Multivariate analysis with tensors and graphs – application to neuroscience

Pierre Humbert

université
PARIS-SACLAY

Multivariate analysis with tensors and graphs – application to neuroscience

Thèse de doctorat de l'Université Paris-Saclay

École Doctorale de Mathématique Hadamard (EDMH) n° 574
Spécialité de doctorat : Mathématiques appliquées
Unité de recherche : ENS Paris-Saclay, Centre Borelli, CNRS, Université Paris-Saclay, 91190, Gif-sur-Yvette, France
Référent : École Normale Supérieure Paris-Saclay

Thèse présentée et soutenue à Paris, le 22 Janvier 2021, par

Pierre HUMBERT

Composition du jury :

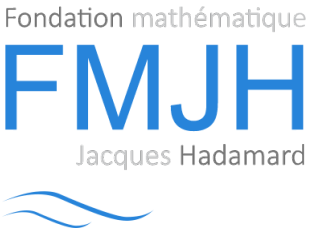
Stéphanie Allassonière Professeur, Université Paris-Descartes & École Polytechnique	Examinatrice
Alexandre Gramfort Directeur de recherche, INRIA, Université Paris-Saclay	Examinateur
Rémi Gribonval Directeur de recherche, INRIA, École Normale Supérieure de Lyon	Président du jury
Cédric Richard Professeur, Université Côte d'Azur	Rapporteur & Examinateur
Dimitri Van De Ville Professeur associé, École Polytechnique Fédérale de Lausanne	Rapporteur & Examinateur
Nicolas Vayatis Professeur, École Normale Supérieure Paris-Saclay	Directeur de thèse
Laurent Oudre Professeur, École Normale Supérieure Paris-Saclay	Co-directeur de thèse
Julien Audiffren Chercheur, Université de Fribourg	Co-encadrant de thèse

Thèse de doctorat

NNT: 2021UPASM019


école
normale
supérieure
paris–saclay


CENTRE
BORELLI


Fondation mathématique
FMJH
Jacques Hadamard


LABEX
Mathématique
Hadamard.

Contents

Remerciements

Mes premiers remerciements vont bien évidemment à mes directeurs de thèse. Nicolas, merci de m'avoir tout de suite fait confiance et de m'avoir permis d'effectuer ma thèse dans un cadre privilégié. Laurent, merci pour ton expertise, ton soutien indéfectible et ton optimisme permanent, il en a fallu à certains moments. Enfin, Julien, merci d'avoir toujours répondu présent quand j'en avais besoin. C'est avec toi que j'ai publié mon premier article, ça ne s'oublie pas. Merci à vous trois d'avoir fait de ces années de thèse une période si enrichissante.

Mes remerciements vont également à Cédric Richard et Dimitri Van De Ville qui ont bien voulu être rapporteurs de ma thèse. Vos remarques, questions et corrections ont été une réelle source d'amélioration du manuscrit. Merci également à Stéphanie Allassonnière, Alexandre Gramfort et Rémi Gribonval d'avoir accepté d'être examinateurs. Vos très nombreuses questions pendant la soutenance m'ont montré votre intérêt pour mon travail et je vous en remercie.

J'ai maintenant une pensée pour tous ceux avec qui j'ai eu la chance d'interagir de près ou de loin. Je pense notamment à l'équipe du Centre Borelli, Christophe, Véronique, Virginie, Alina, Mathilde, Myrto, Alice, Brian, Étienne, Théo, Amir, Firas, Matthieu, Ioannis, Argyris et aux anciens du CMLA, Thomas M., Charles, Cédric, Juan, Rémi et Émile. J'en oublie très certainement.
J'ai bien entendu une pensée toute particulière pour le groupe des Saint-Pères, Thomas D., Ludo, Mounir, Marie, et Antoine sans qui ces années n'auraient pas été si agréables. Je ne peux pas terminer sans évoquer Clément et Batiste. Merci Clément, de m'avoir accueilli à bras ouverts à l'hôpital. J'ai beaucoup appris à tes côtés et nos échanges sont et seront toujours un plaisir. Et enfin, un grand merci à Batiste, sans qui cette thèse n'aurait pas eu la même saveur. On s'appelle demain, après-demain et encore après de toute façon.

Merci à tous les copains de Paris, de Nantes et d'ailleurs. Ils se reconnaîtront.

Merci à ma famille pour son soutien et son affection. Je pense en particulier à ma mère, qui a toujours été là pour moi, mon père, qui m'a transmis son intérêt des sciences, et ma belle-mère, qui a toujours cru en moi. Merci à mes deux sœurs pour tous les bons moments passés ensemble. Merci à vous de m'avoir donné toutes les chances de réussir et surtout de m'avoir aidé à devenir qui je suis.

Merci Zahra de m'avoir soutenu et accompagné tout au long de ces années. Tu m'accompagneras assurément dans les suivantes.

Abstract

How to extract knowledge from multivariate data has emerged as a fundamental question in recent years. Indeed, their increasing availability has highlighted the limitations of standard models and the need to move towards more versatile methods. The main objective of this thesis is to provide methods and algorithms taking into account the structure of multivariate signals. Well-known examples of such signals are images, stereo audio signals, and multichannel ElectroEncephaloGraphy (EEG) signals. Among the existing approaches, we specifically focus on those based on graph or tensor-induced structure which have already attracted increasing attention because of their ability to better exploit the multivariate aspect of data and their underlying structure. Although this thesis takes the study of patients under general anesthesia as a privileged applicative context, methods developed are also adapted to a wide range of multivariate structured data.

The first contribution is the construction and deployment of a complete protocol and recording chain that has enabled us to build a large database of patients under routine general anesthesia. This database contains 88 patients in which 32 EEG signals and physiological variables are recorded synchronously from the moment they enter the operating room up to three hours after the surgery has been completed.

The second contribution consists in elaborating an optimization problem to learn a graph from a set of signals. These signals are assumed to be smooth and to admit a sparse representation in the spectral domain of the same underlying graph. This last property borrowed from graph signal processing is known to carry information related to the cluster structure of this graph. We solve this problem by introducing two algorithms. They are tested on multiple synthetic and real data, including EEG signals recorded during anesthesia.

The third contribution is the inclusion of tensor-induced structures in convolutional dictionary learning methods. More precisely, we add to the initial minimization problem a tensor low-rank constraint for each activation. By taking into account the multivariate structure of signals, the induced low-rank structure brings two major advantages. First, in multiple application contexts the multivariate activations are naturally low-rank. Second, low-rank constraints entail a better robustness with respect to noise, one of the main weaknesses of the activation learning part of the convolutional dictionary learning. Two algorithms are introduced to solve this problem. They are performed on both synthetic and real experiments, from images to EEG signals.

The fourth contribution is based on graph product, an operation built around the two previous structures i.e. graphs and tensors. With this formalism, we provide a simple way to identify the frequency support of multivariate graph signals, a useful information for subsampling, and compression. In addition, we introduce a method to assess the relevance of the graphs chosen a priori. The proposed algorithm is used on a time-feature-space representation of multichannel EEG signals with each dimension encoded by a specific graph.

Finally, the fifth, and last, contribution is more prospective. It consists in a decision support algorithm based on a predictive state representation model which assists anesthesiologists in administering anesthetics during a general anesthesia. In the objective of proposing a practical and comprehensive tool, the model only relied on the four most commonly monitored variables. Performances of the resulting agent are analyzed with divers metrics and through its confrontation to real anesthesiologists.

Notations

General

$\mathbb{1}_{\mathcal{A}}(\cdot)$	Indicator function over the set $\mathcal{A}$
$\langle \cdot, \cdot \rangle$	Inner product
$\star$	Convolution
$\circledast$	Circular convolution

Matrix and vector

$\boldsymbol{x}^\mathsf{T}$, $\boldsymbol{M}^\mathsf{T}$	Transpose of vector $\boldsymbol{x}$, matrix $\boldsymbol{M}$
$\boldsymbol{M}^H$	Hermitian transpose of $\boldsymbol{M}$
$\boldsymbol{M}_{i,j}$, $\boldsymbol{M}_{i,:}$ and $\boldsymbol{M}_{:,j}$	(i,j)-entry, i-th row, and j-th column of a matrix $\boldsymbol{M}$
$\mathrm{tr}(\cdot)$	Trace operator
$\mathrm{diag}(\boldsymbol{x})$	Diagonal matrix containing the vector $\boldsymbol{x}$
$\|\boldsymbol{x}\|_0$	Number of non-zero elements of a vector $\boldsymbol{x}$
$\|\cdot\|_F$, $\|\cdot\|_{2,0}$, $\|\cdot\|_{2,1}$	Frobenius norm, $\ell_{2,0}$-norm, and $\ell_{2,1}$-norm
$\mathbf{0}_N$, $\mathbf{1}_N$	Vector of size N with entries equal to zero or one
$\boldsymbol{I}_N$	Identity matrix in $\mathbb{R}^{N\times N}$

Graph

$G = (\mathcal{V}, \mathcal{E})$	Graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$
N	Number of nodes, i.e. $\mathrm{card}(\mathcal{V})$
$\boldsymbol{W}$, $\boldsymbol{L}$	Weights matrix, combinatorial Laplacian matrix
$\boldsymbol{X}$, $\boldsymbol{\Lambda}$	Matrices with eigenvectors and eigenvalues of $\boldsymbol{L}$
$\diamond$	Graph product

Tensor

p	Order of a tensor
$\boldsymbol{\mathcal{X}}_{i_1,i_2,\ldots,i_p}$	$(i_1, i_2, \ldots, i_p)$-entry of a tensor
$\circ \quad \otimes \quad * \quad \odot$	Outer, Kronecker, Hadamard, and Khatri–Rao product
$[\![\cdot]\!]$	Kruskal operator
$\times_m$	Mode-m product
$\boldsymbol{X}^{(q)}$	q-mode matricization of $\boldsymbol{\mathcal{X}}$
$\mathrm{vec}(\cdot)$	Vectorizarion
$\underset{\leftarrow p}{\odot_{i=1}}$	p Khatri–Rao products in reverse order

Abbreviations

CDL	Convolutional Dictionary Learning
CE	Cross Entropy
CNMF	Convolutive Nonnegative Matrix Factorization
CP	Canonical Polyadic
CPD	Canonical Polyadic Decomposition
CSC	Convolutional Sparse Coding
DL	Dictionary Learning
DoA	Depth of Anesthesia
ECG/EKG	ElectroCardioGraphy
EEG	ElectroEncephaloGraphy
fMRI	functional Magnetic Resonance Imaging
GA	General Anesthesia
GEV	Global Explained Variance
GFT	Graph Fourier Transform
GSP	Graph Signal Processing
HD	Hamming Distance
HMM	Hidden Markov Model
HR	Heart Rate
ICA	Independent Component Analysis
JFT	Joint Fourier Transform
LoC	Loss of Consciousness
MBP	Mean Blood Pressure
NMF	Nonnegative Matrix Factorization
PCA	Principal Component Analysis
PSR	Predictive State Representation
RGG	Random Geometric Graph
RMSE	Root Mean Square Error
RoC	Recovery of Consciousness
ROI	Regions Of Interest
RR	Respiratory Rate
SHMM	Spectral Hidden Markov Model
STF	Space-Time-Frequency
SVD	Singular Value Decomposition
TF	Time-Frequency
TPSR	Transform Predictive State Representation

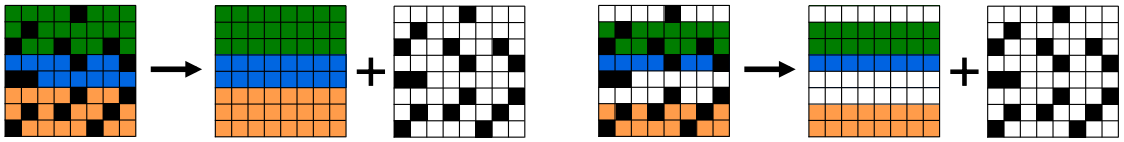
1 Introduction

Contents

1 Context of the thesis

General context. The human body is in a constant equilibrium state known as homeostasis. While this stability is fundamental, it needs a constant and precise regulation of vital organs by the brain. During a General Anesthesia (GA), a part of this stability is undermined by anesthetics. As a result, anesthesiologists must support some vital functions such as the respiratory system.

The objective of a tailored anesthesia is twofold: (i) to avoid excessively deep narcosis, associated with a higher risk of post-operative cognitive dysfunction and delayed awakening, (ii) to prevent under dosing, which is associated with a risk of memorization. To that end, anesthesiologists need to infer, in real-time, the level of consciousness of the patient, also referred to as the Depth of Anesthesia (DoA). Since recently, they can rely on a wide range of physiological variables monitored with a large number of sensors. This remarkable change in the medical field is allowed by the stunning progression of sensors and their systematic use. As a direct consequence, a large amount of signals and time-series is becoming available. Well-known examples of such signals are ElectroCardioGrams (ECG) signals, ElectroEncephaloGrams (EEG) signals, and all physiological variables. This change is particularly noticeable in clinical anesthesia where there was a very limited amount of data until recently. The main question now is how mathematics can help us to transit from all these multivariate raw signals to actionable data and



(a) Low-rank **(b)** Group sparse and low-rank

Figure 1.1: Illustration of (a) the low-rank assumption, (b) the combination of a low-rank and a group sparse assumptions.

knowledge. This is even more crucial, as this subject of interest at the crossroads of medical and mathematical disciplines may lead to important benefits in the treatments of patients but also in our understanding of human physiology.

Collaboration with the medical unit of the Centre Borelli. During this thesis, I collaborated with the medical unit of the Centre Borelli (ex Cognac-G). This center is a research team regrouping mathematicians (statisticians, machine learning specialists, etc.) and medical researchers, gathered around the quantification of human behavior. In particular, I worked in close collaboration with M.D. Clément Dubost, head of the intensive care unit at the "Hôpital d'Instruction des Armées Bégin". Together, we put a lot of effort into designing a complete protocol – from the recording chain to data analysis – in order to jointly propose useful mathematical methods to study patients under anesthesia. In the past, the Centre Borelli has already developed several experimental protocols for a wide range of clinical problems from human locomotion to infants' eye movements. The quantification of the phenomenon of interest has always been made through the analysis of physiological signals recorded with several sensors. The first objective being to extract the relevant information from these signals to understand the physiological mechanisms that produced them. The second objective being to automatize the quantification process in order to provide tools that can be used routinely by practitioners.

2 Motivations

2.1 From data to knowledge by leveraging multivariate structures

Rethinking medicine through its transition into the next-generation cannot be done without significant changes in the way we analyze medical data. Indeed, data sets from current investigations are often much larger, and more complicated, than those of earlier days. This phenomenon is partly due to the democratization of cheap, and easy-to-manipulate sensors, but is also explained by the current leading habit which is to collect any available data from patients in view of a better understanding of different physiological phenomenons. Consequently, multiple signals such as ECG, or EEG signals, are now recorded on an almost daily basis. Their wide diversity and their substantial volume inevitably necessitate vast improvements in data storage, manipulation, as well as advances in analytical methods. To efficiently analyze these data, several approaches were taken. At first, the trend was to emphasize *univariate* data analysis with models including a single output variable. In particular, they focused on how to integrate prior knowledge about data, either by making hypotheses on the class of models to restrict their complexity, or through constraints and regularizations. A classical example illustrating both forms is the ridge regression first proposed by Tikhonov [1963]. Here, a linear model is assumed in the features and a ℓ_2 regularization i.e. a *smoothness* assumption, is added to avoid too large parameters. Another important example is the lasso regression [Tibshirani, 1996] where a ℓ_1 regularization i.e. a

sparsity assumption, is added to induce only few non zero parameters. Other class of models also offer interesting alternatives to add prior knowledge on the structure of some signals. This is for example the case of shift-invariant, or convolutional representations [Garcia-Cardona and Wohlberg, 2018a], which treat a signal as a linear decomposition into few local atoms/patterns. They extract recurrent non-sinusoidal patterns and lead to the discovery of local structures in a set of non-stationary signals like time series, i.e. recordings with a temporal dimension [Lewicki and Sejnowski, 1999; Grosse et al., 2007].

While all these ideas have led to both theoretical and practical advances, there is an inevitable gap between what is being proposed for the univariate case and what we can expect from well-defined statistical models. Indeed, the output signals are often *multivariate* (also called multi-way [Escandar et al., 2014]), and the relations between their variables, or dimensions, must be considered if we want to analyze them adequately. To fill this gap, the statistical and machine learning communities –among others– have placed great emphasis on multivariate analysis through techniques that allow e.g. the presence of more than one output variable [Van Steen and Molenberghs; Hidalgo and Goodman, 2013]. The first natural step to go beyond the univariate case is to consider the bivariate case i.e. matrix-valued data. Many strategies have been proposed to incorporate relations between the different dimensions of such data, highlighting what a multivariate analysis can bring in term of performance and interpretability. Indeed, the bivariate case allows us to consider previously unavailable properties and structures. This is the case of the low-rank structure leveraged in multiple methods such as low-rank Principal Component Analysis (PCA) [Vidal et al., 2016], matrix recovery [Fazel, 2003; Rohde et al., 2011], and matrix completion [Candès and Recht, 2009; Koltchinskii et al., 2011; Negahban and Wainwright, 2011; Recht, 2011]. The combination of both low-rank and sparsity structures also appeared relevant in a number of models. Depending on the combination (see Figure 1.1), it gives rise to more robust and interpretable methods such as sparse PCA [Zou et al., 2006], subspace clustering [Vidal, 2011; Udell et al., 2016; Haeffele and Vidal, 2019], and sparse subspace clustering with outliers [Elhamifar and Vidal, 2013].

A multivariate analysis through graphs. Besides low-rank and sparsity, another promising way to leverage the structure of multivariate data is to use the notion of *graph* (or network). Indeed, the graph brings valuable knowledge on the process that generates the data (e.g. two linked nodes are highly correlated or have very close values) which make it useful in a large range of domains and applications spanning biology [Barabasi and Oltvai, 2004], neuroscience [Richiardi et al., 2013; Preti et al., 2017], clustering [Belkin and Niyogi, 2002; Von Luxburg, 2007], representation learning [William et al., 2017], multi-task learning [Chen et al., 2015a; Nassif et al., 2020], and others [Zhu, 2005; Kolaczyk and Csárdi, 2014]. Being able to build models or learning algorithms from these data, while considering their underlying graph structure, is therefore a major key component to improve performances. What remains is to find a way to incorporate prior information about the structure of signals with a graph. One possibility is to consider undirected probabilistic graphical models where a set of random variables is represented as different nodes of a graph [Koller and Friedman, 2009]. In this representation, an edge between two nodes indicates the conditional dependency between the two corresponding random variables, given the other ones. More recently, *Graph Signal Processing* (GSP) [Shuman et al., 2013; Ortega et al., 2018; Djuric and Richard, 2018], has also appeared to be a powerful alternative framework to extract valuable information from multivariate data. To take into account the structure of a signal, the idea is to consider it as defined on the nodes of a graph and to encode relationships between

its variables via the edges. In this formalism, the graph defines a support, and the signals, now called *graph signals*, are defined on this support. This allows to capture the structure on which a signal evolves, hence providing more information than considering the signal alone. Furthermore, by generalizing standard concepts of signal processing to signals recorded over graphs i.e. graph signals, GSP provides intuitive constraints for the modelization. For instance, the *smoothness* of observations with respect to the true underlying graph is one of the most common and natural assumption [Daitch et al., 2009; Egilmez et al., 2016; Kalofolias, 2016; Chepuri et al., 2017; Dong et al., 2019], which asks for signals to have small local variations among adjacent nodes. Indeed, this property is very natural and is therefore leveraged in a wide range of applications. One can cite multi-task estimation over graph [Nassif et al., 2020] where an underlying graph captures the link between multiple tasks allowing agents to cooperate with each other. This cooperation may be encouraged with a regularization that imposes a certain degree of smoothness between the different decision rules of each agent [Nassif et al., 2018]. Unfortunately, while in these methods the availability of a graph is a core assumption, e.g. in spectral clustering [Von Luxburg, 2007], semi-supervised learning [Zhu, 2005], etc., in most situations no natural graph can be derived or defined. One approach is therefore to infer it from a set of signals assumed to admit the same underlying graph. This task, often referred to as *graph learning* (or graph topology inference), has also received significant attention in various fields such as in statistic, signal processing, biology, and others [Friedman et al., 2008; Hecker et al., 2009; Lim et al., 2015; Moscu et al., 2020]. A review of recent methods for graph topology inference is given in [Dong et al., 2019].

A multivariate analysis through tensors. The inevitable extension of the bivariate case is the multivariate case. Similarly to the transition from the one to the second dimension, new possibilities and thus new strategies become available to leverage the structure of the multivariate data. To this end, a significant amount of works has been concentrated around tensor methods. This growing interest is mainly due to their ability to better exploit the multivariate aspect of the data. Indeed, in part spurred by pioneering works in psychometrics [Cattell, 1944], the list of applications of tensor methods with success encompasses problems in signal processing [Zhou et al., 2013; Cichocki et al., 2015], computer vision [Shashua and Hazan, 2005; Liu et al., 2012], spectral learning of latent variable models [Anandkumar et al., 2014; Janzamin et al., 2019], neuroscience [Beckmann and Smith, 2005; Miwakeichi et al., 2004; Mørup et al., 2006; Becker et al., 2015], etc. Thorough surveys of these techniques with their applications are given in Kolda and Bader [2009]; Grasedyck et al. [2013] and Sidiropoulos et al. [2017]. In this vast literature, one of the most widely used strategies is to directly apply *tensor decomposition* to the data. This often leads to more interpretable results and better performances. Indeed, by factorizing the data in a lower dimensional space, tensor decompositions introduce a compact basis which can describe the data in a concise manner. One important example of such decomposition is the Canonical Polyadic Decomposition (CPD) [Hitchcock, 1927], also known as Parafac or CANDECOMP [Harshman, 1970; Carroll and Chang, 1970], which expresses a tensor as a minimal sum of rank-one tensors. Other decompositions such as the Tucker decomposition [Tucker, 1963], or the higher-order singular value decomposition [De Lathauwer et al., 2000], have also proven to be efficient. For example, these decompositions have led to significant progresses in tensor completion that pertain to tensor recovery [Gandy et al., 2011; Liu et al., 2012; Goulart and Favier; Rauhut et al., 2017]. Another strategy is to include tensor-induced structures in existing methods through additional constraints and regularizations. In Zhou et al. [2013], authors proposed a family of tensor regression models where a *CP low-rank* constraint is added. They also extended these models to Tucker low-rank constraints [Li et al., 2018]. Others focused on multilinear

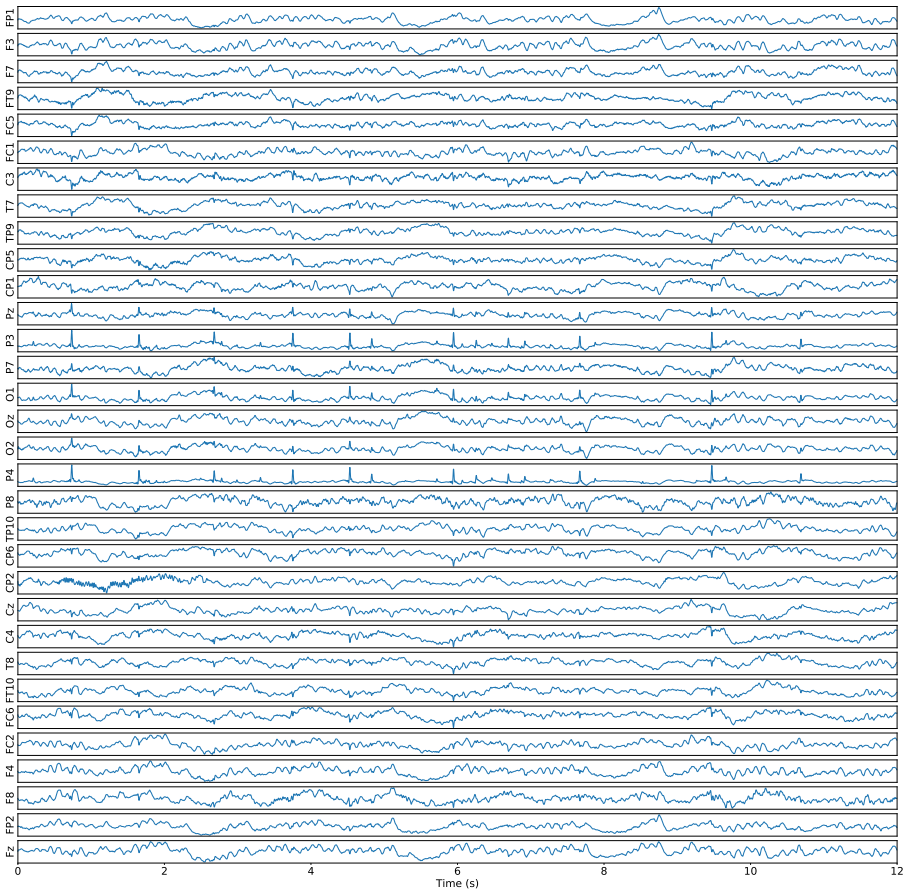


Figure 1.2: Illustration of a 32-channel EEG montage of one patient. On the y-axis of each signal is annotated the name of the corresponding channel.

rank constraints [Rabusseau and Kadri, 2016; Sun and Li, 2017], sparsity constraints on each rank-1 tensor of the CPD [He et al., 2018], etc. This idea of enforcing a particular structure with constraints is also used in several multivariate dictionary learning models [Hawe et al., 2013; Sironi et al., 2014; Dantas et al., 2018; Schwab et al., 2019] or to accelerate convolutional neural networks [Lebedev et al., 2015; Kim et al., 2016; Astrid and Lee, 2017]. Overall, while all these high-order models inevitably bring several difficulties due to the complexity of the manipulated objects, they have proven their usefulness in a wide range of fields showing, once again, the importance of considering the underlying structure of the data to obtain more efficient methods.

2.2 Analysis of consciousness during a general anesthesia

In its more practical aspect, this thesis was built around the necessity to analyze data recorded during a *General Anesthesia* (GA): a drug-induced, reversible condition that includes specific behavioral and physiological traits (unconsciousness, amnesia, analgesia, and akinesia) [Brown et al., 2010]. This unnatural condition is obtained through the use of different drugs (e.g. inhalational hypnotic anesthetics – sevoflurane – or intravenous anesthetics – propofol) which are all reinforcing the GABA inhibitory system in the brain. However, while GA is a cornerstone of modern medicine, and is crucial for the realization of many medical and surgical procedures [Purdon et al., 2013], it may carry some risks (e.g. cognitive dysfunction [Punjasawadwong et al., 2018],

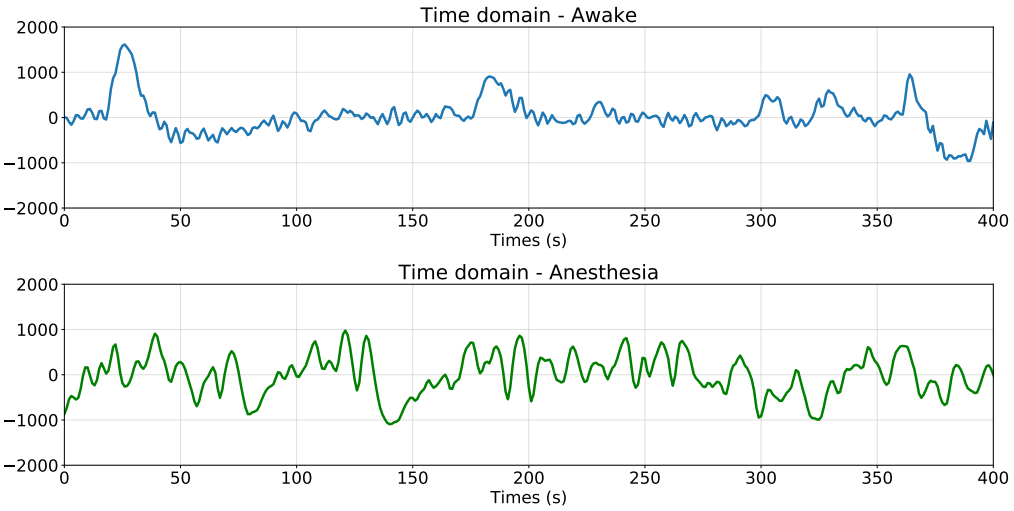


Figure 1.3: On the top, temporal signal in Awake state. On the bottom, temporal signal in Anesthesia state.

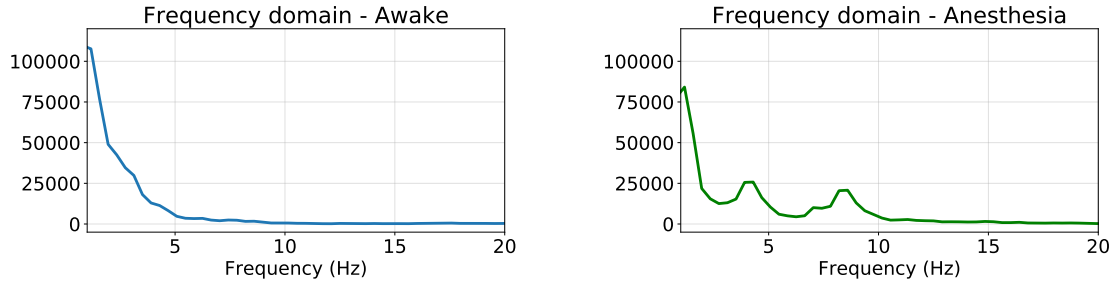


Figure 1.4: On the left, spectrum in Awake state. On the right, spectrum in Anesthesia state.

postoperative delirium [Fritz et al., 2016]). Consequently, a sustained and careful monitoring of the level of consciousness of the patient – also referred to as the Depth of Anesthesia (DoA) – is required. Although there is no consensual definition of the DoA, it has been defined by experts as "the probability of non-response to stimulation, calibrated against the strength of the stimulus, the difficulty of suppressing the response, and the drug-induced probability of non-responsiveness at defined effect site concentrations" [Shafer and Stanski, 2008]. Its precise knowledge is essential to allow accurate titration of the drugs administered. The major objectives are to avoid excessively deep narcosis, associated with a higher risk of post-operative cognitive dysfunction and delayed awakening, and to prevent underdosing, associated with a risk of memorization [Sebel et al., 2004].

The Dynamics of the Brain under Anesthesia. As a direct measurement of the main target of anesthetics i.e. the brain [Merry et al., 2010], ElectroEncephaloGraphy (EEG), which measure the scalp electrical potentials originating from neural currents in the brain, remain the gold-standard to assess the DoA (see Figure 1.2). Indeed, many of the changes that occur in the brain can be readily observed in the EEG [Tong and Thakor, 2009; Sanei and Chambers, 2013; Cohen, 2014]. In consequence, since the 2000's, they have been extensively used to study the phenomenons occurring during a GA [Purdon et al., 2015; Liu and Rinehart, 2016]. A wide range of research has thus showed that GA produces distinct patterns on the EEG which can be described

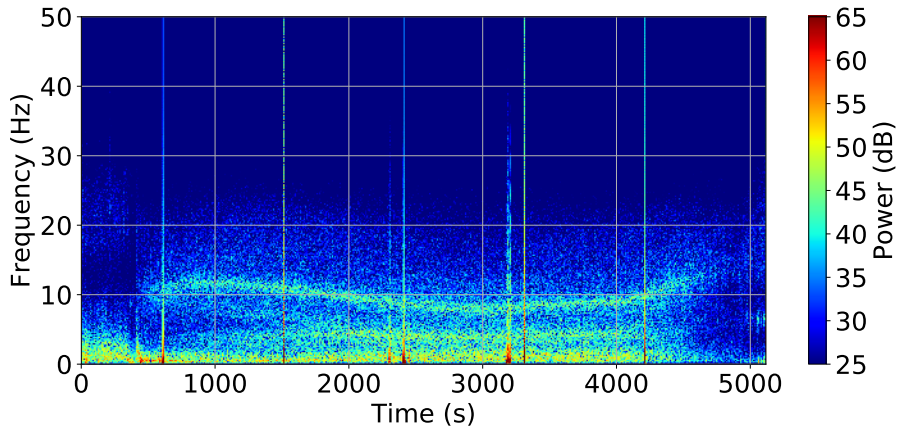


Figure 1.5: Spectrogram of one patient during a general anesthesia induced by propofol and sevoflurane.

in relation to five states in which they appear: Awake or induction, Loss of Consciousness (LoC), Anesthesia or maintenance, Recovery of Consciousness (RoC), and emergence. As the level of general anaesthesia deepens, the best known and most common pattern is a gradual increase in specific frequency bands and signal amplitude. Figure 1.3 illustrate this phenomenon by showing a frontal EEG channel of the same patient in Awake and Anesthesia states. We clearly see changes in the raw data with the apparition of small waves with large amplitudes. These visual changes, present in almost every patient, lead to a modification of the EEG spectrum i.e. the decomposition of the EEG signal into the power in its frequency components (see Figure 1.4). Actually, in an important study conducted by Purdon et al. [2013], researchers have shown that the power of α and δ-waves (respectively in the 8-13 Hz and 1-3Hz ranges) is a promising predictor of the different states of a patient during a GA only induced by propofol. Indeed, they showed that the power of these two ranges of frequencies tend to increase with the induction of the drug. Therefore, their tracking allows to define precisely which state a patient is more likely to be in. A typical evolution of the power of each frequency over time is displayed in Figure 1.5 through a spectrogram. They also find that these modifications at the level of a channel are combined with a spatial-modification called "anteriorization". More precisely, while in the Awake state α-waves are mostly present at the back of the head, with the induction of propofol, these waves start to slowly migrate to the forehead. This process is reversed when the amount of drugs decreases. With this example, we see the importance to go beyond an univariate analysis to fully describe and understand global mechanisms.

A routine clinical context. While these studies allow a better understanding of the GA, they are, in the major part, conducted in an ideal environment. In a clinical context, reality is quite different. First, anesthesiologists use, not one, but multiple drugs to induce the GA. Analysis becomes more difficult as each drug induces its own time-frequency patterns [Purdon et al., 2015]. Second, the analysis of EEG signals suffers from several limitations, especially when data are recorded during real surgeries. Indeed, even if there is no artifact due to muscle contractions (patients are curarized), EEG signals are still prone to low signal to noise ratio, impulsive noise due to sensor malfunctions, and artifacts caused by e.g. electro-surgical devices that are used to cut and cauterize tissue (see Figure 1.6) [Tong and Thakor, 2009]. Thus, it becomes very difficult to use standard methods which assume an ideal theoretical set-up. Third, the use of EEG is time consuming making it unusable for a daily-routine. As a global consequence, other methods, not

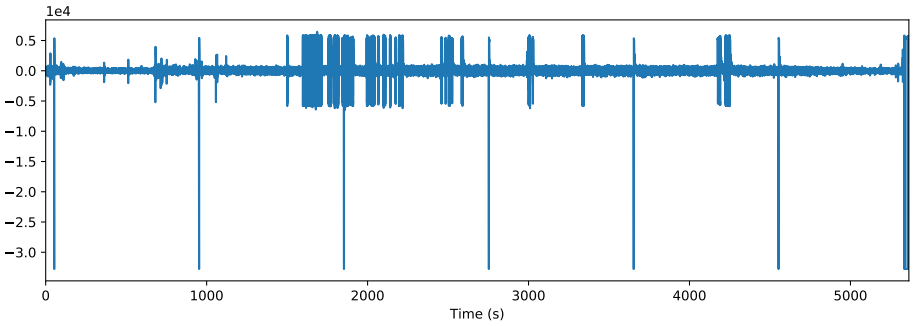


Figure 1.6: EEG recording (in μV) of a patient during anesthesia with a lot of noise (sampling frequency: 100 Hz).

necessarily based on EEG, must be investigated.

To pass through all these issues, during a surgery, several monitoring systems have been proposed for DoA assessment but they all have some limitations [Bruhn et al., 2006]. No point-of-care gold standard monitoring DoA prevails. The most used system is probably the BiSpectral Index (BIS) [Kissin, 2000; Avidan et al., 2008]. It provides a numerical value from 0 to 100 (from no cerebral activity to awake and responsive). However, being largely used, especially in the US, it has a lot of drawbacks such as high inter-individual variability [Whitlock et al., 2011], low performance with volatile anesthetics [George Mychaskiw et al., 2001], high latency and interferences with surgical knife, artifacts from movements or from forced air warming therapy [Hemmerling and Migneault, 2002]. Another index is the sample-entropy introduced by Richman and Moorman [2000]. It is a variant of the approximate entropy that gives information on the complexity of a time series such as EEG signal. In summary, although the EEG is the gold standard for the evaluation of the DoA, it requires additional sensors, it presents some limitations, and it is time consuming. That is why, in a routine clinical context, the best evaluation of the DoA is thought to be, most of the time, the one made by the anesthesiologist on the basis of the physiological variables of the patient.

Altogether, in practice, the ideal DoA monitor should be able to give an evaluation without EEG. Furthermore, while a neural analysis of GA is often centered around useful but old methods of analysis such as time-frequency representation, we believe that recent advances in statistics and machine learning could greatly contribute in a thinner understanding of the complex mechanisms occurring during GA.

3 Contributions

In the following, we detail the contributions of this thesis. To emphasize their versatility, each contribution is supported by a wide variety of experiments, including at least one that is related to GA. Furthermore, for each algorithm we provide an online open-source Python code.

3.1 A database of patients recorded during a general anesthesia

Made in collaboration with M.D. Clément Dubost, the first contribution of this thesis is the construction and deployment of a complete protocol and recording chain to build a large database of patients under routine GA on which we could work. To that end, helped by Brian Berthet-Delteil, Arno Benizri, and Gael de Rocquigny, we continuously recorded synchronously the

physiologic variables routinely monitored during anesthesia together with a 32 channels EEG. All these variables are listed in Table 1.1. Between February 2016 and May 2018, 88 subjects, all from "Hôpital d'Instruction des Armées Bégin, Saint-Mandé, France", have been included in the database. Note that, to the best of our knowledge, this is the first database of patients under routine GA where both multichannel EEGs and physiological variables are recorded synchronously from the moment they enter the operating room up to three hours after the end of surgery.

3.2 Graph learning on multivariate signals

In the second contribution, we consider the graph learning problem i.e. the problem of learning a graph from multivariate graph signals. As already explained, such signals are multivariate observations carrying measurements corresponding to the nodes of an unknown graph, which we desire to infer. The idea of this contribution actually comes from a simple observation. In general, we do not have a graph which is adapted to the signal of interest. One possible idea is thus to learn it. However, as this is an ill-posed problem, we must assume several properties on both signals and associated graph. In our approach, these properties take their inspiration from the field of Graph Signal Processing (GSP) [Shuman et al., 2013; Ortega et al., 2018]. This domain provides intuitive graph-induced structural constraints, and has already proven its success in many applications, especially in neuroscience with the analysis of the brain. Indeed, for instance Huang et al. [2018] show that by constructing a graph from structural connectivity and considering brain activity as graph signals, it is possible to capture relevant brain properties (e.g. cognitive features) with GSP concepts.

More specifically, we elaborate an optimization problem to learn the Laplacian of the underlying graph. To alleviate the ill-posed problem, the graph signals are assumed to behave smoothly with respect to the same underlying graph structure and to admit a sparse representation in the spectral domain of this graph. This last property, referred to as *bandlimitedness* in GSP, is known to carry information related to the cluster structure of the graph [Von Luxburg, 2007; Sardellitti et al., 2019]. The learned graph is therefore a good candidate in the initialization of spectral clustering methods. Note that these two properties are also core assumptions in a lot of methods treating e.g. graph sampling, or interpolation over graphs. To solve this graph learning problem, we propose two algorithms called IGL-3SR and FGL-3SR. Based on a 3-step alternating procedure, both algorithms rely on standard minimization methods – such as manifold gradient descent or linear programming – and have lower complexity compared to previous 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. To highlight the efficiency of our methods, we provide multiple examples ranging from meteorology to EEG analyses.

3.3 Tensor-based convolutional dictionary learning approach

The third contribution results from the combinations of two families of methods to analyze multivariate signals. The first family of methods is called Convolutional Dictionary Learning (CDL) [Wohlberg, 2015; Garcia-Cardona and Wohlberg, 2018a]. It consists in learning atoms – or patterns – which give a sparse approximation of signals. Hence, contrary to Fourier or wavelet bases, the atoms are not predefined, but are learned from the signal itself. This idea of providing a linear decomposition of a signal into few learned atoms, instead of predefined ones, has led to significant results in a wide range of topics, including image classification, image restoration, and signal processing (see [Wohlberg, 2015; Garcia-Cardona and Wohlberg, 2018a]

Variables	Units	abbreviation
EKG		
Electrocardiogram 1	μV	EKG
E-EEG Module		
Electroencephalography (32 channels)	μV	EEG
Basics Module		
Heart Rate	$/min$	HR
Systolic arterial blood pressure	$mmHg$	SBP
Diastolic arterial blood pressure	$mmHg$	DBP
Mean arterial blood pressure	$mmHg$	MBP
Saturated percentage of Dioxygen	$/100\%$	SpO_2
Temperature 1	℃	T1
Temperature 2	℃	T2
Heart rate from arterial line 1	$/min$	P1 HR
Invasive systolic arterial blood pressure 1	$mmHg$	P1 Sys
Invasive diastolic arterial blood pressure 1	$mmHg$	P1 Dia
Invasive mean arterial blood pressure 1	$mmHg$	P1 Mean
Heart rate from arterial line 2	$/min$	P2 HR
Invasive systolic arterial blood pressure 2	$mmHg$	P2 Sys
Invasive diastolic arterial blood pressure 2	$mmHg$	P2 Dia
Invasive mean arterial blood pressure 2	$mmHg$	P2 Mean
ST elevation on lead DII	mm	ST II
ST elevation on lead V5	mm	ST V5
ST elevation on lead aVL	mm	ST aVL
Gaz Analysis Module		
End tidal carbon dioxyde	$mmHg$	Et CO_2
Anesthesia Agent		AA
AA Expiratory Concentration	$/100\%$	AA ET
AA Inspiratory Concentration	$/100\%$	AA FI
Total Minimum Alveolar Concentration	$/100\%$	AA MAC SUM
Fraction inspired of dioxygen	$/m$	Fi O_2
Mean alveolar concentration	$/m$	MAC
Fraction inspired Nitrous Oxide	$/m$	Fi N_2
End tidal Nitrous Oxide	$/m$	Et N_2O
Respiratory Rate	$/min$	RR
BIS Module		
Bispectral Index		BIS
BIS Burst Suppression Ratio	$\%$	BIS BSR
BIS Electromyography	dB	BIS EMG
BIS Signal Quality Index	$\%$	BIS SQI

Table 1.1: Standard variables recorded during a surgery.

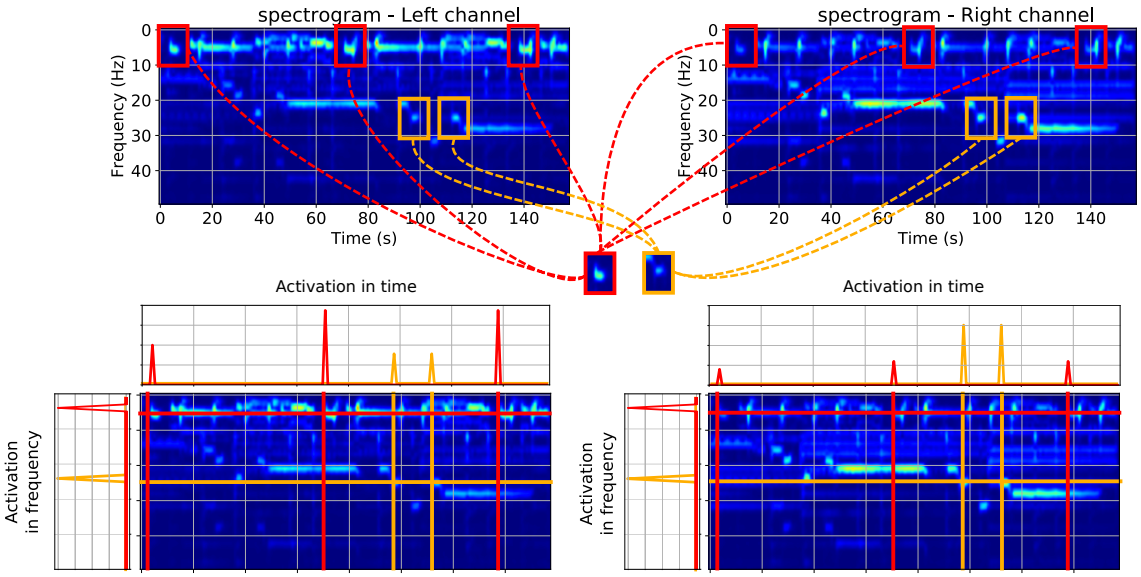


Figure 1.7: Two spectrograms obtained from a stereo music recording. Some repetitive patterns (highlighted in red and orange) are visible on the two spectrograms and suggest that a CDL model may appear as natural for such data. In addition, the low-rank structure of the data is here transferred into the activations tensors rather than into the observed patterns. In other words, although the time-frequency atoms may be complex (and thus without a low-rank structure), the activations (i.e. the time/frequency/channel positions where these atoms appear) clearly exhibit a low-rank structure.

and references therein). Nevertheless, while these methods exhibit interesting properties, they are mainly focused on resolution for univariate signals [Garcia-Cardona and Wohlberg, 2018b], and therefore do not fully take into account the possible interaction between the different modes of multivariate signals. Moreover, they are frequently vulnerable to noise and perturbations such as impulsive noise [Simon and Elad, 2019; Wang et al., 2020].

To take into account these drawbacks, we introduce a tensor CDL model where both activations and atoms are represented by tensors. More precisely, we propose to employ CDL approaches in combination with a second family of methods that include CP low-rank constraints in their modelization. By adding to the initial CDL problem a CP low-rank constraint for each activation, we constrain these activations to be sparse and low-rank. We therefore take into account the multivariate structure of the data and obtain accurate and interpretable results. Note that while the idea of enforcing low-rank constraints for CDL is not novel, it is mainly enforced on the dictionary and not on the activations. Nevertheless, we claim that constraining the activations to be low rank brings two majors advantages. First, in multiple application contexts the low-rank structure naturally appears in the activations rather than in the atoms/dictionary (see Figure 1.7). Second, low-rank constraints on activation entail a better robustness with respect to noise, which is one of the main weaknesses of the activation learning part of CDL [Simon and Elad, 2019]. Another motivation of this model comes from a large number of works which relies on a tensorial representation of multivariate time-series with great success (see e.g. the huge literature considering EEG signals [Miwakeichi et al., 2004; Mørup et al., 2006; De Vos et al., 2007; Becker et al., 2010, 2014, 2015; Dauwels et al., 2011; Mørup, 2011; Zhao et al., 2011; Mahyari et al., 2016]). In these works, signals are frequently analyzed by computing a short-time Fourier transform for each “channel”, resulting in a tensor of order 3 encoding a space-time-frequency representation.

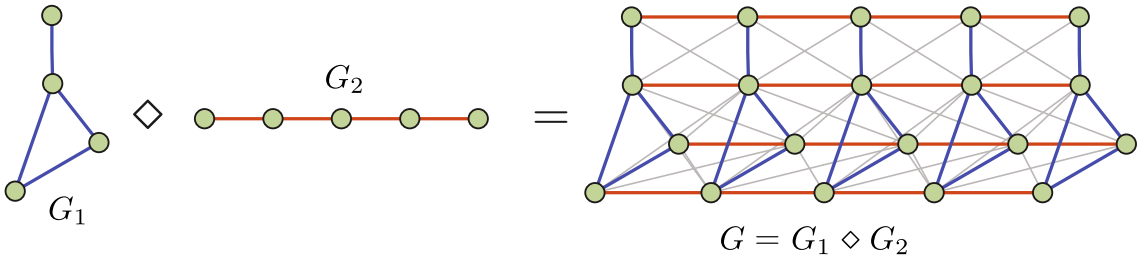


Figure 1.8: Illustration of the product graph G between two graphs G_1 and G_2. $\diamond$ represents either a Cartesian (only colored edges), a Kronecker (only gray edges) or a strong product (all edges) between these two graphs. Figure modified from the original one in [Ortiz-Jiménez et al., 2018].

The resulting tensor is then studied through the prism of the canonical polyadic decomposition to exploit the interactions among multiple modes. Here, while slightly different because we do not directly apply tensor decompositions to the data, coupling the CDL representation with a low-rank constraint also results in (local) representations that are (i) more robust to noise and (ii) easier to understand [Zhao et al., 2011; Zhou et al., 2013; Cong et al., 2015; Rabusseau and Kadri, 2016].

3.4 Graph Product for multivariate graph signals

In the fourth contribution, we propose a simple approach to identify the frequency support of multivariate *time-vertex graph signals* by combining graph and tensor methods. Such signals are related to the notion of *time-vertex* signal processing in GSP where both spatial and temporal interactions are modeled [Grassi et al., 2018]. Although this framework was initially introduced for matrix-value signals, in the multivariate case we need to extend it by considering relationships within any dimension (e.g. time, space, feature space). To this end, one graph per dimension is defined, and these structures are merged using a *graph product* [Imrich and Klavzar, 2000; Hammack et al., 2011; Leskovec et al., 2010; Sandryhaila and Moura, 2014]. An example is given in Figure 1.8. Interestingly, it appears that the resulting complex structure can be easily studied through the tensor formalism. Henceforth, to identify the frequency support of the multivariate graph signal, we first choose one graph per dimension a priori, and then, introduce an optimization problem including tensor-based regularizations adapted to a bandlimitedness assumption. These sparsity regularizations can be specified so as to work only on one dimension (i.e. selection of the best time samples, channels, or features). In addition, by comparing results obtained with the graphs chosen a priori against the ones from random graphs, we provide a simple way to assess their relevance. We apply our method to a tensorial representation of EEG signals highlighting its performance for sampling and compression. While this contribution is focused on time-vertex signals, the core idea can be applied on any multivariate graph signals.

3.5 Apprenticeship learning for a predictive state representation of anesthesia

In this fifth, and last, contribution, we propose a decision support algorithm which assists anesthesiologists in administering drugs in order to maintain an optimal DoA. Derived from a Transform Predictive State Representation algorithm (TPSR) [Littman and Sutton, 2002; Rosencrantz et al., 2004; Boots et al., 2011], our model learns by observing anesthesiologists in practice. This framework, known as apprenticeship learning [Abbeel and Ng, 2004], is particularly useful in the

medical field as it is not based on an exploratory process – a prohibited behavior in healthcare [Gottesman et al., 2018]. TPSR is one particularly powerful and flexible model class employed in the area of sequence prediction. The key insight in this class of models is that observed sequence data is often the manifestation of some underlying, or hidden, dynamics [Hamilton et al., 2014]. By modeling the transition structure between different hidden states and the probabilities governing the emission of observations from these hidden states, a succinct and powerful predictive model can be obtained. Notice that, while the previous contributions are mostly related to EEG analyses, here, to provide a very practical tool for the anesthesiologists we only rely on the four commonly monitored variables during surgery: Heart Rate (HR), Mean Blood Pressure (MBP), Respiratory Rate (RR), and the concentration of anesthetic drug (AAFi). This choice is motivated by the fact that, while an analysis of EEG is mandatory to precisely understand the behavior of brain activity, we believe that a practical tool should be based only on physiological variables routinely monitored and visualized by anesthesiologists. The proposed approach could be of great help for clinicians by improving the fine tuning of the DoA. Furthermore, the possibility to predict the evolutions of variables would help preventing side effects such as low blood pressure. A tool that could autonomously help the anesthesiologist would improve safety-level in the surgical room.

4 Outline of the thesis

This thesis is organized as follows:

- Chapter 2 introduces 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. We solve this problem by introducing an algorithm that combines barrier methods, alternating minimization, and manifold optimization. A relaxed algorithm is also proposed, which allows to scale in time with the graph dimensions. Finally, the two proposed algorithms are tested on several synthetic and real databases, and compared to state-of-the-art approaches.

- Chapter 3 provides a new approach to learn representation of multivariate signal based on tensor and convolutional dictionary learning approaches. We show that a CP low-rank constraint on the multivariate activations allows to take into account their possible (linear) structure, together with allowing a better robustness to noise. Two algorithms either based on ADMM and FISTA are proposed, and a large amount of experiments are performed on both synthetic and real data.

- Chapter 4 proposes a simple approach to identify the frequency support of multivariate time-vertex graph signals. It is built around the notion of graph product and the definition of three graphs that each model the interactions within one dimension (time, space, feature space). By using the tensor formalism, several sparsity methods are proposed. These approaches are tested on multichannel EEG signals in order to assess the sampling and interpolation performances of the proposed framework.

- Chapter 5 introduces a decision support algorithm based on TPSR and apprenticeship learning which assists anesthesiologists in administering drugs in order to maintain the optimal DoA. In the objective of proposing a practical tool, the model only relied on four commonly monitored variables. The performances of the resulting agent is analyzed with diverse metrics and through its confrontation to real anesthesiologists.

5 Publications

Some of the work presented in this document are the result of the following publications:

Mathematical publications:

1. Learning laplacian matrix from bandlimited graph signals, *Le Bars, Batiste* and Humbert, Pierre* and Oudre, Laurent and Kalogeratos, Argyris. In IEEE International Conference on Acoustics, Speech and Signal Processing 2019 (ICASSP). *Authors with equal contribution to this work.*

2. Subsampling of multivariate time-vertex graph signals, *Humbert, Pierre and Oudre, Laurent and Vayatis, Nicolas. In European Signal Processing Conference 2019 (EUSIPCO).*

3. Low rank activations for tensor-based convolutional sparse coding, *Humbert, Pierre and Audiffren, Julien and Oudre, Laurent and Vayatis, Nicolas. In IEEE International Conference on Acoustics, Speech and Signal Processing 2020 (ICASSP).*

4. Detecting multiple change-points in a piece-wise constant varying Ising model, *Le Bars, Batiste and Humbert, Pierre and Kalogeratos, Argyris, and Vayatis, Nicolas. In International Conference on Machine Learning 2020 (ICML).*

5. Tensor convolutional sparse coding with low-rank activations, an application to EEG analysis, *Humbert, Pierre and Oudre, Laurent and Vayatis, Nicolas and Audiffren, Julien. (Submitted).*

6. Robust kernel density estimation with median-of-means principle, *Humbert, Pierre* and Le Bars, Batiste* and Minvielle, Ludovic* and Vayatis, Nicolas. (Submitted). *Authors with equal contribution to this work.*

7. Learning laplacian matrix from graph signals with sparse spectral representation, *Humbert, Pierre* and Le Bars, Batiste* and Oudre, Laurent and Kalogeratos, Argyris, and Vayatis Nicolas. (Submitted). *Authors with equal contribution to this work.*

Medical publications:

1. Learning from an expert, *Humbert, Pierre and Audiffren, Julien and Clément, Dubost and Oudre, Laurent. In Neural Information Processing Systems 2016 (NeurIPS) Workshop on Machine Learning for Health.*

2. Selection of the best electroencephalogram channel to predict the depth of anesthesia, *Dubost, Clément and Humbert, Pierre and Benizri, Arno and Tourtier, Jean-Pierre and Vayatis, Nicolas and Vidal, Pierre-Paul. In Frontiers in Computational Neuroscience.*

3. Prediction of the Depth of anesthesia with hidden Markov model, Dubost, Clément and Humbert, Pierre and De Rocquigny, Gaël and Vayatis, Nicolas and Vidal, Pierre-Paul, In *Virtual Physiological Human 2020 (VPH).*

4. Apprenticeship learning for a predictive state representation of anesthesia, *Humbert, Pierre and Dubost, Clément and Audiffren, Julien and Oudre, Laurent. In IEEE Transactions on Biomedical Engineering (TBME).*

2

Learning Laplacian matrix from graph signals with sparse spectral representation

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. We propose an optimization program to learn the Laplacian of this graph and provide two algorithms to solve it, called IGL-3SR and FGL-3SR. To alleviate this ill-posed problem, signals 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. They are also assumed to behave smoothly with respect to the underlying graph structure. 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 easy to compute closed-form solutions. To justify our approach, we present a probabilistic interpretation of the optimization program as a Factor Analysis Model. Finally, we extensively evaluate both algorithms 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.

Contents

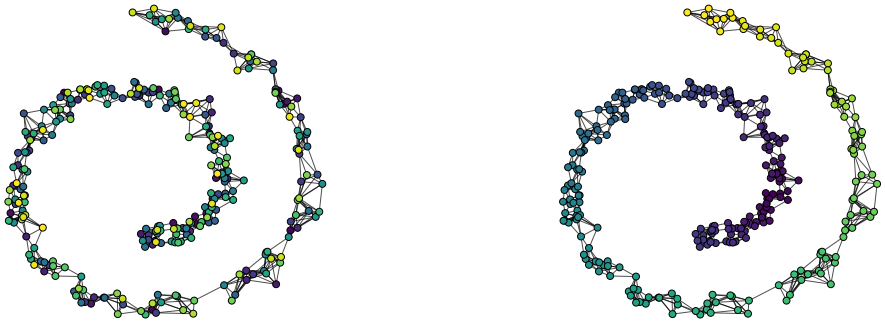
The material of this chapter is based on the following publications:

B. Le Bars*, P. Humbert*, L. Oudre, and A. Kalogeratos. Learning Laplacian Matrix from Bandlimited Graph Signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019. *Authors with equal contribution.

P. Humbert*, B. Le Bars*, L. Oudre, A. Kalogeratos, and N. Vayatis. Learning Laplacian Matrix from Graph Signals with Sparse Spectral Representation. *(Submitted)*. *Authors with equal contribution.

1 Introduction

Graphs are fundamental to model pairwise relations between objects or entities of interest. In the past years, they have proven their efficiency in a large variety of fields from biology [Barabasi and Oltvai, 2004] to neuroscience [Richiardi et al., 2013]. The strength of such concept is explained by its flexibility and its capacity to represent irregular and complex structures that can not be analyze with standard tools. However, while the availability of the graph is a core assumption in many computational tasks, such as spectral clustering [Von Luxburg, 2007], semi-supervised learning [Zhu, 2005], or graph signal processing [Shuman et al., 2013; Sandryhaila and Moura, 2013; Ortega et al., 2018], in most situations no natural graph can be derived or defined. In this situation, one approach is to infer the underlying graph from available data. This task, often referred to as *graph learning*, has also received significant attention in various fields such as machine learning, signal processing, biology, meteorology, etc. [Friedman et al., 2008; Hecker et al., 2009; William et al., 2017; Dong et al., 2019].

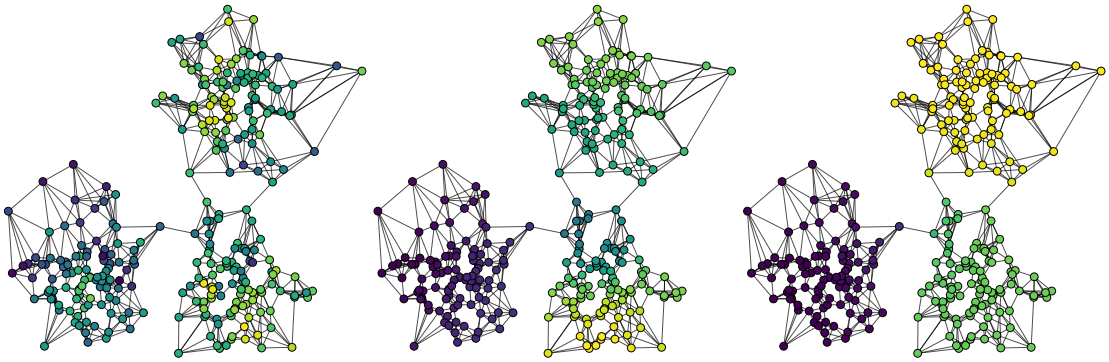


(a) (b)

Figure 2.1: Two graph signals observed on the same graph of 300 nodes. Same colors represent identical values on the nodes. (a) The first signal does not admit smoothness on the graph. (b) The second signal admits smoothness at the level of adjacent nodes. From the definition, smooth graph signals in the vertex domain are signals where neighboring nodes tend to have similar values.

Learning a graph is an ill-posed problem as several graphs can explain the same set of observations. In consequence, previous works have been devoted to introduce underlying models or constraints that would narrow down the range of possible solutions. For instance, physical constraints may be imposed to suggest epidemic models or other information propagation and interaction models [Rodriguez et al., 2011; Du et al., 2012; Gomez-Rodriguez et al., 2016]. From a statistical perspective, the graph learning task is seen as the estimation of the parameters of a certain probability distribution parameterized 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 [Koller and Friedman, 2009; Yang et al., 2015; Wang and Kolar, 2016; Tarzanagh and Michailidis, 2018]. Here, the graph structure encompasses the conditional dependencies between variables. In the particular case of a Gaussian Random Field, the graph learning task consists in estimating the inverse covariance matrix, known as the *precision matrix* [Banerjee et al., 2008]. Several constraints could be imposed on this matrix. For instance, in [Friedman et al., 2008], the proposed estimation method, known as the Graph-Lasso algorithm, relies on the assumption that the precision matrix is sparse.

More recently, *Graph Signal Processing* (GSP) [Shuman et al., 2013; Ortega et al., 2018; Djuric and Richard, 2018], has appeared to be a powerful alternative framework to learn graphs [Pasdeloup et al., 2017; Thanou et al., 2017; Segarra et al., 2017; Dong et al., 2019]. Indeed, GSP generalizes standard concepts and tools of signal processing to multivariate signals recorded over graphs. Hence, notions such as smoothness, sampling, filtering, etc., were adapted to GSP, and then used to learn specific graphs. For instance, the *smoothness* of observations with respect to the true underlying graph is one of the most common assumption [Daitch et al., 2009; Kalofolias, 2016; Egilmez et al., 2016; Chepuri et al., 2017; Dong et al., 2019] to learn graphs on which signals have small local variations among adjacent nodes (Figure 2.1). Another natural assumption is the sparsity of the observations in a graph spectral basis [Valsesia et al., 2018; Sardellitti et al., 2019]. Indeed, in clustering for instance, the vector of labels seen as a signal over the nodes of a graph, exhibits a sparse spectral representation: it is smooth within each cluster and varies from one cluster to another (Figure 2.2). Building such graph is therefore of huge interest for graph-based clustering approaches. 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 [Chen et al., 2015b,d]. The GSP framework



(a) (b) (c)

Figure 2.2: Three smooth graph signals ($N = 300$) with decreasing bandlimitedness: (a) A signal with a 150 sparse spectral representation. (b) A signal with a 6 sparse spectral representation. (c) A signal with a 3 sparse spectral representation. Same colors represent identical values on the nodes.

is also strongly motivated by a wide range of applications where there exist inherent structures behind data observations. One remarkable and elegant application of the GSP is for example in the analysis of brain activity [Huang et al., 2016, 2018] where the main interest lies in its potential to jointly model brain structure as a graph and brain activities as signals residing on the nodes of this graph. The structural and functional connectivity of the brain related to different diseases or external stimuli can then be study at the same time.

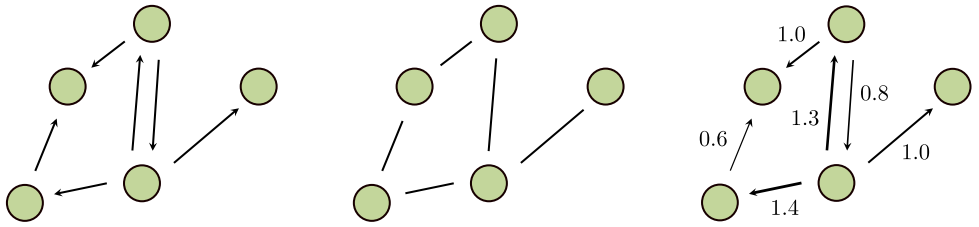
Aim and main contributions. In the present chapter, we introduce an optimization problem to learn a graph from signals that are assumed smooth and with a sparse representation in the spectral domain of the graph. These properties, all borrowed from GSP, can be considered either as constraints or regularizations for the graph learning task, and offer a new perspective on the topic.

The main contributions of this chapter are summarized as follows:

- The graph learning task problem is cast as the minimization of a smooth non-convex objective function over a non-convex set (Section 3). This problem is efficiently solved by introducing an algorithm that combines barrier methods, alternating minimization, and manifold optimization (Section 4). Another algorithm is also proposed, which allows to scale in time with the graph dimensions (Section 5).
- A factor analysis model for smooth graph signals with sparse spectral representation is introduced (Section 6). This model provides a probabilistic interpretation of our optimization problem by linking its objective function to a maximum a posteriori estimation.
- The two proposed algorithms are tested on several synthetic and real data, and compared to state-of-the-art approaches (Section 8). Experimental results show that our approach allows to obtain similar or better performance than standard existing methods while significantly lowering the necessary computing resources.

2 Background on graphs

A graph describes a network by specifying pairs of entities, denoted nodes, that are connected to one another. This connection can be symmetric (e.g. neighborhood) or asymmetric (e.g. prey v.s.



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

Figure 2.3: Graphical representation of a directed (a), undirected (b) and (directed) weighted (c) graph. Directed edges are represented by arrows, and their thickness represents the weight.

predator). We begin this section by providing definitions for directed, undirected, and weighted graphs.

2.1 Definitions from graph theory

Definition 2.1. *(*Directed graph.*) – A directed graph $G = (\mathcal{V}, \mathcal{E})$ is defined via a finite set of nodes (or vertices) $\mathcal{V} = \{1, \cdots, N\}$, and a set of edges $\mathcal{E} = \{(i, j, w_{ij}), i, j \in \mathcal{V}\} \subset \mathcal{V} \times \mathcal{V}$, i.e. pairs of nodes that are considered neighbors. The size of G denotes the number of nodes of G, i.e. card$(\mathcal{V}) = N$.*

In the sequel, we will always assume that a graph has no self-loops (i.e. $\forall u \in \mathcal{V}, (u, u) \notin \mathcal{E}$), and no multiple edges on the same pair of nodes. Furthermore, we will always consider undirected graph. The following definition encodes this notion where connections between entities are symmetric.

Definition 2.2. *(*Undirected graph.*) – An undirected graph $G = (\mathcal{V}, \mathcal{E})$ is a directed graph whose edge set is symmetric, i.e. $\forall (u, v) \in \mathcal{E}, (v, u) \in \mathcal{E}$.*

In many applications, the importance of a connection between two nodes is variable. Assigning a weight to each edge is a very natural way to take this imbalance into account.

Definition 2.3. *(*Weighted graph.*) – A weighted graph is a pair $G = (\mathcal{V}, \mathcal{E})$ with nodes $\mathcal{V} = \{1, \cdots, N\}$, and set edges $\mathcal{E} = \{(i, j, w_{ij}), i, j \in \mathcal{V}\}$ with weights $w_{ij} \in \mathbb{R}^+$ arranged in a weights matrix $\boldsymbol{W} \in \mathbb{R}^{N \times N}$. This graph can be either directed of undirected. It is directed if connections between nodes are asymmetric and the pairs (i, j) are ordered. It is undirected if the pairs (i, j) are not ordered and hence interactions between nodes are symmetric.*

Remark 2.1. *A graph is said to be binary if the weights are in $\{0, 1\}$. In this case, the weights matrix $\boldsymbol{W}$ is called the adjacency matrix and is often denoted $\boldsymbol{A}$.*

Many graph characteristics can be expressed using the weights matrix $\boldsymbol{W}$, making it an important piece of network analysis. From it, we can for example introduce the notion of degree.

Definition 2.4. *(*Degree and degree matrix.*) – The degree of a node i is the number of nodes to whom it is connected and is expressed as $d_i = \sum_{i=1}^{N} \boldsymbol{W}_{i,j}$. The degree matrix $\boldsymbol{D}$ is a diagonal matrix which contains the degree of each node.*

In the following, we will focus mainly on a matrix called the graph Laplacian. While several definitions are proposed in the literature, we consider in this manuscript the *combinatoral graph Laplacian.*

Definition 2.5. *(*Combinatorial graph Laplacian*.) – A graph is entirely described by its combinatorial graph Laplacian matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is the degree matrix and $\boldsymbol{W}$ the weights matrix.*

The Laplacian matrix of a graph is the subject of numerous works, especially in *spectral graph theory* [Chung and Graham, 1997; Mohar, 1992; Das, 2004; Zhang, 2011]. A great deal of attention is dedicated to the eigenvalues and eigenvectors of the Laplacian as they reflect important properties of the associated graph (see e.g. [Alon, 1986; De Abreu, 2007]). Among all the eigenvalues of the Laplacian, one of the most popular is the second smallest, called the *algebraic connectivity*, because this is a convenient value to measure how well a graph is connected [Fiedler, 1973]. For example, a graph is connected if and only if its algebraic connectivity is different from zero (a direct consequence of the Matrix-Tree Theorem [Biggs et al., 1993; De Abreu, 2007]). The associated eigenvector is called the Fiedler vector and is also of great interest [Fiedler, 1975]. We now recall two important propositions related to the spectrum of a Laplacian.

Proposition 2.1. *If G is undirected, with no self-loops, $\boldsymbol{L}$ is a real (symmetric) positive semi-definite matrix and, its eigendecomposition – which is also its singular value decomposition – can be written as $\boldsymbol{L} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\top$, with $\boldsymbol{\Lambda} = diag(\lambda_1, \dots, \lambda_N)$ a diagonal matrix with the eigenvalues and $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$ a matrix with the eigenvectors as columns.*

Proposition 2.2. *Let assume that G has a unique connected component. In this particular case, $\lambda_1 = 0$ and $\boldsymbol{x}_1 = \mathbf{1}_N$, where $\mathbf{1}_N$ is the constant unitary vector of size N.*

When a matrix satisfies these two propositions, one can treat it as a Laplacian and consider the graph associated with it. These two propositions are therefore cornerstone in the graph inference task as they define sufficient constraints to recover a true Laplacian.

2.2 Definitions from GSP

In this section, we introduce basic GSP concepts. A full overview can be found in [Shuman et al., 2013; Ortega et al., 2018] and more recently in [Stanković et al., 2019].

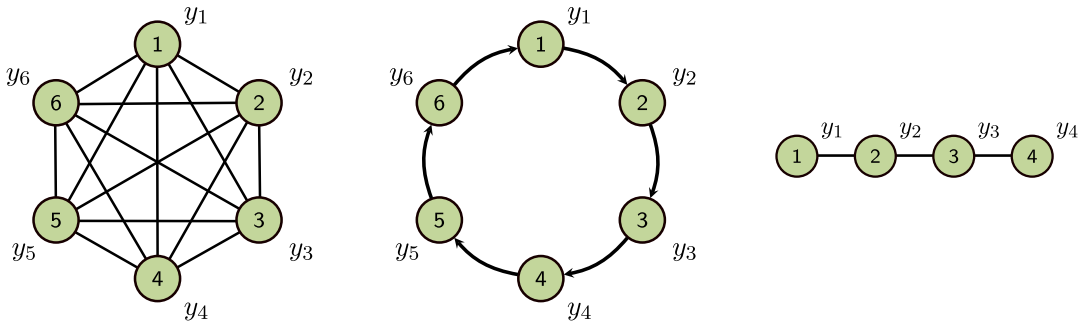
Definition 2.6. *(*Graph signal*.) – A graph signal, or graph function, is defined as a function $y : \mathcal{V} \longrightarrow \mathbb{R}^N$ that assigns a scalar value to each node. This function can be represented as a vector $\boldsymbol{y} \in \mathbb{R}^N$, with $\boldsymbol{y}_i$ the function value at the i-th node.*

It is possible to create a spectral representation of $\boldsymbol{y}$ adapted to a graph using the Graph Fourier Transform (GFT).

Definition 2.7. *(*Graph Fourier Transform*.) – Given a graph G, the GFT of a graph signal $\boldsymbol{y}$ is given by $\boldsymbol{h} = \boldsymbol{X}^\top \boldsymbol{y}$, where the components of $\boldsymbol{h}$ are interpreted as Fourier coefficients, the eigenvalues as distinct frequencies, and the eigenvectors as a decomposition basis.*

This definition is motivated by one important observation. Let consider a directed cycle graph, which is the support of classical time-varying signals (see Figure 2.4(b)). Interestingly, it appears that the eigenvector decomposition of the adjacency or Laplacian matrix gives as eigenvector matrix the Fourier matrix (see e.g. [Segarra et al., 2016; Huang et al., 2016]). Hence, the GFT of a graph signal $\boldsymbol{y}$ (with respect to the cyclic graph) is its discrete Fourier transformation.

The subsequent definitions describe two fundamental properties of graph signals assumed in this chapter.



(a) Complete graph **(b)** Directed cycle graph **(c)** Line graph

Figure 2.4: Three particular graphs: (a) Complete graph, (b) Directed cyclic graph, and (c) Line graph.

Definition 2.8. *(*Spectral sparsity*.) – We say that a graph signal $\boldsymbol{y}$ admits a $k \in \mathbb{N}^+$ sparse spectral representation with respect to a graph G if for $\boldsymbol{h} = \boldsymbol{X}^\top \boldsymbol{y}$*

$$\|\boldsymbol{h}\|_0 = k \ , \tag{2.1}$$

i.e. if the number of non-zero elements in its Fourier coefficient vector is equal to k.

Relation with clusters of a graph. The spectral sparsity is related to the number of clusters of a graph [Von Luxburg, 2007; Sardellitti et al., 2019]. To see this, let consider an ordered vector of two labels $\boldsymbol{y} = (-1, -1, 1, 1)$. In the case where the graph has two connected components i.e. two "perfect clusters", the first two columns of $\boldsymbol{X}$ are $\boldsymbol{x}_1 = (0, 0, 1, 1)$ and $\boldsymbol{x}_2 = (1, 1, 0, 0)$ and the vector $\boldsymbol{y}$ is thus a linear combination of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. As $\boldsymbol{X}$ is an orthogonal matrix, $\langle \boldsymbol{x}_3, \boldsymbol{y} \rangle = \langle \boldsymbol{x}_4, \boldsymbol{y} \rangle = 0$. In other words, $\boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{h}$ admits a 2 sparse spectral representation with respect to this graph.

