Sparse Approximation and Dictionary Learning with Applications to Audio Signals

Daniele Barchiesi

Submitted for the degree of Doctor of Philosophy

Queen Mary, University of London

2013

Sparse Approximation and Dictionary Learning with Applications to Audio Signals

Daniele Barchiesi

Abstract

Over-complete transforms have recently become the focus of a wide wealth of research in signal processing, machine learning, statistics and related fields. Their great modelling flexibility allows to find sparse representations and approximations of data that in turn prove to be very efficient in a wide range of applications. Sparse models express signals as linear combinations of a few basis functions called atoms taken from a so-called dictionary. Finding the optimal dictionary from a set of training signals of a given class is the objective of dictionary learning and the main focus of this thesis. The experimental evidence presented here focuses on the processing of audio signals, and the role of sparse algorithms in audio applications is accordingly highlighted.

The first main contribution of this thesis is the development of a pitch-synchronous transform where the frame-by-frame analysis of audio data is adapted so that each frame analysing periodic signals contains an integer number of periods. This algorithm presents a technique for adapting transform parameters to the audio signal to be analysed, it is shown to improve the sparsity of the representation if compared to a non pitch-synchronous approach and further evaluated in the context of source separation by binary masking.

A second main contribution is the development of a novel model and relative algorithm for dictionary learning of convolved signals, where the observed variables are sparsely approximated by the atoms contained in a convolved dictionary. An algorithm is devised to learn the impulse response applied to the dictionary and experimental results on synthetic data show the superior approximation performance of the proposed method compared to a state-of-the-art dictionary learning algorithm.

Finally, a third main contribution is the development of methods for learning dictionaries that are both well adapted to a training set of data and mutually incoherent. Two novel algorithms namely the incoherent K-SVD and the iterative projections and rotations (IPR) algorithm are introduced and compared to different techniques published in the literature in a sparse approximation context. The IPR algorithm in particular is shown to outperform the benchmark techniques in learning very incoherent dictionaries while maintaining a good signal-to-noise ratio of the representation.

Submitted for the degree of Doctor of Philosophy

Queen Mary, University of London

2013

Contents

List of Figures

List of Algorithms

Acknowledgements

This thesis wouldn't have been written without the support and inspiration of many people. Directly or indirectly, by means of sharp technical insights or through warm words of encouragements, my past three years as a research student have been shaped by a web of personal and professional relationships that I will treasure for my years to come.

My mentors and colleagues at the Centre for Digital Music at Queen Mary University of London have had a profound impact on my journey. Thank you to my supervisor Mark Plumbley who sparked my interest in sparse approximation and supported my research at every stage, encouraging scientific rigour while promoting a relaxed and helping environment. Thank you to the researchers and collaborators whose knowledge and curiosity gave depth and breadth to my own work. Boris Mailhé, Dimitrios Giannoulis, Anssi Klapuri, Ken O'Hanlon, Aris Gretsistas, Maria Jafari, Ivan Damnjanovic, Andrew Nesbit and many others who crossed their paths with mine and left a very bright trace.

My family has been the root and springboard of my achievements. Thank you to my parents Manuela and Enrico, and to all my relatives who have supported me in many ways since I left Italy to become a researcher, despite the fact that most of them are still wondering what am I exactly doing as a job. As an old quote goes, "*you do not really understand something unless you can explain it to your grandmother*", and this has been quite a tough test for my ability to understand dictionary learning.

All the wonderful people who have shared part of their journeys with me during the last three years not only made me a better researcher, but a better person overall. Thank you to my girlfriend May whose love and enthusiasm makes every day brighter. Thank you to my good friends and flatmates Sefki and JohnPaul as I did not think it was possible to have a PhD and so much fun at the same time! Thank you to Mathieu, Mike, Paul, Tim, Fred, Chloé, Pauliñha, Ben, Mel, Sara, Cedric, Federico, Giorgio, and all the people who keep challenging and inspiring me.

Chapter 1

Introduction

1.1 Overview: the big picture

Signal processing essentially consists in extracting meaningful information from data that describe events of interest. Typically, a continuous signal deriving from a process like a sound or an image is sampled in time or space and quantised in magnitude, returning a discrete succession of numbers called *samples*.

The digital samples can be directly used for visualisation or reproduction purposes, but bear little meaning if employed in more sophisticated applications. For example, consider the problems of recognising the type of instrument that plays in a musical recording, removing the noise present in an image, or making a prediction about the future price of a commodity based on its value at times in the past. In all these examples little or nothing can be inferred by looking at the succession of samples. Signal processing acts on digital data *transforming* it in order to provide a *representation* that allow to highlight salient features, separate different components or discern meaningful trends.

The upper plot in Figure 1.1 depicts the samples of the time-domain waveform representing a guitar audio recording and the lower plot shows its time-frequency representation obtained using a Fourier transform[1]. While the former representation can only be used to infer a rough estimate of the amplitude envelope of the sound, the latter provides informa-

[1]The image was obtained using Sonic Visualizer, a tool for audio visualisation and analysis developed at the Centre for Digital Music that can be downloaded from http://www.sonicvisualiser.org/

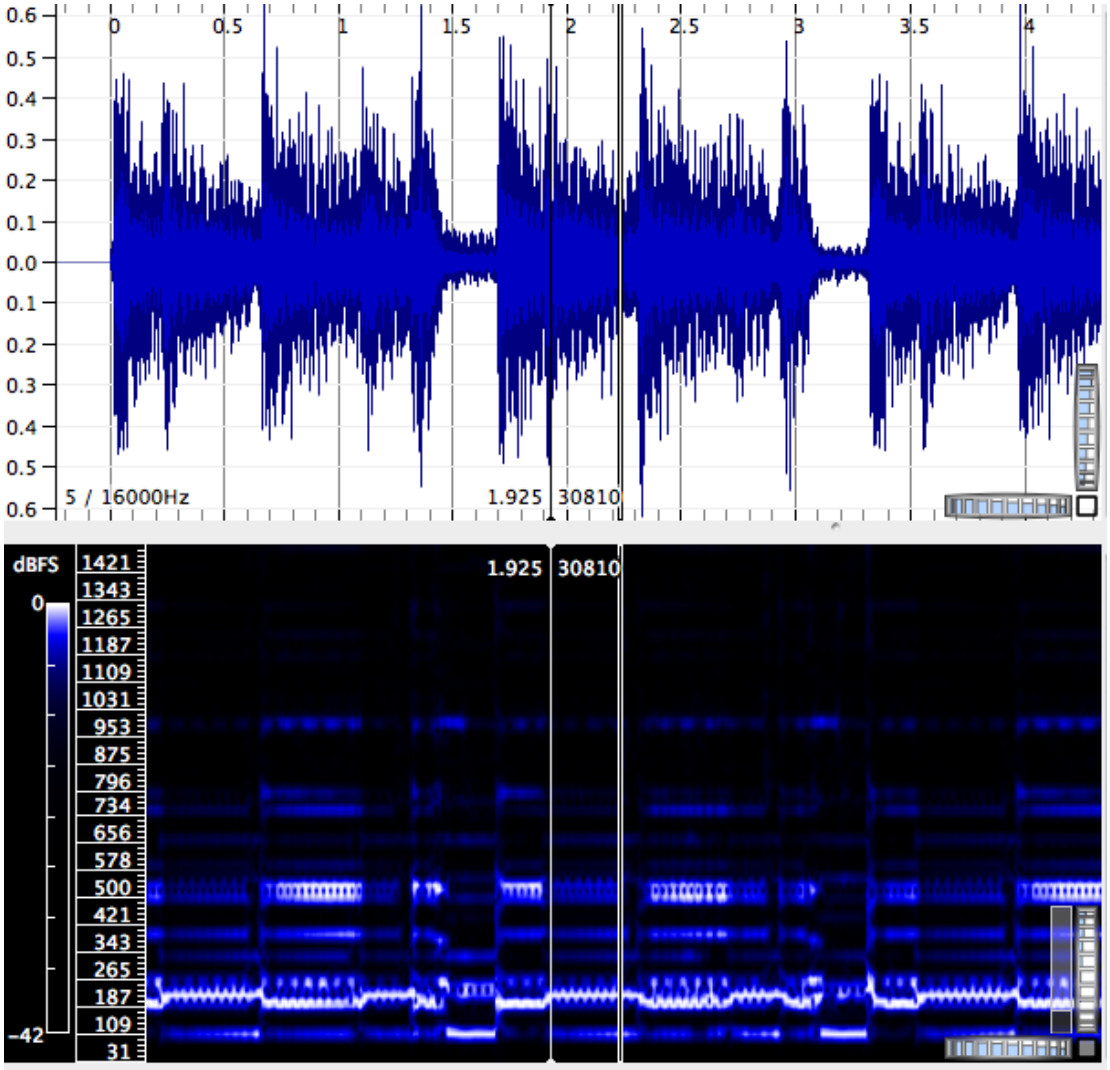


Figure 1.1: Time domain waveform (top) and spectrogram (bottom) of a guitar recording excerpt. In the bottom plot the y axis represent frequencies mapped to a logarithmic scale and the bright areas correspond to regions of the time-frequency plane containing high energy. Note boundaries are visible that are consistent with the envelope that can be inferred by looking at the waveform in the upper plot. In addition, the fundamental frequency and its harmonics are recognisable for each note.

tion about the frequency content of the signal at different times. This is useful for music transcription applications or for inferring the timbre of the instrument and automatically classifying the audio excerpt as played by a guitar.

The representation or approximation of a signal imply the choice of a *dictionary*, that is, a collection of elementary functions called *atoms* that are used to decompose the signal. Any signal that lives in a given space can be represented in an infinite number of ways using different dictionaries provided that the dictionary spans the space, which means that

at least one linear combination of the atoms coincides with the signal to be represented. For example, any two-dimensional vector defined by a pair of (x, y) coordinates can be represented by any dictionary that contains a pair of linearly independent atoms in the two dimensional real space of (x, y) coordinates.

For many decades, *orthonormal* dictionaries have been widely utilised for their mathematical simplicity: in this case, the number of atoms coincides with the dimension of the space containing the signals to be represented, and the transform coefficients are simply computed by calculating inner products. The ubiquitous Fourier transform, the discrete cosine transform, the discrete wavelet transform, the Karhunen-Loève transform derived from the principal component analysis and the class of lapped orthonormal transforms are all examples of orthonormal transforms [70].

More recently, over-complete representations have been investigated for their flexibility and enhanced modelling power. In this case, the dictionary contains a larger number of atoms compared to the dimension of the space containing the signals to be represented, and the representation coefficients are derived using non-linear algorithms. The redundancy introduced by using more atoms than strictly necessary and the enhanced complexity of the algorithms required to compute over-complete representations are often outweighed by the superior adaptivity of this class of transforms to the data to be modelled. A sparse representation or approximation is a transform where the signal is either exactly represented or approximated using only a small number of coefficients with significant magnitude. *Sparsity* is often employed as a measure of adaptivity or modelling power.

The notion of sparsity is deeply rooted in the ubiquitous scientific appeal for conciseness. Sparse approaches have been associated with the principles of parsimony expressed by the famous Occam's razor: competing models of the world should be judged based on the number of assumptions and parameters they require, favouring the ones that provide simple explanation of complex physical phenomena. That is, the models containing a small number of active components.

Continuing with our trivial example, a sparse approximation of vectors in a two-dimensional space only uses one atom and the corresponding coefficient. Over-complete transforms offer an undoubted advantage when seeking sparse approximations of signals,

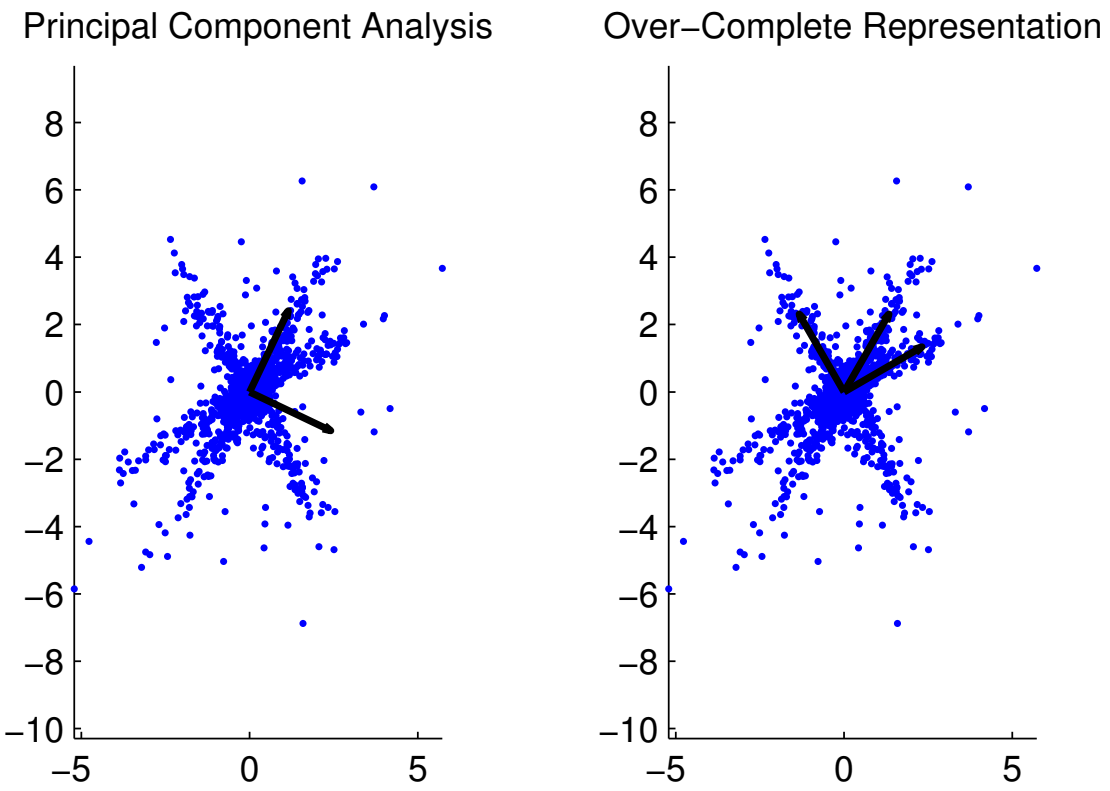


Figure 1.2: Principal component analysis (left) and over-complete dictionary (right) learned on a set of points in a two-dimensional space. PCA learns an orthonormal dictionary whose first atom is oriented in the direction that contains the greater variance in the dataset, but all the points oriented in different directions cannot be well approximated by either one of the atoms. The over-complete dictionary depicted on the right is learned with the objective to provide a sparse approximation of the points in the training set, and the atoms are oriented along the three directions that exhibit most variance in the data.

as illustrated in Figure 1.2 which shows a set of points in the two dimensional real space. The arrows in the left plot represent the atoms of a complete, orthonormal dictionary obtained using principal component analysis (PCA), while the ones in the right plot depict an over-complete dictionary learned from the data in order to provide a sparse approximation. Despite being defined to optimally identify the direction of greater variance within the data, PCA returns a complete dictionary that cannot be aligned with the three directions of prevalent variance and, therefore, cannot lead to a sparse approximation of most of the data. On the contrary, the over-complete dictionary is able to efficiently approximate most of the points using only one of the atoms in the dictionary, and to identify the three directions in the dataset.

Learning over-complete representations for sparse approximation is the objective of

dictionary learning and is the main focus of this thesis. Dictionary learning is an exciting and relatively recent field that has received a great interest in the scientific community [90]. The research endeavours have been devoted to understand the role of dictionaries in sparse problems and the mathematical foundation of sparse representations and approximations, as well as to explore different applications that can greatly benefit from the principles of parsimony and simplicity underlying sparse approaches.

1.2 Thesis structure

The focus of the work presented in this thesis is to present a series of contributions to the field of dictionary learning. Although sparse approximations are used in almost every branch of signal processing, the experimental evidence that will be shown here focuses on the processing of audio signals, and the role of sparse algorithms in audio applications will be accordingly highlighted.

Chapter 2 offers a more formal and thorough background dealing with signal representations. It starts from a description of orthonormal dictionaries and of the class of lapped orthogonal transforms. It then introduces sparse over-complete representations, some of the most popular algorithms used to find sparse representations or approximations and an overview of the methods for dictionary learning that have been proposed in the literature and that are at the basis of most of the main contributions of this thesis.

Chapter 3 is a digression from the main theme of over-complete representations in that it describes a study on different complete transforms to assess their performance for source separation applications. The disjointness of time-frequency representations of simultaneously playing musical instruments is employed as a measure of suitability of a given representation for audio source separation by binary masking. A novel pitch-synchronous lapped orthogonal transform is introduced where the frame-by-frame analysis of audio data is adapted so that each frame analysing periodic signals contains an integer number of periods. Although not strictly regarded as a dictionary learning method, this algorithm presents nonetheless a technique for adapting transform parameters to the audio signal to be analysed and it is shown to improve sparsity of the representation if compared to a non pitch-synchronous approach. The results regarding disjointness, on the other hand, indicate that the modified discrete cosine transform (MDCT) generally

outperforms the proposed pitch-synchronous approach and that sparsity and disjointness are not correlated, an interesting experimental finding that challenges an assumption often made in the source separation literature.

Chapter 4 describes a novel method for dictionary learning of convolved signals. It starts by showing that the sparse approximation of a known sparse signal is greatly degraded if convolution is introduced. From this motivation, a model is proposed where the observed variables are sparsely approximated by the atoms contained in a convolved dictionary, and formulates an algorithm to learn the impulse response applied to the dictionary. Experimental results on synthetic data show the superior approximation performance of the proposed method compared to a state-of-the-art dictionary learning algorithm, and hint at possible applications for de-convolution and source separation.

Chapter 5 deals with learning dictionaries that are both well adapted to a training set of data and mutually incoherent. The mutual coherence is a measure of the similarity between any two different atoms in the dictionary, and learning incoherent dictionaries has been demonstrated to be important for sparse recovery problems. Two novel algorithms, namely the incoherent K-SVD (INK-SVD) and the iterative projections and rotations (IPR) algorithm are introduced and compared to other techniques previously published in the literature. In particular, IPR is applied to the sparse approximation of audio signals and is shown to learn very incoherent dictionaries while maintaining a good signal-to-noise ratio. In addition, experimental evidence is presented in support of the use of incoherent dictionaries for sparse approximation.

Chapter 6 includes a summary of the main topics covered in the thesis.

1.3 Main contributions

This thesis is a report about three years of research on sparse approximation and dictionary learning during which I was lucky to collaborate with other brilliant students and researchers at the Centre for Digital Music at Queen Mary University of London. Some of the topics described in Chapters 3, 4 and 5 are the result of my own work and some others received a substantial contribution from my colleagues. The following list summarises the main contributions of this thesis specifying what parts should be considered to be my own work and what other parts have to be attributed to other researchers.

Pitch-synchronous lapped orthogonal transform : a novel lapped orthogonal transform is described in Section 3.1 that is aimed at analysing a periodic signal using windows containing an integer number of periods. I conceived and implemented the transform, along with the LOTBOX, a Matlab toolbox implementing lapped orthogonal transforms. Dimitrios Giannoulis used the LOTBOX to design and run the experiments on the disjointness of time-frequency representations that are described in Section 3.3.

Dictionary learning of convolved signals : Chapter 4 describes a novel model for dictionary learning of convolved signals and a learning algorithm used to optimize its parameters that was designed and implemented by myself, along with numerical experiments aimed at studying its performance.

Incoherent dictionary learning : Chapter 5 introduces two algorithms for learning dictionaries that are well adapted to a set of training signals and mutually incoherent. The INK-SVD algorithm described in Section 5.3.2 was conceived and implemented by Boris Mailhé, while my contribution consisted in designing and running the numerical experiments presented in Section 5.3.3. The iterative projections and rotations algorithm introduced in Section 5.4 was ideated and implemented by myself, along with the numerical experiments aimed at studying its performance.

1.4 Publications and other deliverables

The following papers have been published or submitted and currently under review in peer reviewed journals and conferences. Most of the results presented in Chapters 3, 4 and 5 are published in these works.

- D. Barchiesi and M. D. Plumbley. Learning Incoherent Dictionaries for Sparse Approximation using Iterative Projections and Rotations. *Submitted and currently under review in the journal "IEEE Transactions on Signal Processing".*

- D. Barchiesi and M. D. Plumbley. Learning Incoherent Dictionaries for Sparse Approximation using Iterative Projections and Rotations. *Sparsity, Dictionaries and Projections in Machine Learning and Signal Processing, ICML Workshop*, June 2012.

- B. Mailhé, D. Barchiesi, and M. D. Plumbley. INK-SVD: Learning incoherent dictionaries for sparse representations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012.

- D. Giannoulis, D. Barchiesi, A. Klapuri, and M. D. Plumbley. On the disjointness of sources in music using different time-frequency representations. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 261–264, October 2011.

- D. Barchiesi and M. D. Plumbley. Dictionary learning of convolved signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5812–5815, May 2011.

- D. Barchiesi and M. D. Plumbley. Dictionary learning of convolved signals. *INSPIRE Network Conference on Information Representation and Estimation*, September 2010.

In addition, the following technical reports and software toolboxes have been produced as part of the research undertaken.

- D. Barchiesi and M. D. Plumbley. Learning Incoherent dictionaries using iterative projections and Lie group optimization. *Technical report n. EECSRR-12-02*, Queen Mary University of London, May 2012.

- D. Barchiesi and M. D. Plumbley. Dictionary Learning of Convolved Signals. *Technical report n. EECSRR-10-04*, Queen Mary University of London, November 2010.

- D. Barchiesi and M. D. Plumbley. Sparse representations for blind deconvolution and source separation. *EECS Postgraduate Conference, Queen Mary University of London*, June 2010. Awarded the "Best Poster Prize".

- D. Barchiesi and M. D. Plumbley. Lapped orthogonal transforms toolbox. Available at http://code.soundsoftware.ac.uk/projects/lots.

- D. Barchiesi and M. D. Plumbley. Incoherent dictionary learning SMALLBox add-on. Available at https://code.soundsoftware.ac.uk/projects/incoherentdl.

Finally, the following publications resulted from projects not related to the scope of the present thesis, and their contributions will not be included in the present work.

- R. Tame, D. Barchiesi and A. Klapuri. Headphone Virtualisation: Improved Localisation and Externalisation of Non- individualised HRTFs by Cluster Analysis. *133rd Convention of the Audio Engineering Society*, May 2012.

- D. Barchiesi and J. Reiss. Reverse Engineering of a Mix. Journal of the Audio Engineering Society, 58(7/8):563-576, July/August 2010.

- D. Barchiesi and J. Reiss. Automatic target mixing using least-squares optimization of gains and equalisation settings. In Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09), pages 7-14, Sep. 2009.

- D. Barchiesi and J. Reiss. Automatic target mixing using genetic optimization of gain and equalisation settings. Digital Music Research Network One-Day Workshop (DMRN+3). Dec. 2008.

1.5 Notation

Table 1.1 indicates the notation adopted in this thesis.

$\boldsymbol{v}$, $\boldsymbol{M}$	Vectors and matrices are indicated by bold lowercase and uppercase letters respectively.
k, K	Scalar values and constants are indicated by lowercase and uppercase letters respectively.
$\boldsymbol{m}_k$	Indicates the vector obtained from the k-th column of the matrix $\boldsymbol{M}$.
$\boldsymbol{m}^k$	Indicates the vector obtained from the k-th row of the matrix $\boldsymbol{M}$.
$\boldsymbol{u} = [\boldsymbol{v}; \boldsymbol{w}]$	Is a vector obtained by the concatenation of vectors $\boldsymbol{u}$ and $\boldsymbol{w}$ along the rows dimension.
$\boldsymbol{A} = [\boldsymbol{B}, \boldsymbol{c}]$	Is a matrix obtained by the concatenation of the matrix $\boldsymbol{B}$ and the vector $\boldsymbol{c}$ along the columns dimension.
Λ	Index sets are indicated by uppercase Greek letters. The restriction of a matrix or vector to the rows (or columns) indexed by a set Λ extends the previous notation and is indicated as $\boldsymbol{M}_\Lambda$ or $\boldsymbol{M}^\Lambda$.
$c^\star$	Indicates the optimal value of the variable c, as returned by an optimization algorithm.
$\hat{\boldsymbol{v}}$	Indicates the Fourier transform of the vector $\boldsymbol{v}$. This notation is extended to matrices where $\hat{\boldsymbol{M}}$ indicates the matrix whose columns are the Fourier transforms of the columns of the matrix $\boldsymbol{M}$.
c^*	Indicates complex conjugate of the variable c.
$(\cdot)^T$, $(\cdot)^{\mathcal{H}}$	Indicates matrix or vector transposition and matrix or vector Hermitian respectively (the latter is a transposition followed by complex conjugation).
$\|\|\boldsymbol{v}\|\|_p$	Indicates the ℓ_p norm of a vector defined as $\|\|\boldsymbol{v}\|\|_p = (\sum_i \|v_i\|^p)^{1/p}$. The limit for $p \to \infty$ is defined as $\|\|\boldsymbol{v}\|\|_\infty = \max_i \|v_i\|$.
$\langle \boldsymbol{v}, \boldsymbol{w} \rangle$	Indicates the inner product between two vectors defined as $\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \sum_{n=1}^{N} v_n^* w_n$.
$\|\|\boldsymbol{M}\|\|_{\mathrm{F}}$	Indicates the Frobenius norm of a matrix defined as $\|\|\boldsymbol{M}\|\|_{\mathrm{F}} = \sqrt{\sum_{i,j} \|m_{ij}\|^2}$.
$\|\|\boldsymbol{M}\|\|_{p,q}$	Indicates the mixed p, q norm of a matrix defined as: $$\|\|\boldsymbol{M}\|\|_{p,q} = \left(\sum_{j=1}^{J} \left(\sum_{i=1}^{I} \|m_{i,j}\|^p \right)^{q/p} \right)^{1/q}.$$ When $p = q$ this norm can be computed by forming a single vector from the elements of $\boldsymbol{M}$ and calculating its norm. In particular, for $p = q = 2$ the matrix mixed norm corresponds to the Frobenius norm.
$\boldsymbol{v} * \boldsymbol{w}$	Indicates the linear convolution of the two vectors defined as $$(\boldsymbol{v} * \boldsymbol{w})[n] = \sum_{i=1}^{I} v[i] w[n-i]$$.

Table 1.1: Notation

Chapter 2

Background

2.1 Bases and dictionaries

Let $\left\{\boldsymbol{\phi}_k \in \mathbb{R}^N\right\}_{k=1}^K$ be a collection of K *atoms* in a space of dimension N. The set of atoms is said to *span* the space and is called a *basis* of $\mathbb{R}^N$ if any signal $\boldsymbol{y} \in \mathbb{R}^N$ can be represented by the following linear combination

$$\boldsymbol{y} = \sum_{k=1}^{K} x_k \boldsymbol{\phi}_k \tag{2.1}$$

where x_k is the coefficient or weight associated with the k-th atom. Note that the space in which the signals live may not necessarily be $\mathbb{R}^N$, but can be any Hilbert space equipped with an inner product $\langle \cdot, \cdot \rangle$ [70]. However, we will restrict our discussion to real or complex signals in finite dimensions unless otherwise specified as this case is relevant to the signal processing applications considered in this work.

Equation (2.1) can be expressed using a compact notation by defining the dictionary matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N \times K}$ as the matrix containing the atoms $\boldsymbol{\phi}_k$ in each one of its columns.

$$\boldsymbol{y} = \boldsymbol{\Phi} \boldsymbol{x} \tag{2.2}$$

where the vector $\boldsymbol{x} \in \mathbb{R}^K$ contains the weights associated to every atom. Equation (2.2) is often referred to as a *synthesis* model in which the signal $\boldsymbol{y}$ is interpreted as synthesised from a finite number of elementary functions, the atoms in the dictionary [30].

2.1.1 Orthonormal dictionaries

An orthonormal dictionary is defined as a set of $K = N$ atoms that satisfy the following property:

$$\langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \tag{2.3}$$

which implies that the squared ℓ_2 norm of the atoms $||\boldsymbol{\phi}_k||_2^2 = \langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_k \rangle = 1$ and that the inner product between any two different atoms is zero. An *orthogonal* dictionary only satisfies the property that different atoms are mutually orthogonal, but does not necessarily contain normalized atoms. Two dictionaries whose atoms have different norms but the same mutual inner products are considered equivalent for representing or approximating a signal through the model (2.2) as the norm differences can be encoded in the magnitude of the coefficients $\boldsymbol{x}$. It is common to work with normalized dictionaries and in the reminder of the thesis we will consider atoms with unit norm unless otherwise specified.

The dictionary matrix $\boldsymbol{\Phi}$ deriving from an orthonormal basis is an orthonormal matrix, so that $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{I}$ is the identity matrix. For orthonormal dictionaries, the coefficients x_k introduced in equation (2.1) are simply calculated by the inner product between the signal $\boldsymbol{y}$ and the atoms:

$$x_k = \langle \boldsymbol{y}, \boldsymbol{\phi}_k \rangle \tag{2.4}$$

In fact, if we express (2.4) in matrix notation as $\boldsymbol{x} = \boldsymbol{\Phi}^T\boldsymbol{y}$, it can be easily shown by substituting in (2.2) that:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{y} = \boldsymbol{I}\boldsymbol{y} = \boldsymbol{y}.$$

Orthonormal transforms include the discrete Fourier transform (DFT), the discrete cosine transform (DCT), the orthonormal discrete wavelet transform (DWT) and the class of lapped orthogonal transforms (LOTs)[1].

When analysing a signal using a dictionary, the coefficients in the transformed domain carry information about the characteristics of the signal based on the properties of the

[1]Here *orthogonal* is used to keep the nomenclature consistent with the literature on the topic, although the dictionaries will be generally assumed to be *orthonormal*.

atoms. For example, the atoms of a N-dimensional DCT-I are defined as [70]:

$$\boldsymbol{\phi}_k[n] = \frac{c_k}{\sqrt{N}} \cos\left[\frac{\pi k}{N}\left(n + \frac{1}{2}\right)\right] \qquad c_k = \begin{cases} 1 & \text{if } k = 0 \\ \sqrt{2} & \text{if } k \neq 0 \end{cases}.$$

Each $\boldsymbol{\phi}_k$ is a cosine function parametrized by the factor $k \in \{0, 1, \ldots, N-1\}$ that determines its frequency, and the inner product between different atoms and the signal to be analysed bears information regarding the activity present at different frequencies. Figure 2.1 shows the time-domain waveform of a 5 seconds piano recording and its DCT transform. It can be seen that most peaks in the transform domain occur between 500Hz and 2000Hz which correspond to a typical range of frequencies present in a piano recording. However, the information resulting from this transform globally pertains to the whole musical excerpt. This occurs because the DCT atoms cover the entire time interval of the musical signal and are only *localised* in frequency (whereas the time-domain representation that represents a signal as a linear combination of Dirac atoms is only localised in time). The Heisenberg uncertainty principle poses a lower bound on the area of the time-frequency plane covered by a given atom, resulting in a trade-off between time and frequency resolution [70].

Had we wanted to achieve a better time resolution and reveal structures such as the note boundaries and harmonic partials visible in Figure 1.1, we would have needed to use atoms that are both localised in time and in frequency. The class of lapped orthogonal transforms provides a framework for constructing orthonormal dictionaries with this type of atoms.

2.1.2 Lapped orthogonal transforms (LOTs)

The simpler way of realizing a globally orthonormal transform by using time-localised functions is dividing the interval $\Gamma \overset{\text{def}}{=} \{1, 2, \ldots, N\}$ into P smaller, disjoint intervals $\gamma_p = \{a_p, \ldots, a_{p+1}\}$, such that the union $\bigcup_{p=1}^{P} \gamma_p = \Gamma$ covers the entire time axis. We can then assign an orthonormal basis $\boldsymbol{\Phi}_p$ such that $\boldsymbol{\Phi}_p \boldsymbol{\Phi}_p^T = \boldsymbol{I}$ locally to each of the intervals and thus define a block orthonormal basis of the space $\mathbb{R}^N$ which can be expressed using

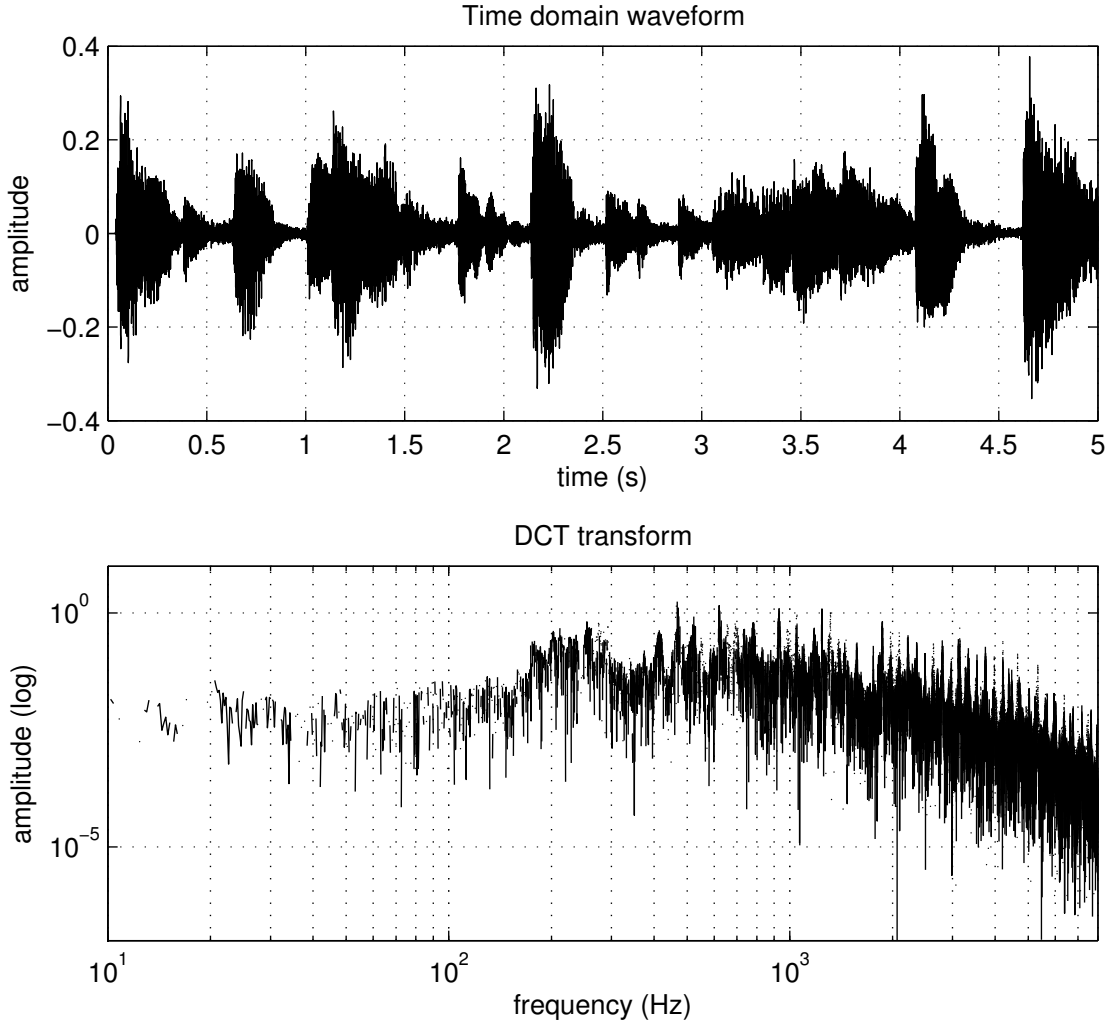


Figure 2.1: Time domain waveform (top) and DCT transform (bottom) of a 5 seconds piano excerpt taken from the RWC database available at http://staff.aist.go.jp/m.goto/RWC-MDB/ (track number 1 of the Jazz Music Database).

a block matrix notation:

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Phi}_P \end{bmatrix} \tag{2.5}$$

It can be trivially shown that $\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{I}$ and, therefore, the dictionary $\boldsymbol{\Phi}$ consists in locally orthonormal bases that are also globally orthonormal. Analysing a signal $\boldsymbol{y}$ with such transform is equivalent to extracting intervals γ_p from $\boldsymbol{y}$ through rectangular windows and applying the respective local transform.

Unfortunately, all periodic transforms (including DFT and DCT) implicitly assume a

periodic extension of the signal to be analysed which creates artificially high frequencies when the value of the signal at the boundaries of each block does not correspond to zero. The spurious coefficients resulting from the windowing effect are misleading in the analysis of the frequency content of the signals and, moreover, lead to a representation that is not *compressible* (i.e., one where the sorted magnitude of the coefficients does not decay slowly), which is inadequate for coding applications.

The lapped orthogonal transform has been introduced in order to partition the signal using smooth, overlapping windows that mitigate the windowing artefacts, while maintaining the local and global orthonormality of the transform [72]. A lapped orthogonal basis of the space $\mathbb{R}^N$ can be written as in equation (2.5), except that consecutive local orthonormal bases are windowed using smooth window boundaries and do overlap. By ensuring that the windows satisfy reconstruction properties and by allowing a maximum overlap of 50% it is possible to ensure that the dictionary $\mathbf{\Phi}$ is globally orthonormal (see also [70] for more details on the theory and algorithms on fast implementations of the LOTs).

Within the framework of LOTs, it is possible to specify different local orthonormal transforms, different partitioning of the signal and overlap between consecutive windows, obtaining a wide range of transforms. The modified cosine transform (MDCT) is a notable example of a LOT that has been widely used in the analysis and coding of audio signals [101, 72, 12]. It is obtained by using type-IV discrete cosine transform (DCT-IV) bases, constant partitioning of the signal and 50% overlap between consecutive windows.

2.2 Over-complete dictionaries

Orthonormal transforms are not the only way to express signals as linear combinations of atoms contained in a dictionary. A dictionary containing atoms $\boldsymbol{\phi}_k \in \mathbb{R}^N$ is said to be *complete* if it spans the space $\mathbb{R}^N$. The following conditions are equivalent in ensuring that the dictionary is complete:

- $\mathbf{\Phi}$ contains N linearly independent atoms.
- The rank of the dictionary $\text{Rank}\,(\mathbf{\Phi}) = N$ equals the size of the space spanned by it.

A full-rank dictionary containing $K > N$ atoms is called *over-complete*. Traditionally, over-complete dictionaries have been analytically designed to provide a signal representation that offers a better time-frequency resolution than what can be achieved by orthonormal transforms. Gabor and wavelet transforms are two notable examples of such over-complete dictionaries.

2.2.1 Gabor and wavelet transforms

The class of Gabor transforms explicitly defines atoms that are localised in time and frequency, providing a representation where the tradeoff between time and frequency resolution can be parametrically adjusted. A discrete Gabor atom is usually indexed with a pair of time and frequency centres $(\tau, \xi) \in \mathbb{Z}$ and is defined as:

$$\boldsymbol{\phi}_{\tau,\xi}[n] = \mathcal{W}[n - \alpha\tau] \exp\left[2\pi i \beta \xi n\right] \tag{2.6}$$

where $\mathcal{W}$ is a windowing function and the parameters α and β control the spacing of the atoms in the time-frequency plane. Figure 2.2 depicts a graphic representation of the time-frequency plane and of a Gabor atom with the relative parameters defining the time-frequency centres of $\boldsymbol{\phi}_{\tau,\xi}$.

Appropriate choices of $\mathcal{W}, \alpha$ and β allow the dictionary to tile and cover the whole time-frequency plane and to provide a complete transform. Generally, the domain of the pair (τ, ξ) is such that the number of coefficients deriving from a Gabor transforms is greater than the number of samples of the signal to be analysed. In this case the Gabor dictionary leads to an over-complete representation that has been often restricted to the class of bi-orthogonal transforms [85], that is, a pair of analysis dictionary $\boldsymbol{\Phi}_a$ and synthesis dictionary $\boldsymbol{\Phi}_s$ that satisfy the following relation:

$$\boldsymbol{\Phi}_s {\boldsymbol{\Phi}_a}^T = \boldsymbol{I}. \tag{2.7}$$

This ensures that, even when working with over-complete representations, the coefficients can be calculated using the inner product $x_{\tau,\xi} = \left\langle \boldsymbol{\phi}_{\tau,\xi}, \boldsymbol{y} \right\rangle$.

Wavelet transforms [70] were proposed to construct non-uniform tilings of the time-frequency plane. The principal idea driving the design of wavelets is that low frequency

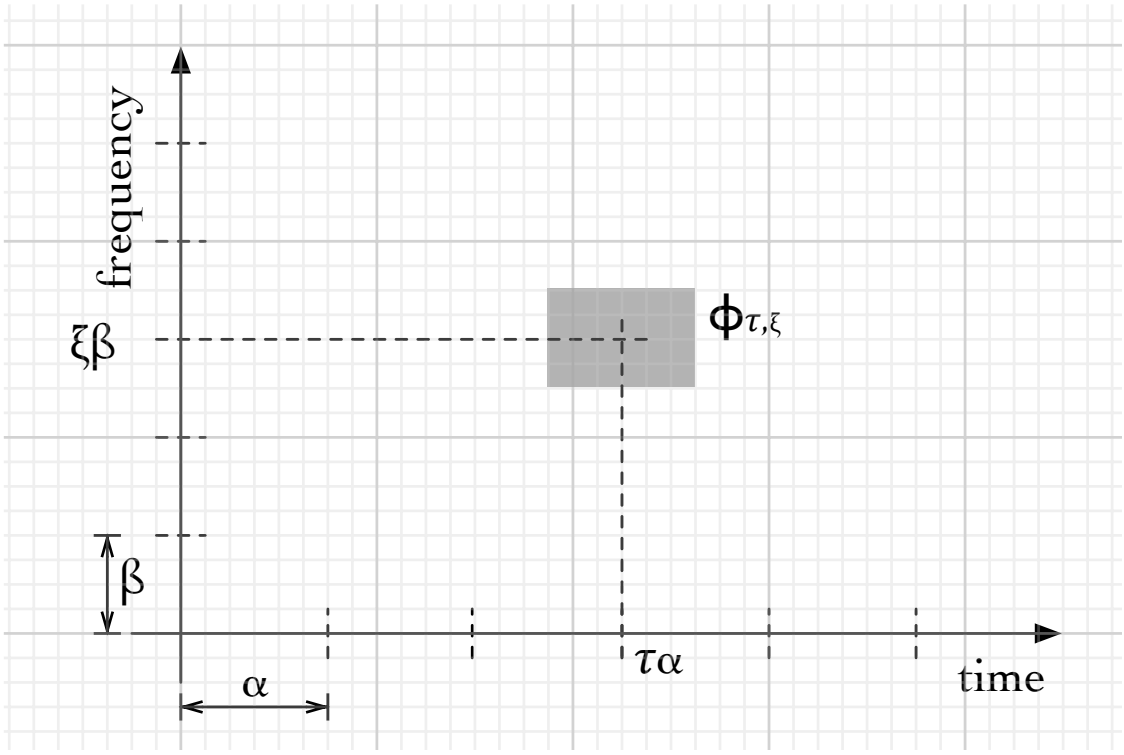


Figure 2.2: Time frequency plane representation of a Gabor atom with time centre loaded at $\tau\alpha$ and frequency centre located at $\xi\beta$. The grey area indicates the tile of the time-frequency plane occupied by the atom and do not necessarily correspond to the support of the function $\boldsymbol{\phi}_{\tau\xi}$ in the time or frequency domain.

signals exhibit long time supports and can be analysed using fine frequency resolutions, whereas high frequency signals are finely localised in time and can be analysed with coarser frequency resolutions. Instead of being defined by their time shifts and frequency modulations as in (2.6), wavelet atoms are shifted and *scaled* versions of a so-called mother wavelet and lead to time-scale representations. Figure 2.3 shows time-frequency tilings of a Gabor transform and of a wavelet transform highlighting the non-uniform partitioning of the time-frequency plane achieved by wavelets.

Generalisations of wavelets such as wavelet packets and cosine trees can be used to partition the time-frequency plane in more adaptive ways [70]. The coefficients of the discrete wavelet transform (DWT) and of its generalisations are computed using fast algorithms that rely on analysis and reconstruction filters, a concept closely related to the analysis and synthesis dictionaries used in bi-orthogonal transforms.

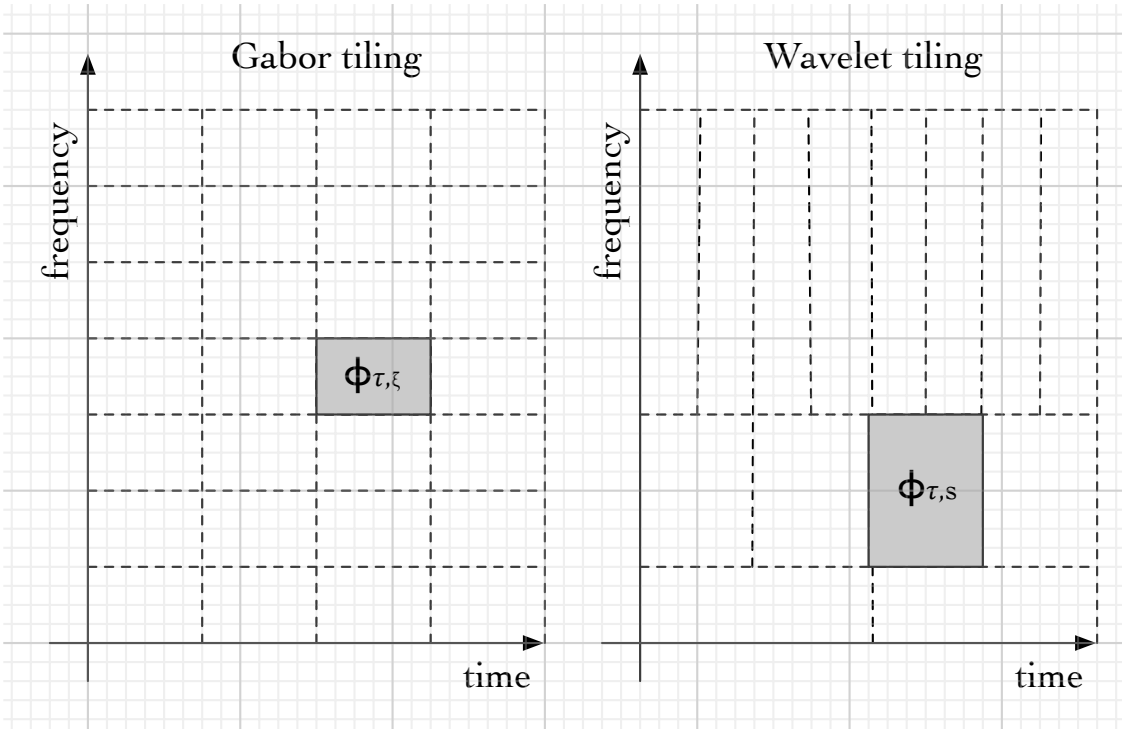


Figure 2.3: Tiling of the time-frequency plane resulting from a Gabor transform (left) and a wavelet transform (right). Gabor atoms $\boldsymbol{\phi}_{\tau,\xi}$ partition the plane uniformly, while wavelet atoms $\boldsymbol{\phi}_{\tau,s}$ employ a time-scale tiling where fine frequency resolution and coarse time resolution is used for low-frequency component while fine time resolution and coarse frequency resolution is used for high-frequency components.

2.3 Sparse models

Gabor and wavelet transforms are only special cases of general over-complete representations. The over-complete *representation* model can be written as is (2.2) with $K > N$:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} \tag{2.8}$$

or relaxed to an over-complete *approximation* model that can be written as

$$\boldsymbol{y} \approx \boldsymbol{\Phi}\boldsymbol{x}. \tag{2.9}$$

Unlike in the case of complete dictionaries, the representation of a signal using an over-complete dictionary is not unique. In fact, given a coefficients vector $\boldsymbol{x}$ such that (2.2) is satisfied, we can construct a second vector $\boldsymbol{x}' = \boldsymbol{x} + \bar{\boldsymbol{x}}$ by choosing $\bar{\boldsymbol{x}} \in \mathcal{N}(\boldsymbol{\Phi})$ in the null-space of the dictionary matrix, so that $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}'$. In other words, an over-complete

representation admits an infinite number of solutions that occupy a space whose dimension equals the dimension of the null-space of $\boldsymbol{\Phi}$, i.e. $K - \text{Rank}\,(\boldsymbol{\Phi})$.

Among all the possible solutions, a *sparse* representation is the one with the smallest number of non-zero coefficients in $\boldsymbol{x}$, and can be defined as a solution of the following optimization problem:

$$\begin{aligned} \boldsymbol{x}^{\star} = \underset{\boldsymbol{x}\in\mathbb{R}^K}{\arg\min}\ ||\boldsymbol{x}||_0 \\ \text{such that } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} \end{aligned} \tag{2.10}$$

where the ℓ_0 pseudo-norm $||\cdot||_0$ counts the number of non-zero coefficients of its argument. A sparse *approximation*, on the other hand, is the natural relaxation of (2.10) where the sparse linear combination is constrained to belong to a so-called ϵ-ball centred around the observed data $\boldsymbol{y}$, that is a region of the space $\mathbb{R}^N$ whose Euclidean distance from the observed signal is not larger than ϵ and quantifies the modelling error:

$$\begin{aligned} \boldsymbol{x}^{\star} = \underset{\boldsymbol{x}\in\mathbb{R}^K}{\arg\min}\ ||\boldsymbol{x}||_0 \\ \text{such that } ||\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}||_2 \leq \epsilon. \end{aligned} \tag{2.11}$$

This error constrained optimization has an alternative formulation, the sparsity constrained sparse approximation, that is defined as seeking the linear combination of S atoms that provides the best approximation in terms of the residual norm of the approximation:

$$\begin{aligned} \boldsymbol{x}^{\star} = \underset{\boldsymbol{x}\in\mathbb{R}^K}{\arg\min}\ ||\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}||_2 \\ \text{such that } ||\boldsymbol{x}||_0 \leq S. \end{aligned} \tag{2.12}$$

It has been shown that a solution to the above problems is NP hard, that is, it cannot be attained by an algorithm in polynomial time [24]. A method to solve (2.10), for example, would search over all possible linear combinations of atoms, starting from approximations that only use one atom and proceeding by increasing the number of active atoms until a combination that exactly represents the signal is found. Algorithms that

follow this strategy can be categorised as *brute force* methods, and require evaluating a large number of possible solutions. Fixing a number of active atoms S, the number of different combinations is $\binom{K}{S} = \frac{K!}{S!(K-S)!}$. Unfortunately, the number given by the binomial expression is computationally impractical for most signal processing tasks that involve dimension of the order of $10^2 - 10^4$.

Searching over all possible linear combinations of S atoms provides a general interpretation of the model (2.9). Given a dictionary containing K atoms, a signal can be approximated by vectors $\tilde{\boldsymbol{y}} = \boldsymbol{\Phi x}$ belonging to the union of all possible subspaces generated by combinations of $S \ll K$ atoms. The notion of union of subspaces can be extended to the *analysis* sparsity model that will be briefly discussed in Section 2.9.1 where the signal is interpreted as *analysed* by a linear operator that promotes co-sparsity (i.e. a large number of zero coefficients in the transformed domain) [31, 44].

2.4 Algorithms for sparse approximations

During the last few decades many algorithms have been proposed and benchmarked in order to approximate the solution to (2.10) in polynomial number of iterations [115] and a significative research effort has been devoted to understand the accuracy of these approximations [92, 113, 111, 114]. The methods can be classified in three different categories: *Greedy* algorithms construct an approximation using generally one atom at a time with the objective of choosing the optimal atom at every step; *convex relaxation* methods rely on the fact that the ℓ_0 pseudo-norm can be approximated by the ℓ_1 norm leading to a convex optimization problem that can be solved in polynomial time; *non-convex optimization* methods are a more recent class of techniques that approximate the ℓ_0 objective with an ℓ_p norm where $0 < p < 1$ leads to non-convex minimisation problems.

