From Local to Global Sparse Modeling

Jeremias Sulam

From Local to Global Sparse Modeling

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Jeremias Sulam

Submitted to the Senate
of the Technion — Israel Institute of Technology
Heshvan 5778 Haifa November 2017

This research was carried out under the supervision of Prof. Michael Elad, in the Faculty of Computer Science.

Some results in this thesis have been published as articles by the author and research collaborators in conferences and journals during the course of the author's doctoral research period, the most up-to-date versions of which being:

List of Publications

- **Jeremias Sulam**, Boaz Ophir and Michael Elad, *Image denoising through multi-scale learnt dictionaries.* 2014 IEEE International Conference onImage Processing (ICIP).
- **Jeremias Sulam** and Michael Elad, *Expected patch log likelihood with a sparse prior.* Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science, 2015
- Javier Turek, **Jeremias Sulam**, Michael Elad and Irad Yavneh. *Fusion of ultrasound harmonic imaging with clutter removal using sparse signal separation.* 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
- **Jeremias Sulam**, Boaz Ophir, Michael Zibulevsky and Michael Elad. *Trainlets: Dictionary learning in high dimensions.* IEEE Transactions on Signal Processing, 2016.
- **Jeremias Sulam***, Yaniv Romano* and Michael Elad, *Gaussian Mixture Diffusion.* IEEE International Conference on the Science of Electrical Engineering (ICSEE), 2016.
- Vardan Papyan*, **Jeremias Sulam*** and Michael Elad. *Working Locally Thinking Globally: Theoretical Guarantees for Convolutional Sparse Coding.* IEEE Transactions on Signal Processing, 2017.
- **Jeremias Sulam**, Yaniv Romano and Ronen Talmon, *Dynamical system classification with diffusion embedding for ECG-based person identification.* Signal Processing, 2017.
- Vardan Papyan, Yaniv Romano, **Jeremias Sulam** and Michael Elad, *Convolutional Dictionary Learning via Local Processing.* IEEE International Conference on Computer Vision (ICCV) 2017.
- **Jeremias Sulam**, Vardan Papyan, Yaniv Romano and Michael Elad, *Multi-Layer Convolutional Sparse Modeling: Pursuit and Dictionary Learning*, Submitted, 2017.

Note: * denotes equal contribution.

Acknowledgements

There are many people to whom I am truly grateful, and without them this collection of works would have not been possible.

First and foremost, I like to sincerely thank my advisor, Professor Michael Elad. Mark Van Doren wrote, "The art of teaching is the art of assisting discovery." Miki has been a terrific teacher: always assisting, explaining and inquiring, showing me how fascinating the process of discovery could be. He has been extremely supportive and it has been inspiring to work with him. For all this, I will always be grateful.

Thank you to my friends, those close and afar, recent and old, as they have always provided me with thoughtful advice and support. Special thanks to those with whom I had the privilege to collaborate and learn: Boaz, Javier, Vardan, Yaniv, Dima – I hope we can continue to work together. I also want thank faculty and lab members whose work has helped me along these few years: Michael Zibulevsky, Ronen Talmon, Irad Yavneh, Yana Katz, Nadav Toledo, Tom Palny and Anna Kleiner: your work is invaluable to everybody who is lucky enough to work here. I also thank my friends at the LSyDNL Lab (UNER), for instructing, accompanying and inspiring me in my first academic and research steps.

Special thanks goes to my loving partner, Heidi, with whom I have learned to support each other in hard moments, as well as rejoice in the happier ones. It is thanks to this PhD that I got to meet you, and it embodies only the beginning of our exciting adventures together.

Last, but not least, I am enormously thankful to my family: to my parents, Alberto and Marga, who always encouraged me to search for answers, understanding and exploration, and to my brother, Ariel, who has always been clever enough to remind me never to stop smiling and having fun in the process. This thesis is dedicated to you all.

The generous financial help of the Technion is gratefully acknowledged.

Contents

List of Figures

Abstract

Signal models have always been central to the development of new and better algorithms. This thesis is concerned with sparse representations modeling, which builds upon the observation that natural signals can be well approximated by a few elements from a vast collection of atoms, commonly termed dictionary. Over the last decade, many works have studied the problem of retrieving the sparse set of atoms that best represent a given measurement signal, and proposing ways of adapting and training this model from real-world data. The latter task, known as the dictionary learning problem, has empowered sparse enforcing methods to achieve remarkable results in many different fields from signal and image processing and restoration to higher level tasks such as detection, classification and several other machine learning applications.

This new sparsity-inspired model, while greatly successful, has typically been applied to small and local signal patches due to the computational constraints that solving these problems entails. More precisely, various computationally efficient algorithms were suggested for solving global problems by addressing a collection of relatively independent local sub-problems. This paradigm results in a series of inconsistencies, however, which we loosely refer to as a *local-global gap*, with both practical and theoretical implications. In this thesis, we will first propose different and complementary strategies to significantly alleviate several of these issues by deploying multi-scale analysis tools and global regularization techniques, such as the expected patch log-likelihood and Laplacian regularization. We will then circumvent these inconsistencies altogether by tackling the problem of learning a sparse representation model for high dimensional signals. Building on the double sparsity model and a cropped wavelets dictionary, this will take us to propose a new dictionary learning algorithm resulting in large trainable atoms, dubbed *Trainlets*. This approach will not only deliver state-of-art results in dictionary learning, but will also enable us to address problems and applications that are simply out of the scope of local methods.

Towards the second part of this thesis, we will consider the Convolutional Sparse Coding (CSC) model, which will be shown to be a (somewhat surprising) answer for the local-global gap. This relatively new model, however, comes with a loose and hardly applicable theoretical analysis. We will expand much of the classical sparse representations theory to the convolutional case, providing uniqueness, stability and recovery guarantees based on a new local measure of sparsity. On the one hand, this will give a theoretical justification to abundant work dealing with algorithmic solutions to this problem. On the other hand, our approach will guide the development of new pursuit and dictionary learning algorithms that, while solving global problems, think and work locally. Towards the last part, and motivated by a very recent connection between

CSC and the rising topic of deep learning, we will analyze the Multi-Layer Convolutional Sparse Coding model, which proposes a global construction composed of a cascade of convolutional layers. We will propose a sound pursuit algorithm for signals following these model assumptions by adopting a projection approach, providing new and improved bounds on the stability of its solution and analyzing different algorithmic alternatives. A dictionary learning algorithm will be naturally derived from our study, enabling to train the nested convolutional filters from real data, and employing them in several applications.

This thesis condenses a tour of different alternatives that seek to better serve global problems while leveraging the powerful locally sparse modeling framework. The outcomes of this work are several new algorithms, practical solutions, novel models and theoretical results that, I hope and believe, will empower the next generation of signal modeling.

Chapter 1

Introduction

Signal models are the expression of our partial understanding of the manifestation of natural phenomena. They intend to condense, in a few mathematical rules, the basic characteristics that describe *all there is to know* about the signal of interest. Clearly, the accuracy of the characterization is problem-dependent, and more accurate descriptions will commonly lead to more complex models. These mathematical constructions, while imperfect, provide progressively better comprehension of real world signals, and have enabled the development of practical applications such as signal and image restoration, detection and classification, among a myriad of other problems.

The concept of parsimony, on the other hand, has been known to thinkers and scholars for a long time. It suffices to refer to Occam's Razon [Sob15] to understand that simplicity and parsimony where admired qualities among philosophers and early scientists. This concept inevitably influenced physics and the modern sciences, from Newton [Haw03] to Solomonof [RH11]. Eventually, around two or three decades ago, the advent of Wavelets begun to influence signal and image modeling, bringing about the understanding that natural signals can be (well) described by a linear combination of only a few building blocks or components. Since then, much progress has been achieved by sparse representations in terms of a redundant collection of signal *atoms*, commonly known dictionary [BDE09]. Backed by elegant theoretical results, this model has led to a series of works dealing either with the problem of the pursuit of such decompositions, or with the design and learning of better atoms from real data [RBE10]. The latter problem, termed dictionary learning, empowered sparse enforcing methods to achieve remarkable results in many different fields from signal and image processing [RPE14, MBS09] to machine learning [JLD13, PCCP14, SPC14].

This new transform model is typically accompanied by hard optimization problems and algorithms that have a high computational complexity [RZE08]. These reasons, together with the curse of dimensionality, causes that whenever this model is deployed to real-world applications, this sparse prior is not enforced on the global signal or image but rather on small local portions, or *patches*, from it [Ela10]. This local paradigm, while powerful and computationally effective, has important limitations that arise from modeling a global signal by simply operating on it locally. These limitations, loosely referred to as the *local-global gap*, are both practical and theoretical.

In restoration applications, this results in texture-like artifacts, particularly noticeable in smooth regions. From an estimation perspective, local neighboring regions do not provide a coherent estimation of their overlap. Could we circumvent these problems by somehow learn a global (but perhaps restricted) model? More importantly, these methods have overlooked a central question: what is the actual global model imposed on signals when working under this local sparse modeling framework? These issues will be our guiding points throughout this work.

1.1 Local Restoration Methods

Our starting point is the SparseLand model and its application to image restoration problems based on a local modeling strategy. Image restoration is any task that aims to recover an image that has undergone some damaging process, usually modeled as resulting from a degradation operator applied to the image with the addition of acquisition noise. Some typical image restoration problems include *image denoising*, where only noise is added to the original image; *image inpaiting*, where some pixels or even areas of the image have been completely removed by the degradation operator; and *image debluring*, resulting in an image that looks out of focus or otherwise blurred.

All these are ill-posed problems, and so it becomes necessary to include some prior information or assumptions into the restoration formulation. This is exactly the role of the signal model: incorporating prior knowledge or beliefs about the unknown signal. In this context, the restoration task consists of obtaining an estimate of the image which is properly related to the measurements but that is likely under the imposed model. In one very popular form, this process reduces – either implicitly or explicitly – to the Maximum a Posteriori (MAP) estimator of the unknown image under some prior. In the last two decades, much effort has been put into developing better models for image restoration, some of these being based on adaptive smoothness [TM98], low-rankness [CLMW11], self-similarity [DFKE06], sparsity [EA06], and combinations of some of these [MBS09, PSWS03]. These priors have become very popular as they often result in computationally effective algorithms with state-of-the-art performance.

1.2 Upgrading Local Restoration Methods

However effective, most state of the art methods share the limitation of working on a single scale and operating on patches of the same size. This local strategy of treating image patches has inherent limitations in terms of the amount of information available per local estimate. Our first contribution [SOE14] tackles this particular point, presenting a patch-based denoising algorithm relying on a sparsity-inspired model (K-SVD [AEB06]) while leveraging a multi-scale analysis framework. This allows us to overcome some of the disadvantages of the popular algorithms by considering patches of different effective size in a natural and computationally practical way. We propose an algorithm in which we look for a sparse representation under an already sparsifying wavelet transform by adaptively training a dictionary on the different decomposition bands of the noisy image, leading to a multi-scale extension of the K-SVD denoising algorithm [EA06].

We then combine the single-scale and multi-scale approaches by merging both outputs by a weighted joint sparse coding strategy. Our experiments on natural images indicate that the proposed method is competitive with state of the art algorithms in terms of PSNR, while giving superior results as to visual quality.

From a Bayesian point of view, most algorithms that operate locally invest their effort on maximizing an a posteriori estimator with respect to some sophisticated prior on the extracted patches, and eventually obtain the final image simply by averaging these processed patches back together. This averaging process, while practical, leads to eventual patches that are in fact not likely under the employed model. Recently, the Expected Patch Log Likelihood (EPLL) method was introduced in [ZW11], arguing that the chosen model should be enforced on the final reconstructed image patches instead. In the context of a Gaussian Mixture Model (GMM), this idea has been shown to lead to state-of-the-art results in image denoising and debluring. In our second contribution [SE15], we combine the EPLL framework with a sparse-representation prior. Our derivations lead to a close yet extended variant of the popular K-SVD algorithm. We show that in order to effectively maximize the EPLL the denoising process should be iterated, and we present a method that intrinsically determines the corresponding local noise thresholds in order to improve the image estimate. Our results show a notable improvement over K-SVD in image denoising and inpainting, achieving comparable performance to that of EPLL with GMM.

More broadly, all these local denoising algorithms (GMM, EPLL, K-SVD, among others) can be understood as global pseudo-linear operators. These algorithms – acting on the entire noisy image – can be thought of having a two-step implementation process: first building a linear operator based on some non-linear decision rules (e.g., which atoms to employ for each patch) and then simply applying this operator to the noisy image in order to obtained the result. This is the approach we take in [SRE16] when studying the resulting operator from the Gaussian Mixture Model (GMM) [PSWS03]. Focusing then on the denoising formulation, we incorporate a graph-based regularization term leveraging the corresponding GMM graph that emerges from this denoiser [Mil13]. From a variational interpretation, the resulting algorithm extends and improves the non-local diffusion algorithm [GO07] by replacing the Non-Local Means kernel [BCM05] with a GMM one. Our results indicate that this approach, termed Gaussian Mixtures Diffusion (GMD), consistently improves over both the original GMM scheme and the non-local diffusion algorithm. Furthermore, GMD is competitive or even better than the state of-the-art method of EPLL.

1.3 High Dimensional Dictionary Learning

Clearly, the local-global gap arises from employing local models to address a global problems. What if one could employ a global model directly? This direction of work is indeed appealing, as it avoids all the problems discussed in the section above. Such a strategy is not as trivial as it might seem, however, since obtaining a model for increasingly higher dimensional signals suffers from the curse of dimensionality, and imposing or training it from real data can result in prohibitive computational costs.

In another contribution [SOZE16], we show how to efficiently handle bigger dimensions and go beyond the small patches in sparsity-based signal and image processing methods. We build our approach based on a new cropped Wavelet decomposition, which enables a multi-scale analysis with virtually no border effects. We then employ this as the base dictionary within a double sparsity model to enable the training of adaptive dictionaries. To cope with the increase of training data, while at the same time improving the training performance, we present an Online Sparse Dictionary Learning (OSDL) algorithm to learn this model effectively, enabling it to handle millions of examples and resulting in large adaptable atoms that we coin *Trainlets*.

The reported results show that not only this approach provides state of the art performance in dictionary learning, but it also allows sparsity-based methods to tackle new problems that remained unreachable until now. For example, in [SE16], we address the specific problem of inpainting large regions of face images. This is a challenging task, as attempting to solve it with local methods is generally infeasible: for the inpainting to be successful, one must have a global model of how a face should look like, what facial features such images contain, etc. We avoid these problems by employing the above Trainlets approach to learn a global dictionary for this class of images. When this model is deployed with a sparse prior, we obtain very plausible reconstructions that outperform competing methods.

1.4 Convolutional Sparse Coding

While the above global method is very effective in modeling relatively large image patches, or even global images from a similar class, it is still too limited when attempting to model arbitrarily large natural images. An elegant and more profound solution to the above-described local-global dichotomy is given by the Convolutional Sparse Coding (CSC) model. This model assumes that the global dictionary is structured as the concatenation of banded Circulant matrices, and in doing so it provides a global model with a shift-invariant local prior. Although several works have presented algorithmic solutions to the global pursuit problem under this new model, no truly-effective guarantees are known for the success of such methods.

Moving to our fifth contribution [PSE17a], we address the theoretical aspects of the sparse convolutional model, providing the first meaningful answers to questions of uniqueness of solutions and success of pursuit algorithms. To this end, we generalize mathematical quantities, such as the ℓ_0 norm, the mutual coherence and the Spark, to their counterparts in the convolutional setting, which intrinsically capture local measures of the global model. We further extend the analysis to a noisy regime, thereby considering signal perturbations and model deviations. We address questions of stability of the sparsest solutions and the success of pursuit algorithms, both greedy and convex. Classical definitions such as the RIP are generalized to the convolutional model, and existing notions such as the ERC are connected to our setting. On the algorithmic side, we propose a simple yet effective approach to solve the global pursuit problem by using simple local processing, thus offering a first of its kind bridge between global modeling of signals and their patch-based local treatment.

1.5 Multi-Layer CSC

While the CSC model has gained increasing attention, a multi-layer (ML) extension of this model was very recently proposed in [PRE17]. Most interestingly, this ML-CSC model consisting of a cascade of convolutional sparse layers, provides a new interpretation of Convolutional Neural Networks (CNNs). Under this framework, the computation of the forward pass in a CNN is equivalent to a pursuit algorithm aiming to estimate the nested sparse representation vectors – or feature maps – from a given input signal. These results are encouraging, as they show for the first time stability guarantees for a problem for which the forward pass provides an approximate solution. Despite having served as a pivotal connection between CNNs and sparse modeling, a deeper understanding of the ML-CSC is still lacking: there are no pursuit algorithms that can serve this model exactly, nor are there conditions to guarantee a non-empty model. While one can easily obtain signals that *approximately* satisfy the ML-CSC constraints, it remains unclear how to simply sample from the model and, more importantly, how one can train the convolutional filters from real data.

In a last contribution of this thesis [SPRE17], we propose a sound pursuit algorithm for the ML-CSC model by adopting a projection approach. We provide new and improved bounds on the stability of the solution of such pursuit and we analyze different practical alternatives to implement this in practice. We show that the training of the filters is essential to allow for non-trivial signals in the model, and we derive an online algorithm to learn the dictionaries from real data, effectively resulting in cascaded sparse convolutional layers. Last, but not least, we demonstrate the applicability of the ML-CSC model for several applications in an unsupervised setting, providing competitive results to state-of-the-art work in the deep-learning arena. This last work represents a bridge between matrix factorization, sparse dictionary learning and sparse auto-encoders, and we analyze these connections in detail.

1.6 Thesis Structure

The organization of this thesis follows, to a large extent, the organization of this introductory chapter. In addition, each chapter is roughly organized following the corresponding related publications.

We will begin by providing background material on Sparse Representation modeling in a general sense in Chapter 2, introducing relevant pursuit algorithms and theoretical guarantees. We will further introduce the problem of dictionary learning and describe a few popular and recent algorithms for this task. We will then move to the novel part of this work, starting by addressing image processing restoration problems in Chapter 3, focusing particularly on image denoising. We will present three strategies to alleviate some of the issues arising from the local-global gap while still working on small image patches. Chapter 4 will be concerned with introducing our approach for high dimensional dictionary learning with Trainlets. We will naturally derive border-effects-free cropped wavelets, introduce the learning algorithm and finally study the performance of this approach on several image processing problems, including

the inpainting of large face images.

We will then undertake a systematic study of the CSC model in Chapter 5, where we will provide new theoretical guarantees for the optimization problems involved and for the algorithms that attempt to solve them. We will then extend this analysis to the Multi-Layer version of the CSC in Chapter 6, providing new and tighter bounds for the recovery of signals satisfying these model assumptions. We will introduce a learning algorithm to adapt the nested filters from real data, and demonstrate the model on several applications.

We will lastly conclude in Chapter 7, where we will comment on open questions and working directions that arise from the works compiled in this thesis.

1.7 Notation

This thesis is compiled based on the results from several publications, often treating different models and applications. Nevertheless, we will strive to maintain a consistent notation throughout this document whenever possible. We will generally refer to vectors with bold lowercase letters to differentiate them from matrices that will be denoted by bold uppercase letters, and from scalar quantities, in non-bold letters. The notation employed in Chapter 5 will represent an exception to this general rule, as we will employ both lower and uppercase bold letters for local and global vectors, respectively. We will make this distinction precise at that point.

While each chapter will employ notation specific to each topic, we will typically refer to signals (or images) by the vectors $\mathbf{y}, \mathbf{x}$ and $\mathbf{z}$, while denoting $\mathbf{v}$ or $\mathbf{n}$ measurement noise or model deviations. Throughout this thesis, we will employ n to depict the signal dimension of local patches, and N to refer to the global dimension. The number of atoms, $\mathbf{d}_j$, in a dictionary $\mathbf{D}$ will be denoted by m. The remaining notation will be specified when needed.

Chapter 2

Preliminaries on Sparse Modelling

Chapter Abstract

Sparse representations modeling assumes that a natural signal can be well described by a linear combination of only a few basic signal components, or atoms, represented as columns from a redundant matrix, termed dictionary. The problem of searching for this sparse set of building blocks, while being NP-hard in general, can be addressed by a variety of approximation algorithms with provable performance bounds. When coupled with different approaches that allow for learning the dictionary from real data, this model provides excellent performance in a variety of signal and image processing applications. In this chapter we review the basics of sparse representation modeling, assesing the type of guarantees that can be claimed and the typical approaches to deploy this model to real data.

2.1 Sparse Solutions to Redundant Systems of Equations

The benefits of having a representation where most of the information is concentrated on only *a small* part of our data have been known for decades. Such representations, where most of its entries are zero except for only a few of them, are called *sparse*. Sparse signals – or signals belonging to the *SparseLand* model – are such that can be represented or well approximated by a linear combination of only a few signal elements, called atoms. Given a collection of atoms, represented by the matrix $\mathbf{D}$ and termed *dictionary*, one can represent an $n-$dimensional signal $\mathbf{x}$ by

$$\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}$$

where $\mathbf{D} \in \mathbb{R}^{n \times m}$, $\boldsymbol{\gamma} \in \mathbb{R}^m$, and $||\boldsymbol{\gamma}||_0 \ll n$, where the l_0 pseudo-norm[1] counts the number of non-zero coefficients in $\boldsymbol{\gamma}$. The dictionary is usually redundant, i.e. $m > n$, and while this choice ruins the benefit of orthogonality between the atoms in $\mathbf{D}$, it enables very sparse representation vectors $\boldsymbol{\gamma}$.

2.2 Sparse Coding

Clearly, if $m > n$ and $\mathbf{D}$ is a full-rank matrix, there exist infinite representations $\boldsymbol{\gamma}$ that can generate $\mathbf{x}$. We therefore regularize this problem by searching for the *sparsest* of all these representations, yielding a formulation known as *sparse coding*, or pursuit. This pursuit problem can be formally posed as follows:

$$(P_0): \quad \min_{\boldsymbol{\gamma}} \ ||\boldsymbol{\gamma}||_0 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\boldsymbol{\gamma}. \tag{2.1}$$

Given the non-convexity (and highly discontinous) ℓ_0 norm, this problem is NP hard in general [DMA97]. Nevertheless, several results have shed light on the circumstances under which a unique solution can be claimed. These guarantees are typically given in terms of properties of the dictionary $\mathbf{D}$, such as the *Spark*, defined as the minimum number of linearly dependent columns in $\mathbf{D}$ [DE03]. Formally,

$$\sigma(\mathbf{D}) = \min_{\boldsymbol{\gamma}} \|\boldsymbol{\gamma}\|_0 \ \text{ s.t. } \ \mathbf{D}\boldsymbol{\gamma} = \mathbf{0}, \ \boldsymbol{\gamma} \neq \mathbf{0}.$$

Based on this property, a solution obeying $\|\boldsymbol{\gamma}\|_0 < \sigma(\mathbf{D})/2$ is necessarily the sparsest one [DE03]. Unfortunately, this bound is of little practical use, as computing the Spark of a matrix is a combinatorial problem – just as hard as solving the problem in Equation (2.1), and infeasible in practice.