Relation with sampling. This property is also crucial for sampling i.e. measuring a graph signal on a reduced set of nodes that allow its stable reconstruction [Chen et al., 2015d; Marques et al., 2016; Lorenzo et al., 2018; Puy et al., 2018; Wang et al., 2018a; Tanaka et al., 2020]. An intuitive way to formalize (irregular) sampling for a graph signal is to introduce a $M \times N$ selection matrix $\boldsymbol{C}$ and to define the sampled signal of size M as

$$\bar{\boldsymbol{y}} = \boldsymbol{C}\boldsymbol{y} \ . \tag{2.2}$$

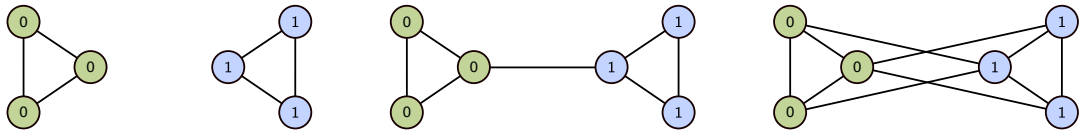
If $\boldsymbol{C}$ is chosen as binary, it has a single non-zero element per row, and at most one non-zero element per column. Hence, the signal $\bar{\boldsymbol{y}}$ is a selection of M out of the N elements of $\boldsymbol{y}$. Now, let assume that $\boldsymbol{y}$ is k-sparse. The sampled signal $\bar{\boldsymbol{y}}$ is then

$$\bar{\boldsymbol{y}} = \boldsymbol{C}\boldsymbol{y} = \boldsymbol{C}\boldsymbol{X}\boldsymbol{h} = \boldsymbol{C}\boldsymbol{X}_k\boldsymbol{h}_k \ , \tag{2.3}$$

where $\boldsymbol{X}_k$, and $\boldsymbol{h}_k$ are $\boldsymbol{X}$ and $\boldsymbol{h}$ without the irrelevant dimensions determined by the sparsity of $\boldsymbol{h}$. If for a specific choice of $\boldsymbol{C}$, the matrix $\boldsymbol{C}\boldsymbol{X}_k \in \mathbb{R}^{K \times K}$ is invertible, $\boldsymbol{h}_k$ can be recovered from $\bar{\boldsymbol{y}}$ and the signal in the original domain can be found from its sampling i.e.

$$\boldsymbol{y} = \boldsymbol{X}_k(\boldsymbol{C}\boldsymbol{X}_k)^{-1}\bar{\boldsymbol{y}} \ . \tag{2.4}$$

This equation shows how the original signal can be interpolated from its samples. However, note that the matrix $\boldsymbol{C}\boldsymbol{X}_k$ has to be invertible. Hence, the key for guaranteeing perfect signal reconstruction is to select a subset of nodes such that the corresponding rows in $\boldsymbol{X}_k$ are linearly independent.



(a) 2-bandlimited **(b)** 3-bandlimited **(c)** 5-bandlimited

Figure 2.5: A graph signal $\boldsymbol{y}$ taking its values in $\{0, 1\}$ on three different graphs. This signal can potentially live on these graphs but only one leads to a sparse graph signal representation. In this illustration, while all the graphs are valid a priori, only the second one favor the sparsity property of $\boldsymbol{y}$ and have one connected component.

Remark 2.2. *(*Sparsity assumption.*) – In GSP this property is known as bandlimitedness. In general, it is assumed that the null components of $\boldsymbol{h}$ are those associated to the largest eigenvalues (frequencies). Indeed, this additional hypothesis permits to fit the fundamental principle of signal processing which suggests that the high-frequency band of a signal should be filtered, as they carry mainly noise and little or no information. This assumption on graph signals is very common, especially in GSP where it is the main hypothesis of several GSP sampling methods [Narang et al., 2013; Anis et al., 2014; Chen et al., 2015b,d; Marques et al., 2016].*

Definition 2.9. *(*Smoothness.*) – A graph signal $\boldsymbol{y}$ is said to be $s \geq 0$* smooth *with respect to a graph G if*

$$\|\boldsymbol{L}^{1/2}\boldsymbol{y}\|_2^2 = \boldsymbol{y}^\intercal \boldsymbol{L}\boldsymbol{y} = \frac{1}{2}\sum_{i,j} \boldsymbol{w}_{ij}\left(\boldsymbol{y}_i - \boldsymbol{y}_j\right)^2 \leq s \cdot \|\boldsymbol{y}\|_2^2 \,. \tag{2.5}$$

Remark 2.3. *(*Smoothness assumption.*) – While this property can be quantified with various metrics, the most common is given by the above definition. From this formula, we see that $\boldsymbol{y}$ gets smoother, thus* (2.5) *lower, when its value at any two nodes gets closer as their edge weight gets larger. This natural property has consequently been widely considered for graph inference [Daitch et al., 2009; Dong et al., 2016; Kalofolias, 2016]. Also note that if $\boldsymbol{x}$ is an eigenvector of the Laplacian matrix $\boldsymbol{L}$ associated to the eigenvalue λ, then $\boldsymbol{x}^\intercal \boldsymbol{L}\boldsymbol{x} = \lambda \boldsymbol{x}^\intercal \boldsymbol{x} = \lambda$.*

2.3 Graph learning task in GSP

To highlight the impact of the sparsity assumption on the graph inference task, we illustrate the interplay between the graph and the data in Figures 2.5. In this example, we consider an unordered signal $\boldsymbol{y}$ tanking its values in $\{0, 1\}$. This signal can potentially be defined on the three graphs from Figure 2.5. Indeed, without any assumption on the properties of the graph signal, they are all valid choices. In the other hand, if we assume that the signal need to admit a sparse spectral representation on its underlying graph (with one connected component), then (b) is the most reasonable candidate. In this chapter, our objective is to learn this graph from a set of observations that are all supposed to share the same underlying graph.

3 Problem statement

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

3.1 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 $\{\boldsymbol{y}^{(k)}\}_{k=1}^{n}$ of size N. We collect them in a matrix $\boldsymbol{Y} = [\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(n)}] \in \mathbb{R}^{N \times n}$. The proposed graph learning framework takes as input the matrix $\boldsymbol{Y}$ and outputs the Laplacian matrix $\boldsymbol{L}$ associated to G (note that both notions are equivalent). Our learning process is based on the following assumptions:

Assumption 2.1. (Assumption on the graph G) – *G is undirected, with no self-loop and has a single connected component.*

With Assumption 2.1, $\boldsymbol{L}$ is a symmetric positive semi-definite matrix with eigenvalue decomposition $\boldsymbol{L} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal$, where $\lambda_1 = 0$ and $x_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N$ (see Proposition 2.1 and 2.2).

Assumption 2.2. (Assumption on the signals $\boldsymbol{Y}$) – *Graph signals $\boldsymbol{Y}$ defined over the true underlying graph G are assumed s-smooth and admit a k-sparse spectral representation, with unknown values for s and k.*

On the spectral sparsity assumption. To further justify the consideration of this property we can see that it is also related to the cluster structure of a graph. Indeed, if a graph has k clusters, a signal that is smooth within each cluster and can vary arbitrarily across different clusters will admit a k-sparse spectral representation. In this context, the non-null weights of h will be necessarily associated to the k first eigenvectors of the corresponding Laplacian matrix as these eigenvectors are also smooth within the clusters [Von Luxburg, 2007]. To enforce such behavior in the graph learning process, i.e. make sure that only the first coefficients of h are non-zero, the bandlimitness property must be combined with the smoothness property.

Figures 2.1 and 2.2 show examples of graph signals that illustrate the intuition behind our two core assumptions on signals.

3.2 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 2.2 (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 $\boldsymbol{L} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal$. The optimization problem incorporates a linear least square regression term depending of $\boldsymbol{Y}$, $\boldsymbol{X}$, and $\boldsymbol{H}$, which controls the distance of the new representation $\boldsymbol{X}\boldsymbol{H}$ to the observations $\boldsymbol{Y}$. In addition, due to Assumption 2.2, we add two penalization terms: One to control the smoothness of the new representation, depending on $\boldsymbol{\Lambda}$ and $\boldsymbol{H}$; the other to control the sparsity on the spectral domain, which only depends on $\boldsymbol{H}$. Finally, as we want to learn a Laplacian matrix satisfying Assumption 2.1, equality and inequality constraints relative to $\boldsymbol{X}$ and $\boldsymbol{\Lambda}$ are necessary. To that end, we introduce the following optimization problem:

$$\min_{\boldsymbol{H},\boldsymbol{X},\boldsymbol{\Lambda}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{H}\|_F^2 + \alpha\|\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\|_F^2 + \beta\|\boldsymbol{H}\|_S \,, \tag{2.6}$$

$$\text{s.t.} \quad \begin{cases} \boldsymbol{X}^\intercal \boldsymbol{X} = \boldsymbol{I}_N,\ \boldsymbol{x}_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N\ , & \text{(a)} \\ (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal)_{k,\ell} \leq 0 \quad k \neq \ell\ , & \text{(b)} \\ \boldsymbol{\Lambda} = \operatorname{diag}(0, \lambda_2, \dots, \lambda_N) \succeq 0\ , & \text{(c)} \\ \operatorname{tr}(\boldsymbol{\Lambda}) = N \in \mathbb{R}_*^+\ , & \text{(d)} \end{cases}$$

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

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

Interpretation of the terms. In the objective function (2.6), the first term corresponds to the quadratic approximation error of $\boldsymbol{Y}$ by $\boldsymbol{X}\boldsymbol{H}$, where $\|\cdot\|_F$ is the Frobenius norm.
The second term is a *smoothness regularization* imposed to the approximation $\boldsymbol{X}\boldsymbol{H}$. Rewriting the smoothness equation (2.5) for the set of graph signals $\boldsymbol{X}\boldsymbol{H}$, we obtain

$$\|\boldsymbol{L}^{1/2}\boldsymbol{X}\boldsymbol{H}\|_F^2 = \|\boldsymbol{X}\boldsymbol{\Lambda}^{1/2}\boldsymbol{X}^\intercal\boldsymbol{X}\boldsymbol{H}\|_F^2 = \|\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\|_F^2 = \sum_{i=1}^{N} \boldsymbol{\lambda}_i \|\boldsymbol{H}_{i,:}\|_2^2\ ,$$

where $\boldsymbol{H}_{i,:}$ is the i-th row of the matrix $\boldsymbol{H}$. This kind of regularization is very common in graph learning [Kalofolias, 2016; Chepuri et al., 2017]. 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 $\boldsymbol{H}$ 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 $\boldsymbol{H}$ as Fourier coefficients.
The last term, $\beta\|\boldsymbol{H}\|_S$, 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 $\boldsymbol{H}$) or $\ell_{2,0}$ (number of rows with ℓ_2-norm different than 0) that induces a row-sparse solution $\widehat{\boldsymbol{H}}$.

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 [Sardellitti et al., 2019]. Therefore, in this work, 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 $\boldsymbol{H}$ "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 $\boldsymbol{H}$. A discussion on the influence of these hyperparameters and an efficient way to fix them is provided in Section 8.3.1. Finally, the first three constraints (2.6a), (2.6b), (2.6c) enforce $\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal$ to be a Laplacian matrix of a graph with a single connected component (Assumption 2.1). More specifically, by definition, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ with $\boldsymbol{W} \in \mathbb{R}_+^{N\times N}$, thus we necessary have $\forall k \neq \ell, \boldsymbol{L}_{k,\ell} = (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal)_{k,\ell} \leq 0$ (constraint (2.6b)). Furthermore, as $\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal$ is the eigendecomposition of the Laplacian matrix of an undirected graph with a single connected component (Assumption 2.1), $\boldsymbol{X}^\intercal\boldsymbol{X} = \boldsymbol{I}_N, \boldsymbol{x}_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N$ and $\lambda_1 = 0 < \lambda_2 \leq \ldots \leq \lambda_N$ (constraints (2.6a) and (2.6c)). The last constraint (2.6d) was proposed in Dong et al. [2016] as to impose structure in the learned graph so that the trivial solution $\widehat{\boldsymbol{\Lambda}} = \mathbf{0}$ is avoided. A discussion about values other than N is made in Kalofolias [2016].

Properties of the objective function (2.6). The objective function (2.6) 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 $\boldsymbol{H}$, $\boldsymbol{X}$, or $\boldsymbol{\Lambda}$, 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 $(2.6a)$ and inequalities $(2.6b)$, 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.

3.3 Reformulation of the problem

As stated in Section 3.2, problem (2.6) is not jointly convex and cannot be solved easily with constraints $(2.6a)$ and $(2.6b)$. In this section, we propose to rewrite constraints $(2.6a)$ and $(2.6b)$, in order to define a new equivalent optimization problem that can be solved with well-known techniques.

3.3.1 Reformulation of the constraint $(2.6a)$

In this section, we show that the constraints (2.6a) 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 4.3 that there exist efficient algorithms that perform optimization over such manifold.

Definition 2.10. (Orthogonal group) – *The space of orthogonal matrices in $\mathbb{R}^{N\times N}$, called orthogonal group, is the space:*

$$Orth(N) = \{\boldsymbol{X} \in \mathbb{R}^{N\times N} \mid \boldsymbol{X}^\intercal \boldsymbol{X} = \boldsymbol{I}_N\} .$$

Lemma 2.1. – *Given $\boldsymbol{X}, \boldsymbol{X}_0 \in \mathbb{R}^{N\times N}$ two orthogonal matrices, both having their first column equal to $\frac{1}{\sqrt{N}}\mathbf{1}_N$ (constraint $(2.6a)$), we have the following equality*

$$\boldsymbol{X} = \boldsymbol{X}_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\intercal \\ \mathbf{0}_{N-1} & [\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:} \end{bmatrix} ,$$

with $[\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:}$ denoting the submatrix of $\boldsymbol{X}_0^\intercal \boldsymbol{X}$ containing everything but the first row and column of itself. Furthermore, $[\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:}$ is in $Orth(N-1)$.

The above lemma allows us to build an equivalent formulation of Problem (2.6) given by the following proposition.

Proposition 2.3. – *Given $\boldsymbol{X}_0 \in \mathbb{R}^{N\times N}$ an orthogonal matrix with first column being equal to $\frac{1}{\sqrt{N}}\mathbf{1}_N$, an equivalent formulation of optimization problem (2.6) is given by*

$$\min_{\boldsymbol{H},\boldsymbol{U},\boldsymbol{\Lambda}} \left\| \boldsymbol{Y} - \boldsymbol{X}_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\intercal \\ \mathbf{0}_{N-1} & \boldsymbol{U} \end{bmatrix} \boldsymbol{H} \right\|_F^2 + \alpha \|\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\|_F^2 + \beta \|\boldsymbol{H}\|_S \triangleq f(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\Lambda}) , \quad (2.7)$$

$$
\text{s.t.} \quad \begin{cases} \boldsymbol{U}^\intercal \boldsymbol{U} = \boldsymbol{I}_{N-1} \ , & (a') \\ \left(\boldsymbol{X}_0 \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U} \end{bmatrix} \boldsymbol{\Lambda} \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U}^\intercal \end{bmatrix} \boldsymbol{X}_0^\intercal \right)_{k,\ell} \leq 0 \quad k \neq \ell \ , & (b') \\ \boldsymbol{\Lambda} = diag(0, \lambda_2, \ldots, \lambda_N) \succeq 0 \ , & (c) \\ tr(\boldsymbol{\Lambda}) = N \in \mathbb{R}_*^+ \ . & (d) \end{cases}
$$

The latter proposition says that since the first column of $\boldsymbol{X}$ is fixed and known, it is sufficient to look for an optimal rotation of a valid matrix $\boldsymbol{X}_0$ that preserves the first column. Such a rotation matrix is given above and is parametrized by a $\boldsymbol{U}$ in $Orth(N-1)$. Note that in practice, to find a matrix $\boldsymbol{X}_0$ satisfying (2.6a), we build the Laplacian of any graph with a single connected component and take its eigenvectors.

3.3.2 Log-barrier method for constraint (2.7b')

In order to deal with constraint (2.7b'), we propose to use a log-barrier method. This barrier function allows us to consider an approximation of problem (2.7) where the inequality constraint (2.7b') is made implicit in the objective function. Denoting by $f(\cdot)$ the objective function of (2.7), we want to solve

$$
\min_{\boldsymbol{H},\boldsymbol{U},\boldsymbol{\Lambda}} \quad f(\boldsymbol{H},\boldsymbol{U},\boldsymbol{\Lambda}) + \frac{1}{t}\phi(\boldsymbol{U},\boldsymbol{\Lambda}) \qquad \text{s.t.} \quad (2.7\text{a'}), (2.7\text{c}), (2.7\text{d}) \ , \tag{2.8}
$$

where t is a fixed positive constant and $\phi(\cdot)$ is the log-barrier function associated to the constraint (2.7b').

Definition 2.11. (Log-barrier function) – *Let the following matrix in* $\mathbb{R}^{N\times N}$*:*

$$
\boldsymbol{h}(\boldsymbol{U},\boldsymbol{\Lambda}) = \boldsymbol{X}_0 \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U} \end{bmatrix} \boldsymbol{\Lambda} \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U} \end{bmatrix}^\intercal \boldsymbol{X}_0^\intercal \ ,
$$

involved in the constraint (2.7b')*. 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:*

$$
\phi(\boldsymbol{U},\boldsymbol{\Lambda}) = -\sum_{k=1}^{N-1}\sum_{\ell>k}^{N} \log\Big(-\boldsymbol{h}(\boldsymbol{U},\boldsymbol{\Lambda})_{k,\ell}\Big) \ , \tag{2.9}
$$

with $\mathbf{dom}(\phi) = \big\{(\boldsymbol{U},\boldsymbol{\Lambda}) \in \mathbb{R}^{(N-1)\times(N-1)} \times \mathbb{R}^{N\times N} \mid \forall 1 \leq k < \ell \leq N, h(\boldsymbol{U},\boldsymbol{\Lambda})_{k,\ell} < 0\big\}$, i.e. its domain is the set of points that strictly satisfy the inequality constraints (2.7b').

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

4 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 (2.8). Given a fixed $t > 0$, we propose to use a block-coordinate descent on $\boldsymbol{H}$, $\boldsymbol{U}$, and $\boldsymbol{\Lambda}$, 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 2.1.

4.1 Optimization with respect to $\boldsymbol{H}$

For fixed $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$, the minimization Problem (2.8) with respect to $\boldsymbol{H}$ is

$$\min_{\boldsymbol{H}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{H}\|_F^2 + \alpha\|\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\|_F^2 + \beta\|\boldsymbol{H}\|_S \;, \quad \text{where } \boldsymbol{X} = \boldsymbol{X}_0 \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U} \end{bmatrix} . \tag{2.10}$$

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 [Ravishankar and Bresler, 2012] (resp. is a particular case of the Group Lasso [Yuan and Lin, 2006] known as Multi-Task Feature Learning [Argyriou et al., 2006]). Moreover, as $\boldsymbol{X}$ is orthogonal, we are able to find closed-form solutions (Proposition 2.4).

Proposition 2.4. (Closed-form solution for the $\ell_{2,0}$ and $\ell_{2,1}$-norms) – *The solutions of Problem* (2.10) *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* (2.10) *is given by the matrix* $\widehat{\boldsymbol{H}} \in \mathbb{R}^{N\times n}$ *where for* $1 \leq i \leq N$,

$$\widehat{\boldsymbol{H}}_{i,:} = \begin{cases} 0 & \text{if } \frac{1}{1+\alpha\lambda_i}\|(\boldsymbol{X}^\intercal\boldsymbol{Y})_{i,:}\|_2^2 \leq \beta \;, \\ \frac{1}{(1+\alpha\lambda_i)}(\boldsymbol{X}^\intercal\boldsymbol{Y})_{i,:} & \text{else .} \end{cases} \tag{2.11}$$

- *Using the* $\ell_{2,1}$*-norm, the optimal solution of* (2.10) *is given by the matrix* $\widehat{\boldsymbol{H}} \in \mathbb{R}^{N\times n}$, *where for* $1 \leq i \leq N$,

$$\widehat{\boldsymbol{H}}_{i,:} = \frac{1}{1+\alpha\lambda_i}\Big(1 - \frac{\beta}{2}\frac{1}{\|(\boldsymbol{X}^\intercal\boldsymbol{Y})_{i,:}\|_2}\Big)_+ (\boldsymbol{X}^\intercal\boldsymbol{Y})_{i,:} \;, \tag{2.12}$$

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

4.2 Optimization with respect to $\boldsymbol{\Lambda}$

For fixed $\boldsymbol{H}$ and $\boldsymbol{U}$, the optimization Problem (2.8) with respect to $\boldsymbol{\Lambda}$ is

$$\min_{\boldsymbol{\Lambda}} \alpha \underbrace{\operatorname{tr}(\boldsymbol{H}\boldsymbol{H}^\intercal\boldsymbol{\Lambda})}_{\|\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\|_F^2} + \frac{1}{t}\phi(\boldsymbol{U}, \boldsymbol{\Lambda}) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{\Lambda} = \operatorname{diag}(0, \lambda_2, \ldots, \lambda_N) \succeq 0 \;, & \text{(c)} \\ \operatorname{tr}(\boldsymbol{\Lambda}) = N \in \mathbb{R}_*^+ \;. & \text{(d)} \end{cases} \tag{2.13}$$

This objective function is differentiable and convex with respect to $\boldsymbol{\Lambda}$, and the constraints define a Simplex. Thus, several convex optimization solvers can be employed, such as those implemented in CVXPY [Diamond and Boyd, 2016]. Popular algorithms are interior-point methods or projected gradient descent methods [Maingé, 2008]. Using one algorithm of the latter type, we compute the gradient of 2.13 and project each iteration onto the Simplex [Duchi et al., 2008].

4.3 Optimization with respect to $\boldsymbol{U}$

For fixed $\boldsymbol{H}$ and $\boldsymbol{\Lambda}$, the optimization Problem (2.8) with respect to $\boldsymbol{U}$ is:

$$\min_{\boldsymbol{U}} \left\| \boldsymbol{Y} - \boldsymbol{X}_0 \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U} \end{bmatrix} \boldsymbol{H} \right\|_F^2 + \frac{1}{t}\phi(\boldsymbol{U}, \boldsymbol{\Lambda}) \qquad \text{s.t.} \qquad \boldsymbol{U}^\intercal\boldsymbol{U} = \boldsymbol{I}_{(N-1)} \;. \quad \text{(a')} \tag{2.14}$$

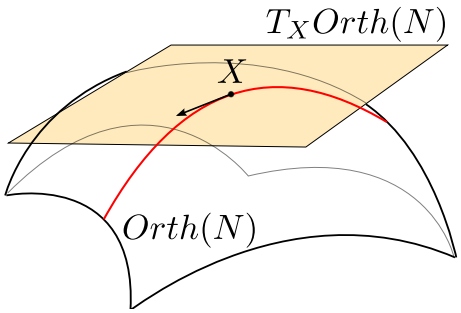


Figure 2.6: The principle of the manifold gradient descend given schematically. $T_{\boldsymbol{X}} Orth(N)$ is the tangent space of $Orth(N)$ at $\boldsymbol{X}$. The red line corresponds to a curve in $Orth(N)$ passing through the point $\boldsymbol{X}$ in the direction of the arrow. At each iteration, considering that $\boldsymbol{X}$ is the point of the current solution, a search direction belonging to $T_{\boldsymbol{X}} Orth(N)$ 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).

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) [Hyvärinen and Oja, 2000; Zou et al., 2006; Shalit and Chechik, 2014]. 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 work, we focus on manifold adaptation of descent algorithms to solve Problem (2.14).

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 2.6 provides pictures this principle.

Definition 2.12. (Tangent space at a point of $Orth(N)$) – *Let $\boldsymbol{X} \in Orth(N)$. The tangent space of $Orth(N)$ at point $\boldsymbol{X}$, denoted by $T_{\boldsymbol{X}} Orth(N)$ is a $\frac{1}{2}N(N-1)$ dimensional vector space defined by:*

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

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:

$$\text{grad}\bar{f}(\boldsymbol{X}) = P_{\boldsymbol{X}}(\nabla_{\boldsymbol{X}} \bar{f}(\boldsymbol{X})) , \tag{2.15}$$

where $P_{\boldsymbol{X}}$ is the projection onto the tangent space at $\boldsymbol{X}$, which is equal to $P_{\boldsymbol{X}}(\boldsymbol{\xi}) = \frac{1}{2}\boldsymbol{X}(\boldsymbol{X}^\intercal\boldsymbol{\xi} - \boldsymbol{\xi}^\intercal\boldsymbol{X})$, and $\nabla_{\boldsymbol{X}}$ is the standard Euclidean gradient. At each iteration, the manifold gradient descent computes the Riemannian gradient (2.15) 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. Note that there are many possible ways to perform this update [Edelman et al., 1998; Absil et al., 2009; Arora, 2009; Meyer, 2011]. Finally, from the last equation, we see that in order to solve problem (2.14) 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 2.5. (Euclidean gradient with respect to $\boldsymbol{U}$) – *The Euclidean gradient of* $f(\cdot)$ *and* $\phi(\cdot)$ *with respect to* $\boldsymbol{U}$ *are:*

$$\nabla_{\boldsymbol{U}} f(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\Lambda}) = -2\big[(\boldsymbol{H}\boldsymbol{Y}^\intercal \boldsymbol{X}_0)_{2:,2:}\big]^\intercal + 2U(\boldsymbol{H}\boldsymbol{H}^\intercal)_{2:,2:} \ ,$$

$$\nabla_{\boldsymbol{U}} \phi(\boldsymbol{U}, \boldsymbol{\Lambda}) = -\sum_{k=1}^{N-1} \sum_{\ell>k}^{N} \frac{\Big(\boldsymbol{B}_{k,\ell} + \boldsymbol{B}_{k,\ell}^\intercal\Big)\boldsymbol{U}\boldsymbol{\Lambda}_{2:,2:}}{\boldsymbol{h}(\boldsymbol{U}, \boldsymbol{\Lambda})_{k,\ell}} \ ,$$

with $\forall 1 \leq k, \ell \leq N, \boldsymbol{B}_{k,\ell} = \big(\boldsymbol{X}_0^\intercal \boldsymbol{e}_k \boldsymbol{e}_\ell^\intercal \boldsymbol{X}_0\big)_{2:,2:}$, *and* $h(\cdot)$ *from Definition 2.11.*

4.4 Log-barrier method and initialization

Choice of the t parameter. The quality of the approximation of Problem (2.7) by Problem (2.8) 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, 2004].

Initialization. At the beginning, our IGL-3SR method requires a feasible solution to initialize the algorithm. One possible choice is to take $\boldsymbol{U}$ as the identity matrix $\boldsymbol{I}_{N-1}$ and to replace $(\boldsymbol{X}_0, \boldsymbol{\Lambda})$ 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 $\boldsymbol{H}$ is not needed as we start directly with the $\boldsymbol{H}$-step.

IGL-3SR is summarized in Algorithm 2.1.

4.5 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 $\boldsymbol{X}^\intercal \boldsymbol{Y}$, 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 [Duchi et al., 2008]. Hence, denoting by $\tau_{\boldsymbol{\Lambda}}$ the number of iterations in each $\boldsymbol{\Lambda}$-step, the complexity is $\mathcal{O}(\tau_{\boldsymbol{\Lambda}}\ nN^2)$.
- 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 [Boyd and Vandenberghe, 2018]. Hence, denoting by $\tau_{\boldsymbol{X}}$ the number of iterations in each X-step, the complexity is $\mathcal{O}(\tau_{\boldsymbol{X}} \cdot nN^2)$.

Overall – The complexity to go through the big loop of IGL-3SR once (i.e. once through each of the $\boldsymbol{H}$, $\boldsymbol{\Lambda}$, and $\boldsymbol{X}$ steps) is of order $\mathcal{O}(\max(\tau_{\boldsymbol{\Lambda}}, \tau_{\boldsymbol{X}}) \cdot nN^2)$. However, recall that $\tau_{\boldsymbol{\Lambda}}$ and $\tau_{\boldsymbol{X}}$ 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.

Algorithm 2.1 The IGL-3SR algorithm with $\ell_{2,1}$-norm

Input: $\boldsymbol{Y} \in \mathbb{R}^{N \times n}, \alpha, \beta$
Input of the barrier method: $t^{(0)}, t_{\max}, \mu$ – see Section 4.4
Output: $\widehat{\boldsymbol{H}}, \widehat{\boldsymbol{X}}, \widehat{\boldsymbol{\Lambda}}$
Initialization: $\boldsymbol{L}_0$ (e.g. with a complete graph) – see Section 4.4

$t \longleftarrow t^{(0)}$
$(\boldsymbol{X}_0, \boldsymbol{\Lambda}) \longleftarrow \text{SVD}(\boldsymbol{L}_0)$
$\boldsymbol{U} \longleftarrow I_{N-1}$
while $t \leq t_{\max}$ **do**
  while not convergence **do**
    ▷ H-step: Compute the closed-form solution of Proposition (2.4)
    for $1 = 1, \ldots, N$ **do**
      $$\boldsymbol{H}_{i,:} \longleftarrow \frac{1}{1+\alpha\lambda_i}\Big(1 - \frac{\beta}{2}\frac{1}{\|(\boldsymbol{X}^\intercal\boldsymbol{Y})_{i,:}\|_2}\Big)_+ (\boldsymbol{X}^\intercal\boldsymbol{Y})_{i,:}$$
    end for
    ▷ Λ-step: Solve Problem (2.13)
    $$\boldsymbol{\Lambda} \longleftarrow \arg\min_{\boldsymbol{\Lambda}} \alpha \operatorname{tr}(\boldsymbol{H}\boldsymbol{H}^\intercal\boldsymbol{\Lambda}) + \frac{1}{t}\phi(\boldsymbol{U}, \boldsymbol{\Lambda}) \quad \text{s.t.} \quad \begin{cases} \boldsymbol{\Lambda} = \operatorname{diag}(0, \lambda_2, \ldots, \lambda_N) \succeq 0\,, \\ \operatorname{tr}(\boldsymbol{\Lambda}) = N \in \mathbb{R}_*^+ \end{cases}$$
    ▷ U-step: Solve Problem (2.14)
    while not convergence **do**
      $$\boldsymbol{U} \longleftarrow \mathit{retraction}(\boldsymbol{U}(\big[(\boldsymbol{H}\boldsymbol{Y}^\intercal\boldsymbol{X}_0)_{2:,2:}\big]\boldsymbol{U} - \boldsymbol{U}^\intercal\big[(\boldsymbol{H}\boldsymbol{Y}^\intercal\boldsymbol{X}_0)_{2:,2:}\big]^\intercal))$$
    end while
  end while
  $t \longleftarrow \mu t$
end while

5 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 (2.6). 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 $\boldsymbol{H}$, $\boldsymbol{X}$, and $\boldsymbol{\Lambda}$, which permits to decompose the problem in three partial minimizations. FGL-3SR relies on a simplification of the minimization step in $\boldsymbol{X}$ by removing the constraint (2.6b). This simplification allows us to compute a closed-form on this step which greatly accelerates the minimization. However, the constraints (2.6a) and (2.6b) are equally important to obtain a valid Laplacian matrix at the end, and reducing the problem does not ensure that the constraint (2.6b) will be satisfied. The following proposition explains why we can get rid of constraint (2.6b) 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 2.6. (Feasible eigenvalues) – *Given any $\boldsymbol{X} \in \mathbb{R}^{N \times N}$ being an orthogonal matrix with first column being equal to $\frac{1}{\sqrt{N}}\mathbf{1}_N$ (constraint (2.6a)), there always exists a matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{N \times N}$*

such that the following constraints are satisfied:

$$\begin{cases} (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal)_{i,j} \leq 0 \quad i \neq j \ , & (2.6b) \\ \boldsymbol{\Lambda} = diag(0, \lambda_2, \dots, \lambda_N) \succeq 0 \ , & (2.6c) \\ tr(\boldsymbol{\Lambda}) = c \in \mathbb{R}_*^+ \ . & (2.6d) \end{cases}$$

In Proposition 2.7 of the next section, we will see that, by ignoring constraint (2.6b) 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 $\boldsymbol{X}$, and right after always optimize with respect to $\boldsymbol{\Lambda}$. Hence, we are sure that we will obtain a proper Laplacian at the end of the process (Proposition 2.6). The initialization and the optimization with respect to $\boldsymbol{H}$ are not concerned by this relaxation and can therefore be performed as in IGL-3SR (see Sections 4.1 and 4.4).

5.1 Optimization with respect to $\boldsymbol{X}$

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

$$\min_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{H}\|_F^2 \qquad \text{s.t.} \quad \boldsymbol{X}^\intercal\boldsymbol{X} = \boldsymbol{I}_N, \ \boldsymbol{x}_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N \ , \quad (2.6a) \tag{2.16}$$

where the constraint $(2.6b)$ is missing. The closed-form solution is given next.

Proposition 2.7. (Closed-form solution of Problem (2.16)) – *Let $\boldsymbol{X}_0$ be any matrix that belongs to the constraints set (2.6a), and $\boldsymbol{M} = (\boldsymbol{X}_0^\intercal\boldsymbol{Y}\boldsymbol{H}^\intercal)_{2:,2:}$ the submatrix containing everything but the input's first row and first column. Finally, let $\boldsymbol{P}\boldsymbol{D}\boldsymbol{Q}^\intercal$ be the SVD of $\boldsymbol{M}$. Then, the problem admits the following closed form solution:*

$$\widehat{\boldsymbol{X}} = \boldsymbol{X}_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\intercal \\ \mathbf{0}_{N-1} & \boldsymbol{P}\boldsymbol{Q}^\intercal \end{bmatrix} \ . \tag{2.17}$$

In practice, $\boldsymbol{X}_0$ can be fixed to the current value of $\boldsymbol{X}$.

5.2 Optimization with respect to $\boldsymbol{\Lambda}$

With respect to $\boldsymbol{\Lambda}$, the optimization Problem (2.6) becomes:

$$\min_{\boldsymbol{\Lambda}} \ \alpha \ \underbrace{\text{tr}(\boldsymbol{H}\boldsymbol{H}^\intercal\boldsymbol{\Lambda})}_{\|\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\|_F^2} \qquad \text{s.t.} \quad \begin{cases} (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal)_{i,j} \leq 0 \quad i \neq j \ , & \text{(b)} \\ \boldsymbol{\Lambda} = \text{diag}(0, \lambda_2, \dots, \lambda_N) \succeq 0 \ , & \text{(c)} \\ \text{tr}(\boldsymbol{\Lambda}) = N \in \mathbb{R}_*^+ \ , & \text{(d)} \end{cases} \tag{2.18}$$

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 2.2.

5.3 Computational complexity of FGL-3SR

Considering a graph with N nodes and n graph signals:

Algorithm 2.2 The FGL-3SR algorithm with $\ell_{2,1}$-norm

Input : $\boldsymbol{Y} \in \mathbb{R}^{N \times n}$, α, β
Output : $\widehat{\boldsymbol{H}}$, $\widehat{\boldsymbol{X}}$, $\widehat{\boldsymbol{\Lambda}}$
Initialization: $\boldsymbol{L}_0$ (e.g. with a complete graph) – see Section 4.4
$(\boldsymbol{X}, \boldsymbol{\Lambda}) \longleftarrow \text{SVD}(\boldsymbol{L}_0)$
for $t = 1, 2, \ldots$ **do**
  ▷ H-step: Compute the closed-form solution of Proposition (2.4)
  for $1 = 1, \ldots, N$ **do**
    $$\boldsymbol{H}_{i,:} \longleftarrow \frac{1}{1+\alpha\lambda_i}\Big(1 - \frac{\beta}{2}\frac{1}{\|(\boldsymbol{X}^\top\boldsymbol{Y})_{i,:}\|_2}\Big)_+ (\boldsymbol{X}^\top\boldsymbol{Y})_{i,:}$$
  end for
  ▷ X-step: Compute the closed-form solution of Proposition (2.7)
  $\boldsymbol{M} \longleftarrow (\boldsymbol{X}^\top\boldsymbol{Y}\boldsymbol{H}^\top)_{2:,2:}$
  $(\boldsymbol{P}, \boldsymbol{D}, \boldsymbol{Q}^\top) \longleftarrow \text{SVD}(\boldsymbol{M})$
  $$\boldsymbol{X} \longleftarrow \boldsymbol{X}\begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\top \\ \boldsymbol{0}_{N-1} & \boldsymbol{P}\boldsymbol{Q}^\top \end{bmatrix}$$
  ▷ Λ-step: Solve the linear Program (2.18)
  $$\boldsymbol{\Lambda} \longleftarrow \arg\min_{\boldsymbol{\Lambda}} \alpha\,\text{tr}(\boldsymbol{H}\boldsymbol{H}^\top\boldsymbol{\Lambda}) \qquad \text{s.t.} \quad \begin{cases} (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\top)_{i,j} \leq 0 \quad i \neq j \\ \boldsymbol{\Lambda} = \text{diag}(0, \lambda_2, \ldots, \lambda_N) \succeq 0 \\ \text{tr}(\boldsymbol{\Lambda}) = N \in \mathbb{R}_*^+ \end{cases}$$
end for

- H-step (non-iterative) – The closed-form solution requires to compute the matrix product $\boldsymbol{X}^\top\boldsymbol{Y}$, which is of complexity $\mathcal{O}(nN^2)$.
- X-step (non-iterative) – The closed-form solution requires to compute the SVD of $(\boldsymbol{X}_0^\top\boldsymbol{Y}\boldsymbol{H}^\top)_{2:,2:} \in \mathbb{R}^{(N-1)\times(N-1)}$, which is of complexity $\mathcal{O}(N^3)$ [Cline and Dhillon, 2006].
- Λ-step – Solving the LP can be done with interior-point methods or with the ellipsoid method [Vandenberghe, 2010]. 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 [Bubeck, 2015].

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 [Sardellitti et al., 2019]) relies on a semi-definite program and is of order at least $\mathcal{O}(N^8)$ (see Section 7). As will be clearly demonstrated in Section 8, in practice the empirical execution time of FGL-3SR is lower than IGL-3SR and ESA-GL.

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

The two proposed algorithms are based on a modification of the initial optimization problem (2.6). Indeed, both of them relax the constraint (2.6*b*), $\forall k \neq \ell,\ (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\top)_{k,\ell} \leq 0$, but with 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 (2.6*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 keep the set of variables in the initial set of constraints, essential for the matrix $\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal$ to be a proper Laplacian. On the other hand, FGL-3SR does not use a log-barrier function to relax the constraint (2.6*b*), but instead, removes it at the $\boldsymbol{X}$-step. Recall that we are able to do that because we know from Lemma 2.6 that for any $\boldsymbol{X}$ returned by the X-step (5.1), there exist a $\boldsymbol{\Lambda}$ making $\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal$ a Laplacian. This relaxation has the advantage to drastically speed-up the X-step while loosing the convergence property and the decreasing over the initial constraints set.

6 A probabilistic interpretation

In this section, we introduce a new representation model adapted to smooth graph signals with sparse spectral representation. The goal of this model is to provide a probabilistic interpretation of Problem (2.6) and link its objective function to a maximum a posteriori estimation (Proposition 2.8).

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

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{h} + \boldsymbol{m}_y + \boldsymbol{\varepsilon} \ , \tag{2.19}$$

where $\boldsymbol{m}_y \in \mathbb{R}^N$ is the mean of the graph signal $\boldsymbol{y}$ and $\boldsymbol{\varepsilon}$ is a Gaussian noise with zero mean and covariance $\sigma^2 \boldsymbol{I}_N$. Here, the latent variable $\boldsymbol{h} \in \mathbb{R}^N$ controls $\boldsymbol{y}$ through the eigenvector matrix $\boldsymbol{X}$ of $\boldsymbol{L}$. The choice of the representation matrix $\boldsymbol{X}$ is particularly adapted since it reflects the topology of the graph and provides a spectral embedding of its vertices. Moreover, as seen in Section 3, $\boldsymbol{X}$ can be interpreted as a graph Fourier basis, which makes it an intuitive choice for the representation matrix. In a noiseless scenario with $\boldsymbol{m}_y = \boldsymbol{0}_N$, $\boldsymbol{h}$ actually corresponds to the GFT of $\boldsymbol{y}$.

To comply with the spectral sparsity assumption (Assumption 2.2), we now propose a distribution that allows $\boldsymbol{h}$ 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 $\boldsymbol{h}$ is

$$\boldsymbol{h}|\gamma \sim \mathcal{N}(\boldsymbol{0}_N, \widetilde{\boldsymbol{\Lambda}}^\dagger) \ , \tag{2.20}$$

where $\widetilde{\boldsymbol{\Lambda}}^\dagger$ is the Moore-Penrose pseudo-inverse of the diagonal matrix containing the values $\{\lambda_i \cdot \mathbb{1}\{\gamma_i = 1\}\}_{i=1}^N$. 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 $\boldsymbol{h}_i$ 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 $\boldsymbol{h}_i$ concentrates more around 0, leading to small value of $\lambda_i \boldsymbol{h}_i^2$. 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 $\lambda_1 = 0$, $\boldsymbol{h}_1$ follows a centered degenerate Gaussian, i.e $\boldsymbol{h}_1$ 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. [2016], which was only focused on the smoothness assumption.

Definition 2.13. (Prior and conditional distributions) – *The following equations summarize the prior and important conditional distributions of our model:*

$$p(\boldsymbol{h}_i|\gamma_i,\lambda_i) \propto \exp(-\lambda_i \boldsymbol{h}_i^2)\cdot \mathbb{1}\{\gamma_i=1\} + \mathbb{1}\{\boldsymbol{h}_i=0,\gamma_i=0\}\ , \tag{2.21}$$

$$p(\boldsymbol{y}|\boldsymbol{h},\boldsymbol{X}) \propto \exp(-\frac{1}{\sigma^2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{h}-\boldsymbol{m}_y\|_2^2)\ , \tag{2.22}$$

$$p(\gamma_i) \propto p_i^{\gamma_i}(1-p_i)^{1-\gamma_i}\ . \tag{2.23}$$

For simplicity, in the following we consider that $\boldsymbol{m}_y = \boldsymbol{0}_N$ and $p_1 = 0$.

Lemma 2.2. – *Assume the proposed Model (2.19). If $p_1 = 0$ and $p_i \in (0,1)$, $\forall i \geq 2$, then:*

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

Definition 2.14. (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 2.8. (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:*

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

This proposition tells us that *for a given Laplacian matrix, the maximum a posteriori estimate of $\boldsymbol{h}$ would corresponds to the minimum of Problem (2.6).*

7 Related work on GSP-based graph learning methods

We now 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 8.

1. **GL-SigRep** [Dong et al., 2016]**:**
   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:

$$\min_{\boldsymbol{L},\tilde{\boldsymbol{Y}}}\|\boldsymbol{Y}-\tilde{\boldsymbol{Y}}\|_F^2 + \alpha\|\boldsymbol{L}^{1/2}\tilde{\boldsymbol{Y}}\|_F^2 + \beta\|\boldsymbol{L}\|_F^2 \quad \text{s.t.} \quad \begin{cases}\boldsymbol{L}_{k,\ell} = \boldsymbol{L}_{\ell,k} \leq 0 \quad k\neq\ell\ , \\ \boldsymbol{L}\boldsymbol{1} = \boldsymbol{0}_N\ , \\ \mathrm{tr}(\boldsymbol{L}) = N \in \mathbb{R}_*^+\ .\end{cases} \tag{2.24}$$

   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 (2.24) consists in an alternating minimization over $\boldsymbol{L}$ and $\tilde{\boldsymbol{Y}}$. With respect to $\tilde{\boldsymbol{Y}}$ the problem has a closed-form solution whereas for $\boldsymbol{L}$, 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.

2. **ESA-GL** [Sardellitti et al., 2019]**:**
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 work is two-fold. First, ESA-GL does not include the smoothness assumption while learning the Fourier basis $\boldsymbol{X}$. 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 5.3), and hence is prohibitive for large graphs. The first step consists in fitting an orthonormal basis such that the observed graph signals $\boldsymbol{Y}$ admit a sparse representation with respect to this basis. They consider the problem

$$\min_{\boldsymbol{H},\boldsymbol{X}} \|\boldsymbol{Y}-\boldsymbol{X}\boldsymbol{H}\|_F^2 \qquad \text{s.t.} \quad \begin{cases} \boldsymbol{X}^\intercal\boldsymbol{X}=\boldsymbol{I}_N,\ \boldsymbol{x}_1=\frac{1}{\sqrt{N}}\mathbf{1}_N\ , \\ \|\boldsymbol{H}\|_{2,0}\leq K\in\mathbb{N}\ , \end{cases} \tag{2.25}$$

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

$$\min_{\boldsymbol{L}\in\mathbb{R}^{N\times N},\boldsymbol{C}_K\in\mathbb{R}^{K\times K}} \operatorname{tr}(\widehat{\boldsymbol{H}}_K^T\boldsymbol{C}_K\widehat{\boldsymbol{H}}_K)+\mu\|\boldsymbol{L}\|_F^2 \quad \text{s.t.} \quad \begin{cases} \boldsymbol{L}_{k,\ell}=\boldsymbol{L}_{\ell,k}\leq 0 \quad k\neq\ell\ , \\ \boldsymbol{L}\mathbf{1}_N=\mathbf{0}_N\ , \\ \boldsymbol{L}\widehat{X}_K=\widehat{\boldsymbol{X}}_K\boldsymbol{C}_K, \quad \boldsymbol{C}_K\succeq 0\ , \\ \operatorname{tr}(\boldsymbol{L})=N\in\mathbb{R}_*^+\ , \end{cases} \tag{2.26}$$

where $\boldsymbol{C}_K\in\mathbb{R}^{K\times K}$ and $\widehat{\boldsymbol{X}}_K$ corresponds to the columns of $\widehat{\boldsymbol{X}}$ associated to the non-zero rows of $\widehat{\boldsymbol{H}}$ denoted $\widehat{\boldsymbol{H}}_K$. Thus, the second step aims at estimating a Laplacian that enforces the smoothness of the learned signal representation $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{H}}$. 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 $\boldsymbol{C}_K$, the readers shall refer to the aforementioned paper.

8 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 8.2 and 8.3 use synthetic data for first comparing the different methods and then study the influence of the hyperparameters; Section 8.4 displays several examples on real-world data.

All experiments were conducted on a personal laptop with 4-core 2.5GHz Intel CPUs and Linux/Ubuntu OS. For the $\boldsymbol{\Lambda}$-step of both algorithms, we use the Python's CVXPY package [Diamond and Boyd, 2016]. For the $\boldsymbol{X}$-step of IGL-3SR, we use the conjugate gradient descent solver combined with an adaptive line search, both provided by Pymanopt [Townsend et al., 2016], a Python toolbox for optimization on manifolds. Note that this package only requires the gradients given in Proposition 2.5. The source code of our implementations is available at https://github.com/pierreHmbt/GL-3SR.

8.1 Evaluation metrics

We provide visual and quantitative comparisons of the learned Laplacian $\widehat{\boldsymbol{L}}$ and its weight matrix $\widehat{\boldsymbol{W}}$ using the performance measures: *Recall*, *Precision*, and F_1-*measure*, which are standard for this type of evaluation [Pasdeloup et al., 2017]. 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 Pasdeloup et al. [2017], the F_1-measure is computed on a thresholded version of the estimated weight matrix $\widehat{\boldsymbol{W}}$. This threshold is equal to the average value of the off-diagonal entries of $\widehat{\boldsymbol{W}}$ (same process as in [Sardellitti et al., 2019]).

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

$$\rho(\boldsymbol{L}, \widehat{\boldsymbol{L}}) = \frac{\sum_{ij}(\boldsymbol{L}_{ij} - L_m)(\widehat{\boldsymbol{L}}_{ij} - \widehat{\boldsymbol{L}}_m)}{\sqrt{\sum_{ij}(\boldsymbol{L}_{ij} - \boldsymbol{L}_m)^2}\sqrt{\sum_{ij}(\widehat{\boldsymbol{L}}_{ij} - \widehat{\boldsymbol{L}}_m)^2}} \, , \tag{2.27}$$

where $\boldsymbol{L}_m$ and $\widehat{\boldsymbol{L}}_m$ are the average values of the entries of the true and estimated Laplacian matrices, respectively. This $\rho(\cdot)$ function evaluates the quality of the weights distribution over the edges.

8.2 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 (2.21) that we presented in Section 6. 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 $\widehat{\boldsymbol{W}}$, as explained in Section 8.1.

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 2.4. Indeed, for $\ell_{2,1}$-norm, the sparsity of $\widehat{\boldsymbol{H}}$ is only controlled by β (Equation (2.12)). On the contrary, when using the $\ell_{2,0}$-norm, the value of α also influences the sparsity (Equation (2.11)). 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 8.3.1, easier to tune.

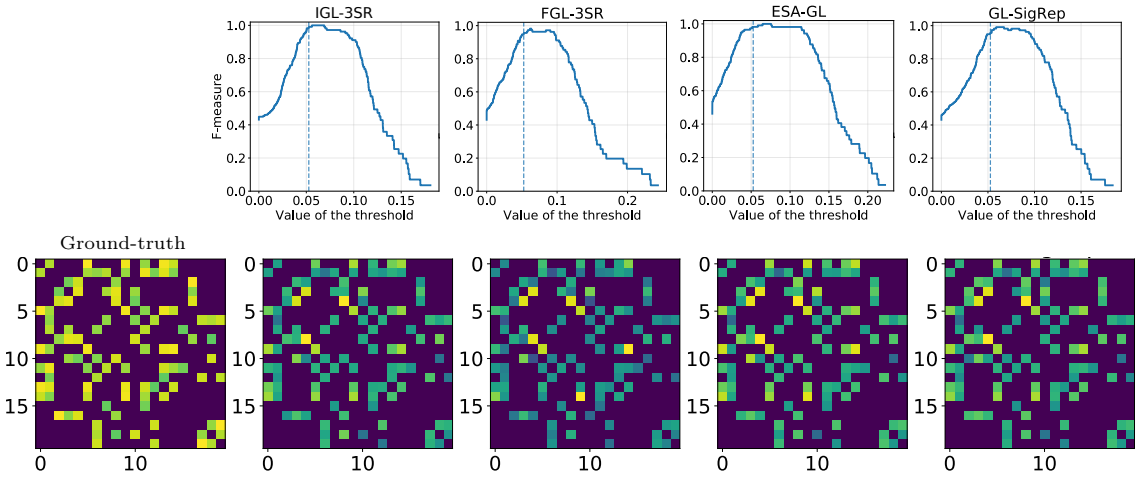
		RG graph model				*ER graph model*			
N	**Metrics**	**IGL-3SR**	**FGL-3SR**	**ESA-GL**	**GL-SigRep**	**IGL-3SR**	**FGL-3SR**	**ESA-GL**	**GL-SigRep**
20	Precision ↑	**0.973** (±**0.042**)	0.952 (±0.042)	0.899 (±0.054)	0.929 (±0.068)	**0.952** (±**0.045**)	0.819 (±0.080)	0.931 (±0.045)	0.704 (±0.125)
	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 (±00.028)	0.816 (±0.068)	**0.941** (±**0.038**)	0.779 (±0.071)
	$\rho(\boldsymbol{L}, \overline{\boldsymbol{L}})$ ↑	**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)
	$\rho(\boldsymbol{L}, \overline{\boldsymbol{L}})$ ↑	**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)	–
	$\rho(\boldsymbol{L}, \overline{\boldsymbol{L}})$ ↑	**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 2.1: 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.

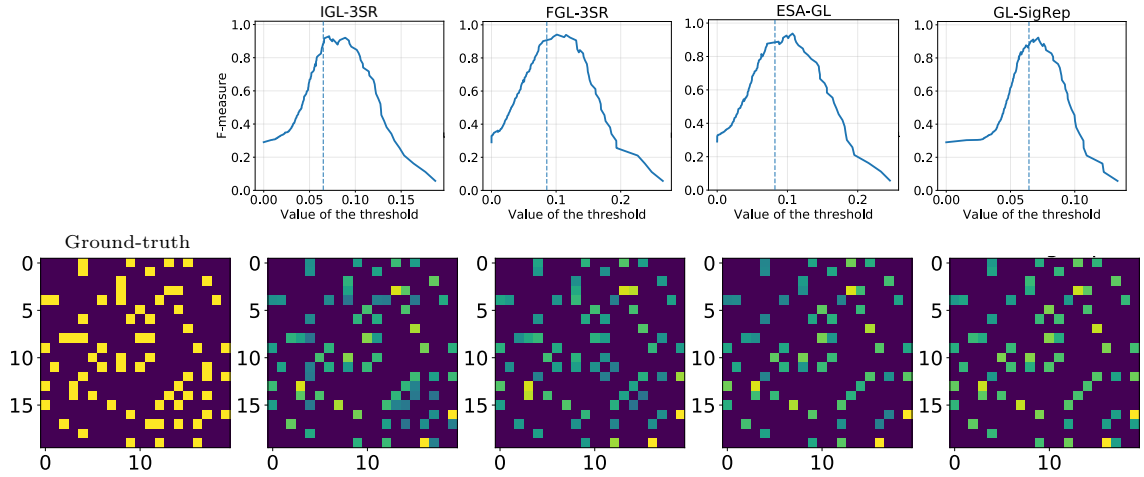
Quantitative results. Average evaluation metrics and their standard deviation are collected in Table 2.1. The results show that the use of the sparsity constraint improves the quality of the learned graphs. Indeed, the two proposed methods IGL-3SR and FGL-3SR, as well as ESA-GL, have better overall performance 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 2.7 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 2.1, indicate that in this sampling process the proposed FGL-3SR method managed to infer accurate graphs despite the relaxation.

Speed performance. Figure 2.8 displays the evolution of the empirical computation time as the number of nodes increases. For each algorithm, time per iteration is: i) for IGL-3SR and FGL-3SR, the time needed for the computation of the 3 steps one time; ii) for ESA-3SR,the time needed for the computation of the quadratic program; iii) for GL-SigRep, the time needed for the computation of its two steps one time. 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.



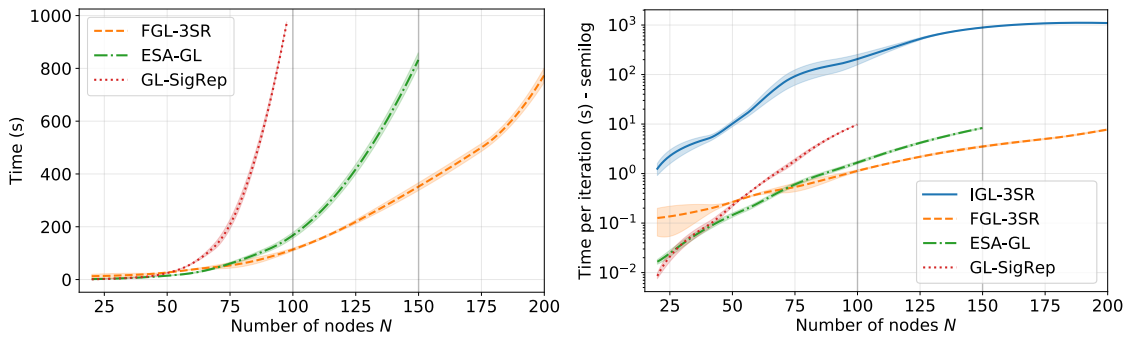
(a) Graph learning on RG synthetic graphs.



(b) Graph learning on ER synthetic graphs.

Figure 2.7: 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.

IGL-3SR v.s. FGL-3SR. In terms of numerical performance, IGL-3SR is better than FGL-3SR (Table 2.1). Indeed, except for graphs of size 20, metrics relative to the recovery of the true graph give better results. On the contrary, in terms of computational time aspect FGL-3SR is better than IGL-3SR (see Figure 2.8). Indeed, no matter the size of the graph, FGL-3SR has a time per iteration lower than IGL-3SR. This is due to the fact that contrary to IGL-3SR which solves two out of three sub-problems with iterative methods, FGL-3SR solves two sub-problems via closed-form solutions which are efficiently computable. In conclusion, when the number of nodes is small, IGL-3SR is preferred. If not, one should use FGL-3SR.



(a) Standard scale. **(b)** Semi-log scale.

Figure 2.8: 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.

8.3 Influence of the hyperparameters

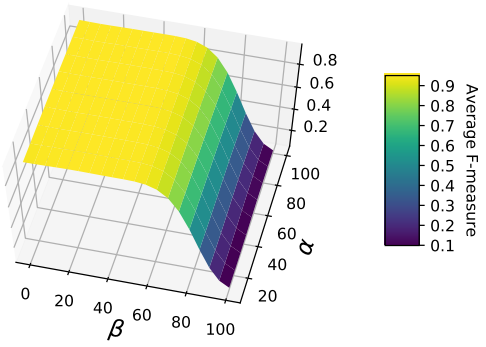
We now study how hyperparameters of IGL-3SR and FGL-3SR influence their overall performance, with respect to the F_1-measure. This study is made on a RG graph with $N = 20$ nodes and 10-bandlimited signals $\boldsymbol{Y}$ in $\mathbb{R}^{20\times 1000}$.

8.3.1 Influence of α and β

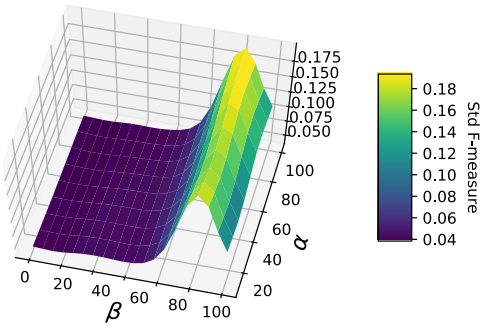
We first highlight the influence of α and β on FGL-3SR. We run and collect the F_1-measure for 20 values of α (resp. β) in $[10^{-5}, 100]$ (resp. $[10^{-5}, 60]$). The resulting heatmaps are displayed in Figure 2.9. The most important observation is that the value of α does not seem to impact the quality 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 2.10) 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 $\widehat{\boldsymbol{H}}$. 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.

8.3.2 Influence of t

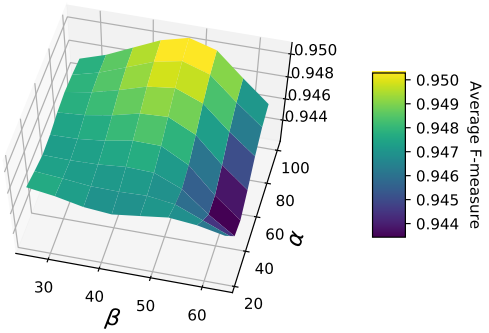
We now highlight the influence of t on IGL-3SR. Figure 2.11 shows the learned graphs for several values of $t \in [10,\ 10^4]$. This experiment brings two main messages: first, when t is too low, the learned graph is very close to the complete graph, whereas when t increases the learned graph becomes 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 objective function during the $\boldsymbol{\Lambda}$-step of Problem (2.8) is equivalent to the minimization of $\operatorname{tr}(\boldsymbol{H}\boldsymbol{H}^\top\boldsymbol{\Lambda}) + \frac{1}{\alpha t}\phi(\boldsymbol{U}, \boldsymbol{\Lambda})$.



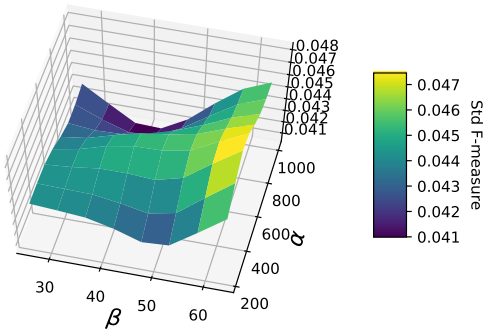
(a) Average of the F_1-measure.



(b) Standard deviation of the F_1-measure.

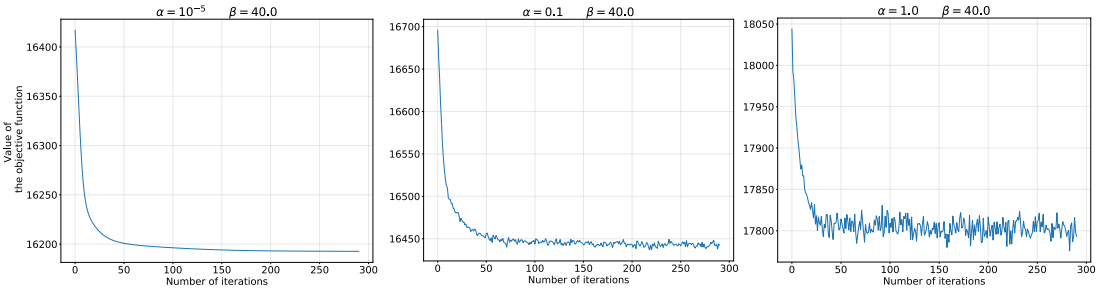


(c) Average of the F_1-measure (focus).



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

Figure 2.9: 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 2.10: 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) $\alpha = 1$.

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 [Boyd and Vandenberghe, 2004]. 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 [Boyd and Vandenberghe, 2004].

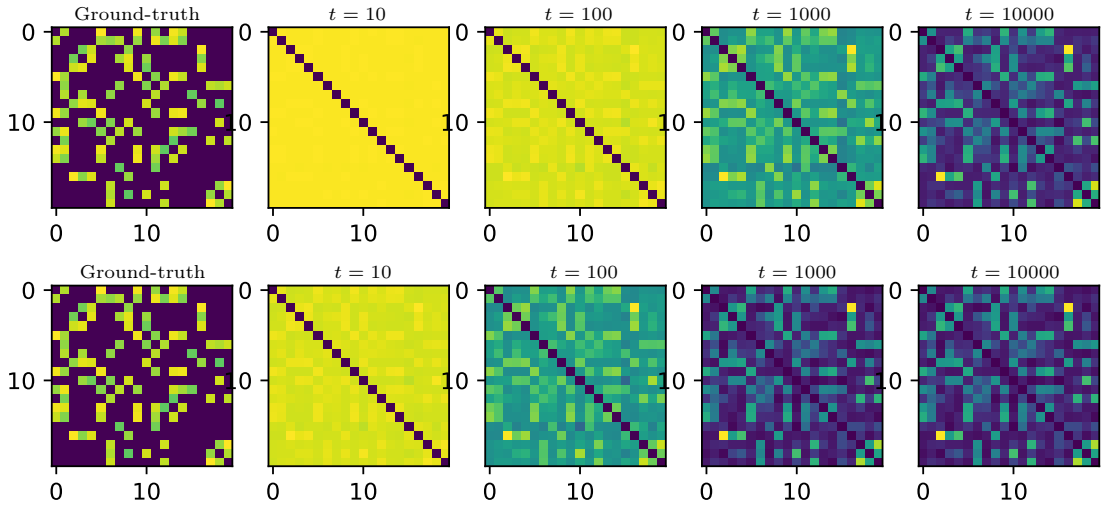


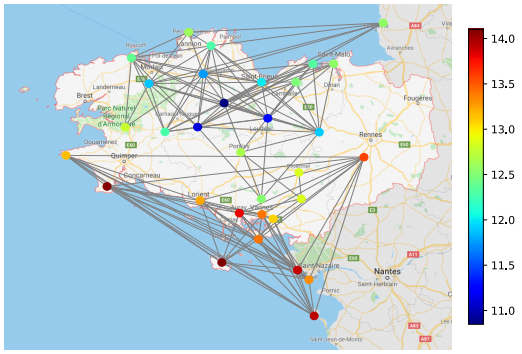
Figure 2.11: Learned graphs with increasing t values: (top row) $\alpha = 10^{-4}$, (bottom row) $\alpha = 10^{-3}$.

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 2.10). 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 4.4, can be used to tune t [Boyd and Vandenberghe, 2004]. Hence, according to our experiments, it remains only β as a critical hyperparameter to tune for both these methods. Based on Figure 2.9, 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.

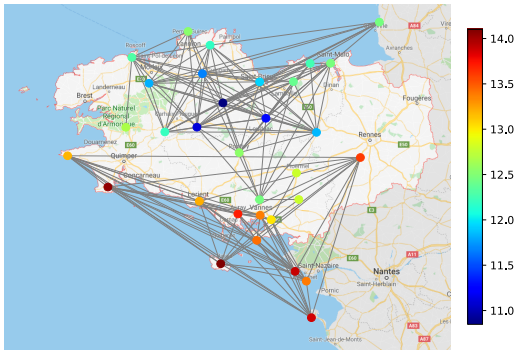
8.4 Temperature data

We used hourly temperature (C°) measurements on 32 weather stations in Brittany, France, during a period of 31 days [Chepuri et al., 2017]. The dataset contains $24 \times 31 = 744$ multivariate observations, i.e. $\boldsymbol{Y} \in \mathbb{R}^{32 \times 744}$, 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.

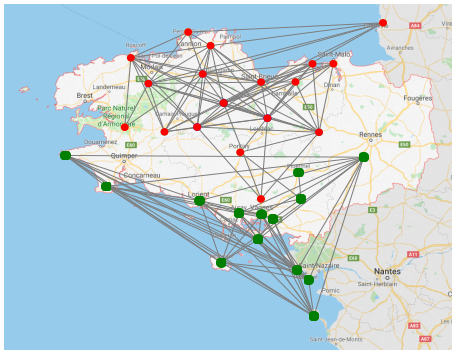
The graphs obtained with each of the method are displayed in Figure 2.12 (a-b). They are in accordance with the one found in Chepuri et al. [2017] on the same dataset. Both the proposed methods provide similar results, which shows that the relaxation used in FGL-3SR has a moderate influence in practice 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 [Ng et al., 2001] 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 of the region of Britanny (Figure 2.12 (c-d)), which is an expected natural segmentation.



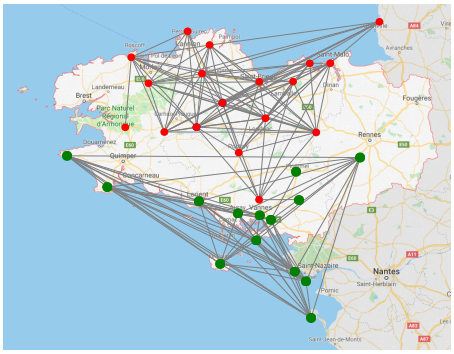
(a) Learned graph by IGL-3SR.



(b) Learned graph by FGL-3SR.



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



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

Figure 2.12: (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.

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 2.13 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.

8.5 Cancer genome data

In this second experiment, we consider the RNA-Seq Cancer Genome Atlas Research Network dataset [Weinstein et al., 2013]. The data set contains the information of 801 individuals, each of them characterized by a set of 20531 genetic features and being labeled by one out of 5 types of cancer: breast carcinoma (BRCA), colon adenocarcinoma (COAD), kidney renal clear-cell carcinoma (KIRC), lung adenocarcinoma (LUAD), and prostate adenocarcinoma (PRAD).

The goal of the considered task is to learn a graph of the $N = 801$ individuals (the nodes) using the $n = 20531$ genetic features (the samples seen here as graph signals) and determine if this graph is able to group the individuals according to their tumor type. The number of nodes being

(a) **(b)**

Figure 2.13: (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).

large, we propose to use FGL-3SR and, as previously, to use spectral clustering [Ng et al., 2001] on the learned graph the find the cluster mapping.

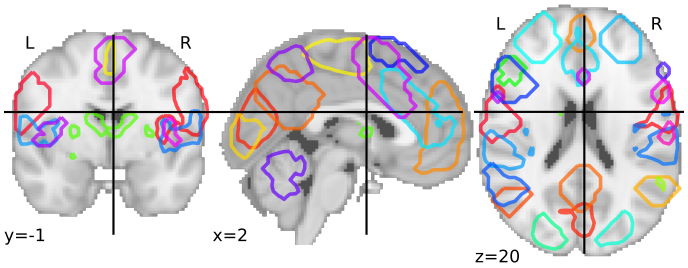
As the number of nodes of the graph is too large, ESA-GL and Sig-Rep are not able to run in reasonable time. Therefore, we compare FGL-3SR to two other state-of-the-art methods, which are however not GSP-oriented but rather specialized to obtain a graph that facilitates data clustering. The two competitors are namely the Constrained Laplacian Rank (CLR) algorithm [Nie et al., 2016] that builds a special graph from the available data, and the Structured Graph Learning (SGL) algorithm [Kumar et al., 2019, 2020] that take as input the sample covariance matrix of the data. As quality measure we use the clustering accuracy, which has also been used in the associated papers of the competitors from where we obtain their reported results. The results for the three methods are respectively:

$$\text{FGL-3SR: } 0.9887, \qquad \text{CLR: } 0.9862, \qquad \text{SGL: } 0.9987.$$

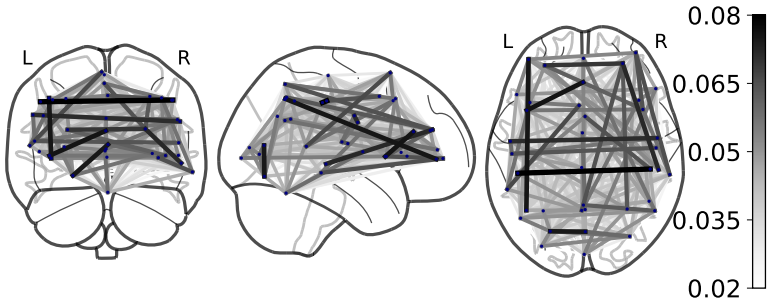
The first interesting result is that FGL-3SR presents similar accuracy to CLR and SGL, even though it is not a graph learning method specially designed for clustering like the competitors. Secondly, while FGL-3SR comes second in terms of accuracy after SGL, two important remarks need to be made about the SGL method: 1) it must fix the right number of clusters of the learned graph a priori to obtain such result; 2) it has an additional hyperparameter to tune compared to FGL-3SR. Therefore comparably, bearing in mind the above results, the fact that SGL is fine-tailored for the undertaken clustering tasks and that it has higher tuning complexity, and finally the limitations of ESA-GL and Sig-Rep that prevent them from being applied in this scenario, FGL-3SR seems to be a promising alternative for large-scale graph-based learning applications.

8.6 Results on the ADHD dataset

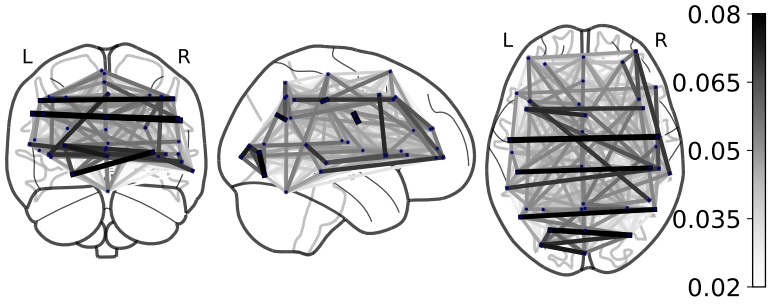
In this third experiment, we consider the Attention Deficit Hyperactivity Disorder (ADHD) dataset [Bellec et al., 2017] composed of functional Magnetic Resonance Imaging (fMRI) data. ADHD is a mental pathophysiology characterized by an excessive activity [Boyle et al., 2011]. We study the resting-state fMRI of 20 subjects with ADHD and 20 healthy subjects available from Nilearn [Abraham et al., 2014]. Each of the 40 fMRI consists in a series of images measuring the brain activity. These images are processed as follows. We split the brain into $N = 39$ Regions Of Interest (ROIs) with the Multi-Subject Dictionary Learning atlas [Varoquaux et al., 2011] (see Figure 2.14a). Each ROI defines a node and the signal value at a certain node is the aggregation of



(a) Indicative Regions of Interest (ROIs) from Varoquaux et al. [2011].



(b) ADHD subject.



(c) Healthy subject.

Figure 2.14: (a) Indicative ROIs from the Multi-Subject Dictionary Learning atlas extracted in Varoquaux et al. [2011] 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.

the pixel values over the associated ROI. Each image is thereby transformed into a graph signal. For each of the 40 subjects, we therefore have access to a matrix in $\mathbb{R}^{n\times 39}$, where n is the number of images in the fMRI of the subject (i.e. the number of graph signals).

We propose to estimate a graph for each subject independently. Examples of learned graphs with FGL-3SR for an ADHD subject and a healthy subject are displayed in Figure 2.14. 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 [Damoiseaux et al., 2006; Smith et al., 2009]. 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 [Abraham et al., 2017; Dadi et al., 2019]. 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 (2.27) 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 [Abraham et al., 2017] 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., 2018] 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.

9 Electroencephalography microstates analysis through graphs

In this preliminary experiment, we are interested in multichannel ElectroEncephaloGraphy (EEG). EEG is an important method to access real-time information about the global function of neuronal networks. Traditionally, its analysis relies on the study of the different frequency bands present in each channel. However, due to the time-frequency uncertainty principle, studying the frequency inherently sacrifices the high temporal resolution of the EEG. To overcome this issue and analyze short-lasting fluctuations of neuronal activity, several methods in the time domain have been proposed. These methods are based on the seminal work of Lehmann et al. [1987] which first considers the temporal evolution of the topography of the scalp electric field instead of the frequencies. With this idea, they obtain a global measure of the brain activity with high temporal resolution. They show that the topography does not change randomly and continuously over time but instead remains stable for 80 to 120 milliseconds. These periods of quasi-stability are termed *EEG microstates* and are a window to better understand the behavior of the brain activity (see e.g. [Musso et al., 2010; Van de Ville et al., 2010]). A full review and introduction can be found in [Michel and Koenig, 2018] and [Poulsen et al., 2018]. In this experiment, we aim to study the microstates during a general anesthesia. For a better understanding of these microstates, we first extract them from the EEG signals and then learn their underlying graphs with FGL-3SR.

Dataset. The data consists in 32 EEG signals recorded at 250 Hz during a General Anesthesia (GA) for 10 patients. Signals are first filtered using a bandpass filter between 1 and 20Hz, to remove the potential drift below 1Hz, and to keep the frequencies below 20Hz that characterize GA [Brown et al., 2010]. We also remove some artifacts using Independent Component Analysis (ICA) and set the reference to average – an important parameter to study microstates. Finally, for each patient, we crop the signals to only keep times relative to the "Anesthesia state" [Brown et al., 2010].

Segment the signals and graph learning of the microstates. EEG microstate segmentation is performed based on a standard procedure. The local maxima of the global field power (GFP) [Lehmann and Skrandies, 1980] are extracted from the EEG. Then, several runs of the *modified K-means* algorithm are performed [Pascual-Marqui et al., 1995], using different random initializations. The run resulting in the best segmentation, as measured by the Global Explained Variance (GEV) [Poulsen et al., 2018] is kept. Through this segmentation, we extract and group the similar temporal parts of the EEG, hence satisfying the i.i.d. hypothesis of the factor analysis model introduced in Section 6. Furthermore, as the topography remains stable during a few milliseconds,

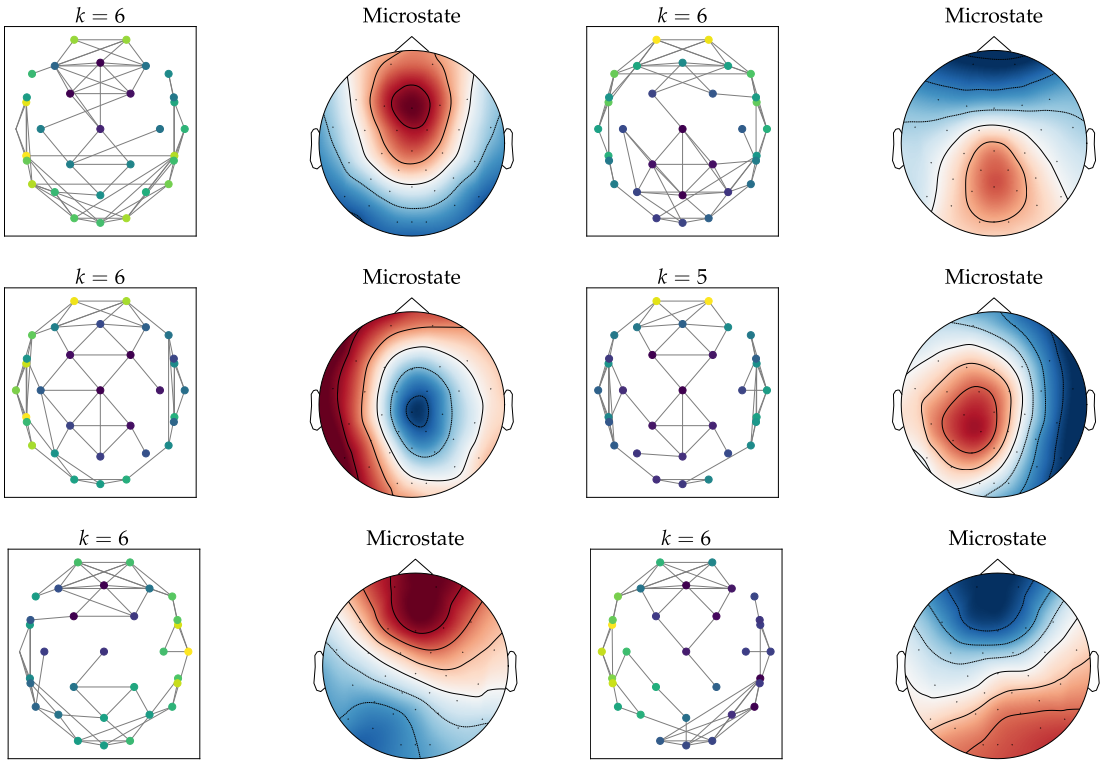


Figure 2.15: Left: The learned graph with k being the sparsity obtained. Middle: The topography of the microstate. Right: The topography of the second eigenvector of the learned graph.

the assumption that the signal is stationary during this short period of time is valid. We then learn with FGL-3SR the structure of each found microstate using the signals relative to each cluster. Hyperparameters are set in order to obtain visually relevant graphs.