2.4.1 Greedy algorithms

The first method appearing in the literature to approximate the solution of (2.10) is the matching pursuit (MP) algorithm proposed by Mallat and Zhang [71]. A signal $\boldsymbol{y}$ is iteratively approximated using the atoms $\boldsymbol{\phi}_k$ contained in the dictionary $\boldsymbol{\Phi}$ through the following steps:

I - Initialise the coefficients as the zero vector $\boldsymbol{x} = \boldsymbol{0}$ and set the residual as $\boldsymbol{r} = \boldsymbol{y}$.

Algorithm 1: Matching pursuit (MP)

Input: $\boldsymbol{y}, \boldsymbol{\Phi}, I, \epsilon$
Output: $\boldsymbol{x}^{\star}$
// Initialisation
1 $i \leftarrow 1$;
2 $\boldsymbol{r} \leftarrow \boldsymbol{y}$;
3 **while** $i \leq I$ *or* $||\boldsymbol{r}||_2 \leq \epsilon$ **do**
// Atom selection
4 $\boldsymbol{c} = \boldsymbol{\Phi}^T \boldsymbol{r}$;
5 $k^{\star} = \arg\max_k |c_k|$;
// Residual update
6 $\boldsymbol{r} \leftarrow \boldsymbol{r} - c_{k^{\star}} \boldsymbol{\phi}_{k^{\star}}$;
7 $i \leftarrow i + 1$;
8 **end**

II - Compute the inner products between the residual and the atoms in the dictionary $c_k = \langle \boldsymbol{r}, \boldsymbol{\phi}_k \rangle$.

III - Select the atom that results in the largest absolute inner product $k^{\star} = \arg\max_{k=\{1,\ldots,K\}} |c_k|$.

IV - Update the residual by subtracting the contribution of the optimal atom $\boldsymbol{r} \leftarrow \boldsymbol{r} - c_{k^{\star}} \boldsymbol{\phi}_{k^{\star}}$.

V - Repeat steps II to IV until a stopping criterion is met.

The *orthogonal* marching pursuit (OMP) [82] has been proposed as an improved greedy algorithm where the atom selection is unchanged but the residual update is performed by projecting the current residual onto the subspace spanned by the atoms selected up to that point. Algorithms 1 and 2 summarise the steps of MP and OMP respectively.

In the OMP algorithm, the set of active indexes Λ is defined and initialised as the empty set. Inner products are calculated between the residual and a sub-dictionary whose atom indexes are restricted to be in Λ^c that is the complement of the active set in line 5 (i.e, only the inner products of unused atoms are evaluated at each step). After selecting the atom exhibiting the larger absolute inner product as in MP and updating Λ to include the chosen index, the residual update is performed by calculating the vector of coefficients $\boldsymbol{x}^{\star}_{\Lambda}$ derived from projecting the signal onto the subspace spanned by the active atoms. This is achieved by computing the Moore-Penrose pseudo-inverse $\boldsymbol{\Phi}^{\dagger}_{\Lambda}$ of the sub-dictionary $\boldsymbol{\Phi}^{\dagger}_{\Lambda} \stackrel{\text{def}}{=} (\boldsymbol{\Phi}_{\Lambda}{}^T \boldsymbol{\Phi}_{\Lambda})^{-1} \boldsymbol{\Phi}_{\Lambda}{}^T$ that contains the active atoms in line 8.

Algorithm 2: Orthogonal matching pursuit (OMP)

Input: $\boldsymbol{y}, \boldsymbol{\Phi}, I, \epsilon$
Output: $\boldsymbol{x}^\star$
// Initialisation
$i \leftarrow 1$;
$\boldsymbol{r} \leftarrow \boldsymbol{y}$;
$\Lambda \leftarrow \emptyset$;
while $i \leq I$ *or* $||\boldsymbol{r}||_2 \leq \epsilon$ **do**
  // Atom and support selection
  $\boldsymbol{c} = (\boldsymbol{\Phi}_{\Lambda^c})^T \boldsymbol{r}$;
  $k^\star = \arg\min_k |c_k|$;
  $\Lambda \leftarrow \Lambda \bigcup k^\star$;
  // Residual update
  $\boldsymbol{x}^\star_\Lambda = \boldsymbol{\Phi}^\dagger_\Lambda \boldsymbol{y}$;
  $\boldsymbol{r} \leftarrow \boldsymbol{y} - \boldsymbol{\Phi}_\Lambda \boldsymbol{x}^\star_\Lambda$;
  $i \leftarrow i + 1$;
end

Unlike in the MP algorithm, in the OMP inner products are computed only for the atoms that do not belong to the active set because the residual at each step is orthogonal to the space spanned by the atoms belonging to the active set. This means that, at each iteration, the inner products $\langle \boldsymbol{r}, \boldsymbol{\phi}_k \rangle = 0 \ \ \forall k \in \Lambda$ and the same atom cannot be selected twice. Moreover, if the dictionary is a basis that spans the space $\mathbb{R}^N$ the algorithm converges to a representation with zero residual error after at most N steps.

The advantage of using the OMP algorithm in terms of convergence properties comes at the computational cost incurred by computing one pseudo-inverse per iteration. This essentially solves the least squares problem $\boldsymbol{x}^\star_\Lambda = \arg\min_{\boldsymbol{x}_\Lambda} ||\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}_\Lambda||_2$.

More recent greedy algorithms have been proposed by modifying and improving the strategy followed by MP and OMP . The regularised orthogonal matching pursuit (ROMP) [77], the compressive sampling matching pursuit (CoSaMP)[76], the subspace pursuit [19] are all examples of recent contributions to the class of greedy algorithms for sparse representation that are oriented to compressive sampling applications and explicitly offer convergence guarantees in terms of the restricted isometry property (see [16] for an overview of compressive sampling, a popular technique for the acquisition of sparse signals based on sparse representations that is briefly described in Section 2.9.1).

Algorithm 3: Iterative hard thresholding (IHT)

Input: $\boldsymbol{y}, \boldsymbol{\Phi}, S, I, \epsilon$
Output: $\boldsymbol{x}^\star$
// Initialisation
$i \leftarrow 1$;
$\boldsymbol{r} \leftarrow \boldsymbol{y}$ **while** $i \leq I$ *or* $||\boldsymbol{r}||_2 \leq \epsilon$ **do**
  // Gradient descent
  $\boldsymbol{c} = \boldsymbol{\Phi}^T \boldsymbol{r}$;
  $\boldsymbol{x} \leftarrow \boldsymbol{x} - \boldsymbol{c}$;
  // Hard thresholding
  $\boldsymbol{x} \leftarrow [\boldsymbol{x}]_\lambda$;
  // Residual update
  $\boldsymbol{r} \leftarrow \boldsymbol{y} - \boldsymbol{\Phi x}$
end

The iterative hard thresholding (IHT) algorithm [8] is an example of a greedy approach to the solution of a penalised problem of the form:

$$\boldsymbol{x}^\star = \underset{\boldsymbol{x} \in \mathbb{R}^K}{\arg\min} \; ||\boldsymbol{y} - \boldsymbol{\Phi x}||_2 + \lambda \, ||\boldsymbol{x}||_0 \, . \tag{2.13}$$

Algorithm 3 summarises its main steps. This strategy resembles a gradient descent optimization because at each step the vector $\boldsymbol{c} = \boldsymbol{\Phi}^T \boldsymbol{r} = \nabla \boldsymbol{x} ||\boldsymbol{y} - \boldsymbol{\Phi x}||_2^2$ is calculated and subtracted to the previous solution. The updated solution is then element-wise hard-thresholded using the thresholding parameter λ:

$$[x_k]_\lambda = \begin{cases} x_k & \text{if } |x_k| \geq \sqrt{\lambda} \\ 0 & \text{if } |x_k| < \sqrt{\lambda} \end{cases} \tag{2.14}$$

and based on the new solution the residual is updated accordingly. Depending on λ one or more new components may enter or leave the active set at any iteration, but generally the solution is forced to be sparse by the thresholding step. The HIT algorithm has been shown to be as reliable as the OMP algorithm in retrieving the representation of a sparse signal from an incomplete set of measurements [8, 93].

2.4.2 Convex relaxation algorithms

A convex relaxation of the problem (2.10) consists in formulating a sparse representation problem as:

$$\boldsymbol{x}^{\star} = \underset{\boldsymbol{x}\in\mathbb{R}^K}{\arg\min} \; ||\boldsymbol{x}||_1 \quad \text{such that } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} \tag{2.15}$$

where the ℓ_1 norm is used in place of the ℓ_0 pseudo-norm as it is the closest convex surrogate function to the original objective of (2.10).

The rationale behind convex relaxation strategies relies on a geometric insight about the space occupied by the infinite number of solutions of an over-complete system of equations. Figure 2.4 depicts the case $K = 2, N = 1$ that can be easily visualised in two dimensions, although the concepts described can be generalised in higher dimensions.

The constraint set of (2.15), that is, the set of solutions of an over-determined system of equations is an affine expression that defines an hyper-plane embedded in the space $\mathbb{R}^K$ occupied by the representation coefficients $\boldsymbol{x}$. In the case of our example, we consider the plane $\mathbb{R}^2$ and a line which corresponds to the set of solutions that satisfy $\boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{y}$. A sparse representation lies on this hyperplane and results in a small number of non-zero coefficients compared to the dimension K. In our case, a sparse representation is one where only one of the two coefficients differs from zero, i.e. the intersection between the line $\boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{y}$ and either one of the axis x_1 or x_2.

The circle and diamond shapes in Figure 2.4 represent contours lines with constant ℓ_2 and ℓ_1 norm respectively, and different outward concentric levels correspond to increasing values of the relative norm. It can be seen from the figure that seeking the solution with the smallest ℓ_1 norm promotes sparsity in that it tends to correspond to the corners of the ℓ_1 contours that intersect with the axis.

The optimization (2.15) has been proposed by Chen et al. as the basis pursuit (BP) algorithm [17] whose goal is to select from an over-complete dictionary the optimal basis that minimises the ℓ_1 norm of the representation coefficients. The BP algorithm turns the optimization (2.15) into a standard linear program by defining an augmented dictionary $\bar{\boldsymbol{\Phi}} \stackrel{\text{def}}{=} [\boldsymbol{\Phi}, -\boldsymbol{\Phi}]$ which include negative copies of the atoms in its columns and solving the

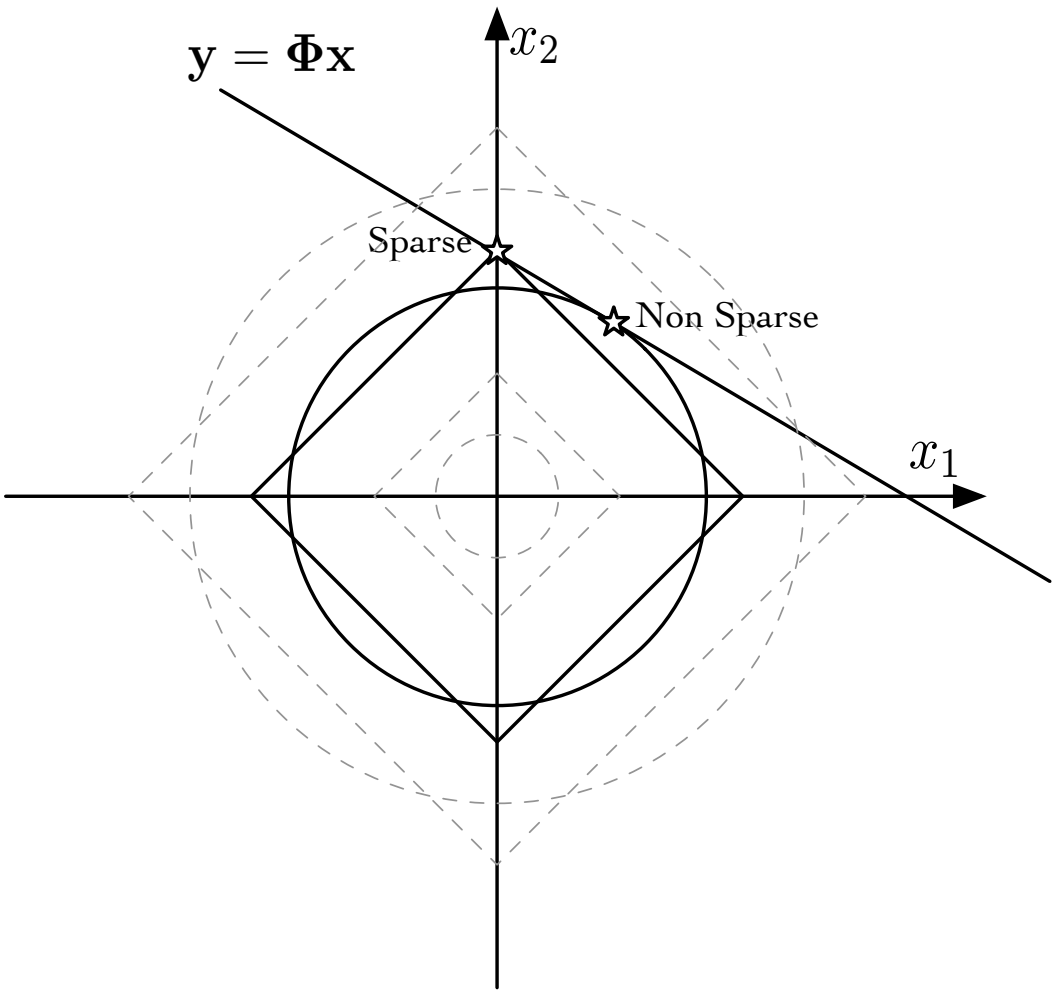


Figure 2.4: Coefficients space and solutions of an under-determined system of equations with $K = 2$ and $N = 1$. The line $\boldsymbol{\Phi x} = \boldsymbol{y}$ represent the set of points that satisfy the representation constraint of (2.10). The minimum ℓ_1 and ℓ_2 solutions are indicated as the intersection of the constraint set with the contours lines representing locations of equal ℓ_1 norm (diamonds) and ℓ_2 norm (circles). The sparse representation is located in correspondence with the minimum ℓ_1 solution unless the constraint set lies on the ℓ_1 contour line, in which case any non-sparse solution would be equivalent to a sparse one in terms of ℓ_1 minimisation.

optimization problem:

$$\begin{aligned} \bar{\boldsymbol{x}}^{\star} = \underset{\bar{\boldsymbol{x}} \in \mathbb{R}^{2K}}{\arg\min}\ & \mathbf{1}^T \bar{\boldsymbol{x}} \\ \text{such that } & \boldsymbol{y} = \bar{\boldsymbol{\Phi}} \bar{\boldsymbol{x}} \\ & \bar{\boldsymbol{x}} \succeq \mathbf{0} \end{aligned} \tag{2.16}$$

where $\bar{\boldsymbol{x}} \in \mathbb{R}^{2K}$ is an augmented coefficients vector whose elements are constrained to be greater than zero (here we used the notation $\succeq$ to indicate element-wise inequality) and $\mathbf{1}$ indicates a vector of ones and is introduced to express the ℓ_1 norm as an inner product $\langle \mathbf{1}, \bar{\boldsymbol{x}} \rangle = ||\bar{\boldsymbol{x}}||_1$. This linear program can be solved with any suitable convex optimization method [11] and results in the optimal $\bar{\boldsymbol{x}}^\star$ that can be easily translated to the solution $\boldsymbol{x}^\star$ of (2.15) by splitting it in two consecutive vectors $\bar{\boldsymbol{x}}^\star = [\boldsymbol{v}^\star; \boldsymbol{u}^\star]$ of length K and subtracting the second vector to the first one $\boldsymbol{x}^\star = \boldsymbol{v}^\star - \boldsymbol{u}^\star$.

The basis pursuit de-noising algorithm (BPD) is a natural generalisation of (2.15) which can be used to approximate a signal that cannot be exactly represented by a linear combination of a few atoms

$$\boldsymbol{x}_\lambda^\star = \underset{\boldsymbol{x} \in \mathbb{R}^K}{\arg\min} \frac{1}{2} ||\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}||_2^2 + \lambda ||\boldsymbol{x}||_1 . \tag{2.17}$$

In this unconstrained minimisation the parameter λ rules the trade-off between the approximation accuracy expressed by the first term of the objective function and the sparsity of the solution promoted by the minimisation of the ℓ_1 norm of the coefficients. Assuming additive Gaussian noise, the parameter λ can be set proportionally to the variance of the noise, so that the noiseless case corresponding to $\lim_{\lambda \to 0} \boldsymbol{x}_\lambda^\star$ coincides with the solution of (2.15). The optimization (2.17) is a quadratic program that can be solved with any standard convex optimization algorithm [11] and has a convenient Bayesian interpretation as the maximum a posteriori (MAP) estimate of the signal under the assumptions that the noise follows a Gaussian distribution and that the coefficients follow a Laplacian distribution.

More precisely, consider the sparse approximation model (2.9) that can be written as $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{n}$ by introducing the noise vector $\boldsymbol{n}$ and suppose that $\boldsymbol{x}$ and $\boldsymbol{n}$ are drawn independently from the respective prior distributions $P_{\boldsymbol{x}}$ and $P_{\boldsymbol{n}}$. The likelihood of observing a signal given that this is generated from some given coefficients is $p(\boldsymbol{y}|\boldsymbol{x}) = P_{\boldsymbol{n}}(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x})$ and the prior probability of observing the coefficients is $p(\boldsymbol{x}) = P_{\boldsymbol{x}}(\boldsymbol{x})$. The Bayes rule

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})} \tag{2.18}$$

linking the *posterior* conditional probability of the coefficients given that we observe a

given signal to the likelihood and the prior probability of the coefficients allows to define the maximum-a-posteriori solution as:

$$\begin{aligned} \boldsymbol{x}^{\star}_{\text{MAP}} &= \arg\max_{\boldsymbol{x}\in\mathbb{R}^K} p(\boldsymbol{x}|\boldsymbol{y}) \\ &= \arg\min_{\boldsymbol{x}\in\mathbb{R}^K} -\log p(\boldsymbol{x}|\boldsymbol{y}) \\ &= \arg\min_{\boldsymbol{x}\in\mathbb{R}^K} -\log p(\boldsymbol{y}|\boldsymbol{x}) - \log p(\boldsymbol{x}). \end{aligned} \tag{2.19}$$

Assuming a Gaussian noise prior $P_{\boldsymbol{n}}(\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}) \propto \exp\left(-||\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}||_2^2/2\right)$, a Laplacian prior on the coefficients $P_{\boldsymbol{x}}(\boldsymbol{x}) \propto \exp\left(-\lambda\sum_k |x_k|\right)$ and substituting into (2.19) we obtain

$$\boldsymbol{x}^{\star}_{\text{MAP}} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^K} \frac{1}{2}||\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}||_2^2 + \lambda||\boldsymbol{x}||_1 \tag{2.20}$$

that coincides with the BPD formulation (2.17). It is worth noting that this MAP solution is not the only valid Bayesian interpretation of penalised least-squares problems, as shown by Gribonval [43].

Another strategy for defining and solving a convex relaxation of the sparse representation problem is the least absolute shrinkage and selection operator (LASSO) algorithm proposed by Tibshirani [110] and further developed by Osborne et al. [80]. The optimization problem is formulated as follows:

$$\begin{aligned} \boldsymbol{x}^{\star}_{\kappa} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^K} \; & ||\boldsymbol{y}-\boldsymbol{\Phi}\boldsymbol{x}||_2^2 \\ \text{such that } \; & ||\boldsymbol{x}||_1 \leq \kappa \end{aligned} \tag{2.21}$$

where the parameter κ controls the level of sparsity of the solution.

Figure 2.5 offers a geometric interpretation of the LASSO algorithm that is worth comparing to Figure 2.4. Here $\boldsymbol{x}^{\star}_K$ indicates a large value of the parameter κ that satisfies $\boldsymbol{y} = \boldsymbol{\Phi}x^{\star}_K$, i.e. a solution for which the value of the objective function in (2.21) is zero. For lower values of κ, the solution $\boldsymbol{x}^{\star}_{\kappa}$ is the intersection between the constraint set (that is, the shaded area delimited by the contours plot of the ℓ_1 norm) and the contours plot of the quadratic cost function of (2.21) depicted as dashed ellipsoids. Once again we can see that the ℓ_1 norm promotes sparsity, while the ℓ_2 solution would be one where both

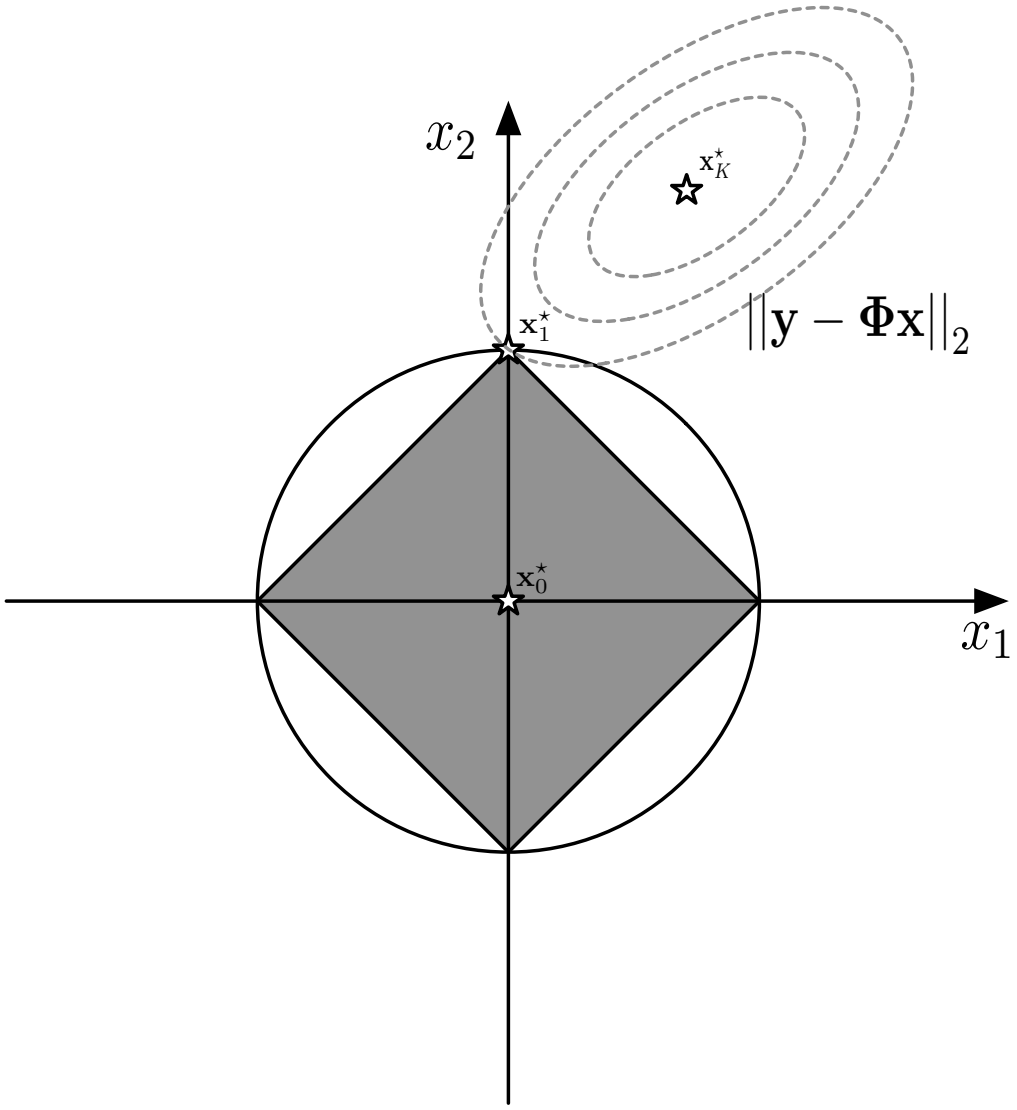


Figure 2.5: Geometric interpretation of the LASSO algorithm. The solution $\boldsymbol{x}_\kappa^\star$ lies on the intersection between the constraint set bounded by the ℓ_1 norm ball and the contours levels representing increasing values of the quadratic function that is the objective of (2.21). The homotopy algorithm finds the solution of (2.21) following a greedy strategy that starts from $\boldsymbol{x}_0^\star = \mathbf{0}$ and follows the solution path resulting from increasing κ.

the components x_1 and x_2 are active.

The *homotopy* algorithm [80] introduced to solve (2.21) is an iterative method that starts form the solution $\boldsymbol{x}_0^\star = \mathbf{0}$ and traces a solution path which follows increasing values of the parameter κ until the desired constraint is reached. Enlarging the feasible set by increasing the value of κ causes new atoms to enter the active set Λ and may result in

other atoms to exit it. The least angle regression (LARS) algorithm proposed by Efron et al. [29] is a simplification of homotopy where atoms are only allowed to enter the active set every time the solution gets updated.

Drori and Donoho [28] reviewed the ℓ_1 minimisation algorithms and built theoretical and empirical phase-transition diagrams which show for what values of sparsity and dictionary redundancy the various algorithms are able to recover a sparse solution to an under-determined system of equations. They use the parameter $\sigma = N/K$ which measures the redundancy of the dictionary as the ratio between the dimension of the signals and the number of atoms and the sparsity level $\rho = S/N$ as the ratio between the number of active atoms and the dimension of the signals. As σ decreases implying more redundant dictionaries, the sparsity level ρ must also decrease to guarantee correct recovery. For appropriate values of σ and ρ, LASSO, LARS and OMP succeed in recovering an S-sparse solution in S steps. Moreover, Donoho and Tsaig show [26] how the LARS algorithm bridges the gap between ℓ_1 optimization algorithms and greedy methods such as OMP by iteratively finding a solution of (2.21) adding one atom to the active set at each step.

2.5 Applications of sparse over-complete models

This section gives an overview of some of the algorithms that appeared in the literature during the last decade which make use of sparse representations or approximations to solve typical signal processing problems. The research on sparse methods and the advances in the related field of compressed sensing are very popular fields with contributions regularly appearing from a vast research community (see, for example, the blog Nuit Blanche[2] for almost-daily updates on the latest events and publications). Therefore, this is by no means an exhaustive list of contributions, but rather a partial account focused on applications to audio signals analysis and processing.

Audio coding and de-noising

Sparse representations are by definition suitable for audio coding where the goal is to represent (in the case of lossless coding) or approximate (in the case of lossy coding) an audio signal using the smallest possible bit-rate, that is, the smallest amount of information or amount of significant coefficients per second.

[2] http://nuit-blanche.blogspot.co.uk/

Davies and Daudet [23] devise a modulated complex lapped transform (MCLT) that is a generalisation of lapped orthogonal transforms for the coding of audio signals and suggest a multi-resolution analysis where an over-complete sparse approximation is used for audio processing. The authors also define an iteratively reweighed least squares algorithm for coding audio signals using the proposed MCLT. Ravelli et al. [88] suggest instead an union of 8 MDCT transforms for audio coding and shows through a comparison with state-of-the-art algorithms the superior performance of sparse over-complete approximations at low bit-rates.

Audio signals often display regular harmonic structures and recurring patterns both in time and frequency. Starting from this motivation, Daudet [22] introduced the molecular matching pursuit algorithm where molecular structures are defined as groups of atoms that occur together in a time-frequency representation. The distinction between tonal and transient molecules makes the algorithm suitable for audio analysis and coding. Extending this work, Leveau et al. [59] suggest to employ instrument specific harmonic molecules for the representation of audio signals. These are grouped in successive time frames and also used for polyphonic instrument recognition.

The work by Vincent and Plumbley [117] follows a similar rationale in proposing a Bayesian probabilistic model to represent audio signals at very low bit-rates using note-like representations consisting of harmonic partials. The resulting so-called object coding represents an ambitious goal that blurs the boundaries between coding and transcription, another challenging application in audio signal processing. Coding is also intimately linked with de-noising, as sparse representations that capture salient features of audio signals through significant coefficients are also likely to discard any additive noise in the set of non-significant coefficients. This is the rationale behind the basis pursuit de-noising algorithm [17] and of more recent algorithms specifically tailored to audio signals [37].

Apart from designing atoms that are specifically tailored to the representation of audio signals, advances in coding applications can be pursued by studying the distribution of sparse approximation coefficients. Kowalski and Torrésani [57] propose a probabilistic model of the analysis coefficients resulting from the inner product between the atoms in a dictionary made of a union of bases and the signal to be coded. The coefficients are further classified in significant and not significant components and this distinction proved

to be useful for de-noising applications.

In the area of speech processing, a popular technique is the linear predictive coding (LPC) where current samples of speech are expressed as linear combination of past samples in an auto-regressive model. Giacobello et al. [39] propose a reweighed ℓ_1 algorithm for LPC of speech that leads to a sparse residual in the time domain, which can be in turn sparsely coded [40].

Audio restoration

Coding is not the only application of sparse approximation for the processing of audio signals. The audio in-painting framework introduced by Adler et al. [3] by analogy with the perhaps better known image in-painting includes several problems in audio restoration such as bandwidth extension, packet loss, de-clipping and impulsive noise removal which are tackled using an unified model. The signal is decomposed using a frame based sparse time-frequency transform (usually a discrete cosine or discrete Gabor dictionary), the locations of un-reliable data are assumed to be known and the audio is restored in every frame by solving an inverse problem using the orthogonal matching pursuit algorithm. The results obtained in terms of signal-to-noise-ratio are comparable to state-of-the-art algorithms and commercial software. As a particular example of audio in-painting, Moussallam et al. [74] employ a decomposition with full-bandwidth atoms on a signal with reduced bandwidth for bandwidth extension.

Source separation

Source separation is another application where sparse models have been used extensively. Its formal definition is given in Section 3.3.1, along with distinctions between different categories of source separation problems. Generally speaking, the goal of this class of applications is to extract a set of unknown source signals from a set of mixtures. For example, the so-called *cocktail party* problem consists in extracting a single speech conversation from a mixture of background chattering; whereas in a musical audio processing scenario source separation aims at separating different instruments that are playing simultaneously from mixture observations.

Zibulevsky and Pearlmutter [123] rely on the assumption that source signals are sparse in a given dictionary to propose a maximum a posteriori estimation of the sources given the observed mixtures, and present results on the separation of audio signals. A similar

assumption is present in the work of Georgiev et al. [38] whose goal is to identify a mixing matrix that is then inverted to retrieve the source signals. Kowalski et al. [58] propose an optimization framework for undetermined convolutive source separation based on sparsity of the source signals and using an iterative thresholding algorithm. Sudhakar et al. [107] devise a framework for filter identification from convolutive mixtures that exploits the sparsity of the filters in the time domain and the sparsity of the source signals in a transformed domain. Sudhakar and Gribonval [108] also tackle the problem of permutation indeterminacy suffered by frequency domain methods for convolutive blind source separation observing that the ℓ_1 norm of filter matrix increases with permutations and seeking therefore to optimize the filters with minimal ℓ_1 norm in the time domain. Benichoux et al. [7] propose a sparse approach to the recovery of multiple room impulse responses that is based on a statistical model of the impulse responses sparsity and envelope. Bobin et al. [9] use morphological component analysis for source separation where the source signals are assumed to be sparse in dictionaries that are dissimilar for different sources. Finally, Gribonval and Lesage [45] summarise the research contributions and challenges encountered by sparse approaches for blind sources separation.

Additional applications

Other applications of sparse approaches include speech recognition and classification. For the former, Fazel and Chakrabartty [34] propose the sparse auditory reproducing kernel features as representations that are coded using a dictionary of gamma tone functions [104] and used for speech recognition. In the latter application Huang and Aviyente [49] propose a framework that joins discrimination methods such as linear discriminant analysis to the sparse representation optimization with the objective to promote sparsity of the representation.

2.6 Dictionary learning for sparse approximation

The sparse models (2.8) and (2.9) described in Section 2.3 rely on the assumption that signals can be expressed as a sparse linear combination of atoms contained in a given dictionary. Although there exist dictionaries which have been designed to mathematically model the properties of certain classes of functions and can be used to sparsely approximate or represent these signals, a more adaptive solution consists in *learning* the

dictionary from examples of data of a given class [90].

Given a set of M observed signals $\{\boldsymbol{y}_m\}_{m=1}^{M} \in \mathbb{R}^N$ which can be stacked along the columns of the matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times M}$, the goal of dictionary learning is to optimize a dictionary $\boldsymbol{\Phi} \in \mathcal{D} \subseteq \mathbb{R}^{N \times K}$ belonging to a class of admissible dictionaries, such that:

$$\boldsymbol{Y} \approx \boldsymbol{\Phi}\boldsymbol{X} \tag{2.22}$$

and the matrix $\boldsymbol{X} \in \mathbb{R}^{K \times M}$ which contains the coefficients of the representations along its columns is sparse. This means that each signal $\boldsymbol{y}_m$ is associated with a sparse representation $\boldsymbol{x}_m$ which contains a small number of nonzero coefficients.

The model (2.22) contains an inherent ambiguity in that, given a solution pair $(\boldsymbol{\Phi}, \boldsymbol{X})$ it is possible to define an equivalent solution $(\boldsymbol{\Phi}' = \boldsymbol{\Phi}\boldsymbol{A}, \boldsymbol{X}' = \boldsymbol{B}\boldsymbol{X})$ by multiplying the dictionary and the coefficients by a pair of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ such that $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{I}$ is the identity. As will be discussed in Chapter 5 it is possible to leverage this ambiguity to promote desirable properties of the dictionary. However, a scaling ambiguity corresponding to the case where $\boldsymbol{A}$ is a diagonal matrix and $\boldsymbol{B} = \boldsymbol{A}^{-1}$ will be from now on avoided by defining the set of admissible dictionaries $\mathcal{D} = \{\boldsymbol{\Phi} : ||\boldsymbol{\phi}_k||_2 = 1 \quad \forall k\}$ as the one where atoms are of unit norm. For the reminder of the thesis normalized dictionaries will be considered without an explicit notation except when otherwise specified.

The optimization problems defined to learn dictionaries from a matrix $\boldsymbol{Y}$ of training signals follow the ones introduced to find sparse approximations. A sparsity constrained formulation of dictionary learning can be written by analogy with (2.12) as follows:

$$\begin{aligned}(\boldsymbol{\Phi}^\star, \boldsymbol{X}^\star) = \underset{\boldsymbol{\Phi} \in \mathcal{D}, \boldsymbol{X} \in \mathbb{R}^{K \times M}}{\arg\min} \; & ||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} \\ \text{such that } & ||\boldsymbol{x}_m||_0 \leq S \quad \forall m\end{aligned} \tag{2.23}$$

where the sparsity of the representation coefficients is enforced in the approximation of every signal and the objective function is the Frobenius norm of the residual. Likewise,

an error-constrained optimization can be defined in analogy to (2.11) as:

$$(\boldsymbol{\Phi}^\star, \boldsymbol{X}^\star) = \underset{\boldsymbol{\Phi}\in\mathcal{D}, \boldsymbol{X}\in\mathbb{R}^{K\times M}}{\arg\min} ||\boldsymbol{X}||_{0,0} \quad \text{such that } ||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} \leq \epsilon \tag{2.24}$$

where the mixed norm notation is extended to the ℓ_0 pseudo-norm applied to the matrix $\boldsymbol{X}$ which counts its total number of non-zero elements, and the parameter ϵ determines the allowed error of the sparse approximation of the training data. Finally, an un-constrained optimization can be defined as:

$$(\boldsymbol{\Phi}^\star, \boldsymbol{X}^\star) = \underset{\boldsymbol{\Phi}\in\mathcal{D}, \boldsymbol{X}\in\mathbb{R}^{K\times M}}{\arg\min} ||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} + \lambda\, ||\boldsymbol{X}||_{0,0} \tag{2.25}$$

where the parameter λ rules the tradeoff between sparsity and approximation error.

Given the ambiguity inherent in the dictionary learning model and the NP-hard nature of sparse approximation optimizations, (2.23), (2.24) and (2.25) are not convex. Even substituting the ℓ_0 pseudo-norm with the ℓ_1 norm as in the sparse approximation formulation (2.15) does not resolve this issue as the interplay between sparse approximation coefficients and dictionary makes optimizing both variables at the same time extremely challenging. One common strategy employed by dictionary learning algorithms is to tackle the optimizations in a block-coordinate descent fashion, starting from an initial dictionary $\boldsymbol{\Phi}^{(0)}$ and performing the following two steps at each iteration t:

Sparse coding : given a fixed dictionary $\boldsymbol{\Phi}^{(t)}$ the matrix of spare representation coefficients $\boldsymbol{X}^{(t)}$ can be computed as a standard sparse approximation problem using any solver that is suitable to the particular formulation. For example, if dictionary learning is defined as a sparsity constrained optimization, then any method that seeks a best S-term approximant to the observed signals can be employed, such as OMP or LARS.

Dictionary update : given a fixed matrix of sparse approximation coefficients $\boldsymbol{X}^{(t)}$, the dictionary $\boldsymbol{\Phi}^{(t+1)}$ is updated in order to improve the objective of the dictionary learning optimization, subject to optional constraints.

It is worth noting that the space $\mathcal{D}$ of dictionaries with unit-norm atoms is not a convex

set, as shown in appendix A.1. This implies that the result of the dictionary update step is a dictionary that does not necessarily contain normalized atoms. However, a normalization step can be added such that:

$$\boldsymbol{\Phi}^{(t+1)} \leftarrow \boldsymbol{\Phi}^{(t+1)}\boldsymbol{\Xi}^{-1} \tag{2.26}$$

$$\boldsymbol{X}^{(t)} \leftarrow \boldsymbol{\Xi}\boldsymbol{X}^{(t)} \tag{2.27}$$

where $\boldsymbol{\Xi}$ is a diagonal matrix whose elements $\xi_{k,k} = \left|\left|\boldsymbol{\phi}_k^{(t+1)}\right|\right|_2$ contain the norm of the dictionary. This way, every atom in the updated dictionary is normalized and the coefficients in the matrix $\boldsymbol{X}^{(t)}$ are updated such that the product $\boldsymbol{\Phi}^{(t+1)}\boldsymbol{\Xi}^{-1}\boldsymbol{\Xi}\boldsymbol{X}^{(t)} = \boldsymbol{\Phi}^{(t+1)}\boldsymbol{X}^{(t)}$ remains unchanged.

Several algorithms have been proposed to solve the dictionary update step and pursue a local minima of the relevant optimization problem. Some of them are described in the next section and the interested reader can find more information in the review paper by Rubinstein et al. [90].

2.7 Algorithms for dictionary learning

2.7.1 SparseNet

Olshausen and Fields in their seminal paper [78] propose a dictionary learning algorithm aimed at representing vectors obtained from patches of natural images. In their formulation, the authors define a penalised optimization problem in the form:

$$(\boldsymbol{\Phi}^\star, \boldsymbol{X}^\star) = \underset{\boldsymbol{\Phi}\in\mathcal{D},\boldsymbol{X}\in\mathbb{R}^{N\times M}}{\arg\min} \frac{1}{2}\left|\left|\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}\right|\right|_{\mathrm{F}}^2 + \lambda\mathcal{P}(\boldsymbol{X}) \tag{2.28}$$

where the first term is the usual quadratic function of the residual which ensures that the representations is close to the observed data in an ℓ_2 measure, the second term is a penalty that induces sparsity in the representation and the parameter λ controls the relative importance of the two objectives.

The authors experimented with different sparsity inducing penalty functions, including $\mathcal{P}(\boldsymbol{X}) = \sum_{k,m}\log(1+x_{k,m})$ and $\mathcal{P}(\boldsymbol{X}) = ||\boldsymbol{X}||_{1,1} \overset{\text{def}}{=} \sum_{k,m}|x_{k,m}|$, and tackled the optimization of (2.28) with a gradient descent strategy. Algorithm 4 summarises the steps of the SparseNet dictionary learning algorithm. For a given number of iterations, the sparse

Algorithm 4: SparseNet dictionary learning

Input: $\boldsymbol{Y}, \boldsymbol{\Phi}^{(0)}, I, \lambda, \eta$
Output: $\boldsymbol{\Phi}^{\star}, \boldsymbol{X}^{\star}$
// Initialisation
1 $i \leftarrow 1$;
2 **while** $i \leq I$ **do**
  // Sparse coding
3 **for** $m = 1 : M$ **do**
4 $\boldsymbol{x}_m \leftarrow \underset{\boldsymbol{x} \in \mathbb{R}^K}{\arg\min}\ ||\boldsymbol{y}_m - \boldsymbol{\Phi}\boldsymbol{x}_m||_2^2 + \lambda \mathcal{P}(\boldsymbol{x}_m)$;
5 **end**
  // Dictionary update
6 $\bar{\boldsymbol{\Phi}} \leftarrow (\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X})\boldsymbol{X}^T$;
7 $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi} - \eta\bar{\boldsymbol{\Phi}}$;
  // Dictionary normalization
8 $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi}\boldsymbol{\Xi}$;
9 $i \leftarrow i + 1$;
10 **end**

coding is performed on each signal independently by using the current dictionary and solving the optimization in Line 4 using any suitable sparse approximation algorithm. The dictionary update is then performed in a batch fashion by computing the gradient of the cost function w.r.t. the dictionary $\bar{\boldsymbol{\Phi}} \stackrel{\text{def}}{=} \nabla_{\boldsymbol{\Phi}}\left(||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\text{F}}^2\right) = (\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X})\boldsymbol{X}^T$. The dictionary is then updated by standard gradient descent using the step size η and normalized through the diagonal matrix $\boldsymbol{\Xi}$ whose (k,k)-th elements $\xi_{k,k} = 1/\,||\boldsymbol{\phi}_k||_2$ are the inverse of the norm of the corresponding atom.

As the BPD algorithm can be interpreted as a MAP estimation of the approximation coefficients under a Gaussian noise distribution and a Laplacian coefficients distribution, the SparseNet algorithm and the successive dictionary learning technique proposed by Lewicki and Sejnowski [60] can be thought in probabilistic terms as an approximated maximum likelihood (ML) estimation of the dictionary. Let $p(\boldsymbol{Y}|\boldsymbol{\Phi})$ be the likelihood of observing a set of signals $\boldsymbol{Y}$ given a dictionary $\boldsymbol{\Phi}$. This cannot be directly maximised but it can be expressed in terms of the likelihood $p(\boldsymbol{Y}|\boldsymbol{\Phi}, \boldsymbol{X})$ (that is the probability of observing a set of signals $\boldsymbol{Y}$ given a dictionary and a matrix of approximation coefficients) and the prior probability of the coefficients $p(\boldsymbol{X})$:

$$p(\boldsymbol{Y}|\boldsymbol{\Phi}) = \int_{\boldsymbol{X}} p(\boldsymbol{Y}, |\boldsymbol{\Phi}, \boldsymbol{X}) p(\boldsymbol{X}) d\boldsymbol{X} \tag{2.29}$$

where the integral is a marginalisation over the *latent* variable $\boldsymbol{X}$. Unfortunately the computation of this integral is not practical and therefore an *approximated* ML strategy consists in considering the *mode* of the distribution $p(\boldsymbol{Y}, |\boldsymbol{\Phi})$ instead, which in turns leads to the iterative method consisting of sparse coding and dictionary update stages described in Section 2.6 (see [4, 60] for more details on the probabilistic interpretation of dictionary learning).

Interestingly, the set of dictionary atoms $\boldsymbol{\phi}_k$ learned from natural images in the paper by Olshausen and Fields [78] resulted to be spatially localised, oriented and bandpass functions. These properties are not present in the atoms learned using non-sparse techniques such as principal component analysis and are believed to describe the behaviour of the receptive fields of the cells in the primary visual cortex, a conjecture that arose the interest of the neurobiology research community on sparse representations.

A similar first order approach has been adopted by Smith and Lewicki [104] who learned atoms from speech and natural sounds. In this case, the resulting functions resemble asymmetric sinusoids with sharp attacks and gradual decays of different length (so-called gammatone functions), a property that is thought to be common to the impulse response of the cochlear filters that process sounds in our inner ear.

2.7.2 Method of optimal directions and K-SVD

Engan et al. [32] proposed the method of optimal directions (MOD) where the optimization explicitly constraints a sparse solution as in (2.23). To tackle this optimization they use a block coordinate descent method where the sparse representation step can be employed with any algorithm which attempts to find an optimal k-term approximation, such as OMP or LARS, and the subsequent dictionary update step is performed computing the pseudo-inverse of the current sparse representation. Algorithm 5 summarises the optimization followed by MOD.

The sparse coding is performed on each signal by fixing a maximum number of active atoms S, and the dictionary update is carried out by computing the pseudo-inverse of the current sparse approximation coefficients as in line 6. This provides the locally optimal solution to the minimisation $\boldsymbol{\Phi}^\star = \underset{\boldsymbol{\Phi} \in \mathbb{R}^{N \times K}}{\arg\min} ||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}}$.

The K-SVD algorithm introduced by Aaron et al. [4] aims at minimising the same optimization problem, but differs from MOD in the dictionary update step. Given the

Algorithm 5: Method of optimal directions (MOD)

Input: $\boldsymbol{Y}, \boldsymbol{\Phi}^{(0)}, I, S$
Output: $\boldsymbol{\Phi}^{\star}, \boldsymbol{X}^{\star}$
// Initialisation
1 $i \leftarrow 1$;
2 **while** $i \leq I$ **do**
  // Sparse coding
3 **for** $m = 1 : M$ **do**
4 $\boldsymbol{x}_m \leftarrow \underset{\boldsymbol{x} \in \mathbb{R}^K}{\arg\min} \; ||\boldsymbol{y}_m - \boldsymbol{\Phi}\boldsymbol{x}_m||_2^2 \quad \text{s.t. } ||\boldsymbol{x}_m||_0 \leq S;$
5 **end**
  // Dictionary update
6 $\boldsymbol{\Phi} \leftarrow \boldsymbol{Y}\boldsymbol{X}^{\dagger}$;
  // Dictionary normalization
7 $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi}\boldsymbol{\Xi}$;
8 $t \leftarrow t + 1$;
9 **end**

objective function of (2.23) $\mathcal{C}(\boldsymbol{\Phi}, \boldsymbol{X}) = ||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}}$, the approximant term can be written as a sum of rank-1 matrices:

$$\begin{aligned} \mathcal{C}(\boldsymbol{\Phi}, \boldsymbol{X}) &= \left|\left| \boldsymbol{Y} - \sum_{k=1}^{K} \boldsymbol{\phi}_k \boldsymbol{x}^k \right|\right|_{\mathrm{F}} \\ &= \left|\left| \left(\boldsymbol{Y} - \sum_{k' \neq k} \boldsymbol{\phi}_{k'} \boldsymbol{x}^{k'} \right) - \boldsymbol{\phi}_k \boldsymbol{x}^k \right|\right|_{\mathrm{F}}. \end{aligned}$$

Let the partial residual matrix be $\boldsymbol{E}_k \overset{\text{def}}{=} \boldsymbol{Y} - \sum_{k' \neq k} \boldsymbol{\phi}_{k'} \boldsymbol{x}^{k'}$, then the atom $\boldsymbol{\phi}_k$ and the corresponding row of sparse approximation coefficients $\boldsymbol{x}^k$ can be jointly optimized to locally minimise the cost function $\mathcal{C}$ by calculating the best rank-1 approximation of $\boldsymbol{E}_k$. Moreover, since the support of the sparse approximation coefficients should not be modified during the dictionary update step, $\boldsymbol{E}_k$ and its rank-1 approximation are restricted to the columns corresponding to the signals that use the k-th atom in their sparse approximation, that is, the indexes corresponding to non-zero elements of the vector $\boldsymbol{x}_k$.

Algorithm 6 summarises the K-SVD algorithm. While the sparse coding step included in lines 3 to 5 does not differ from the one performed by MOD, the dictionary update is carried out on each atom $\boldsymbol{\phi}_k$ independently using the following strategy:

I - For each dictionary atom $\boldsymbol{\phi}_k$, the set Λ_k of nonzero elements of the k-th row of $\boldsymbol{X}$

Algorithm 6: K-SVD dictionary learning

Input: $\boldsymbol{Y}, \boldsymbol{\Phi}^{(0)}, I, S$
Output: $\boldsymbol{\Phi}^{\star}, \boldsymbol{X}^{\star}$
// Initialisation
1 $i \leftarrow 1$;
2 **while** $i \leq I$ **do**
// Sparse coding
3 **for** $m = 1 : M$ **do**
4 $\boldsymbol{x}_m \leftarrow \underset{\boldsymbol{x} \in \mathbb{R}^K}{\arg\min} \; ||\boldsymbol{y}_m - \boldsymbol{\Phi}\boldsymbol{x}_m||_2^2 \quad \text{s.t. } ||\boldsymbol{x}_m||_0 \leq S$;
5 **end**
// Dictionary update
6 **for** $k = 1 : K$ **do**
7 $\Lambda_k \leftarrow i \subseteq \{1, \ldots, M\}$ s.t. $x_{k,i} \neq 0$;
8 $\boldsymbol{E}_k \leftarrow \left[\boldsymbol{Y} - \sum_{j \neq k} \boldsymbol{\phi}_j \boldsymbol{x}^j\right]_{\Lambda_k}$;
9 $(\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}) \leftarrow \text{SVD}\,(\boldsymbol{E}_k)$;
10 $\boldsymbol{\phi}_k \leftarrow \boldsymbol{u}_1$;
11 $\boldsymbol{x}_{\Lambda_k} \leftarrow \sigma_{1,1} \boldsymbol{v}_1{}^T$;
12 **end**
// Dictionary normalization
13 $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi}\boldsymbol{\Xi}$;
14 $i \leftarrow i + 1$;
15 **end**

(that is, the set of training data which use the k-th atom in their approximation) is identified in line 7.

II - A partial residual matrix is calculated and its columns are restricted to the active set of signals that use the k-th atom for their sparse approximation in line 8.

III - The atom $\boldsymbol{\phi}_k$ and the coefficients $[\boldsymbol{x}^k]_{\Lambda_k}$ are updated using the solution of the best rank-1 approximation of the matrix $\boldsymbol{E}_k$, which can be calculated using its SVD in lines 9 to 11.

Dai et al. [20] extended the simultaneous update of dictionary and sparse approximation coefficients to arbitrary subsets of atoms and relative coefficients in the active set Λ using a gradient descent and line search strategy. Their so-called simultaneous codeword optimization (SIMCO) strategy is also regularised by a penalty function that promotes well conditioned sub-dictionaries, a concept closely related to the mutual coherence measure for incoherent dictionary learning that will be developed in Chapter 5.

Dictionary learning has been interpreted by several authors as a generalisation of

vector quantisation [4, 112, 103]. In vector quantisation, a set of representative vectors or codebook is learned from the training set, and each point is represented by one of these vectors. The k-means algorithm is perhaps the simplest and most widely used vector quantisation strategy; it starts by choosing k vectors in the training set at random as the initial codebook and it iterates the following steps: i) assign each point in the training set to the closest of the k elements of the codebook ii) update each element of the codebook with the mean of the vectors associated with it at the previous step. Once the vector quantisation algorithm has run for a certain number of iterations and a codebook has been defined, each point in the training set is approximated by one of the elements of the codebook, that is a sparse representation with the number of active atoms $S = 1$. A dictionary learning algorithm, on the other hand, allows each point of the training set to be approximated by a sparse linear combination of the elements of the codebook, or atoms of the dictionary. The name K-SVD echoes the k-means algorithm, but also indicates that the dictionary is optimized by performing K singular value decompositions.

In a recent contribution, Mailhé and Plumbley analysed the local optimality of the SPARSENET, MOD and K-SVD dictionry updates for the objective (2.23), showing that K-SVD can perform better than the other ones, especially if initialised with the solution returned by the SPARSENET algorithm [64].

2.8 Applications of dictionary learning

Classification

Traditionally, the goal of dictionary learning is to optimize a set of atoms that provide a sparse representation of observed data. Since the seminal papers by Chen et al. [17], sparse representation have been used for de-noising purposes and this remains one of the main applications where dictionary learning algorithms have been employed with great success. From a machine learning prospective, this is an unsupervised learning problem where a low dimensional model is learned from a set of training data. However, there are supervised tasks like classification that can benefit from learning adaptive dictionaries for sparse representations [69, 68, 65].