Other guarantees are given in terms of the *mutual coherence* of the dictionary, $\mu(\mathbf{D})$. This

[1] From this point onward, we will refer to ℓ_0 norm as norm, in a slight abuse of nomenclature.

measure quantifies the similarity of atoms in the dictionary, defined in [DE03] as:

$$\mu(\mathbf{D}) = \max_{i \neq j} \frac{|\mathbf{d}_i^T \mathbf{d}_j|}{\|\mathbf{d}_i\|_2 \, \|\mathbf{d}_j\|_2}.$$

Unlike the Spark, this quanatity is easily computable. A relation between the Spark and the mutual coherence was shown in [DE03], stating that $\sigma(\mathbf{D}) \geq 1 + \frac{1}{\mu(\mathbf{D})}$. This, in turn, enables the formulation of a practical uniqueness bound guaranteeing that $\boldsymbol{\gamma}$ is the unique solution of the P_0 problem if:

$$\|\boldsymbol{\gamma}\|_0 < \frac{1}{2}\left(1 + 1/\mu(\mathbf{D})\right). \tag{2.2}$$

Due to the combinatorial nature of these problems, the usual way to tackle such a pursuit is to approximate its solution instead of finding it exactly. Two approaches are possible: either minimizing the problem in (2.1) with a greedy strategy, or else relaxing the non-convex l_0 norm by some convex alternative such as the popular l_1 norm. Within the first family of methods, the Orthogonal Matching Pursuit (OMP) [PRK93] has shown to yield a good compromise between accuracy and complexity, and it has become a popular option. On the other hand, FOCUSS [GR97], shrinkage algorithms [BD08, Ela06] and other convex optimization techniques [Tro06] enable to approximate the solution of the problem above in the l_1 case. Interestingly, both families of methods have been proven to recover the true solution of the P_0 problem if the representation vector is sparse *enough*. This sparsity (or rather, cardinality) bound depends on the mutual coherence of the dictionary just as before [DET06, Tro04, DE03, GN03] and detailed in Equation (2.2).

In real world applications, due to noisy measurements and model imperfections, the idealistic setting portrayed above is not directly applicable. Consider one is given the measurements $\mathbf{y} = \mathbf{D}\boldsymbol{\gamma} + \mathbf{n}$, where $\mathbf{n}$ is a nuisance vector of bounded energy, $\|\mathbf{n}\|_2 \leq \epsilon$. In this case, one can extend the P_0 problem to consider these signal perturbations and enforcing the model only approximately, obtaining:

$$(P_0^\epsilon): \quad \min_{\boldsymbol{\gamma}} \|\boldsymbol{\gamma}\|_0 \;\; \text{s.t.} \;\; \|\mathbf{D}\boldsymbol{\gamma} - \mathbf{y}\|_2 \leq \epsilon. \tag{2.3}$$

Unlike the noiseless case, given a solution to this problem, one can not claim its uniqueness but instead can guarantee that it will be close enough to the true vector $\boldsymbol{\gamma}$ that generated the signal $\mathbf{y}$. This kind of stability results have been derived in recent years by leveraging the Restricted Isometry Property (RIP) [CT05]. A matrix $\mathbf{D}$ is said to have a k-RIP with constant δ_k if this is the smallest quantity such that

$$(1 - \delta_k)\|\boldsymbol{\gamma}\|_2^2 \leq \|\mathbf{D}\boldsymbol{\gamma}\|_2^2 \leq (1 + \delta_k)\|\boldsymbol{\gamma}\|_2^2,$$

for every $\boldsymbol{\gamma}$ satisfying $\|\boldsymbol{\gamma}\|_0 = k$. Based on this property, it was shown that assuming $\boldsymbol{\gamma}$ is sparse enough, the distance between $\boldsymbol{\gamma}$ and the solution to the P_0^ϵ problem is bounded [Ela10]. Similar stability claims can be formulated in terms of the mutual coherence also, by exploiting its relationship with the RIP property. For example, from [Ela10], if $\hat{\boldsymbol{\gamma}}$ is the solution to the

problem in (2.3), and if $\|\boldsymbol{\gamma}\|_0 = k \leq \frac{1}{2}(1 + 1/\mu(\mathbf{D})$, then

$$\|\hat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}\|_2^2 \leq \frac{4\epsilon^2}{1 - (1-2k)\mu(\mathbf{D})}.$$

Success guarantees of practical algorithms, such as the Orthogonal Matching Pursuit (OMP) and the Basis Pursuit Denoising (BPDN), have also been derived under this regime. In the same spirit of the aforementioned stability results, both approaches where shown to recover a solution close to the true sparse vector as long as some sparsity constraint, relying on the mutual coherence of the dictionary and the noise energy, is met [Tro04, DET06].

Another useful property for analyzing the success of pursuit methods, initially proposed in [Tro04], is the Exact Recovery Condition (ERC). Formally, one says that the ERC is met for a support $\mathcal{T}$ with a constant θ whenever

$$\theta = 1 - \max_{i \notin \mathcal{T}} \|\mathbf{D}_{\mathcal{T}}^{\dagger} \mathbf{d}_i\|_1 > 0,$$

where we have denoted by $\mathbf{D}_{\mathcal{T}}^{\dagger}$ the Moore-Penrose pseudoinverse of the dictionary restricted to support $\mathcal{T}$, and $\mathbf{d}_i$ refers to the i^{th} atom in $\mathbf{D}$. Assuming the above is satisfied, the stability of both the OMP and BP was proven in [Tro06]. Moreover, in an effort to provide a more intuitive result, the ERC was shown to hold whenever the total number of non-zeros in $\mathcal{T}$ is less than a certain number, which is a function of the mutual coherence.

2.3 Dictionary Learning

Clearly, the choice of the dictionary is a central issue as the ability to obtain a sparse representation for a given signal will depend on how well the respective atoms can represent it. Initially, analytically defined dictionaries were – and still are – used for sparse coding [CDS01]. These dictionaries are defined in terms of algorithms or transformations rather than explicit matrices, and as such are have a very efficient implementation. Common examples of these include the Discrete Cosine Transform [Jai79], wavelets [Mal08], contourlets [DV02], shearlets [K$^+$12], among other. However, even though some of these analytical constructions are optimal (in an approximation rate sense) for certain classes of functions, the mathematical rules by which they are defined might not necessarily provide *the most* sparsifying transform for real data.

A better alternative is to train the dictionary from real data. In this problem, the objective is to obtain the most representative set of atoms such that they enable a good (i.e., sparser) representation for many real cases. Consider one gathers a collection of N training examples arranged in a matrix $\mathbf{Y} \in \mathbb{R}^{n \times N}$. The dictionary learning problem can then be posed as minimizing a reconstruction term, subject to the sparsity of each representation vector and a constraint on the atoms (to resolve a norm ambiguity):

$$\min_{\mathbf{D},\boldsymbol{\Gamma}} ||\mathbf{Y} - \mathbf{D}\boldsymbol{\Gamma}||_F^2 \text{ s.t. } \begin{cases} ||\boldsymbol{\gamma}_i||_0 \leq k & \forall i \\ ||\mathbf{d}_j||_2 = 1 & \forall j \end{cases}, \tag{2.4}$$

where $\mathbf{\Gamma} \in \mathbb{R}^{m \times N}$ are the corresponding sparse vectors, ordered column wise.

This problem is highly non-convex, not only for inheriting the ℓ_0 norm of the sparse coding problem, but also because of the multiplication of the factors. Many algorithms have been proposed to minimize the above objective, and though different, they generally follow an alternating minimization strategy. Under this iterative framework, at each t^{th} iteration, on frizzes one variable (say, the current estimate $\mathbf{D}^{t-1}$), and then minimizes the cost in (2.4) just with respect the sparse codes $\mathbf{\Gamma}^t$. This is nothing but the pursuit problem we commented on the previous subjection for a number of N signals. Again, greedy approaches and relaxation methods can be employed for this stage.

Once the matrix $\mathbf{\Gamma}^t$ has been updated, one keeps this variable fixed and minimizes the cost with respect to the dictionary $\mathbf{D}$. In other words, and up to the norm constraint (which is easily amendable with a subsequent projection) the problem reduces to:

$$\min_{\mathbf{D}} ||\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}^t||_F^2. \tag{2.5}$$

It is in this step that most approaches differ from one another. For example, in one of its earliest forms (the MOD algorithm from [EAH00]) the matrix $\mathbf{D}$ was simply found by the least-squares solution,

$$\mathbf{D}^t = \mathbf{Y}\mathbf{\Gamma}^{t+}.$$

This whole process is then iterated until convergence, or until a sufficient representation error has been obtained.

More sophisticated approaches have been proposed. Among them, the K-SVD algorithm [AEB06] has been extremely popular in a myriad of different applications. The main characteristic of this approach is to employ a sequential atom-wise update of the dictionary. This way, one is interested in minimizing the cost in (2.5) only with respect to the, say, j^{th} atom. One can write this problem as

$$\min_{\mathbf{d}_j} ||\mathbf{E}_j - \mathbf{d}_j \mathbf{\Gamma}_j^T||_F^2,$$

where $\mathbf{E}_j = \mathbf{Y} - \sum_{i \neq j} \mathbf{d}_i \mathbf{\Gamma}_j^T$ is the error term of the j^{th} atom and $\mathbf{\Gamma}_j^T$ denotes the j^{th} row of the matrix $\mathbf{\Gamma}$. This problem has a solution in terms of the SVD decomposition of the error matrix $\mathbf{E}_j$ – thus, the name of the algorithm – providing a rank-1 approximation to $\mathbf{E}_j$ However, in order to maintain the sparsity of $\mathbf{\Gamma}$, this decomposition is performed not on the entire error matrix but rather on a reduced version of it, containing only those signals that employ the atom $\mathbf{d}_j$ in their decomposition.

Before moving on, we comment on a popular alternative method, the Online Dictionary Learning (ODL) algorithm by [MBPS10]. Both of the above methods (K-SVD and MOD) are *batch* learning algorithms: the entire set of training examples is needed in order to perform a dictionary update step. This approach can be prohibitive in high dimensional scenarios or in cases with a very large dataset. The ODL proposes a solution that works in an online regime, by minimizing a quadratic surrogate of the expected loss function. Generally speaking, this algorithm also alternates between sparse coding and a dictionary update stage, but it does so

one sample at a time. At iteration t, one first solves the following pursuit for the t^{th} signal example $\mathbf{y}_t$

$$\boldsymbol{\gamma}_t = \arg\min_{\boldsymbol{\gamma}} \frac{1}{2}\|\mathbf{y}_t - \mathbf{D}\boldsymbol{\gamma}\|_2^2 + \lambda\|\boldsymbol{\gamma}\|_1,$$

solved in practice with the LARS algorithm [EHJ+04]. Then, the update of $\mathbf{D}$ is driven by the surrogate loss

$$\min_{\mathbf{D}} \frac{1}{t}\sum_{i=1}^{t}\left(\frac{1}{2}||\mathbf{y}_i - \mathbf{D}\boldsymbol{\gamma}_i||_2^2 + \lambda\|\boldsymbol{\gamma}_i\|_1\right).$$

The ODL algorithm searches for the dictionary that minimizes the cost above in a block-coordinate descent manner, which results in a closed-form update that leverages the past information of previously seen samples, and it does so in an efficient way by keeping two auxiliary matrices. After adding a few modifications (like managing mini-batches, scaling past data and pruning unused atoms), this algorithm has shown state-of-the-art performance in a number of different applications [MBPS09, MBPS10].

2.4 Sparse Modeling and Inverse Problems

Inverse problems in image processing consist of recovering an original image that has been degraded. Denoising, debluring and inpainting are specific and common such examples, in which one is given the measurements $\mathbf{y}$, generally modeled as

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \tag{2.6}$$

where $\mathbf{A}$ is a known linear[2] operator and $\mathbf{n}$ represents measurement noise, assumed to be independent and normally distributed. Put formally, these problems attempt to estimate the underlying image $\mathbf{x}$ given the measurement $\mathbf{y}$. Since these restoration problems are ill-posed, it becomes necessary to include image priors as regularizers, which results in developing a Maximum a Posteriori (MAP) estimator for the unknown image $\hat{\mathbf{x}}$. This can be formulated as an optimization problem where we look for an estimate which is close enough to the measured image while being likely under this prior. Most state of the art methods employ, either implicitly of explicitly, some prior knowledge of this form [EA06, PSWS03, MBS09, DFKE06].

In the field of sparse representations, this restoration task can be generally formulated as follows. Considering the denoising case for simplicity (i.e., $\mathbf{A} = \mathbf{I}$), and letting $\mathbf{P}_i$ denote a patch-extraction operator that extracts an n-dimensional patch from $\mathbf{y}$, one is usually interested in an optimization problem like the following [EA06, MESM08]:

$$\min_{\mathbf{x},\{\boldsymbol{\gamma}_i\},\mathbf{D}} \quad \frac{1}{2}||\mathbf{x} - \mathbf{y}||_2^2 + \lambda\sum_i \frac{1}{2}||\mathbf{D}\boldsymbol{\gamma}_i - \mathbf{P}_i\mathbf{x}||_2^2 + \mu_i||\boldsymbol{\gamma}_i||_0,$$

In words, this loss searches for an estimate $\mathbf{x}$ closed to $\mathbf{y}$ such that every patch from it has an

[2]While we will restrict our study to known (and linear) degradation operators, unknown degradation operators have also been considered in the literature, for example in [SLE15].

approximation in terms of sparse vectors $\boldsymbol{\gamma}_i$ and dictionary $\mathbf{D}$.

Noticing the similarities with the dictionary learning problem studied above, this is a generalization of it where all training examples come from a real (high-dimensional) image. Therefore, one can employ the same alternating minimization approach to approximate the solution of this problem as well. In practice, this reduces to taking all overlapping patches $\mathbf{P}_i\mathbf{x}$, and then iterating between sparse coding and dictionary updates. Once the iterates converge, the patches are finally merged back together by averaging. When the update of the dictionary is done with an SVD step, this results in the K-SVD denoising algorithm [EA06].

Is this optimal? Why should we reconstruct the global image only once? what are really the model assumptions imposed on the global image $\mathbf{x}$? We will gradually answer these questions in the coming chapters.

Chapter 3

Upgrading Local Methods

Chapter Abstract

Over the last decade, a number of algorithms have shown promising results in removing additive white Gaussian noise from natural images, and though different, they all share in common a patch based strategy by locally denoising overlapping patches. While this decreases the complexity of the problem, it also causes noticeable artifacts when dealing with large smooth areas. In this chapter we present two different patch-based denoising algorithms relying on a sparsity-inspired model (K-SVD) that significantly alleviate these problems. The first one employs a multi-scale analysis framework, in which we look for a sparse representation under an already sparsifying wavelet transform by adaptively training a dictionary on the different decomposition bands of the noisy image itself, leading to a multi-scale version of the K-SVD algorithm. The second approach focuses on a relatively recent idea, the Expected Patch Log Likelihood (EPLL). This framework argues that the chosen model should be enforced on the final reconstructed image patches, and not just on the intermediate ones. We will show how to combine the EPLL with a sparse-representation prior, and our derivations will lead to a close yet extended variant of the popular K-SVD image denoising algorithm. Finally, we study the global properties of the denoising operator resulting from the GMM denoising algorithm, and we leverage it to employ a Laplacian regularization. This method shares similarities with variational approaches, and can be thought of as a generalization of the non-linear diffusion algorithm. In all cases, we will see how local patch-based algorithms can be boosted to better serve global restoration problems.

3.1 Local Priors and Image Restoration

Inverse problems in image processing consist of recovering an original image that has been degraded. Denoising, debluring and inpaiting are specific and common such examples. Put formally, these problems attempt to recover an underlying image $\mathbf{x}$ given the measurement $\mathbf{y}$ such that

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \tag{3.1}$$

where $\mathbf{A}$ is a known linear operator and $\mathbf{n}$ represents measurement noise. In dealing with this problem, it is common to work with image priors as regularizers and develop a Maximum a Posteriori (MAP) estimator for the unknown image $\hat{\mathbf{x}}$. This can be formulated as an optimization problem where we look for an estimate which is close enough to the measured image while being likely under this prior. Most state of the art methods employ, either implicitly of explicitly, some prior knowledge in the form of smoothness [TM98], self-similarity [DFKE06], sparsity [EA06], and combinations of some of these [MBS09, PSWS03]. Learning specifics priors from real data, by adapting them to the image and problem at hand, has shown to enable better performance under this approach [MBPS09, RB09]. However, this learning process is computationally expensive and it is usually restricted to small dimensions, which leads naturally to the modeling of small image patches [AEB06, WF07].

When deploying a sparse enforcing prior, as briefly mentioned at the end of Chapter 2, this can be done in terms of the following optimization problem,

$$\min_{\mathbf{x},\{\boldsymbol{\gamma}_i\},\mathbf{D}} \quad \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \frac{\lambda}{2}\sum_i ||\mathbf{D}\boldsymbol{\gamma}_i - \mathbf{P}_i\mathbf{x}||_2^2 + \mu_i||\boldsymbol{\gamma}_i||_0.$$

This expression seeks for an underlying image $\mathbf{x}$ that would be close to $\mathbf{y}$ if one applies the degradation operator $\mathbf{A}$, and such that every patch from it, $\mathbf{P}_i\mathbf{x}$, can be expressed as $\mathbf{D}\boldsymbol{\gamma}_i$ for a sparse vector $\boldsymbol{\gamma}_i$. We make use of the linear operator $\mathbf{P}_i$ that extracts a small image patch of length n (typically 8×8 - 11×11) from an image of dimension N. In addition, one minimizes the expression with respect to the dictionary $\mathbf{D}$ as well, adapting the model to the data.

This problem is highly non-smooth and non-convex. The popular K-SVD algorithm [EA06] minimizes the expression above by taking an alternating minimization approach: first fixing the dictionary and the target image and optimizing with respect to the sparse vectors $\boldsymbol{\gamma}_i$. Then, keeping these fixed and updating the dictionary $\mathbf{D}$. One might repeat this process several times, until fixing those and updating the image $\mathbf{x}$. The loss above is quadratic with respect to $\mathbf{x}$, and so this step simply amounts to aggregating the image patches and averaging with appropriate weights.

This local strategy, while very effective both in terms of computational complexity and restoration performance, is known to cause significant artifacts in the estimated images. In fact, these problems are not unique to sparse-enforcing methods as they appear in practically all restoration method that employ a patch-based approach. This can be clearly seen in Figure 3.1, were we depict a natural image contaminated with Gaussian noise of standard deviation of 35,

Figure 3.1: One image from the dataset in [oCD] contaminated with Gaussian noise of $\sigma = 35$ (left), and the denoising results by the state-of-the-art method of BM3d (right) [DFKE07] with PSNR = 33.16.

and the results obtained by the state-of-the-art algorithm BM3D [DFKE07]. As one can see, despite the PSNR measure being considerably good (33.16 dB), the resulting image contains severe texture-like artifacts, particularly noticeable in smooth areas of large images – which, ironically, are often not appreciated in the smaller and popular images used in computer vision community. These artifacts are mainly caused by a lack of agreement of the local (per patch) estimates of the image value at that locations. This is the problem we intend to solve in the works presented in this Chapter.

3.2 Multi-Scale Sparse Modeling and Restoration

An appealing direction to *globalize* patch-based algorithms is through a multi-scale approach. Indeed, working at different scales can provide a broader analysis of image patches, forcing them to consider more global information. This is the approach we took in [SOE14], where we proposed to merge the K-SVD denoising algorithm [EA06] with a wavelet analysis, in a similar way to the approach taken in [OLE11]. This leads to an effective sparse decomposition of the image content using different scale atoms in a natural way. As a result, the potential of the K-SVD denoising algorithm is exploited beyond the single scale limitations, reaching state of the art results. In this section we describe this idea in detail, tie it and contrast it to existing work, and demonstrate the effectiveness of the proposed scheme.

3.2.1 Related Work

The idea of combining dictionary learning with a multi-scale analysis framework is not new. In [SO02], the authors proposed to train wavelet coefficients with a sparsity inducing prior on a wavelet pyramidal decomposition structure, achieving slightly better results for compression. Later, the authors in [MSE07] used different size patches taken from a quadtree structure to train a multi-scale dictionary, in a first extension of the K-SVD algorithm to a multi-scale scheme.

The work presented in [OLE11] introduced the construction of true multi-scale dictionaries by learning patch based atoms in the analysis domain of the wavelet transform. In this case, the resulting dictionary appears as the multiplication of a wavelet synthesis matrix with a learnt dictionary in the wavelet domain, i.e., $\tilde{\mathbf{D}} = \mathbf{W_S D}$, where $\mathbf{W_S}$ is the synthesis matrix of a wavelet (inverse) transform. However, choosing an orthogonal wavelet with periodic extension enables the authors to work in the analysis domain instead, by solving the following optimization problem:

$$\min_{\mathbf{D},\mathbf{x}} ||\mathbf{W_A y} - \mathbf{D\Gamma}||_F^2 \text{ subject to } ||\boldsymbol{\gamma}_i||_0 \leq k, \ \forall i,$$

where $\mathbf{W_A}$ is the analysis operator (wavelet transform) matrix. This expression suggests to adapt the atoms to sparsely represent the wavelet coefficients of the different training examples. In this sense, the expression represents a slight abuse of notation as $\mathbf{W_A y}$ denotes small dimensional patches taken from the wavelet coefficients of the image $\mathbf{y}$, arranged column-wise. Moreover, the authors proposed to train different sub-dictionaries $\mathbf{D_b}$ per band by employing K-SVD on 8×8 patches of the wavelet sub images. This simple scheme allows to work with different sized atoms, since a patch in a first decomposition level implies an effective patch of four times its area in the image domain. Once the collection of sub-dictionaries is trained, the authors in [OLE11] use a *global* framework for the sparse coding stage, where the patches from different scales *compete* for additional coefficients selecting the one that gives the most profit in terms of the residual energy, with a global variant of the OMP algorithm.

All these approaches have looked for a better representation of some class of data or images in terms of some dictionary. As such, they fail to treat the denoising task competitively, as indeed demonstrated in [OLE11]. In [EA06], the K-SVD denoising algorithm was formally derived by proposing a global image prior that forces patch-based local sparsity over patches in every location of the image. The problem is solved iteratively using an error threshold for the sparse coding which depends on σ, the noise standard deviation, treating each patch *independently*. We will make use of this concept and extend it to a multi-scale framework.

3.2.2 Our Contribution

In this work we propose to continue and extend the work in [OLE11], and tackle specifically the denoising problem. In [OLE11] the authors have shown an example of a naive denoising through *M-term* approximation, using a global pursuit. The results reported in this method were not competitive with the single-scale K-SVD. In this section we propose to adapt the multi-scale sub-dictionaries to the noisy image itself and treat the pursuit locally. This resembles the work in [EA06], but in a multi-scale scenario. Each band of the decomposition is treated separately, training a subdictionary for each band, which is then used to denoise the corresponding wavelet coefficients. In a final stage, the multi-scale K-SVD and the traditional (single-scale) K-SVD denoised images are combined though a weighted joint sparse coding in order to benefit from the advantages that each bring. This last step allows us to maximize the information shared between the two images, and obtain a better estimate for the original signal.