Results. We extract 6 microstates from the signal and learn their associated graphs (see Figure 2.15). They explain 80% of the variance i.e. GEV equals 0.80 and are consistent with those of Shi et al. [2020]. All graphs returned by FGL-3SR are structured, well reflecting the topology of microstates. This underlines the pertinence of our approach for the visualization and the study of microstates with graphs. One other advantage of this approach is that, unlike standard methods, graphs allow us to finely analyze the spatial relationships between the EEG channels (the links between the nodes). In other terms, we do not only analyze the topography, which is a partial information (averaging). Thus, instead of only comparing the topology of microstates, we can compare their structure via appropriate metrics on graphs [Maretic et al., 2019]. Furthermore, these graphs allow to apply a wide range of other useful methods such as filtering, sampling, spectral clustering, blind source separation of graph signals [Miettinen et al., 2020], and even computation of similarity between the topography using the Wasserstein distance between the underlaying graphs [Maretic et al., 2019]. Note that for the 6 learned graph, the hyperparameter β was set to the same value and led to the same sparsity for $\boldsymbol{H}$. To conclude, we believe that the graph representation may allow a thinner analysis of the structure of the microstates during anesthesia.

10 Conclusion

This chapter presented a data-driven graph learning approach by employing a combination of two assumptions. The first is standard in the related literature and concerns the *smoothness* of graph signals with respect to the underlying graph structure. The second is the *spectral sparsity* assumption, a consequence of 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 and subsampling, etc. Finally, by including the two assumptions in a probabilistic model, we link our optimization problem to a maximum a posteriori estimation and pave the way for further statistical understanding.

11 Technical proofs

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

$\boldsymbol{x}^\intercal, \boldsymbol{M}^\intercal$	Transpose of vector $\boldsymbol{x}$, matrix $\boldsymbol{M}$.
$\text{tr}(\boldsymbol{M})$	Trace of matrix $\boldsymbol{M}$.
$\text{diag}(\boldsymbol{x})$	Diagonal matrix containing the vector $\boldsymbol{x}$.
$\boldsymbol{M}_{k,l}$	(k,l)-th element of the matrix $\boldsymbol{M}$.
$\boldsymbol{M}_{k,:}$	k-th row of $\boldsymbol{M}$.
$\boldsymbol{M}_{:,l}$	l-th column of $\boldsymbol{M}$.
$\boldsymbol{M}_{k:,l:}$	Submatrix containing the elements of $\boldsymbol{M}$ from the k-th row to the last row, and from the l-th column to the last column.
$\boldsymbol{M} \succeq 0$	$\boldsymbol{M}$ is a positive semi-definite matrix.
$\boldsymbol{M}^\dagger$	The Moore-Penrose pseudoinverse of $\boldsymbol{M}$.
$\boldsymbol{e}_k$	Vector containing zeros except a 1 at position k.
$\boldsymbol{I}_n$	Identity matrix of size n.
$\boldsymbol{0}_n$	Vector of size n containing only zeros.
$\boldsymbol{1}_n$	Vector of size n containing only ones.
$\mathbb{1}_{\mathcal{A}}(\cdot)$	The indicator function over the set $\mathcal{A}$.
$\|\boldsymbol{x}\|_0$	The number of non-zero elements of a vector $\boldsymbol{x}$.
$\|\cdot\|_F$	The Frobenius norm.
$\|\cdot\|_{2,0}$	The $\ell_{2,0}$-norm, with $\|\boldsymbol{M}\|_{2,0} = \sum_{i=1} \mathbb{1}_{\{\|\boldsymbol{M}_{i,:}\|_2 \neq 0\}}$.
$\|\cdot\|_{2,1}$	The $\ell_{2,1}$-norm, with $\|\boldsymbol{M}\|_{2,1} = \sum_{i=1} \|\boldsymbol{M}_{i,:}\|_2$.
∇f	Gradient of the function f.
$\langle \cdot, \cdot \rangle$	Inner product function.
$Orth(N)$	The set of all orthogonal matrices of size $N \times N$.

Table 2.2: Table of notations used throughout the chapter.

Lemma 2.1 – *Given $\boldsymbol{X}, \boldsymbol{X}_0 \in \mathbb{R}^{N\times N}$ two orthogonal matrices with first column equals to $\frac{1}{\sqrt{N}}\mathbf{1}_N$ (constraint $(2.6a)$), we have the following equality:*

$$\boldsymbol{X} = \boldsymbol{X}_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\intercal \\ \mathbf{0}_{N-1} & [\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:} \end{bmatrix} ,$$

with $[\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:}$ denoting the submatrix of $\boldsymbol{X}_0^\intercal \boldsymbol{X}$ containing everything but the first row and column of itself. Furthermore, remark that $[\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:}$ is in $Orth(N-1)$.

Proof. Let consider $\boldsymbol{X}, \boldsymbol{X}_0 \in \mathbb{R}^{N\times N}$ two orthogonal matrix with first column equals to $\frac{1}{\sqrt{N}}\mathbf{1}_N$. We have the following equalities:

$$\boldsymbol{X}_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\intercal \\ \mathbf{0}_{N-1} & [\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:} \end{bmatrix} = \begin{bmatrix} \vdots & \vdots \\ \boldsymbol{X}_{0(:,1)} & \boldsymbol{X}_{0(:,2:)}[\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:} \\ \vdots & \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots \\ \frac{1}{\sqrt{N}}\mathbf{1}_N & \boldsymbol{X}_{:,2:} \\ \vdots & \vdots \end{bmatrix} = \boldsymbol{X} .$$

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

$$[\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:} \left[[\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:}\right]^\intercal = \boldsymbol{X}_{0,(2:,:)}^\intercal \boldsymbol{X}_{:,2:} \left[\boldsymbol{X}_{0,(2:,:)}^\intercal \boldsymbol{X}_{:,2:}\right]^\intercal = \boldsymbol{X}_{0,(2:,:)}^\intercal \boldsymbol{X}_{:,2:} \boldsymbol{X}_{:,2:}^\intercal [\boldsymbol{X}_{0,(2:,:)}^\intercal]^\intercal = I_{N-1}.$$

By symmetry we conclude that $[\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:} \in Orth(N-1)$. □

Proposition 2.3 – *Given $\boldsymbol{X}_0 \in \mathbb{R}^{N\times N}$ an orthogonal matrix with first column equals to $\frac{1}{\sqrt{N}}\mathbf{1}_N$, an equivalent formulation of optimization problem (2.6) is given by:*

$$\min_{\boldsymbol{H},\boldsymbol{U},\boldsymbol{\Lambda}} \left\| \boldsymbol{Y} - \boldsymbol{X}_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\intercal \\ \mathbf{0}_{N-1} & \boldsymbol{U} \end{bmatrix} \boldsymbol{H} \right\|_F^2 + \alpha \|\boldsymbol{\Lambda}^{1/2}\boldsymbol{H}\|_F^2 + \beta \|\boldsymbol{H}\|_S \triangleq f(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\Lambda}) ,$$

$$\text{s.t.} \quad \begin{cases} \boldsymbol{U}^\intercal \boldsymbol{U} = \boldsymbol{I}_{N-1} , & (a') \\ \left(\boldsymbol{X}_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\intercal \\ \mathbf{0}_{N-1} & \boldsymbol{U} \end{bmatrix} \boldsymbol{\Lambda} \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\intercal \\ \mathbf{0}_{N-1} & \boldsymbol{U}^\intercal \end{bmatrix} \boldsymbol{X}_0^\intercal \right)_{k,\ell} \leq 0 \quad k \neq \ell , & (b') \\ \boldsymbol{\Lambda} = diag(0, \lambda_2, \ldots, \lambda_N) \succeq 0 , & (c) \\ tr(\boldsymbol{\Lambda}) = N \in \mathbb{R}_*^+ . & (d) \end{cases}$$

Proof. From the previous lemma, we know that $\boldsymbol{X}$ can be decompose into two orthogonal matrices $\boldsymbol{X}_0$ and $\boldsymbol{U} = [\boldsymbol{X}_0^\intercal \boldsymbol{X}]_{2:,2:}$. Hence, we can optimize with respect to $\boldsymbol{U}$ instead of $\boldsymbol{X}$ and the second part of the constraint $(2.6a)$ is automatically satisfied. To make the equivalence, we just replace $\boldsymbol{X}$ from the main optimization problem to $\boldsymbol{X}_0 \begin{bmatrix} 1 & \mathbf{0}_{N-1}^\intercal \\ \mathbf{0}_{N-1} & \boldsymbol{U} \end{bmatrix}$ where $\boldsymbol{U}$ is now imposed to be orthogonal. □

Proposition 2.4 (Closed-form solution for the $\ell_{2,0}$ and $\ell_{2,1}$-norms) – *The solutions of problem* (2.10) *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* (2.10) *is given by the matrix* $\widehat{\boldsymbol{H}} \in \mathbb{R}^{N\times n}$ *where for* $1 \leq i \leq N$,

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

- *Using the* $\ell_{2,1}$*-norm, the optimal solution of* (2.10) *is given by the matrix* $\widehat{\boldsymbol{H}} \in \mathbb{R}^{N\times n}$ *where for* $1 \leq i \leq N$,

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

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

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

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

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

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

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

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

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

$$\frac{\partial}{\partial \boldsymbol{H}_{i,:}} \tilde{f}_i(\boldsymbol{H}_{i,:}) = 2(1+\alpha\lambda_i)\boldsymbol{H}_{i,:} - 2(\boldsymbol{X}^\intercal \boldsymbol{Y})_{i,:} = 0 \ ,$$

$$\widehat{\boldsymbol{H}}_{i,:} = (\boldsymbol{X}^\intercal \boldsymbol{Y})_{i,:}/(1+\alpha\lambda_i) \ .$$

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

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

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

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

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.

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

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

$$\min_{\boldsymbol{H}_{i,:}\in\mathbb{R}^n} \quad (1+\alpha\lambda_i)\|\boldsymbol{H}_{i,:}\|_2^2 - 2\langle(\boldsymbol{X}^\intercal\boldsymbol{Y})_{i,:}, \boldsymbol{H}_{i,:}\rangle + \beta\|\boldsymbol{H}_{i,:}\|_2 \ . \tag{2.28}$$

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

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

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

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

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

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

By taking the norm of the previous equation, we obtain

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

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

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

which concludes the proof. □

Proposition 2.5 (Euclidean gradient with respect to $\boldsymbol{U}$) – *The Euclidean gradient of f and ϕ with respect to $\boldsymbol{U}$ are*

$$\begin{aligned}
\nabla_{\boldsymbol{U}} f(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\Lambda}) &= -2\big[(\boldsymbol{H}\boldsymbol{Y}^\intercal \boldsymbol{X}_0)_{2:,2:}\big]^\intercal + 2U(\boldsymbol{H}\boldsymbol{H}^\intercal)_{2:,2:} \ , \\
\nabla_{\boldsymbol{U}} \phi(\boldsymbol{U}, \boldsymbol{\Lambda}) &= -\sum_{k=1}^{N-1}\sum_{\ell>k}^{N} \frac{\Big(\boldsymbol{B}_{k,\ell} + \boldsymbol{B}_{k,\ell}^\intercal\Big)\boldsymbol{U}\boldsymbol{\Lambda}_{2:,2:}}{\boldsymbol{h}(\boldsymbol{U}, \boldsymbol{\Lambda})_{k,\ell}} \ .
\end{aligned}$$

with $\forall 1 \leq k, \ell \leq N, \boldsymbol{B}_{k,\ell} = \big(\boldsymbol{X}_0^\intercal \boldsymbol{e}_k \boldsymbol{e}_\ell^\intercal \boldsymbol{X}_0\big)_{2:,2:}$, *and* $h(\cdot)$ *from Definition 2.11.*

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

$$f(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\Lambda}) = -2\mathrm{tr}\left(\boldsymbol{Y}^\intercal \boldsymbol{X}_0 \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & U \end{bmatrix} \boldsymbol{H}\right) + \mathrm{tr}\left(\boldsymbol{H}^\intercal \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & U^\intercal U \end{bmatrix} \boldsymbol{H}\right) .$$

The corresponding gradient is the following.

$$\begin{aligned}
\nabla_{\boldsymbol{U}} f(\boldsymbol{H}, \boldsymbol{U}, \boldsymbol{\Lambda}) &= -2\nabla_{\boldsymbol{U}}\mathrm{tr}\left(\boldsymbol{Y}^\intercal \boldsymbol{X}_0 \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U} \end{bmatrix} \boldsymbol{H}\right) + \nabla_{\boldsymbol{U}}\mathrm{tr}\left(\boldsymbol{H}^\intercal \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U}^\intercal\boldsymbol{U} \end{bmatrix} \boldsymbol{H}\right) \\
&= -2\nabla_{\boldsymbol{U}}\mathrm{tr}\left(\boldsymbol{H}\boldsymbol{Y}^\intercal \boldsymbol{X}_0 \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U} \end{bmatrix}\right) + \nabla_{U}\mathrm{tr}\left(\boldsymbol{H}\boldsymbol{H}^\intercal \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U}^\intercal\boldsymbol{U} \end{bmatrix}\right) \\
&= -2\nabla_{\boldsymbol{U}}\Big((\boldsymbol{H}\boldsymbol{Y}^\intercal \boldsymbol{X}_0)_{1,1} \cdot 1 + \mathrm{tr}\big((\boldsymbol{H}\boldsymbol{Y}^\intercal \boldsymbol{X}_0)_{2:,2:}\boldsymbol{U}\big)\Big) \\
&\quad + \nabla_{\boldsymbol{U}}\Big((\boldsymbol{H}\boldsymbol{H}^\intercal)_{1,1} \cdot 1 + \mathrm{tr}\big((\boldsymbol{H}\boldsymbol{H}^\intercal)_{2:,2:}\boldsymbol{U}^\intercal\boldsymbol{U}\big)\Big) \\
&= -2\big[(\boldsymbol{H}\boldsymbol{Y}^\intercal \boldsymbol{X}_0)_{2:,2:}\big]^\intercal + 2\boldsymbol{U}(\boldsymbol{H}\boldsymbol{H}^\intercal)_{2:,2:} \ .
\end{aligned}$$

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

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

We can write the h function as:

$$\begin{aligned}\boldsymbol{h}(\boldsymbol{U}, \boldsymbol{\Lambda})_{k,\ell} &= \big\langle \boldsymbol{e}_k\boldsymbol{e}_\ell^\intercal, \boldsymbol{h}(U, \boldsymbol{\Lambda})\big\rangle = \Big\langle \boldsymbol{X}_0^\intercal\boldsymbol{e}_k\boldsymbol{e}_\ell^\intercal\boldsymbol{X}_0, \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U}\end{bmatrix} \boldsymbol{\Lambda} \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U}\end{bmatrix}^\intercal \Big\rangle \\ &= \Big\langle \boldsymbol{X}_0^\intercal\boldsymbol{e}_k\boldsymbol{e}_\ell^\intercal\boldsymbol{X}_0, \begin{bmatrix} \lambda_1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U}\boldsymbol{\Lambda}_{2:,2:}\boldsymbol{U}^\intercal\end{bmatrix} \Big\rangle = \operatorname{tr}\Big(\boldsymbol{X}_0^\intercal\boldsymbol{e}_\ell\boldsymbol{e}_k^\intercal\boldsymbol{X}_0 \begin{bmatrix} 0 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{U}\boldsymbol{\Lambda}_{2:,2:}\boldsymbol{U}^\intercal\end{bmatrix}\Big) \\ &= \big(\boldsymbol{X}_0^\intercal\boldsymbol{e}_\ell\boldsymbol{e}_k^\intercal\boldsymbol{X}_0\big)_{1,1}\cdot 0 + \operatorname{tr}\Big(\big(\boldsymbol{X}_0^\intercal\boldsymbol{e}_\ell\boldsymbol{e}_k^\intercal\boldsymbol{X}_0\big)_{2:,2:}\boldsymbol{U}\boldsymbol{\Lambda}_{2:,2:}\boldsymbol{U}^\intercal\Big) \\ &= \operatorname{tr}\Big(\boldsymbol{B}_{k,\ell}^\intercal\boldsymbol{U}\boldsymbol{\Lambda}_{2:,2:}\boldsymbol{U}^\intercal\Big) \,.\end{aligned}$$

In conclusion we have $\nabla_{\boldsymbol{U}}\boldsymbol{h}(\boldsymbol{U}, \boldsymbol{\Lambda})_{k,\ell} = \Big(\boldsymbol{B}_{k,\ell} + \boldsymbol{B}_{k,\ell}^\intercal\Big)\boldsymbol{U}\boldsymbol{\Lambda}_{2:,2:}$, which finishes the proof. □

Proposition 2.6 (Feasible eigenvalues) – *Given any $\boldsymbol{X} \in \mathbb{R}^{N\times N}$ being an orthogonal matrix with first column equals to $1/\sqrt{N}$ (constraint (2.6a)), there always exist a matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{N\times N}$ such that the following constraints are satisfied:*

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

Proof. Let us consider a positive real value $c > 0$. Taking $\boldsymbol{\Lambda} = \operatorname{diag}(0, c, \ldots, c)/(N-1)$ leads to $\operatorname{tr}(\boldsymbol{\Lambda}) = c$ and $\forall i \neq j, (\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^\intercal)_{i,j} = -c/N < 0$. 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 2.7 (Closed-form solution of problem (2.16)) – *Consider the optimization problem (2.16). Let $\boldsymbol{X}_0$ be any matrix that belongs to the constraints set (a), and $\boldsymbol{M} = (\boldsymbol{X}_0^\intercal\boldsymbol{Y}\boldsymbol{H}^\intercal)_{2:,2:}$ the submatrix containing everything but the input's first row and first column. Finally, let $\boldsymbol{P}\boldsymbol{D}\boldsymbol{Q}^\intercal$ be the SVD of $\boldsymbol{M}$. Then, the problem admits the following closed form solution*

$$\widehat{\boldsymbol{X}} = \boldsymbol{X}_0 \begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{P}\boldsymbol{Q}^\intercal\end{bmatrix} \,.$$

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

$$\widehat{\boldsymbol{G}} = \underset{\boldsymbol{G}}{\operatorname{argmin}} \Bigg\| \boldsymbol{Y} - \boldsymbol{X}_0 \underbrace{\begin{bmatrix} 1 & \boldsymbol{0}_{N-1}^\intercal \\ \boldsymbol{0}_{N-1} & \boldsymbol{G}\end{bmatrix}}_{\triangleq\ \tilde{\boldsymbol{G}}} \boldsymbol{H} \Bigg\|_F^2 \,, \tag{2.29}$$

s.t. $\boldsymbol{G}^\mathsf{T}\boldsymbol{G} = I_{N-1}$. This is obtained by replacing $\boldsymbol{X}$ with $\boldsymbol{X}_0\tilde{\boldsymbol{G}}$.

Solving the above Equation (2.29) is equivalent to finding:

$$\widehat{\boldsymbol{G}} = \arg\max_{\boldsymbol{G}} \operatorname{tr}\left(\boldsymbol{H}\boldsymbol{Y}^\mathsf{T}\boldsymbol{X}_0\tilde{\boldsymbol{G}}\right) = \arg\max_{\boldsymbol{G}} \operatorname{tr}\left(\boldsymbol{M}^\mathsf{T}\boldsymbol{G}\right),$$

s.t. $\boldsymbol{G}^\mathsf{T}\widehat{\boldsymbol{G}} = I_{N-1}$. Then, as proved in Zou et al. [2006], we finally have $\boldsymbol{G}^* = \boldsymbol{P}\boldsymbol{Q}^\mathsf{T}$, which completes the proof. □

Lemma 2.2 – *Assume the proposed Model (2.19). If $p_1 = 0$ and $p_i \in (0,1)$, $\forall i \geq 2$, then,*

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

Proof. Based on the Factor Analysis model and the independence of $\boldsymbol{h}_i$'s,

$$\begin{aligned}\log(p(\boldsymbol{h}|\boldsymbol{y},\boldsymbol{X},\boldsymbol{\Lambda})) &\propto \log\left(p(\boldsymbol{y}|\boldsymbol{h},\boldsymbol{X},\boldsymbol{\Lambda})\right) + \log\left(p(\boldsymbol{h}|\boldsymbol{X},\boldsymbol{\Lambda})\right) \\ &\propto -\frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{h}\|_2^2 + \sum_{i=1}^{N}\log\left(p(\boldsymbol{h}_i|\lambda_i)\right) . \end{aligned} \tag{2.30}$$

Let us now focus on $\log\left(p(\boldsymbol{h}_i|\lambda_i)\right)$, for which we have

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

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:

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

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

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

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

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

Finally we can compute $\log\left(p(\boldsymbol{h}_i|\lambda_i)\right)$:

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

Note that with our parametrization, the particular case $i=1$ leads to $\log\left(p(\boldsymbol{h}_1|\lambda_1)\right)=0$. Now plugging our result in equation (2.30) and multiplying on both side by -1, we get our final result. □

Proposition 2.8 (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:*

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

Proof. To show that the p_i's are well-defined and belongs to $(0,1)$, it suffices to apply Lemma 2.3 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:

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

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

$$
p_i \log(\frac{\lambda_i}{\sqrt{2\pi}}) - \log(p_i) - \log(\frac{\lambda_i}{\sqrt{2\pi}}) = C \ . \tag{2.31}
$$

Plugging the equation (2.31) into the final result of proposition 1, we obtain:

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

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

Lemma 2.3. *Let $C > 0$. For any $x > 0$,*

$$
0 \leq -W\left(-\frac{e^{-C}\log(x)}{x}\right) / \log(x) \leq 1 \ . \tag{2.32}
$$

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

$$
\frac{\partial}{\partial x}\left[-W\left(-\frac{e^{-C}\log(x)}{x}\right) / \log(x)\right] = \frac{W\left(-\frac{e^{-C}\log(x)}{x}\right)\left(W\left(-\frac{e^{-C}\log(x)}{x}\right) + \log(x)\right)}{x\log^2(x)\left(W\left(-\frac{e^{-C}\log(x)}{x}\right) + 1\right)} \ . \tag{2.33}
$$

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

$$
-1/e < -e^{-(C+1)} = \min_{x>0} -\frac{e^{-C}\log(x)}{x} \leq -\frac{e^{-C}\log(x)}{x} \ . \tag{2.34}
$$

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,

$$-e^{-C}\frac{\log(x)}{x} < -\frac{\log(x)}{x} \iff W\left(-e^{-C}\frac{\log(x)}{x}\right) < W\left(\frac{-\log(x)}{x}\right) = -\log(x) \tag{2.35}$$

$$\iff W\left(-e^{-C}\frac{\log(x)}{x}\right) + \log(x) < 0\ . \tag{2.36}$$

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,

$$-\frac{1}{e} \leq -\frac{\log(x)}{x} < -e^{-C}\frac{\log(x)}{x} \iff W\left(-\frac{\log(x)}{x}\right) = -\log(x) < W\left(-e^{-C}\frac{\log(x)}{x}\right) \tag{2.37}$$

$$\iff W\left(-e^{-C}\frac{\log(x)}{x}\right) + \log(x) > 0\ . \tag{2.38}$$

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 (2.32). 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 $W(y) < \log(y)$ [Hoorfar and Hassani, 2007]. Let $f(x) = -\frac{e^{-C}\log(x)}{x}$, for x small enough we have:

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

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

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

□

3

Tensor-based convolutional dictionary learning with CP low-rank activations

Abstract

The goal of this chapter is to provide algorithms for Convolutional Dictionary Learning (CDL) taking into account the underlying linear structure of the multivariate input signals. In this view, we add to the initial CDL problem a tensor constraint enforcing the activation maps to be sparse and Canonical Polyadic (CP) low-rank. We propose two algorithms, called T-ConvADMM and T-ConvFISTA, for the minimization. Based on a 2-steps alternating procedure, they both rely on an optimization in the Fourier domain to efficiently solve the problem despite the increasing complexity induced by the tensor representation. Their benefits in term of convergence, complexity, and interpretability of the learned dictionary and activations are discussed in details. Then, we evaluate these two algorithms on a wide range of synthetic data. Experiments show that (i) the low-rank model entails a better robustness to noise and perturbations, resulting in accurate, sparse and interpretable encoding of the signals, (ii) algorithms are computationally faster than previous ones in several cases. Finally, multiple real-data applications, ranging from image processing to electroencephalogram analysis, are performed, highlighting the important advantages and versatility of this tensor CP low-rank formulation.

Contents

The material of this chapter is based on the following publications:

P. Humbert, J. Audiffren, L. Oudre, and N. Vayatis. Low rank activations for tensor-based convolutional sparse coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

P. Humbert, L. Oudre, N. Vayatis, and J. Audiffren. Tensor Convolutional Sparse Coding with Low-Rank activations, an application to EEG analysis. *(Submitted)*.

1 Introduction

The linear decomposition of a signal into few atoms of a learned dictionary instead of a predefined one such as discrete cosine transform, wavelets, curvelets, etc., has led to state-of-the-art results in a wide range of topics, including image denoising [Elad and Aharon, 2006], image classification [Raina et al., 2007; Mairal et al., 2009; Yang et al., 2009], and other signal processing tasks [Huang and Aviyente, 2007; Févotte et al., 2009; Peyré, 2009; Mairal et al., 2010]. Recently, its convolutional counterpart known as Convolutional Dictionary Learning (CDL) or Convolutional Sparse Coding (CSC), has gained renewed interest. The central idea behind CDL is to replace the traditional patch-based representation with a global shift-invariant one. Various algorithms built around the Alternating Direction Method of Multipliers (ADMM) or the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) have been suggested to efficiently handle the associated CDL problem. But interestingly, they mainly focused on a resolution for univariate signals or images [Garcia-Cardona and Wohlberg, 2018a] while multivariate data with a natural tensor structure are encountered in many scientific areas.

One approach to still apply vector-based algorithms on multivariate data is to vectorized them by stretching their elements. However, this naive processing ignores the multidimensional structure of the input and is frequently sub-optimal. A powerful idea to effectively exploit the structural information is to use multilinear analysis and low-rank tensor decomposition techniques [Kolda and Bader, 2009]. Indeed, by providing essential tools for handling multivariate data they naturally simplify the adaptation of machine learning and statistical methods to tensors. Until recently, a lot of works have considered with great success the tensor framework e.g. in regression [Zhou

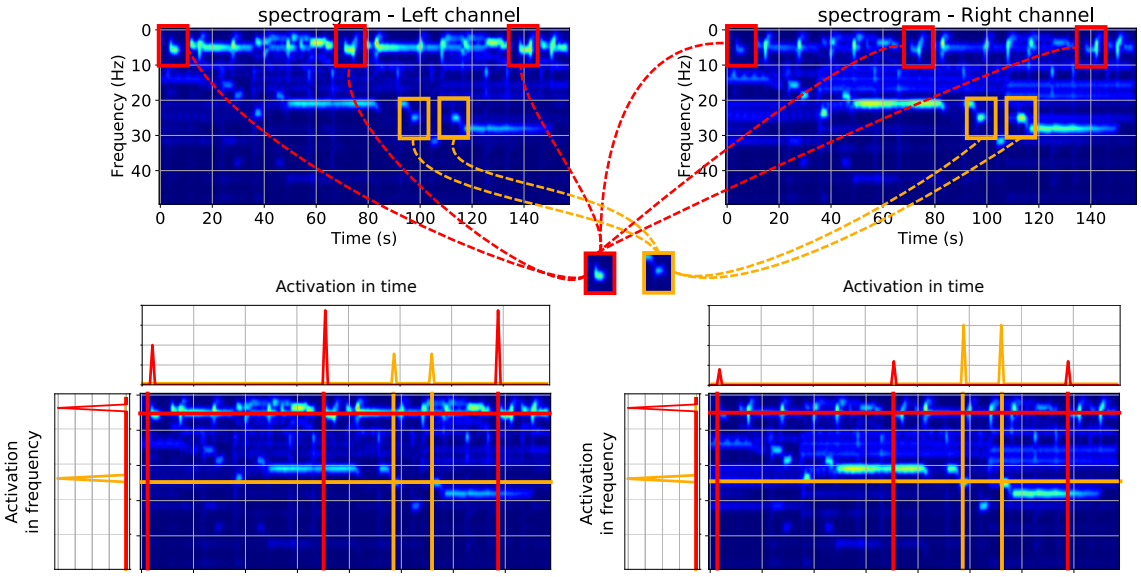


Figure 3.1: Spectrograms of a stereo audio jazz signal.

et al., 2013; Rabusseau and Kadri, 2016; Li et al., 2017; He et al., 2018], image completion [Liu et al., 2012], decomposition of spectrograms or scalograms of EEG data [Cong et al., 2015], processing of audio signals [Wang et al., 2013].

Contributions. In this chapter, we provide two algorithms based on ADMM or FISTA for CDL taking into account the underlying structure of the multivariate/tensor signals. Unlike previous works, we do not rely on a low-rank constraint on the atoms. Instead, we extend the standard minimization CDL problem to a tensorial one with an additional CP low-rank decomposition constraint on the activation maps. The idea of enforcing low-rank constraints in CDL is not novel: Rigamonti et al. [2013] and Sironi et al. [2014] used the idea of separable filters for learning a low-rank collection of atoms in order to improve computational runtime. More recent publications including [Quesada et al., 2018; Silva et al., 2018; Quesada et al., 2019] have also successfully used low-rank (or even rank-1) constraints on 2-D dictionary. Yet, in all these approaches, the low-rank constraints have been enforced on the dictionary/atoms. However, in several applicative contexts, the low-rank structure naturally appears in the activations rather than in the atoms/dictionary. To illustrate the relevance of our new approach, we display in Figure 3.1 an example of two spectrograms obtained from a stereo music recording. Both spectrograms exhibit a low-rank structure. This is a known property for such time-frequency representations, which is commonly used for signal decomposition or source separation. Some repetitive patterns (highlighted in red and orange) are also visible on the spectrograms which suggests that a CDL model may appear as natural for such data. However, the strong low-rank structure of the data is here transferred into the activations tensors rather than into the observed patterns. In other words, although the time-frequency atoms may be complex (and thus without a low-rank structure), the activations (i.e. the time/frequency/channel positions where these atoms appear) clearly are low-rank. In this example, this phenomenon may be explained by the harmonic structure of the audio signals, to the tempo grid used by the instruments or to the fact that both channels approximately capture the same audio scene. Such observations can also be made for sequences of images, where the structure lies in the locations of the patterns rather that in the individual atoms.

The organization of this chapter is as follow. We first recall the standard CDL model and the most important algorithms to solve the associated problem in Section 2. Then, we introduce our multivariate CDL problem, referred to as Kruskal Convolutional Dictionary Learning (K-CDL) in Section 3. We propose two algorithms based on ADMM or FISTA to solve it (Section 4). Their properties are analyzed and discuss in details. Finally, in Section 6, we conduct multiple empirical analysis on synthetic and real data to highlight the performances of our approach.

2 Convolutional dictionary learning

The Dictionary Learning problem (DL) was introduced in the context of modeling receptive fields in human vision by Olshausen and Field [1996, 1997]. As their results were considered impressive by the scientific community, DL enjoyed early success and found many applications in image processing (e.g. for discovering and visualizing the underlying structure of natural image patches). However, DL is mostly a patch-based method, and thereby does not capture the correlation between local neighborhoods. To circumvent this drawback, following the work of Lewicki and Sejnowski [1999] in discrete 1D time-varying signals, Grosse et al. [2007] introduced its extension called Convolutional Dictionary Learning (CDL). This work was generalized to images by Mørup et al. [2008]. The main idea behind CDL is to replace the traditional patch-based model with a global shift-invariant one. In this way, a dictionary of patterns/atoms (small signals) is learned so that the input signals can be represented approximately by a superposition of only a small number of them, called "active". For any input signal, these active basis functions produce a sparse signal representation that concisely represents that signal.

Formally, given a finite set of N signals $\boldsymbol{y}_1, \dots, \boldsymbol{y}_N$ in $\mathbb{R}^M$ and a scalar $\lambda > 0$, the ℓ_1-regularized CDL problem is

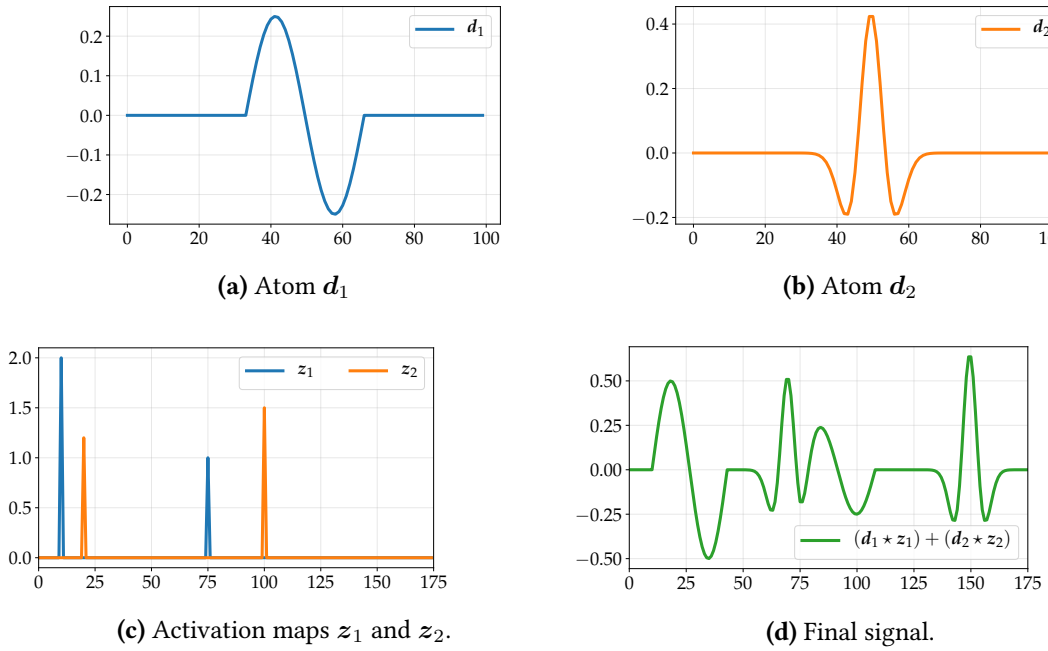
$$\min_{\{\boldsymbol{d}_k\}_{k=1}^K, \{\boldsymbol{z}_{n,k}\}_{n,k=1}^{N,K}} \frac{1}{2} \sum_{n=1}^{N} \left(\|\boldsymbol{y}_n - \sum_{k=1}^{K} \boldsymbol{d}_k \circledast \boldsymbol{z}_{n,k}\|_2^2 + \lambda \sum_{k=1}^{K} \|\boldsymbol{z}_{n,k}\|_1 \right) , \qquad (3.1)$$
$$\text{s.t.} \quad \|\boldsymbol{d}_k\|_2 \leq 1 \quad \forall k = 1, \dots, K$$

where the $\boldsymbol{d}_k \in \mathbb{R}^W$ are the *atoms*, the $\boldsymbol{z}_{n,k} \in \mathbb{R}^M$ are the *activation maps*, and $\circledast$ denotes the (circular) convolutional operator (see Appendix for more details). An example of such representation is given in Figure 3.2.

For clarity, in the sequel, we will drop the index k or n when obvious e.g. $\{\boldsymbol{d}_k\}_{k=1}^K$ will be denoted by $\{\boldsymbol{d}_k\}$.

In its simplest form, the CDL problem (3.1) involves two important components: a sparsity regularization and a unit-norm constraint.

The sparsity regularization. A natural regularization to encourage sparsity of the activations maps is the ℓ_0-regularization. However, with this (semi)-norm, the problem is often intractable and research has either focus for an approximate solution using a greedy algorithm, or for a convex relaxation. A typical convex relaxation for this problem is the equation (3.1), where a ℓ_1-regularization is preferred. This relaxation can be shown to consistently estimate the solution of the ℓ_0 problem under some assumptions on the sparsity of the solution and the design of the



(a) Atom $\boldsymbol{d}_1$

(b) Atom $\boldsymbol{d}_2$

(c) Activation maps $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$.

(d) Final signal.

Figure 3.2: Two atoms are displayed in (a) and (b). The first one in blue is a sinusoidal and the second one in orange is the "Mexican hat function". For each atom, the corresponding activation map is represented in (c). The resulting signal from the CSC model is displayed in (d).

dictionary [Donoho and Elad, 2003; Fuchs, 2004]. While other sparsity-based penalties may be considered e.g. group sparsity, in this chapter, we will focus on the most frequently employed, the ℓ_1-regularization (see [Mairal et al., 2014] for a complete review).

The unit-norm constraint. The most common constraint imposed on the atoms is to have a unit norm as in equation (3.1). This is an important constraint since multiplying an atom $\boldsymbol{d}_k$ by a scalar $a > 1$ and all $\{\boldsymbol{z}_{n,k}\}_{n=1}^N$ by $1/a$, does not change the value of the objective function even if the ℓ_1-norm is decreased by a factor $1/a$. Thus, without the unit norm constraint, the $\{\boldsymbol{z}_{n,k}\}_{n=1}^N$ tend to 0 and the norm of $\boldsymbol{d}_k$ explodes. Other constraints have also been proposed, such as smoothness constraints enforced by regularizing the gradient with its ℓ_2-norm.

Even though the CDL problem is not jointly convex in $(\{\boldsymbol{d}_k\}, \{\boldsymbol{z}_{n,k}\})$, it is convex with respect to each variable when the other one is fixed. A natural optimization scheme for minimizing the objective function is therefore to alternate between the minimization with respect to the atoms $\{\boldsymbol{d}_k\}$ when the activation maps $\{\boldsymbol{z}_{n,k}\}$ are fixed and vice versa. This strategy known as *alternating minimization* or *block coordinate descent* [Ortega and Rheinboldt, 2000] has proven to be very effective in solving a wide range of optimization problems such as iteratively reweighted least squares, robust regression, or sparse recovery [Daubechies et al., 2010]. Note that, we consider an optimization problem for which it is not possible, in general, to guarantee that we are going to obtain the global minimum. Furthermore, this problem exhibits several symmetries and admits multiple global optima which can be an issue in practice [Mairal et al., 2014]. Optimizing with respect to $\{\boldsymbol{z}_{n,k}\}$ is often referred as *Convolutional Sparse Coding* (CSC) and will be our main focus in this chapter.

2.1 Convolutional sparse coding

As the activation maps are independent across the signals $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$, we can solve the CSC problem for only one of them.

Given a signal $\boldsymbol{y}$, the CSC problem is

$$\min_{\{\boldsymbol{z}_k\}} \frac{1}{2}\|\boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{d}_k \circledast \boldsymbol{z}_k\|_2^2 + \lambda \sum_{k=1}^{K} \|\boldsymbol{z}_k\|_1 \ , \tag{3.2}$$

where $\circledast$ is the circular convolution (see Appendix).

In the sequel, without further information, we will always use this convolution and focus on the CSC instead of the CDL. Several algorithms have been proposed to solve the CSC problem. The work of Kavukcuoglu et al. [2010] extends to CSC the coordinate descent methods introduced by Friedman et al. [2007]. The Feature Sign Search algorithm proposed in Grosse et al. [2007] solves at each step a quadratic sub-problem for an active set of the estimated nonzero coefficients. More recently, Papyan et al. [2017a] and Zisselman et al. [2019] have introduced respectively the Slice-Based Dictionary Learning (SBDL) and the Local Block Coordinate Descent (LoBCoD) algorithms. The two most important algorithms remain the ones of [Bristow et al., 2013] and [Chalasani et al., 2013] which are described below.

2.1.1 Convolutional sparse coding with ADMM

Zeiler et al. [2010] were the first to propose an efficient algorithm for the CSC problem (3.2) by introducing an auxiliary variable to separate the convolution from the ℓ_1-regularization. This important idea of separating the fidelity term from the sparsity term is now widely used in contemporary methods. To do so, solvers often rely on the Alternating Direction Method of Multipliers (ADMM) [Glowinski and Marroco, 1975; Gabay and Mercier, 1976] in the Fourier domain for the computational convenience of convolutions [Bristow et al., 2013; Wohlberg, 2014, 2015]. The algorithm who popularize ADMM for both the CSC and CDL is called Fast Convolutional Sparse Coding (FCSC) [Bristow et al., 2013]. In this paper, authors have shown remarkable improvements in efficiency by exploiting the Parseval's equality and the convolutional theorem for solving (3.2).

The steps to solve the CSC problem with ADMM are straightforward. We first consider the splitting

$$f(\{\boldsymbol{z}_k\}) = \frac{1}{2}\|\boldsymbol{y} - \sum_{k=1}^{K} \boldsymbol{d}_k \circledast \boldsymbol{z}_k\|_2^2 \quad , \quad \psi(\{\boldsymbol{z}_k\}) = \lambda \sum_{k=1}^{K} \|\boldsymbol{z}_k\|_1 \ , \tag{3.3}$$

where f is the fidelity term which control the difference between the input and its reconstruction, and ψ is the regularization term. Then, by introducing K auxiliary variables $\{\boldsymbol{t}_k\}$, we rewrite the main equation (3.2)

$$\begin{aligned} \min_{\{\boldsymbol{t}_k, \boldsymbol{z}_k\}} \ & f(\{\boldsymbol{z}_k\}) + \psi(\{\boldsymbol{t}_k\}) \\ \text{s.t.} \quad & \boldsymbol{z}_k = \boldsymbol{t}_k \quad \forall k = 1, \ldots, K \ . \end{aligned}$$

The corresponding iterations of the ADMM algorithm with a scalar $\rho > 0$ and the $\{\boldsymbol{u}_k\}$ as dual variables are given by

$$\{\boldsymbol{z}_k^{(s+1)}\} = \arg\min_{\{\boldsymbol{z}_k\}} f(\{\boldsymbol{z}_k\}) + \frac{\rho}{2}\sum_{k=1}^{K}\|\boldsymbol{z}_k - \boldsymbol{t}_k^{(s)} + \boldsymbol{u}_k^{(s)}\|_2^2 \,, \tag{3.4}$$

$$\{\boldsymbol{t}_k^{(s+1)}\} = \arg\min_{\{\boldsymbol{t}_k\}} \psi(\{\boldsymbol{t}_k\}) + \frac{\rho}{2}\sum_{k=1}^{K}\|\boldsymbol{z}_k^{(s+1)} - \boldsymbol{t}_k + \boldsymbol{u}_k^{(s)}\|_2^2 \,, \tag{3.5}$$

$$\{\boldsymbol{u}_k^{(s+1)}\} = \boldsymbol{u}_k^{(s)} + \boldsymbol{z}_k^{(s+1)} - \boldsymbol{t}_k^{(s+1)} \,. \tag{3.6}$$

Subproblem (3.5) admits the well-known closed-form solution [Tibshirani, 1996]

$$\forall k = 1, \dots, K, \qquad \boldsymbol{t}_k^{(s+1)} = \mathcal{S}_{\lambda/\rho}(\boldsymbol{z}_k^{(s+1)} + \boldsymbol{u}_k^{(s)}) \,,$$

where $\mathcal{S}_\gamma(\cdot)$ is the soft-thresholding operator i.e. for a vector $\boldsymbol{x} \in \mathbb{R}^m$

$$\mathcal{S}_\gamma(\boldsymbol{x})[i] = \text{sign}(\boldsymbol{x}_i)\max(|\boldsymbol{x}_i| - \gamma, 0) \,.$$

Subproblem (3.4) also admits a closed-form solution (in certain conditions). However, finding the solution is a computationally demanding process due to the size of the matrices involved. As proposed by Bristow and Lucey [2014], one way to address this issue is to use both the Parseval's and convolution theorems in order to take advantage of the convolutional structure of the problem. Forgetting the iteration index, rewriting the objective function of (3.4) in the Fourier domain gives

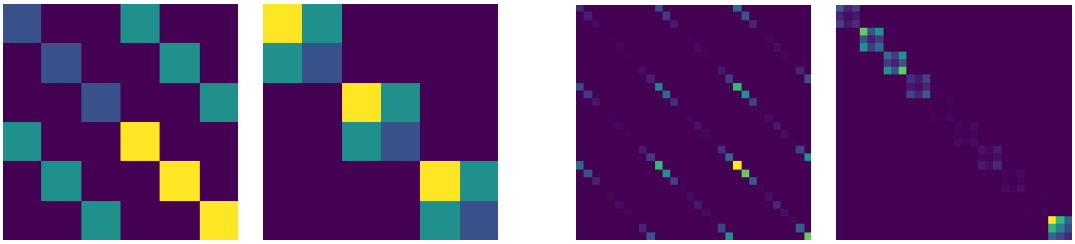
$$\frac{1}{2M}\|\widehat{\boldsymbol{y}} - \sum_{k=1}^{K}\widehat{\boldsymbol{d}}_k * \widehat{\boldsymbol{z}}_k\|_2^2 + \frac{\rho}{2M}\sum_{k=1}^{K}\|\widehat{\boldsymbol{z}}_k - \widehat{\boldsymbol{t}}_k + \widehat{\boldsymbol{u}}_k\|_2^2 \,, \tag{3.7}$$

where $\widehat{\cdot}$ denotes the frequency representation of a signal, $*$ is the component-wise product, and each $\widehat{\boldsymbol{d}_k}$ is in $\mathbb{C}^M$. As $\boldsymbol{u} * \boldsymbol{v} = \text{diag}(\boldsymbol{u})\boldsymbol{v}$, the component-wise product is rewritten in a matrix product. Then, by introducing the matrix $\widehat{\boldsymbol{D}} = [\text{diag}(\widehat{\boldsymbol{d}_1}), \dots, \text{diag}(\widehat{\boldsymbol{d}_K})]$ in $\mathbb{C}^{M\times KM}$, and the three vectors $\widehat{\boldsymbol{z}} = [\widehat{\boldsymbol{z}}_1^\intercal, \dots, \widehat{\boldsymbol{z}}_K^\intercal]^\intercal$, $\widehat{\boldsymbol{t}} = [\widehat{\boldsymbol{t}}_1^\intercal, \dots, \widehat{\boldsymbol{t}}_K^\intercal]^\intercal$, and $\widehat{\boldsymbol{u}} = [\widehat{\boldsymbol{u}}_1^\intercal, \dots, \widehat{\boldsymbol{u}}_K^\intercal]^\intercal$ in $\mathbb{C}^{KM}$, the first term of (3.7) becomes $\|\widehat{\boldsymbol{y}} - \widehat{\boldsymbol{D}}\widehat{\boldsymbol{z}}\|_2^2$ and its minimum is given by the solution in $\widehat{\boldsymbol{z}}$ of

$$\left(\widehat{\boldsymbol{D}}^H\widehat{\boldsymbol{D}} + \rho\boldsymbol{I}\right)\widehat{\boldsymbol{z}} = \left(\widehat{\boldsymbol{D}}^H\widehat{\boldsymbol{y}} + \rho(\widehat{\boldsymbol{t}} - \widehat{\boldsymbol{u}})\right) \,, \tag{3.8}$$

where $(\cdot)^H$ stands for the Hermitian transpose. Here, the matrix $\left(\widehat{\boldsymbol{D}}^H\widehat{\boldsymbol{D}} + \rho\boldsymbol{I}\right)$ is of size $KM \times KM$ and can be expensive to inverse (when possible). Fortunately, as $\widehat{\boldsymbol{D}}$ is block diagonal, $\widehat{\boldsymbol{D}}^H\widehat{\boldsymbol{D}}$ is a particular diagonal block matrix known as *band matrix* (see Figures 3.3a and 3.3c). Hence, it is possible to permute rows and columns in order to only solve M independent $K \times K$ linear systems (see Figures 3.3b and 3.3d). More precisely, this system is actually composed of M independent system, which correspond to each frequency computed by the FFT. The solution of the initial problem can then be retrieved using the inverse Fourier transform. The full algorithm to solve the CSC based on ADMM is described in Algorithm 3.1.

Remark 3.1. *In the general case, the necessity to find such permutation comes from the graph community where they want to exhibit adjacency matrices with small bandwidth. Two popular algorithms are the reverse Cuthill-McKee algorithm [Cuthill and McKee, 1969] later improved by the GPS algorithm [Gibbs et al., 1976].*



(a) (b) (c) (d)

Figure 3.3: Visualization of the Gram matrix $\widehat{\boldsymbol{D}}^H \widehat{\boldsymbol{D}}$ before and after a reordering. The two left matrices (a, b) correspond to the Gram matrix without reordering (a) and with reordering (b). The two right matrices (c, d) also correspond to the Gram matrix without reordering (c) and with reordering (d) but for a higher dimension.

Algorithm 3.1 ADMM for CSC

Input: signal $\boldsymbol{y}$, dictionary $\boldsymbol{D}$, regularization and ADMM parameters λ, ρ, tolerance ε
Initialization: $\boldsymbol{z}^{(0)}$
Precompute $\widehat{\boldsymbol{y}}$ and $\widehat{\boldsymbol{D}}$ using the FFT
$\boldsymbol{t}^{(0)} \longleftarrow \boldsymbol{z}^{(0)}$
$\boldsymbol{u}^{(0)} \longleftarrow (0, \cdots, 0)$
repeat

▷ Update of $\boldsymbol{z}$ via equation (3.4)
Compute $\widehat{\boldsymbol{z}}^{(s)}, \widehat{\boldsymbol{t}}^{(s)}$ and $\widehat{\boldsymbol{u}}^{(s)}$ using the FFT
Solve the linear systems $\left(\widehat{\boldsymbol{D}}^H \widehat{\boldsymbol{D}} + \rho \boldsymbol{I}\right) \widehat{\boldsymbol{z}} = \left(\widehat{\boldsymbol{D}}^H \widehat{\boldsymbol{y}} + \rho(\widehat{\boldsymbol{t}}^{(s)} + \widehat{\boldsymbol{u}}^{(s)})\right)$
Compute $\boldsymbol{z}^{(s+1)}$ using the inverse FFT

▷ Update of $\boldsymbol{t}$ via equation (3.5)
$\boldsymbol{t}^{(s+1)} \longleftarrow \mathcal{S}_{\lambda/\rho}(\boldsymbol{z}^{(s+1)} + \boldsymbol{u}^{(s)})$

▷ Update of $\boldsymbol{u}$ via equation (3.6)
$\boldsymbol{u}^{(s+1)} \longleftarrow \boldsymbol{u}^{(s)} + \boldsymbol{z}^{(s+1)} - \boldsymbol{t}^{(s+1)}$

until $\|\boldsymbol{z}^{(s+1)} - \boldsymbol{z}^{(s)}\|_\infty \leq \varepsilon$

Convergence and complexity The ADMM algorithm is proven to converge to the optimal solution [Gabay, 1983]. Furthermore, in practice, this algorithm often gives an estimate with sufficient accuracy within tens of iterations. Indeed, with alternate minimization, each iteration does not need to find an optimal point, but a point with medium accuracy. Unfortunately, simple examples show that ADMM can be very slow to converge to high accuracy [Boyd et al., 2011].

The complexity of ADMM-based solvers such as FCSC are easily obtained by the analysis of each step. The first step requires the FFT which gives a complexity of $\mathcal{O}(KM\log(M))$. Then, as already mentioned, we need to solve M independent linear systems of size $K \times K$ which gives a complexity of $\mathcal{O}(K^3 M)$ when using direct method such as Gaussian elimination or Cholesky decomposition. Finally, the soft-threshold part and the dual variable updates give a complexity of $\mathcal{O}(KM)$.

2.1.2 Convolutional sparse coding with FISTA

Using the Fast Iterative Soft Thresholding Algorithm (FISTA) [Beck and Teboulle, 2009] to solve the CSC problem (3.2) was first proposed by Chalasani et al. [2013]. Based on the Iterative Soft Thresholding Algorithm (ISTA) [Daubechies et al., 2004], this popular proximal method has the advantage of being a simple gradient-based algorithm involving very simple computations. Furthermore, compared to ISTA, FISTA performs an extra step known as the *Nesterov's momentum* which accelerates its convergence.

The steps to solve the CSC problem with FISTA are straightforward. We first consider the same splitting (3.3) used in ADMM (Section 2.1.1). Then we alternate between i) a gradient descent step on the fidelity term f i.e.

$$\boldsymbol{z}_k^{(s+1/2)} = \boldsymbol{z}_k^{(s)} - \eta \nabla f\left(\{\boldsymbol{z}_k^{(s)}\}_{k=1}^K\right) \qquad \text{with } \eta > 0\ , \tag{3.9}$$

ii) the proximal operator of $\eta \cdot \psi(\cdot)$

$$\forall k = 1, \dots, K, \qquad \boldsymbol{z}_k^{(s+1)} = \text{prox}_{\eta\cdot\psi}\left(\boldsymbol{z}_k^{(s+1/2)}\right) = \mathcal{S}_{\eta\lambda}\left(\boldsymbol{z}_k^{(s+1/2)}\right)\ ,$$

where $\mathcal{S}_\gamma(\cdot)$ is the soft-thresholding operator introduced earlier, and iii) the Nesterov's momentum relative to FISTA (see Algorithm 3.2). Once again, we can take advantage of the FFT and perform the descent step in the frequency domain. The descent step is thus given by

$$\widehat{\boldsymbol{z}}_k^{(s+1/2)} = \widehat{\boldsymbol{z}}_k^{(s)} - \eta' \nabla \widehat{f}\left(\{\widehat{\boldsymbol{z}}_k^{(s)}\}_{k=1}^K\right) \qquad \text{with } \eta' > 0\ .$$

To express the gradient in a nice formulation, and forgetting the iteration index, we introduce the matrix $\widehat{\boldsymbol{D}} = [\text{diag}(\widehat{\boldsymbol{d}}_1), \dots, \text{diag}(\widehat{\boldsymbol{d}}_K)]$ in $\mathbb{C}^{M\times KM}$, and the vector $\widehat{\boldsymbol{z}} = [\widehat{\boldsymbol{z}}_1^\intercal, \dots, \widehat{\boldsymbol{z}}_K^\intercal]^\intercal$ in $\mathbb{C}^{KM}$. The fidelity term becomes $\|\widehat{\boldsymbol{y}} - \widehat{\boldsymbol{D}}\widehat{\boldsymbol{z}}\|_2^2$ and the gradient with respect to $\widehat{\boldsymbol{z}}$ is now given by

$$\nabla \widehat{f}\left(\{\widehat{\boldsymbol{z}}_k\}_{k=1}^K\right) = \nabla \widehat{f}(\widehat{\boldsymbol{z}}) = -\widehat{\boldsymbol{D}}^H(\widehat{\boldsymbol{y}} - \widehat{\boldsymbol{D}}\widehat{\boldsymbol{z}}) = \widehat{\boldsymbol{D}}^H(\widehat{\boldsymbol{D}}\widehat{\boldsymbol{z}} - \widehat{\boldsymbol{y}})\ ,$$

where $(\cdot)^H$ stands for the Hermitian transpose.

Convergence and complexity FISTA has an optimal theoretical convergence rate guarantee of $\mathcal{O}(1/t^2)$ [Beck and Teboulle, 2009] which makes it very efficient to solve the CSC problem. The proof of convergence and the convergence rates do not depend on the particular structure of the CSC problem and can also be proven. Unlike for the simple proximal scheme (ISTA), we cannot guarantee that the sequence of iterates generated by the accelerated version (FISTA) is itself convergent. Furthermore, it should be noted that accelerated schemes are not necessarily descent algorithms, in the sense that the objective does not necessarily decrease at each iteration [Bach et al., 2012].

The complexity of FISTA-based solvers are easily obtained by the analysis of each step. The pre-computations of $\widehat{\boldsymbol{y}}$ and $\{\widehat{\boldsymbol{d}}_k\}$ is of complexity $\mathcal{O}((K+1)M\log(M))$. Then, the first step of FISTA requires the FFT which gives a complexity of $\mathcal{O}(KM\log(M))$. The computation of the gradient only relies on simple matrix multiplications and have a complexity of $\mathcal{O}(KM)$ instead of $\mathcal{O}(KM^2)$ due to the diagonal-block structure of $\widehat{\boldsymbol{D}}$. Finally, the soft-threshold part and the dual variable updates give a complexity of $\mathcal{O}(KM)$.

Algorithm 3.2 FISTA for CSC

Input: signal $\boldsymbol{y}$, dictionary $\boldsymbol{D}$, regularization and step parameters λ, η' ($\eta' = 1/L$, the inverse of Lipschitz constant if calculate), tolerance ε
Initialization: $\boldsymbol{z}^{(0)}$
Precompute $\widehat{\boldsymbol{y}}$ and $\widehat{\boldsymbol{D}}$ using the FFT
$t^{(0)} \longleftarrow 1$
$\boldsymbol{w}^{(0)} \longleftarrow \boldsymbol{z}^{(0)}$
repeat
    ▷ Update of $\boldsymbol{w}$ via a proximal gradient step (ISTA)
        Compute $\widehat{\boldsymbol{z}}^{(s)}$ using the FFT
        $\widehat{\boldsymbol{w}}^{(s+1/2)} \longleftarrow \widehat{\boldsymbol{z}}^{(s)} - \frac{1}{L}\widehat{\boldsymbol{D}}^H(\widehat{\boldsymbol{D}}\widehat{\boldsymbol{z}}^{(s)} - \widehat{\boldsymbol{y}})$
        Compute $\boldsymbol{w}^{(s+1/2)}$ using the inverse FFT
        $\boldsymbol{w}^{(s+1)} \longleftarrow \mathcal{S}_{\eta\lambda}\left(\boldsymbol{w}_k^{(s+1/2)}\right)$ ▷ λ/L and not λ
    ▷ Nesterov momentum step (FISTA)
    $t^{(s+1)} \longleftarrow \frac{1 + \sqrt{1 + 4 \cdot t^{(s)^2}}}{2}$
    $\boldsymbol{z}^{(s+1)} \longleftarrow \boldsymbol{w}^{(s+1)} + \frac{t^{(s)} - 1}{t^{(s+1)} + 1}(\boldsymbol{w}^{(s+1)} - \boldsymbol{w}^{(s)})$
until $\|\boldsymbol{z}^{(s+1)} - \boldsymbol{z}^{(s)}\|_\infty \leq \varepsilon$

2.2 Dictionary update

We now quickly focus on the problem of learning a dictionary.

Given the activation maps $\{\boldsymbol{z}_{n,k}\}$, the CDL problem becomes

$$\min_{\{\boldsymbol{d}_k\}_{k=1}^K} \frac{1}{2}\sum_{n=1}^N \|\boldsymbol{y}_n - \sum_{k=1}^K \boldsymbol{d}_k \circledast \boldsymbol{z}_{n,k}\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{d}_k\|_2 \leq 1 \quad \forall k = 1, \dots, K \ . \tag{3.10}$$

Conversely to the sparse coding, here the activation maps are fixed and we want to find a common dictionary for all the signals $\{\boldsymbol{y}_n\}$. In the past years, a lot of algorithms have been proposed to solve this problem. In the following, we quickly present some of them.

2.2.1 Proximal gradient descent

Since in equation (3.10) the constraint on the atoms is convex, it is possible to use a proximal gradient descent to solve the CDL. Let us denote by $\mathcal{I}_\Omega$ the indicator function of the constraint set $\Omega = \{\boldsymbol{x} \mid \|\boldsymbol{x}\|_2 \leq 1\}$ i.e. Ω is the unit ball. Problem (3.10) is then equivalent to

$$\min_{\{\boldsymbol{d}_k\}} \frac{1}{2}\sum_{n=1}^N \|\boldsymbol{y}_n - \sum_{k=1}^K \boldsymbol{d}_k \circledast \boldsymbol{z}_{n,k}\|_2^2 + \mathcal{I}_\Omega(\boldsymbol{D}) \ , \tag{3.11}$$

where the constraint is now a penalization-term. The proximal operator of $\mathcal{I}_\Omega$ is the projection proj_Ω onto Ω. As Ω is the ℓ_2 unit ball, this operator is separable for each atom and can be computed

Algorithm	Time complexity
Deconvolutional Networks[Zeiler et al., 2010]	$T($ $\underbrace{KM}_{\text{Conjugate gradient}}$ $\cdot$ $\underbrace{KMW}_{\text{Spatial convolutions}}$ $+$ $\underbrace{KD}_{\text{Shrinkage}}$ $)$
Fast CSC[Bristow et al., 2013]	$T($ $\underbrace{K^3M}_{\text{Linear systems}}$ $+$ $\underbrace{KM\log(M)}_{\text{FFTs}}$ $+$ $\underbrace{KM}_{\text{Shrinkage}}$ $)$
Fast and Flexible CSC[Heide et al., 2015]	$\underbrace{K^3M + (T-1)K^2M}_{\text{Linear systems}} + T($ $\underbrace{KM\log(M)}_{\text{FFTs}}$ $+$ $\underbrace{KM}_{\text{Shrinkage}}$ $)$
(Wolbergh) FCSC-ShM[Wohlberg, 2015]	$T($ $\underbrace{KM}_{\text{Linear systems}}$ $+$ $\underbrace{KM\log(M)}_{\text{FFTs}}$ $+$ $\underbrace{KM}_{\text{Shrinkage}}$ $)$
ConvFISTA CSC[Chalasani et al., 2013]	$T($ $\underbrace{KM}_{\text{Gradient}}$ $+$ $\underbrace{KM\log(M)}_{\text{FFTs}}$ $+$ $\underbrace{KM}_{\text{Shrinkage}}$ $)$
SBDL (CSC + CDL)[Papyan et al., 2017a]	$T($ $\underbrace{KMW + M(k^3 + Kk^2)}_{\text{LARS}}$ $+$ $\underbrace{MK^2}_{\text{Gram}}$ $+$ $\underbrace{Mk(W+K) + WK^2}_{\text{K-SVD}}$ $)$
LoBCoD (CSC + CDL)[Zisselman et al., 2019]	$T($ $\underbrace{KMW + M(k^3 + Kk^2)}_{\text{LARS}}$ $+$ $\underbrace{MK^2}_{\text{Gram}}$ $+$ $\underbrace{M(W + Wk + K)}_{\text{Stochastic-LoBCoD}}$ $)$

Table 3.1: T is the number of iteration, K the number of atoms, M the size of the signal, W the size of the atoms, and k is the maximum number of nonzeros per *"needle"* (see [Zisselman et al., 2019]). Note that, in the worst case, $k = K$. FCSC-ShM is FCSC with an iterative application of the Sherman-Morrison equation.

via a closed-form

$$\text{proj}_{\Omega}(\boldsymbol{d}_k) = \frac{\boldsymbol{d}_k}{\max\left(\|\boldsymbol{d}_k\|_2, 1\right)} \ .$$

At each iteration, the proximal/projected gradient descent algorithm performs a gradient step for the smooth and convex fidelity term i.e. the left term in 3.11. Then, it used the proximal operator of $\mathcal{I}_{\Omega}$ i.e. the projection, to compute the next point. Like ISTA, this algorithm can be accelerated using the Nesterov's momentum and is called Accelerated Proximal Gradient Descent.

2.2.2 Alternate direction method of multipliers

In their paper, Bristow et al. [2013] also introduced a method for the dictionary update based on ADMM. As for the proximal gradient descent algorithm, this method first introduces the indicator function $\mathcal{I}_{\Omega}$ for the constraint. Then, it splits the objective function in two groups of variables $\{\boldsymbol{d}_k\}, \{\tilde{\boldsymbol{d}}_k\}$ and constrains these variables to be equal. Problem (3.10) becomes

$$\begin{aligned} \min_{\{\boldsymbol{d}_k\},\{\tilde{\boldsymbol{d}}_k\}} \ & \frac{1}{2}\sum_{n=1}^{N}\|\boldsymbol{y}_n - \sum_{k=1}^{K}\boldsymbol{d}_k \circledast \boldsymbol{z}_{n,k}\|_2^2 + \mathcal{I}_{\Omega}(\tilde{\boldsymbol{D}}) \\ \text{s.t.} \quad & \boldsymbol{d}_k = \tilde{\boldsymbol{d}}_k \quad \forall k = 1, \dots, K \ . \end{aligned}$$

The update is made as for ADMM-based solver for CSC (see Subsection 2.1.1).

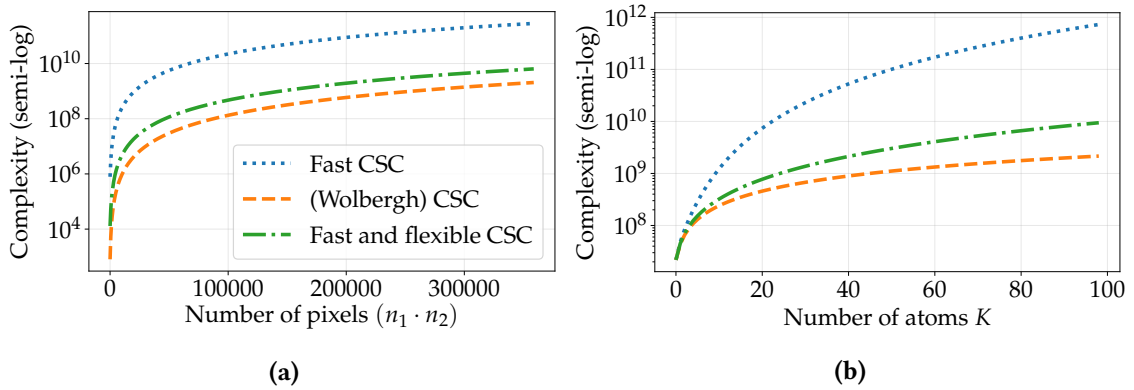


Figure 3.4: Evolution of the complexity (in semi-log) for data in $\mathbb{R}^{n_1 \times n_2 \times 3}$ (color images) when (a) n_1, n_2 vary and $K = 64$ (b) K varies and $n_1 = n_2 = 500$. Only the Fourier-based methods are reported.

2.2.3 Block coordinate descent, K-SVD

Mairal et al. [2010] proposed an algorithm based on the block coordinate descent. The block coordinate descent updates at each iteration only one of the dictionary atoms with all the other fixed. The atoms are updated using the coordinate-wise proximal gradient descent step.

Aharon et al. [2006] proposed a method based on the computation of K Singular Value Decomposition to update the dictionary. This algorithm can be seen as an extension of the K-Means algorithm and it has been adapted for convolutional dictionary learning in [Yellin et al., 2017].

2.3 Comparison of the solvers in the convolutional setting

While we only present the two leading CSC solvers in the above sections, there exist other algorithms build upon them which improve their theoretical algorithmic complexity. We collected all of them in Table 3.1. We also displayed the evolution of their theoretical complexity for typical dimension in Figure 3.4.

Up to date, the most effective algorithm in term of theoretical complexity is due to Wohlberg [2015] and is based on ADMM. They show that the complexity of solving the linear system (3.8) can be reduced to $\mathcal{O}(KM)$ with a careful analysis of the matrices involved and the use of the Sherman-Morrison formula [Sherman and Morrison, 1950]. However, the comparative review made by Garcia-Cardona and Wohlberg [2018a] indicates a very wide range of performances across the existing methods. For example, their results show that FISTA with frequency domain computation of the gradient is a viable alternative to ADMM-based solvers. In term of scalability, they show that methods based on FISTA or with parallel implementation are scalable to relatively large training sets, e.g. 100 images of 512×512 pixels. Finally, they note that while the computation time seems to only increase linearly with the number of training inputs and the number of dictionary atoms, the increase is more than linear with the size of the inputs, preventing the use of these methods for large inputs.

2.4 Theoretical guarantees for convolutional representation

Since the convolutional setting is equivalent to the vectorial case, previous works on DL can be directly applied for CDL. The objective function of the DL is not jointly convex. Thus, the

alternate minimization approach is not guarantee in general to converge to a global minimum. Furthermore, the problem has several symmetries and admits multiple global optima (possibility of arbitrary permutation, sign ambiguities, etc.) [Mairal et al., 2014]. The first theoretical studies of alternate minimization for DL were for vectorial data. Agarwal et al. [2016] show that, under certain conditions (enough samples and observed signals do not have noise or outliers), if data are generated using a dictionary, there exists a polynomial time algorithm which permits to estimate this dictionary. Under presence of noise and outlier points, Gribonval et al. [2015] show the sample complexity of dictionary retrieval methods and quantify the effect of the assumptions made in the model. These results can be improved for CDL by taking into account the particular structure of the data. Papyan et al. [2017b] introduce quantities which extend the different concepts used in sparse coding literature to convolutional settings and highlights the properties of dictionary elements critical for the uniqueness of the coding signal. In their second paper, Papyan et al. [2016] study the recovery capacities of classical convolutional sparse coding algorithms for noisy observations. Note that recent advances on *deconvolution model* also provide theoretical guarantees of reconstruction but only when there is a single atom (see e.g. [Zhang et al., 2017; Kuo et al., 2019; Lau et al., 2019; Qu et al., 2019b,a; Qu et al.; Shi and Chi, 2019, 2020]).

3 Tensor-based convolutional dictionary learning

Although the CDL problem for univariate data or images is widely study and well understood, its ability to take into account multivariate data is not well established. The generalization of the CDL problem to more dimensions can naturally be studied through the lens of tensor algebra. Indeed, this particular algebra provides an efficient framework to manipulate such data (see Appendix 9 for some remainders). One of the most important notion from tensor algebra is undoubtedly the generalization of the matrix rank which allows to efficiently take into account (or constrained) the underlaying structure of the tensor. However, this extension to multivariate signals is not trivial and several issues appear e.g. non-unity of the notion of rank, apparition of symmetries, non-convexity of the CSC problem. To deal with these issues, in the following, we carefully describe each component of our optimization problem. It includes additional constraints mandatory to obtain good results. Furthermore, as the number of parameters increases exponentially with the number of modes, we describe efficient procedure to reduce the complexity of the algorithms and handle such amount of data.

We now present how we extend the CDL problem to tensor data in order to take advantage of the underlying structure of this particular object.

Let $\boldsymbol{\mathcal{Y}}_1, \dots, \boldsymbol{\mathcal{Y}}_N \in \mathbb{Y} \triangleq \mathbb{R}^{n_1 \times \cdots \times n_p}$ be N tensor inputs of order $p > 0$ i.e. multidimensional signals. We define the regularized *Kruskal Convolutional Dictionary Learning problem* (K-CDL) as

$$\min_{\{\boldsymbol{\mathcal{D}}_k \boldsymbol{\mathcal{Z}}_{n,k}\}} \frac{1}{2} \sum_{n=1}^{N} \left(\left\| \boldsymbol{\mathcal{Y}}_n - \sum_{k=1}^{K} \boldsymbol{\mathcal{D}}_k \circledast_{1,\cdots,p} \boldsymbol{\mathcal{Z}}_{n,k} \right\|_F^2 + \varphi\left(\boldsymbol{\mathcal{Z}}_{n,1}, \cdots, \boldsymbol{\mathcal{Z}}_{n,K}; \boldsymbol{\alpha}\right) + \psi\left(\boldsymbol{\mathcal{Z}}_{n,1}, \cdots, \boldsymbol{\mathcal{Z}}_{n,K}; \boldsymbol{\beta}\right) \right) \tag{3.12}$$

$$\text{s.t.} \quad \begin{cases} \text{CP-rank}(\boldsymbol{\mathcal{Z}}_{n,k}) \leq R & \forall n, k\ , \quad \text{(a)} \\ \boldsymbol{\mathcal{D}}_k \in \mathbb{D}, \|\boldsymbol{\mathcal{D}}_k\|_F \leq 1 & \forall k\ , \quad \text{(b)} \end{cases}$$

with $\varphi(\cdot)$ a sparsity regularization, $\psi(\cdot)$ a regularization explained below, and $\boldsymbol{\alpha}, \boldsymbol{\beta} \succeq 0$ two vectors of hyperparameters.

In this formulation, the $\{\boldsymbol{\mathcal{Z}}_{n,k}\} \in \mathbb{Y}$ are (multivariate) sparse activation maps which specify where the (multivariate) atoms $\{\boldsymbol{\mathcal{D}}_k\}$, in $\mathbb{D} \triangleq \mathbb{R}^{w_1 \times \cdots \times w_p}$, $(w_1 \leq n_1, \cdots, w_p \leq n_p)$, are placed in the input signals. To take advantage of the tensor structure, we add a Canonical Polyadic (CP) low-rank constraint (3.12 a) on the activation maps. The formulation of the K-CDL problem therefore relies on four important constraints and regularizations explained bellow.

The CP low-rank constraint (a). This constraint controls the linear links between the different modes of the activations maps and thus takes into account the structure of the data. In the following, we choose to embed this constraint using the Kruskal operator $[\![\cdot]\!]$ (see Definition 3.4 in Appendix). Hence, each activation $\boldsymbol{\mathcal{Z}}_{n,k}$ is replaced by $[\![\boldsymbol{Z}_{n,k,1}, \cdots, \boldsymbol{Z}_{n,k,p}]\!]$ where the $\{\boldsymbol{Z}_{n,k,q}\}$ are in $\mathbb{R}^{n_q \times R}$. This approach is the generalization of the *Burer-Monteiro heuristic* for matrix [Burer and Monteiro, 2003].

The unit-ball constraints (b). The constraint on the $\{\boldsymbol{\mathcal{D}}_k\}$ prevents the scaling indeterminacy between the atoms and the activations as in the standard CDL. While in this chapter we only consider the unit-ball constraint, it can be easily modified to learn dictionaries with other structures.