Rodriguez and Sapiro [89] introduced a dictionary learning algorithm for representation and discrimination whose goal is twofold: on one hand, the set of learned atoms is

optimized in order to provide a sparse representation of training data, as in traditional dictionary learning. On the other hand, in the representation coefficients domain training data belonging to the same class should be close to each other (in an ℓ_2 measure), and far apart from data belonging to different classes. Given a set of training data, each vector $\boldsymbol{y}_m$ is associated with a class $c_m \in \mathcal{C} = \{1, \ldots, Q\}$, where Q is the total number of classes, so to define a supervised classification problem. The mixed objective is accomplished by specifying a dictionary learning problem as follows:

$$(\boldsymbol{\Phi}^\star, \boldsymbol{X}^\star) = \underset{\boldsymbol{\Phi}\in\mathcal{D}, \boldsymbol{X}\in\mathbb{R}^{K\times M}}{\arg\min} ||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}}^2 + \theta\mathcal{C}(\boldsymbol{X}) \quad \text{such that } ||\boldsymbol{x}_m||_0 \leq S \quad \forall m \tag{2.30}$$

where the first term is the usual quadratic function of the residual, the penalty $\mathcal{C}(\boldsymbol{X})$ is a linear discriminant function of the coefficients and the parameter θ controls the relative importance of the two objectives. Let Λ_q be the subset of indexes corresponding to signals belonging to the q-th class, and let $\bar{\boldsymbol{x}}_{\Lambda_q} = \frac{1}{M_q}\sum_{m\in\Lambda_q}\boldsymbol{x}_m$ be the q-th class centroid (that is, the mean of the sparse approximation coefficients of signals belonging to class q). $\mathcal{C}(\boldsymbol{X})$ is minimised when the intra-class variance of the vectors in $\boldsymbol{X}_{\Lambda_q}$ is small and the inter-class variance of the class centroids $\bar{\boldsymbol{x}}_{\Lambda_q}$ is large. The optimization of function (2.30) is tackled by a *supervised* K-SVD algorithm where the dictionary update is performed in the same way of the original method [4], while the sparse representation step is accomplished using the class supervised simultaneous orthogonal matching pursuit (SSOMP), a modified OMP which takes into account the linear discriminant penalty during the atom selection stage.

Schnass and Vandergheynst proposed a different approach to the classification problem [96, 95] which accomplish the same goal of simultaneous representation and discrimination in the coefficients domain where the dictionary $\boldsymbol{\Phi}$ is assumed to be a concatenation of class specific orthonormal bases $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_{(1)} \cdots \boldsymbol{\Phi}_{(Q)}]$ each of which satisfies:

$$\frac{\left|\left|{\boldsymbol{\Phi}_{(j)}}^T\boldsymbol{y}_i\right|\right|_2}{\left|\left|{\boldsymbol{\Phi}_{(i)}}^T\boldsymbol{y}_i\right|\right|_2} < 1 \quad \forall j \neq i \in \Lambda_i. \tag{2.31}$$

This means that the norm of the representation coefficients of points belonging to class

i must be greater when the data are analysed with the respective orthonormal base $\mathbf{\Phi}_{(i)}$ rather than with any other $\mathbf{\Phi}_{(j)}, j \neq i$. The problem here becomes optimizing a set of Q orthonormal bases which meet the above constraint. This is accomplished by a projection onto convex sets algorithm, which is an iterative method that alternatively projects an initial set $\{\mathbf{\Phi}_q\}_{q=1}^{Q}$ onto the sets of orthonormal basis and onto the set of basis that satisfy the above condition respectively.

Interestingly, although no sparsity is explicitly involved in this algorithm, the incoherence objective between set of orthonormal bases and data belonging to a given class means that a large number of inner products $\left\langle \boldsymbol{y}_{(i)}, \boldsymbol{\phi}_{(j)k} \right\rangle \approx 0$ is close to zero and can be linked with the concept of co-sparsity mentioned in section 2.9.1.

Music transcription and source separation

Abdallah and Plumbley [2] propose a dictionary learning algorithm that is formulated as a probabilistic model and is inspired by the method published in [60] to learn atoms that efficiently represent the magnitude spectrum of polyphonic music. When applied to synthetic harpsichord musical excerpts, the learned atoms display a harmonic structure that resembles the spectrum of single notes, while the matrix of approximation coefficients can be interpreted as an activation matrix that indicates which notes are active at any specific time. Polyphonic music transcription can be thus tackled with this technique in a unsupervised fashion that is similar to the approach followed by non-negative matrix factorization algorithms.

Scholler and Purwins [97] employ a first-order dictionary learning such as the one detailed in Section 2.7.1 to learn atoms from mixtures of percussion sounds that contain several classes of percussion instruments. They use matching pursuit to code audio signals and, once the sparse approximation coefficients are obtained, they train a classifier using a support vector machine in order to discriminate between bass drums, snare drum and hi-hat, a task that is essential in drums transcription. Their results show that features obtained from sparse approximation coefficients are more robust to noise than traditional timbre descriptor features.

Bobin et al. [10] proposed the morphological component analysis (MCA) as a novel sparse model where observed data are approximated using a sum of sparse linear combinations of atoms belonging to different dictionaries with dissimilar structures (e.g., edges

and textures present in images that can be efficiently represented using curvelets and local cosine functions respectively). In a successive work [9] MCA is extended to a multi-channel case and employed to tackle source separation problems using an algorithm that resembles dictionary learning by gradient descent.

In the multi-channel MCA a matrix of observed signals is modelled as a mixture of morphological components that are in turn approximated using several different dictionaries. The columns of the unknown mixing matrix and the sparse approximation coefficients of the corresponding morphological component are optimized following a two-steps strategy that resembles the one introduced in Section 2.6 to learn dictionaries for sparse approximation. In particular, the coefficients are updated using a soft-thresholding method and the mixing matrix weights using a gradient descent update.

2.9 Additional background

This section presents additional topics on sparse approximation and dictionary learning that are not essential for understanding the reminder of the thesis and its main contributions, but complement what discussed so far to offer a more comprehensive overview of the field and its related themes.

2.9.1 Additional models and algorithms for sparse approximation

Non-convex optimization for sparse approximation

The family of ℓ_1 optimization algorithms has been used as a convex relaxation strategy to solve sparse approximation problems that are expressed in terms of an ℓ_0 pseudo-norm, as in the optimizations (2.10), (2.11), (2.12) and (2.13). Alternatively, ℓ_p norms with $0 < p < 1$ can be considered and generally lead to the formulation of non-convex optimization problems.

The sparse reconstruction by separable approximation (SpaRSA) algorithm proposed by Wright et al. [119] is a general framework for the solution of minimisation problems of the form:

$$\boldsymbol{x}^\star = \underset{\boldsymbol{x}\in\mathbb{R}^K}{\arg\min} \; ||\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}||_2^2 + \lambda\mathcal{P}(\boldsymbol{x}) \tag{2.32}$$

where the first term of the minimisation is a quadratic error term and the function $\mathcal{P}(\boldsymbol{x}) = \sum_k \mathcal{P}_k(x_k)$ is a separable sparsity-inducing penalty function (that is, a function that

can be expressed as a sum of functions of the individual coefficients). This has the main advantage of being readily applicable to cases where the penalisation term is not necessarily the ℓ_1 norm, but can be any non-convex ℓ_p norm with $0 < p < 1$.

Wipf and Nagarajan [118] review so-called iterative re-weighted schemes to solve the penalised least-square problem (2.32). The general idea common to the various techniques is to start from an estimate of the solution $\boldsymbol{x}^{(0)}$ and a set of initial weights $\boldsymbol{w}^{(0)}$. The representation coefficients are iteratively updated by solving at each iteration t the problem

$$\boldsymbol{x}^{(t+1)} = \underset{\boldsymbol{x}\in\mathbb{R}^K}{\arg\min} \; ||\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}||_2^2 + \lambda \sum_k w_k^{(t)} \mathcal{P}_k(x_k) \tag{2.33}$$

and the weights are also updated usually employing a function $\mathcal{Q}$ proportional to the inverse of each component $w_k^{(t+1)} = \mathcal{Q}\left(1/x_k^{(t+1)}\right)$.

The authors in [118] also propose a novel re-weighted optimization formulation that can be extended to non-separable penalisation functions and compare their approach to convex relaxation methods reporting superior performance for sparse recovery and approximation.

Compressive sampling

Compressed sensing or compressive sampling is a novel model for the acquisition of signals that relies on sparse representations [16]. One of the classical tenets in signal processing is that a function must be sampled at a *Nyquist* frequency that is twice its maximum frequency in order to be able to accurately reconstruct it. By assuming that a signal $\boldsymbol{y} \in \mathbb{R}^N$ is sparse in a given dictionary $\boldsymbol{\Phi} \in \mathbb{R}^{N\times K}$, compressive sampling allows to reconstruct it from $P < N$ measurements realized through the measurement matrix $\boldsymbol{M} \in \mathbb{R}^{P\times N}$. The compressive sampling model can be expressed as:

$$\boldsymbol{z} = \boldsymbol{M}\boldsymbol{y} = \boldsymbol{M}\boldsymbol{\Phi}\boldsymbol{x} \tag{2.34}$$

where $\boldsymbol{z} \in \mathbb{R}^P$ is the observable measurement and $\boldsymbol{y}$ is the unknown signal to be recovered. Knowing $\boldsymbol{M}$ and $\boldsymbol{\Phi}$, the coefficients $\boldsymbol{x}$ can be retrieved from $\boldsymbol{z}$ by solving the sparse

representation problem:

$$\boldsymbol{x}^{\star} = \underset{\boldsymbol{x}\in\mathbb{R}^K}{\arg\min} \; ||\boldsymbol{x}||_0 \quad \text{such that } \boldsymbol{z} = \boldsymbol{M}\boldsymbol{\Phi}\boldsymbol{x} \tag{2.35}$$

and the signal can be reconstructed by $\boldsymbol{y}^{\star} = \boldsymbol{\Phi}\boldsymbol{x}^{\star}$.

In the context of compressive sampling, it is crucial that algorithms for sparse representation succeed in recovering the sparse representation coefficients that generate the signal to allow its reconstruction (or, less strictly, it is necessary that the *support* Λ of the sparse representation is correctly recovered as the magnitude of the coefficients can be easily retrieved by calculating the pseudo-inverse of the sub-matrix restricted to the support $\boldsymbol{x}^{\star}_{\Lambda} = (\boldsymbol{M}\boldsymbol{\Phi})^{\dagger}_{\Lambda}\boldsymbol{y}$). The restricted isometry property (RIP) is a condition that links the properties of the measurement matrix, the dictionary and the sparse vectors of coefficients to the success of sparse recovery, which in turn ensures the reliability of compressive sampling [14].

Let $\boldsymbol{A} \stackrel{\text{def}}{=} \boldsymbol{M}\boldsymbol{\Phi}$, for each sparsity level S and for any S-sparse vector of coefficients $\boldsymbol{x}$ the restricted isometry constant δ_S is the smaller positive number for which the following relation holds:

$$(1-\delta_S)\,||\boldsymbol{x}||_2^2 \leq ||\boldsymbol{A}\boldsymbol{x}||_2^2 \leq (1+\delta_S)\,||\boldsymbol{x}||_2^2\,. \tag{2.36}$$

This condition loosely means that the Euclidean length of an S-sparse signal is almost preserved by the application of the linear operator $\boldsymbol{A}$ as long as the RIP constant is not close to 1, which in turn implies that S-sparse signals do not belong to the null-space of the matrix $\boldsymbol{A}$. Candès et al. proved that ℓ_1 minimisation algorithms succeed in recovering the sparse representation coefficients as long as $\delta_{2S} < \sqrt{2} - 1$ [14]. The RIP is in turn dependent on the cross-coherence between the atoms of the dictionary and the rows of the measurement matrix, which is defined as the maximum absolute correlation between the two set of vectors:

$$\mu(\boldsymbol{M}, \boldsymbol{\Phi}) \stackrel{\text{def}}{=} \underset{p,k}{\arg\max} \; \langle \boldsymbol{m}^p, \boldsymbol{\phi}_k \rangle. \tag{2.37}$$

A low cross-coherence means that the RIP condition is satisfied for a large set of sparse signals, and by choosing $\boldsymbol{M}$ to be a random matrix the probability of achieving a low cross-

coherence is nearly optimal, meaning that random sampling matrices offer an universal encoding strategy, as suggested by Candès and Tao [15].

Analysis sparsity

The sparse models introduced in Section 2.3 assume that a signal $\boldsymbol{y}$ is *synthesised* from a small number of atoms, that is, a *synthesis* sparse model. An alternative view that has emerged in recent years is the so-called *analysis* sparse model [75, 44, 31]. In this case, a signal $\boldsymbol{y} \in \mathbb{R}^N$ is multiplied or *analysed* by an analysis matrix $\boldsymbol{M} \in \mathbb{R}^{K \times N}$ with $K \geq N$ resulting in a vector of coefficients $\boldsymbol{x} \in \mathbb{R}^K$.

$$\boldsymbol{x} = \boldsymbol{M}\boldsymbol{y}. \tag{2.38}$$

The coefficients are said to be *co-sparse* if the number of zero components is large. Geometrically, this means that the signal lives in a space that is orthogonal to many rows of the analysis matrix $\boldsymbol{M}$. An analysis sparse approximation problem can be defined in parallel to the one proposed for the sparse synthesis model.

$$\tilde{\boldsymbol{y}}^\star = \underset{\tilde{\boldsymbol{y}} \in \mathbb{R}^N}{\arg\min} \; ||\boldsymbol{y} - \tilde{\boldsymbol{y}}||_2^2 + \lambda \, ||\boldsymbol{M}\tilde{\boldsymbol{y}}||_0 \tag{2.39}$$

where $\tilde{\boldsymbol{y}}$ is a signal approximant that is optimized following a tradeoff between the quadratic representation error and the level of co-sparsity.

As pointed out by Elad at al. [31] the synthesis and analysis models are equivalent for (under)-complete representations but dramatically differ in an over-complete setting. Understanding the relations between synthesis and analysis models and the potentialities of the latter in the range of applications that have been successfully tackled with the former is an open and thriving research field. Furthermore, a parallel between algorithms for sparse synthesis and analysis approximation is emerging, and is an object of current research.

2.9.2 Additional models and algorithms for dictionary learning

Dictionary learning for ℓ_1 exact sparse coding

Dictionary learning for sparse approximation allows for a non-zero residual that accounts for any modelling error between a set of observed data and their sparse approximation.

We can alternatively define an exact sparse model as:

$$(\boldsymbol{\Phi}^{\star}, \boldsymbol{X}^{\star}) = \underset{\boldsymbol{\Phi}\in\mathcal{D}}{\arg\min} \; ||\boldsymbol{X}||_{1,1} \quad \text{such that } \boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{X} \tag{2.40}$$

where the signals are exactly represented by linear combinations of the atoms in $\boldsymbol{\Phi}$ and the sparse objective function is relaxed to the $\ell_{1,1}$ mixed norm of the representation coefficients. Plumbley [84] proposed a gradient descent algorithm for solving this problem that makes use of the geometry of a dual problem that is defined analogously to the one proposed to find the solution of the basis pursuit (2.15). This method starts from an initial dictionary that satisfies the exact sparse representation constraint, and calculates updates that minimise the objective function while remaining in the constraint set.

Majorization algorithm

Yaghoobi et al. proposed an algorithm for dictionary learning using a majorization method [120] that offers an alternative optimization framework to the strategies detailed so far. In this technique, an optimization problem is formulated to solve a penalised objective such as the one defined in (2.28) using a convex penalisation term, and the set of admissible dictionaries $\mathcal{D}$ (that is usually constrained to contain atoms with unit norm) is relaxed to be either the set of dictionaries with bounded Frobenious norm or the set of dictionaries with atoms of bounded ℓ_2 norm. These two constraint sets can be shown to be convex and, therefore, both the sparse coding and the dictionary update step can be performed using convex optimization tools without the need of normalization.

A majorisation-minimisation algorithm is employed to solve the sub-problems of dictionary learning. This is a general strategy for convex optimization that, given a convex function $\mathcal{F}(\omega)$ starts from an initial point $\omega^{(0)}$ and performs at each iteration t the following operations:

I - Define a surrogate function $\mathcal{G}\left(\omega^{(t)}, \xi\right) \geq \mathcal{F}(\omega)$ that majorises the original function using, for example, a second-order Taylor expansion of the function $\mathcal{F}(\omega)$ around the point $\omega^{(t)}$.

II - Find the value $\xi^{\star} = \underset{\xi}{\arg\min}\, \mathcal{G}\left(\omega^{(t)}, \xi\right)$ that minimises the surrogate function.

III - Update $\omega^{(t+1)} = \xi^\star$, calculate $\mathcal{F}\left(\omega^{(t+1)}\right)$ and iterate from step I.

Tackling the dictionary learning problem with a majorisation-minimisation strategy allows one to benefit from the wide body of research undertaken in the field of convex optimization and to specify additional constraints on the dictionary.

Online algorithms

The methods described so far perform a batch learning of the dictionary $\boldsymbol{\Phi}$ from a set of M training samples. Alternatively, it is possible to optimize the dictionary in an online fashion, continuously updating the atoms as new training samples become available.

Mairal et al. [66, 67] proposed a method to solve the problem:

$$(\boldsymbol{\Phi}^\star, \boldsymbol{X}^\star)^{(T_0)} = \underset{\boldsymbol{\Phi}\in\mathcal{D}, \boldsymbol{X}\in\mathbb{R}^{K\times M}}{\arg\min} \left|\left|\boldsymbol{Y}^{(T_0)} - \boldsymbol{\Phi}\boldsymbol{X}^{(T_0)}\right|\right|_{\mathrm{F}} + \lambda \left|\left|\boldsymbol{X}^{(T_0)}\right|\right|_1 \tag{2.41}$$

where the super-script T_0 indicates that the matrices of signals and coefficients contain online data acquired at discrete times $t = 1, \ldots, T_0$. This is an online version of the unconstrained minimisation (2.25) where the sparsity penalty has been relaxed to the ℓ_1 norm of the approximation coefficients. The sparse coding is performed using LARS and the dictionary update is performed one atom at a time using a block coordinate descent strategy. For this algorithm, the use of mini-batches (that is, a small number of training samples considered in each optimization step), has been shown to improve the representation performance in image processing applications.

Skretting and Engan [103] developed an online version of the MOD algorithm where all the dictionary atoms are updated every time a new training vector becomes available using a fast matrix inversion to compute the pseudo-inverse of the current approximation coefficients. In addition, the authors introduced a forgetting factor which allows a *search then converge* strategy: the dictionary atoms are allowed to change abruptly during the first iterations of the algorithm when only a few training samples are considered. Then, as the training set becomes bigger, the forgetting factor has the effect of stabilising the algorithm promoting its convergence to a fixed point.

Sparse dictionaries

Jafari and Plumbley [52] propose a greedy dictionary learning algorithm for sparsely approximate speech signals. Atoms are selected from the training data by iteratively

choosing the sparsest speech frames, promoting sparsity in the dictionary itself as well as in the approximation coefficients. The resulting greedy adaptive dictionary learning algorithm (GAD) has been used for speech approximation and de-nonising with performance comparable to the principal component analysis method. The GAD has also been extended by the same authors to the analysis of other audio signals than speech [51].

Rubinstein et Al. [91] offer a different view on sparse atoms by learning dictionaries of the form $\boldsymbol{\Phi} = \boldsymbol{\Psi}\boldsymbol{A}$ where $\boldsymbol{\Psi}$ is a fixed dictionary and $\boldsymbol{A}$ is a matrix of sparse representation coefficients. By optimizing $\boldsymbol{A}$, each atom $\boldsymbol{\phi}_k = \boldsymbol{\Psi}\boldsymbol{a}_k$ is a sparse linear combination of the vectors contained in the fixed dictionary $\boldsymbol{\Psi}$. The advantage of this approach is that the fixed dictionary can bear a fast implementation of the matrix-vector multiplication speeding up the learning process and constituting a trade-off between the flexibility of adaptive dictionary learning algorithms and the fast implementation of most traditional transforms.

Shift-invariant dictionary learning

Shift-invariant dictionary learning consists in optimizing a set of atoms that can be arbitrarily shifted in time or space to approximate a single observed signal belonging to a very high dimensional space. This is particularly suited, for example, for the modelling of audio signals because atoms can be learned from a single audio stream rather than from a set of lower dimensional training samples that are obtained by a windowing operation.

Jost et al. proposed the matching of time-invariant filters (MoTIF) algorithm to learn shift-invariant atoms [54]. In this method, a shifting operator is used to place a given atom at a particular time shift and the corresponding delay is optimized in order to maximise the correlation between the atom and the training signal, so that a matching pursuit strategy can be employed to find both the optimal shift and the optimal combination of atoms. An adjoint shifting operator is then introduced to extract portions of the training signal that are approximated using a given atom and that should be used in the dictionary update step. In the MoTIF algorithm this is realized by solving a generalised eigenvalue problem, while in the shift-invariant K-SVD proposed by Mailhé et al. [63] the atoms update is obtained jointly with the optimization of the corresponding set of sparse approximation coefficients, as for the original K-SVD algorithm.

2.9.3 Other matrix factorisation models and algorithms

Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a popular technique which has been proven to be successful for audio signal processing applications, such as source separation and automatic music transcription [36, 81, 35, 53]. Although it is not traditionally included among dictionary learning methods, the goal of NMF is exactly the one described in equation (2.22), that is, approximating a matrix of observed data with a product between a dictionary that contains elementary atoms and a matrix of coefficients which describe which atoms are contributing to the observed variables. The NMF optimization problem can be expressed as follows:

$$(\boldsymbol{\Phi}^\star, \boldsymbol{X}^\star) = \underset{\boldsymbol{\Phi}\in\mathcal{D}, \boldsymbol{X}\in\mathbb{R}^{K\times M}}{\arg\min} \mathcal{L}(\boldsymbol{Y}, \boldsymbol{\Phi}\boldsymbol{X}) \quad \text{such that } \boldsymbol{\Phi}, \boldsymbol{X} \succeq \mathbf{0} \tag{2.42}$$

where the objective is a function of the approximation error and the symbol $\succeq$ indicates element-wise inequality, implying that all the variables considered are non-negative element-wise. In many applications of NMF to audio signals, the power spectrum or magnitude spectrum resulting from a short time Fourier transform (STFT) are modelled as the product between the atoms in the columns of the matrix $\boldsymbol{\Phi}$ and the correspondent coefficients stored in the matrix $\boldsymbol{X}$. A notable difference between dictionary learning and NMF is that often the number of atoms $K < N$ is smaller than the dimension of the spectrograms, resulting in an over-determined system of equations. In choosing the loss function $\mathcal{L}(\boldsymbol{Y}, \boldsymbol{\Phi}\boldsymbol{X})$, the Itakura-Saito divergence [35] is often preferred to the usual euclidian distance due to its scaling invariance and expectation-maximisation (EM) strategies are employed in the optimization. Alternatively, it is possible to perform underdetermined NMF whenever the number of atoms $K > N$. In this case sparsity can be used to constraint the solution, which makes NMF a non-negative version of dictionary learning.

Latent variable decompositions

Shashanka et al. [99] introduced a latent variable model which resembles NMF, but whose probabilistic formulation allows for richer, more sophisticated models to be defined starting from its general framework. In this work, each vector of the magnitude spectrum

deriving from a (STFT) of audio signals is interpreted as a scaled histogram of a random process containing two latent variables whose generative model can be explained as follows:

- Latent variable s determines which instrument/speech contributes to the magnitude spectrum according to its probability distribution $P_t(s)$, which is time-dependent.
- Latent variable z determines which multinomial component (specific to the instrument picked at the previous stage) contributes to the magnitude spectrum according to its probability distribution $P_t(z|s)$, also time-dependent.
- A frequency bin f is selected according to the multinomial distribution of the component picked at previous stages $P_s(f|z)$. This is fixed in time and represents one of the atoms of the representation.

The process is repeated and generates the magnitude spectrum in each STFT window, such that its probability distribution can be expressed by marginalization over the latent variables:

$$P_t(f) = \sum_s P_t(s) \sum_{z \in \{\boldsymbol{z}_s\}} P_t(z|s) P_s(f|s)$$

where the set $\{\boldsymbol{z}_s\}$ contains all the components associated with the s-th instrument. It is possible to express this probabilistic model as a matrix factorization analogous to NMF and learn the atoms and time-dependent activations with an EM strategy. This has been proved to be effective for signal processing tasks such as blind source separation [86] and dereverberation [102].

2.10 Summary

This chapter contains an overview of the main concepts of the most relevant literature in the fields of sparse approximation and dictionary learning. Starting from the definition of dictionaries and their role in signal processing, orthonormal bases have been defined as a traditional way of analysing and processing signals. The lapped orthogonal transform has been presented as a framework for realizing transforms that are applied to signals in a high dimensional space and that are both globally and locally orthonormal. They will be the starting point for the realization of a pitch-synchronous transform that is described in Chapter 3, and is at the base of popular transforms used for audio processing.

Over-complete dictionaries have been introduced as a more general and flexible class of transforms than orthonormal dictionaries and lead to the concept of sparse approximations, that is, expressing signals as a linear combination of a small number of atoms in the dictionary. Sparse approximations models and algorithms have been presented, including greedy methods, convex relaxation algorithms and non-convex optimization strategies. Selected applications of sparse approximations and the related models of compressive sampling and analysis sparsity have been detailed.

Sparse approximation relies on dictionaries that are suitable for expressing signals using a small number of active atoms, and dictionary learning answers the problem of learning such dictionaries from a set of training data of a given class, an unsupervised task that has been interpreted as a generalisation of vector quantisation algorithms. A few selected algorithms for dictionary learning have been detailed and others have been more briefly mentioned, along with related methods for matrix factorization. Finally, besides the applications of sparse approximations that are allowed by learning dictionaries adapted to a given class of signal, a number of applications that are specifically associated with dictionary learning have been presented, including supervised problems such as classification and unsupervised ones such as music transcription and source separation.

Chapter 3

Studying sparsity and disjointness of audio transforms

The work presented in this chapter resulted from a collaboration with Dimitrios Giannoulis, a PhD student at the Centre for Digital Music at Queen Mary University of London. The study of disjointness of audio time-frequency transforms appeared in a joint publication at the Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) [41].

Although there has been constant communication and substantial overlap between the work undertaken by myself and by Dimitrios Giannoulis, my main contribution consisted in the design and implementation of the pitch-synchronous lapped orthogonal transform (LOT) presented in Section 3.1 and of the LOTBOX detailed in Appendix B.1, whereas Dimitrios focused on the experimental evaluation described in Section 3.3.

3.1 Pitch-synchronous transforms using LOTs

Lapped orthogonal transforms (LOTs) have been introduced in Section 2.1.2 as a way to perform a window-based analysis of one dimensional signals using bases that are both locally and globally orthonormal [72]. Within the framework of LOTs different types, lengths and overlaps of the local orthonormal bases can be specified, making LOTs a class of parametric transforms. Some notable examples include the non-overlapping STFT obtained using Fourier local bases, constant window length and no overlap and the MDCT

obtained using DCT-IV local bases, constant window length and 50% overlap.

When analysing audio signals, it is possible to adapt the various parameters to the local characteristics of the recording, switching for example between DCT and wavelet bases whenever the signal is estimated to be a steady state or transient part of the audio recording; or by dynamically adapting the window lengths of the local orthonormal transforms to the signal to be analysed. In this section this latter strategy is employed to obtain a pitch-synchronous LOT.

Pitch-synchronous transforms

When examining the frequency content of audio signals with a window-based transform, frames are extracted from the signal stream and independently analysed by employing time-frequency transforms. When analysing periodic signals, spurious frequency components are introduced whenever the windowing process extracts a fractional number of periods of the function to be analysed, because treating frames independently from each other introduces sudden jumps at the frames boundaries. For this reason it is beneficial to include in each window an integer number of periods of the function to be transformed.

The effect of different window lengths on the absolute value of the Fourier transform of a periodic function can be seen in Figure 3.1. A pure sinusoid is analysed in three ways:

- *Rectangular window, pitch synchronous*: an integer number of periods are extracted using a rectangular window. The magnitude of the corresponding Fourier transform displays a clear peak in correspondence to the frequency of the sinusoid and a steep decay in adjacent frequencies.

- *Hann window, non pitch synchronous*: a fractional number of periods are extracted using a Hann window $\boldsymbol{h} \in \mathbb{R}^N$ defined as $\boldsymbol{h}[n] \stackrel{\text{def}}{=} \frac{1}{2}\left(1 - \cos\left(2\pi \frac{n-1}{N-1}\right)\right)$. The signal is set to zero at the boundaries of the window, and the magnitude of the corresponding Fourier transform decays very quickly, but the peak around the frequency of the sinusoid is less clear.

- *Rectangular window, non pitch synchronous*: a fractional number of periods are extracted using a rectangular window. The magnitude of the corresponding Fourier

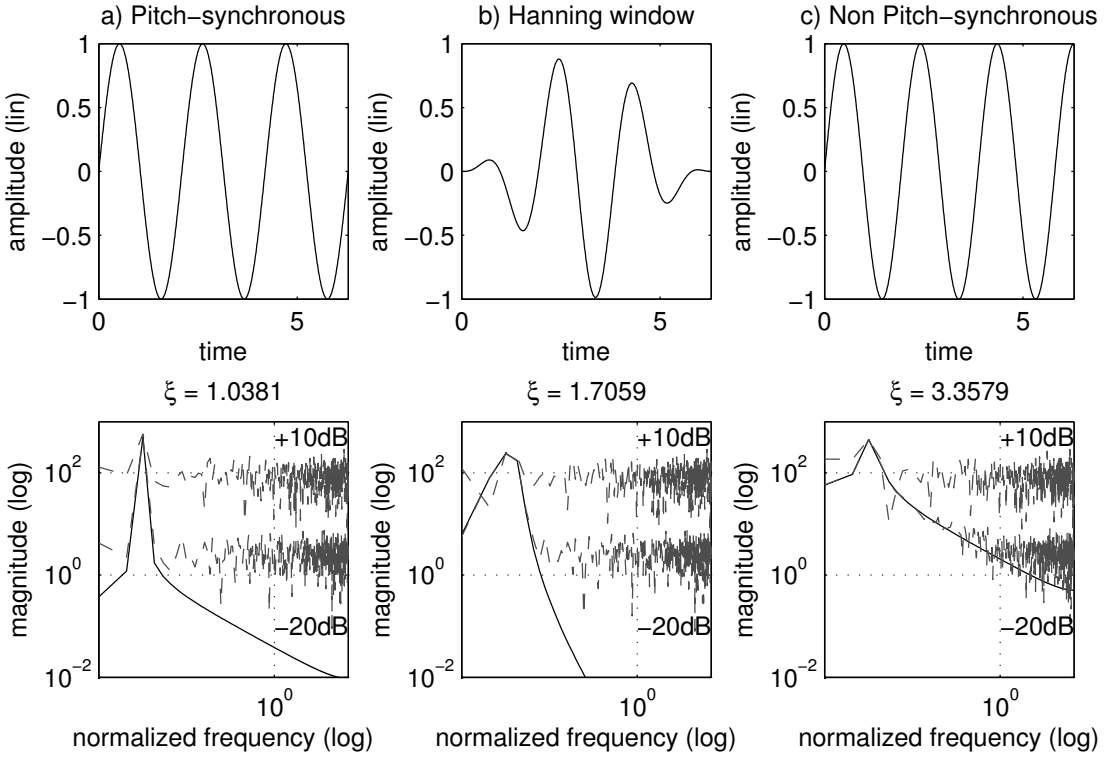


Figure 3.1: The figures display the frequency analysis of a sinusoidal function obtained by windowing the signal using different window lengths. In a) the function is analysed by windowing 3 of its periods and the magnitude of its Fourier transform in the corresponding plot clearly displays a peak in correspondence to the frequency of the sinusoid and a steep decay in adjacent frequencies. In b) the function is analysed by windowing 3.25 periods using a Hann window and the magnitude of the corresponding Fourier transform decays quickly but displays a less clear peak. Finally, in c) the sinusoid is analysed by windowing 3.25 periods using a rectangular window and the magnitude of the corresponding Fourier transform displays a clear peak but decays more slowly. The ξ sparsity measure reveals that the magnitude of the Fourier transform of a) is the sparsest, followed by b) then c). The grey lines in the lower row of plots represent the Fourier transform of the signals when corrupted by Gaussian noise at $+10$dB and -20dB respectively. As expected the pitch-synchronous transform exhibits a clearer peak leading to a better discrimination of the fundamental frequency.

transform displays a clear peak but decays slowly around it, making the frequency analysis less robust to noise or interference, as highlighted in Figure 3.1.

This toy example can be generalised to the analysis of general periodic signals analysed with any linear transform. Given the fact that periodic functions can be represented as sums of pure sinusoids through their Fourier series, and given that a linear transform of a sum of functions coincides with the sum of the transforms, a window-based linear transform that analyses a periodic signal by extracting an integer number of periods will

result in a representation that is free from spurious components. This desirable property is reflected in the compressibility of a transform, that is the rate at which the sorted magnitude of the transformed coefficients decays to zero, and is visible in the lower row of plots in Figure 3.1.

The notion of compressibility has been employed in the context of approximation theory to measure how much information about a signal is contained in a subset of its larger coefficients in a transformed domain [25]. When a signal is compressible it means that it can be reconstructed using a small number of the most significant coefficients in the transformed domain leading to a small approximation error. This is the principle at the root of sparse approximation as it is defined in Section 2.3 and, in particular, in equation (2.12).

Compressibility can be therefore associated to sparsity, and one simple measure of sparsity proposed in the literature is defined as the ratio between the ℓ_1 and the ℓ_2 norms of the coefficients in the transformed domain [109]:

$$\xi(\boldsymbol{v}) = \frac{||\boldsymbol{v}||_1}{||\boldsymbol{v}||_2} \tag{3.1}$$

where $\boldsymbol{v}$ is a vector in $\mathbb{R}^N$ or $\mathbb{C}^N$. The rationale behind this measure is that, for realistic, possibly noisy signals, the transformed coefficients will rarely be exactly zero, but will follow a distribution that exhibits a strong peak around values very close to zero. The ℓ_1 norm is a good measure of this approximate sparsity because it is the closest convex approximation of the ℓ_0 sparsity objective (a fact that also motivates the convex relaxation algorithms for sparse approximation detailed in Section 2.4.2). In Figure 3.1 the ξ sparsity measure is indicated, showing that the rectangular window pitch synchronous transform is the most compressible, followed by the Hann window non pitch synchronous and by the rectangular window non pitch synchronous transforms.

Background work on pitch-synchronous audio transforms

Previous work on pitch-synchronous transforms include a method proposed by Abad [1] for note detection in a polyphonic musical mix. In this algorithm a wavelet bases is designed so that the atoms' frequencies are located at the discrete intervals defined by the equal tempered tuning employed by instruments in western music. Evangelista [33], on the other hand, devised a pitch-synchronous wavelet transform where the atoms

are explicitly modelled on the periodic components of the signal to be analysed. Both the mentioned pitch-synchronous wavelet transform and the novel pitch-synchronous LOT algorithm detailed in the next Section rely on estimating the frequency content of a signal to be analysed and use this information as a parameter of the relevant transform. The main difference consists in that the proposed method employs LOTs rather than wavelets.

3.1.1 The pitch-synchronous LOT algorithm

A LOT can be used for a frame-by-frame analysis of periodic signals and the length of each frame can be freely adjusted as long as the overlap between consecutive windows does not exceed 50% of the window length. The main idea behind the pitch-synchronous LOT is to locally adapt the window lengths to the pitch of the signal to be analysed, in order to have local orthonormal transforms of signals containing an integer number of periods.

To this aim, given a pitched monophonic audio signal, a pitch estimation algorithm can be employed to extract the fundamental frequency of the signal at each instant in time and used to define a set of window lengths that are used as parameters of the LOT. Figure 3.2 visually displays the main stages of the pitch synchronous LOT transform, and Algorithm 7 details them in a pseudo-code format.

Let $\boldsymbol{y} \in \mathbb{R}^N$ be the signal to be analysed, w_0 an initial window length and ϵ a tolerance level, the algorithm proceeds through the following steps:

I - Compute a function $\boldsymbol{f} \in \mathbb{R}^N$ that contains estimates of the fundamental frequency of the audio signal at each sample in time and a relative salience function $\boldsymbol{g} \in \mathbb{R}^N$ that indicates how reliable the estimation is at each sample (discarding, for example, portions of audio that are silent or very noisy). Any suitable pitch estimation function can be utilised for this purpose, in the present implementation of the pitch synchronous LOT, the methods by Klapuri and Virtanen have been chosen for this task [55, 56].

II - Threshold the function $\boldsymbol{f}$ so that only reliable estimates are kept while setting the unreliable frequency estimates to zero. This is done by using the tolerance

parameter ϵ that is given as an input to the algorithm.

$$\tilde{f}_n \stackrel{\text{def}}{=} \texttt{Thresh}(f_n) = \begin{cases} f_n & \text{if } \; g_n \geq \epsilon \, ||\boldsymbol{g}||_\infty \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

III - Compute an adaptive partitioning of the time axis by defining a set of window lengths $\boldsymbol{w} \in \mathbb{R}^Q$ that are adapted to the fundamental frequency of the signal. This is accomplished through the following steps:

- Define two index counters n and q for the vectors $\tilde{\boldsymbol{f}}$ and $\boldsymbol{w}$ respectively.
- Start by assigning the current window length w_q to the fixed reference length w_0 that is defined as an input to the algorithm.
- Define a time interval that starts from the current pointer n and extends for a length given by w_q. If the norm of $\tilde{\boldsymbol{f}}$ in this interval is greater than zero (that is, following from the previous thresholding step, if there are reliable frequency estimates in this interval), adjust the current window length by performing iteratively the following:
  - Calculate the average frequency in the interval $[n, n + w_q]$:

$$\bar{f} = \frac{1}{w_q} \sum_{i=n}^{n+w_q} f_i \tag{3.3}$$

  - Adjust the current window length w_q by calculating the closest number of samples to w_0 that contain an integer number of periods based on $\bar{f}$.

$$\bar{w}_q = \left\lfloor \left\lfloor w_0 \bar{f}/\text{Fs} \right\rfloor \text{Fs}/\bar{f} + 0.5 \right\rfloor \tag{3.4}$$

    where $\lfloor \cdot \rfloor$ indicates the floor operation that returns the largest integer smaller or equal than its argument and Fs is the sampling frequency of the signal $\boldsymbol{y}$.
- Update the index q by adding one and the index n by adding w_q.

IV - Perform a LOT analysis of the signal $\boldsymbol{y}$ using the vector of window lengths $\boldsymbol{w}$ as the parameter governing the length of the local orthonormal transforms. Note

Algorithm 7: Pitch-synchronous analysis of audio signals.

Input: $\boldsymbol{y}, w_0, \epsilon, \mathrm{Fs}$
Output: $\boldsymbol{x}$
// Calculate $f0$ and salience functions
$[\boldsymbol{f}, \boldsymbol{g}] \leftarrow \texttt{PitchEst}(\boldsymbol{y})$;
$\boldsymbol{f} \leftarrow \texttt{Thresh}(\boldsymbol{f}, \boldsymbol{g}, \epsilon)$;
// Calculate adaptive partitioning
$n \leftarrow 1$;
$q \leftarrow 1$;
while $n \leq \texttt{length}(\boldsymbol{y}) - w_0$ **do**
  $w_q \leftarrow w_0$;
  if $||\boldsymbol{f}[n : n + w_0]|| > 0$ **then**
    for *i=1:10* **do**
      $\bar{f} \leftarrow \frac{1}{\boldsymbol{w}[q]} \sum \boldsymbol{f}\,[n : n + w_q]$;
      $w_q \leftarrow \left\lfloor \left\lfloor w_0 \bar{f}/\mathrm{Fs} \right\rfloor \mathrm{Fs}/\bar{f} + 0.5 \right\rfloor$;
    end
  end
  $n \leftarrow n + w_q$;
  $q \leftarrow q + 1$;
end
$\boldsymbol{x} \leftarrow \mathrm{LOT}(\boldsymbol{y}, \boldsymbol{w})$

that the type of local orthonormal transforms and their overlap are parameters not addressed by this algorithm and can be specified according to the nature of the signal to be analysed. However, in the case of overlapping windows the length of the transforms should be updated accordingly, in order to ensure that an integer number of periods of the signal are analysed in each window.

The main motivation driving the design of a pitch-synchronous LOT consists in obtaining a compressible or sparse representation of periodic signals. This is the focus of the next section which describes experimental results on a periodic audio signal.

3.2 Sparsity of LOTs

This section deals with the evaluation of the sparsity of different LOTs for the analysis of a periodic audio signal. An oboe recording taken from the Iowa musical instruments dataset[1] was used for evaluation. The audio sample consists in a chromatic scale recorded with sampling rate at 44100Hz and quantised at 16 bits per sample. An oboe was chosen because its regular harmonic structure and the absence of vibrato makes it particularly amenable to a sparse modelling. Experiments with complex multi-track recordings which

[1] http://theremin.music.uiowa.edu/MIS.html

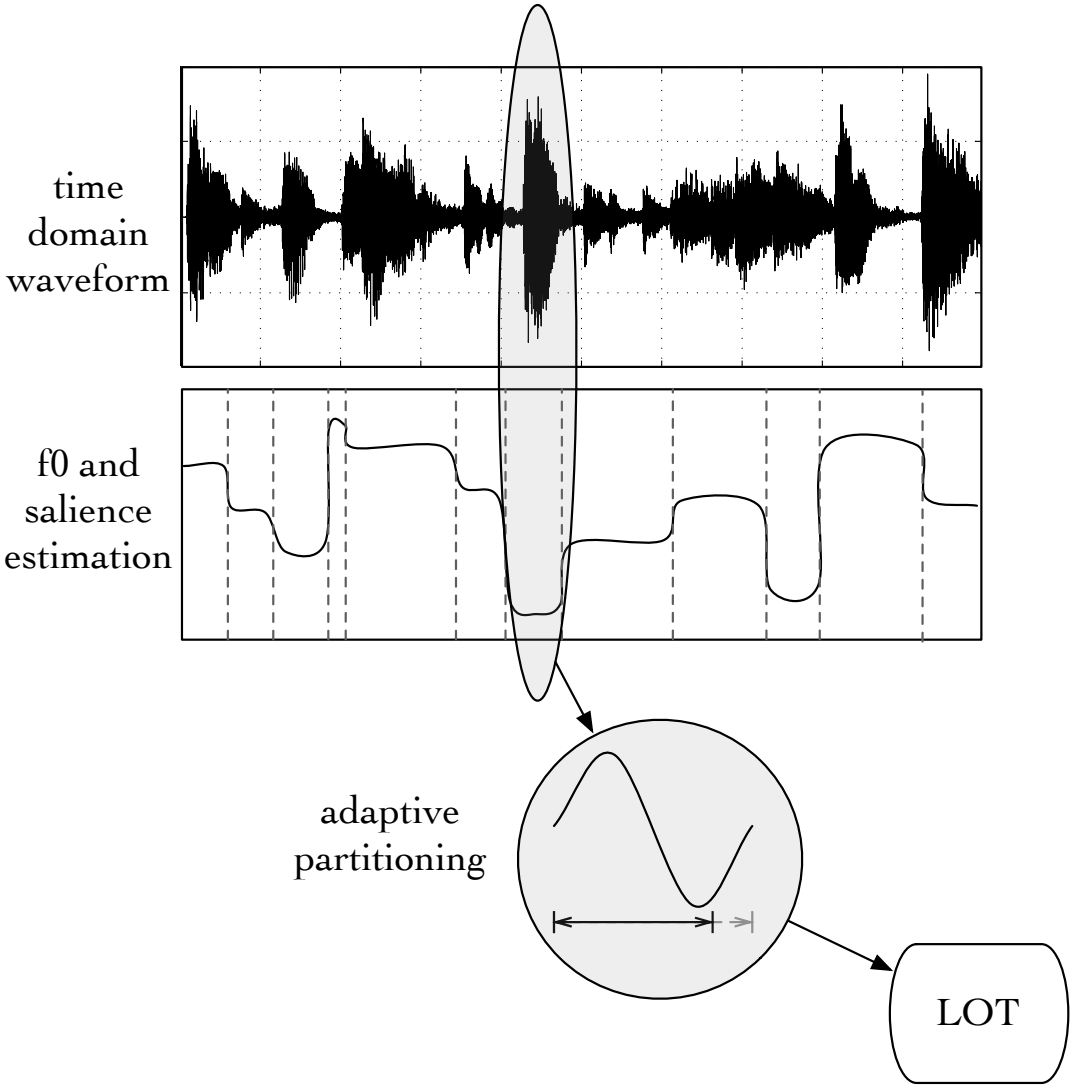


Figure 3.2: Pitch-synchronous analysis of an audio signal. The time-domain waveform is analysed by a pitch estimation algorithm and an $f0$ track is produced which estimates the fundamental frequency of the signal at each instant in time. The $f0$ track is complemented by a salience function that describes the reliability of the estimation, discarding for examples estimates on silent or noisy portions of the audio signal. An adaptive partitioning of the time axis is performed by adjusting window lengths to contain an integer number of periods of signal based on the estimated fundamental frequency. The set of window lengths are finally utilised as a parameter of the LOT where the length of the local orthonormal bases are adapted to the pitch of the audio signal.

constitute a more realistic dataset will be described in the next section. In all the transforms we used windows of 2048 samples (corresponding to about 46ms, a duration that is suitable for analysing harmonic structures in the audio signals).

Figure 3.3 shows the results of the experiment that are summarised in table 3.1:

The waveform in the time domain has a sparsity measure $\xi \approx 660$. As soon as a

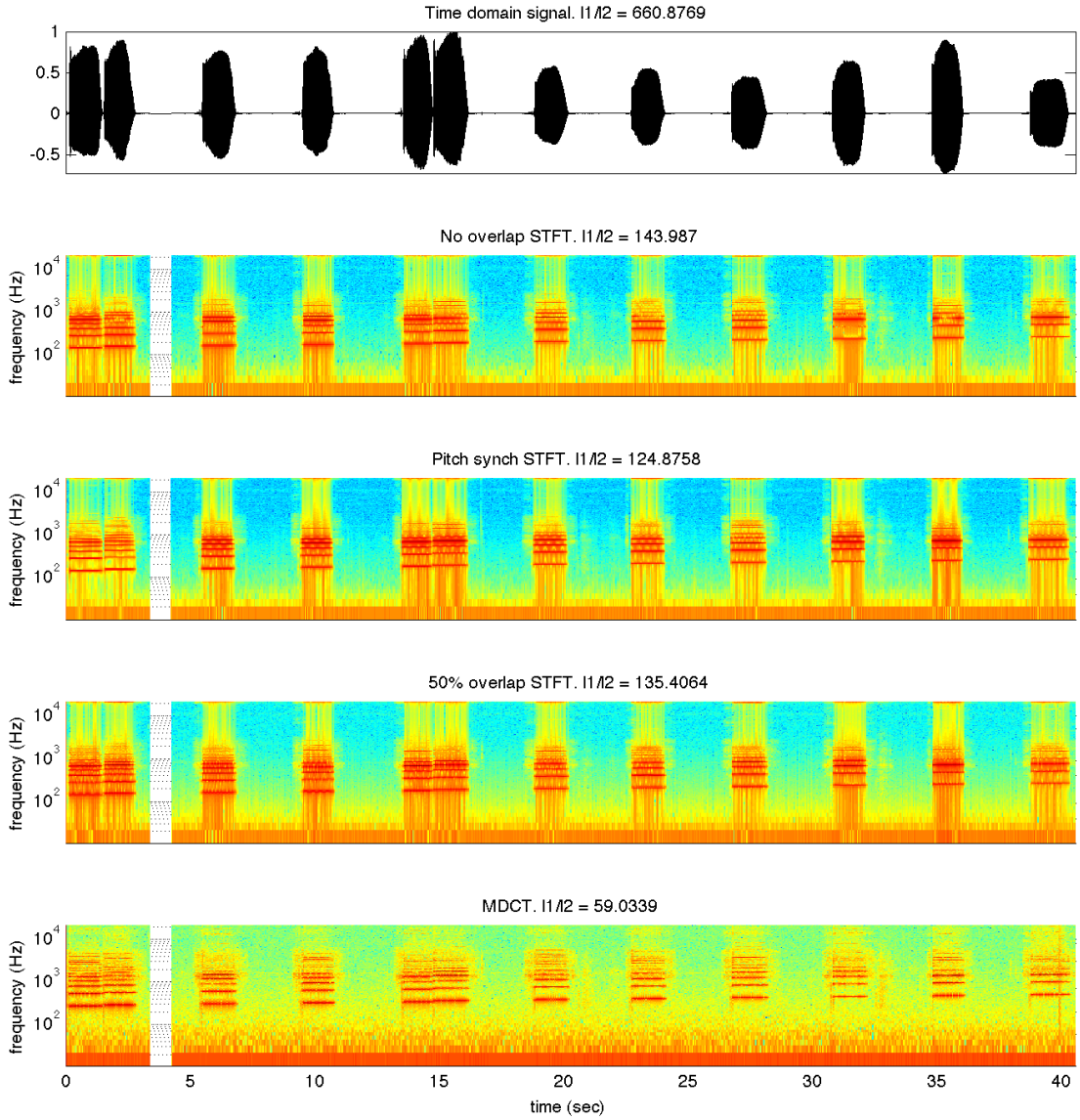


Figure 3.3: Sparsity of different LOTs applied on an oboe recording

STFT with no overlapping windows is applied, the sparsity measure drops to $\xi \approx 144$. The harmonic partials of the oboe notes are visible, although blurred by the spurious components introduced by the window artefacts. A STFT with 50% overlapping windows achieves a sparsity level $\xi \approx 135$, while a non overlapping STFT with adaptive windows length achieves a slightly better sparsity $\xi \approx 125$. In this case, the proposed pitch-synchronous LOT was used. The best sparsity is by far achieved when analysing the audio recording using MDCT. In this case ξ drops to approximately 59. As can be seen, the structure of the harmonic partials is clear and the windowing artefacts are not prominent.

The results obtained analysing the oboe recording indicate that the proposed pitch

Transform	**Overlap**	**Window Length**	ξ	$\tilde{H}$
Identity	n.a.	n.a.	661	12.4
STFT	0	Constant	144	0.7
STFT	50%	Constant	135	0.6
STFT	0	Pitch-synchronous	125	0.5
DCT-IV	50%	Constant	59	0.1

Table 3.1: Sparsity index ξ and empirical entropy $\tilde{H}$ of different LOTs applied to an oboe recording.

synchronous LOT leads to a more compressible representation if compared to a non adaptive STFT with zero or 50% overlap. However, it is outperformed by the MDCT which achieves a much more sparse representation (as measured by the sparsity index ξ). Additional informal experiments indicated that realizing a pitch-synchronous MDCT does not have a noticeable impact in the sparsity of the transformed coefficients, and indicates that for the purpose of analysis or approximation, MDCT should be regarded as the preferred choice when dealing with audio signals.

In addition to the ξ sparsity measure, the entropy H of the transformed signal was evaluated. This quantity has been used in information theory to describe the information content of a signal [98], and it is defined for a probability distribution $p(x)$ as follows:

$$H = -\int p(x) \log_2 p(x). \tag{3.5}$$

When observing a vector of coefficients $\boldsymbol{x}$, as returned by the LOTs algorithm, empiric probabilities can be computed by considering a histogram $h_l \in \mathbb{R}^L$ that counts the number of occurrences of the coefficients that fit in a range indexed by l (for the purpose of this experiment the number of value ranges was set to $L = N$). A vector of empiric probabilities $\tilde{p}_l = h_l/N$ can be obtained by dividing the number of occurrences by the size of the vector. A discretised version of equation (3.5) can be defined as:

$$\tilde{H} = -\sum_{\tilde{p}_l > 0} \tilde{p}_l \log_2 \tilde{p}_l \tag{3.6}$$

where the sum is only taken on non-zero values of $\tilde{p}_l$, i.e. values that occur at least once.