3.2.3 Multi-scale K-SVD Denoising

Consider a noisy image $\mathbf{y}$, its wavelet transform as a collection of band images $\mathbf{y}_b^W = (\mathbf{W}_A\mathbf{y})_b$, and its estimated denoised versions $\hat{\mathbf{x}}_b^W$, $b = 1, ..., L = 3S + 1$, with S decomposition levels. Generalizing the work in [EA06], we propose a global maximum *a posteriori* (MAP) estimator for denoising the image in the wavelet domain as

$$\min_{\boldsymbol{\gamma}_{i,b},\mathbf{D}_b,\mathbf{x}_b^W} \quad \frac{\lambda}{2}||\mathbf{y}_b^W - \mathbf{x}_b^W||_2^2 + \sum_i \mu_{i,b}||\boldsymbol{\gamma}_{i,b}||_0 + \sum_i ||\mathbf{D}_b\boldsymbol{\gamma}_{i,b} - \mathbf{P}_{i,b}\mathbf{x}_b^W||_2^2, \quad \forall b$$

where $\mathbf{x}_{i,b}$ is the sparse vector for the $(i)-$patch in the decomposition band b, $\mathbf{P}_{i,b}$ the patch-extraction operator acting on the sub-image $\mathbf{x}_b^W$, and λ is a penalty parameter. This optimization problem can be solved iteratively by first considering a fixed set of dictionaries $\mathbf{D}_b$ and obtaining the vectors $\boldsymbol{\gamma}_{i,b}$ by any pursuit method. Then the sub dictionaries are updated using a K-SVD step. These steps are repeated for a fixed number of iterations. Finally, we update $\mathbf{x}_b^W$ by

$$\hat{\mathbf{x}}_b^W = \left(\lambda\mathbf{I} + \sum_i \mathbf{P}_{i,b}^T\mathbf{P}_{i,b}\right)^{-1}\left(\lambda\mathbf{y}_b^W + \sum_i \mathbf{P}_{i,b}^T\mathbf{D}_b\boldsymbol{\gamma}_{i,b}\right).$$

After the different sub band images have been denoised in the wavelet domain, the multi-scale denoised image is obtained by applying the inverse wavelet transform. Note that by working on patches of the same size in all decomposition levels, we consider different-scale *effective* patches in the image domain. This gives our algorithm a more global outlook than that of the regular K-SVD denoising algorithm, and involves essentially the same computational complexity, plus the forward and backward wavelet transform. The complexity analysis detailed in [OLE11] is still valid here.

3.2.4 Fusing Single and Multi-Scale Results

After this multi-scale K-SVD denoising stage, we go one step further. While working on the wavelet coefficients on the different scales $1, 2, \ldots, S$, we miss considering the *scale 0*. Following this motivation, we propose to merge the outcome of the original (*single-scale*) K-SVD denoised image $\hat{\mathbf{x}}_{ss}$ with the output of the *multi-scale* K-SVD algorithm proposed here, $\hat{\mathbf{x}}_{ms}$. Both of these have some remaining noise and different artifacts, but correspond to the same underlying image. We aim to recover the information common to both of them by a weighted joint sparse coding, as motivated by [YQR13] and shown in Fig. 3.2. Consider each patch in the single scale and multi-scale images given by $\mathbf{y}_{ss}$ and $\mathbf{y}_ms$, respectively. We concatenate corresponding patches of both images with a weighting factor β as $\tilde{\mathbf{y}} = [\ \mathbf{y}_{ms}^T\sqrt{1+\beta}\ ,\ \mathbf{y}_{ss}^T\sqrt{1-\beta}\ ]^T \in \mathbb{R}^{2n}$. We may then use the dictionary given by $\mathbf{A} = [\ \mathbf{D}^T\sqrt{1+\beta}\ ,\ \mathbf{D}^T\sqrt{1-\beta}\ ]^T \in \mathbb{R}^{2n\times m}$ to obtain the sparse vector $\boldsymbol{\alpha} \in \mathbb{R}^m$ by the OMP algorithm. Finally, the denoised patch will be given by $\hat{\mathbf{z}} = \mathbf{D}\boldsymbol{\alpha}/\sqrt{2}$, in order to preserve the initial energy.

As we will see later, the multi scale K-SVD algorithm outperforms the single scale K-SVD specially in the presence of high noise due to the increasing patch-like artefacts, which the multi

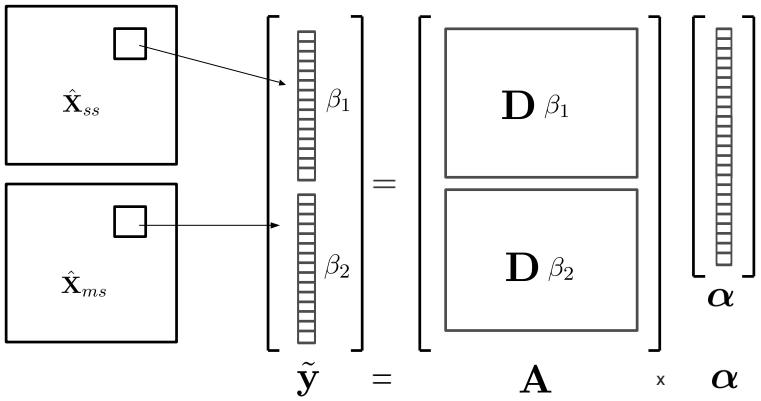


Figure 3.2: Joint sparse coding stage of the Fused K-SVD denoising algorithm. $\beta_1 = \sqrt{1+\beta}$, $\beta_2 = \sqrt{1-\beta}$.

scale approach is more robust to. This indicates that β should be close to 1 in such cases, and close to 0 when the noise level is lower. One could just propose a function $\beta = f(\sigma_\eta)$ accordingly, or choose an adaptive method that optimizes this parameter for each patch. For the sake of simplicity we consider here a linear function of the initial noise level, from $\beta = 0$ for $\sigma = 0$ to $\beta = 0.9$ for $\sigma = 50$. Certainly other choices are possible, and the implications of this choice will be commented later on.

3.2.5 Experiments

In this section we present the results of a denoising experiment on landscape images from the online NOAA library [oCD]. We chose these images as they contain large scenery areas that are poorly treated by typical patch-based denoising methods. One of this images is depicted in the top left corner of Fig.3.4. Fifteen images from this dataset, size 870×1360, were contaminated by white Gaussian noise with zero mean and variable standard deviation σ. For the multi-scale decomposition we used a discrete Meyer wavelet, with 2 decomposition levels. By choosing a unitary transform, the stoping criteria for the sparse coding stage in the denoising algorithm is simply $\epsilon = c \cdot \sigma$, where $c = 1.15$ following [EA06].

We evaluate our denoising results with two image quality measures: the popular Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [WBSS04]. While simple and practical, the PSNR relies only on the absolute difference pixel by pixel, and does not provide a good signal fidelity measure [WB09]. As such, its ability to compare images from a human perception point of view is poor. The SSIM is somehow a more complete image quality measure, which builds upon the idea that human perception is highly adaptive to structural information from images and visual scenes [WBSS04]. We include in the results those obtained by the BM3D algorithm [DFKE06], computed with the code made available by the authors, and with their recommended parameters. We also compare our performance against the regular single-scale

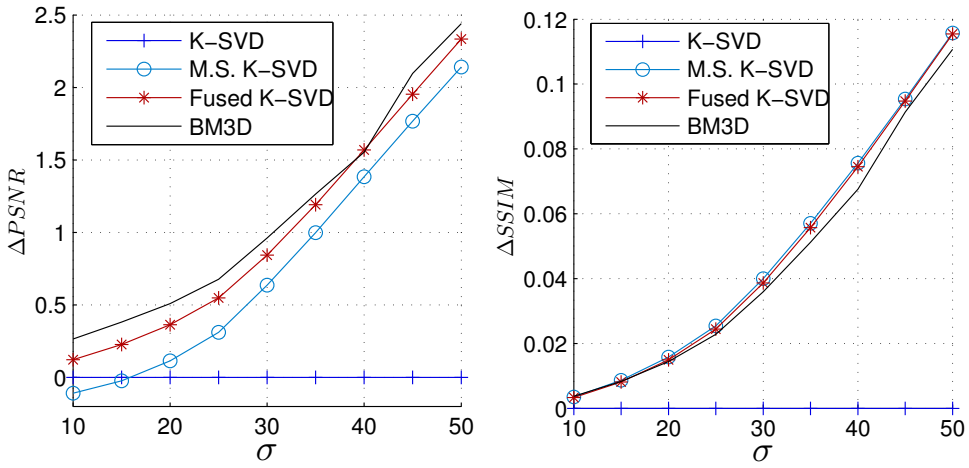


Figure 3.3: Denoising results by the K-SVD [EA06], BM3D [DFKE06], multi-scale K-SVD and Fused K-SVD algorithms, averaged over 15 testing images from the NOAA database [oCD]. Left: PSNR gain with respect to K-SVD. Right: SSIM gain with respect to K-SVD.

K-SVD. Note that all three methods use 8×8 patches.

We may also benefit from choosing an appropriate initial dictionary [EA06]. To this end, we trained a single-scale and a multi-scale dictionary on 20 natural images (outside the above set of test images), for the single-scale and multi-scale versions of the K-SVD algorithm, respectively. The same single scale initial dictionary was later used to merge the final outcome of the *Fused K-SVD* algorithm, as described in the previous section. In this case we use OMP with an error threshold of $\epsilon = 0.1 \cdot \sigma_\eta$, where this factor has been chosen empirically, accounting not only for the remaining noise but also for the difference in the artefacts of the two images.

In Fig. 3.3 we present the averages over all testing images for the different algorithms, relative to that of K-SVD. The multi-scale K-SVD outperforms the single-scale K-SVD in almost the whole range of noise variance, and the Fused K-SVD and BM3D present the best results, with the latest being slightly higher in terms of PSNR. Note that the last weighted joint sparse coding stage enables an extra boost, and the full fused algorithm improves the results by 0.2 - 0.3 dB compared to the plain multi-scale K-SVD. Turning to the SSIM results, the artefacts on the smooth areas in the regular K-SVD denoised images are strongly penalized by this measure. Multi-scale K-SVD and Fused K-SVD seem to be the best, with our methods slightly outperforming BM3D. Fusion gives no gain with respect to this measure. In Fig. 3.4 we depict the results of the K-SVD, BM3D and Fused K-SVD on the example image.

The reason for this difference in both measures should not be surprising. While BM3D makes little mistakes in terms of absolute value, these errors are more *noticeable* when there are large smooth areas, which causes the annoying texture artefacts that can be seen in the images in Fig. 3.4. It is in these areas where our method shows its greatest benefits. The coding of the deeper decomposition levels implies choosing big atoms yielding nicely codded smooth patches. These atoms are treated considering a more global approach than just looking at a 8×8 patch in the

Figure 3.4: One image from the dataset in [oCD], and its denoising results (noise level $\sigma_\eta = 35$). Top left: original image. Top right: K-SVD (PSNR = 31.01, SSIM = 0.857). Bottom left: BM3D (PSNR = 33.16, SSIM = 0.923). Bottom right: Fused K-SVD algorithm (PSNR = 33.16, SSIM = 0.940). Note the artefacts in the single scale patch based methods.

image domain. This makes the method more robust to higher noise levels, where the texture artefacts become stronger. However, this advantage comes at the cost of losing some details in the sharp edges of the image. The absolute error at these points are slightly higher that those made by BM3D, as noted by the PSNR results.

To finish this section, we have a word about the standard images such us *Lena, Barbara, etc.* In these cases the performance of the Fused K-SVD algorithm is between 0.3-0.4 dB (PSNR) and 0.002-0.01 (SSIM) lower than BM3D. Note that these images are small (512×512) and hardly present any smooth areas of considerable size. Even in these images, however, there is a notable improvement over the regular K-SVD in both measures (up to 0.55 dB in PSNR and 0.035 in SSIM).

3.3 Expected Patch Log Likelihood

We now leave behind the multi-scale analysis approach, and focus in the formulation of the restoration problem instead. As we saw, image restoration methods typically work by breaking the image into small overlapping patches, solving their MAP estimate, and tiling the results

Figure 3.5: Sub images extracted from the denoising results of the image shown in Fig. 3.4, by the K-SVD [EA06] (left), the BM3D (center) and by the fused K-SVD method (right).

back together by averaging them [EA06, DFKE06, BCM05]. Recently, Zoran and Weiss [ZW11] proposed a general framework based on the simple yet appealing idea that the *resulting final* patches should be likely under some specific prior, and not the intermediate ones. Their approach is based on maximizing the *Expected Patch Log Likelihood* (EPLL) which yields the average likelihood of a patch on the final image under some prior. This idea is general in the sense that it can be applied to any patch-based prior for which a MAP estimator can be formulated. In particular, the authors in [ZW11] employed the classic Gaussian Mixture Model prior achieving state of the art results in image denoising and deblurring.

As we have discussed in Chapter 2, the idea that a natural signal or image patch can be well represented by a linear combination of a few atoms from a dictionary is a very strong prior. This leads to the natural question, could we use the EPLL framework with a sparsity-inspired prior? If so, how is this related to existing methods that explicitly target this problem and what is there to gain from this approach? In this section we explore and formally address these questions, showing that indeed benefit can be found in employing EPLL with a patch sparsity-based prior.

We thus begin this section by briefly reviewing the EPLL framework as described in [ZW11].

Given an image $\mathbf{x}$, the Expected Patch Log Likelihood under some prior p is defined as

$$EPLL_p(\mathbf{x}) = \sum_i log\ p(\mathbf{P}_i\mathbf{x}),$$

where $\mathbf{P}_i$ extracts the i^{th} patch from $\mathbf{x}$. Therefore, given the corruption model in Eq. (3.1) we can propose to minimize the following cost function:

$$f_p(\mathbf{x}|\mathbf{y}) = \frac{\lambda}{2}||\mathbf{A}\mathbf{x} - \mathbf{y}||_2^2 - EPLL_p(\mathbf{x}),$$

where the first term represents the log likelihood of the image. To get around the hard optimization of this function, the authors in [ZW11] propose to use a *Half Quadratic Splitting* strategy by defining auxiliary patches $\{\mathbf{z}^i\}$ for each patch $\mathbf{P}_i\mathbf{x}$, and then minimizing

$$c_{p,\beta}(\mathbf{x}, \{\mathbf{z}^i\}|\mathbf{y}) = \frac{\lambda}{2}||\mathbf{A}\mathbf{x} - \mathbf{y}||_2^2 + \sum_i \frac{\beta}{2}||\mathbf{P}_i\mathbf{x} - \mathbf{z}^i||_2^2 - log\ p(\mathbf{z}^i) \tag{3.2}$$

iteratively, while increasing the value of β. Note that for $\beta \to \infty$, $\mathbf{z}^i \to \mathbf{P}_i\mathbf{x}$, so this parameter controls the distance between the auxiliary patches and the patches of the image $\mathbf{x}$. For a fixed value of β, the cost function is again broken into a two step inner minimization: first fix $\{\mathbf{z}^i\}$ and solve for $\mathbf{x}$ by

$$\mathbf{x} = \left(\lambda\mathbf{A}^T\mathbf{A} + \beta\sum_i \mathbf{P}_i^T\mathbf{P}_i\right)^{-1}\left(\lambda\mathbf{A}^T\mathbf{y} + \beta\sum_i \mathbf{P}_i^T\mathbf{z}^i\right). \tag{3.3}$$

Then, fix $\mathbf{x}$ and solve for $\{\mathbf{z}^i\}$ by solving the MAP estimate for each patch under the prior in consideration. This process should be repeated 4-5 times, before increasing β and repeating the whole process again. Each time, the patches are taken from *the image estimate* at each iteration.

Within the EPLL scheme, the choice of β is crucial. In [ZW11] the authors set this parameter manually to be $\frac{1}{\sigma^2}[1, 4, 8, 16, 32, \dots]$, where σ is the noise standard deviation. In the same work it is also suggested that β could be determined as $\beta = \frac{1}{\sigma^2}$, where σ is estimated in every iteration by an *off-the-shelf* white Gaussian noise estimator.

3.4 EPLL with a Sparse Prior

In the original formulation, Zoran and Weiss propose to use a Gaussian Mixture Model (GMM) prior which is learnt off-line from a large number of examples. In their case, the MAP estimator for each patch is simply given by the Wiener filter solution for the Gaussian component with the highest conditional weight [ZW11]. However, the EPLL approach is a generic framework for potentially any patch-based prior. We now turn to explore the formulation of an equivalent problem with a sparsity inducing prior.

3.4.1 Cost function formulation

Consider the signal $\mathbf{z} = \mathbf{D}\boldsymbol{\gamma}$, where $\mathbf{D}$ is a redundant dictionary of size $n \times m$ ($n < m$), and the vector $\boldsymbol{\gamma}$ is sparse; i.e., $||\boldsymbol{\gamma}||_0 \ll n$, where the l_0 pseudo-norm $||\cdot||_0$ basically counts the non zero elements in $\boldsymbol{\gamma}$. Assuming that this is the model we impose on our patches $\mathbf{z}^i$, Eq. (3.2) becomes

$$c_{\mu,\beta}(\mathbf{x}, \{\boldsymbol{\gamma}_i\}|\mathbf{y}) = \frac{\lambda}{2}||\mathbf{A}\mathbf{x} - \mathbf{y}||_2^2 + \sum_i \frac{\beta}{2}||\mathbf{D}\boldsymbol{\gamma}_i - \mathbf{P}_i\mathbf{x}||_2^2 + \mu_i||\boldsymbol{\gamma}_i||_0. \tag{3.4}$$

In this case, μ_i reflects the trade-off between the accuracy of the representation and the sparsity of $\boldsymbol{\gamma}_i$. For the case $\beta = 1$, this last expression corresponds exactly to the formulation of the K-SVD denoising algorithm in [EA06], where $\mathbf{A} = \mathbf{I}$. In this work, Elad and Aharon proposed to use a block-coordinate minimization that starts by fixing $\mathbf{x} = \mathbf{y}$, and then seeking the optimal $\boldsymbol{\gamma}_i$ solving the MAP estimator for each patch:

$$\hat{\boldsymbol{\gamma}}_i = \arg\min_{\boldsymbol{\gamma}} \mu_i||\boldsymbol{\gamma}_i||_0 + ||\mathbf{D}\boldsymbol{\gamma}_i - \mathbf{P}_i\mathbf{x}||_2^2. \tag{3.5}$$

Though this problem is NP-hard in general, its solution can be well approximated by greedy or pursuit algorithms [DET06]. In particular, the Orthogonal Matching Pursuit (OMP) [Tro04] can be used with the noise energy as an error threshold to yield an approximation of the solution to Problem (3.5), and we employ this method in our work due to its simplicity and efficienty [RZE08]. This way, μ_i is handled implicitly by replacing the second term by a constraint of the form

$$\min_{\boldsymbol{\gamma}} ||\boldsymbol{\gamma}||_0 \;\text{ subject to }\; ||\mathbf{D}\boldsymbol{\gamma} - \mathbf{P}_i\mathbf{x}||_2^2 \leq nc\sigma^2, \tag{3.6}$$

where c is a constant factor set to 1.15 in [EA06]. Given the estimated sparse vectors $\{\hat{\boldsymbol{\gamma}}_i\}$, the algorithm proceeds by updating for the unknown image $\mathbf{x}$ which results in an equivalent expression to that in Eq. (3.3) - for a specific value of β. When denoising is done locally (training the dictionary on the corrupted patches) the dictionary gets updated together with the sparse vectors by using a K-SVD step. This adaptive method that trains the dictionary on the noisy image itself has proven to be better than using a dictionary trained offline.

The initial claim in [EA06] is that the above block-coordinate minimization should be iterated. In practice, however, repeating this process is problematic since after updating $\mathbf{x}$, the noise level has changed and it is spatially varying. Therefore, the sparse coding stage has no known thresholds to employ. Thus, the algorithm in [EA06] does not iterate after updating $\mathbf{x}$.

Increasing β, as practiced in [ZW11], forces the distance $||\mathbf{D}\boldsymbol{\gamma}_i - \mathbf{P}_i\mathbf{x}||_2$ to be smaller. Therefore, iterating the above algorithm for increasing values of β is equivalent to iterating the process described for the K-SVD with smaller thresholds. As we see, the algorithm proposed in [EA06] applies only the first iteration of the EPLL scheme with a sparse-enforcing prior, therefore losing important denoising potential. A synthetic example is shown in Fig. 3.6, where we compare the algorithms in [EA06] and [ZW11] with the method proposed in this section.

We now turn to address the matter of the threshold design for later stages of the K-SVD in order to practice the EPLL concept in an effective way.

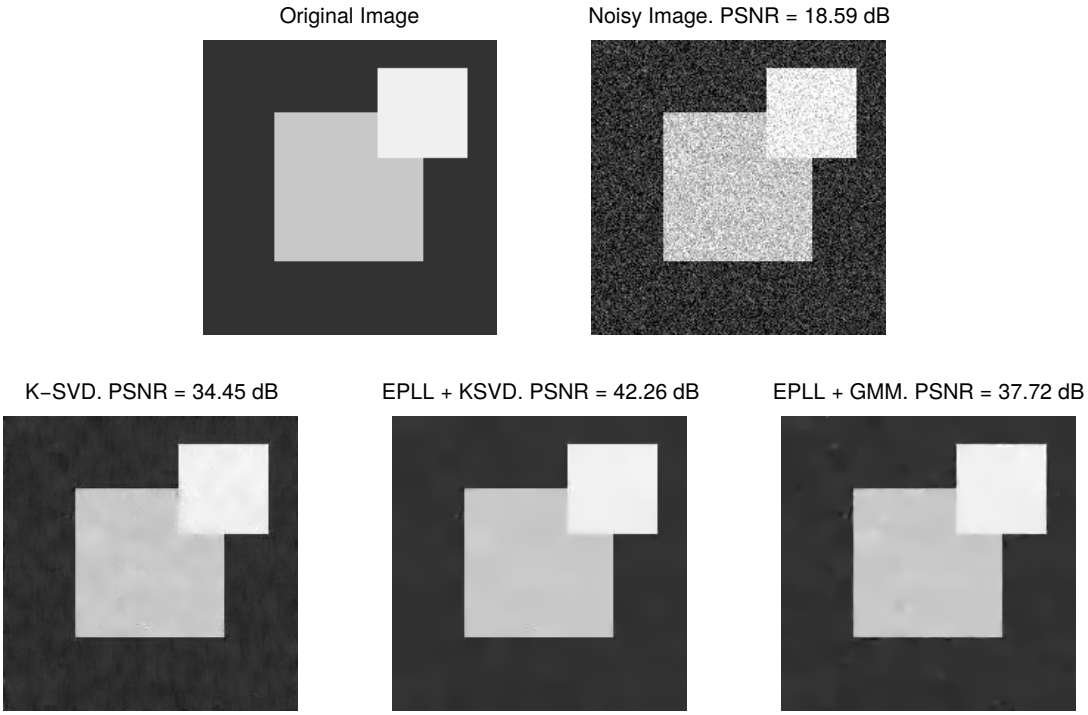


Figure 3.6: Denoising of a synthetic image ($\sigma = 30$). A similar demonstration was presented in [ZW11], showing the benefits of the EPLL framework under a GMM approach. Note the texture-like resulting artifacts in the result by K-SVD. This problems is notably reduced by the EPLL with a Sparse Prior, the method we present in this work. We include for comparison the result by [ZW11]. The evolution of the Peak Signal to Noise Ratios are depicted in Fig 3.9.

3.4.2 Sparse coding thresholds

Consider the threshold in the sparse coding stage, at each iteration k, to be ν_k^2. Naturally, in the first iteration of the process that aims to minimize Eq. (3.4) we set this threshold to be exactly the noise energy σ^2 for all patches; i.e. $\nu_1^2 = \sigma^2$. In the following iterations, however, instead of trying to estimate the remaining noise with an *of-the-shelf* algorithm, we propose an intrinsic alternative by using the information we already have about each patch.

