perform the aforementioned tasks.

Neighborhood selection strategy. Due to the intractability of the normalizing constant $Z(\cdot)$, classical maximum likelihood approaches are difficult to apply in practice. Hence, an intuitive approach is to extend the neighborhood selection strategy introduced for the static setting in [132] to our time-varying setting. Instead of maximizing the global likelihood of Eq. (4.1), this approach maximizes, for each node $a \in V$, the conditional likelihood of the node knowing the other nodes in $V \setminus a$. The conditional probability of observing a node's value, knowing the others, when $X \sim \mathcal{I}(\Omega)$, is:

$$\mathbb{P}_{\omega_a}(x_a | x_{\setminus a}) = \frac{\exp\left\{2x_a \sum_{b \in V \setminus a} x_b \omega_{ab}\right\}}{\exp\left\{2x_a \sum_{b \in V \setminus a} x_b \omega_{ab}\right\} + 1}, \tag{4.3}$$

where ω_a denotes the a-th column (or row) of Ω that is used to parametrize the probability function of Eq. (4.3). Here, $x_{\setminus a}$ denotes the vector x without the coordinate a.

For each node, we thus propose to maximize a penalized version of the conditional likelihood of Eq. 4.3. The detailed procedure is explained below.

3.1 Optimization program

The neighborhood selection strategy works as follows. For each node $a = 1, \dots, p$, we solve the regularized optimization program:

$$\widehat{\omega}_a = \underset{\omega \in \mathbb{R}^{p-1 \times n}}{\arg\min}\ \mathcal{L}_a(\omega) + pen_{\lambda_1, \lambda_2}(\omega). \tag{4.4}$$

In this equation, $\mathcal{L}_a(\omega)$ stands for the node-wise negative conditional log-likelihood of node a, knowing $x_{\setminus a}^{(i)}$:

$$\mathcal{L}_a(\omega) \triangleq -\sum_{i=1}^n \log\left(\mathbb{P}_{\omega^{(i)}}(x_a^{(i)} | x_{\setminus a}^{(i)})\right) \tag{4.5}$$

$$= \sum_{i=1}^n \log\left\{\exp\left(\omega^{(i)\top} x_{\setminus a}^{(i)}\right) + \exp\left(-\omega^{(i)\top} x_{\setminus a}^{(i)}\right)\right\} - \sum_{i=1}^n x_a^{(i)} \omega^{(i)\top} x_{\setminus a}^{(i)}, \tag{4.6}$$

where $\omega^{(i)}$ is the i-th column of ω. The last line is obtained by plugging Eq. (4.3) in Eq. (4.5) with $\omega^{(i)}$ instead of ω_a.

With such objective function, we learn at each timestamp i the neighborhood $\omega^{(i)}$ of node a via a penalized logistic regression method.

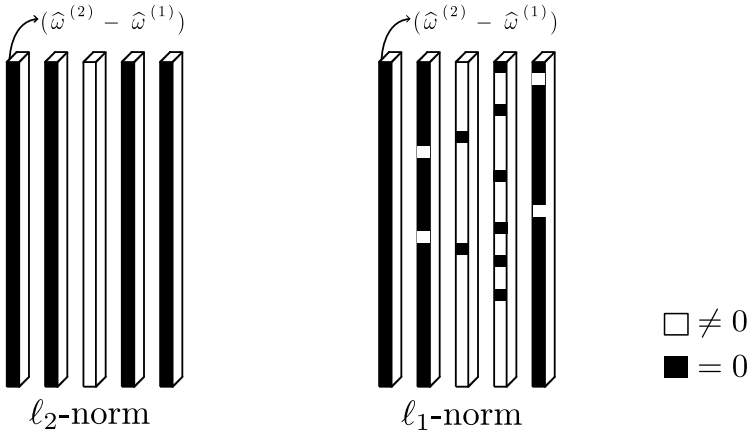


Figure 4.1: Comparison of the learned parameter vectors when using either ℓ_2 or ℓ_1-norm in the fused penalty. White squares indicates dimensions at which the two consecutive parameter vectors are different. Black squares where they are equal. The presence of at least one white square indicates a change-point.

Penalty term. Provided two hyperparameters, $\lambda_1, \lambda_2 > 0$, we propose the following penalty term for Eq. 4.4:

$$pen_{\lambda_1,\lambda_2}(\omega) = \lambda_1 \sum_{i=2}^{n} \left\| \omega^{(i)} - \omega^{(i-1)} \right\|_2 + \lambda_2 \sum_{i=1}^{n} \left\| \omega^{(i)} \right\|_1 .$$

The overall two-term penalty is necessary for recovering efficiently the piece-wise constant graph structure. The second term is quite standard: it allows the estimated parameter vectors to be sparse and thus imposes structure in the learned graphs. On the other hand, without the first term, we would fit for each timestamp $i \in \{1, \ldots, n\}$ a parameter vector $\omega^{(i)}$ that perfectly matches the observation $x^{(i)}$ (in terms of likelihood). In such a situation, we would obtain as many different parameter vectors ω as there is different samples, making the piece-wise constant assumption of Eq. 4.2 impossible to recover. This is why we propose a group-fused penalty, consisting in the ℓ_2-norm of the difference between two consecutive parameter vectors. The sum of the ℓ_2-norms acts as a group-lasso penalty on temporal difference between consecutive parameter vectors, which encourages the two vectors to be equal. This allows us to learn efficiently an evolving piece-wise constant structure and also to detect the change-points. In the past, the authors of [5] proposed to use a sum of ℓ_1-norm in order to impose structure in the variation of the weight matrices. However, such penalization does not allow to recover a piece-wise constantly evolving graph structure. This problem is illustrated in figure 4.1: by penalizing each dimension independently, the ℓ_1-norm easily leads to consecutive parameter vectors having different values, making the piece-wise constant assumption more difficult to recover.

In conclusion, the hyperparameter λ_1 controls the number of estimated change-points – the larger λ_1 is, the fewer the estimated number of change-points will be. Similarly, when λ_2 increases, the sparsity of each parameter vector increases as well. A priori, choosing the hyperparameters is not straightforward. However, since our objective function corresponds to a penalized logistic regression problem, we can use existing model selection criteria. We discuss further about this aspect in the experimental section.

3.2 Change-point detection and structure estimation

Assume that the optimization program (4.4) is solved. The set of estimated change-points $\widehat{\mathcal{D}}$ is:

$$\widehat{\mathcal{D}} = \left\{ \widehat{T}_j \in \{2, \dots, n\} : \left\| \widehat{\omega}_a^{(\widehat{T}_j)} - \widehat{\omega}_a^{(\widehat{T}_j - 1)} \right\|_2 \neq 0 \right\}.$$

Namely, this corresponds to the set of timestamps at which the estimated parameter vectors have changed. For each submodel $j = 1, \dots, |\widehat{\mathcal{D}}| + 1$, the a-th column of $\Theta^{(j)}$ is estimated by $\widehat{\theta}_a^{(j)} \triangleq \widehat{\omega}_a^{(\widehat{T}_{j-1})} = \dots = \widehat{\omega}_a^{(\widehat{T}_j - 1)}$. The non-zero elements of $\widehat{\theta}_a^{(j)}$ indicate the *neighborhood* of a.

One should notice that this estimation leads to a non-symmetric weight matrix. To overcome this problem, is was proposed in [95, 132] to either use the $\min$ or $\max$ operator. In the present work, to estimate the structure of the j-th graph, we take:

$$\widehat{\mathcal{E}}^{(j)} = \{(a, b) : \max(|\widehat{\theta}_{ab}^{(j)}|, |\widehat{\theta}_{ba}^{(j)}|) > 0\},$$

where $\widehat{\theta}_{ab}^{(j)}$ is the b-th element of $\widehat{\theta}_a^{(j)}$, and conversely for $\widehat{\theta}_{ba}^{(j)}$. In this case, there is an edge between two nodes if at least one of them contains the other node in its neighborhood.

4 Theoretical analysis

In this section, we present two change-point consistency theorems for TVI-FL. The theorems state that, as the number of samples n tends to infinity, the change-points are estimated more and more precisely.

4.1 Technical assumptions

We denote by $\widehat{D} = |\widehat{\mathcal{D}}|$ (the set's cardinality) the total number of detected change-points, respectively for the real changes $D = |\mathcal{D}|$, and by $[D]$ the set of indices $\{1, \dots, D\}$. Let us now define two important quantities. The first is the minimal time difference between two change-points:

$$\Delta_{\min} \triangleq \min_{j \in [D]} |T_j - T_{j-1}|.$$

The second quantity is the minimal variation in the model parameters between two change-points, which is given by:

$$\xi_{\min} \triangleq \min_{a \in V, j \in [D]} \|\theta_a^{(j+1)} - \theta_a^{(j)}\|_2.$$

We now introduce three standard assumptions on the Ising model inference and change-points detection. They are assumed to be true for each node $a \in V$.

(A1) *There exist two constants* $\phi_{\min} > 0$ *and* $\phi_{\max} < \infty$ *such that* $\forall j \in [D+1]$, $\phi_{\min} \leq \Lambda_{\min}\left(\mathbb{E}_{\Theta^{(j)}}[X_{\setminus a} X_{\setminus a}^\top]\right)$ *and* $\phi_{\max} \geq \Lambda_{\max}\left(\mathbb{E}_{\Theta^{(j)}}[X_{\setminus a} X_{\setminus a}^\top]\right)$.

Here $\Lambda_{\min}(\cdot)$ and $\Lambda_{\max}(\cdot)$ denote, respectively, the smallest and largest eigenvalues of the input matrix. (A1) is standard in such problems: it ensures that the covariates are not too dependent, making the model identifiable [95, 132]. In fact, this assumption is always verified if the support of the model is sufficiently large. Indeed, if at least p linearly independent

vectors have a non-zero probability to be observed, then the matrix $\mathbb{E}_{\Theta^{(j)}}[X_{\setminus a}X_{\setminus a}^\top]$ will have full rank.

(A2) *There exists a constant M such that* $\max_{j\in[D+1]}\|\theta_a^{(j)}\|_2 \leq M$.

(A3) *The sequence $\{T_j\}_{j=1}^D$ satisfies, for each j, $T_j = \lfloor n\tau_j \rfloor$, where $\lfloor x \rfloor$ is the largest integer smaller than or equal to x and $\{\tau_j\}_{j=1}^D$ is a fixed, unknown sequence of the change-point fractions belonging to* $[0, 1]$.

This last assumption says that as n grows, the new observations are sampled uniformly across all the $D + 1$ sub-models.

4.2 Main results

Next, we present our theoretical results on *change-point consistency*. The proofs are made for one node, a, but generalize to all the other nodes. We first provide the optimality conditions necessary to demonstrate the main results.

Lemma 4.1. *(Optimality Conditions) A matrix $\widehat{\omega}$ is optimal for our problem iff there exists a collection of subgradient vectors $\{\hat{z}^{(i)}\}_{i=2}^n$ and $\{\hat{y}^{(i)}\}_{i=1}^n$, with $\hat{z}^{(i)} \in \partial \left\|\widehat{\omega}^{(i)} - \widehat{\omega}^{(i-1)}\right\|_2$ and $\hat{y}^{(i)} \in \partial \left\|\widehat{\omega}^{(i)}\right\|_1$, such that $\forall k = 1, \ldots, n$ we have:*

$$\sum_{i=k}^{n} x_{\setminus a}^{(i)} \left\{ \tanh\left(\widehat{\omega}^{(i)\top} x_{\setminus a}^{(i)}\right) - \tanh\left(\omega_a^{(i)\top} x_{\setminus a}^{(i)}\right) \right\} - \sum_{i=k}^{n} x_{\setminus a}^{(i)} \left\{ x_a^{(i)} - \mathbb{E}_{\Omega^{(i)}}\left[X_a | X_{\setminus a} = x_{\setminus a}^{(i)}\right] \right\} + \lambda_1 \hat{z}^{(k)} + \lambda_2 \sum_{i=k}^{n} \hat{y}^{(i)} = \mathbf{0}_{p-1}, \quad (4.7)$$

where $\tanh$ *is the hyperbolic tangent function, $\mathbf{0}_{p-1}$ is the zero vector of size $p-1$, $\hat{z}^{(1)} = \mathbf{0}_{p-1}$, and*

$$\hat{z}^{(i)} = \begin{cases} \frac{\widehat{\omega}^{(i)} - \widehat{\omega}^{(i-1)}}{\left\|\widehat{\omega}^{(i)} - \widehat{\omega}^{(i-1)}\right\|_2} & \text{if } \widehat{\omega}^{(i)} - \widehat{\omega}^{(i-1)} \neq 0, \\ \in \mathcal{B}_2(0,1) & \text{otherwise,} \end{cases}$$

$$\hat{y}^{(i)} = \begin{cases} \text{sign}(\widehat{\omega}^{(i)}) & \text{if } x \neq 0, \\ \in \mathcal{B}_1(0,1) & \text{otherwise.} \end{cases}$$

Proof. The proof is given in the Appendix. It consists in writing the sub-differential of the objective function and say, thanks to the convexity, that 0 belongs to it. □

Theorem 4.1. *(Change-point consistency) Let $\{x_i\}_{i=1}^n$ be a sequence of observations drawn from the model presented in Sec. 2. Suppose (A1-A3) hold, and assume that $\lambda_1 \asymp \lambda_2 = \mathcal{O}(\sqrt{\log(n)/n})$. Let $\{\delta_n\}_{n\geq 1}$ be a non-increasing sequence that converges to 0, and such that $\forall n > 0$, $\Delta_{\min} \geq n\delta_n$, with $n\delta_n \to +\infty$. Assume further that (i) $\frac{\lambda_1}{n\delta_n \xi_{\min}} \to 0$, (ii) $\frac{\sqrt{p-1}\lambda_2}{\xi_{\min}} \to 0$, and (iii) $\frac{\sqrt{p\log(n)}}{\xi_{\min}\sqrt{n\delta_n}} \to 0$. Then, if the correct number of change-points are estimated, we have $\widehat{D} = D$ and:*

$$\mathbb{P}(\max_{j=1,\ldots,D} |\hat{T}_j - T_j| \leq n\delta_n) \underset{n\to\infty}{\longrightarrow} 1. \quad (4.8)$$

Proof. We extend the proof given in [75, 95] to the particular case of the Ising model. While the major steps are essentially the same, the Lemmas needed have been adapted to our case. We give here the proof's main steps.

Thanks to the union bound, the probability of the complementary in Eq. (4.8) can be upper bounded by:

$$\mathbb{P}(\max_{j=1,\dots,D} |\hat{T}_j - T_j| > n\delta_n) \leq \sum_{j=1}^{D} \mathbb{P}(|\hat{T}_j - T_j| > n\delta_n).$$

To prove Eq. (4.8), it is now sufficient to show $\forall j = 1, \dots, D$ that $\mathbb{P}(|\hat{T}_j - T_j| > n\delta_n) \to 0$. Let us define the event $C_n = \{|\hat{T}_j - T_j| < \frac{\Delta_{\min}}{2}\}$ and its complementary C_n^c. The rest of the proof is divided in two parts: bounding the good scenario, i.e. show that $\mathbb{P}(\{|\hat{T}_j - T_j| > n\delta_n\} \cap C_n) \to 0$, and doing the same for the bad scenario, i.e $\mathbb{P}(\{|\hat{T}_j - T_j| > n\delta_n\} \cap C_n^c) \to 0$.

To bound the good scenario, the proof applies Lemma 4.1 to bound the considered probability by three others probabilities. These latter are then asymptotically bounded by 0, thanks to a combination of Assumptions (A1-A3), assumptions of the theorem and concentration inequalities related to the considered time-varying Ising model (given by the Lemmas of the Appendix).

To bound the bad case scenario, the three following complementary events are defined:

$$\begin{aligned}
D_n^{(l)} &\triangleq \left\{\exists j \in [D], \widehat{T}_j \leq T_{j-1}\right\} \cap C_n^c, \\
D_n^{(m)} &\triangleq \left\{\forall j \in [D], T_{j-1} < \widehat{T}_j < T_{j+1}\right\} \cap C_n^c, \\
D_n^{(r)} &\triangleq \left\{\exists j \in [D], \widehat{T}_j \geq T_{j+1}\right\} \cap C_n^c.
\end{aligned}$$

Thus, it suffices to prove that $\mathbb{P}(\{|\hat{T}_j - T_j| > n\delta_n\} \cap D_n^{(l)})$, $\mathbb{P}(\{|\hat{T}_j - T_j| > n\delta_n\} \cap D_n^{(m)})$, and $\mathbb{P}(\{|\hat{T}_j - T_j| > n\delta_n\} \cap D_n^{(r)}) \to 0$ as $n \to \infty$. To prove this, similar arguments to those used for the good case are employed. □

Note that with $\delta_n = \log(n)^\gamma / n$, for any $\gamma > 1$ and $\xi_{\min} = \Omega(\sqrt{\log(n)/\log(n)^\gamma})$, the conditions of the theorem are met. With this parameterization, we obtain a convergence rate of order $\mathcal{O}(\log(n)^\gamma / n)$ for the estimation of the change-points. More precisely, for any $\delta > 0$ and sufficiently large n, we have with probability at least $1 - \delta$ that

$$\frac{1}{n} \max_{j=1,\dots,D} |\hat{T}_j - T_j| \leq \frac{1}{n} \log(n)^\gamma.$$

In conclusion, we obtain the same rate of convergence to that of the single change-point detection method given in [138]. It is almost optimal up to a logarithmic factor. The main drawback of the previous theorem is that it assumes that the number of change-points have been correctly estimated. In practice this is complicated to verify, while proving that the right number of change-points are consistently estimated is also difficult to get for this type of methods [75]. Nevertheless, in practice we may have an idea about an upper bound on the true number of change-points.

The next proposition provides a consistency result when the number of change-points is overestimated. Let us first introduce the metric $d(A\|B)$ defined as:

$$d(A\|B) = \sup_{b \in B} \inf_{a \in A} |b - a|. \tag{4.9}$$

Proposition 4.1. *Let $\{x_i\}_{i=1}^n$ be a sequence of observations drawn from the model presented in Sec. 2. Assume the conditions of Theorem 4.1 are respected. Then, if for a fix $D_{\max} < \infty$, we have $D \leq \widehat{D} \leq D_{\max}$ then:*

$$\mathbb{P}(d(\widehat{\mathcal{D}} \| \mathcal{D}) \leq n\delta_n) \underset{n\to\infty}{\longrightarrow} 1.$$

Proof. A detailed proof is provided in Appendix. The proof applies multiple times the different tricks used to prove Theorem 4.1 and the Lemmas also given in the Appendix. □

Proposition 1 is of fundamental importance as it tells us that, even though the number of change-points has been overestimated, asymptotically, all the true change-points belong to the set of estimated change-points.

5 Experimental study

This section provides numerical arguments showing the empirical performance of TVI-FL. All the experiments were implemented using Python and conducted on a personal laptop. The code of TVI-FL is available online[1], so as a Jupyter Notebook reproducing results and figures of the real-world example.

5.1 Optimization procedure

Despite being non-differentiable, the convexity of the objective function allows the use of existing convex optimization algorithms of the literature. In this work, we use the python package CVXPY [48] that allows us to solve our problem efficiently. Note also that the optimization for each node is independent to the other nodes, and hence the approach allows efficient parallel implementations.

In the situation where more than one data vector is observed at each timestamp, one has simply to replace the node-wise negative log-likelihood in Eq. 4.5 with:

$$-\sum_{i=1}^{n}\sum_{l=1}^{n^{(i)}} \log\left(\mathbb{P}_{\omega^{(i)}}(x_a^{(il)}|x_{\setminus a}^{(il)})\right), \tag{4.10}$$

where $n^{(i)}$ stands for the number of data vectors observed at timestamp i, and $x^{(il)}$ for the l-th observed vector at time i.

Tuning the hyperparameters λ_1 and λ_2. As stated in Sec. 3.1, it is possible to employ any model selection technique suited for logistic regression. In the experiments, we use and compare two techniques. The first is the *Akaike Information Criterion* (AIC) that computes the average of the following quantity for all nodes:

$$\mathrm{AIC}(\widehat{\omega}_a) \triangleq 2\mathcal{L}_a(\widehat{\omega}_a) + 2\,\mathrm{Dim}(\widehat{\omega}_a)\ , \tag{4.11}$$

where

$$\mathrm{Dim}(\widehat{\omega}_a) = \sum_{i=1}^{n}\left(\mathbb{1}\{\widehat{\omega}_a^{(i)} \neq \widehat{\omega}_a^{(i-1)}\}\sum_{b\in V\setminus a}\mathbb{1}\{\mathrm{sign}(\widehat{\omega}_{ab}^{(i)}) \neq 0\}\right)$$

[1] https://github.com/BatisteLB/TVI-FL

counts the number of parameters that are estimated. By convention, $\widehat{\omega}_a^{(0)} = \widehat{\omega}_a^{(1)}$. In this case, set of hyperparameters that minimize the AIC are finally selected.

The second technique, based on cross-validation (CV), assumes that more than one sample is observed at each moment in time $i = 1, \ldots, n$. Thus, the time-series can be split in a part for the learning phase and another part for the testing phase. In our experiments, we selected the hyperparameters maximizing the AUC i.e. the area under the ROC-curve associated to the classification score (the probability to be equal to either 1 or -1).

For both model selection techniques, AIC and CV, the hyperparameters are found using either the standard random-search or grid-search strategies.

5.2 Experimental setup

Baseline method. As mentioned in Sec. 1, no existing work in the literature deals properly with the considered multiple change-point detection task. Several methods deal with varying Gaussian graphical models [95, 177], varying Ising models with smooth structural changes over time [96], or the detection of a single change-point in the varying Ising model [138]. The closest work we can compare with is the Tesla method [5, 96]. Its major difference to our approach is the use of the ℓ_1-norm instead of the ℓ_2-norm as a fused-penalty. This difference is very significant, theoretically and practically.

Indeed, *using an ℓ_1-norm fused-penalty does not encourage the recovery of a graphical model that evolves piece-wise constantly as a whole*, which makes it less adaptable to recover change-points. More specifically, such a term does not encourage two consecutive parameter vectors to be equal at every dimensions: the regularization only affects each dimension independently. Thus, despite the edge weights may evolve *independently* in a piece-wise constant fashion, those changes occur at arbitrary timestamps and does not aggregate to a globally piece-wise constant behavior. An illustration of this phenomena and a comparison with the ℓ_2-norm can be found in the Appendix. Nonetheless, the same way the standard linear regression can be used to recover sparse parameters, Tesla can still be used to recover change-points in practice. Hence, we choose this method as our baseline because, despite the lack of any theoretical guarantee, it can still be applicable, provided a sufficiently large sample size and appropriately tuned regularization.

Performance metrics. We use two suitable metrics to evaluate the quality of TVI-FL on the learned graphs and change-points. The first one, very standard in change-point detection tasks [157] and known as the *Hausdorff metric*, measures the longest temporal distance between a change-point in $\mathcal{D}$ and its prediction in $\widehat{\mathcal{D}}$:

$$h(\mathcal{D}, \widehat{\mathcal{D}}) \triangleq \frac{1}{n} \max \left\{ \max_{t \in \mathcal{D}} \min_{\hat{t} \in \widehat{\mathcal{D}}} |t - \hat{t}|, \max_{\hat{t} \in \widehat{\mathcal{D}}} \min_{t \in \mathcal{D}} |t - \hat{t}| \right\}.$$

The lower this metric is, the better is the estimation. The second one, the F_1*-score*, measures the goodness of the learned graphs structures (high value is better) by the quantity:

$$F_1 = \frac{2 \times precision \times recall}{precision + recall},$$

which combines the two following classic measures:

$$precision = \frac{1}{n}\sum_{i=1}^{n}\sum_{a<b}\frac{\mathbb{1}\{(a,b)\in\widehat{\mathcal{E}}_i \wedge (a,b)\in\mathcal{E}_i\}}{\mathbb{1}\{(a,b)\in\widehat{\mathcal{E}}_i\}},$$
$$recall = \frac{1}{n}\sum_{i=1}^{n}\sum_{a<b}\frac{\mathbb{1}\{(a,b)\in\widehat{\mathcal{E}}_i \wedge (a,b)\in\mathcal{E}_i\}}{\mathbb{1}\{(a,b)\in\mathcal{E}_i\}}.$$

5.3 Application to synthetic data

Simulation design. We compare the performance of our method TVI-FL against Tesla using several independent synthetic datasets. We first fix certain characteristics for all generated datasets: each of them has $n = 100$ timestamps, $|D| = 2$ change-points at the 51-st and 81-st timestamps, hence resulting in 3 submodels being valid respectively for 50, 30, and 20 timestamps. We consider the graph structure of each submodel to be an independent random d-regular graph of $p = 20$ nodes, where at each time the degree of the all nodes can be $d \in \{2, 3, 4\}$.

To generate a piece-wise constant Ising model:

- We first pick a degree value $d \in \{2, 3, 4\}$ and draw independently 3 random d-regular graphs, one for each submodel. Same as in [5], the edge weights are drawn from a uniform distribution taken over $[-1, -0.5] \cup [0.5, 1]$.
- For each submodel, we draw observations using Gibbs sampling with a burn-in period of 1000 samples. Moreover, we collect one observation every 20 samples (lag) to avoid dependencies between them. In fact, instead of a single observation, for each timestamp $i \in \{1, \ldots, n\}$ we generate multiple observations $n^{(i)}$ in $\{4, 6, 8\}$, which requires to use the likelihood of Eq. (4.10). Besides, to be able to perform CV, we also sample 5 more observations per timestamp and use them only in the testing phase.

With the above procedure we generate 10 different piece-wise constant Ising models for each degree d, which makes 30 models to learn in total. In addition, for each model, we generate 3 different sets of observations, one for each $n^{(i)} \in \{4, 6, 8\}$, that constitute the individual learning problems of our evaluation. This results in 90 experiments in total.

For each experiment, we use a random-search strategy to find the best pair of hyperparameters (λ_1, λ_2) in $[4, 15] \times [30, 40]$. This is done individually for the TVI-FL and Tesla methods. The selected hyperparameters are those minimizing the AIC or maximizing the AUC (see Sec. 5.1).

Results. The average value and standard deviation of the corresponding h-score and F_1-score over each group of 10 experiments are reported in Tab. 4.1. The results clearly show that TVI-FL outperforms Tesla, regardless which model selection criterion we consult. This was expected as Telsa is not designed to recover Ising models that are evolving piece-wise constantly (see Sec. 5.2). Furthermore, while in some cases Tesla finds a number of change-points closer to the true number, the associated h-scores are still higher than those of TVI-FL. Yet, Tesla is still not irrelevant to the task and in fact there are cases in which it reaches competitive performance scores to those of TVI-FL. Another finding is that AIC seems to favor a low number of estimated changes-points. It achieves better h-scores for this simulated process, while the AUC criterion seems to give priority to the recovery of the graph structure, illustrated by higher F_1-scores.

Degree	Observations per timestamps	Method	AIC h-score ↓	AIC F_1-score ↑	AIC $\widehat{D}$	AUC h-score ↓	AUC F_1-score ↑	AUC $\widehat{D}$
$d=2$	$n^{(i)}=4$	TVI-FL	**0.046 ± (0.024)**	**0.694 ± (0.103)**	7.400 ± (3.137)	0.221 ± (0.035)	**0.876 ± (0.030)**	26.100 ± (7.739)
		Tesla	0.106 ± (0.087)	0.649 ± (0.190)	12.700 ± (7.682)	**0.184 ± (0.051)**	0.841 ± (0.041)	25.100 ± (4.784)
	$n^{(i)}=6$	TVI-FL	**0.129 ± (0.058)**	**0.816 ± (0.073)**	9.700 ± (2.759)	**0.147 ± (0.071)**	**0.875 ± (0.027)**	15.300 ± (3.378)
		Tesla	0.178 ± (0.130)	0.748 ± (0.167)	12.900 ± (5.540)	0.164 ± (0.062)	0.841 ± (0.048)	19.000 ± (2.530)
	$n^{(i)}=8$	TVI-FL	**0.082 ± (0.081)**	0.833 ± (0.095)	7.400 ± (3.040)	**0.099 ± (0.073)**	**0.891 ± (0.024)**	11.000 ± (3.873)
		Tesla	0.124 ± (0.071)	**0.846 ± (0.047)**	13.600 ± (2.010)	0.178 ± (0.066)	0.853 ± (0.039)	14.700 ± (3.348)
$d=3$	$n^{(i)}=4$	TVI-FL	**0.080 ± (0.069)**	**0.563 ± (0.089)**	7.000 ± (2.683)	**0.204 ± (0.035)**	**0.734 ± (0.024)**	23.100 ± (6.715)
		Tesla	0.278 ± (0.319)	0.353 ± (0.072)	3.200 ± (2.891)	0.208 ± (0.029)	0.611 ± (0.041)	29.200 ± (3.187)
	$n^{(i)}=6$	TVI-FL	**0.055 ± (0.064)**	**0.617 ± (0.161)**	6.300 ± (3.494)	**0.130 ± (0.051)**	**0.743 ± (0.034)**	12.800 ± (2.821)
		Tesla	0.302 ± (0.241)	0.346 ± (0.060)	2.000 ± (1.183)	0.173 ± (0.044)	0.616 ± (0.041)	22.600 ± (2.245)
	$n^{(i)}=8$	TVI-FL	**0.091 ± (0.073)**	**0.714 ± (0.130)**	8.000 ± (2.530)	**0.127 ± (0.073)**	**0.764 ± (0.032)**	10.400 ± (2.154)
		Tesla	0.311 ± (0.231)	0.361 ± (0.098)	2.600 ± (2.615)	0.162 ± (0.052)	0.633 ± (0.045)	18.700 ± (3.716)
$d=4$	$n^{(i)}=4$	TVI-FL	**0.101 ± (0.082)**	**0.453 ± (0.111)**	6.500 ± (3.324)	**0.232 ± (0.026)**	**0.644 ± (0.041)**	29.400 ± (4.317)
		Tesla	0.444 ± (0.273)	0.347 ± (0.044)	2.875 ± (1.900)	0.234 ± (0.017)	0.518 ± (0.046)	34.625 ± (1.654)
	$n^{(i)}=6$	TVI-FL	**0.099 ± (0.064)**	**0.501 ± (0.130)**	5.667 ± (2.309)	**0.183 ± (0.044)**	**0.664 ± (0.041)**	16.778 ± (3.258)
		Tesla	0.258 ± (0.236)	0.355 ± (0.035)	2.500 ± (1.118)	0.215 ± (0.032)	0.503 ± (0.040)	26.000 ± (4.472)
	$n^{(i)}=8$	TVI-FL	**0.077 ± (0.076)**	**0.528 ± (0.158)**	5.556 ± (3.624)	**0.169 ± (0.064)**	**0.678 ± (0.049)**	12.444 ± (4.524)
		Tesla	0.251 ± (0.230)	0.357 ± (0.044)	2.625 ± (0.696)	0.219 ± (0.027)	0.518 ± (0.054)	24.000 ± (2.398)

Table 4.1: Results for the model with the lowest AIC, and that with the highest AUC. The average ± (std) of the metrics is reported.