The sparsity of a signal is related to its entropy, since a large number of coefficients that are exactly (or approximately) zero and a small number of significative coefficients

leads to a small entropy $\tilde{H}$. Table 3.1 confirms this trend showing that the smallest entropy is reached by the MDCT transform. Sparsity and entropy can be associated to the cost of coding a signal (as measured by the bits needed to represent its symbols or coefficients), and it is small for representations with small $\tilde{H}$. It is worth mentioning that, in the case of a pitch-synchronous transform, the cost of coding must also take into account the overhead needed to store the windows locations.

The results presented in this section do not necessarily imply that a strategy consisting in adaptively adjusting the parameters of a LOT to the properties of the signal to be analysed should be generally discarded in favour of a standard MDCT. Other adaptive transforms can be designed starting from the framework of LOTs, for example by adjusting the type of the local orthonormal transforms. Additionally, further investigation can be carried out to better understand why an adaptive MDCT does not significantly outperform a non-adaptive one.

These are interesting problems that are for the moment being left for future investigation. The reminder of this chapter focuses on studying LOTs in the context of source separation, which is one of the most popular and widely studied problems in audio signal processing.

3.3 Measuring disjointness of time-frequency representations

3.3.1 Source separation

Source separation [18] is a classic signal processing application that deals with analysing some observable signals to extract a set of unknown sources based on a given mixing model. Let $\left\{\boldsymbol{s} \in \mathbb{R}^N\right\}_{m=1}^{M}$ be a set of source signals and $\left\{\boldsymbol{y} \in \mathbb{R}^N\right\}_{p=1}^{P}$ a set of mixtures. A so-called instantaneous mixing model can be written as:

$$\boldsymbol{y}_p = \sum_{m=1}^{M} a_{m,p} \boldsymbol{s}_m \tag{3.7}$$

where $a_{m,p}$ is the mixing weight describing the contribution of the m-th source to the p-th mixture.

A more accurate mixing model for describing how audio sources contribute to observable mixtures is the convolutive mixing model, where a set of impulse responses $\{\boldsymbol{h}_{m,p}\}$ are introduced to describe the acoustic path leading from the m-th source to the p-th

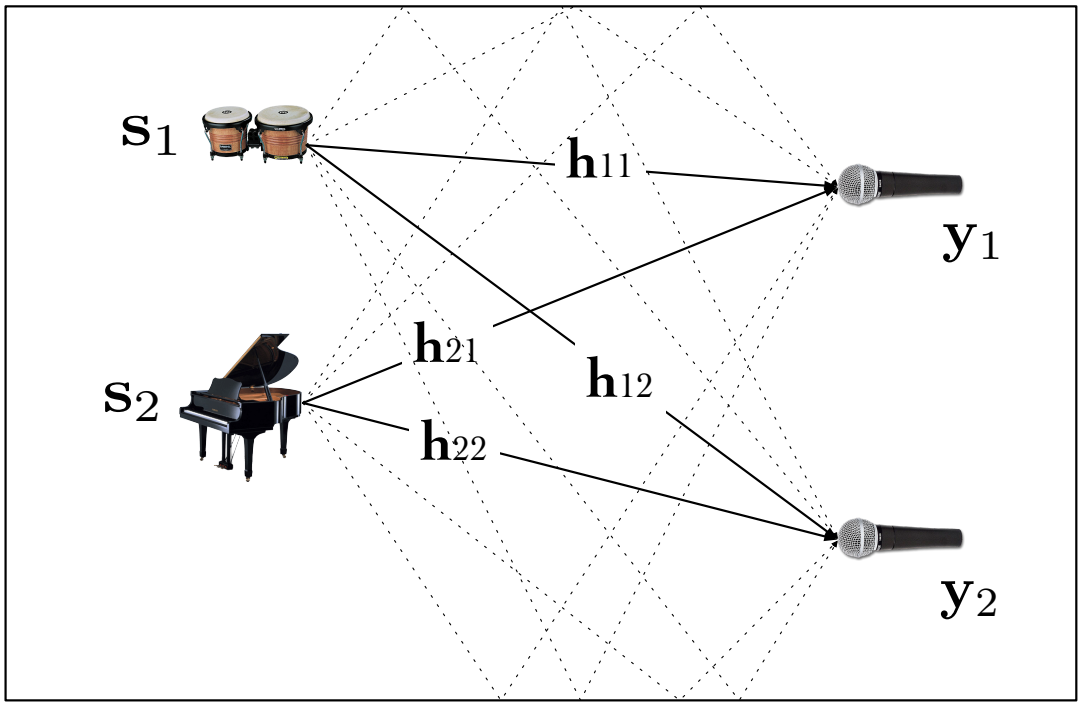


Figure 3.4: Convolutive mixing model. A set of source signals are recorded using a set of microphones to obtain observed mixtures. An impulse response which models the acoustic reflections in the ambient is associated to each acoustic path that goes from each source to each microphone.

mixture:

$$\boldsymbol{y}_p = \sum_{m=1}^{M} \boldsymbol{h}_{m,p} * \boldsymbol{s}_m. \tag{3.8}$$

The convolutive mixing model is displayed graphically in Figure 3.4.

Whenever the mixing weights or the impulse responses are unknown the source separation problem is said to be *blind*. A further distinction is usually introduced with regards to the number of sources and mixtures: if $M \leq P$ the source separation is said to be *overdetermined* or *exactly determined*, while if $M > P$ the problem is described as *underdetermined*. This nomenclature comes from the fact that a source separation is an inverse problem whose determinacy depends on the relative number of unknowns and equations.

Considering the simple instantaneous mixing model, for example, let $\boldsymbol{S} \in \mathbb{R}^{N \times M}$ be a matrix of source signals containing the vectors $\boldsymbol{s}_m$ in each of its columns and let $\boldsymbol{Y} \in \mathbb{R}^{N \times P}$ contain the observed mixtures in each of its columns. Then a mixing matrix

$\boldsymbol{A} \in \mathbb{R}^{M \times P}$ can be defined in order to write (3.7) in a compact matrix notation:

$$\boldsymbol{Y} = \boldsymbol{S}\boldsymbol{A}. \tag{3.9}$$

Given the mixing matrix (or its estimate $\tilde{\boldsymbol{A}}$ in the case of a blind source separation), estimating a set of source signals is done by solving the inverse problem:

$$\boldsymbol{S}^{\star} = \underset{\boldsymbol{S} \in \mathbb{R}^{N \times M}}{\arg\min} \left|\left| \boldsymbol{Y} - \boldsymbol{S}\tilde{\boldsymbol{A}} \right|\right|_{\mathrm{F}}. \tag{3.10}$$

Note that the equation above has an analytic solution $\boldsymbol{S}^{\star} = \boldsymbol{Y}\tilde{\boldsymbol{A}}^{\dagger}$ if $\boldsymbol{A}$ is square and invertible (i.e., only if the problem is determined). More generally, and specifically when dealing with under-determinate source separation, the source signals can be estimated as $\boldsymbol{s}_m^{\star} = \mathcal{F}_m(\boldsymbol{Y})$ where $\mathcal{F}_m$ is a suitable function used to extract the m-th source from the matrix of observed mixtures. This is the approach followed by binary masking algorithms.

3.3.2 Underdetermined blind source separation by binary masking

Source separation problems are particularly challenging when the mixing matrix is unknown and the number of sources is greater than the number of mixtures. In particular, the estimation of a set of sources from a single mixture will be considered from now on, and in this underdetermined blind setting a popular strategy that has proved to be successful in the case of audio source separation is the use of binary masks [122].

The overall structure of a binary masking algorithm can be described as follows: a mixture $\boldsymbol{y}$ is mapped by a linear operator $\mathcal{T}$ into a transformed domain (usually a time-frequency or time-scale representation), and M binary masks $\{\mathcal{M}_m\}_{m=1}^{M}$ are defined in order to extract the source signals. After each mask has been applied in the transformed domain, the corresponding source $\boldsymbol{s}_m^{\star}$ can be estimated by inverting the transform:

$$\boldsymbol{s}_m^{\star} = \mathcal{F}_m(\boldsymbol{y}) = \mathcal{T}^{-1}\left(\mathcal{M}_m \mathcal{T}(\boldsymbol{y})\right). \tag{3.11}$$

Choosing a suitable transform is crucial for the success of the source separation algorithm. In particular, the operator $\mathcal{T}$ should lead to a representation where the coefficients belonging to different sources do not overlap with each other, so that a suitable mask can be used to extract each source with the maximum fidelity and the minimum possible

interference from the other ones.

W-Disjoint orthogonality as a measure of disjointness

Let us consider a linear mixture of M sources $\boldsymbol{y} = \sum_{m=1}^{M} \boldsymbol{s}_m$. The transformed mixture can be written as:

$$\mathcal{T}(\boldsymbol{y}) = \sum_{m=1}^{M} \mathcal{T}(\boldsymbol{s}_m) \tag{3.12}$$

We can define an oracle mask $\mathcal{M}_m^\star$ for each source (given that the individual sources are available) as:

$$\mathcal{M}_m^\star = \begin{cases} 1 & \text{if } |\mathcal{T}(\boldsymbol{s}_m)| > |\mathcal{T}(\boldsymbol{z}_m)| \\ 0 & \text{otherwise} \end{cases} \tag{3.13}$$

where $\boldsymbol{z}_m \stackrel{\text{def}}{=} \sum_{j \neq m} \boldsymbol{s}_j$ is the mixture of the sources interfering with $\boldsymbol{s}_m$.

Given a mask $\mathcal{M}_m$ and the original sources, we can measure the preserved signal ratio (PSR) defined as the portion of the energy of the m-th source that is preserved after the masking operation in the transformed domain:

$$\text{PSR}_m = \frac{||\mathcal{M}_m(\mathcal{T}(\boldsymbol{s}_m))||_2^2}{||\mathcal{T}(\boldsymbol{s}_m)||_2^2} \tag{3.14}$$

and the signal to interference ratio (SIR), which measures the amount of interference caused by the interfering sources after the masking:

$$\text{SIR}_m = \frac{||\mathcal{M}_m(\mathcal{T}(\boldsymbol{s}_m))||_2^2}{||\mathcal{M}_m(\mathcal{T}(\boldsymbol{z}_m))||_2^2} \tag{3.15}$$

Finally, the W-disjoint orthogonality (WDO) is defined as:

$$\text{WDO}_m = \text{PSR}_m - \frac{\text{PSR}_m}{\text{SIR}_m}$$

For reference purpose, we keep the term *w-disjoint orthogonality* that was originally coined by the authors in [122] to emphasise the dependance of the disjointness on the window used in a STFT, even though the model presented in this section is more general because it refers to any linear invertible transform.

In an ideal situation, the PSR tends to 1 and the SIR tends to infinity, leading to a WDO ≈ 1, while a WDO value close to zero or negative indicates that the ideal mask

extracts equal or more energy from the interference rather than from the desired source.

It is worth stressing that WDO is an oracle measure that requires the original source signals in order to evaluate if a given transform leads to a representation where the sources can be separated. Therefore, a WDO close to one should be considered as a necessary but not sufficient condition for the success of source separation by binary masking. Nonetheless, it provides with a useful estimate of how well a transform can perform for this given task.

3.3.3 Experimental setting and results

The following transforms were tested and compared in terms of the disjointness of their representations.

Short time Fourier transform (STFT) : 50% frame overlap and a Hamming window were employed since they are a common choice of parameters that ensure the invertibility of the transform using inverse Fourier transform and overlap-add synthesis.

Constant Q transform (CQT) [13]: a time-frequency representation similar to STFT but with logarithmic frequency resolution so that the Q-factors (ratios of the centre frequencies to bandwidths) of all the frequency bins are the same. This is suitable for pitched audio signals because the fundamental frequency and and the frequency of the harmonics of most instruments are logarithmically spaced.

Pitch-synchronous STFT : the method proposed in Section 3.1 realized as a particular instance of LOT where the time-domain signal is analysed by using windows whose length is adapted to its pitch. Only the bass guitar was chosen to be analysed with a pitch-synchronous transform because it consists of a periodic, usually monophonic signal whose pitch can be reliably estimated.

Modified discrete cosine transform (MDCT) : realized as a particular instance of LOT with fixed window size, 50% overlap and DCT-IV local bases.

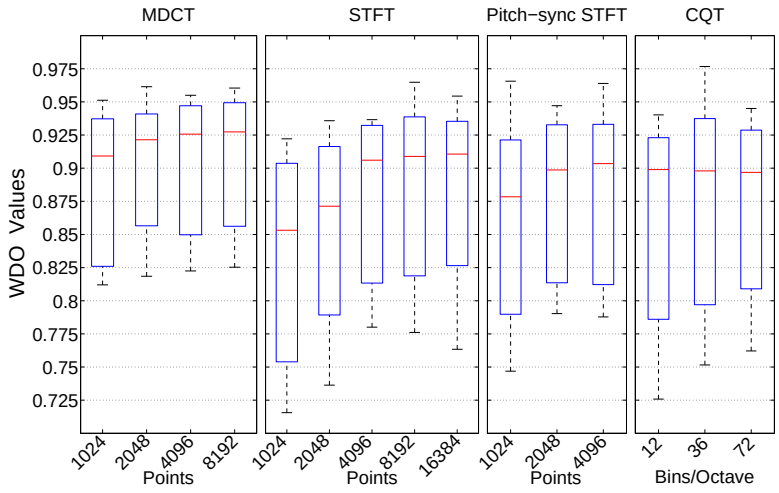
We used a dataset of 18 multitrack songs of various genres from pop-rock to heavy metal so as to have a representative and heterogeneous collection of modern popular music. The tracks we focused on per song are: guitar, bass, drums and vocals. For measuring the disjointness the measurements were performed on random 2.9 seconds segments from the

songs that were previously normalized to unit energy. During the selection, we ensured that none of the sources to be measured were silent in the segment.

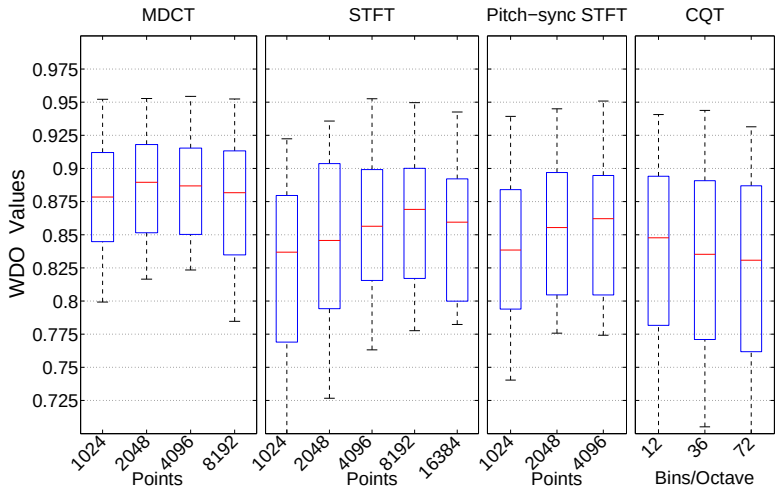
Figure 3.5 shows the boxplot of the WDO measured for different pairs of instruments from all the songs in the dataset (here the pitch-synchronous STFT only appears with pairs including the bass guitar). Focusing on the median disjointness, we can observe that MDCT outperforms the other transforms in all cases and that, within the MDCT subplots, a window length of 2048 samples (about 46ms given that the sampling rate is 44.1kHz) leads to the best results in most of the cases, except for when analysing the bass which, being a periodic and low frequency signal, benefits from a higher frequency resolution allowed by longer windows. Moreover, we can observe that certain pairs of instruments like bass and vocals or drums and vocals are more disjoint than the rest, which can be explained by the fact that these instruments are usually not highly harmonically or rhythmically correlated.

Given the wide range of musical genres considered in the evaluation, the data presented in Figure 3.5 exhibit a high variance. For this reason, in Figure 3.6 we present WDO measurements relative to the value achieved by a MDCT with 1024 samples windows. This allows a comparison of the improvement or decrease in disjointness achieved by the various methods with respect to the reference in each song. In these plots, the same trends just described can be identified, but the variance of the data is much smaller, showing that our evaluation is statistically significant if we consider relative measures. In other words, the plot highlights that it is possible to achieve a consistent improvement in the WDO of pairs of instruments by using MDCT with a 1024 samples window length rather than any other of the tested transforms.

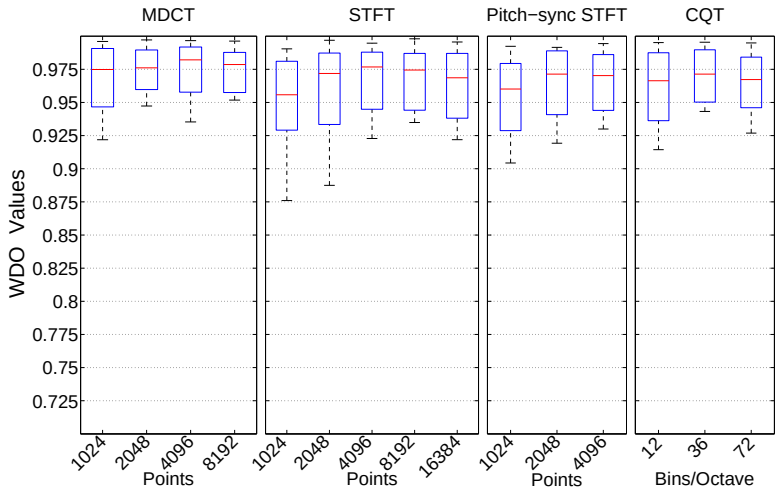
WDO measurements were also conducted for each track considering the interference resulting from all the other instruments (rather than simply on pairs of instruments as in the previous results). In this case, we observed that the median disjointness is around values contained in the range $[0.6, 0.7]$, a decrease between 16% and 28% with respect to the WDO $= 0.843$ reported for a four tracks speech mixture [122]. Again, this can be explained by the harmonic and rhythmic correlation present in most musical recordings and absent in independent speech tracks.



(a) guitar/bass

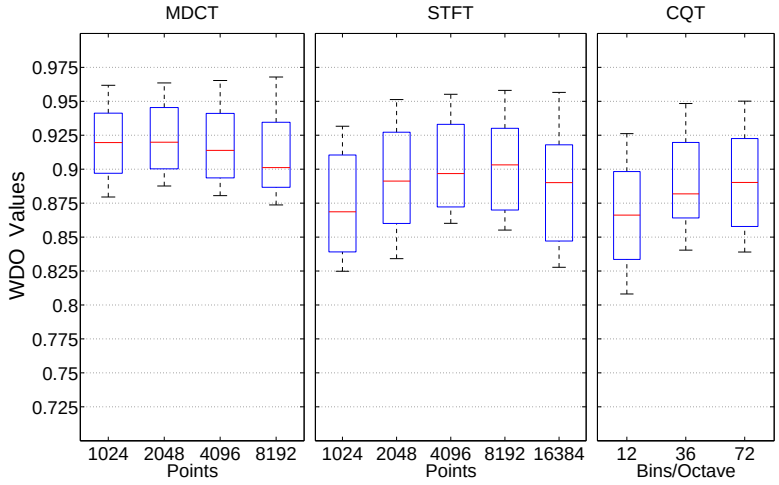


(b) drums/bass

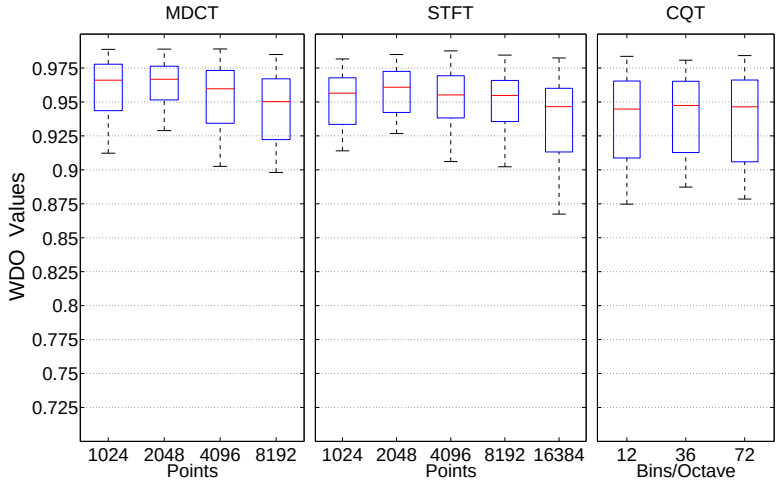


(c) bass/vocals

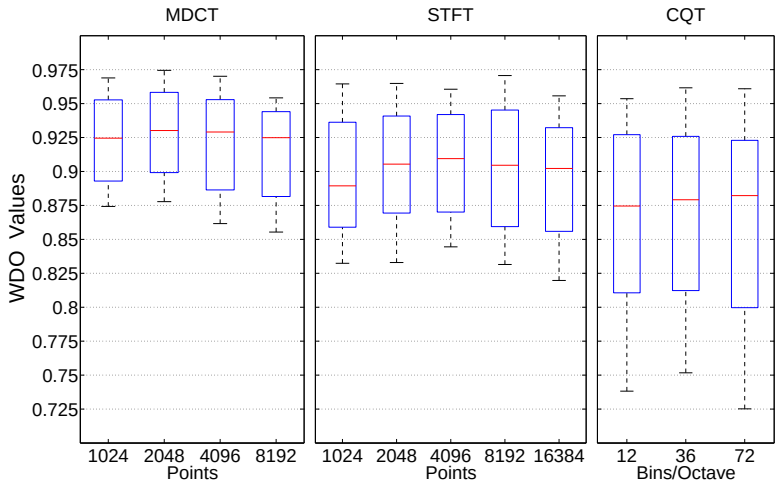
Figure 3.5: WDO measurements of different pairs of instruments for various transforms. The central mark of the boxplots is the median, the edges of the boxes are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.



(d) guitar/vocals

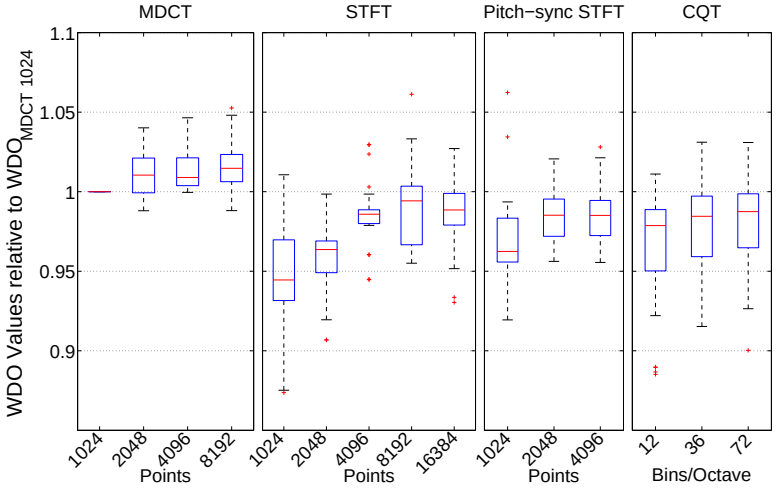


(e) drums/vocals

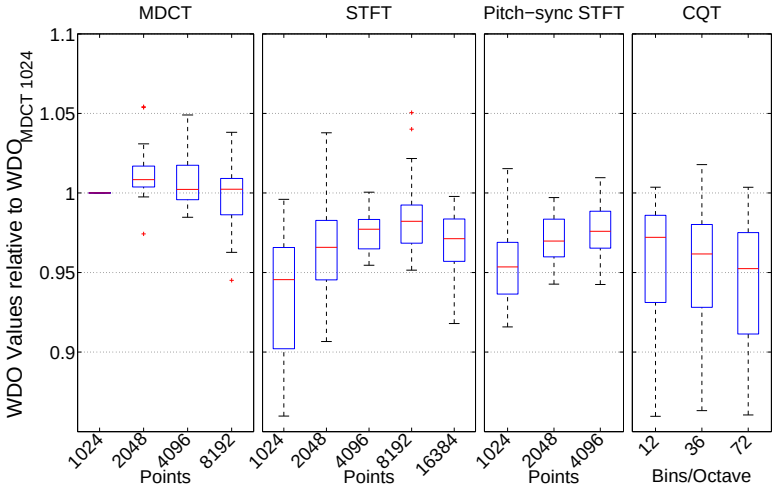


(f) guitar/drums

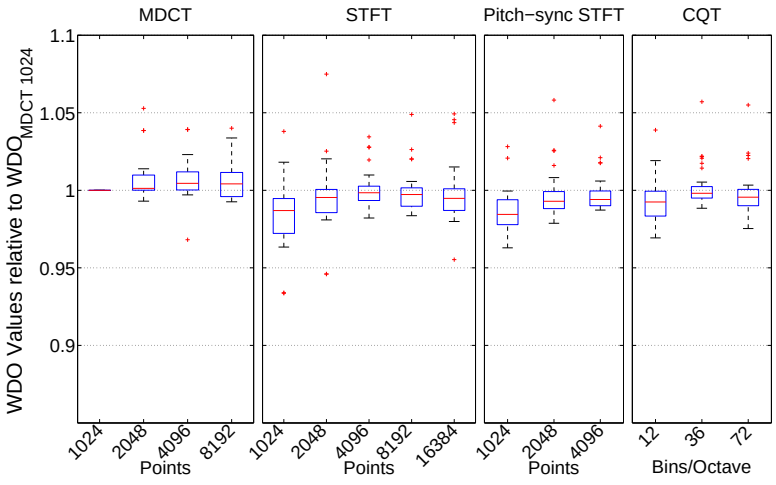
Figure 3.5: (continued) WDO measurements of different pairs of instruments for various transforms. The central mark of the boxplots is the median, the edges of the boxes are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.



(a) guitar/bass

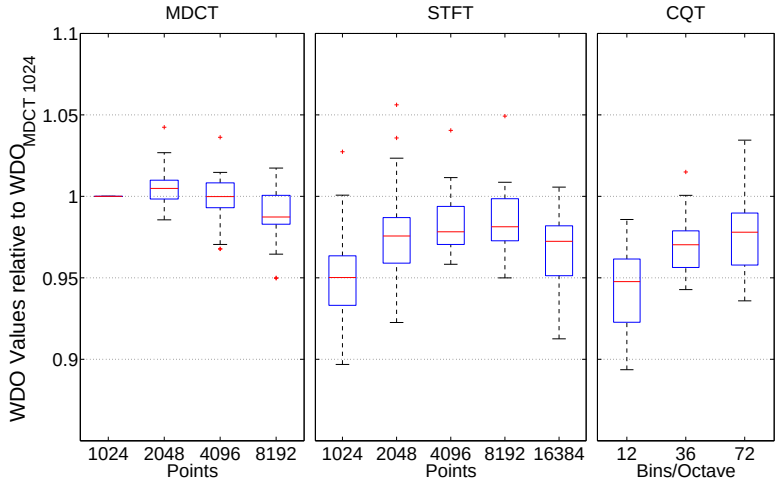


(b) drums/bass

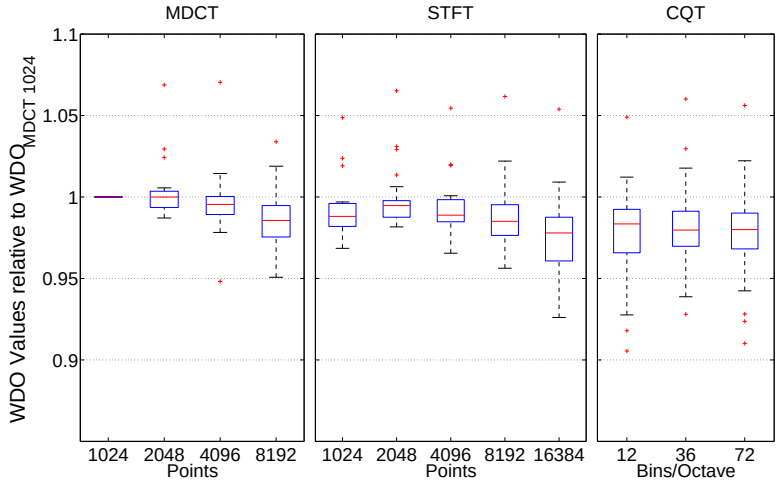


(c) bass/vocals

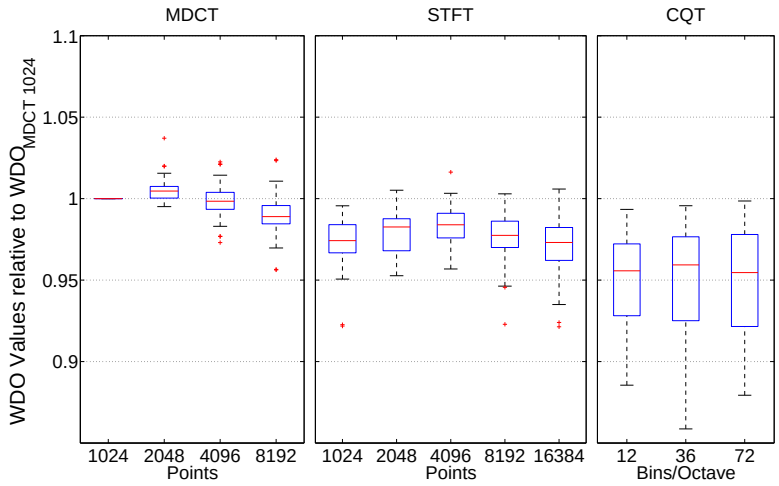
Figure 3.6: Ratios of WDO measurements relative to $\text{WDO}_{\text{MDCT1024}}$ for different pairs of instruments. The central mark of the boxplots is the median, the edges of the boxes are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.



(d) guitar/vocals



(e) drums/vocals



(f) guitar/drums

Figure 3.6: (continued) ratios of WDO measurements relative to $\text{WDO}_{\text{MDCT1024}}$ for different pairs of instruments. The central mark of the boxplots is the median, the edges of the boxes are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.

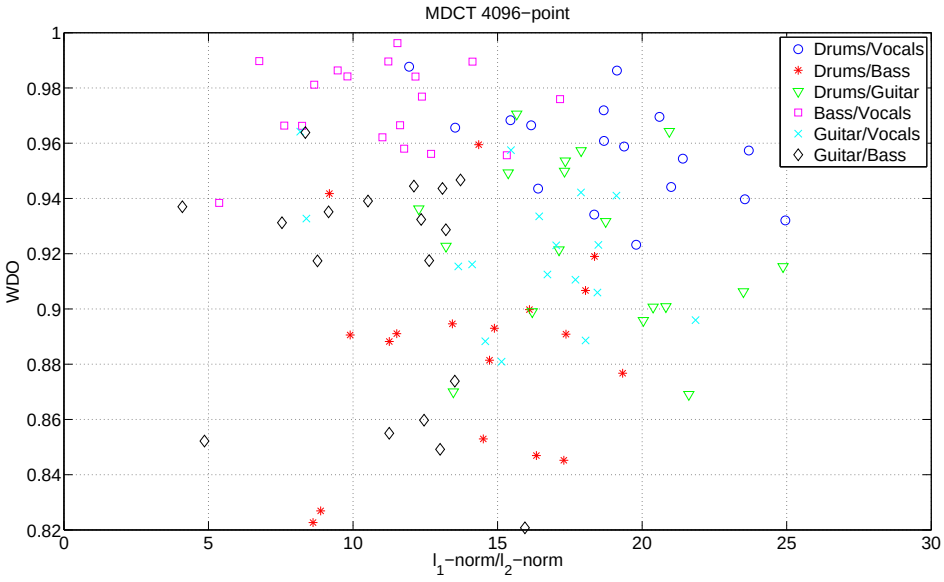


Figure 3.7: Correlation between sparsity and disjointness for different pairs of instruments. Different markers correspond to different pairs of instruments and the points for each marker type are different audio tracks taken from the database used in the evaluation described in Section 3.3.3. The ξ measure appearing on the x-axis is the average sparsity measures of the pair of instruments considered.

3.3.4 Correlation between sparsity and disjointness

Additional analysis was carried out on the mixtures of pairs of musical tracks to assess the correlation between disjointness and sparsity, revealing that there is not any significant correlation between the two quantities.

Considering MDCT with a window size of 4096 samples, both the WDO between different pairs of instruments and the sparsity measure ξ were computed in every frame singularly for each of the two instruments in a pair. The two measures were then averaged over the total number of frames and over the two instruments. The scatter plot in Figure 3.7 displays the results obtained.

As can be noted, no significant correlation can be identified between the two quantities, suggesting that in the analysis of musical audio signals using MDCT with a 4096 window size, a sparse representation does not imply a disjoint one and vice versa. This result contradicts a common assumption made in the source separation community which links the sparsity of a transform to the degree of disjointness of different sources in the transform domain (see [58, 122] for example).

Although additional investigation is required to understand the relation between ξ

and WDO and, more in general, between the sparsity of a transform and its effect on source separation performance, a possible explanation of the results obtained considers the nature of musical audio signals. Unlike speech signals that are analysed in a vast number of source separation studies including [122], the musical audio signals exhibit harmonic spectra whose components are much more likely to overlap between different pairs of instruments. Therefore, a transform that provides a sparse representation of such components does not result in a disjoint transform if the components naturally overlap.

3.4 Summary

In this chapter a novel pitch-synchronous transform LOT has been presented. A quasi-periodic signal can be defined as a function that is periodic on time intervals of limited duration. The design of the pitch-synchronous LOT aims at realizing a frame-by-frame analysis of a quasi-periodic signal where the length of every frame is adapted to the frequency of the signal in order to contain an integer number of its periods.

The proposed method utilises a pitch estimation step that can be carried out with any suitable algorithm and is used to infer a set of window lengths to be inputed as a parameter of a LOT. The framework of LOTs was chosen as the starting point for the realization of the pitch-synchronous transform because of its great flexibility, its fast implementation realized through a window-based algorithm and its wide use in the analysis of audio signals.

Numerical examples on a pure sinusoidal signal and on a monophonic oboe recording showed that a non-overlapping, pitch-synchronous STFT achieves a sparser representation of the signals (as measured in terms of the ratio between the ℓ_1 and the ℓ_2 norms of the representation coefficients) if compared to a non-pitch synchronous STFT. However, the pitch-synchronous transform is outperformed by a MDCT, and realizing a pitch-synchronous MDCT does not lead to significant improvements like in the case of the STFT counterpart. This negative result is worth of further investigation.

Different LOTs including MDCT, standard STFT and the proposed pitch-synchronous STFT have been compared along with the CQT transform to assess the disjointness of their representation coefficients in the context of undetermined source separation by binary masking of a set of musical tracks. Again, the MDCT proved to be overall the best

choice as it led to the more disjoint representation (as measured by the WDO). This result confirms previous investigation [109] and further motivates the use of MDCT for the processing of audio signals.

A counterintuitive result was obtained regarding the correlation between sparsity and disjointness of a transform: considering the best performing transform in terms of disjointness, no correlation was found between the sparsity of the coefficients in the transformed domain and the overlap of their supports. This contradicts the commonly made assumption that sparsity induces disjointness, at least in the case of the audio signals employed in the experiment shown.

Chapter 4

Dictionary learning of convolved signals

Sparse approximation techniques have been extensively used for de-noising purposes. For example, Chen et al. [17] show in their seminal paper on basis pursuit that it is possible to effectively remove Gaussian noise from sparse signals by solving the approximation problem (2.17).

The penalised minimisation defined in equation (2.17) results in a mixed objective that minimises the reconstruction error of the signal and the ℓ_1 norm of the approximation coefficients, inducing sparsity in the vector of approximation coefficients. The rationale behind their approach is that the non-zero coefficients of the approximation correspond to atoms that capture much more salient information about the signal than about the additive noise. This also implies that a small additive random perturbation of a signal that admits an exact sparse representation using a given dictionary should not lead to a large approximation error when approximating the perturbed signal with the same dictionary.

On the other hand, *convolution* can be thought as a different perturbation that is introduced for example every time a physical phenomenon is measured by means of transducers (e.g. recording an audio signal by means of a microphone). In this case, the recorded variable can be modelled as the convolution of the original signal of interest, the *source signal*, with the impulse response of the system in which the measurement takes place. Unlike additive noise, this process greatly affects the approximation residual obtained using sparse representation algorithms, as will be shown in Section 4.2.

Starting from this motivation, this chapter deals with learning a convolved dictionary which can be used to sparsely represent the observations, given the assumption that the underlying source signals belong to a class for which there exists an analytic or learned dictionary that leads to a sparse representation. Lou et al. [61] employed a similar idea involving sparse approximation on a convolved dictionary for de-blurring of natural images, with the substantial difference that in this paper the impulse response is known a priori, while the method proposed here aims at learning it from data. Hence the main contribution of the work presented in this chapter consists in interpreting the blind estimation of the unknown channel as a dictionary learning problem, and in devising an optimization algorithm to learn its paramenters.

4.1 Sparse approximation and convolution model

Suppose that a set of M source signals $\left\{\boldsymbol{s}_m \in \mathbb{R}^N\right\}_{m=1}^M$ admits an exact sparse representation in a dictionary $\boldsymbol{\Phi} \in \mathcal{D} \subseteq \mathbb{C}^{N \times K}$:

$$\boldsymbol{s}_m = \boldsymbol{\Phi}\boldsymbol{x}_m \quad ||\boldsymbol{x}_m||_0 \leq S \; \forall m = 1, \ldots, M. \tag{4.1}$$

This means that each source signal has an exact sparse representation using the dictionary $\boldsymbol{\Phi}$ with at most S active atoms.

Suppose that we do not directly observe the variables $\boldsymbol{s}_m$ but rather a set of convolved observations $\left\{\boldsymbol{y}_m \in \mathbb{R}^{(N+L-1)}\right\}_{m=1}^M$:

$$\boldsymbol{y}_m[n] = \sum_{l=0}^{L-1} s_m[l]h[n-l] \tag{4.2}$$

that are the result of a single input single output (SISO) causal convolutive system characterised by the impulse response $\boldsymbol{h}$ of length L.

Let

$$\breve{\boldsymbol{h}} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{h} \\ \\ \boldsymbol{0} \end{bmatrix} \tag{4.3}$$

be a vector in $\mathbb{R}^{N+L-1}$ resulting from zero-padding the impulse response with $N-1$ zeros. $\mathcal{T}(\boldsymbol{v})$ is a Toeplitz operator that takes a vector $\boldsymbol{v} \in \mathbb{R}^I$ as an input and returns a matrix $\boldsymbol{V} = \mathcal{T}(\boldsymbol{v}) \in \mathbb{R}^{I \times I}$ whose columns contain circularly shifted versions of the

vector, so that $V_{i,j} = v_{[(i-j+I) \bmod I]+1}$. The Topeplitz convolutive matrix $\boldsymbol{H} \stackrel{\text{def}}{=} \mathcal{T}\left(\breve{\boldsymbol{h}}\right) \in \mathbb{R}^{(N+L-1)\times(N+L-1)}$ contains shifted versions of the zero-padded impulse response in each column:

$$\boldsymbol{H} \stackrel{\text{def}}{=} \begin{bmatrix} h[0] & 0 & 0 & 0 & 0 \\ h[1] & h[0] & \ddots & \vdots & \vdots \\ \vdots & h[1] & \ddots & 0 & \vdots \\ h[L-1] & \vdots & \ddots & h[0] & 0 \\ 0 & h[L-1] & \ddots & h[1] & h[0] \\ \vdots & 0 & \ddots & \vdots & h[1] \\ \vdots & \vdots & \ddots & h[L-1] & \vdots \\ 0 & 0 & 0 & 0 & h[L-1] \end{bmatrix}. \tag{4.4}$$

A new dictionary $\boldsymbol{\Psi}$ containing convolved atoms can be written as:

$$\boldsymbol{\Psi} = \boldsymbol{H}\breve{\boldsymbol{\Phi}} \tag{4.5}$$

where the zero-padded dictionary $\breve{\boldsymbol{\Phi}} \in \mathbb{R}^{(N+L-1)\times K}$ can be expressed as:

$$\breve{\boldsymbol{\Phi}} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{0} \end{bmatrix}. \tag{4.6}$$

The observed signals resulting from the convolution described by equation (4.2) can be stacked into the columns of a matrix $\boldsymbol{Y} \in \mathbb{R}^{(N+L-1)\times M}$, and the model can be written in a compact matrix form as:

$$\boldsymbol{Y} = \boldsymbol{H}\breve{\boldsymbol{\Phi}}\boldsymbol{X} = \boldsymbol{\Psi}\boldsymbol{X}. \tag{4.7}$$

This means that the observed signals cannot be sparsely represented using the original dictionary $\boldsymbol{\Phi}$, but a new dictionary $\boldsymbol{\Psi}$ whose atoms $\boldsymbol{\psi}_k = \boldsymbol{h} * \boldsymbol{\phi}_k$ are obtained by the convolution between the original atoms and the impulse response of the system.

Whenever the impulse response $\boldsymbol{h}$ of the measurement system is unknown, one could still use the atoms of the dictionary $\breve{\boldsymbol{\Phi}}$ to attempt a sparse approximation of the observed signals. However this results in a large approximation error, as shown in Section 4.2.

Variable	Value	Description
Ensemble	RSE	Real Fourier dictionary ensemble
M	500	Number of observed signals
K	200	Number of atoms of the dictionary
$\|\|\boldsymbol{x}\|\|_0 / K$	0.05	Normalised diversity of source signals
S	15	Sparsity constraint for OMP-S
ϵ	10^{-2}	Error constraint for OMP-E

Table 4.1: Parameters of the experiment studying the effect of convolution on sparse approximation. The dimensions of the problem and the values of parameters such as sparsity constraints and error constraints are within ranges typically used in signal processing applications that employ a frame-by-frame processing of audio and image data.

4.2 Effect of convolution on sparse approximation

The goal of this section is to show the poor conditioning of sparse approximation in the presence of convolution. That is, how much the approximation error grows relative to the Euclidean distance between anechoic source signals and convoluted observations, when these are approximated using the dictionary $\breve{\boldsymbol{\Phi}}$ rather than $\boldsymbol{\Psi}$.

To this aim, a matrix of observed signals $\boldsymbol{Y}$ was synthesised by generating sparse linear combinations of the atoms contained in a dictionary $\boldsymbol{\Psi}$, as in the model (4.7). The parameters used for the simulation are summarised in Table 4.1. In particular, we firstly defined $M = 500$ source signals of dimension $N = 100$ as sparse linear combinations of the atoms contained in a two times over-complete real Fourier dictionary, that is one of the standard matrix ensembles implemented in the SPARSELAB toolbox[1]. The normalized diversity of the source signals, defined as the ratio between the number of nonzero coefficients of the representations and the number of atoms in the dictionary, was set to $||\boldsymbol{x}||_0 / K = 0.05$. We produced the observations by convolving the sources with a sparse non-negative impulse response $\boldsymbol{h}$ of length $L = 50$.

Both the sparsity constrained orthogonal matching pursuit (OMP-S) and the error constrained orthogonal matching pursuit (OMP-E) were tested as sparse approximation algorithms. They are two alternative formulations corresponding to the optimizations introduced in equations (2.12) and (2.11), and are based on the algorithm introduced in [82], but differ in their stopping criteria. The former aims at solving the following

[1] http://sparselab.stanford.edu

optimization:

$$\begin{aligned} &\underset{\boldsymbol{X}\in\mathbb{R}^{K\times M}}{\arg\min} \ ||\boldsymbol{Y}-\boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} \\ &\text{such that } \ ||\boldsymbol{x}_m||_0 \leq S \quad \forall m=1,\ldots,M \end{aligned} \tag{4.8}$$

while the latter attempts to solve the problem:

$$\begin{aligned} &\underset{\boldsymbol{X}\in\mathbb{R}^{K\times M}}{\arg\min} \ ||\boldsymbol{X}||_{0,0} \\ &\text{such that } \ ||\boldsymbol{Y}-\boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} \leq \epsilon. \end{aligned} \tag{4.9}$$

Hence, testing the sparsity constrained formulation involves assessing the residual norm of the sparse approximation for a given number of active atoms S, while evaluating the error constrained algorithm means measuring the number of active atoms employed to approximate the signals to a given level of accuracy.

The experiments were run multiple times varying the number of non-zero elements of $\boldsymbol{h}$ from 1 to L, starting from the identity operator of the convolution operation $\boldsymbol{\delta}_0$ and linearly increasing the number of non-zero coefficients of the impulse response. This causes an increasing average distance between the source signals and the observations measured by:

$$\bar{d}\left(\boldsymbol{S},\boldsymbol{Y}\right)=\frac{1}{M}\left|\left|\boldsymbol{Y}-\breve{\boldsymbol{S}}\right|\right|_{\mathrm{F}}=\frac{1}{M}\sum_{m=1}^{M}\left|\left|\boldsymbol{y}_m-\breve{\boldsymbol{\Phi}}\boldsymbol{x}_m\right|\right|_2. \tag{4.10}$$

Here the columns of $\breve{\boldsymbol{S}}$ and $\boldsymbol{Y}$ contain the sources and convolved observations respectively. The sizes of the variable is within the range of values commonly used for a frame-by-frame processing of audio or images.

Figures 4.1 and 4.2 depict the results of the experiment for the two versions of OMP. Let us first analyse the results for OMP-S. We ran the sparse approximation using the dictionary $\breve{\boldsymbol{\Phi}}$ on the convolved variables $\boldsymbol{Y}$ setting the number of active atoms to $S=15$. The sparsity constraint S was chosen to be 50% larger than the true number of active atoms used to synthesise the anechoic source signals to be resilient to modelling errors. In a more realistic situation where the signals to be analysed are not synthesised from a known dictionary, the fact that the residual norm of OMP is monotonically decreasing as a function of the number of atoms ensures a better approximation quality when choosing

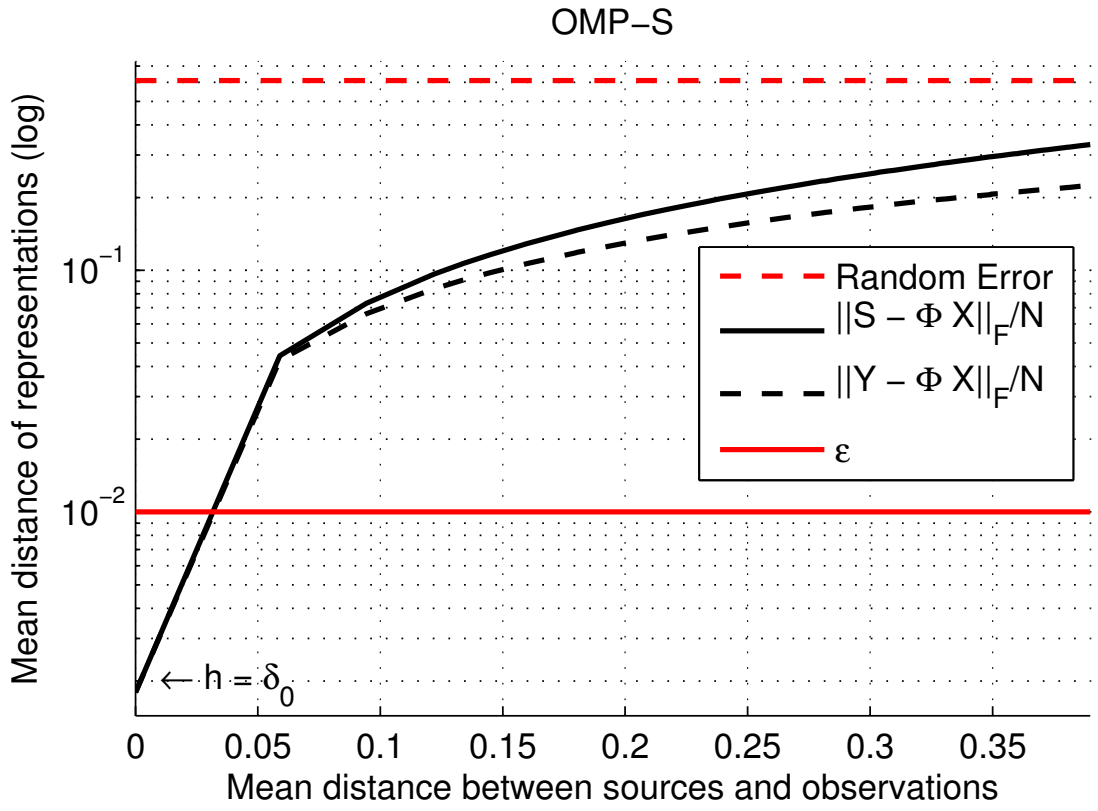


Figure 4.1: OMP-S results on convolved signals (averaged values over 100 trials of the experiment). The two black curves represent the average distance between the reconstructed signals and the sources or the observed variables respectively. For comparison purpose, the red dashed line represents the average distance between the observed variables and random signals, while the red solid line is the error tolerance defined for OMP-E (whose results are detailed in Figure 4.2).

a larger number of active atoms.

As can be seen in the left side of the plot 4.1, when the impulse response is simply $\boldsymbol{h} = \boldsymbol{\delta}_0$ the source and observed variables are the same and OMP-S is able to represent them with negligible error. However, as the convolution causes the observed $\boldsymbol{Y}$ to differ from $\breve{\boldsymbol{S}}$, the error in the representation quickly increases, to the point where OMP-S becomes almost comparable with a random approximation (that is, to the error resulting from approximating the observed signals with random vectors).

The behaviour of OMP-E is similar: fixing a tolerance $\epsilon = 10^{-2}$, the algorithm is able to represent the observed signals using the right number of active elements in the trivial case $\boldsymbol{h} = \boldsymbol{\delta}_0$. However, as soon as the average distance $\bar{d}(\boldsymbol{S}, \boldsymbol{Y})$ increases, the number of active elements needed rapidly rises over 80% of a completely dense representation.

The steep error curve displayed in the two figures confirms that convolution is a

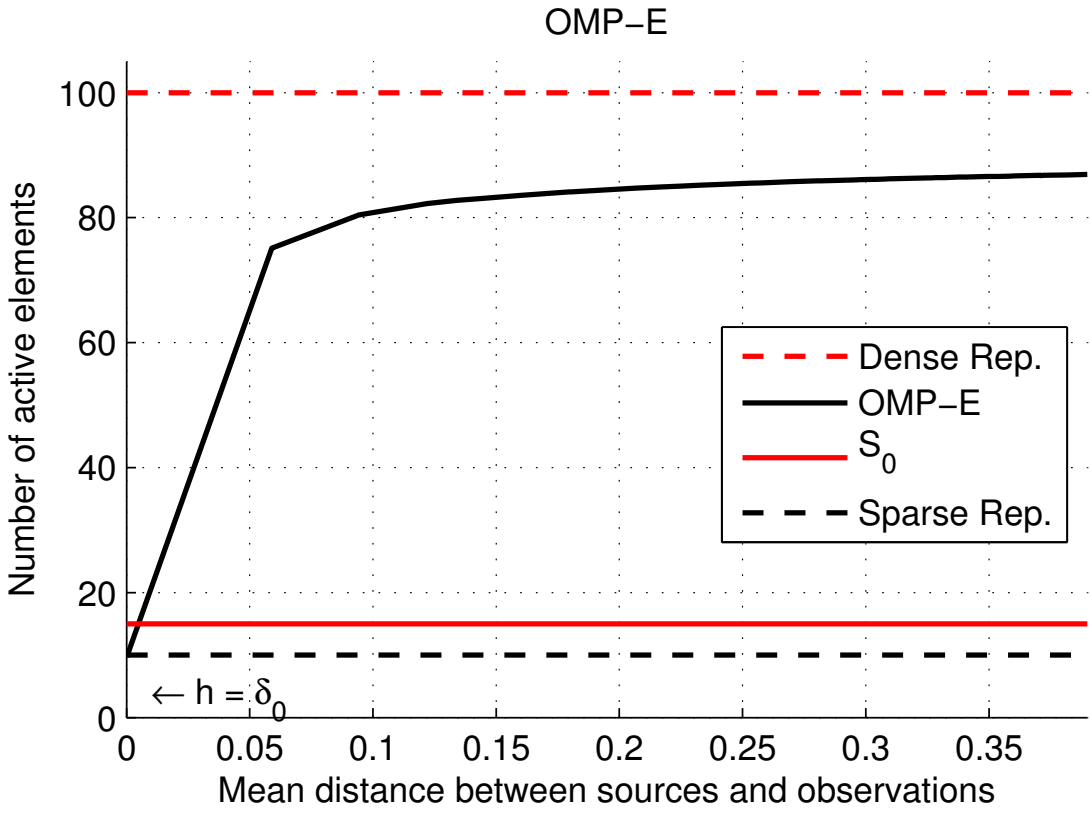


Figure 4.2: OMP-E results on convolved signals (averaged values over 100 trials of the experiment). The black and red dashed lines represent the true number of active atoms used to generate the test signals and the number of active atoms of a completely dense representation respectively. S is the constraint parameter used for OMP-S.

process that has a strong impact on the quality of sparse approximation, and motivates the design of an algorithm that can be used to learn the impulse response $\boldsymbol{h}$ and, therefore, approximate the observed variables using the atoms in the convolved dictionary $\boldsymbol{\Psi}$.