Consider the general problem of estimating the remaining noise after applying K-SVD on the noisy image; i.e., the first iteration of our method. From a global perspective, the estimated image can be expressed as

$$\hat{\mathbf{x}} = \left(\lambda \mathbf{A}^T\mathbf{A} + \sum_i \mathbf{P}_i^T\mathbf{P}_i\right)^{-1}\left(\lambda \mathbf{A}^T + \sum_i \mathbf{P}_i^T\mathbf{D}_{S_i}\mathbf{D}_{S_i}^{+}\mathbf{P}_i\right)\mathbf{y},$$

where S_i denotes the support of the sparse vector $\hat{\boldsymbol{\gamma}}_i$ chosen in the OMP, and $\mathbf{D}_{S_i}$ is the set of the corresponding atoms in the dictionary. Leaving aside the selection of the support of each

sparse vector, we can represent this operation by a linear operator as

$$\hat{\mathbf{x}} = \mathbf{L}(\mathbf{x} + \mathbf{n}). \tag{3.7}$$

Assuming for a moment that $\mathbf{x} \approx \mathbf{L}\mathbf{x}$, we could express the remaining noise as $\mathbf{n}_r = \mathbf{L}\mathbf{n}$, from which we could obtain the full covariance matrix as $Cov(\mathbf{n}_r) = \sigma^2\mathbf{L}\mathbf{L}^T$. Then, we could either take into consideration the full covariance matrix, or make the simplifying assumption of white noise by considering just the diagonal of $Cov(\mathbf{n}_r)$. Though appealing, this approach does not work in practice because $||\hat{\mathbf{x}} - \mathbf{L}\mathbf{x}||_2$ is considerably large, and thus the estimate of the remaining noise is considerably low. Also, note that $\mathbf{L}$ is a banded matrix of size $N^2 \times N^2$, where N is the number of pixels, and so the estimation of its covariance matrix is computationally intractable for practical purposes.

We thus turn to a similar but local alternative that will enable a practical solution. Each patch consists of the true underlying vector $\mathbf{z}_{0i}$ and a noise component $\mathbf{v}_i$, $\mathbf{z}_i = \mathbf{z}_{0i} + \mathbf{v}_i$. Given the chosen support S_i, $\hat{\mathbf{z}}_i$ is obtained as a projection onto the span of the selected atoms:

$$\hat{\mathbf{z}}_i = \mathbf{D}_{S_i}\mathbf{D}_{S_i}^{+}\mathbf{z}_i = \mathbf{D}_{S_i}\mathbf{D}_{S_i}^{+}\left(\mathbf{z}_{0i} + \mathbf{v}_i\right).$$

Assuming now that $\mathbf{z}_{0i} \approx \mathbf{D}_{S_i}\mathbf{D}_{S_i}^{+}\mathbf{z}_{0i}$ (if the correct support of the signal was chosen by the OMP), the contribution of the noise to the patch estimate would be given by $\hat{\mathbf{v}}_i^r = \mathbf{D}_{S_i}\mathbf{D}_{S_i}^{+}\mathbf{v}_i$. This is an analogue assumption to that made for Eq. (3.7), but now for each patch instead of the global image. This way, considering the covariance matrix of the remaining noise $Cov(\hat{\mathbf{v}}_i^r)$, the mean squared error estimate at the i^{th} patch and iteration k will be given by $\frac{1}{n}tr\{Cov(\hat{\mathbf{v}}_i^r)\}$, leading to

$$\left(\hat{\sigma}_i^k\right)^2 = |S_i|\frac{\nu_k^2}{n}.$$

Therefore, the estimate of the remaining noise in each patch is simply proportional to the number of atoms used for that patch. Of course, the remaining noise is no longer white after the back projection step, but we make this assumption in order to simplify further derivations.

Generalizing this patch analysis to the entire image, we can estimate the average remaining noise in the image $\mathbf{x}$ by performing an estimate in the spirit of Eq. (3.3), tilling back and averaging the local estimates as

$$\mathbf{R}^k = \frac{\lambda\nu_k^2\mathbf{I} + \sum_i \mathbf{P}_i^T\mathbf{1}\left(\hat{\sigma}_i^k\right)^2}{\lambda\nu_k^2\mathbf{I} + \sum_i \mathbf{P}_i^T\mathbf{P}_i} = \Phi\left((\hat{\sigma}_i^k)^2\right), \tag{3.8}$$

where the operator $\Phi(\cdot)$ relocates the local estimates $\hat{\sigma}_i^k$ with the corresponding weighting. This way, $\mathbf{R}^k$ stands for an estimation of the energy of the remaining noise pixel-wise, equivalent – but not equal, due to our simplifying assumptions – to the diagonal of $Cov(\mathbf{n}_r)$. An example is shown in Fig. 3.7 for the popular image Lena. We see that $\mathbf{R}^k$ provides a fair estimate of the information in the diagonal of the full covariance matrix of the remaining noise $Cov(\mathbf{n_r})$, and

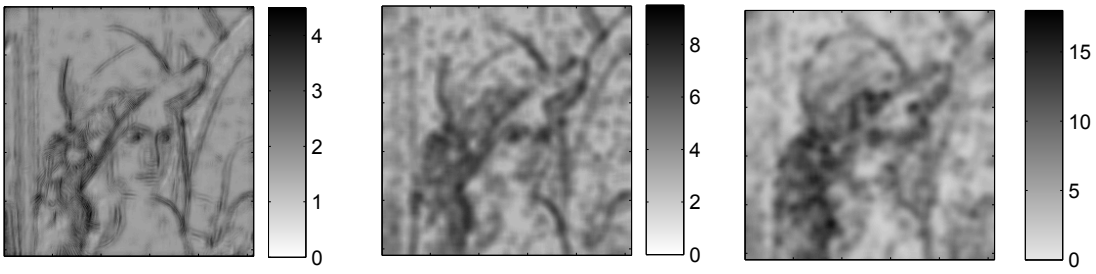


Figure 3.7: Left: plot of the diagonal of the covariance matrix $Cov(\mathbf{n_r})$ after the first iteration of denoising the image Lena ($\sigma = 20$). Center: the corresponding plot of the estimated $\mathbf{R}^k$ in Eq. (3.8), and right: the corresponding average of the standard deviation per patch of the true error image.

that it is closer to the average of the standard deviation per patch of the true error image. The reader should also note that computing $\mathbf{R}^k$ is considerably cheaper than the computation of the operator in (3.7), since we only compute the local covariance matrices and their weighted average, and the matrix in the denominator of Eq. (3.8) is a diagonal one. Therefore we use $\mathbf{R}^k$ to derive the threshold for the next iteration.

From this point two possibilities arise: use $\mathbf{R}^k$ to evaluate a local patch-based noise energy, eventually denoising each patch with a different threshold, or finding a new global and common threshold for all the patches. The first option, while elegant, is slightly more complex, as you cannot benefit from fast (batch) sparse coding implementations that require all signals to employ the same error threshold. In addition, this option was found not to yield significant improvements when compared to the second and simpler approach. Thus, in the following we adopt the later global alternative.

The reader should bare in mind that the thresholds should tend to zero as we iterate, corresponding to $\beta \to \infty$. Certainly, this implies that our thresholds will not reflect the *real* remaining noise. As an example, in Fig. 3.8 we present the evolution of the PSNR by the proposed method for the image Lena for different thresholds. We see that if the threshold is not changed with the iterations, the PSNR of the resulting image $\mathbf{x}$ decreases after the first iteration. On the other hand, if we set the threshold to be the variance of the real remaining noise (by having access to an oracle and the original image), the PSNR initially increases but eventually decreases since the threshold do not tend to zero. We include for comparison the results of our threshold-setting method.

This way, in what follows we propose to use an heuristic that provides decreasing thresholds and which has been proven to be robust. In the subsequent iterations, we set the threshold ν_k^2 to be the mode of the values in $\mathbf{R}^k$. Furthermore, we have found that the multiplication by a constant factor δ improves the performance in our method. To this end, assuming independence between the remaining noise and the patch estimate, and considering the residual per patch

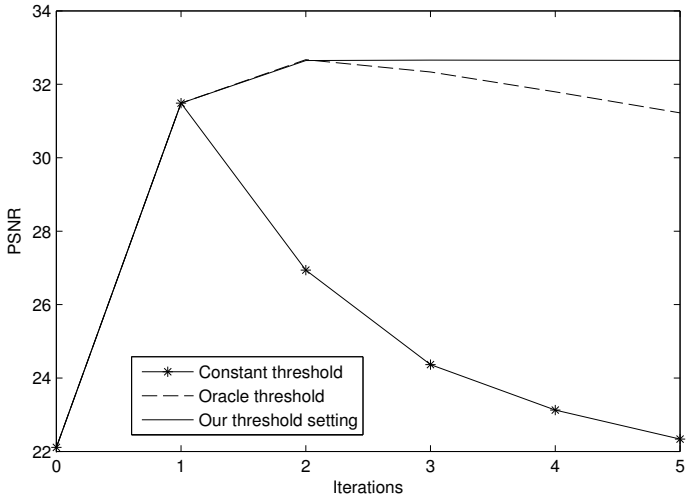


Figure 3.8: PSNR evolution of the EPLL scheme with a sparse-representation prior for denoising the image Lena ($\sigma = 20$) and three different threshold settings: a) using a constant threshold for all the iterations (equal to the initial noise energy σ^2); b) using an oracle threshold by setting it to be the variance of the real error image (having access to the original image); and c) our threshold setting method.

$\mathbf{r}_i = \mathbf{z}_i - \hat{\mathbf{z}}_i$, we have that $\tilde{\sigma}_i^2 = \sigma^2 - Var(\mathbf{r}_i)$[1]. With these estimates we can perform an analogue of Eq. (3.8) and obtain its mode, $\tilde{\nu}^2$. We then define the factor $\delta = \tilde{\nu}^2/\nu_k^2$, and set the thresholds for the next iteration to be $\nu_k^2 = \delta \cdot mode(\mathbf{R}^k)$. A full description of our algorithm is depicted in Algorightm 3.1. This way, one can think of the effective σ^{k+1} to be given by $\sqrt{\nu^k \delta}$.

In the following iterations the assumption about the independence between the remaining noise and the patch estimate will be very week, and so $\tilde{\sigma}_i^2$ will not be accurate. Thus, δ is determined after the first iteration only and kept fixed for the subsequent steps, while the estimate ν_k^2 provides decreasing estimates every time. An example of the obtained ν_k^2's can be seen in Fig. 3.9.

3.4.3 Results

To gain some insight into the performance of our method and as a motivating example, in Fig. 3.6 we present the denoising results on a synthetic image obtained by the regular K-SVD algorithm, and the one achieved by applying the EPLL approach with the sparse-enforcing prior. A similar demonstration was presented in [ZW11], and we include the results of this method as well. The K-SVD denoised image presents texture artifacts common to patch-based algorithms, while in the image denoised with our method the final patches are far more likely under the prior that we try to learn from the image itself.

Fig. 3.9 depicts the evolution of the PSNR of the denoised image in each iteration for this experiment. Note that given a fixed dictionary, solving the MAP estimate for each patch with a

[1]The variance is calculated as $Var(\mathbf{r}) = \frac{1}{n-1}\sum_j (\mathbf{r}_j - \bar{\mathbf{r}})^2$, where $\bar{\mathbf{r}}$ is the mean of $\mathbf{r}$.

Algorithm 3.1 EPLL with a Sparse Prior, given the noisy image $\mathbf{y}$ with a noise standard deviation of σ and an initial dictionary $\mathbf{D}_0$.

Initialization: $\mathbf{x} = \mathbf{y}$. $\mathbf{D} = \mathbf{D}_0$, $\delta = 1$, $k = 1$, $\nu_k^2 = \sigma^2$.

for $OuterIter = 1 : 3 - 4$ **do**

- $\{\mathbf{D}^{k+1}, \mathbf{x}^{k+1}\} = \underset{\boldsymbol{\gamma}_i, \mathbf{D}, \mathbf{x}}{argmin}\ \lambda||\mathbf{x}^k - \mathbf{y}||_2^2 + \sum_i ||\mathbf{D}^k\boldsymbol{\gamma}_i - \mathbf{P}_i\mathbf{x}^k||_2^2 + \mu_i||\boldsymbol{\gamma}_i||_0$, by K-SVD with error threshold ν_k^2
- get local estimates $\left(\hat{\sigma}_i^k\right)^2 = |S_i|\frac{\nu_k^2}{n}\ ,\ \forall i$
- get global estimate $\mathbf{R}^k = \Phi\left((\hat{\sigma}_i^k)^2\right)$ with Eq. (3.8)

if $k = 1$ **then**

- $\nu_{k+1}^2 = mode(\mathbf{R}^k)$
- $\tilde{\sigma}_i^2 = \sigma^2 - Var(\mathbf{r}_i),\ \forall i$
- $\tilde{\nu}^2 = mode\left(\Phi(\tilde{\sigma}_i^2)\right)$
- $\delta = \tilde{\nu}^2/\nu_{k+1}^2$

end

- $\nu_{k+1}^2 = \delta \cdot mode(\mathbf{R}^k)$
- $k = k + 1$

end

Output: $\mathbf{x}, \mathbf{D}$.

sparse prior implies applying OMP on each of them. This corresponds to the EPLL+OMP curve. On the other hand, we could minimize Eq. (3.6) w.r.t $\mathbf{D}$ as well by applying a K-SVD step, updating the dictionary as well as the sparse vectors; this is the curve depicted as EPLL+K-SVD. The constant dotted line corresponds to the original K-SVD algorithm. Note that the result after the first iteration in our method is worse than the one obtained by K-SVD where $c = 1.15$. Choosing $c = 1$ in our case, however, enables further improvement as we proceed maximizing the Expected Patch Log Likelihood. Notice also that our method converges in considerable fewer iterations than the method of [ZW11]. The right side of Fig. 3.9 shows the evolution of the thresholds ν_k used in the successive iterations, as well as the values $1/\sqrt{\beta}$ used by EPLL-GMM.

The improvement obtained by training the dictionary in each iteration of our method is both important and intuitive. It is known that applying K-SVD on a noisy image achieves good denoising results but yields somewhat noisy atoms [EA06]. By training the dictionary $\mathbf{D}$ in the progressively cleaner estimates $\mathbf{x}$ we obtain cleaner and more well defined atoms, which are later used to perform further denoising. In the top row of Fig. 3.10 we present 8 atoms trained on a noisy version of the image Lena after the first iteration, while the lower row shows the same atoms after 4 iterations.

3.4.4 Inpaiting

We next present results on image inpainting. In this particular application of image restoration, the signal is the outcome of a linear operator that deletes a number of pixels from the original

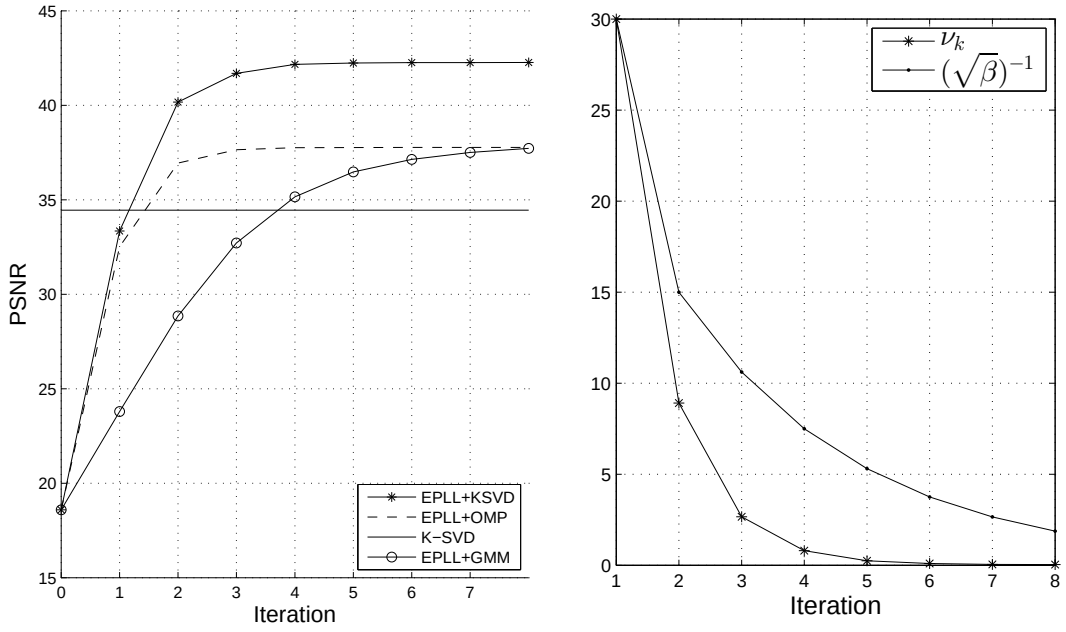


Figure 3.9: Left: PSNR evolution by EPLL with a sparsity inducing prior on the synthetic image in Fig. 3.6, compared to the original K-SVD algorithm [EA06] and the EPLL-GMM of [ZW11]. Right: sequence of thresholds ν_k determined by the proposed method and the equivalent $1/\sqrt{\beta}$ by the method of [ZW11].

image $\mathbf{x}$, plus the measurement noise. By considering a sparse prior on the original signal, we can formulate an equivalent problem to that of Eq. (3.4), where $\mathbf{A}$ is the missing-pixels mask. The corresponding cost function can be minimized in a block coordinate manner, coding for the unknown sparse representation and updating the dictionary. In this case, however, the threshold in the OMP has to consider only the energy of existing pixel in each patch [MESM08]. This again represents the first iteration of the Half Splitting strategy proposed in [ZW11], and we may perform the next iterations by estimating the remaining noise as explained above. Furthermore, after the first iteration our estimate includes values of the missing pixel. We can then make use of the previous denoising strategy to tackle the next iteration, by having knowledge of the supports used to inpaint each patch, as it was previously explained.

Table 3.1 shows the results on inpainting the popular images *peppers* and *Lena* with 25%,

Missing Pixels	25%		50%		75%	
K-SVD	29.67	28.81	27.92	27.27	23.64	23.86
EPLL+K-SVD	**29.71**	**28.85**	**28.18**	**27.39**	**23.81**	**24.07**

Table 3.1: Inpaiting results in terms of Peak Signal to Noise Ratio (PSNR) for 25%, 50% and 75% missing pixels for the images *peppers* (left subcolumns) and *Lena* (right subcolumns), with additive white Gaussian noise ($\sigma = 20$).

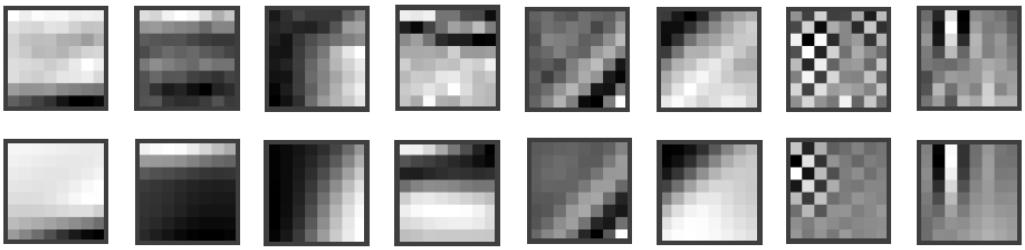


Figure 3.10: Atoms from a dictionary trained on a noisy version of the image Lena. The top row corresponds to the atoms after the first iteration of our method (essentially, after applying K-SVD), while the lower row corresponds to the same atoms after 4 iterations of the EPLL with a sparsity enforcing prior.

50% and 75% missing pixels, with additive white Gaussian noise ($\sigma = 20$). As it can be seen, the EPLL scheme leads to a slight improvement in the K-SVD inpainting results, with increased effect for higher missing pixels rates. The same concept could be applied to more sophisticated algorithms that use a sparsity-based prior, such as the state-of-the-art method of [RPE14].

3.4.5 Denoising

We conclude this section by presenting results on denoising of 12 images from the Kodak database, for different noise levels. We compare here the performance of the K-SVD denoising algorithm in [EA06] and our approach of the EPLL framework with a sparse prior (EPLL-K-SVD, where the dictionary is also updated in each iteration). In all cases we performed 4 iterations of this method, as this was found to be a convenient compromise between runtime and performance. For both K-SVD methods, an initial dictionary with 1024 atoms was trained on overlapping 8×8 patches from 9 training images using K-SVD. We include for completion the results achieved by the EPLL with a Gaussian Mixture Model (GMM) as the image prior from [ZW11].

In Fig. 3.11 we present the relative increase in PSNR, averaged over all 12 images. The EPLL with a Sparse enforcing Prior shows a clear improvement over the regular K-SVD. Furthermore, the complete implementation of the denoising algorithm closes the gap between the original K-SVD and EPLL-GMM, having comparable performance: our method achieves the best results for lower noise energy while EPLL with GMM is better for higher noise levels. In Fig.3.12 and Fig.3.13 we present two examples of denoised images by the three methods. Note, lastly, how artifacts are notably reduced in the resulting images processed by our method.

3.5 Gaussian Mixture Diffusion

As we have seen, most state-of-the-art denoising algorithms employ a patch-based approach by enforcing a local model or prior, such as self similarity, sparse representation, or Gaussian Mixture Model (GMM). While applying these models, these algorithms implicitly build a notion of similarity between the image pixels. This can be formulated as an image-adaptive linear-filter

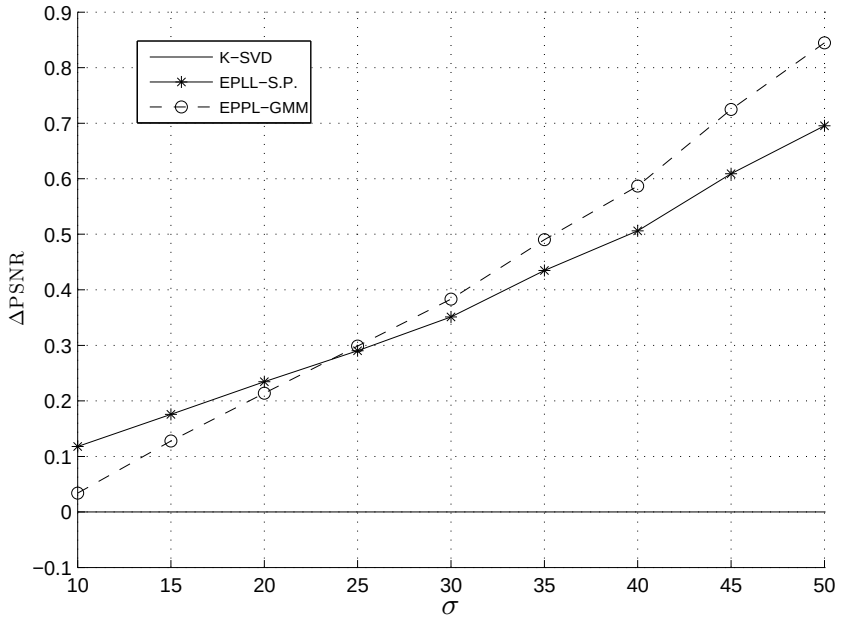


Figure 3.11: Denoising results averaged over 12 images from the Kodak Dataset with respect to K-SVD [EA06] by EPLL with GMM [ZW11] and the method presented here: EPLL with Sparse Prior, in terms of the Peak Signal to Noise Ratio (PSNR).

which is then used to denoise or restore the degraded image. In this final section, we focus on such a filter emerging from the GMM, study its properties and construct a graph Laplacian from it.

Given the noisy measurements $\mathbf{y}$, the image restoration task can be expressed in terms of an optimization problem, minimizing a cost function over the unknown image $\mathbf{x}$:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda\mathcal{R}(\mathbf{x}). \tag{3.9}$$

From a maximum a posteriori perspective, the first term corresponds to the log-likelihood function while $\mathcal{R}(\mathbf{x})$ enforces the specific model on the unknown image, with parameter λ. This last term acts as the regularizer, promoting *smoothness* or other qualities that – we believe – characterize natural images.