We show that empirically it is possible to obtain both low h-score and high F_1-score via better hyperparameters tuning. Specifically, for each experiment and for each degree d, we select the model with the highest F_1-score when the associated h-score $\leq h_{\min}$, with $h_{\min} \in \{0, .01, .02, .03\}$. This allows, respectively at most 0 to 3 timestamps of offset error between an estimated and a real change-point. In the results of Figure 4.2 we observe that even with very low h-score, high F_1-score are reachable. Furthermore, the TVI-FL method always provides better F_1-score than Tesla, confirming once again its superior performance.

5.4 Finding change-points in the real world: a voting dataset

Dataset and setup. In this section we evaluate the empirical performance of the TVI-FL method in a real-world use case. In particular, we analyze the different votes of the Illinois House of Representatives during the period of the 114-th and 115-th US Congresses (2015-2019), which are available at `voteview.com` [109]. The Illinois House of Representatives has 18 seats (one per district), each one corresponds to a US Representative belonging to the Democratic or the Republican parties. A Representative may or may not get reelected at the end of a Congress, which affects if he/she will retain his/her seat in the new Congress. The specific dataset we used contains 1264 votes, each of them represented by a vector of size $p = 18$, where a dimension is equal to 1 if the respective Representative of that seat has voted *Yes*, and -1 if it has voted *No*. When no information is provided about the vote of a seat (e.g. due to an absence), we impute the majority vote of its party.

It is always difficult to interpret a large number of change-points. For this reason, we choose to use the AIC criterion, which was found in Sec. 5 to favor smaller number of change-points. As for model tuning, we use a grid-search strategy to find the best values for the hyperparameters.

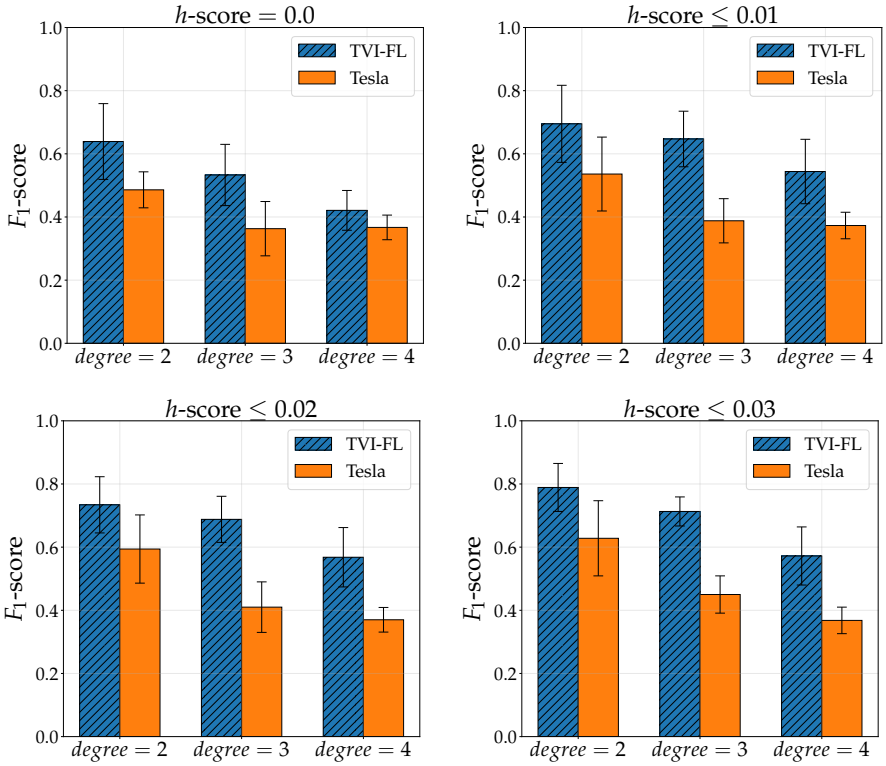


Figure 4.2: Average value of the best F_1-score obtained when the h-score is below a certain threshold. These thresholds are respectively (from top left to bottom right), 0.0, 0.01, 0.02, 0.03, i.e. at most 0, 1, 2, or 3 timestamps of offset error between an estimated and a real change-point. Each pairs of bars corresponds to different d-regular graphs, with $d \in \{2, 3, 4\}$. The error bars correspond to $\pm$ (std).

Results. Figure 4.3 (bottom) shows the cumulative function of the votes of each of the 18 seats, in temporal order, and the three change-points (dashed vertical line) detected by TVI-FL. The first two change-points are difficult to interpret; it seems though that the second one corresponds to the pre-election period when a Congress comes to its end and votes get usually less polarized. Nevertheless, it must be noted that the structural changes of the first two change-points are significantly lower compared to the third one. In fact, this last estimated change-point corresponds exactly to the time at which the Congress has changed. This significant change-point seems due to the non-reelection of some Representatives. More specifically, the Representative of 10-th seat was the only one who was not reelected at the end of the 114-th Congress: the Republican Robert Dold, who was replaced by the Democrat Brad Schneider. This switch apparently lead to a significant variation in the structure of the underlying graph. Figure 4.3 (top) shows the graphs of positive weights, before and after this significant change-point. As expected, two clusters appear, one with the seats of Democrats and the other with those of the Republicans. Moreover, the 10-th seat becomes more connected with the cluster of Democrats after the time of change: the node loses 3 connections to the Republican cluster and gains 5 connections to Democrats and gets connected with all of them. More generally, all its weights with the Republican cluster decreases, contrarily to its weights with the Democratic cluster that do increase. This observation explains the origins of the structural change. Finally, it is interesting

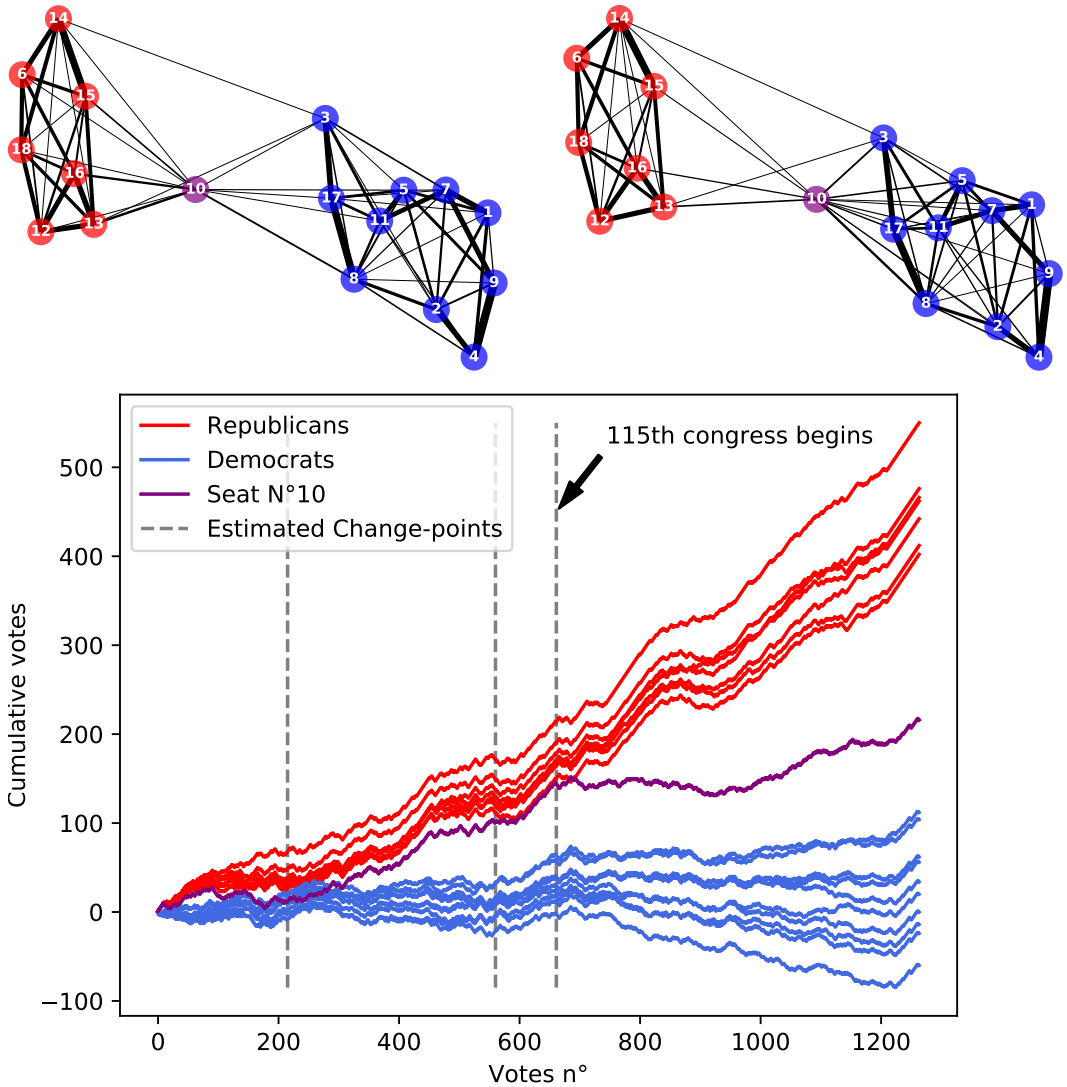


Figure 4.3: (Top) The two graphs before and after the strongest estimated change-point: the third one that indeed corresponds to the end of the 114-th Congress. (Bottom) The cumulative functions of votes of the 18 seats over the two Congresses.

to observe that before and after the change-point, the 10-th seat is the only one well-connected to both political groups. This makes us to conclude that this seat is represented by a *super-collaborator*, a role that some Representatives get by acting more independently and position themselves in the middle of the parties [10]. Similarly, it is not surprising for Dan Lipinski, who had the 3-rd seat, to present in the learned graphs 2 connections with Republicans, as he is known to be a conservative Democrat.

Overall, this experiment shows that TVI-FL is suited to find change-points in a real-world binary dataset, while also to recover the underlying evolving graph structure. This way, it increases the interpretability of the detected change-points. After applying the Telsa method on the same problem, we observed similar results and for this reason we omit them from the presentation.

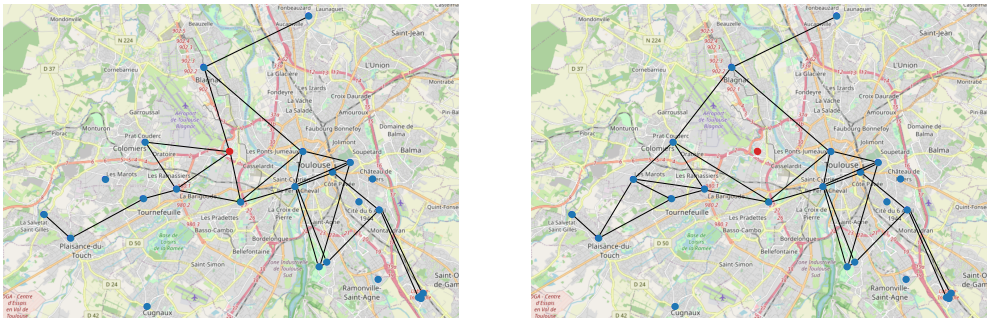


Figure 4.4: Learned graphs at two different timestamps. Only the edges with positive weights are represented and the abnormal BS is in red. (Left) A graph learned before the BS failure, recorded on the 30th day. (Right) A graph learned after this day.

5.5 Application to Sigfox dataset

We finally apply TVI-FL to the Sigfox dataset used in the previous chapters. Recall that in consists in a set of Sigfox messages recorded over a period of five months at the level of 34 Base Stations. In this experiment, we propose to directly apply our algorithm to the 4 last months, referred as the testing dataset in the experiment of chapter 2. Over this dataset, the Sigfox messages are recorded daily, resulting in $n = 120$ days/timestamps over which we select randomly $n_i = 200$ messages per day. Around the 30th day, one BS has been declared as working poorly. The goal of this experiment is therefore to see if we detect such timestamp as a change-point and if the learned graphs are well-estimated.

Here again, we selected the two hyperparameters using the AIC selection strategy. It resulted in the detection of 12 change-points for which most of them were difficult to interpret. However, only one change-point was detected in the neighborhood of the failing BS, actually detected at the 30th day, beginning time of failure. Moreover, the change in the neighborhood of this BS is characterized by a global decreasing in its associated weights. Such decreasing seems to be characteristic of an anomaly since it somehow means that the BS is sharing fewer messages with its neighbors than before. We illustrate such change in the network structure by representing a graph learned before and a graph learned after the 30th day (figure 4.4).

In the figure, only the positive weights are represented and we clearly see a loss of edges in the neighborhood of the red node. Moreover, we also see that the learned graphs are in accordance with the spatial distribution of the different BSs, an indicator of the goodness of fit of the estimated graphs.

6 Conclusions

The aim of this chapter was to answer the last two objectives raised in the introduction of this thesis. In other words, the detection of changes in the underlying structure of binary vectors. We proposed TVI-FL, an efficient way to learn a time-varying Ising model with piece-wise constantly evolving structure. Our method is able to both detect the changepoints at which the structure of the model changes and the structure themselves. Our work is the first to provide change-point consistency theorems in this context. Those theoretical guarantees are reinforced by an empirical study. Using two different model selection criteria,

the proposed method is showed to outperform the closest baseline algorithm. Moreover, although not specifically designed to work on Sigfox network data, we illustrated the applicability of TVI-FL to them in a promising last experiment.

Nevertheless, there are still few works to be addressed. These include for example the proof of consistent graph structure recovery (sparsistency) or the use of the recent Interaction Screening Objective [166] in place of the standard conditional likelihood.

7 Technical proofs

7.1 Main results

In the following, we recall and prove the main results given in the chapter. The proofs uses in many situations the different lemmas given next.

Lemma 4.1. *(Optimality Conditions) A matrix $\widehat{\omega}$ is optimal for our problem iff there exists a collection of subgradient vectors $\{\hat{z}^{(i)}\}_{i=2}^n$ and $\{\hat{y}^{(i)}\}_{i=1}^n$, with $\hat{z}^{(i)} \in \partial \left\|\widehat{\omega}^{(i)} - \widehat{\omega}^{(i-1)}\right\|_2$ and $\hat{y}^{(i)} \in \partial \left\|\widehat{\omega}^{(i)}\right\|_1$, such that $\forall k = 1, \ldots, n$ we have:*

$$\sum_{i=k}^{n} x_{\backslash a}^{(i)} \left\{ \tanh\left(\widehat{\omega}^{(i)\top} x_{\backslash a}^{(i)}\right) - \tanh\left(\omega_a^{(i)\top} x_{\backslash a}^{(i)}\right) \right\}$$
$$- \sum_{i=k}^{n} x_{\backslash a}^{(i)} \left\{ x_a^{(i)} - \mathbb{E}_{\Omega^{(i)}}\left[X_a | X_{\backslash a} = x_{\backslash a}^{(i)} \right] \right\} + \lambda_1 \hat{z}^{(k)} + \lambda_2 \sum_{i=k}^{n} \hat{y}^{(i)} = \mathbf{0}_{p-1},$$

where tanh *is the hyperbolic tangent function, $\mathbf{0}_{p-1}$ is the zero vector of size $p-1$, $\hat{z}^{(1)} = \mathbf{0}_{p-1}$, and*

$$\hat{z}^{(i)} = \begin{cases} \frac{\widehat{\omega}^{(i)} - \widehat{\omega}^{(i-1)}}{\left\|\widehat{\omega}^{(i)} - \widehat{\omega}^{(i-1)}\right\|_2} & \text{if } \widehat{\omega}^{(i)} - \widehat{\omega}^{(i-1)} \neq 0, \\ \in \mathcal{B}_2(0,1) & \text{otherwise,} \end{cases}$$

$$\hat{y}^{(i)} = \begin{cases} \text{sign}(\widehat{\omega}^{(i)}) & \text{if } x \neq 0, \\ \in \mathcal{B}_1(0,1) & \text{otherwise.} \end{cases}$$

Proof. The proof follows those of [75, 95] and [64]. We first introduce the following change of variables:

$$\gamma^{(i)} = \begin{cases} \omega^{(i)} & \text{if} \quad i = 1 \\ \omega^{(i)} - \omega^{(i-1)} & \text{otherwise.} \end{cases}$$

Thus $\omega^{(i)} = \sum_{l=1}^{i} \gamma^{(l)}$, which leads to a change in the objective function 4.4 presented in Section 3.1.

$$\{\hat{\gamma}^{(i)}\}_{i=1}^n = \underset{\gamma \in \mathbb{R}^{p-1 \times n}}{\arg\min} \sum_{i=1}^{n} \log \left\{ \exp\left(\sum_{l=1}^{i} \gamma^{(l)\top} x_{\backslash a}^{(i)} \right) + \exp\left(- \sum_{l=1}^{i} \gamma^{(l)\top} x_{\backslash a}^{(i)} \right) \right\}$$
$$- \sum_{i=1}^{n} x_a^{(i)} \sum_{l=1}^{i} \gamma^{(l)\top} x_{\backslash a}^{(i)} + \lambda_1 \sum_{i=2}^{n} \left\| \gamma^{(i)} \right\|_2 + \lambda_2 \sum_{i=1}^{n} \left\| \sum_{l=1}^{i} \gamma^{(l)} \right\|_1 . \tag{4.12}$$

This problem is convex, thus a necessary and sufficient condition for $\{\hat{\gamma}^{(i)}\}_{i=1}^n$ to be a solution is that for all $k = 1, \ldots, n$, the $(p-1)$-dimensional zero-vector $\mathbf{0}$, belongs to the

subdifferential of (4.12), taken with respect to $\gamma^{(k)}$:

$$\mathbf{0} \in \sum_{i=k}^{n} x_{\setminus a}^{(i)} \left(\tanh \left(\sum_{l=1}^{i} \hat{\gamma}^{(l)\top} x_{\setminus a}^{(i)} \right) - x_a^{(i)} \right) + \lambda_1 \partial \left\| \hat{\gamma}^{(k)} \right\|_2 + \lambda_2 \sum_{i=k}^{n} \partial \left\| \sum_{l=1}^{i} \hat{\gamma}^{(l)} \right\|_1 .$$

Recall that

$$\partial \|x\|_2 = \begin{cases} \left\{ \frac{x}{\|x\|_2} \right\} & \text{if} \quad x \neq 0 \\ \mathcal{B}_2(0,1) & \text{otherwise} \end{cases}$$
$$\partial \|x\|_1 = \begin{cases} \{\text{sign}(x)\} & \text{if} \quad x \neq 0 \\ \mathcal{B}_1(0,1) & \text{otherwise} \end{cases}.$$

Reapplying the change of variable, we obtain:

$$\mathbf{0} = \sum_{i=k}^{n} x_{\setminus a}^{(i)} \left(\tanh \left(\hat{\omega}^{(i)\top} x_{\setminus a}^{(i)} \right) - x_a^{(i)} \right) + \lambda_1 \hat{z}^{(k)} + \lambda_2 \sum_{i=k}^{n} \hat{y}^{(i)}$$

Noting that $\mathbb{E}_{\Omega^{(i)}} \left[X_a | X_{\setminus a} = x_{\setminus a}^{(i)} \right] = \tanh \left(\omega_a^{(i)\top} x_{\setminus a}^{(i)} \right)$, we obtain the final result. □

Theorem 4.1. *(Change-point consistency) Let $\{x_i\}_{i=1}^n$ be a sequence of observations drawn from the piece-wise constant Ising model presented in Sec. 2. Suppose (A1-A3) hold, and assume that $\lambda_1 \asymp \lambda_2 = \mathcal{O}(\sqrt{\log(n)/n})$. Let $\{\delta_n\}_{n\geq 1}$ be a non-increasing sequence that converges to 0, and such that $\forall n > 0$, $\Delta_{\min} \geq n\delta_n$, with $n\delta_n \to +\infty$. Assume further that (i) $\frac{\lambda_1}{n\delta_n \xi_{\min}} \to 0$, (ii) $\frac{\sqrt{p-1}\lambda_2}{\xi_{\min}} \to 0$, and (iii) $\frac{\sqrt{p \log(n)}}{\xi_{\min}\sqrt{n\delta_n}} \to 0$. Then, if the correct number of change-points are estimated, we have $\widehat{D} = D$ and:*

$$\mathbb{P}(\max_{j=1,\ldots,D} |\hat{T}_j - T_j| \leq n\delta_n) \underset{n\to\infty}{\longrightarrow} 1.$$

Proof. The proof follows the steps given in [64, 75, 95]. First of all,Thanks to the union bound,

$$\mathbb{P}(\max_{j=1,\ldots,D} |\hat{T}_j - T_j| > n\delta_n) \leq \sum_{j=1}^{D} \mathbb{P}(|\hat{T}_j - T_j| > n\delta_n),$$

thus it suffices to show for each $j = 1, \ldots, D$, that $\mathbb{P}(|\hat{T}_j - T_j| > n\delta_n) \to 0$. We denote by $A_{n,j}$ the event $\left\{ |\hat{T}_j - T_j| > n\delta_n \right\}$.

Similarly to [95], we first consider the good case where we assume that the event $C_n = \left\{ |\hat{T}_j - T_j| < \frac{\Delta_{\min}}{2} \right\}$ occurs.

Bounding the good case

For each $j = 1, \ldots, D$, we are going to show that $\mathbb{P}(A_{n,j} \cap C_n) \longrightarrow 0$. In particular, we suppose that $\hat{T}_j \leq T_j$ as the proof for $\hat{T}_j \geq T_j$ will be the same by symmetry.

Applying Lemma 4.1 with $k = \hat{T}_j$ and $k = T_j$, subtracting one with the other and applying the ℓ_2-norm, we obtain

$$
\begin{aligned}
0 = &\Bigg\| \sum_{i=\hat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \left\{ \tanh\left(\widehat{\omega}^{(i)\top} x_{\setminus a}^{(i)}\right) - \tanh\left(\omega_a^{(i)\top} x_{\setminus a}^{(i)}\right) \right\} \\
&- \sum_{i=\hat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \left\{ x_a^{(i)} - \mathbb{E}_{\Omega^{(i)}}\left[X_a | X_{\setminus a} = x_{\setminus a}^{(i)} \right] \right\} + \lambda_1 (\hat{z}^{(\widehat{T}_j)} - \hat{z}^{(T_j)}) + \lambda_2 \sum_{i=\hat{T}_j}^{T_j-1} \hat{y}^{(i)} \Bigg\|_2 \\
&\geq \Bigg\| \sum_{i=\hat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \left\{ \tanh\left(\widehat{\omega}^{(i)\top} x_{\setminus a}^{(i)}\right) - \tanh\left(\omega_a^{(i)\top} x_{\setminus a}^{(i)}\right) \right\} \\
&- \sum_{i=\hat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \left\{ x_a^{(i)} - \mathbb{E}_{\Omega^{(i)}}\left[X_a | X_{\setminus a} = x_{\setminus a}^{(i)} \right] \right\} \Bigg\|_2 - \left\| \lambda_2 \sum_{i=\hat{T}_j}^{T_j-1} \hat{y}^{(i)} \right\|_2 - \left\| \lambda_1 (\hat{z}^{(\widehat{T}_j)} - \hat{z}^{(T_j)}) \right\|_2
\end{aligned}
$$

We have $\left\| \lambda_1 (\hat{z}^{(\widehat{T}_j)} - \hat{z}^{(T_j)}) \right\|_2 \leq 2\lambda_1$ and $\left\| \lambda_2 \sum_{i=\hat{T}_j}^{T_j-1} \hat{y}^{(i)} \right\|_2 \leq (T_j - \hat{T}_j)\sqrt{p-1}\lambda_2$. Furthermore, one may notice that for all $i \in \left\{ \hat{T}_j, \ldots, T_j - 1 \right\}$, $\widehat{\omega}^{(i)} = \hat{\theta}_a^{j+1}$ and $\omega_a^{(i)} = \theta_a^j$. Adding and subtracting $\tanh\left((\theta_a^{j+1})^\top x_{\setminus a}^{(i)} \right)$, then applying again the triangle inequality leads to the following result:

$$
2\lambda_1 + (T_j - \hat{T}_j)\sqrt{p-1}\lambda_2 \geq \|R_1\|_2 - \|R_2\|_2 - \|R_3\|_2 \tag{4.13}
$$

with

$$
R_1 = \sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \left\{ \tanh\left((\theta_a^j)^\top x_{\setminus a}^{(i)} \right) - \tanh\left((\theta_a^{j+1})^\top x_{\setminus a}^{(i)} \right) \right\} \tag{4.14}
$$

$$
R_2 = \sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \left\{ \tanh\left((\hat{\theta}_a^{j+1})^\top x_{\setminus a}^{(i)} \right) - \tanh\left((\theta_a^{j+1})^\top x_{\setminus a}^{(i)} \right) \right\} \tag{4.15}
$$

$$
R_3 = \sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \left\{ x_a^{(i)} - \mathbb{E}_{\Theta^{(j)}}\left[X_a | X_{\setminus a} = x_{\setminus a}^{(i)} \right] \right\} \tag{4.16}
$$

The event (4.13) occurs with probability one and it can be showed that it is included in the event:

$$
\{2\lambda_1 + (T_j - \hat{T}_j)\sqrt{p-1}\lambda_2 \geq \frac{1}{3}\|R_1\|_2\} \cup \{\|R_2\|_2 \geq \frac{1}{3}\|R_1\|_2\} \cup \{\|R_3\|_2 \geq \frac{1}{3}\|R_1\|_2\}
$$

Thus, we have:

$$
\begin{aligned}
\mathbb{P}(A_{n,j} \cap C_n) \leq &\mathbb{P}(A_{n,j} \cap C_n \cap \{2\lambda_1 + (T_j - \hat{T}_j)\sqrt{p-1}\lambda_2 \geq \frac{1}{3}\|R_1\|_2\}) \\
&+ \mathbb{P}(A_{n,j} \cap C_n \cap \{\|R_2\|_2 \geq \frac{1}{3}\|R_1\|_2\})
\end{aligned}
$$

$$+ \mathbb{P}(A_{n,j} \cap C_n \cap \{\|R_3\|_2 \geq \frac{1}{3}\|R_1\|_2\})$$
$$\triangleq \mathbb{P}(A_{n,j,1}) + \mathbb{P}(A_{n,j,3}) + \mathbb{P}(A_{n,j,3})$$

Now, We are going to show that each one of the three events has a probability that converges to 0 as n grows. Let's focus on $A_{n,j,1}$. Applying the mean-value theorem, we have for all $i = \widehat{T}_j, \ldots, T_j - 1$:

$$\tanh\left((\theta_a^j)^\top x_{\setminus a}^{(i)}\right) - \tanh\left((\theta_a^{j+1})^\top x_{\setminus a}^{(i)}\right) = (1 - \tanh^2(\bar{\theta}^{iT} x_{\setminus a}^{(i)})) x_{\setminus a}^{(i)\top}(\theta_a^j - \theta_a^{j+1}) \tag{4.17}$$

with $\bar{\theta}^i = \alpha^i \theta_a^j + (1-\alpha^i)\theta_a^{j+1}$, for a certain $\alpha^i \in [0,1]$. Combining (4.17) with the definition of R_1, we obtain:

$$\|R_1\|_2 = \left\| \sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \left\{ \tanh\left((\theta_a^j)^\top x_{\setminus a}^{(i)}\right) - \tanh\left((\theta_a^{j+1})^\top x_{\setminus a}^{(i)}\right) \right\} \right\|_2 \tag{4.18}$$

$$= (T_j - \widehat{T}_j) \left\| \frac{1}{T_j - \widehat{T}_j} \sum_{i=\widehat{T}_j}^{T_j-1} (1 - \tanh^2(\bar{\theta}^{iT} x_{\setminus a}^{(i)})) \times x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top} (\theta_a^j - \theta_a^{j+1}) \right\|_2 \tag{4.19}$$

$$\geq (T_j - \widehat{T}_j) \times \Lambda_{\min}\left(\frac{1}{T_j - \widehat{T}_j} \sum_{i=\widehat{T}_j}^{T_j-1} (1 - \tanh^2(\bar{\theta}^{iT} x_{\setminus a}^{(i)})) x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top} \right) \times \left\|\theta_a^j - \theta_a^{j+1}\right\|_2 \tag{4.20}$$

Since, $\forall j$, $\left\|\theta_a^j\right\|_2 \leq M$ (A2), we have $\left\|\bar{\theta}^i\right\|_2 \leq M$ and $|\bar{\theta}^{iT} x_{\setminus a}^{(i)}| \leq M \cdot \sqrt{p-1}$. Thus, there exist a constant $\tilde{M} > 0$ such that $1 - \tanh^2(\bar{\theta}^{iT} x_{\setminus a}^{(i)}) \geq \tilde{M}$. Combining this with the fact that each matrix $x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top}$ are positive semidefinite, we have:

$$\|R_1\|_2 \geq (T_j - \widehat{T}_j)\tilde{M}\Lambda_{\min}\left(\frac{1}{T_j - \widehat{T}_j} \sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top} \right) \xi_{\min} \tag{4.21}$$

Thus, the event $\{2\lambda_1 + (T_j - \hat{T}_j)\sqrt{p-1}\lambda_2 \geq \frac{1}{3}\|R_1\|_2\}$ is included in the event

$$2\lambda_1 + (T_j - \hat{T}_j)\sqrt{p-1}\lambda_2 \geq (T_j - \widehat{T}_j)\tilde{M}\Lambda_{\min}\left(\frac{1}{T_j - \widehat{T}_j} \sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top} \right) \xi_{\min} \tag{4.22}$$

Denoting by $\{4.22\}$ the event of equation (4.22), we have:

$$\mathbb{P}(A_{n,j,1}) \leq \mathbb{P}(A_{n,j} \cap C_n \cap \{4.22\})$$
$$\leq \mathbb{P}\left(A_{n,j} \cap C_n \cap \{4.22\} \cap \{\Lambda_{\min}\left(\frac{1}{T_j - \widehat{T}_j} \sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top} \right) > \frac{\phi_{\min}}{2}\} \right)$$
$$+ \mathbb{P}\left(A_{n,j} \cap C_n \cap \{\Lambda_{\min}\left(\frac{1}{T_j - \widehat{T}_j} \sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top} \right) \leq \frac{\phi_{\min}}{2}\} \right)$$

Using Lemma 4.3 with $v_n = n\delta_n$ and $\epsilon = \frac{\phi_{\min}}{2}$, we can bound the right-hand side of the upper equation. We also re-write the first term so that we obtain:

$$\begin{aligned}\mathbb{P}(A_{n,j,1}) &\leq \mathbb{P}(A_{n,j} \cap C_n \cap \{\frac{2\lambda_1}{T_j - \widehat{T}_j} + \sqrt{p-1}\lambda_2 > \frac{\tilde{M}\phi_{\min}}{2}\xi_{\min}\}) \\ &\qquad + c_1 \exp\left(-\frac{\epsilon^2 n\delta_n}{2} + 2\log(n)\right) \\ &\leq \mathbb{P}(\xi_{\min}^{-1}\frac{2\lambda_1}{n\delta_n} + \xi_{\min}^{-1}\sqrt{p-1}\lambda_2 > \frac{\tilde{M}\phi_{\min}}{2}) + c_1 \exp\left(-\frac{\epsilon^2 n\delta_n}{2} + 2\log(n)\right)\end{aligned} \tag{4.23}$$

Thanks to (iii), we have $n\delta_n$ that goes to infinity faster than $\log(n)$, thus the second term of the sum goes to 0 as n grows. Furthermore, using (i) and (ii) we have:

$$\mathbb{P}(\xi_{\min}^{-1}\frac{2\lambda_1}{n\delta_n} + \xi_{\min}^{-1}\sqrt{p-1}\lambda_2 > \frac{\tilde{M}\phi_{\min}}{2}) \underset{n\to 0}{\longrightarrow} \mathbb{P}(0+0 > \frac{\tilde{M}\phi_{\min}}{2}) = 0$$

Which concludes that $\mathbb{P}(A_{n,j,1}) \to 0$.