4.3 Dictionary learning of convolved signals

The results shown in Section 4.2 are not surprising if we consider the model presented in Section 4.1: the observed variables are no longer sparsely represented using the atoms in the dictionary $\breve{\boldsymbol{\Phi}}$, but can be represented using the the dictionary $\boldsymbol{\Psi} = \boldsymbol{H}\breve{\boldsymbol{\Phi}}$ whose atoms $\boldsymbol{\psi}_k = \boldsymbol{h} * \boldsymbol{\phi}_k$ are the convolution of the original atoms and the impulse response $\boldsymbol{h}$. Since the impulse response of the system is unknown, the objective of this section is to develop a novel dictionary learning algorithm in order to learn it from the observed signals, leveraging the assumption that the resulting convolved dictionary should lead to a sparse approximation with small residual error.

Besides the approximation objective described above, the research presented in this section can be contextualised and linked to source separation tasks. Jaureguiberry et al. [53] presented a method for nonnegative matrix factorisation in which equalization filters are learned in a supervised setting from a set of known source signals and used for separating filtered observations. Their contribution is linked to the proposed algorithm in that it includes a convolutive filter (the equalisation process) into the analysis of audio sources, but uses NMF instead of the general dictionary learning considered in this thesis. In addition, Benichoux et al. [7] employed a study of the statistical properties of reverberation filters to realise a constrained estimation of room impulse responses. Mr. Benichoux and I initiated a collaborative research project aimed at combining and extending our algorithms. We tackled a convolutive source separation task by constraining the time domain representation of the reverberation filters and the sparse approximation of the source signals. Although preliminary results have been encouraging, this research project is not yet mature enough to constitute a significant contribution and to be included in this thesis.

4.3.1 Dictionary learning in the Fourier domain

The general strategy employed in the optimization of the dictionary learning of convolved signals follows an alternate optimization of sparse approximation coefficients and impulse response that is common to dictionary learning algorithms.

Before describing the optimization process used to learn the impulse response, a frequency domain formulation of the convolution model is derived in this section. This is useful to express equation (4.7) in terms of the impulse response $\boldsymbol{h}$ rather than the convolution matrix $\boldsymbol{H}$ and define a relative impulse response optimization step that will be described in Section 4.3.2.

Let us first define a cost function $\mathcal{C}\left(\boldsymbol{h}, \boldsymbol{X}\right)$ that represents the total error of the approximation as a function of the impulse response and of the sparse approximation coefficients:

$$\mathcal{C}\left(\boldsymbol{h}, \boldsymbol{X}\right) = \frac{1}{2}\left|\left|\boldsymbol{Y} - \boldsymbol{H}\breve{\boldsymbol{\Phi}}\boldsymbol{X}\right|\right|_{\mathrm{F}}^{2} \tag{4.11}$$

We can express (4.11) in the frequency domain by multiplying both the observations matrix $\boldsymbol{Y}$ and their sparse approximation $\boldsymbol{H}\breve{\boldsymbol{\Phi}}\boldsymbol{X}$ by the DFT matrix $\boldsymbol{F} \in \mathbb{C}^{(N+L-1)\times(N+L-1)}$

whose columns contain the Fourier basis:

$$\boldsymbol{f}_l = \frac{1}{\sqrt{N+L-1}} \exp\left[\frac{2\pi i}{N+L-1} ln\right]. \tag{4.12}$$

Here the normalized version of the DFT, such that $\boldsymbol{F}^{\mathcal{H}}\boldsymbol{F} = \boldsymbol{I}$ is used. Since it is defined as an orthonormal transform, the Fourier operator does not modify the magnitude of the residual expressed in equation (4.11) and leads to an equivalent cost function in the Fourier domain:

$$\mathcal{C}\left(\hat{\boldsymbol{h}}, \boldsymbol{X}\right) = \frac{1}{2}\left|\left|\boldsymbol{F}^{\mathcal{H}}\boldsymbol{Y} - \boldsymbol{F}^{\mathcal{H}}\boldsymbol{H}\breve{\boldsymbol{\Phi}}\boldsymbol{X}\right|\right|_{\mathrm{F}}^2 \tag{4.13}$$

where the *hat* symbol identifies variables in the Fourier domain and will be applied from now on column-wise if applied to matrix arguments.

Note that, given a vector $\boldsymbol{v}$, every element of its Fourier transform $\hat{v}_l = \langle \boldsymbol{f}_l, \boldsymbol{v} \rangle$ results from the inner product with the Fourier bases. When using a matrix notation, the Hermitian operator $(\cdot)^{\mathcal{H}}$ must be used in place of the matrix transposition to calculate the inner products because $\boldsymbol{F}$ contains complex variables.

There is a relation that links convolution of two vectors in the time domain to the element-wise multiplication of the components of their respective Fourier transforms [79]. The element-wise multiplication between the components of a vector $\boldsymbol{v}$ and the components of the vector $\boldsymbol{w}$ can be expressed by the product $\mathcal{D}(\boldsymbol{v})\boldsymbol{w}$, where the operator $\mathcal{D}(\cdot)$ returns a diagonal matrix whose diagonal entries are the elements of its vector argument.

Let $\boldsymbol{\psi}_k = \boldsymbol{h} * \boldsymbol{\phi}_k$ be a convolved atom. This can be expressed in terms of the Fourier transforms $\hat{\boldsymbol{\phi}}_k$ and $\hat{\boldsymbol{h}}$ as follows:

$$\hat{\boldsymbol{\psi}}_k = \mathcal{D}\left(\hat{\tilde{\boldsymbol{h}}}\right)\hat{\tilde{\boldsymbol{\phi}}}_k \tag{4.14}$$

where $\tilde{\boldsymbol{h}}$ and $\tilde{\boldsymbol{\phi}_k}$ are obtained periodically extending the vectors $\boldsymbol{h}$ and $\boldsymbol{\phi}_k$. Equation (4.14) is named *circular* convolution from the fact that it is equivalent to the convolution between vectors that have been circularly or periodically extended. This is equivalent to the *linear* convolution defined in equation (4.7) if the vectors have been zero-padded as described in Section 4.1. In this case convolution in the time domain is equivalent to element-wise multiplication in the frequency domain, and the cost function (4.13) can be expressed as:

$$\hat{\mathcal{C}}\left(\hat{\boldsymbol{h}}, \boldsymbol{X}\right) = \frac{1}{2}\left|\left|\hat{\boldsymbol{Y}} - \mathcal{D}\left(\hat{\breve{\boldsymbol{h}}}\right)\hat{\breve{\boldsymbol{\Phi}}}\boldsymbol{X}\right|\right|_{\mathrm{F}}^2. \tag{4.15}$$

For clarity of notation, the zero-pad symbol $\breve{(\cdot)}$ will be from now on omitted in the equations, assuming that variables have been zero-padded as a pre-processing step.

One last simplification of the cost function (4.15) can be derived by considering a property of the Fourier transform of real signals. Given a real vector $\boldsymbol{v} \in \mathbb{R}^N$, the Hermitian symmetry of its Fourier transform $\hat{\boldsymbol{v}}$ implies that $\hat{v}_{N-j+1} = \hat{v}_j^* \; \forall j = 1, \dots, \lfloor (N+L)/2 \rfloor + 1$. This constraint can be taken into account by only estimating the first $J = \lceil (N+L)/2 \rceil$ Fourier coefficients of the vector $\hat{\boldsymbol{h}}$, and setting the remainders as complex conjugate.

The cost function in the frequency domain (4.15) becomes:

$$\mathcal{C}\left(\hat{\boldsymbol{h}}_{1:J}, \boldsymbol{X}\right) = \frac{1}{2} \left|\left| \hat{\boldsymbol{Y}}^{1:J} - \mathcal{D}\left(\hat{\boldsymbol{h}}\right)^{1:J} \hat{\boldsymbol{\Phi}}^{1:J} \boldsymbol{X} \right|\right|_{\mathrm{F}}^2 \tag{4.16}$$

where the superscripts $(\cdot)^{1:J}$ indicate that only the first J rows of the various variables are taken into account.

4.3.2 Block coordinate descent optimization

This section describes an optimization algorithm aimed at solving the problem:

$$\begin{aligned} (\boldsymbol{h}^\star, \boldsymbol{X}^\star) = \underset{\boldsymbol{h} \in \mathbb{R}^L, \boldsymbol{X} \in \mathbb{R}^{K \times M}}{\arg\min} \; & ||\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} \\ \text{such that } & ||\boldsymbol{x}_m||_0 \leq S \quad \forall m. \end{aligned} \tag{4.17}$$

This optimization follows from the convolutive model introduced in Section 4.1 and is akin to the dictionary learning problem as defined in equation (2.23), with the notable difference that here a dictionary $\boldsymbol{\Phi}$ is kept fixed and the impulse response vector $\boldsymbol{h}$ is optimized instead.

The joint minimisation of the cost function (4.17) over the variables $\boldsymbol{h}$ and $\boldsymbol{X}$ is an underdetermined problem in that the number of variables is $KM + L$ and the number of observations is NM, with $K \geq N > L$. A so-called *block coordinate descent* optimization can be employed following the same strategy used by dictionary learning algorithms where the two variables are updated one at a time and the optimization of one is based on the previous value of the other.

This is an iterative optimization strategy that starts from an initial guess of the impulse response $\boldsymbol{h}^{(0)}$ (that can be initialised, for example, as a random vector or as the

convolution identity $\boldsymbol{\delta}_0$), and proceeds for a fixed number of iterations $i = 1, \ldots, I$ by solving the following sub-problems at each iteration i:

Source signals optimization : given a fixed $\boldsymbol{h}^{(i)}$, a convolved dictionary $\boldsymbol{\Psi}^{(i)} = \mathcal{T}\left(\boldsymbol{h}^{(i)}\right)\boldsymbol{\Phi}$ is computed and used to obtain a sparse approximation $\boldsymbol{X}^{(i)}$ of the observed signals $\boldsymbol{Y}$.

Impulse response optimization : given a matrix of sparse approximation coefficients $\boldsymbol{X}^{(i)}$, a new impulse response $\boldsymbol{h}^{(i+1)}$ is computed in order to minimise (4.11) or its equivalent in the frequency domain (4.16).

Source Signals optimization

Given a fixed impulse response $\boldsymbol{h}^{(i)}$, the cost function (4.11) (or, its equivalent in the frequency domain (4.16)) represents a classic sparse approximation problem where we seek a matrix $\boldsymbol{X}$ that minimises the residual norm of the representation over the dictionary $\boldsymbol{\Psi} = \boldsymbol{H}^{(i)}\breve{\boldsymbol{\Phi}}$, given a constraint on the sparsity of the solution vectors.

$$\begin{aligned} &\underset{\boldsymbol{X}\in\mathbb{R}^{K\times M}}{\arg\min} \; ||\boldsymbol{Y} - \boldsymbol{\Psi}\boldsymbol{X}||_{\mathrm{F}}^2 \\ &\text{such that } \; ||\boldsymbol{x}_m||_0 \leq S \quad \forall\, m = 1, \ldots, M. \end{aligned} \tag{4.18}$$

This can be tackled using any suitable sparse approximation algorithm such as OMP-S. Alternatively, the sparsity assumption can be relaxed to an ℓ_1 constraint, which leads to a convex problem that can be solved with various methods, such as basis pursuit [17] or homotopy [80].

Impulse Response optimization

The impulse response optimization step can be solved starting from (4.16) and defining an equivalent quadratic program. Given that the Frobenious norm of a matrix $\boldsymbol{M}$ can be expressed as the trace of the Gram matrix $\boldsymbol{G} \overset{\text{def}}{=} \boldsymbol{M}^{\mathcal{H}}\boldsymbol{M}$, $||\boldsymbol{M}||_{\mathrm{F}} = \mathrm{Tr}\,(\boldsymbol{G})$, the cost function defined in equation (4.16) can be written as:

$$\hat{\mathcal{C}}(\hat{\boldsymbol{h}}_{1:J}) = \frac{1}{2}\,\mathrm{Tr}\left[\left(\boldsymbol{Y}^{1:J} - \mathcal{D}\left(\hat{\boldsymbol{h}}\right)^{1:J}\hat{\boldsymbol{\Phi}}^{1:J}\boldsymbol{X}\right)^{\mathcal{H}}\left(\boldsymbol{Y}^{1:J} - \mathcal{D}\left(\hat{\boldsymbol{h}}\right)^{1:J}\hat{\boldsymbol{\Phi}}^{1:J}\boldsymbol{X}\right)\right]. \tag{4.19}$$

To avoid double superscripts in the notation, the row indexes $(\cdot)^{1:J}$ will be omitted from now on, implicitly assuming that the impulse response in the Fourier domain is optimized

only considering its first J components. Expanding the above equation while omitting the terms that do not depend on $\hat{\boldsymbol{h}}$ leads to the definition of an impulse response optimization problem which can be written as:

$$\hat{\boldsymbol{h}}^{\star}_{1:J} = \underset{\hat{\boldsymbol{h}}_{1:J}}{\arg\min}\ \frac{1}{2}\left[\mathrm{Tr}\left(\hat{\boldsymbol{S}}^{\mathcal{H}}\mathcal{D}\left(\left|\hat{\boldsymbol{h}}\right|^{2}\right)\hat{\boldsymbol{S}}\right) - \mathrm{Tr}\left(\hat{\boldsymbol{Y}}^{\mathcal{H}}\mathcal{D}\left(\hat{\boldsymbol{h}}\right)\hat{\boldsymbol{S}}\right) - \mathrm{Tr}\left(\hat{\boldsymbol{S}}^{\mathcal{H}}\mathcal{D}\left(\hat{\boldsymbol{h}}^{*}\right)\hat{\boldsymbol{Y}}\right)\right] \tag{4.20}$$

where the operator $(\cdot)^*$ indicates complex conjugate and the matrix $\hat{\boldsymbol{S}}$ contains the Fourier transform of the estimated source signals $\boldsymbol{S} = \boldsymbol{\Phi}\boldsymbol{X}$.

We can simplify this expression considering a property of diagonal matrices. Let $\boldsymbol{D}$ be a diagonal matrix, and $\boldsymbol{A}$ and $\boldsymbol{B}$ two arbitrary matrices.

$$\begin{aligned}
\mathrm{Tr}\left(\boldsymbol{A}^{\mathcal{H}}\boldsymbol{D}\boldsymbol{B}\right) &= \sum_{i}\left[\boldsymbol{A}^{\mathcal{H}}\boldsymbol{D}\boldsymbol{B}\right]_{i,i} \\
&= \sum_{i}\sum_{j}\left[\boldsymbol{A}^{\mathcal{H}}\right]_{i,j}\left[\boldsymbol{D}\boldsymbol{B}\right]_{j,i} \\
&= \sum_{i}\sum_{j} a^{*}_{j,i}\sum_{k} d_{j,k}b_{k,i} \\
&= \sum_{j} d_{j,j}\sum_{i} b_{j,i}a^{*}_{j,i} \\
&= \mathrm{d}\left(\boldsymbol{D}\right)^{T}\mathrm{d}\left(\boldsymbol{B}\boldsymbol{A}^{\mathcal{H}}\right)
\end{aligned}$$

where the operator $\mathrm{d}\left(\cdot\right)$ returns a vector whose elements are the diagonal entries of its matrix argument.

Therefore, the unconstrained minimisation of the frequency response $\hat{\boldsymbol{h}}$ becomes:

$$\hat{\boldsymbol{h}}^{\star} = \underset{\hat{\boldsymbol{h}}}{\arg\min}\ \frac{1}{2}\left[\hat{\boldsymbol{h}}^{\mathcal{H}}\mathcal{D}\left(\hat{\boldsymbol{h}}\right)\mathrm{d}\left(\hat{\boldsymbol{S}}\hat{\boldsymbol{S}}^{\mathcal{H}}\right) - \hat{\boldsymbol{h}}^{T}\mathrm{d}\left(\hat{\boldsymbol{S}}\hat{\boldsymbol{Y}}^{\mathcal{H}}\right) - \hat{\boldsymbol{h}}^{\mathcal{H}}\mathrm{d}\left(\hat{\boldsymbol{Y}}\hat{\boldsymbol{S}}^{\mathcal{H}}\right)\right]. \tag{4.21}$$

Let's introduce for clarity of notation the vectors

$$\boldsymbol{\alpha} \overset{\text{def}}{=} \mathrm{d}\left(\hat{\boldsymbol{S}}\hat{\boldsymbol{S}}^{\mathcal{H}}\right) \tag{4.22}$$

$$\boldsymbol{\beta} \overset{\text{def}}{=} \mathrm{d}\left(\hat{\boldsymbol{Y}}\hat{\boldsymbol{S}}^{\mathcal{H}}\right). \tag{4.23}$$

The cost function in equation (4.21) can be written as:

$$\frac{1}{2}\left[\hat{\boldsymbol{h}}^{\mathcal{H}}\mathcal{D}\left(\hat{\boldsymbol{h}}\right)\boldsymbol{\alpha}-\hat{\boldsymbol{h}}^{T}\boldsymbol{\beta}^{*}-\hat{\boldsymbol{h}}^{\mathcal{H}}\boldsymbol{\beta}\right]=\frac{1}{2}\left[\hat{\boldsymbol{h}}^{\mathcal{H}}\mathcal{D}\left(\boldsymbol{\alpha}\right)\hat{\boldsymbol{h}}-\hat{\boldsymbol{h}}^{\mathcal{H}}\boldsymbol{\beta}^{*}-\hat{\boldsymbol{h}}^{\mathcal{H}}\boldsymbol{\beta}\right] \tag{4.24}$$

where the equivalence comes from the commutative property of the element-wise multiplication $\mathcal{D}\left(\hat{\boldsymbol{h}}\right)\boldsymbol{\alpha}=\mathcal{D}\left(\boldsymbol{\alpha}\right)\hat{\boldsymbol{h}}$. Minimising this cost function with respect to the impulse response means solving a quadratic program in the complex variable $\hat{\boldsymbol{h}}$. This is equivalent to the least-square problem:

$$\hat{\boldsymbol{h}}^{\star}=\underset{\hat{\boldsymbol{h}}\in\mathbb{C}^{J}}{\arg\min}\ \left|\left|\mathcal{D}\left(\boldsymbol{\alpha}\right)^{\frac{1}{2}}\hat{\boldsymbol{h}}-\mathcal{D}\left(\boldsymbol{\alpha}\right)^{-\frac{1}{2}}\boldsymbol{\beta}\right|\right|_{2}^{2}. \tag{4.25}$$

Equation (4.25) can be expressed in the time domain by considering the Fourier transform $\hat{\boldsymbol{h}}=\boldsymbol{F}^{\mathcal{H}}\boldsymbol{h}$:

$$\boldsymbol{h}^{\star}=\underset{\boldsymbol{h}\in\mathbb{R}^{N+L-1}}{\arg\min}\ \frac{1}{2}\left|\left|\mathcal{D}\left(\boldsymbol{\alpha}^{1/2}\right)\boldsymbol{F}^{\mathcal{H}}\boldsymbol{h}-\mathcal{D}\left(\boldsymbol{\alpha}^{-1/2}\right)\boldsymbol{\beta}\right|\right|_{2}^{2}. \tag{4.26}$$

The length of the impulse response L can be taken into account in the optimization by constraining the last $N-1$ components of the optimization variable in (4.26) to be zero.

For clarity of notation we define the variables

$$\boldsymbol{\Gamma}\overset{\text{def}}{=}\ \mathcal{D}\left(\boldsymbol{\alpha}^{1/2}\right)\boldsymbol{F}^{\mathcal{H}} \tag{4.27}$$

$$\boldsymbol{\xi}\overset{\text{def}}{=}\ \mathcal{D}\left(\boldsymbol{\alpha}^{-1/2}\right)\boldsymbol{\beta}. \tag{4.28}$$

The optimization (4.26) becomes:

$$\boldsymbol{h}^{\star}=\underset{\boldsymbol{h}\in\mathbb{R}^{L}}{\arg\min}\ \frac{1}{2}\left|\left|\boldsymbol{\Gamma}_{1:L}\boldsymbol{h}-\boldsymbol{\xi}\right|\right|_{2}^{2} \tag{4.29}$$

which is over-determined because $\boldsymbol{\Gamma}_{1:L}\in\mathbb{C}^{(N+L-1)\times L}$ and can be solved by computing the pseudo-inverse:

$$\boldsymbol{h}^{\star}=\boldsymbol{\Gamma}_{1:L}^{\dagger}\boldsymbol{\xi}. \tag{4.30}$$

Constrained Optimization

In the experiments presented in Section 4.2 we employed a sparse and non-negative impulse response that was convolved with the synthetic source signals. The choice of sparsity

and non-negativity constraints is particularly suited for audio signals in that the early reflections coming from the surfaces of the ambient in which the signals are recorded can be modelled as a sparse non-negative impulse response [6].

These constraints can be introduced into the impulse response optimization (4.29). Firstly, we can assume that the vector $\boldsymbol{h}$ is non negative, turning the minimisation (4.29) into a non negative least squares problem which can be solved via quadratic programming. We can then constraint the ℓ_1 norm of the solution to be smaller than a fixed value Q_1, inducing sparsity on the impulse response coefficients.

This can be done in a very simple way considering that the ℓ_1 norm of a nonnegative vector is the sum of its entries. The optimization problem becomes:

$$\begin{aligned} \boldsymbol{h}^\star = \underset{\boldsymbol{h}\in\mathbb{R}^L}{\arg\min}\ & \frac{1}{2}\left|\left|\boldsymbol{\Gamma}_{1:L}\boldsymbol{h}-\boldsymbol{\xi}\right|\right|_2^2 \\ \text{such that } & \boldsymbol{h}\geq\boldsymbol{0} \\ & \boldsymbol{1}^{\mathcal{H}}\boldsymbol{h}\leq Q_1. \end{aligned} \tag{4.31}$$

The labels D$\boldsymbol{h}$-BCD and S$\boldsymbol{h}$-BCD are the acronyms of *dense* $\boldsymbol{h}$ *block coordinate descent* and *sparse* $\boldsymbol{h}$ *block coordinate descent* and will be employed from now on to identify the two optimization problems (4.29) and (4.31).

Ambiguities

The cost function of the problem (4.17) contains an inherent ambiguity because it is possible to multiply the matrices $\boldsymbol{H}$ and $\boldsymbol{X}$ by an arbitrary scalar and its inverse resulting in the same function value, that is:

$$\left|\left|\boldsymbol{Y}-\boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{X}\right|\right|_{\mathrm{F}} = \left|\left|\boldsymbol{Y}-(C\boldsymbol{H})\,\boldsymbol{\Phi}\left(\frac{1}{C}\boldsymbol{X}\right)\right|\right|_{\mathrm{F}} \ \forall\, C\in\mathbb{R}\setminus\{0\}\,. \tag{4.32}$$

To prevent the optimization from introducing a large discrepancy between the norms of the impulse response matrix $\boldsymbol{H}$ and of the sparse approximation coefficients matrix $\boldsymbol{X}$, a normalization step is added in a way that is analogous to the normalization introduced after the dictionary update step of dictionary learning algorithms that is explained in Section 2.6.

In a traditional dictionary learning setting, the normalization step relies on an am-

biguity of the objective function where the dictionary and the matrix of approximation coefficients can be multiplied by a diagonal matrix and its inverse respectively maintaining the same value. By defining a diagonal normalization matrix $\boldsymbol{\Xi}$ whose diagonal elements contain the inverse of the ℓ_2 norm of the atoms of the dictionary returned by the dictionary update step, the atoms are kept normalized.

In the learning setting described here such ambiguity does not hold, since given an arbitrary diagonal matrix $\boldsymbol{D} \neq C\boldsymbol{I}$ which is not trivially a scaled identity matrix:

$$||\boldsymbol{Y} - \boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} \neq \left|\left|\boldsymbol{Y} - (\boldsymbol{D}\boldsymbol{H})\,\boldsymbol{\Phi}\left(\boldsymbol{D}^{-1}\boldsymbol{X}\right)\right|\right|_{\mathrm{F}}. \tag{4.33}$$

Instead of keeping the atoms of the dictionary normalized, we choose not to modify the initial Frobenious norm of the dictionary:

$$||\boldsymbol{H}\boldsymbol{\Phi}||_{\mathrm{F}}^2 = ||\boldsymbol{\Phi}||_{\mathrm{F}}^2 = K \tag{4.34}$$

where we assumed a normalized dictionary $\boldsymbol{\Phi}$ and K is the number of atoms.

Therefore, once the impulse response has been optimized solving the problem (4.29) or (4.31), we redistribute the energy between the matrix $\boldsymbol{H}$ and the coefficients $\boldsymbol{X}$ so that the equality (4.34) is satisfied.

$$\begin{aligned} R &= \frac{\sqrt{K}}{||\boldsymbol{H}\boldsymbol{\Phi}||_{\mathrm{F}}} \\ \boldsymbol{h} &\leftarrow R\boldsymbol{h} \\ \boldsymbol{X} &\leftarrow \frac{1}{R}\boldsymbol{X} \end{aligned}$$

This is analogous to the normalization step introduced after the dictionary update of dictionary learning algorithms that is described in Section 2.6. Note that other more general ambiguities such as multiplication with a diagonal or permutation matrix do not occur in this case.

Algorithm 8 summarises the optimization of the dictionary learning of convolved signals model.

Algorithm 8: Dictionary learning of convolved signals

Input: $\boldsymbol{Y}, \boldsymbol{\Phi}, \boldsymbol{h}^{(0)}, I$
Output: $\boldsymbol{h}^{\star}, \boldsymbol{X}^{\star}$
// Initialisation
$i \leftarrow 1$;
while $i \leq I$ **do**
  // Source signals optimization
  $\boldsymbol{H} \leftarrow \mathcal{T}\left(\breve{\boldsymbol{h}}\right)$;
  $\boldsymbol{\Psi} \leftarrow \boldsymbol{H}\breve{\boldsymbol{\Phi}}$;
  $\boldsymbol{X} \leftarrow \underset{\boldsymbol{X}\in\mathbb{R}^{K\times M}}{\arg\min} \; ||\boldsymbol{Y} - \boldsymbol{\Psi}\boldsymbol{X}||_{\mathrm{F}}^2 \quad \text{s.t. } ||\boldsymbol{x}_m||_0 \leq S \quad \forall\, m$;
  $\boldsymbol{S} \leftarrow \breve{\boldsymbol{\Phi}}\boldsymbol{X}$;
  // Impulse response optimization
  $\boldsymbol{\alpha} \leftarrow \mathrm{d}\left(\hat{\boldsymbol{S}}\hat{\boldsymbol{S}}^{\mathcal{H}}\right)$;
  $\boldsymbol{\beta} \leftarrow \mathrm{d}\left(\hat{\boldsymbol{Y}}\hat{\boldsymbol{S}}^{\mathcal{H}}\right)$;
  $\boldsymbol{\Gamma} \leftarrow \mathcal{D}\left(\boldsymbol{\alpha}^{1/2}\right)\boldsymbol{F}^{\mathcal{H}}$;
  $\boldsymbol{\xi} \leftarrow \mathcal{D}\left(\boldsymbol{\alpha}^{1/2}\right)\boldsymbol{\beta}$;
  // Choose constrained or unconstrained optimization of the impulse response
  switch $\boldsymbol{h}$ *optimization type* **do**
    case D$\boldsymbol{h}$-BCD
      $\boldsymbol{h} \leftarrow$ solution of optimization (4.29);
    endsw
    case S$\boldsymbol{h}$-BCD
      $\boldsymbol{h} \leftarrow$ solution of optimization (4.31);
    endsw
  endsw
  // Impulse response and coefficients normalization
  $R \leftarrow \frac{\sqrt{K}}{||\boldsymbol{H}\boldsymbol{\Phi}||_{\mathrm{F}}}$;
  $\boldsymbol{h} \leftarrow R\boldsymbol{h}$;
  $\boldsymbol{X} \leftarrow \frac{1}{R}\boldsymbol{X}$;
end

4.4 Numerical experiments

In this section we will describe several numerical tests performed on synthetic data in order to evaluate the proposed block coordinate descent optimization and to compare it with the K-SVD dictionary learning algorithm [4]. The main goal is to assess whether the proposed model and algorithm are better suited to learn signals generated as sparse linear combinations of convolved signals compared to a standard dictionary learning algorithm. The source signals were generated according to the model described in Section 4.1 using the parameters defined in Section 4.2 and were convolved with a sparse non-negative

impulse response with normalized diversity $||\boldsymbol{h}||_0 / L = 0.05$.

4.4.1 Sparse vs dense impulse response estimation

To tackle the optimization problem (4.17) we used the sparsity-constrained version of OMP and chose the sparsity parameter S to allow for 50% more active elements than originally used to generate the signals, as previously done in the numerical experiments presented in Section 4.2. For the impulse response estimation step we may or may not introduce additional constraints, which leads to:

1. S$\boldsymbol{h}$-BCD : at each step of the algorithm, the impulse response $\boldsymbol{h}$ is updated solving the optimization problem (4.31) using a nonnegative version of the LASSO algorithm[2], which constrains the solution to be sparse and nonnegative (again, we set the sparsity constraint to allow for a 50% tolerance on the number of active elements). The LASSO algorithm was chosen because it solves an ℓ_1 constrained minimisation while at the same time using a fixed number of active atoms, and it is therefore a valid technique for the solution of the problem (4.31).

2. D$\boldsymbol{h}$-BCD : at each step of the algorithm, the impulse response $\boldsymbol{h}$ is updated by solving the optimization problem (4.29). As explained in Section 4.3.2, this is an overdetermined problem whose solution can be derived analytically and corresponds to the least-squares solution of the system.

The two methods above, along with the K-SVD dictionary learning algorithm[3] were initialised with the known dictionary $\boldsymbol{\Phi}$ and an initial impulse response $\boldsymbol{h}^{(0)}$ whose elements were generated randomly from a Gaussian distribution with zero mean and unit variance. The learning algorithms were run for 50 iterations.

Figure 4.3 shows the average distance $\bar{d}$ of the representations defined as in equation (4.10). The data are displayed on a logarithmic scale and averaged over 100 independent trials of the experiment that correspond to different random initialisations.

As can be seen, D$\boldsymbol{h}$-BCD is the only method which improves its value during all the iterations. On the other hand, the convergence of K-SVD and S$\boldsymbol{h}$-BCD is significantly slower, with the latter performing worse. In other words, the impulse response learned

[2]we used the `SolveLasso` function contained in the SPARSELAB toolbox.
[3]we used the `ksvd` function available as part of the SMALLBOX toolbox

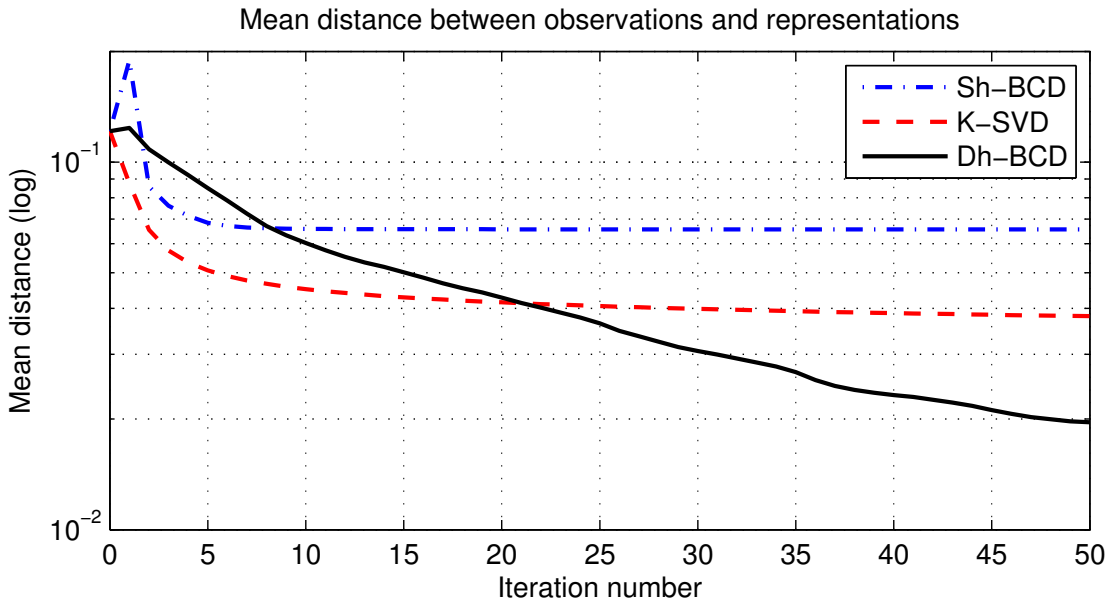


Figure 4.3: Average distance between observed convolved variables and sparse approximation model as a function of the iteration number of BCD and K-SVD (average over 100 trials of the experiment).

using D$\boldsymbol{h}$-BCD allows a sparse approximation of the convolved signals which is more effective in terms of the residual error if compared to the K-SVD algorithm or with the constrained S$\boldsymbol{h}$-BCD .

Figure 4.4 offers a more precise comparison between K-SVD and D$\boldsymbol{h}$-BCD by showing the boxplot of the average distance as a function of the iteration number. The plots display the distribution of the average distance at each iteration over the 100 independent trials that were run starting from random initialisations of $\boldsymbol{h}^{(0)}$ and convolved observations. The data are arranged and displayed according to their percentile, with the boxes comprising points that fall between the 25-th and the 75-th percentile, the central mark indicating the median that corresponds to the 50-th percentile and whiskers extending until points that fall within values not considered outliers (a data point is considered an outlier if its value falls outside intervals above and below the boxes of size 1.5 times the size of the boxes).

In the upper plot referring to D$\boldsymbol{h}$-BCD the median error drops from 10^{-1} at the first iteration to a value lower than 10^{-2} at iteration 50. Similarly, all the average distances not considered outliers drop to values below 10^{-2} by the 50-th iteration, with a few outliers remaining to values around 10^{-1}. On the other hand, K-SVD seems to be more robust to

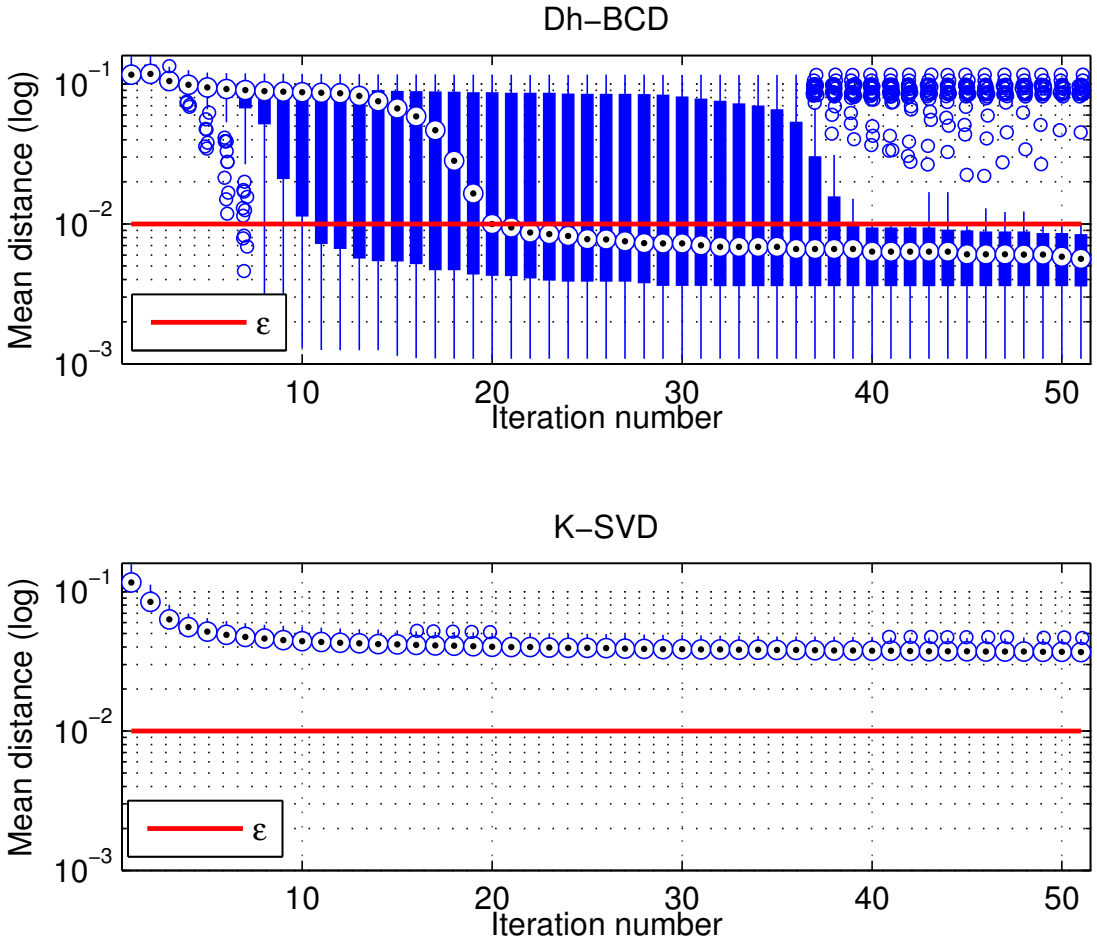


Figure 4.4: Boxplot comparison of the average distance obtained with D$\boldsymbol{h}$-BCD and K-SVD over 100 trials of the experiment. For each iteration, the central mark is the median, the edges of the boxes are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.

outliers than the proposed methods but consistently achieves a worst average error.

Similar results have been obtained by considering an impulse response $\boldsymbol{h}$ of length $L = 60$ samples generated according to the image method proposed by Allen and Berkley [6] and implemented by McGovern [73]. The image method is a technique for generating the impulse response describing the acoustic path between a source and a microphone situated in a room of known dimensions. It is based on the assumption that acoustic waves can be modelled as beams that are reflected by the surfaces of the room, and provides a realistic simulation of room reverberation. The room parameters chosen in this simulation are illustrated in Figure 4.5. The results of the experiment are shown in Figures 4.6 and 4.7. As for the previous experiment, the K-SVD algorithm is outperformed

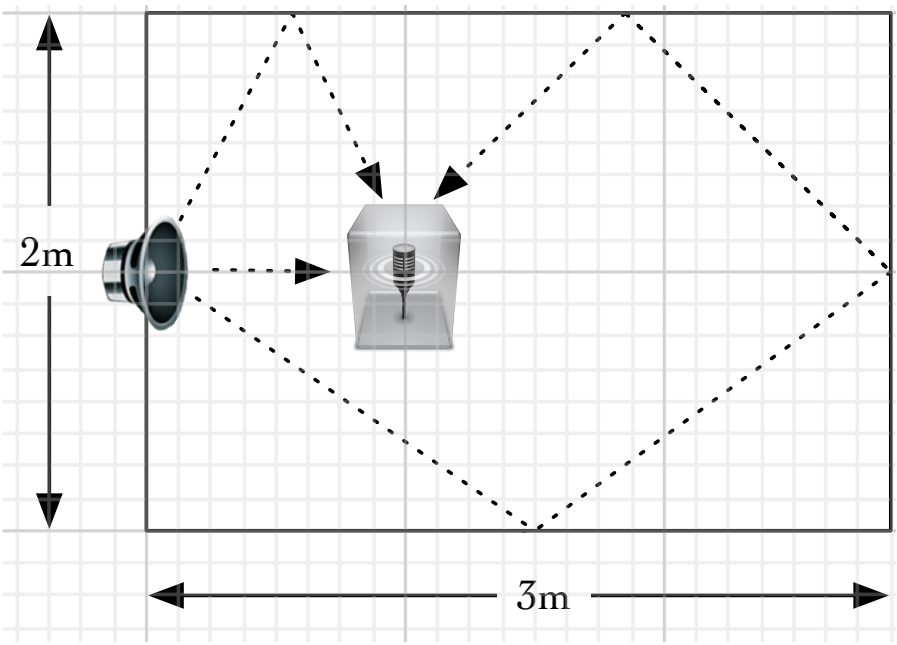


Figure 4.5: Room parameters used to generate an impulse response according to the image method. The floor dimensions (width times length) are 2x3 meters and the height of the room is 3m. The loudspeaker is located at position $[0, 1, 1.5]$m, while the microphone is located at position $[1, 1, 1.5]$m, assuming that the coordinates represent length, width and height respectively. The reflective coefficient of the walls modelling the degree of acoustic reflectivity of the material in the room was set to 0.5 from a range that goes from 0 (totally absorbent) to 1 (totally reflective).

by the D$\boldsymbol{h}$-BCD mouthed, and similar trends can be observed regarding the box plots in Figures 4.4 and 4.7.

In general, the fact that D$\boldsymbol{h}$-BCD outperforms S$\boldsymbol{h}$-BCD suggests that constraining the solution to belong to the feasible set from where the test data were generated is not a good strategy, while performing an unconstrained optimization of the impulse response allows for the necessary flexibility required to minimise the non-convex cost function whenever the initialisation is far from a local minimum. Moreover, the fact that K-SVD is outperformed by D$\boldsymbol{h}$-BCD indicates that taking into account the particular structure of the dictionary and reducing therefore the number of free parameters of the optimization from the whole set of atoms to the impulse response coefficients can lead to significant improvements.

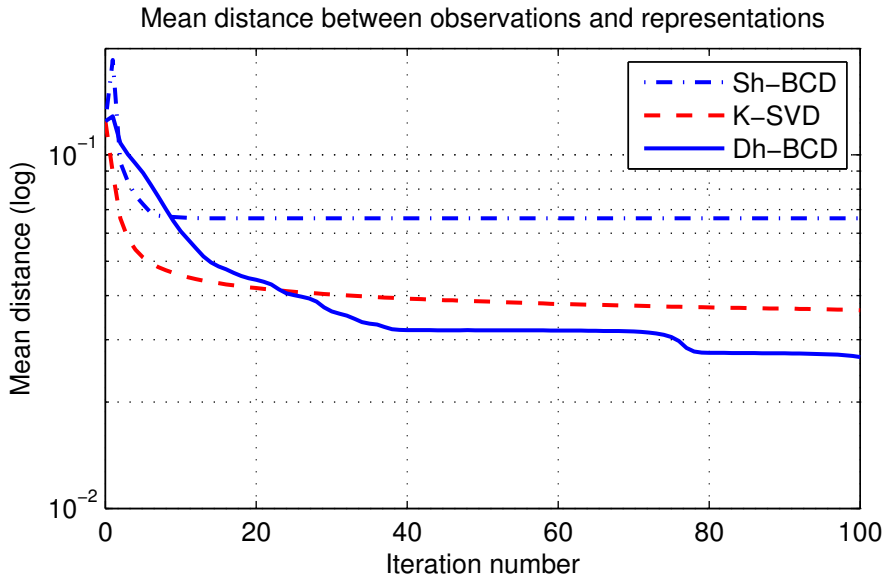


Figure 4.6: Average distance between observed convolved variables and sparse approximation model as a function of the iteration number of BCD and K-SVD (average over 20 trials of the experiment).

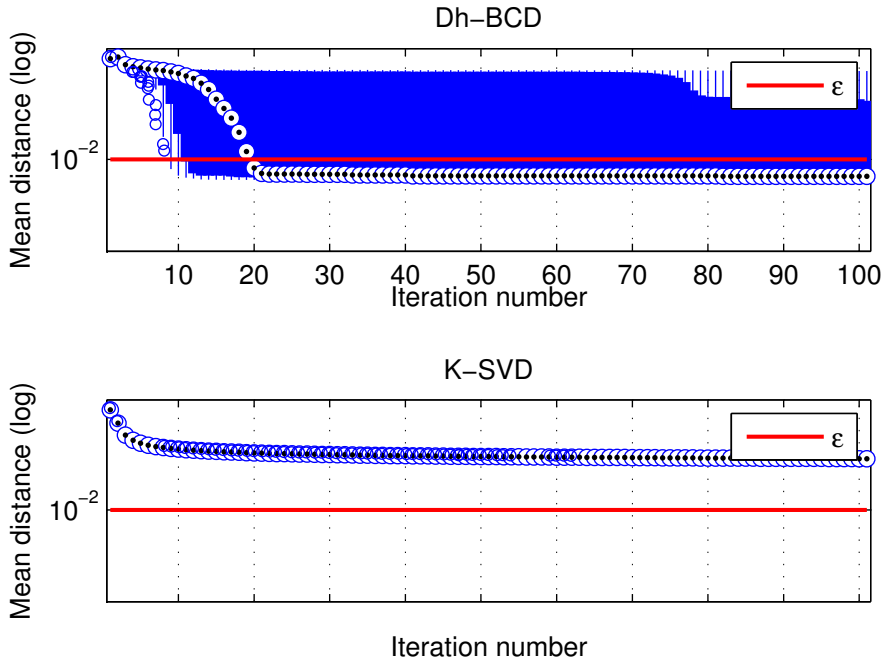


Figure 4.7: Boxplot comparison of the average distance obtained with D$\boldsymbol{h}$-BCD and K-SVD over 20 trials of the experiment. For each iteration, the central mark is the median, the edges of the boxes are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and outliers are plotted individually.

4.4.2 Sparsity phase-transition

The block coordinate descent strategy described in section 4.3.2 proved to be effective in learning a dictionary for a problem with sparse source signals and a sparse non-negative impulse response. In particular, the version employing an unconstrained estimation of the impulse response is able to provide a representation of the observed signals with small residual error. However, as in every sparse approximation problem, the number of active atoms contributing to the synthesis of the input data plays a crucial role. For this reason, we tested the algorithms varying the normalized diversity of source signals and impulse response between 1% and 25% of the respective dimensions, again comparing the results with the K-SVD algorithm. Figure 4.8 shows the contours plot of the residual error achieved at the end of the optimization by the various methods, along with a comparison plot which shows the best performing technique in each point of the sources/impulse response normalized diversity plan.

As we might expect, the two variants of the proposed block coordinate descent method perform well when the source signals and the impulse response are sparse, exhibiting a slightly stronger dependence on the sources normalized diversity. The results for the K-SVD algorithm, on the other hand, seem to depend strongly on the normalized diversity of the impulse response, presenting also a slight drop in correspondence with a source normalized diversity of 0.05. Overall, the comparison plot reveals that, as long as the sources normalized diversity is below 10% of the signals dimension N, and the impulse response is sufficiently sparse, then K-SVD is outperformed by D$\boldsymbol{h}$-BCD . This condition is not unrealistic and corresponds to the common assumption $S \ll N$ made throughout most of the literature on sparse representation.

4.5 Summary

In this chapter a novel method for dictionary learning of convolved signals was presented. Starting from the observation that convolution strongly affects the residual error of sparse approximation, a new strategy for the sparse approximation of convolved signals has been devised.

A model where observed signals are obtained as sparse linear combinations of a convolved dictionary was used to obtain an optimization problem aimed at learning from

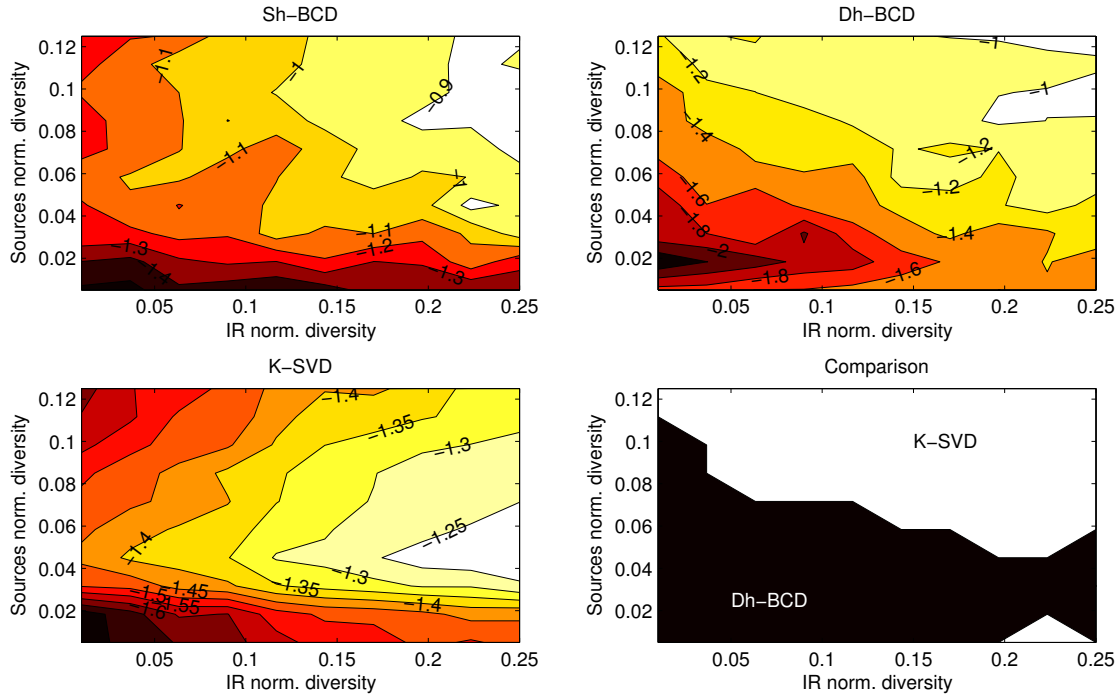


Figure 4.8: BCD and K-SVD results for various densities of source signals and impulse response (average over 20 trials). The values appearing along the contours plots represent the average distance between the observed data and their sparse approximation achieved after 50 iterations by the various algorithms on a logarithmic scale. Since the dictionary is two times over-complete, a completely dense representation corresponds to a source normalized diversity $||\boldsymbol{x}||_0 / K = 0.5$.

observed data a matrix of sparse approximation coefficients and an impulse response applied to the dictionary atoms. Following a strategy analogous to the one employed in traditional dictionary learning which consists in sparse coding followed by dictionary update, the matrix $\boldsymbol{X}$ and the vector $\boldsymbol{h}$ are optimized keeping the dictionary $\boldsymbol{\Phi}$ fixed.

Both an unconstrained and a constrained version of the impulse response optimization step labelled D$\boldsymbol{h}$-BCD and S$\boldsymbol{h}$-BCD respectively have been employed to approximate sets of synthetic signals and compared to the K-SVD dictionary learning algorithm. The results show that D$\boldsymbol{h}$-BCD outperforms the two other methods. They suggest that, in the case of convolved observations generated from anechoic signals that admit a sparse representation in a known dictionary, learning the impulse response can be more efficient than learning a new dictionary with a standard dictionary learning algorithm. Moreover, the unconstrained version of the proposed algorithm outperforms the constrained one, even if the observed data have been generated according to a constrained model.

Finally, numerical experiments comparing signals generated with different levels of

normalized diversity show the range of parameters within which the above claim holds. Overall, as long as signals and impulse responses are sparse, D$\boldsymbol{h}$-BCD is a better choice than K-SVD for sparse approximation.

Future research should investigate dictionary learning of convolved signals on non-synthetic data, including audio and images. In addition, the work initiated in collaboration with Mr. Benichoux on constrained convolutive source separation that has been mentioned in Section 4.3 can prove to be a worthwhile research avenue for the future.

Chapter 5

Incoherent dictionary learning

5.1 Learning incoherent dictionaries

This Chapter deals with learning dictionaries for sparse approximation that allow one to express a set of training signals with a small residual error and contain atoms that are dissimilar to each other. This mixed objective comprising extrinsic and intrinsic properties of the dictionary is analysed in the context of sparse recovery and sparse approximation, and novel optimisation algorithms are proposed to address it.