Broadly speaking, often times the denoising process can be decomposed into two stages. The first one involves highly non-linear decisions which enforce quite sophisticated *local* priors on small patches extracted from the image, whereas the second stage accounts for projections and averaging in order to obtain the final *global* image. Interestingly, as pointed out in [Mil13, RE15], once the non-linear part is fixed, these algorithms can be formulated as an image-adaptive linear

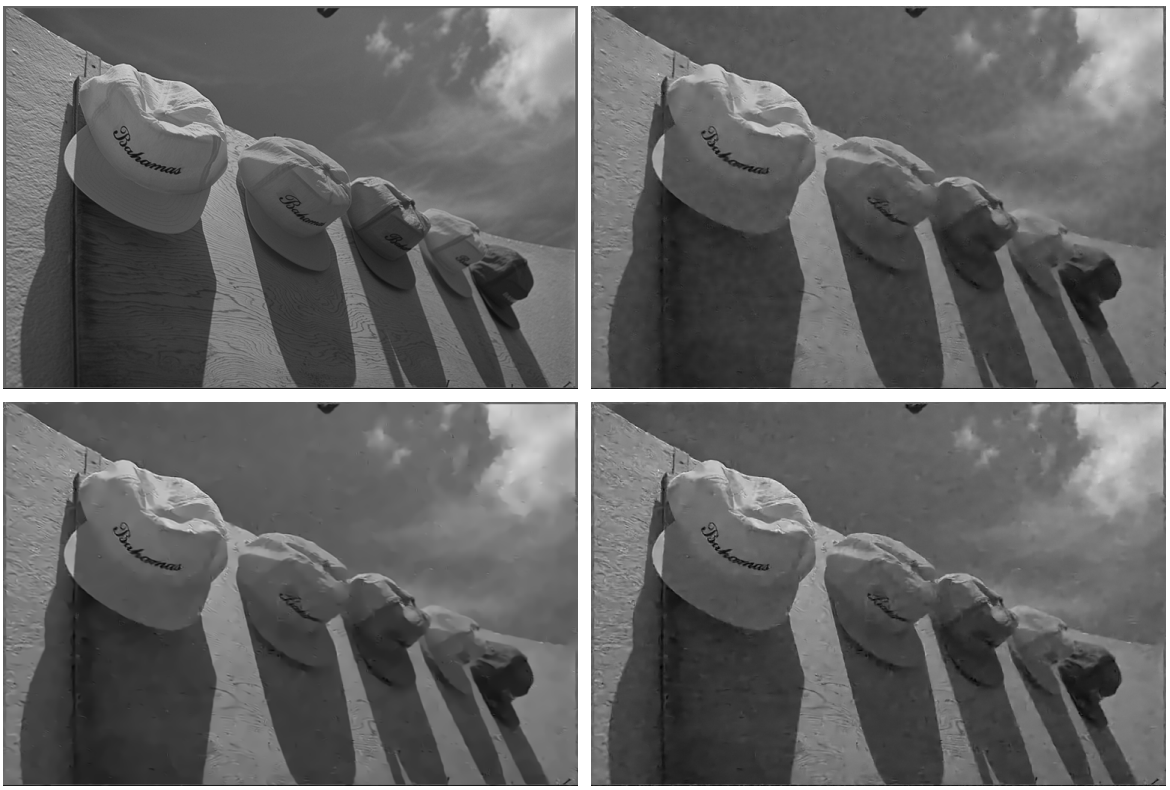


Figure 3.12: Denoising results of an image from the Kodak Database corrupted with a noise standard deviation of $\sigma = 25$. Top left: original image. Top right: K-SVD (PSNR = 32.14 dB). Bottom left: EPLL with Sparse Prior (PSNR = 32.42 dB). Bottom Right: EPLL with GMM (PSNR = 32.25 dB).

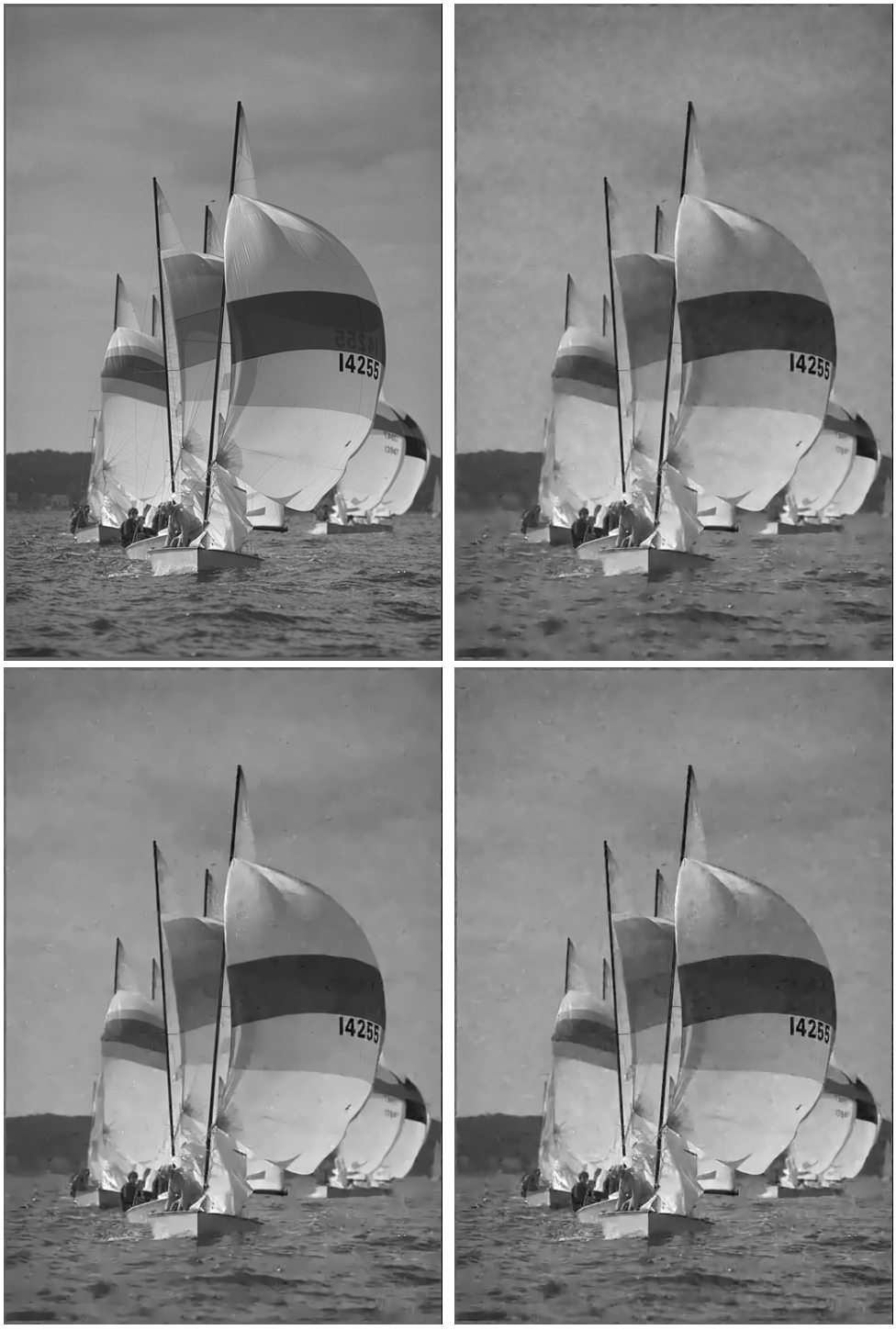


Figure 3.13: Denoising results of an image from the Kodak Database, initially corrupted with additive white Gaussian nose ($\sigma = 25$). Top left: Original Image, top right: K-SVD (PSNR = 31.42 dB), bottom left: EPLL with Sparse Prior (PSNR = 31.83 dB), bottom right: EPLL with GMM (PSNR = 31.85 dB).

filter, $\mathbf{W}$. This way, the entire denoising process in this framework can be expressed as

$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{y}, \tag{3.10}$$

where $\hat{\mathbf{x}} \in \mathbf{R}^N$ is the denoised image, and $\mathbf{W} \in \mathbf{R}^{N \times N}$ is the matrix form of the denoiser (we will describe this operator in more detail in Section 3.5.1).

On the one hand, Equation (3.10) states that each denoised pixel in $\hat{\mathbf{x}}$ is the outcome of a weighted average over the noisy image pixels, where the weights are determined by the specific denoising algorithm. On the other hand, this shows that most denoising algorithms implicitly build a notion of similarity between the i-th and j-th image pixels, given by the entry $\mathbf{W}(i,j)$. Based on this observation, the denoiser can be formulated as a weighted graph, where the vertices refer to the image pixels, and the (weighted) edges represent the pixels' similarity. Previous works have addressed the graph formulation (and its properties) of the K-SVD [RE15], the Non-Local Means, the Bilateral and the LARK kernels [Mil13]. Yet, despite the popularity of the GMM prior in the image processing community, this analysis has not been addressed for its resulting operator – this will be the first concern of our work.

Recently, the graph formulation has been employed to regularize the denoising process [GO07, Mil13, RE15], designing an image-adaptive term $\mathcal{R}(\mathbf{x})$ in Equation (3.9). This term is usually expressed in terms of the Laplacian operator, defined by $\mathbf{L} = \mathbf{I} - \mathbf{W}$, where $\mathbf{I} \in \mathbf{R}^{N \times N}$ is the identity matrix and $\mathbf{W}$ is induced by different denoisers (from Equation (3.10)). Broadly speaking, the eigenvectors that correspond to the small eigenvalues of $\mathbf{L}$ encapsulate most of the structure of the underlying signal [MS14]. As such, one may propose a graph-based regularization term that penalizes those components in $\mathbf{x}$ corresponding to the large eigenvalues of $\mathbf{L}$; e.g., as done in [GO07, ELB08],

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \mathbf{x}^T \mathbf{L} \mathbf{x}. \tag{3.11}$$

Notice that when moving from Equation (3.9) to Equation (3.11), we are constraining ourselves to priors that have a graph-Laplacian interpretation. In these cases, the performance of the resulting algorithm depends on the choice of $\mathbf{L}$, which is in turn determined by the similarity measure we use to construct the corresponding graph.

The problem in Equation (3.11) is certainly not the only way to enforce a graph-based regularization when dealing with inverse problems. Recent works have also considered replacing the data-fidelity term by a weighted norm induced by the matrix $\mathbf{W}$, as in [KM14]. Another alternative, presented in [RE15], is to enforce the reconstructed image $\mathbf{x}$ to be close to the filtered image $\mathbf{W}\mathbf{y}$. Formally,

$$\mathcal{J}(\mathbf{x}) = \min_{\mathbf{x}} \frac{1}{2}\|\mathbf{W}\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \mathbf{x}^T \mathbf{L} \mathbf{x}. \tag{3.12}$$

This formulation generally provides better results than the problem in Equation (3.11), as it is related to boosting methods. In particular, this is the cost function minimized by the SOS boosting [RE15].

Interestingly, the minimization of this kind of problems can be interpreted from a variational

perspective. The Non-Local Diffusion (NLD) algorithm [GO07] suggests a non-local generalization of the diffusion framework by employing a functional defined over a set of pixels which are not necessarily near each other. Unlike the conventional diffusion methods, the minimization of the corresponding functional amounts to a diffusion process between pixels that can now be *far apart* by leveraging some notion of similarity or affinity. In particular, the NLD employed the affinity measure induced by the Non Local Means (NLM) kernel [BCM05], defining the distance between pixels as a function of the Euclidean distance between their corresponding patches. The resulting algorithm effectively minimizes the cost function in Equation (3.11), where the Laplacian is the one corresponding to the NLM operator.

In this work, we explore the algorithm resulting from the problem in Equation (3.12) in the case of a Laplacian operator induced by the GMM prior. We provide a detailed analysis of the denoiser resulting from GMM, and employ the formulation in terms of a non-local diffusion process. This way, our also work extends and improves the non-local diffusion algorithm of [GO07] by (1) employing a similarity measure induced by the GMM operator, and (2) considering the cost function in Equation (3.12) instead of the original problem in (3.11). As we will show in the experimental section, our proposed Gaussian Mixture Diffusion (GMD) approach outperforms both the initial formulation of the NLD with the NLM kernel [GO07], and the original GMM algorithm. Interestingly, the GMD is also competitive or even better than the EPLL [ZW11], which builds upon GMM as well.

3.5.1 Gaussian Mixture Model

GMM is a popular prior for natural image patches, which has been shown to be very effective in several image restoration tasks [PSWS03,ZW11]. This prior models the distribution of patches as the sum of multivariate Gaussians learned from real data. Applying this prior for image denoising accounts to formulating a MAP estimator for each independent patch from the corrupted image. This can be approximated by choosing the Gaussian with the highest conditional weight for each patch, and then applying a plain Wiener filter with the corresponding covariance matrix [ZW11]. Finally, a patch averaging step is applied in order to obtain the final denoised image.

Given K (learned) Gaussian distributions, characterized by their covariance matrices Σ_k, with zero mean[2], denoising each patch $\mathbf{z}_i \in \mathbf{R}^n$ can be formally expressed by the following minimization problem

$$\hat{\mathbf{p}}_i = \arg\min_{\mathbf{p}} \|\mathbf{p} - \mathbf{z}_i\|_2^2 + \sigma^2 \mathbf{p}^T \boldsymbol{\Sigma}_{k(i)}^{-1} \mathbf{p},$$

where $k(i)$ is the index of the chosen Gaussian with highest conditional weight [ZW11] for the i^{th} patch, and $\hat{\mathbf{p}}_i$ is its estimated clean version. This problem has a closed form solution in terms of the Wiener filter, given by

$$\hat{\mathbf{p}}_i = \left(\mathbf{I} + \sigma^2 \boldsymbol{\Sigma}_{k(i)}^{-1}\right)^{-1} \mathbf{z}_i \;=\; \mathbf{F}_i \, \mathbf{z}_i. \tag{3.13}$$

Next, the denoised patches $\hat{\mathbf{p}}_i$ are merged together by averaging. This is done by minimizing

[2]For simplicity, we make the common assumption that the image patches have zero-mean.

the following cost function

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \mu\|\mathbf{y}-\mathbf{x}\|_2^2 + \sum_{i=1}^{N} \|\hat{\mathbf{p}}_i - \mathbf{R}_i\mathbf{x}\|_2^2, \tag{3.14}$$

where $\hat{\mathbf{x}} \in \mathbf{R}^N$ is the estimated (denoised) global image, and $\mathbf{R}_i \in \mathbf{R}^{n\times N}$ is a matrix that extracts the i^{th} patch from the image. Following the procedure described in [RE15], the closed form solution of Equation (3.14) is given by

$$\begin{aligned}\hat{\mathbf{x}} &= \left(\mu\mathbf{I} + \sum_i \mathbf{R}_i^T\mathbf{R}_i\right)^{-1}\left(\mu\mathbf{I} + \sum_i \mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i\right)\mathbf{y} \\ &= \mathbf{W}_{\text{GMM}}\mathbf{y},\end{aligned} \tag{3.15}$$

where the filters $\mathbf{F}_i$ are the Wiener filters in Equation (3.13). In the above derivation we have used the fact that $\mathbf{z}_i = \mathbf{R}_i\mathbf{y}$. Notice that this linear operator can be written as a matrix $\mathbf{W}_{\text{GMM}}$, and thus the denoised image is simply expressed as $\hat{\mathbf{x}} = \mathbf{W}_{\text{GMM}}\ \mathbf{y}$.

While the GMM model is a popular choice for image denoising, a formal analysis of $\mathbf{W}_{\text{GMM}}$ has not yet been addressed. In this section, we present the properties of this filter and provide their corresponding proofs in Appendix 3.7.1. We should note that all these properties are also shared by the KSVD filter [RE15].

Theorem 3.1. *Under the assumption of periodic boundary conditions*[3]*, the matrix* $\mathbf{W}_{GMM}$*, defined in Equation* (3.15)*, has the following properties:*

1. $\mathbf{W}_{GMM} = \mathbf{W}_{GMM}^T$*: it is symmetric.*
2. $\mathbf{W}_{GMM} \succ 0$*: it is positive definite, and has minimal eigenvalue equal to* $\frac{\mu}{\mu+n}$*.*
3. $\|\mathbf{W}_{GMM}\|_2 \leq 1$*: its spectral radius* ≤ 1*.*

3.5.2 The Proposed Approach

Based on the properties provided in Theorem 3.1, we can draw interesting conclusions. As it was done for the K-SVD operator matrix in [RE15], the matrix $\mathbf{W}_{\text{GMM}}$ can be decomposed into a similarity matrix $\mathbf{K}_{\text{GMM}}$ and a normalization matrix $\mathcal{D}$. Formally,

$$\begin{aligned}\mathbf{W}_{\text{GMM}} =& \left(\mu\mathbf{I} + \sum_i \mathbf{R}_i^T\mathbf{R}_i\right)^{-1}\left(\mu\mathbf{I} + \sum_i \mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i\right) \\ =& \mathcal{D}^{-1}\mathbf{K}_{\text{GMM}}.\end{aligned}$$

As a consequence, a graph-Laplacian can be constructed from this operator by

$$\mathbf{L}_{\text{GMM}} = \mathbf{I} - \mathbf{W}_{\text{GMM}},$$

[3] By assuming cyclic boundary conditions, the following holds: $\sum_i \mathbf{R}_i^T\mathbf{R}_i = n\mathbf{I}$, where n is the dimension of the image patch.

where the eigenvalues of $\mathbf{L}_{\text{GMM}}$ are in $[0, 1)$.

The denoising algorithm is obtained by minimizing the function $\mathcal{J}(\mathbf{x})$, defined in Equation (3.12), which can be done using a gradient descent strategy. As such, the estimated image is found by iterating:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \gamma \nabla \mathcal{J}(\mathbf{x})$$

where

$$\nabla \mathcal{J}(\mathbf{x}) = \mathbf{x} - \mathbf{W}\mathbf{y} + 2\lambda \mathbf{L}\mathbf{x}. \tag{3.16}$$

For a given step-size γ and regularizer-strength ρ (fixed for each noise level), we run the gradient descent process a fixed number of iterations. In practice, we do not build $\mathbf{L}$ nor $\mathbf{W}$ explicitly, but rather apply it by using the local filters $\mathbf{F}_i$. In the case of the GMM, this corresponds to knowing the Gaussians chosen for each patch. Before moving to the experimental section, we should note that the non-local diffusion method of [GO07] employs a minimization driven by an update very similar to that in Equation (3.16), where $\mathbf{y}$ is used instead of $\mathbf{W}\mathbf{y}$ – thus, the name. This shows more clearly the connection to the NLD algorithm, as well as providing a variational interpretation to the SOS boosting algorithm [RE15].

3.5.3 Experimental Results

In this section we present image denoising results corresponding to the minimization of the cost function in Equation (3.12) for the Laplacian matrix $\mathbf{L}$ induced by the GMM prior, for various standard test images and noise levels. As for the parameters the proposed GMD, for all noise levels we set $\gamma = 0.1$, $\lambda = 2$, and 2 diffusion steps are applied.

We compare the proposed approach to the Non Local Diffusion work [GO07], which corresponds to a diffusion process guided by the graph built with the NLM kernel (minimizing Equation (3.11)). In addition, we also compare the GMD to the plain GMM denoiser as a baseline. We include for completion the results obtained by the EPLL [ZW11]. This algorithm essentially minimizes a cost function similar to the one in Equation (3.9), where the prior is enforced *on the reconstructed patches*. This idea boils down to applying a GMM-based denoiser iteratively, with a set of parameters which need to be tuned. Note that the EPLL algorithm is still a patch-based method which also updates its operator (choosing the Gaussian Mixtures) at every iteration, whereas in our approach these remain constant.

Table 3.2 provides a comparison between NLD, GMM, EPLL and the proposed GMD approach in terms of Peak Signal to Noise Ratio (PSNR). As can be seen, for $\sigma = 20$, GMD achieves the best reconstruction performance. For $\sigma = 30$, EPLL and GMD obtain comparable results, whereas for $\sigma = 50$, the EPLL slightly outperforms GMD. We remind the reader that the EPLL, unlike our approach, updates its operator at each iteration. We believe that by updating the matrix $\mathbf{L}$ (i.e., re-run the Gaussian selection step) the GMD results can be further improved, too. However, we choose not to include this step in our algorithm in order to focus the attention on the minimization of the problem in Equation (3.12).

Figure 3.14 provides a visual comparison between the NLD approach of [GO07], GMM and

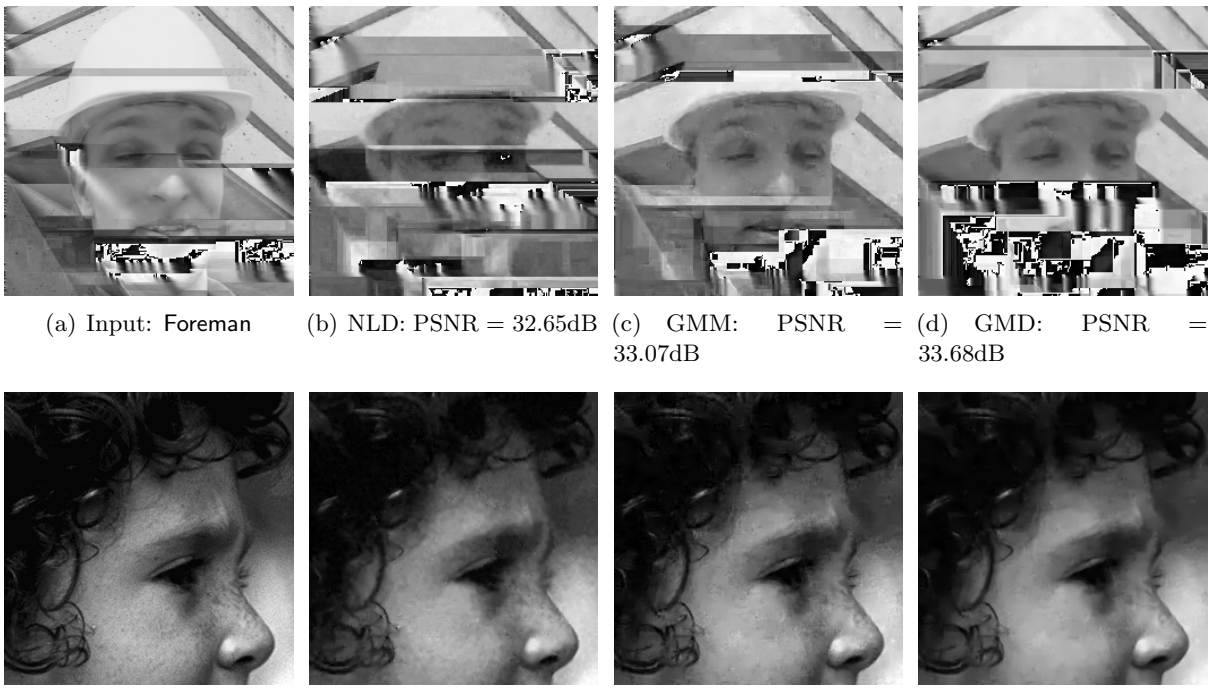
Table 3.2: Denoising results for various noise levels and images, given in terms of PSNR. The best result is highlighted.