We now focus on the event $A_{n,j,2}$. Let $\bar{T}_j \triangleq \lfloor 2^{-1}(T_j + T_{j+1})\rfloor$ and remark that between T_j and $\bar{T}_j$, $\widehat{\omega}^{(i)} = \widehat{\theta}^{j+1}$. Now, using Lemma 4.1 with $k = \bar{T}_j$ and $k = T_j$ and similar operation used to show equation (4.13), we have:

$$\begin{aligned}&2\lambda_1 + (\bar{T}_j - T_j)\sqrt{p-1}\lambda_2 \\ &\geq \left\| \sum_{i=T_j}^{\bar{T}_j - 1} x_{\setminus a}^{(i)} \left(\tanh\left((\hat{\theta}_a^{j+1})^\top x_{\setminus a}^{(i)}\right) - \tanh\left((\theta_a^{j+1})^\top x_{\setminus a}^{(i)}\right)\right)\right\|_2 \qquad (4.24) \\ &\quad - \left\| \sum_{i=T_j}^{\bar{T}_j - 1} x_{\setminus a}^{(i)} \left(x_a^{(i)} - \mathbb{E}_{\Theta^{(j+1)}}\left[X_a | X_{\setminus a} = x_{\setminus a}^{(i)}\right]\right)\right\|_2 \qquad (4.25)\end{aligned}$$

In the following we note $\varepsilon_{j+1}^i \triangleq x_a^{(i)} - \mathbb{E}_{\Theta^{(j+1)}}\left[X_a | X_{\setminus a} = x_{\setminus a}^{(i)}\right]$. Using the fact that $\left\|\hat{\theta}_a^{j+1}\right\|_2$ is necessarily bounded, Lemma 4.3 with $\epsilon = \phi_{\min}/2$ and similar arguments that we used for $A_{n,j,1}$, we can write that the first term in the right-hand side of the previous equation is lower-bounded by:

$$(T_j - \bar{T}_j)\tilde{\tilde{M}}\frac{\phi_{\min}}{2}\left\|\hat{\theta}_a^{j+1} - \theta_a^{j+1}\right\|_2$$

with probability tending to one. Here, $\tilde{\tilde{M}}$ corresponds to a positive constant derived the same way as $\tilde{M}$ in the previous part of the proof. In consequence, we can write

$$\left\|\hat{\theta}_a^{j+1} - \theta_a^{j+1}\right\|_2 \leq \frac{8\lambda_1 + 4(\bar{T}_j - T_j)\sqrt{p-1}\lambda_2 + 4\left\|\sum_{i=T_j}^{\bar{T}_j - 1} x_{\setminus a}^{(i)}\varepsilon_{j+1}^i\right\|_2}{\tilde{\tilde{M}}\phi_{\min}(T_{j+1} - T_j)} \tag{4.26}$$

which holds with probability tending to one.

Furthermore, with probability also tending to one it can be shown using the same arguments used to prove equation (4.21) that $\|R_1\|_2 \geq (T_j - \widehat{T}_j)\tilde{M}\phi_{\min}\xi_{\min}/2$ and $\|R_2\|_2 \leq \left\|\hat{\theta}_a^{j+1} - \theta_a^{j+1}\right\|_2 \phi_{\max}(T_j - \widehat{T}_j)/2$. Combining that with equation (4.31), we can write:

$$\mathbb{P}(A_{n,j,2}) \leq c_1 \exp\left(-c_2 n\delta_n + 2\log(n)\right) + \mathbb{P}\Bigg(A_{n,j} \cap C_n \cap$$
$$\left\{\frac{1}{3}\tilde{\tilde{M}}\tilde{M}\phi_{\min}^2\phi_{\max}^{-1}\xi_{\min}(T_{j+1} - T_j) \leq 8\lambda_1 + 4(\bar{T}_j - T_j)\sqrt{p-1}\lambda_2 + 4\left\|\sum_{i=T_j}^{\bar{T}_j - 1} x_{\setminus a}^{(i)}\varepsilon_{j+1}^i\right\|_2\right\}\Bigg)$$
$$\leq \mathbb{P}(c_3\phi_{\min}^2\phi_{\max}^{-1}\xi_{\min}\Delta_{\min} \leq \lambda_1) + \mathbb{P}(c_4\phi_{\min}^2\phi_{\max}^{-1}\xi_{\min} \leq \sqrt{p-1}\lambda_2)$$
$$+ \mathbb{P}\left(c_5\phi_{\min}^2\phi_{\max}^{-1}\xi_{\min} \leq (\bar{T}_j - T_j)^{-1}\left\|\sum_{i=T_j}^{\bar{T}_j - 1} x_{\setminus a}^{(i)}\varepsilon_{j+1}^i\right\|_2\right) + c_1\exp\left(-c_2 n\delta_n + 2\log(n)\right)$$

With $c_1, \ldots, c_5$ positive constants.

The first two terms tends to 0 as n goes to infinity thanks to the hypothesis (i) and (ii) of the theorem. Indeed, since $\Delta_{\min} > n\delta_n$ and $(n\delta_n\xi_{\min})^{-1}\lambda_1 \to 0$ (i), the first term tends to $\mathbb{P}(c_3\phi_{\min}^2\phi_{\max}^{-1} \leq 0) = 0$ and the second term tends to 0 since $\xi_{\min}^{-1}\sqrt{p-1}\lambda_2 \to 0$ (ii). The fourth term directly tends to 0. Applying Lemma 4.4, we can upper bound the third term by:

$$\mathbb{P}\left(c_5\phi_{\min}^2\phi_{\max}^{-1}\xi_{\min} \leq (\bar{T}_j - T_j)^{-1/2}2\sqrt{p\log(n)}\right) + c_6\exp(-2p\log(n))$$
$$\leq \mathbb{P}\left(c_5\phi_{\min}^2\phi_{\max}^{-1}\xi_{\min} \leq (n\delta_n)^{-1/2}2\sqrt{p\log(n)}\right) + c_6\exp(-2p\log(n))$$

with c_6 an other positive constant.

Since $(\xi_{\min}\sqrt{n\delta_n})^{-1}\sqrt{p\log(n)} \to 0$ (iii), the previous equation tends to 0, which make $\mathbb{P}(A_{n,j,2})$ tends to 0 as well.

Finally, we upper bound the probability on the event $A_{n,j,3}$. As before, we know that $\|R_1\|_2 \geq (T_j - \widehat{T}_j)\tilde{M}\phi_{\min}\xi_{\min}/2$ with probability at least $1 - c_1\exp(-c_2 n\delta_n + 2\log(n))$, thus we have:

$$\mathbb{P}(A_{n,j,3}) \leq \mathbb{P}\left(\frac{\tilde{M}\phi_{\min}\xi_{\min}}{6} \leq \frac{\|R_3\|_2}{T_j - \widehat{T}_j}\right) + c_1\exp(-c_2 n\delta_n + 2\log(n))$$

Using Lemma 4.5, we can upper bound the first term by:

$$\mathbb{P}\left(\frac{\tilde{M}\phi_{\min}\xi_{\min}}{6} \leq 2\sqrt{\frac{p\log(n)}{T_j - \widehat{T}_j}}\right) + c_2\exp(-c_3\log(n))$$
$$\leq \mathbb{P}\left(\frac{\tilde{M}\phi_{\min}\xi_{\min}}{6} \leq 2\sqrt{\frac{p\log(n)}{n\delta_n}}\right) + c_2\exp(-c_3\log(n))$$

which tends to 0 thanks to (iii). Since the symmetric case follows exactly the same arguments, we have shown that $\mathbb{P}(A_{n,j} \cap C_n) \to 0$. We now need to prove that $\mathbb{P}(A_{n,j} \cap C_n^c) \to 0$.

Bounding the bad case

Again, the proof follows the one of [64, 75, 95]. Let define the following complementary events:

$$D_n^{(l)} \triangleq \left\{\exists j \in [D], \widehat{T}_j \leq T_{j-1}\right\} \cap C_n^c \tag{4.27}$$

$$D_n^{(m)} \triangleq \left\{\forall j \in [D], T_{j-1} < \widehat{T}_j < T_{j+1}\right\} \cap C_n^c \tag{4.28}$$

$$D_n^{(r)} \triangleq \left\{\exists j \in [D], \widehat{T}_j \geq T_{j+1}\right\} \cap C_n^c. \tag{4.29}$$

We can write $\mathbb{P}(A_{n,j} \cap C_n^c) = \mathbb{P}(A_{n,j} \cap D_n^{(l)}) + \mathbb{P}(A_{n,j} \cap D_n^{(m)}) + \mathbb{P}(A_{n,j} \cap D_n^{(r)})$. Again, the goal is to prove that the three terms tends to 0. We will assume that $\widehat{T}_j \leq T_j$ as the other case can be done by symmetry. Let's first focus on the middle term, it has been shown in [64, 75, 95] that it can be upper bounded in the following way:

$$\begin{aligned}
&\mathbb{P}(A_{n,j} \cap D_n^{(m)}) \\
&\leq \mathbb{P}(A_{n,j} \cap \{(\widehat{T}_{j+1} - T_j) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)}) + \mathbb{P}(\{(T_{j+1} - \widehat{T}_{j+1}) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)}) \\
&\leq \mathbb{P}(A_{n,j} \cap \{(\widehat{T}_{j+1} - T_j) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)}) \\
&\quad + \sum_{k=j+1}^{D} \mathbb{P}(\{(\widehat{T}_{k+1} - T_k) \geq \frac{\Delta_{\min}}{2}\} \cap \{(T_k - \widehat{T}_k) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)})
\end{aligned} \tag{4.30}$$

Let's bound the first term. Assuming the event $A_{n,j} \cap \{(\widehat{T}_{j+1} - T_j) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)}$ and applying Lemma 4.1 with $k = \widehat{T}_j$ and $k = T_j$, we can prove similarly as equation (4.31) that:

$$\begin{aligned}
\left\|\hat{\theta}_a^{j+1} - \theta_a^j\right\|_2 &\leq \frac{4\lambda_1 + 2(T_j - \widehat{T}_j)\sqrt{p-1}\lambda_2 + 2\left\|\sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \varepsilon_j^i\right\|_2}{\tilde{\tilde{M}}\phi_{\min}(T_j - \widehat{T}_j)} \\
&\leq c_1\phi_{\min}^{-1}(n\delta_n)^{-1}\lambda_1 + c_2\phi_{\min}^{-1}\sqrt{p-1}\lambda_2 + c_3\phi_{\min}^{-1}(T_j - \widehat{T}_j)^{-1}\left\|\sum_{i=\widehat{T}_j}^{T_j-1} x_{\setminus a}^{(i)} \varepsilon_j^i\right\|_2
\end{aligned}$$

with probability tending to one. Using Lemma 4.5 we can bound the third term and obtain:

$$\left\|\hat{\theta}_a^{j+1} - \theta_a^j\right\|_2 \leq c_1\phi_{\min}^{-1}(n\delta_n)^{-1}\lambda_1 + c_2\phi_{\min}^{-1}\sqrt{p-1}\lambda_2 + c_3\phi_{\min}^{-1}(\sqrt{n\delta_n})^{-1}\sqrt{p\log(n)}$$

with probability tending to one. Similarly, applying the same lemmas with $k = T_j$ and either $k = \widehat{T}_{j+1}$, if $\widehat{T}_{j+1} \leq T_{j+1}$ or $k = T_{j+1}$ otherwise, we have:

$$\left\|\hat{\theta}_a^{j+1} - \theta_a^{j+1}\right\|_2 \leq c_4\phi_{\min}^{-1}(n\delta_n)^{-1}\lambda_1 + c_5\phi_{\min}^{-1}\sqrt{p-1}\lambda_2 + c_6\phi_{\min}^{-1}(\sqrt{n\delta_n})^{-1}\sqrt{p\log(n)}$$

with probability tending to one.
Since $\xi_{\min} \leq \left\|\theta_a^j - \theta_a^{j+1}\right\|_2 \leq \left\|\hat{\theta}_a^{j+1} - \theta_a^j\right\|_2 + \left\|\hat{\theta}_a^{j+1} - \theta_a^{j+1}\right\|_2$, we finally upper bound the considered probability by:

$$\begin{aligned}&\mathbb{P}(A_{n,j} \cap \{(\widehat{T}_{j+1} - T_j) \geq \frac{\Delta_{\min}}{2}\} \cap D_n^{(m)}) \\ &\quad \leq \mathbb{P}(\xi_{\min} \leq c_7\phi_{\min}^{-1}(n\delta_n)^{-1}\lambda_1 + c_8\phi_{\min}^{-1}\sqrt{p-1}\lambda_2 + c_9\phi_{\min}^{-1}(\sqrt{n\delta_n})^{-1}\sqrt{p\log(n)})\end{aligned}$$

Which tends to 0 thanks to the hypothesis (i), (ii) and (iii). The other probabilities in the upper bound on $\mathbb{P}(A_{n,j} \cap D_n^{(m)})$ also tends to 0. The proof follows exactly the previous one. We proved that $\mathbb{P}(A_{n,j} \cap D_n^{(m)}) \to 0$, we will now show the same for $\mathbb{P}(A_{n,j} \cap D_n^{(l)})$. The proof exactly follows the one of [64] where it has been showed that:

$$\begin{aligned}\mathbb{P}(D_n^{(l)}) &\leq \sum_{j=1}^{D} 2^{j-1}\mathbb{P}(\max\{l \in [D] : \widehat{T}_l \leq T_{l-1}\}) \\ &\leq 2^{D-1}\sum_{j=1}^{D}\sum_{l>j}\mathbb{P}(\{T_l - \widehat{T}_l \geq \frac{\Delta_{\min}}{2}\} \cap \{\widehat{T}_{l+1} - T_l \geq \frac{\Delta_{\min}}{2}\})\end{aligned}$$

Now, as shown in [64] and with the same arguments used to bound the elements of (4.30), we have $\mathbb{P}(D_n^{(l)}) \to 0$. Similarly we can show $\mathbb{P}(D_n^{(r)}) \to 0$ as $n \to 0$. Finally we have $\mathbb{P}(A_{n,j} \cap C_n^c) \to 0$, which concludes the proof. □

Proposition 4.1. *Let $\{x_i\}_{i=1}^n$ be a sequence of observation drawn from the model presented in Sec. 2. Assume the condition of Theorem 1 are respected. Then, if for a fix $D_{\max}$ we have $D \leq \widehat{D} \leq D_{\max}$ then:*

$$\mathbb{P}(d(\widehat{\mathcal{D}}\|\mathcal{D}) \leq n\delta_n) \underset{n\to\infty}{\longrightarrow} 1.$$

Proof. As stated upper, the proof, here again, follows the one of [75]. We are going to show that:

$$\begin{aligned}&\mathbb{P}(\{d(\widehat{\mathcal{D}}\|\mathcal{D}) \geq n\delta_n\} \cap \{D \leq \widehat{D} \leq D_{\max}\}) \\ &\quad \leq \sum_{K=D}^{D_{\max}}\mathbb{P}(\{d(\widehat{\mathcal{D}}\|\mathcal{D}) \geq n\delta_n\} \cap \{\widehat{D} = K\}) \underset{n\to\infty}{\longrightarrow} 0.\end{aligned}$$

First, we note that for $K = D$, we have
$\mathbb{P}(\{d(\widehat{\mathcal{D}}\|\mathcal{D}) \geq n\delta_n\} \cap \{\widehat{D} = K\}) \underset{n\to\infty}{\longrightarrow} 0$ thanks to Theorem 1. Thus it suffices to show that:

$$\sum_{K=D+1}^{D_{\max}} \mathbb{P}(\{d(\widehat{\mathcal{D}}\|\mathcal{D}) \geq n\delta_n\} \cap \{\widehat{D} = K\})$$
$$\leq \sum_{K=D+1}^{D_{\max}} \sum_{k=1}^{D} \mathbb{P}(\forall 1 \leq l \leq K, |\widehat{T}_l - T_k| \geq n\delta_n) \underset{n\to\infty}{\longrightarrow} 0.$$

Like in [75], we rewrite the event $\{\forall 1 \leq l \leq K, |\widehat{T}_l - T_k| \geq n\delta_n\}$ as the disjoint union of the events:

$$E_{n,k,1} = \{\forall 1 \leq l \leq K, |\widehat{T}_l - T_k| \geq n\delta_n \text{ and } \widehat{T}_l < T_k\}$$
$$E_{n,k,2} = \{\forall 1 \leq l \leq K, |\widehat{T}_l - T_k| \geq n\delta_n \text{ and } \widehat{T}_l > T_k\}$$
$$E_{n,k,3} = \{\exists 1 \leq l \leq K-1, |\widehat{T}_l - T_k| \geq n\delta_n, |\widehat{T}_{l+1} - T_k| \geq n\delta_n \text{ and } \widehat{T}_l < T_k < \widehat{T}_{l+1}\}$$

and propose to show that the probability of each events tends to 0 as n grows. Let's begin with $\mathbb{P}(E_{n,k,1})$ and note that it is equal to:

$$\mathbb{P}(E_{n,k,1} \cap \{\widehat{T}_K > T_{k-1}\}) + \mathbb{P}(E_{n,k,1} \cap \{\widehat{T}_K \leq T_{k-1}\})$$

First, we are going to upper bound the left-hand element of the previous equation. Applying Lemma 4.1 with $t = \widehat{T}_K$ and $t = T_k$, we can prove similarly to the equation (4.13) in the good case scenario of the previous theorem that:

$$2\lambda_1 + (T_k - \widehat{T}_K)\sqrt{p-1}\lambda_2 \geq \left\|R'_1\right\|_2 - \left\|R'_2\right\|_2 - \left\|R'_3\right\|_2$$

with

$$R'_1 = \sum_{i=\widehat{T}_K}^{T_k-1} x_{\setminus a}^{(i)} \left\{ \tanh\left((\theta_a^k)^T x_{\setminus a}^{(i)}\right) - \tanh\left((\theta_a^{k+1})^T x_{\setminus a}^{(i)}\right) \right\}$$
$$R'_2 = \sum_{i=\widehat{T}_K}^{T_k-1} x_{\setminus a}^{(i)} \left\{ \tanh\left((\hat{\theta}_a^{K+1})^T x_{\setminus a}^{(i)}\right) - \tanh\left((\theta_a^{k+1})^T x_{\setminus a}^{(i)}\right) \right\}$$
$$R'_3 = \sum_{i=\widehat{T}_K}^{T_k-1} x_{\setminus a}^{(i)} \left\{ x_a^{(i)} - \mathbb{E}_{\Theta^{(k)}}\left[X_a | X_{\setminus a} = x_{\setminus a}^{(i)}\right] \right\}.$$

Like in the previous theorem, we can upperbound $\mathbb{P}(E_{n,k,1} \cap \{\widehat{T}_k > T_{k-1}\})$ by:

$$\mathbb{P}(E_{n,k,1}^{(1)}) + \mathbb{P}(E_{n,k,1}^{(2)}) + \mathbb{P}(E_{n,k,1}^{(3)})$$

where

$$E_{n,k,1}^{(1)} = \{2\lambda_1 + (T_k - \widehat{T}_K)\sqrt{p-1}\lambda_2 \geq \frac{1}{3}\left\|R'_1\right\|_2\}$$

$$E^{(2)}_{n,k,1} = \{\left\|R'_2\right\|_2 \geq \frac{1}{3}\left\|R'_1\right\|_2\}$$
$$E^{(3)}_{n,k,1} = \{\left\|R'_3\right\|_2 \geq \frac{1}{3}\left\|R'_1\right\|_2\}$$

To show that $\mathbb{P}(E^{(1)}_{n,k,1})$ tends to 0 it suffices to follow the proof used to show that $\mathbb{P}(A_{n,j,1})$ tends to 0 in the good scenario of the previous theorem.

Similarly, to show that $\mathbb{P}(E^{(2)}_{n,k,1})$ tends to 0 it suffices to follow the proof used for $\mathbb{P}(A_{n,j,2})$. Applying lemma 4.1 with $t = T_k$ ans $t = T_{k+1}$ we can show that with probability tending to one:

$$\left\|\widehat{\theta}^{K+1}_a - \theta^{k+1}_a\right\|_2 \leq \frac{4\lambda_1 + 2(T_{k+1} - T_k)\sqrt{p-1}\lambda_2 + 2\left\|\sum_{i=T_k}^{T_{k+1}} x^{(i)}_{\setminus a}\varepsilon^i_{j+1}\right\|_2}{\tilde{\tilde{M}}\phi_{\min}(T_{k+1} - T_k)} \tag{4.31}$$

The rest follows exactly the arguments used to show the limit of $\mathbb{P}(A_{n,j,2})$.

Finally, $\mathbb{P}(E^{(3)}_{n,k,1})$ tends to 0 the same way $\mathbb{P}(A_{n,j,3})$ was tending to 0 in the previous proof.

The proof to show that $\mathbb{P}(E_{n,k,1} \cap \{\widehat{T}_K \leq T_{k-1}\})$ tends to 0 is the same. It suffices to apply lemma 4.1 with $t = T_{k-1}$ and $t = T_k$ to split the event in 3 sub-events and follow the proof. By symmetry, we also have $\mathbb{P}(E_{n,k,2}) \to 0$.

Let's now focus on $E_{n,k,3}$. Like in [75], the event is split is four independent events:

$$E_{n,k,3} = E^{(1)}_{n,k,3} \cup E^{(2)}_{n,k,3} \cup E^{(3)}_{n,k,3} \cup E^{(4)}_{n,k,3}$$

with

$$E^{(1)}_{n,k,3} = E_{n,k,3} \cap \{T_{k-1} < \widehat{T}_l < \widehat{T}_{l+1} < T_{k+1}\}$$
$$E^{(2)}_{n,k,3} = E_{n,k,3} \cap \{T_{k-1} < \widehat{T}_l < T_{k+1}, \widehat{T}_{l+1} > T_{k+1}\}$$
$$E^{(3)}_{n,k,3} = E_{n,k,3} \cap \{\widehat{T}_l < T_{k-1}, T_{k-1} < \widehat{T}_{l+1} < T_{k+1}\}$$
$$E^{(4)}_{n,k,3} = E_{n,k,3} \cap \{\widehat{T}_l < T_{k-1}, \widehat{T}_{l+1} > T_{k+1}\}$$

To prove that each one of the previous events have a probability that tends to 0 as n grows, we invite the reader to read the proof of [75]. It consist in multiple applications of the different Lemmas, the same way we used them in the previous part. Only the time at which lemma 4.1 is used changes and are given by [75]. This concludes the proof. □

Supplementary Lemmas

Below, the different lemmas necessary to prove the main results are given.

Lemma 4.2. *Let $\{x^{(i)}\}_{i=1}^n$ be a set of i.i.d observation sampled from an Ising model with parameter $\Theta \in \mathbb{R}^{p\times p}$ and assume that assumption (A1) is satisfied. Then, $\forall r, l \in [n]$ such*

that $l < r$ and $r - l > v_n$ with v_n a positive serie, we have $\forall \epsilon > 0$:

$$\mathbb{P}\left(\Lambda_{\min}\left(\frac{1}{r-l+1}\sum_{i=l}^{r} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top}\right) \leq \phi_{\min} - \epsilon\right) \leq 2(p-1)^2 \exp\left(-\frac{\epsilon^2 v_n}{2}\right) \tag{4.32}$$

and

$$\mathbb{P}\left(\Lambda_{\max}\left(\frac{1}{r-l+1}\sum_{i=l}^{r} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top}\right) \geq \phi_{\max} + \epsilon\right) \leq 2(p-1)^2 \exp\left(-\frac{\epsilon^2 v_n}{2}\right) \tag{4.33}$$

Proof. Let $\widehat{\Sigma} = \frac{1}{r-l+1}\sum_{i=l}^{r} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top}$ and $\Sigma = \mathbb{E}\left[X_{\setminus a} X_{\setminus a}^{\top}\right]$.

We first proove the inequality (4.32). Recall that for a symmetric matrix M, we have $\Lambda_{\max}(M) \leq \|M\|_F$, the Frobenius norm of M. We have

$$\begin{aligned}
\Lambda_{\min}(\widehat{\Sigma}) &= \min_{\|v\|_2=1} v^{\top}\widehat{\Sigma}v && (4.34)\\
&\geq \min_{\|v\|_2=1} v^{\top}\Sigma v - \max_{\|v\|_2=1} v^{\top}(\widehat{\Sigma}-\Sigma)v && (4.35)\\
&\geq \Lambda_{\min}(\Sigma) - \Lambda_{\max}(\widehat{\Sigma}-\Sigma) && (4.36)\\
&\geq \phi_{\min} - \left\|\widehat{\Sigma}-\Sigma\right\|_F && (4.37)
\end{aligned}$$

Let $s_{mq}^{(i)}$ be the (m,q)-th coordinate of $x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top} - \Sigma$ and $\frac{1}{r-l+1}\sum_{i=l}^{r} s_{mq}^{(i)}$ the one of $\widehat{\Sigma} - \Sigma$. Note that $\mathbb{E}\left[s_{mq}^{(i)}\right] = 0$ and $|s_{mq}^{(i)}| \leq 2$. Let's analyze the quantity $\mathbb{P}\left(\left\|\widehat{\Sigma}-\Sigma\right\|_F > \epsilon\right)$ with $\epsilon > 0$:

$$\begin{aligned}
\mathbb{P}\left(\left\|\widehat{\Sigma}-\Sigma\right\|_F > \epsilon\right) &= \mathbb{P}\left((\sum_{m,q} s_{mq}^2)^{1/2} > \epsilon\right) && (4.38)\\
&= \mathbb{P}\left(\sum_{m,q} s_{mq}^2 > \epsilon^2\right) && (4.39)\\
&\leq \sum_{m,q} \mathbb{P}\left(s_{mq}^2 > \epsilon^2\right) && (4.40)\\
&\leq \sum_{m,q} \mathbb{P}\left(|s_{mq}| > \epsilon\right) && (4.41)
\end{aligned}$$

Thanks to Hoeffding's inequality, we have $\mathbb{P}\left(|s_{mq}| > \epsilon\right) \leq 2\exp\left(-\frac{\epsilon^2(r-l+1)}{2}\right)$. Since $r - l > v_n$, we also have $\mathbb{P}\left(|s_{mq}| > \epsilon\right) \leq 2\exp\left(-\frac{\epsilon^2 v_n}{2}\right)$. It follows from (4.41) that $\mathbb{P}\left(\left\|\widehat{\Sigma}-\Sigma\right\|_F > \epsilon\right) \leq 2(p-1)^2\exp\left(-\frac{\epsilon^2 v_n}{2}\right)$. We deduce that:

$$\mathbb{P}\left(\Lambda_{\min}(\widehat{\Sigma}) \geq \phi_{\min} - \epsilon\right) \geq 1 - 2(p-1)^2\exp\left(-\frac{\epsilon^2 v_n}{2}\right), \tag{4.42}$$

which concludes the proof for (4.32).

To prove (4.33) it suffices to note that $\Lambda_{\max}(\widehat{\Sigma}) \leq \phi_{\max} + \left\|\widehat{\Sigma}-\Sigma\right\|_F$ and use the same arguments. □

Lemma 4.3. *Let $\{x^{(i)}\}_{i=1}^n$ be a set of i.i.d observation sampled from an Ising model with parameter $\Theta \in \mathbb{R}^{p\times p}$ and assume that assumption (A1) is satisfied.*
Let R and L be two random variable such that $R, L \in [n]$, $L < R$ and $R - L > v_n$ almost surely, with v_n a positive serie. For a fixed node a and any $\epsilon > 0$, there exist a constant $c_1 > 0$ such that:

$$\mathbb{P}\left(\Lambda_{\min}\left(\frac{1}{R-L+1}\sum_{i=L}^{R} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top}\right) \leq \phi_{\min} - \epsilon\right) \leq c_1 \exp\left(-\frac{\epsilon^2 v_n}{2} + 2\log(n)\right) \tag{4.43}$$

and

$$\mathbb{P}\left(\Lambda_{\max}\left(\frac{1}{R-L+1}\sum_{i=L}^{R} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top}\right) \geq \phi_{\max} + \epsilon\right) \leq c_1 \exp\left(-\frac{\epsilon^2 v_n}{2} + 2\log(n)\right) \tag{4.44}$$

Proof. We note $\widehat{\Sigma}(L,R) = \frac{1}{R-L+1}\sum_{i=L}^{R} x_{\setminus a}^{(i)} x_{\setminus a}^{(i)\top}$ and $\mathcal{I} \triangleq \left\{(l,r) \in [n]^2 : r - l > v_n\right\}$
We first prove the inequality (4.43):

$$\mathbb{P}\left(\Lambda_{\max}\left(\widehat{\Sigma}(L,R)\right) \geq \phi_{\max} + \epsilon\right) = \sum_{(l,r)\in\mathcal{I}} \mathbb{P}\left(\Lambda_{\max}\left(\widehat{\Sigma}(L,R)\right), L = l, R = r\right) \tag{4.45}$$

$$\leq \sum_{(l,r)\in\mathcal{I}} \mathbb{P}\left(\Lambda_{\max}\left(\widehat{\Sigma}(L,R)\right) \middle| L = l, R = r\right) \tag{4.46}$$

Using Lemma 4.2 we can bound (4.46):

$$(4.46) \leq \sum_{(l,r)\in\mathcal{I}} 2(p-1)^2 \exp\left(-\frac{\epsilon^2 v_n}{2}\right) \tag{4.47}$$

$$\leq |\mathcal{I}| c_1 \exp\left(-\frac{\epsilon^2 v_n}{2}\right) \tag{4.48}$$

$$\leq n^2 c_1 \exp\left(-\frac{\epsilon^2 v_n}{2}\right) \tag{4.49}$$

$$\leq c_1 \exp\left(-\frac{\epsilon^2 v_n}{2} + 2\log(n)\right) \tag{4.50}$$

with $c_1 = 2(p-1)$. This concludes the proof for (4.43). Same arguments are used to prove (4.44). □

Lemma 4.4. *Let $\{x^{(i)}\}_{i=1}^n$ be a set of independent observation sampled from the time-varying Ising model (Section 2). Then, $\forall j \in [D]$ and $\forall r, l \in \{T_j, \ldots, T_{j+1} - 1\}$ such that $l < r$, we have:*

$$\mathbb{P}\left(\frac{1}{r-l+1}\left\|R_3(l,r)\right\|_2 \leq 2\sqrt{\frac{p\log(n)}{r-l+1}}\right) \geq 1 - 2(p-1)\exp\left(-2p\log(n)\right) \tag{4.51}$$

with $R_3(l,r) = \sum_{i=l}^{r} x_{\setminus a}^{(i)}\left\{x_a^{(i)} - \mathbb{E}_{\Theta^j}\left[X_a | X_{\setminus a} = x_{\setminus a}^{(i)}\right]\right\}$

Proof. Let Z_{ij} be the the j-th element of the vector $\frac{1}{r-l+1}x_{\backslash a}^{(i)}\left\{x_a^{(i)} - \mathbb{E}_\Theta\left[X_a|X_{\backslash a} = x_{\backslash a}^{(i)}\right]\right\}$. Note that $|Z_{ij}| \leq \frac{2}{r-l+1}$ and $\mathbb{E}\left[Z_{ij}\right] = 0$. Let $\epsilon > 0$, we have:

$$\begin{aligned}
\mathbb{P}\left(\frac{1}{r-l+1}\left\|R_3(l,r)\right\|_2 \geq \epsilon\right) &= \mathbb{P}\left(\sqrt{\sum_{j\neq a}(\sum_{i=l}^{r} Z_{ij})^2} \geq \epsilon\right) \\
&= \mathbb{P}\left(\sum_{j\neq a}(\sum_{i=l}^{r} Z_{ij})^2 \geq \epsilon^2\right) \\
&\leq \sum_{j\neq a}\mathbb{P}\left(|\sum_{i=l}^{r} Z_{ij}| \geq \epsilon\right) \\
&\leq 2(p-1)\exp\left(-\frac{\epsilon^2(r-l+1)}{2}\right)
\end{aligned}$$

Now, if we fix $\epsilon = 2\sqrt{\frac{p\log(n)}{r-l+1}}$, we obtain:

$$\mathbb{P}\left(\frac{1}{r-l+1}\left\|R_3(l,r)\right\|_2 \leq 2\sqrt{\frac{p\log(n)}{r-l+1}}\right) \geq 1 - 2(p-1)\exp\left(-2p\log(n)\right)$$

□

Lemma 4.5. *Let $\{x^{(i)}\}_{i=1}^n$ be a set of independent observation sampled from the time-varying Ising model (Section 2). We have:*

$$\mathbb{P}\left(\bigcap_{j\in[D]}\bigcap_{l,r\in\mathcal{I}_j}\left\{\frac{1}{r-l+1}\left\|R_3^j(l,r)\right\|_2 \leq 2\sqrt{\frac{p\log(n)}{r-l+1}}\right\}\right) \geq 1 - c_2\exp\left(-c_3\log(n)\right) \tag{4.52}$$

with $R_3^j(l,r) = \sum_{i=l}^r x_{\backslash a}^{(i)}\left\{x_a^{(i)} - \mathbb{E}_{\Theta^j}\left[X_a|X_{\backslash a} = x_{\backslash a}^{(i)}\right]\right\}$, c_2, c_3 some positive constants and $\mathcal{I}_j \triangleq \left\{(l,r) \in \{T_j, \ldots, T_{j+1}-1\}^2 : r > l\right\}$.