5.1.1 Sparse approximation and dictionary learning models

We consider a sparse synthesis model where a signal $\boldsymbol{y} \in \mathbb{R}^N$ is approximated by a sparse linear combination of atoms $\{\boldsymbol{\phi}_k\}_{k=1}^K, \boldsymbol{\phi}_k \in \mathbb{R}^N$, as already introduced in Section 2.3. Arranging the atoms along the columns of the *dictionary* matrix $\boldsymbol{\Phi}$, we can express the model as in (2.9):

$$\boldsymbol{y} \approx \boldsymbol{\Phi}\boldsymbol{x} \tag{5.1}$$

where $\boldsymbol{x}$ is a sparse vector of representation coefficients, with $||\boldsymbol{x}||_0 \leq S$. The parameters of this model can be determined by solving a sparse approximation problem and optimizing:

$$\begin{aligned} \boldsymbol{x}^\star = \arg\min_{\boldsymbol{x}} \ & ||\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{x}||_2 \\ \text{such that } & ||\boldsymbol{x}||_0 \leq S \end{aligned} \tag{5.2}$$

as already introduced in (2.12). A batch sparse approximation model is defined as in (2.22) by stacking a set of observed signals $\{\boldsymbol{y}_m \in \mathbb{R}^N\}_{m=1}^M$ along the columns of the matrix $\boldsymbol{Y} \in \mathbb{R}^{N \times M}$:

$$\boldsymbol{Y} \approx \boldsymbol{\Phi} \boldsymbol{X} \tag{5.3}$$

where $\boldsymbol{X}$ is a sparse matrix whose columns contain the vectors $\boldsymbol{x}_m$ of representation coefficients. Corresponding dictionary learning problems can be proposed as detailed in Section 2.6 and tackled through the alternate optimization consisting in sparse coding followed by dictionary update as detailed in Section 2.7.

The sparse approximation (5.2) that is at the core of the sparse coding step of dictionary learning has been proved to be an NP hard problem [24], and a great number of sub-optimal algorithms that run in polynomial time [17, 76, 77, 82] have been developed in order to tackle it, as already detailed in Section 2.4. An important research effort has been devoted to understand how the different strategies and algorithms for sparse modelling perform in different settings. For example, sparse recovery deals with retrieving a sparse signal from a set of incomplete measurements and has applications in the field of compressive sampling [16], while sparse approximation is concerned with how efficiently a general signal can be approximated by linear combinations of a few atoms from an over-complete dictionary [25, 111].

The theorems that have been proposed in the literature to this aim link the success of the algorithms with the *coherence* of the dictionary.

5.1.2 Dictionary coherence and its role in the performance of sparse algorithms

The coherence of a dictionary indicates the degree of similarity between different atoms or different collections of atoms. A simple measure that has been proposed in the literature is the mutual coherence $\mu(\boldsymbol{\Phi})$, which is defined as the maximum absolute inner product between any two different atoms of the dictionary:

$$\mu(\boldsymbol{\Phi}) \stackrel{\text{def}}{=} \max_{i \neq j} \left| \langle \boldsymbol{\phi}_i, \boldsymbol{\phi}_j \rangle \right| \tag{5.4}$$

and is zero for orthogonal bases. For clarity of notation, in the reminder of this Chapter the dependancy on the dictionary $\boldsymbol{\Phi}$ will be omitted whenever unambiguous from the

context.

Tropp [111] showed that, given a sparse signal generated according to the model (5.1), the orthogonal matching pursuit algorithm (OMP) [82] is guaranteed to retrieve the correct support of the representation coefficients from an observed signal $\boldsymbol{y}$ and a given dictionary $\boldsymbol{\Phi}$ if:

$$\mu < \frac{1}{(2S-1)} \tag{5.5}$$

and further refined this bound by considering the cumulative coherence which involves the sum of correlations between an atom $\boldsymbol{\phi}_i$ and an S-dimensional sub-dictionary that does not include it. The bound (5.5) is referred to as a worst-case bound, and it is linked to the condition number of an arbitrary sub-dictionary of the matrix $\boldsymbol{\Phi}$ [114]. Less pessimistic results can be obtained by considering random sub-dictionaries, an insight that leads to average-case bounds expressed as the probability of success or failure of a sparse recovery algorithm [93].

In general, the results reported in the literature indicate that sparse recovery succeeds in a wide range of problem settings whenever the mutual coherence μ is low and the cumulative coherence grows slowly as a function of the number of active atoms. In particular, equation (5.5) implies that only signals which are synthesised from $S < \frac{1}{2} + \frac{1}{2\mu}$ active atoms are guaranteed to be correctly recovered. However, it can be proved [106] that for a $N \times K$ dictionary, the mutual coherence is lower-bounded by:

$$\mu \geq \sqrt{\frac{K-N}{N(K-1)}}. \tag{5.6}$$

As an illustrative example, a dictionary containing $K = 200$ atoms in $N = 100$ dimensions has a mutual coherence that is lower-bounded by $\mu \geq 0.07$, and the sparse representation of a signal generated with such a dictionary is guaranteed to be correctly retrieved if the number of active atoms is $S_{\max} \leq 7$.

Based on results for sparse recovery, Gribonval and Vandergheynst [47] extended the work of Tropp [111] and proved a stability result regarding the matching pursuit (MP) [71] algorithm. In particular they prove that, given a signal $\boldsymbol{y}$ and an optimal S-term sparse approximation $\hat{\boldsymbol{y}}_S$ (that is, the solution that would be returned by a combinatorial search over all the possible sets of S atoms),

- At each step $t < T_S$, MP produces an approximation $\tilde{\boldsymbol{y}}^{(t)}$ using the correct atoms from the support of the optimal approximation $\hat{\boldsymbol{y}}_S$.
- The approximation error at step T_S given by $\left|\left|\boldsymbol{y} - \tilde{\boldsymbol{y}}^{(T_S)}\right|\right|_2 \leq C \, ||\boldsymbol{y} - \hat{\boldsymbol{y}}_S||_2$ is upper-bounded by the approximation error attained by the optimal approximation multiplied by a constant.

The number of steps T_S that can be proved to select the correct support of the sparse approximation is inversely proportional to the coherence of the dictionary. In other words, a low coherence ensures that the MP algorithm correctly identify the support of a *brute force* combinatorial solution for a large number of steps.

It is fair to stress that the bounds detailed above are only sufficient (and not necessary) conditions for proving stability and recovery results. They do not imply that incoherent dictionaries are necessarily better for sparse approximation. However, experimental results presented in Section 5.5 will highlight the advantages of mutually incoherent dictionaries in the context of sparse approximation.

5.1.3 Learning incoherent dictionaries

If the advantage of learning incoherent dictionaries for coding applications lies in the success of approximation algorithms, the results on sparse recovery place the emphasis on retrieving the true support of the signals to be analysed. This is a desirable property whenever sparse approximations are sought in order to reveal an underlying structure or clustering in the data.

For example, morphological component analysis [10, 9] decomposes a signal over a set of dictionaries that have been previously learned from different training data consisting of morphologically dissimilar classes (i.e., edges and textures for an image, or different classes of instruments for a musical audio signal). The mutual incoherence between different learned sets of atoms is a prerequisite that allows for a sparse coding where the position of the non-zero coefficients can be informative for classification and source separation applications.

In addition, Dai et al. [20] recently observed that the K-SVD dictionary learning algorithm [4] can converge to ill-conditioned dictionaries that perform poorly for sparse approximation. They proposed a novel technique to address this issue that introduces a

penalised optimisation in the dictionary update step. Tropp [114] showed that the coherence of a dictionary is linked to the condition number of its sub-dictionaries (i.e, matrices defined by selecting a subset of the atoms), and used this relation to prove average-case results on sparse recovery for ℓ_1 based algorithms. This implies that achieving a low mutual coherence results in well-conditioned sub-dictionaries and further motivates the objective of the work presented in this Chapter.

Finally, Gleichman and Eldar introduced the *blind* compressed sensing framework [42] as a generalisation of the compressive sampling technique introduced in Section 2.9.1 where the dictionary in which the signals are supposed to be sparse is unknown. In their formulation, a set of compressively sampled variables $\boldsymbol{Z} = \boldsymbol{M}\boldsymbol{\Phi}\boldsymbol{X}$ is derived from a known measurement matrix $\boldsymbol{M}$ and an unknown dictionary $\boldsymbol{\Phi}$. Dictionary learning is employed in order to factorise the observed data as $\boldsymbol{Z} \approx \boldsymbol{\Psi}\boldsymbol{X}$ and a post-processing step is employed to factorise the learned dictionary in the product $\boldsymbol{\Psi} \approx \boldsymbol{M}\boldsymbol{\Phi}$ to then reconstruct the signals $\boldsymbol{Y} = \boldsymbol{\Phi}\boldsymbol{X}$ exploiting constraints on $\boldsymbol{\Phi}$. In this context, learning an incoherent dictionary $\boldsymbol{\Psi}$ can promote a unique factorization and a correct recovery of the signals $\boldsymbol{Y}$.

5.2 Previous work on incoherent dictionaries

This Section presents previous work on learning incoherent dictionaries, including methods that inspired the algorithms that constitute the main contributions of this Chapter or provided benchmark techniques for the evaluation of the proposed techniques.

5.2.1 Constructing Grassmannian frames with iterative projections

A Grassmannian frame is a collection of atoms that have unit norm and minimal mutual coherence. It can be proved that, for an $N \times K$ dictionary, the mutual coherence is bounded by (5.6), and the lower bound is reached when the dictionary is an equiangular tight frame, that is, a Grassmannian frame where any pair of different atoms have the same absolute inner product [106]. It is also worth noting that equiangular tight frames do not exist for any pair (N, K), but necessarily (and not sufficiently) require $K \leq \frac{1}{2}N(N+1)$ if the atoms are real or $K \leq N^2$ if the atoms are complex.

Constructing Grassmannian frames is an open research problem for which there is generally no analytic solution. One possible approach is to use an iterative projection method [116]. To illustrate this algorithm, we define two constraint sets, namely the

structural constraint set $\mathcal{K}_{\mu_0}$ as the set of symmetric square matrices with unit diagonal values and off-diagonal values with magnitude smaller or equal than μ_0:

$$\mathcal{K}_{\mu_0} \stackrel{\text{def}}{=} \{\boldsymbol{K} \in \mathbb{R}^{K\times K} : \boldsymbol{K} = \boldsymbol{K}^T, \operatorname{diag}(\boldsymbol{K}) = \mathbf{1}, \max_{i>j} |k_{i,j}| \leq \mu_0 \leq 1\}. \tag{5.7}$$

and the spectral constraint set $\mathcal{F}$ as the set of symmetric positive semidefinite square matrices with rank smaller than or equal to N:

$$\mathcal{F} \stackrel{\text{def}}{=} \left\{\boldsymbol{F} \in \mathbb{R}^{K\times K} : \boldsymbol{F} = \boldsymbol{F}^T, \operatorname{eig}(\boldsymbol{F}) \geq \mathbf{0}, \operatorname{rank}(\boldsymbol{F}) \leq N\right\}$$

In the above expressions, the operators $\operatorname{diag}(\cdot)$ and $\operatorname{eig}(\cdot)$ return the vector of diagonal elements and the vector of eigenvalues of their arguments respectively.

The iterative projection algorithm starts from an initial dictionary $\boldsymbol{\Phi}$, calculates its Gram matrix $\boldsymbol{G} \stackrel{\text{def}}{=} \boldsymbol{\Phi}^{\mathcal{H}}\boldsymbol{\Phi}$, and iteratively projects it onto the sets $\mathcal{K}_{\mu_0}$ and $\mathcal{F}$ until a stopping criterion is met.

- *Projection onto the structural constraint set.* Given an arbitrary Gram matrix $\boldsymbol{G}$, its projection $\boldsymbol{K} = \mathcal{P}_{\mathcal{K}_{\mu_0}}(\boldsymbol{G})$ onto the structural constraint set can be obtained by setting its diagonal values to one and by limiting the magnitude of its off-diagonal values:

  1. Set $\operatorname{diag}(\boldsymbol{K}) = \mathbf{1}$
  2. Limit the off-diagonal elements so that, for $i \neq j$,

$$k_{i,j} = \operatorname{Limit}(g_{i,j}, \mu_0) \stackrel{\text{def}}{=} \begin{cases} g_{i,j} & \text{if} \quad |g_{i,j}| \leq \mu_0 \\ \mu_0 & \text{if} \quad g_{i,j} > \mu_0 \\ -\mu_0 & \text{if} \quad g_{i,j} < -\mu_0 \end{cases}$$

- *Projection onto the spectral constraint set.* Given an arbitrary dictionary $\boldsymbol{\Phi}$, its Gram matrix $\boldsymbol{G}$ is by construction a symmetric, positive semidefinite matrix. Its projection $\boldsymbol{F} = \mathcal{P}_{\mathcal{F}}(\boldsymbol{G})$ onto the spectral constraint set $\mathcal{F}$ can be obtained through the following steps:

  1. Calculate an eigenvalue decomposition (EVD) $\boldsymbol{G} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$

2. Threshold the eigenvalues by keeping only the N largest positive ones.

$$\bar{\boldsymbol{\Lambda}} = [\text{Thresh}(\boldsymbol{\Lambda}, N)]_{i,i} \stackrel{\text{def}}{=} \begin{cases} \lambda_{i,i} & \text{if} \quad i \leq N \text{ and } \lambda_{i,i} > 0 \\ 0 & \text{if} \quad i > N \text{ or } \lambda_{i,i} \leq 0 \end{cases}$$

where the eigenvalues in $\boldsymbol{\Lambda}$ are ordered from the largest to the smallest. Following this step, at most N eigenvalues of the Gram matrix are different from zero. It is worth noting that in the original formulation of the IP algorithm [116] the N largest eigenvalues in the matrix $\boldsymbol{\Lambda}$ are set to K/N as this results in the spectrum of the Gram matrix of an equiangular tight frame. However, relaxing this constraint as proposed here led to better numerical results.

3. Update the Gram matrix as $\boldsymbol{F} = \boldsymbol{Q}\,\text{Thresh}(\boldsymbol{\Lambda}, N)\boldsymbol{Q}^T$, so that $\text{rank}(\boldsymbol{F}) \leq N$.

Once the Gram matrix has been iteratively projected onto the two sets and the stopping criterion has been met, it is factorized as the product

$$\boldsymbol{G} = \boldsymbol{\Phi}^T\boldsymbol{\Phi} \tag{5.8}$$

through the following steps:

1. Calculate an eigenvalue decomposition (EVD) $\boldsymbol{G} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$

2. Set $\boldsymbol{\Phi} = \text{Thresh}(\boldsymbol{\Lambda}, N)^{\frac{1}{2}}\boldsymbol{Q}^T$

so that $\boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{Q}\,\text{Thresh}(\boldsymbol{\Lambda}, N)\boldsymbol{Q}^T$.

Note that at this point, the dictionary is not guaranteed to have a mutual coherence bounded by μ_0. The intersection between the sets $\mathcal{F}$ and $\mathcal{K}_{\mu_0}$ may be empty for certain values of N, K and μ_0 (in fact, it is empty whenever μ_0 is lower than the bound (5.6)). The iterative projections algorithm is only guaranteed to converge to an accumulation point [116] consisting of a pair of matrices $\bar{\boldsymbol{F}} \in \mathcal{F}$ and $\bar{\boldsymbol{K}} \in \mathcal{K}_{\mu_0}$ that are not necessarily located at a minimal distance between the constraint sets. However, we found in our numerical experiments that the algorithm works well for values of μ_0 close to the lower bound (5.6), providing a dictionaries with constrained mutual coherence.

Algorithm 9 summarises the steps of the IP method.

Algorithm 9: Iterative projections (IP)

Input: $\boldsymbol{\Phi}, \mu_0, I$
Output: $\boldsymbol{\Phi}^\star$
// Initialisation
$i \leftarrow 1$;
// Calculate Gram matrix
$\boldsymbol{G} \leftarrow \boldsymbol{\Phi}^T\boldsymbol{\Phi}$;
while $i \leq I$ *or* $\mu(\boldsymbol{\Phi}) \leq \mu_0$ **do**
  // Project Gram onto the structural constraint set
  $\text{diag}(\boldsymbol{G}) \leftarrow \mathbf{1}$;
  $\boldsymbol{G} \leftarrow \text{Limit}(\boldsymbol{G}, \mu_0)$;
  // Project Gram onto the spectral constraint set
  $[\boldsymbol{Q}, \boldsymbol{\Lambda}] \leftarrow \text{EVD}(\boldsymbol{G})$;
  $\boldsymbol{\Lambda} \leftarrow \text{Thresh}(\boldsymbol{\Lambda}, N)$;
  $\boldsymbol{G} \leftarrow \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$;
  $i \leftarrow i + 1$;
end
// Obtain incoherent dictionary from its Gram matrix
$\boldsymbol{\Phi}^\star \leftarrow \boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{Q}^T$;

5.2.2 Method of optimal coherence-constrained directions (MOCOD)

Ramirez et al. [87] proposed a dictionary learning algorithm inspired by the method of optimal directions (MOD) [32] in which the sparse approximation is performed using a novel penalty term derived from a probabilistic formulation of the sparse model (5.1), and the dictionary update step is modified in order to promote mutually incoherent atoms.

In particular, the incoherence objective is pursued by introducing in the dictionary learning optimization the term $||\boldsymbol{G} - \boldsymbol{I}||_{\text{F}}$ where each element g_{ij} of the Gram matrix $\boldsymbol{G} \stackrel{\text{def}}{=} \boldsymbol{\Phi}^T\boldsymbol{\Phi}$ contains the inner product between the i-th and the j-th atom of the dictionary. This expression measures the Frobenius distance between the Gram matrix of the dictionary and the identity matrix, which corresponds to the Gram matrix of an orthonormal dictionary whose mutual coherence is zero.

Overall, the optimization presented in [87] reads as:

$$(\boldsymbol{\Phi}^\star, \boldsymbol{X}^\star) = \underset{\boldsymbol{\Phi}, \boldsymbol{X}}{\arg\min}\ ||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\text{F}}^2 + \tau \sum_{m,n} \log(|x_{km}| + \beta) + \zeta\, ||\boldsymbol{G} - \boldsymbol{I}||_{\text{F}}^2 + \eta \sum_{k=1}^{K} \left(||\boldsymbol{\phi}_k||_2^2 - 1\right)^2. \tag{5.9}$$

In this unconstrained minimisation, the first term represents the modelling error, while the desired properties of dictionary and representation coefficients are enforced through penalty terms. In particular, the penalty factor multiplied by τ promotes sparsity of

the representation coefficients, while the factors multiplied by ζ and η promote mutual incoherence and unit norm of the dictionary atoms respectively.

In order to solve this optimization, the sparse approximation is followed by a MOCOD dictionary update step, obtained by setting to zero the derivative of the above cost function with respect to the dictionary $\boldsymbol{\Phi}$. The resulting update can be written as [87]:

$$\boldsymbol{\Phi}' = \left(\boldsymbol{Y}\boldsymbol{X}^T + 2(\zeta+\eta)\boldsymbol{\Phi}\right)\left[\boldsymbol{X}\boldsymbol{X}^T + 2\zeta\boldsymbol{G} + 2\eta\,\mathrm{diag}(\boldsymbol{G})\right]^{-1}.$$

Note that setting to zero the penalty factors ζ and η results in the MOD update [32].

5.2.3 Incoherent dictionary design and dictionary preconditioning

Yaghoobi et al. [121] proposed a dictionary design method for coding of audio signals where the parameters of gammatone atoms [104] are optimized in order to minimise the mutual coherence of the resulting dictionary. In this work, the authors are inspired by the iterative projections method described in Section 5.2.1 (that also is at the core of one of the IPR algorithm described in Section 5.4), and show through experimental results the advantages of using an incoherent dictionary for sparse recovery and sparse approximation. Despite the similarity in the motivation and in part of the optimization technique between the work by Yaghoobi et al. and the algorithm that will be proposed in Section 5.4, dictionary design is substantially different from dictionary learning: while the former involves optimizing the parameters of a set of parametric functions that are designed to be suited for a given class of signals, the latter is adapted to an arbitrary set of observed variables and can therefore be extended to classes of signals for which an efficient dictionary is not known. Moreover, in the case of dictionary design there is not a mixed objective consisting of good approximation and mutual incoherence because the former is implicitly assumed given the nature of the parametric functions and of the signals to be analysed. For this reason the experimental comparisons in the reminder of this chapter are limited to dictionary learning algorithms.

Apart from incoherent dictionary learning or design, Schnass and Vandergheynst [94] presented a method for dictionary preconditioning that aims at tackling the problem of coherent dictionaries for sparse recovery. In this work, a sensing matrix is multiplied by a coherent dictionary in order to obtain an equivalent sparse recovery problem with

low cross-cumulative coherence (i.e. the cumulative coherence between atoms of the sensing matrix and atoms of the dictionary), and improve the performance of greedy sparse approximation algorithms. Although related to the present work, we choose not to further detail or benchmark this algorithm as it does not involve dictionary learning.

5.3 Incoherent K-SVD

The work presented in this Section resulted from a collaboration with Dr. Boris Mailhé, a post-doctoral research assistant at the Centre for Digital Music at Queen Mary University of London. The incoherent K-SVD algorithm appeared in a joint publication at the International Conference on Acoustics, Speech and Signal Processing (ICASSP) [62].

Although there has been constant communication with my co-author while working on this project, my main contribution consisted in the design and implementation of the experimental section that is aimed at evaluating the incoherent K-SVD algorithm. The ideation and implementation of the method itself is to be attributed to Boris Mailhé.

Both the method and the experimental results presented in Section 5.4 have been designed and implemented by myself.

5.3.1 Dictionary de-correlation

Apart from a penalised optimization such as that described in Section 5.2.2, an alternative strategy for learning incoherent dictionaries can be pursued by including a de-correlation step into the iterative scheme illustrated in Section 2.6. At each iteration of the dictionary learning algorithm consisting of sparse approximation followed by dictionary update, we add the following optimization problem:

$$\begin{aligned} \mathbf{\Phi}^\star = \underset{\mathbf{\Phi}\in\mathcal{D}}{\arg\min}\ & \mathcal{C}(\mathbf{\Phi}) \\ \text{such that } & \mu(\mathbf{\Phi}) \leq \mu_0 \end{aligned} \tag{5.10}$$

where the objective $\mathcal{C}(\mathbf{\Phi})$ is a cost function that expresses the approximation quality of the dictionary and μ_0 is a fixed target mutual coherence level. Therefore, an incoherent dictionary learning algorithm realized with a de-correlation step starts from an initial $\mathbf{\Phi}^{(0)}$ and proceeds by solving the following sub-problems at each iteration t:

Sparse coding : given a fixed dictionary $\boldsymbol{\Phi}^{(t)}$, a sparse approximation $\boldsymbol{X}^{(t)}$ is optimized using any suitable algorithm.

Dictionary update : given a fixed matrix of approximation coefficients $\boldsymbol{X}^{(t)}$, a new (possibly mutually coherent) dictionary $\tilde{\boldsymbol{\Phi}}^{(t+1)}$ is updated in order to improve the objective of the dictionary learning optimization, subject to optional constraints.

Dictionary de-correlation : given $\boldsymbol{X}^{(t)}$ and $\tilde{\boldsymbol{\Phi}}^{t+1}$, a de-correlated dictionary $\boldsymbol{\Phi}^{(t+1)}$ is optimized according to (5.10).

The mutual coherence constraint present in (5.10) is non-convex, as shown in Appendix A.1. Therefore a gradient descent optimization of (5.10) is not guaranteed to keep the solution into the constraint set.

5.3.2 The INK-SVD algorithm

In the INK-SVD algorithm, the de-correlation problem consists in finding the closest dictionary to a given dictionary (in a Frobenius norm sense), subject to a mutual coherence constraint. The optimization (5.10) can be explicitly written as:

$$\begin{aligned} \boldsymbol{\Phi}^{\star} = \underset{\boldsymbol{\Phi}\in\mathcal{D}}{\arg\min} \ \left|\left|\tilde{\boldsymbol{\Phi}} - \boldsymbol{\Phi}\right|\right|_{\mathrm{F}} \\ \text{such that } \mu(\boldsymbol{\Phi}) \leq \mu_0 \end{aligned} \tag{5.11}$$

where $\tilde{\boldsymbol{\Phi}}$ is the matrix resulting from the dictionary update stage of the learning algorithm. In order to devise an algorithm to tackle this optimization, let us first consider a simple example consisting in a dictionary formed by only two atoms.

De-correlation of two atoms

Let the initial dictionary $\tilde{\boldsymbol{\Phi}}$ be composed of only two atoms $\tilde{\boldsymbol{\phi}}_1$ and $\tilde{\boldsymbol{\phi}}_2$ of unit norm with a correlation higher than μ_0. In this simple case we can directly express the optimum of Problem (5.11).

Let us assume without loss of generality that $\left\langle \tilde{\boldsymbol{\phi}}_1, \tilde{\boldsymbol{\phi}}_2 \right\rangle > 0$ (the opposite case can be derived by considering the pair $(\tilde{\boldsymbol{\phi}}_1, -\tilde{\boldsymbol{\phi}}_2)$) and let $\tilde{\theta}$ be the half-angle between $\tilde{\boldsymbol{\phi}}_1$ and $\tilde{\boldsymbol{\phi}}_2$. Problem (5.11) only has two degrees of freedom because of the normalization constraint. We choose the half-angle $\theta^{\star}$ between $\boldsymbol{\phi}_1^{\star}$ and $\boldsymbol{\phi}_2^{\star}$ and the angle α between the directions of

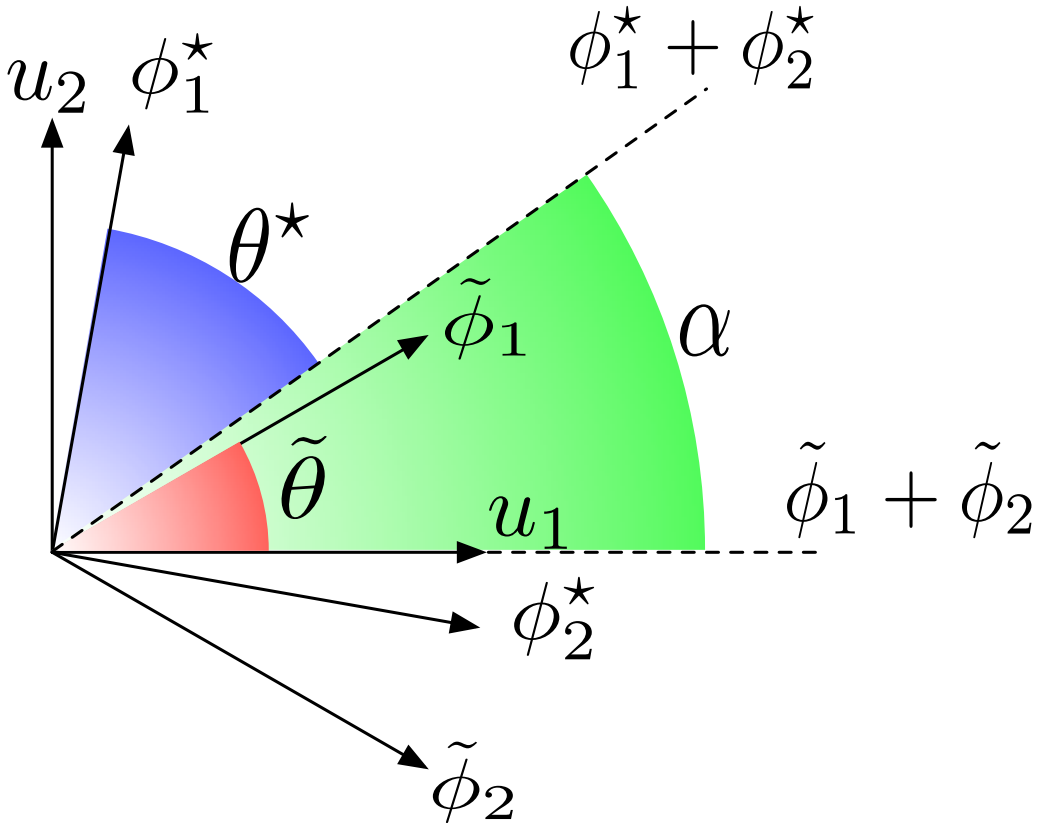


Figure 5.1: De-correlation of two atoms. For the optimal de-correlation we have $\alpha = 0$ and the pair $(\boldsymbol{\phi}_1^\star, \boldsymbol{\phi}_2^\star)$ would be symmetric with respect to $\boldsymbol{u}_1$.

the sums $\tilde{\boldsymbol{\phi}}_1 + \tilde{\boldsymbol{\phi}}_2$ and $\boldsymbol{\phi}_1^\star + \boldsymbol{\phi}_2^\star$ for parameters as shown on Figure 5.1. In the orthonormal basis

$$(\boldsymbol{u}_1, \boldsymbol{u}_2) = \left(\frac{\tilde{\boldsymbol{\phi}}_1 + \tilde{\boldsymbol{\phi}}_2}{\left|\left|\tilde{\boldsymbol{\phi}}_1 + \tilde{\boldsymbol{\phi}}_2\right|\right|_2}, \frac{\tilde{\boldsymbol{\phi}}_1 - \tilde{\boldsymbol{\phi}}_2}{\left|\left|\tilde{\boldsymbol{\phi}}_1 - \tilde{\boldsymbol{\phi}}_2\right|\right|_2} \right) \tag{5.12}$$

all the considered vectors have a simple expression:

$$\tilde{\boldsymbol{\Phi}} = \left(\tilde{\boldsymbol{\phi}}_1, \tilde{\boldsymbol{\phi}}_2\right) = \begin{pmatrix} \cos\tilde{\theta} & \cos\tilde{\theta} \\ \sin\tilde{\theta} & -\sin\tilde{\theta} \end{pmatrix} \tag{5.13}$$

$$\boldsymbol{\Phi}^\star = (\boldsymbol{\phi}_1^\star, \boldsymbol{\phi}_2^\star) = \begin{pmatrix} \cos(\alpha + \theta^\star) & \cos(\alpha - \theta^\star) \\ \sin(\alpha + \theta^\star) & \sin(\alpha - \theta^\star) \end{pmatrix}. \tag{5.14}$$

We can then express the mutual coherence constraint as:

$$|\langle \boldsymbol{\phi}_1^\star, \boldsymbol{\phi}_2^\star \rangle| = |\cos 2\theta^\star| \leq \mu_0 \tag{5.15}$$

and the objective function as:

$$\begin{aligned}
\left|\left|\tilde{\boldsymbol{\phi}}_1 - \boldsymbol{\phi}_1^\star\right|\right|_2^2 &= 2 - 2\cos(\tilde{\theta} - \theta^\star - \alpha) \\
\left|\left|\tilde{\boldsymbol{\phi}}_2 - \boldsymbol{\phi}_2^\star\right|\right|_2^2 &= 2 - 2\cos(\tilde{\theta} - \theta^\star + \alpha) \\
\left|\left|\tilde{\boldsymbol{\Phi}} - \boldsymbol{\Phi}^\star\right|\right|_{\mathrm{F}}^2 &= 4 - 4\cos(\tilde{\theta} - \theta^\star)\cos(\alpha).
\end{aligned} \tag{5.16}$$

If we assume without loss of generality that $\cos(\tilde{\theta} - \theta^\star) > 0$, then the cost function (5.16) is minimal for $\alpha = 0$ and $\theta^\star$ as close to $\tilde{\theta}$ as possible: Problem (5.11) is solved by rotating $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ symmetrically with respect to their mean until their correlation reaches μ_0. The angle $\theta^\star$ is the angle that reaches the equality in Equation (5.15):

$$\cos 2\theta^\star = \mu_0 \tag{5.17}$$

$$\theta^\star = \frac{\arccos \mu_0}{2} \tag{5.18}$$

and the dictionary $\Phi^\star$ is given by equation (5.14).

General case

In the general case, the previous method provides the steepest descent direction if only one pair of atoms reaches the maximal correlation. However, the coherence function is non-convex with respect to $\boldsymbol{\Phi}$ so following a steepest descent does not guarantee to find a global minimum. Instead of using a descent method, we chose to de-correlate the dictionary by iterating de-correlations of pairs of atoms. The core idea is simple: as long as there are any atoms with correlation higher than μ_0, select a pair of them and de-correlate them with the method explained in Section 5.3.2.

However, decorrelating two atoms can potentially change correlations with other atoms in the dictionary, so finding the next pair would require to update the correlations after each pair de-correlation. We speed up the process by decorrelating some pairs in parallel. Instead of selecting one pair of atoms at a time, we partition the whole dictionary into high correlation pairs (and single atoms that do not need to be modified), decorrelate all those pairs and only then update the correlations. This is detailed on Algorithm 10.

The partitioning detailed in Algorithm 11 is performed in a greedy way: starting with

Algorithm 10: INK-SVD decorrelation

Input: $\tilde{\boldsymbol{\Phi}}, \mu_0$
Output: $\boldsymbol{\Phi}^\star$
while $\mu(\tilde{\boldsymbol{\Phi}}) > \mu_0$ **do**
  $E = \text{partition}(\tilde{\boldsymbol{\Phi}}, \mu_0)$;
  for $\forall(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) \in E$ **do**
    decorrelate$(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)$;
  end
end

Algorithm 11: INK-SVD partition

Input: $\tilde{\boldsymbol{\Phi}}, \mu_0$
Output: E
`// Initialisation`
$\boldsymbol{\Phi} \leftarrow \tilde{\boldsymbol{\Phi}}$;
$E \leftarrow \emptyset$;
while $\mu(\boldsymbol{\Phi}) > \mu_0$ **do**
  $(i, j) = \arg\max_{i,j} \left|(\boldsymbol{\Phi}^T\boldsymbol{\Phi} - \boldsymbol{I})_{i,j}\right|$;
  $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi} \setminus \{\boldsymbol{\phi}_i, \boldsymbol{\phi}_j\}$;
  $E \leftarrow E \cup \{(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j)\}$;
end

the whole dictionary, pairs of atoms with the highest correlation are grouped together and removed from the set of considered atoms until there are no pairs left with correlation higher than μ_0.

5.3.3 Experimental results

We tested the INK-SVD dictionary learning algorithm in order to assess if it converges to a dictionary that exhibits bounded mutual coherence and good approximation quality. The test signal we used is the musical excerpt `music03_16kHz`, a 16 kHz guitar recording that is part of the data included in SMALLBOX [21], a Matlab toolbox for testing and benchmarking dictionary learning algorithms used in our evaluation. This contains the code needed to reproduce the results presented here[1] and will be further detailed in Appendix B.2. A musical audio signal was chosen because previous informal experiments resulted in K-SVD learning a highly coherent dictionary for this type of data. Additional experiments about incoherent dictionary learning that make use of different audio signals are detailed in Section 5.5.4.

[1] `http://small-project.eu/software-data/smallbox`

We divided the recording into 50% overlapping blocks of 256 samples (corresponding to 16ms) with rectangular windows and arranged the resulting vectors as columns of the training data matrix $\boldsymbol{Y}$. Then, we initialised three twice over-complete dictionaries for sparse approximation using respectively:

- a randomly chosen subset of the training data $\boldsymbol{Y}$
- an over-complete DCT dictionary
- an over-complete Gabor dictionary.

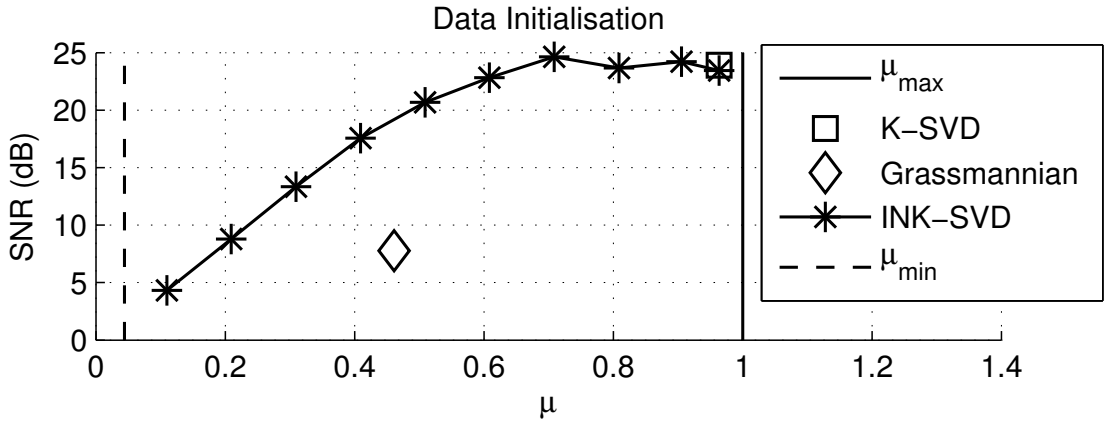
We run the K-SVD dictionary learning algorithm for 20 iterations, allowing for 12 non-zero coefficients in each representation (which corresponds to about 5% of active elements if compared with the dimension of the signals N). We included the proposed INK-SVD de-correlation algorithm and compared it with the iterative projection algorithm that is detailed in section 5.2.1, using the implementation presented in [30, p.30].

Figure 5.2 depicts the results of the experiment. The y-axis illustrates the signal-to-noise-ratio SNR achieved by the dictionary at the end of the optimization, that is defined as:

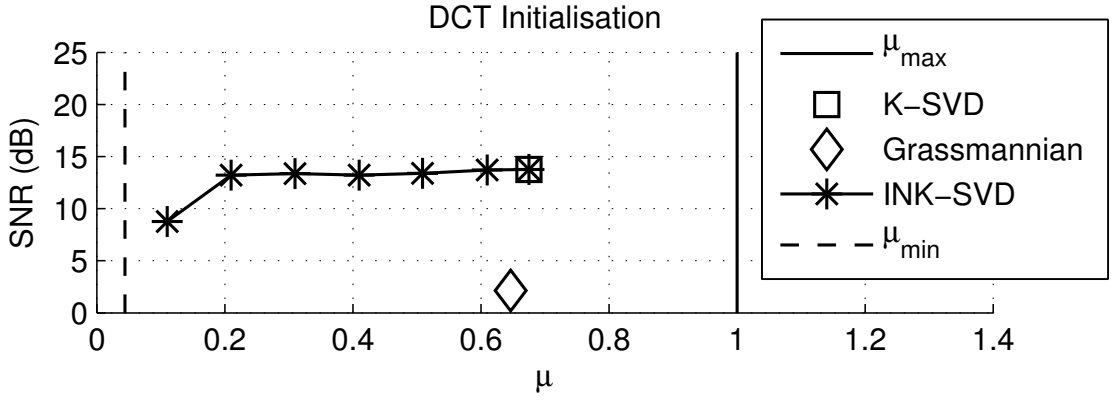
$$\text{SNR}(\boldsymbol{Y}, \boldsymbol{\Phi X}) = 20 \log_{10} \frac{||\boldsymbol{Y}||_{\text{F}}}{||\boldsymbol{Y} - \boldsymbol{\Phi X}||_{\text{F}}}. \tag{5.19}$$

The three plots correspond to the three different dictionary initialisations. In particular, we note that when the dictionary is initialised with random examples from the training data, K-SVD achieves a good approximation quality of about SNR $\approx$ 24dB at the expense of a high mutual coherence $\mu \approx 0.95$. On the other hand, INK-SVD is able to achieve a lower coherence $\mu = 0.5$ while maintaining a SNR$> 20dB$ and, after this value, the approximation quality drops linearly with the mutual coherence. The iterative projection method (labelled as Grassmannian) achieves a correlation $\mu \approx 0.45$, but with a worst SNR $\approx 8dB$.

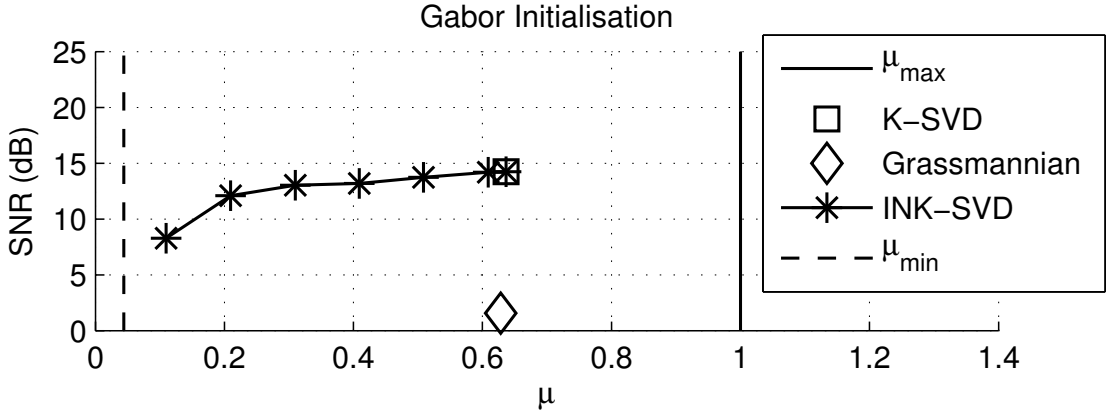
The other two plots corresponding to DCT and Gabor initialisations display overall a poorer approximation quality. In these cases, Grassmannian fails to significantly decorrelate the dictionaries and achieves a very poor SNR, while INK-SVD is able to decorrelate the dictionaries up to $\mu = 0.2$ with a small loss in approximation accuracy.



(a)



(b)



(c)

Figure 5.2: Signal to noise ratio as a function of the coherence value for different choices of dictionary initialisation and de-correlation functions. The levels $\mu_{max} = 1$ and $\mu_{min} = \sqrt{(K-N)/N(K-1)}$ indicate the maximum and minimum coherence attainable by a $N \times K$ dictionary.

5.4 Iterative projections and rotations algorithm

Both the iterative projections algorithm introduced in Section 5.2.1 and the INK-SVD algorithm described in Section 5.3.2 do not take into account the objective of dictionary learning, that is to approximate a set of training signals using a sparse model. The former simply aims at de-correlating a dictionary by lowering its mutual coherence to a fixed level μ_0, while the latter attempts to solve the optimization (5.11) whose objective is to converge to an incoherent dictionary that is close (in a Frobenoius norm) to the matrix returned by the dictionary update step of dictionary learning. This is motivated by the assumption that dictionaries that are closer to each other are supposed to be suitable to approximate a certain class of signals and, therefore, if solving (5.11) leads to a dictionary with lower mutual coherence that is close to $\tilde{\boldsymbol{\Phi}}$, then it will be similarly well adapted to approximate the signals in $\boldsymbol{Y}$.

A better strategy consists in including the dictionary learning objective (2.23) into the dictionary de-correlation optimization (5.10) by setting $\mathcal{C}(\boldsymbol{\Phi}) = ||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}}$, and overall seeking a solution to the following optimization problem:

$$\begin{aligned} \boldsymbol{\Phi}^{\star} = \underset{\boldsymbol{\Phi}\in\mathcal{D}}{\arg\min}\ & ||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} \\ \text{such that } & \mu(\boldsymbol{\Phi}) \leq \mu_0 \\ & ||\boldsymbol{x}_m||_0 \leq S \quad \forall m \end{aligned} \tag{5.20}$$

For this purpose, after performing a sparse approximation that satisfies the sparsity constraint and a dictionary update, we employ a dictionary de-correlation that is based on the iterative projections algorithm which consists of two steps:

I - *Dictionary de-correlation*: obtained through an iterative projection algorithm, this step ensures that the mutual coherence constraint is satisfied.

II - *Dictionary rotation*: this step optimizes the dictionary with respect to the objective function (5.20) without affecting its mutual coherence.

5.4.1 Dictionary rotation

The iterative projection algorithm can be used to de-correlate a dictionary starting from the matrix returned by the dictionary update step. However, we found that optimizing

the Gram matrix with the only objective being reducing the mutual coherence means that the decomposition (5.8) is likely to lead to an updated dictionary that exhibits a poor approximation performance, as shown by the numerical experiments presented in Section 5.3.3. To resolve this issue, we employ a dictionary rotation[2] which does not modify the mutual coherence and that is optimized for the dictionary learning objective (5.20).

The key observation to be made is that the decomposition (5.8) is not unique, since for any orthonormal matrix $\boldsymbol{W}$ such that $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$ we obtain:

$$(\boldsymbol{W}\boldsymbol{\Phi})^T(\boldsymbol{W}\boldsymbol{\Phi}) = \boldsymbol{\Phi}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{\Phi} = \boldsymbol{\Phi}^T\boldsymbol{\Phi} = \boldsymbol{G}.$$

Therefore, it is possible to apply an orthonormal matrix to the dictionary obtained from the iterative projection algorithm in order to minimise the residual norm expressed in (5.20). The resulting optimization problem can be expressed as follows:

$$\boldsymbol{W}^\star = \underset{\boldsymbol{W}\in\mathcal{O}(N)}{\arg\min}\ \frac{1}{2}\,||\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}}^2 \tag{5.21}$$

where $\mathcal{O}(N)$ is the set of $N \times N$ orthonormal matrices. After tackling the optimization (5.21) with a Lie group method detailed in Appendix D, I found that a closed-form solution to this problem can be traced back to an algorithm proposed by Horn et al. [48] to align sets of points measured in different coordinate systems for stereo photogrammetry and robotics applications.

Let us define $\tilde{\boldsymbol{Y}} \overset{\text{def}}{=} \boldsymbol{\Phi}\boldsymbol{X}$ as the matrix containing the sparse approximation of the observed data. The minimisation problem (5.21) can be expressed using the identity $||\boldsymbol{A}||_F^2 = \mathrm{Tr}(\boldsymbol{A}^T\boldsymbol{A})$ as:

$$\boldsymbol{W}^\star = \underset{\boldsymbol{W}\in\mathcal{O}(N)}{\arg\min}\ \mathrm{Tr}\left(\boldsymbol{Y}^T\boldsymbol{Y}\right) + \mathrm{Tr}\left(\tilde{\boldsymbol{Y}}^T\tilde{\boldsymbol{Y}}\right) - 2\,\mathrm{Tr}\left(\boldsymbol{Y}^T\boldsymbol{W}\tilde{\boldsymbol{Y}}\right).$$

Since the first two terms do not depend on $\boldsymbol{W}$ and since for every pair of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, $\mathrm{Tr}(\boldsymbol{A}\boldsymbol{B}) = \mathrm{Tr}(\boldsymbol{B}\boldsymbol{A})$, we can instead consider the maximisation problem:

$$\boldsymbol{W}^\star = \underset{\boldsymbol{W}\in\mathcal{O}(N)}{\arg\max}\ \mathrm{Tr}\left(\boldsymbol{W}\tilde{\boldsymbol{Y}}\boldsymbol{Y}^T\right). \tag{5.22}$$

[2] *Rotation* is from now on employed with an abuse of terminology, referring to any linear transformation obtained through an orthonormal matrix that include flips and rotations.

The notation $\boldsymbol{C} \stackrel{\text{def}}{=} \tilde{\boldsymbol{Y}}\boldsymbol{Y}^T$ indicates the sample covariance between the observed signals and their sparse approximations, which can be decomposed using an SVD as $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. The objective function in (5.22) can be written as:

$$\operatorname{Tr}\left(\boldsymbol{W}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right) = \operatorname{Tr}\left(\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{W}\boldsymbol{U}\right) = \operatorname{Tr}\left(\boldsymbol{\Sigma}\boldsymbol{Q}\right)$$

where the matrix $\boldsymbol{Q} \stackrel{\text{def}}{=} \boldsymbol{V}^T\boldsymbol{W}\boldsymbol{U}$ is orthonormal as it results from the product of three orthonormal matrices. Considering that $\boldsymbol{\Sigma}$ is diagonal, the following holds:

$$\operatorname{Tr}\left(\boldsymbol{\Sigma}\boldsymbol{Q}\right) = \sum_{n=1}^{N} \sigma_n q_{nn}.$$

The singular values σ_n are non-negative by definition, and the entries q_{nn} are upper-bounded by 1 because the norm of the vectors $\boldsymbol{q}_n$ is unitary. Therefore, the value $q_{nn}^\star = 1$ maximises the above equation, and implies $\boldsymbol{Q}^\star = \boldsymbol{I}$. This can be obtained by setting:

$$\boldsymbol{W}^\star = \boldsymbol{V}\boldsymbol{U}^T.$$

5.4.2 Optimisation algorithm

The whole dictionary decorrelation could be performed only once after dictionary learning, but we found in our numerical experiments that this strategy led to poor approximation results, as exposed in Section 5.5.5. Instead, we choose to rotate the dictionary at every step of the iterative projections that are performed after every dictionary update. This strategy leads to an algorithm that adapts the dictionary to the approximation objective (5.20) at each step of the de-correlation.

Considering the dictionary decorrelation alone, we initialise the algorithm with the dictionary $\boldsymbol{\Phi}^{(0)}$ returned by the update step of dictionary learning and perform at each iteration t the following steps summarised in Algorithm 12:

I - Compute the Gram matrix: $\boldsymbol{G}^{(t)} = {\boldsymbol{\Phi}^{(t)}}^T\boldsymbol{\Phi}^{(t)}$.

II - Calculate the projection onto the structural constraint set: $\boldsymbol{K}^{(t)} = \mathcal{P}_{\mathcal{K}_{\mu_0}}\left(\boldsymbol{G}^{(t)}\right)$.

III - Factorise $\boldsymbol{K}^{(t)}$ as in (5.8) including thresholding its eigenvalues. This returns an updated dictionary $\boldsymbol{\Phi}^{(t+1)}$ whose Gram matrix $\boldsymbol{G}^{(t+1)} = \mathcal{P}_{\mathcal{F}}(\boldsymbol{K}^{(t)})$ is projected onto

the spectral constraint set.

IV - Rotate the dictionary using an optimal orthonormal transform by updating $\boldsymbol{\Phi}^{(t+1)} = \boldsymbol{W}^{\star}\boldsymbol{\Phi}^{(t+1)}$.

Note that the rotation step does not modify the Gram matrix of the dictionary because this does not change the pair-wise correlations between atoms, and therefore is irrelevant for the purpose of the convergence of the iterative projections algorithm to a dictionary with bounded coherence. The convergence analysis of the general dictionary learning optimization described by (5.20) is very difficult and is outside the scope of the present work. The interested reader can find insights on related problems by reading the work of Aaron et al. [5], Gribonval and Schnass [46] or Mailhé and Plumbley [64].

Nonetheless, it is worth highlighting the fact that the rotation step finds the optimal solution of the problem (5.21), and therefore is guaranteed to improve (or leave unchanged) the cost function (5.20) without violating its constraints set. This is sufficient to say that adding a rotation step to the dictionary de-correlation algorithm improves the approximation quality of dictionary learning if compared to the iterative projections algorithm alone.

The IPR algorithm includes the calculation of the optimal rotation matrix described in 5.4.1 which replaces our early formulation based on a Lie group method. As well as offering a closed-form solution to a problem that was previously tackled with an iterative method, this substantially improved the computational time required by the algorithm and allowed for a simpler analysis of its complexity.

Since $M \geq K \geq N$, the running time of the algorithm per iteration is dominated (in order) by the following steps:

- Computation of the EVD of the Gram matrix $\boldsymbol{G}$ requiring $\mathcal{O}(K^3)$ operations.
- Computation of the covariance matrix $\boldsymbol{C}$ requiring $\mathcal{O}(N^2M)$ operations.
- Computation of the SVD of the covariance matrix $\boldsymbol{C}$ requiring $\mathcal{O}(N^3)$ operations.

In the numerical experiments presented in Section 5.5, we observed that these three operations accounted for around 90% of the computational time required by every iteration of the IPR algorithm, whose order of magnitude is comparable to the one relative to the time required by running a dictionary update step using K-SVD or MOD.