The sparsity regularization $\varphi(\cdot)$. The regularization $\varphi(\cdot)$ on the activations is here to advantage sparse solutions. There is multiple ways to induce this sparsity. One popular choice in tensor regression is to add an ℓ_1-norm over the Kruskal operator of each activations in the objective function. However, this may leads to a complicated optimization problem [Chen et al., 2012; Tan et al., 2012]. Another popular choice is to impose sparsity on each Rank-1 component of the CP decomposition of the activations i.e.

$$\varphi : \underbrace{(\boldsymbol{Z}_{n,1,1}, \cdots, \boldsymbol{Z}_{n,K,p}; \boldsymbol{\alpha})}_{\cong\, (\boldsymbol{\mathcal{Z}}_{n,1}, \cdots, \boldsymbol{\mathcal{Z}}_{n,K}; \boldsymbol{\alpha})} \longmapsto \sum_{k=1}^{K} \sum_{r=1}^{R} \boldsymbol{\alpha}_{k,r} \left\| \boldsymbol{Z}_{n,k,1}(r,:) \circ \cdots \circ \boldsymbol{Z}_{n,k,p}(r,:) \right\|_1 \quad \text{with} \quad \boldsymbol{\alpha} \succeq 0\ , \tag{3.13}$$

where the $\{\boldsymbol{Z}_{n,k,q}\} \in \mathbb{R}^{n_q \times R}$ are the one from the CP decomposition. This constraint can be beneficial in multiple ways as discuss by He et al. [2018]. Nevertheless, as the CP decomposition may not be unique, the problem may suffer from parameter identifiability issues [Mishra et al., 2017]. Moreover, this is not a separable function with respect to the CP components $\{\boldsymbol{Z}_{n,k,q}\}$. Regarding these issues, we propose a regularization constraint called *Mode sparsity constraint*, defined by

$$\varphi : (\boldsymbol{Z}_{n,1,1}, \cdots, \boldsymbol{Z}_{n,K,p}; \boldsymbol{\alpha}) \longmapsto \sum_{k=1}^{K} \sum_{q=1}^{p} \boldsymbol{\alpha}_{k,q} \left\| \boldsymbol{Z}_{n,k,q} \right\|_1 \quad \text{with} \quad \boldsymbol{\alpha} \succeq 0\ . \tag{3.14}$$

This constraint induces the sparsity of each element of the CP-decomposition for every activations independently. The sparsity in each mode is therefore controlled without the impact of the other

modes i.e. the regularization (and not the objective function) is separable in each $\{\boldsymbol{Z}_{n,k,q}\}$. One additional advantage is that the multi-convolutional operator is well-adapted to such property of separability. When necessary, we can also add a positive constraint on the activation maps.

Identifiability and the $\psi(\cdot)$ constraint. The CP decomposition is known to be unique when it satisfies the Kruskal condition [Kruskal, 1989], but only up to permutation of the normalized factor matrices. In other words, the CP decomposition is unchanged by scaling or permutation, and the $\{\boldsymbol{Z}_{n,k,q}\}$ that solve equation (3.12) may not be unique. The scaling indeterminacy makes the optimization difficult as there is a continuous manifold of equivalent solutions. This difficulty is handled in (3.12) via $\psi(\cdot)$, a ridge-based penalization (e.g. $\sum_{q=1}^{p}\sum_{k=1}^{K}\boldsymbol{\beta}_{n,k,q}\|\boldsymbol{Z}_{k,q}\|_F^2$, $(\boldsymbol{\beta}_{1,1},\cdots,\boldsymbol{\beta}_{K,p})\succeq 0$) to (3.12) (see [Acar et al., 2011] and [Paatero, 1997]). On the contrary, the minimizers up to a permutation are isolated equivalent minimizers, and thus do not negatively impact the optimization [Acar et al., 2011].

Remark 3.2. *When $R=1$, the representation induced by the K-CDL is closed to the "Low rank tensor deconvolution" from Phan et al. [2015].*

Remark 3.3. *In recent tensor regression works, some authors prefer to add a combination of trace norm and ℓ_1-norm in the objective function to automatically infer the rank [Song and Lu, 2017]. However, Bengua et al. [2017] showed that the trace norm may not be appropriate for capturing the global correlation of a tensor leading us to our solution. Furthermore, we will see that the use of the Kruskal operator allows to split the K-CDL problem into smaller problems with less complexity and parameters to infer.*

In the following we are mostly interesting in solving the K-CDL problem with atoms fixed i.e. the Kruskal-CSC (K-CSC) problem.

Given a signal $\boldsymbol{\mathcal{Y}}$, and with regard to the previous remarks, the *elastic-net* K-CSC problem is

$$\min_{\{[\![\boldsymbol{Z}_{k,1},\cdots,\boldsymbol{Z}_{k,p}]\!]\}_k} \frac{1}{2}\left\|\boldsymbol{\mathcal{Y}}-\sum_{k=1}^{K}\boldsymbol{\mathcal{D}}_k \circledast_{1,\cdots,p}[\![\boldsymbol{Z}_{k,1},\cdots,\boldsymbol{Z}_{k,p}]\!]\right\|_F^2 + \sum_{q=1}^{p}\boldsymbol{\alpha}_q\sum_{k=1}^{K}\|\boldsymbol{Z}_{k,q}\|_1+\sum_{q=1}^{p}\boldsymbol{\beta}_q\sum_{k=1}^{K}\|\boldsymbol{Z}_{k,q}\|_F^2\,, \tag{3.15}$$

where the $\{\boldsymbol{Z}_{k,q}\}$ are in $\mathbb{R}^{n_q\times R}$ and the $\|\cdot\|_F^2$ is added to improve the minimization process, as previously discussed.

For simplicity, for all q we have set $\boldsymbol{\alpha}_{1,q}=\cdots=\boldsymbol{\alpha}_{K,q}$, and $\boldsymbol{\beta}_{1,q}=\cdots=\boldsymbol{\beta}_{K,q}$. Furthermore, in the following we set $N=1$.

4 Resolution of the problem

Even though the K-CDL problem (3.12) is not convex, it is convex with respect to each of the Z-block $\{(\boldsymbol{Z}_{1,q},\cdots,\boldsymbol{Z}_{K,q})\}_{q=1}^{p}$, or D-block $(\boldsymbol{\mathcal{D}}_1,\cdots,\boldsymbol{\mathcal{D}}_K)$ when the other ones are fixed. Furthermore, the two regularizations are separable with respect to these blocks. A natural optimization scheme for minimizing the objective function is therefore to use a *block-coordinate strategy* or *alternating minimization* [Hildreth, 1957; Ortega and Rheinboldt, 2000; Nikolova and Tan, 2017]. The main idea is to split the main non-convex problem into several convex subproblems;

1) by freezing the D-block and all except one Z-block at a time (referred as $\boldsymbol{\mathcal{Z}}$-step) 2) by only freezing all the Z-blocks (referred as $\boldsymbol{\mathcal{D}}$-step). Although this algorithm monotonically decreases the objective function, a stationary point is not guaranteed to be a local minimum (it can be a saddle point). Fortunately, we will see that in practice the block relaxation algorithm almost always converges to at least a local minimum.

The $\boldsymbol{\mathcal{Z}}$-step or activations update. To solve (3.15), we also use an iterative strategy. For q varying between 1 and p, we consider

$$\min_{\boldsymbol{Z}_{1,q},\cdots,\boldsymbol{Z}_{K,q}} \frac{1}{2}\left\|\boldsymbol{\mathcal{Y}} - \sum_{k=1}^{K} \boldsymbol{\mathcal{D}}_k \circledast_{1,\cdots,p} [\![\boldsymbol{Z}_{k,1},\cdots,\boldsymbol{Z}_{k,q},\cdots \boldsymbol{Z}_{k,p}]\!]\right\|_F^2 + \boldsymbol{\alpha}_q \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q}\|_1 + \boldsymbol{\beta}_q \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q}\|_F^2 \,. \tag{3.16}$$

One basic solution is to rewrite the problem as a regression one (without the convolution) and to use tensor regression solvers [Zhou et al., 2013; Li et al., 2017; He et al., 2018]. However, it requires the construction of a very large circulant tensor which is not tractable in practice due to memory limitation. In the following, we propose two efficient algorithms based on either ADMM or FISTA to solve (3.16).

Let first introduce two functions and three important properties which will be useful in the following.

$$f\left(\{\boldsymbol{Z}_{k,q}\}_{k,q=1}^{K,p}\right) = \frac{1}{2}\left\|\boldsymbol{\mathcal{Y}} - \sum_{k=1}^{K} \boldsymbol{\mathcal{D}}_k \circledast_{1,\cdots,p} [\![\boldsymbol{Z}_{k,1},\cdots,\boldsymbol{Z}_{k,p}]\!]\right\|_F^2, \tag{3.17}$$

$$g\left(\{\boldsymbol{Z}_{k,q}\}_{k=1}^{K}\right) = \boldsymbol{\alpha}_q \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q}\|_1 + \boldsymbol{\beta}_q \sum_{k=1}^{K} \|\boldsymbol{Z}_{k,q}\|_F^2 \,. \tag{3.18}$$

In this equation, f is *the fidelity term* that controls the difference between the input and its reconstruction, and g is the summation of the regularizations.

Lemma 3.1. (Mode-wise DFT) – *Given the CP-decomposition of a tensor $\boldsymbol{\mathcal{X}} = [\![\boldsymbol{X}_1,\cdots,\boldsymbol{X}_p]\!]$, the DFT can be performed mode-wise, i.e.*

$$\widehat{\boldsymbol{\mathcal{X}}} = \sum_{r=1}^{R} \widehat{\boldsymbol{x}}_r^{(1)} \circ \cdots \circ \widehat{\boldsymbol{x}}_r^{(p)} \triangleq [\![\widehat{\boldsymbol{X}}_1,\cdots,\widehat{\boldsymbol{X}}_p]\!] \,. \tag{3.19}$$

The complexity of computing $(\widehat{\boldsymbol{X}}_1,\cdots,\widehat{\boldsymbol{X}}_p)$ using the FFT goes from $\boldsymbol{\mathcal{O}}(\prod_{i=1}^{p} n_i \log(\prod_{i=1}^{p} n_i))$ to $\boldsymbol{\mathcal{O}}(R\sum_{i=1}^{p} n_i \log(n_i)))$. Notice that the DFT is only performed on the second dimension of each factor matrix, i.e. $\widehat{\boldsymbol{X}}_q = [\widehat{\boldsymbol{X}_q(:,1)} \mid \ldots \mid \widehat{\boldsymbol{X}_q(:,R)}]$.

We see from this lemma the important advantage of separable signals over non-separable ones in term of complexity.

Theorem 3.1. (Equality in the Fourier domain) – *The orthogonality of the Fourier basis implies a Plancherel formula. Therefore, in the Fourier domain, the fidelity term $f(\cdot)$ is equal to*

$$f\left(\{\boldsymbol{Z}_{k,q}\}_{k,q=1}^{K,p}\right) = \frac{1}{2\prod_{i=1}^{p} N_i}\left\|\widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^{K}\widehat{\boldsymbol{\mathcal{D}}}_k * [\![\widehat{\boldsymbol{Z}}_{k,1},\cdots,\widehat{\boldsymbol{Z}}_{k,p}]\!]\right\|_F^2 \tag{3.20}$$

$$\triangleq \frac{1}{\prod_{i=1}^{p} N_i}\widehat{f}\left(\{\widehat{\boldsymbol{Z}}_{k,q}\}_{k,q=1}^{K,p}\right) , \tag{3.21}$$

where $\widehat{\cdot}$ denotes the frequency representation of a signal, $*$ is the component-wise product, and $\widehat{f}$ denotes the fidelity term in the Fourier domain up to the factor $\frac{1}{\prod_{i=1}^{p} N_i}$.

Corollary 3.1. (A compact vectorized formulation) – *The following equality holds*

$$\widehat{f}\left(\{\widehat{\boldsymbol{Z}}_{k,q}\}_{k,q=1}^{K,p}\right) = \frac{1}{2}\left\|\widehat{\boldsymbol{y}}^{(q)} - \widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}}\otimes\boldsymbol{I})\widehat{\boldsymbol{z}}^{(q)}\right\|_F^2 , \tag{3.22}$$

where $\widehat{\boldsymbol{y}}^{(q)}$ is the vectorization of the folding of $\widehat{\boldsymbol{\mathcal{Y}}}$ along the dimension q, $\widehat{\boldsymbol{z}}^{(q)} = [\widehat{\boldsymbol{z}}_1^{(q)\intercal},\ldots,\widehat{\boldsymbol{z}}_K^{(q)\intercal}]^\intercal$ where $\forall k, \widehat{\boldsymbol{z}}_k^{(q)}$ is the vectorization of the matrix $\widehat{\boldsymbol{Z}}_{k,q}$,$\widehat{\boldsymbol{\Gamma}} = [diag(\widehat{\boldsymbol{d}_1}^{(n)}),\ldots,diag(\widehat{\boldsymbol{d}_K}^{(n)})]$ with $\boldsymbol{d}_k^{(q)}$ the vectorization of the folding of $\widehat{\boldsymbol{\mathcal{D}}_k}$ along the dimension q, and

$$\widehat{\boldsymbol{A}} = \begin{pmatrix}\widehat{\boldsymbol{B}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{B}}_K\end{pmatrix} \quad \text{where} \quad \widehat{\boldsymbol{B}}_k = (\overset{\leftarrow p}{\odot}_{i=1,i\neq q}\,\widehat{\boldsymbol{Z}}_{k,i}) . \tag{3.23}$$

Here, $\widehat{\boldsymbol{\Gamma}} \in \mathbb{C}^{n_1\cdots n_p \times K n_1\cdots n_p}$, $\widehat{\boldsymbol{A}} \in \mathbb{C}^{K\prod_{1,i\neq q}^{p} n_i \times KR}$, $\boldsymbol{I} \in \mathbb{R}^{n_q\times n_q}$, and $\widehat{\boldsymbol{z}}^{(q)} \in \mathbb{C}^{KRn_q}$. Thus, the design matrix $\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}}\otimes\boldsymbol{I})$ is in $\mathbb{C}^{n_1\cdots n_p\times KRn_q}$.

4.1 T-ConvADMM: ADMM-based solver for K-CSC

We now introduce an ADMM-based solver for the K-CSC (3.15). Considering the previous splitting of the objective function, the iterations of the ADMM algorithm with a scalar $\rho > 0$ and $\{\boldsymbol{U}_k\}$ as dual variables are given by

$$\{\boldsymbol{Z}_{k,q}^{(s+1)}\} = \arg\min_{\{\boldsymbol{Z}_{k,q}\}} f(\{\boldsymbol{Z}_{k,q}\}) + \frac{\rho}{2}\sum_{k=1}^{K}\|\boldsymbol{Z}_{k,q} - \boldsymbol{T}_k^{(s)} + \boldsymbol{U}_k^{(s)}\|_F^2 , \tag{3.24}$$

$$\{\boldsymbol{T}_k^{(s+1)}\} = \arg\min_{\{\boldsymbol{T}_k\}} g(\{\boldsymbol{T}_k\}) + \frac{\rho}{2}\sum_{k=1}^{K}\|\boldsymbol{Z}_{k,q}^{(s+1)} - \boldsymbol{T}_k + \boldsymbol{U}_k^{(s)}\|_F^2 , \tag{3.25}$$

$$\{\boldsymbol{U}_k^{(s+1)}\} = \boldsymbol{U}_k^{(s)} + \boldsymbol{Z}_{k,q}^{(s+1)} - \boldsymbol{T}_k^{(s+1)} . \tag{3.26}$$

As g is fully separable, subproblem (3.25) admits the closed-form solution

$$\forall k = 1,\ldots,K, \qquad \boldsymbol{T}_k^{(s+1)} = \frac{1}{1+2\boldsymbol{\beta}_q/\rho}\mathcal{S}_{\boldsymbol{\alpha}_q/\rho}(\boldsymbol{Z}_{k,q}^{(s+1)} + \boldsymbol{U}_k^{(s)}) ,$$

where $\mathcal{S}_\gamma(\cdot)$ is the soft-thresholding operator. Subproblem (3.24) also admits a closed-form solution (with conditions). However, this solution is difficult to compute due to the size of the

Algorithm 3.3 T-ConvADMM, ADMM for K-CSC

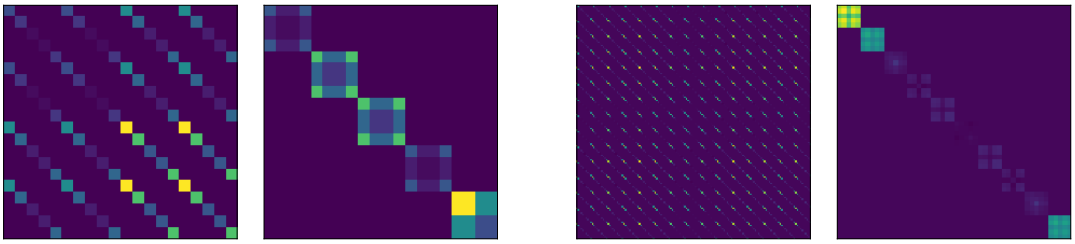
Input: signal $\boldsymbol{\mathcal{Y}}$, dictionary $\boldsymbol{\mathcal{D}}_1, \cdots, \boldsymbol{\mathcal{D}}_K$, regularization and ADMM parameters λ, ρ, tolerance ε
Precompute $\widehat{\boldsymbol{\mathcal{Y}}}$ and $\{\widehat{\boldsymbol{\mathcal{D}}}_k\}$
repeat
  for q in $\{1, \cdots, p\}$ **do**
    $\widehat{\boldsymbol{y}}^{(q)}, \{\widehat{\boldsymbol{d}}_k^{(q)}\} \longleftarrow \text{vec}(\widehat{\boldsymbol{Y}}^{(q)}), \{\text{vec}(\widehat{\boldsymbol{\mathcal{D}}}_k^{(q)})\}$
    Precompute $\{\widehat{\boldsymbol{Z}}_{k,i}\}_{k=1,i=1,i\neq q}^{K,p} \longleftarrow \{\text{DFT}(\boldsymbol{Z}_{k,i})\}_{k=1,i=1,i\neq q}^{K,p}$
    $\widehat{\boldsymbol{D}} \longleftarrow \widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$
    repeat
      ▷ Update of $\boldsymbol{Z}$ via equation (3.24)
        $\widehat{\boldsymbol{Z}}^{(s)}, \widehat{\boldsymbol{T}}^{(s)}, \widehat{\boldsymbol{U}}^{(s)} \longleftarrow \text{DFT}(\boldsymbol{Z}^{(s)}), \text{DFT}(\boldsymbol{T}^{(s)}), \text{DFT}(\boldsymbol{U}^{(s)})$
        $\widehat{\boldsymbol{z}}^{(s)}, \widehat{\boldsymbol{t}}^{(s)}, \widehat{\boldsymbol{u}}^{(s)} \longleftarrow \text{vec}(\widehat{\boldsymbol{Z}}^{(s)}), \text{vec}(\widehat{\boldsymbol{T}}^{(s)}), \text{vec}(\widehat{\boldsymbol{U}}^{(s)})$
        $\widehat{\boldsymbol{z}}^{(s+1)} \longleftarrow \text{Solve } \left(\widehat{\boldsymbol{D}}^H \widehat{\boldsymbol{D}} + \rho \boldsymbol{I}\right) \widehat{\boldsymbol{z}} = \left(\widehat{\boldsymbol{D}}^H \widehat{\boldsymbol{y}} + \rho(\widehat{\boldsymbol{t}}^{(s)} + \widehat{\boldsymbol{u}}^{(s)})\right)$
        $\widehat{\boldsymbol{Z}}^{(s+1)} \longleftarrow \text{Matricization of } \widehat{\boldsymbol{z}}^{(s+1)}$
        $\boldsymbol{Z}^{(s+1)} \longleftarrow \text{IDFT}(\widehat{\boldsymbol{Z}}^{(s+1)})$
      ▷ Update of $\boldsymbol{T}$ via equation (3.25)
        $\boldsymbol{T}^{(s+1)} \longleftarrow \text{prox}_{\rho,\alpha_q,\beta_q}(\boldsymbol{Z}^{(s+1)} + \boldsymbol{U}^{(s)})$
      ▷ Update of $\boldsymbol{u}$ via equation (3.26)
        $\boldsymbol{U}^{(s+1)} \longleftarrow \boldsymbol{U}^{(s)} + \boldsymbol{Z}^{(s+1)} - \boldsymbol{T}^{(s+1)}$
    until $\|\boldsymbol{Z}^{(s+1)} - \boldsymbol{Z}^{(s)}\|_\infty \leq \varepsilon$
  end for
until $\|\boldsymbol{\mathcal{Z}}^{(s+1)} - \boldsymbol{\mathcal{Z}}^{(s)}\|_\infty \leq \varepsilon$

matrices involved. One way to solve it efficiently is to exploit the Parseval's and convolution theorems (3.1) in order to take advantage of the convolutional structure of the problem (as in the univariate case). Using the previous propositions, the solution of (3.24) in the Fourier domain is given by the solution in $\widehat{\boldsymbol{z}}$ of

$$\left((\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I}) + \rho\boldsymbol{I}\right)\widehat{\boldsymbol{z}} = \left((\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{y}} + \rho(\widehat{\boldsymbol{t}} - \widehat{\boldsymbol{u}})\right) , \tag{3.27}$$

where $(\cdot)^H$ stands for the Hermitian transpose. The matrix $\left((\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I}) + \rho\boldsymbol{I}\right)$ is of size $KRn_q \times KRn_q$ which can be expensive to inverse. Fortunately, it has a particular diagonal block structure (see Figures 3.5a and 3.5c). Hence, we can permute rows and columns to only solve n_q independent $KR \times KR$ linear systems (see Figures 3.5b and 3.5d).

Complexity of T-ConvADMM. The complexity of T-ConvADMM is easily obtained by the analysis of each step. The pre-computation of the tensor $\widehat{\boldsymbol{\mathcal{Y}}}$ and $\{\widehat{\boldsymbol{\mathcal{D}}}_k\}$ is of complexity $\mathcal{O}((K+1)(M\log(M)))$ with $M = \prod_{i=1}^p n_i$. Then, given a particular mode q, we pre-compute the FFT of the remaining $\widehat{\boldsymbol{Z}}_{k,i}, (i \neq q)$. By Lemma (3.1), these operations have a complexity of $\mathcal{O}(KR(p-1)\sum_{i=1,i\neq q}^p n_i \log(n_i))$. Finally, as in the standard ADMM-based solvers, an analysis of the matrices involved leads to solve n_q linear systems of size KR. When using Gaussian



(a) (b) (c) (d)

Figure 3.5: Visualization of $(\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$ before and after a reordering. The two left matrices (a, b) correspond to the Gram matrix without (a) and with reordering (b). The two right matrices (c, d) also correspond to the Gram matrix without (c) and with reordering (d) but for a higher dimension.

elimination or Cholesky decomposition the complexity is therefore of $\mathcal{O}((KR)^3 n_q)$. However, it is possible to take advantage of iterative methods to reduced the complexity. Finally, the soft-threshold part and the dual variable updates are of complexity $\mathcal{O}(Kn_q)$. As we have this complexity for every modes, the overall complexity is $\mathcal{O}((KR)^3 \sum_{i=1}^p n_i)$.

4.2 T-ConvFISTA: FISTA-based solver for K-CSC

To solve the K-CSC (3.15) with FISTA, we introduce the following splitting

$$f\left(\{\boldsymbol{Z}_{k,q}\}_{k=1}^K\right) = \frac{1}{2}\left\|\boldsymbol{\mathcal{Y}} - \sum_{k=1}^K \boldsymbol{\mathcal{D}}_k \circledast_{1,\cdots,p} [\![\boldsymbol{Z}_{k,1}, \cdots, \boldsymbol{Z}_{k,p}]\!]\right\|_F^2 + \boldsymbol{\beta}_q \sum_{k=1}^K \|\boldsymbol{Z}_{k,q}\|_F^2 \tag{3.28}$$

$$= f_1\left(\{\boldsymbol{Z}_{k,q}\}_{k=1}^K\right) + f_2\left(\{\boldsymbol{Z}_{k,q}\}_{k=1}^K\right) \tag{3.29}$$

$$\varphi\left(\{\boldsymbol{Z}_{k,q}\}_{k=1}^K\right) = \boldsymbol{\alpha}_q \sum_{k=1}^K \|\boldsymbol{Z}_{k,q}\|_1 \ , \tag{3.30}$$

and alternate between i) a gradient descent on $f(\cdot)$, ii) the proximal operator over $\varphi(\cdot)$, and iii) the Nesterov's momentum. As φ is separable, its proximal operator is given for each $\boldsymbol{Z}_{k,q}$ by the soft-thresholding operator. The gradient descent step is performed in the Fourier domain. This "trick" decreases the complexity of the gradient computation. A nice formulation of the gradient in the Fourier domain is given by the following lemma.

Corollary 3.2. *The partial derivative of f_1 with respect to $\boldsymbol{Z}_{\ell,q}$ is given by*

$$\frac{\partial}{\partial \boldsymbol{Z}_{\ell,q}} f_1\left(\{\boldsymbol{Z}_{k,q}\}\right) = \mathit{IDFT}\left[\left(\left(\widehat{\boldsymbol{Y}}^{(q)} - \sum_{k=1}^K \widehat{\boldsymbol{D}}_k^{(q)} * [\![\widehat{\boldsymbol{Z}}_{k,1}, \cdots, \widehat{\boldsymbol{Z}}_{k,p}]\!]\right) * \overline{\widehat{\boldsymbol{D}}}_\ell^{(q)}\right)\overline{\boldsymbol{B}}_\ell\right] \ . \tag{3.31}$$

Using proposition (3.1), we also have a vectorial formulation for the gradient given by

$$\nabla_{\mathit{vec}(\{\boldsymbol{Z}_{k,q}\})} f_1\left(\{\boldsymbol{Z}_{k,q}\}\right) = \mathit{IDFT}\left[(\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\left(\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\widehat{\boldsymbol{z}}^{(q)} - \widehat{\boldsymbol{y}}^{(q)}\right)\right] \ . \tag{3.32}$$

Significant speed-up. There are several ways to improve the speed of this algorithm in a given implementation. For instance, the computation of $(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\widehat{\boldsymbol{z}}^{(q)}$ can be performed in $\mathcal{O}(KR\prod_{i=1}^p n_i)$ operations instead of $\mathcal{O}(KRn_q\prod_{i=1}^p n_i)$ (naive computation) by noticing that

$$(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\widehat{\boldsymbol{z}}^{(q)} = (\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\mathrm{vec}([\widehat{\boldsymbol{Z}}_{1,q} \mid \cdots \mid \widehat{\boldsymbol{Z}}_{K,q}]) = \mathrm{vec}([\widehat{\boldsymbol{Z}}_{1,q} \mid \cdots \mid \widehat{\boldsymbol{Z}}_{K,q}]\widehat{\boldsymbol{A}}^\mathsf{T}) \ .$$

In addition, we can exploit distributed computation by using a parallel matrix-vector multiplication. In our specific case where $\prod_{i=1}^{p} n_i \gg KRn_q$, we can precompute the Gram matrix $(\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\boldsymbol{\Gamma}^H\boldsymbol{\Gamma}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$ and $(\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{y}}^{(q)}$ to improve efficiency. All the work being in computing this Gram matrix which is now done only once.

These computations are also parallelizable using an all-reduce method. This means, for example, that the Gram matrix can be computed only keeping a single $(\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H)_{i,:}$ in working memory at a given time, so it is feasible to solve a lasso problem with extremely large $\prod_{i=1}^{p} n_i$ on a single machine, as long as KRn_q is modest [Parikh et al., 2014].

Proposition 3.1. *The matrix $(\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$ is composed of K^2 blocks equal to*

$$\left(\left(\overset{\leftarrow p}{\odot}_{i=1,i\neq q} \widehat{\boldsymbol{Z}}_{k,i}\right)^H \otimes I\right) \overline{diag(\widehat{\boldsymbol{d}}_k^{(q)})} diag(\widehat{\boldsymbol{d}}_\ell^{(q)}) \left(\left(\overset{\leftarrow p}{\odot}_{i=1,i\neq q} \widehat{\boldsymbol{Z}}_{\ell,i}\right) \otimes I\right) . \tag{3.33}$$

Each of these blocks can be computed in $\mathcal{O}(R^2 \prod_{i=1,i\neq q}^{p} n_i)$. Hence, the full matrix can be computed in $\mathcal{O}((KR)^2 \prod_{i=1,i\neq q}^{p} n_i)$ operations. Furthermore, this matrix is a $(KRn_q \times KRn_q)$ banded matrix (as explained before). Its product with $\widehat{\boldsymbol{z}}^{(q)}$ can therefore be made in only $\mathcal{O}((KR)^2 n_q)$ operations.

Complexity of T-ConvFISTA. The complexity of T-ConvFISTA is easily obtained by the analysis of each step. The pre-computation of the tensor $\widehat{\boldsymbol{\mathcal{Y}}}$ and $\{\widehat{\boldsymbol{\mathcal{D}}}_k\}$ is of complexity $\mathcal{O}((K+1)(M\log(M)))$ with $M = \prod_{i=1}^{p} n_i$. Then, given a particular mode q, we pre-compute the FFT of the remaining $\widehat{\boldsymbol{Z}}_{k,i}, (i \neq q)$. By Lemma (3.1), these operations have a complexity of $\mathcal{O}(KR(p-1)\sum_{i=1,i\neq q}^{p} n_i \log(n_i))$. Finally, we perform the gradient step in the frequency domain. Each computation of the gradient is of complexity $\mathcal{O}((KR)^2 n_q)$ if the Gram matrix is precomputed. The overall complexity is therefore dominating by $\mathcal{O}((KR)^2 n_q)$ for typical value of parameters. As we do this process for every mode, we obtain an overall complexity of $\mathcal{O}((KR)^2 \sum_{i=1}^{p} n_i)$.

4.3 Some additional remarks

Comparison of the complexity with previous CSC solvers. We collect the theoretical complexity of our two solvers in Table 3.2. In addition, a comparison of the evolution of the complexity between the standard Fourier-based solvers is displayed in Figure 3.6. The theoretical complexity of our tensor-based solvers is much smaller than the complexity of the other methods with a dominant term $\mathcal{O}((KR)^2 \max(n_i))$ instead of $\mathcal{O}(KM\log(M)) = \mathcal{O}(K\prod_{i=1}^{p} n_i \log(\prod_{j=1}^{p} n_j))$ for FCSC with iterative application of the Sherman-Morrison equation (FCSC-ShM) [Wohlberg, 2015] or even $\mathcal{O}(K\prod_{i=1}^{p} n_i \prod_{i=1}^{p} w_i)$ for LoBCoD (CSC) while being the most recent solver. As an example, for a multispectral images of size $(n_1 \times n_2 \times n_3) = (128 \times 128 \times 128)$ with 12 atoms and a rank set to $R = 3$, $(KR)^2(n_1+n_2+n_3) = 497,664$ while $Kn_1n_2n_3\log(n_1n_2n_3) = 366,316,018$.

Originality and advantages of the low-rank method To date, most works have focused only on the 2-D case with a low-rank constraint enforced on the atoms, i.e in the patterns observed in the data [Garcia-Cardona and Wohlberg, 2018b]. However, in several applicative contexts, data are multilinear and the low-rank structure naturally appears in the activations rather than in the atoms/dictionary. To take these observations into account, in the K-CDL we extend the standard

Algorithm 3.4 T-ConvFISTA (sub-problem)

Input: signal $\boldsymbol{\mathcal{Y}}$, dictionary $\boldsymbol{\mathcal{D}}_1, \cdots, \boldsymbol{\mathcal{D}}_K$, regularization and step parameters α, β, η ($\eta = 1/L$, the inverse of Lipschitz constant if calculate), tolerance ε
Initialization: $\boldsymbol{Z}^{(0)}$

Precompute: $\widehat{\boldsymbol{\mathcal{Y}}}$, $\{\widehat{\boldsymbol{\mathcal{D}}}_k\}$, $\boldsymbol{G}$ and $(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\widehat{\boldsymbol{y}}^{(q)}$

$t^{(0)} \longleftarrow 1$
repeat

▷ Update of $\boldsymbol{W}$ via a proximal gradient step (ISTA)
Compute $\widehat{\boldsymbol{Z}}^{(s)}$ using the FFT
$\widehat{\boldsymbol{z}}^{(s)} \longleftarrow \text{vec}(\widehat{\boldsymbol{Z}}^{(s)})$
$\widehat{\boldsymbol{w}}^{(s+1/2)} \longleftarrow \widehat{\boldsymbol{z}}^{(s)} - \eta \left(\boldsymbol{G}\widehat{\boldsymbol{z}}^{(s)} - (\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\widehat{\boldsymbol{y}}^{(q)} \right)$
$\widehat{\boldsymbol{W}}^{(s+1/2)} \longleftarrow$ Matricization of $\widehat{\boldsymbol{w}}^{(s+1/2)}$
Compute $\boldsymbol{W}^{(s+1/2)}$ using the IFFT

▷ Update of $\boldsymbol{W}$ via a proximal step (ISTA)
$\boldsymbol{W}^{(s+1)} \longleftarrow \text{prox}_{\eta,\alpha,\beta} \left(\boldsymbol{W}_k^{(s+1/2)} \right)$

▷ Nesterov momentum step (FISTA)
$t^{(s+1)} \longleftarrow \dfrac{1 + \sqrt{1 + 4 \cdot t^{(s)^2}}}{2}$
$\boldsymbol{Z}^{(s+1)} \longleftarrow \boldsymbol{W}^{(s+1)} + \dfrac{t^{(s)} - 1}{t^{(s+1)} + 1}(\boldsymbol{W}^{(s+1)} - \boldsymbol{W}^{(s)})$

until $\|\boldsymbol{Z}^{(s+1)} - \boldsymbol{Z}^{(s)}\|_\infty \leq \varepsilon$

CDL problem to a tensorial one with an additional low-rank CP decomposition constraint on the activation maps. This is an important modification both in term of representation and complexity implying five main advantages:

1. First, the low-rank constraint allows to exploit the underlying structural information of the input signals. This has already been proved to be very effective in various contexts from image processing to EEG signals decomposition (see e.g. [Guo et al., 2012; Liu et al., 2013]). In image processing for example, previous works have shown that the vectorization of an image removes the inherent spatial structure of it while a low rank tensor regression produces more interpretable results [Zhou et al., 2013].

2. Second, because the activations are decomposed in each mode, they are much more interpretable than those of the standard CDL. This is a mandatory property when working on complex data such as EEG recordings.

3. Third, low-rank constraints on activations entail a better robustness with respect to noise [Zhou et al., 2013; Zhao et al., 2011; Cong et al., 2015; Rabusseau and Kadri, 2016], which is one of the main weakness of the activation learning part of CDL [Simon and Elad, 2019].

Algorithm	Time complexity (Z-step)
T-ConvADMM	$T\Big(\underbrace{(KR)^3 n_q}_{\text{Linear system}} + \underbrace{KRn_q \log(M)}_{\text{FFTs}} + \underbrace{KRn_q}_{\text{Shrinkage}}\Big)$
T-ConvFISTA	$T\Big(\underbrace{(KR)^2 n_q}_{\text{Gradient}} + \underbrace{KRn_q \log(M)}_{\text{FFTs}} + \underbrace{KRn_q}_{\text{Shrinkage}}\Big)$

Table 3.2: T is the number of iterations, K the number of atoms, $M = \prod_{i=1}^{n_p} n_i$, and R the CP-rank.

4. A fourth advantage is the drastic reduction of the number of unknown activation parameters. Indeed, it goes from $K \prod_{i=1}^{p} n_i$ (unconstrained model) to $KR \sum_{i=1}^{p} n_i$. This reduction in dimension, and consequently in computational cost, is substantial.

5. Finally, the low-rank constraint imposes that each activation $\boldsymbol{\mathcal{Z}}_k$ can be written as the sum of at most R separable filters (product of multiple one dimensional filters). The K-CDL is therefore a *separable convolution problem*. This property allows to significantly speed up the calculus of the convolution and of the solvers (Section 4). Indeed, filtering an $(n_1 \times n_2)$ image with a $(w_1 \times w_2)$ non-separable atom is $\mathcal{O}(n_1 n_2 (w_1 + w_2))$. By contrast, it is instead of $\mathcal{O}(n_1 n_2 w_1 w_2)$ for a non-separable atom. This cost reduction becomes even more desirable when dealing with higher order inputs.

How to do the initialization? The initialization of the factor matrices $\{\boldsymbol{Z}_{k,q}\}$ can highly impact the performance of the algorithms. While there are many possible ways to do this initialization, one easy and effective approach is to choose random factor matrices, a strategy already used in the CP-ALS algorithm [Battaglino et al., 2018]. Notice that, unlike the standard initialization of FISTA with vector of zeros, we must choose random factor matrices without to much sparsity. Indeed, at each step of the algorithms, we construct a "new dictionary" $\left(\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})\right)$ based on the factors. Hence, if some initial factors are too sparse, this new dictionary contains a lot of zeros and we may not be able to solve our problem properly. One extreme case is when we choose all factor matrices equal to zeros. The new dictionary is then filled with zeros and we cannot find a solution of the global problem. In the following we initialize the $\{\boldsymbol{Z}_{k,q}\}$ with random Uniform matrices.

Simultaneously sparse and low-rank. It has been shown in [Richard et al., 2012] that, being low-rank is not an equivalent of sparsity for matrices, but that being low-rank and sparse can actually be seen as two orthogonal concepts. However, while the estimation of simultaneously sparse and low-rank matrices could be desirable, a balance between the two constraints has to be found as the two regularizations may have adversarial influence. In our setting, this is achieved by using a Ivanov regularization for the rank (CP-rank $\leq R$) and a Tykhonov regularization for the sparsity (e.g. $\sum_{k,q} \boldsymbol{\alpha}_{k,q} \|\boldsymbol{Z}_{k,q}\|_1$). This means that the solution should be as sparse as possible while having a CP-rank less than or equal to R.

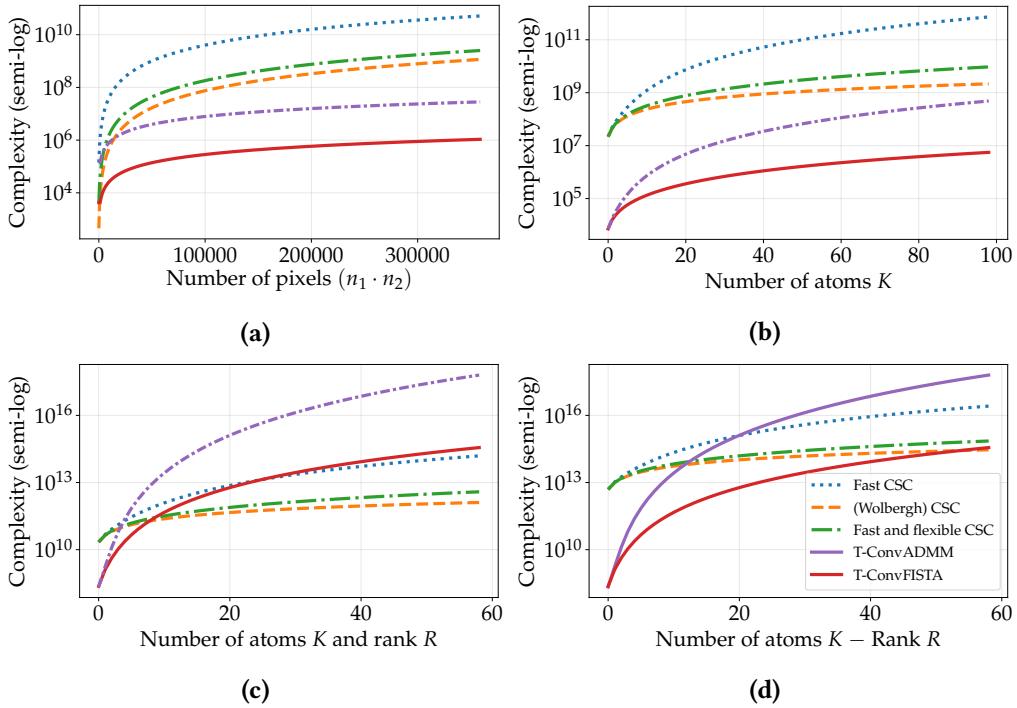


Figure 3.6: Evolution of the theoretical complexity (in semi-log) for data in $\mathbb{R}^{n_1 \times n_2 \times 3}$ (color images) when (a) n_1, n_2 vary and $K = 36, R = 1$, (b) K varies and $n_1 = n_2 = 500, R = 1$, (c) $K = R$ varies and $n_1 = n_2 = 500$, and (d) $K = R$ varies and $n_1 = n_2 = n_3 = 500$ (multispectral image). For (c) and (d), we set the number of iterations of standard methods to 1000 and for our methods to 500 for the number of inner iterations and 20 for the other. Recall that all complexity are given without taking into account the sparsity. This is therefore the worst complexity possible. Only the Fourier-based methods are reported.

4.4 Dictionary update, $\mathcal{D}$-step.

Given the activations $\{\boldsymbol{\mathcal{Z}}_{n,k}\}$, the dictionary update aims at improving how the model reconstructs the inputs $\boldsymbol{\mathcal{Y}}_1, \cdots, \boldsymbol{\mathcal{Y}}_N$ by solving

$$\min_{\forall k, \boldsymbol{\mathcal{D}}_k \in \mathbb{D}, \ \|\boldsymbol{\mathcal{D}}_k\|_F \leq 1} \frac{1}{2} \sum_{n=1}^{N} \left\| \boldsymbol{\mathcal{Y}}_n - \sum_{k=1}^{K} \boldsymbol{\mathcal{D}}_k \circledast_{1,\cdots,p} \boldsymbol{\mathcal{Z}}_{n,k} \right\|_F^2. \tag{3.34}$$

This step presents no significant difference with existing methods. The problem is smooth and convex and can be solved using the algorithms presented in Section 2.

5 Related works

We now briefly present some methods related to the CDL problem or its variants to better understand where our contribution lies in this vast literature. We collect on Table 3.3 a selective list of algorithms. We divided this list in three categories. The fist one contains algorithms for the standard CDL problem of Section 2. Complete reviews are provided in [Wohlberg, 2015] and [Garcia-Cardona and Wohlberg, 2018a]. The second category contains very recent algorithms taking into consideration the separability/rank of a (2-D) dictionary. Finally, the last category contains our two algorithms.

Method	Solver	Rank constraint	Application
Category 1 – Standard			
(Lewicki and Sejnowski [1999] 1999)		None	Representation of 1-D speech data
FS-EXACT (Grosse et al. [2007] 2007)	Feature Search	None	Classification of 1-D audio signals
DeconvNet (Zeiler et al. [2010] 2010)		None	2-D image representation/denoising
Kavukcuoglu et al. [2010]	Coordinate Descent	None	2-D image representation
ConvFISTA (Chalasani et al. [2013] 2013)	FISTA	None	2-D image representation
FCSC (Bristow et al. [2013] 2013) Kong and Fowlkes 2014)	ADMM	None	2-D image representation
FCSC-SM (Wohlberg [2014, 2015] 2014, 2016)	ADMM	None	2-D image representation
FFCSC (Heide et al. [2015] 2015)	ADMM	None	2-D image representation
ALS-CTD (Huang and Anandkumar [2015] 2015)	Alternating Least-Square	None	1-D synthetic data
CONSENSUS (Šorel and Šroubek [2016] 2016)	ADMM	None	2-D image representation
SBDL (Papyan et al. [2017a] 2017)	Local-ADMM	None	2-D image inpainting/separation
DICOD (Moreau et al. [2018] 2018)	Coordinate Descent	None	1-D synthetic data
LoBCoD (Zisselman et al. [2019] 2019)	Coordinate Descent	None	2-D image inpainting/fusion
Category 2 – Rank constraint on the dictionary			
SEP-COMB (Rigamonti et al. [2013] 2013)		Tensor: CP-rank	N-D signal representation
SEP-COMB + SEP-TD (Sironi et al. [2014] 2014)		Tensor: CP-rank	N-D signal representation
(Silva et al. [2017] 2017)	FISTA / ADMM	Matrix: rank	2-D image representation
(Quesada et al. [2018] 2018)		Matrix: rank	2-D image representation
Pair-SepF (Silva et al. [2018] 2018)	Accelerated Proximal Gradient	Matrix: rank	2-D image representation
(Dupré La Tour et al. [2018] 2018)	Coordinate descent	Matrix: rank	Electromagnetic Brain Signals
Comb-SepF (Quesada et al. [2019] 2019)		Matrix: rank	2-D image representation
Our methods – Rank constraint on the activations			
T-ConvADMM (2020)	ADMM	Tensor: CP-rank	N-D signal representation
T-ConvFista (2020)	FISTA	Tensor: CP-rank	N-D signal representation

Table 3.3: Selective list of CDL/CSC solvers.

Standard CDL. The notion of *translation invariant representation* of a signal was proposed by Simoncelli et al. [1992] after they observed that block-based wavelet algorithms were sensitive to translation and scaling of the input signal. Later, Lewicki and Sejnowski [1999] proposed an algorithm to find this efficient representation by inferring the best temporal positions of given 1-D functions in a kernel basis i.e. dictionary. Their main idea was to infer the values and temporal locations of the non-zero coefficients and then to refine the result through a modified conjugate gradient local search. The generalization of the work of Lewicki and Sejnowski [1999] to a 2-D convolution is due to Grosse et al. [2007] and is now referred as the (multivariate) CDL problem presented in Section 2 (in contrast to the univariate CDL for 1-D atoms). In their paper, they first expressed the problem with a ℓ_1-norm regularization and convolutional constraints. Then, they used a frequency domain method combined with the feature sign search minimization algorithm [Lee et al., 2007]. While the efficiency of this representation has led to a wide range of applications, the large-scale nature of them has placed great demands on the computational efficiency of the algorithms. This has given rise to a range of optimization approaches for CSC and CDL. For instance, Chalasani et al. [2013] introduced a convolutional extension of the FISTA algorithm for sparse inference called ConvFISTA (see Section 2). Then, Bristow et al. [2013] proposed the FCSC algorithm, a Fourier method based on ADMM (see Section 2). The FCSC has been progressively improved in [Wohlberg, 2014, 2015; Šorel and Šroubek, 2016]. Up to now, the state-of-the-art algorithms always operate in the frequency domain to exploit the convolutional structure of the problem. However, while this is the first step towards making CDL practical, these frequency methods can introduce boundary artifacts. To address this issue, Heide et al. [2015] proposed to incorporate a particular matrix in the optimization problem. They derive a flexible formulation

and propose an efficient ADMM-based solution called FFCSC which splits the objective into a sum of simpler convex functions. Very recently, Papyan et al. [2017a]; Moreau et al. [2018]; Zisselman et al. [2019] proposed more localized strategies. Note that while their algorithms operate in the batch mode (i.e., all the samples have to be accessed in each iteration), recent works study online learning to improve scalability [Degraux et al., 2017; Liu et al., 2017; Wang et al., 2018b; Liu et al., 2018].

Standard CDL with a low-rank constraint on the dictionary. The idea of learning separable atoms in the multivariate CDL was first introduced in tensorial computer vision by Rigamonti et al. [2013] and Sironi et al. [2014]. They proposed two methods to learn high-order CP low-rank dictionary: a first one learns low-rank atoms thanks to a nuclear norm, the other learns low-rank atoms *a posteriori*. However, note that (i) both methods cannot be directly applied to learn low-rank activations as there is an additional sparsity constraint, (ii) in their formulation this is the full dictionary (called filters bank) which is assumed to be low-rank. Thus, the original atoms are approximated by a weighted sum of shared rank-1 atoms e.g. several two-dimensional atoms are stacked together to form a 3-dimensional tensor and this resulting tensor is decomposed in a sum of rank-1 tensors. Interestingly, in these two papers, they empirically showed that using separable atoms as dictionaries in CSC or convolutional neural network applications provides significant improvements in computational performance with respect to non-separable implementations, while giving little loss in accuracy or reconstruction quality. From this observation, very recently, some papers have re-focused on the 2-D multivariate CDL problem and assumed or learned separable/low-rank 2-D filter banks [Silva et al., 2017; Quesada et al., 2018; Silva et al., 2018; Dupré La Tour et al., 2018]. The first one, [Silva et al., 2017], introduced a computationally efficient algorithm when the dictionary atoms are given and already separable. The two others, [Quesada et al., 2018; Silva et al., 2018], proposed to directly learn the separable 2-D atoms. A slightly modification of this separable CDL problem is proposed by Quesada et al. [2019] where they empirically showed that this alternative formulation provides a reduction in computation time over the stantard CSC and CDL algorithms.

Tensor and dictionary. Instead of trying to extend the multivariate CDL to tensor, another approach is to use a tensor-based representation including particular tensor operations. In Bibi and Ghanem [2017], authors used the t-product (see definition in [Kilmer and Martin, 2011]) to provide another tensor CDL formulation that has the potential to uncover high dimensional correlation among channels, but is also computationally expensive. Finally, Jiang et al. [2018] and Gong et al. [2020] exploit other products such as the t-linear combination but do not consider convolutional models.

Relation between K-CDL and CDL With specific choices on the parameters or on the dimension values, the K-CDL problem reduces to well-known CDL ones. Hence, it can be seen as a generalization of several approaches in the literature.

- For vector-valued atoms and signals ($p = 1$), the K-CDL reduces to the 1-D CDL, known as univariate CDL, presented in Section 2.
- When $p > 1$ and $R = +\infty$ (i.e. no low-rank constraint), the K-CDL also reduces to the CDL presented in Section 2, known as Multivariate CDL.
- When $p = 2$, $R < +\infty$ and $w_2 = 1$, the K-CDL reduces to models which impose a matrix rank structure on the dictionary.

- Finally, when $R = 1$, the representation induced by the K-CDL is closed to the *Low rank tensor deconvolution model* from Phan et al. [2015] which is, however, not proposed as a CDL model.

6 Experiments

To illustrate and compare the effectiveness and efficiency of our two tensor-based solvers, we consider in this section a wide range of synthetic and real data. To make comparisons that are as fair as possible, each algorithm is implemented in Python using Tensorly [Kossaifi et al., 2019] (for tensor algebra in Python), Sporco [Wohlberg, 2017] (a Python package for convolutional sparse representations with some C/C++ modules), and standard python libraries. Furthermore, to save memory and reduce the time complexity, both methods are implemented with sparse matrix packages. We also compare our methods to the two leading batch CDL algorithms presented in the previous sections: FCSC with iterative application of the Sherman-Morrison equation [Bristow et al., 2013; Wohlberg, 2015], and ConvFISTA in the Fourier domain [Chalasani et al., 2013; Wohlberg, 2015]. They are both implemented in Sporco. All subsequent simulations are run on a machine through Linux/Ubuntu with 16-core of 2.5GHz Intel CPUs and 64GB of RAM.

For the convenience of the reader, we list here the CDL algorithms compared and the acronyms we use throughout this section: ADMM with tensor-based rank constraint (T-ConvADMM) of Section 4.1, FISTA with tensor-based rank constraint (T-ConvFISTA) of Section 4.2, FCSC with iterative application of the Sherman-Morrison equation (FCSC-ShM or FCSC for short) [Bristow et al., 2013; Wohlberg, 2015], FISTA in the Fourier domain (ConvFISTA) [Chalasani et al., 2013; Wohlberg, 2015]. Note that, based on the previous analysis of the complexity, we choose to use T-ConvFISTA with precomputation of the Gram matrix. For the dictionary update, we also use ADMM with iterative application of the Sherman-Morrison equation [Wohlberg, 2015].

6.1 Evaluation on synthetic data

We now present a large range of results on synthetic data.

Dataset. Small-scale and large-scale experiments are performed by considering two main different datasets:

- A *small-scale dataset* which contains 10 independent input signals of size $(25 \times 25 \times 25)$. Each signal is generated as follows. We draw $K = 3$ atoms of size $(5 \times 5 \times 5)$ according to an Uniform distribution with values in $[-1, 1]$ and normalize them. Then, we set the maximal CP-rank to $R^* = 2$ and draw sparse activations from a Bernoulli-Uniform distribution with Bernoulli parameter equal to 0.2, and range of values in $[-1, 1]$. Finally, we generate the input tensor according to the convolutional model induced by the K-CDL (3.12).

- A *large-scale dataset* which is generated as the small-scale dataset but with input signals of size $(128 \times 128 \times 128)$ and Bernoulli parameter equal to 0.02.

- These two dataset are extended with their noisy counterpart called *noisy small-scale dataset* and *noisy large-scale dataset*. Following Wohlberg [2015], for each input, we construct noisy input signals by adding Multivariate Gaussian noise of progressively high variance to obtain a Signal to Noise Ratio (SNR) with respect to the original input of $25.5, 9.5$, and 3.0dB corresponding to a noise's variance approximatively equals to $5.29e{-}6, 2.25e{-}4$ and, $9.00e{-}4$; (standard

CSC		*Small-scale dataset*		*Large-scale dataset*	
CP-rank	**Metrics**	**T-ConvADMM**	**T-ConvFISTA**	**T-ConvADMM**	**T-ConvFISTA**
$R=1$	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$\mathbf{0.016}\,(\pm\mathbf{0.005})$	$\mathbf{0.016}\,(\pm\mathbf{0.005})$	$\mathbf{0.025}\,(\pm\mathbf{0.002})$	$\mathbf{0.025}\,(\pm\mathbf{0.002})$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$\mathbf{0.013}\,(\pm\mathbf{0.003})$	$\mathbf{0.013}\,(\pm\mathbf{0.003})$	$\mathbf{0.016}\,(\pm\mathbf{0.003})$	$\mathbf{0.016}\,(\pm\mathbf{0.003})$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ ↑	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ ↑	0%	0%	0%	0%
$R=2$	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$1.346 \cdot e{-}7\,(\pm 8.996 \cdot e{-}8)$	$\mathbf{1.966 \cdot e{-}8}\,(\pm\mathbf{6.716 \cdot e{-}9})$	$\mathbf{7.575 \cdot e{-}11}\,(\pm\mathbf{5.528 \cdot e{-}12})$	$2.804 \cdot e{-}10\,(\pm 2.672 \cdot e{-}10)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$8.041 \cdot e{-}8\,(\pm 5.312 \cdot e{-}8)$	$\mathbf{1.261 \cdot e{-}8}\,(\pm\mathbf{4.025 \cdot e{-}9})$	$\mathbf{4.476 \cdot e{-}11}\,(\pm\mathbf{2.556 \cdot e{-}12})$	$1.736 \cdot e{-}10\,(\pm 1.723 \cdot e{-}10)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ ↑	94%	**96**%	80%	**85**%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ ↑	**96**%	**98**%	**90**%	**90**%
$R=3$	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$\mathbf{3.195 \cdot e{-}7}\,(\pm\mathbf{4.351 \cdot e{-}7})$	$7.126 \cdot e{-}7\,(\pm 2.348 \cdot e{-}7)$	$\mathbf{6.439 \cdot e{-}10}\,(\pm\mathbf{3.972 \cdot e{-}10})$	$1.771 \cdot e{-}8\,(\pm 7.898 \cdot e{-}8)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$\mathbf{1.954 \cdot e{-}7}\,(\pm\mathbf{2.533 \cdot e{-}7})$	$4.266 \cdot e{-}7\,(\pm 1.355 \cdot e{-}7)$	$\mathbf{4.253 \cdot e{-}10}\,(\pm\mathbf{2.833 \cdot e{-}10})$	$1.200 \cdot e{-}8\,(\pm 4.459 \cdot e{-}9)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ ↑	**72**%	44%	**90**%	60%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ ↑	**96**%	**96**%	**90**%	56%
$R=4$	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$\mathbf{4.154 \cdot e{-}7}\,(\pm\mathbf{1.494 \cdot e{-}7})$	$9.290 \cdot e{-}7\,(\pm 2.851 \cdot e{-}7)$	$\mathbf{8.893 \cdot e{-}10}\,(\pm\mathbf{4.922 \cdot e{-}10})$	$4.365 \cdot e{-}8\,(\pm 1.030 \cdot e{-}8)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$\mathbf{2.646 \cdot e{-}7}\,(\pm\mathbf{9.219 \cdot e{-}8})$	$5.512 \cdot e{-}7\,(\pm 1.796 \cdot e{-}7)$	$\mathbf{5.248 \cdot e{-}10}\,(\pm\mathbf{2.771 \cdot e{-}10})$	$2.689 \cdot e{-}8\,(\pm 5.989 \cdot e{-}9)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ ↑	**72**%	20%	**100**%	**100**%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ ↑	**98**%	**100**%	**100**%	**100**%
CSC		***Small-scale dataset***		***Large-scale dataset***	
CP-rank	**Metrics**	**FCSC-ShM** [Bristow et al., 2013]	**ConvFISTA** [Chalasani et al., 2013]	**FCSC-ShM** [Bristow et al., 2013]	**ConvFISTA** [Chalasani et al., 2013]
–	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$3.072 \cdot e{-}5\,(\pm 7.682 \cdot e{-}6)$	$3.211 \cdot e{-}5\,(\pm 5.364 \cdot e{-}6)$	$2.840 \cdot e{-}5\,(\pm 3.403 \cdot e{-}6)$	$1.630 \cdot e{-}5\,(\pm 1.000 \cdot e{-}6)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$2.031 \cdot e{-}5\,(\pm 4.601 \cdot e{-}6)$	$8.746 \cdot e{-}5\,(\pm 5.234 \cdot e{-}6)$	$1.873 \cdot e{-}5\,(\pm 2.128 \cdot e{-}6)$	$1.435 \cdot e{-}5\,(\pm 1.376 \cdot e{-}6)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}6\}$ ↑	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}6\}$ ↑	0%	0%	0%	0%

Table 3.4: Results return on the CSC task on dataset without noise. For T-ConvADMM and T-ConvFISTA, $R = 1, 2, 3$, or 4. Mean and standard deviation are reported. For the RMSE the lowest the better. For the other ones, the higher the better.

deviation $\sim 0.0023, 0.015$, and 0.03). Recall that the definition of the SNR between a signal $\boldsymbol{y}_{ref}$ and a comparison one $\boldsymbol{y}_{noisy} = \boldsymbol{y}_{ref} + \boldsymbol{\varepsilon}$ is

$$\text{SNR}(\boldsymbol{y}_{ref}, \boldsymbol{y}_{noisy}) = 10 \log_{10} \left(\frac{\text{Var}(\boldsymbol{y}_{ref})}{\text{MSE}(\boldsymbol{y}_{ref}, \boldsymbol{y}_{noisy})} \right) , \tag{3.35}$$

where MSE denotes the Mean Squared Error. SNR is an asymmetric decibel measurement (dB) used to compare the level of a signal to the level of background noise.

Metrics. We use four metrics to evaluate our methods:

- The Root Mean Square Error (RMSE) between the true input signal (resp. the true activation maps) and the reconstruction. The lower the better. We denote them RMSE($\boldsymbol{\mathcal{Y}}$) and RMSE($\boldsymbol{\mathcal{Z}}$).

- The number of times a method reaches a "correct" minimizers among all the initializations e.g. RMSE under $\varepsilon = 1.e{-}6$. This metric reflects the sensitivity of an algorithm to its initializations. The higher the better. We denoted them $\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < \varepsilon\}$ and $\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < \varepsilon\}$.

6.1.1 Evaluation of the K-CSC (known dictionary)

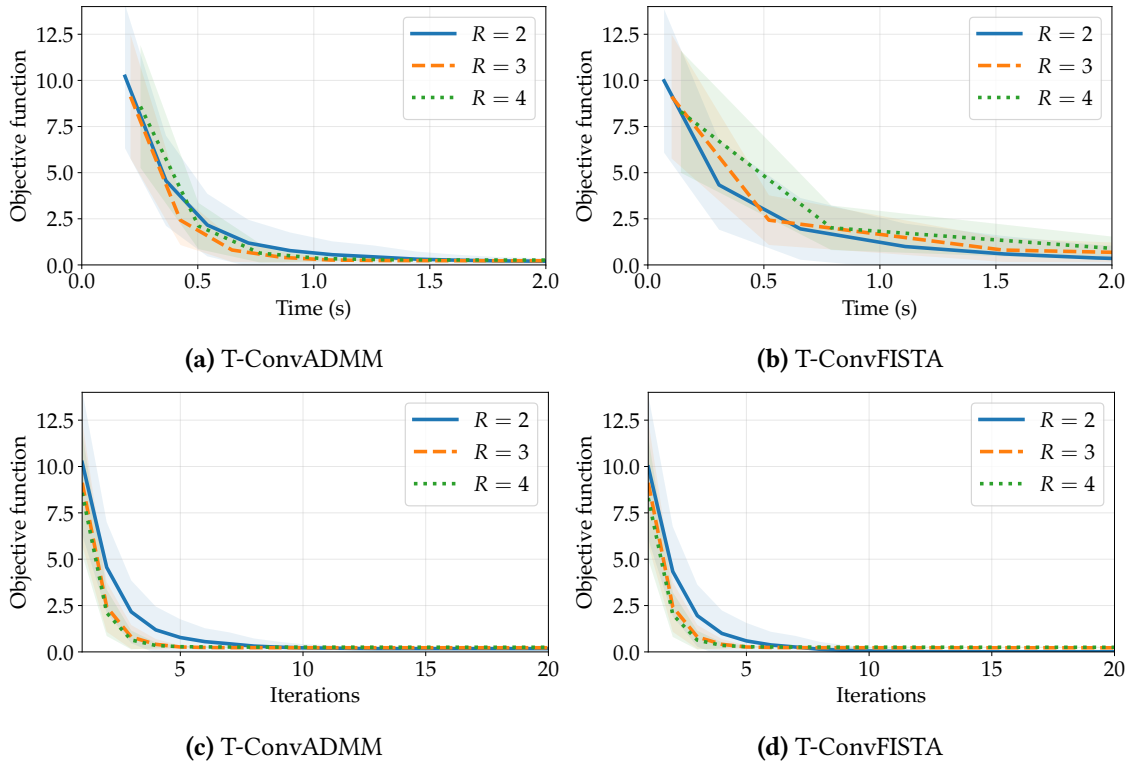
In this experiment, we only compare the performances of the methods on the CSC as this is where stands our major contribution. The true dictionary is therefore given at the begining. The $\{\boldsymbol{Z}_{k,q}\}$ are initialized with random Uniform matrices.

CSC $\mathbf{R=2}$		***Noisy small-scale dataset***		***Noisy large-scale dataset***	
SNR	**Metrics**	**T-ConvADMM**	**T-ConvFISTA**	**T-ConvADMM**	**T-ConvFISTA**
25.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$\mathbf{3.988 \cdot e{-}4\,(\pm 4.121 \cdot e{-}5)}$	$3.999 \cdot e{-}4\,(\pm 4.427 \cdot e{-}5)$	$\mathbf{6.523 \cdot e{-}5\,(\pm 5.970 \cdot e{-}7)}$	$7.606 \cdot e{-}5\,(\pm 1.253 \cdot e{-}6)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$\mathbf{2.397 \cdot e{-}4\,(\pm 2.331 \cdot e{-}5)}$	$2.403 \cdot e{-}4\,(\pm 2.528 \cdot e{-}5)$	$\mathbf{3.820 \cdot e{-}5\,(\pm 4.431 \cdot e{-}7)}$	$4.469 \cdot e{-}5\,(\pm 1.1666 \cdot e{-}6)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}3\}$ ↑	**98**%	94%	86%	**90**%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}3\}$ ↑	**98**%	96%	86%	**90**%
9.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$2.513 \cdot e{-}3\,(\pm 1.046 \cdot e{-}4)$	$\mathbf{2.492 \cdot e{-}3\,(\pm 9.024 \cdot e{-}5)}$	$\mathbf{4.254 \cdot e{-}4\,(\pm 9.016 \cdot e{-}6)}$	$4.958 \cdot e{-}4\,(\pm 7.733 \cdot e{-}6)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$1.509 \cdot e{-}3\,(\pm 6.113 \cdot e{-}5)$	$\mathbf{1.495 \cdot e{-}3\,(\pm 5.066 \cdot e{-}5)}$	$\mathbf{2.504 \cdot e{-}4\,(\pm 8.241 \cdot e{-}6)}$	$2.913 \cdot e{-}4\,(\pm 7.194 \cdot e{-}6)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 2.5e{-}3\}$ ↑	**84**%	**84**%	84%	**88**%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 2.5e{-}3\}$ ↑	**96**%	98%	84%	**90**%
3.0dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$5.224 \cdot e{-}3\,(\pm 3.302 \cdot e{-}4)$	$\mathbf{4.847 \cdot e{-}3\,(\pm 3.166 \cdot e{-}4)}$	$\mathbf{8.835 \cdot e{-}4\,(\pm 2.140 \cdot e{-}5)}$	$9.918 \cdot e{-}4\,(\pm 1.529 \cdot e{-}5)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$3.147 \cdot e{-}3\,(\pm 2.039 \cdot e{-}4)$	$\mathbf{2.894 \cdot e{-}3\,(\pm 1.805 \cdot e{-}4)}$	$\mathbf{5.187 \cdot e{-}4\,(\pm 1.708 \cdot e{-}5)}$	$5.828 \cdot e{-}4\,(\pm 1.434 \cdot e{-}5)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 5.e{-}3\}$ ↑	41%	**52**%	84%	**88**%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 4.e{-}3\}$ ↑	84%	**86**%	84%	**88**%
CSC		***Noisy small-scale dataset***		***Noisy large-scale dataset***	
SNR	**Metrics**	**FCSC-ShM** [Bristow et al., 2013]	**ConvFISTA** [Chalasani et al., 2013]	**FCSC-ShM** [Bristow et al., 2013]	**ConvFISTA** [Chalasani et al., 2013]
25.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$2.292 \cdot e{-}3\,(\pm 1.220 \cdot e{-}5)$	$2.109 \cdot e{-}3\,(\pm 2.547 \cdot e{-}4)$	$1.732 \cdot e{-}3\,(\pm 8.707 \cdot e{-}6)$	$1.732 \cdot e{-}3\,(\pm 8.703 \cdot e{-}6)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$1.454 \cdot e{-}3\,(\pm 7.326 \cdot e{-}5)$	$1.311 \cdot e{-}3\,(\pm 2.027 \cdot e{-}4)$	$1.050 \cdot e{-}3\,(\pm 2.794 \cdot e{-}6)$	$1.050 \cdot e{-}3\,(\pm 2.791 \cdot e{-}5)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 1.e{-}3\}$ ↑	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 1.e{-}3\}$ ↑	0%	0%	10%	10%
9.5dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$6.734 \cdot e{-}3\,(\pm 7.117 \cdot e{-}4)$	$6.673 \cdot e{-}3\,(\pm 6.847 \cdot e{-}4)$	$6.689 \cdot e{-}3\,(\pm 3.344 \cdot e{-}5)$	$6.689 \cdot e{-}3\,(\pm 3.344 \cdot e{-}4)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$4.393 \cdot e{-}3\,(\pm 6.060 \cdot e{-}4)$	$4.367 \cdot e{-}3\,(\pm 5.919 \cdot e{-}4)$	$4.405 \cdot e{-}3\,(\pm 3.011 \cdot e{-}3)$	$4.406 \cdot e{-}3\,(\pm 3.010 \cdot e{-}3)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 2.5e{-}3\}$ ↑	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 2.5e{-}3\}$ ↑	0%	0%	0%	0%
3.0dB	RMSE($\boldsymbol{\mathcal{Y}}$) ↓	$1.215 \cdot e{-}2\,(\pm 1.252 \cdot e{-}3)$	$1.209 \cdot e{-}2\,(\pm 1.202 \cdot e{-}3)$	$1.211 \cdot e{-}2\,(\pm 6.243 \cdot e{-}4)$	$1.211 \cdot e{-}2\,(\pm 6.242 \cdot e{-}4)$
	RMSE($\boldsymbol{\mathcal{Z}}$) ↓	$7.800 \cdot e{-}3\,(\pm 1.033 \cdot e{-}3)$	$7.774 \cdot e{-}3\,(\pm 1.009 \cdot e{-}3)$	$7.812 \cdot e{-}3\,(\pm 5.252 \cdot e{-}4)$	$7.813 \cdot e{-}3\,(\pm 5.251 \cdot e{-}4)$
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Y}}) < 5.e{-}3\}$ ↑	0%	0%	0%	0%
	$\#\{\text{RMSE}(\boldsymbol{\mathcal{Z}}) < 4.e{-}3\}$ ↑	0%	0%	0%	0%

Table 3.5: Results return on the CSC task on dataset with noise. For T-ConvADMM and T-ConvFISTA, R is set to the true value, $R^* = 2$. Mean and standard deviation are reported. With SNR$= -17$dB, the best result with T-ConvADMM and T-ConvFISTA was obtained by the tensor full of zero (no activations) for the small-scale dataset. For the two standard methods, the best result was obtained by the tensor full of zero regardless the size of the data.

Noiseless scenario. We start with the noiseless case. For each one of the 20 input signals, we run our methods with $R = 1, 2, 3, 4$ and for five different initializations. This makes a total of 400 runs. The metric $\#\{\text{RMSE}(\cdot) < \varepsilon\}$ is therefore calculated on 50 initializations. Each time, the reconstruction giving the lowest RMSE($\boldsymbol{\mathcal{Y}}$) among the five tries is kept.

Quantitative results are collected in Table 3.4. Both T-ConvADMM and T-ConvFISTA give competitive results with RMSE under $1.e-7$ as soon as $R \geq 2$. Furthermore, as expected, the best results are obtained when the estimated rank R is equal to the true one, i.e. when $R = R^* = 2$. Notice that, although surprising, an overestimation of the rank does not penalize the performance and still leads to very low RMSE – under $1.e-7$. We also collected results of the standard methods in Table 3.4 (bottom). With RMSE only around $1.e{-}5$, we clearly outperformed FCSC-ShM and ConvFISTA. This was expected as they do not take into account the underlying rank structure, i.e. the separability of the activations. In addition, for the two datasets we display on Figures 3.7 and 3.8 the values of the objective function (average on all the runs) in term of times or iterations. Curves with respect to times slightly advantage T-ConvADMM against T-ConvFISTA. Furthermore, they show the advantage of correctly estimating R^* as, with a high R, methods are more expensive (at least at the beginning). This is in line with the complexity section. Nevertheless, note that our implementation takes into account the sparsity of the matrices involved. Hence, thanks to a proper tuning of the hyperparameters, even if R is too large our methods quickly converge – the unnecessary column of the activation maps being set to 0. In comparison, results obtained in Table 3.4 take ~ 500 seconds for both FCSC-ShM and ConvFISTA against ~ 200 seconds for our methods. This gives a difference of more than a factor 2. Another



(a) T-ConvADMM **(b)** T-ConvFISTA

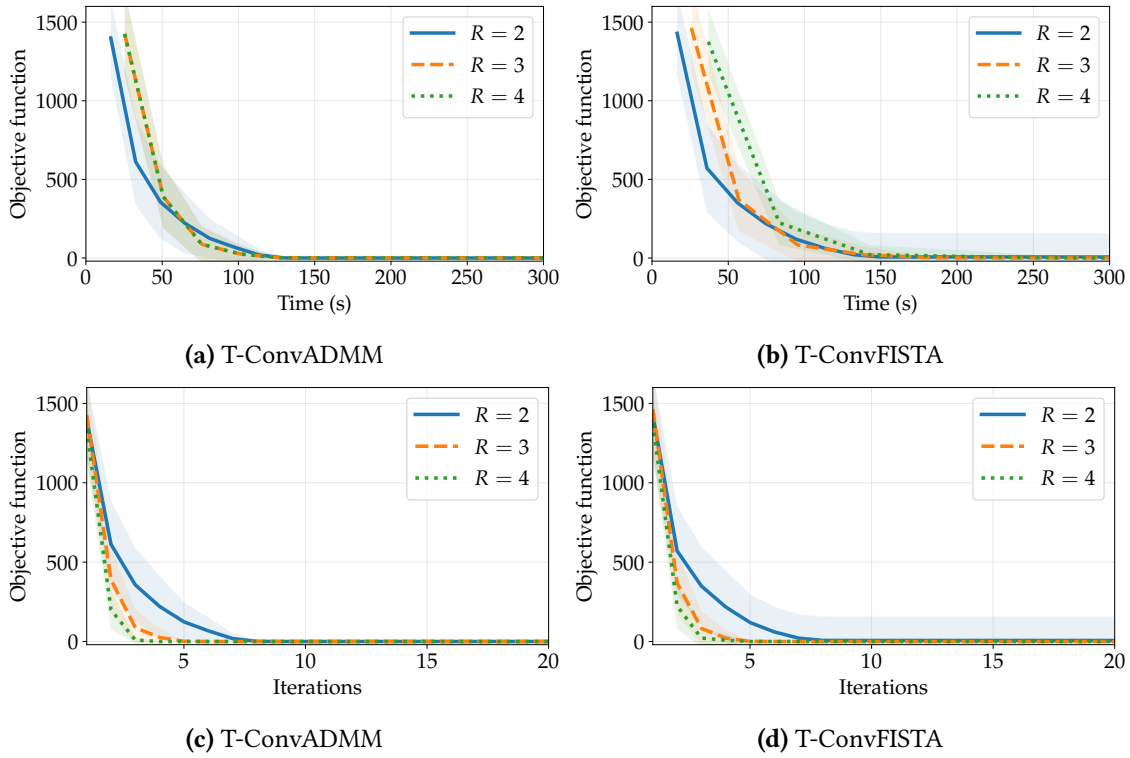
(c) T-ConvADMM **(d)** T-ConvFISTA

Figure 3.7: Average curves with standard deviation of the convergence of our two methods on the *small-scale dataset* with respect to (a, b) the times and (c, d) the number of iterations and for $R = 2, 3, 4$.

very interesting result is the convergence in term of iterations. Curves with respect to iterations do not present significant difference between our two methods. More importantly, they also do not present significant difference between the two datasets and converge in approximatively 10 iterations.

Results with noise. We now study the noisy case. This is an important experiment as while the CSC model has been successfully used for image processing problems, it still falls behind traditional patch-based methods on simple tasks such as denoising [Simon and Elad, 2019]. For each input signal, we run our methods with five different initializations. The metric $\#\{\text{RMSE}(\cdot) < \varepsilon\}$ is therefore calculated on 50 initializations. Each time, the reconstruction giving the lowest $\text{RMSE}(\boldsymbol{\mathcal{Y}})$ among the five tries is kept. We set $R = R^* = 2$ during all the experiment.

Quantitative results are collected in Table 3.5. The most remarkable result is that, even under strong noise, T-ConvADMM and T-ConvFISTA yield very good reconstructions. Figure 3.9 provides a visual example of this important property. We see that T-ConvADMM reconstructs the input signal with high accuracy when SNR $\sim$ 3.0dB while FCSC is completely defective and mostly overfits the noise. This was expected because the noise does not share the low-rank structure of the signal. The K-CSC model, which includes a low-rank constraint, succeeds to not capture it and thus recovers the true signal with accuracy. In other words, taking into account the low-rank structure of the signal eliminates the noise and allows a better recovery of the activations. Furthermore, notice that since for both datasets $R^* = 2$, the larger the signal, the more "restrictive" the rank constraint is. This leads to lower RMSEs on the large-scale dataset



(a) T-ConvADMM **(b)** T-ConvFISTA

(c) T-ConvADMM **(d)** T-ConvFISTA

Figure 3.8: Average curves with standard deviation of the convergence of our two methods on the *large-scale dataset* with respect to (a, b) the times and (c, d) the number of iterations and for $R = 2, 3, 4$.

than on the small-scale dataset.