Proof. The proof is a simple application of Lemma 4.4:

$$\begin{aligned}
&\mathbb{P}\left(\bigcup_{j\in[D]}\bigcup_{l,r\in\mathcal{I}_j}\left\{\frac{1}{r-l+1}\left\|R_3^j(l,r)\right\|_2 \geq 2\sqrt{\frac{p\log(n)}{r-l+1}}\right\}\right) \\
&\leq \sum_{j\in[D]}\sum_{l,r\in\mathcal{I}_j}\mathbb{P}\left(\frac{1}{r-l+1}\left\|R_3^j(l,r)\right\|_2 \geq 2\sqrt{\frac{p\log(n)}{r-l+1}}\right) \\
&\leq 2Dn^2(p-1)\exp\left(-2p\log(n)\right) \\
&\leq c_2\exp\left(-2p\log(n) + 2\log(n)\right) \\
&\leq c_2\exp\left(-c_3\log(n)\right)
\end{aligned}$$

since $p > 1$. This concludes the proof. □

Conclusion and perspectives

In this thesis, several contributions have been made in the context of vector data that are observed on network structures. Such data, known as graph vectors or graph signals, are recorded in many real-world scenarios and there is an increasing need to design learning algorithms adapted to them. In this work, two *a priori* distinct tasks has been considered. On one hand, the problem of event detection, which can be split in two different problems i.e. anomaly detection and change-point detection. On the other hand, the graph learning problem that is useful in the most common scenario where the underlying network structure over which the vectors are observed is unknown. Although a priori distinct, these two tasks were successfully linked in the final chapter, where the graph was allowed to change in time. This thesis work also illustrated the idea that a concrete problem, in our case the detection of a Sigfox Base Station failure, can lead to more general and theoretical questions. Finally, the implementation of the different algorithms that has been proposed were made available online to allow and encourage the reproducibility of the results, which was a commitment from the beginning of the thesis project.

Regarding the various contributions presented in the manuscript and the research perspectives they suggest: In Chapter 2, we presented a simple novelty detection algorithm aiming to detect abnormal level of communication activity at the level of a node in a communication network. The algorithm relied on the intuition that the level of activity of a node can be predicted by looking at the level of activity recorded at its neighboring nodes. Thanks to an access to a normal data set and conventional supervised learning methods, the relationship between nodes activity has been learned. Afterwards, an anomaly was detected whenever the predicted level of activity was far from the real one. The method was shown to perform well on both synthetic and real-world data. In particular, it was shown to solve the problem raised by the industrial collaboration, i.e. the objective of detecting BSs failures in Sigfox network.

There are still some research perspectives to investigate. This includes direct use, when it is known, of the network structure on which the data are observed. For example at Sigfox, the knowledge of the positions of the BSs allows us to reduce the dimension of the learning problem by selecting only neighboring BSs. Can this knowledge have another use? Moreover, one can think about inferring the graph structure during the learning phase. This could be done with, for example, a logistic regression, in the manner of the Ising models of Chapter 4. Finally, a proper theoretical investigation on prediction-error based anomaly detection algorithms should be done in future works. In particular, this could be made in the framework of [36] or [68] which characterize the quality of a scoring function using notions related to the estimation of minimum volume sets.

In Chapter 3, we elaborated a new graph learning algorithm in the framework of GSP. The graph vectors were assumed to enjoy a sparse representation in the graph spectral domain, a feature which is known to carry information related to the cluster structure of a graph and which is also a key hypothesis of sampling algorithms. The signals were

as well assumed to behave smoothly with respect to the underlying graph structure. To tackle the problem, we proposed a new optimization program that learns the Laplacian of the graph and we provided two algorithms to solve it, called IGL-3SR and FGL-3SR. Based on a 3-steps alternating procedure, both algorithms relied on standard minimization methods such as manifold gradient descent or linear programming. While IGL-3SR ensures convergence, FGL-3SR acts as a relaxation and is significantly faster since its alternating process relies on multiple closed-form solutions.

Some further investigation remains to be done. In particular, one could think of studying the convergence properties of the FGL-3SR algorithm. Also, a statistical inference method could be considered to learn the graph. The factor analysis model presented in Section 5 of Chapter 3 seems to be a good basic model for a statistical estimation of the parameters. The latter could be learned using the EM algorithm or variational methods.

Finally, in Chapter 4, we combined the problems of event detection, in particular change-point detection, with the task of graph inference. This time, the problem was tackled by developing a probabilistic framework where the graph vectors were assumed to be drawn from Ising models. We assumed that the graph structures, i.e the parameter of the Ising models, were allowed to change over time, in a piece-wise constant fashion. Thus, the objective was to identify both the moments at which significant changes occurred in the Ising model, as well as the underlying graph structure governing the signal behavior segment-wise. For this purpose, we proposed to estimate the neighborhood of each node by maximizing a penalized version of its conditional log-likelihood. The objective of the penalization was twofold: it imposed sparsity in the learned graphs and, thanks to a fused-type penalty, it also enforced them to evolve piece-wise constantly. In the end, we provided two change-points consistency theorems and demonstrated the performance of our method on several synthetic data sets and real-world examples.

Here again, some research perspectives remain. Among them, the investigation of the *sparsistency* which corresponds to the consistency of the estimated graph structures: when the number of sample grows in each segment, we must tend to recover the edges of the underlying graph. Another track for further study is the use of a different objective function, e.g. the Interaction Screening Objective [166], which has been shown to be of good quality from both a computational and a statistical point of view in the static scenario.

Overall, the objectives set at the beginning of the manuscript have been met and, as stated above, there are still many directions to explore. In addition to these, a last track was investigated at the end of the thesis, the problem of robustness in machine learning and in particular for the task of non-parametric density estimation (see Chapter A in appendix). This recent interest suggests that studying the robustness of the methods proposed throughout the manuscript is also an interesting track of research that will be investigated.

A

Robust Kernel Density Estimation with Median-of-Means principle

Contents

Abstract

The work presented here is significantly different from that previously studied. For the sake of consistency with the rest of the thesis, it is therefore placed here rather than in the body of the document. In this additional chapter, we introduce a robust nonparametric density estimator combining the popular Kernel Density Estimation method and the Median-of-Means principle (MoM-KDE). This estimator is shown to achieve robustness to any kind of anomalous data, even in the case of adversarial contamination. In particular, while previous works only prove consistency results under known contamination model, this work provides finite-sample high-probability error-bounds without *a priori* knowledge on the outliers. Finally, when compared with other robust kernel estimators, we show that MoM-KDE achieves competitive results while having significant lower computational complexity.

Associated publication:
Robust Kernel Density Estimation with Median-of-Means principle,
Humbert, Pierre*, Le Bars, Batiste*, Minvielle, Ludovic*, and Vayatis, Nicolas

To be Submitted.
* Authors with equal contribution to this work

1 Introduction

Over the past years, the task of learning in the presence of outliers has become an increasingly important objective in both statistics and machine learning. Indeed, in many situations, training data can be contaminated by undesired samples, which may badly affect the resulting learning task, especially in adversarial settings. Building robust estimators and algorithms that are resilient to outliers is therefore becoming crucial in many learning procedures. In particular, the inference of a probability density function from a contaminated random sample is of major concerns.

Density estimation methods are mostly divided into parametric and nonparametric techniques. Among the nonparametric family, the Kernel Density Estimator (KDE) is probably the most known and used for both univariate and multivariate densities [125, 142, 148], but it also known to be sensitive to dataset contaminated by outliers [92, 93, 163]. The construction of robust KDE is therefore an important area of research, that can have useful applications such as anomaly detection and resilience to adversarial data corruption. Yet, only few works have proposed such robust estimators.

Kim and Scott [93] proposed to combine KDE with ideas from M-estimation to construct the so-called Robust Kernel Density Estimator (RKDE). However, no consistency results were provided and robustness was rather shown experimentally. Later, RKDE was proven to converge to the true density, however at the condition that the dataset remains uncorrupted [162]. More recently, Vandermeulen and Scott [163] proposed another robust estimator, called Scaled and Projected KDE (SPKDE). Authors proved the L_1-consistency of SPKDE under a variant of the Huber's ε-contamination model where two strong assumptions are made [83]. First, the contamination parameter ε is known, and second, the outliers are drawn from an uniform distribution when outside the support of the true density. Unfortunately, as they did not provided rates of convergence, it still remains unclear at which speed SPKDE converges to the true density. Finally, both RKDE and SPKDE require iterative algorithms to compute their estimators, increasing the overall complexity of their construction.

In statistical analysis, another idea to construct robust estimators is to use the Median-of-Means principle (MoM). Introduced by Nemirovsky and Yudin [122], Jerrum et al. [86], and Alon et al. [8], the MoM was first designed to estimate the mean of a real random variable. It relies on the simple idea that rather than taking the average of all the observations, the sample is split in several non-overlapping blocks over which the mean is computed. The MoM estimator is then defined as the median of these means. Easy to compute, the MoM properties have been studied by Minsker [119] and Devroye et al. [45] to estimate the means of heavy-tailed distributions. Furthermore, due to its robustness to outliers, MoM-based estimators have recently gained a renewed of interest in the machine learning community [105, 106].

Contributions. In this work, we propose a new robust nonparametric density estimator based on the combination of the Kernel Density Estimation method and the Median-of-

Means principle (MoM-KDE). We place ourselves in a more general framework than the classical Huber contamination model, called $\mathcal{O} \cup \mathcal{I}$, which gets rid of any assumption on the outliers. We demonstrate the statistical performance of the estimator through finite-sample high-confidence error bounds in the L_∞-norm and show that MoM-KDE's convergence rate is the same as KDE without outliers. Additionally, we prove the consistency in the L_1-norm, which is known to reflect the global performance of the estimate. To the best of our knowledge, this is the first work that presents such results in the context of robust kernel density estimation, especially under the $\mathcal{O} \cup \mathcal{I}$ framework. Finally, we demonstrate the empirical performance of MoM-KDE on both synthetic and real data and show the practical interest of such estimator as it has a lower complexity than the baseline RKDE and SPKDE.

2 Median-of-Means Kernel Density Estimation

We first recall the classical kernel density estimator. Let $X_1, \cdots, X_n$ be independent and identically distributed (i.i.d.) random variables that have a probability density function (pdf) $f(\cdot)$ with respect to the Lebesgue measure on $\mathbb{R}^d$. The Kernel Density Estimate of f (KDE), also called the *Parzen–Rosenblatt estimator*, is a nonparametric estimator given by

$$\hat{f}_n(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right) , \tag{A.1}$$

where $h > 0$ and $K : \mathbb{R}^d \longrightarrow \mathbb{R}_+$ is an integrable function satisfying $\int K(u)du = 1$ [158]. Such a function $K(\cdot)$ is called a *kernel* and the parameter h is called the *bandwidth* of the estimator. The bandwidth is a smoothing parameter that controls the bias-variance tradeoff of $\hat{f}_n(\cdot)$ with respect to the input data.

While this estimator is central in statistic, a major drawback is its weakness against outliers [91–93, 163]. Indeed, as it assigns uniform weights $1/n$ to every points regardless of whether X_i is an outlier or not, inliers and outliers contribute equally in the construction of the KDE, which results in undesired "bumps" over outlier locations in the final estimated density (see Figure A.1). In the following, we propose a KDE-based density estimator robust to the presence of outliers in the sample set. These outliers are considered in a general framework described in the next section.

2.1 Outlier setup

Throughout the chapter, we consider the $\mathcal{O} \cup \mathcal{I}$ framework introduced by Lecué and Lerasle [105]. This very general framework allows the presence of outliers in the dataset and relax the standard i.i.d. assumption on each observation. We therefore assume that the n random variables are partitioned into two (unknown) groups: a subset $\{X_i \mid i \in \mathcal{I}\}$ made of inliers, and another subset $\{X_i \mid i \in \mathcal{O}\}$ made of outliers such that $\mathcal{O} \cap \mathcal{I} = \emptyset$ and $\mathcal{O} \cup \mathcal{I} = \{1, \ldots, n\}$. While we suppose the $X_{i \in \mathcal{I}}$ are i.i.d. from a distribution that admits a density f with respect to the Lebesgue measure, no assumption is made on the outliers $X_{i \in \mathcal{O}}$. Hence, these outlying points can be dependent, adversarial, or not even drawn from a proper probability distribution.

The $\mathcal{O} \cup \mathcal{I}$ framework is related to the well-known Huber's ε-contamination model [83] where it is assumed that data are i.i.d. with distribution $g = \varepsilon f_{\mathcal{I}} + (1-\varepsilon) f_{\mathcal{O}}$, and $\varepsilon \in [0, 1)$; the distribution $f_{\mathcal{I}}$ being related to the inliers and $f_{\mathcal{O}}$ to the outliers. However, there are several important differences. First, in the $\mathcal{O} \cup \mathcal{I}$ the proportion of outliers is fixed and equals to $|\mathcal{O}|/n$, whereas it is random in the Huber's ε-contamination model [107]. Second, the $\mathcal{O} \cup \mathcal{I}$ is less restrictive. Indeed, contrary to Huber's model which considers that inliers and outliers are respectively i.i.d from the same distributions, $\mathcal{O} \cup \mathcal{I}$ does not make a single assumption on the outliers.

2.2 MoM-KDE

We now present our main contribution, a robust kernel density estimator based on the MoM. This estimator is essentially motivated by the fact that the classical kernel density estimation at one point corresponds to an empirical average (see Equation (A.1)). Therefore, the MoM principle appears to be an intuitive solution to build a robust version of the KDE. A formal definition of MoM-KDE is given below.

Definition A.1. *(MoM Kernel Density Estimator) Let $1 \leq S \leq n$, and let $B_1, \cdots, B_S$ be a random partition of $\{1, \cdots, n\}$ into S non-overlapping blocks B_s of equal size $n_s \triangleq n/S$. The MoM Kernel Density Estimator (MoM-KDE) of f at x_0 is given by*

$$\hat{f}_{MoM}(x_0) \propto \text{Median}\left(\hat{f}_{n_1}(x_0), \cdots, \hat{f}_{n_S}(x_0)\right) , \tag{A.2}$$

where $\hat{f}_{n_s}(x_0)$ is the value of the standard kernel density estimator at x_0 obtained via the samples of the s-th block B_s. Note that $\hat{f}_{MoM}(\cdot)$ is not necessarily a density as its integral may not be equal to 1. When needed, we thus normalize it by its integral the same way it is proposed by Devroye and Lugosi [44].

Broadly speaking, MoM estimators appear to be a good tradeoff between the unbiased but non robust empirical mean and the robust but biased median [106]. A visual example of the robustness of MoM-KDE is displayed in Figure A.1. We now give a simple example highlighting the robustness of MoM-KDE.

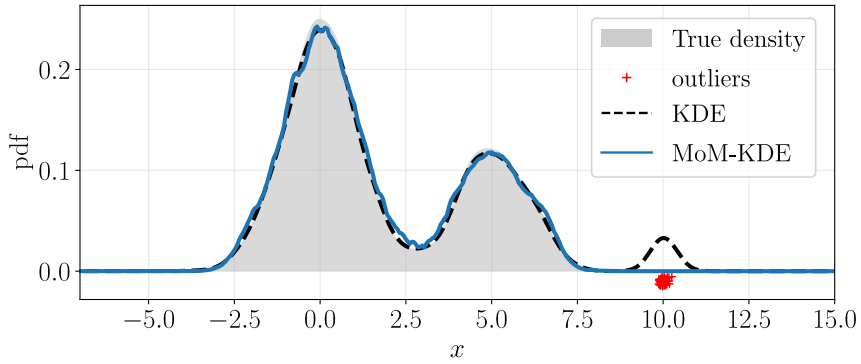
Example A.1. *(MoM-KDE v.s. Uniform KDE) Let the inliers be i.i.d. samples from a uniform distribution on the interval $[-1, 1]$ and the outliers be i.i.d. samples from another uniform distribution on $[-3, 3]$. Let the kernel function be the uniform kernel, $x_0 = 2$ and $h \in (0, 1)$. Then if $S > 2|\mathcal{O}|$, we obtain*

$$|\hat{f}_{MoM}(x_0) - f(x_0)| = 0 \quad a.s. \quad \textit{and} \quad \mathbb{P}\left(|\hat{f}_n(x_0) - f(x_0)| = 0\right) = (1 - h/3)^{|\mathcal{O}|} \neq 1 \, .$$

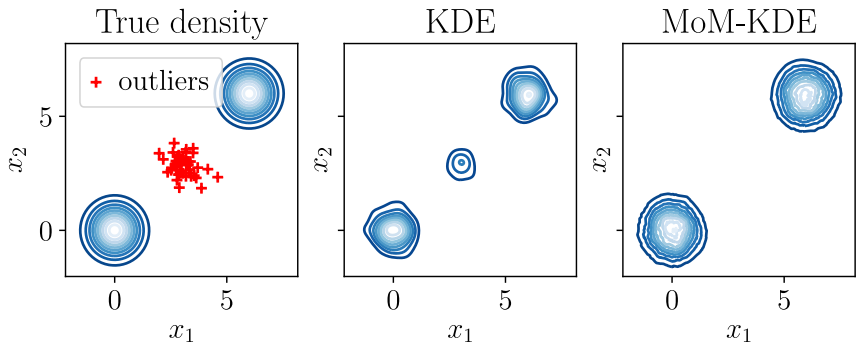
This result shows that the MoM-KDE makes (almost surely) no error at the point x_0. On the contrary, the KDE here has a non-negligible probability to make an error.

2.3 Time complexity

The complexity of MoM-KDE to evaluate one point is the same as the standard KDE, $\mathcal{O}(n)$; $\mathcal{O}(S \cdot \frac{n}{S})$ for the block-wise evaluation and $\mathcal{O}(n)$ to compute the median with the *median-of-medians algorithm* [20]. Since RKDE and SPKDE are KDEs with modified weights, they also perform the evaluation step in $\mathcal{O}(n)$ time. However, these weights need to be learnt,



(a) One-dimensional



(b) Two-dimensional

Figure A.1: True density, outliers, KDE, and MoM-KDE. (a) Estimates from a 1-D true density and outliers from a normal density centered in $\mu_{\mathcal{O}} = 10$ with variance $\sigma^2_{\mathcal{O}} = 0.1$. (b) Estimates from a 2-D true density and outliers from a normal density centered in $\mu_{\mathcal{O}} = (3, 3)$ with variance $\sigma^2_{\mathcal{O}} = 0.5 I_2$.

Table A.1: Computational complexity

Method	Learning	Evaluation	Iterative method
KDE [125]	–	$\mathcal{O}(n)$	no
RKDE [93]	$\mathcal{O}(n_{iter} \cdot n^2)$	$\mathcal{O}(n)$	yes
SPKDE [163]	$\mathcal{O}(n_{iter} \cdot n^2)$	$\mathcal{O}(n)$	yes
MoM-KDE	–	$\mathcal{O}(n)$	no

thus requiring an additional non-negligible computing capacity. Indeed, each one of them rely on an iterative method – the iteratively reweighted least squares algorithm and the projected gradient descent algorithm, that both have a complexity of $\mathcal{O}(n_{iter} \cdot n^2)$, where n_{iter} is the number of needed iterations to reach a reasonable accuracy. MoM-KDE on the other hand does not require any learning procedure. Note that the evaluation step can be accelerated through several ways, hence potentially reducing computational time of all these competing methods [7, 15, 71, 172]. Theoretical time complexities are gathered in Table A.1.

3 Theoretical analysis

In this section, we give a finite-sample high-probability error bound in the L_∞-norm for MoM-KDE under the $\mathcal{O} \cup \mathcal{I}$ framework. To our knowledge, we are the first to provide such error bounds in robust kernel density estimation under this framework. In particular, our objective is to prove that even with a contaminated dataset, MoM-KDE achieves a similar convergence rate than KDE without outliers [88, 150, 169]. In order to build this high-probability error bound, it is assumed, among other standard hypotheses, that the true density is Holder-continuous, a smoothness property usually considered in KDE analysis [88, 158, 169]. In addition, we show the consistency in the L_1-norm. In this last result, we will see that the aforementioned assumptions are not necessary to obtain the consistency. In the following, we give the necessary definitions and assumptions to perform our non-asymptotic analysis.

3.1 Setup and assumptions

Let us first list the usual assumptions, notably on the considered kernel function, that will allow us to derive our results. They are standard in KDE analysis, and are chosen for their simplicity of comprehension [88, 158]. More general hypotheses could be made in order to obtain the same results, notably assuming kernel of order ℓ (see for example the works of Tsybakov [158] and Wang et al. [169]).

Assumption 1. (Bounded density) $\|f\|_\infty < \infty$.

We make the following assumptions on the kernel K.

Assumption 2. (Density kernel) $\forall u \in \mathbb{R}^d, K(u) \geq 0$, and $\int K(u)du = 1$.

Assumption 3. (Spherically symmetric and non-increasing) There exists a non-increasing function $k : \mathbb{R}_+ \longrightarrow \mathbb{R}_+$ such that $K(u) = k(\|u\|)$ for all $u \in \mathbb{R}^d$, where $\|\cdot\|$ is any norm of $\mathbb{R}^d$.

Assumption 4. (Exponentially decaying tail) There exists positive constants $\rho, C_\rho, t_0 > 0$ such that for all $t > t_0$

$$k(t) \leq C_\rho \cdot \exp(-t^\rho) \ .$$

All the above assumptions are respected by most of the popular kernels, in particular the Gaussian, Exponential, Uniform, Triangular, Cosine kernel, etc. Furthermore, the last assumption implies that for any $m > 0$, we have $\int \|u\|^m K(u)du < \infty$ (finite norm moment) [88]. Finally, when taken together, these assumptions imply that the kernel satisfies the VC property [169]. Theses are key properties to provide the bounds presented in the next section.

Before stating our main results, we recall the definition of the Holder class of functions.

Definition A.2. *(Holder class) Let T be an interval of $\mathbb{R}^d$, and $0 < \alpha \leq 1$ and $L > 0$ be two constants. We say that a function $f : T \to \mathbb{R}$ belongs to the Holder class $\Sigma(L, \alpha)$ if it satisfies*

$$\forall x, x' \in T, \quad |f(x) - f(x')| \leq L\|x - x'\|^\alpha \ . \tag{A.3}$$

This definition implies a smoothness regularization on the function f, and is a convenient property to bound the bias of KDE-based estimators.

3.2 L_∞ and L_1 consistencies of MoM-KDE

This section states our central finding, a L_∞ finite-sample error bound for MoM-KDE that proves its consistency and yields the same convergence rate as KDE with uncontaminated data. The latter is given by the following Lemma partly proven by Sriperumbudur and Steinwart [150] and verified several times in the literature [65, 88, 169].

Lemma A.1. *(L_∞ error-bound of the KDE without anomalies) Suppose that f belongs to the class of densities $\mathcal{P}(\alpha, L)$ defined as*

$$\mathcal{P}(\alpha, L) \triangleq \left\{ f \mid f \geq 0, \int f(x)dx = 1, \text{ and } f \in \Sigma(\alpha, L) \right\} , \tag{A.4}$$

where $\Sigma(\alpha, L)$ is the Holder class of function on $\mathbb{R}^d$ (Definition A.2). Grant assumptions 1 to 4 and let $n > 1$, $h \in (0, 1)$ and $S \geq 1$ such that $nh^d \geq S$ and $nh^d \geq |\log(h)|$. Then with probability at least $1 - \exp(-S)$, we have

$$\|\hat{f}_n - f\|_\infty \leq C_1 \sqrt{\frac{S|\log(h)|}{nh^d}} + C_2 h^\alpha , \tag{A.5}$$

where $C_2 = L \int \|u\|^\alpha K(u)du < \infty$ and C_1 is a constant that only depends on $\|f\|_\infty$, the dimension d, and the kernel properties.

This Lemma comes from the well-known bias-variance decomposition, where we separately bound the variance (see Theorem 3.1 of Sriperumbudur and Steinwart [150]) and the bias (see e.g. [158] or [136]). It shows the consistency of KDE without anomalies, as soon as $h \to 0$ and $nh^d \to \infty$.

We now present our main result. Its objective is to show that even under the $\mathcal{O} \cup \mathcal{I}$ framework, we do not need any additional hypothesis – besides the ones of the previous lemma – to show that MoM-KDE achieves the same convergence rate as KDE when used with uncontaminated data.

Proposition A.1. *(ℓ_∞-error-bound of the MoM-KDE under the $\mathcal{O} \cup \mathcal{I}$) Suppose that f belongs to the class of densities $\mathcal{P}(\alpha, L)$ and grant assumptions 1 to 4. Let S be the number of blocks, $\delta > 0$ such that $S > (2 + \delta)|\mathcal{O}|$, and $\Delta = (1/(2 + \delta) - |\mathcal{O}|/S)$. Then, for any $h \in (0, 1)$, δ sufficiently small, and $n \geq 1$ such that $nh^d \geq S \log(2(2 + \delta)/\delta)$, and $nh^d \geq S|\log(h)|$, we have with probability at least $1 - \exp(-2\Delta^2 S)$,*

$$\|\hat{f}_{MoM} - f\|_\infty \leq C_1 \sqrt{\frac{S \log\left(\frac{2(2+\delta)}{\delta}\right) |\log(h)|}{nh^d}} + C_2 h^\alpha , \tag{A.6}$$

where $C_2 = L \int \|u\|^\alpha K(u)du < \infty$ and C_1 is a constant that only depends on $\|f\|_\infty$, the dimension d, and the kernel properties.

The proof is given in the supplementary material. From equation (A.6), the optimal choice of the bandwidth is $h \asymp \left(\frac{S\log(n)}{n}\right)^{1/(2\alpha+d)}$ leading to the final rate of $\left(\frac{S\log(n)}{n}\right)^{\alpha/(2\alpha+d)}$. This convergence rate is the same (up to a constant) to the one of KDE without anomalies, with the same exponential control (Lemma A.1). Note that when there is no outlier, i.e. $|\mathcal{O}| = 0$, the bound holds for $S = 1$, and we recover the classical KDE minimax optimal rate [169]. In addition, the previous proposition states that the convergence of the MoM-KDE only depends on the number of outliers in the dataset, and not on their "type". This estimator is therefore robust in a wide range of scenarios, including the adversarial one.

We now give a ℓ_1-consistency result under mild hypotheses, which is known to reflect the global performance of the estimate. Indeed, small ℓ_1 error leads to accurate probability estimation [43].

Proposition A.2. *(ℓ_1-consistency in probability) If $n/S \to \infty$, $h \to 0$, $nh^d \to \infty$, and $S > 2|\mathcal{O}|$, then*

$$\|\hat{f}_{MoM} - f\|_1 \xrightarrow[n\to\infty]{\mathcal{P}} 0\,. \tag{A.7}$$

This result is obtained by bounding the left-hand part by the errors in the healthy blocks only, i.e. those without anomalies. Under the hypothesis of the proposition, these errors are known to converge to 0 in probability [169]. The complete proof is given in supplementary material. Contrary to SPKDE [163], no assumption on the outliers generation process is necessary to obtain this consistency result. Moreover, while we need to assume that the proportion of outliers is perfectly known to prove the convergence of SPKDE, the MoM-KDE converges whenever the number of outliers is overestimated.

3.3 Influence function in the $\mathcal{O} \cup \mathcal{I}$ framework

As a measure of robustness, we now introduce an Influence Function (IF) adapted to the $\mathcal{O} \cup \mathcal{I}$ framework. It is inspired from the classical IF, first proposed by [74], which measures how an estimator changes when the initial distribution is modified by adding a small amount of contamination at a point x'. Therefore, it provides a notion of stability in the Huber model framework [9, 42]. We now define a similar concept under the $\mathcal{O} \cup \mathcal{I}$.

Definition A.3. *(IF$_{\mathcal{O}\cup\mathcal{I}}$) Let $T_n(x_0; \mathcal{I}_n)$ be a density estimator evaluated at x_0 and learned with an healthy data set $\mathcal{I}_n = \{X_i\}_{i=1}^n$. Let $m \in \mathbb{N}$ and $x' \in \mathbb{R}^d$. The IF$_{\mathcal{O}\cup\mathcal{I}}$ is defined as:*

$$\text{IF}_{\mathcal{O}\cup\mathcal{I}}(x_0, x', m; \mathcal{I}_n, T_n) \triangleq \left|T_n(x_0; \mathcal{I}_n) - T_n(x_0; \mathcal{I}_n \cup \{x'\}_{i=1}^m)\right|\,,$$

where by healthy points we mean inliers i.e. samples that are independently drawn from the true density function.