σ**Image**	House	Saturn	Foreman	Lena	Peppers	Girl	Woman	**Averages**
Non Local Diffusion (NLM kernel)								
20	32.27	35.21	32.65	31.35	31.51	30.03	30.99	32.00
30	30.00	32.57	30.42	29.36	26.65	28.60	28.86	29.49
50	27.14	29.64	27.76	26.86	27.09	26.72	26.06	27.40
GMM								
20	32.59	35.54	33.07	32.23	32.11	30.56	31.81	32.56
30	30.84	33.38	31.24	30.48	30.56	29.38	29.86	30.82
50	28.13	30.33	28.66	28.03	28.23	27.82	27.26	28.35
EPLL-GMM								
20	32.98	36.68	33.63	32.60	32.51	30.71	32.08	33.03
30	**31.22**	34.23	31.66	**30.78**	**30.90**	**29.54**	**30.04**	**31.20**
50	**28.76**	**31.15**	**29.16**	**28.41**	**28.68**	28.00	**27.57**	**28.82**
GMD (Proposed)								
20	**33.07**	**36.78**	**33.68**	**32.61**	**33.53**	**30.75**	**32.14**	**33.22**
30	31.20	**34.35**	**31.70**	30.70	30.86	29.53	29.97	31.19
50	28.42	31.12	29.08	28.23	28.47	**28.07**	27.27	28.66

ours GMD method. As can be seen the GMD reconstruction has less artifacts than the baseline methods, complying with the quantitative PSNR measure.

3.6 Chapter Conclusion

In this chapter, after having shown the limitations of patch-based approaches in the form of artifacts, we initially presented a multi-scale extension of the K-SVD denoising algorithm by proposing a global MAP estimator for the denoised image in the wavelet domain. We tackled this minimization problem iteratively in terms of the K-SVD algorithm per band, applying a multi-scale patch denoising of the image. We then boosted the results by fusing the single scale and multi-scale K-SVD outcome images by a weighted sparse coding step. The results obtained by this method show the potential benefits of working within a multi-scale framework, as we are able to combine bigger effective atoms that give rise to clear smooth areas, in which most current methods fail. The combination of the regular and multi-scale K-SVD denoised images could be improved by proposing a patch-based adaptive weight instead of a global one, and the joint sparse coding alternative is effective, but not necessarily the only one. An orthogonal wavelet transform enabled a simple multi-scale analysis, but other multi-scale transforms might yield improvements on this framework and are worth exploring. Moreover, this multi-scale approach is not restricted to the K-SVD algorithm, and the question posed in the introduction still holds for other methods. We then moved to show that maximizing the Expected Patch Log Likelihood with a sparse inducing prior leads naturally to a formulation of which the K-SVD algorithm represents the first iteration. In its original form, this method performed only one update of the image due to technical difficulties in assessing the remaining noise level. We have shown how to circumvent this issue and go beyond this first iteration, intrinsically determining the coding threshold in



(a) Input: Foreman (b) NLD: PSNR = 32.65dB (c) GMM: PSNR = 33.07dB (d) GMD: PSNR = 33.68dB

(e) Input: Girl (f) NLD: PSNR = 30.03dB (g) GMM: PSNR = 30.56dB (h) GMD: PSNR = 30.75dB

Figure 3.14: Denoising of the images Foreman (a-d) and Girl (e-h), when $\sigma = 20$.

each step. This work completes the one in [EA06], providing the full path to the numerical minimization of the original cost function and exploiting all the potential of the sparse inducing prior. The resulting algorithm shows a clear improvement over K-SVD in all the experiments. In denoising in particular, EPLL with a sparse prior achieved comparable performance to the state of the art method of EPLL with a GMM prior. Interestingly, both priors yield comparable results when applied within the EPLL framework. Lastly, we introduced a graph interpretation of the GMM denoiser, followed by an analysis of the resulting operator. The denoising effect is obtained by minimizing a cost function with a graph-Laplacian regularization, as suggested by the SOS formulation. We have shown that the proposed approach can be understood from a variational perspective, resulting in close variant of the NLD algorithm. Following the experimental results, it was evidenced that our approach is more effective than the traditional NLD counterpart. Our results not only outperform those by the regular GMM denoising algorithm, but they are also competitive with those of the state-of-the-art EPLL method. We believe that updating the graph (and the corresponding operator) along the iterations (as done by the EPLL and the SOS boosting) will result in increased performance, and this is a promising direction of future work.

More broadly, we have seen local-patch based approaches provide effective and powerful priors. Nevertheless, one can boost the performance of these algorithm in several ways once a more global analysis is performed.

3.7 Chapter Appendix

3.7.1 Properties of the GMM Matrix

In this appendix we provide the proves for Theorem 3.1.

Proof. We will start by showing that property 1 holds. Under the periodic boundary conditions, we have that

$$\begin{aligned}\mathbf{W}_{\text{GMM}} &= \left(\mu\mathbf{I}+\sum_i \mathbf{R}_i^T\mathbf{R}_i\right)^{-1}\left(\mu\mathbf{I}+\sum_i \mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i\right)\\ &= \frac{1}{\mu+n}\left(\mu\mathbf{I}+\sum_i \mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i\right).\end{aligned}$$

Recall that the filters $\mathbf{F}_i$, expressed in Equation (3.13) are symmetric (and more so, positive definite) as they are the inverse of symmetric matrices. Therefore, $\mathbf{W}_{\text{GMM}}$ is the sum of symmetric matrices, and it is then also symmetric.

The second property can be deduced using the same rationale. The matrices given by $\mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i$ are symmetric and positive semidefinite. Thus, their sum $\sum_i \mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i$ is also positive semidefinite. Moreover, we can express

$$\mathbf{W}_{\text{GMM}} = \frac{\mu}{\mu+n}\mathbf{I}+\frac{1}{\mu+n}\left(\sum_i \mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i\right).$$

The first term in this sum is obviously positive definite, while the second is positive semidefinite. From this, the minimal eigenvalue of the GMM matrix $\lambda_{min}(\mathbf{W}_{\text{GMM}}) = \frac{\mu}{\mu+n} > 0$, and $\mathbf{W}_{\text{GMM}} \succ 0$.

To prove the last property, consider the operator norm of $\mathbf{W}_{\text{GMM}}$ given the (square of the) decomposition presented above:

$$\begin{aligned}\|\mathbf{W}_{\text{GMM}}\|_2 &= \left\|\frac{\mu}{\mu+n}\mathbf{I}+\frac{1}{\mu+n}\left(\sum_i \mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i\right)\right\|_2\\ &\leq \left\|\frac{\mu}{\mu+n}\mathbf{I}\right\|_2+\left\|\frac{1}{\mu+n}\left(\sum_i \mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i\right)\right\|_2.\end{aligned} \tag{3.17}$$

Note first that the operator norm of the first term is given by $\frac{\mu}{\mu+n}$. Focusing on the second term, we consider a similar decomposition to that presented in [RE15] (Appendix B). The sum in the last Equation, over all N patches, considers overlapping structures. We can decompose this term by considering the sum over $\{\Omega_j\}_{j=1}^n$ groups of *non overlapping* patches only. With

this, we have that

$$\left\|\sum_i \mathbf{R}_i^T\mathbf{F}_i\mathbf{R}_i\right\|_2 = \left\|\sum_j^n \sum_{k\in\Omega_j} \mathbf{R}_k^T\mathbf{F}_k\mathbf{R}_k\right\|_2 \leq \sum_j^n \|\mathcal{M}_j\|_2\,,$$

where we have denoted $\mathcal{M}_j = \sum_{k\in\Omega_j} \mathbf{R}_k^T\mathbf{F}_k\mathbf{R}_k$. Notice that $\mathcal{M}$ is a block diagonal matrix, having the filters $\mathbf{F}_k$ as leading minors.

To show that $\|\mathcal{M}_j\|_2 \leq 1$ we will rely on the definition of the induced norm. Consider thus any vector $\mathbf{x}\in\mathbb{R}^N$ such that $\|\mathbf{x}\|_2 = 1$. Moreover, denote $\mathbf{R}_k\mathbf{x} = \mathbf{x}_k \in \mathbb{R}^n$. Then,

$$\|\mathcal{M}_j\mathbf{x}\|_2^2 = \left\|\sum_{k\in\Omega_j} \mathcal{M}_j^k\mathbf{x}\right\|_2^2 = \left\|\sum_{k\in\Omega_j} \mathbf{R}_k^T\mathbf{F}_k\mathbf{x}_k\right\|_2^2. \tag{3.18}$$

Due to the fact that $\mathbf{R}_k^T\mathbf{R}_j = \mathbf{0}$, $\forall i \neq k$ (because the corresponding patches are non-overlapping), we have that

$$\begin{aligned}\|\mathcal{M}_j\mathbf{x}\|_2^2 &\leq \sum_{k\in\Omega_j} \left\|\mathbf{R}_k^T\mathbf{F}_k\mathbf{x}_k\right\|_2^2, \\ &\leq \sum_{k\in\Omega_j} \left\|\mathbf{R}_k^T\right\|_2^2 \|\mathbf{F}_k\mathbf{x}_k\|_2^2, \leq \sum_{k\in\Omega_j} \|\mathbf{F}_k\mathbf{x}_k\|_2^2\,,\end{aligned} \tag{3.19}$$

where we have used the multiplicative property of the operator norm and the fact that $\|\mathbf{R}_k^T\|_2 = 1$.

Looking now at the square of the operator norm of $\mathbf{F}_k\mathbf{x}_k$:

$$\|\mathbf{F}_k\mathbf{x}_k\|_2^2 = \left\|(\mathbf{I}+\sigma^2\mathbf{\Sigma}_k^{-1})^{-1}\mathbf{x}_k\right\|_2^2 \leq \|\mathbf{x}_k\|_2^2\,, \tag{3.20}$$

where the inequality holds since $\lambda_{max}(\mathbf{I}+\sigma^2\mathbf{\Sigma}_k^{-1}) \geq 1$, as $\mathbf{\Sigma}_k^{-1} \succ 0$.

By using Equations (3.18), (3.19), (3.20), and the fact that $\|\mathbf{x}\|_2^2 = \sum_{k\in\Omega_j} \|\mathbf{x}_k\|_2^2$, we have that

$$\begin{aligned}\frac{\|\mathcal{M}_j\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} &= \frac{\left\|\sum_{k\in\Omega_j} \mathbf{R}_k^T\mathbf{F}_k\mathbf{x}_k\right\|_2^2}{\|\sum_{k\in\Omega_j} \mathbf{x}_k\|_2^2} \leq \frac{\sum_{k\in\Omega_j} \|\mathbf{F}_k\mathbf{x}_k\|_2^2}{\sum_{k\in\Omega_j} \|\mathbf{x}_k\|_2^2} \\ &\leq \frac{\sum_{k\in\Omega_j} \|\mathbf{x}_k\|_2^2}{\sum_{k\in\Omega_j} \|\mathbf{x}_k\|_2^2} = 1.\end{aligned}$$

Therefore, the squared maximal singular value (i.e. the operator norm) of $\mathcal{M}_j$ is ≤ 1. Incorpo-

rating this into Equation (3.17), we have that

$$\begin{aligned}\|\mathbf{W}_{\mathrm{GMM}}\|_2 \leq & \left\| \frac{\mu}{\mu+n}\mathbf{I} \right\|_2 + \left\| \frac{1}{\mu+n}\left(\sum_i \mathbf{R}_i^T \mathbf{F}_i \mathbf{R}_i\right) \right\|_2 \\ \leq & \frac{\mu}{\mu+n} + \frac{1}{\mu+n}\sum_{j=1}^{n} \|\mathcal{M}_j\|_2 \\ \leq & \frac{\mu}{\mu+n} + \frac{n}{\mu+n} \leq \ 1.\end{aligned}$$

Chapter 4

Trainlets

Chapter Abstract

Sparse representation has shown to be a very powerful model for real world signals, and has enabled the development of applications with notable performance. Combined with the ability to learn a dictionary from signal examples, sparsity-inspired algorithms are often achieving state-of-the-art results in a wide variety of tasks. However, these methods have traditionally been restricted to small dimensions mainly due to the computational constraints that the dictionary learning problem entails. In the context of image processing, this implies handling small image patches. In this chapter, we show how to efficiently handle bigger dimensions and go beyond the small patches in sparsity-based signal and image processing methods. We build our approach based on a new cropped Wavelet decomposition, which enables a multi-scale analysis with virtually no border effects. We then employ this as the base dictionary within a double sparsity model to enable the training of adaptive dictionaries. To cope with the increase of training data, while at the same time improving the training performance, we present an Online Sparse Dictionary Learning (OSDL) algorithm to train this model effectively, enabling it to handle millions of examples. The derivations that we now present show that dictionary learning can be up-scaled to tackle a new level of signal dimensions, obtaining large adaptable atoms that we call *Trainlets*.

4.1 Dictionary Learning for High Dimensional Signals

As presented in Chapter 2, the dictionary learning problem consist in adapting a matrix $\mathbf{D}$ to represent a set of signal examples (given by the columns of a matrix $\mathbf{Y}$) as sparse as possible. Formally, this can be written as

$$\min_{\mathbf{D},\mathbf{\Gamma}} \quad \frac{1}{2}\|\mathbf{Y}-\mathbf{D}\mathbf{\Gamma}\|_F^2 \quad \text{subject to} \quad \|\boldsymbol{\gamma}_i\|_0 \leq p \quad \forall i, \tag{4.1}$$

While several works have proposed different algorithmic solutions to minimize this cost function, they involve computationally complex and time consuming algorithms. As a consequence, all dictionary learning approaches (and applications) have typically been restricted to small dimensional signals. In the context of image processing, small signals imply handling small image patches. Most state-of-the-art methods for image restoration exploit such a localized patch based approach [DFKE07, MBS09, ZW11]. As we have already presented extensively in the previous chapter, small overlapping patches (7×7 - 11×11) are extracted from the corrupted image and treated relatively independently according to some image model [DFKE06, ZW11], sparse representations being a popular choice [DZSW11, YWHM10, EA06, RPE14]. The full image estimation is lastly formed by merging together the small restored patches by overlapping and averaging.

Some works have attempted to handle larger two dimensional patches (i.e., greater than 16×16) with some success. In [OLE11], and later in [SOE14], traditional K-SVD is applied in the Wavelet domain. These works implicitly manage larger patches while keeping the atom dimension small, noting that small patches of Wavelet coefficients translate to large regions in the image domain. In the context of Convolutional Networks, on the other hand, the work in [BSH12] has reported encouraging state-of-art result on patches of size 17×17.

Though adaptable, explicit dictionaries are computationally expensive to apply. Some efforts have been done in designing fast dictionaries that can be both applied and learned efficiently. This requirement implies constraining the degrees of freedom of the explicit matrix in some way, i.e. imposing some structure on the dictionary. One such possibility is the search for adaptable separable dictionaries, as in [HSK13], or the search of a dictionary which is an image in itself as in [AE08,BMBP11], lowering the degrees of freedom and obtaining (close to) shift invariant atoms. Another, more flexible alternative, has been the pursuit of sparse dictionaries [RZE10, YD09]. In these works the dictionary is composed of a multiplication of two matrices, one of which is sparse. The work in [LMG15] takes this idea a step further, composing a dictionary from the multiplication of a sequence of sparse matrices. In the interesting work reported in [CMTD15] the dictionary is modeled as a collection of convolutions with sparse kernels, lowering the complexity of the problem and enabling the approximation of popular analytically-defined atoms. All of these works, however, have not addressed dictionary learning on real data of considerably higher dimensions or with a considerably large dataset.

A related but different model to the *synthesis sparse* model studied so far is the *analysis model* [EMR07, RE14]. In this framework, a dictionary $\mathbf{W}$ is learned such that $\|\mathbf{W}\mathbf{y}\|_0 \ll n$. A

close variant is the Transform Learning model, where it is assumed that $\mathbf{W}\mathbf{y} \approx \boldsymbol{\gamma}$ and $\|\boldsymbol{\gamma}\|_0 \ll n$, as presented in [RB13b]. This framework presents interesting advantages due to the very cheap sparse coding stage. An online transform learning approach was presented in [RWB15], and a sparse transform model was presented in [RB13a], enabling the training on bigger image patches. In our work, however, we constrain ourselves to the study of synthesis dictionary models.

We give careful attention to the model proposed in [RZE10]. In this work a double sparse model is proposed by combining a fixed separable dictionary with an adaptable sparse component. This lowers the degrees of freedom of the problem in Equation (4.1), and provides a feasible way of treating high dimensional signals. However, the work reported in [RZE10] concentrated on 2D and 3D-DCT as a base-dictionary, thus restricting its applicability to relatively small patches.

In this chapter we expand on this model, showing how to efficiently handle bigger dimensions and go beyond the small patches in sparsity-based signal and image processing methods. This model provides the flexibility of incorporating multi-scale properties in the learned dictionary, a property we deem vital for representing larger signals. For this purpose, we propose to replace the fixed base dictionary with a new multi-scale one. We build our approach on *cropped* Wavelets, a multi-scale decomposition which overcomes the limitations of the traditional Wavelet transform to efficiently represent small images (expressed often in the form of severe border effects).

Another aspect that has limited the training of large dictionaries has been the amount of data required and the corresponding amount of computations involved. As the signal size increases, a (significant) increase in the number of training examples is needed in order to effectively learn the inherent data structure. While traditional dictionary learning algorithms require many sweeps of the whole training corpus, this is no longer feasible in our context. Instead, we will look to online learning methods, such as Stochastic Gradient Decent (SGD) [Bot98]. These methods have gained prominence in recent years with the advent of big data, and have been used in the context of traditional (unstructured) dictionary learning [MBPS10] and in training the special structure of the Image Signature Dictionary [AE08]. We will present an Online Sparse Dictionary Learning (OSDL) algorithm to effectively train the double-sparsity model. This approach enable us to handle very large training sets while using high dimensional signals, achieving faster convergence than the batch alternative and providing a better treatment of local minima, which are abundant in non-convex dictionary learning problems.

4.2 Sparse Dictionaries

Learning dictionaries for large signals requires adding some constraint to the dictionary, otherwise signal diversity and the number of training examples needed make the problem intractable. Often, these constraints are given in terms of a certain structure. One such approach is the double-sparsity model [RZE10]. In this model the dictionary is assumed to be a multiplication of a fixed operator $\boldsymbol{\Phi}$ (we will refer to it as the base dictionary) by a sparse adaptable matrix $\mathbf{A}$. Every atom in the effective dictionary $\mathbf{D}$ is therefore a linear combination of few and arbitrary atoms from the base dictionary. Formally, this means that the training procedure requires

solving the following problem:

$$\min_{\mathbf{A},\mathbf{\Gamma}} \frac{1}{2}\|\mathbf{Y}-\mathbf{\Phi}\mathbf{A}\mathbf{\Gamma}\|_F^2 \quad \text{s.t.} \quad \begin{cases} \|\boldsymbol{\gamma}_i\|_0 \leq p & \forall i \\ \|\mathbf{a}_j\|_0 = k & \forall j \end{cases}. \tag{4.2}$$

Note that the number of columns in $\mathbf{\Phi}$ and $\mathbf{A}$ might differ, allowing flexibility in the redundancy of the effective dictionary. The authors in [RZE10] used an over-complete Discrete Cosine Transform (ODCT) as the base dictionary in their experiments. Using Wavelets was proposed but never implemented due both to implementation issues (the traditional Wavelet transform is not entirely separable) and to the significant border-effects Wavelets have in small-to-medium sized patches. We address both of these issues in the following section.

As for the training of such a model, the update of the dictionary is now constrained by the number of non-zeros in the columns of $\mathbf{A}$. In [RZE10] a variant of the K-SVD algorithm (termed Sparse K-SVD) was proposed for updating the dictionary. As the work in [AEB06], this is a batch method that updates every atom sequentially. In the context of the double-sparsity structure, this task is converted into a sparse-coding problem, and approximated by the greedy OMP algorithm.

In the recent inspiring work reported in [LMG15] the authors extended the double-sparsity model to a scenario where the base dictionary itself is a multiplication of several sparse matrices, that are to be learned. While this structure allows for a clear decrease in the computational cost of applying the dictionary, its capacity to treat medium-size problems is not explored. The proposed algorithm involves a hierarchy of matrix factorizations with multiple parameters to be set, such as the number of levels and the sparsity of each level.

4.3 Cropped Wavelets

The double sparsity model relies on a base-dictionary which should be computationally efficient to apply. The ODCT dictionary has been used for this purpose in [RZE10], but its applicability to larger signal sizes is weak. Indeed, as the patch size grows – getting closer to an image size – the more desirable a multi-scale analysis framework becomes. The separability of the base dictionary provides a further decrease in the computational complexity. Applying two (or more) 1D dictionaries on each dimension separately is typically much more efficient than an equivalent non-separable multi-dimensional dictionary. We will combine these two characteristics as guidelines in the design of the base dictionary for our model.

4.3.1 Optimal Extensions and Cropped Wavelets

The two dimensional Wavelet transform has shown to be very effective in sparsifying natural (normal sized) images. When used to analyze small or medium sized images, not only is the number of possible decomposition scales limited, but more importantly the border effects become a serious limitation. Other works have pointed out the importance of the boundary conditions in the context of deconvolution [AF13, Ree05]. However, our approach is different from these, as

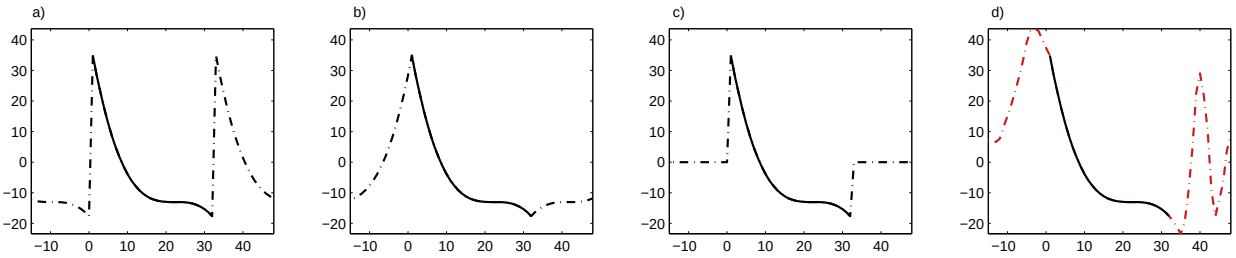


Figure 4.1: Different border treatments: a) periodic, b) symmetric, c) zero-padding, and d) the resulting optimized extension signal $\bar{\mathbf{f}} = \mathbf{W}_s \mathbf{g}_w$.

we will focus on the basis elements rather than on the signal boundaries, and in the pursuit of the corresponding coefficients.

In order to build (bi-)orthogonal Wavelets over a finite (and small) interval, one usually assumes their periodic or symmetric extension onto an infinite axis. A third alternative, zero-padding, assumes the signal is zero outside of the interval. However, none of these alternatives provides an optimal approximation of the signal borders. In general, all these methods do not preserve their vanishing moments at the boundary of the interval, leading to additional non-zero coefficients corresponding to the basis functions that overlap with the boundaries [Mal08]. An alternative is to modify the Wavelet filters such that they preserve their vanishing moments at the borders of the interval, although constructing such Wavelets while preserving their orthogonality is complicated [CDV93].