6.1.2 Evaluation of the K-CDL (unknown dictionary)

We now quickly evaluate our algorithms on the full K-CDL. We use the datasets of the previous section, set $R = 2$, and use T-ConvFISTA combined with the FCSC solver with Sherman–Morrison iterates for the $\boldsymbol{\mathcal{D}}$-step [Wohlberg, 2015]. This solver is preferred to T-ConvADMM as it provides similar results on the K-CSC without the necessity of tuning the ρ parameter (we calculate the Lipschitz constant instead). The activations $\{\boldsymbol{\mathcal{Z}}_{k,q}\}$ and the atoms $\{\boldsymbol{\mathcal{D}}_k\}$ are initialized with random Uniform matrices or tensors. Then, we normalize the atoms to satisfy the ℓ_2 constraint.

Results. On noiseless signals, we obtain a range of RMSEs comparable to those obtained with standard methods when $R \geq R^*$. However, on noisy signals, we observe that T-ConvFISTA returns better results than FCSC-ShM and ConvFista even if the number of active coefficients is lower (see Figure 3.10, for an example on the same signal of Figure 3.9). We now compare the time performance of T-ConvFISTA with the other solvers. To be as fair as possible, we employ the strategy proposed in [Mairal et al., 2010] and re-implemented the other two methods in pure Python. Their code now share an important part with our algorithm, and we can draw meaningful comparisons, which would have been difficult otherwise. Figure 3.11 shows the average time until convergence (i.e. until the relative convergence tolerance becomes lower than $1e-4$ [Boyd et al., 2011]). While it is important to remark that the relative speeds of each methods are dependent of their choice of hyperparameters as well as on the sparsity of the signals, we observe that (i) T-ConvFISTA with the optimizations discussed in Section 4.2 is significantly faster

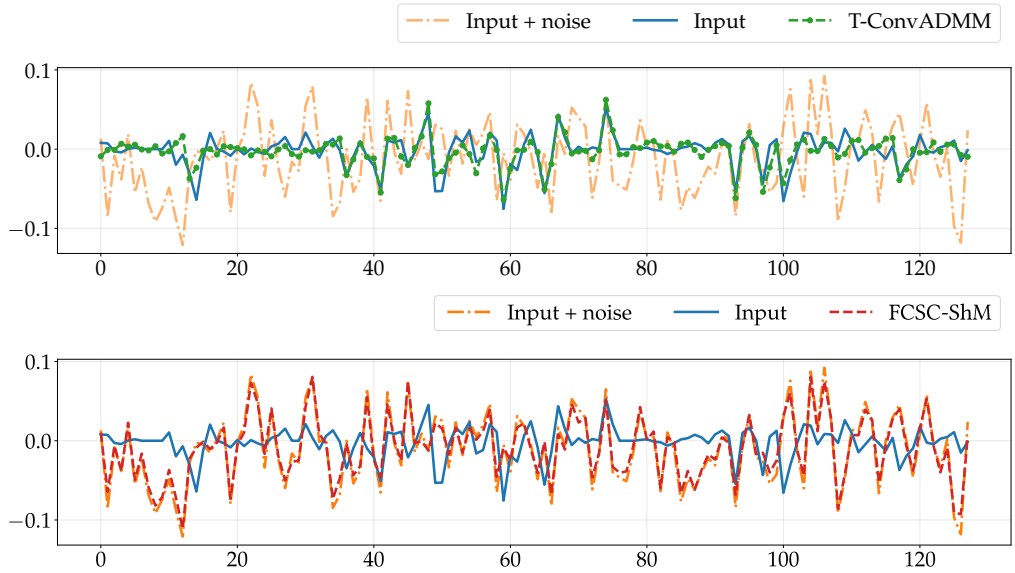


Figure 3.9: One tube of 3-rd order of: (Top) input + noise (SNR of 3.0dB), input, and reconstruction with T-ConvADMM. (Bottom) input + noise (SNR of 3.0dB), input, and reconstruction with FCSC-ShM.

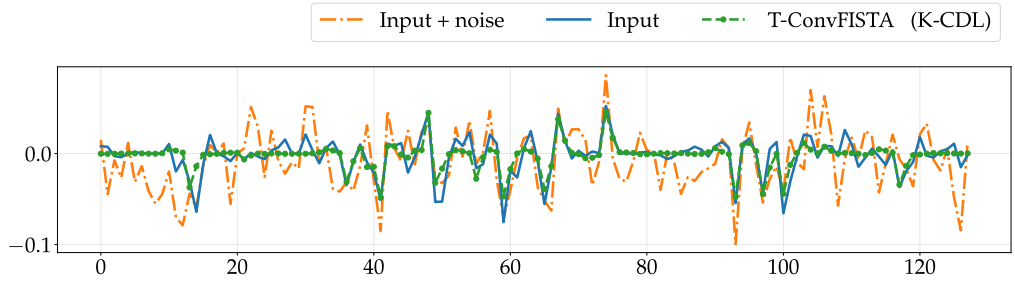


Figure 3.10: One tube of 3-rd order of: input + noise (SNR of 3.0dB), input, and reconstruction with T-ConvFISTA.

than its regular counterpart and (ii) T-ConvFISTA is faster than FCSC-ShM and ConvFista, even if the advantage decreases as R increases. This is in line with the time complexity of each algorithm (see Table 3.2).

6.2 Examples on real data

In this section, we use T-ConvFISTA on a wide range of real data. We start with images and show that it is possible to accurately reconstruct them even with CP low-rank activations. Then, we extract time-frequency patterns related to musical instruments in audio signals. Finally, we consider multichannel ElectroEncephaloGram (EEG) and ElectroCardioGram (ECG) signals. We show that the separability of the activations is an important property allowing to segment the signal or to easily understand its underlying structure. Hyperparameters are set in order to bring enough sparsity while not deteriorating the reconstructions.

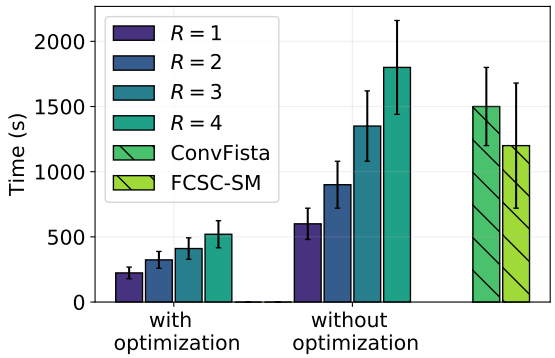
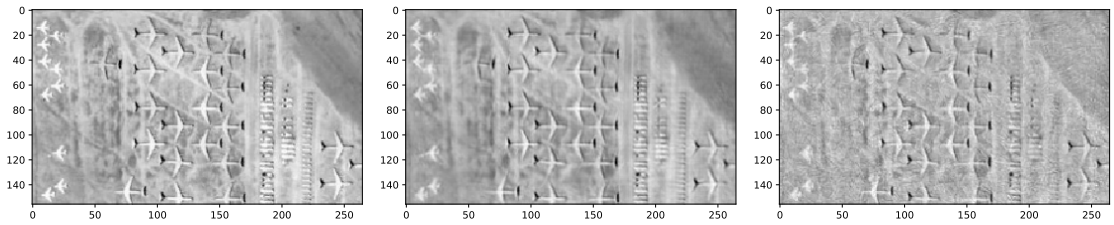


Figure 3.11: Time until convergence of T-ConvFISTA on the dictionary learning process ($\boldsymbol{\mathcal{Z}}$ + $\boldsymbol{\mathcal{D}}$ steps), with and without the optimizations discussed in Section 4.2. The standard deviation are indicated using black lines.



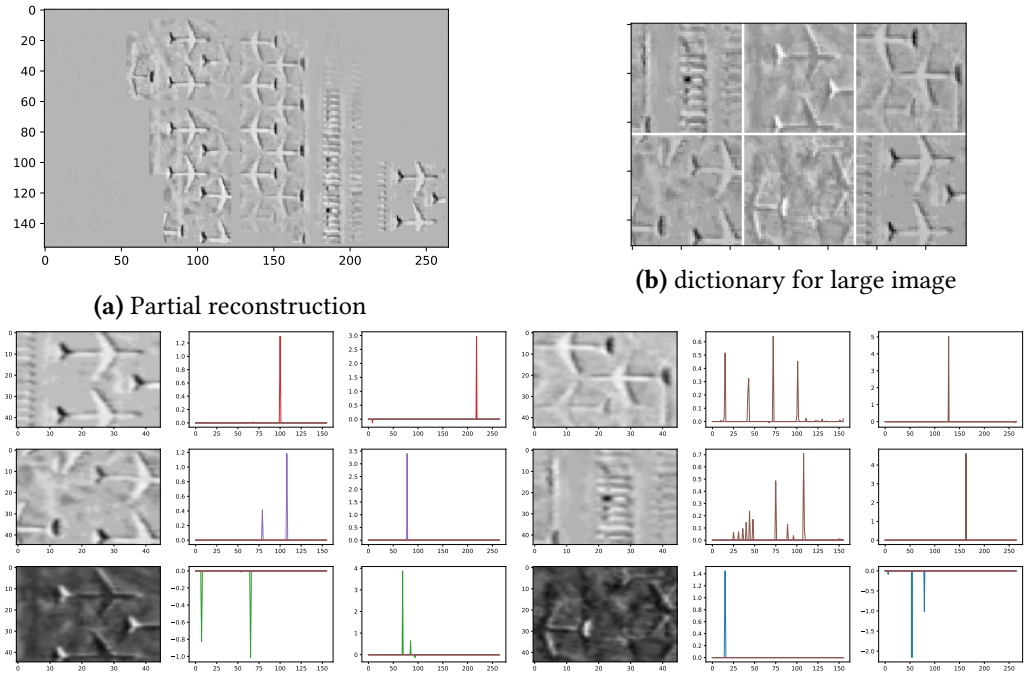
(a) True **(b)** T-ConvFISTA **(c)** FCSC

Figure 3.12: Black and white satellite view of an airport. In the middle, the reconstruction of the initial image with our method. On the right with classical method.

6.2.1 Gray images – 2rd order tensor (matrix).

We first consider the matrix case with a black and white satellite view of an airport of size (150×250) from Hearn and Reichel [2014] (see Figure 3.12 (a)). As this image admits obvious low-rank activation maps due to its redundancy and to its patterns alignment (e.g. planes or cars), we set $R = 3$ and learn 6 atoms.

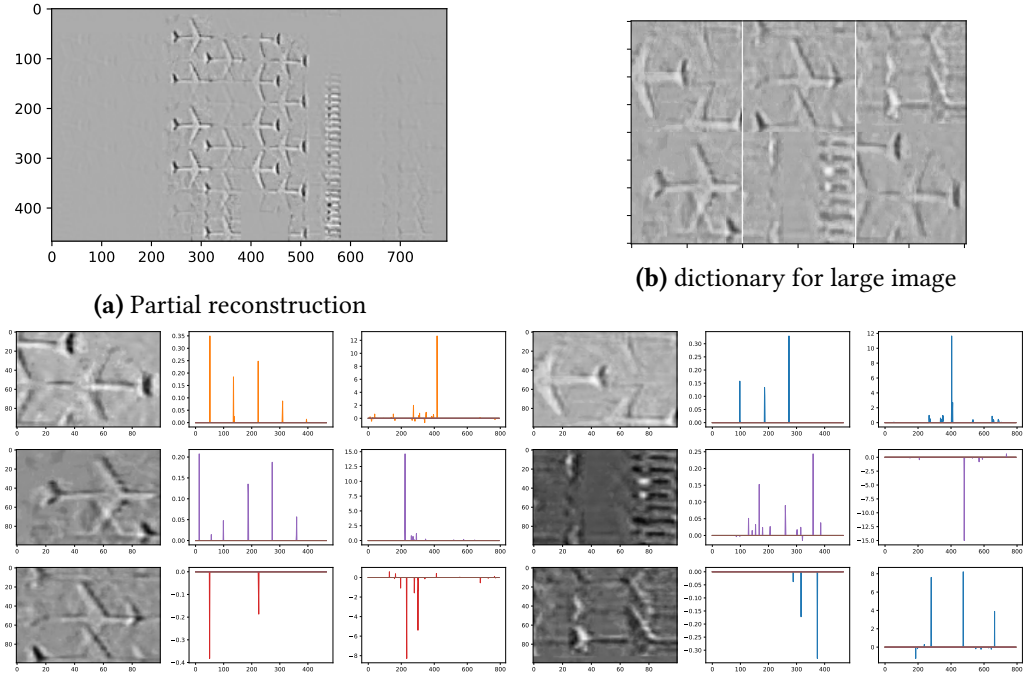
Interestingly, even with this very low-rank constraint we are able to efficiently reconstruct the initial image (Figure 3.12 (b)) and find relevant atoms (Figure 3.13). This is an important behavior since this means that even if the image does not present a global low-rank structure (i.e. the matrix representing the image is not low-rank), it exists patterns with low-rank activations. We display the full results in Figure 3.14. Note that, activations are rank-1 and not 3 as find by T-ConvFISTA. In addition, we also display results for the same image but of size (500×800) in order to highlight the capacity of the algorithm to treat large data (Figure 3.14).



(a) Partial reconstruction

(b) dictionary for large image

Figure 3.13: Illustration of the reconstruction with T-ConvFISTA on the medium scale image. On top, a partial reconstruction and the learn dictionary. Then, four atoms with their activation maps.



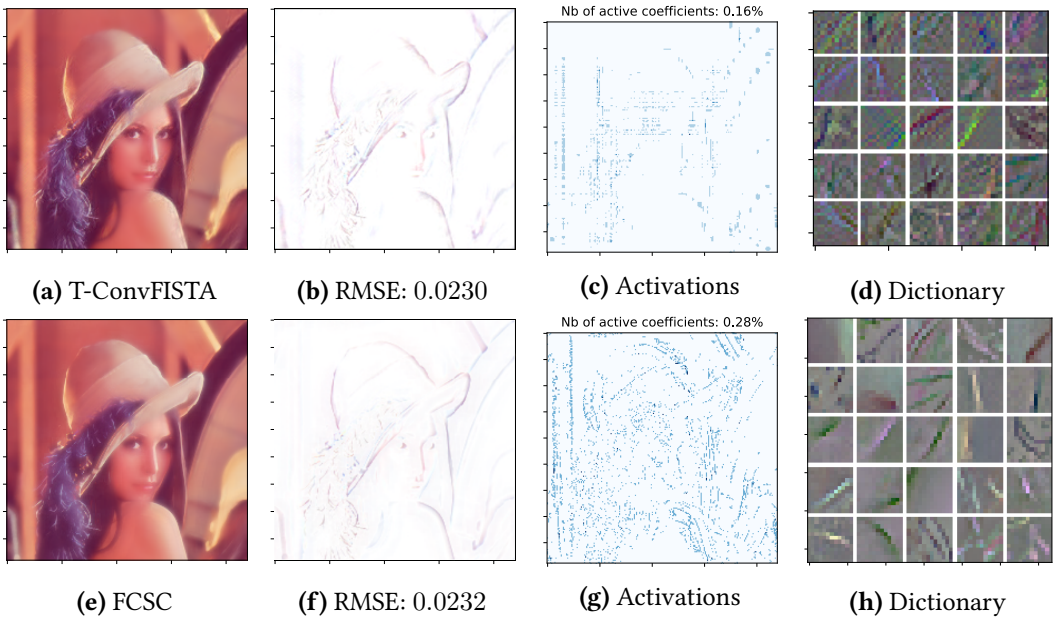
(a) Partial reconstruction

(b) dictionary for large image

Figure 3.14: Illustration of the reconstruction with T-ConvFISTA on the large scale image. On top, a partial reconstruction and the learn dictionary. Then, four atoms with their activation maps.

(a) True **(b)** T-ConvFISTA **(c)** FCSC

Figure 3.15: On the left, the Lena image. On the middle, the reconstruction obtained with T-ConvFISTA. On the middle, the reconstruction obtained with FCSC.



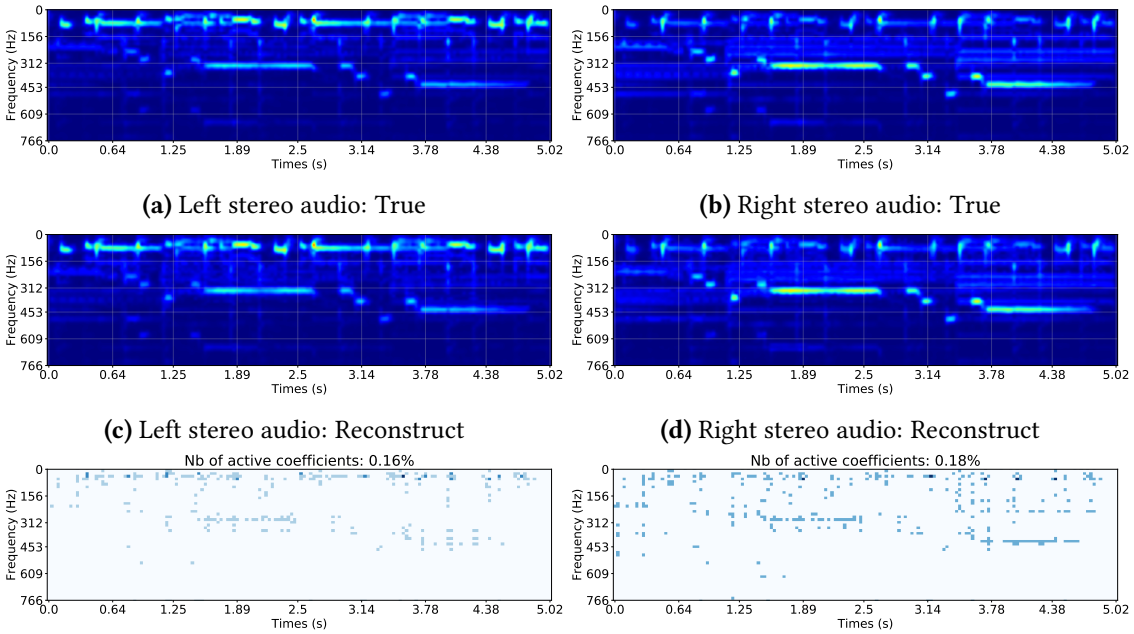
(a) T-ConvFISTA **(b)** RMSE: 0.0230 **(c)** Activations **(d)** Dictionary

(e) FCSC **(f)** RMSE: 0.0232 **(g)** Activations **(h)** Dictionary

Figure 3.16: From left to right. The full reconstruction, the reconstruction on the filtered image, the activations, and the learned dictionary.

Color images – 3rd order tensor. We now consider the famous Lena image encoded in the RGB space (Figure 3.15 (a)). We set $R = 10$ and learn 25 color atoms of size $(12 \times 12 \times 3)$.

Results for T-ConvFISTA and FCSC are displayed on Figures 3.15 (b, c) and 3.16. While the image seems less structured than the previous one, we see that our method stills efficiently reconstruct it. To compare the sparsity, we force the two methods to return equivalent RMSEs. From 3.16 (c) and (g), we see that to reconstruct the image with an RMSE of ~ 0.023, T-ConvFISTA need much less activations than FCSC: 0.16% against 0.28%. We therefore clearly see that, even if this image does not have an obvious structure, our algorithm is able to find it and to learn it. Interestingly, although we set $R = 10$, it always returns activations with CP-rank smaller than 6.



(a) Left stereo audio: True

(b) Right stereo audio: True

(c) Left stereo audio: Reconstruct

(d) Right stereo audio: Reconstruct

(e) Left stereo audio: Activations

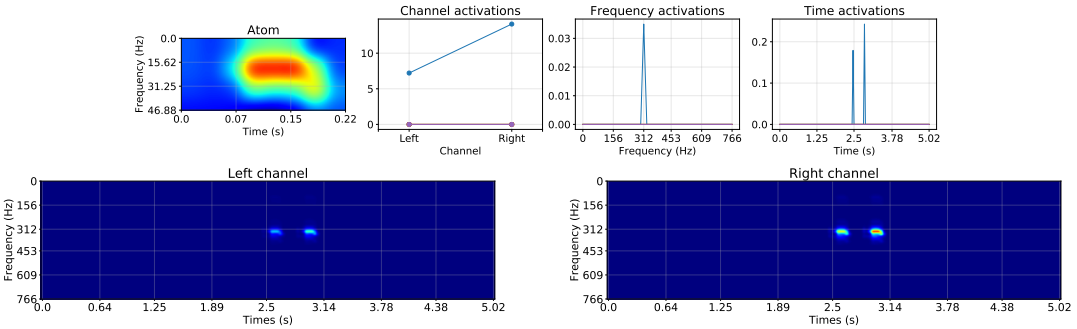
(f) Right stereo audio: Activations

Figure 3.17: Results on the jazz signal. On top the true spectrograms of the left and right channels. On the middle, the reconstructions. On the bottom, the activations obtained by adding up the activations of all atoms.

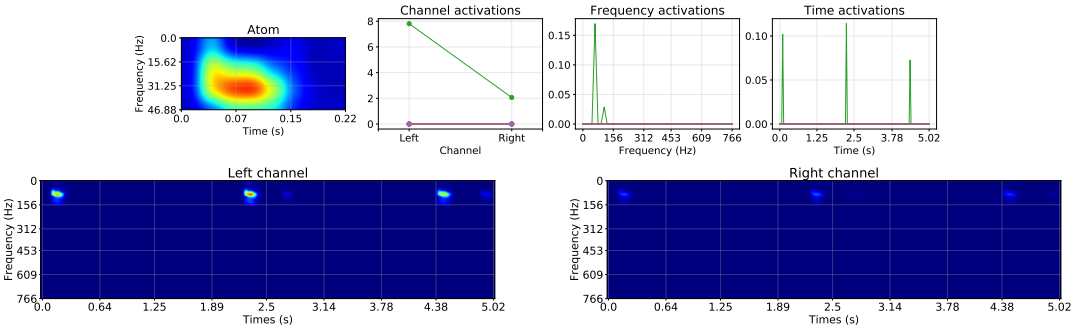
6.2.2 Audio signal – 3rd order tensor with low-rank structure.

Identifying recurring patterns in audio signal is an important problem in many scientific domains. A popular model to achieve this is nonnegative matrix factorization (NMF) [Lee and Seung, 1999]. A more recent model is the convolutive nonnegative matrix factorization (CNMF) [O'grady and Pearlmutter, 2006]. It extends the classic NMF by introducing a convolutional structure into the low-rank model reconstruction and thus, captures short-term temporal dependencies in the data. However, these two methods never deal with stereo or multidimensional signals. In this example, we propose to use T-ConvFISTA to learn a dictionary (i.e. short-lived temporal patterns) on a stereo audio signal. This stereo signal is 5 seconds long and recorded at 8000Hz, for a total of $2 \times 40000 = 80000$ points. For each signal (one per channel), we compute a short-time Fourier transform to obtain its spectrogram. Window size is set to 512 samples with 50% overlap : only the first 50 bins have been conserved ($0 - 781.25$ Hz). The final data consists in a third order tensor of size $(2 \times 50 \times 158)$. We reconstruct the input using $K = 25$ frequency-time atoms of size $(1 \times 4 \times 8)$ (i.e. atoms with 46.875 Hz bandwidth of 0.224 seconds). The maximal CP-rank of each associated activation is set to $R = 5$.

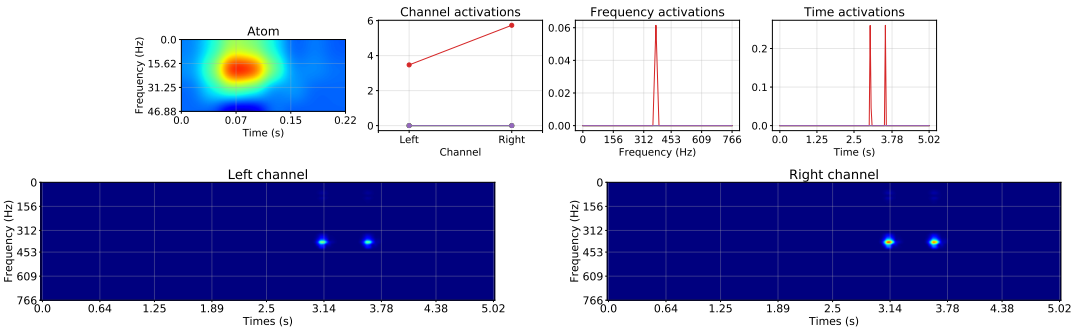
We obtain a RMSE of $3.415e{-}3$ with 0.17% active coefficients while with FCSC we obtain a RMSE of $4.048e{-}3$ with more than 0.34% active coefficients. For reference, the RMSE is equal to $1.060e{-}2$ when the reconstruction is full of 0. The results are displayed on Figure 3.17. Atoms and activations returned by our method are displayed Figure 3.18. Since in this audio signal the different instruments play at different frequency we can isolate them: the first two atoms of Figure 3.18(e) correspond to the drums and the two last ones (Figure 3.18 (f)) to the guitar.



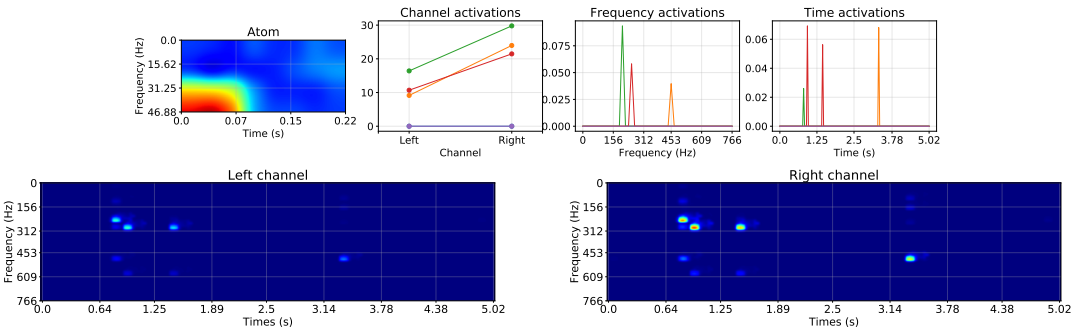
(a) Left stereo audio: atom of drums **(b)** Right stereo audio: atom of drums



(c) Left stereo audio: atom of drums **(d)** Right stereo audio: atom of drums



(e) Left stereo audio: atom of guitar **(f)** Right stereo audio: atom of guitar



(g) Left stereo audio: atom of guitar **(h)** Right stereo audio: atom of guitar

Figure 3.18: On each of the four group of images: From left to right, the learned atom, the activations relative to the first dimension (channel), the activations relative to the second dimension (frequency), and the activations relative to the third dimension (time). Then, the two spectrograms corresponding to the reconstruction.

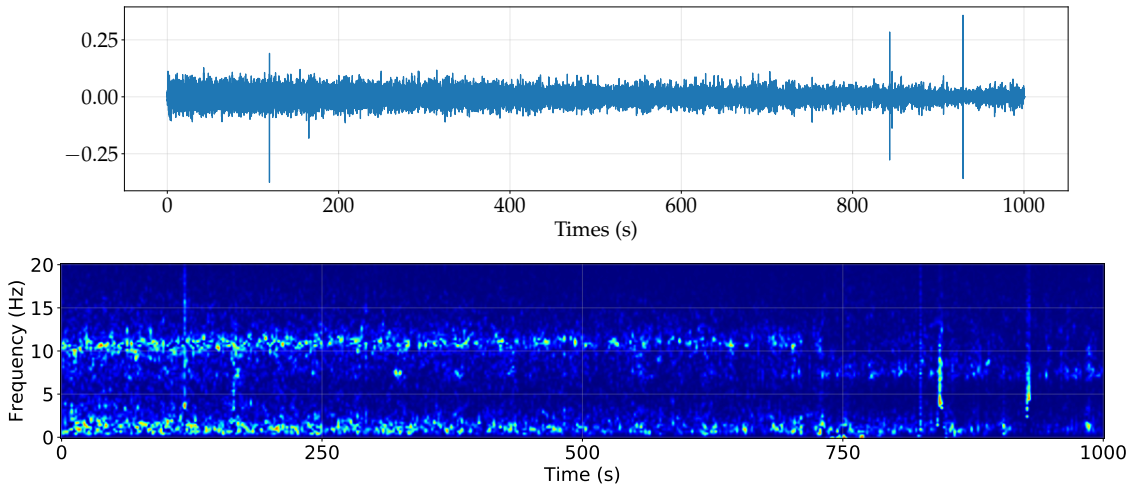


Figure 3.19: EEG recording (in μV) of a patient during GA (sampling frequency: 250Hz). On top the raw signal of one channel and on bottom its spectrogram.

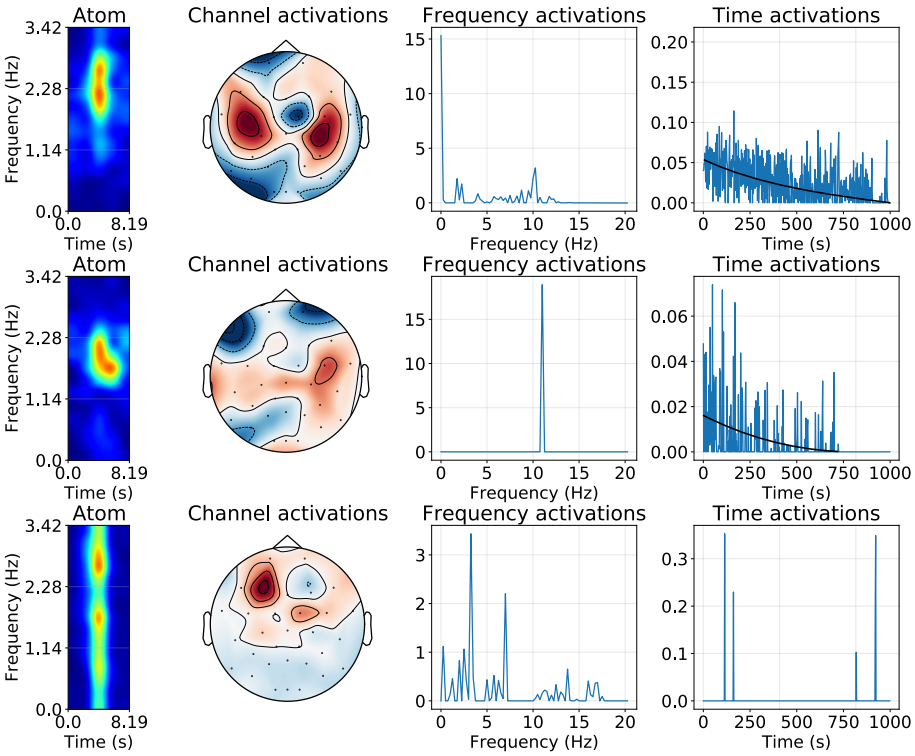
6.3 Signals recorded during a general anesthesia

6.3.1 A study of multichannel electroencephalography signals

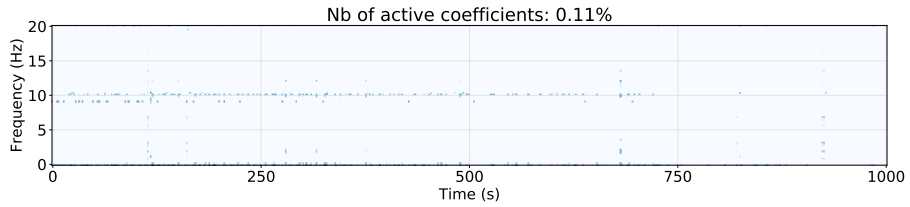
We now consider multichannel EEG signals that record brain activity with sensors covering a large part of the head. Note that, this is a difficulet dataset because there is a lot of noise, impulsion noise, and deficient measures. This two-dimensional measurement is stored in a matrix $\boldsymbol{X}$ in $\mathbb{R}^{N_s \times N_t}$ where N_s is the number of sensors and N_t is the number of samples.

Justification of the model. Assuming a static propagation, the $\boldsymbol{X}$ matrix can be factorized into a lead-field matrix $\boldsymbol{A}$ and a signal matrix of N_r sources, denoted $\boldsymbol{S} \in \mathbb{R}^{N_r \times N_t}$, such that $\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S}$ [Becker et al., 2015]. The goal is now to find a transformation allowing to produce a relevant data tensor from $\boldsymbol{X}$. A frequently used idea is to compute a short-time Fourier transform on each channel to obtain a Space-Time-Frequency (STF) representation $\boldsymbol{\mathcal{Y}}$ [Miwakeichi et al., 2004; Mørup et al., 2006; Becker et al., 2010, 2014; Zhao et al., 2011]. In previous methods, authors assume that the time and frequency variables separate in order to justify a CP decomposition of the tensor. While no theoretical validation that justifies this application has been performed [Becker et al., 2014], all these works show that tensor decomposition inherently exploits the interactions among multiple modes. Here, we adopt a slightly different point of view as we do not assume that the full tensor $\boldsymbol{\mathcal{Y}}$ is tri-linear. Instead, we only assume that it results from the summation of K relevant atoms with associated tri-linear activations. The sparsity of the activations is supported by recent results on neuroscience which postulate that neural activity consists more of transient bursts of isolated events rather than rhythmically sustained oscillations [van Ede et al., 2018]. Such activities could be described not only by their frequency and amplitude but also by their rate, duration, and shape suggesting that multivariate CDL is well-adapted to analyze them.

Data and parameters. The data consists in 32 EEG signals recorded at 250 Hz during a General Anesthesia (GA). We crop the full signal to keep only an important phase of the GA known as the "Recovery of Consciousness" (RoC) [Purdon et al., 2013]. Each signal is then of 1000 seconds (see Figure 3.19). With all channels included, it corresponds to $8,000,000$ points. To construct



(a) Three atoms with their activations



(b) Activations of all the atoms

Figure 3.20: (a) Three atoms of interest with their activations. From left to right: the time-frequency atom, the channel activations (mode 1), the frequency activations (mode 2), and the time activations (mode 3). (b) The activations obtained by adding up the activations of all atoms. Topographies are made with MNE-Python [Gramfort et al., 2013].

the STF representation, the signal is first filtered using a bandpass filter between 1 and 20Hz, to remove the potential drift below 1Hz, and to keep the frequencies below 20Hz that characterize GA [Brown et al., 2010]. Then, on each channel a short time Fourier transform is used with window size equals to 1024 samples and 50% overlap: only the first 82 bins have been conserved $(0 - 20$ Hz). We stack the 32 spectrograms in a final tensor $\boldsymbol{\mathcal{Y}}$ of size $(32 \times 82 \times 490)$. During a GA, patients are static and EEG signals do not present many patterns. As a consequence, we set $R = 2$, and only learn $K = 5$ atoms of size $(1 \times 15 \times 5)$ corresponding to time-frequency atoms covering 8.19 seconds and a band of frequencies of 3.42Hz. To reconstruct the 1-D initial signal from the spectrograms we apply the inverse short time Fourier transform.

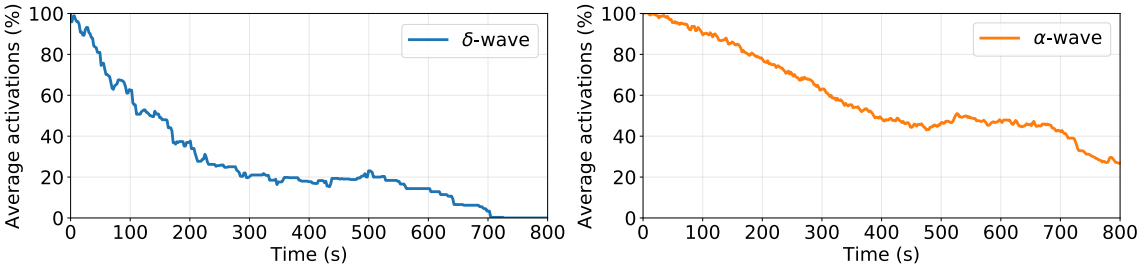


Figure 3.21: Evolution of the time activations for the first and second atoms of Figure 3.20 which are relative to the δ and α waves.

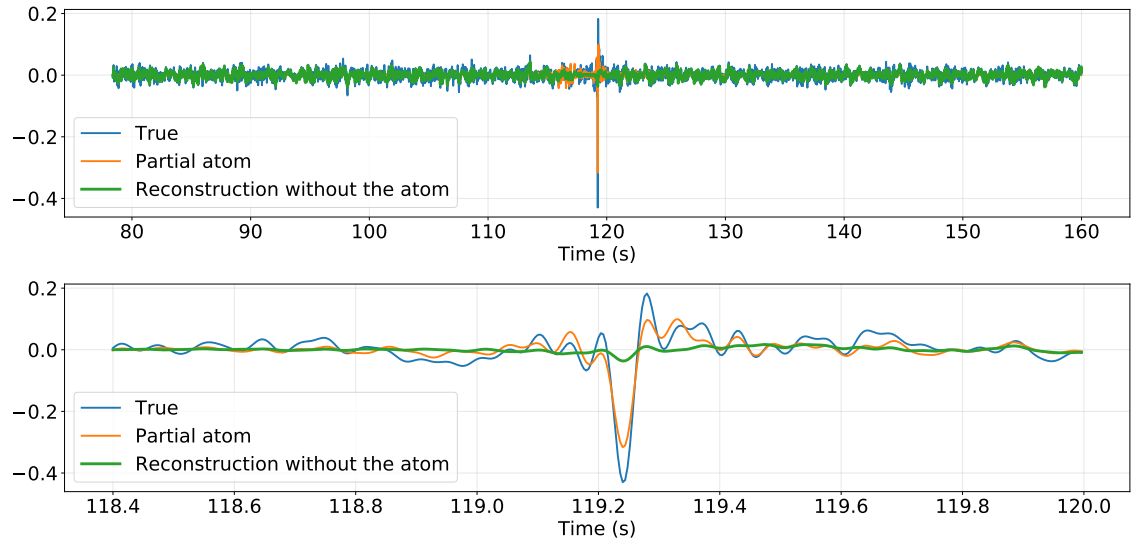
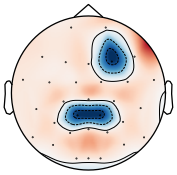


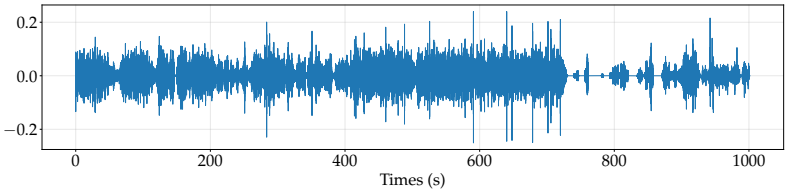
Figure 3.22: The first column shows the atoms learn with T-ConvFISTA on EEG signals. The three other columns show the corresponding activation map for each dimension.

Learned dictionary and activations. Three learned atoms with their activations are displayed in Figure 3.20. One important property is the high interpretability of our results. Indeed, as we decompose the activations into the modes (channels $\times$ frequencies $\times$ times), we can study each one of them independently. For example, from the frequency activations (mode 2), we see that the first two atoms are relative to important frequencies in anesthesia referred as α and θ-waves. Regarding their time activations (mode 3), they decrease with time (see Figure 3.21). This is a common behavior that occurs during a GA induced by propofol [Purdon et al., 2013]. Indeed, it is known that when sedation begins, α and θ-waves appear. Then, during the ROC stage, they gradually disappear and fade away. The third atom corresponds to important spikes which may be explain by impulsional noise. From the channel activations (mode 1), we see that most of its contribution is on one channel. However, due to the propagation of the electricity on all the scalp, the other sensors also record these spikes at the same time. The activation tensor relative to this particular atom is therefore rank-1 (as found by the algorithm). Notice that, thanks to its identification, we can remove its contribution from the final reconstruction in order to not observe the spikes (Figure 3.22).

Robustness to noise and reconstruction. Via the channel activations (mode 1) of one learned atom we identify three deficient channels: 10 (CP1), 21 (CP2), and 28 (F4) (Figure 3.23). In a



(a) Spatial activations



(b) Raw signal of a bad channel

Figure 3.23: (a) Spatial activations before removing the bad channels. (b) Raw signal at one of the bad channel.

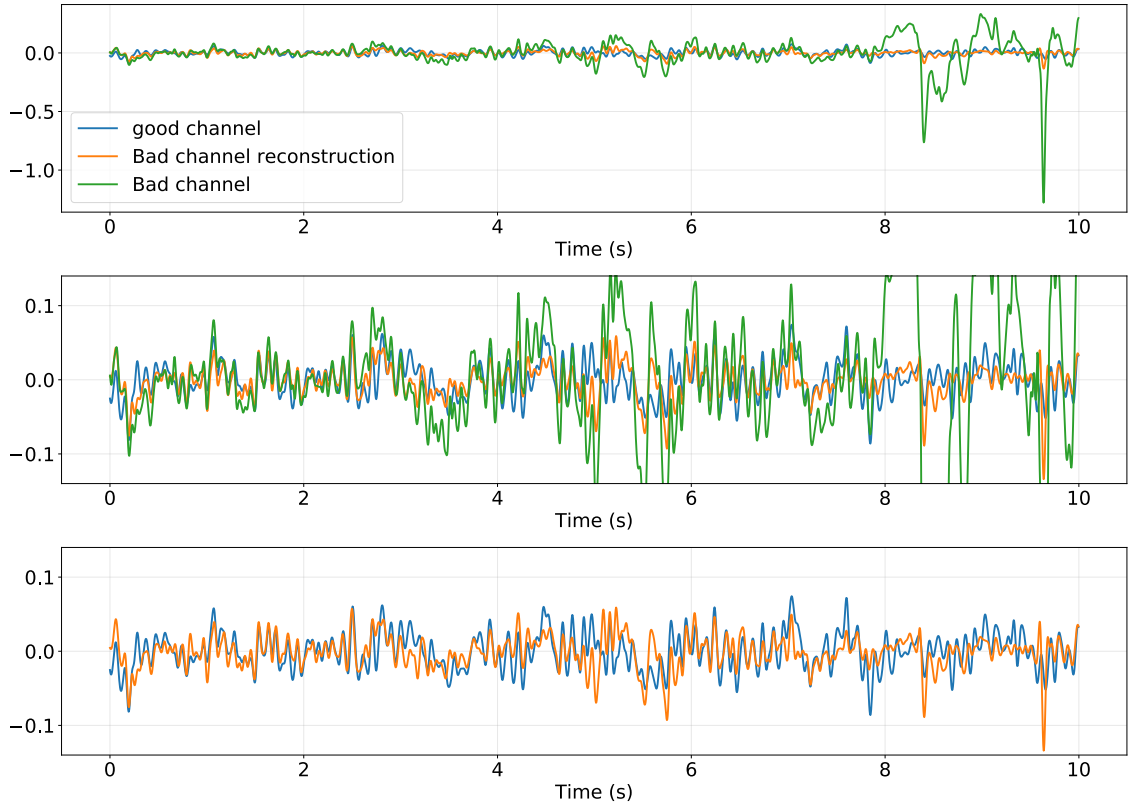


Figure 3.24: On top, raw signal of a good channel (blue), a bad channel (green), and a reconstruction of the bad channel with T-ConvFISTA (orange). The other two figures are more focused on signals.

clinical context, these channels are at spatial positions where the cap can come off. The sensors then only pick up noise at these positions. Fortunately, as show in the synthetic experiments, due to the low-rank constraint, the model assumes links between the channels and is robust to strong noise. In our case, this lead to an automatic reconstruction of the bad channels using the good ones. in Figure 3.24 for instance, we see a bad channel (in green) presenting a lot of noise, especially after 8 seconds. Using the other channels (e.g. the blue one), our algorithm reconstructs the initial signal (in orange).

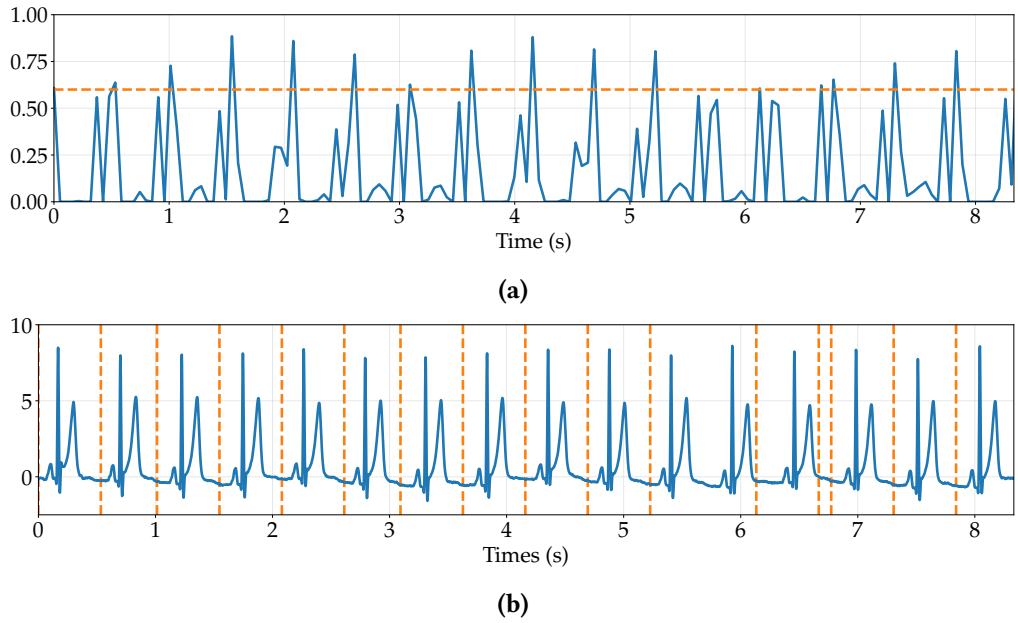


Figure 3.25: On top, one time activation map with threshold (orange dash line). On bottom, result of the EKG signal detection on a small part of it. Each vertical orange dash line is obtained automatically.

6.3.2 Electrocardiogram: automatic detection of the P-QRS-T complex

An ElectroCardioGram signal (EKG) is characterized by five main events referred as P, QRS complex (three events) and T. Each one has a specific role during the cardiac cycle and their abnormalities will lead to different diagnoses [Thakor and Zhu, 1991; Taillefer et al., 1997]. To date, the gold-standard of EKG analysis remains human analysis, except in specific situations such as continuous ST-segment monitoring during anesthesia of high-risk cardiac patients [Landesberg et al., 2002]. The PR interval is known to be linked to the autonomic nervous system [Shouldice et al., 2003]. Drugs used during anesthesia are blocking the autonomous tone, explaining in a large part the side-effects of anesthesia [Gelman and Mushlin, 2004]. An automatic and real time detection of the PR interval appears potentially interesting. In this example, we show how to use T-ConvFISTA to detect P-QRS-T complexes easily.

The EKG signal is recorded at 600Hz during ~ 25 minutes for a total of $1,070,000$ points. Before applying our method, we decompose it with a short-time Fourier transform to obtain a Time-Frequency (TF) representation. Window size is set to 64 samples with 50% overlap so that we keep a high temporal accuracy while drastically reducing the signal in time. Only the first 10 bins have been conserved. The final signal is of size (10×33439). As the spectrogram exhibits a lot of regularity and small variability, we set $R = 2$ and $K = 1$. To reduce the time complexity, we learn the atom on the beginning of the signal. When the atom is learned, we only perform the CSC on the full EKG signal and therefore enjoy an important reduction in complexity. Finally, to detect the complex, we apply a threshold on the time activations map (3.25 (a)). An illustration of the final segmentation is given in Figure 3.25 (b). Note that, as standard methods do not allow to independently control the sparsity in dimension, the activations are spread on the frequency axis. Hence, we obtain a poor quality atom compared to the one return by T-ConvFISTA and the reconstruction becomes noisy (Figure 3.26 (c, d)).

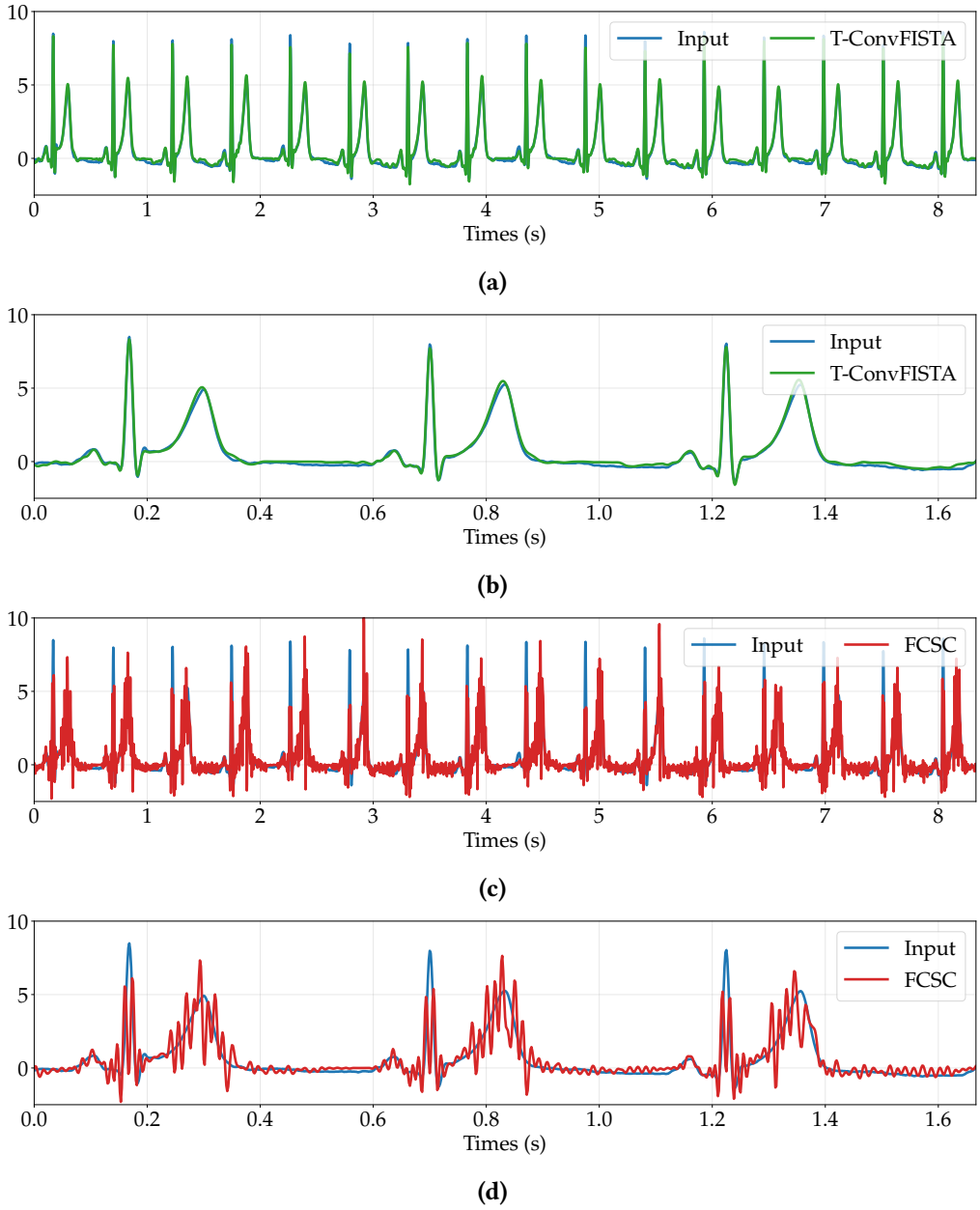


Figure 3.26: Results on EKG signals. (a, b) are the reconstruction with T-ConvFISTA. (c, d) are the reconstruction with FCFC.

7 Conclusion

In this chapter, we generalized the CDL problem to multivariate signals. More particularly, using tensor algebra we supposed that the activation maps are sparse and CP low-rank. We proposed two algorithms based on ADMM and FISTA to efficiently solve the associated minimization problem. The two algorithms are evaluated and compared on both synthetic and real data. We showed that they provide better results than conventional algorithms in term of reconstruction, sparsity, and interpretability. On real data we showed that the ability of our methods to split the activation maps in each mode allows a better comprehension of the input signal.

8 Appendix

8.1 Proofs of the chapter

Lemma 3.2. (Mode-wise DFT) – *Given the CP-decomposition of a tensor* $\boldsymbol{\mathcal{X}} = [\![\boldsymbol{X}_1, \cdots, \boldsymbol{X}_p]\!]$, *the DFT can be performed mode-wise i.e.*

$$\widehat{\boldsymbol{\mathcal{X}}} = \sum_{r=1}^{R} \widehat{\boldsymbol{x}}_r^{(1)} \circ \cdots \circ \widehat{\boldsymbol{x}}_r^{(p)} \ . \tag{3.36}$$

The complexity of the computation of $\widehat{\boldsymbol{\mathcal{X}}}$ *using the FFT goes from* $\boldsymbol{\mathcal{O}}(\prod_{i=1}^{p} n_i \log(\prod_{i=1}^{p} n_i))$ *to* $\boldsymbol{\mathcal{O}}(R \sum_{i=1}^{p} n_i \log(n_i)))$.

Proof. Using the definition of the CP-decompositions, the proof is straightforward. Furthermore, as we only perform 1-D FFT, we obtain the given complexity. □

Theorem 3.2. (Equality in the Fourier domain) – *In the Fourier domain, the fidelity term* $f(\cdot)$ *is equal to*

$$f\left(\{\boldsymbol{Z}_{k,q}\}_{k=1}^{K}\right) = \frac{1}{2\prod_{i=1}^{p} N_i} \left\| \widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{D}}}_k * [\![\widehat{\boldsymbol{Z}}_{k,1}, \cdots, \widehat{\boldsymbol{Z}}_{k,p}]\!] \right\|_F^2 , \tag{3.37}$$

where $\widehat{\cdot}$ *denotes the frequency representation of a signal, and* $*$ *is the component-wise product.*

Proof. The proof rests on several equalities and properties.

$$\begin{aligned}
&\|\boldsymbol{\mathcal{Y}} - \sum_{k=1}^{K} \boldsymbol{\mathcal{D}}_k \circledast \sum_{r=1}^{R} \boldsymbol{z}_{k,r}^{(1)} \circ \cdots \circ \boldsymbol{z}_{k,r}^{(p)}\|_F^2 \\
&= \frac{1}{2\prod_{i=1}^{p} N_i} \|\widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^{K} \mathrm{DFT}(\boldsymbol{\mathcal{D}}_k \circledast \sum_{r=1}^{R} \boldsymbol{z}_{k,r}^{(1)} \circ \cdots \circ \boldsymbol{z}_{k,r}^{(p)})\|_F^2 \quad \text{(Parseval's theorem – Plancherel)} \\
&= \frac{1}{2\prod_{i=1}^{p} N_i} \|\widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{D}}}_k * \sum_{r=1}^{R} \mathrm{DFT}(\boldsymbol{z}_{k,r}^{(1)} \circ \cdots \circ \boldsymbol{z}_{k,r}^{(p)})\|_F^2 \quad \text{(convolution theorem)} \\
&= \frac{1}{2\prod_{i=1}^{p} N_i} \|\widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{D}}}_k * \sum_{r=1}^{R} \widehat{\boldsymbol{z}}_{k,r}^{(1)} \circ \cdots \circ \widehat{\boldsymbol{z}}_{k,r}^{(p)}\|_F^2 \quad \text{(separable)} \\
&= \frac{1}{2\prod_{i=1}^{p} N_i} \|\widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{D}}}_k * [\![\widehat{\boldsymbol{Z}}_{k,1}, \cdots, \widehat{\boldsymbol{Z}}_{k,p}]\!]\|_F^2 \quad \text{(Kruskal operator)} \ .
\end{aligned}$$

□

Corollary 3.3. (A compact vectorized formulation) – *The following equality holds*

$$f\left(\{\boldsymbol{Z}_{k,q}\}_{k=1}^{K}\right) = \frac{1}{2} \left\| \widehat{\boldsymbol{y}}^{(q)} - \widehat{\boldsymbol{\Gamma}} (\widehat{\boldsymbol{A}} \otimes \boldsymbol{I}) \widehat{\boldsymbol{z}}^{(q)} \right\|_F^2 , \tag{3.38}$$

where $\widehat{\boldsymbol{y}}^{(q)}$ *is the vectorization of the folding of* $\widehat{\boldsymbol{\mathcal{Y}}}$ *along the dimension* q, $\widehat{\boldsymbol{z}}^{(q)} = [\widehat{\boldsymbol{z}}_1^{(q)^\intercal}, \ldots, \widehat{\boldsymbol{z}}_K^{(q)^\intercal}]^\intercal$ *where* $\forall k, \widehat{\boldsymbol{z}}_k^{(q)}$ *is the vectorization of the matrix* $\boldsymbol{Z}_{k,q}$, $\widehat{\boldsymbol{\Gamma}} = [\mathit{diag}(\widehat{\boldsymbol{d}_1}^{(n)}), \ldots, \mathit{diag}(\widehat{\boldsymbol{d}_K}^{(n)})]$ *with*

$\boldsymbol{d}_k^{(q)}$ *the vectorization of the folding of* $\widehat{\boldsymbol{\mathcal{D}}}_k$ *along the dimension* q*, and*

$$\widehat{\boldsymbol{A}} = \begin{pmatrix} \widehat{\boldsymbol{B}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{B}}_K \end{pmatrix} \quad \textit{where} \quad \widehat{\boldsymbol{B}}_k = (\overset{\hookleftarrow p}{\odot}_{i=1,i\neq q} \widehat{\boldsymbol{Z}}_{k,i}) \ . \tag{3.39}$$

Here, $\widehat{\boldsymbol{\Gamma}} \in \mathbb{C}^{n_1\cdots n_p \times Kn_1\cdots n_p}$*,* $\widehat{\boldsymbol{A}} \in \mathbb{C}^{K\prod_{1,i\neq q}^{p} n_i \times KR}$*,* $\boldsymbol{I} \in \mathbb{R}^{n_q \times n_q}$*, and* $\widehat{\boldsymbol{z}}^{(q)} \in \mathbb{C}^{KRn_q}$*. Thus, the design matrix* $\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$ *is in* $\mathbb{C}^{n_1\cdots n_p \times KRn_q}$*.*

Proof. The proof mainly rests on the proposition (3.4) and on the formulation of the previous theorem.

$$\begin{aligned}
&\|\widehat{\boldsymbol{\mathcal{Y}}} - \sum_{k=1}^{K} \widehat{\boldsymbol{\mathcal{D}}}_k * [\![\widehat{\boldsymbol{Z}}_{k,1}, \cdots, \widehat{\boldsymbol{Z}}_{k,p}]\!]\|_F^2 = \|\widehat{\boldsymbol{Y}}^{(q)} - \sum_{k=1}^{K} \widehat{\boldsymbol{D}}_k^{(q)} * \widehat{\boldsymbol{Z}}_{k,q} \left(\overset{\hookleftarrow p}{\odot}_{i=1} \widehat{\boldsymbol{Z}}_k^{(i)}\right)^{\mathsf{T}} \|_F^2 \quad \text{(matricization)} \\
&= \|\widehat{\boldsymbol{y}}^{(q)} - \sum_{k=1}^{K} \widehat{\boldsymbol{d}}_k^{(q)} * \left(\overset{\hookleftarrow p}{\odot}_{i=1} \widehat{\boldsymbol{Z}}_k^{(i)} \otimes I\right) \mathrm{vec}(\widehat{\boldsymbol{Z}}_{k,q})\|_F^2 \quad \text{(vectorization)} \\
&= \|\widehat{\boldsymbol{y}}^{(q)} - \sum_{k=1}^{K} \mathrm{diag}(\widehat{\boldsymbol{d}}_k^{(q)}) \left(\overset{\hookleftarrow p}{\odot}_{i=1} \widehat{\boldsymbol{Z}}_k^{(i)} \otimes I\right) \mathrm{vec}(\widehat{\boldsymbol{Z}}_{k,q})\|_F^2 \qquad (\boldsymbol{x} * \boldsymbol{y} = \mathrm{diag}(\boldsymbol{x})\boldsymbol{y}) \\
&= \|\widehat{\boldsymbol{y}}^{(q)} - \sum_{k=1}^{K} \mathrm{diag}(\widehat{\boldsymbol{d}}_k^{(q)}) \widehat{\boldsymbol{C}}_k \widehat{\boldsymbol{z}}_k\|_F^2 \ ,
\end{aligned}$$

where the last line is just notations. To obtain the final equality, we stack the matrices $\{\mathrm{diag}(\widehat{\boldsymbol{d}}_k^{(q)})\}$ and construct a block-diagonal matrix such that the block are the $\{\boldsymbol{C}_k\}$. Finally we obtain the following equality.

$$\begin{pmatrix} \widehat{\boldsymbol{C}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{C}}_K \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{B}}_1 \otimes \boldsymbol{I} & & \\ & \ddots & \\ & & \widehat{\boldsymbol{B}}_K \otimes \boldsymbol{I} \end{pmatrix} = \begin{pmatrix} \widehat{\boldsymbol{B}}_1 & & \\ & \ddots & \\ & & \widehat{\boldsymbol{B}}_K \end{pmatrix} \otimes \boldsymbol{I} \ ,$$

where $\widehat{\boldsymbol{B}}_k = (\overset{\hookleftarrow p}{\odot}_{i=1,i\neq q} \widehat{\boldsymbol{Z}}_{k,i})$. This end the proof. □

Proposition 3.2. *The matrix* $(\widehat{\boldsymbol{A}}^H \otimes \boldsymbol{I})\widehat{\boldsymbol{\Gamma}}^H\widehat{\boldsymbol{\Gamma}}(\widehat{\boldsymbol{A}} \otimes \boldsymbol{I})$ *is composed of* K^2 *blocks equals to*

$$\left(\overset{\hookleftarrow p}{\odot}_{i=1,i\neq q} \widehat{\boldsymbol{Z}}_{k,i})^H \otimes I\right) \overline{\mathit{diag}(\widehat{\boldsymbol{d}_k}^{(q)})} \mathit{diag}(\widehat{\boldsymbol{d}_\ell}^{(q)}) \left((\overset{\hookleftarrow p}{\odot}_{i=1,i\neq q} \widehat{\boldsymbol{Z}}_{\ell,i}) \otimes I\right) \ . \tag{3.40}$$

Each of these blocks can be computed in $\boldsymbol{\mathcal{O}}(R^2\prod_{i=1}^{p} n_i)$*. Hence, the full matrix can be computed in* $\boldsymbol{\mathcal{O}}((KR)^2\prod_{i=1}^{p} n_i)$ *operations. Furthermore, this matrix is a* $(KRn_q \times KRn_q)$ *banded matrix (as explain before). Its product with* $\widehat{\boldsymbol{z}}^{(q)}$ *can therefore be made in only* $\boldsymbol{\mathcal{O}}((KR)^2 n_q)$ *operations.*

Proof. The first step of the proof requires to write $\widehat{\boldsymbol{\Gamma}}^H$ as the Kronecker product of two specific matrices in order to use the equality $(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{AC} \otimes \boldsymbol{BD})$. Recall that $\widehat{\boldsymbol{\Gamma}}^H$ is a block-diagonal matrix, i.e. $\widehat{\boldsymbol{\Gamma}}^H = [\mathrm{diag}(\widehat{\boldsymbol{d}_1}^{(q)}), \cdots, \mathrm{diag}(\widehat{\boldsymbol{d}_K}^{(q)})]$. Hence, we can decompose each

diagonal-block $\text{diag}(\widehat{\boldsymbol{d}_k}^{(q)})$ into smaller diagonal matrices as follow

$$\text{diag}(\widehat{\boldsymbol{d}_k}^{(q)}) = \sum_{i=1}^{N_{\backslash q}} \text{diag}(e_i) \otimes \Delta_{k,i} \quad \text{with} \quad N_{\backslash q} = \prod_{i=1, i\neq q}^{p} n_i \ ,$$

with $\text{diag}(e_i) \in \mathbb{R}^{N_{\backslash q} \times N_{\backslash q}}$ and $\Delta_{k,i} \in \mathbb{C}^{n_q \times n_q}$ being the i-th diagonal block of $\text{diag}(\widehat{\boldsymbol{d}_k}^{(q)})$ (i.e. $\Delta_{k,i} = \text{diag}(\widehat{\boldsymbol{d}_k}^{(q)})_{(i\cdot n_q:(i+1)\cdot n_q),(i\cdot n_q:(i+1)\cdot n_q)}$). As $(\text{diag}(e_i) \otimes \Delta_{k,i})$ is decomposed in two matrices of the proper dimension, we can used the equality $(\boldsymbol{A} \otimes \boldsymbol{B})(\boldsymbol{C} \otimes \boldsymbol{D}) = (\boldsymbol{AC} \otimes \boldsymbol{BD})$ and we have

$$\begin{aligned}
&\left((\odot_{i=1,i\neq q}^{p} \widehat{\boldsymbol{Z}}_{k,i})^H \otimes I\right) \overline{\text{diag}(\widehat{\boldsymbol{d}_k}^{(q)})} \text{diag}(\widehat{\boldsymbol{d}_\ell}^{(q)}) \left((\odot_{i=1,i\neq q}^{p} \widehat{\boldsymbol{Z}}_{\ell,i}) \otimes I\right) \\
&= \left(\widehat{\boldsymbol{B}}_k^H \otimes I\right) \sum_{i=1}^{N_{\backslash q}} \left(\text{diag}(e_i) \otimes \overline{\Delta_{k,i}}\right) \sum_{j=1}^{N_{\backslash q}} \left(\text{diag}(e_j) \otimes \Delta_{\ell,j}\right) \left(\widehat{\boldsymbol{B}}_\ell \otimes I\right) \\
&= \sum_{i=1}^{N_{\backslash q}} \sum_{j=1}^{N_{\backslash q}} \left(\widehat{\boldsymbol{B}}_k^H \otimes I\right) \left(\text{diag}(e_i) \otimes \overline{\Delta_{k,i}}\right) \left(\text{diag}(e_j) \otimes \Delta_{\ell,j}\right) \left(\widehat{\boldsymbol{B}}_\ell \otimes I\right) \\
&= \sum_{i=1}^{N_{\backslash q}} \sum_{j=1}^{N_{\backslash q}} \left(\widehat{\boldsymbol{B}}_k^H \text{diag}(e_i)\text{diag}(e_j) \widehat{\boldsymbol{B}}_\ell \otimes \overline{\Delta_{k,i}} \Delta_{\ell,j}\right) \\
&= \sum_{i=1}^{N_{\backslash q}} \left(\widehat{\boldsymbol{B}}_k^H \text{diag}(e_i)\text{diag}(e_i) \widehat{\boldsymbol{B}}_\ell \otimes \overline{\Delta_{k,i}} \Delta_{\ell,i}\right) \\
&= \sum_{i=1}^{N_{\backslash q}} \left((\text{diag}(e_i)\widehat{\boldsymbol{B}}_k)^H \text{diag}(e_i) \widehat{\boldsymbol{B}}_\ell \otimes \overline{\Delta_{k,i}} \Delta_{\ell,i}\right) = \sum_{i=1}^{N_{\backslash q}} \left(\overline{\widehat{\boldsymbol{B}}}_k(i,:) \circ \widehat{\boldsymbol{B}}_\ell(i,:) \otimes \overline{\Delta_{k,i}} \Delta_{\ell,i}\right) \ .
\end{aligned}$$

The outer product of two vectors in $\mathbb{C}^{1\times R}$ is of complexity $\boldsymbol{\mathcal{O}}(R^2)$. This product is made for each $1 \leq i \leq N_{\backslash q}$ and for each K^2 blocks. Hence, the overall complexity is $\boldsymbol{\mathcal{O}}((KR)^2 \prod_{i=1,i\neq q}^{p} n_i)$. □

9 Notation and preliminaries on tensor

In the sequel, we recall the tensor algebra concepts which allowed us to extend the CDL to multivariate signals. Please refer to [Kolda and Bader, 2009; Sidiropoulos et al., 2017] for a more in-depth introduction on the tensor algebra topic.

9.1 Some important definitions and formulas

A tensor is a multidimensional array extending the notion of vectors and matrices. Formally, a p-th order tensor is an element of the tensor product of $p \in \mathbb{N}_*$ vector spaces, denoted $\boldsymbol{\mathcal{X}} \in \mathbb{X} \triangleq \mathbb{R}^{n_1 \times \cdots \times n_p}$ and addressed by p indexes. Whereas in matrices we can extract rows or columns, in tensors we can extract slices, fibers, or elements. A slice of a tensor is the matrix obtained by fixing all its indexes except two, while a fiber is a vector obtained by fixing all its indexes except one. Slice, fiber, or element are denoted in equivalent ways $\boldsymbol{\mathcal{X}}_{:,:,\cdots,i_p}$, $\boldsymbol{\mathcal{X}}_{:,\cdots,i_p}$, or $\boldsymbol{\mathcal{X}}_{i_1,\cdots,i_p}$.

9.1.1 Some products

In this section, we review some useful products and their properties, as they pertain to tensor computations. These operations greatly facilitates the understanding of this particular algebra and lightens the notations. We start by three important matrix products.

Definition 3.1. (Kronecker, Khatri-Rao, and Hadamard product) – *The Kronecker product between* $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *and* $\boldsymbol{B} \in \mathbb{R}^{k\times \ell}$ *is denoted* $\boldsymbol{A} \otimes \boldsymbol{B}$*. The result is a matrix of size* $(mk) \times (n\ell)$ *such that*

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{pmatrix} a_{1,1}\boldsymbol{B} & \cdots & a_{1,n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m,1}\boldsymbol{B} & \cdots & a_{m,n}\boldsymbol{B} \end{pmatrix} .$$

The Khatri-Rao product [Smilde et al., 2005] between $\boldsymbol{A} \in \mathbb{R}^{m\times k}$ *and* $\boldsymbol{B} \in \mathbb{R}^{n\times k}$ *is denoted* $\boldsymbol{A} \odot \boldsymbol{B}$*. The result is a matrix of size* $(mn) \times (k)$ *such that*

$$\boldsymbol{A} \odot \boldsymbol{B} = [\boldsymbol{a}_{:,1} \otimes \boldsymbol{b}_{:,1}, \cdots, \boldsymbol{a}_{:,k} \otimes \boldsymbol{b}_{:,k}] .$$

The Hadamard product, or component-wise product, between $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ *and* $\boldsymbol{B} \in \mathbb{R}^{m\times n}$ *is denoted* $\boldsymbol{A} * \boldsymbol{B}$*. The result is also a matrix of size* $m \times n$ *such that* $(\boldsymbol{A} * \boldsymbol{B})_{i,j} = \boldsymbol{A}_{i,j} \cdot \boldsymbol{B}_{i,j}$.

Definition 3.2. (Inner product and induced norm) – *Let* $\boldsymbol{\mathcal{X}}$ *and* $\boldsymbol{\mathcal{Y}}$ *be two tensors in* $\mathbb{X}$*. The inner product between* $\boldsymbol{\mathcal{X}}$ *and* $\boldsymbol{\mathcal{Y}}$ *is given by*

$$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \sum_{i_1=1}^{n_1} \cdots \sum_{i_p=1}^{n_p} \boldsymbol{\mathcal{X}}_{i_1,\cdots,i_p} \boldsymbol{\mathcal{Y}}_{i_1,\cdots,i_p} = vec(\boldsymbol{\mathcal{X}})^\intercal vec(\boldsymbol{\mathcal{Y}}) .$$

The norm induced by this inner product is the Frobenius norm denoted $\| \cdot \|_F$*, and such that* $\|\boldsymbol{\mathcal{X}}\|_F = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle^{1/2}$ *i.e. the square root of the sum of the squares of all the elements of* $\boldsymbol{\mathcal{X}}$*.*

Multiplication between tensors and matrices is defined using the m-mode product.

Definition 3.3. (Mode-m product) – *For* $m \in \{1, \cdots, p\}$ *and* $\boldsymbol{A}$ *in* $\mathbb{R}^{n_m \times n_q}$*, the mode-*$m$ *product between* $\boldsymbol{\mathcal{X}}$ *and* $\boldsymbol{A}$ *is given by*

$$(\boldsymbol{\mathcal{X}} \times_m \boldsymbol{A})_{i_1,\cdots,i_{m-1},j,i_{m+1},\cdots,p} = \sum_{k=1}^{n_m} \boldsymbol{\mathcal{X}}_{i_1,\cdots,i_{m-1},k,i_{m+1},\cdots,p} \boldsymbol{A}_{k,j} .$$

The mode product of $\boldsymbol{\mathcal{X}}$ with two proper matrices $\boldsymbol{U}, \boldsymbol{V}$ admits the two following fundamental properties

$$\begin{aligned} &\boldsymbol{\mathcal{X}} \times_m \boldsymbol{U} \times_n \boldsymbol{V} = \boldsymbol{\mathcal{X}} \times_n \boldsymbol{V} \times_m \boldsymbol{U} \qquad (m \neq n) \\ &\boldsymbol{\mathcal{X}} \times_m \boldsymbol{U} \times_m \boldsymbol{V} = \boldsymbol{\mathcal{X}} \times_m \boldsymbol{U}\boldsymbol{V} . \end{aligned}$$

An illustration of this product is given in Figure 3.27.

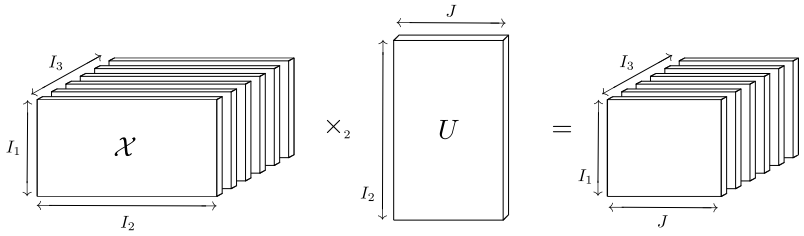


Figure 3.27: Illustration of the Mode-2 product with a third order tensor.

9.1.2 Canonical Polyadic Decomposition and tensor rank

Tensor algebra has many similarities but also many striking differences with matrix algebra. One of the main difference is related to the definition of the rank which is not unique as in the matrix case. Many definitions exist in the literature which are not equivalent in general. The most well known is called the *Canonical Polyadic rank* (CP-rank) of a tensor.

Proposition 3.3. (Canonical Polyadic Decomposition/PARAFAC and CP-rank) – *For any tensor* $\boldsymbol{\mathcal{X}} \in \mathbb{X}$*, there exist* $R > 0$*, and,* $\boldsymbol{x}_r^{(i)} \in \mathbb{R}^{n_i}$*,* $1 \leq i \leq p$*,* $1 \leq r \leq R$*, such that*

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \boldsymbol{x}_r^{(1)} \circ \cdots \circ \boldsymbol{x}_r^{(p)} \ . \tag{3.41}$$

The smallest R *for which such decomposition exists is called the Canonical Polyadic rank of* $\boldsymbol{\mathcal{X}}$ *(CP-rank*$(\boldsymbol{\mathcal{X}})$ *or rank*$(\boldsymbol{\mathcal{X}})$ *for short), and in this case* (3.41) *is referred to as the CP decomposition of* $\boldsymbol{\mathcal{X}}$*.*

Definition 3.4. (Kruskal operator [Kruskal, 1977]) – *With the notation of Proposition 3.3, the Kruskal operator* $[\![\cdot]\!]$ *is defined as*

$$[\![\boldsymbol{X}_1, \cdots, \boldsymbol{X}_p]\!] \triangleq \sum_{r=1}^{R} \boldsymbol{x}_r^{(1)} \circ \cdots \circ \boldsymbol{x}_r^{(p)} \ ,$$

where $\boldsymbol{X}_i = \left[\boldsymbol{x}_1^{(i)} \mid \ldots \mid \boldsymbol{x}_R^{(i)} \right] \in \mathbb{R}^{n_i \times R}$*,* $1 \leq i \leq p$*.*

Remark 3.4. *A CP-decomposition is always possible for a (possibly) non-optimal* R *by considering the canonical basis.*

9.1.3 Matricization and vectorization

Matricization, also known as *unfolding* or *flattening*, is the process of reordering the elements of a tensor into a matrix. For instance, we can rearranged a tensor in $\mathbb{R}^{n_1 \times n_2 \times n_3}$ into a matrix in $\mathbb{R}^{(n_1 \cdot n_2) \times n_3}$. The matricization operation permits a better comprehension of the tensor object and is very useful in practice (e.g. optimization). Before the introduction of a proper definition, we recall that a *slice* of a tensor is the matrix obtained by fixing all its indexes except two. As an illustration let us consider a third order tensor $\boldsymbol{\mathcal{X}}$ in $\mathbb{R}^{n_1 \times n_2 \times n_3}$. A slice i is here denoted by $\boldsymbol{\mathcal{X}}(i, :, :)$ or $\boldsymbol{\mathcal{X}}_{i::}$. The two other slices are defined equally.