Given this definition, IF$_{\mathcal{O}\cup\mathcal{I}}$ quantifies how much the value at x_0 of an estimated density function changes whenever the healthy dataset is increased by m points located at x'. Therefore, the link with the notion of stability is made obvious: the smaller IF$_{\mathcal{O}\cup\mathcal{I}}$ is, the more stable and thus robust the estimator is.

In the next proposition, we provide a lower bound on the number of added samples m over which the $\mathrm{IF}_{\mathcal{O}\cup\mathcal{I}}$ of the MoM-KDE is lower that the one of KDE with high probability.

Proposition A.3. *Let $x', x_0 \in \mathbb{R}^d$ and $\mathcal{I}_n$ be an healthy data set. Grant assumptions 1 to 4 and denote*

$$a \triangleq \sum_{i\in\mathcal{I}_n} K\left(\frac{X_i - x_0}{h}\right), \quad b \triangleq K\left(\frac{x' - x_0}{h}\right) .$$

Let $S > 2m$ with $m \in [\![0, \frac{n}{2}[\![$ the number of added samples and $\delta > 0$ such that $|b - a/n| > C_\rho\sqrt{2\delta S/n}$.
If $m \geq \frac{C_\rho\sqrt{2n\delta S}}{|b-a/n|-C_\rho\sqrt{2\delta S/n}}$, then with probability higher than $1 - 4\exp(-\delta)$ we have:

$$\mathit{IF}_{\mathcal{O}\cup\mathcal{I}}(x_0, x', m; \mathcal{I}_n, \hat{f}_{MoM}) \leq \mathit{IF}_{\mathcal{O}\cup\mathcal{I}}(x_0, x', m; \mathcal{I}_n, \hat{f}_{KDE}) .$$

Given the previous proposition, the lower bound on m over which the MoM-KDE is better than KDE is not necessarily easy to interpret. When everything is fixed except x_0 and x', we see that the bound is low whenever $|b - a/n| = |K(\frac{x_0-x'}{h}) - \frac{1}{n}\sum K(\frac{x_0-X_i}{h})|$ is large. A sufficient condition for this is to take x' far from the sampling set $\mathcal{I}_n$, i.e take x' as an outlier. Under this condition, the bound will get even lower whenever x_0 gets closer to x'.

4 Numerical experiments

In this section, we display numerical results supporting the relevance of MoM-KDE. All experiences were run over a personal laptop computer using `Python`. The code of MoM-KDE is made available online[1].

Comparative methods. In the following experiments, we propose to compare MoM-KDE to the classical KDE and two robust versions of KDE, called RKDE [93] and SPKDE [163].

As previously explained, RKDE takes the ideas of robust M-estimation and translate it to kernel density estimation. Authors point out that classical KDE estimator can be seen as the minimizer of a squared error loss in the Reproducing Kernel Hilbert Space $\mathcal{H}$ corresponding to the chosen kernel. Instead of minimizing this loss, they propose to minimize a robust version of it, $\sum_i \rho(\|\phi(X_i) - g\|_{\mathcal{H}})$, with respect to $g \in \mathcal{H}$. Here ϕ is the canonical feature map and $\rho(\cdot)$ is either the robust Huber or Hampel function. The solution of the newly expressed problem is then found using the iteratively reweighted least squares algorithm.

SPKDE proposes to scale the standard KDE in a way that it decontaminates the dataset. This is done by minimizing the function $\|\beta\hat{f}_n - g\|_2$ with respect to g, belonging to the convex hull of $\{k_h(\cdot, X_i)\}_{i=1}^n$. Here, β is an hyperparameter that controls the robustness and $\hat{f}_n$ is the KDE estimator. The minimization is shown to be equivalent to a quadratic program over the simplex, solved via projected gradient descent.

[1] https://github.com/lminvielle/mom-kde. For the sake of comparison, we also implemented RKDE and SPKDE.

Metrics. The performance of the MoM-KDE is measured through three metrics, two are used to measure the similarity between the estimated and the true density, and one describes performances of an anomaly detector based on the estimated density. The first one is the Kullback-Leibler divergence [98] which is the most used in robust KDE [91–93, 163]. Used to measure the similarity between distributions, it is defined as

$$D_{\mathrm{KL}}(\hat{f}\|f) = \int \hat{f}(x) \log\left(\frac{\hat{f}(x)}{f(x)}\right) dx \ .$$

As the Kullback-Leibler divergence is non-symmetric and may have infinite values when distributions do not share the same support, we also consider the Jensen-Shannon divergence [57, 110]. It is a symmetrized version of D_{KL}, with positive values, bounded by 1 (when the logarithm is used in base 2), and has found applications in many fields, such as deep learning [70] or transfer learning [143]. It is defined as

$$D_{\mathrm{JS}}(\hat{f}\|f) = \frac{1}{2}\left(D_{\mathrm{KL}}(\hat{f}\|g) + D_{\mathrm{KL}}(f\|g)\right), \quad \text{with} \quad g = \frac{1}{2}(\hat{f}+f) \ .$$

Motivated by real-world application, the third metric is not related to the true density, which is usually not available in practical cases. Instead, we quantify the capacity of the learnt density to detect anomalies using the well-known Area Under the ROC Curve criterion (AUC). An input point x_0 is considered abnormal whenever $\hat{f}(x_0)$ is below a given threshold.

Hyperparameters. All estimators are built using the Gaussian kernel. The number of blocks in MoM-KDE is selected on a regular grid of 20 values between 1 and $2|\mathcal{O}|+1$ in order to obtain the lowest D_{JS}. The bandwidth h is chosen for KDE via the pseudo-likelihood k-cross-validation method [159], and used for all estimators. The construction of RKDE follows exactly the indications of its authors [93] and $\rho(\cdot)$ is taken to be the Hampel function as they empirically showed that it is the most robust. For SPKDE, the true ratio of anomalies is given as an input parameter.

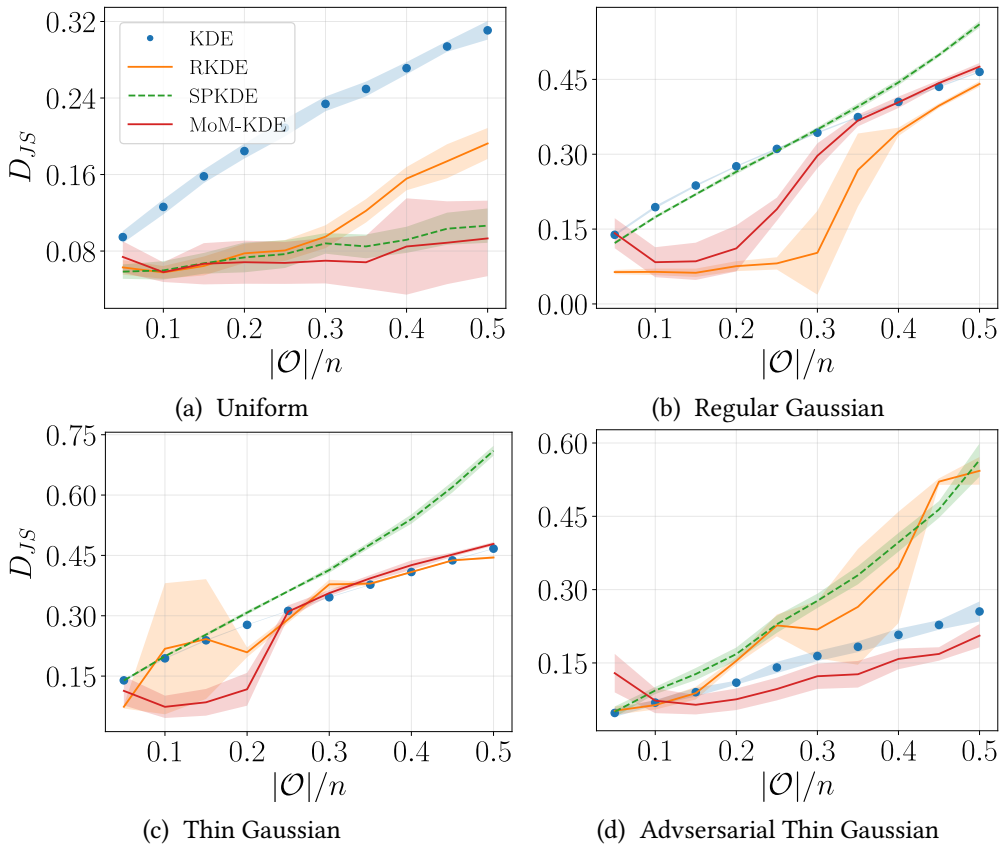
4.1 Results on synthetic data.

To evaluate the efficiency of the MoM-KDE against KDE and its robust competitors, we set up several outlier situations. In all theses situations, we draw $N = 1000$ inliers from an equally distributed mixture of two normal distribution $\mathcal{N}(\mu_1, \sigma_1)$ and $\mathcal{N}(\mu_2, \sigma_2)$ with $\mu_1 = 0$, $\mu_2 = 6$, and $\sigma_1 = \sigma_2 = 0.5$. The outliers however are sampled through various schemes:

(a) **Uniform.** A uniform distribution $U([\mu_1 - 3, \mu_2 + 3])$ which is the classical setting used for outlier simulation.

(b) **Regular Gaussian.** A *similar*-variance normal distribution $\mathcal{N}(3, 0.5)$ located between the two inlier clusters.

(c) **Thin Gaussian.** A *low*-variance normal distribution $\mathcal{N}(3, 0.01)$ located between the two inliers clusters.



(a) Uniform (b) Regular Gaussian

(c) Thin Gaussian (d) Advsersarial Thin Gaussian

Figure A.2: Density estimation with synthetic data. The displayed metric is the Jensen-Shannon divergence. A lower score means a better estimation of the true density.

(d) **Adversarial Thin Gaussian.** A low variance normal distribution $\mathcal{N}(0, 0.01)$ located on one of the inliers' Gaussian mode. This scenario can be seen as adversarial as an ill-intentioned agent may hide wrong points in region of high density. It is the most challenging setting for standard robust estimators as they are in general robust to outliers located outside the support of the density we wish to estimate.

For all situations, we consider several ratios of contamination and set the number of outliers $|\mathcal{O}|$ in order to obtain a ratio $|\mathcal{O}|/n$ ranging from 0.05 to 0.5 with 0.05-wide steps. Finally, to evaluate the pertinence of our results, for each set of parameters, data are generated 10 times.

We display in Figure A.2 the results over synthetic data using the D_{JS} score. The average scores and standard deviations over the 10 experiments are represented for each outlier scheme and ratio. Overall, the results show the good performance of MoM-KDE in all the considered situations. Furthermore, they highlight the dependency of the two competitors to the type of outliers. Indeed, as SPKDE is designed to handle uniformly distributed outliers, the algorithm struggles when confronted with differently distributed outliers (see Figure A.2 b, c, d). RKDE performs generally better, but fails against adversarial contamination, which may be explained by its tendency to down-weight points located in low-density

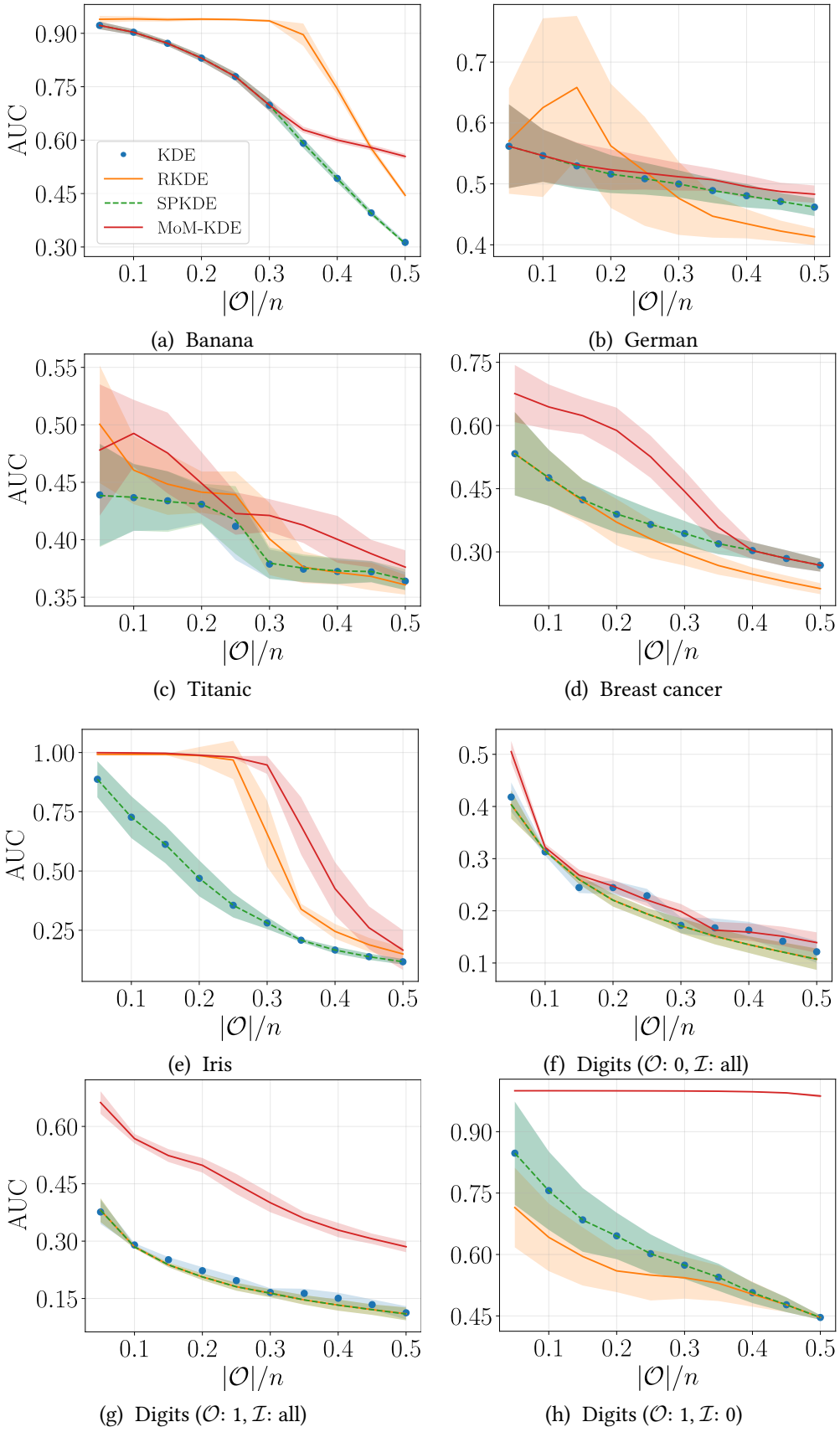
regions, which for this particular case correspond to the inliers. Results over D_{KL} and AUC showing very similar results, they are not reported here.

4.2 Results on real data.

Experiments are also conducted over six classification datasets: Banana, German, Titanic, Breast-cancer, Iris, and Digits. They contain respectively $n = 5300, 1000, 2201, 569, 150$ and 1797 data points having $d = 2, 20, 3, 30, 4$ and 64 input dimensions. They are all publicly available either from open repositories [2] (for the first three) or directly from Scikit-learn package (for the last three) [128]. We follow the approach of Kim and Scott [93] that consists in setting the class labeled 0 as outliers and the rest as inliers. To artificially control the outlier proportion, we randomly downsample the abnormal class to reach a ratio $|\mathcal{O}|/n$ ranging from 0.05 to 0.5 with 0.05-wide steps. When a dataset does not contain enough outliers to reach a given ratio, we similarly downsample the inliers. For each dataset and each ratio, the experiments are performed 50 times, the random downsampling resulting in different learning datasets. The empirical performance is evaluated through the capacity of each estimator to detect anomalies, which we measure with the AUC.

Results are displayed in Figure A.3. With the Digits dataset, we also explore additional scenarios with changing inlier and outlier classes (specified in figure titles). Overall, results are in line with performances observed over synthetic experiments, achieving good results in comparison to its competitors. Note that even in the highest dimensional scenarios, i.e. Digits and Breast cancer ($d = 64$ and $d = 30$), MoM-KDE still behaves well, outperforming its competitors.

[2] http://www.raetschlab.org/Members/raetsch/benchmark/



(a) Banana (b) German

(c) Titanic (d) Breast cancer

(e) Iris (f) Digits ($\mathcal{O}$: 0, $\mathcal{I}$: all)

(g) Digits ($\mathcal{O}$: 1, $\mathcal{I}$: all) (h) Digits ($\mathcal{O}$: 1, $\mathcal{I}$: 0)

Figure A.3: Anomaly detection with real datasets, measured with AUC over varying outlier proportion. A higher score means a better detection of the outliers. For Digits, we specify which classes are chosen to be inliers ($\mathcal{I}$) and outliers ($\mathcal{O}$).

5 Conclusion

This additional chapter introduced MoM-KDE, a new efficient way to perform robust kernel density estimation. The method has been shown to be consistent in both L_∞ and L_1 error-norm in presence of very generic outliers, enjoying a similar rate of convergence than the KDE without outliers. MoM-KDE achieved good empirical results in various situations while having a lower computational complexity than its competitors.

This work proposed to use the coordinate-wise median to construct its robust estimator. Future works will investigate the use of other generalization of the median in high dimension, e.g. the geometric median. In addition, further investigation will include a deeper statistical analysis under the hurdle contamination model in order to analyse the minimax optimality [112] of MoM-KDE.

6 Technical proofs

Lemma A.1. *(L_∞ error-bound of the KDE without anomalies) Suppose that f belongs to the class of densities $\mathcal{P}(\alpha, L)$ defined as*

$$\mathcal{P}(\alpha, L) \triangleq \left\{ f \mid f \geq 0, \int f(x)dx = 1, \text{ and } f \in \Sigma(\alpha, L) \right\} ,$$

where $\Sigma(\alpha, L)$ is the Hölder class of function on $\mathbb{R}^d$. Grant assumptions 1 to 4 and let $n > 1$, $h \in (0, 1)$ and $S \geq 1$ such that $nh^d \geq S$ and $nh^d \geq |\log(h)|$. Then with probability at least $1 - \exp(-S)$, we have

$$\|\hat{f}_n - f\|_\infty \leq C_1 \sqrt{\frac{S|\log(h)|}{nh^d}} + C_2 h^\alpha ,$$

where $C_2 = L \int \|u\|^\alpha K(u)du < \infty$ and C_1 is a constant that only depends on $\|f\|_\infty$, the dimension d, and the kernel properties.

Proposition A.1. *(L_∞ error-bound of the MoM-KDE under the $\mathcal{O} \cup \mathcal{I}$) Suppose that f belongs to the class of densities $\mathcal{P}(\alpha, L)$ and grant assumptions 1 to 4. Let S be the number of blocks, $\delta > 0$ such that $S > (2 + \delta)|\mathcal{O}|$, and $\Delta = (1/(2 + \delta) - |\mathcal{O}|/S)$. Then, for any $h \in (0, 1)$, δ sufficiently small, and $n \geq 1$ such that $nh^d \geq S \log(2(2 + \delta)/\delta)$, and $nh^d \geq S|\log(h)|$, we have with probability at least $1 - \exp(-2\Delta^2 S)$,*

$$\|\hat{f}_{MoM} - f\|_\infty \leq C_1 \sqrt{\frac{S \log\left(\frac{2(2+\delta)}{\delta}\right) |\log(h)|}{nh^d}} + C_2 h^\alpha ,$$

where $C_2 = L \int \|u\|^\alpha K(u)du < \infty$ and C_1 is a constant that only depends on $\|f\|_\infty$, the dimension d, and the kernel properties.

Proof. From the definition of the MoM-KDE, we have the following implication [106]

$$\left\{ \sup_x \left| \hat{f}_{MoM}(x) - f(x) \right| \geq \varepsilon \right\} \Longrightarrow \left\{ \sup_x \sum_{k=1}^{S} I\left(\left| \hat{f}_{n_s}(x) - f(x) \right| > \varepsilon \right) \geq S/2 \right\} .$$

Thus to upper-bound the probability of the left-hand event, it suffices to upper-bound the probability of the right-hand event. Moreover, we have

$$\begin{aligned}
& \left| \hat{f}_{n_s}(x) - f(x) \right| \leq \sup_x \left| \hat{f}_{n_s}(x) - f(x) \right| \\
\Longrightarrow \quad & I\left(\left| \hat{f}_{n_s}(x) - f(x) \right| > \varepsilon \right) \leq I\left(\sup_x \left| \hat{f}_{n_s}(x) - f(x) \right| > \varepsilon \right) \\
\Longrightarrow \quad & \sum_{k=1}^{S} I\left(\left| \hat{f}_{n_s}(x) - f(x) \right| > \varepsilon \right) \leq \sum_{s=1}^{S} I\left(\sup_x \left| \hat{f}_{n_s}(x) - f(x) \right| > \varepsilon \right) \\
\Longrightarrow \quad & \sup_x \sum_{s=1}^{S} I\left(\left| \hat{f}_{n_s}(x) - f(x) \right| > \varepsilon \right) \leq \sum_{s=1}^{S} I\left(\sup_x \left| \hat{f}_{n_s}(x) - f(x) \right| > \varepsilon \right) ,
\end{aligned}$$

which implies that

$$\mathbb{P}\left(\sup_x \sum_{s=1}^{S} I\left(\left|\hat{f}_{n_s}(x) - f(x)\right| > \varepsilon\right) \geq S/2\right) \leq \mathbb{P}\left(\sum_{s=1}^{S} I\left(\sup_x \left|\hat{f}_{n_s}(x) - f(x)\right| > \varepsilon\right) \geq S/2\right) .$$

Let $Z_s = I\left(\sup_x \left|\hat{f}_{n_s}(x) - f(x)\right| > \varepsilon\right)$ and let $\mathcal{S} = \left\{s \in \{1, \cdots, S\} \mid B_s \cap \mathcal{O} = \emptyset\right\}$ i.e. the set of indices s such that the block B_s does not contain any outliers. Since $\sum_{s \in \mathcal{S}^C} I(\cdot)$ is bounded by $|\mathcal{O}|$, almost surely, the following holds.

$$\begin{aligned}
&\sum_{s=1}^{S} I\left(\sup_x \left|\hat{f}_{n_s}(x) - f(x)\right| > \varepsilon\right) = \sum_{s=1}^{S} Z_s = \sum_{s \in \mathcal{S}} Z_s + \sum_{s \in \mathcal{S}^C} Z_s \\
&\leq \sum_{s \in \mathcal{S}} Z_s + |\mathcal{O}| \\
&= \sum_{s \in \mathcal{S}} [Z_k - \mathbb{E}(Z_s) + \mathbb{E}(Z_s)] + |\mathcal{O}| \\
&= \sum_{s \in \mathcal{S}} [Z_s - \mathbb{E}(Z_s)] + \sum_{s \in \mathcal{S}} \mathbb{E}(Z_s) + |\mathcal{O}| \\
&\leq \sum_{s=1}^{S} [Z_s - \mathbb{E}(Z_s)] + S \cdot \mathbb{E}(Z_1) + |\mathcal{O}| \\
&\leq \sum_{s=1}^{S} [Z_s - \mathbb{E}(Z_s)] + S \cdot \mathbb{P}\left(\sup_x \left|\hat{f}_{n_1}(x) - f(x)\right| > \varepsilon\right) + |\mathcal{O}| ,
\end{aligned} \tag{A.8}$$

where Z_1 is assumed, without loss of generality, to be associated to a block not containing outliers. This block always exists thanks to the hypothesis $S > (2+\delta)|\mathcal{O}|$.

Let $\varepsilon = C_1\sqrt{\frac{S\log(\frac{2(2+\delta)}{\delta})|\log(h)|}{nh^d}} + C_2 h^\alpha$, then using Lemma 1 with $S = \log(\frac{2(2+\delta)}{\delta})$, we have

$$\mathbb{P}\left(\sup_x \left|\hat{f}_{n_1}(x) - f(x)\right| > \varepsilon\right) \leq \frac{\delta}{2(2+\delta)} .$$

Combining this last inequality with equation (A.8) leads to

$$\mathbb{P}\left(\sum_{s=1}^{S} I\left(\sup_x \left|\hat{f}_{n_s}(x) - f(x)\right| > \varepsilon\right) \geq S/2\right) \leq \mathbb{P}\left(\sum_{s=1}^{S} [Z_s - \mathbb{E}(Z_s)] + S \cdot \frac{\delta}{2(2+\delta} + |\mathcal{O}| \geq S/2\right)$$

$$\leq \mathbb{P}\left(\sum_{s=1}^{S}[Z_s - \mathbb{E}\left(Z_s\right)] \geq S\left(\frac{1}{2} - \frac{\delta}{2(2+\delta)} - \frac{|\mathcal{O}|}{S}\right)\right)$$
$$\leq \mathbb{P}\left(\sum_{s=1}^{S}[Z_s - \mathbb{E}\left(Z_s\right)] \geq S\left(\frac{1}{2+\delta} - \frac{|\mathcal{O}|}{S}\right)\right)$$

Tacking $\Delta = \left(\frac{1}{2+\delta} - \frac{|\mathcal{O}|}{S}\right) > 0$ and applying Hoeffding's inequality to the right-hand side of the previous equation gives

$$\mathbb{P}\left(\sum_{s=1}^{S} I\left(\sup_x \left|\hat{f}_{n_s}(x) - f(x)\right| > \varepsilon\right) \geq S/2\right) \leq e^{-2S\Delta^2} ,$$

which concludes the proof.

□

Proposition A.2. *(L_1-consistency in probability) If $n/S \to \infty$, $h \to 0$, $nh^d \to \infty$, and $S > 2|\mathcal{O}|$, then*

$$\|\hat{f}_{MoM} - f\|_1 \xrightarrow[n\to\infty]{\mathcal{P}} 0 .$$

Proof. We first rewrite the MoM-KDE as

$$\hat{f}_{MoM}(x) = \sum_{s=1}^{S} \hat{f}_{n_s}(x) I_{A_s}(x) ,$$

where $A_s = \left\{x \mid \hat{f}_{MoM}(x) = \hat{f}_{n_s}(x)\right\}$. Without loss of generality, we assume that

$$A_k \bigcap_{s\neq\ell}^{S} A_\ell = \emptyset, \quad \bigcup_{s=1}^{S} A_s = \mathbb{R}^d, \quad \text{and} \quad \sum_{s=1}^{S} I_{A_s}(x) = 1 .$$

$$\begin{aligned}
\int \left|\hat{f}_{MoM}(x) - f(x)\right| dx &= \int \left|\sum_{s=1}^{S} \hat{f}_{n_s}(x) I_{A_s}(x) - f(x)\right| dx \\
&= \int \left|\sum_{s=1}^{S} \left(\hat{f}_{n_s}(x) - f(x)\right) I_{A_s}(x)\right| dx \\
&\leq \int \sum_{s=1}^{S} \left|\hat{f}_{n_s}(x) - f(x)\right| I_{A_s}(x) dx \\
&= \sum_{s=1}^{S} \int_{A_s} \left|\hat{f}_{n_s}(x) - f(x)\right| dx \\
&= \sum_{s\in\mathcal{S}} \int_{A_s} \left|\hat{f}_{n_s}(x) - f(x)\right| dx + \sum_{s\in\mathcal{S}^C} \int_{A_s} \left|\hat{f}_{n_s}(x) - f(x)\right| dx .
\end{aligned} \tag{A.9}$$

From the L_1-consistency of the KDE in probability, if the number of anomalies grows at a small enough speed [43], the left part is bounded, i.e.

$$\sum_{s\in\mathcal{S}}\int_{A_s}\left|\hat{f}_{n_s}(x)-f(x)\right|dx \leq \sum_{s\in\mathcal{S}}\int\left|\hat{f}_{n_s}(x)-f(x)\right|dx \xrightarrow[n\to\infty]{\mathcal{P}} 0\ . \tag{A.10}$$

We now upper-bound the right part of equation (A.9). Let consider a particular block A_s where $s\in\mathcal{S}^C$. In this block, the estimator f_{n_s} is selected and is calculated with samples containing anomalies. As $\forall x\in A_s$, $f_{n_s}(x)$ is the median (by definition), if $S>2|\mathcal{O}|$, we can always find a $s'\in\mathcal{S}$ such that $f_{n_s}(x)\leq f_{n_{s'}}(x)$ or $f_{n_s}(x)\geq f_{n_{s'}}(x)$.
Now let denote $A_s^+=\left\{x\in A_s \mid \hat{f}_{n_s}(x)\geq f(x)\right\}$ and $A_s^-=\left\{x\in A_s \mid \hat{f}_{n_s}(x)< f(x)\right\}$. We have $A_s^+\cup A_s^-=A_s$ and each one of these blocks can be decomposed respectively into $|\mathcal{S}|$ sub-blocks (not necessarily disjoint) $\{A_s^{s',+}\}_{s'\in\mathcal{S}}$ and $\{A_s^{s',-}\}_{s'\in\mathcal{S}}$ such that $\forall s'\in\mathcal{S}$, $A_s^{s',+}=\left\{x\in A_s \mid \hat{f}_{n_{s'}}(x)\geq \hat{f}_{n_s}(x)\geq f(x)\right\}$ and $A_s^{s',-}=\left\{x\in A_s \mid \hat{f}_{n_{s'}}(x)\leq \hat{f}_{n_s}(x)< f(x)\right\}$. Finally, the right-hand term of equation (A.9) can be upper-bounded by

$$\begin{aligned}
\sum_{s\in\mathcal{S}^C}\int_{A_s}\left|\hat{f}_{n_s}(x)-f(x)\right|dx &\leq \sum_{s\in\mathcal{S}^C}\int_{A_s^+}\left|\hat{f}_{n_s}(x)-f(x)\right|dx+\int_{A_s^-}\left|\hat{f}_{n_s}(x)-f(x)\right|dx\\
&\leq \sum_{s\in\mathcal{S}^C}\sum_{s'\in\mathcal{S}}\int_{A_s^{s',+}}\left|\hat{f}_{n_s}(x)-f(x)\right|dx+\int_{A_s^{s',-}}\left|\hat{f}_{n_s}(x)-f(x)\right|dx\\
&\leq \sum_{s\in\mathcal{S}^C}\sum_{s'\in\mathcal{S}}\int_{A_s^{s',+}}\left|\hat{f}_{n_{s'}}(x)-f(x)\right|dx+\int_{A_s^{s',-}}\left|\hat{f}_{n_{s'}}(x)-f(x)\right|dx\\
&\leq \sum_{s\in\mathcal{S}^C}\sum_{s'\in\mathcal{S}}\int\left|\hat{f}_{n_{s'}}(x)-f(x)\right|dx+\int\left|\hat{f}_{n_{s'}}(x)-f(x)\right|dx
\end{aligned}$$

Since $\forall s'\in\mathcal{S}$ we have $\int|\hat{f}_{n_{s'}}(x)-f(x)|dx \xrightarrow[n\to\infty]{\mathcal{P}} 0$, we can conclude using similar arguments as those used for (A.10) that $\sum_{s\in\mathcal{S}^C}\int_{A_s}|\hat{f}_{n_s}(x)-f(x)|dx \xrightarrow[n\to\infty]{\mathcal{P}} 0$, which concludes the proof.