Algorithm 12: Iterative projections and rotations (IPR)

Input: $\boldsymbol{Y}, \boldsymbol{\Phi}, \boldsymbol{X}, \mu_0, I$
Output: $\boldsymbol{\Phi}^{\star}$
// Initialisation
$i \leftarrow 1$;
while $i \leq I$ *and* $\mu(\boldsymbol{\Phi}) > \mu_0$ **do**
  // Calculate Gram matrix
  $\boldsymbol{G} \leftarrow \boldsymbol{\Phi}^T\boldsymbol{\Phi}$;
  // Project ont structural constraint set
  $\mathrm{diag}(\boldsymbol{G}) \leftarrow \mathbf{1}$;
  $\boldsymbol{G} \leftarrow \mathrm{Limit}(\boldsymbol{G}, \mu_0)$;
  // Factorise Gram matrix and project onto spectral constraint set
  $[\boldsymbol{Q}, \boldsymbol{\Lambda}] \leftarrow \mathrm{EVD}(\boldsymbol{G})$;
  $\boldsymbol{\Lambda} \leftarrow \mathrm{Thresh}(\boldsymbol{\Lambda}, N)$;
  $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Lambda}^{1/2}\boldsymbol{Q}^T$;
  // Rotate dictionary
  $\boldsymbol{C} \leftarrow \boldsymbol{Y}(\boldsymbol{\Phi}\boldsymbol{X})^T$;
  $[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] \leftarrow \mathrm{SVD}(\boldsymbol{C})$;
  $\boldsymbol{W} \leftarrow \boldsymbol{V}\boldsymbol{U}^T$;
  $\boldsymbol{\Phi} \leftarrow \boldsymbol{W}\boldsymbol{\Phi}$;
  $i \leftarrow i + 1$;
end

5.5 Numerical Experiments

We tested the proposed IPR decorrelation method with the K-SVD dictionary learning algorithm in order to assess if it converges to a dictionary that exhibits bounded mutual coherence and good approximation quality. The test signal we used is the same employed in the experiments described in Section 5.3.3.

We divided the recording into 50% overlapping blocks of 256 samples (corresponding to 16ms) with rectangular windows and arranged the resulting time-domain signals as columns of the training data matrix $\boldsymbol{Y}$. Then, we initialised a twice over-complete dictionary for sparse approximation using either a randomly chosen subset of the training data or an over-complete Gabor dictionary. We run the dictionary learning algorithms for 50 iterations, allowing for $S = 12$ non-zero coefficients in each representation (which corresponds to about 5% of active elements if compared with the dimension of the audio frames N). When testing the algorithm proposed in [87], we used OMP as a sparse approximation step setting the stopping criterion to the maximum number of active atoms S and MOCOD for the dictionary update. INK-SVD and IPR were implemented using OMP for the sparse approximation step and K-SVD for the dictionary update. Table 5.1 summarises

Algorithm (Reference)	Sparse Approximation	Dictionary Update	Dictionary Decorrelation
Sapiro et al. [87]	OMP	MOCOD	-
Mailhé et al. [62]	OMP	K-SVD	INK-SVD
Proposed method	OMP	K-SVD	IPR

Table 5.1: Algorithms for learning incoherent dictionaries

the tested algorithms.

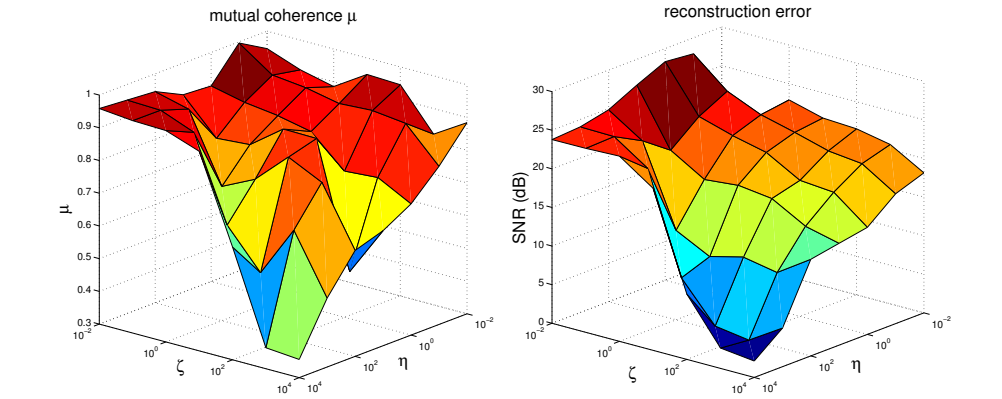
5.5.1 MOCOD updates

The unconstrained optimization illustrated in (5.9) relies on the penalty factors ζ and η in order to promote incoherence of the dictionary and unit norm of the atoms respectively. To evaluate the MOCOD dictionary update for the purpose of incoherent dictionary learning, we tested different values of these factors on a logarithmic scale between 10^{-2} and 10^{4}, assessing the resulting mutual coherence and signal-to-noise ratio (SNR) achieved by the optimized dictionary, the latter being defined as:

$$\text{SNR}(\boldsymbol{Y}, \boldsymbol{\Phi X}) = 20 \log_{10} \frac{||\boldsymbol{Y}||_{\text{F}}}{||\boldsymbol{Y} - \boldsymbol{\Phi X}||_{\text{F}}}.$$

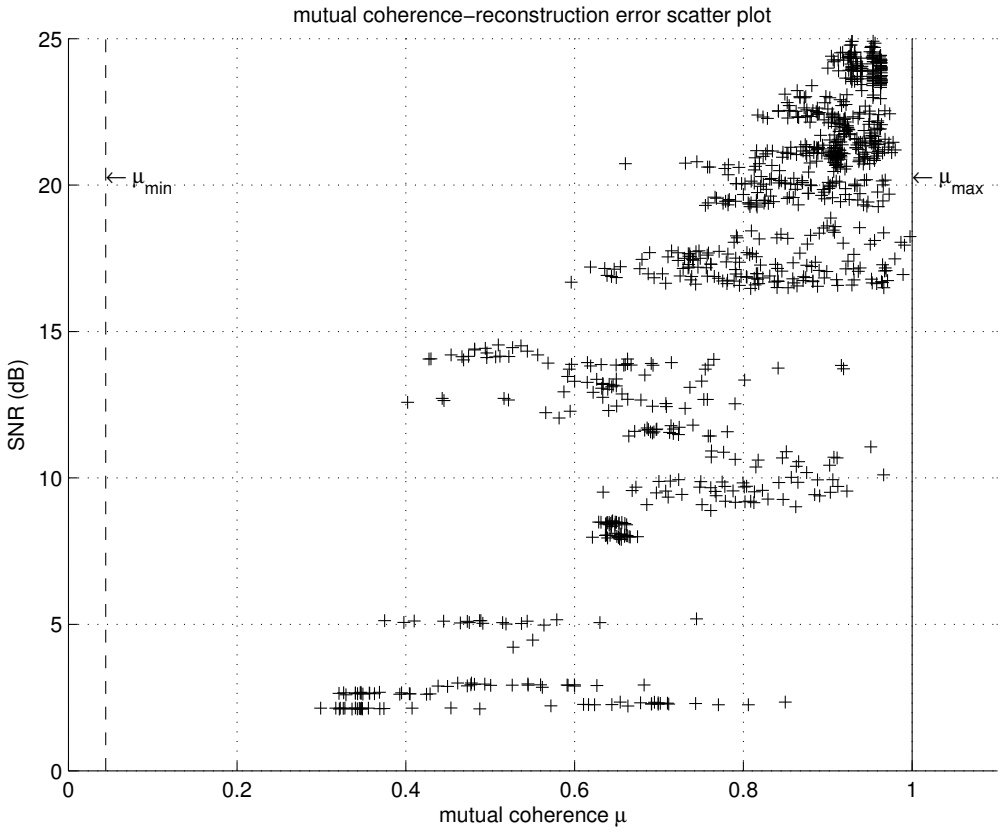
Figure 5.3 depicts the results of our experiment using respectively randomly chosen data from the training set (which is the default initialisation of the original implementation of the K-SVD algorithm) and a twice over-complete Gabor dictionary for the initialisation. We run the experiment 20 times to increase the significance of our results whenever the initialisation involved choosing a random subset of the training data as the initial dictionary.

When $\zeta \to 0$ and $\eta \to \infty$, the optimization (5.9) converges to a standard dictionary learning where the atoms are not forced to be incoherent, but are constrained to have unit norm. This case corresponds to the left corner of the surface plots in Figure 5.3. We can note that a *data* initialisation produces a highly coherent dictionary with the best approximation quality, while a *Gabor* initialisation results in a lower coherence at the expense of a worse SNR. Continuing our analysis in the case of data initialisation, keeping $\eta \to \infty$ and increasing the coherence penalty factor ζ results in a dictionary with lower mutual coherence, but also in a worse approximation quality. This behaviour is further illustrated by the mutual coherence-reconstruction scatter plot, which depicts μ



(a)

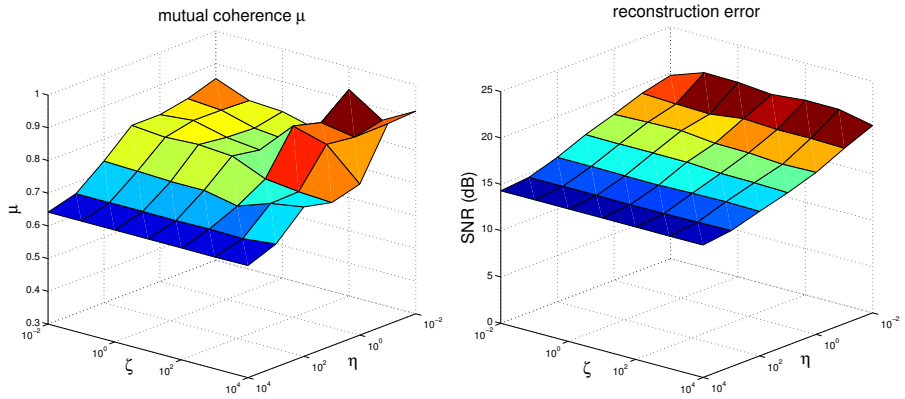
(b)



(c)

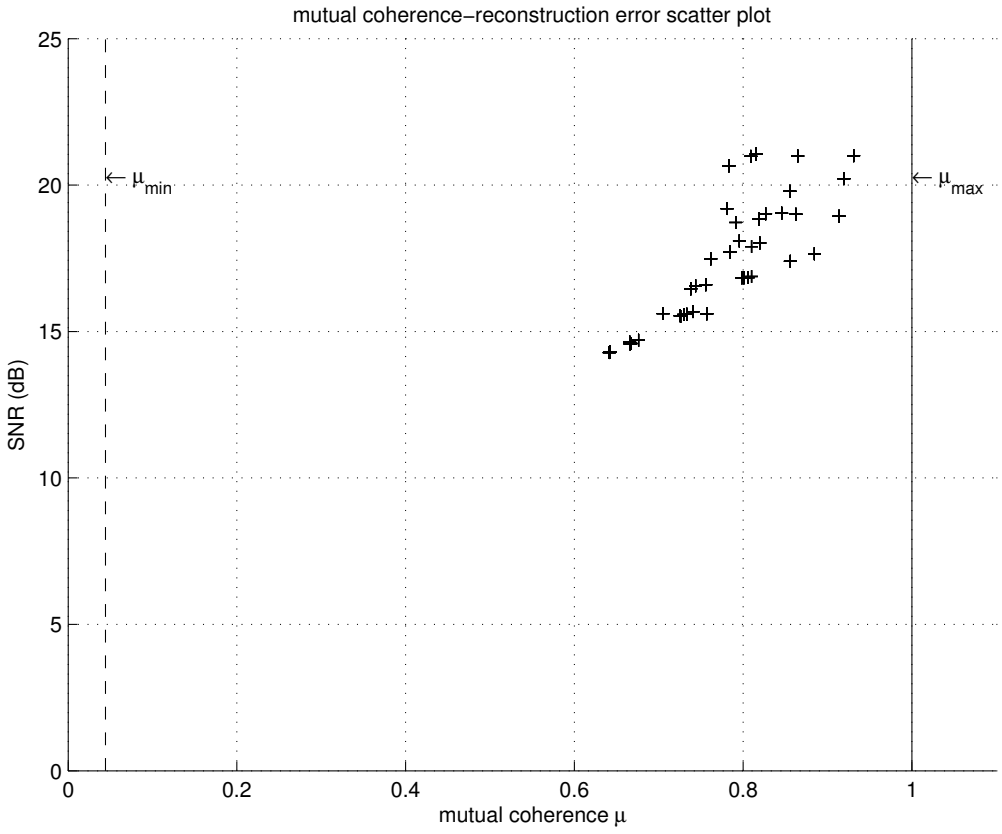
Figure 5.3: mutual coherence and reconstruction error achieved using the MOCOD dictionary update and randomly chosen samples from the training set as the initial dictionary. The surf plots show the mutual coherence and SNR of the sparse approximation as a function of the two regularisation parameters η and ζ in equation (5.9). In the scatter plots, the points correspond to dictionaries obtained at different trials and with different values of the parameters η and ζ. The levels $\mu_{\max} = 1$ and $\mu_{\min} = \sqrt{(K-N)/N(K-1)}$ indicate the maximum and minimum coherence attainable by a $N \times K$ dictionary.

against SNR of the sparse approximation for every learned dictionary and exhibits a clear (although highly variable) trend. In the case of Gabor initialisation, on the other hand,



(d)

(e)



(f)

Figure 5.3: (continued) mutual coherence and reconstruction error achieved using the MOCOD dictionary update and a Gabor frame as the initial dictionary. The surf plots show the mutual coherence and SNR of the sparse approximation as a function of the two regularisation parameters η and ζ in equation (5.9). In the scatter plots, the points correspond to dictionaries obtained at different trials and with different values of the parameters η and ζ. The levels $\mu_{\max} = 1$ and $\mu_{\min} = \sqrt{(K-N)/N(K-1)}$ indicate the maximum and minimum coherence attainable by a $N \times K$ dictionary.

it seems that the parameter ζ does not affect mutual coherence and reconstruction error for high values of η, while decreasing the penalty factor η has generally a negative effect

on both μ and SNR of the learned dictionaries.

To understand the poor performance of the MOCOD algorithm, especially when initialised with a Gabor dictionary, we inspected μ and SNR of the sparse approximation at every iteration, along with the percentage change of the dictionary with respect to the Frobenious norm, that is defined as:

$$100 \frac{\left|\left|\mathbf{\Phi}^{(t+1)} - \mathbf{\Phi}^{(t)}\right|\right|_{\mathrm{F}}}{\left|\left|\mathbf{\Phi}^{(t)}\right|\right|_{\mathrm{F}}} \tag{5.23}$$

were $\mathbf{\Phi}^{(t)}$ indicates the dictionary at iteration t.

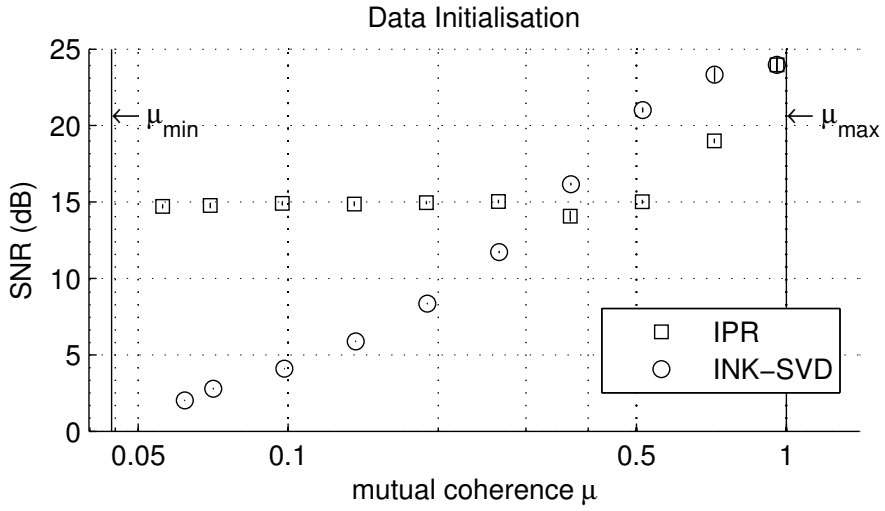
The main observation that underlies the poor performance of MOCOD is that the percentage change of the dictionary does not converge to zero and, therefore, the algorithm does not converge to a fixed point of the objective function (5.9). Whenever η is small (that is, when the dictionary atoms are not forced to be unit norm), the optimization is very unstable and we often observed that the mutual coherence ends being greater than the one of the initial dictionary, especially for low values of ζ.

When η is large, the algorithm still does not converge to a fixed point of the objective function, but the mutual coherence and SNR are much more stable. In this case different initialisations lead to different behaviours: in the case of *data* initialisation, the mutual coherence drops and the SNR oscillates, while in the case of *Gabor* initialisation, the SNR does not change significantly and the mutual coherence slightly increases. Moreover, the minimum mutual coherence achieved by MOCOD in the results shown is never smaller than $\mu = 0.3$, and further experiments with penalisation terms $\eta = \zeta = 10^{10}$ confirmed that the algorithm is unable to reach lower mutual coherence levels.

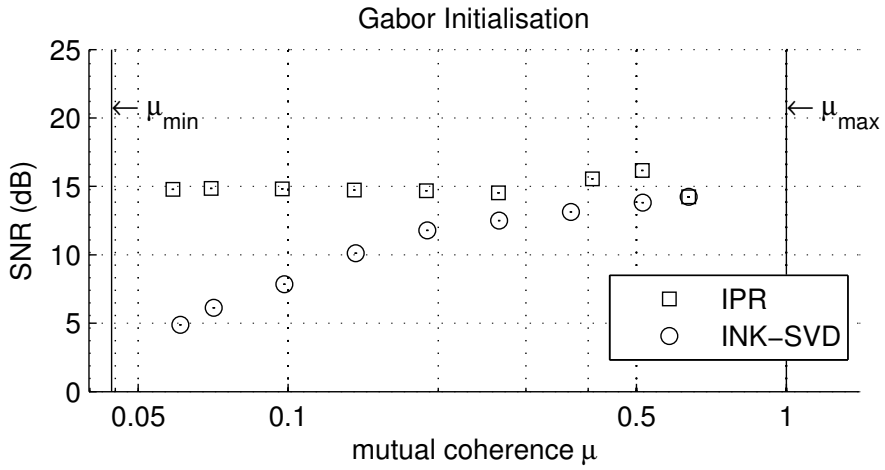
Unlike MOCOD, INK-SVD and the proposed IPR algorithm allow us to set a target coherence μ_0 and to run the dictionary decorrelation iteratively until it is achieved.

5.5.2 IPR and INK-SVD

After experimenting with different combinations of dictionary learning and decorrelation iteration numbers, we found that consistently good results can be achieved by performing 50 iterations of the K-SVD dictionary learning combined with 5 iterations of the relevant decorrelation method. This also led to comparable running times, as will be discussed



(a)



(b)

Figure 5.4: Mutual coherence and reconstruction error achieved using the proposed iterative projections and rotations (IPR) algorithm and INK-SVD dictionary decorrelation, initialised with (a) randomly chosen samples from the training set as the initial dictionary or (b) a twice over-complete Gabor dictionary. The error bars in (a) represent the standard deviation resulting from 10 independent trials of the experiment and indicate that the results are consistent, regardless the random element introduced in the initialisation.

in Section 5.5.3. We set the target mutual coherence in logarithmically spaced intervals from $\mu = 0.05$ to $\mu = 1$ and compared the two algorithms by evaluating the achieved SNR. When applying the methods to an initial dictionary formed by randomly selected vectors from the training set, we run the experiment for 10 independent trials to obtain more significant results.

Figure 5.4 depicts the results of our experiment. As can be noted, both algorithms succeed in matching the target coherence levels for both initialisations except for the

lower end on the left side of the plots, with IPR performing slightly better in achieving the smallest mutual coherence in the case of data initialisation, reaching a value of around 0.055 compared to the 0.06 of INK-SVD. Whenever the target coherence μ_0 is bigger than the coherence level achieved without dictionary decorrelation, the two methods simply act as a K-SVD without any mutual coherence constraint. In the case of data initialisation, we can observe that INK-SVD obtains a good SNR for mutual coherence values greater than $\mu = 0.3$, after that its performance degrades substantially. On the contrary, the proposed IPR does not perform as well for high coherence values, but does not significantly degrade from $\mu = 0.3$ to $\mu = 0.05$.

The results for Gabor initialisation, on the other hand, favour the proposed algorithm showing a better SNR and no significant approximation degradation for all the target coherence values.

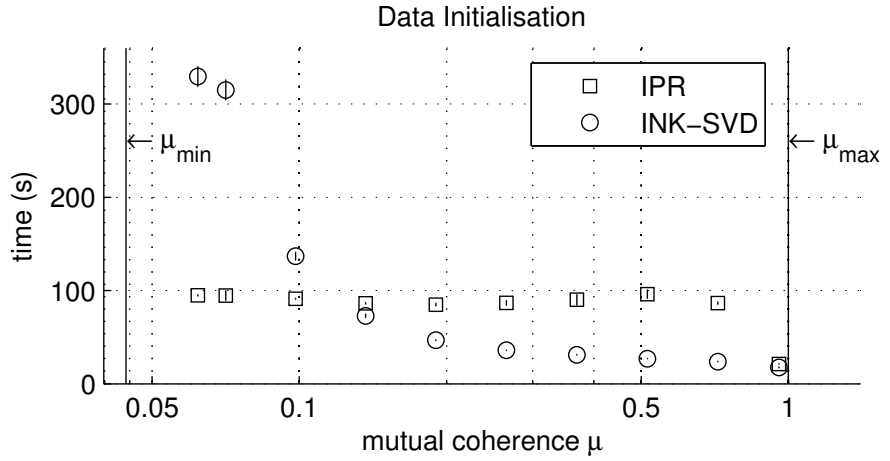
5.5.3 Running times

Figure 5.5 shows the running times of the IPR and INK-SVD algorithms for different coherence levels, tested on a iMac with a 3.06GHz Intel Core 2 Duo processor running MATLAB R2011a and the `cputime` function. The IPR values are not dependant on the coherence level and are just below 100 seconds, whereas INK-SVD takes longer to compute less coherent dictionaries. This is because INK-SVD acts in a greedy fashion by decorrelating pair of atoms until the target mutual coherence is reached (or until a maximum number of iterations) and therefore the number of pairs of atoms to decorrelate increases for low values of the target coherence.

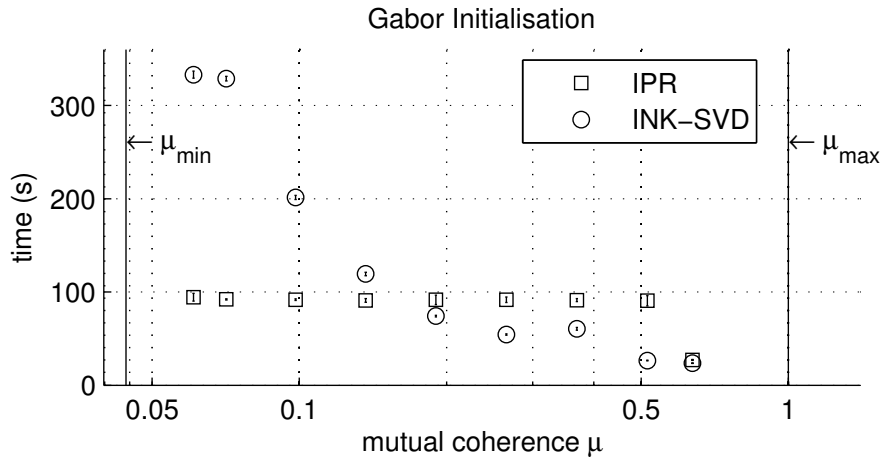
The time required to compute a non de-correlated dictionary can be found in the right end of the plots and is around 20 seconds, which is also consistent with the average time of 23 seconds needed by the MOCOD algorithm. This means that the cost of IPR is about 5 times the cost of a standard K-SVD for the problem sizes considered in our experiments.

5.5.4 Sparse approximation results

The relation between the coherence of a dictionary and its approximation properties for different classes of signals is a complex topic. In this Section a formal convergence analysis of the tested dictionary learning algorithms is not attempted as this is outside the scope of the present work. However, I will present some experimental results which suggest



(a)



(b)

Figure 5.5: Running times of IPR and INK-SVD for different mutual coherence levels and dictionaries initialised with (a) randomly chosen samples from the training set or (b) a twice over-complete Gabor dictionary. The error bars indicate the standard deviation resulting from 10 independent trials of the experiments.

that incoherent dictionaries are indeed useful for sparse approximation, also pointing the interested reader to other related work that might fuel further research in this area.

The trade-off between mutual coherence and SNR of the sparse approximation visible in Figures 5.3d, 5.4a and 5.4b is consistent with the fact that the different decorrelation methods aim at solving penalised or constrained optimization problems. If we compare the general dictionary learning problem introduced in Section 2.6 to the incoherent formulations presented in this thesis, the penalty factors used to promote incoherence in the unconstrained optimization (5.9) and the feasible set consisting of dictionaries with bounded mutual coherence in the constrained problem (5.20) suggest that an incoher-

ent dictionary is expected to have a worse approximation performance if compared to a coherent one. On the other hand, dictionary learning is a non-convex optimization problem that to the best of our knowledge lacks strong and general convergence results, relying instead on the ability of practical algorithms to converge to local minima of the optimization cost function.

Additional assumptions regarding the intrinsic properties of a learned dictionary can be promoted through penalised or constrained problems in order to *steer* the optimization towards a local minimum. This is the approach followed by Dai et al. [20], who devised a penalised optimization to learn dictionaries where the condition number of groups of atoms employed in the sparse approximation of the signals in the training set is low. In cases where a standard K-SVD would converge to a dictionary with ill-posed sub-dictionaries, the authors documented the superior performance of their proposed method for the sparse approximation of the signals in the training set. A mutually incoherent dictionary learned through the IPR algorithm is designed to approach the spectral properties of an orthonormal transform, and therefore it is reasonable to expect the condition number of its sub-dictionaries to be low. However, a more thorough investigation is necessary to fully support this claim.

For the purpose of the experimental evaluation of the IPR algorithm, we tested whether the mutual coherence versus SNR trade-off is consistent over different training and testing signals. We considered the following test material:

- `music03_16kHz`, a 5 seconds guitar recording distributed as part of the SMALLBOX that was used to train the dictionaries in the experiments presented so far.

- `track n.6` of the jazz section of the RWC music database[3], which is a 30 seconds electric guitar recording.

- `track n.1` of the jazz section of the RWC music database, which is a 30 seconds acoustic piano recording.

After running the IPR dictionary learning algorithm on the guitar recording `track n.6` using the *data* initialisation, the same problem parameters specified in Section 5.5 and the target mutual coherence levels specified in Section 5.5.2, we employed the learned

[3]available at http://staff.aist.go.jp/m.goto/RWC-MDB/

dictionaries to approximate the two remaining test signals, using the OMP algorithm and 5% of active atoms, as in the learning phase.

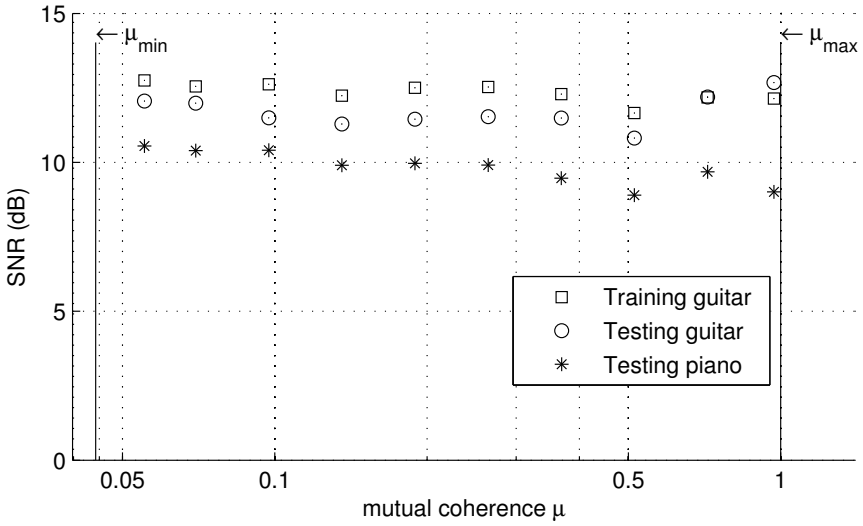


Figure 5.6: Mutual coherence versus SNR of the sparse approximation using a dictionary learned from `track n.6` of the jazz section of the RWC database using data initialisation, OMP and 5% of active atoms in the sparse coding step of dictionary learning. In the testing phase, OMP with 5% of active atoms was also used to approximate signals from the training set, from `music03_16kHz` that is a different guitar recording and from `track n.1` of the jazz section of the RWC database that is a piano recording.

Figure 5.6 displays the results of the experiment. If we compare these values to the ones presented in Figure 5.4a, we can note that the trade-off between mutual coherence and SNR is no longer present, and that the approximation of the training set (which in the case of the training guitar is inversely proportional to the residual norm in the cost function (5.20)) is around 12 and 13 dB for the two guitar signals and around 10 dB for the piano signal. The absence of a steep peak in correspondence with a dictionary with high mutual coherence and the overall worse approximation performance can be explained by the fact that `music03_16kHz` is a relatively short signal (5 seconds), that as a consequence when learning a dictionary from this signal the number of training vectors compared to the size of the dictionary is relatively small and that we observed a few signals that could be approximated very well using only one atom in the dictionary. This does not happen when learning a dictionary from a longer training set obtained using `track n.6` (a 30 seconds signals) and results in overall worse but more consistent results.

Figure 5.6 shows that essentially a dictionary with a low mutual coherence is as good as a coherent dictionary when used to approximate the training set and the guitar recording

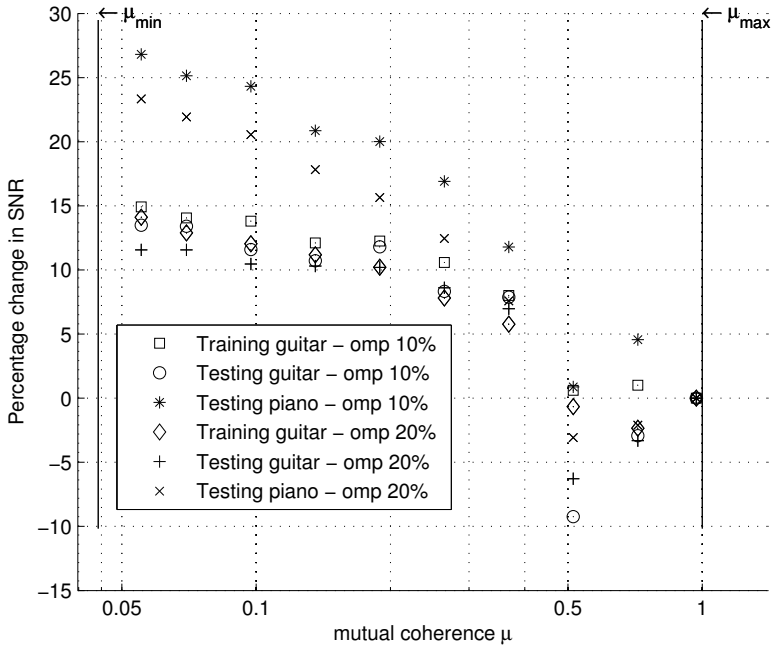
used as one of the testing sets. There is however a slight improvement in the SNR obtained when approximating the piano signal, which means that in this case a dictionary with a low mutual coherence better generalises its approximation capabilities to a different class of signals.

This insight is further confirmed by the data presented in Figure 5.7, which depict the percentage change in the SNR of the sparse approximation between dictionaries with different mutual coherences and dictionaries returned by the baseline K-SVD algorithm that does not enforce any mutual coherence constraint. The percentage improvement between a baseline value b and a test data b' is defined as $100\frac{b'-b}{b}$. In the case considered here, let $\text{SNR}\left(\boldsymbol{Y}, \boldsymbol{\Phi}_{\text{K-SVD}}\boldsymbol{X}_{\text{K-SVD}}\right)$ be the signal to noise ratio resulting form the dictionary and coefficients matrix returned by the K-SVD algorithm. The values shown in Figure 5.7 are defined as

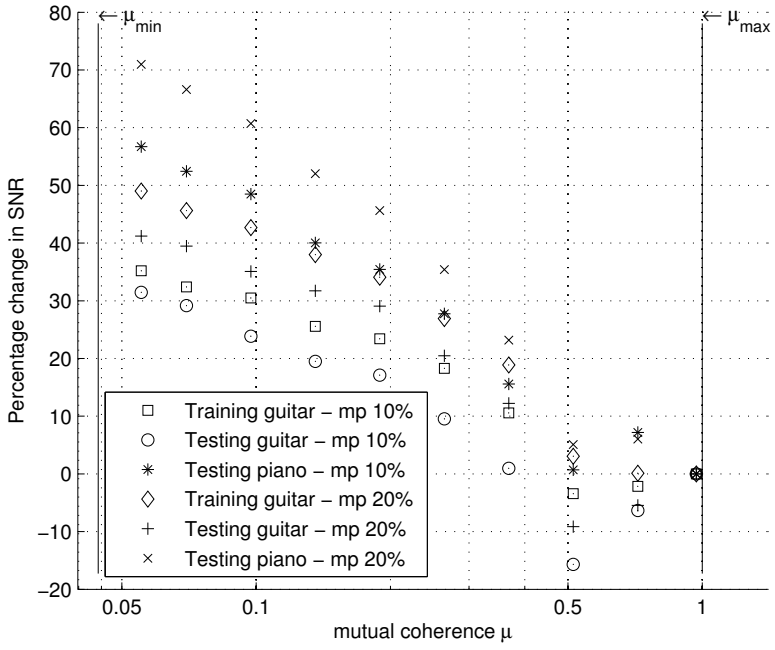
$$100\frac{\text{SNR}\left(\boldsymbol{Y}, \boldsymbol{\Phi}\boldsymbol{X}\right) - \text{SNR}\left(\boldsymbol{Y}, \boldsymbol{\Phi}_{\text{K-SVD}}\boldsymbol{X}_{\text{K-SVD}}\right)}{\text{SNR}\left(\boldsymbol{Y}, \boldsymbol{\Phi}_{\text{K-SVD}}\boldsymbol{X}_{\text{K-SVD}}\right)} \tag{5.24}$$

and represent the relative improvement of the SNR of the sparse approximation that can be achieved by using incoherent dictionaries relative to the baseline. This is in turn a function of the approximation error $||\boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{X}||_{\text{F}}$ that reaches infinity in the case of perfect reconstruction and monotonically decreases dropping below zero when the approximation error resulting from incoherent dictionaries exceeds the baseline error resulting from the K-SVD algorithm. This measure was chosen instead of a simple comparison of approximation errors because being based on SNR values it is independent from the norm of the signal to be analysed. Moreover, by comparing different values of SNR, it is independent from the absolute value of the baseline SNR.

Figure 5.7 reveals that, when learning an incoherent dictionary using a given number of active elements during the sparse coding step, a substantial improvement can be obtained over coherent dictionaries when approximating signals using a larger number of active atoms (in this case either 10% or 20% compared to the dimension of the training signals N). This improvement in the lower end of mutual coherence values ranged from 10% to 27% in Figure 5.7(a) depending on the signal to be approximated. It is more significative in the case corresponding to the approximation of piano signals through a dictionary learned on guitar signals, suggesting that the mutual coherence constraints improves the generalisation of the approximation performance.



(a)



(b)

Figure 5.7: Mutual coherence versus percentage change in the SNR of the sparse approximation using a dictionary learned from `track n.6` of the jazz section of the RWC database using OMP and 5% of active atoms in the sparse coding step of dictionary learning. In the testing phase, (a) OMP or (b) MP with 10% or 20% of active atoms were used to approximate signals from the training set, from `music03_16kHz` that is a different guitar recording and from `track n.1` of the jazz section of the RWC database that is a piano recording.

Moreover, consistent and more significant results can be obtained if the sparse approximation is performed using the MP algorithm, as shown in Figure 5.7(b). In this case, the percentage improvement over a baseline coherent dictionary reached values over 70%. It is worth noting that in this last example, the labels MP 10% and MP 20% refer to the iterations of the sparse approximation algorithm that do not necessarily coincide to the number of active atoms employed in the approximation given that MP allows for atoms to be selected multiple times.

5.5.5 Additional experiments

IPR *as a post-processing step*

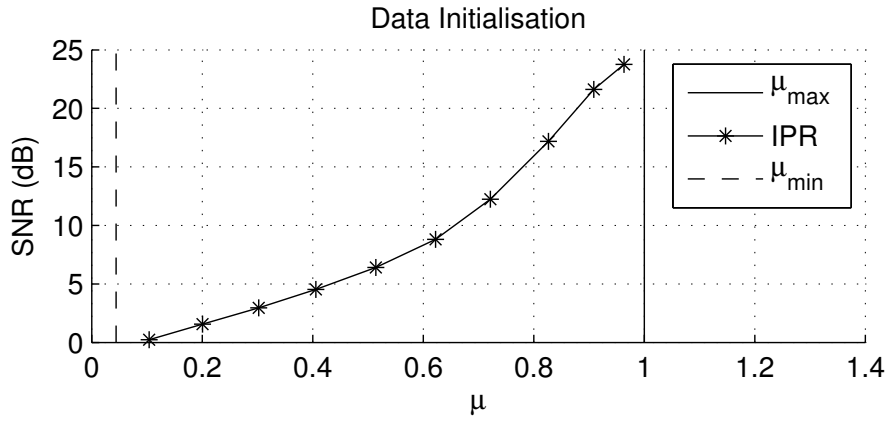
To analyse wether the dictionary decorrelation could be performed only once after running an unconstrained dictionary learning algorithm, IPR was tested after running K-SVD as a post-processing step. The experiment parameters were the same as those described in Section 5.5. The following two strategies were employed:

- Perform the de-correlation only once, starting from the dictionary learned by K-SVD and setting the coherence parameter in linearly spaced intervals of 0.1 increment in the range $\mu = [0.1, 1]$.
- Perform the de-correlation iteratively reducing the mutual coherence from the value of the learned dictionary to $\mu = 0.1$ in steps of 0.1, starting each de-correlation with the dictionary returned by the previous step.

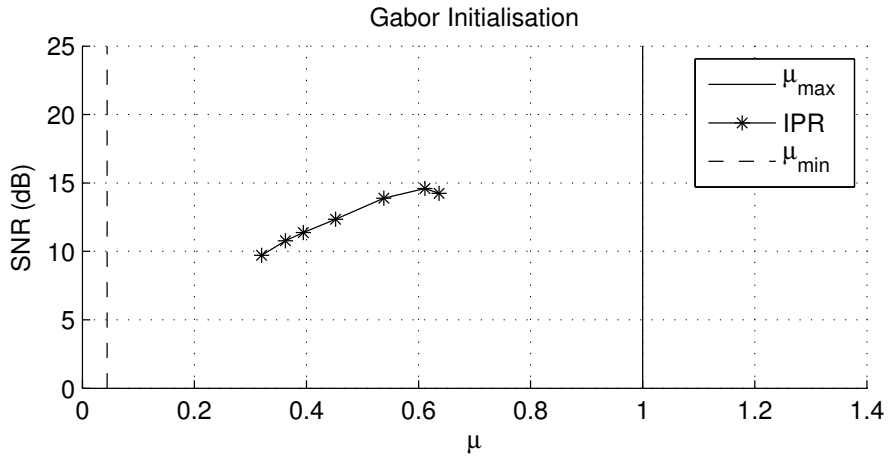
Figure 5.8 depicts the results obtained with the latter strategy which led to a slightly better outcome compared to the former. The results, however, are far from the ones achieved by including the de-correlation within the dictionary learning algorithm. The lower graph also shows that this approach is unable to reach the target coherence levels in the case of Gabor initialisation.

5.6 Summary and topics for further research

In this chapter we introduced the incoherent dictionary learning problem that consists in learning a dictionary that is both well adapted to a set of training signals and mutually incoherent. The motivation of learning incoherent dictionaries comes in part from theoretical results that show how a low mutual coherence is a sufficient condition for proving



(a)



(b)

Figure 5.8: SNR of the sparse approximation obtained by using IPR as a post-processing step after learning a coherent dictionary with K-SVD using (a) data and (b) Gabor dictionary initialisation.

the success of sparse recovery algorithms, in part from experimental results regarding sparse approximations and in part from intuitions about several application areas where incoherent dictionaries might prove a superior performance.

The iterative projection algorithm and the method of coherence constrained directions have been detailed as previous attempt to design or learn incoherent dictionaries. A dictionary de-correlation step has been proposed as an additional sub-problem of dictionary learning after the dictionary update stage, and two novel algorithms, namely the incoherent K-SVD and the iterative projections and rotations method, have been introduced to solve it.

The mixed objective of low mutual coherence and high approximation quality achieved by the learned dictionaries has been assessed through numerical experiments on audio sig-

nals. Unlike IP and MOCOD, the proposed IN-KSVD and IPR proved to be able to achieve very low mutual coherence levels, close to the lower bound that has been derived in the context of frame theory. IPR in particular also achieved a consistently high SNR of the sparse approximation regardless the level of mutual coherence, and was found to be faster than INK-SVD, making its performance superior overall. Additional numerical experiments on different audio signals demonstrated how incoherent dictionaries are desirable for sparse approximation whenever the number of active atoms used in the testing phase is larger than the number of active atoms employed during the learning phase. This suggests better generalisation properties of incoherent dictionaries.

Exploring the applications of the proposed work is one of the main objectives for future investigation. Incoherent dictionary learning can be applied to coding technologies, both for audio and for other types of signals that are amenable to sparse approximations. In this context, being able to control the mutual coherence of a dictionary can be used as a proxy for setting the generalisation capability of the atoms. Incoherent dictionaries can be thus employed whenever the training signals available for learning are a subset of a class of functions expected to exhibit a larger variance or to span a larger subspace.

Supervised problems can also benefit from the proposed algorithm in the context of dictionary learning for classification or morphological component analysis. IPR acts on the Gram matrix of the dictionary by thresholding the correlation between different atoms. This approach could be adapted to de-correlate only certain subsets of the dictionary that correspond to different morphological components or sources so to have sub-dictionaries that are constrained to have a low cross-coherence.

I have been recently awarded funding for a research proposal that includes investigating the incoherent dictionary learning in the context of audio scene classification. This project concerns the analysis of non-musical non-speech audio sources. It is aimed at designing an automatic tagging system that categorizes different events based on the sounds they produce (e.g., identifying different sports programs from their audio track or different acoustics scenes and events as proposed in the IEEE AASP challenge[4]). For this purpose, foreground and background audio sources (such as the sound of a racket hitting a ball and the background chattering of the audience in a tennis match), can be

[4]http://www.elec.qmul.ac.uk/digitalmusic/sceneseventschallenge/

separated and independently input to a machine learning algorithm responsible for the classification. The separation can be performed by assuming that the spectral representations of foreground and background sounds span incoherent sub-spaces, or consists in morphologically diverse components.

To conclude on the scope of future research regarding incoherent dictionary learning, an interesting topic for investigation consists in extending the de-correlation strategy to more accurate measures of coherence, such as the cumulative coherence proposed by Tropp [111]. This should also be complemented by a more accurate theoretical understanding of the interplay between coherence and approximation performance.

Chapter 6

Conclusions

6.1 Summary of main contributions

This thesis dealt with dictionary learning for sparse approximation, orthonormal and over-complete transforms and their application to the analysis of audio signals. The contributions of the work presented here can be included in three main areas corresponding to the topics covered in Chapters 3, 4 and 5.

Starting from Chapter 3 that studies sparsity and disjointness of audio transforms, the principal conclusions to be drawn from the work presented are as follows.

- The MDCT is overall the best choice among a range of popular transforms currently used for the analysis of audio signals.
  - It provides a *sparse* representation of audio signals that makes it suitable for coding applications, outperforming other LOTs for this task.
  - It provides a *disjoint* representation of pairs of musical audio signals making it suitable for source separation applications. It outperforms other LOTs and the CQT for this task.
  - It leads to better results compared to the pitch-synchronous STFT that is a novel adaptive LOT proposed for analysing quasi-periodic functions such as pitched musical audio signals.

Further research would be needed to better study the merits of a pitch-synchronous

MDCT and to investigate the relationship between sparsity and disjointness of a transform, two quantities that do not appear to be correlated as commonly assumed.

From the study of adaptive *orthonormal* transforms presented in Chapter 3, the reminder of the thesis is dedicated to the investigation of *over-complete* transforms in the context of dictionary learning for sparse approximation. Chapter 4 focuses on the sparse approximation of convolved signals.

- When a group of signals are sparse in a known dictionary and are convolved to an impulse response generating a set of convolved signals, it is more efficient to learn the convolution filter than a new dictionary for sparse approximation.

  - A novel model and algorithm for dictionary learning of convolved signals is formulated. The proposed technique learns an impulse response instead of an entire dictionary and allows to approximate the observed variables with a smaller error compared to the K-SVD dictionary learning algorithm.

  - In the analysis of audio signals, it is relevant to consider an impulse response constrained to be sparse and non-negative. In this case, the corresponding constrained optimisation S$\boldsymbol{h}$-BCD is outperformed by the unconstrained D$\boldsymbol{h}$-BCD. This suggests that optimising the objective in this situation requires a tradeoff between allowing for too many degrees of freedom, as in the case of K-SVD, and allowing for too few, as in S$\boldsymbol{h}$-BCD.

  - D$\boldsymbol{h}$-BCD outperforms K-SVD whenever the source signals are *sparse* (that is, synthesised from a number of atoms that is smaller than 10% compared to the dimension of the signals). This conclusion was obtained by testing the proposed method and K-SVD for different levels of normalized diversity of the source signals and of the impulse response.

  Additional research should be carried out to assess the performance of D$\boldsymbol{h}$-BCD when applied to real-world signals rather than synthetic ones.

From the analysis of convolved signals presented in Chapter 4, Chapter 5 is dedicated to intrinsic properties of dictionaries and to how they are relevant in applications.

- Dictionaries for sparse approximation that are well adapted to a set of training signals can be constrained to be also mutually incoherent. This property is important for sparse recovery and is desirable for sparse approximation.

  - Two novel algorithms, the INK-SVD and the IPR are presented and compared to existing methods for incoherent dictionary learning. The two proposed techniques are the only ones that allow to achieve mutual coherence levels close to the theoretical lower-bound.

  - The IPR in particular allows to learn dictionaries with very low mutual coherence without significantly affecting their approximation performance. IPR is also computationally less expensive than INK-SVD making it the best performing method overall.

  - Experiments suggest that mutually incoherent dictionaries can achieve better generalisation compared to coherent ones when more active atoms are used during the testing phase compared to the learning phase.

  Additional investigation is needed to better understand the relation between mutual coherence and approximation capabilities of a dictionary, as well as to explore additional applications of incoherent dictionaries.

6.2 Back to the big picture

Signal transforms have been introduced in Section 1.1 as a way to extract meaningful information from data. They are used to infer properties and realize processes that are useful in applications, helping to make sense of and leverage on the huge amount of data that is produced nowadays.

Throughout this thesis, various signal models have been introduced that typically decompose observed signals into elementary building blocks. The pitch-synchronous LOT presented in Chapter 3, for example, expresses signals as combinations of basis functions that are localised in both time and frequency, providing a sparse representation of the data. The model for dictionary learning of convolved signals introduced in Chapter 4 decomposes a set of convolved observations into the product of an impulse response matrix, a dictionary and a set of coefficients. The incoherent dictionary learning, on the other

hand, relies on a model comprising only dictionary and approximation coefficients and introduces a constraint on an important intrinsic property of the dictionary.

There exist an infinite number of possible models and decompositions, each of which essentially aims at extracting or identifying information based on assumptions about the nature of the signals to be analysed. This thesis expands the toolbox available to the signal processing community by proposing novel models and algorithms, and offers comparisons with existing methods that highlight the strengths and limitations of the various techniques.

This last point highlights that there is not a single answer which can be used in every situation, but that there are principles that guide the design of new solutions built on an existing body of knowledge to push its boundaries. The principle of parsimony at the core of sparse approximation is one of those, and in the context of the *big picture* of signal processing, it can lead to a succinct explanation of properties of signals that have its roots in the dictionary utilized.

The investigation in dictionary learning for sparse approximation is still in many ways in its infancy. As in every thriving research field, many unanswered questions stem from every answered query, and the creative potential for adapting old methods and developing new ones is great and compelling.

The ideas and results presented in this work are a step contributing to this journey.

Appendix A

Derivations

A.1 On the convexity of the set of admissible dictionaries

A.1.1 The set of dictionaries with unit norm atoms is non-convex

In this appendix we prove that the set of dictionaries with unit-norm atoms $\mathcal{D} \stackrel{\text{def}}{=} \{\boldsymbol{\Phi} \in \mathbb{R}^{N\times K} : ||\boldsymbol{\phi}_k||_2 = 1\}$ is non-convex.

Let us consider a simple case where $N = K = 2$ and let us define two dictionaries $\boldsymbol{\Phi}, \boldsymbol{\Psi} \in \mathcal{D}$ that contain unit-norm atoms. A set is convex if and only if taking any two of its elements, their convex combination also lies in the set. Therefore, we need to study a dictionary $\boldsymbol{\Xi} \stackrel{\text{def}}{=} \theta\boldsymbol{\Phi} + (1-\theta)\boldsymbol{\Psi}$ with $\theta \in [0,1]$ resulting from the convex combination of the two elements of $\mathcal{D}$.

The constraint characterising the set $\mathcal{D}$ can be expressed in terms of the Gram matrix of the dictionaries. In particular, considering the matrix $\boldsymbol{\Phi}$, the following holds:

$$\boldsymbol{G}(\boldsymbol{\Phi}) \stackrel{\text{def}}{=} \boldsymbol{\Phi}^T\boldsymbol{\Phi} = \begin{bmatrix} \langle\boldsymbol{\phi}_1, \boldsymbol{\phi}_1\rangle & \langle\boldsymbol{\phi}_1, \boldsymbol{\phi}_2\rangle \\ \langle\boldsymbol{\phi}_2, \boldsymbol{\phi}_1\rangle & \langle\boldsymbol{\phi}_2, \boldsymbol{\phi}_2\rangle \end{bmatrix} = \begin{bmatrix} 1 & A \\ A & 1 \end{bmatrix} \tag{A.1}$$

where $A \in [-1,1]$ and the same holds for $\boldsymbol{G}(\boldsymbol{\Psi}) = \begin{bmatrix} 1 & B \\ B & 1 \end{bmatrix}$ with $B \in [-1,1]$.