Definition 3.5. (q-mode matricization of a tensor) – *Let* $\boldsymbol{\mathcal{X}}$ *be a tensor in* $\mathbb{R}^{n_1 \times \cdots \times n_p}$*. The* q*-mode matricization of* $\boldsymbol{\mathcal{X}}$ *is a matrix in* $\mathbb{R}^{n_q \times \prod_{i=1, i \neq q}^{p} n_q}$ *denoted* $\boldsymbol{X}^{(q)}$ *and obtained by stacking all slices of* $\boldsymbol{\mathcal{X}}$ *except the* q*-th.*

Converting a tensor to a matrix is useful both computationally and theoretically as there exist connections between the matricization, and the Kruskal operator. One of the most important proposition is given in the following.

Proposition 3.4. (Matricization of the Kruskal operator) – *Let $\boldsymbol{\mathcal{X}}$ be a tensor in $\mathbb{X}$ with CP-decomposition $[\![\boldsymbol{X}_1, \cdots, \boldsymbol{X}_p]\!]$. Then,*

$$\boldsymbol{X}^{(q)} = \boldsymbol{X}_q \left(\boldsymbol{X}_p \odot \cdots \odot \boldsymbol{X}_{q+1} \odot \boldsymbol{X}_{q-1} \odot \cdots \odot \boldsymbol{X}_1\right)^\intercal = \boldsymbol{X}_q \left(\overset{\hookleftarrow p}{\underset{i=1, i\neq q}{\odot}} \boldsymbol{X}_i \right)^\intercal ,$$

where $\odot$ is the Khatri–Rao product (see definition 3.1) and $\overset{\hookleftarrow p}{\underset{i=1}{\odot}}$ denotes the product of p Khatri–Rao products in reverse order. We can also vectorized this formula which gives us

$$\begin{aligned} vec(\boldsymbol{X}^{(q)}) &= \left(\boldsymbol{X}_p \odot \cdots \odot \boldsymbol{X}_{q+1} \odot \boldsymbol{X}_{q-1} \odot \cdots \odot \boldsymbol{X}_1 \otimes \boldsymbol{I}\right) vec(\boldsymbol{X}_q) \\ &= \left(\overset{\hookleftarrow p}{\underset{i=1, i\neq q}{\odot}} \boldsymbol{X}_i \otimes \boldsymbol{I}_{n_q} \right) vec(\boldsymbol{X}_q) , \end{aligned}$$

where $\boldsymbol{I}_{n_q}$ is the identity matrix of size $(n_q \times n_q)$.

We now go back to the CDL. As we want to extend it to tensor signals, we are confronted to the problem of a correct definition of the convolutional operator. Fortunately, the convolutional operator for multidimensional signals is well defined and does not differ much from the one for one-dimensional signals. We recall its properties in the next section.

9.2 How to perform the convolution for discrete signals?

The CDL equation (3.1) contains the convolution operator $\star$. However, for discrete signals (seen as vectors), there exists several ways to perform such convolution. In this section, we address this issue by presenting the different ways to proceed.

The standard adaptation of the convolution for discrete signals leads to the following definition.

Definition 3.6. (Discrete convolution) – *Let consider two discrete functions f, g defined on all the set of integer $\mathbb{Z}$ i.e. with infinite support. The convolution between this two functions is called the discrete convolution and is given by*

$$(f \star g)[n] = \sum_{k=-\infty}^{+\infty} f[k] g[n-k] .$$

Here, we use the notation $f[\cdot]$ to highlight the discrete structure of the functions.

In this definition, we have considered discrete signals with infinite support i.e. $f[n]$ is defined for all n in $\mathbb{Z}$. However, in practice, f is usually known over a finite domain, (e.g. $0 \leq n < N$) and the convolution must be modified to take into account this border effects. To compute the discrete convolution between two discrete functions f, g with finite support, one approach is to assume that values outside the domain of consideration are 0 (also referred as *Dirichlet boundary* [Bristow and Lucey, 2014]). Another popular approach is to extend f, g with a periodization by introducing two functions $\tilde{f}$ and $\tilde{g}$ such that

$$\tilde{f}[n] = f[n \bmod N] , \ \tilde{g}[n] = g[n \bmod N] .$$

Here, $\tilde{f}$ and $\tilde{g}$ are two discrete functions with period N (see Figure 3.28 for an example). This strategy leads to the definition of the *circular convolution*.

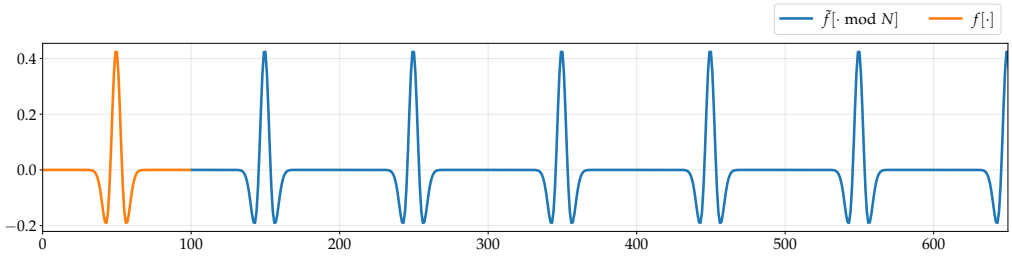


Figure 3.28: Periodization of the "finite" function f draw in orange.

Definition 3.7. (Circular discrete convolution) – *Let consider two functions $\tilde{f}, \tilde{g}$ defined on $\{0, \dots, N-1\}$ with period N. The circular convolution between $\tilde{f}$ and $\tilde{g}$ is given by*

$$(\tilde{f} \circledast \tilde{g})[n] = \sum_{k=0}^{N-1} \tilde{f}[k]\tilde{g}[n-k] \ .$$

The premise behind the circular convolution approach is to develop a relation between the Convolution theorem and the Discrete Fourier Transform in order to calculate the convolution between two finite-extent, discrete-valued signals. Indeed, remark that, $\tilde{f} \circledast \tilde{g}$ is a signal of period N. It can therefore be decomposed in a Fourier basis like classical periodic signals which gives rise to the following important theorem.

Definition 3.8. (Discrete Fourier Transform (DFT)) – *Let consider a function f defined on $\{0, \dots, N-1\}$ with period N. The Discrete Fourier Transform (DFT) of f is given by*

$$\widehat{f}[k] = \sum_{n=0}^{N-1} f[n] \exp\left(-\frac{i2\pi kn}{N}\right) \ ,$$

and the Inverse DFT (IDFT) of $\widehat{f}$ is given by

$$f[n] = \frac{1}{N} \sum_{k=0}^{N-1} \widehat{f}[k] \exp\left(\frac{i2\pi kn}{N}\right) \ ,$$

where $\widehat{\cdot}$ denotes the frequency representation of a signal.

Theorem 3.3. (Discrete convolution theorem) – *If f and g have period N, then the DFT of $h = f \circledast g$ is*

$$\widehat{h}[n] = \widehat{f}[n] \cdot \widehat{g}[n] \ , \quad \text{or in vector notation} \quad \widehat{h} = \widehat{f} * \widehat{g} \ ,$$

where $\widehat{\cdot}$ denotes the frequency representation of a signal, and $*$ is the component-wise product.

This theorem is the core of most methods that solve the CDL problem as it allows to take advantage of the Fast Fourier Transform (FFT) to significantly reduces the complexity of the algorithms. Indeed, a direct computation of $\widehat{h}$ – with the summation – requires $\mathcal{O}(N^2)$ multiplications. With the FFT the complexity reduces to $\mathcal{O}(N \log(N))$.

Remark 3.5. *If f and g do not have the same support, we extend the one with the lowest support with zeros (zero-padding).*

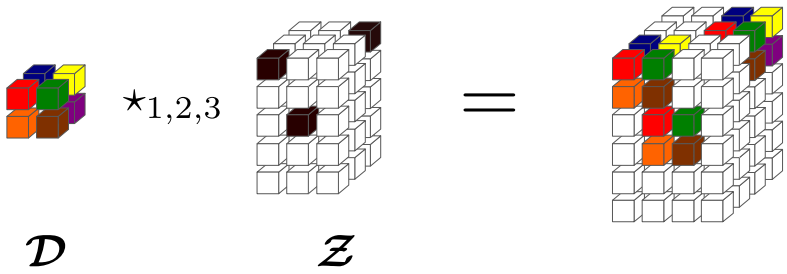


Figure 3.29: Illustration of the multidimensional convolution (Dirichlet boundary version) with 3-th order tensors, where each cube represents a dimension and each axis an order. Notice that the result has one additional dimension in each order.

9.3 How to perform the convolution for multidimensional signals?

The standard adaptation of the convolution for multivariate discrete signals (seen as tensors) leads to the following definition.

Definition 3.9. (Discrete convolution) – *Let consider two p-dimensional discrete functions $\mathcal{F}, \mathcal{G}$ defined on all the set of integer $\mathbb{Z}^p$ i.e. with infinite support. The convolution between this two functions is called the discrete convolution and is given by*

$$(\mathcal{F} \star \mathcal{G})[n_1, \cdots, n_p] = \sum_{k_1=-\infty}^{+\infty} \cdots \sum_{k_p=-\infty}^{+\infty} \mathcal{F}[k_1, \cdots, k_p]\mathcal{G}[n_1 - k_1, \cdots, n_p - k_p] \ .$$

When the convolution is only performed on some dimensions, we use the symbol $\star_{1,2,\dots}$ where the subscript numbers are the dimension involved (see Figure 3.29).

Remark 3.6. *For unidimensional signal, $\star_1$ reduces to the 1-D discrete convolutional operator.*

In this definition, we have considered discrete multidimensional signals with infinite support. To compute the discrete convolution between two discrete function $\mathcal{F}, \mathcal{G}$ with finite support, one approach is to assume that values outside the domain of consideration are 0 (also referred as *Dirichlet boundary* [Bristow and Lucey, 2014]). However, as in the univariate case (see Section 9.2), to develop a relation between the Convolution theorem and the DFT, we use the circular convolution for multivariate discrete signals.

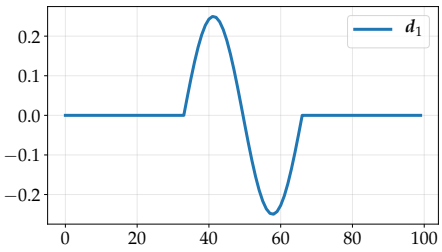
Let consider the periodization of $\mathcal{F}$ and $\mathcal{G}$,

$$\begin{aligned}\tilde{\mathcal{F}}[n_1, \cdots, n_p] &= \mathcal{F}[n_1 \bmod N_1, \cdots, n_p \bmod N_p] \\ \tilde{\mathcal{G}}[n_1, \cdots, n_p] &= \mathcal{G}[n_1 \bmod N_1, \cdots, n_p \bmod N_p] \ .\end{aligned}$$

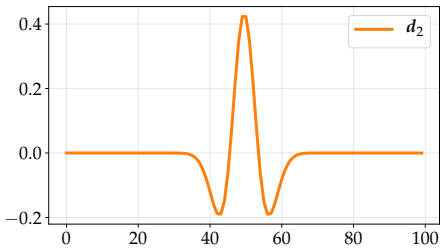
Here, $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}$ are now two discrete functions with period $(N_1, \cdots, N_p)$ (each of the modes are periodic one-dimensional signals). The *circular convolution* is defined as follow.

Definition 3.10. (Circular discrete convolution) – *Let consider two functions $\mathcal{F}, \mathcal{G}$ defined on $\{0 \dots, N_1 - 1\} \times \cdots \times \{0 \dots, N_p - 1\}$ with both a period of $(N_1, \cdots, N_p)$. The circular convolution between $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}$ is given by*

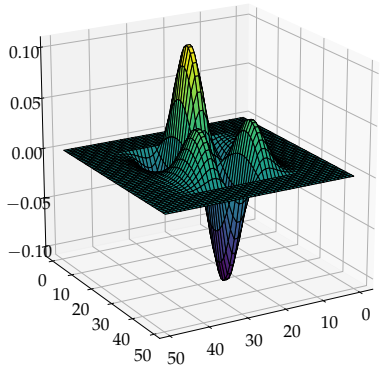
$$(\tilde{\mathcal{F}} \circledast \tilde{\mathcal{G}})[n_1, \cdots, n_p] = \sum_{k_1=0}^{N_1-1} \cdots \sum_{k_p=0}^{N_p-1} \tilde{\mathcal{F}}[k_1, \cdots, k_p]\tilde{\mathcal{G}}[n_1 - k_1, \cdots, n_p - k_p] \ .$$



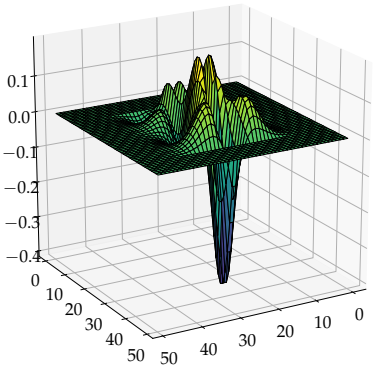
(a) Atom $\boldsymbol{d}_1$



(b) Atom $\boldsymbol{d}_2$



(c)



(d)

Figure 3.30: (c) Illustration of a two dimensional separable function $\boldsymbol{\mathcal{F}} = f_1 \circ f_2$ with f_1 and f_2 being the atoms (a) and (b). (c) Illustration of a two dimensional multi-separable function $\boldsymbol{\mathcal{F}} = \sum_{k=1}^{4} f_{1,k} \circ f_{1,k}$ with $\{f_{1,k}\}$ and $\{f_{2,k}\}$ being different dilatations of the atoms (a) and (b).

$\tilde{\boldsymbol{\mathcal{F}}} \circledast \tilde{\boldsymbol{\mathcal{G}}}$ is a signal of period $(N_1, \cdots, N_p)$ and can be decomposed in a Fourier basis like classical periodic signals which give rises to the following important theorem.

Theorem 3.4. (Discrete convolution theorem) – *If $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$ have period $(N_1, \cdots, N_p)$, then the DFT of $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{F}} \circledast \boldsymbol{\mathcal{G}}$ is*

$$\widehat{\boldsymbol{\mathcal{H}}}[n_1, \cdots, n_p] = \widehat{\boldsymbol{\mathcal{F}}}[n_1, \cdots, n_p] * \widehat{\boldsymbol{\mathcal{G}}}[n_1, \cdots, n_p] \ , \quad \text{or in tensor notation} \quad \widehat{\boldsymbol{\mathcal{H}}} = \widehat{\boldsymbol{\mathcal{F}}} * \widehat{\boldsymbol{\mathcal{G}}} \ ,$$

where $\widehat{\cdot}$ denotes the frequency representation of a signal, and $*$ is the component-wise product.

Remark 3.7. *If $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$ do not have the same support, we extend the one with the lowest support with zeros (zero-padding).*

A direct computation of $\widehat{\boldsymbol{\mathcal{H}}}$ with the summation requires $\mathcal{O}(\prod_{i=1}^{p} N_i^2)$ multiplications. With the p-dimensional FFT the complexity becomes $\mathcal{O}(\sum_{i=1}^{p} N_i \log(\sum_{i=1}^{p} N_i))$. We have extensively use this theorem to accelerate our algorithms.

9.4 Separable signals

One important difference of multivariate signals over the univariate ones is the notion of *separability*. With this notion, we can avoid the complexity introduced by the additional dimensions. This not only simplifies formulas, but also leads to fast numerical algorithms.

Definition 3.11. (Separable discrete signal) – *A discrete signal $\boldsymbol{\mathcal{F}}$ is said to be separable if it can be write as the outer product of univariate signals i.e.*

$$\boldsymbol{\mathcal{F}} = f_1 \circ \cdots \circ f_p \ , \tag{3.42}$$

where the $\{f_i\}_{i=1}^{p}$ are univariate discrete signals.

When $\boldsymbol{\mathcal{F}}$ is write as a tensor, we see that $\boldsymbol{\mathcal{F}}$ is separable if its CP-rank is equal to 1. We can easily extend this definition to *"multi"*-separable function by considering signals equals to the summation of multiple separable signals, i.e. tensor with CP-rank > 1. This extension allows to consider separable signals with more complex structure (see examples in Figure 3.30).

4

Subsampling of multivariate time-vertex graph signals

Abstract

In this chapter, we present an approach for processing and subsampling multivariate time-vertex graph signals. The main idea is to model the relationships within each dimension (time, space, feature space) with different graphs and to merge these structures with graph products. Our technique based on a tensor formalism aims at identifying the frequency support of the graph signal in order to preserve its content after subsampling. Results are provided on real electroencephalogram signals.

Contents

The material of this chapter is based on the following publication:

P. Humbert, L. Oudre, and N. Vayatis. Subsampling of Multivariate Time-Vertex Graph Signals. In *European Signal Processing Conference (EUSIPCO)*, 2019.

1 Introduction

Graph Signal Processing (GSP) [Shuman et al., 2013; Ortega et al., 2018] has emerged as a powerful field to analyze structured data as it allows, for instance, to handle complex signal such as those recorded with sensor networks. Indeed, by assuming that each component of the signal lies on a graph node, complex spatial interactions or dependencies can be taken into account for several tasks such as sampling, filtering, or reconstruction [Sandryhaila and Moura, 2013; Chen et al., 2015b,c,d; Marques et al., 2016].
In most works, this type of signal, called graph signal, only refers to a single time instance (e.g. its acquisition time) and hence encodes the variation of an instantaneous observation over an underlying graph structure. Therefore, very often the time variations are not taken into account in the processing of such signals: studies consider either one time-sample [Wagner et al., 2005; Jain et al., 2014; Mohan et al., 2014], where only the spatial dimension is analyzed, or an average on a time window. In order to deal with temporal graph signals, recent works have introduced the notion of *time-vertex* signal processing, where both spatial and temporal interactions are modeled [Grassi et al., 2018]. In this context, another Graph Fourier Transform (GFT), called Joint Fourier Transform (JFT), has been introduced [Sandryhaila and Moura, 2014; Loukas and Foucard, 2016] and efficiently used in several examples such as video inpainting, seismic epicenter localization [Grassi et al., 2018], and recovery of high-dimensional processes evolving over a graph (spread modeling) [Loukas and Perraudin, 2019].
In the case of multivariate sensor networks, or feature-based representations, one solution may consist in treating each feature or modality individually. However, the underlying assumption is that all variables are independent, which is not true in a lot of typical situations, such as meteorological data which could be composed of several correlated variables (temperature, atmospheric pressure, rainfalls) over time and space. In this situation, a third graph layer is needed to also model the links between the different modalities. As a result, multivariate time-vertex graph signals should be modeled with three types of interactions: one in time, one in space, and one in feature space. By combining the notion of graph product with the tensor formalism, we show in this chapter that it is possible to extend the notion of GFT to multivariate graph signals, and to provide efficient algorithms for processing them.

This chapter provides a framework for processing multivariate time-vertex graph signals, based on the notion of graph product and the definition of three graphs that each model the interactions within one dimension (time, space, feature space). By using the tensor formalism, several sparsity methods are provided, that can be specified so as to work only on one dimension (i.e. selection of the best time samples, sensors or features). These approaches are tested on real ElectroEncephaloGram (EEG) signals in order to assess the sampling and interpolation performances of the proposed framework.

2 Background and notations

We first recall the notations used in this chapter and introduce the product graph.

2.1 Tensor algebra

Let $d_1, d_2, \ldots, d_p \in \mathbb{N}_*$ and $\mathbb{Y} = \mathbb{R}^{d_1} \times \ldots \times \mathbb{R}^{d_p} \triangleq \mathbb{R}^{d_1 \times \cdots \times d_p}$ be the product of p $\mathbb{R}$-vector spaces. Recall that an element of $\boldsymbol{\mathcal{Y}} \in \mathbb{Y}$ is called a tensor of order p. In the following, $\boldsymbol{\mathcal{Y}}$ will be used indifferently to denote the multilinear form in $\mathbb{Y}^*$ and its representation in the canonic

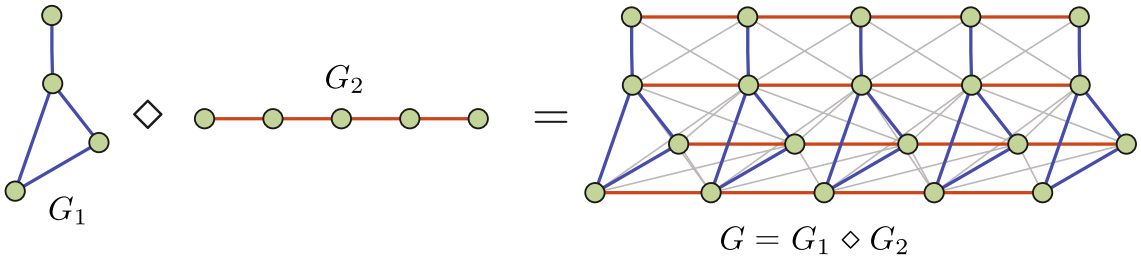


Figure 4.1: Illustration of the product graph $\mathcal{G}$ between two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$. $\diamond$ represents either a Cartesian (only colored edges), a Kronecker (only gray edges) or a strong product (all edges) between these two graphs. Figure modified from the original one in [Ortiz-Jiménez et al., 2018].

base of $\mathbb{Y}$, the choice being clear from the context. The *mode-m matrix product* between a tensor $\boldsymbol{\mathcal{Y}}$ and a matrix $\boldsymbol{X} \in \mathbb{R}^{j \times d_m}$ in coordinate notation is $(\boldsymbol{\mathcal{Y}} \times_m \boldsymbol{X})_{i_1,\cdots,i_{m-1},j,i_{m-1},\cdots,i_p} \triangleq \sum_{k=1}^{d_m} \boldsymbol{\mathcal{Y}}_{i_1,\cdots,i_{m-1},k,i_{m-1},\cdots,n} \boldsymbol{X}_{j,k}$ and is equivalent to $\boldsymbol{\mathcal{Y}} \times_m \boldsymbol{X} \Leftrightarrow \boldsymbol{X}\boldsymbol{\mathcal{Y}}^{(m)}$ where $\boldsymbol{\mathcal{Y}}^{(m)}$ denotes the tensor $\boldsymbol{\mathcal{Y}}$ unfolded along axis m. The operator $\otimes$ represent the Kronecker product. When multiple products are necessary, we use the upper version of these notations, $\bigtimes$ and $\bigotimes$. See Appendix 9 for a complete presentation.

2.2 Product graph

Let $G = (\mathcal{V}, \mathcal{E})$ be a directed weighted graph with nodes $\mathcal{V} = \{1, \cdots, N\}$, edges $\mathcal{E} = \{(i, j, w_{ij}), i, j \in \mathcal{V}\}$, and weights $w_{ij} \in \mathbb{R}^+$. As stated in the previous chapter, the Laplacian matrix $\boldsymbol{L}$ of the graph is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$ is the degree matrix and $\boldsymbol{W}$ the weights matrix (Definition 2.5). For simplicity, we assume that $\boldsymbol{L}$ is diagonalizable. Its eigendecomposition is $\boldsymbol{L} = \boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{X}^{-1}$, with $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ a diagonal matrix with the eigenvalues and $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ a matrix with the eigenvectors as columns. If $\boldsymbol{L}$ is not diagonalizable, Jordan decomposition into generalized eigenvectors is used.

Let $G_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $G_2 = (\mathcal{V}_2, \mathcal{E}_2)$ be two graphs with N_1 and N_2 vertices and Laplacian $\boldsymbol{L}_1 = \boldsymbol{X}_1\boldsymbol{\Lambda}_1\boldsymbol{X}_1^{-1}$, $\boldsymbol{L}_2 = \boldsymbol{X}_2\boldsymbol{\Lambda}_2\boldsymbol{X}_2^{-1}$, respectively. A product graph of G_1 and G_2, denoted by the symbol $\diamond$, is the graph with Laplacian equal to

$$\bar{\boldsymbol{L}} = (\boldsymbol{X}_1 \otimes \boldsymbol{X}_2)\boldsymbol{\Lambda}_\diamond(\boldsymbol{X}_1^{-1} \otimes \boldsymbol{X}_2^{-1}) \,, \tag{4.1}$$

where $\boldsymbol{\Lambda}_\diamond$ depends of the choice of the product [Imrich and Klavzar, 2000; Hammack et al., 2011; Leskovec et al., 2010; Sandryhaila and Moura, 2014] (see Figure 4.1).

2.3 Graph signal processing

A bivariate graph signal can be represented as a matrix $\boldsymbol{Y} \in \mathbb{R}^{N_1 \times N_2}$, where $Y_{i,j}$ is the value at the i-th node of G_1 and j-th node of $G_2^{(i)}$. Using the Graph Fourier Transform (GFT) it is possible to create a spectral representation $\boldsymbol{H}$ of $\boldsymbol{Y}$ defined as

$$\boldsymbol{H} = \boldsymbol{Y} \times_1 \boldsymbol{X}_1^{-1} \times_2 \boldsymbol{X}_2^{-1} \,. \tag{4.2}$$

The eigenvalues can be interpreted as distinct frequencies, the components of $\boldsymbol{H}$ as Fourier coefficients, and the eigenvectors as a decomposition basis. Notice that if G_1 is a cycle graph, $\boldsymbol{X}_1$ is the Discrete Fourier Transform matrix, and the GFT formula (4.2) is exactly the JFT. Hence, the JFT could be seen as a particular case of the multidimensional GFT.

With the tensor formalism used in (4.2), it is straightforward to extend the previous definitions to product graph with more than two related graphs. Given a collection of M graphs $(G_m)_{m=1}^M$ with $(N_m)_{m=1}^M$ vertices and Laplacian $(\boldsymbol{L}_m = \boldsymbol{X}_m \boldsymbol{\Lambda}_m \boldsymbol{X}_m^{-1})_{m=1}^M$, the Laplacian of the (full) product graph is

$$\bar{\boldsymbol{L}} = (\bigotimes_{m=1}^{M} \boldsymbol{X}_m) \boldsymbol{\Lambda}_\diamond (\bigotimes_{m=1}^{M} \boldsymbol{X}_m^{-1}) \ , \tag{4.3}$$

where $\boldsymbol{\Lambda}_\diamond$ is a matrix which depends of the choice of the product. As an example, if we choose the cartesian product, $\boldsymbol{\Lambda}_\diamond = \bigoplus_{m=1}^{M} \boldsymbol{\Lambda}_m$ where $\bigoplus$ is the Kronecker sum [Merris, 1998].

The GFT of a tensor graph signal $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{N_1 \times \cdots \times N_M}$ is therefore

$$\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{Y}} \times_{m=1}^{M} \boldsymbol{X}_m^{-1} \ . \tag{4.4}$$

This definition is the most important one as this is from it that we can identify the spectral support of multivariate signals.

3 Method

In this section, we propose to use tensor algebra to represent multivariate time-vertex graph signals. Using the extended version of GFT, we propose a subsampling technique that aims at recovering the whole signals by using a subset of features, time samples, or sensors.

3.1 Framework for processing multivariate time-vertex graph signals

In the context of graph signals obtained from multivariate sensor networks, data can be stored in a tensor $\boldsymbol{\mathcal{Y}}$ in $\mathbb{R}^{F \times T \times S}$, where F is the number of features recorded by the sensor, T is the number of time samples, and S is the number of sensors. Interactions between the different dimensions can be modelled with three different graphs that each encodes the interactions for one dimension:

- G_F – This graph quantifies the similarity between the different features or modalities of the data. There are several techniques to build such a graph. An intuitive approach is to consider a weighted correlation graph where the weights between two nodes corresponds to the absolute Pearson correlation coefficient between the modalities or features.

- G_T – This graph controls the interactions between time samples. One common choice is to use a directed cycle graph of size T, which links each sample to the next sample. This type of dependencies can be seen as a Markov process where the value of a sample only depends on the previous sample. This graph is widely used in the GSP community since, for this graph, the Graph Fourier Transform corresponds to the classical Fourier Transform. The weight (adjacency) matrix of G_T is a circulant matrix which is known to have as eigenvector matrix the discrete Fourier transform matrix [Huang et al., 2016; Loukas and Foucard, 2016; Segarra et al., 2016; Ortega et al., 2018]. Although G_T is directed, its Laplacian is still well defined by $\boldsymbol{L}_T = \boldsymbol{I}_T - \boldsymbol{W}_T$, where $\boldsymbol{I}_T$ is the identity matrix of size $(T \times T)$.

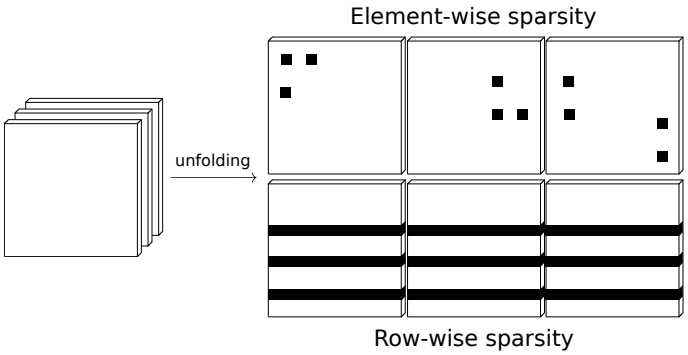


Figure 4.2: Illustration of the difference between the two sparsity norms. On the top, the element-wise sparsity norm. On the bottom, the row-wise sparsity norm. A black square means zero-value.

- G_S – In a sensor network, this graph models the interactions between the sensors. When dealing with a physical network, this graph can be based on physical links that exist between the sensors. When these interactions are unknown, an intuitive choice consists in building the graph in order to reflect the spatial closeness of each sensor. In general case, this graph is undirected and the edge weights can be built with the Gaussian function

$$\boldsymbol{W}_S(i,j) = \exp\Big(- \|s_i - s_j\|_2^2/\sigma^2 \Big) \,, \tag{4.5}$$

  where s_i is the spatial position of the i-th node of G_S.

Notice that since G_F and G_S are undirected, with no self-loops, and with a single connected component, their Laplacian are symmetric positive semi-definite and $\boldsymbol{X}_F^{-1} = \boldsymbol{X}_F^\intercal$, and $\boldsymbol{X}_S^{-1} = \boldsymbol{X}_S^\intercal$.

3.2 Identifying the support of the tensor graph signal

Most graph signal subsampling techniques are based on the assumption that the signal representation in the GFT domain is sparse [Narang et al., 2013; Anis et al., 2014; Chen et al., 2015b,d]. A graph signal with this property is called bandlimited with respect to its graph. When the frequency support of a graph signal is not known, we need to identify it in order to design a proper sampling and interpolation procedure. This problem leads to the following sparse signal reconstruction minimization

$$\min_{\boldsymbol{\mathcal{H}}} \quad \|\boldsymbol{\mathcal{Y}} \bigtimes_{m=1}^{M} \boldsymbol{X}_m^{-1} - \boldsymbol{\mathcal{H}}\|_F^2 + \Omega(\boldsymbol{\mathcal{H}}) \,, \tag{4.6}$$

where Ω is a regularization function imposing some sparsity on $\boldsymbol{\mathcal{H}}$. There are several valid choices for Ω. However, to obtain bandlimited tensor graph signal, we need to design a function which imposes sparsity on slices. We propose to use the two following functions, illustrated on Figure 4.2:

1. **General Sparsity (GS) constraint:**

$$\Omega : (\boldsymbol{\mathcal{H}}, \alpha) \longmapsto \alpha \|\boldsymbol{\mathcal{H}}\|_0 \,. \tag{4.7}$$

Notice that $\boldsymbol{\mathcal{H}}$ can be complex (e.g. if the graph is directed). In this case, $\|\cdot\|_0$ is naturally defined as the number of non-negative coefficients $\left(\mathrm{Re}(\boldsymbol{\mathcal{H}}_{i_1,\cdots,i_p})^2 + \mathrm{Im}(\boldsymbol{\mathcal{H}}_{i_1,\cdots,i_p})^2\right)$ i.e. both the real and the imaginary parts are equal to zero. This function is the equivalent of the vectorial zero semi-norm for tensor object. When using this semi-norm, the solution of (4.6) is given by the hard-threshold operator $\mathcal{S}_\alpha$

$$\boldsymbol{\mathcal{H}}^* = \mathcal{S}_\alpha\Big(\boldsymbol{\mathcal{Y}} \bigtimes_{m=1}^{M} \boldsymbol{X}_m^{-1}\Big) \,. \tag{4.8}$$

Although this sparsity constraint is very simple to implement, it does not allow us to control in which dimension the sparsity occurs. In particular, this behavior is not adapted to the bandlimitedness assumption.

2. **Controlled Sparsity (CS) constraint:**

$$\Omega : (\boldsymbol{\mathcal{H}}, (\alpha_m)_{m=1}^M) \longmapsto \sum_{m=1}^{M} \alpha_m \|\boldsymbol{\mathcal{H}}^{(m)}\|_{2,0} \,. \tag{4.9}$$

This function imposes zeros on the rows of the unfolding $\boldsymbol{\mathcal{H}}$ which make it more adapted for the bandlimitedness assumption (see Figure 4.2). Considering each norm/mode independently, the solution of the subproblem is obtained by sorting the rows of $\left(\boldsymbol{\mathcal{Y}} \times_{m=1}^{M} \boldsymbol{X}_m^{-1}\right)^{(m)}$ by their ℓ_2-norm and then selecting the rows with norms lower than α_m (row/column-wise hard thresholding) [Baraniuk et al., 2010]. The complexity of this sorting process is $\mathcal{O}(\prod_{k=1}^{M} N_k + N_m \log(N_m))$. Following this observation, we propose the following optimization problem and the algorithm 4.1 to solve it

$$\min_{\boldsymbol{\mathcal{H}}} \quad \|\boldsymbol{\mathcal{Y}} \bigtimes_{m=1}^{M} \boldsymbol{X}_m^{-1} - \boldsymbol{\mathcal{H}}\|_F^2 \tag{4.10}$$

$$\text{s.t.} \qquad \Big(\|\boldsymbol{\mathcal{H}}^{(m)}\|_{2,0} \leq K_m\Big)_{m=1}^{M} \,, \tag{4.11}$$

where each $K_m \in \mathbb{R}$ control the sparsity of the m-th dimension. Contrary to the previous constraint, this one is adapted to the bandlimited property. Indeed, thanks to the parameters K_m it is possible to impose different sparsity constraints for the three different domains (time, space, feature space).

3.3 Selecting the best nodes and reconstruction

The sparsity in the frequency domain allows to subsample graph signals by selecting few elements from each graph domain. This task is referred to as subsampling. Sampling a subset of nodes from multiple graph $(G_m)_{m=1}^M$ is equivalent to selecting a subset of rows and columns from each associated $\boldsymbol{X}_m$. Fortunately, as the support of the tensor graph signal is now estimated (see previous section), the columns which need to be kept are known and we only need to select the best subset of rows for each $\boldsymbol{X}_m$. When only one graph is considered, several methods exist in order to efficiently find a proper subset. For high-dimensional data, greedy methods (algorithms that select one node at a time) are very useful. Several authors have proved submodularity of different optimality criteria such as D-optimality [Shamaiah et al., 2010], and frame potential

Algorithm 4.1 CS constraint

Input : $\mathcal{Y}$, $(\boldsymbol{L}_m)_m^M$, and $(K_m)_m^M$.
Output : $\mathcal{H}$
for $m = 1, \cdots, M$ **do**
  $\boldsymbol{X}_m \longleftarrow \text{eigen}(\boldsymbol{L}_m)$
end for
$\mathcal{H} \longleftarrow \mathcal{Y} \times_{m=1}^{M} \boldsymbol{X}_m^{-1}$
for $m = 1, \cdots, M$ **do**
  for $j = 1, \cdots, N_{m^-}$ **do**
    $\boldsymbol{y}_j \longleftarrow \|\mathcal{H}_{:,j}^{(m)}\|_F^2$
  end for
  $s \longleftarrow \text{argsort}(\boldsymbol{y})[0 : K_m]$
  $\mathcal{H}^{(m)}[s] \longleftarrow 0$
end for

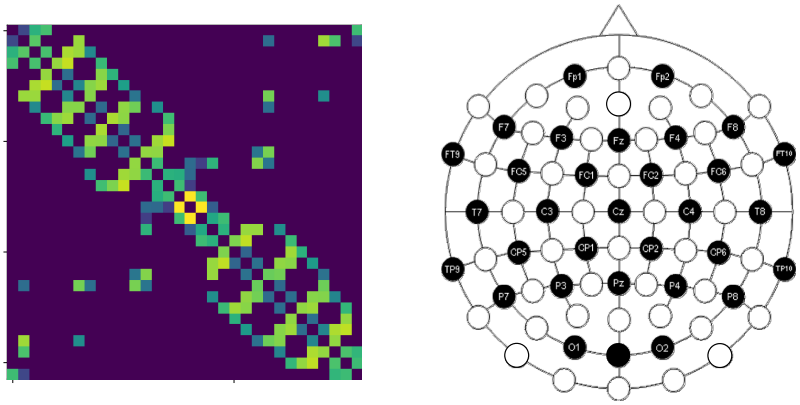


Figure 4.3: Weight matrix of the graph G_S and template 2-D layouts of the sensors.

[Ranieri et al., 2014]. We can also follows ideas of Ortiz-Jiménez et al. [2018, 2019] which proposed low-complexity greedy algorithms based on submodular functions to sample signals that reside on the vertices of a product graph.

4 Results

In this section, we test our different strategies on real EEG data.

4.1 Data

Dataset. The dataset consists of $S = 32$ EEG signals collected at 250 Hz during a general anesthesia with electrodes attached on the brain of a patient. For each EEG signal, we compute the spectrogram through Short-Time Fourier-Transform with time-windows of 256 samples and with 50% overlap. Then, we compute the energies in $F = 12$ frequency bands equally spaced between 0.1 Hz and 12 Hz (in order to retrieve the delta, theta and alpha waves that are relevant for anesthesia [Brown et al., 2010; Purdon et al., 2013]). The final tensor graph signal $\mathcal{Y}$ is in $\mathbb{R}^{F \times T \times S} = \mathbb{R}^{12 \times 233 \times 32}$.

Graph construction. As explained in the previous section, we construct three graphs, G_F, G_T, G_S, respectively as a weighted correlation graph, a cycle graph, and a spatially weighted graph. The graph G_S related to the spatial component is constructed using the spatial position of each channels in 3-D space; low weights are removed (see Figure 4.3).

4.2 Subsampling and reconstruction

The different techniques described in this work are tested on the EEG data. Then, for a given percentage of removed nodes, we reconstruct the data and compute the Root Mean Square Error (RMSE). Results with four different sparsity constraints are displayed in solid lines in Figure 4.4:

- General Sparsity (GS) with a total of 74560 nodes. The conserved nodes can appear in any time/space/feature space positions.
- Controlled Sparsity (CS) on the feature space dimension F with a total of 12 nodes. Only a few energy signals are kept and other are reconstructed by using the correlations between the modalities.
- Controlled Sparsity (CS) on the time dimension T with a total of 233 nodes. Only a few time samples are kept for the reconstruction: this task is linked with signal interpolation and to the classical definition of signal subsampling.
- Controlled Sparsity (CS) on the spatial dimension S with a total of 32 nodes. Only a few EEG sensors are used to reconstruct others, based on their spatial interactions.

The performances of the GS constraint configuration are very satisfactory since it is possible to reconstruct the whole data set with a 0.02 RMSE by removing up to 80% of the nodes. This means that the data is actually very sparse in the frequency domain and that the information can be well represented in sparse domains. However, the main drawback of this approach is that, since the selected nodes can appear in any domain (time, space, feature space), subsampling may be difficult to implement.

The results obtained with Controlled Sparsity (CS) constraints are very contrasted. To obtain a 0.04 RMSE, it is equivalent to remove 30% of the time samples or 60% of the sensors or of the frequency bands. It therefore appears that the graph structure is especially relevant in the subsampling process for this two last dimensions. For the F and S dimensions, results appear similar up to $\sim$ 10% of removed nodes but differ for larger percentages. Indeed, removing more sensors seems to have a slightly stronger effect than removing more modalities (before 80%). This is probably due to the fact that the main phenomena occurring during anesthesia appear in the alpha band between 8 Hz and 12 Hz which spans several of the 10 considered frequency bands. Therefore, a strong correlation exists between modalities that enables a fairly good reconstruction. For the T dimension, the RMSE increases linearly with the number of removed nodes, which is probably due to the relatively weak interactions modeled in the G_T graph. Although the performances of the Controlled Sparsity configuration appear to be worse than the General Sparsity, it is interesting to notice that for this configuration, the subsampling experiment can directly be used to select sensors, lower the sampling frequency or to choose the relevant frequency band to monitor during anesthesia.

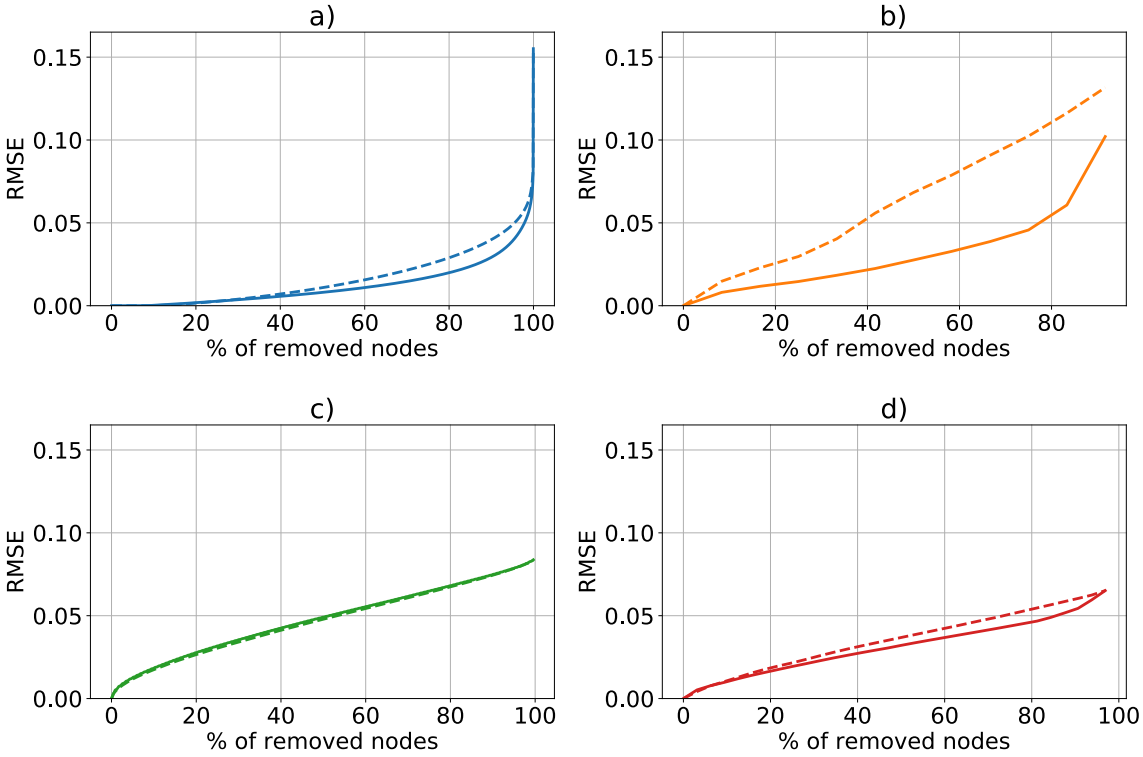


Figure 4.4: Evolution of the RMSE with the percentage of removed dimensions for a) General Sparsity (GS) or Controlled Sparsity (CS) on the b) Feature space F c) Time T d) Spatial S dimensions. The dotted plots correspond to configurations where all graphs have been replaces by random ER graphs.

4.3 Importance of the graph structure

Intuitively, the structure of the graphs used for sampling and reconstruction is crucial. To prove this point, we propose in this experiment to replace one or all of the graphs G_F, G_T, G_S with a random Erdős–Rényi (ER) graph. For the General Sparsity (GS) constraint, all graphs are random and for the Controlled Sparsity (CS) constraint only the graph of interest is random. The resulting reconstruction performances are displayed in dotted lines on Figure 4.4.

For the General Sparsity constraint, the performances decrease with the use of random graphs: for 80% of removed nodes, the RSME is now 0.03 instead of 0.02. As far as the Controlled Sparsity constraints are concerned, and as seen in the previous subsection, the graph structure is especially important for the F and S dimensions. In particular, when considering the spatial dimension, the RMSE is significantly larger with the random graph, which shows that the proposed spatial modeling is here useful for the sampling/reconstruction process. Interestingly, although the directed cycle graph has been a very common model for dealing with the temporal aspects of time-vertex signals [Loukas and Foucard, 2016; Grassi et al., 2018], it here appears that this graph does not bring the necessary structure for the sampling task: results are here similar when this graph is replaced by a random graph. Instead of the simple Markov formulation, a more structured graph (learned via bandlimited signals for example) could probably better model the relationships between time samples.

5 Conclusion

In this chapter, we imposed with three different graphs relationships between the three dimensions, time, space, and feature space of multivariate time-vertex graph signals. To be able to sample these signals, we provided an efficient algorithm which identify their graph frequency support. In addition, we introduced a way to assess the relevance of the graphs chosen a priori by comparing our results with those obtained when random graphs are taken into account. The results showed the importance of the graphs in this algorithm and support for the relevance of the controlled sparsity constraints to recover multivariate bandlimited signals.

5

Apprenticeship learning for a predictive state representation of anesthesia

Abstract

In this chapter, we present a decision support algorithm which assists anesthesiologists in administering anesthetics in order to maintain an optimal DoA. (DoA). Derived from a Transform Predictive State Representation algorithm, our model learns by observing anesthesiologists in practice. This framework, known as apprenticeship learning, is particularly useful in the medical field as it is not based on an exploratory process – a prohibited behavior in healthcare. The model only relies on four commonly monitored variables: Heart Rate, Mean Blood Pressure, Respiratory Rate, and concentration of anesthetic drug. The performances of the model is analyzed with metrics derived from the Hamming distance and cross entropy. They demonstrate that low rank dynamical system had the best performances on both predictions and simulations. Then, a confrontation of our agent to a panel of six real anesthesiologists demonstrate that 95.7 % of the actions are valid. These results strongly support the hypothesis that TPSR based models convincingly embed the behavior of anesthesiologists including only four variables that are commonly assessed to predict the DoA. The proposed approach could be of great help for clinicians by improving the fine tuning of the DoA. Furthermore, the possibility to predict the evolutions of the variables would help preventing side effects such as low blood pressure. A tool that could autonomously help the anesthesiologist would thus improve safety-level in the surgical room.

Contents

The material of this chapter is based on the following publication:

P. Humbert, C. Dubost, J. Audiffren, and L. Oudre. Apprenticeship Learning for a Predictive State Representation of Anesthesia. In *IEEE Transactions on Biomedical Engineering (TBME)* .

1 Introduction

In the early 2010's, the 4th National Audit Project (NAP4) estimated that 2.9 million General Anesthesia (GA) were performed annually in the UK [Woodall and Cook, 2010]. As this practice carries risks (cardiovascular complication [Golubovic et al., 2018], cognitive dysfunction [Punjasawadwong et al., 2018] and postoperative delirium [Fritz et al., 2016]), a sustained and intense attention of the anesthesiologists is imperative to evaluate the level of consciousness of the patient, also referred to as the Depth of Anesthesia (DoA). However, its precise estimation remains an open problem and a constant monitoring of many physiological variables such as heart rate or blood pressure is needed to prevent complications. Since this large amount of information is intractable for the human brain, modern monitors provide multiple auditory and visual warnings, to inform and alert anesthesiologists when physiological variables begin to deteriorate. Unfortunately, those additional indications, while originally meant to help, tend to cause information overload [Stevenson et al., 2013], and often fail to be fully processed. Moreover, due to the global problematic of cost efficiency and human resource limitations, it has become common for anesthesiologists to manage two surgical rooms at the same time [Merry et al., 2010]. In this context, the development of autonomous agents[1] which assist the anesthesiologists managing the delivery of drugs during a GA has become crucial to ease the decision making process, reduce the daily workload and personalize the anesthetic administration, all of this allowing a potentially significant improvement in care.

Several methods have been introduced to fully automate a particular task using closed-loop control models. These methods are used in many fields and cover a wide range of applications [Zhang et al., 1993; Wang et al., 2010; Herrero et al., 2018; Romero-Ugalde et al., 2018]. The automation of the delivery of drugs in anesthesia is one of them [Gentilini et al., 2001; Ionescu et al., 2008; Sawaguchi et al., 2008; Dumont, 2012]. Conventional control techniques have been proposed, such as proportional integral-derivative control [O'hara et al., 1991]. However, these methods perform poorly when applied to processes with variable time delays, nonlinearities, and non-negligible process noise [Tang et al., 2001]. More advanced techniques commonly associated with intelligent systems were studied, including bayesian filtering [Ching et al., 2013], fuzzy control [Moore et al., 2009], and reinforcement learning algorithms as markov decision processes [Borera et al., 2012; Moore et al., 2014]. The latter are receiving significant interest in the medical community [Prasad et al., 2017; Moore et al., 2011] as they provide efficient models and strong training patterns for autonomous agent that are mathematically sound and have already proven

[1] In the chapter we define agents as Apprenticeship Learning based models.

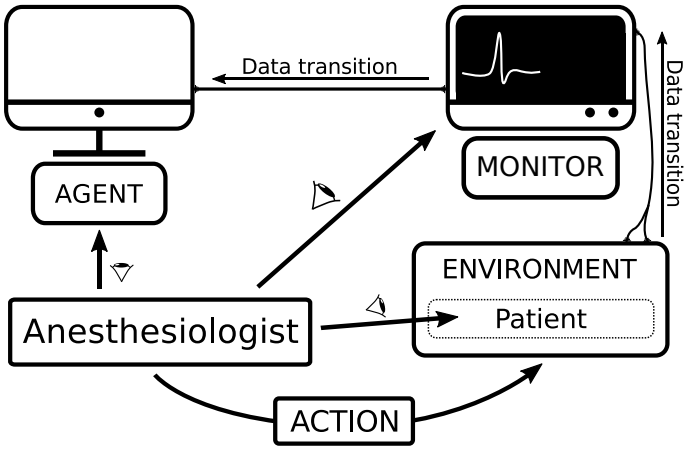


Figure 5.1: Diagram of the agent and the specific environment. The environment (i.e. the patient) provides observable data (i.e. physiological variables). The monitor records this data and transmits it to the agent. The anesthesiologist chooses an action based on the action suggested by the agent, the values given by the monitor and the behavior of the patient.

their usefulness in other areas (e.g. robotic programming [Kaelbling et al., 1996; Kober et al., 2013]). However, the definition of a proper and accurate reward function – a mandatory part of reinforcement learning methods – is nearly intractable for complex problems [Kuderer et al., 2015]. Moreover, while the free exploration of the policies space is a key part of the learning process in reinforcement learning algorithms, this is a prohibitive behavior in healthcare. We refereed to [Yu et al., 2019] for a complete survey on reinforcement learning in healthcare.

The use of apprenticeship learning (also called learning by watching, imitation learning, learning from demonstration)[2] [Abbeel and Ng, 2004] permits to overcome these drawbacks as the learning process in this framework only need observations of experts without the need for exploration. Moreover, models derived from Predictive State Representations (PSRs) [Littman and Sutton, 2002], such as Transformed PSRs (TPSRs) [Rosencrantz et al., 2004], rely entirely on observable quantities – an especially desirable property when the underlying latent state (in this case, *consciousness*) is complex and poorly understandood. Based on spectral learning algorithms, TPSR increases the compactness of the space of relevant states. From a mathematical perspective, many theoretical results demonstrate the rich expressiveness of these models. For instance, [Littman and Sutton, 2002] – influenced by [Rivest and Schapire, 1994] – showed that PSRs are as flexible and powerful as partially observable markov decision process while providing much more compact representations.

In this study, we introduce a novel decision support tool that predicts in real-time whether anesthesiologists should *reduce the drug dose, do nothing or increase the drug dose* given previous sequences of actions and observations (see Figure 5.1 for an illustration). To this end, we combine Apprenticeship Learning principles and TPSR model to solve major problems of control techniques. The resulting approach presents significant advantages, including the fact that the model learns *"how anesthesiologists do"*, instead of trying to learn a complex model of consciousness and deducing *"how anesthesia should work"*. Another major contribution is that our model only relies on a high-resolution recording of the Heart Rate (HR), the Mean Blood Pressure (MBP), the

[2] A slight difference is now made between apprenticeship learning and imitation learning in the literature.

Respiratory Rate (RR) and the concentration of anesthetic drug (AAFi). These four variables are constantly influenced by the drug and are mandatory monitored, making the resulting model suitable for daily use. We also introduce a simple algorithm to homogenize the acquired physiological data and decrease the intra-patient variability. Indeed, the patient's age and gender, as well as disease and surgical intervention are known to affect response to anesthetics [Schnider et al., 1998]. Finally, models were evaluated 1) quantitatively with metrics derived from the Hamming distance and cross entropy 2) with a confrontation to six real anesthesiologists on three cases. This confrontation provides additional metrics to fully evaluate our model and is a mandatory prerequisite for medical application.

This work is organized as follows. We recall the PSR model and its learning process in Section 2. Then, we introduce our main contribution, the construction of a TPSR-based autonomous agent to assist the anesthesiologists managing the delivery of drugs during a GA. (Section 3). We also define and discuss our methodology and preprocessing choices. In Section 3.3 we assess the performance of the model with respect to multiple different metrics (Section 3.4) and with three evaluations done by a panel of experts in anesthesiology (Section 3.5). Finally, the performances, advantages and drawbacks of our approach are discussed in the last section (Section 5).

2 Predictive state representation

From the angle embraced in this work, we consider a GA as a discrete-time dynamical system where at each time step, the environment (i.e. the patient) generates observable data (i.e. physiological variables) from a set $\mathcal{O}$. Recorded by a medical device, these data are transmitted to the agent which takes an action from a set of possible actions $\mathcal{A} = \{0, 1, 2\} = \{$*reduce the drug dose, do nothing, increase the drug dose*$\}$. Finally, the environment moves to an (unknown) hidden state and produces new observations.

In the present work, we used PSR based models to learn this system. The algorithm of PSR was first introduced by [Littman and Sutton, 2002]. The authors showed the advantages of this model over Markovian approaches and discussed the improvement brought by possible non-linear models. Following this idea, [Singh et al., 2003; Rudary and Singh, 2004] have focused on improving the learning process of the PSR models. The algorithm used in this chapter, called Transformed Predictive State Representations (TPSR), was introduced in [Rosencrantz et al., 2004], where the authors presented the multiple advantages over PSRs, namely removing the problems of local minima in the associated minimization problem and producing a more compact representation. This mathematical model is described below; we refer to [Rivest and Schapire, 1994; Littman and Sutton, 2002; Rosencrantz et al., 2004; Boots et al., 2011] for an in depth presentation.

2.1 Background on PSR and TPSR

A linear PSR can be seen as a complete description of a dynamical system. Formally, it consists of two infinite countable sets $\overline{\mathcal{H}}$ and $\overline{\mathcal{T}}$ and a system-dynamics matrix $\mathcal{D}$ defined as follows:

- The elements of $\overline{\mathcal{H}}$ (resp. $\overline{\mathcal{T}}$), called *histories* (resp. *tests*) and referring to the past (resp. the future), are defined by

$$\overline{\mathcal{H}} := \left\{ h \in (\mathcal{A} \times \mathcal{O})^k \mid k \in \mathbb{N} \right\},$$

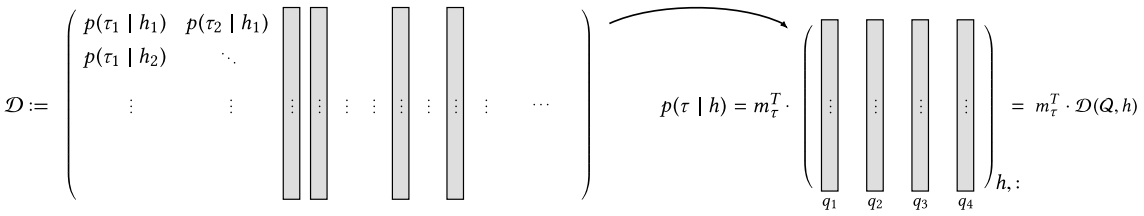


Figure 5.2: Illustration of the PSR framework. On the left the system-dynamic matrix $\mathcal{D}$. The gray columns involved in the construction of the matrix on the right are core tests.

$$\overline{\mathcal{T}} := \left\{ \tau \in (\mathcal{A} \times \mathcal{O})^\ell \mid \ell \in \mathbb{N}_* \right\} .$$

In other words, they consist in an ordered *sequences* of action-observations pairs $(a, \boldsymbol{o}) \in \mathcal{A} \times \mathcal{O}$, denoted by $h = a_1 \boldsymbol{o}_1 a_2 \boldsymbol{o}_2 \cdots a_k \boldsymbol{o}_k$ (resp. $\tau = a^1 \boldsymbol{o}^1 a^2 \boldsymbol{o}^2 \cdots a^\ell \boldsymbol{o}^\ell$).

- The *system-dynamics matrix* $\mathcal{D}$, containing an infinite number of columns and rows, has its elements equal to

$$\mathcal{D}(\tau_i, h_j) = \mathcal{D}_{j,i} := p(\tau_i \mid h_j) = \frac{p(h_j, \tau_i)}{p(h_j)} , \tag{5.1}$$

where p denotes the probability associated with the law of the dynamical system for all pairs (τ, h) in $(\overline{\mathcal{T}} \times \overline{\mathcal{H}})$ – in other words, $p(\tau_i \mid h_j)$ denotes the probability of observing τ_i in the future given that h_j was observed in the immediate past. If $p(h_j) = 0$ we set $p(\tau_i \mid h_j) = 0$. The rank of $\mathcal{D}$ characterizes the complexity of the system and is commonly referred to as its *linear dimension.*

- Any family $\mathcal{Q} := \{q_1, \cdots, q_k\}$, $k \in \mathbb{N}$, of linearly independent columns of $\mathcal{D}$ is called a sufficient set of core tests (core set for short) if $|\mathcal{Q}| = \text{rank}(\mathcal{D})$ ($|\cdot|$ denotes the cardinality of a set). The elements of the core set form a base of the vector space spawned by the columns of $\mathcal{D}$. Therefore, for any $\tau \in \overline{\mathcal{T}}$, there exists an unique weight vector $\boldsymbol{m_\tau}$ such that for all h

$$\mathcal{D}(\tau, h) = p(\tau \mid h) = \boldsymbol{m_\tau}^T p(\mathcal{Q} \mid h) . \tag{5.2}$$

In this equation, $p(\mathcal{Q} \mid h)$ is called the *belief vector* and is defined as

$$\begin{cases} p(\mathcal{Q} \mid h) := \big(p(q_1 \mid h), \ldots, p(q_{|\mathcal{Q}|} \mid h)\big)^T & \text{if } \ h \neq \emptyset , \\ p(\mathcal{Q} \mid \emptyset) := \boldsymbol{m_0}^T & \text{otherwise} , \end{cases} \tag{5.3}$$

with $\boldsymbol{m_0}$ denoting the (unknown) initial condition of the system and $\emptyset$ being the empty history. Similarly, we define $\mathcal{D}(\mathcal{Q})$ as the submatrix of $\mathcal{D}$ that contains the columns relative to the core set i.e. $[\mathcal{D}(\mathcal{Q}, h)^T]_i = [p(\mathcal{Q} \mid h)^T]_i = p(q_i \mid h)$ (see Figure 5.2).

Discovery problem Finding a core set is called the *discovery problem.* This is important as for any such $\mathcal{Q}$, the knowledge of $\mathcal{D}(\mathcal{Q})$ – as well as the initial distribution $\boldsymbol{m_0}$– is enough to fully describe the dynamical system [Singh et al., 2004].

Basically, there are two main approaches to solve this problem and learn PSRs [Hamilton et al., 2014]. The first one is a discovery-based technique (see e.g. [Wolfe et al., 2005; James and Singh, 2004; James et al., 2005]) leading to an explicit knowledge of $\mathcal{Q}$. The second one is a subspace-based technique which is used here and referred to as Transformed PSRs (TPSRs). The latter uses spectral methods to find a subspace isomorph to the vector space generated by $\mathcal{Q}$ instead of determining $\mathcal{Q}$ exactly. To use TPSR model, we applied the spectral algorithm introduced by Boots et al. [2011] which learns several matrices (namely $B_{a\boldsymbol{o}}$, $\boldsymbol{b_\infty}$ and $\boldsymbol{b_*}$, defined below) from sequences of action-observation pairs. This algorithm provides compact and accurate models and permits to predict the most likely future sequences of actions and states efficiently.

We now recall the matrices involved in this algorithm. For $\mathcal{H} \subset \overline{\mathcal{H}}$ and $\mathcal{T} \subset \overline{\mathcal{T}}$, two finite subsets, let define

- $P_\mathcal{H} \in \mathbb{R}^{|\mathcal{H}|}$ that contains the probability of every event in $\mathcal{H}$ i.e. $P_\mathcal{H}(h_j) = [P_\mathcal{H}]_j := p(h_j)$.

- $P_{\mathcal{T},\mathcal{H}} \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{H}|}$ where entry (i,j) is the joint probability of (h_j, τ_i) i.e. $P_{\mathcal{T},\mathcal{H}}(\tau_i, h_j) = [P_{\mathcal{T},\mathcal{H}}]_{i,j} := p(h_j, \tau_i)$.

- $P_{\mathcal{T},a\boldsymbol{o},\mathcal{H}} \in \mathbb{R}^{|\mathcal{T}|\times|\mathcal{H}|}$ (one matrix for each unique pair $a\boldsymbol{o}$) where entry (i,j) of $P_{\mathcal{T},a\boldsymbol{o},\mathcal{H}}$ is the probability of the history h_j, the next action-observation pair $a\boldsymbol{o}$, and the subsequent test τ_i i.e. $P_{\mathcal{T},a\boldsymbol{o},\mathcal{H}}(\tau_i, h_j) = [P_{\mathcal{T},a\boldsymbol{o},\mathcal{H}}]_{i,j} := p(h_j, a\boldsymbol{o}, \tau_i)$.

Let $k \in \mathbb{N}$ and $a_1 o_1 \dots a_k o_k \in (\mathcal{A} \times \mathcal{O})^k$. For any $t \leq k$, let $h_t = a_1 o_1 \dots a_t o_t$ and $\boldsymbol{b_t} = p(\mathcal{Q}|h_t)$ the associated belief vector. Thus, the belief vector at time $(t+1)$ can be expressed as $\boldsymbol{b_{t+1}} = p(\mathcal{Q} \mid h_t a\boldsymbol{o}_t)$. The equation binding $\boldsymbol{b_t}$ and $\boldsymbol{b_{t+1}}$ is called the *update rule* and is given by

$$\boldsymbol{b_{t+1}} = \frac{B_{a\boldsymbol{o}_t}\boldsymbol{b_t}}{\boldsymbol{b}_\infty^T B_{a\boldsymbol{o}_t}\boldsymbol{b_t}} \,, \tag{5.4}$$

where

$$\begin{cases} B_{a\boldsymbol{o}_t} = U^T P_{\mathcal{T},a\boldsymbol{o}_t,\mathcal{H}}(U P_{\mathcal{T},\mathcal{H}})^\dagger & \text{is a transition matrix,} \\ \boldsymbol{b}_\infty^T = P_\mathcal{H}^T (U^T P_{\mathcal{T},\mathcal{H}})^\dagger & \text{is a normalizer } (\forall h, \boldsymbol{b}_\infty^T p(\mathcal{Q} \mid h) = 1) \,, \\ \boldsymbol{b_*} = U^T P_{\mathcal{T},\mathcal{H}} \mathbf{1}_{|\mathcal{H}|} & \text{is the initial state.} \end{cases} \tag{5.5}$$

Here, $\mathbf{1}_{|\mathcal{H}|}$ is the ones-vector of length $|\mathcal{H}|$, $\dagger$ denotes the Moore–Penrose pseudo inverse and U contains the left singular vectors of $P_{\mathcal{T},\mathcal{H}}$.

Predictions With the previously defined matrices, for any sequence of u (action, observation) pairs ($u \in \mathbb{N}_*$), we have

$$\begin{cases} p(a_{t+1}\boldsymbol{o}_{t+1} \mid h_t) = \boldsymbol{b}_\infty^T B_{a\boldsymbol{o}_{t+1}}\boldsymbol{b_t} \quad (\text{for } u = 1) \,, \\ p(a_{t+1}\boldsymbol{o}_{t+1}, \cdots, a_{t+u}\boldsymbol{o}_{t+u} \mid h_t) = \boldsymbol{b}_\infty^T B_{a\boldsymbol{o}_{t+u}} \cdots B_{a\boldsymbol{o}_{t+1}}\boldsymbol{b_t} \,. \end{cases} \tag{5.6}$$

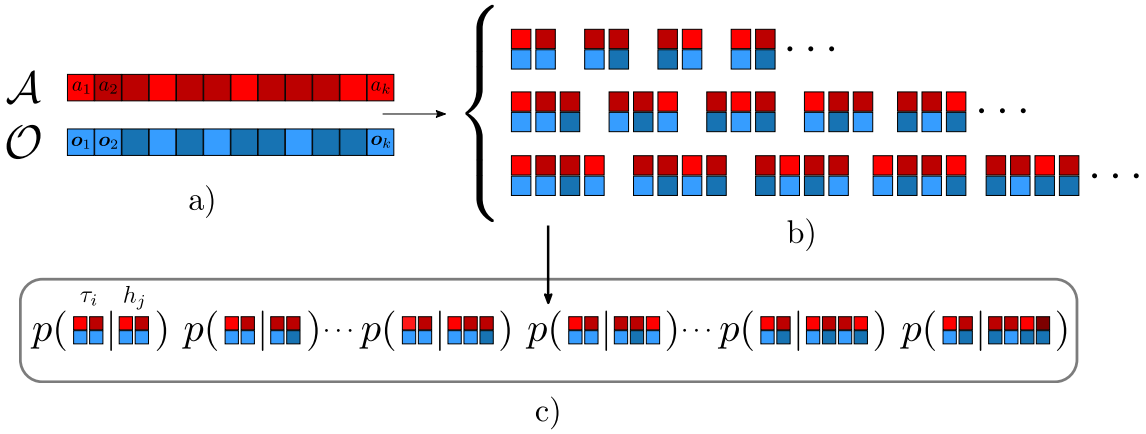


Figure 5.3: a) Sequence of action in $\mathcal{A}$ and sequence of observations in $\mathcal{O}$. Each shade of red (resp. blue) represent a different action (resp. observation). b) Extraction of unique tuples of (actions, observations) of size 2, 3 and 4. c) Example of estimation of the probability of a test of size 2 given all possible histories.

This equation is the key to provide an estimator of the probability $p(\cdot)$.

For further discussion on those equations, we refer the reader to the work of Boots et al. [2011] where theoretical aspects and relation to the matrices of PSRs were discussed. The methodology to predict actions and/or observations in GA is discussed Section 3.

2.2 Methodological choices.

In this subsection, we present our strategy to adapt the TPSR to the problem of closed-loop control of anesthesia. Namely, the introduction of new variables to control the maximum length of each sequence and the use of specific algorithms to compute the different matrices.

Maximal length of a sequence. The computation of the matrices $\overline{\mathcal{T}}$ and $\overline{\mathcal{H}}$ is intractable in practice as they are indexed over an infinite set. To circumvent this problem, we introduced $M_{\mathcal{H}} \in \mathbb{N}_*$ (resp. $M_{\mathcal{T}} \in \mathbb{N}_*$) the maximal length of each history (resp. each test) and restricted ourselves to the learning of $\mathcal{H}_{M_{\mathcal{H}}} := \left\{h \in (\mathcal{A}\times\mathcal{O})^k \mid k \in \mathbb{N}_{\leqslant M_{\mathcal{H}}}\right\}$ and $\mathcal{T}_{M_{\mathcal{T}}} := \left\{\tau \in (\mathcal{A}\times\mathcal{O})^\ell \mid \ell \in \mathbb{N}_{\leqslant M_{\mathcal{T}}}\backslash\{0\}\right\}$. With such a restriction we assumed that $\mathcal{H}_{M_{\mathcal{H}}}$ was sufficient i.e. it allowed to solve the discovery problem – this hypothesis was validated by our experimental results (Section 3.3). In the following, we referred those two sets by $\mathcal{H}$ and $\mathcal{T}$ to simplify the notation. It is worth noting that

$$|\mathcal{H}| \approx ((n_{th}+1)^4|\mathcal{A}|)^{M_{\mathcal{H}}} ,$$

and that the same can be stated for $\mathcal{T}$. Consequently, both sets grow exponentially with $M_{\mathcal{H}}$ and $M_{\mathcal{T}}$. The two numbers $M_{\mathcal{H}}$ and $M_{\mathcal{T}}$ were considered as parameters of the problem.

Learning problem. We computed the estimators $\widehat{P}_{\mathcal{H}}$, $\widehat{P}_{\mathcal{T},\mathcal{H}}$ and $\left(\widehat{P}_{\mathcal{T},a\boldsymbol{o},\mathcal{H}}\right)_{a\boldsymbol{o}}$ of the true TPSR matrices using the entire training set (in other words, all observed combinations were processed). Then, we used a randomized SVD algorithm [Halko et al., 2011] to compute the Singular Value Decomposition (SVD) of $\widehat{P}_{\mathcal{T},\mathcal{H}}$ and obtain its left singular vectors $\widehat{U}$. Algorithm 5.1 summarizes *the learning problem* and an illustration is provided in Figure 5.3.

Algorithm 5.1 Learning problem

Input : M preprocessed trajectories $(\widehat{S_1}, \cdots, \widehat{S_M})$, integers $M_{\mathcal{H}}$, $M_{\mathcal{T}}$, R
Output : $\boldsymbol{b}_{\boldsymbol{\infty}}^T, \boldsymbol{b}_*$ and $\left(B_{a\boldsymbol{o}}\right)_{a\boldsymbol{o}}$
$N \longleftarrow \sum_{m=1}^{M} \sum_{\ell=1}^{M_{\mathcal{H}}} \sum_{k=1}^{|\widehat{S_m}|-\ell} 1$
for $j \in \{1, \cdots, |\mathcal{H}|\}$ **do**
    $\widehat{p}(h_j) \longleftarrow \frac{1}{N} \sum_{m=1}^{M} \sum_{\ell=1}^{M_{\mathcal{H}}} \sum_{k=1}^{|\widehat{S_m}|-\ell} \mathbb{1}_{\{\widehat{S_m}_{(k:k+\ell)}=h_j\}}$
    $[\widehat{P}_{\mathcal{H}}]_j \longleftarrow \widehat{p}(h_j)$
end for
for $(i,j) \in \{1, \cdots, |\mathcal{T}|\} \times \{1, \cdots, |\mathcal{H}|\}$ **do**
    $[\widehat{P}_{\mathcal{T},\mathcal{H}}]_{i,j} \longleftarrow \widehat{p}(h_j, \tau_i)$
    for all $a\boldsymbol{o}$ **do**
        $[\widehat{P}_{\mathcal{T},a\boldsymbol{o},\mathcal{H}}]_{i,j} \longleftarrow \widehat{p}(h_j, a\boldsymbol{o}, \tau_i)$
    end for
end for
$\widehat{U} \longleftarrow$ randomize-SVD$(\widehat{P}_{\mathcal{T},\mathcal{H}}, R)$
$\widehat{\boldsymbol{b}}_{\boldsymbol{\infty}}^T \longleftarrow \widehat{P}_{\mathcal{H}}^T (\widehat{U}^T \widehat{P}_{\mathcal{T},\mathcal{H}})^\dagger$
$\widehat{\boldsymbol{b}}_* \longleftarrow \widehat{U}^T \widehat{P}_{\mathcal{T},\mathcal{H}} \mathbf{1}_{|\mathcal{H}|}$
for all $a\boldsymbol{o}$ **do**
    $\widehat{B}_{a\boldsymbol{o}} \longleftarrow \widehat{U}^T \widehat{P}_{\mathcal{T},a\boldsymbol{o}_t,\mathcal{H}} (\widehat{U} \widehat{P}_{\mathcal{T},\mathcal{H}})^\dagger$
end for

The agent predictions were made using a maximum likelihood approach on the distribution given by equation (5.6).

$$\begin{cases} \arg\max p(a_{t+1}\boldsymbol{o}_{t+1} \mid h_t) \quad (\text{for } u = 1)\ , \\ \arg\max p(a_{t+1}\boldsymbol{o}_{t+1}, \cdots, a_{t+u}\boldsymbol{o}_{t+u} \mid h_t)\ . \end{cases} \tag{5.7}$$

Ties were broken at random.

2.3 Toy example

Here, we give some intuition of the inner working of the TPSR on a simple toy example. The source-code of this example is accessible at https://reine.cmla.ens-cachan.fr/p.humbert/TPSR_implementation. The dataset consisted on a sequence of actions $\overrightarrow{a}$ and a sequence of observations $\overrightarrow{o}$ display in Figure 5.4.

The sequence of action $\overrightarrow{a}$ presents two interesting features. First, the pattern $(0, 1, 2)$ is repeated almost all the way. Moreover 0 are always followed by 1 i.e $p(1 \mid 0) = 1$. On the contrary, 1 are never followed by 0 i.e $p(0 \mid 1) = 0$. Second, there is a "breakpoint" in the repetition of the pattern with six "2''. A visualization of these two sequences is displayed in blue Figure 5.5 b).

To emphasize the importance of the observations sequence $\overrightarrow{o}$, we considered two distinct datasets.

$$\begin{aligned}\overrightarrow{a} &= (\overbrace{0,1,2}^{\text{pattern}},0,1,2,\overbrace{2,2,2,2,2,2}^{\text{“rupture”}},0,1,2,0,1,2,0,1,2,0,1,2)\,,\\ \overrightarrow{o} &= (0,2,1,0,0,0,1,1,1,2,2,2,0,1,2,0,1,1,2,1,2,0,1,2)\,.\end{aligned}$$

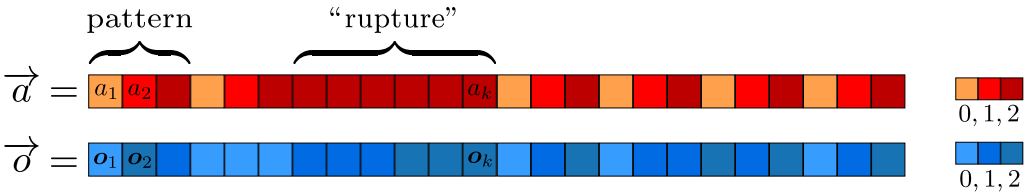


Figure 5.4: Sequence of actions and of observations constituting the dataset. At each color is associated 0, 1 or 2.