□

Proposition A.3. *Let $x', x_0\in\mathbb{R}^d$ and $\mathcal{I}_n$ be an healthy data set. Grant assumptions 1 to 4 and denote*

$$a\triangleq\sum_{i\in\mathcal{I}}K\left(\frac{X_i-x_0}{h}\right),\quad b\triangleq K\left(\frac{x'-x_0}{h}\right)\ .$$

Let $S>2m$ with $m\in[\![0,\frac{n}{2}[\![$ the number of added samples and $\delta>0$ such that $|b-a/n|>C_\rho\sqrt{2\delta S/n}$.
If $m\geq\frac{C_\rho\sqrt{2n\delta S}}{|b-a/n|-C_\rho\sqrt{2\delta S/n}}$, then with probability higher than $1-4\exp(-\delta)$ we have:

$$IF_{\mathcal{O}\cup\mathcal{I}}(x_0,x',m;\mathcal{I}_n,\hat{f}_{MoM})\leq IF_{\mathcal{O}\cup\mathcal{I}}(x_0,x',m;\mathcal{I}_n,\hat{f}_{KDE})\ .$$

Proof.

– The $\text{IF}_{\mathcal{O}\cup\mathcal{I}}$ for the KDE is

$$\begin{aligned}
&\text{IF}_{\mathcal{O}\cup\mathcal{I}}(x_0, x', m, \mathcal{I}_n; \hat{f}_{KDE}) \\
=&\left| \frac{1}{(n+m)h^d} \left(\sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right) + \sum_{i=1}^{m} K\left(\frac{x' - x_0}{h}\right) \right) - \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right) \right| \\
=&\frac{1}{h^d} \left| \frac{1}{n+m} \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right) - \frac{1}{n} \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right) + \frac{m}{n+m} K\left(\frac{x' - x_0}{h}\right) \right| \\
=&\frac{1}{h^d} \left| \left(\frac{1}{n+m} - \frac{1}{n}\right) \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right) + \frac{m}{n+m} K\left(\frac{x' - x_0}{h}\right) \right| \\
=&\frac{1}{h^d} \left| \left(\frac{1}{n+m} - \frac{1}{n}\right) a + \left(\frac{m}{n+m}\right) b \right| = \frac{1}{h^d} \left| \frac{nb - a}{n^2/m + n} \right| ,
\end{aligned}$$

with $a \triangleq \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right)$, and $b \triangleq K\left(\frac{x' - x_0}{h}\right)$.

– $\text{IF}_{\mathcal{O}\cup\mathcal{I}}$ for the MoM-KDE

Let $S > 2m$ be the number of blocks in the MoM-KDE, $\{B_s\}_{s=1}^{S}$ and $\{\tilde{B}_{s'}\}_{s'=1}^{S}$ be respectively the blocks of the contaminated data set $\mathcal{I}_n \cup \{x'\}^m$ and the healthy data set. We have:

$$\begin{aligned}
&\text{IF}_{\mathcal{O}\cup\mathcal{I}}(x_0, x', m, \mathcal{I}_n; \hat{f}_{MoM}) \\
&\quad = \frac{1}{h^d} \Bigg| \text{Median} \left\{ \frac{S}{n+m} \left(\sum_{i \in \mathcal{I}_n \cap B_s} K\left(\frac{X_i - x_0}{h}\right) + \sum_{i \in \{x'\} \cap B_s} K\left(\frac{x' - x_0}{h}\right) \right) \right\}_{s=1}^{S} \\
&\qquad\qquad - \text{Median} \left\{ \frac{S}{n} \sum_{i \in \tilde{B}_{s'}} K\left(\frac{X_i - x_0}{h}\right) \right\}_{s'=1}^{S} \Bigg| \\
&\quad \leq \quad \frac{1}{h^d} \left| \frac{S}{n+m} \sum_{i \in B_s} K\left(\frac{X_i - x_0}{h}\right) - \frac{S}{n} \sum_{i \in \tilde{B}_{s'}} K\left(\frac{X_i - x_0}{h}\right) \right| ,
\end{aligned}$$

where $\tilde{B}_{s'}$ is the block selected by the median for the healthy MoM-KDE. The inequality is obtained by noticing that, with $S > 2m$, there always exists an healthy block B_s that makes it true.

Finally, denoting

$$\left| \sum_{i \in B_s} \frac{S}{n+m} K\left(\frac{X_i - x_0}{h}\right) - \sum_{i \in \tilde{B}_{s'}} \frac{S}{n} K\left(\frac{X_i - x_0}{h}\right) \right| = \left| Z^{(s)} - Z^{(s')} \right| ,$$

and using both the triangular and Hoeffding inequalities, and the fact that $\mathbb{E}(Z^{(s)}) = \mathbb{E}(Z^{(s')})$, we have

$$h^d \cdot \text{IF}_{\mathcal{O}\cup\mathcal{I}}(x_0, x', m, \mathcal{I}_n; \hat{f}_{MoM}) \leq \left| Z^{(s)} - Z^{(s')} - \mathbb{E}(Z^{(s)}) + \mathbb{E}(Z^{(s')}) \right|$$

$$\leq \left|Z^{(s)} - \mathbb{E}(Z^{(s)})\right| + \left|Z^{(s')} - \mathbb{E}(Z^{(s')})\right| \ ,$$

and for $t > 0$,

$$\mathbb{P}\left(\left|Z^{(s)} - \mathbb{E}(Z^{(s)})\right| \geq t\right) \leq 2\exp\left(-\frac{2t^2(n+m))}{SC_\rho^2}\right)$$
$$\mathbb{P}\left(\left|Z^{(s')} - \mathbb{E}(Z^{(s')})\right| \geq t\right) \leq 2\exp\left(-\frac{2t^2 n}{SC_\rho^2}\right) = 2\exp\left(-\frac{2n_s t^2}{C_\rho^2}\right) \ ,$$

where $n_s = n/S$. Furthermore, given two real-valued random variables X, Y, we know that for $t > 0$,

$$\mathbb{P}\left(|X| + |Y| \geq 2t\right) \leq \mathbb{P}\left(|X| \geq t\right) + \mathbb{P}\left(|Y| \geq t\right) \ .$$

Therefore, we have

$$\mathbb{P}\left(\left|Z^{(s)} - \mathbb{E}(Z^{(s)})\right| + \left|Z^{(s')} - \mathbb{E}(Z^{(s')})\right| \geq t\right) \leq 4\exp\left(-\frac{n_s t^2}{2C_\rho^2}\right) \ .$$

Setting $t = \dfrac{C_\rho\sqrt{2\delta}}{\sqrt{n_s}}$ with $\delta > 0$, finally gives $\mathbb{P}\left(\left|Z^{(s)} - Z^{(s')}\right| < t\right) \geq 1 - 4\exp(-\delta)$. We now know that with probability $1 - 4\exp(-\delta)$, $\mathrm{IF}_{\mathcal{O}\cup\mathcal{I}}(x_0, x', m, \mathcal{I}_n; \hat{f}_{MoM}) < \dfrac{1}{h^d}\dfrac{C_\rho\sqrt{2\delta}}{\sqrt{n_s}}$ and we seek for which value of m, this value is smaller than the $\mathrm{IF}_{\mathcal{O}\cup\mathcal{I}}$ of the KDE i.e.

$$\begin{aligned}
\frac{C_\rho\sqrt{2\delta}}{\sqrt{n_s}} \leq \left|\frac{nb-a}{n^2/m+n}\right| \iff & \quad n^2/m + n \leq \frac{\sqrt{n_s}|nb-a|}{C_\rho\sqrt{2\delta}} \\
\iff & \quad 1/m \leq \left(\frac{\sqrt{n_s}|nb-a|}{C_\rho\sqrt{2\delta}} - n\right)/n^2 \\
\iff & \quad m \geq \frac{n^2}{\dfrac{\sqrt{n_s}|nb-a|}{C_\rho\sqrt{2\delta}} - n} \\
\iff & \quad m \geq \frac{n^2}{\dfrac{\sqrt{n}|nb-a|}{C_\rho\sqrt{2\delta S}} - n} \\
\iff & \quad m \geq \frac{n}{\dfrac{\sqrt{n}|b-a/n|}{C_\rho\sqrt{2\delta S}} - 1} \ .
\end{aligned}$$

□

B

Introduction en français

Contents

1 Contexte de la thèse

1.1 Contexte general

Au cours des dernières décennies, la quantité croissante de données multivariées disponibles dans tous les secteurs, de la médecine à l'industrie en passant par les réseaux sociaux, a entraîné un besoin important d'analyse et de modélisation de ces données dans le but d'accomplir diverses tâches. Parmi celles-ci, les tâches de détection d'anomalies et de détection de ruptures (appelées *détection d'événements* lorsque l'on se réfère aux deux) au sein de cette quantité massive de données sont d'une importance majeure. En quelques mots, la détection d'anomalies cherche à trouver, dans un ensemble de données, une petite quantité d'entre elles qui s'écarte du comportement *normal* de la grande majorité. Cette tâche peut être appliquée à n'importe quel ensemble de données et, dans une application réelle, elle peut par exemple caractériser la découverte d'un élément dysfonctionnel dans la chaîne de production d'une entreprise. D'autre part, la détection de rupture est liée au vaste domaine de l'analyse de séries temporelles. Ici, on cherche à déterminer des moments où le modèle sous-jacent aux données a changé (Figure B.1). En sciences de la vie ou en biologie, cette tâche peut par exemple correspondre à la détection d'un changement d'état (e.g. un individu se réveillant, le début de la puberté etc.)

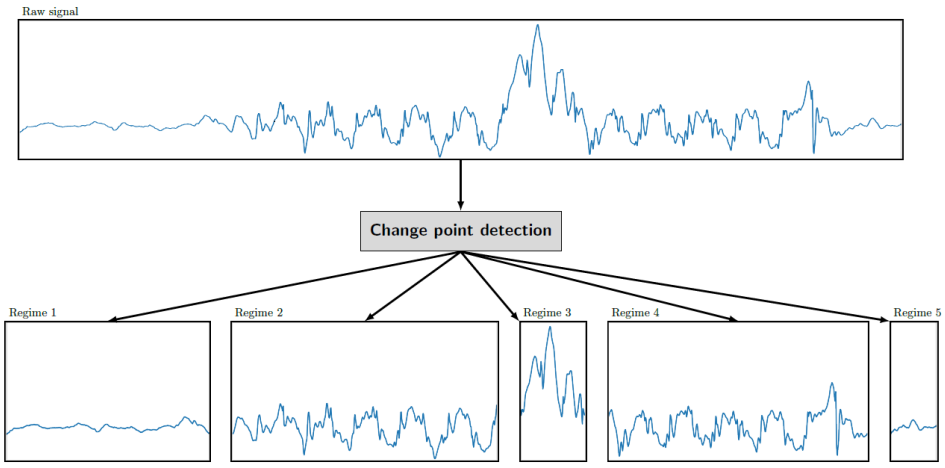


Figure B.1: Illustration simple d'un problème de détection de rupture [157]. Dans cet exemple, le signal est univarié, 4 points de rupture sont détectés, résultant en 5 sous-modèles.

Parmi cette grande quantité de données, un certain nombre d'entre elles apparaissent naturellement sur une structure en réseau, interconnectant les observations ou les variables qui les caractérisent. C'est notamment le cas des données provenant de réseaux sociaux, réseaux de communication, réseaux de transport ou de réseaux de capteurs, où elles sont collectées au niveau des noeuds d'un graphe. Par exemple dans un réseau social, les noeuds du graphe associé peuvent correspondre aux utilisateurs et une arête peut indiquer le lien social qui réside entre deux d'entre eux. Dans un réseau de capteurs, une arête peut simplement caractériser la distance spatiale entre deux capteurs. Généralement, le graphe apporte des connaissances sur le processus qui génère les données (par exemple, les valeurs observées à deux noeuds liés dans le graphe peuvent être fortement corrélés ou être très proches), et construire des modèles ou des algorithmes d'apprentissage – y compris des méthodes de détection d'anomalies ou de détection de ruptures – à partir de ce type de données, tout en prenant en compte la structure de réseau associée, est important pour améliorer les performances d'apprentissage.

Ce type de données est appelé vecteurs sur graphe ou signaux sur graphe. Comme indiqué plus haut, elles se réfèrent simplement à des données vectorielles dont chaque composante est associée à un unique noeud d'un même graphe. Alors que dans certains cas, le graphe est naturellement donné et donc connu *a priori* (par exemple, dans le cas du réseau social ou du réseau de capteurs donné ci-dessus), il existe de nombreux cas où les données admettent une structure de graphe sous-jacente qui n'est pas disponible et que l'on cherche à apprendre à partir de ces données. C'est par exemple le cas en biologie, où l'on s'intéresse à savoir quels gènes (ou protéines) sont exprimés les uns avec les autres [62, 94]. Plus généralement, ce besoin d'inférence de graphe peut apparaître pour tout type de données pour lesquelles on souhaite savoir quelles variables sont liées à quelles autres. Cette tâche peut avoir un fort impact sur la visualisation et la compréhension des données traitées, mais aussi, comme dit précédemment, sur la capacité à construire des algorithmes d'apprentissage plus efficaces.

Ce travail de thèse est profondément ancré dans les différents sujets mentionnés ci-dessus. Il se concentre en particulier sur la détection d'événements dans un ensemble de

vecteurs sur graphe, que le graphe soit connu ou non. Il se concentre également sur la tâche d'inférence du graphe lui-même, et sur la détection de changement dans la structure du graphe sous-jacente aux données.

1.2 Contexte industriel

Cette thèse de doctorat a été réalisée grâce au programme CIFRE (Convention Individuelle de Formation par la Recherche) et l'ANRT (Agence Nationale de la Recherche et de la Technologie). Elle a été sponsorisée par Sigfox, un opérateur mondial de télécommunication dédié à l'Internet des objets (IoT). Plus précisément, Sigfox possède des antennes, appelées stations de base (BS), qui sont installées sur des tours (de la même manière qu'un opérateur de téléphonie mobile) et reçoivent des transmissions de données provenant d'objets tels que des capteurs de stationnement, des alarmes à incendie, des compteurs d'eau ou d'électricité, etc. Ces objets sont détenus par des clients, et le rôle de Sigfox est essentiellement de collecter les données transmises par les objets, via les BS, et de les renvoyer sur un cloud auquel le client a accès. La particularité de Sigfox est son protocole simple, qui permet aux objets d'envoyer des petites quantités de données (12 bytes maximum par "message"), de manière peu coûteuse en énergie et qui sont reçues efficacement par les BS sur de très longue distance. Sans entrer dans les détails, décrivons brièvement le processus d'envoi d'un message dans le protocole Sigfox.

Soit un objet cherchant à transmettre une petite quantité de données (température, préssion, informations binaires, etc.) à son propriétaire. L'information est encodée dans un signal qui est envoyé trois fois, à trois niveaux de fréquence différents. Les signaux sont envoyés sans aucun protocole de sélection de stations de base alentours, ils sont simplement envoyés "dans l'air", en espérant qu'ils seront reçus par au moins une station de base environnante (grâce à une vaste couverture, les signaux sont souvent reçus par un grand nombre d'entre elles). Une fois que certaines des stations de base alentours ont reçu au moins un des trois signaux, celui-ci est décodé et envoyé dans un cloud en utilisant un

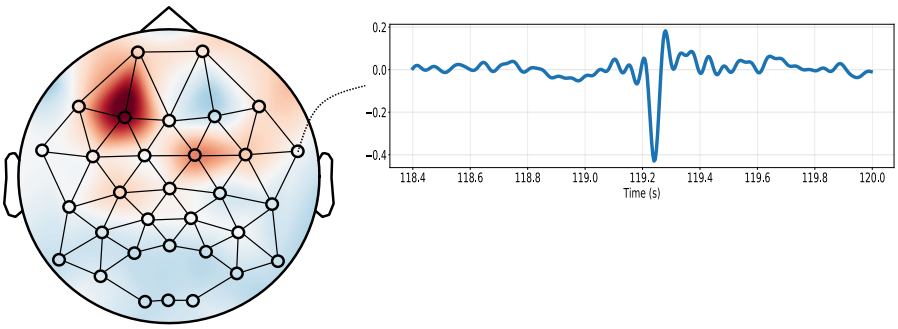


Figure B.2: Exemple de signaux enregistrés sur une structure de graphe. À gauche, des capteurs mesurant l'activité cérébrale dans le temps, représenté par un électroencéphalogramme (EEG), sont placés sur une tête humaine. Leur position sur la tête induit une structure graphique qui encode leur proximité. Dans cet éxemple, un signal sur graphe correspond à un vecteur de taille égale au nombre de capteurs (c'est-à-dire de noeuds) contenant les valeurs d'EEG à un moment précis. Dans ce scénario, on peut s'attendre à ce que les noeuds proches dans le graphe aient des valeurs similaires. À droite, la série temporelle d'observations enregistrées à un seul noeud.

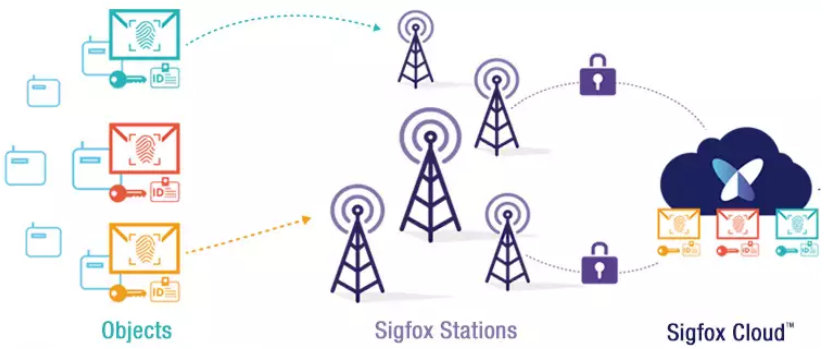


Figure B.3: Résumé de l'architecture de Sigfox.

protocole internet standard tel que la 3G. Un schéma de l'architecture du réseau Sigfox est fourni par la figure B.3. Pour une description plus complète de la technologie Sigfox et des réseaux LPWAN en général, nous invitons le lecteur à consulter [29].

Au départ, l'objectif principal de la collaboration avec Sigfox était de proposer et de développer des méthodes d'apprentissage pour la détection d'anomalies au niveau d'une station de base (par exemple une panne). Il s'agissait d'un sujet nouveau, mal traité par les chercheurs et les ingénieurs de l'entreprise, mais dont la nécessité se faisait de plus en plus importante en raison de l'expansion significative du réseau. Jusqu'alors, les données collectées étaient très peu utilisées à cette fin et seules des méthodes simples, basées sur des dépassements de seuils, étaient employées. Les seuils étaient alors fixés a priori par les experts de terrain et conduisaient à un taux de faux positifs très élevé.

Les données collectées au niveau de chaque BS sont diverses, allant de données matérielles (par exemple la température du moteur) aux informations logicielles (par exemple la version du système d'exploitation utilisée). Toutefois selon les experts, les plus importantes pour détecter les anomalies sont celles liées au spectre des signaux "observés" par la station de base. Malheureusement, l'entiereté du spectre n'est pas collecté et seules quelques statistiques qui le résument le sont. Ces dernières étant essentiellement quelques quantiles des intensités enregistrées sur l'ensemble du spectre. Après plusieurs semaines d'analyse de données, il a finalement été décidé que seules les informations de "réception" de chaque station de base seraient conservées. En d'autres termes, les informations sur l'activité du réseau, c'est-à-dire pour chaque message envoyé dans le réseau, quelle BS l'a reçu ou non. Cette décision est essentiellement motivée par le fait que ces données sont brutes (contrairement aux données du spectre qui sont déjà traitées) et qu'elles comportent a priori peu d'erreurs. En outre, on s'attend intuitivement à ce qu'une défaillance au niveau d'une station de base ait un impact direct sur son niveau d'activité (par exemple, avec une diminution du nombre total de signaux reçus).

Une propriété intéressante de ces données de réception est qu'elles apparaissent naturellement sur une structure de graphe, induite par la distribution spatiale des stations de base. En prenant par exemple le vecteur qui indique, pour un certain message envoyé, quelle station de base l'a reçu ou non, on observe empiriquement que des stations de base proches ont plus de chances de recevoir un même signal, et inversement. Un autre exemple est le vecteur qui spécifie le nombre de signaux reçus par chaque station de base sur une période donnée. Dans ce cas, les stations de base proches auront des valeurs fortement corrélées (voir figure B.4).

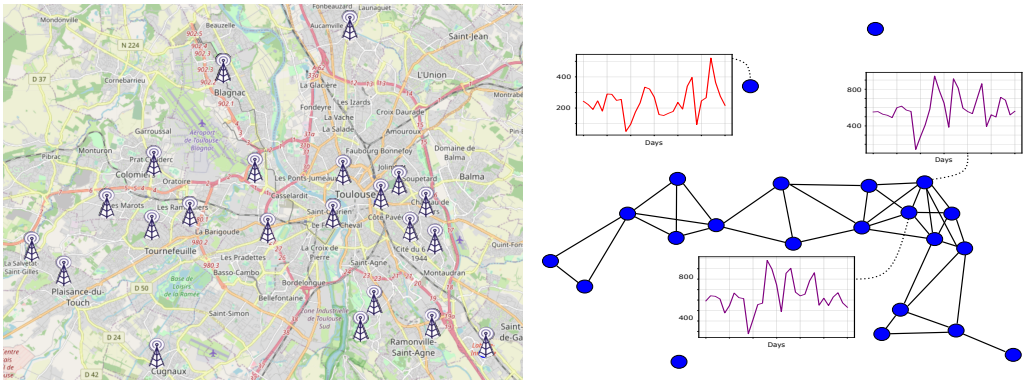


Figure B.4: À gauche, un sous-ensemble de stations de base situées près de Toulouse, en France. À droite, une représentation naïve du réseau sous forme de graphe, qui relie deux noeuds si les stations de base sont situées à moins de 2km l'une de l'autre. Les différentes courbes indiquent pour 3 des noeuds, le nombre de messages reçus quotidiennement sur un mois. Il est clair que les deux noeuds reliés (avec des courbes violettes) sont plus similaire, tant en termes d'échelle que de corrélation, que le noeud isolé (avec une courbe rouge).

Au vu de ces observations et de la structure en réseau inhérente aux données, nous pouvons conclure qu'elles peuvent être considérées comme des vecteurs sur graphes, faisant ainsi le lien vers le problème de détection d'événements dans un ensemble de signaux sur graphes. De plus, rapidement, et comme expliqué en détail dans la section suivante, il a été réalisé que le graphe spatial induit n'était pas toujours adapté aux données, ce qui a conduit au deuxième axe de la thèse, à savoir l'inférence de la structure de graphe elle-même.

2 Objectifs et motivations

Dans ce qui suit, nous décrivons les différents objectifs qui seront au coeur de ce document.

Détecter des changements de comportement ou des anomalies au niveau des noeuds d'un réseau de communication. Ce premier objectif est appliqué et essentiellement motivé par le problème industriel soulevé par la collaboration avec Sigfox. Cependant, ce problème peut se poser pour de nombreux autres type de réseaux de communication (e.g. certain réseaux sociaux ou réseaux d'ordinateurs), et afin que ce travail de thèse soit le plus générale possible, nous chercherons à proposer une ou plusieurs méthodes de détection d'événements applicable pour un large spectre de réseaux de communication, incluant Sigfox. Si dans le cadre de notre collaboration industrielle, une telle tâche pourrait se résumer à la détection d'une panne de station de base, dans un réseau de capteurs [174] elle peut caractériser la détection d'un problème au niveau d'un capteur ou de la valeur qu'il quantifie. Dans un réseau d'ordinateurs ou un réseau social, un tel problème de détection peut se poser, par exemple, pour des questions de sécurité du réseau où l'anomalie peut provenir d'une attaque (piratage, fraude à l'identité, etc.) [6]. Les exemples précédents illustrent bien l'importance d'une telle tâche : les anomalies non détectées peuvent avoir des répercussions importantes sur les performances du réseau considéré. En résumé, nos objectifs seront, tout d'abord, de donner une définition simple d'un réseau de communication, qui puisse correspondre à un large éventail de réseaux. De même, nous examinerons des

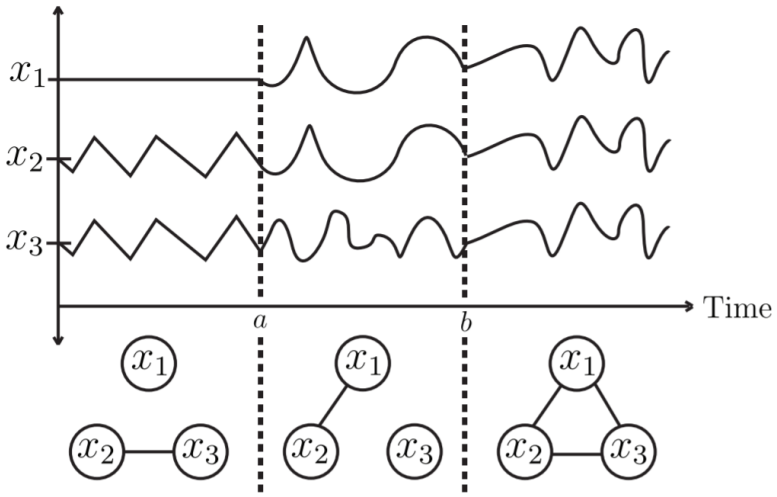


Figure B.5: Illustration de la tâche de détection de ruptures combinée à l'inférence de graphes. Chaque dimension de la série temporelle correspond à un noeud du graphe [73].

notions simples d'anomalies qui peuvent être observées pour différents types de réseaux. Enfin, nous proposerons un moyen de détecter les anomalies. Afin de rester proche de notre application industrielle, nous nous limiterons à la détection au niveau des noeuds, c'est-à-dire à la détection d'événements au niveau d'une seule entité (par exemple une station de base, un ordinateur, etc.).

Détecter des ruptures ou des anomalies dans un ensemble de signaux sur graphe. Cette tâche plus générale a été initialement motivée par la conclusion que nous avons faite dans la section précédente, à savoir le fait que nos données sont des vecteurs sur graphes. De plus, comme expliqué dans la section précédente, ces types de signaux sont fréquemment observés dans le monde réel, et il est donc important de construire des algorithmes de détection d'anomalies ou de détection de ruptures qui leur soient adaptés. Lorsque le graphe est connu, il constitue une connaissance *a priori* du modèle qui génère les données. Il est donc concevable que les performances des algorithmes puissent être améliorées en utilisant ces informations supplémentaires [55]. La connaissance du graphe permet notamment la construction de nouvelles variables ou représentations basées sur celui-ci, et ainsi, permet de découvrir des types d'événements plus complexes [31]. Typiquement, une anomalie détectée au niveau d'un noeud peut provenir d'une valeur anormale, par rapport à elle-même *et* par rapport à ses voisins. Lorsque le graphe est inconnu, peu de conclusions supplémentaire peuvent être faite. Cependant, savoir qu'il existe une structure de graphe sous-jacente indique une chose importante : si l'on veut détecter une anomalie ou un point de rupture au niveau d'une dimension spécifique (c'est-à-dire un noeud spécifique), les valeurs observées aux autres noeuds doivent être prises en compte également puisqu'elles sont liées par les arêtes du graphe [102]. Un graphe inconnu suggère toutefois une première étape d'apprentissage du graphe lui-même, afin de mieux comprendre les données et d'appliquer des algorithmes de détection d'événements plus adaptés. Cette procédure d'apprentissage constitue notre troisième objectif.

Apprendre la structure du graphe sous-jacente à des données vectorielles. Cet objectif se pose dans de nombreux domaines et peut être appliqué à tout type de données vectorielles. La tâche est ici d'inférer la structure de graphe sous-jacente, et bien sûr inconnue, à des données. En d'autres termes, l'objectif est d'apprendre des relations entre des variables, c'est-à-dire avec quelles autres variables, une variable d'intérêt est plus similaire ou liée (en terme de corrélation, d'indépendance conditionnelle, d'échelle, de valeurs, etc.). Cette procédure d'apprentissage se fait en utilisant un ensemble de vecteurs, tous supposés admettre le même graphe sous-jacent. Au cours du processus d'apprentissage, des pénalités structurelles peuvent être imposées, notamment sur la parcimonie du graphe [50, 61, 132], et plusieurs d'entre elles seront étudiées dans cette thèse.

L'apprentissage d'une telle structure de graphe a de nombreuses applications différentes. Tout d'abord, elle permet de comprendre et d'interpêter les données vectorielles considérées grâce à la visualisation simple apportée par le graphe. De plus, une telle tâche d'apprentissage relie souvent les données à un modèle. C'est par exemple le cas des champs aléatoires de Markov [97], qui dans de nombreuses situations suppose une relation linéaire entre les variables, relation déterminée par le graphe lui-même [79, 132]. Dans de tels cas, on peut prédire la valeur d'un noeud en fonction des valeurs des autres noeud. Cela illustre bien l'applicabilité du graphe appris. Enfin, le graphe peut également être utilisé dans de nombreux algorithmes d'apprentissage qui nécessitent un graphe, typiquement l'algorithme du partitionnement spectral [123], certains algorithmes d'apprentissage semi-supervisés [17], dans le cadre du *traitement du signal sur graphe* [124] etc.

L'une des applications les plus célèbres de l'apprentissage des graphes dans le monde réel se trouve en biologie, avec les réseaux d'interaction des gènes. Ceux-ci mettent en évidence les gènes qui sont la plupart du temps exprimés ensemble. À Sigfox, on pourrait penser que le graphe est directement donné par la position spatiale des stations de base (Figure B.4), vu que que des stations de base voisines recevront probablement beaucoup de messages en commun. En pratique, ce n'est pas toujours vrai, deux stations de base situées à une faible distance (à vol d'oiseau) l'une de l'autre peuvent être séparées par un mur ou être à des altitudes différentes, ce qui les rend différentes dans leur capacité à recevoir des signaux. Cette observation nous a permis de conclure que la tâche d'apprentissage de graphe pouvait également être intéressante avec les données de réception Sigfox.

Détecter des changements dans la structure de graphe sous-jacente à des données vectorielles. Cet objectif peut être considéré comme une combinaison de l'apprentissage de graphe et de la détection de ruptures. En fait, contrairement à la grande majorité des techniques de détection de ruptures qui recherchent un changement significatif dans la moyenne d'une série temporelle, la tâche ici est de détecter un changement dans le graphe sous-jacent aux données. L'objectif est donc double : trouver des moments entre lesquels toutes les données vectorielles observées ont la même structure de graphe sous-jacente et apprendre ces graphes. Une illustration de cette tâche est fournie par la Figure B.5 pour une série temporelle à valeur réelle et à 3 dimensions. En plus de déterminer les moments où le système a changé, les méthodes qui répondent à cet objectif tirent également parti des avantages liés à l'inférence des graphes (voir le paragraphe précédent), c'est-à-dire la modélisation, l'applicabilité de certains algorithmes d'apprentissage automatique, etc. En particulier, la visualisation que cet apprentissage permet entraîne une forte compréhension et une grande capacité d'interprétation des points de rupture trouvés [104].