Let us write now the Gram matrix of the convex combination $\boldsymbol{G}(\boldsymbol{\Xi})$ in terms of $\boldsymbol{G}(\boldsymbol{\Phi})$

and $\boldsymbol{G}(\boldsymbol{\Psi})$.

$$\begin{aligned}\boldsymbol{G}(\boldsymbol{\Xi}) &= (\theta\boldsymbol{\Phi} + (1-\theta)\boldsymbol{\Psi})^T (\theta\boldsymbol{\Phi} + (1-\theta)\boldsymbol{\Psi}) \\ &= \theta^2\boldsymbol{G}(\boldsymbol{\Phi}) + (1-\theta)^2 \boldsymbol{G}(\boldsymbol{\Psi}) + \left(\theta - \theta^2\right)\left(\boldsymbol{\Phi}^T\boldsymbol{\Psi} + \boldsymbol{\Psi}^T\boldsymbol{\Phi}\right).\end{aligned}$$

If we define the matrix

$$\boldsymbol{C} \stackrel{\text{def}}{=} \boldsymbol{\Phi}^T\boldsymbol{\Psi} = \begin{bmatrix} c_{11} = \langle\boldsymbol{\phi}_1, \boldsymbol{\psi}_1\rangle & c_{12} = \langle\boldsymbol{\phi}_1, \boldsymbol{\psi}_2\rangle \\ c_{21} = \langle\boldsymbol{\phi}_2, \boldsymbol{\psi}_1\rangle & c_{22} = \langle\boldsymbol{\phi}_2, \boldsymbol{\psi}_2\rangle \end{bmatrix}$$

containing inner products between the atoms in $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, $\boldsymbol{G}(\boldsymbol{\Xi})$ can be expressed as:

$$\begin{aligned}\boldsymbol{G}(\boldsymbol{\Xi}) &= \theta^2\boldsymbol{G}(\boldsymbol{\Phi}) + (1-\theta)^2\boldsymbol{G}(\boldsymbol{\Psi}) + \left(\theta - \theta^2\right)\left(\boldsymbol{C} + \boldsymbol{C}^T\right) \\ &= \theta^2 \begin{bmatrix} 1 & A \\ A & 1 \end{bmatrix} + (1-\theta)^2 \begin{bmatrix} 1 & B \\ B & 1 \end{bmatrix} + \left(\theta - \theta^2\right) \begin{bmatrix} 2c_{11} & c_{12} + c_{21} \\ c_{12} + c_{21} & 2c_{22} \end{bmatrix}\end{aligned}$$

The constraint that requires the atoms in $\boldsymbol{\Xi}$ to be normalized can be expressed as follows:

$$\theta^2 + (1-\theta)^2 + 2c_{11}\left(\theta - \theta^2\right) = \theta^2 + (1-\theta)^2 + 2c_{22}\left(\theta - \theta^2\right) = 1$$

which can be turned into the following:

$$\begin{aligned}\left(\theta^2 - \theta\right)(1 - c_{11}) &= 0 \\ \left(\theta^2 - \theta\right)(1 - c_{22}) &= 0.\end{aligned}$$

The equalities are satisfied in three cases: if $\theta = 0$ or $\theta = 1$ then the convex combination returns trivially either $\boldsymbol{\Phi}$ or $\boldsymbol{\Psi}$ and does not give any information about the convexity of the set $\mathcal{D}$. For $\theta \in (0, 1)$, the above constraints are satisfied if and only if $c_{11} = c_{22} = 1$, which in turns implies that the inner products $\langle\boldsymbol{\phi}_1, \boldsymbol{\psi}_1\rangle = \langle\boldsymbol{\phi}_2, \boldsymbol{\psi}_2\rangle = 1$ that only holds if $\boldsymbol{\Phi} = \boldsymbol{\Psi}$. The same proof generalises to the case where the number atoms and their dimension is greater than 2 leading to a higher number of equations of the same form.

The non-convexity of the set $\mathcal{D}$ implies that when updating a dictionary $\boldsymbol{\Phi} \in \mathcal{D}$ using, e.g. a gradient descent update as in the SparseNet algorithm described in Section

2.7.1 or any other dictionary update that do not explicitly constrain the update to the admissible set, then the resulting dictionary might not belong to $\mathcal{D}$ anymore. This is solved using a normalization step after the dictionary update.

A.1.2 The set of dictionaries with bounded mutual coherence is not convex

Let us now consider the set of dictionaries with bounded mutual coherence that is introduced in Section 5.3.1 as the constraint set of the dictionary de-correlation problem; it can be shown using a simple example that this set is also not convex.

Let $N = K = 2$ and consider a pair of orthonormal dictionaries $\mathbf{\Phi} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{\Psi} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Since they contain two orthonormal atoms, the mutual coherence of both dictionaries is zero; if we take their convex combination:

$$\mathbf{\Xi} = \theta\mathbf{\Phi} + (1-\theta)\mathbf{\Psi} = \begin{bmatrix} \theta & 1-\theta \\ 1-\theta & \theta \end{bmatrix} \tag{A.2}$$

then the mutual coherence of the resulting dictionary is:

$$\mu(\mathbf{\Xi}) = 2\theta(1-\theta) \tag{A.3}$$

that is zero for $\theta = 0$ or $\theta = 1$ and greater than zero for $\theta \in (0, 1)$.

The non-convexity of the set of dictionaries with bounded mutual coherence implies that the optimization (5.10) defined for de-correlating a dictionary cannot be solved with standard convex optimization tools.

Appendix B

Software

B.1 LOTBox

LOTBOX is a Matlab toolbox that implements lapped orthogonal transforms (LOTs) which I developed as the accompanying software for the work presented in Chapter 3.

The toolbox can be downloaded from the software repository: `https://code.soundsoftware.ac.uk/projects/lots` as a zip archive and installed simply by adding all the files to the current Matlab path.

The main functions implemented in the LOTBOX are the following:

- `lot.m` and `ilot.m`: implementation of the forward and inverse lapped orthonormal transform. The function takes as an input the signal and the parameters of the transform (that are a set of window lengths, the type of local orthonormal transforms, the length of the tail determining the amount of overlap between consecutive windows and the type of tail function). The forward and inverse transforms are implemented using a fast algorithm described in [70].

- `lappedwindow.m`: the fast algorithm for calculating LOTs employed by `lot.m` and `ilot.m` requires extracting frames from a signal using a window defined through this function. The frames are then processed using local orthonormal transforms.

- `dct.m` and `idcti.m`: implementation of the forward and inverse DCT of types I-IV. The DCT-IV in particular is needed to compute the MDCT as a special case of LOT.

- `lotplot.m`: plot function that displays the coefficients returned by the forward LOT in a spectrogram-like fashion.

Although many software implementation of special cases of LOTs such as STFT or MDCT are available, to the best of my knowledge at the time of development there was not a Matlab toolbox allowing a computation of LOTs that includes specifying different window lengths or local orthonormal transforms exploiting the full potential and flexibility of this class of transforms.

The code implementing the pitch-synchronous LOT described in Algorithm 7 is unfortunately not available for download. This is because the algorithm for pitch estimation that was employed as part of it is now protected by a copyright licence that does not allow neither commercial nor academic distribution. Although this affects the immediate reproducibility of the results presented in Chapter 3, similar experiments can be performed by selecting another suitable pitch estimation algorithm.

B.2 SMALLBox

SMALLBOX [21] is a Matlab toolbox for rapid prototyping and benchmarking of dictionary learning techniques that is being developed by a team at the Centre for Digital Music at Queen Mary University of London.

The toolbox can be downloaded from the software repository `https://code.soundsoftware.ac.uk/projects/smallbox` along with extensive documentation.

SMALLBOX comprises a collection of toolboxes developed by third party organisations for convex optimisation, sparse approximation and dictionary learning. By providing a common framework for these tools, it allows to test and develop new algorithms or modifications of existing methods while maintaining a common interface between the various components. As part of the development of SMALLBOX, I contributed to the design of an *add-on structure* that allows to realize additional code which is not included in the core of the SMALLBOX distribution, but that can be nonetheless interfaced to its components.

The *incoherent dictionary learning* SMALLBOX add-on was developed to as the accompanying software to Chapter 5. It can be downloaded from

`https://code.soundsoftware.ac.uk/projects/smallbox` along with a relative documentation that explains the SMALLBOX add-ons installation process.
The main functions implemented in the incoherent dictionary learning SMALLBOX add-on are the following:

- `dico_update_mocod.m`: implements the MOCOD dictionary update described in Section 5.2.2.
- `SMALL_test_mocod.m`: script that tests the MOCOD dictionary learning algorithm used to obtained Figure 5.3.
- `dico_decorr_symetric.m`: implements the INK-SVD dictionary decorrelation algorithm described in Section 5.3.2.
- `SMALL_test_coherence2.m`: script that tests the IPR and INK-SVD dictionary learning algorithms used to obtained Figure 5.4.
- `ipr.m`: implements the IPR dictionary decorrelation Algorithm 12.

In addition to the functions that implement or test the proposed algorithms, the folder `classes` contains a object-oriented implementation of dictionaries for sparse approximation, including functions for calculating quantities such as the mutual coherence of a dictionary. The dictionary class structure can be useful for research on sparse approximation beyond the scope of incoherent dictionary learning. Its object-oriented modularity makes it particularly suited to serve as a starting point for a more comprehensive toolbox.

Appendix C

Dictionary learning of convolved signals with overlap and save model

---

This appendix presents a variation of the model for dictionary learning of convolved signals and of the relative optimisation algorithm that is adapted to cases when the convolved observations result from the frame-by-frame analysis of an underlying high-dimensional signal.

C.1 Overlap and save algorithm

The model introduced in Section 4.1 and the optimization described in Section 4.3 assume that the observed signals $\boldsymbol{y}_m \in \mathbb{R}^{N+L-1}$ are generated according to equation (4.2) independently for each index m, whereas in some cases (such as the analysis of audio signals, for example) they are instead the consequence of a frame-by-frame processing of an underlying high-dimensional source signal. In order to update the model and optimisation to this situation, we consider the way high-dimensional signals are numerically convolved with impulse responses in a frame-by-frame fashion. Perhaps the simplest and widely known algorithm for block based convolution of one dimensional signals is the *overlap and save* method presented for the first time by Stockham Jr. [105]. Its block diagram is shown in Figure C.1: a potentially infinite anechoic signal is divided into frames of length N and each of them is convolved with the impulse response of length $L < N$ by the multiplication of the respective Fourier

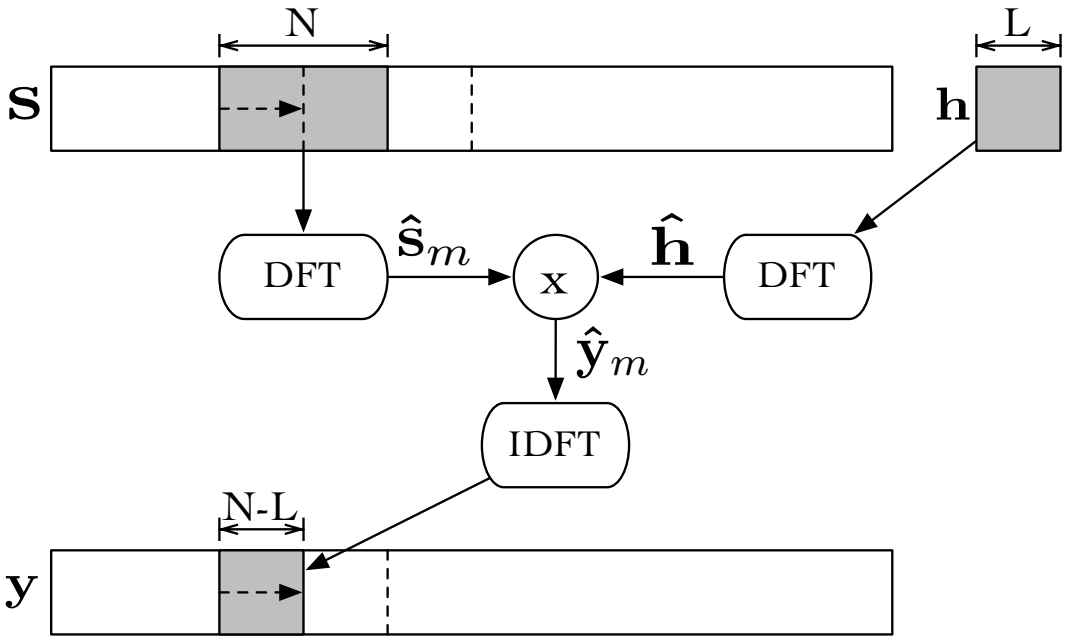


Figure C.1: Overlap and save algorithm for block-based linear convolution. A source signal is divided into blocks of length N and convolved with an impulse response of length L through multiplication of the respective Fourier transforms. After taking the IDFT of the resulting signal, the last $N-L$ samples are appended to the vector $\boldsymbol{y}$ and the successive frame taken from the source $\boldsymbol{s}$ is shifted by $N-L$ samples.

transforms. Since this results in a circular convolution of the two sequences as explained in Section 4.3, only the last $N-L$ samples of the output are selected and appended to the output vector containing the linear convolution of the two sequences (the interested reader can refer to [105] for a more thorough explanation of this algorithm).

The overlap and save algorithm can be expressed using a compact matrix notation by considering the matrix $\boldsymbol{Y} \in \mathbb{R}^{(N-L)\times M}$ as containing non overlapping frames of the convolved observation, the matrix $\boldsymbol{S} \in \mathbb{R}^{N\times M}$ containing in its columns frames of the source signal that overlap by $N-L$ frames and a partial Fourier matrix $\tilde{\boldsymbol{F}} \stackrel{\text{def}}{=} \boldsymbol{F}^{L+1:N} \in \mathbb{C}^{(N-L)\times N}$ which contains the last $N-L$ rows of the IDFT matrix of dimension N.

$$\begin{aligned}\boldsymbol{Y} &= \tilde{\boldsymbol{F}}\,\mathcal{D}\left(\hat{\mathring{\boldsymbol{h}}}\right)\hat{\boldsymbol{S}} \\ &= \tilde{\boldsymbol{F}}\,\mathcal{D}\left(\hat{\mathring{\boldsymbol{h}}}\right)\hat{\boldsymbol{\Phi}}\boldsymbol{X} \\ &= \tilde{\boldsymbol{F}}\hat{\boldsymbol{\Psi}}\boldsymbol{X}. \end{aligned} \tag{C.1}$$

This expression is similar to the model introduced in (4.7) except from the fact that the vectors $\boldsymbol{s}_m$ and $\breve{\boldsymbol{h}}$ whose Fourier transforms are multiplied together are of length N rather than $N + L - 1$, and that the partial Fourier matrix $\tilde{\boldsymbol{F}}$ is introduced. Learning the parameters of this model can be done following a method similar to the one described in Section 4.3.2 consisting of a block coordinate descent optimisation of sparse approximation coefficients and impulse response.

C.2 Dictionary learning of convolved signals block coordinate descent optimization with overlap and save model

Source signals optimization

Regarding the optimization of the source signals given a fixed impulse response, the model (C.1) can be used to define an optimization problem where a matrix of approximation coefficients is computed by solving the following:

$$\begin{aligned} \boldsymbol{X}^{\star} = \underset{\boldsymbol{X} \in \mathbb{R}^{K \times M}}{\arg\min} \ & \left|\left| \boldsymbol{Y} - \tilde{\boldsymbol{F}} \hat{\boldsymbol{\Psi}} \boldsymbol{X} \right|\right|_{\mathrm{F}} \\ \text{such that } & ||\boldsymbol{x}_m||_0 \leq S \quad \forall m = 1, \ldots, M. \end{aligned} \tag{C.2}$$

Here the convolved dictionary in the Fourier domain $\hat{\boldsymbol{\Psi}} = \mathcal{D}\left(\hat{\breve{\boldsymbol{h}}}\right)\hat{\boldsymbol{\Phi}}$ is obtained using the current estimate of the impulse response. To avoid writing double superscripts, the zero-pad operation on the impulse response will be implicitly assumed from now on and omitted from the notation. The estimated source signals can be obtained by the expression $\boldsymbol{S} = \boldsymbol{\Phi}\boldsymbol{X}$.

Impulse response optimization

In order to derive an expression for the optimization of the impulse response, we need to express the cost function

$$\mathcal{C}(\boldsymbol{h}) = \left|\left| \boldsymbol{Y} - \tilde{\boldsymbol{F}} \hat{\boldsymbol{\Psi}} \boldsymbol{X} \right|\right|_{\mathrm{F}}^2 \tag{C.3}$$

as a function of the vector $\boldsymbol{h}$.

We start by considering that the squared Frobenious norm of a matrix $\boldsymbol{B}$ is the trace of the Gram matrix $\boldsymbol{B}^{\mathcal{H}}\boldsymbol{B}$. Fixing an estimate of the source signals $\boldsymbol{S}$, the objective

function can therefore be written as

$$\begin{aligned}\mathcal{C}\left(\hat{\boldsymbol{h}}\right) &= \frac{1}{2}\operatorname{Tr}\left[\left(\boldsymbol{Y}-\tilde{\boldsymbol{F}}\,\mathcal{D}\left(\hat{\boldsymbol{h}}\right)\hat{\boldsymbol{S}}\right)^{\mathcal{H}}\left(\boldsymbol{Y}-\tilde{\boldsymbol{F}}\,\mathcal{D}\left(\hat{\boldsymbol{h}}\right)\hat{\boldsymbol{S}}\right)\right] \\ &= \frac{1}{2}\left\{\operatorname{Tr}\left[\hat{\boldsymbol{S}}^{\mathcal{H}}\,\mathcal{D}\left(\hat{\boldsymbol{h}}^{*}\right)\tilde{\boldsymbol{F}}^{\mathcal{H}}\tilde{\boldsymbol{F}}\,\mathcal{D}\left(\hat{\boldsymbol{h}}\right)\hat{\boldsymbol{S}}\right]-\operatorname{Tr}\left[\hat{\boldsymbol{S}}^{\mathcal{H}}\,\mathcal{D}\left(\hat{\boldsymbol{h}}^{*}\right)\tilde{\boldsymbol{F}}^{\mathcal{H}}\boldsymbol{Y}\right]-\operatorname{Tr}\left[\boldsymbol{Y}^{\mathcal{H}}\tilde{\boldsymbol{F}}\,\mathcal{D}\left(\hat{\boldsymbol{h}}\right)\hat{\boldsymbol{S}}\right]+C\right\}\end{aligned}$$

with $C=\operatorname{Tr}\left[\boldsymbol{Y}^{\mathcal{H}}\boldsymbol{Y}\right]$ a constant that does not depend on the impulse response. Before analysing the terms of this equation, we need two simple lemmas about traces.

Let $\boldsymbol{B}$ and $\boldsymbol{C}$ be two arbitrary matrices and $\boldsymbol{\Lambda}=\mathcal{D}\left(\boldsymbol{\lambda}\right)$ be a diagonal matrix. Then, we can write:

$$\begin{aligned}\operatorname{Tr}\left[\boldsymbol{B}\boldsymbol{\Lambda}\boldsymbol{C}\right] &= \sum_{i}\left[\boldsymbol{B}\boldsymbol{\Lambda}\boldsymbol{C}\right]_{ii}=\sum_{i}\sum_{j}B_{ij}\left[\boldsymbol{\Lambda}\boldsymbol{C}\right]_{ji}=\sum_{i}\sum_{j}B_{ij}\sum_{k}\Lambda_{jk}C_{ki} \\ &= \sum_{i}\sum_{j}B_{ij}\lambda_{j}C_{ji}=\sum_{j}\lambda_{j}\sum_{i}C_{ji}B_{ij}=\sum_{j}\lambda_{j}\left[\boldsymbol{C}\boldsymbol{B}\right]_{jj} \\ &= \boldsymbol{\lambda}^{T}\,\mathrm{d}\left(\boldsymbol{C}\boldsymbol{B}\right)\end{aligned}$$

Therefore, the following equality holds:

$$\operatorname{Tr}\left[\boldsymbol{B}^{\mathcal{H}}\boldsymbol{\Lambda}^{\mathcal{H}}\boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{B}\right]=\boldsymbol{\lambda}^{\mathcal{H}}\,\mathrm{d}\left(\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\boldsymbol{\Lambda}^{\mathcal{H}}\boldsymbol{C}\right)$$

Assuming that the matrix $\boldsymbol{C}$ is Hermitian (that is $\boldsymbol{C}^{\mathcal{H}}=\boldsymbol{C}$), and given that $\mathrm{d}\left(\boldsymbol{M}\right)=\left[\mathrm{d}\left(\boldsymbol{M}^{\mathcal{H}}\right)\right]^{*}$ we can write:

$$\operatorname{Tr}\left[\boldsymbol{B}^{\mathcal{H}}\boldsymbol{\Lambda}^{\mathcal{H}}\boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{B}\right]=\boldsymbol{\lambda}^{\mathcal{H}}\left[\mathrm{d}\left(\boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\right)\right]^{*}$$

and derive an expression for the diagonal elements of the matrix

$$\begin{aligned}\left[\boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\right]_{ii} &= \sum_{j}C_{ij}\left[\boldsymbol{\Lambda}\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\right]_{ji}=\sum_{j}C_{ij}\sum_{k}\Lambda_{jk}\left[\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\right]_{ki} \\ &= \sum_{j}C_{ij}\lambda_{j}\left[\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\right]_{ji}=\sum_{j}C_{ji}\left[\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\right]_{ji}\lambda_{j} \\ \mathrm{d}\left(\boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\right) &= \left[\boldsymbol{C}\circ\left(\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\right)\right]\boldsymbol{\lambda}\end{aligned}$$

where $\circ$ indicates the element-wise or Hadamard product of two matrices. Therefore, we

have:

$$\mathrm{Tr}\left[\boldsymbol{B}^{\mathcal{H}}\boldsymbol{\Lambda}^{\mathcal{H}}\boldsymbol{C}\boldsymbol{\Lambda}\boldsymbol{B}\right] = \boldsymbol{\lambda}^{\mathcal{H}}\left[\boldsymbol{C}\circ\left(\boldsymbol{B}\boldsymbol{B}^{\mathcal{H}}\right)\right]^{*}\boldsymbol{\lambda}$$

Equipped with these two results, we can express (C.3) as:

$$\begin{aligned}\mathcal{C}(\hat{\boldsymbol{h}}) &= \frac{1}{2}\left[\hat{\boldsymbol{h}}^{\mathcal{H}}\boldsymbol{\Gamma}\hat{\boldsymbol{h}} - \hat{\boldsymbol{h}}^{\mathcal{H}}\boldsymbol{\beta} - \hat{\boldsymbol{h}}^{T}\boldsymbol{\beta}^{*} + C\right] \\ &= \frac{1}{2}\left[\hat{\boldsymbol{h}}^{\mathcal{H}}\boldsymbol{\Gamma}\hat{\boldsymbol{h}} - 2\Re\left(\hat{\boldsymbol{h}}^{\mathcal{H}}\boldsymbol{\beta}\right) + C\right]\end{aligned}$$

where:

$$\begin{aligned}\boldsymbol{\Gamma} &\stackrel{\text{def}}{=} \left[\left(\tilde{\boldsymbol{F}}^{\mathcal{H}}\tilde{\boldsymbol{F}}\right)\circ\left(\hat{\boldsymbol{S}}\hat{\boldsymbol{S}}^{\mathcal{H}}\right)\right]^{*} \\ \boldsymbol{\beta} &\stackrel{\text{def}}{=} \mathrm{d}\left(\tilde{\boldsymbol{F}}^{\mathcal{H}}\boldsymbol{Y}\hat{\boldsymbol{S}}^{\mathcal{H}}\right)\end{aligned}$$

Since $\boldsymbol{\Gamma}$ is a symmetric positive definite matrix, this is a standard quadratic function of the complex variable $\hat{\boldsymbol{h}}$. The optimization of the impulse response given the current matrix of approximation coefficients $\boldsymbol{X}$ can be turned into a quadratic optimization problem involving the current estimate of the source signal:

$$\boldsymbol{h}^{\star} = \boldsymbol{F}\left\{\arg\min_{\hat{\boldsymbol{h}}}\frac{1}{2}\left[\hat{\boldsymbol{h}}^{\mathcal{H}}\boldsymbol{\Gamma}\hat{\boldsymbol{h}} - 2\Re\left(\hat{\boldsymbol{h}}^{\mathcal{H}}\boldsymbol{\beta}\right) + C\right]\right\} \tag{C.4}$$

This optimization can be solved by calculating the pseudo-inverse of the matrix $\boldsymbol{\Gamma}$, which results in:

$$\boldsymbol{h}^{\star} = \boldsymbol{F}\boldsymbol{\Gamma}^{\dagger}\boldsymbol{\beta}. \tag{C.5}$$

Appendix D

A Lie group method for dictionary rotation

The dictionary rotation performed as part of the IPR algorithm to adapt the dictionary $\boldsymbol{\Phi}$ to the representation of a set of training signals $\boldsymbol{Y}$ can be realized by multiplying $\boldsymbol{\Phi}$ by an orthonormal matrix optimized according to (5.21), which is the closed-form solution discussed in Section 5.4.1. Alternatively, a Lie group method can be employed as described in this appendix.

Re-stating the problem (5.21), we are seeking the solution of a least-squares problem involving the minimisation of the residual norm, subject to an orthonormal constraint on the matrix to be optimized:

$$\underset{\boldsymbol{W}\in\mathcal{O}(N)}{\arg\min}\ ||\boldsymbol{Y}-\boldsymbol{W}\boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} \tag{D.1}$$

where $\mathcal{O}(N)$ is the space of orthonormal matrices of dimension N. This optimization is similar to a problem encountered for non-negative independent component analysis (NN-ICA), making it possible to borrow methods employed in that field for our purpose. We refer the interested reader to [83] for an exhaustive explanation of NN-ICA and the relative optimization techniques. Here, we limit our discussion to the one that has been employed in the IPR algorithm, namely a conjugate gradient optimization constrained to the $\mathcal{SO}(N)$ manifold of special orthogonal matrices with positive determinant.

D.1 Constrained optimization in the $\mathcal{SO}(N)$ manifold

The set $\mathcal{O}(N)$ is a manifold embedded in the space of general $N \times N$ matrices. If we associate to this set the matrix multiplication operation, we obtain a *group*, which is defined as an algebraic structure consisting of a set together with an operation which satisfies the following properties:

1 - Closure under the operation: the multiplication of any two orthogonal matrices returns an orthogonal matrix.

2 - Associativity: matrix multiplication is associative. Given the matrices $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{C}$, the equality $\boldsymbol{ABC} = (\boldsymbol{AB})\boldsymbol{C} = \boldsymbol{A}(\boldsymbol{BC})$ holds.

3 - Existence of an identity element: the orthogonal identity matrix $\boldsymbol{I}$ maps any matrix $\boldsymbol{A}$ to itself $\boldsymbol{IA} = \boldsymbol{A}$.

4 - Existence of an inverse element: the set includes, for every element $\boldsymbol{W} \in \mathcal{O}(N)$, an inverse element $\boldsymbol{W}^{-1} \in \mathcal{O}(N)$, such that $\boldsymbol{W}^{-1}\boldsymbol{W} = \boldsymbol{I}$. For orthogonal matrices, $\boldsymbol{W}^{-1} = \boldsymbol{W}^T$.

It has been proved that the group described so far is a disconnected Lie group, which loosely means that we can associate a system of coordinates, as in a vector space $\mathbb{R}^{N \times N}$, to a local region of the manifold (much like two-dimensional cartographic maps are associated with local regions of the earth), but that we can only move smoothly from one point to another in the manifold if these do not belong to *disconnected* regions [83]. We would rather consider *connected* Lie groups, where this complication does not occur and we can move around the manifold in every direction. The subset $\mathcal{SO}(N) \subset \mathcal{O}(N)$ of orthogonal matrices with determinant equal to one, with the matrix multiplication operation, is a connected Lie group. Therefore, we choose to modify the problem (5.21) by imposing the constraint $\boldsymbol{W} \in \mathcal{SO}(N)$. This results in a proper *rotation*[1] of the dictionary expressed by the following:

$$\hat{\boldsymbol{W}} = \underset{\boldsymbol{W} \in \mathcal{SO}(N)}{\arg\min} \; ||\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}} \,. \tag{D.2}$$

[1] As opposed to the more general, improper rotation that results from solving (5.21).

In order to solve (D.2), one option is to choose an update that is locally tangent to the manifold (by exploiting the local isomorphism between the manifold and the relative vector space, as in the cartographic analogy) and then to project back the updated matrix onto the manifold $\mathcal{SO}(N)$ [27]. However, we found that this method exhibited slow convergence in our experiments. Instead, we perform the optimization in a Lie algebra associated to the constraint manifold.

D.2 Conjugate gradient descent in the Lie algebra $\mathfrak{so}(N)$

A Lie algebra is a vector space with an associated binary operation called Lie bracket (see [83] for a more detailed exposition). It can be shown that the space of skew-symmetric matrices, that is, any matrix $\boldsymbol{B}$ that satisfies $\boldsymbol{B} = -\boldsymbol{B}^T$, with the matrix commutator operation $[\boldsymbol{A}, \boldsymbol{B}] = \boldsymbol{AB} - \boldsymbol{BA}$ is a Lie algebra associated with the constraint manifold $\mathcal{SO}(N)$, and we denote it by $\mathfrak{so}(N)$. Moreover, any element belonging to this Lie algebra can be mapped into an element belonging to the Lie group $\mathcal{SO}(N)$ by a matrix exponential (and vice versa using the matrix logarithm). That is, for every $\boldsymbol{B} \in \mathfrak{so}(N)$, $\exp(\boldsymbol{B}) \in \mathcal{SO}(N)$. Here the matrix exponential is defined as

$$\exp(\boldsymbol{B}) = \sum_{p=1}^{\infty} \frac{1}{p!} \boldsymbol{B}^p. \tag{D.3}$$

A Lie group method [50] can be used to optimize a cost function working in the Lie algebra while satisfying the manifold constraint. Its steps can be summarised as follows:

I - Start from a matrix $\boldsymbol{B} = \log(\boldsymbol{W}) \in \mathfrak{so}(N)$, for example from the zero matrix, that corresponds to the matrix logarithm of the identity $\boldsymbol{0} = \log(\boldsymbol{I}) \in \mathfrak{so}(N)$.

II - Find an update $\Delta\boldsymbol{B}$ that improves the cost function and move in the Lie algebra to an updated $\boldsymbol{B}' = \boldsymbol{B} + \Delta\boldsymbol{B}$.

III - Map the updated matrix onto the constraint manifold $\boldsymbol{V} = \exp(\boldsymbol{B}') \in \mathcal{SO}(N)$.

IV - Calculate $\boldsymbol{W}' = \boldsymbol{VW} \in \mathcal{SO}(N)$.

It is possible to perform steps I to IV iteratively by using the method of *parallel transport* (the interested reader can find more detailed information in [83] and references therein), which allows us to work in the Lie algebra $\mathfrak{so}(N)$ and use any of the tools

developed for numerical optimization in vector spaces. In our proposed algorithm, we employ a conjugate gradient optimization that consists of the following steps at each iteration $i = \{1, 2, \ldots, I\}$:

I - Calculate the gradient of the unconstrained cost function $\mathcal{C}(\boldsymbol{W}) = \frac{1}{2} ||\boldsymbol{Y} - \boldsymbol{W}\boldsymbol{\Phi}\boldsymbol{X}||_{\mathrm{F}}^2$.

$$(\nabla_{\boldsymbol{W}}\mathcal{C})^{(i)} = \left(\boldsymbol{W}^{(i)}\boldsymbol{\Phi}\boldsymbol{X} - \boldsymbol{Y}\right)(\boldsymbol{\Phi}\boldsymbol{X})^T \tag{D.4}$$

II - Map the gradient to the Lie algebra, obtaining:

$$\boldsymbol{R}^{(i)} = 2\,\mathrm{skew}\left[(\nabla_{\boldsymbol{W}}\mathcal{C})^{(i)}\left(\boldsymbol{W}^{(i)}\right)^T\right] \tag{D.5}$$

where $\mathrm{skew}(\boldsymbol{A}) = \frac{1}{2}\left(\boldsymbol{A} - \boldsymbol{A}^T\right)$ is the skew-symmetric component of the matrix $\boldsymbol{A}$.

III - Find a conjugate search direction in the Lie algebra as:

$$\boldsymbol{H}^{(i)} = -\boldsymbol{R}^{(i)} + \gamma\boldsymbol{H}^{(i-1)}$$

where

$$\gamma = \frac{\left\langle \boldsymbol{R}^{(i)}, \boldsymbol{R}^{(i)} - \boldsymbol{R}^{(i-1)} \right\rangle}{\left\langle \boldsymbol{R}^{(i-1)}, \boldsymbol{R}^{(i-1)} \right\rangle}$$

is the Polak-Ribière formula [100] and $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \mathrm{Tr}[\boldsymbol{A}^T\boldsymbol{B}]$ indicates the matrix inner product.

IV - Perform a line search in the direction $\boldsymbol{H}^{(i)}$ as:

$$t^{\star(i)} = \underset{t\in\mathbb{R}}{\arg\min}\, \mathcal{C}\left(\exp\left(t\boldsymbol{H}^{(i)}\right)\right)$$

V - Update the orthogonal matrix as:

$$\boldsymbol{W}^{(i+1)} = \exp\left(t^{\star(i)}\boldsymbol{H}^{(i)}\right)\boldsymbol{W}^{(i)}$$

The steps of the IPR algorithms with the Lie group method dictionary rotation are summarised in Algorithm 13.

Algorithm 13: Iterative projections and rotations with Lie group method rotation

Input: $\boldsymbol{\Phi}, \boldsymbol{Y}, \boldsymbol{X}, \mu_0, I_{IP}, I_R$
Output: $\boldsymbol{\Phi}^\star$
// Initialisation
1 $i_{IP} \leftarrow 1$;
2 $i_R \leftarrow 1$;
3 **while** $i_{IP} \leq I_{IP}$ *and* $\mu(\boldsymbol{\Phi}) > \mu_0$ **do**
  // Perform one iteration of the iterative projections algorithm 9
4 $\boldsymbol{\Phi} \leftarrow \text{IP}(\boldsymbol{\Phi}, \mu_0, 1)$;
  // Rotate dictionary
5 $\boldsymbol{W} \leftarrow \boldsymbol{I}$ // Initialise rotation matrix
6 $\boldsymbol{H} \leftarrow \boldsymbol{0}$ // Initialise search direction
7 **for** $i_R = 1 : I_R$ **do**
    // Find an update direction and step in the Lie algebra
8 $\nabla_{\boldsymbol{W}} \leftarrow (\boldsymbol{W}\boldsymbol{\Phi}\boldsymbol{X} - \boldsymbol{Y})(\boldsymbol{\Phi}\boldsymbol{X})^T$;
9 $\boldsymbol{R} \leftarrow 2\,\text{skew}\left[(\nabla_{\boldsymbol{W}}\mathcal{C})\,\boldsymbol{W}^T\right]$;
10 $\boldsymbol{H} \leftarrow -\boldsymbol{R} + \gamma\boldsymbol{H}$;
11 $t \leftarrow \underset{t\in\mathbb{R}}{\arg\min}\, \mathcal{C}\left(\exp\left(t\boldsymbol{H}\right)\right)$;
    // Map the update to the constraint manifold
12 $\boldsymbol{W} \leftarrow \exp\left(t\boldsymbol{H}\right)\boldsymbol{W}$;
13 **end**
14 $\boldsymbol{\Phi} \leftarrow \boldsymbol{W}\boldsymbol{\Phi}$;
15 $i_{IP} \leftarrow i_{IP} + 1$
16 **end**

It is worth noting about this technique that it is a first-order method to solve (D.2) which only requires the computation of the unconstrained gradient at line 8 to define an update direction in the Lie algebra $\mathfrak{so}(N)$. However, the minimisation at line 11 that is needed to define an optimal step-size and the matrix exponential at line 12 employed to map the updated matrix onto the manifold $\mathcal{SO}(N)$ are computationally expensive and largely outweigh the resources needed to compute the closed-form solution of the rotation step detailed in Algorithm 12 of Section 5.4.1. Moreover, the Lie group method is an iterative algorithm that restricts the admissible set of solutions of (5.21) to the Lie manifold $\mathcal{SO}(N)$ that is a sub-set of the space of orthonormal matrices $\mathcal{O}(N)$.

Although Algorithm 13 remains an interesting application of Lie group methods to a dictionary rotation problem and might be a useful starting point if the optimization (5.21) is substituted by a more general objective function that does not admit a closed-form solution, the strategy detailed in Section 5.4.1 is comparatively faster, more general and more accurate than the method described in this appendix, and it is

therefore preferred in the implementation of the IPR algorithm.

Bibliography

[1] C. A. Abad. Pitch-synchronous multiresolution analysis of music signals. Master's thesis, Universitat Pompeu Fabra, 2007.

[2] S. Abdallah and M. D. Plumbley. Unsupervised analysis of polyphonic music by sparse coding. *IEEE Trans on Neural Networks*, 17(1):179–196, Jan. 2006.

[3] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley. Audio inpainting. *IEEE Trans. on Audio, Speech and Language Processing*, 20(3):922–932, Mar. 2012.

[4] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11):4311–4322, Nov. 2006.

[5] M. Aharon, M. Elad, and A. M. Bruckstein. On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them. *Linear Algebra and its Applications*, 416(1):48–67, Jul. 2006.

[6] J. B. Allen and D. A. Berkley. Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, 65(4):943–950, Apr. 1979.

[7] A. Benichoux, E. Vincent, and R. Gribonval. A compressed sensing approach to the simultaneous recording of multiple room impulse responses. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2011. Submitted to WASPAA'11.

[8] T. Blumensath and M. E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5):629–654, 2008.

[9] J. Bobin, Y. Moudden, and J.-L. Starck. Enhanced source separation by morphological component analysis. In *Proceedings of the IEEE International*

Conference on Acoustics, Speech and Signal Processing (ICASSP), volume 5, pages 833–866, 2006.

[10] J. Bobin, J.-L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho. Morphological component analysis: An adaptive thresholding strategy. *IEEE Trans. on Image Processing*, 16(11):2675–2681, Nov. 2007.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[12] K. Brandeburg. MP3 and AAC explained. In *Proceedings of the 17th Audio Engineering Society International Conference on High Quality Audio Coding*, number 17-009, Aug. 1999.

[13] J. C. Brown. Calculation of a constant Q spectral transform. *J. Acoust. Soc. Am.*, 89(1):425–434, 1991.

[14] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math*, 59(8):1207–1223, Aug. 2006.

[15] E. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. on Information Theory*, 52(12):5406–5425, Dec. 2006.

[16] E. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, Mar. 2008.

[17] S. S. Chen, D. L. Donoho, and M. A. Sauders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, Mar. 2001.

[18] A. Cichocki and S.-i. Amari. *Adaptive Blind Signal and Image Processing*. Wiley and Sons, 2002.

[19] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Trans. on Information Theory*, 55(5):2230–2249, May 2009.

[20] W. Dai, T. Xu, and W. Wang. Dictionary learning and update based on simultaneous codeword optimization (SIMCO). In *Proceedings of the IEEE*

International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2037–2040, 2012.

[21] I. Damnjanovic, M. E. P. Davies, and M. D. Plumbley. SMALLbox - An evaluation framework for sparse representations and dictionary learning algorithms. In *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pages 418–425, 2010.

[22] L. Daudet. Sparse and structured decompositions of signals with the molecular matching pursuit. *IEEE Trans. on Audio, Speech and Language Processing*, 14(5):1808–1816, Sep. 2006.

[23] M. E. Davies and L. Daudet. Sparse audio representations using the MCLT. *Signal Processing*, 86(3):457–470, 2006.

[24] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, 1997.

[25] R. A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.

[26] D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1-norm minimization problems when the solution may be sparse. *IEEE Trans. on Information Theory*, 54(11):4789–4812, Nov. 2008.

[27] S. C. Douglas. Self-stabilized gradient algorithms for blind source separation with orthogonality constraints. *IEEE Trans on Neural Networks*, 11(6):1490–1497, Nov. 2000.

[28] I. Drori and D. L. Donoho. Solution of ℓ_1 minimization problems by LARS/HOMOTOPY methods. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 3, pages 636–639, May 2006.

[29] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):pp. 407–451, 2004.

[30] M. Elad. *Sparse and redundant representations.* Springer, 2010.

[31] M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Sep. 2006.

[32] K. Engan, S. O. Aase, and J. H. Husøy. Method of optimal directions for frame design. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 5, pages 2443–2446, 1999.

[33] G. Evangelista. Pitch-synchronous wavelet representation of speech and music signals. *IEEE Trans. on Signal Processing*, 41(12):3313–3330, Dec. 1993.

[34] A. Fazel and S. Chakrabartty. Sparse auditory reproducing kernel (SPARK) features for noise-robust speech recognition. *IEEE Trans. on Audio, Speech and Language Processing*, 20(4):1362–1371, May 2012.

[35] C. Févotte. Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[36] 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, Mar. 2009.

[37] C. Févotte, B. Torrésani, L. Daudet, and S. J. Godsill. Sparse linear regression with structured priors and application to denoising of musical audio. *IEEE Trans. on Audio, Speech and Language Processing*, 16(1):174–185, 2008.

[38] P. Georgiev, F. Theis, and A. Cichocki. Sparse component analysis and blind source separation of undetermined mixtures. *IEEE Trans on Neural Networks*, 16(4):992–996, Jul. 2005.

[39] D. Giacobello, M. G. Chistensen, M. N. Murthi, S. H. Jensen, and M. Moonen. Enhancing sparsity in linear prediction of speech by iteratively reweighted 1-norm minimization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.

[40] D. Giacobello, C. Græsbøll, M. N. Murthi, S. H. Jensen, and M. Moonen. Retrieving sparse patterns using a compressed sensing framework: Applications to speech coding based on sparse linear prediction. *IEEE Signal Processing Letters*, 17(1):103–106, Jan. 2010.

[41] D. Giannoulis, D. Barchiesi, A. Klapuri, and M. D. Plumbley. On the disjointness of sources in music using different time-frequency representations. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 261–264, 2011.

[42] S. Gleichman and Y. C. Eldar. Blind compressed sensing. *IEEE Trans. on Information Theory*, 57(10):6958 – 6975, Oct. 2011.

[43] R. Gribonval. Should penalized least squares regression be interpreted as a maximum a posteriori estimation? *IEEE Trans. on Signal Processing*, 59(5):2405–2410, 2011.

[44] R. Gribonval. Sparsity & co.: An overview of analysis vs synthesis in low-dimensional signal models. In *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, page 14, Jun. 2011.

[45] R. Gribonval and S. Lesage. A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. In *Proceedings of the 14th European Symposium on Artificial Neural Networks (ESANN)*, pages 323–330, 2006.

[46] R. Gribonval and K. Schnass. Dictionary identification: Sparse matrix-factorisation via ℓ_1-minimisation. *IEEE Trans. on Information Theory*, 56(7):3523–3539, Jul. 2010.

[47] R. Gribonval and P. Vandergheynst. On the exponential convergence of matching pursuit in quasi-incoherent dictionaries. *IEEE Trans. on Information Theory*, 52(1):255–261, Jan. 2006.

[48] B. K. Horn, H. M. Hilden, and S. Negahdaripour. Closed-form solution of absolute

orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5:1127–1135, July 1988.

[49] K. Huang and S. Aviyente. Sparse representation for signal classification. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[50] A. Iseries, H. Munthe-Kaas, S. P. Nørsett, and A. Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000.

[51] M. G. Jafari and M. D. Plumbley. A doubly sparse greedy adaptive dictionary learning algorithm for music and large-scale data. In *Proceedings of the Audio Engineering Society Convention*, number 8087, May 2010.

[52] M. G. Jafari and M. D. Plumbley. Fast dictionary learning for sparse representations of speech signals. *IEEE Journal of Selected Topics in Signal Processing*, 5:1025–1031, Sep. 2011.

[53] X. Jaureguiberry, P. Leveau, S. Maller, and J. J. Burred. Adaptation of source-specific dictionaries in non-negative matrix factorization for source separation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[54] P. Jost, P. Vandergheynst, S. Lesage, and R. Gribonval. MoTIF: An efficient algorithm for learning translation invariant dictionaries. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 5, pages 857–860, Jul. 2006.

[55] A. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, pages 216–221, 2006.

[56] A. Klapuri and T. Virtanen. Representing musical sounds with an interpolating state model. *IEEE Trans. on Audio, Speech and Language Processing*, 18(3):613–624, Mar. 2010.

[57] M. Kowalski and B. Torrésani. Random models for sparse signals expansion on union of bases with application to audio signals. *IEEE Trans. on Signal Processing*, 56(8):3468–3481, Aug. 2008.

[58] M. Kowalski, E. Vincent, and R. Gribonval. Beyond the narrowband approximation: Wideband convex methods for under-determined reverberant audio source separation. *IEEE Trans. on Audio, Speech and Language Processing*, 18(7):1818–1829, Sep. 2010.

[59] P. Leveau, E. Vincent, G. Richard, and L. Daudet. Instrument-specific harmonic atoms for mid-level music representation. *IEEE Trans. on Audio, Speech and Language Processing*, 16(1):116–128, Jan. 2008.

[60] M. Lewicki and T. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, Feb. 2000.

[61] Y. Lou, A. Bertozzi, and S. Soatto. Direct sparse deblurring. *Journal of Mathematical Imaging and Vision*, pages 1–12, 2010.

[62] B. Mailhé, D. Barchiesi, and M. D. Plumbley. INK-SVD: Learning incoherent dictionaries for sparse representations. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3573–3576, 2012.

[63] B. Mailhé, S. Lesage, R. Gribonval, F. Bimbot, and P. Vandergheynst. Shift-invariant dictionary learning for sparse representations: Extending K-SVD. In *Proceedings of the 16th European Signal Processing Conference (EUSIPCO)*, Aug. 2008.

[64] B. Mailhé and M. D. Plumbley. Local optimality of dictionary learning algorithms. In *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, page 67, 2011.

[65] J. Mairal, F. Bach, and J. Ponce. Task-Driven Dictionary Learning. Technical Report 7400, Institut National de Recherche Informatique et en Automatique (INRIA), Sep. 2010.

[66] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the international conference on machine learning*, 2009.

[67] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, Jan. 2010.

[68] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[69] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2008.

[70] S. Mallat. *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Number ISBN: 012-4-66606-X. Academic Press, London, UK, Sep. 1999.

[71] S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Trans. on Signal Processing*, 41(12):3397–3415, Dec. 1993.

[72] H. S. Malvar. *Signal Processing with Lapped Transforms*. Number ISBN: 978-0-89006-467-2. Artech Print on Demand, Jan. 1992.

[73] S. G. McGovern. A model for room acoustics. available at http://www.sgm-audio.com/research/rir/rir.html, 2004.

[74] M. Moussallam, P. Leveau, and S. M. A. Sbaï. Sound enhancement using sparse approximation with speclets. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 221–224, Mar. 2010.

[75] S. Nam, M. E. Davies, M. Elad, and R. Gribonval. Cosparse analysis modeling - uniqueness and algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5804–5807, 2011.

[76] D. Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26:301–321, 2008.

[77] D. Needell and R. Vershynin. Uniform uncertanty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of Computational Mathematics*, 9(3):317–334, 2009.

[78] B. 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, Jun. 1996.

[79] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Number ISBN 0130834432. Pearson, 1998.

[80] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis*, 20(3):389–404, Jul. 2000.

[81] A. Ozerov and C. Févotte. Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation. *IEEE Trans. on Audio, Speech and Language Processing*, 18(3):550–563, Mar. 2010.

[82] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27th Asilomar Confrence on Signals, Systems and Computers*, volume 1, pages 40–44, Nov. 1993.

[83] M. D. Plumbley. Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras. *Neurocomputing*, 67:161–197, 2005.

[84] M. D. Plumbley. Dictionary learning for L1-exact sparse coding. In M. E. Davies, C. J. James, S. A. Abdallah, and M. D. Plumbley, editors, *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, pages 406–413. Springer, 2007.

[85] S. Qian and D. Chen. Discrete gabor transform. *IEEE Trans. on Signal Processing*, 41(7):2429–2438, Jul. 1993.

[86] B. Raj and P. Smaragdis. Latent variable decomposition of spectrograms for single channel speaker separation. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 17–20, Oct. 2005.

[87] I. Ramírez, F. Lecumberry, and G. Sapiro. Sparse modeling with universal priors and learned incoherent dictionaries. Technical Report 2279, Institute for Mathematics and its Applications, University of Minnesota, Sep. 2009.

[88] E. Ravelli, G. Richard, and L. Daudet. Union of MDCT bases for audio coding. *IEEE Trans. on Audio, Speech and Language Processing*, 16(8):1361–1372, Nov. 2008.

[89] F. Rodriguez and G. Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries. Technical Report 2213, Institute for Mathematics and its Applications, University of Minnesota, 2008.

[90] R. Rubinstein, A. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, Jun. 2010.

[91] R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity: Learning sparse dictionaries for sparse signal approximation. *IEEE Trans. on Signal Processing*, 58(3):1553–1564, Mar. 2010.

[92] K. Schnass, R. Gribonval, H. Rauhut, and P. Vandergheynst. Atoms of all channels, unite! Average case analysis of multi-channel sparse recovery using greedy algorithms. *Journal of Fourier Analysis and Applications*, 14(5):655–687, 2008.

[93] K. Schnass and P. Vandergheynst. Average performance analysis for thresholding. *IEEE Signal Processing Letters*, 14(11):828–831, Nov. 2007.

[94] K. Schnass and P. Vandergheynst. Dictionary preconditioning for greedy algorithms. *IEEE Trans. on Signal Processing*, 56(5):1994–2002, May 2008.

[95] K. Schnass and P. Vandergheynst. Classification via incoherent subspaces. Submitted to Rejecta Mathematica, 2010.

[96] K. Schnass and P. Vandergheynst. A union of incoherent spaces model for classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5490–5493, Mar. 2010.

[97] S. Scholler and H. Purwins. Sparse approximations for drum sound classification. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):933–940, Sep. 2011.

[98] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, July 1948.

[99] M. Shashanka, B. Raj, and P. Smaragdis. Probabilistic latent variable models as non-negative factorizations. *Computational Intelligence and Neuroscience Journal*, May 2008.

[100] J. R. Shewchuk. An Introduction to the Conjugate Gradient Method Withouth the Agonizing Pain. Unpublished draft, available at http://www.cs.cmu.edu/ quake-papers/painless-conjugate-gradient.pdf, Aug. 1994.

[101] S. Shlien. The modulated lapped transform, its time-varying forms, and its applications to audio coding standards. *IEEE Trans. on Speech and Audio Processing*, 5(4):359–366, Jul. 1997.

[102] R. Singh, B. Raj, and P. Smaragdis. Latent-variable decomposition based dereverberation of monaural and multi-channel signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1914–1917, 2010.

[103] K. Skretting and K. Engan. Recursive least squares dictionary learning algorithm. *IEEE Trans. on Signal Processing*, 58(4):2121–2130, Apr. 2010.

[104] E. C. Smith and M. Lewicki. Efficient auditory coding. *Nature*, 439(23):978–982, Feb. 2006.

[105] T. G. Stockham Jr. High speed convolution and correlation. In *Proceedings of the Spring Joint Computer Conference, AFIPS*, volume 28, pages 229–233, Apr. 1966.

[106] T. Strohmer and R. W. J. Heath. Grassmannian frames with applications to coding and communication. *Applied and Computational Harmonic Analysis*, 14(3):257–275, 2003.

[107] P. Sudhakar, S. Arberet, and R. Gribonval. Double sparsity: Towards blind estimation of multiple channels. In *Proceedings of the Ninth International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, Sep. 2010.

[108] P. Sudhakar and R. Gribonval. Sparse filter models for solving permutation indeterminacy in convolutive blind source separation. In *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 2009.

[109] V. Y. F. Tan and C. Févotte. A study of the effect of source sparsity for various transforms on blind audio source separation performance. In *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 2005.

[110] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society, Series B (Methodological)*, 58(1):267–288, Jan. 1996.

[111] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. on Information Theory*, 50(10):2231–2242, Oct. 2004.

[112] J. A. Tropp. *Topics in sparse approximation*. PhD thesis, University of Texas at Austin, Aug. 2004.

[113] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Trans. on Information Theory*, 52(3):1030–1051, Mar. 2006.

[114] J. A. Tropp. On the conditioning of random subdictionaries. *Applied and Computational Harmonic Analysis*, 25(1):1–24, Jul. 2008.

[115] J. A. Tropp. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, Jun. 2010.

[116] J. A. Tropp, I. S. Dhillon, R. W. J. Heath, and T. Strohmer. Designing structured tight frames via an alternating projection method. *IEEE Trans. on Information Theory*, 51(1):188–209, Jan. 2005.

[117] E. Vincent and M. D. Plumbley. Low bitrate object coding of musical audio using bayesian harmonic models. *IEEE Trans. on Audio, Speech and Language Processing*, 15(4):1273–1282, May 2007.

[118] D. Wipf and S. Nagarajan. Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. *IEEE Journal of Selected Topics in Signal Processing, Special Issue on Compressed Sensing*, 4(2):317–329, Apr. 2010.

[119] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. on Signal Processing*, 57(7):2479–2493, Jul. 2009.

[120] M. Yaghoobi, T. Blumensath, and M. E. Davies. Dictionary learning for sparse approximations with the majorization method. *IEEE Trans. on Signal Processing*, 57(6):2178–2191, 2009.

[121] M. Yaghoobi, L. Daudet, and M. E. Davies. Parametric dictionary design for sparse coding. *IEEE Trans. on Signal Processing*, 57(12):4800–4810, Dec. 2009.

[122] Ö. Yılmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Trans. on Signal Processing*, 52(7):1830–1847, Jul. 2004.

[123] M. Zibulevsky and B. A. Pearlmutter. Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation*, 13(4):863–882, Apr. 2001.