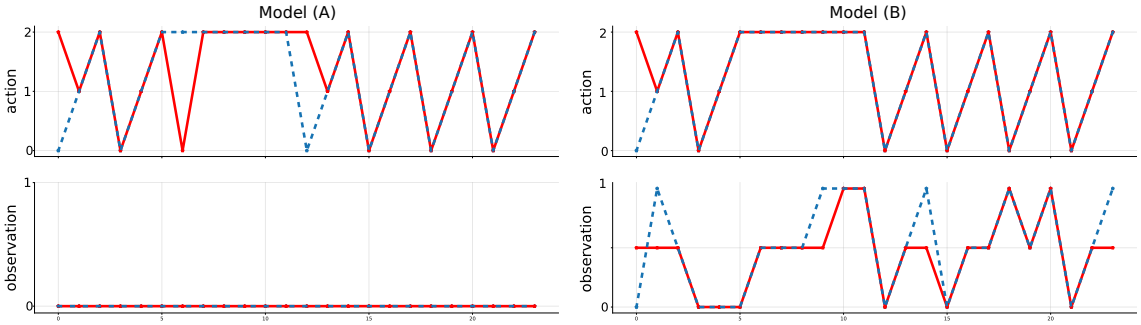


Figure 5.5: For each figure, on the top are represented the actions and on the bottom the observations. Figures on the left: Resulting curves when considering model (A). Figures on the right: Resulting curves when considering model (B). The blue dot curves are the true sequences. The red curves are predicted by the TPSR model.

- model (A) – Dataset was composed of $\overrightarrow{a}$ and a sequence of observations uniquely composed of 0 which does not bring any information (Figure 5.5 Model (A)),

- model (B) – Dataset was composed of $\overrightarrow{a}$ and $\overrightarrow{o}$ (Figure 5.5 Model (B)).

In both cases, we considered history and test with a maximal size of 2 (i.e. $M_{\mathcal{H}} = M_{\mathcal{T}} = 2$) and computed estimators of the different matrices $\widehat{P}_{\mathcal{H}}$, $\widehat{P}_{\mathcal{T},\mathcal{H}}$ and $\left(\widehat{P}_{\mathcal{T},a\boldsymbol{o},\mathcal{H}}\right)_{a\boldsymbol{o}}$ (learning part of the algorithm). Then, the core test was found via an SVD (discovery problem). Finally, at any given time t, the agent provided the most probable pair (action, observation) at time $t+1$ using equation (5.6) and a maximum likelihood approach.

On Figure 5.5, we displayed in red the results of the prediction. For the model (A), we observe that the TPSR learned to predict the pattern $(0,1,2)$, but cannot anticipate the “breakpoint” sequence of “2” – as no information is brought by the observation in this model. On the other hand, in model (B), we see that the TPSR used the observation information to predict the “breakpoint”. Note that since the most present action in the dataset is “2” this is the action predicted at $t = 0$. This underlines the importance of observations for acute prediction of actions.

Sex (F/M)	Age (year)	Weight (kg)	Height (cm)
10/21	60 ± 20	82 ± 14	176 ± 7

Table 5.1: Demographic description of the participants. The values presented are means and standard deviations.

3 Methods

The goal of our model is to maintain the patient under a deep anesthesia state qualified as "surgical anesthesia". The anesthesia usually requires the use of two types of drugs: morphinomimetic in order to control the pain and hypnotic drugs to ensure that the patient remains asleep. In our model we only focused on the administration of the hypnotic agent (which is made continuously under general anesthesia), in this case the gas sevoflurane. This gas is administered to the patient thanks to the endotracheal tube and rapidly reaches the brain. It is the actions to do on the gas administration that we aimed at modeling, among the three possibilities: decrease, do nothing, or increase the gas concentration.

3.1 Dataset

Study participants. The study has been approved by the ethics committee of the French society of anesthesiology (SFAR) under the number IRB 00010254-2016-018. Patients were included from March to May 2017 in a single observational center, the Begin military teaching hospital, Saint-Mandé, France. They were included if they were scheduled for an outgoing surgery for inguinal hernia repair under GA, if they gave their consent to the study and if their comorbidity score was low (classified ASA 1 or 2 [Daabiss, 2011]). They were excluded if they presented complications during the surgery (cardiac arrhythmias, variation of the blood pressure or cardiac frequency more than 20 % compared to the baseline value, or unplanned hospitalization). A summary on the 31 participants is available in Table 5.1.

Anesthesia protocol. The anesthesia protocol was in accordance with the declaration of Helsinki. Four anesthesiologists were included in the study. All the patients were pre-oxygenated via face-mask by 100% oxygen for at least 3 minutes before induction. Sufentanil $0.3\ \mu g/kg$ of ideal-body weight was injected rapidly followed 3 minutes later by $2-4\ mg/kg$ propofol in combination with ketamine $20\ mg$. When required for the surgery, patients were paralyzed following induction with a bolus of $0.17\ mg/kg$ of cisatracurium. After tracheal intubation, patients were ventilated with tidal volume of $6\ mg/kg$ ideal-body weight, $5\ cmH_2O$ Positive end-expiratory Pressure (Peep) and a respiratory rate between 10 and 14 to maintain $EtCO_2$ between 30 and 40 $mmHg$. Anesthesia was maintained with sevoflurane MAC age-adjusted (e.g. 1.0), a volatile anesthetic agent [Patel and L. Goa, 1996]. Dose adjustments were made by the anesthesiologist in charge of the patient depending on clinical variables available. Once asleep, patients received a single bolus of local anesthesia when indicated for the surgery.

Data. During the surgery, patients were continuously monitored with a multiparametric device, the Carescape monitor B850, from General Electrics (GE) HealthcareTM Finland Oy, Helsinki, Finland. Variables were recorded synchronously with a sampling frequency of 1Hz during the anesthesia. We selected 4 standard physiological variables (listed in Table 5.2) providing a dataset

Variables	Units	abbreviation
Basics Module – 1 Hz		
Heart Rate	/min	HR
Mean arterial blood pressure	mmHg	MBP
Gaz Analysis Module – 1 Hz		
Respiratory Rate	/min	RR
AA Inspiratory Concentration	/100 %	AA FI

Table 5.2: Selected variables classified by modules. For each of the variables, sampling frequency, unit and abbreviation are provided.

of 4 trajectories for each patient. The anesthetics drugs influence all the organs and especially the cardiopulmonary system. Therefore, the four variables that we selected are all constantly influenced by the drug [De Hert and Moerman, 2015]. Moreover, they are mandatory monitored, making the resulting model suitable for daily use since no additional sensors are needed. All these variables are in accordance with the recommendation of the American Society of Anesthesiologists. This choice was also motivated by our aim to provide a decision support tool. Additionally, it should be noticed that the dimension of the system-dynamics matrix $\mathcal{D}$ from the TPSR increases exponentially with the number of variables considered. Therefore, the choice of a restricted number of variables reduce the complexity of the learning problem, acting as an additional regularization.

3.2 Preprocessing

To homogenize the data, noise and trend of all trajectories were removed via a Simple Moving Average filter (SMA) with a windows size n of $\{5, 15, 30\}$ seconds and no overlap. The random process underlying each physiological variable was assumed to be locally stationary, as their variations were relatively slow, which justified the use of SMA for small values of n.

Observations. Each observation $\boldsymbol{o} \in \mathcal{O}$ consisted of quadruplets $\big($HR, MBP, RR, AAFi$\big)$ discretized using n_{th} thresholds ($n_{th} \in \mathbb{N}_{>2}$) and taking their values in the set $\{0, 1, \cdots, n_{th}\}$ – where 0 represents low values, and n_{th} high values. The discretization was calculated using Ckmeans, a clustering algorithm based on K-means which has been proven to outperform it in the one-dimensional case [Wang and Song, 2011]. We made an exception for AAFi, which was discretized according to common anesthetic heuristics (i.e. with thresholds between 1% and 3%). The purpose of this calibration procedure was 1) to reduce the inter-patient variability while keeping the intra-patient variability by mapping similar physiological states into the same discretized state– a key part of the problem, as incoherent discretization led to contradictory events, 2) to train a model that automatically adapts to the demographic characteristics of patients (e.g. age, height, weight, BMI). The number of thresholds used in the discretization is a parameter of the model and is evaluated in our experiments. To allow real-time use of the model, preprocessing parameters were estimated during a calibration phase. An example of discretization is displayed Figure 5.6.

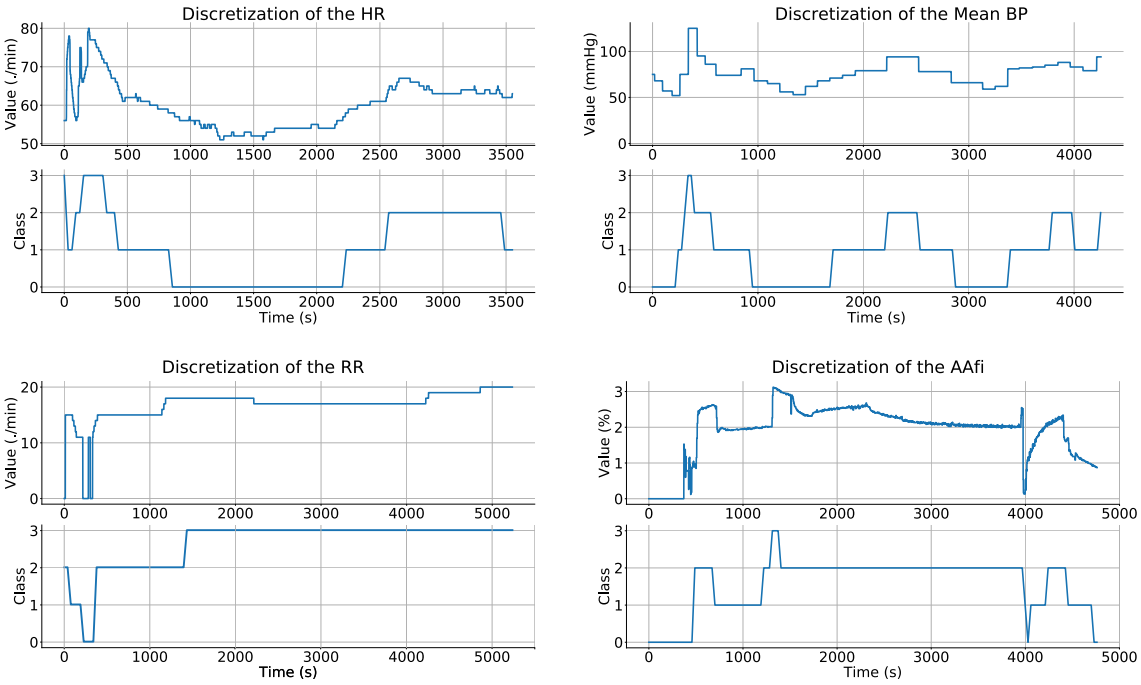


Figure 5.6: Example of a discretization on the four variables, HR, MBP, RR and AAFi with $n_{th} = 3$. For each variable, on the top the raw signal recorded by the monitor during the GA. On the bottom, its discretization in four classes via CKmean.

Names	**Symbols**
Window of the SMA	n
Number of thresholds	n_{th}
Prior on the rank of $\mathcal{D}$	R
Maximal length of a history	$M_{\mathcal{H}}$
Maximal length of a test	$M_{\mathcal{T}}$

Table 5.3: Name and symbol of each adjustable parameters of the model.

Actions. The actions were derived from the AAFi variable which represents the amount of drug administrated to a patient. The considered set of possible actions was $\mathcal{A} = \{0, 1, 2\} = \{$*Reduce drug dose, Do nothing, Increase drug dose*$\}$ formally defined by

- action 0 (Reduce drug dose) – Significant decrease of the AAFi (by at least 10%),
- action 1 (Do nothing) – No significant increase or decrease of the AAFi,
- action 2 (Increase drug dose) – Significant increase of the AAFi (by at least 10%).

More precisely, actions are labeled as follows. Between two regularly spaced sampling points (distant by e.g. 30 s), the action is labeled 2 (resp 0) if the AAFi has increased by at least 10% (resp decreasing by at least 10%). Otherwise, the action is labelled 1. The data pipeline we used for our model is illustrated in Figure 5.7.

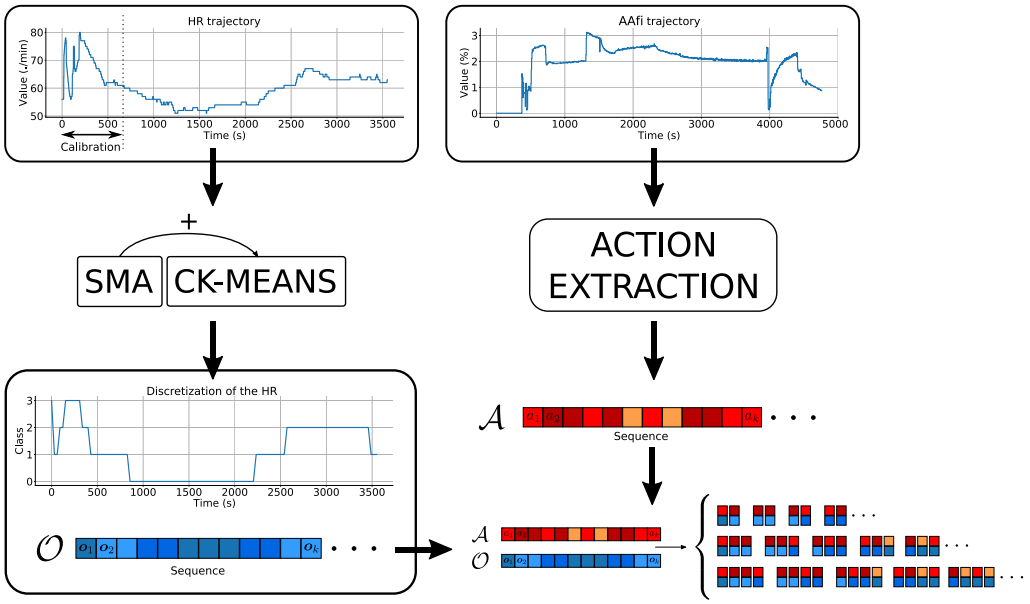


Figure 5.7: On the left: Preprocessing and discretization procedure for the HR variable. The raw signal (trajectory) is filtered and discretized via the combination of the SMA and CKmean to obtain a sequence of observations. On the right: Extraction of the action from the AAFi variable. The raw signal (trajectory) is filtered and actions are extracted to obtain a sequence of actions. Then, a sequence actions/observations is made in order to fit in the TPSR framework.

3.3 Evaluation process

We now present the different experiments made to evaluate the performances of our model. First, we conducted an extensive analysis of the different parameters and their respective influence to identify the best set of parameters, using cross-validation and multiple metrics (see Section 3.4). Second, we compared the performance of the resulting model with a Spectral Hidden Markov Model (SHMM) [Hsu et al., 2012; Minh et al., 2012], i.e. HMM learned with a spectral algorithm. Finally, our model and its associated agent were confronted to a panel of six anesthesiologists assessing three cases.

3.4 Quantitative analysis setup

Prior to any evaluations, the dataset was randomly split into a Training-set (60%), a Validation-set (20%) and a Test-set (20%). We repeated this procedure five times, and average the results over the five random splits.

Classical metrics. In the first experiment, we evaluated the discrepancy between actions predicted by the agent and actions of the experts. The agent predictions were selected using a maximum likelihood approach on the distribution given by equation (5.6) – ties were broken at random. The metric used in this experiment was the averaged Hamming Distance (HD) between two sequences $(\tau, \widehat{\tau})$ of length μ – a classical metric for PSRs, closely related to the One-Step Prediction Accuracy [Downey et al., 2017] – (Equation 5.8).

$$\mathrm{HD}(\tau, \widehat{\tau}) = \frac{1}{\mu} \sum_{i=1}^{\mu} \mathbf{1}_{\tau[i] \neq \widehat{\tau}[i]} \ . \tag{5.8}$$

	TPSR			SHMM		
Metrics	$n = 30$	$n = 15$	$n = 5$	$n = 30$	$n = 15$	$n = 5$
HD-A	**0.416**($\pm$0.022)	0.438($\pm$0.042)	0.458($\pm$0.016)	0.592($\pm$0.064)	0.542($\pm$0.122)	0.599($\pm$0.057)
CE-A$_{0,2}$	**1.087**($\pm$0.129)	1.407($\pm$0.141)	1.595($\pm$0.054)	–	–	–
SCE-A$_{0,2}$	**0.628**($\pm$0.056)	0.782($\pm$0.153)	1.108($\pm$0.095)	–	–	–
HD-O						
HR	**0.390**($\pm$0.032)	0.445($\pm$0.070)	0.479($\pm$0.070)	0.759($\pm$0.049)	0.741($\pm$0.057)	0.820($\pm$0.034)
Mean BP	**0.342**($\pm$0.041)	0.345($\pm$0.026)	0.526($\pm$0.082)	0.799($\pm$0.052)	0.735($\pm$0.106)	0.741($\pm$0.081)
RR	**0.199**($\pm$0.008)	0.293($\pm$0.038)	0.389($\pm$0.085)	0.729($\pm$0.085)	0.712($\pm$0.120)	0.732($\pm$0.060)
AAFi	0.197($\pm$0.051)	0.174($\pm$0.034)	**0.156**($\pm$0.041)	0.688($\pm$0.089)	0.787($\pm$0.093)	0.797($\pm$0.040)
Mean HD-O	**0.271**($\pm$0.024)	0.284($\pm$0.031)	0.377($\pm$0.054)	0.717($\pm$0.021)	0.757($\pm$0.035)	0.734($\pm$0.016)

Table 5.4: Results of the quantitative analysis with respect to the n parameter – all the other parameters were optimized with cross validation. For every metrics, the best values were the smallest ones. Metrics reported are the Hamming Distance of Action (HD-A) and Observation (HD-O), the Mean HD-O, the Cross Entropy of Action 0 and 2 (CE-A$_{0,2}$) and the Sliding Cross Entropy of Action 0, 2 (SCE-A$_{0,2}$). On the left, results for our TPSR model, on the right, results for the SHMM. For more details on the metrics, see Subsection 3.4

We also computed the distance of actions or observations sequences separately. Let $\tau|_a$ be the sequence of actions provided by the dataset and $\widehat{\tau}|_a$ the one found with the algorithm e.g. $\tau|_a = (a^1 a^2 a^3 a^4 a^5 a^6 a^7 a^8 a^9) = (1, 1, 1, 2, 1, 1, 1, 0, 1)$. We defined the HD of Actions (HD-A) by

$$\text{HD-A}(\tau, \widehat{\tau}) := \text{HD}(\tau|_a, \widehat{\tau}|_a).$$

The HD of Observations (HD-O) is defined similarly. Finally, we used the cross entropy measure in Action 0 –*Reduce drug dose*– or 2 –*Increase drug dose*– and referred it by CE-A$_{0,2}$. This metric is defined as follows. Suppose that at time t the expert takes the action $i \in \{0, 2\}$, then

$$\text{CE-A}_{0,2}(t, i) = -\log\left(p(\widehat{\tau}|_a(t) = i \in \{0, 2\})\right). \tag{5.9}$$

Metric taking into account a delay. Due to anesthetics latency, the action of an anesthesiologist will only be noticed on the recorded variables after a short time delay. Indeed, the time to reach equilibrium point after a modification of the concentration of sevoflorane is approximately 1 minute (considering a supply of fresh gas of 0.4 L/mn) [Philip et al., 2012]. This phenomenon is not captured by HD-A, HD-0 or CE-A$_{0,2}$. We introduce here a new metric called Sliding Cross Entropy on Action 0 –*Reduce drug dose*– or 2 –*Increase drug dose*– (SCE-A$_{0,2}$) to address this problem. SCE-A$_{0,2}^{(\delta)}$ is defined as follows. Suppose that at time t the expert takes the action $i \in \{0, 2\}$, then let

$$p_{t,i,\delta} = p\Big(\exists s \in [t - \delta, t + \delta] \quad \text{s.t.} \quad \forall s > s' \geq t - \delta\,,$$
$$\widehat{\tau}|_a(s') = 1 \text{ and } \widehat{\tau}|_a(s) = i \in \{0, 2\}\Big)\,.$$

In other words, $p_{t,i,\delta}$ represents the probability of the event where the agent takes the correct action, but with a possible time latency of δ – and that the agent only do neutral action (i.e. action

1) before this. Then SCE-$A_{0,2}$ is simply defined as

$$\text{SCE-A}_{0,2}^{(\delta)}(t,i) = -\log(p_{t,i,\delta}) \ . \tag{5.10}$$

When no delay is considered ($\delta = 0$), SCE-$A_{0,2}$ is CE-$A_{0,2}$. During experiments, the delay was set to 1 minute.

SHMM comparison. In a third step, we compared our TPSR model with the best set of parameters to a tuned SHMM. We used the same metrics as in the previous experiments.

3.5 Real expert evaluation method

The evaluation of reinforcement learning algorithms in healthcare is complex and special care needs to be taken [Gottesman et al., 2018, 2019]. Hence, for an exhaustive and thorough evaluation of our method, we confronted the best model of the quantitative analysis and its corresponding agent with a panel of six anesthesiologists from the anesthesia-intensive care department of the Begin military teaching hospital. This experiment provides additional metrics to fully evaluate a generative model and is a mandatory prerequisite for medical application. The evaluation was conducted as follows. To begin the confrontation, each anesthesiologist was presented with sequences where only the previous actions and the four discretized selected variables were displayed. Then, the three following experiments were conducted and results collected.

- **Experiment** 1 – At each time, and given the real previous sequences, the anesthesiologist chose an action in $\mathcal{A} = \{$*Reduce drug dose, Do nothing, Increase drug dose*$\}$.
  Those actions were recorded and we measured the disagreement rate between the actions taken by the anesthesiologist and the actions predicted by the agent. This experiment quantifies the capacity of the agent to make the right decisions at the right time.

- **Experiment** 2 – At each time, and given the real previous sequences, the agent predicted an action in $\mathcal{A} = \{$*Reduce drug dose, Do nothing, Increase drug dose*$\}$ and the anesthesiologist labeled it as

  - good: the action is the best choice,
  - acceptable: the action is not optimal but still a good choice,
  - dangerous: the action may lead to future complications.

  We measured the frequency of each label. This experiment provides a qualitative evaluation of the actions of the agent, even if they differ from the real anesthesiologist. Indeed, due to anesthetic latency and the nature of our problem, actions that differ from the anesthesiologist might still be valid choices.

- **Experiment** 3 – At each time, and given the previous generated sequences, the anesthesiologist chose an action in $\mathcal{A} = \{$*Reduce drug dose, Do nothing, Increase drug dose*$\}$ and predicted the evolutions of each variables.
  For each variable, we measured the agreement rate between the prediction made by the anesthesiologist to the one made by the agent. This experiment qualitatively evaluate the capacity of our trained model to predict a plausible evolution of the dynamical system given an action.

It should be noted that agreement with human experts in experiment 2 may have been influenced by the lack of a *blind* evaluation. That is why the other two experiments were carefully design to avoid this problem, and their results are in concordance with experiment 2.

4 Results

4.1 Quantitative analysis

Results of the quantitative analysis. We evaluated the ability of each set of parameters to predict the right pairs (action, observations) with the metrics defined in Section 3.4. For each parameter, the following values were compared: $n \in \{5, 15, 30\}$, $n_{th} \in \{3, 4, 5\}$, $M_{\mathcal{H}} \in \{2, 3, 6\}$, $M_{\mathcal{T}} \in \{2, 3, 6\}$ and R was set to $\{50, 100, 300, 400\}$. It is important to note that n played a very crucial role in our model as it significantly modified the data during the preprocessing. Results of the best set of parameters for each value of n are displayed in Table 5.4.

The best result was obtained for $(n = 30, n_{th} = 3, M_{\mathcal{H}} = 6, M_{\mathcal{T}} = 3, R = 400)$ (an example of agent sequence is displayed in Figure 5.8). This model was used for the confrontation with anesthesiologists. It is interesting to note that the agent tended to predict action and observation with a slight time delay. This aspect was emphasized by the evaluation with the SCE-$A_{0,2}$. Furthermore, the curves of Figure 5.8 illustrate that the prediction of physiological variables was accurate and generally differed because of a slight delay.

Contribution of the variables.

- Contribution of AAFi – The AAFi variable is used both as an observation and for the computation of the actions. Hence, the question of whether AAFi influences the model by making the prediction obvious is crucial. To highlight the fact that our model is able to predict the action without simply relying on previous AAFi levels, we conducted additional experiments where AAFi was not included in the model. As a baseline, we also computed the results when no variables were included in the model. These results are presented in Table 5.5. These additional experiments showed that the removal of the AAFi variable in the model only mildly reduced out model performance in term of the **SCE-A**$_{0,2}$ metric: around 0.722 instead of 0.628 for the original model (with AAfi). In comparison, removing all observations (i.e only relying on actions) leads to a **SCE-A**$_{0,2}$ of 0.913. This additional experiment suggests that while AAFi is an important variable for the prediction, it does not trivially contain all the required information. The good results obtained by the agent are therefore not explained by the presence of the AAFi variable in the observations.

- Contribution of RR – The RR is an important variable for monitoring the patient's state. However, in our protocol, the patient is artificially ventilated, i.e. RR is regulated to maintain $EtCO_2$ at a certain level. To study the importance of this variable, we have computed extra results without the RR variable in the model (see Table IV). It turns out that for $n = 30$, **SCE-A**$_{0,2}$ was equal to 0.635 (against 0.628 when RR is in the model and 0.722 when AAFi is not in the model). This shows that the importance of this variable in our model remains limited. However, we believe that the presence of this variable still makes sense in a clinical setting, especially in critical situations. Indeed, under general anesthesia when the depth of anesthesia is appropriate to perform surgery, patients stop breathing spontaneously. The breathing is thus performed artificially by a ventilator, where the RR is set by the anesthesiologist. In such

Metrics	TPSR **with** AAFi $n = 30$	TPSR **without** AAFi $n = 30$	TPSR **without** RR $n = 30$	TPSR **with** no obs. $n = 30$
HD-A	**0.416**($\pm$0.022)	0.439($\pm$0.012)	0.419($\pm$0.001)	0.456($\pm$0.012)
CE-A$_{0,2}$	**1.087**($\pm$0.129)	1.145($\pm$0.071)	1.124($\pm$0.018)	1.161($\pm$0.015)
SCE-A$_{0,2}$	**0.628**($\pm$0.056)	0.722($\pm$0.029)	0.635($\pm$0.021)	0.913($\pm$0.013)

Table 5.5: Additional results of the quantitative analysis. For every metrics, the best values were the smallest ones. Metrics reported are the Hamming Distance of Action (HD-A), the Cross Entropy of Action 0 and 2 (CE-A$_{0,2}$) and the Sliding Cross Entropy of Action 0, 2 (SCE-A$_{0,2}$). For more details on the metrics, see Subsection 3.4

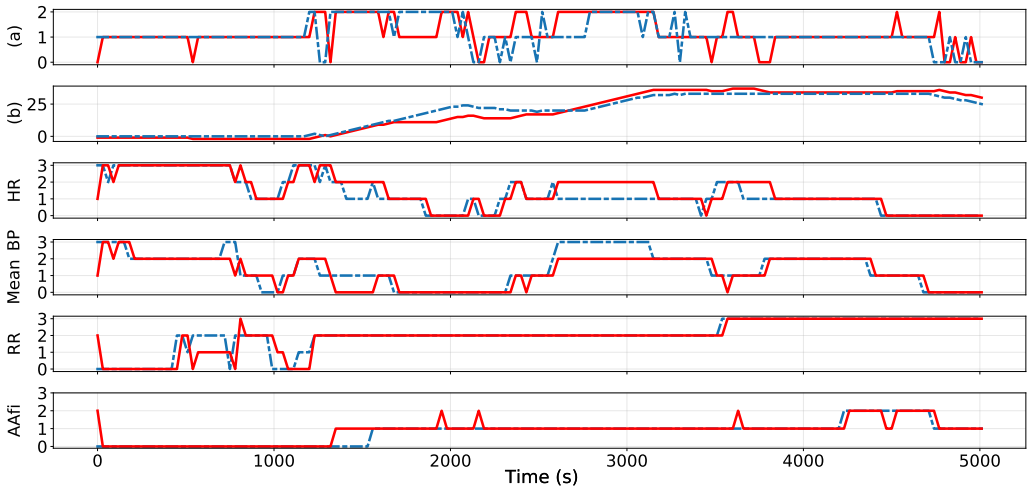


Figure 5.8: Result of the model with the most promising parameters on one patient. At the top, the two graphs show the results of the prediction of actions. (a) – comparison of the real actions (blue dotted line) with those predicted by our agent (red line); (b) – cumulative sum of the real sequence of actions (blue dotted line) and of the predicted (red line). The next four graphs are the results of the prediction of physiological variables. For each graph, in blue dotted line the real sequences and in red line the predictions.

a condition, the stability of the RR represents the good tolerance of the patient towards the mechanical ventilation and becomes an important indicator of under dosage of anesthesia when the variability increases. In our current experiment, the dataset does not contain any critical situations as every surgery have been unremarkable as regard as the anesthesia. Hence, the RR does not significantly contribute to the model performance at this time. However, we anticipate that this is an important indicator of awakening. Thus, RR can be considered an alert variable, which could be used to introduce hard coded behavior in the model: for instance, when it exceeds a certain threshold, the algorithm could send an alarm and exit the closed-loop system, handing the matter back to the anesthesiologist. This is classical approach to closed-loop system.

SHMM. We evaluated the performances of the SHMM for each of our discretization size and for a maximal rank of 400. The results were reported in Table 5.4. We also present in Figure 5.9 the prediction of the best SHMM model on the same patient as Figure 5.8. Figure 5.8 c) is a good

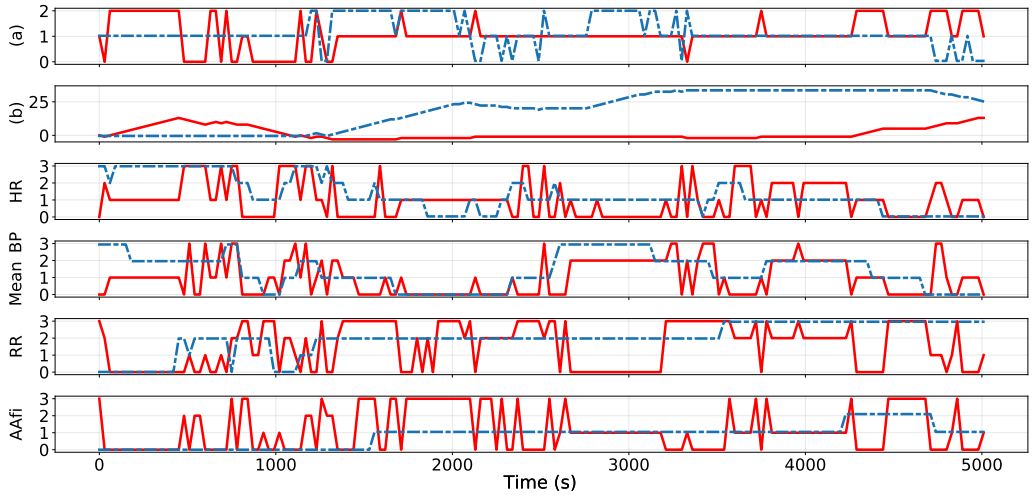


Figure 5.9: Result of the best SHMM model on the same patient of Figure 5.8. At the top, the two graphs show the results of the prediction of actions. (a) – comparison of the real actions (blue dotted line) with those predicted by our agent (red line); (b) – cumulative sum of the real sequence of actions (blue dotted line) and of the predicted (red line). The next four graphs are the results of the prediction of physiological variables. For each graph, in blue dotted line the real sequences and in red line the predictions.

representation of the performance of models – the closer the two curves are, the better the model is. It appears that TPSR significantly outperformed SHMM in all the experiments.

4.2 Real expert evaluation

We asked six consultants anesthesiologists to evaluate our best model. Results from the three experiments introduced in Subsection 3.5 are presented in the Table 5.6. The results were in accordance with those of the paragraph 4.1.

- Experiment 1 showed an accuracy rate close to the one found in the quantitative evaluation section.
- Experiment 2 showed that 95.7 % of the actions were considered valid by the experts. This high rate of concordance was expected due to the long-latency of the anesthetics drugs.
- Experiment 3 demonstrated that the agent can predict the evolution of the variables in the upcoming minutes secondary to any given action.

Exp-1	**Exp-2**		**Exp-3**	
0.371	*Good*	0.632	*HR*	0.914
	Acceptable	0.325	*Mean BP*	0.879
	Dangerous	0.043	*RR*	0.789
			AAFi	0.828

Table 5.6: Evaluation of the best models ($n = 30, n_{th} = 3, M_{\mathcal{H}} = 6, M_{\mathcal{T}} = 3, R = 400$) by a panel of anesthesiologists. Experiment 1: Rate of disagreement between agent and anesthesiologist actions. Experiment 2: rate on actions classify as (good/acceptable/dangerous). Experiment 3: Rate of agreement between agent and anesthesiologists observations. See Section 3.5 for more details on the three experiments.

5 Discussion and future works

Linear dimension. Interestingly, the distribution of the singular values of $\widehat{P}_{\mathcal{T},\mathcal{H}}$ (which is linked to the linear dimension of the TPSR) was found to be similar regardless of the number of included patients. Furthermore, the number of singular values close to zero was significant for several values of horizons, justifying the low rank approximation of the matrix $P_{\mathcal{T},\mathcal{H}}$. Our experiments revealed that models with low rank dynamical system demonstrate strong performances on both predictions and simulations. These results justify the choice of TPSRs over regular PSRs. Moreover, they may have significant consequences in the medical field as the evaluation of DoA through physiological variables could require much less information than presumed i.e. the space of latent states relative to a patient under GA could actually be relatively small.

Influence of parameters. Throughout our experiments, we observed that different values of $M_{\mathcal{H}}$ and $M_{\mathcal{T}}$ yield similar performances. There might be four possible explanations for this phenomenon.

1. The dynamic system does not have a very long memory. This hypothesis is reasonable, as generally, anesthesiologists do not concentrate on a long period of time, partly because of all the simultaneous tasks required.

2. The population included is homogeneous as we only included patients undergoing inguinal hernia repair under GA. No patient in the population had any significant past medical history nor underwent any side-effect during the GA.

3. Values of the horizon parameters that have a significant impact are large, and thus require significantly larger dataset to observe.

4. The discretization process and the values of the parameter n reduce the long time range dependency of the dynamic system.

Future works might try to evaluate each of these hypotheses.

Additionally, the experiments showed that $n_{th} = 3$ is the best choice for this parameter as it achieve the best trade-off between a) the generalization of the discretization which reduces the inter-patient variability and b) the accuracy of the physiological variable trajectories. Recall that the research of the best set of parameters is more indicative on the behavior of the algorithm than

on which parameters need to be actually set for a clinical use. Indeed, the number of patients included is not large enough to properly optimize all the hyperparameters of the models, and current values may change on a larger cohort.

Action and observation prediction. In our experiments, the agent was able to accurately predict the evolution over time of the physiological variables. This performance was expected for discrete AAFi and RR, which exhibit very small variations. However, the small errors on all the observations imply that the agent has learned the complete dynamic system properly. Conversely, predictions were slightly less accurate on actions. This might be explained by the multiplicity of the strategy (policy) exhibited by the experts. Nevertheless, simulations have shown that the actions taken by the agent were validated by the experts. Furthermore, in the first experiments, a significant part of the error was due to small time latency – the agent taking action a few seconds before or after the expert. This behavior was highlighted by the SCE-$A_{0,2}$, a specific metric relevant in our GA scenario. Since those actions would have produced similar results, the good results of the SCE-$A_{0,2}$ demonstrated that the HD metric artificially under estimated the global performance of the agent. The labels of the actions in our model may be seen as relatively inaccurate, since they are restricted to three basics action and that the exact dose of AAFi to be added if necessary is not predicted. Such precision, while theoretically possible by using the continuous extension of the TPSR model [Hefny et al., 2017] would require a significantly larger cohort of patient to be properly calibrated.

SHMM comparison. These experiments highlighted the advantage of our approach over the SHMM. This observation was in line with previous results (see e.g. [Singh et al., 2004; Boots et al., 2013]). One explanation is that, contrary to TPSR, SHMM tends to scale poorly with the complexity of the system to be modeled. However, the implementation of PSRs requires more computational power.

Anesthesiologists feedback. The confrontation with the experts in anesthesiology showed that our agent was coherent and followed an expected policy most of the time. Moreover, all the experts agreed that $n = 30$ appears to be the most realistic value for this parameter. Nevertheless, they also highlighted that there was a latency of the AAFi variable in some situations, particularly when using low flow of fresh gas.

Clinical relevance. The interest of our agent is double: helping at maintaining a patient at the optimal DoA and predict the occurrence of cardiovascular side-effects (with the idea to avoid them). The workload in the surgical theater imposes that an anesthesiologist is often in charge of two surgery rooms plus the post-anesthesia care unit. A tool that could autonomously help the anesthesiologist would thus improve the safety-level in the surgical room. With such a workload, for a low-risk patient undergoing a low-risk surgery, the anesthesiologist in charge may eventually remember a few characteristics of the patient and usually the pre-induction values of HR and MBP. Once anesthesia level is stabilized and surgery has started, it seems reasonable to consider that the anesthesiologist will leave the patient under the nurse-anesthetist care and will only watch the patient every 10-minute. If we consider that the anesthesiologist will remember the pre-induction, post-induction HR, MBP, RR and AAFi, for one patient we end with: 4 values, every 10-minute meaning 24 values every hour to assess the DoA and status of the patient. As opposed, our agent will take into account all the variables available every second. For a low-risk patient with MBP assesses every 5 minutes this will represent 10.820 values every hour.

Limitations. Despite the strong performances of our model during our experimental evaluations, the PSR approach of the GA setting suffer several drawbacks. First, the model is very dependent on the discretization. Indeed, it is a key component that influences the entire learning process as a too fine or too wide discretization leads to an incorrect estimation of the matrices involved in the model. Second, the lack of a preexisting efficient simulator, as well as a gold-standard for the DoA, greatly limit the possibility to improve the performances above what is observed in the expert trajectories.
In its current state, our method is merely a proof of concept for the feasibility of maintaining the anesthesia using carefully trained multimodal algorithm. More experiments and recordings including patients in multiple settings and hospitals will be needed before considering this method as fully valid. It is the authors' belief that the clinical staff will be likely to accept this new approach, as automatic closed-loop anesthesia protocols are already existing, based on the bispectral index [Liu and Rinehart, 2016]. Our method can be seen as an improvement over the exiting protocols, as it takes into account multiple physiological signals as input.

Future works. Beyond the influence of the horizon parameter, we believe that the recording of other relevant physiological variables with additional sensors (e.g. electroencephalogram, muscular sensors, galvanic skin response, ...) could improve the performance of the model. Moreover, a wider range of surgery type in the dataset could bring valuable information on the behavior of the agent. The next step will aim at increasing the population in order to test the generalization of our algorithm in other settings such as in intensive care unit.

6 Conclusion

In the present chapter, we combined apprenticeship learning techniques and model derived from existing PSR, known as TPSR. The resulting agent learned a policy of maintaining the optimal DoA using expert trajectories. The use of machine learning models based on observable variables during GA is pertinent due to the high number of information intractable for the human brain. The performances of the resulting model are promising and convincingly embedded the general behavior of an anesthesiologist. These preliminary results are very encouraging and demonstrate that cardio-pulmonary changes induced by GA can relatively easily be predicted by apprenticeship-learning based algorithm allowing a potentially significant improvement in care.

6

Conclusion and perspectives

In this thesis, five contributions were proposed. The first one was the construction and deployment of a complete protocol and recording chain that has enabled us to build a large database of patients under routine General Anesthesia (GA). This contribution was motivated by the privileged applicative context of this work which was the study of patients under anesthesia. Then, because signals recorded during GA are mainly multivariate, e.g. multichannel ElectroEncephalogram (EEG) recordings, three contributions focused on methods processing multivariate signals efficiently. More specifically, in Chapters 2, 3, and 4, we proposed several methods built on graphs and tensors which are known to exploit the underlying multivariate data structure. Finally, in the last chapter, we made a more prospective contribution consisting in a first attempt at automatic and individual administration of anesthetics for patients under GA relying on reinforcement learning techniques. We further summarized and gave some future perspectives of each chapter in the following.

In Chapter 2, we introduced an optimization problem to learn the underlying graph from a set of graph signals supposed to share the same structure. This graph learning task being ill-posed, two constraints known as *smoothness* and *sparse spectral representation* were included. Borrowed from graph signal processing, these two constraints allow to learn a graph which reflects the topology of the data. The main idea behind the inclusion of the second constraint was to find a graph which makes signals *bandlimited* over it. This important feature being known to carry information related to the cluster structure of the graph, makes this graph a good candidate in the initialization of spectral clustering methods. A first algorithm, called IGL-3SR, was proposed to solve this problem by combining barrier methods, alternating minimization, and manifold optimization. A relaxed algorithm, called FGL-3SR, was also introduced, which allows to scale in time with the graph dimensions. The numerical experiments of this chapter showed that both algorithms display competitive results with regards to previous methods. Three interesting directions of research would be (i) to prove the convergence of these algorithms to at least a local minimum, (ii) to derive concentration bounds on the estimated Laplacian matrix, (iii) to consider dynamic graph topologies i.e. network structures which change over time. To date, several works have already addressed these questions but on other related settings. For instance, [Kumar et al., 2019, 2020] introduced provably convergent algorithms when considering Gaussian graphical models and studied structural constraints on the eigenvalues of the Laplacian. In Sardellitti et al. [2019], authors studied the conditions under which the Laplacian matrix can be recovered uniquely when signals are considered sparse over the underlying graph. However, their results are only deterministic, and providing statistical properties based on probabilistic model could be of interest. Beside these questions, several researches in multitask learning [Argyriou et al., 2006; Jacob et al., 2009] may be useful to propose more efficient learning algorithms. A more applicative direction would be to conduct further experiments to see whether learned graphs

from EEG signals provide a better overview of the different states occurring during GA. This was already a fruitful idea in the analysis of brain activity [Richiardi et al., 2013; Huang et al., 2016, 2018], but graphs were then given a priori.

Chapter 3 introduced a multivariate Convolutional Dictionary Learning (CDL) problem, called Kruskal CDL, K-CDL for short, where the multivariate activations are assumed to be CP low-rank. By taking into account the structure of the activations, this model has two major advantages over the standard CDL. First, as it decomposes the activations into the sum of rank-1 tensors, results are highly interpretable. Second, it turned out that the CP low-rank constraint allows to entail a better robustness with respect to noise, one of the main weaknesses of the activation learning part of CDL methods. In this chapter, two algorithms, called T-ConvADMM and T-ConvFISTA, were proposed for the K-CDL problem. We proved that by acting in the frequency domain and by using the particular structure of the matrices involved in the optimization at our advantage, they have a theoretical complexity which increases quadratically in the number of atoms and with the rank. Overall, experiments on synthetic and real data showed that the K-CSC is a valuable alternative to CDL when signals are multivariate. Interestingly, this has also been the case even if signals seem to have a richer structure like images. In the future, it would be interesting to study the statistical influence of the sparse and CP low-rank assumptions on the estimator, especially on its robustness to noise. This has already been a subject of interest in numerous other settings with sparse or low-rank matrix and tensor recovery [Fazel, 2003; Rohde et al., 2011; Donoho et al., 2014; Yang et al., 2016; de Morais Goulart, 2016; Rauhut et al., 2017; Li et al., 2020], compressed sensing [Candès et al., 2006; Donoho, 2006], matrix and tensor completion [Candès and Recht, 2009; Koltchinskii et al., 2011; Negahban and Wainwright, 2011; Recht, 2011; Liu et al., 2012; Gandy et al., 2011], etc. It would also be interesting to focus on how to overcome the non-convexity of the vast majority of the tensor decomposition/factorization problems [Haeffele and Vidal, 2015]. Another important line of research would be to study other sparsity-induced structure and rank constraints in convolutional representation. Note that this is already done in tensor regression with e.g. sparsity constraint on each rank-1 tensor of the CP decomposition [He et al., 2018], Tucker low-rank constraint [Li et al., 2018], multilinear rank constraint [Rabusseau and Kadri, 2016; Sun and Li, 2017], etc. A generalization of these decompositions, known as *tensor network models*, could also be investigated [Orús, 2014; Cichocki et al., 2016; Li and Sun].

Chapter 4 proposed a method to recover the spectral support of bandlimited multivariate time-vertex graph signals defined on a product of graphs. By taking into account the three dimensions time, space, and feature of a multichannel EEG signal, we highlighted the importance of the underlying graphs for sampling. In addition, we introduced a simple way to assess the relevance of the graphs chosen a priori by comparing our results with those obtained when random graphs are choose instead. Results showed the importance of graphs in this algorithm and support for the relevance of *controlled sparsity constraint* to recover multivariate sparse (bandlimited) signals. An interesting direction of research would be to learn the different graphs constituting the product graph with the method proposed in Chapter 2. This idea has already been investigated when the multivariate graphs signals are only assumed smooth with respect to the underlying product graph [Kadambari and Chepuri, 2020; Lodhi and Bajwa, 2020]. A more applicative contribution would be to use this method for the selection of relevant EEG channels. By identifying the scalp area providing valuable information about brain activity under GA, we could select the most optimal EEG channel to characterize the DoA. This would be of great help to the anesthesiologist who could only rely on a subset of these channels [Dubost et al., 2019]. Furthermore, this could be an interesting way to answer the more general question of channel selection (see e.g. [Arvaneh

et al., 2011; Alotaiby et al., 2015]).

Finally, Chapter 5 is a first attempt to propose a decision support tool based on a predictive state representation model which assists anesthesiologists in administering anesthetics during a general anesthesia and to maintain the optimal Depth of Anesthesia (DoA). This algorithm based on a predictive state representation model exhibits interesting quantitative results. In addition, because a precise evaluation of the quantitative performances of reinforcement learning algorithms are difficult to obtain in healthcare applications [Gottesman et al., 2018, 2019; Yu et al., 2019], a confrontation with real anesthesiologists was performed. These results strongly support the hypothesis that this model convincingly embedded the behavior of anesthesiologists. Nevertheless, this model could be improved in several ways (i) by assuming continuous and not discrete observations using e.g. a kernel density estimation method from Boots et al. [2011]), and (ii) by increasing the number of available actions as in Moore et al. [2014]. Several recent works on apprenticeship learning (or imitation learning), especially the ones focusing on how to effectively learn from imperfect demonstrations, could also be considered [Ho and Ermon, 2016; Wu et al., 2019]. However, great care must be taken. In particular, we strongly believe that these types of approaches should only be considered as part of a support system to accompany the anesthesiologists, not a replacement.

Developing these models, we strove to make their codes available online with documentation to facilitate their use in the community. Indeed, as re-implementing recent technical methods is a major time-consuming task. We believe open-source projects are of major interest. Moreover, we plan to release the database to benefit the community at large.

On a concluding note, we would like to stress that questions that motivated this thesis lie beyond the GA. Indeed, multivariate data are now present in multiple datasets and will undoubtedly become increasingly complex. The methods and algorithms described in this work have therefore a great potential, and can already be successfully used in countless situations as shown throughout this thesis.

7

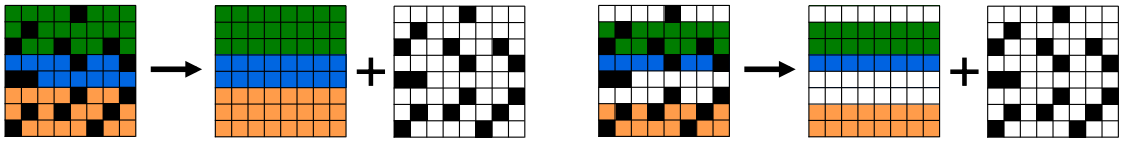
Résumé en français

1 Contexte de la thèse

Contexte général. Le corps humain est dans un état constant d'équilibre appelé homéostasie. Si cette stabilité est fondamentale, elle nécessite une régulation constante et précise des organes vitaux par le cerveau. Lors d'une Anesthésie Générale (AG), une partie de cette stabilité est mise à mal par les anesthésiques. Les anesthésistes doivent alors soutenir eux même certaines fonctions vitales, comme le système respiratoire, en personnalisant l'anesthésie.

L'objectif d'une anesthésie personnalisée est double : (i) éviter une narcose associée à un risque plus élevé de dysfonctionnement cognitif postopératoire et de réveil tardif, (ii) prévenir un sous-dosage, associé à un risque de mémorisation. Les anesthésistes doivent donc déduire, en temps réel, le niveau de conscience du patient, également appelé profondeur de l'anesthésie (DoA en anglais) et ainsi adapter leurs dosages. Depuis peu, ils peuvent s'appuyer sur une large gamme de variables physiologiques mesurées par de nombreux capteurs. Ce remarquable changement dans le domaine médical est en parti due à l'amélioration des capteurs et à leur utilisation systématique. Une conséquence directe est la disponibilité de grande quantité de données. Les exemples les plus connus de ces données sont les signaux mesurés par électrocardiographie (ECG), électroencéphalographie (EEG) ou encore toutes les variables physiologiques. Ce changement est particulièrement notable dans le domaine de l'anesthésie clinique, où la quantité de données était très limitée. La question principale est maintenant de savoir comment les mathématiques peuvent nous aider à transformer ces signaux bruts en des données où il est possible d'extraire des connaissances. Cette question au carrefour des mathématiques et de la médecine est d'autant plus cruciale en ce qu'elle pourrait conduire à d'importants avancées dans la manière de soigner les patients mais aussi dans notre compréhension de la physiologie humaine.

Collaboration avec l'unité médicale du Centre Borelli. Au cours de cette thèse, j'ai collaboré avec l'unité médicale du Centre Borelli (ex Cognac-G). Ce centre est une équipe de recherche regroupant des mathématiciens (statisticiens, spécialistes de l'apprentissage automatique, etc.) et des chercheurs en médecine, tous réunis autour de la quantification du comportement humain. J'ai notamment travaillé en étroite collaboration avec le docteur Clément Dubost, chef du service de réanimation de l'Hôpital d'Instruction des Armées Bégin. Ensemble nous avons conçu un protocole complet - de la chaîne d'enregistrement à l'analyse des données - dans le but de proposer des méthodes mathématiques utiles à l'étude des patients sous anesthésie. Par le passé, le Centre Borelli a déjà développé plusieurs protocoles expérimentaux pour des problèmes cliniques allant de la locomotion humaine aux mouvements oculaires des nourrissons. La quantification du phénomène d'intérêt a toujours été faite grâce à l'analyse de signaux enregistrés avec plusieurs capteurs. Le premier objectif étant d'extraire les informations pertinentes de ces signaux pour



(a) Rang faible **(b)** Parcimonie groupée et rang faible

Figure 7.1: Illustration de (a) l'hypothèse de rang faible, (b) la combinaison des hypothèses de rang faible et de parcimonie.

en comprendre les mécanismes physiologiques qui les ont produits. Le second objectif étant d'automatiser le processus de quantification afin de fournir des outils utilisables en routine.

2 Motivations

2.1 Comprendre les données brutes par leurs structures multivariées

Repenser la médecine ne peut se faire sans changements importants dans la façon dont nous analysons les données médicales. En effet, les données issus des recherches actuelles sont souvent beaucoup plus volumineux et plus complexes que celles d'autrefois. Ce phénomène est en partie dû à la démocratisation des capteurs bon marché et faciles à manipuler qui simplifient la collecte systématique de nombreuses données sur les patients. Par conséquent, désormais, de multiples signaux, tels que les signaux ECG ou EEG, sont enregistrés presque quotidiennement. Or, leur grande diversité et leur volume important nécessitent inévitablement des améliorations dans les techniques de stockage et de manipulation de données, ainsi que des avancées dans les méthodes d'analyse. Pour étudier efficacement ces données, plusieurs approches ont été adoptées. Dans un premier temps, la tendance était de se focaliser sur l'analyse des données *univariées* avec des modèles comprenant une seule variable à expliquer. Les recherches se sont surtout concentrés sur l'intégration de connaissances préalables sur les données soit en faisant des hypothèses sur la classe de modèles pour en restreindre la complexité, soit par le biais de contraintes et de régularisations. Un exemple classique illustrant cette dernière approches est la régression ridge proposée pour la première fois par Tikhonov [1963]. Dans ce cas, un modèle linéaire est supposé et une régularisation ℓ_2, c'est-à-dire une hypothèse de *régularité*, est ajoutée afin d'éviter des coefficients trop grands. Un autre exemple important est la régression lasso [Tibshirani, 1996] où une régularisation ℓ_1, c'est-à-dire une hypothèse de *parcimonie*, est ajoutée. D'autres modèles existent pour ajouter des connaissances préalables sur la structure des signaux. C'est par exemple le cas des représentations convolutives [Garcia-Cardona and Wohlberg, 2018a], qui permettent d'extraire des motifs récurrents non sinusoïdaux et conduisent ainsi à la découverte de structures locales dans des signaux non stationnaires comme les séries temporelles [Lewicki and Sejnowski, 1999; Grosse et al., 2007].

Bien que toutes ces idées aient conduit à des avancées tant théoriques que pratiques, un écart existe entre les résultats obtenus dans le cas univarié et ce que l'on peut attendre de modèles statistiques bien définis. En effet, les signaux à expliquer sont souvent *multivariés* et les relations entre leurs variables, ou dimensions, doivent être prises en compte si l'on veut les analyser de manière adéquate. Pour combler cette lacune, les statisticiens se sont penché sur l'analyse multivariée et ont développé des techniques permettant, par exemple, la présence de plus d'une variable de sortie [Van Steen and Molenberghs; Hidalgo and Goodman, 2013]. Pour aller au-delà du cas univarié, une première étape naturelle est de considérer le cas bivarié où la variable à expliquer

est matricielle. De nombreuses stratégies ont été proposées pour incorporer les relations entre les différentes dimensions de ces données, mettant en évidence ce qu'une analyse multivariée peut apporter en termes de performance et d'interprétabilité. Un intéret du cas bivarié est qu'il nous permet de considérer des propriétés et des structures jusqu'alors indisponibles. C'est le cas de la structure de rang faible exploitée dans de nombreuses méthodes comme l'Analyse en Composantes Principales (ACP) de rang faible [Vidal et al., 2016], la reconstruction de matrice [Fazel, 2003; Rohde et al., 2011] (matrix recovery en anglais), et la complétion de matrice [Candès and Recht, 2009; Koltchinskii et al., 2011; Negahban and Wainwright, 2011; Recht, 2011]. La combinaison des structures de rang faible et de parcimonie est également apparue pertinente dans un certain nombre d'applications. Selon la combinaison (voir figure 7.1), cela donne lieu à des méthodes plus robustes et interprétables telles que l'ACP parcimonieuse [Zou et al., 2006], la classification non-supervisée de sous-espaces [Vidal, 2011; Udell et al., 2016; Haeffele and Vidal, 2019], et la classification non-supervisée de sous-espaces parcimonieux avec valeurs aberrantes [Elhamifar and Vidal, 2013].

Analyse multivariée à l'aide de graphes. Outre les notions de rang faible et de parcimonie, une autre façon d'exploiter la structure des données multivariées consiste à utiliser la notion de *graphe*. En effet, le graphe peux apporter des connaissances précieuses sur le processus qui génère les données (par ex. deux nœuds liés sont fortement corrélés ou ont des valeurs très proches), ce qui le rend utile dans bon nombre de domaines et d'applications allant de la biologie [Barabasi and Oltvai, 2004] aux neurosciences [Richiardi et al., 2013; Preti et al., 2017], en passant par la classification non-supervisé [Belkin and Niyogi, 2002; Von Luxburg, 2007], l'apprentissage par représentation [William et al., 2017], l'apprentissage multitâche [Chen et al., 2015a; Nassif et al., 2020], etc. [Zhu, 2005; Kolaczyk and Csárdi, 2014]. Construire des modèles ou des algorithmes d'apprentissage en tenant compte de la structure de graphe des données, est donc un élément clé pour améliorer les performances. Il reste à trouver un moyen d'incorporer ces informations structurelles dans les modèles et les méthodes. Une possibilité est de considérer des modèles graphiques probabilistes non dirigés où un ensemble de variables aléatoires est représenté par les différents nœuds d'un graphe [Koller and Friedman, 2009]. Dans cette représentation, une arête entre deux nœuds indique la dépendance conditionnelle entre les deux variables aléatoires correspondantes, sachant les autres. Plus récemment, le *Traitement des Signaux sur Graphes* (GSP en anglais) [Shuman et al., 2013; Ortega et al., 2018; Djuric and Richard, 2018], est apparu comme une puissante alternative pour extraire des informations de signaux multivariées. Pour prendre en compte la structure du signal, l'idée est de le considérer comme défini sur les nœuds d'un graphe et d'encoder les relations entre ses variables via les arêtes. Dans ce formalisme, le graphe définit un support, et les signaux, désormais appelés *signaux sur graphes*, sont définis sur ce support. Cela permet de capturer la structure sur laquelle un signal évolue, fournissant ainsi plus d'informations que si l'on considère le signal seul. De plus, en généralisant les concepts standards du traitement du signal aux signaux sur graphes, le GSP fournit des contraintes intuitives pour la modélisation. Par exemple, la *régularité* des observations par rapport au vrai graphe sous-jacent est l'une des hypothèses la plus courante qui demande à ce que les signaux aient de petites variations entre les nœuds adjacents. [Daitch et al., 2009; Egilmez et al., 2016; Kalofolias, 2016; Chepuri et al., 2017; Dong et al., 2019]. Cette propriété très naturelle est exploitée dans beaucoup d'applications. On peut citer l'estimation multi-tâche sur graphe [Nassif et al., 2020] où le graphe capture le lien entre plusieurs tâches permettant aux agents de coopérer entre eux. Cette coopération peut être encouragée par une régularisation qui impose un certain degré de proximité entre les différentes règles de décision de chaque agent [Nassif et al., 2018]. Bien souvent, la connaissance du graphe

est une hypothèse de base. C'est le cas par exemple pour la classification spectral non-supervisée [Von Luxburg, 2007]. Malheureusement, dans la plupart des situations, aucun graphe ne peut être défini. Une approche consiste donc à le déduire d'un ensemble de signaux supposés admettre le même graphe sous-jacent. Cette tâche, souvent appelée *apprentissage de graphes* (ou inférence de topologie de graphe), a fait l'objet d'une attention toute particulière dans divers domaines tels que la statistique, le traitement du signal, la biologie, etc. [Friedman et al., 2008; Hecker et al., 2009; Lim et al., 2015; Moscu et al., 2020]. Une revue de la littérature des méthodes récentes d'apprentissage de graphes est disponible dans [Dong et al., 2019].

Analyse multivariée à l'aide de tenseurs L'extension naturelle du cas bivarié est le cas multivarié où la variable à expliquer est maintenant tensorielle. Comme pour le passage de la première à la deuxième dimension, de nouvelles possibilités, et donc de nouvelles stratégies, sont disponibles pour exploiter la structure de ces données multivariées. Un grand nombre de travaux se sont concentrés sur les méthodes tensorielles. Cet intérêt croissant est principalement dû à leur capacité à mieux exploiter l'aspect multivarié des données. En effet, en partie poussée par les travaux pionniers de Cattell [1944] en psychométrie, l'application des méthodes tensorielles a eu du succès en traitement du signal [Zhou et al., 2013; Cichocki et al., 2015], vision par ordinateur [Shashua and Hazan, 2005; Liu et al., 2012], apprentissage spectral de modèles à variables latentes [Anandkumar et al., 2014; Janzamin et al., 2019], neurosciences [Beckmann and Smith, 2005; Miwakeichi et al., 2004; Mørup et al., 2006; Becker et al., 2015], etc. Des études approfondies de ces méthodes sont disponibles dans Kolda and Bader [2009]; Grasedyck et al. [2013] et Sidiropoulos et al. [2017]. Dans cette vaste littérature, l'une des stratégies les plus utilisées consiste à appliquer directement une *décomposition tensorielle* aux données. Cela conduit souvent à des résultats plus interprétables et à de meilleures performances. En effet, en factorisant les données dans un espace de dimension inférieure, les décompositions tensorielles introduisent une base qui peut décrire les données de manière plus concise. Un exemple important d'une telle décomposition est la Décomposition Canonique Polyadique (DCP) [Hitchcock, 1927], également connue sous le nom de Parafac ou CANDECOMP [Harshman, 1970; Carroll and Chang, 1970]. Celle-ci exprime un tenseur comme une somme minimale de tenseurs de rang un. D'autres décompositions, telles que la décomposition de Tucker [Tucker, 1963], ou la décomposition en valeurs singulières d'ordre supérieur [De Lathauwer et al., 2000], se sont avérées efficaces. Ces décompositions ont notamment conduit à des progrès significatifs en complétion tensorielle [Gandy et al., 2011; Liu et al., 2012; Goulart and Favier; Rauhut et al., 2017]. Une autre stratégie consiste à imposer des structures tensorielles dans des méthodes déjà existantes par le biais de contraintes et de régularisations supplémentaires. Dans Zhou et al. [2013], les auteurs ont proposé une famille de modèles de régression tensorielle où une contrainte *de rang CP faible* est ajoutée. Ils ont également étendu ces modèles aux contraintes de rang de Tucker faible [Li et al., 2018]. D'autres se sont intéressés aux contraintes de rang multilinéaires [Rabusseau and Kadri, 2016; Sun and Li, 2017], aux contraintes de parcimonie sur chaque tenseur de rang 1 de la DCP [He et al., 2018], etc. L'idée d'imposer une structure particulière par des contraintes est également utilisée dans plusieurs modèles d'apprentissage de dictionnaires multivariés [Hawe et al., 2013; Sironi et al., 2014; Dantas et al., 2018; Schwab et al., 2019] ou même pour accélérer les réseaux de neurones convolutifs [Lebedev et al., 2015; Kim et al., 2016; Astrid and Lee, 2017]. Globalement, si tous ces modèles apportent inévitablement plusieurs difficultés dues à la grande complexité des objets manipulés, ils ont prouvé leur utilité et montré, une fois de plus, qu'il est important de bien prendre en compte la structure des données pour obtenir de meilleurs résultats.

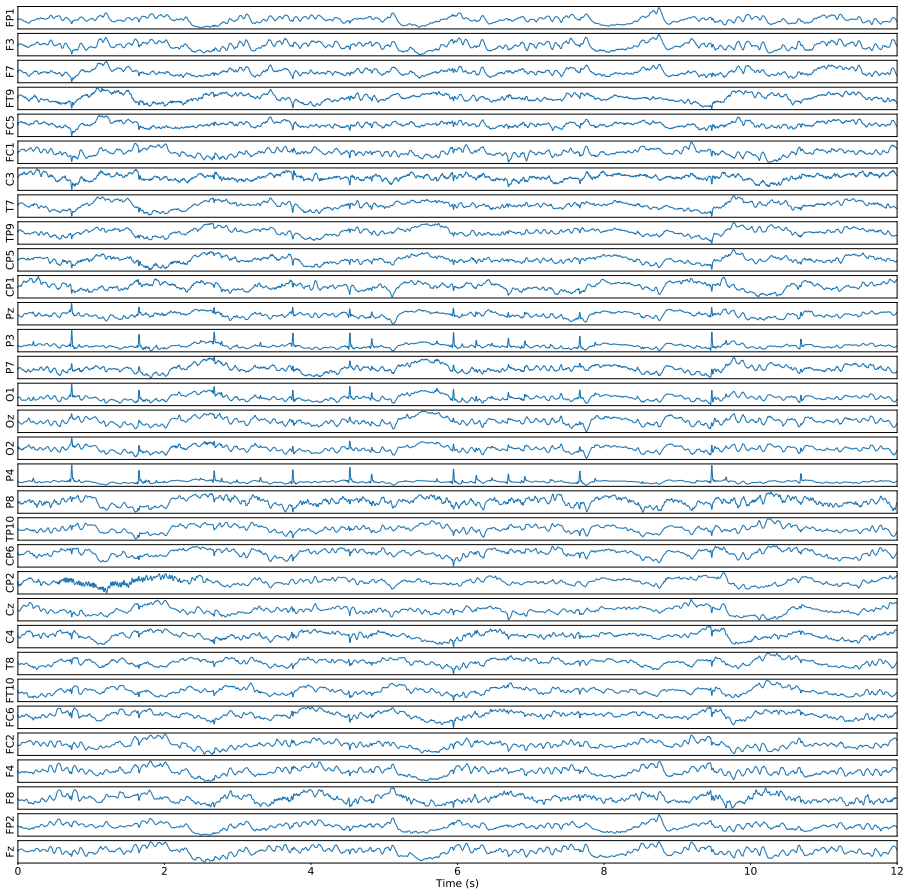


Figure 7.2: Représentation de 32 canaux EEG d'un patient. Sur l'axe y de chaque signal est annoté le nom du canal correspondant.

2.2 Analyse de la conscience pendant une anesthésie générale

Cette thèse s'est également construite autour de la nécessité d'analyser des données enregistrées lors d'une *Anesthésie Générale* (AG). AG est un état réversible induit qui comprend des traits comportementaux et physiologiques spécifiques (inconscience, amnésie, analgésie et akinésie) [Brown et al., 2010]. Cet état non naturel est obtenu principalement par l'utilisation de différentes drogues (par exemple, des anesthésiques hypnotiques inhalés - le sévoflurane - ou des anesthésiques intraveineux - le propofol) qui agissent sur les récepteurs inhibiteurs GABA du cerveau. Cependant, bien que l'AG soit une pierre angulaire de la médecine moderne, et qu'elle soit cruciale dans de nombreuses procédures chirurgicales [Purdon et al., 2013], elle peut comporter certains risques (par exemple, dysfonctionnement cognitif [Punjasawadwong et al., 2018], délire postopératoire [Fritz et al., 2016]). Une surveillance permanente de l'état de conscience du patient – également appelée profondeur d'anesthésie (DoA en anglais) – est donc nécessaire. Bien qu'il n'existe pas de définition consensuelle du DoA, elle a été définie par les experts comme "la probabilité de non-réponse à une stimulation, calibrée en fonction de la force du stimulus, de la difficulté à supprimer la réponse et de la probabilité de non-réponse induite par le médicament à des concentrations définies au site d'effet" [Shafer and Stanski, 2008]. Sa connaissance précise est essentielle pour permettre un titrage précis des anesthésiques administrés. Les principaux objectifs sont d'éviter une narcose, associée à un risque plus élevé de dysfonctionnement cognitif postopératoire et de

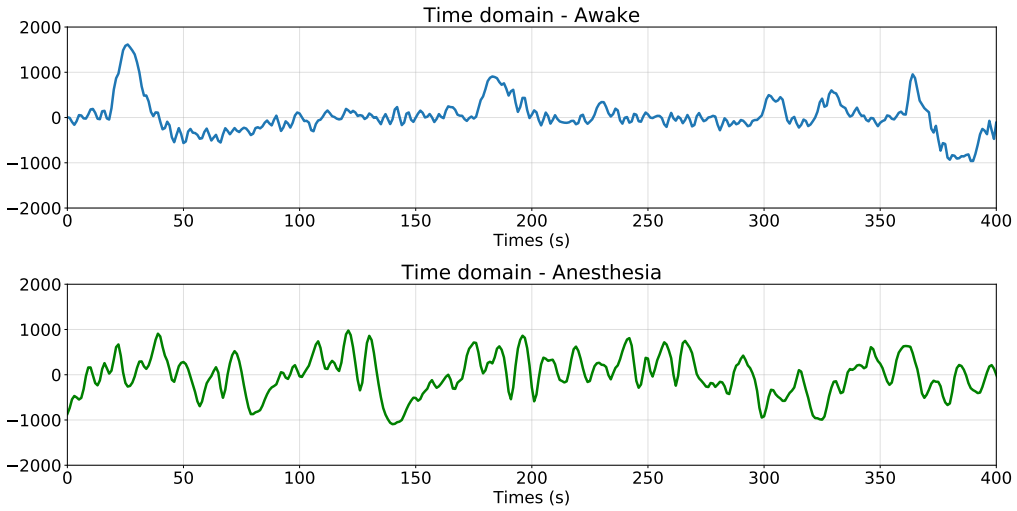


Figure 7.3: En haut, signal temporel pendant au réveil. En bas, signal temporel pendant l'anesthésie.

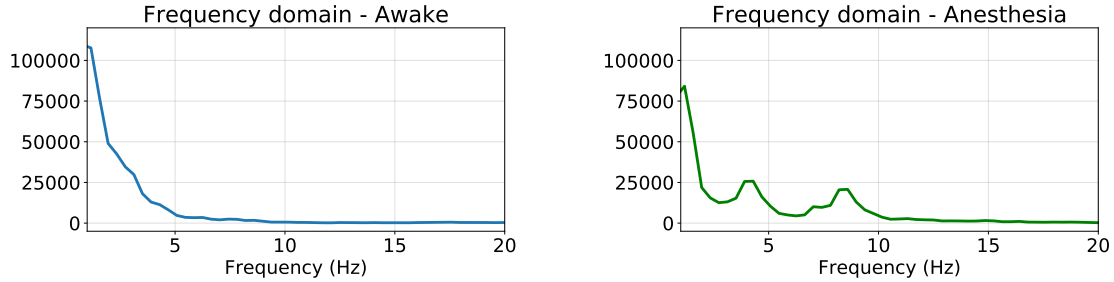


Figure 7.4: A gauche, spectre d'un signal EEG au réveil. A droite, spectre d'un signal EEG pendant l'anesthésie.

réveil tardif, et de prévenir un sous-dosage, associé à un risque de mémorisation [Sebel et al., 2004].

Fonctionnement du cerveau pendant une anesthésie. L'analyse des signaux mesurés par ElectroEncéphaloGraphie (EEG) reste la référence pour évaluer la DoA (voir figure 7.2). En effet, ces signaux sont une mesure directe de la principale cible des anesthésiques, le cerveau [Merry et al., 2010]. Ainsi, bon nombre des changements se produisant dans le cerveau peuvent y être facilement observés [Tong and Thakor, 2009; Sanei and Chambers, 2013; Cohen, 2014]. Fort de ce constat, depuis les années 2000, l'EEG est largement utilisée pour étudier les phénomènes survenant lors d'une AG [Purdon et al., 2015; Liu and Rinehart, 2016]. Les recherches ont ainsi montré que l'AG induit certains comportements dans les signaux EEG qui peuvent être décrits en fonction de cinq états : L'éveil, la Perte de Conscience (PdC), L'anesthésie, le Rétablissement de la Conscience (RdC), et enfin, l'émergence. Lorsque l'anesthésie s'approfondit, le schéma le plus connu et le plus courant est une augmentation progressive de bandes de fréquences spécifiques et de l'amplitude du signal. La figure 7.3 illustre ce phénomène en montrant le canal EEG frontal d'un patient réveillé puis sous anesthésie. On y voit clairement des changements dans les données brutes avec l'apparition de petites ondes de grandes amplitudes. Ces changements visuels, présents chez presque tous les patients, entraînent une modification du spectre du signal EEG (voir Figure 7.4). Dans une importante étude menée par Purdon et al. [2013], les chercheurs

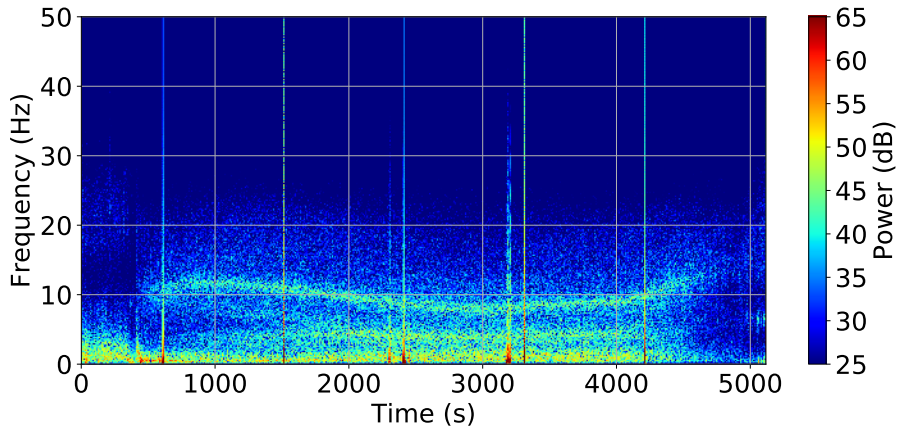


Figure 7.5: Spectrogramme d'un patient sous anesthésie générale induite par propofol et sévoflurane.

ont montré que la puissance des ondes α et δ (respectivement dans les bandes 8-13 Hz et 1-3Hz) est un bon indicateur des différents états d'un patient pendant une AG lorsqu'elle est induite par du propofol. En effet, ils ont montré que la puissance de ces deux bandes de fréquences tend à augmenter à mesure que la quantité de drogue augmente. En conséquence, leur suivi permet de définir précisément dans quel état un patient est le plus susceptible de se trouver. A l'aide d'un spectrogramme, l'évolution de la puissance de chaque fréquence au cours du temps est représentée sur la figure 7.5. Ils constatent également que ces modifications au niveau d'un canal sont combinées à une modification spatiale appelée "antériorisation". Plus précisément, alors qu'à l'état d'éveil, les ondes alpha sont principalement présentes à l'arrière du crâne, après induction du propofol, ces ondes migrent lentement vers le front. Ce processus s'inverse lorsque la quantité de propofol diminue. Par cet exemple, nous comprenons l'importance d'aller au-delà d'une analyse univariée pour décrire et comprendre pleinement les mécanismes globaux.

A routine clinical context. Si ces études permettent de mieux comprendre l'AG, elles sont, pour la plupart, menées dans un environnement idéal. En clinique, la réalité est tout autre. Tout d'abord, les anesthésistes utilisent non pas un, mais plusieurs anesthésiques pour induire l'AG. L'analyse devient alors plus difficile car chacun d'eux induit ses propres comportement temps-fréquence [Purdon et al., 2015]. Deuxièmement, les méthodes d'analyse des signaux EEG sont peu robuste aux bruits. Un problème courant surtout lorsque les données sont enregistrées pendant des interventions chirurgicales. En effet, même s'il n'y a pas d'artefact dû aux contractions musculaires (les patients sont curarisés), les signaux EEG sont toujours sujets à un faible rapport signal/bruit, à un bruit impulsif dû à des dysfonctionnements du capteur et à des artefacts causés, par exemple, par des appareils utilisés pour couper et cautériser les tissus (voir figure 7.6) [Tong and Thakor, 2009]. Il devient donc très difficile d'utiliser les méthodes standard qui supposent une configuration théorique idéale. Troisièmement, l'utilisation des méthodes d'EEG prend du temps, ce qui les rend inutilisables au quotidienne. Ces trois exemples nous montrent que d'autres méthodes, pas nécessairement basées sur l'EEG, doivent être étudiées.