Travailler avec des signaux sur graphe binaires. Ce dernier objectif est essentiellement motivé par le fait que les données vectorielles considérées sont binaires. En effet, rappelons que dans notre cadre d'application, une donnée brute est un vecteur, caractérisant un message Sigfox, qui encode quelle BS a reçu le signal ou non. Néanmoins, ce problème reste important dans de nombreux autres contextes [5], notamment parce qu'il est souvent moins étudié que pour les données vectorielles ou les séries temporelles à valeur réelle.

3 Préliminaires

Dans cette section, nous proposons de rappeler brièvement quelques fondamentaux sur la détection d'événements, à savoir la détection d'anomalies et la détection de ruptures, sur la théorie des graphes et sur les vecteurs observés sur graphes. L'objectif est de fournir quelques définitions, propriétés et algorithmes de base qui seront utiles dans le reste du manuscrit.

3.1 Détection d'anomalies

Dans sa version la plus classique, le problème de détection d'anomalies cherche à trouver dans un ensemble de données, une petite quantité de vecteurs qui ont été générés par une distribution de probabilité différente de celle générant la majorité des points. Cette formulation simple a motivé de nombreuses méthodes statistiques de détection d'anomalies qui supposent essentiellement que les anomalies se trouvent dans des régions de faible densité. Parmi ces méthodes, celle de [58] qui suppose que les données *normales* sont générées par une distribution de probabilité connue à l'avance et qui considère les points se trouvant dans une région de faible probabilité comme anormaux.

Tout en étant la plus générale, la formulation précédente correspond en fait à un cadre particulier de détection d'anomalies. En effet, sur la base des étiquettes disponibles, les tâches de détection d'anomalies peuvent être divisées en trois types. La détection d'anomalie *supervisée*, qui consiste à entraîner l'algorithme sur la base d'un ensemble de données étiquetées comprenant des observations normales et anormales. Ce cadre est donc fortement lié à un problème de classification supervisée [151]. Le second scénario est le

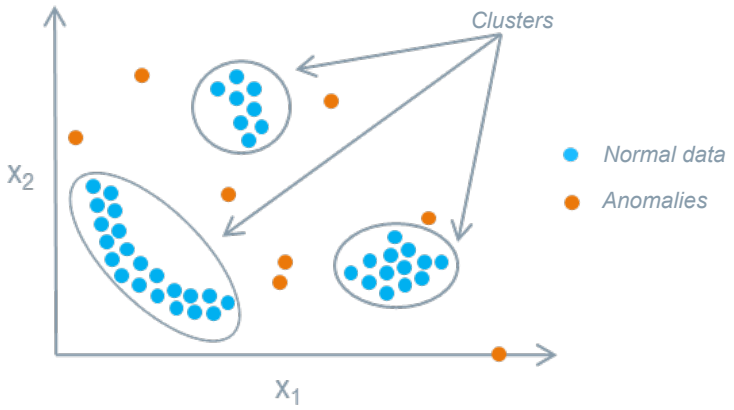


Figure B.6: Un exemple d'anomalies pour un ensemble de données bidimensionnelles.

cadre dit de *détection de nouveauté*, également appelé classification à une classe ou détection d'anomalie semi-supervisée. Dans ce cadre, seules des données normales sont disponibles pour la phase d'apprentissage. C'est le cas dans les applications où les comportements normaux sont connus mais où, par exemple, les intrusions ou les attaques sont inconnues et doivent être détecté. Ce scénario est celui considéré au chapitre 2. Enfin, la détection d'anomalie *non supervisé*, qui fait référence à celle présentée dans le paragraphe précédent : aucune étiquette n'est disponible et l'ensemble des données d'apprentissage contient des données normales et anormales.

La plupart des algorithmes de détection d'anomalies ne se contentent pas d'associé le vecteur d'entrée à une valeur binaire, indiquant si celui-ci est normal ou non. Ils renvoient plutôt une fonction à valeur réelle, appelée *fonction de score*, qui produit, pour un vecteur d'entrée donné, un score d'anomalie à valeur réelle. La force de l'utilisation d'une telle fonction de score est qu'elle permet de classer les échantillons du moins anormal au plus anormal. Cette fonction est très puissante lorsque l'on a beaucoup de données à analyser et que l'on veut classer certaines anomalies par ordre de priorité. De plus, si l'on veut une sortie binaire, il suffit de fixer un seuil au-dessus duquel le score sera considéré comme anormal.

Il existe une grande variété d'algorithmes de détection d'anomalies, de ceux basés sur l'inférence de la densité des données [25, 140, 141, 165], à ceux basés par exemple sur les arbres de décision [111]. Cette grande variété d'algorithme est accentuée par les différents étiquetages expliqués plus haut, mais aussi par le type de données analysées. Nous nous sommes ici concentrés sur les données vectorielles classiques, mais il pourrait s'agir de données temporelles ou même textuelles. Il serait donc impossible de dresser ici une liste exhaustive de ces méthodes, et pour des études complètes, on peut se référer à [28, 127]. Dans ce qui suit, nous présentons un algorithme efficace de détection d'anomalies, pour des données vectorielles standard, qui donne de bons résultats dans les scénarios d'étiquetage non supervisé et semi-supervisé.

3.1.1 SVM à une classe

La machine à vecteurs de support (SVM) à une classe, introduite pour la première fois dans [140], étend le SVM standard pour la classification à deux classes au problème de la détection de nouveauté. En effet, plutôt que d'avoir accès à un ensemble de données étiquetées avec des étiquettes positives et négatives, il suppose que les données d'entrée appartiennent toutes à la classe 1 (la classe normale). Ensuite, au lieu de construire un hyperplan séparant deux classes, il construit un hyperplan séparant les points d'entrée (mappé dans un espace de redescription), de l'origine de l'espace, traité ici comme le seul point de la seconde classe. Formellement, dans sa version à *marge souple*, le SVM à une classe fonctionne comme suit.

Soient $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$, n observations et $\Phi : \mathbb{R}^d \to \mathcal{H}$ une fonction a valeures dans $\mathcal{H}$, un espace de Hilbert à noyau reproduisant [13] associé au noyau k. Pour séparer les données de l'origine, le SVM à une classe résout le programme quadratique suivant :

$$\begin{aligned} &\min_{\omega \in \mathbb{R}^d, \xi \in \mathbb{R}^n, \rho \in \mathbb{R}} \frac{1}{2}\|\omega\|^2 + \frac{1}{n}\sum_{i=1}^{n} \xi_i - \nu\rho \\ &\text{s.t} \quad \langle w, \Phi(x^{(i)}) \rangle \geq \rho - \xi_i, \quad i = 1, \ldots, n \end{aligned}$$

$$\xi_i \geq 0,$$

où $\nu \in (0, 1)$ est un hyperparamètre qui empêche le sur-apprentissage et permet à l'ensemble des données d'entraînement de contenir des valeurs anormales. En fait, on peut montrer que ν correspond à une borne supérieure sur la fraction d'anomalies qui peuvent être présete dans l'ensemble d'apprentissage. La fonction de score utilisée pour détecter les anomalies est donnée par

$$f(x) = \text{sign}(\langle w, \Phi(x^{(i)})\rangle - \rho),$$

qui, comme souhaité, sera positive pour la plupart des vecteurs d'apprentissage.

D'après la définition précédente, le SVM à une classe peut être vu comme un algorithme d'estimation d'un espace de volume minimum contenant presque toutes les données d'entrée. Il est lié à l'estimation d'un ensemble de volume minimal [141] de masse $1 - \nu$, c'est-à-dire l'ensemble de volume minimal, par rapport à la mesure de Lebesgue, ayant une masse de $1 - \nu$, par rapport à la mesure de probabilité des données normales.

3.1.2 Mesures de qualité

Pouvoir évaluer la qualité des algorithmes de détection d'anomalies est, comme pour toute autre méthode d'apprentissage, très important. Lorsque des étiquettes sont disponibles, toutes les mesures de qualité utilisées pour évaluer la qualité d'une fonction de score ou d'un algorithme de classification binaires peuvent également être utilisées. On pourra par exemple analyser la courbe ROC et l'aire sous cette courbe (AUC). L'un des inconvénients de l'utilisation de ces mesures est qu'elles ne sont pas particulièrement adaptées à des classes déséquilibrées. Pour cette raison, des mesures liées à la classe normale, telles que le taux de faux positifs, peuvent être préférées.

Lorsque peu ou pas de labels sont disponibles, la question de l'évaluation de la qualité des algorithmes reste ouverte. Certaines pistes, cherchant à étendre la notion de courbes ROC au scénario sans étiquette, ont notamment été étudiées dans [67] et sont liées aux courbes d'excès de masse et Masse-Volume. Toutefois, ces méthodes ne seront pas prises en compte dans le présent manuscrit et, le cas échéant, des étiquettes seront disponibles.

3.2 Détection de ruptures

La détection de ruptures est une tâche particulière de l'analyse des séries temporelles. Son objectif est de trouver des moments où des changements significatifs se sont produits dans le modèle sous-jacent à une série temporelle. Comme indiqué dans la Section 1.1, cette tâche a de nombreuses applications, que ce soit dans le traitement de la parole [76], en climatologie [134], dans l'analyse des données de trafic d'un réseau [108] etc. Une illustration de ce problème de segmentation est fournie dans la figure B.1.

La tâche de détection de ruptures peut être divisée en deux grandes catégories : la détection *hors ligne* dite *a posteriori*, et celle *en ligne* dite *séquentielle*. Dans le cadre hors ligne, la segmentation est effectuée après que le signal ait été entièrement observé. Au contraire, dans le scénario en ligne, on cherche à trouver les moments de rupture en temps réel, au fur et à mesure que les vecteurs sont observées. Dans cette section, nous nous concentrons sur le scénario hors ligne, lui-même considéré au chapitre 4. Ce problème de détection de rupture hors ligne peut à nouveau être divisé en deux catégories, le cas où le

nombre de rupture à retrouver est *connu* et lorsqu'il est *inconnu*. La plupart du temps, la résolution des deux problèmes diffère avec l'ajout, dans le programme d'optimisation, d'un terme pénalisant le nombre de rupture estimé. Avant d'aller plus loin, décrivons maintenant notre problème de manière plus formelle.

Nous considérons que le cadre statistique décrit dans [157]. Soit $\{x^{(i)}\}_{i=1}^{n}$ une série temporelle à valeurs dans $\mathbb{R}^d$ et supposée stationnaire par morceaux, ce qui signifie qu'il existe des instances de temps $\mathcal{T}^\star = \{t_1^\star, \dots, t_{K^\star}^\star\} \subset \{1 \dots, n\}$ auxquelles le modèle sous-jacent à la série temporelle à changé. L'objectif de la détection de rupture est de trouver les temps $\mathcal{T}^\star$, et donc le nombre de ruptures lorsque celui-ci est inconnu. Pour cela, la plupart des méthodes que l'on trouve dans la littérature recherchent un ensemble d'indices de temps $\mathcal{T} = \{t_k\}_{k=1}^{K} \subset \{1 \dots, n\}$, estimant $\mathcal{T}^\star$ et tel qu'ils minimisent une fonction de la forme

$$\sum_{k=0}^{K} c(\{x^{(i)}\}_{i=t_k+1}^{t_{k+1}}), \tag{B.1}$$

où $t_0 = 0, t_{K+1} = n$ et $c(\cdot)$ est une fonction de coût qui évalue la qualité de la ségmentation. Quand le nombre de ruptures est connu $K = K^\star$, sinon, un terme pénalisant la taille de $\mathcal{T}$ est ajouté à la fonction objective.

Les méthodes de détection de ruptures diffèrent ensuite sur deux aspects principaux. Soit sur la fonction de coût utilisée, généralement liée au modèle sous-jacent aux données (e.g. paramétrique ou non), soit sur la méthode utilisée pour résoudre le problème de minimisation mentionné ci-dessus. Le problème étant combinatoire, de nombreuses solutions ont été proposées (méthode gourmande, programmation dynamique etc.). La grande variété des fonctions de coût et des méthodes de minimisation ne nous permet pas d'être exhaustifs et nous invitons le lecteur à consulter la revue de [157] pour des exemples et des discussions approfondies. Néanmoins, nous donnons ci-dessous un exemple de modèle et de fonction de coût associée qu'il nous semble important de connaître.

Example B.1. *L'approche du maximum de vraisemblance.*
Dans cet exemple, les échantillions de la séries temporelle sont supposés indépendants et identiquement distribués (iid) par morceaux. En d'autres termes, pour une famille de densités paramétriques $\{f(\cdot|\theta)\}$ donnée, nous avons $\forall i = 1 \dots, n$:

$$x^{(i)} \sim \sum_{k=1}^{K^\star} f(\cdot|\theta_k) \mathbb{1}\{t_k^\star \leq k < t_{k+1}^\star\}.$$

Une façon d'apprendre les paramètres d'un tel modèle, et donc d'estimer les ruptures, est de maximiser la vraisemblance. La fonction de coût correspondante est la suivante :

$$c(\{x^{(i)}\}_{i=t_k+1}^{t_{k+1}}) = -\sup_{\theta} \sum_{i=t_k+1}^{t_{k+1}} \log f(x^{(i)}|\theta).$$

Le cadre précédent est probablement l'un des plus considérés pour la détection de rupture [60, 101, 144]. Le modèle i.i.d. par morceaux est par ailleurs celui considéré au chapitre 4, mais avec une fonction de coût légèrement différente.

3.2.1 Mesures de qualité

Là encore, en supposant l'accès aux véritables points de rupture, de nombreuses mesures ont été proposées pour évaluer la qualité des algorithmes de segmentation. Parmi elles, des métriques basées sur celles de la classification binaire (rupture ou non) telles que le score F_1. Cependant, ces dernières ne prennent pas en compte l'aspect temporel du problème et on préférera par exemple utiliser la métrique de Hausdorff. Celle-ci définit l'erreur $h(\mathcal{T}^\star, \mathcal{T})$ de l'ensemble des points de rupture estimés par rapport aux points réels comme étant la plus grande distance temporelle entre un point de changement réel et un estimé :

$$h(\mathcal{T}^\star, \mathcal{T}) \triangleq \max \left\{ \max_{t^\star \in \mathcal{T}^\star} \min_{t \in \mathcal{T}} |t - t^\star|, \max_{t \in \mathcal{T}} \min_{t^\star \in \mathcal{T}^\star} |t - t^\star| \right\}.$$

Une telle mesure a l'avantage de pénaliser à la fois une sur-segmentation et une sous-segmentation.

Alors que les mesures précédente évaluent la qualité d'un algorithme de segmentation de manière empirique, il est également important d'évaluer la qualité d'un algorithme d'un point de vue théorique. Cela se concrétise avec démontrant la consistence des estimateurs [157], qui précise que lorsque le nombre d'échantillons dans chaque segment tend vers l'infini, on a $\mathbb{P}(K = K^\star) \to 0$ et $n^{-1} h(\mathcal{T}^\star, \mathcal{T}) \to 0$ en probabilité.

3.3 Théorie des graphes et modèles pour les vecteurs sur graphes

3.3.1 Définitions basiques

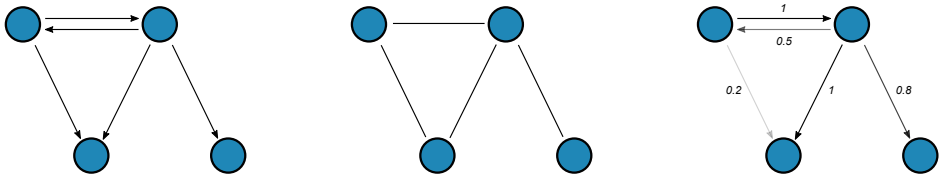
Les graphes sont des objets mathématiques décrivant des systèmes potentiellement complexes via un ensemble d'entités interconnectées, appelées noeuds. Ils apparaissent dans de nombreux domaines et applications, notamment ceux qui font intervenir la notion de réseaux, tels que les réseaux biologiques, les réseaux de neurones, les réseaux de capteurs, les réseaux informatiques, les réseaux de télécommunication, les réseaux sociaux, les réseaux de transport etc. Les graphes sont alors les outils les plus utilisés pour décrire et modéliser ces réseaux. Dans ce qui suit, nous donnons quelques définitions et concepts de base de la théorie des graphes.

Definition B.1. (Graphe dirigé.) *Un graphe dirigé $G = (\mathcal{V}, \mathcal{E})$ est défini via un ensemble fini de noeuds $\mathcal{V} = \{v_1, \ldots, v_p\}$ et un ensemble d'arêtes $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$, qui relient des paires de noeuds entre eux. Si $(u, v) \in \mathcal{E}$, on dit que u est un parent de v et que v est un enfant de u.*

Dans ce qui suit, on considère que pour tout noeud u dans $\mathcal{V}$, $(u, u) \notin \mathcal{E}$. De plus, on peut se référer à un noeud $v_i \in \mathcal{V}$ simplement par son indice i.

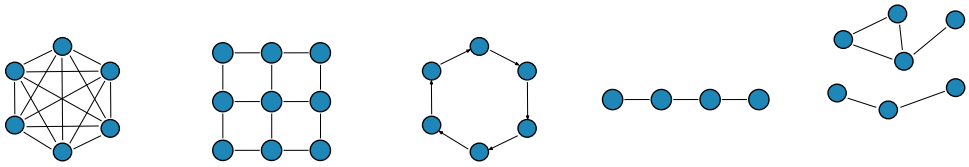
Definition B.2. (Graph non dirigé.) *Un graphe non dirigé $G = (\mathcal{V}, \mathcal{E})$ est un graphe dirigé dont l'ensemble des arêtes $\mathcal{E}$ est symétrique. Autrement dit, $\forall (u, v) \in \mathcal{E}$, $(v, u) \in \mathcal{E}$. Dans ce contexte il n'y a donc pas de notion de parent ou d'enfant et deux noeuds connectés sont simplement appelés voisins.*

Definition B.3. (Graphe pondéré.) *Un graphe pondéré $G = (\mathcal{V}, \mathcal{E})$ est un graphe dont l'ensemble des arêtes $\mathcal{E} = \{(u, v, w_{uv}), u, v \in \mathcal{V}\}$ associe à chaque arête $(u, v) \in \mathcal{E}$ un poids $w_{uv} \in \mathbb{R}_+$. Si le graphe est en plus non orienté, on a $\forall (u, v) \in \mathcal{E}$, $w_{uv} = w_{vu}$.*



(a) Dirigé (b) Non dirigé (c) Pondéré

Figure B.7: Exemples de graphes dirigés, non dirigés et pondérés avec quatre nœuds.



(a) Complet (b) Grille (c) Cyclique (d) Ligne (e) Arbitraire

Figure B.8: Quelques exemples de graphes importants. Le dernier (e) admet deux composantes connexes.

Remark B.1. *Bien que largement supposée, l'hypothèse de positivité des poids n'est pas obligatoire.*

Dans ce qui suit, sauf indication contraire, les graphes sont supposés non dirigés.

Definition B.4. (Matrice d'adjacence.) *Soit $G = (\mathcal{V}, \mathcal{E})$ un graphe de taille p. Sa matrice d'adjacence $A \in \{0,1\}^{p \times p}$ est une matrice binaire dont les entrées indiquent la présence ou l'absence d'arête. $\forall i, j \in \{1, \dots, p\}$:*

$$A_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in \mathcal{E} \\ 0 & \text{otherwise.} \end{cases}$$

La matrice d'adjacence décrit entièrement le graphe qui lui est associé, ce qui la rend utile pour les manipulations mathématiques et la construction de certaines caractéristiques. Par exemple, le nombre total d'arêtes d'un graphe correspond simplement à $||A||_1$ et le **degré** d'un noeud i est $\sum_j A_{ij}$. A est toujours symétrique pour les graphes non dirigés et sa généralisation aux graphes pondérés est la **matrice de poids** W dont les entrées correspondent aux poids des arêtes. Dans ce qui suit, les graphes sont supposés pondérés.

Definition B.5. (Degré and matrice des degrés.) *Soit $G = (\mathcal{V}, \mathcal{E})$ un graphe pondéré de taille p avec une matrice de poids W. $\forall i \in \{1, \dots, p\}$, le degré du noeud v_i est $d_i = \sum_j W_{ij}$. La matrice de degrés D du graphe est la matrice diagonale qui contient les degrés de tout les noeuds.*

Definition B.6. (Matrice Laplacienne.) *La matrice Laplacienne d'un graphe G avec matrice de poids W et matrice de degrés D est la matrice $L = D - W$.*

Tout comme la matrice d'adjacence, la matrice laplacienne décrit entièrement le graphe qui lui est associé. En particulier, elle est connue pour contenir d'importantes caractéristiques topologiques du graphe et pour être liée à la théorie spectrale des graphes. Par exemple, le nombre de valeurs propres nulles de L correspond au nombre de *composante connexe* du graphe. Une composante connexe étant un sous-ensemble de noeuds pour lesquels il existe toujours un chemin entre eux, et pour lesquels il n'existe aucun chemin avec un autre noeud. Une illustration d'un graphe avec deux composantes connexes est donnée sur la figure B.8e.

Definition B.7. (Vecteur sur graphe.) *Un vecteur sur graphe, également appelé signal sur graphe ou fonction sur graphe, est une fonction $x : \mathcal{V} \to \mathbb{R}$ qui assigne une valeur réelle à tous les noeuds d'un graphe $G = (\mathcal{V}, \mathcal{E})$. Cette fonction peut être représentée par un simple vecteur $x \in \mathbb{R}^p$ avec x_i la valeur de x au noeud i.*

Ce dernier objet permet de définir des données vectorielles qui sont observées sur une structure en réseau, un aspect fondamental des données qui sont considérées tout au long de ce travail de thèse. Néanmoins, une question demeure : qu'apporte le graphe à la modélisation de ce type de vecteurs ? En effet, pris comme cela un vecteur sur graphe reste un simple vecteur. Dans les sections suivantes, nous présentons deux points de vue différents qui répondent à cette question. Ces deux cadres sont tous deux considérés dans des chapitres indépendant de la thèse.

3.3.2 Le cadre du traitement du signal sur graphe

Le traitement du signal sur graphe (Graph Signal Processing - GSP) [124, 147] est un domaine relativement récent dont l'objectif est d'étendre la plupart des outils développés dans le domaine du traitement du signal et des images au traitement de signaux sur graphes. Ainsi, des notions telles que la régulariré des signaux, l'échantillonnage ou la représentation spectrale d'un signal ont été développées pour ce type de données. En fait, les signaux temporels et les images sont considérés comme des cas particuliers de signaux sur graphe où le graphe associé correspond soit à une ligne pour un signal temporel, soit à une grille pour une image (voir figure B.8). Dans le contexte du GSP le graphe peut maintenant être quelconque. Enfin, dans ce cadre, et comme pour les signaux temporels ou les images, la valeur enregistrée à un noeud est vue comme une version décalée des valeurs enregistrées aux noeuds voisins.

Dans la suite nous rappelons quelques notions de base du GSP et les propriétés supposées être partagées par la plupart des vecteurs sur graphes.

Definition B.8. (Régularité.) *Soit $G = (\mathcal{V}, \mathcal{E})$ un graphe de taille p avec une matrice de poids W, L sa matrice laplacienne, et $y \in \mathbb{R}^p$ un signal sur ce graphe. Nous disons que y a une régularité de niveau s par rapport au graphe G si*

$$y^\mathsf{T} L y = \frac{1}{2} \sum_{i,j \in [p]} w_{ij} \left(y_i - y_j\right)^2 \leq s \ .$$

La définition précédente donne une notion de régularité pour les signaux sur graphes. Intuitivement, un signal sur graphe y a une régularité de niveau s par rapport à G si les noeuds adjacents du graphe ont des valeurs de signal suffisamment proche. Plus s est petit, plus le signal sur graphe est lisse. Dans le cadre du GSP, les signaux sur graphes sont la plupart du temps supposés lisses par rapport à leurs graphes associés (s petit).

Remark B.2. *Le cas particulier où $s = 0$ implique que tous les noeuds voisins ont la même valeur.*

Definition B.9. (Transformée de Fourier graphique.) *Soit $G = (\mathcal{V}, \mathcal{E})$ un graphe non orienté et $L = X\Lambda X^\mathsf{T}$ la décomposition en éléments propres de sa matrice laplacienne. La transformée de Fourier graphique (GFT) d'un signal sur graphe $y \in \mathbb{R}^p$ est donnée par*

$$h = X^\mathsf{T} y,$$

où les composantes de h sont interprétées comme des coefficients de Fourier, les valeurs propres Λ comme des fréquences et les vecteurs propres X comme une base de Fourier.

Cette définition est initialement motivée par le fait qu'appliquée aux signaux temporels ou aux images, on retrouve la transformée de Fourier classique. De plus, il est empiriquement observé que les vecteurs propres de X associés aux plus petites valeurs propres de Λ présentent moins de variabilité entre les valeurs des noeuds que ceux associés aux plus grandes, ce qui motive également la comparaison avec l'analyse de Fourier.

Definition B.10. (Parcimonie spectrale.) *Soit $k \in \mathbb{N}^+$, on dit qu'un signal sur graphe y admet une représentation spectrale k-parcimonieuse (ou bien que y est k-bandlimité) par rapport à un graphe G, si pour $h = X^\mathsf{T} y$ nous avons*

$$\|h\|_0 \leq k \ , \tag{B.2}$$

où $\|h\|_0$ correspond au nombre d'éléments non nuls de h.

Avec cette définition, y admet une représentation spectrale k-parcimonieuse si le nombre d'éléments non nuls dans son vecteur de coefficients de Fourier est inférieur ou égal à k. Dans le cadre du GSP, être k-bandlimité est la seconde propriété supposée être partagée par la plupart des vecteurs sur graphes. Lorsque k est petit, cela indique notamment que le signal peut être reconstruit à partir d'un petit nombre de valeurs de noeuds. De plus, combiné à la notion de régularité, les coefficients nuls auront plus de chances d'être associés à des grandes valeurs propres (autrement dit des fréquences haute). Cela est en général observé lors de l'analyse spectrale de signaux temporels standards où la grande variabilité des valeurs observées à des noeuds voisins s'explique principalement par la présence de bruit.

3.3.3 Un cadre probabiliste

Dans la partie précédente, les vecteurs et leur graphe associé étaient liés entre eux par des propriétés issues du traitement du signal, à savoir la régularité ou lissité et la parcimonie de la représentation spectrale. Dans le cadre présenté ici, les signaux sur graphes sont vu comme des vecteurs aléatoires tirés selon une distribution de probabilité particulière : un *champ aléatoire de Markov* (Markov Random Fields - MRF). Pour ce type de distribution, le graphe encode une structure de dépendance particulière qui est expliquée dans ce qui suit.

Definition B.11. (Indépendance conditionnelle.) *Soit X, Y et Z trois variables aléatoires à valeur réelle. On désigne par $F_{X|Z=z}(x)$ (respectivement $F_{Y|Z=z}(y)$) la fonction de répartition (cdf) de X (respectivement Y) sachant $Z = z$. On dit alors que X et Y sont conditionnellement indépendantes sachant Z, dénoté $X \perp\!\!\!\perp Y|Z$ si et seulement si $\forall x, y, z$ nous avons :*

$$F_{X,Y|Z=z}(x, y) = F_{X|Z=z}(x) F_{Y|Z=z}(y).$$

Dans une certaine mesure, le fait que X et Y soient conditionnellement indépendants par rapport à Z nous indique que, étant donné Z, connaître X n'apporte aucune information sur Y et inversement.

Remark B.3. *Deux variables conditionnellement indépendantes peuvent être dépendantes et inversement.*

Definition B.12. (Champ aléatoire de Markov.) *Soit $G = (\mathcal{V}, \mathcal{E})$ un graphe non orienté et $X = (X_i)_{i \in \mathcal{V}}$ un vecteur aléatoire dont les entrées sont indexées par les noeuds $\mathcal{V}$. Nous disons que X est tiré d'un MRF associé à G si les propriétés suivantes sont respectées :*

(a) $X_u \perp\!\!\!\perp X_v \mid X_{\mathcal{V}\setminus\{u,v\}}$*, pour toute arête* $(u, v) \notin \mathcal{E}$*.*

(b) $X_u \perp\!\!\!\perp X_{\mathcal{V}\mathcal{N}(u)} \mid X_{\mathcal{N}(u)}, \forall u \in \mathcal{V}$ *où* $\mathcal{N}(u) = \{v \in \mathcal{V} : (u, v) \in \mathcal{E}\}$ *est le* voisinage *de* u*.*

(c) $X_A \perp\!\!\!\perp X_B \mid X_S$*, pour tout sous-ensembles disjoint* $A, B, S \subset \mathcal{V}$ *tels que* S *sépare* A *et* B *i.e. tout chemin de* A *à* B *(et inversement) passe par* S*.*

Remark B.4. *On peut montrer [100] que* $(c) \Rightarrow (b) \Rightarrow (a)$*. Pour certaines distributions de probabilité, la réciproque est également vraie, c'est notamment le cas des variables admettant une fonction de densité positive.*

Compte tenu de la définition précédente, nous comprenons que la construction de MRF n'est pas évidente. Pour cette raison, les MRF sont souvent réduits à la classe des distributions de probabilité qui se *factorise*, une classe de distribution qui a la particularité de valider les trois propriétés requises pour être un MRF.

Definition B.13. (Factorisation.) *Soit $G = (\mathcal{V}, \mathcal{E})$ un graphe non orienté et X un vecteur aléatoire indexé par V admettant une mesure de probabilité $\mathbb{P}_X(\cdot)$. Rappelons qu'une clique est un sous-ensemble de noeuds qui sont tous reliés entre eux. On dit que $\mathbb{P}_X(\cdot)$ se factorise en G si elle est de la forme :*

$$\mathbb{P}_X(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C), \tag{B.3}$$

où $\mathcal{C}$ est l'ensemble de toutes les cliques de G, $\psi_C(\cdot)$ sont des fonctions de potentiel *postitives et Z est une constante de normalisation.*

Les fonctions de potentiel peuvent être arbitraires et, pour une distribution de probabilité donnée qui se factorise, ne sont pas nécessairement définies de manière unique. Les modèles graphiques gaussiens et les modèles d'Ising sont des exemples célèbres de distributions qui factorisent, et qui sont donc des MRF. Le premier est facile à caractériser : pour tout graphe G admettant une matrice de poids W, le vecteur $X \sim \mathcal{N}(\cdot, W^{-1})$ se factorise en G. Quant aux modèles d'Ising, ils sont considérés au chapitre 4 dans lequel des rappels les concernant y sont faits.

4 Organisation du manuscrit

La thèse est organisée comme suit. Chaque chapitre peut être lu indépendamment.

- Chapitre 2: Détection d'anomalies dans des réseaux de communication : applications à Sigfox.

  Ce chapitre est essentiellement consacré à la résolution du premier objectif décrit dans la section 2. La tâche de détection d'anomalies est prise sous l'angle de l'analyse de l'activité d'un réseau de communication. En d'autres termes, les anomalies sont détectées sur la base, par exemple, d'un nombre anormal d'interactions ou de quantité d'informations échangées entre deux noeuds. Ce cadre très général nous permet de considérer une large classe de réseaux de communication, qui inclue Sigfox.