We begin our derivation by looking closely at the zero-padding case. Let $\mathbf{f} \in \mathbf{R}^n$ be a finite signal. Consider $\bar{\mathbf{f}} = \mathbf{P}\mathbf{f}$, the zero-padded version of $\mathbf{f}$, where $\mathbf{P} \in \mathbf{R}^{L \times n}$, $L > n$ (L is "big enough"). Considering the Wavelet analysis matrix $\mathbf{W}_a$ of size $L \times L$, the Wavelet representation coefficients are obtained by applying the Discrete Wavelet Transform (DWT) to $\bar{\mathbf{f}}$, which can be written as $\mathbf{g}_w = \mathbf{W}_a \bar{\mathbf{f}}$. Note that this is just a projection of the (zero-padded) signal onto the orthogonal Wavelet atoms. As for the inverse transform, the padded signal is recovered by applying the inverse Wavelet transform or Wavelet synthesis operator $\mathbf{W}_s$ ($\mathbf{W}_s = \mathbf{W}_a^T$, assuming orthogonal Wavelets), of size $L \times L$ to the coefficients $\mathbf{g}_w$. Lastly, the padding is discarded (multiplying by $\mathbf{P}^T$) to obtain the final signal in the original finite interval:

$$\hat{\mathbf{f}} = \mathbf{P}^T \mathbf{W}_s \mathbf{g}_w = \mathbf{P}^T \mathbf{W}_s \left(\mathbf{W}_a \mathbf{P} \mathbf{f} \right) = \mathbf{f}.$$

Zero-padding is not an option of preference because it introduces discontinuities in the function $\bar{\mathbf{f}}$ that result in large (and many) Wavelet coefficients, even if $\mathbf{f}$ is smooth inside the finite interval. This phenomenon can be understood from the following perspective: we are seeking the representation vector $\mathbf{g}_w$ that will satisfy the perfect reconstruction of $\mathbf{f}$,

$$\mathbf{P}^T \mathbf{W}_s \mathbf{g}_w = \mathbf{f}.$$

The matrix $\mathbf{P}^T \mathbf{W}_s$ serves here as the effective dictionary that multiplies the representation in order to recover the signal. This relation is an under-determined linear system of equations with

n equations and L unknowns, and thus it has infinitely many possible solutions.

In fact, zero padding chooses a very specific solution to the above system, namely, $\mathbf{g}_w = \mathbf{W}_a\mathbf{P}\mathbf{f}$. This is nothing but the projection of the signal onto the adjoint of the above-mentioned dictionary, since $\mathbf{W}_a\mathbf{P} = (\mathbf{P}^T\mathbf{W}_s)^T$. While this is indeed a feasible solution, such a solution is expected to have many non-zeros if the atoms are strongly correlated. This indeed occurs for the finite-support Wavelet atoms that intersect the borders, and which are cropped by $\mathbf{P}^T$.

To overcome this problem, we propose the following alternative optimization objective:

$$\mathbf{g}_w = \arg\min_{\mathbf{g}} \|\mathbf{g}\|_0 \quad \text{s.t.} \quad \mathbf{P}^T\mathbf{W}_s\mathbf{g} = \mathbf{f},$$

i.e., seeking the sparsest solution to this under-determined linear system. Note that in performing this pursuit, we are implicitly extending the signal $\mathbf{f}$ to become $\bar{\mathbf{f}} = \mathbf{W}_s\mathbf{g}_w$, which is the smoothest possible with respect to the Wavelet atoms (i.e., it is sparse under the Wavelet transform). At the same time, we keep using the original Wavelet atoms with all their properties, including their vanishing moments. On the other hand, we pay the price of performing a pursuit instead of a simple back-projection. In particular, we use OMP to approximate the solution to this sparse coding problem. To conclude, our treatment of the boundary issue is obtained by applying the cropped Wavelets dictionary $\mathbf{W}_c = \mathbf{P}^T\mathbf{W}_s$, and seeking the sparsest representation with respect to it, implicitly obtaining an extension of $\mathbf{f}$ without boundary problems.

To illustrate our approach, in Fig. 4.1 we show the typical periodic, symmetric and zero-padding border extensions applied to a random smooth function, as well as the ones obtained by our method. As can be seen, this extension – which is nothing else than Wavelet atoms that *fit* in the borders in a natural way – guarantees not to create discontinuities which result in denser representations[1]. Note that we will not be interested in the actual extensions explicitly in our work.

To provide further evidence on the better treatment of the borders by the cropped Wavelets, we present the following experiment. We construct 1,000 random smooth functions $\mathbf{f}$ of length 64 (3rd degree polynomials), and introduce a random step discontinuity at sample 32. These signals are then normalized to have unit l_2-norm. We approximate these functions with only 5 Wavelet coefficients, and measure the energy of the point-wise (per sample) error (in l_2-sense) of the reconstruction. Fig. 4.2 shows the mean distribution of these errors[2]. As expected, the discontinuity at the center introduces a considerable error. However, the traditional (periodic) Wavelets also exhibit substantial errors at the borders. The proposed cropped Wavelets, on the other hand, manage to reduce these errors by avoiding the creation of extra discontinuities.

Practically speaking, the proposed cropped Wavelet dictionary can be constructed by taking a Wavelet synthesis matrix for signals of length L and cropping it. Also, and because we will be

[1] A similar approach was presented in [ZM00] in the context of compression. The authors proposed to optimally extend the borders of an irregular shape in the sense of minimal l_1-norm of the representation coefficients under a DCT transform.

[2] The m-term approximation with Wavelets is performed with the traditional non-linear approximation scheme. In this framework, orthogonal Wavelets with periodic extensions perform better than symmetric extensions or zero-padding, which we therefore omit from the comparison. We used for this experiment Daubechies Wavelets with 13 taps. All random variables were chosen from Gaussian distributions.

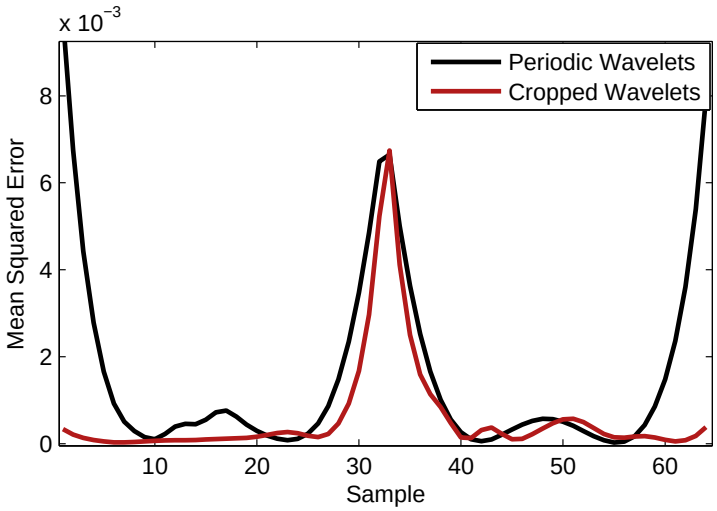


Figure 4.2: Mean approximation (using 5 coefficients) error *per sample* of smooth functions of length 64 with a discontinuity at sample 32.

making use of greedy pursuit methods, each atom is normalized to have unit l_2 norm. This way, the cropped Wavelets dictionary can be expressed as

$$\mathbf{\Phi}_1^c = \mathbf{P}^T \, \mathbf{W}_s \, \boldsymbol{\mathcal{W}},$$

where $\boldsymbol{\mathcal{W}}$ is a diagonal matrix of size $L \times L$ with values such that each atom (column) in $\mathbf{\Phi}_1^c$ (of size $n \times L$) has a unit norm[3]. The resulting transform is no longer orthogonal, but this – now redundant – Wavelet dictionary solves the borders issues of traditional Wavelets enabling for a lower approximation error.

Just as in the case of zero-padding, the redundancy obtained depends on the dimension of the signal, the number of decomposition scales and the length of the support of the Wavelet filters (refer to [Mal08] for a thorough discussion). In practice, we set $L = 2^{\lceil log_2(n) \rceil + 1}$; i.e, twice the closest higher power of 2 (which reduces to $L = 2n$ if n is a power of two, yielding a redundancy of at most 2) guaranteeing a sufficient extension of the borders.

4.3.2 A Separable 2-D Extension

The one-dimensional Wavelet transform is traditionally extended to treat two-dimensional signals by constructing two-dimensional atoms as the separable product of two one-dimensional ones, per scale [Mal08]. This yields three two-dimensional Wavelet functions at each scale j, implying a decomposition which is only separable per scale. This means cascading this two-dimensional transform on the approximation band at every scale.

An alternative extension is a completely separable construction. Considering all the basis

[3]Because the atoms in $\mathbf{W}_s$ are compactly supported, some of them may be identically zero in the central n samples. These are discarded in the construction of $\mathbf{\Phi}_1^c$.

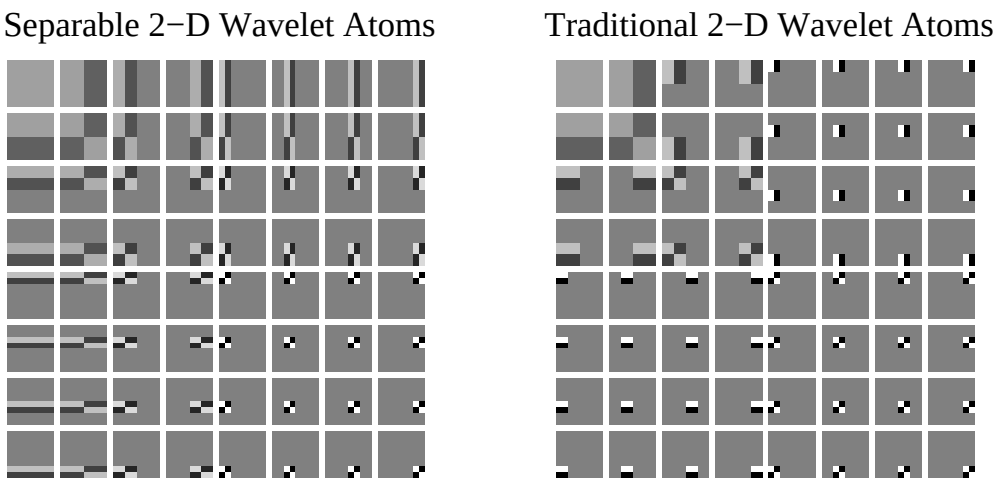


Figure 4.3: 2-D atoms of the Wavelet (Haar) transform for patches of size 8×8 – the separable versus the the traditional construction.

elements of the 1-D DWT (in *all* scales) arranged column-wise in the matrix $\mathbf{\Phi}_1$, the 2-D separable transform can be represented as the Kronecker product $\mathbf{\Phi}_2 = \mathbf{\Phi}_1 \otimes \mathbf{\Phi}_1$. This way, all properties of the transform $\mathbf{\Phi}_1$ translate to each of the dimensions of the 2-dimensional signal on which $\mathbf{\Phi}_2$ is applied. Now, instead of cascading down a two-dimensional decomposition, the same 1-D Wavelet transform is applied first to all the columns of the image and then to all the rows of the result (or vice versa). In relatively small images, this alternative is simpler and faster to apply compared to the traditional cascade. This modification is not only applicable to the traditional Wavelet transform, but also to the cropped Wavelets dictionary introduced above. In this 2-D set-up, both vertical and horizontal borders are implicitly extended to provide a sparser Wavelet representation.

We present in Fig. 4.3 the 2-D atoms of the Wavelet (Haar) Transform for signals of size 8×8 as an illustrative example. The atoms corresponding to the coarsest decomposition scale and the diagonal bands are the same in both separable and non-separable constructions. The difference appears in the vertical and horizontal bands (at the second scale and below). In the separable case we see elongated atoms, mixing a low scale in one direction with high scale in the other.

4.3.3 Approximation of Real-World Signals

While it is hard to rank the performance of separable versus non-separable *analytical* dictionaries or transforms in the general case, we have observed that the separable Wavelet transform provides sparser representations than the traditional 2-D decomposition on small-medium size images. To demonstrate this, we take 1,000 image patches of size 64×64 from popular test images, and compare the m-term approximation achieved by the regular two-dimensional Wavelet transform, the completely separable Wavelet transform and our separable and cropped Wavelets. A small

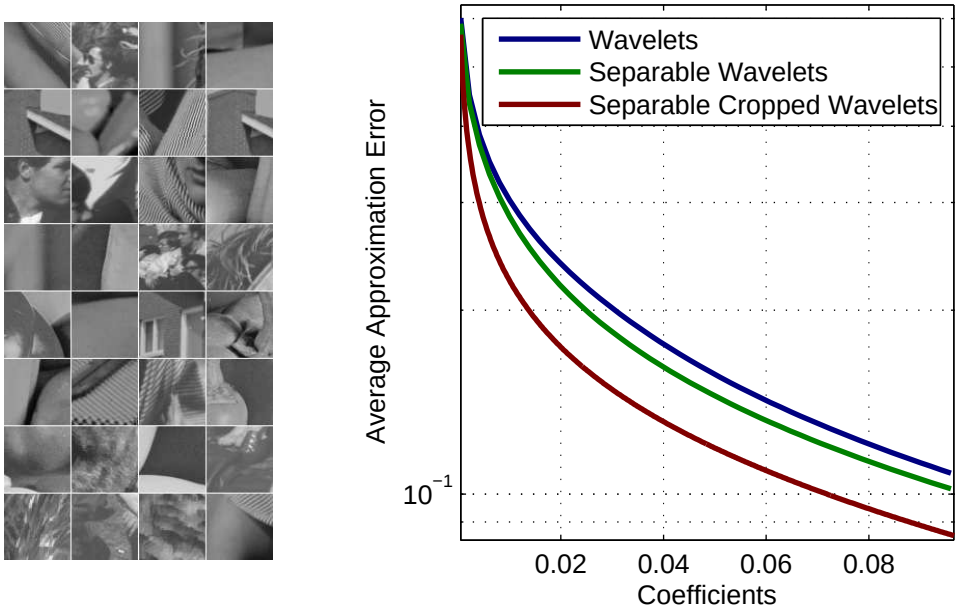


Figure 4.4: Left: Random set of some of the images used for the the M-Term approximation Experiment. Right: M-Term approximation by the traditional 2-D Wavelets and the separable and cropped Wavelets on real images of size 64×64.

subset of these patches is presented on the left of Fig. 4.4. These large patches are in themselves small images, exhibiting the complex structures characteristic of real world images.

As we see from the results in Fig. 4.4 (right), the separability provides some advantage over regular Wavelets in representing the image patches. Furthermore, the proposed separable cropped Wavelets give an even better approximation of the data with fewer coefficients.

Before concluding this section, we make the following remark. It is well known that Wavelets (separable or not) are far from providing an optimal representation for general images [Mal08, CD00, DV05]. Nonetheless, in this work these basis functions will be used only as the base dictionary, while our learned dictionary will consist of linear combinations thereof. It is up to the learning process to close the gap between the sub-optimal representation capability of the Wavelets, and the need for a better two dimensional representation that takes into account edge orientation, scale invariance, and more.

4.4 Online Sparse Dictionary Learning

As seen previously, the *de-facto* method for training the doubly sparse model has been a batch-like process. When working with higher dimensional data, however, the required amount of training examples and the corresponding computational load increase. In this *big-data* (or *medium-data*) scenario, it is often unfeasible or undesired to perform several sweeps over the entire data set. In some cases, the dimensionality and amount of data might restrict the learning process to only a couple of iterations. In this regime of work it may be impossible to even store all training samples in memory during the training process. In an extreme online learning set-up, each data sample is

seen only once as new data flows in. These reasons lead naturally to the formulation of an online training method for the double-sparsity model. In this section, we first introduce a dictionary learning method based on the Normalized Iterative Hard-Thresholding algorithm [BD10]. We then use these ideas to propose an Online Sparse Dictionary Learning (OSDL) algorithm based on the popular Stochastic Gradient Descent technique, and show how it can be applied efficiently to our specific dictionary learning problem.

4.4.1 NIHT-based Dictionary Learning

A popular practice in dictionary learning, which has been shown to be quite effective, is to employ a block coordinate minimization over this non-convex problem. This often reduces to alternating between a sparse coding stage, throughout which the dictionary is held constant, and a dictionary update stage in which the sparse coefficients (or their support) are kept fixed. We shall focus on the second stage, as the first remains unchanged, essentially applying sparse coding to a group of examples. Embarking from the objective as given in Equation (4.2), the problem to consider in the dictionary update stage is the following:

$$\min_{\mathbf{A}} \underbrace{\frac{1}{2}||\mathbf{Y}-\mathbf{\Phi}\mathbf{A}\mathbf{\Gamma}||_F^2}_{f(\mathbf{A})} \quad \text{s.t.} \quad ||\mathbf{a}_j||_0 = k \quad \forall j,$$

where $\mathbf{\Phi}$ is the base dictionary of size $n \times L$ and $\mathbf{A}$ is a matrix of size $L \times m$ with k non-zeros per column. Many dictionary learning methods undertake a sequential update of the atoms in the dictionary ([AEB06, MBPS10, RZE10]). Following this approach, we can consider m minimization problems of the following form:

$$\min_{\mathbf{a}_j} \underbrace{\frac{1}{2}||\mathbf{E}_j-\mathbf{\Phi}\mathbf{a}_j\boldsymbol{\gamma}_j^T||_F^2}_{f(\mathbf{a}_j)} \quad \text{s.t.} \quad ||\mathbf{a}_j||_0 = k, \tag{4.3}$$

where $\mathbf{E}_j$ is the error given by $\mathbf{Y}-\sum_{i\neq j}\mathbf{\Phi}\mathbf{a}_i\boldsymbol{\gamma}_i^T$ and $\boldsymbol{\gamma}_i^T$ denotes the i-th row of $\mathbf{\Gamma}$. This problem produces the j-th column in $\mathbf{A}$, and thus we sweep through $j = 1, \ldots, m$ to update all of $\mathbf{A}$.

The Normalized Iterative Hard-Thresholding (NIHT) [BD10] algorithm is a popular sparse coding method in the context of Compressed Sensing [BD08]. This method can be understood as a projected gradient descent algorithm. We can propose a dictionary update based on the same concept. Note that we could rewrite the cost function in Equation (4.3) as $f(\mathbf{a}_j) = \frac{1}{2}||\mathbf{E}_j-\mathcal{H}_j\mathbf{a}_j||_F^2$, for an appropriate operator $\mathcal{H}_j$. Written in this way, we can perform the dictionary update in terms of the NIHT by iterating:

$$\mathbf{a}_j^{t+1} = \mathcal{P}_k\left[\mathbf{a}_j^t - \eta_j^t\,\mathcal{H}_j^*\left(\mathbf{E}_j - \mathcal{H}_j\mathbf{a}_j^t\right)\right], \tag{4.4}$$

where $\mathcal{H}_j^*$ is the adjoint of $\mathcal{H}_j$, $\mathcal{P}_k$ is a Hard-Thresholding operator that keeps the k largest non-zeros (in absolute value), and η_j^t is an appropriate step-size. This algorithm iterates over Equation (4.4) until convergence *per* atom in the dictionary update stage.

The choice of the step size is critical. Noting that $\mathcal{H}_j^* \left(\mathbf{Y} - \mathcal{H}_j \mathbf{a}_j \right) = \nabla f(\mathbf{a}_j)$, in [BD10] the authors propose to set this parameter per iteration as:

$$\eta_j^t = \frac{\| \nabla f(\mathbf{a}_j^t)_{S_j} \|_F^2}{\| \mathcal{H} \nabla f(\mathbf{a}_j^t)_{S_j} \|_F^2}, \tag{4.5}$$

where S_j denotes the support of $\mathbf{a}_j^t$. With this step size, the estimate $\hat{\mathbf{a}}^{t+1}$ is obtained by performing a gradient step and hard-thresholding as in Equation (4.5). Note that if the support of $\hat{\mathbf{a}}_j^{t+1}$ and $\mathbf{a}_j^t$ are the same, setting η_j^t as in Equation (4.5) is indeed optimal, as it is the minimizer of the quadratic cost w.r.t. η_j^t. In this case, we simply set $\mathbf{a}_j^{t+1} = \hat{\mathbf{a}}_j^{t+1}$. If the support changes after applying $\mathcal{P}_k$, however, the step-size must be diminished until a condition is met, guaranteeing a decrease in the cost function[4]. Following this procedure, the work reported in [BD10] shows that the algorithm in Equation (4.4) is guaranteed to converge to a local minimum of the problem in Equation (4.3).

Consider now the algorithm given by iterating between 1) sparse coding of all examples in $\mathbf{Y}$, and 2) atom-wise dictionary update with NIHT in Equation (4.3). An important question that arises is: will this simple algorithm converge? Let us assume that the pursuit succeeds, obtaining the sparsest solution for a given sparse dictionary $\mathbf{A}$, which can indeed be guaranteed under certain conditions. Moreover, pursuit methods like OMP, Basis Pursuit and FOCUSS perform very well in practice when $k \ll n$ (refer to [BDE09] for a thorough review). For the cases where the theoretical guarantees are not met, we can adopt an external interference approach by comparing the best solution using the support obtained in the previous iteration to the one proposed by the new iteration of the algorithm, and choosing the best one. This small modification guarantees a decrease in the cost function at every sparse coding step. The atom-wise update of the dictionary is also guaranteed to converge to a local minimum for the above mentioned choice of step sizes. Performing a series of these alternating minimization steps ensures a monotonic reduction in the original cost function in Equation (4.1), which is also bounded from below, and thus convergence to a fixed point is guaranteed.

Before moving on, a word on the characteristics of trained dictionaries is in place. The recovery guarantees of pursuit methods is generally formulated in terms of properties of the dictionary, such as its mutual coherence or its Restricted Isometry Property (RIP) [Ela10]. While dictionary learning provides better and sparser representations for real data, this adaptive process generally deteriorates these measures. A trained dictionary does not typically exhibit low correlation between its atoms, and so the corresponding results (which are worst-case scenario analyses) say very little about the quality of the obtained dictionary. As we shall see in the results section, this does not imply a deterioration of its practical performance; on the contrary, their effectiveness in image representation and restoration is greatly improved.

[4]The step size is decreased by $\eta_j^t = c\, \eta_j^t$, where $c < 1$. We refer the reader to [BD08] and [BD10] for further details.

Algorithm 4.1 Stochastic NIHT for Sparse Dictionary Learning.

Initialization: Training samples $\{\mathbf{y}_i\}$, base-dictionary $\mathbf{\Phi}$, initial sparse matrix $\mathbf{A}^0$

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

  Draw $\mathbf{y}_i$ at random

  $\boldsymbol{\gamma}_i \leftarrow$ Sparse Code $(\mathbf{y}_i, \mathbf{\Phi}, \mathbf{A}^i)$

  $\mathcal{S}_i = Support(\boldsymbol{\gamma}_i)$

  for $j = 1, \ldots, |\mathcal{S}_i|$ **do**

    Update $\mathbf{a}_{\mathcal{S}(j)}^{i+1}$ with Equation (4.6) and step size $\frac{\eta^*_{\mathcal{S}(j)}}{1+i/T}$

  end

end

Result: Sparse Dictionary $\mathbf{A}$

4.4.2 From Batch to Online Learning

As noted in [AE08, MBPS10], it is not compulsory to accumulate all the examples to perform an update in the gradient direction. Instead, we turn to a stochastic (projected) gradient descent approach. In this scheme, instead of computing the expected value of the gradient by the sample mean over all examples, we estimate this gradient over a single randomly chosen example $\mathbf{y}_i$. We then update the atoms of the dictionary based on this estimation using:

$$\mathbf{a}_j^{t+1} = \mathcal{P}_k \left[\mathbf{a}_j^t - \eta^t \, \nabla f \left(\mathbf{a}_j^t, \mathbf{y}_i, \boldsymbol{\gamma}_i, \right) \right]. \tag{4.6}$$

Since these updates might be computationally costly (and because we are only performing an alternating minimization over problem (4.2)), we might stop after a few iterations of applying Equation (4.6). We also restrict this update to those atoms that are used by the current example (since others have no contribution in the corresponding gradient). In addition, instead of employing the step size suggested by the NIHT algorithm, we employ the common approach of using decreasing step sizes throughout the iterations, which has been shown beneficial in stochastic optimization [Bot98]. To this end, and denoting by η_j^* the step size resulting from the NIHT, we employ an effective learning rate of $\frac{\eta_j^*}{1+t/T}$, with a manually set parameter T. This modification does not compromise the guarantees of a decrease in the cost function (for the given random sample i), since this factor is always smaller than one. We outline the basic stages of this method in Algorithm 4.1.