Pour surmonter toutes ces difficultés, plusieurs systèmes de surveillance ont été proposés pour l'évaluation de la DoA au cours d'une intervention chirurgicale. Tous présentent quelques limites et il n'existe pas encore de "gold-standard" de surveillance de la DoA. Le système sans doute le plus utilisé est l'indice BiSpectral (BIS) [Kissin, 2000; Avidan et al., 2008]. Il fournit une valeur

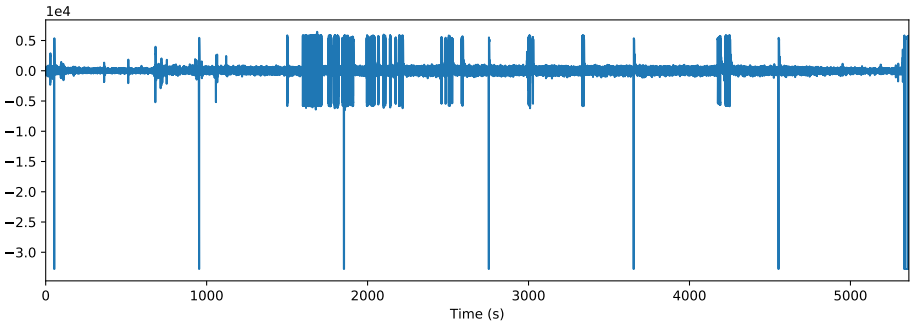


Figure 7.6: EEG enregistré pendant une anesthésie générale présentant beaucoup de bruit. L'unité de l'axe des y est μV.

numérique de 0 à 100 (de l'absence d'activité cérébrale à l'éveil). Cependant, bien qu'il soit souvent utilisé, en particulier aux États-Unis, il présente de nombreux inconvénients tels qu'une grande variabilité inter-individuelle [Whitlock et al., 2011], de faibles performances avec les anesthésiques volatils [George Mychaskiw et al., 2001], et une latence élevée. En somme, l'EEG semble être la meilleure méthode pour évaluer la DoA, bien qu'elle nécessite des capteurs supplémentaires, présente certaines limites et prend du temps. C'est pourquoi, en clinique, la meilleure évaluation de la DoA est, la plupart du temps, celle réalisée par l'anesthésiste sur la base des variables physiologiques du patient.

En résumé, dans la pratique, un monitoring idéal de l'AG devrait être capable de donner une évaluation sans EEG. En outre, alors que l'analyse de l'AG est souvent centrée sur d'anciennes méthodes d'analyse, telles que la représentation temps-fréquence, nous pensons que les progrès récents en matière de statistiques et d'apprentissage automatique pourraient grandement contribuer à une compréhension plus fine des mécanismes complexes qui se produisent pendant l'AG.

3 Contributions

Nous détaillons ici les différentes contributions de cette thèse. Afin de souligner leur polyvalence, chaque contribution est accompagné d'une grande variété d'expériences, dont au moins une liée à l'anesthésie générale. De plus, pour chaque algorithme, un code Python open-source est disponible en ligne.

3.1 Une base de données de patients sous anesthésie générale

Fruit d'une collaboration avec le docteur Clément Dubost, la première contribution de cette thèse est la construction et le déploiement d'une chaîne de mesures nous ayant permis de constituer une base de données de patients sous AG. Aidés par Brian Berthet-Delteil, Arno Benizri et Gaël de Rocquigny, nous avons enregistré en continue et de manière synchrone les variables physiologiques de routine d'une anesthésie ainsi que 32 canaux EEG. De février 2016 à mai 2018, 88 sujets, tous issus de "l'Hôpital d'Instruction des Armées Bégin, Saint-Mandé, France", ont été inclus dans la base de données. Notons qu'à notre connaissance, il s'agit de la première base de données de patients sous AG où à la fois des EEG multicanaux et des variables physiologiques sont enregistrés de manière synchrone depuis l'entrée en salle d'opération et jusqu'à trois heures après la fin de l'intervention.

3.2 Apprentissage de graphes

Pour deuxième contribution, nous considérons le problème de l'apprentissage d'un graphe à partir de signaux multivariés sur graphes. Comme nous l'avons déjà expliqué, ces signaux sont associés à un graphe inconnu, que nous souhaitons apprendre. L'idée de cette contribution vient d'une observation simple. En général, nous ne disposons pas d'un graphe adapté au signal d'intérêt. Une idée possible est donc de l'apprendre. Cependant, comme il s'agit d'un problème mal posé, nous devons supposer plusieurs propriétés sur les signaux observés et le graphe associé. Dans notre approche, ces propriétés s'inspirent du domaine du Traitement des Signaux sur Graphes (GSP en anglais) [Shuman et al., 2013; Ortega et al., 2018]. Ce domaine fournit des contraintes structurelles intuitives induites par les graphes, et a déjà fait ses preuves dans de nombreuses applications, notamment en neurosciences avec l'analyse du cerveau. En effet, Huang et al. [2018] montrent qu'en construisant un graphe à partir de la connectivité structurelle et en considérant l'activité cérébrale comme des signaux sur graphes, il est possible de capturer des propriétés cérébrales pertinentes (par exemple, des caractéristiques cognitives) avec des concepts du GSP.

Plus précisément, nous élaborons un problème d'optimisation pour apprendre le Laplacien du graphe sous-jacent. Pour rendre identifiable ce problème mal posé, les signaux observés sont supposés se comporter de manière régulière/lisse sur le même graphe et admettre une représentation parcimonieuse dans le domaine spectral de ce graphe. Cette dernière propriété de *largeur de bande faible* est connue pour porter des informations liées au nombre de clusters du graphe [Von Luxburg, 2007; Sardellitti et al., 2019]. Le graphe appris est donc un bon candidat dans l'initialisation des méthodes de classification spectral non-supervisée. Notez que ces deux propriétés sont également des hypothèses de base dans un grand nombre de méthodes comme par exemple l'échantillonnage sur graphes, ou l'interpolation sur graphes. Pour résoudre ce problème d'apprentissage, nous proposons deux algorithmes appelés IGL-3SR et FGL-3SR. Basés sur une procédure alternée, les deux algorithmes s'appuient sur des méthodes de minimisation standard – telles que la descente du gradient sur variétés riemanniennes ou l'optimisation linéaire. Alors que IGL-3SR est assuré de converger, FGL-3SR est une relaxation du problème de base et est donc significativement plus rapide que les autres méthodes. Pour mettre en évidence l'efficacité de nos méthodes, nous fournissons de nombreux exemples allant de la météorologie à l'analyses de signaux EEG.

3.3 Apprentissage de dictionnaires convolutifs tensorielles

La troisième contribution résulte de la combinaison de deux familles de méthodes d'analyse de signaux multivariés. La première famille de méthodes est l'apprentissage de dictionnaires convolutifs (CDL en anglais) [Wohlberg, 2015; Garcia-Cardona and Wohlberg, 2018a]. Elle consiste à apprendre des atomes - ou motifs - locaux permettant une reconstruction parcimonieuses des signaux. Ainsi, contrairement aux méthodes considérant des bases de Fourier ou d'ondelettes, ici, les atomes ne sont pas prédéfinis et sont appris à partir du signal lui-même. Cette idée de fournir une décomposition linéaire d'un signal en quelques atomes locaux appris, au lieu d'atomes prédéfinis, a conduit à des résultats significatifs dans de nombreux domaine comme la classification d'images, la restauration d'images et le traitement du signal (voir [Wohlberg, 2015; Garcia-Cardona and Wohlberg, 2018a]). Néanmoins, bien que ces méthodes aient des propriétés intéressantes, elles sont principalement axées sur le traitement de signaux univariés [Garcia-Cardona and Wohlberg, 2018b], et ne prennent donc pas pleinement en compte l'interaction possible entre les différents dimensionss des signaux multivariés. De plus, elles sont bien souvent

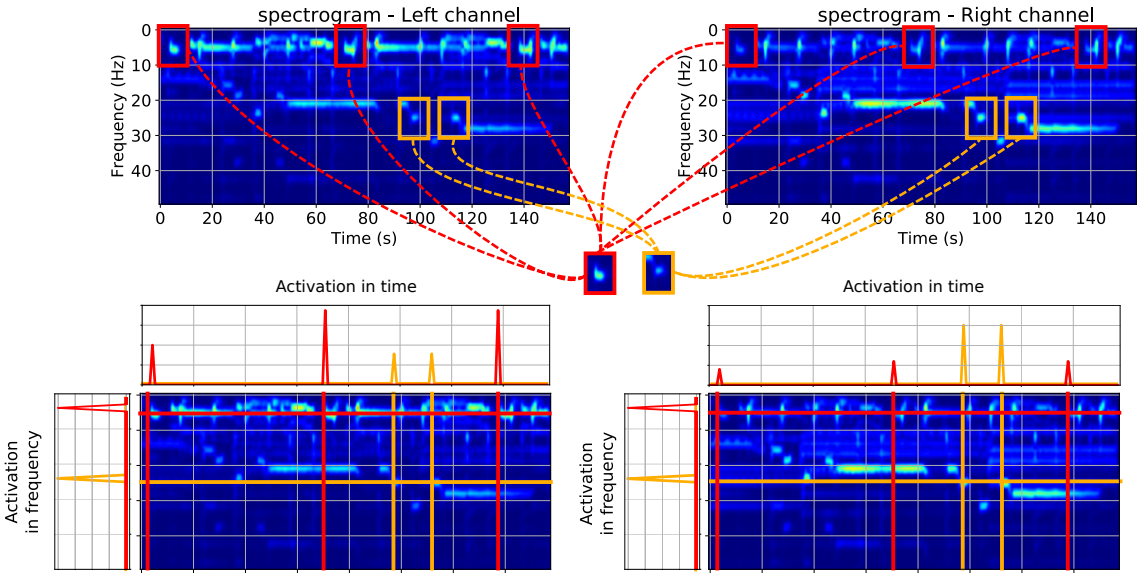


Figure 7.7: Deux spectrogrammes obtenus à partir d'un signal musical stéréo. Certains atomes se répètent (surlignés en rouge et orange) tout en étant visibles sur les deux spectrogrammes. Cette observation suggère que le modèle de CDL est pertinent sur ces données. De plus, une structure de rang faible se retrouve dans les tenseurs d'activations (et non dans les atomes). En d'autres termes, bien que les atomes temps-fréquence sont complexes (et donc sans structure de rang faible), les activations (c'est-à-dire les positions temps/fréquence/canal où ces atomes apparaissent) présentent clairement une structure de rang faible.

vulnérables au bruit et aux perturbations telles que le bruit impulsionnel [Simon and Elad, 2019; Wang et al., 2020].

En prenant en compte ces inconvénients, nous introduisons un modèle CDL tensoriel où les activations et les atomes sont représentés par des tenseurs. Plus précisément, nous proposons de combiner les approches CDL avec une seconde famille de méthodes qui incluent des contraintes de rang CP faible dans leur modélisation. En plus d'ajouter au problème CDL initial une contrainte de rang CP faible pour chaque activation, nous contraignons ces activations à être parcimonieuses. Nous prenons ainsi en compte la structure multivariée des données et obtenons de meilleurs résultats aussi bien en terme de reconstruction que d'interprétabilité. Il est à noter que l'idée d'imposer des contraintes de rang faible dans le CDL n'est pas nouvelle mais est principalement imposée sur le dictionnaire et non sur les activations. Néanmoins, contraindre les activations à être de rang faible apporte deux avantages majeurs. Premièrement, dans de nombreux contextes, la structure de rang faible apparaît naturellement dans les activations plutôt que dans les atomes/le dictionnaire (voir la figure 7.7). Deuxièmement, les contraintes de rang faible sur les activations impliquent une meilleure robustesse au bruit, une des principales faiblesses du problème d'apprentissage d'activations du CDL [Simon and Elad, 2019]. Le succés d'un grand nombre de travaux s'appuyant sur une représentation tensorielle de séries temporelles multivariées (voir par exemple l'importante littérature sur le traitement des signaux EEG [Miwakeichi et al., 2004; Mørup et al., 2006; De Vos et al., 2007; Becker et al., 2010, 2014, 2015; Dauwels et al., 2011; Mørup, 2011; Zhao et al., 2011; Mahyari et al., 2016]) est également source de motivation. Dans ces travaux, les signaux sont généralement analysés en calculant une transformée de Fourier à court terme pour chaque "canal", ce qui donne un tenseur d'ordre 3 espace-temps-fréquence. Ce tenseur est alors étudié à travers le prisme de la décomposition canonique polyadique pour

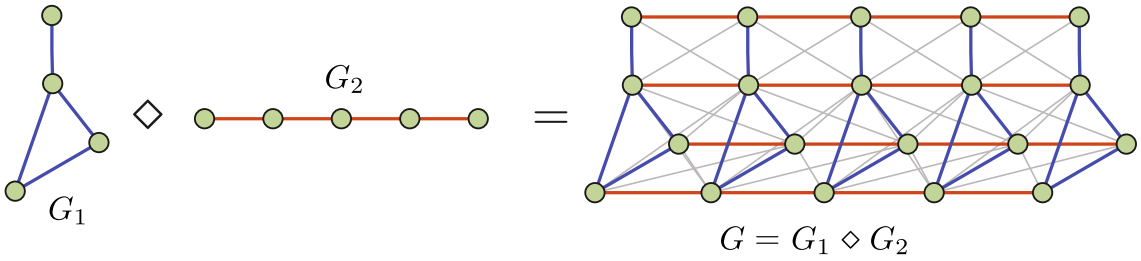


Figure 7.8: Illustration d'un produit de graphe G entre deux graphes G_1 et G_2. $\diamond$ représente un produit Cartésien (seulement les arêtes colorées), un produit de Kronecker (seulement les arêtes grises) ou un "strong product" (toutes les arêtes) entre ces deux graphes. Figure inspirée de [Ortiz-Jiménez et al., 2018].

exploiter les interactions entre les multiples modes. Notons que notre approche est légèrement différent car nous n'appliquons pas directement les décompositions tensorielles aux données. Néanmoins, la combinaison de la représentation CDL avec une contrainte de rang faible aboutit également à des représentations (locales) qui sont (i) plus robustes au bruit et (ii) plus faciles à comprendre [Zhao et al., 2011; Zhou et al., 2013; Cong et al., 2015; Rabusseau and Kadri, 2016].

3.4 Produit de graphes pour l'analyse de signaux multivariés sur graphes

Dans cette quatrième contribution, nous proposons une approche simple pour identifier le support fréquentielle des *signaux multivariés temporels sur graphes*. Ces signaux sont liés à la notion du traitement des *signaux temporels sur graphes* où les interactions spatiales et temporelles sont modélisées [Grassi et al., 2018]. Bien que ce cadre ait été initialement introduit pour les signaux matricielles, nous l'étendons au cas multivarié (par exemple en considérant les relations entre les dimensions, temps, espace, variables). À cette fin, un graphe par dimension est défini. Ces graphes sont alors fusionnés à l'aide d'un *produit de graphes* [Imrich and Klavzar, 2000; Hammack et al., 2011; Leskovec et al., 2010; Sandryhaila and Moura, 2014]. Un exemple est donné par la figure 7.8. Il apparaît que la structure complexe qui en résulte peut être facilement étudiée à travers le formalisme tensoriel. Ainsi, pour identifier le support fréquentiel d'un signal sur graphe, nous choisissons (a priori) un graphe par dimension, puis nous introduisons un problème d'optimisation incluant des régularisations tensorielles adaptées à l'hypothèse de largeur de bande faible. Ces régularisations parcimonieuses peuvent être spécifiées de manière à ne considérer qu'une seule dimension (c'est-à-dire la sélection uniquement des meilleurs nœuds temporels ou canaux ou variables). De plus, en comparant les résultats obtenus avec les graphes choisis a priori à ceux obtenus à partir de graphes aléatoires, nous fournissons un moyen simple d'évaluer leur pertinence. Nous appliquons notre méthode à une représentation tensorielle de signaux EEG en mettant en évidence ses performances pour l'échantillonnage et la compression. Bien que cette contribution se concentre sur les signaux temporels sur graphes, elle peut être appliquée à n'importe quel signal multivarié sur graphe.

3.5 Support décisionnel grâce à l'apprentissage par mimétisme.

Dans cette cinquième et dernière contribution, nous proposons un algorithme qui aide les anesthésistes à administrer les anesthésiques pour maintenir une DoA optimale. Dérivé d'un algorithme appelé *Transform Predictive State Representation* (TPSR) [Littman and Sutton, 2002; Rosencrantz et al., 2004; Boots et al., 2011], notre modèle apprend en observant les anesthésistes dans la pratique. Ce cadre, connu sous le nom d'apprentissage par mimétisme [Abbeel and Ng, 2004], est

particulièrement utile dans le domaine médical car il ne repose pas sur un processus exploratoire – un comportement prohibé dans le cas présent [Gottesman et al., 2018]. Le TPSR est une classe de modèles particulièrement puissante et flexible utilisée dans le domaine de la prédiction séquentielle. L'idée principale de cette classe de modèles est que les données observées sont souvent la manifestation d'une dynamique sous-jacente cachée. En modélisant la structure de transition entre différents états cachés et les probabilités d'occurrence des observations, on peut obtenir un modèle prédictif succinct et puissant. Notons que, bien que les contributions précédentes soient principalement liées à l'analyse des signaux EEG, ici, pour fournir un outil utilisable en pratique par les anesthésistes, nous nous basons uniquement sur les quatre variables couramment surveillées pendant la chirurgie : La fréquence cardiaque, la pression artérielle, la fréquence respiratoire et la concentration expirée d'anesthésique. Ce choix est motivé par le fait que, bien qu'une analyse des signaux EEG soit obligatoire pour comprendre précisément le comportement de l'activité cérébrale, nous pensons qu'un outil pratique devrait être basé uniquement sur des variables physiologiques couramment surveillées et visualisées par les anesthésistes. Cette approche pourrait être d'une grande aide pour les anesthésistes afin de prédire l'évolution des variables et ainsi prévenir les effets secondaires tels que l'hypotension artérielle. En résumé, ce support décisionnel pourrait aider l'anesthésiste à améliorer le soin et la sécurité des patients.

Bibliography

P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, page 1, 2004. 26, 135, 169

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. 57

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. 58, 59

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009. 42

E. Acar, D. M. Dunlavy, and T. G. Kolda. A scalable optimization approach for fitting canonical tensor decompositions. *Journal of Chemometrics*, 25(2):67–86, 2011. 85

A. Agarwal, A. Anandkumar, P. Jain, and P. Netrapalli. Learning sparsely used overcomplete dictionaries via alternating minimization. *SIAM Journal on Optimization*, 26(4):2775–2799, 2016. 83

M. Aharon, M. Elad, and A. Bruckstein. K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006. 82

N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986. 34

T. Alotaiby, F. E. Abd El-Samie, S. A. Alshebeili, and I. Ahmad. A review of channel selection algorithms for eeg signal processing. *EURASIP Journal on Advances in Signal Processing*, 2015 (1):66, 2015. 157

A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research (JMLR)*, 15:2773–2832, 2014. 18, 162

A. Anis, A. Gadde, and A. Ortega. Towards a sampling theorem for signals on arbitrary graphs. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3864–3868. IEEE, 2014. 36, 127

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 41–48, 2006. 41, 155

R. Arora. On learning rotations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 55–63, 2009. 42

M. Arvaneh, C. Guan, K. K. Ang, and C. Quek. Optimizing the channel selection and classification accuracy in eeg-based bci. *IEEE Transactions on Biomedical Engineering (TBME)*, 58(6):1865–1873, 2011. 156

M. Astrid and S.-I. Lee. CP-decomposition with tensor power method for convolutional neural networks compression. In *IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 115–118. IEEE, 2017. 19, 162

M. S. Avidan, L. Zhang, B. A. Burnside, K. J. Finkel, A. C. Searleman, J. A. Selvidge, L. Saager, M. S. Turner, S. Rao, M. Bottros, et al. Anesthesia awareness and the bispectral index. *New England Journal of Medicine (NEJM)*, 358(11):1097–1108, 2008. 22, 165

F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012. 79

O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research (JMLR)*, 9:485–516, 2008. 31

A.-L. Barabasi and Z. N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004. 17, 30, 161

R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010. 128

C. Battaglino, G. Ballard, and T. G. Kolda. A practical randomized CP tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 39(2):876–901, 2018. 92

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. 79

H. Becker, P. Comon, L. Albera, M. Haardt, and I. Merlet. Multiway space-time-wave-vector analysis for source localization and extraction. In *European Signal Processing Conference (EUSIPCO)*, pages 1349–1353. IEEE, 2010. 25, 107, 168

H. Becker, L. Albera, P. Comon, M. Haardt, G. Birot, F. Wendling, M. Gavaret, C. G. Bénar, and I. Merlet. EEG extended source localization: tensor-based vs. conventional methods. *NeuroImage*, 96:143–157, 2014. 25, 107, 168

H. Becker, L. Albera, P. Comon, R. Gribonval, F. Wendling, and I. Merlet. Brain-source imaging: from sparse to tensor models. *IEEE Signal Processing Magazine*, 32(6):100–112, 2015. 18, 25, 107, 162, 168

C. F. Beckmann and S. M. Smith. Tensorial extensions of independent component analysis for multisubject fmri analysis. *Neuroimage*, 25(1):294–311, 2005. 18, 162

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 585–591, 2002. 17, 161

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. 57

J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Transactions on Image Processing*, 26(5): 2466–2479, 2017. 85

A. Bibi and B. Ghanem. High order tensor formulation for convolutional sparse coding. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1772–1780, 2017. 95

N. Biggs, N. L. Biggs, and B. Norman. *Algebraic graph theory*, volume 67. Cambridge University Press, 1993. 34

B. Boots, S. M. Siddiqi, and G. J. Gordon. Closing the learning-planning loop with predictive state representations. *The International Journal of Robotics Research (IJRR)*, 30(7):954–966, 2011. 26, 136, 138, 139, 157, 169

B. Boots, G. Gordon, and A. Gretton. Hilbert space embeddings of predictive state representations. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 92–101, 2013. 152

E. C. Borera, B. L. Moore, and L. D. Pyeatt. Partially observable markov decision process for closed-loop anesthesia control. In *European Conference on Artificial Intelligence (ECAI)*, pages 949–954. IOS Press, 2012. 134

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 43, 54, 55

S. Boyd and L. Vandenberghe. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge University Press, 2018. 43

S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011. 78, 100

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. 57

H. Bristow and S. Lucey. Optimization methods for convolutional sparse coding. *arXiv preprint arXiv:1406.2407*, 2014. 77, 118, 120

H. Bristow, A. Eriksson, and S. Lucey. Fast convolutional sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 391–398, 2013. 76, 81, 94, 96, 97, 98

E. N. Brown, R. Lydic, and N. D. Schiff. General anesthesia, sleep, and coma. *New England Journal of Medicine (NEJM)*, 363(27):2638–2650, 2010. 19, 59, 108, 129, 163

J. Bruhn, P. Myles, R. Sneyd, and M. Struys. Depth of anaesthesia monitoring: what's available, what's validated and what's next? *British Journal of Anaesthesia (BJA)*, 97(1):85–94, 2006. 22

S. Bubeck. Convex optimization: algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015. 46

S. Burer and R. D. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003. 84

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009. 17, 156, 161

E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2): 489–509, 2006. 156

J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319, 1970. 18, 162

R. B. Cattell. "parallel proportional profiles" and other principles for determining the choice of factors by rotation. *Psychometrika*, 9(4):267–283, 1944. 18, 162

R. Chalasani, J. C. Principe, and N. Ramakrishnan. A fast proximal method for convolutional sparse coding. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–5, 2013. 76, 79, 81, 94, 96, 97, 98

J. Chen, C. Richard, and A. H. Sayed. Diffusion LMS over multitask networks. *IEEE Transactions on Signal Processing*, 63(11):2733–2748, 2015a. 17, 161

K. Chen, K.-S. Chan, and N. C. Stenseth. Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):203–221, 2012. 84

S. Chen, A. Sandryhaila, and J. Kovačević. Sampling theory for graph signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3392–3396. IEEE, 2015b. 31, 36, 124, 127

S. Chen, A. Sandryhaila, J. M. F. Moura, and J. Kovačević. Signal recovery on graphs: variation minimization. *IEEE Transactions on Signal Processing*, 63(17):4609–4624, 2015c. 124

S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević. Discrete signal processing on graphs: sampling theory. *IEEE Transactions on Signal Processing*, 63(24):6510–6523, 2015d. 31, 35, 36, 124, 127

S. P. Chepuri, S. Liu, G. Leus, and A. O. Hero. Learning sparse graphs under smoothness prior. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6508–6512. IEEE, 2017. 18, 31, 38, 55, 161

S. Ching, M. Y. Liberman, J. J. Chemali, M. B. Westover, J. D. Kenny, K. Solt, P. L. Purdon, and E. N. Brown. Real-time closed-loop control in a rodent model of medically induced coma using burst suppression. *Anesthesiology*, 119(4):848–860, 2013. 134

F. R. Chung and F. C. Graham. *Spectral graph theory*. Number 92. American Mathematical Society, 1997. 34

A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan. Tensor decompositions for signal processing applications: from two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015. 18, 162

A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, D. P. Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016. 156

A. K. Cline and I. S. Dhillon. Computation of the singular value decomposition. 2006. 46

M. X. Cohen. *Analyzing neural time series data: theory and practice*. MIT press, 2014. 20, 164

F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi. Tensor decomposition of EEG signals: a brief review. *Journal of Neuroscience Methods*, 248:59–69, 2015. 26, 73, 91, 169

E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th National Conference*, pages 157–172. ACM, 1969. 77

M. Daabiss. American society of anaesthesiologists physical status classification. *Indian Journal of Anaesthesia (IJA)*, 55(2), 2011. 142

K. Dadi, M. Rahim, A. Abraham, D. Chyzhyk, M. Milham, B. Thirion, G. Varoquaux, and A. D. N. Initiative. Benchmarking functional connectome-based predictive models for resting-state fMRI. *NeuroImage*, 192:115–134, 2019. 58

S. I. Daitch, J. A. Kelner, and D. A. Spielman. Fitting a graph to vector data. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 201–208, 2009. 18, 31, 36, 161

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 (PNAS)*, 103(37):13848–13853, 2006. 58

C. F. Dantas, J. E. Cohen, and R. Gribonval. Learning fast dictionaries for sparse representations using low-rank tensor decompositions. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 456–466. Springer, 2018. 19, 162

K. C. Das. The Laplacian spectrum of a graph. *Computers & Mathematics with Applications*, 48 (5-6):715–724, 2004. 34

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11): 1413–1457, 2004. 79

I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1): 1–38, 2010. 75

J. Dauwels, K. Srinivasan, R. M. Ramasubba, and A. Cichocki. Multi-channel EEG compression based on matrix and tensor decompositions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 629–632. IEEE, 2011. 25, 168

N. M. M. De Abreu. Old and new results on algebraic connectivity of graphs. *Linear Algebra and its Applications*, 423(1):53–73, 2007. 34

S. De Hert and A. Moerman. Sevoflurane. *F1000Research*, 4(F1000 Faculty Rev), 2015. 143

L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000. 18, 162

J. H. de Morais Goulart. *Estimation of structured tensor models and recovery of low-rank tensors*. PhD thesis, 2016. 156

M. De Vos, L. De Lathauwer, B. Vanrumste, S. Van Huffel, and W. Van Paesschen. Canonical decomposition of ictal scalp EEG and accurate source localisation: Principles and simulation study. *Computational Intelligence and Neuroscience*, 2007, 2007. 25, 168

K. Degraux, U. S. Kamilov, P. T. Boufounos, and D. Liu. Online convolutional dictionary learning for multimodal imaging. In *IEEE International Conference on Image Processing (ICIP)*, pages 1617–1621. IEEE, 2017. 95

S. Diamond and S. Boyd. CVXPY: a Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research (JMLR)*, 17:1–5, 2016. 41, 49

P. M. Djuric and C. Richard, editors. *Cooperative and Graph Signal Processing – Principles and Applications*. Elsevier, 2018. 17, 31, 161

X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst. Learning Laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23):6160–6173, 2016. 36, 38, 47, 48

X. Dong, D. Thanou, M. Rabbat, and P. Frossard. Learning graphs from data: a signal representation perspective. *IEEE Signal Processing Magazine*, 36(3):44–63, 2019. 18, 30, 31, 161, 162

D. Donoho, M. Gavish, et al. Minimax risk of matrix denoising by singular value thresholding. *The Annals of Statistics*, 42(6):2413–2440, 2014. 156

D. L. Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006. 156

D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences (PNAS)*, 100(5):2197–2202, 2003. 75

C. Downey, A. Hefny, and G. Gordon. Practical learning of predictive state representations. *arXiv preprint arXiv.org/pdf/1702.04121*, 2017. 145

N. Du, L. Song, M. Yuan, and A. J. Smola. Learning networks of heterogeneous influence. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2780–2788, 2012. 31

C. Dubost, P. Humbert, A. Benizri, J.-P. Tourtier, N. Vayatis, and P.-P. Vidal. Selection of the best electroencephalogram channel to predict the depth of anesthesia. *Frontiers in Computational Neuroscience*, 13, 2019. 156

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 25th International Conference on Machine Learning (ICML)*, pages 272–279, 2008. 41, 43

G. A. Dumont. Closed-loop control of anesthesia - a review. *IFAC Proceedings Volumes*, 45(18): 373–378, 2012. 134

T. Dupré La Tour, T. Moreau, M. Jas, and A. Gramfort. Multivariate convolutional sparse coding for electromagnetic brain signals. In *Advances in Neural Information Processing Systems (NeurIPS*, pages 3292–3302, 2018. 94, 95

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. 42

H. E. Egilmez, E. Pavez, and A. Ortega. Graph learning from data under structural and Laplacian constraints. *arXiv preprint arXiv:1611.05181*, 2016. 18, 31, 161

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006. 72

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(11):2765–2781, 2013. 17, 161

G. M. Escandar, H. C. Goicoechea, A. M. de la Peña, and A. C. Olivieri. Second-and higher-order data generation and calibration: a tutorial. *Analytica chimica acta*, 806:8–26, 2014. 17

S. M. Fazel. Matrix rank minimization with applications. 2003. 17, 156, 161

C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis. *Neural Computation*, 21(3):793–830, 2009. 72

M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973. 34

M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975. 34

J. Friedman, T. Hastie, H. Höfling, R. Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007. 76

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008. 18, 30, 31, 162

B. A. Fritz, P. L. Kalarickal, H. R. Maybrier, M. R. Muench, D. Dearth, Y. Chen, K. E. Escallier, B. Abdallah, N. Lin, and M. S. Avidan. Intraoperative electroencephalogram suppression predicts postoperative delirium. *Anesthesia & Analgesia*, 122(1):234–242, 2016. 20, 134, 163

J.-J. Fuchs. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, 50(6):1341–1344, 2004. 75

D. Gabay. Chapter ix applications of the method of multipliers to variational inequalities. In *Studies in Mathematics and its Applications*, volume 15, pages 299–331. Elsevier, 1983. 78

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976. 76

S. Gandy, B. Recht, and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011. 18, 156, 162

C. Garcia-Cardona and B. Wohlberg. Convolutional dictionary learning: a comparative review and new algorithms. *IEEE Transactions on Computational Imaging*, 4(3):366–381, 2018a. 17, 23, 72, 82, 93, 160, 167

C. Garcia-Cardona and B. Wohlberg. Convolutional dictionary learning for multi-channel signals. In *52nd Asilomar Conference on Signals, Systems, and Computers*, pages 335–342. IEEE, 2018b. 25, 90, 167

S. Gelman and P. S. Mushlin. Catecholamine-induced changes in the splanchnic circulation affecting systemic hemodynamics. *Anesthesiology*, 100(2):434–439, 2004. 111

A. Gentilini, M. Rossoni-Gerosa, C. W. Frei, R. Wymann, M. Morari, A. M. Zbinden, and T. W. Schnider. Modeling and closed-loop control of hypnosis by means of bispectral index (BIS) with isoflurane. *IEEE Transactions on Biomedical Engineering (TBME)*, 48(8):874–889, 2001. 134

I. George Mychaskiw, M. Horowitz, V. Sachdev, and B. J. Heath. Explicit intraoperative recall at a bispectral index of 47. *Anesthesia & Analgesia*, 92(4):808–809, 2001. 22, 166

N. E. Gibbs, W. G. Poole, Jr, and P. K. Stockmeyer. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal on Numerical Analysis*, 13(2):236–250, 1976. 77

R. Glowinski and A. Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue Française d'Automatique, Informatique, et Recherche Opérationelle*, 9(R2):41–76, 1975. 76

M. Golubovic, D. Stanojevic, M. Lazarevic, V. Peric, T. Kostic, M. Djordjevic, S. Zivic, and D. J. Milic. A risk stratification model for cardiovascular complications during the 3-month period after major elective vascular surgery. *BioMed Research International*, 2018, 2018. 134

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 (JMLR)*, 17:3092–3120, 2016. 31

X. Gong, W. Chen, and J. Chen. A low-rank tensor dictionary learning method for hyperspectral image denoising. *IEEE Transactions on Signal Processing*, 68:1168–1180, 2020. 95

O. Gottesman, F. Johansson, J. Meier, J. Dent, D. Lee, S. Srinivasan, L. Zhang, Y. Ding, D. Wihl, X. Peng, et al. Evaluating reinforcement learning algorithms in observational health settings. *arXiv preprint arXiv:1805.12298*, 2018. 27, 147, 157, 170

O. Gottesman, F. Johansson, M. Komorowski, A. Faisal, D. Sontag, F. Doshi-Velez, and L. A. Celi. Guidelines for reinforcement learning in healthcare. *Nature Medecine*, 25(1):16–18, 2019. 147, 157

J. H. d. M. Goulart and G. Favier. An iterative hard thresholding algorithm with improved convergence for low-rank tensor recovery. In *European Signal Processing Conference (EUSIPCO)*, pages 1701–1705. IEEE. 18, 162

A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, et al. Meg and eeg data analysis with mne-python. *Frontiers in Neuroscience*, 7:267, 2013. 108

L. Grasedyck, D. Kressner, and C. Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013. 18, 162

F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud. A time-vertex signal processing framework: scalable processing and meaningful representations for time-series on graphs. *IEEE Transactions on Signal Processing*, 66(3):817–829, 2018. 26, 124, 131, 169

R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and M. Seibert. Sample complexity of dictionary learning and other matrix factorizations. *IEEE Transactions on Information Theory*, 61(6):3469–3486, 2015. 83

R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariant sparse coding for audio classification. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 149–158. AUAI Press, 2007. 17, 74, 76, 94, 160

W. Guo, I. Kotsia, and I. Patras. Tensor learning for regression. *IEEE Transactions on Image Processing*, 21(2):816–827, 2012. 91

B. D. Haeffele and R. Vidal. Global optimality in tensor factorization, deep learning, and beyond. *arXiv preprint arXiv:1506.07540*, 2015. 156

B. D. Haeffele and R. Vidal. Structured low-rank matrix factorization: Global optimality, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42(6): 1468–1482, 2019. 17, 161

N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011. 139

W. Hamilton, M. M. Fard, and J. Pineau. Efficient learning and planning with compressed predictive states. *Journal of Machine Learning Research (JMLR)*, 15(1):3395–3439, 2014. 27, 138

R. Hammack, W. Imrich, and S. Klavžar. *Handbook of product graphs*. CRC press, 2011. 26, 125, 169

R. A. Harshman. Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis. 1970. 18, 162

S. Hawe, M. Seibert, and M. Kleinsteuber. Separable dictionary learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 438–445, 2013. 19, 162

L. He, K. Chen, W. Xu, J. Zhou, and F. Wang. Boosted sparse and low-rank tensor regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1009–1018, 2018. 19, 73, 84, 86, 156, 162

T. A. Hearn and L. Reichel. Fast computation of convolution operations via low-rank approximation. *Applied Numerical Mathematics*, 75:136–153, 2014. 102

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. 18, 30, 162

A. Hefny, C. Downey, Z. Marinho, W. Sun, S. Srinivasa, and G. J. Gordon. Predictive state models for prediction and control in partially observable environments. In *Conference on Robot Learning (CoRL)*, 2017. 152

F. Heide, W. Heidrich, and G. Wetzstein. Fast and flexible convolutional sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5135–5143, 2015. 81, 94

T. M. Hemmerling and B. Migneault. Falsely increased bispectral index during endoscopic shoulder surgery attributed to interferences with the endoscopic shaver device. *Anesthesia & Analgesia*, 95(6):1678–1679, 2002. 22

P. Herrero, T. M. Rawson, A. Philip, L. S. P. Moore, A. H. Holmes, and P. Georgiou. Closed-loop control for precision antimicrobial delivery: an in silico proof-of-concept. *IEEE Transactions on Biomedical Engineering (TBME)*, 65(10):2231–2236, 2018. 134

B. Hidalgo and M. Goodman. Multivariate or multivariable regression? *American Journal of Public Health*, 103(1):39–40, 2013. 17, 160

C. Hildreth. A quadratic programming procedure. *Naval Research Logistics Quarterly*, 4(1):79–85, 1957. 85

F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927. 18, 162

J. Ho and S. Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4565–4573, 2016. 157

A. Hoorfar and M. Hassani. Approximation of the Lambert W function and hyperpower function. *Research Report Collection*, 10(2), 2007. 70

D. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012. 145

F. Huang and A. Anandkumar. Convolutional dictionary learning through tensor factorization. In *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges*, pages 116–129, 2015. 94

K. Huang and S. Aviyente. Sparse representation for signal classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 609–616, 2007. 72

W. Huang, L. Goldsberry, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro. Graph frequency analysis of brain signals. *IEEE Journal of Selected Topics in Signal Processing*, 10(7):1189–1203, 2016. 32, 34, 126, 156

W. Huang, T. A. Bolton, J. D. Medaglia, D. S. Bassett, A. Ribeiro, and D. Van De Ville. A graph signal processing perspective on functional brain imaging. *Proceedings of the IEEE*, 106(5): 868–885, 2018. 23, 32, 156, 167

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000. 42

W. Imrich and S. Klavzar. *Product graphs: structure and recognition*. Wiley, 2000. 26, 125, 169

C. M. Ionescu, R. De Keyser, B. C. Torrico, T. De Smet, M. M. Struys, and J. E. Normey-Rico. Robust predictive control strategy applied for propofol dosing using BIS as a controlled variable during anesthesia. *IEEE Transactions on Biomedical Engineering (TBME)*, 55(9):2161–2170, 2008. 134

L. Jacob, J.-p. Vert, and F. R. Bach. Clustered multi-task learning: A convex formulation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 745–752, 2009. 155

R. Jain, J. Moura, and C. Kontokosta. Big data+ big cities: graph signals of urban air pollution [exploratory sp]. *Signal Processing Magazine*, 31(5):130–136, 2014. 124

M. R. James and S. Singh. Learning and discovery of predictive state representations in dynamical systems with reset. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004. 138

M. R. James, B. Wolfe, and S. P. Singh. Combining memory and landmarks with predictive state representations. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 734–739, 2005. 138

M. Janzamin, R. Ge, J. Kossaifi, A. Anandkumar, et al. Spectral learning on matrices and tensors. *Foundations and Trends® in Machine Learning*, 12(5-6):393–536, 2019. 18, 162

F. Jiang, X.-Y. Liu, H. Lu, and R. Shen. Efficient multi-dimensional tensor sparse coding using t-linear combination. In *AAAI Conference on Artificial Intelligence*, 2018. 95

S. K. Kadambari and S. P. Chepuri. Learning product graphs from multidomain signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5665–5669. IEEE, 2020. 156

L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: a survey. *Journal of Artificial Intelligence Research (JAIR)*, 4:237–285, 1996. 135

V. Kalofolias. How to learn a graph from smooth signals. In *Proceedings of the Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 920–929, 2016. 18, 31, 36, 38, 161

K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun. Learning convolutional feature hierarchies for visual recognition. In *Advances in Neural Information Processing Systems (NeurIPS*, pages 1090–1098, 2010. 76, 94

M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011. 95

Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin. Compression of deep convolutional neural networks for fast and low power mobile applications. In *International Conference on Learning Representations (ICLR)*, 2016. 19, 162

I. Kissin. Depth of anesthesia and bispectral index monitoring. *Anesthesia & Analgesia*, 90(5): 1114–1117, 2000. 22, 165

J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: a survey. *The International Journal of Robotics Research (IJRR)*, 32(11):1238–1274, 2013. 135

E. D. Kolaczyk and G. Csárdi. *Statistical analysis of network data with R*, volume 65. Springer, 2014. 17, 161

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009. 18, 72, 115, 162

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 17, 31, 161

V. Koltchinskii, K. Lounici, A. B. Tsybakov, et al. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011. 17, 156, 161

B. Kong and C. C. Fowlkes. Fast convolutional sparse coding (FCSC). 94

J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic. Tensorly: tensor learning in Python. *Journal of Machine Learning Research (JMLR)*, 20(1):925–930, 2019. 96

J. B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18(2):95–138, 1977. 117

J. B. Kruskal. Rank, decomposition, and uniqueness for 3-way and n-way arrays. *Multiway Data Analysis*, pages 7–18, 1989. 85

M. Kuderer, S. Gulati, and W. Burgard. Learning driving styles for autonomous vehicles from demonstration. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2641–2646. IEEE, 2015. 135

S. Kumar, J. Ying, J. V. d. Cardoso, D. P. Palomar, et al. Structured graph learning via Laplacian spectral constraints. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 57, 155

S. Kumar, J. Ying, and D. P. Palomar. A unified framework for structured graph learning via spectral constraints. *Journal of Machine Learning Research (JMLR)*, 21(22):1–60, 2020. 57, 155

H.-W. Kuo, Y. Lau, Y. Zhang, and J. Wright. Geometry and symmetry in short-and-sparse deconvolution. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3570–3580. PMLR, 2019. 83

G. Landesberg, M. Mosseri, Y. Wolf, Y. Vesselov, and C. Weissman. Perioperative myocardial ischemia and infarction identification by continuous 12-lead electrocardiogram with online st-segment monitoring. *Anesthesiology*, 96(2):264–270, 2002. 111

Y. Lau, Q. Qu, H.-W. Kuo, P. Zhou, Y. Zhang, and J. Wright. Short and sparse deconvolution-a geometric approach. In *International Conference on Learning Representations (ICLR)*, 2019. 83

V. Lebedev, Y. Ganin, M. Rakhuba, I. Oseledets, and V. Lempitsky. Speeding-up convolutional neural networks using fine-tuned CP-decomposition. In *International Conference on Learning Representations (ICLR)*, 2015. 19, 162

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999. 105

H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 801–808, 2007. 94

D. Lehmann and W. Skrandies. Reference-free identification of components of checkerboard-evoked multichannel potential fields. *Electroencephalography and Clinical Neurophysiology*, 48 (6):609–621, 1980. 59

D. Lehmann, H. Ozaki, and I. Pal. EEG alpha map series: brain micro-states by space-oriented adaptive segmentation. *Electroencephalography and Clinical Neurophysiology*, 67(3):271–288, 1987. 59

J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: an approach to modeling networks. *Journal of Machine Learning Research (JMLR)*, 11(2), 2010. 26, 125, 169

M. S. Lewicki and T. J. Sejnowski. Coding time-varying signals using sparse, shift-invariant representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 730–736, 1999. 17, 74, 94, 160

C. Li and Z. Sun. Evolutionary topology search for tensor network decomposition. 156

X. Li, J. Haupt, and D. Woodruff. Near optimal sketching of low-rank tensor regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3466–3476, 2017. 73, 86

X. Li, D. Xu, H. Zhou, and L. Li. Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences*, 10(3):520–545, 2018. 18, 156, 162

X. Li, Z. Zhu, A. Man-Cho So, and R. Vidal. Nonconvex robust low-rank matrix recovery. *SIAM Journal on Optimization*, 30(1):660–686, 2020. 156

N. Lim, F. d'Alché Buc, C. Auliac, and G. Michailidis. Operator-valued kernel-based vector autoregressive models for network inference. *Machine Learning*, 99(3):489–513, 2015. 18, 162

M. L. Littman and R. S. Sutton. Predictive representations of state. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1555–1561, 2002. 26, 135, 136, 169

J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(1):208–220, 2012. 18, 73, 156, 162

J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(1):208–220, 2013. 91

J. Liu, C. Garcia-Cardona, B. Wohlberg, and W. Yin. Online convolutional dictionary learning. In *IEEE International Conference on Image Processing (ICIP)*, pages 1707–1711. IEEE, 2017. 95

J. Liu, C. Garcia-Cardona, B. Wohlberg, and W. Yin. First-and second-order methods for online convolutional dictionary learning. *SIAM Journal on Imaging Sciences*, 11(2):1589–1628, 2018. 95

N. Liu and J. Rinehart. Closed-loop propofol administration: routine care or a research tool? What impact in the future? *Anesthesia and analgesia*, 122(1):4, 2016. 20, 153, 164

M. A. Lodhi and W. U. Bajwa. Learning product graphs underlying smooth graph signals. *arXiv preprint arXiv:2002.11277*, 2020. 156

P. Lorenzo, S. Barbarossa, and P. Banelli. Sampling and recovery of graph signals. In *Cooperative and Graph Signal Processing*, pages 261–282. Elsevier, 2018. 35

A. Loukas and D. Foucard. Frequency analysis of time-varying graph signals. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 346–350. IEEE, 2016. 124, 126, 131

A. Loukas and N. Perraudin. Stationary time-vertex signal processing. *EURASIP Journal on Advances in Signal Processing*, 2019(1):36, 2019. 124

A. G. Mahyari, D. M. Zoltowski, E. M. Bernat, and S. Aviyente. A tensor decomposition-based approach for detecting dynamic network states from eeg. *IEEE Transactions on Biomedical Engineering (TBME)*, 64(1):225–237, 2016. 25, 168

P.-E. Maingé. Strong convergence of projected subgradient methods for nonsmooth and nonstrictly convex minimization. *Set-Valued Analysis*, 16(7-8):899–912, 2008. 41

J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised dictionary learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1033–1040, 2009. 72

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research (JMLR)*, 11:19–60, 2010. 72, 82, 100

J. Mairal, F. Bach, J. Ponce, et al. Sparse modeling for image and vision processing. *Foundations and Trends® in Computer Graphics and Vision*, 8(2-3):85–283, 2014. 75, 83

H. P. Maretic, M. El Gheche, G. Chierchia, and P. Frossard. GOT: an optimal transport framework for graph comparison. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13876–13887, 2019. 60

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. 35, 36, 124

R. Merris. Laplacian graph eigenvectors. *Linear Algebra and its Applications*, 278(1-3):221–236, 1998. 126

A. F. Merry, J. B. Cooper, O. Soyannwo, I. H. Wilson, and J. H. Eichhorn. International standards for a safe practice of anesthesia 2010. *Canadian Journal of Anesthesia/Journal canadien d'Anesthésie*, 57(11):1027–1034, 2010. 20, 134, 164

G. Meyer. *Geometric optimization algorithms for linear regression on fixed-rank matrices*. PhD thesis, 2011. 42

C. M. Michel and T. Koenig. EEG microstates as a tool for studying the temporal dynamics of whole-brain neuronal networks: a review. *NeuroImage*, 180:577–593, 2018. 59

J. Miettinen, S. A. Vorobyov, and E. Ollila. Blind source separation of graph signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5645–5649. IEEE, 2020. 60

H. Q. Minh, M. Cristani, A. Perina, and V. Murino. A regularized spectral algorithm for hidden markov models with applications in computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2384–2391. IEEE, 2012. 145

A. Mishra, D. K. Dey, and K. Chen. Sequential co-sparse factor regression. *Journal of Computational and Graphical Statistics*, 26(4):814–825, 2017. 84

F. Miwakeichi, E. Martınez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi. Decomposing EEG data into space-time-frequency components using parallel factor analysis. *NeuroImage*, 22(3):1035–1045, 2004. 18, 25, 107, 162, 168

D. Mohan, M. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet. Wavelets on graphs with application to transportation networks. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1707–1712. IEEE, 2014. 124

B. Mohar. Laplace eigenvalues of graphs - a survey. *Discrete Mathematics*, 109(1-3):171–183, 1992. 34

B. L. Moore, L. D. Pyeatt, and A. G. Doufas. Fuzzy control for closed-loop, patient-specific hypnosis in intraoperative patients: a simulation study. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3083–3086. IEEE, 2009. 134

B. L. Moore, A. G. Doufas, and L. D. Pyeatt. Reinforcement learning: a novel method for optimal control of propofol-induced hypnosis. *Anesthesia & Analgesia*, 112(2):360–367, 2011. 134

B. L. Moore, L. D. Pyeatt, V. Kulkarni, P. Panousis, K. Padrez, and A. G. Doufas. Reinforcement learning for closed-loop propofol anesthesia: a study in human volunteers. *Journal of Machine Learning Research (JMLR)*, 15(1):655–696, 2014. 134, 157

T. Moreau, L. Oudre, and N. Vayatis. DICOD: distributed convolutional coordinate descent for convolutional sparse coding. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 3626–3634, 2018. 94, 95

M. Mørup. Applications of tensor (multiway array) factorizations and decompositions in data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):24–40, 2011. 25, 168

M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred. Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG. *NeuroImage*, 29(3):938–947, 2006. 18, 25, 107, 162, 168

M. Mørup, M. N. Schmidt, and L. K. Hansen. Shift invariant sparse coding of image and music data. *Journal of Machine Learning Research (JMLR)*, 5, 2008. 74

M. Moscu, R. Borsoi, and C. Richard. Online graph topology inference with kernels for brain connectivity estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1200–1204. IEEE, 2020. 18, 162

F. Musso, J. Brinkmeyer, A. Mobascher, T. Warbrick, and G. Winterer. Spontaneous brain activity and eeg microstates. a novel eeg/fmri analysis approach to explore resting-state networks. *Neuroimage*, 52(4):1149–1161, 2010. 59

S. Narang, A. Gadde, and A. Ortega. Signal processing techniques for interpolation in graph structured data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5445–5449. IEEE, 2013. 36, 127

R. Nassif, S. Vlaski, C. Richard, and A. H. Sayed. A regularization framework for learning over multitask graphs. *IEEE Signal Processing Letters*, 26(2):297–301, 2018. 18, 161

R. Nassif, S. Vlaski, C. Richard, J. Chen, and A. H. Sayed. Multitask learning over graphs: an approach for distributed, streaming machine learning. *IEEE Signal Processing Magazine*, 37(3): 14–25, 2020. 17, 18, 161

S. Negahban and M. J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, pages 1069–1097, 2011. 17, 156, 161

A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 849–856, 2001. 55, 57

F. Nie, X. Wang, M. I. Jordan, and H. Huang. The constrained Laplacian rank algorithm for graph-based clustering. In *AAAI Conference on Artificial Intelligence*, 2016. 57

M. Nikolova and P. Tan. Alternating proximal gradient descent for nonconvex regularised problems with multiconvex coupling terms. 2017. 85

P. D. O'grady and B. A. Pearlmutter. Convolutive non-negative matrix factorisation with a sparseness constraint. In *IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, pages 427–432, 2006. 105

D. A. O'hara, G. J. Derbyshire, F. J. Overdyk, D. K. Bogen, and B. E. Marshall. Closed-loop infusion of atracurium with four different anesthetic techniques. *Anesthesiology*, 74(2):258–263, 1991. 134

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. 74

B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, 37(23):3311–3325, 1997. 74

A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst. Graph signal processing: overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018. 17, 23, 30, 31, 34, 124, 126, 161, 167

J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. SIAM, 2000. 75, 85

G. Ortiz-Jiménez, M. Coutino, S. P. Chepuri, and G. Leus. Sampling and reconstruction of signals on product graphs. In *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 713–717. IEEE, 2018. 26, 125, 129, 169

G. Ortiz-Jiménez, M. Coutino, S. P. Chepuri, and G. Leus. Sparse sampling for inverse problems with tensors. *IEEE Transactions on Signal Processing*, 67(12):3272–3286, 2019. 129

R. Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349:117–158, 2014. 156

P. Paatero. A weighted non-negative least squares algorithm for three-way 'PARAFAC' factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 38(2):223–242, 1997. 85

V. Papyan, J. Sulam, and M. Elad. Working locally thinking globally-part II: Stability and algorithms for convolutional sparse coding. *arXiv preprint arXiv:1607.02009*, 2016. 83

V. Papyan, Y. Romano, J. Sulam, and M. Elad. Convolutional dictionary learning via local processing. In *IEEE International Conference on Computer Vision (ICCV)*, pages 5296–5304, 2017a. 76, 81, 94, 95

V. Papyan, J. Sulam, and M. Elad. Working locally thinking globally: theoretical guarantees for convolutional sparse coding. *IEEE Transactions on Signal Processing*, 65(21):5687–5701, 2017b. 83

N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3): 127–239, 2014. 90

R. D. Pascual-Marqui, C. M. Michel, and D. Lehmann. Segmentation of brain electrical activity into microstates: model estimation and validation. *IEEE Transactions on Biomedical Engineering (TBME)*, 42(7):658–665, 1995. 59

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. 31, 50

S. Patel and K. L. Goa. Sevoflurane: a review of its pharmacodynamic and pharmacokinetic porperties and its clinical use in general anaesthesia. *Drugs*, 51:658–700, 04 1996. 142

G. Peyré. Sparse modeling of textures. *Journal of Mathematical Imaging and Vision*, 34(1):17–31, 2009. 72

A.-H. Phan, P. Tichavskỳ, and A. Cichocki. Low rank tensor deconvolution. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2169–2173. IEEE, 2015. 85, 96

J. H. Philip, B. K. Philip, and S. Leeson. Gas man version 4. 1 teaches inhalation kinetics. *British Journal of Anaesthesia (BJA)*, 108(suppl_2):215–277, 2012. 146

A. T. Poulsen, A. Pedroni, N. Langer, and L. K. Hansen. Microstate EEGlab toolbox: an introductory guide. *bioRxiv*, (289850), 2018. 59

N. Prasad, L.-F. Cheng, C. Chivers, M. Draugelis, and B. E. Engelhardt. A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017. 134

M. G. Preti, T. A. Bolton, and D. Van De Ville. The dynamic functional connectome: State-of-the-art and perspectives. *Neuroimage*, 160:41–54, 2017. 17, 161

Y. Punjasawadwong, W. Chau-in, M. Laopaiboon, S. Punjasawadwong, and P. Pin-on. Processed electroencephalogram and evoked potential techniques for amelioration of postoperative delirium and cognitive dysfunction following non-cardiac and non-neurosurgical procedures in adults. *Cochrane Database of Systematic Reviews*, (5), 2018. 19, 134, 163

P. L. Purdon, E. T. Pierce, E. A. Mukamel, M. J. Prerau, J. L. Walsh, K. F. K. Wong, A. F. Salazar-Gomez, P. G. Harrell, A. L. Sampson, A. Cimenser, et al. Electroencephalogram signatures of loss and recovery of consciousness from propofol. *Proceedings of the National Academy of Sciences (PNAS)*, 110(12):E1142–E1151, 2013. 19, 21, 107, 109, 129, 163, 164

P. L. Purdon, A. Sampson, K. J. Pavone, and E. N. Brown. Clinical electroencephalography for anesthesiologistspart I: background and basic signatures. *Anesthesiology*, 123(4):937–960, 2015. 20, 21, 164, 165

G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst. Random sampling of bandlimited signals on graphs. *Applied and Computational Harmonic Analysis*, 44(2):446–475, 2018. 35

Q. Qu, X. Li, and Z. Zhu. Exact recovery of multichannel sparse blind deconvolution via gradient descent. 83

Q. Qu, X. Li, and Z. Zhu. Exact and efficient multi-channel sparse blind deconvolution—a nonconvex approach. In *53rd Asilomar Conference on Signals, Systems, and Computers*, pages 640–644. IEEE, 2019a. 83

Q. Qu, X. Li, and Z. Zhu. A nonconvex approach for exact and efficient multichannel sparse blind deconvolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4015–4026, 2019b. 83

J. Quesada, P. Rodriguez, and B. Wohlberg. Separable dictionary learning for convolutional sparse coding via split updates. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4094–4098. IEEE, 2018. 73, 94, 95

J. Quesada, G. Silva, P. Rodriguez, and B. Wohlberg. Combinatorial separable convolutional dictionaries. In *2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*, pages 1–5. IEEE, 2019. 73, 94, 95

G. Rabusseau and H. Kadri. Low-rank regression with tensor responses. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1867–1875, 2016. 19, 26, 73, 91, 156, 162, 169

R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 759–766, 2007. 72

J. Ranieri, A. Chebira, and M. Vetterli. Near-optimal sensor placement for linear inverse problems. *IEEE Transactions on Signal Processing*, 62(5):1135–1146, 2014. 129

H. Rauhut, R. Schneider, and Ž. Stojanac. Low rank tensor recovery via iterative hard thresholding. *Linear Algebra and its Applications*, 523:220–262, 2017. 18, 156, 162

S. Ravishankar and Y. Bresler. Learning sparsifying transforms. *IEEE Transactions on Signal Processing*, 61(5):1072–1086, 2012. 41

B. Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research (JMLR)*, 12(Dec):3413–3430, 2011. 17, 156, 161

E. Richard, P.-A. Savalle, and N. Vayatis. Estimation of simultaneously sparse and low rank matrices. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012. 92

J. Richiardi, S. Achard, H. Bunke, and D. Van De Ville. Machine learning with brain graphs: predictive modeling approaches for functional imaging in systems neuroscience. *IEEE Signal Processing Magazine*, 30(3):58–70, 2013. 17, 30, 156, 161

J. S. Richman and J. R. Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6): H2039–H2049, 2000. 22

R. Rigamonti, A. Sironi, V. Lepetit, and P. Fua. Learning separable filters. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2754–2761, 2013. 73, 94, 95

R. L. Rivest and R. E. Schapire. Diversity-based inference of finite automata. In *28th Annual Symposium on Foundations of Computer Science*, volume 41, pages 555–589. IEEE, 1994. 135, 136

M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 561–568, 2011. 31

A. Rohde, A. B. Tsybakov, et al. Estimation of high-dimensional low-rank matrices. *The Annals of Statistics*, 39(2):887–930, 2011. 17, 156, 161

H. M. Romero-Ugalde, V. Le Rolle, J.-L. Bonnet, C. Henry, P. Mabo, G. Carrault, and A. I. Hernández. Closed-loop vagus nerve stimulation based on state transition models. *IEEE Transactions on Biomedical Engineering (TBME)*, 65(7):1630–1638, 2018. 134

M. Rosencrantz, G. Gordon, and S. Thrun. Learning low dimensional predictive representations. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, 2004. 26, 135, 136, 169

M. R. Rudary and S. P. Singh. A nonlinear predictive state representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 855–862, 2004. 136

A. Sandryhaila and J. M. Moura. Discrete signal processing on graphs. *IEEE Transactions on Signal Processing*, 61(7):1644–1656, 2013. 30, 124

A. Sandryhaila and J. M. Moura. Big data analysis with signal processing on graphs: representation and processing of massive data sets with irregular structure. *IEEE Signal Processing Magazine*, 31(5):80–90, 2014. 26, 124, 125, 169

S. Sanei and J. A. Chambers. *EEG signal processing*. John Wiley & Sons, 2013. 20, 164

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. 23, 31, 35, 38, 46, 49, 50, 155, 167

Y. Sawaguchi, E. Furutani, G. Shirakami, M. Araki, and K. Fukuda. A model-predictive hypnosis control system under total intravenous anesthesia. *IEEE Transactions on Biomedical Engineering (TBME)*, 55(3):874–887, 2008. 134

T. W. Schnider, C. F. Minto, P. L. Gambus, C. Andresen, D. B. Goodale, S. L. Shafer, and E. J. Youngs. The influence of method of administration and covariates on the pharmacokinetics of propofol in adult volunteers. *Anesthesiology*, 88(5):1170–1182, 1998. 136

E. Schwab, B. D. Haeffele, R. Vidal, and N. Charon. Global optimality in separable dictionary learning with applications to the analysis of diffusion MRI. *SIAM Journal on Imaging Sciences*, 12(4):1967–2008, 2019. 19, 162

P. S. Sebel, T. A. Bowdle, M. M. Ghoneim, I. J. Rampil, R. E. Padilla, T. J. Gan, and K. B. Domino. The incidence of awareness during anesthesia: a multicenter united states study. *Anesthesia & Analgesia*, 99(3):833–839, 2004. 20, 164

S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro. Reconstruction of graph signals through percolation from seeding nodes. *IEEE Transactions on Signal Processing*, 64(16):4363–4378, 2016. 34, 126

S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro. Network topology inference from spectral templates. *IEEE Transactions on Signal and Information Processing over Networks*, 3(3):467–483, 2017. 31

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. 59

S. Shafer and D. Stanski. Defining depth of anesthesia. In *Modern Anesthetics*, pages 409–423. Springer, 2008. 20, 163

U. Shalit and G. Chechik. Coordinate-descent for learning orthogonal matrices through givens rotations. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 548–556, 2014. 42

M. Shamaiah, S. Banerjee, and H. Vikalo. Greedy sensor selection: leveraging submodularity. In *IEEE Conference on Decision and Control (CDC)*, pages 2572–2577, 2010. 128

A. Shashua and T. Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 792–799, 2005. 18, 162

J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *The Annals of Mathematical Statistics*, 21(1):124–127, 1950. 82

L. Shi and Y. Chi. Manifold gradient descent solves multi-channel sparse blind deconvolution provably and efficiently. *arXiv preprint arXiv:1911.11167*, 2019. 83

L. Shi and Y. Chi. Manifold gradient descent solves multi-channel sparse blind deconvolution provably and efficiently. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5730–5734. IEEE, 2020. 83

W. Shi, Y. Li, Z. Liu, J. Li, G. Wang, X. Yan, and G. Wang. Non-canonical microstate becomes salient in high density EEG during propofol-induced altered states of consciousness. *International Journal of Neural Systems*, 2020. 60

R. Shouldice, C. Heneghan, P. Nolan, and P. Nolan. PR and PP ECG intervals as indicators of autonomic nervous innervation of the cardiac sinoatrial and atrioventricular nodes. In *First International IEEE EMBS Conference on Neural Engineering, 2003. Conference Proceedings*, pages 261–264. IEEE, 2003. 111

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. 17, 23, 30, 31, 34, 124, 161, 167

N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017. 18, 115, 162

G. Silva, J. Quesada, P. Rodríguez, and B. Wohlberg. Fast convolutional sparse coding with separable filters. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6035–6039. IEEE, 2017. 94, 95

G. Silva, J. Quesada, and P. Rodríguez. Efficient separable filter estimation using rank-1 convolutional dictionary learning. In *IEEE 28th International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2018. 73, 94, 95

D. Simon and M. Elad. Rethinking the csc model for natural images. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2271–2281, 2019. 25, 91, 99, 168

E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607, 1992. 94

S. Singh, M. R. James, and M. R. Rudary. Predictive state representations: a new theory for modeling dynamical systems. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 512–519. AUAI Press, 2004. 137, 152

S. P. Singh, M. L. Littman, N. K. Jong, D. Pardoe, and P. Stone. Learning predictive state representations. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 712–719, 2003. 136

A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua. Learning separable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(1):94–106, 2014. 19, 73, 94, 95, 162

A. Smilde, R. Bro, and P. Geladi. *Multi-way analysis: applications in the chemical sciences*. John Wiley & Sons, 2005. 116

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, A. R. Laird, et al. Correspondence of the brain's functional architecture during activation and rest. *Proceedings of the National Academy of Sciences (PNAS)*, 106(31): 13040–13045, 2009. 58

X. Song and H. Lu. Multilinear regression for embedded feature selection with application to fMRI analysis. In *AAAI Conference on Artificial Intelligence*, 2017. 85

M. Šorel and F. Šroubek. Fast convolutional sparse coding using matrix inversion lemma. *Digital Signal Processing*, 55:44–51, 2016. 94

L. Stanković, M. Daković, and E. Sejdić. Introduction to graph signal processing. In *Vertex-Frequency Analysis of Graph Signals*, pages 3–108. Springer, 2019. 34

R. A. Stevenson, J. J. Schlesinger, and M. T. Wallace. Effects of divided attention and operating room noise on perception of pulse oximeter pitch changesa laboratory study. *Anesthesiology*, 118(2):376–381, 2013. 134

W. W. Sun and L. Li. Store: sparse tensor response regression and neuroimaging analysis. *Journal of Machine Learning Research (JMLR)*, 18(1):4908–4944, 2017. 19, 156, 162

R. Taillefer, M. E Gordon DePuey, J. E. Udelson, G. A. Beller, Y. Latour, and F. Reeves. Comparative diagnostic accuracy of Tl-201 and Tc-99m sestamibi SPECT imaging (perfusion and ECG-gated SPECT) in detecting coronary artery disease in women. *Journal of the American College of Cardiology (JACC)*, 29(1):69–77, 1997. 111

X. Tan, Y. Zhang, S. Tang, J. Shao, F. Wu, and Y. Zhuang. Logistic tensor regression for classification. In *International Conference on Intelligent Science and Intelligent Data Engineering*, pages 573–581. Springer, 2012. 84

Y. Tanaka, Y. C. Eldar, A. Ortega, and G. Cheung. Sampling on graphs: from theory to applications. *arXiv preprint arXiv:2003.03957*, 2020. 35

K.-S. Tang, K. F. Man, G. Chen, and S. Kwong. An optimal fuzzy PID controller. *IEEE Transactions on Industrial Electronics*, 48(4):757–765, 2001. 134

D. A. Tarzanagh and G. Michailidis. Estimation of graphical models through structured norm minimization. *Journal of Machine Learning Research (JMLR)*, 18, 2018. 31

N. V. Thakor and Y.-S. Zhu. Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection. *IEEE Transactions on Biomedical Engineering (TBME)*, 38(8):785–794, 1991. 111

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. 31

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. 16, 77, 160

A. N. Tikhonov. On the solution of ill-posed problems and the method of regularization. In *Doklady Akademii Nauk*, volume 151, pages 501–504. Russian Academy of Sciences, 1963. 16, 160

S. Tong and N. V. Thakor. *Quantitative EEG analysis methods and clinical applications*. Artech House, 2009. 20, 21, 164, 165

J. Townsend, N. Koep, and S. Weichwald. Pymanopt: a Python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research (JMLR)*, 17: 1–5, 2016. 49

L. R. Tucker. Implications of factor analysis of three-way matrices for measurement of change. *Problems in Measuring Change*, 15:122–137, 1963. 18, 162

M. Udell, C. Horn, R. Zadeh, S. Boyd, et al. Generalized low rank models. *Foundations and Trends® in Machine Learning*, 9(1):1–118, 2016. 17, 161

D. Valsesia, G. Fracastoro, and E. Magli. Sampling of graph signals via randomized local aggregations. *IEEE Transactions on Signal and Information Processing over Networks*, 5(2):348–359, 2018. 31

D. Van de Ville, J. Britz, and C. M. Michel. EEG microstate sequences in healthy humans at rest reveal scale-free dynamics. *Proceedings of the National Academy of Sciences (PNAS)*, 107(42): 18179–18184, 2010. 59

F. van Ede, A. J. Quinn, M. W. Woolrich, and A. C. Nobre. Neural oscillations: sustained rhythms or transient burst-events? *Trends in Neurosciences*, 41(7):415–417, 2018. 107

K. Van Steen and G. Molenberghs. Multivariate and multidimensional analysis. *Encyclopedia of Life Support Systems (EOLSS)*. 17, 160

L. Vandenberghe. The CVXOPT linear and quadratic cone program solvers. 2010. 46

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. 57, 58

R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(2):52–68, 2011. 17, 161

R. Vidal, Y. Ma, and S. Sastry. *Generalized Principal Component Analysis*, volume 40. Springer, 2016. 17, 161

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 17, 18, 23, 30, 35, 37, 161, 162, 167

R. Wagner, H. Choi, R. Baraniuk, and V. Delouille. Distributed wavelet transform for irregular sensor network grids. In *IEEE/SP Workshop on Statistical Signal Processing*, pages 1196–1201, 2005. 124

F. Wang, Y. Wang, and G. Cheung. A-optimal sampling and robust reconstruction for graph signals via truncated neumann series. *IEEE Signal Processing Letters*, 25(5):680–684, 2018a. 35

H. Wang and M. Song. Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming. *The R journal*, 3(2), 2011. 143

J. Wang and M. Kolar. Inference for high-dimensional exponential family graphical models. In *Artificial Intelligence and Statistics*, pages 1042–1050, 2016. 31

J. Wang, C. Xu, X. Xie, and J. Kuang. Multichannel audio signal compression based on tensor decomposition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 286–290. IEEE, 2013. 73

Y. Wang, E. Dassau, and I. F. J. Doyle. Closed-loop control of artificial pancreatic β -cell in type 1 diabetes mellitus using model predictive iterative learning control. *IEEE Transactions on Biomedical Engineering (TBME)*, 57(2):211–219, 2010. 134

Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Scalable online convolutional sparse coding. *IEEE Transactions on Image Processing*, 27(10):4850–4859, 2018b. 95

Y. Wang, J. T. Kwok, and L. M. Ni. Generalized convolutional sparse coding with unknown noise. *IEEE Transactions on Image Processing*, 29:5386–5395, 2020. 25, 168

J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander, J. M. Stuart, C. G. A. R. Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113, 2013. 56

E. L. Whitlock, A. J. Villafranca, N. Lin, B. J. Palanca, E. Jacobsohn, K. J. Finkel, L. Zhang, B. A. Burnside, H. A. Kaiser, A. S. Evers, et al. Relationship between bispectral index values and volatile anesthetic concentrations during the maintenance phase of anesthesia in the b-unaware trial. *Anesthesiology*, 115(6):1209–1218, 2011. 22, 166

L. H. William, Y. Rex, and J. Leskovec. Representation learning on graphs: methods and applications. *IEEE Data Engineering Bulletin*, 2017. 17, 30, 161

B. Wohlberg. Efficient convolutional sparse coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7173–7177. IEEE, 2014. 76, 94

B. Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing*, 25(1):301–315, 2015. 23, 76, 81, 82, 90, 93, 94, 96, 100, 167

B. Wohlberg. SPORCO: a Python package for standard and convolutional sparse representations. In *Proceedings of the 15th Python in Science Conference*, pages 1–8, 2017. 96

B. Wolfe, M. R. James, and S. Singh. Learning predictive state representations in dynamical systems without reset. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 980–987, 2005. 138

N. Woodall and T. Cook. National census of airway management techniques used for anaesthesia in the UK: first phase of the fourth national audit project at the royal college of anaesthetists. *British Journal of Anaesthesia (BJA)*, 106(2):266–271, 2010. 134

Y.-H. Wu, N. Charoenphakdee, H. Bao, V. Tangkaratt, and M. Sugiyama. Imitation learning from imperfect demonstration. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 6818–6827, Long Beach, California, USA, 09–15 Jun 2019. PMLR. 157

D. Yang, Z. Ma, and A. Buja. Rate optimal denoising of simultaneously sparse and low rank matrices. *The Journal of Machine Learning Research (JMLR)*, 17(1):3163–3189, 2016. 156

E. Yang, P. Ravikumar, G. I. Allen, and Z. Liu. Graphical models via univariate exponential family distributions. *Journal of Machine Learning Research (JMLR)*, 16:3813–3847, 2015. 31

J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1794–1801. IEEE, 2009. 72

F. Yellin, B. D. Haeffele, and R. Vidal. Blood cell detection and counting in holographic lens-free imaging by convolutional sparse dictionary learning and coding. In *IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 650–653. IEEE, 2017. 82

C. Yu, J. Liu, and S. Nemati. Reinforcement learning in healthcare: a survey. *arXiv preprint arXiv:1908.08796*, 2019. 135, 157

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. 41

M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535, 2010. 76, 81, 94

X. Zhang, J. A. Ashton-Miller, and C. S. Stohler. A closed-loop system for maintaining constant experimental muscle pain in man. *IEEE Transactions on Biomedical Engineering (TBME)*, 40(4): 344–352, 1993. 134

X.-D. Zhang. The Laplacian eigenvalues of graphs: a survey. *arXiv preprint arXiv:1111.2897*, 2011. 34

Y. Zhang, Y. Lau, H.-w. Kuo, S. Cheung, A. Pasupathy, and J. Wright. On the global geometry of sphere-constrained sparse blind deconvolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4894–4902, 2017. 83

Q. Zhao, C. F. Caiafa, D. P. Mandic, L. Zhang, T. Ball, A. Schulze-Bonhage, and A. S. Cichocki. Multilinear subspace regression: an orthogonal tensor decomposition approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1269–1277, 2011. 25, 26, 91, 107, 168, 169

H. Zhou, L. Li, and H. Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013. 18, 26, 72, 86, 91, 162, 169

X. Zhu. *Semi-supervised learning with graphs*. PhD thesis, Doc. Thesis, Carnegie Mellon Univ., 2005. 17, 18, 30, 161

E. Zisselman, J. Sulam, and M. Elad. A local block coordinate descent algorithm for the CSC model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8208–8217, 2019. 76, 81, 94, 95

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006. 17, 42, 67, 161

université PARIS-SACLAY | ÉCOLE DOCTORALE de mathématiques Hadamard (EDMH)

Titre: Tenseurs et graphes pour l'analyse multivariée – application aux neurosciences.

Mots clés: Tenseur, graphe, analyse multivariée, neurosciences

Résumé: Comment extraire l'information contenue dans des données multivariées est devenue une question fondamentale ces dernières années. En effet, leur disponibilité croissante a mis en évidence les limites des modèles standards et la nécessité d'évoluer vers des méthodes plus polyvalentes. L'objectif principal de cette thèse est de fournir des méthodes et des algorithmes prenant en compte la structure des signaux multivariés. Des exemples bien connus de tels signaux sont les images, les signaux audios stéréo, et les signaux d'électroencéphalographie multicanaux. Parmi les approches existantes, nous nous concentrons spécifiquement sur celles basées sur la structure induite par les graphes ou les tenseurs qui ont déjà attiré une attention croissante en raison de leur capacité à mieux exploiter l'aspect multivarié des données et leur structure sous-jacente. Bien que cette thèse prenne l'étude de l'anesthésie générale comme contexte applicatif privilégié, les méthodes développées sont adaptées à un large spectre de données structurées multivariées.

Title: Multivariate analysis with tensors and graphs – application to neuroscience

Keywords: Tensor, graph, multivariate analysis, neuroscience

Abstract: How to extract knowledge from multivariate data has emerged as a fundamental question in recent years. Indeed, their increasing availability has highlighted the limitations of standard models and the need to move towards more versatile methods. The main objective of this thesis is to provide methods and algorithms taking into account the structure of multivariate signals. Well-known examples of such signals are images, stereo audio signals, and multichannel electroencephalography signals. Among the existing approaches, we specifically focus on those based on graph or tensor-induced structure which have already attracted increasing attention because of their ability to better exploit the multivariate aspect of data and their underlying structure. Although this thesis takes the study of patients under general anesthesia as a privileged applicative context, methods developed are also adapted to a wide range of multivariate structured data.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France