  Dans ce chapitre, nous commençons par un bref aperçu de la littérature sur la détection d'anomalies dans les réseaux, en mettant l'accent sur les méthodes faisant appel à des représentation graphique du reseau et de son activité. Nous présentons également un algorithme simple de détection de nouveauté qui vise à détecter un niveau anormal d'activité au niveau d'un noeud. Cet algorithme repose sur l'intuition que le niveau d'activité d'un noeud peut être déterminé ou prédit en examinant le niveau d'activité enregistré au niveau des noeuds voisins. En utilisant un ensemble de données normales et des méthodes d'apprentissage supervisé conventionnelles, on cherche alors à apprendre cette relation entre l'activité des noeuds. Ainsi, une anomalie est détectée lorsqu'un niveau d'activité prédit est loin du niveau d'activité réellement observé.

  Cette méthode se montre performante aussi bien sur des données synthétiques que sur des données provenant du réseau Sigfox, ce qui nous permet de conclure à la résolution du premier objectif. De plus, l'approche présentée est liée à d'autres objectifs présentés en section 2, en particulier à la tâche de détection d'événements pour des vecteurs sur graphes et d'inférence de graphes, sujets au coeur de la thèse dans les chapitres suivants.

- Chapitre 3: Inférence de structure à partir de signaux sur graphes réguliés et bandlimités.

  Dans ce chapitre, on étudie le problème de l'apprentissage de la structure sous-jacente à un ensemble de vecteurs, c'est-à-dire le graphe sur lequel ils sont observés. Ce chapitre est donc lié au troisième objectif décrit dans la section 2 et se place dans le cadre du traitement du signal sur graphe. On suppose que les vecteurs sur graphe admettent une représentation parcimonieuse dans le domaine spectral du graphe, une propriété qui notamment caractérise des clusters dans un graphe. De plus, les signaux sont supposés régulié par rapport au graphe sous-jacent. Afin d'inférer le graphe, nous proposons un programme d'optimisation qui apprend la matrice Laplacienne associée. Nous présentons deux algorithmes pour le résoudre, appelés IGL-3SR et FGL-3SR. Basés sur une procédure de descente par block, les deux algorithmes reposent sur des méthodes de minimisation standard – telles que de la descente de gradient sur une variété ou de la programmation linéaire – et sont de compléxité moins grande que les algorithmes de l'état de l'art. Alors que IGL-3SR est assuré de converger, FGL-3SR procède à une relaxation qui lui permet d'être significativement

plus rapide. Les deux algorithmes sont évalués sur des données synthétiques et réelles.

- Chapitre 4: Détecter des changements dans la structure de graphe d'un modèle Ising qui évolue dans le temps.

  Ce dernier chapitre aborde les deux derniers objectifs définis en section 2. On se place dans le cadre probabiliste définie dans la section précedente et en particulier on suppose que les vecteurs sur graphe sont tirés selon un modèle d'Ising. Le chapitre se concentre sur la detection de plusieurs ruptures dans un modèle d'Ising qui évolue dans le temps de manière constante par morceaux. L'objectif est d'identifier à la fois les moments où des changements significatifs se produisent dans le modèle d'Ising, ainsi que d'éstimer les structures de graphe sous-jacentes. Pour cela, nous proposons d'estimer le voisinage de chaque noeud en maximisant une version pénalisée de sa log-vraisemblance conditionnelle. L'objectif de la pénalisation que l'on présente est double : elle impose de la parcimonie dans les graphes que l'on apprend et elle les oblige également à évoluer de manière constante par morceaux. En utilisant peu d'hypothèses, nous fournissons deux théorèmes de consistence des ruptures estimées. Ces théorèmes sont les premiers dans le contexte de détection d'un nombre inconnu de rupture dans un modèle d'Ising. Pour finir, des résultats expérimentaux sur plusieurs ensembles de données synthétiques et réels démontrent la performance de notre méthode.

5 Publications

Tous les travaux présentés dans ce manuscrit ont donné lieu à des publications dans des conférences et des revues internationales.

- B. Le Bars and A. Kalogeratos, A Probabilistic Framework to Node-level Anomaly Detection in Communication Networks, In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 2188-2196, 2019
- B. Le Bars[1], P. Humbert[1], L. Oudre and A. Kalogeratos, Learning Laplacian Matrix from Bandlimited Graph Signals, In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2937-2941, 2019
- P. Humbert[1] , B. Le Bars[1] , L. Oudre, A. Kalogeratos and N. Vayatis, Learning Laplacian Matrix from Graph Signals with Sparse Spectral Representation, *Submitted to JMLR*, 2020
- B. Le Bars, P. Humbert, A. Kalogeratos and N. Vayatis, Learning the piece-wise constant graph structure of a varying Ising model, In *Proceedings of the* 37^{th} *International Conference on Machine Learning (ICML)*, 2020
- P. Humbert[1], B. Le Bars[1], L. Minvielle[1] and N. Vayatis, Robust Kernel Density Estimation with Median-of-Means principle, *To be submitted*, 2020

[1]Authors with equal contribution to the work.

Bibliography

[1] A. Abraham, F. Pedregosa, M. Eickenberg, P. Gervais, A. Mueller, J. Kossaifi, A. Gramfort, B. Thirion, and G. Varoquaux. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8:14, 2014.

[2] A. Abraham, M. P. Milham, A. Di Martino, R. C. Craddock, D. Samaras, B. Thirion, and G. Varoquaux. Deriving reproducible biomarkers from multi-site resting-state data: an autism-based example. *NeuroImage*, 147:736–745, 2017.

[3] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[4] C. C. Aggarwal, Y. Zhao, and S. Y. Philip. Outlier detection in graph streams. In *Proc. of the IEEE Intern. Conf. on Data Engineering*, pages 399–409, 2011.

[5] A. Ahmed and E. P. Xing. Recovering time-varying networks of dependencies in social and biological studies. *Proceedings of the National Academy of Sciences*, 106 (29):11878–11883, 2009.

[6] M. Ahmed, A. Mahmood, and J. Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

[7] G. G. Alexander and W. M. Andrew. Rapid evaluation of multiple density models. In *AISTATS*, 2003.

[8] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 58(1):137–147, 1999.

[9] Donald WK Andrews. Stability comparison of estimators. *Econometrica: Journal of the Econometric Society*, pages 1207–1235, 1986.

[10] C. Andris, D. Lee, M. J. Hamilton, M. Martino, C. E. Gunning, and J. A. Selden. The rise of partisanship and super-cooperators in the US House of Representatives. *PloS one*, 10(4):e0123507, 2015.

[11] A. Anis, A. Gadde, and A. Ortega. Towards a sampling theorem for signals on arbitrary graphs. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3864–3868, 2014.

[12] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems*, pages 41–48, 2007.

[13] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[14] R. Arora. On learning rotations. In *Advances in Neural Information Processing Systems*, pages 55–63, 2009.

[15] A. Backurs, P. Indyk, and T. Wagner. Space and time efficient kernel density estimation in high dimensions. In *Advances in Neural Information Processing Systems*, pages 15773–15782, 2019.

[16] O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. of Machine Learning Research*, 9(Mar):485–516, 2008.

[17] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine learning*, 56(1-3):209–239, 2004.

[18] P. Bellec, C. Chu, F. Chouinard-Decorte, Y. Benhajali, D. S. Margulies, and R. C. Craddock. The neuro bureau ADHD-200 preprocessed repository. *Neuroimage*, 144: 275–286, 2017.

[19] K. Bleakley and J.-P. Vert. The group fused lasso for multiple change-point detection. *arXiv preprint arXiv:1106.4199*, 2011.

[20] M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest, and R. E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7:448–461, 1973.

[21] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[22] S. Boyd and L. Vandenberghe. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge University Press, 2018.

[23] C. A. Boyle, S. Boulet, L. A. Schieve, R. A. Cohen, S. J. Blumberg, M. Yeargin-Allsopp, S. Visser, and M. D. Kogan. Trends in the prevalence of developmental disabilities in US children, 1997–2008. *Pediatrics*, 127(6):1034–1042, 2011.

[24] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[25] M. Breunig, H. Kriegel, R. Ng, and J. Sander. Lof: identifying density-based local outliers. In *ACM SIGMOD Record*, volume 29, pages 93–104, 2000.

[26] S. Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[27] L. Bybee and Y. Atchadé. Change-point computation for large graphical models: a scalable algorithm for gaussian graphical models with change-points. *J. of Machine Learning Research*, 19(1):440–477, 2018.

[28] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *Computing Surveys*, 41(3):15:1–15:58, 2009.

[29] B. S. Chaudhari and M. Zennaro. *LPWAN Technologies for IoT and M2M Applications*. Elsevier Science, 2020.

[30] S. Chen, A. Sandryhaila, and J. Kovačević. Sampling theory for graph signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3392–3396, 2015.

[31] Y. Chen, X. Mao, D. Ling, and Y. Gu. Change-point detection of gaussian graph signals with partial information. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3934–3938. IEEE, 2018.

[32] H. Cheng, P.-N. Tan, C. Potter, and S. Klooster. Detection and characterization of anomalies in multivariate time series. In *Proc. of the SIAM Intern. Conf. on Data Mining*, pages 413–424, 2009.

[33] S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero. Learning sparse graphs under smoothness prior. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6508–6512, 2017.

[34] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pages 129–136, 2010.

[35] F. R. K. Chung and F. C. Graham. *Spectral graph theory*. Number 92. American Mathematical Society, 1997.

[36] S. Clémençon and J. Jakubowicz. Scoring anomalies: a m-estimation formulation. In *Artificial Intelligence and Statistics*, pages 659–667, 2013.

[37] A. K. Cline and I. S. Dhillon. Computation of the singular value decomposition. 2006.

[38] M. Corneli, P. Latouche, and F. Rossi. Multiple change points detection and clustering in dynamic networks. *Statistics and Computing*, pages 1–19, 2017.

[39] K. Dadi, M. Rahim, A. Abraham, D. Chyzhyk, M. Milham, B. Thirion, and G. Varoquaux. Benchmarking functional connectome-based predictive models for resting-state fMRI. *Neuroimage*, 192:115–134, 2019.

[40] S. I. Daitch, J. A. Kelner, and D. A. Spielman. Fitting a graph to vector data. In *Proceedings of the International Conference on Machine Learning*, pages 201–208, 2009.

[41] J. S. Damoiseaux, S. Rombouts, F. Barkhof, P. Scheltens, C. J. Stam, S. M. Smith, and C. F. Beckmann. Consistent resting-state networks across healthy subjects. *Proceedings of the national academy of sciences*, 103(37):13848–13853, 2006.

[42] Michiel Debruyne, Andreas Christmann, Mia Hubert, and Johan AK Suykens. Robustness and stability of reweighted kernel based regression. Technical report, 2008.

[43] L. Devroye and L. Gyorfi. *Nonparametric Density Estimation: The L*1 *View*. New York: John Wiley & Sons, 1985.

[44] L. Devroye and G. Lugosi. *Combinatorial methods in density estimation*. Springer Science & Business Media, 2012.

[45] L. Devroye, M. Lerasle, G. Lugosi, and R. I. Oliveira. Sub-gaussian mean estimators. *The Annals of Statistics*, 44(6):2695–2725, 2016.

[46] J. Di and E. Kolaczyk. Complexity-penalized estimation of minimum volume sets for dependent data. *J. of Multivariate Analysis*, 101(9):1910–1926, 2010.

[47] S. Diamond and S. Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17:1–5, 2016.

[48] S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *J. of Machine Learning Research*, 17(83):1–5, 2016.

[49] P. M. Djuric and C. Richard, editors. *Cooperative and Graph Signal Processing – Principles and Applications*. Elsevier, 2018.

[50] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst. Learning laplacian matrix in smooth graph signal representations. *Trans. Signal Processing*, 64(23):6160–6173, 2016.

[51] X. Dong, D. Thanou, M. Rabbat, and P. Frossard. Learning graphs from data: A signal representation perspective. *preprint arXiv:1806.00848*, 2018.

[52] N. Du, L. Song, M. Yuan, and A. J. Smola. Learning networks of heterogeneous influence. In *Advances in Neural Information Processing Systems*, pages 2780–2788, 2012.

[53] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the International Conference on Machine Learning*, pages 272–279, 2008.

[54] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

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

[56] H. E. Egilmez, E. Pavez, and A. Ortega. Graph learning from data under structural and Laplacian constraints. *preprint arXiv:1611.05181*, 2016.

[57] D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information theory*, 49(7):1858–1860, 2003.

[58] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*, 2000.

[59] F. Fazayeli and A. Banerjee. Generalized direct change estimation in ising model structure. In *International Conference on Machine Learning*, pages 2281–2290, 2016.

[60] K. Frick, A. Munk, and H. Sieling. Multiscale change point inference. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 495–580, 2014.

[61] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[62] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004.

[63] A. J. Gibberd and J. DB Nelson. Regularized estimation of piecewise constant gaussian graphical models: The group-fused graphical lasso. *Journal of Computational and Graphical Statistics*, 26(3):623–634, 2017.

[64] A. J. Gibberd and S. Roy. Multiple changepoint estimation in high-dimensional gaussian graphical models. *arXiv preprint arXiv:1712.05786*, 2017.

[65] E. Giné and A. Guillou. Rates of strong uniform consistency for multivariate kernel density estimators. In *Annales de l'Institut Henri Poincare (B) Probability and Statistics*, volume 38, pages 907–921. Elsevier, 2002.

[66] S. Goel, D. M. Kane, and A. R. Klivans. Learning ising models with independent failures. In *Conference on Learning Theory*, pages 1449–1469, 2019.

[67] N. Goix. How to evaluate the quality of unsupervised anomaly detection algorithms? *arXiv preprint arXiv:1607.01152*, 2016.

[68] N. Goix, A. Sabourin, and S. Clémençon. On anomaly ranking and excess-mass curves. In *Artificial Intelligence and Statistics*, pages 287–295, 2015.

[69] M. Gomez-Rodriguez, L. Song, H. Daneshmand, and B. Schölkopf. Estimating diffusion networks: Recovery conditions, sample complexity & soft-thresholding algorithm. *Journal of Machine Learning Research*, 17:3092–3120, 2016.

[70] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[71] A. G. Gray and A. W. Moore. Nonparametric density estimation: Toward computational tractability. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 203–211. SIAM, 2003.

[72] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier detection for temporal data: A survey. *IEEE Trans. on Knowledge and Data Engineering*, 26(9):2250–2267, 2014.

[73] D. Hallac, Y. Park, S. Boyd, and J. Leskovec. Network inference via the time-varying graphical lasso. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 205–213. ACM, 2017.

[74] Frank R Hampel. The influence curve and its role in robust estimation. *Journal of the american statistical association*, 69(346):383–393, 1974.

[75] Z. Harchaoui and C. Lévy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492):1480–1493, 2010.

[76] Z. Harchaoui, F. Vallet, A. Lung-Yut-Fong, and O. Cappé. A regularized kernel-based approach to unsupervised audio segmentation. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1665–1668. IEEE, 2009.

[77] N. Heard, D. Weston, K. Platanioti, and D. Hand. Bayesian anomaly detection methods for social networks. *The Annals of Applied Statistics*, 4(2):645–662, 2010.

[78] M. Hecker, S. Lambeck, S. Toepfer, E. Van Someren, and R. Guthke. Gene regulatory network inference: data integration in dynamic models — A review. *Biosystems*, 96 (1):86–103, 2009.

[79] H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods. *J. of Machine Learning Research*, 10(Apr):883–906, 2009.

[80] Y. Hong. On computing the distribution function for the poisson binomial distribution. *Computational Statistics & Data Analysis*, 59:41–51, 2013.

[81] A. Hoorfar and M. Hassani. Approximation of the lambert w function and hyperpower function. *Research report collection*, 10(2), 2007.

[82] L. Huang, X. Nguyen, M. Garofalakis, M. I. Jordan, A. Joseph, and N. Taft. In-network PCA and anomaly detection. In *Advances in Neural Information Processing Systems*, pages 617–624, 2007.

[83] P. J. Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[84] P. Humbert, B. Le Bars, L. Oudre, A. Kalogeratos, and N. Vayatis. Learning laplacian matrix from graph signals with sparse spectral representation.

[85] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[86] M. R. Jerrum, L. G. Valiant, and V. V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[87] T. Ji, D. Yang, and J. Gao. Incremental local evolutionary outlier detection for dynamic social networks. In *Proc. of the Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, pages 1–15. Springer, 2013.

[88] H. Jiang. Uniform convergence rates for kernel density estimation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1694–1703. JMLR. org, 2017.

[89] V. Kalofolias. How to learn a graph from smooth signals. In *Proceedings of the Conference on Artificial Intelligence and Statistics*, pages 920–929, 2016.

[90] H. Keshavarz, G. Michailidis, and Y. Atchade. Sequential change-point detection in high-dimensional gaussian graphical models. *arXiv preprint arXiv:1806.07870*, 2018.

[91] J. Kim and C. Scott. Robust kernel density estimation. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3381–3384. IEEE, 2008.

[92] J. Kim and C. Scott. On the robustness of kernel density m-estimators. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 697–704. Citeseer, 2011.

[93] J. Kim and C. D. Scott. Robust kernel density estimation. *Journal of Machine Learning Research*, 13(Sep):2529–2565, 2012.

[94] H. Kitano. Computational systems biology. *Nature*, 420(6912):206–210, 2002.

[95] M. Kolar and E. P. Xing. Estimating networks with jumps. *Electronic journal of statistics*, 6:2069, 2012.

[96] M. Kolar, L. Song, A. Ahmed, and E. P. Xing. Estimating time-varying networks. *The Annals of Applied Statistics*, 4(1):94–123, 2010.

[97] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[98] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[99] M. Latapy, T. Viard, and C. Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1):61, 2018.

[100] S. L. Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[101] M. Lavielle and E. Moulines. Least-squares estimation of an unknown number of shifts in a time series. *Journal of time series analysis*, 21(1):33–59, 2000.

[102] B. Le Bars and A. Kalogeratos. A probabilistic framework to node-level anomaly detection in communication networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2188–2196. IEEE, 2019.

[103] B. Le Bars, P. Humbert, L. Oudre, and A. Kalogeratos. Learning laplacian matrix from bandlimited graph signals. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2937–2941, 2019.

[104] B. Le Bars, P. Humbert, Argyris K., and N. Vayatis. Learning the piece-wise constant graph structure of a varying ising model. In *Proceedings of the* 37^{th} *International Conference on Machine Learning (ICML)*, 2020.

[105] G. Lecué and M. Lerasle. Learning from mom's principles: Le cam's approach. *Stochastic Processes and their applications*, 129(11):4385–4410, 2019.

[106] G. Lecué, M. Lerasle, and T. Mathieu. Robust classification via mom minimization. *Machine Learning*, 2020.

[107] M. Lerasle. Lecture notes: Selected topics on robust statistical learning theory. *arXiv preprint arXiv:1908.10761*, 2019.

[108] C. Lévy-Leduc and F. Roueff. Detection and localization of change-points in high-dimensional network traffic data. *The Annals of Applied Statistics*, 3(2):637–662, 2009.

[109] J. B. Lewis, K. Poole, H. Rosenthal, A. Boche, A. Rudkin, and L. Sonnet. Voteview: Congressional roll-call votes database. *https://voteview.com/*, 2020.

[110] F. Liese and I. Vajda. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412, 2006.

[111] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[112] H. Liu and C. Gao. Density estimation with contamination: minimax rates and theory of adaptation. *Electronic Journal of Statistics*, 13(2):3613–3653, 2019.

[113] M. Londschien, S. Kovács, and P. Bühlmann. Change point detection for graphical models in presence of missing values. *arXiv preprint arXiv:1907.05409*, 2019.

[114] P.-E. Maingé. Strong convergence of projected subgradient methods for nonsmooth and nonstrictly convex minimization. *Set-Valued Analysis*, 16(7-8):899–912, 2008.

[115] D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, A. Aderhold, R. Bonneau, and Y. Chen. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796, 2012.

[116] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro. Sampling of graph signals with successive local aggregations. *IEEE Transactions on Signal Processing*, 64(7):1832–1843, 2016.

[117] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, 34(3):1436–1462, 2006.

[118] G. Meyer. *Geometric optimization algorithms for linear regression on fixed-rank matrices*. PhD thesis, 2011.

[119] S. Minsker. Geometric median and robust estimation in banach spaces. *Bernoulli*, 21 (4):2308–2335, 2015.

[120] S. K. Narang, A. Gadde, and A. Ortega. Signal processing techniques for interpolation in graph structured data. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5445–5449, 2013.

[121] J. Neil, C. Hash, A. Brugh, M. Fisk, and C. Storlie. Scan statistics for the online detection of locally anomalous subgraphs. *Technometrics*, 55(4):403–414, 2013.

[122] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley Interscience, New-York, 1983.

[123] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2001.

[124] A. Ortega, P. Frossard, J. Kovačević, J. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5): 808–828, 2018.

[125] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[126] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat. Characterization and inference of graph diffusion processes from observations of stationary signals. *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

[127] A. Patcha and J. M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, 51(12):3448–3470, 2007.

[128] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. of Machine Learning Research*, 12:2825–2830, 2011.

[129] L. Peel and A. Clauset. Detecting change points in the large-scale structure of evolving networks. In *Proc. of the AAAI Conf. on Artificial Intelligence*, volume 15, pages 1–11, 2015.

[130] C. E. Priebe, J. M. Conroy, D. J. Marchette, and Y. Park. Scan statistics on Enron graphs. *Computational & Mathematical Organization Theory*, 11(3):229–247, 2005.

[131] S. Ranshous, S. Shen, D. Koutra, S. Harenberg, C. Faloutsos, and N. Samatova. Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3):223–247, 2015.

[132] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional ising model selection using l1-regularized logistic regression. *The Annals of Statistics*, 38(3): 1287–1319, 2010.

[133] S. Ravishankar and Y. Bresler. Learning sparsifying transforms. *IEEE Transactions on Signal Processing*, 61(5):1072–1086, 2012.

[134] J. Reeves, J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of applied meteorology and climatology*, 46(6):900–915, 2007.

[135] Z. Ren, T. Sun, C.-H. Zhang, and H. H. Zhou. Asymptotic normality and optimalities in estimation of large gaussian graphical models. *The Annals of Statistics*, 43(3): 991–1026, 2015.

[136] P. Rigollet and R. Vert. Optimal rates for plug-in estimators of density level sets. *Bernoulli*, 15(4):1154–1178, 2009.

[137] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proc. of the Int. Conf. on Machine Learning*, pages 561–568, 2011.

[138] S. Roy, Y. Atchadé, and G. Michailidis. Change point estimation in high dimensional markov random-field models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1187–1206, 2017.

[139] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo. Graph topology inference based on sparsifying transform learning. *IEEE Transactions on Signal Processing*, 67(7): 1712–1727, 2019.

[140] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[141] C. D. Scott and R. D. Nowak. Learning minimum volume sets. *J. of Machine Learning Research*, 7(Apr):665–704, 2006.

[142] D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[143] N. Segev, M. Harel, S. Mannor, K. Crammer, and R. El-Yaniv. Learn on source, refine on target: A model transfer learning framework with random forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1811–1824, Sep. 2017.

[144] A. Sen and M. S. Srivastava. On tests for detecting change in mean. *The Annals of statistics*, pages 98–108, 1975.

[145] B. Sen, N. C Borle, R. Greiner, and M. R. G. Brown. A general prediction model for the detection of ADHD and autism using structural and functional MRI. *PloS one*, 13 (4):e0194856, 2018.

[146] U. Shalit and G. Chechik. Coordinate-descent for learning orthogonal matrices through givens rotations. In *International Conference on Machine Learning*, pages 548–556, 2014.

[147] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *Signal Processing Magazine*, 30(3): 83–98, 2013.

[148] B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

[149] S. M. Smith, P. T. Fox, K. L. Miller, D. C. Glahn, P. M. Fox, C. E. Mackay, N. Filippini, K. E. Watkins, R. Toro, and A. R. Laird. Correspondence of the brain's functional architecture during activation and rest. *Proceedings of the National Academy of Sciences*, 106(31):13040–13045, 2009.

[150] B. Sriperumbudur and I. Steinwart. Consistency and rates for clustering with dbscan. In *Artificial Intelligence and Statistics*, pages 1090–1098, 2012.

[151] Y. Sun, A. Wong, and M. S. Kamel. Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, 23(04):687–719, 2009.

[152] W. Tang and F. Tang. The poisson binomial distribution–old & new. *arXiv preprint arXiv:1908.10024*, 2019.

[153] D. A. Tarzanagh and G. Michailidis. Estimation of graphical models through structured norm minimization. *Journal of Machine Learning Research*, 18, 2018.

[154] D. Thanou, X. Dong, D. Kressner, and P. Frossard. Learning heat diffusion graphs. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):484–499, 2017.

[155] M. Thottan, G. Liu, and C. Ji. Anomaly detection approaches for communication networks. In *Algorithms for next generation networks*, pages 239–261. Springer, 2010.

[156] J. Townsend, N. Koep, and S. Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17:1–5, 2016.

[157] C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. *Signal Processing*, page 107299, 2019.

[158] A. B. Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

[159] Berwin A Turlach. Bandwidth selection in kernel density estimation: A review. In *CORE and Institut de Statistique*. Citeseer, 1993.

[160] D. Valsesia, G. Fracastoro, and E. Magli. Sampling of graph signals via randomized local aggregations. *preprint arXiv:1804.06182*, 2018.

[161] L. Vandenberghe. The CVXOPT linear and quadratic cone program solvers. 2010.

[162] R. Vandermeulen and C. Scott. Consistency of robust kernel density estimators. In *Conference on Learning Theory*, pages 568–591, 2013.

[163] R. A. Vandermeulen and C. Scott. Robust kernel density estimation by scaling and projection in hilbert space. In *Advances in Neural Information Processing Systems*, pages 433–441, 2014.

[164] G. Varoquaux, A. Gramfort, F. Pedregosa, V. Michel, and B. Thirion. Multi-subject dictionary learning to segment an atlas of brain spontaneous activity. In *Proceedings of the Biennial International Conference on Information Processing in Medical Imaging*, pages 562–573. Springer, 2011.

[165] R. Vert and J.-P. Vert. Consistency and convergence rates of one-class svms and related algorithms. *Journal of Machine Learning Research*, 7(May):817–854, 2006.

[166] M. Vuffray, S. Misra, A. Lokhov, and M. Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2016.

[167] X. Wan, E. Milios, N. Kalyaniwalla, and J. Janssen. Link-based event detection in email communication networks. In *Proc. of the ACM Symp. on Applied Computing*, pages 1506–1510, 2009.

[168] B. Wang and Y. Qi. Fast and scalable learning of sparse changes in high-dimensional gaussian graphical model structure. In *International Conference on Artificial Intelligence and Statistics*, pages 1691–1700, 2018.

[169] D. Wang, X. Lu, and A. Rinaldo. Dbscan: Optimal rates for density-based cluster estimation. *Journal of Machine Learning Research*, 20(170):1–50, 2019.

[170] H. Wang, M. Tang, Y. R. Park, and C. Priebe. Locality statistics for anomaly detection in time series of graphs. *IEEE Trans. on Signal Processing*, 62(3):703–717, 2014.

[171] J. Wang and M. Kolar. Inference for high-dimensional exponential family graphical models. In *Artificial Intelligence and Statistics*, pages 1042–1050, 2016.

[172] Z. Wang and D. W. Scott. Nonparametric density estimation for high-dimensional data—algorithms and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 11(4):e1461, 2019.

[173] L. H. William, Y. Rex, and J. Leskovec. Representation learning on graphs: Methods and applications. *preprint arXiv:1709.05584*, 2017.

[174] M. Xie, S. Han, B. Tian, and S. Parvin. Anomaly detection in wireless sensor networks: A survey. *Journal of Network and computer Applications*, 34(4):1302–1325, 2011.

[175] L. Xue, H. Zou, and T. Cai. Nonconcave penalized composite conditional likelihood estimation of sparse ising models. *The Annals of Statistics*, 40(3):1403–1429, 2012.

[176] E. Yang, P. Ravikumar, G. I. Allen, and Z. Liu. Graphical models via univariate exponential family distributions. *Journal of Machine Learning Research*, 16:3813–3847, 2015.

[177] J. Yang and J. Peng. Estimating time-varying graphical models. *Journal of Computational and Graphical Statistics*, pages 1–12, 2019.

[178] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68 (1):49–67, 2006.

[179] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[180] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

université PARIS-SACLAY

ÉCOLE DOCTORALE
de mathématiques
Hadamard (EDMH)

Titre: Détection d'événements et inférence de structure pour des vecteurs sur graphes

Mots clés: Graphes, Réseaux, Détection d'anomalies et de ruptures, Champs aléatoire de Markov

Résumé: Cette thèse aborde différents problèmes autour de l'analyse et la modélisation de signaux sur graphes, autrement dit des données vectorielles observées sur des graphes. Nous nous intéressons en particulier à deux tâches spécifique. La première est le problème de détection d'événements, c'est-à-dire la détection d'anomalies ou de ruptures, dans un ensemble de vecteurs sur graphes. La seconde tâche consiste en l'inférence de la structure de graphe sous-jacente aux vecteurs contenus dans un ensemble de données.

Dans un premier temps notre travail est orienté vers l'application. Nous proposons une méthode pour détecter des pannes ou des défaillances d'antenne dans un réseau de télécommunication. La méthodologie proposée est conçue pour être efficace pour des réseaux de communication au sens large et tient implicitement compte de la structure sous-jacente des données.

Dans un deuxième temps, une nouvelle méthode d'inférence de graphes dans le cadre du Graph Signal Processing est étudiée. Dans ce problème, des notions de régularité local et global, par rapport au graphe sous-jacent, sont imposées aux vecteurs.

Enfin, nous proposons de combiner la tâche d'apprentissage des graphes avec le problème de détection de ruptures. Cette fois, un cadre probabiliste est considéré pour modéliser les vecteurs, supposés ainsi être distribués selon un certain champ aléatoire de Markov. Dans notre modélisation, le graphe sous-jacent aux données peut changer dans le temps et un point de rupture est détecté chaque fois qu'il change de manière significative.

Title: Event detection and structure inference for graph vectors

Keywords: Graphs, Networks, Anomaly and change-point detection, Markov Random Fields

Abstract: This thesis addresses different problems around the analysis and the modeling of graph signals i.e. vector data that are observed over graphs. In particular, we are interested in two tasks. The first one is the problem of event detection, i.e. anomaly or change-point detection, in a set of graph vectors. The second task concerns the inference of the graph structure underlying the observed graph vectors contained in a data set.

At first, our work takes an application-oriented aspect in which we propose a method for detecting antenna failures or breakdowns in a telecommunication network. The proposed approach is designed to be effective for communication networks in a broad sense and it implicitly takes into account the underlying graph structure of the data.

In a second time, a new method for graph structure inference within the framework of Graph Signal Processing is investigated. In this problem, notions of both local and global smoothness, with respect to the underlying graph, are imposed to the vectors.

Finally, we propose to combine the graph learning task with the change-point detection problem. This time, a probabilistic framework is considered to model the vectors, assumed to be distributed from a specific Markov Random Field. In the considered modeling, the graph underlying the data is allowed to evolve in time and a change-point is actually detected whenever this graph changes significantly.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France