An important question that now arises is whether shifting from a batch training approach to this online algorithm preserves the convergence guarantees described above. Though plenty is known in the field of stochastic approximations, most of the existing results address convergence guarantees for convex functions, and little is known in this area regarding projected gradient algorithms [BB08]. For non-convex cases, convergence guarantees still demand the cost function to be differentiable with continuous derivatives [Bot98]. In our case, the l_0 pseudo-norm makes a proof of convergence challenging, since the problem becomes not only non-convex but also (highly) discontinuous.

Algorithm 4.2 Online Sparse Dictionary Learning (OSDL) algorithm.

Initialization: Training samples $\{\mathbf{y}_i\}$, base-dictionary $\mathbf{\Phi}$, initial sparse matrix $\mathbf{A}^0$

$\mathbf{G_\Phi} = \mathbf{\Phi}^T\mathbf{\Phi}$; $\mathbf{U} = \mathbf{0}$

for $t = 1, \ldots, T$ **do**

Draw a mini-batch $\mathbf{Y}_t$ at random

$\mathbf{\Gamma}_t \leftarrow$ Sparse Code $(\mathbf{Y}_t, \mathbf{\Phi}, \mathbf{A}^t, \mathbf{G}^t)$

$\eta^t = ||\nabla f(\mathbf{A}^t_\mathcal{S})||_F / \|\mathbf{\Phi}\nabla f(\mathbf{A}^t_\mathcal{S})\mathbf{\Gamma}^\mathcal{S}_t\|_F$

$\mathbf{U}^{t+1}_\mathcal{S} = \gamma\mathbf{U}^t_\mathcal{S} + \eta^t \ \nabla f(\mathbf{A}^t_\mathcal{S})$

$\mathbf{A}^{t+1}_\mathcal{S} = \mathcal{P}_k\left[\mathbf{A}^t_\mathcal{S} - \mathbf{U}^{t+1}_\mathcal{S}\right]$

Update columns and rows of $\mathbf{G}$ by $\left(\mathbf{A}^{t+1}\right)^T \mathbf{G_\Phi} \mathbf{A}^{t+1}_\mathcal{S}$

end

Result: Sparse Dictionary $\mathbf{A}$

That said, one could re-formulate the dictionary learning problem using a non-convex but continuous and differentiable penalty function, moving from a constrained optimization problem to an unconstrained one. We conjecture that convergence to a fixed point of this problem can be reached under the mild conditions described in [Bot98]. Despite these theoretical benefits, we choose to maintain our initial formulation in terms of the l_0 measure for the sake of simplicity (note that we need no parameters other than the target sparsity). Practically, we saw in all our experiments that convergence is reached, providing numerical evidence for the behavior of our algorithm.

4.4.3 OSDL In Practice

We now turn to describe a variant of the method described in Algorithm 4.1, and outline other implementation details. The atom-wise update of the dictionary, while providing a specific step-size, is computationally slower than a global update. In addition, guaranteeing a decreasing step in the cost function implies a line-search per atom that is costly. For this reason we propose to replace this stage by a global dictionary update of the form

$$\mathbf{A}^{t+1} = \mathcal{P}_k\left[\mathbf{A}^t - \eta^t \ \nabla f(\mathbf{A}^t)\right],$$

where the thresholding operator now operates in each column of its argument. While we could maintain a NIHT approach in the choice of the step-size in this case as well, we choose to employ

$$\eta^\star = \frac{||\nabla f(\mathbf{A}_S)||_F}{||\mathbf{\Phi}\nabla f(\mathbf{A}_S)X||_F}. \tag{4.7}$$

Note that this is the square-root of the value in Equation (4.5) and it may appear as counter-intuitive. We shall present a numerical justification of this choice in the following section.

Secondly, instead of considering a single sample $\mathbf{y}_t$ per iteration, a common practice in

stochastic gradient descent algorithms is to consider mini-batches $\{\mathbf{y}_i\}$ of N examples arranged in the matrix $\mathbf{Y}_t$. As explained in detail in [RZE08], the computational cost of the OMP algorithm can be reduced by precomputing (and storing) the Gram matrix of the dictionary $\mathbf{D}$, given by $\mathbf{G} = \mathbf{D}^T\mathbf{D}$. In a regular online learning scheme, this would be infeasible due to the need to recompute this matrix for each example. In our case, however, the matrix needs only to be updated once per mini-batch. Furthermore, only a few atoms get updated each time. We exploit this by updating only the respective rows and columns of the matrix $\mathbf{G}$. Moreover, this update can be done efficiently due to the sparsity of the dictionary $\mathbf{A}$.

Stochastic algorithms often introduce different strategies to regularize the learning process and try to avoid local minimum traps. In our case, we incorporate in our algorithm a momentum term $\mathbf{U}^t$ controlled by a parameter $\gamma \in [0, 1]$. This term helps to attenuate oscillations and can speed up the convergence by incorporating information from the previous gradients. This algorithm, termed Online Sparse Dictionary Learning (OSDL) is depicted in Algorithm 4.2. In addition, many dictionary learning algorithms [AEB06, MBPS10] include the replacement of (almost) unused atoms and the pruning of similar atoms. We incorporate these strategies here as well, checking for such cases once every few iterations.

4.4.4 Complexity Analysis

We now turn to address the computational cost of the proposed online learning scheme[5]. As was thoroughly discussed in [RZE10], the sparse dictionary enables an efficient sparse coding step. In particular, any multiplication by $\mathbf{D}$, or its transpose, has a complexity of $\mathcal{T}_D = \mathbf{\Omega}(km + \mathcal{T}_{\mathbf{\Phi}})$, where m is the number of atoms in $\mathbf{\Phi}$ (assume for simplicity $\mathbf{A}$ square), k is the atom sparsity and $\mathcal{T}_{\mathbf{\Phi}}$ is the complexity of applying the base dictionary. For the separable case, this reduces to $\mathcal{T}_{\mathbf{\Phi}} = \mathbf{\Omega}(n\sqrt{m})$.

Using a sparse dictionary, the sparse coding stage with OMP (in its Cholesky implementation) is $\mathbf{\Omega}(pn\sqrt{m} + pkm)$ per example. Considering N examples in a mini-batch, and assuming $n \propto m$ and $p \propto n$, we obtain a complexity of $\mathbf{\Omega}\left(Nn^2(\sqrt{m} + p)\right)$.

Moving to the update stage in the OSDL algorithm, calculating the gradient $\nabla f(A_{\mathcal{S}})$ has a complexity of $\mathrm{T}_{\nabla f} = \mathbf{\Omega}((k|\mathcal{S}| + n\sqrt{m})N)$, and so does the calculation of the step size. Recall that $\mathcal{S}$ is the set of atoms used by the current samples, and that $|\mathcal{S}| < m$; i.e., the update is applied only on a subset of all the atoms. Updating the momentum variable grows as $\mathbf{\Omega}(|\mathcal{S}|m)$, and the hard thresholding operator is $\mathbf{\Omega}(|\mathcal{S}|mlog(m))$. In a pessimistic approach, assume $|\mathcal{S}| \propto n$.

Putting these elements together, the OSDL algorithm has a complexity of $\mathbf{\Omega}(Nn^2(\sqrt{m} + k) + m^2log(m))$ per mini-batch. The first term depends on the number of examples per mini-batch, and the second one depends only on the size of the dictionary. For high dimensions (large n), the first term is the leading one. Clearly, the number of non-zeros per atom k determines the computational complexity of our algorithm. While in this study we do not address the optimal way of scaling k, experiments shown hereafter suggest that its dependency with n might in

[5] We analyze the complexity of just the OSDL for simplicity. The analysis of Algorithm 4.1 is similar, adding the complexity of the line search of the step sizes.

fact be less than linear. The sparse dictionary provides a computational advantage over the online learning methods using explicit dictionaries, such as [MBPS10], which have complexity of $\mathbf{\Omega}(Nn^3)$.

4.5 Application to Image Processing

In this section we present a number of experiments to illustrate the behaviour of the method presented in the previous section. We start with a detailed experiment on learning an image-specific dictionary. We then move on to demonstrations on image denoising and image compression. Finally we tackle the training of universal dictionaries on millions of examples in high dimensions.

4.5.1 Image-Specific Dictionary Learning

To test the behaviour of the proposed approach, we present the following experiment. We train an adaptive sparse dictionary in three setups of increasing dimension: with patches of size 12×12, 20×20 and 32×32, all extracted from the popular image Lena, using a fixed number of non-zeros in the sparse coding stage (4, 10 and 20 non-zeros, respectively). We also repeat this experiment for different levels of sparsity of the dictionary $\mathbf{A}$. We employ the OSDL algorithm, as well as the method presented in Algorithm 4.1 (in its mini-batch version, for comparison). We also include the results by Sparse K-SVD, which is the classical (batch) method for the double sparsity model, and the popular Online Dictionary Learning (ODL) algorithm [MBPS09]. Note that this last method is an online method that trains a dense (full) dictionary. Training is done on 200,000 examples, leaving 30,000 as a test set.

The sparse dictionaries use the cropped Wavelets as their operator $\mathbf{\Phi}$, built using the Symlet Wavelet with 8-taps. The redundancy of this base dictionary is 1.75 (in 1-D), and the matrix $\mathbf{A}$ is set to be square, resulting in a total redundancy of just over 3. For a fair comparison, we initialize the ODL method with the same cropped Wavelets dictionary. All methods use OMP in the sparse coding stage. Also, note that the ODL[6] algorithm is implemented entirely in C, while in our case this is only true for the sparse coding, giving the ODL somewhat of an advantage in run-time.

The results are presented in Fig. 4.5, showing the representation error on the test set, where each marker corresponds to an epoch. The atom sparsity refers to the number of non-zeros per column of $\mathbf{A}$ with respect to the signal dimension (i.e., 5% in the 12×12 case implies 7 non-zeros). Several conclusions can be drawn from these results. First, as expected, the online approaches provide a much faster convergence than the batch alternative. For the low dimensional case, there is little difference between Algorithm 4.1 and the OSDL, though this difference becomes more prominent as the dimension increases. In these cases, not only does Algorithm 4.1 converge slower but it also seems to be more prone to local minima.

As the number of non-zeros per atom grows, the representation power of our sparse dictionary increases. In particular, OSDL achieves the same performance as ODL for an atom sparsity of 25% for a signal dimension of 144. Interestingly, OSDL and ODL achieve the same performance

[6] We used the publicly available SPArse Modeling Software package, at http://spams-devel.gforge.inria.fr/.

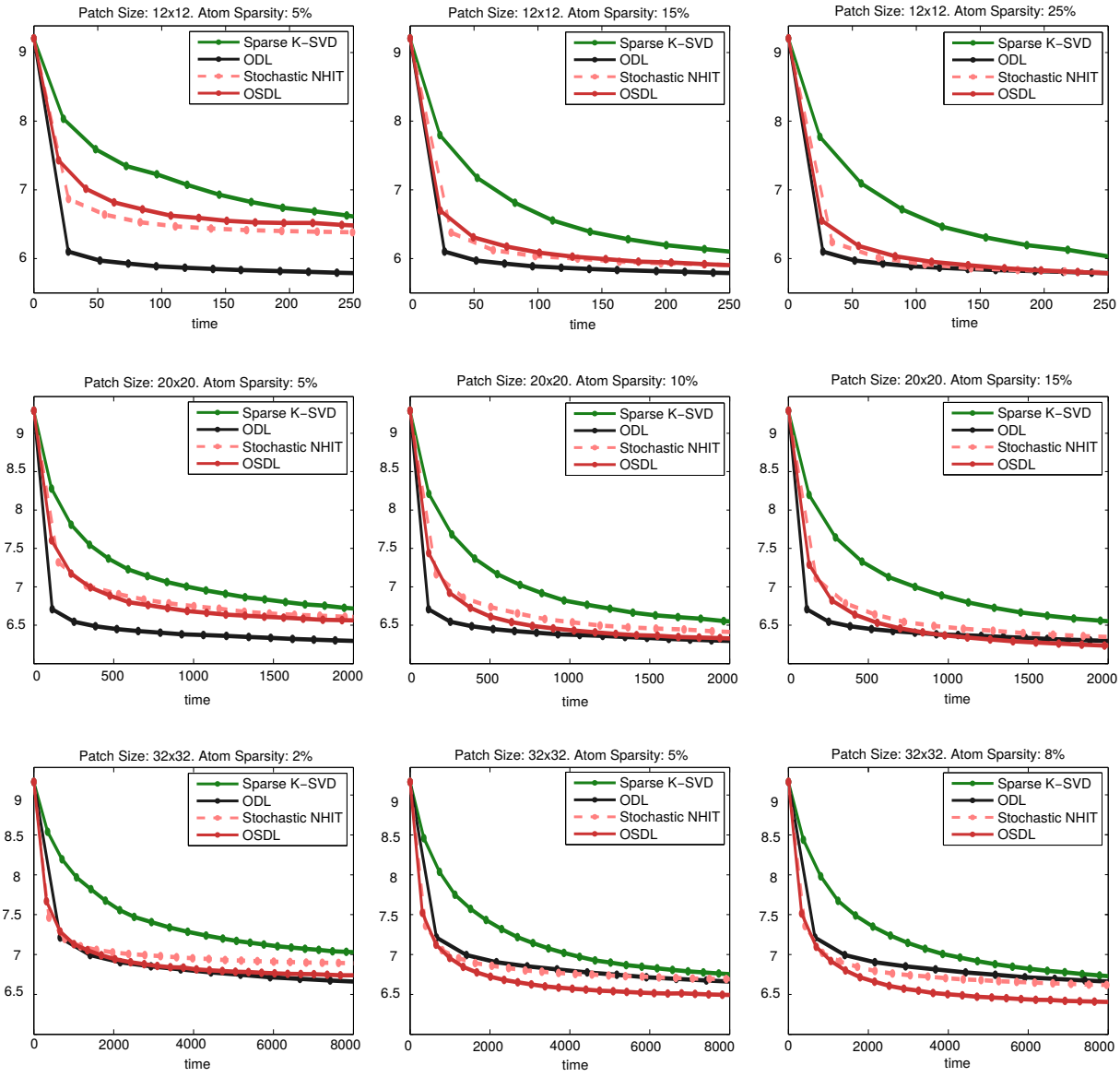


Figure 4.5: Experiment 1: Dictionary learning by Sparse K-SVD, by the Stochastic NIHT presented in Algorithm 4.1, the ODL algorithm [MBPS09] and by the Online Sparse Dictionary Learning (OSDL).

for decreasing number of non-zeros in $\mathbf{A}$ as the dimension increases: 10% for the 20×20 case and $\approx 2\%$ for the 32×32. In this higher dimensional setting, not only does the sparse dictionary provide faster convergence but it also achieves a lower minimum. The lower degrees of freedom of the sparse dictionary prove beneficial in this context[7], where the amount of training data is limited and perhaps insufficient to train a full dictionary. This example suggests that indeed k could grow slower than linearly with the dimension n.

Before moving on, we want to provide some empirical evidence to support the choice of the step size in the OSDL algorithm. In Fig. 4.6 we plot the atom-wise step sizes obtained by Algorithm 4.1, η_j^* (i.e., the optimal values from the NIHT perspective), together with their mean value, as a function of the iterations for the 12×12 case for illustration. In addition, we show the global step sizes of OSDL as in Equation (4.7). As can be seen, this choice provides a fair approximation to the mean of the individual step sizes. Clearly, the square of this value would be too conservative, yielding very small step sizes and providing substantially slower convergence.

4.5.2 Image Restoration Demonstration

In the context of image restoration, most state-of-the-art algorithms take a patch-based approach. While the different algorithms differ in the models they enforce on the corrupted patches (or the prior they chose to consider, in the context a Bayesian formulation) the general scheme remains very much the same: overlapping patches are extracted from the degraded image, then restored more or less independently, before being merged back together by averaging. Though this provides an effective option, this *locally-focused* approach is far from being optimal. As noted in several recent works ([SOE14, SE15, RE15]), not looking at the image as a whole causes inconsistencies between adjacent patches which often result in texture-like artifacts. A possible direction to seek for a more global outlook is, therefore, to allow for bigger patches.

We do not intended to provide a complete image restoration algorithm in this work. Instead, we will show that benefit can indeed be found in using bigger patches in image restoration – given an algorithm which can cope with the dimension increase. We present an image denoising experiment of several popular images, for increasing patch sizes. In the context of sparse representations, an image restoration task can be formulated as a Maximum a Posteriori formulation [EA06]. In the case of a sparse dictionary, this problem can be posed as:

$$\min_{\mathbf{z},\boldsymbol{\gamma}_i,\mathbf{A}} = \frac{\lambda}{2}||\mathbf{z}-\mathbf{y}||_2^2 + \sum_i ||\boldsymbol{\Phi}\mathbf{A}\boldsymbol{\gamma}_i - \mathbf{P}_i\mathbf{z}||_2^2 + \mu_i||\boldsymbol{\gamma}_i||_0,$$

where $\mathbf{z}$ is the image estimate given the noisy observation $\mathbf{y}$, $\mathbf{P}_i$ is an operator that extracts the i^{th} patch from a given image and $\boldsymbol{\gamma}_i$ is the sparse representation of the i^{th} patch. We can minimize this problem by taking a similar approach to that of the dictionary learning problem: use a block-coordinate descent by fixing the unknown image $\mathbf{z}$, and minimizing w.r.t the sparse vectors $\boldsymbol{\gamma}_i$ and the dictionary (by any dictionary learning algorithm). We then fix the sparse

[7] Note that this limitation needed to be imposed for a comparison with Sparse K-SVD. Further along this section we will present a comparison without this limitation.

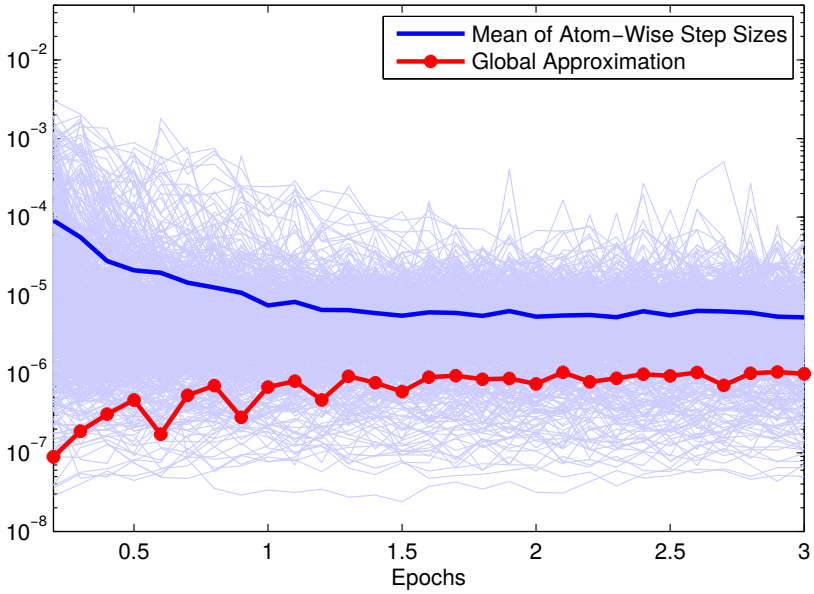


Figure 4.6: Step sizes η_j^* obtained by the atom-wise NIHT algorithm together with their mean value, and the global approximation by OSDL.

vectors and update the image $\mathbf{z}$. Note that even though this process should be iterated (as effectively shown in [SE15]) we stick to the first iteration of this process to make a fair comparison with the K-SVD based algorithms.

For this experiment, denoted as Experiment 4, we use both Sparse K-SVD and OSDL, for training the double sparsity model. Each method is run with the traditional ODCT and with the cropped Wavelets dictionary, presented in this chapter. We include as a reference the results of the K-SVD denoising algorithm [EA06], which trains a regular (dense) dictionary with patches of size 8×8. The dictionary sparsity was set to be 10% of the signal dimension. Regarding the size of the dictionary, the redundancy was determined by the redundancy of the cropped Wavelets (as explained in Section 4.3.1), and setting the sparse matrix $\mathbf{A}$ to be square. This selection of parameters is certainly not optimal. For example, we could have set the redundancy as an increasing function of the signal dimension. However, learning such increasingly redundant dictionaries is limited by the finite data of each image. Therefore, we use a square matrix $\mathbf{A}$ for all patch sizes, leaving the study of other alternatives for future work. 10 iterations were used for the K-SVD methods and 5 iterations for the OSDL.

Fig. 4.7 presents the averaged results over the set of 10 publicly available images used by [LBM13], where the noise standard deviation was set to $\sigma = 30$. Note how the original algorithm presented in [RZE10], Sparse K-SVD with the ODCT as the base dictionary, does not scale well with the increasing patch size. In fact, once the base dictionary is replaced by the cropped Wavelets dictionary, the same algorithm shows a jump in performance of nearly 0.4 dB. A similar effect is observed for the OSDL algorithm, where the cropped Wavelets dictionary performs the best.

Employing even greater patch sizes eventually results in decreasing denoising quality, even for the OSDL with Cropped Wavelets. Partially, this could be caused by a limitation of the

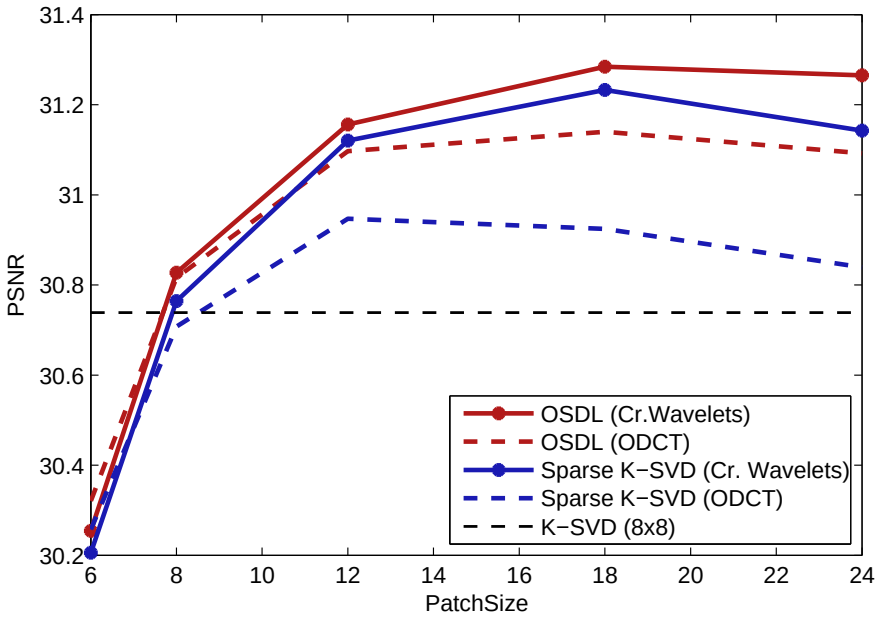


Figure 4.7: Experiment 4: Denoising results as a function of the patch size for Sparse K-SVD and OSDL, which an overcomplete DCT dictionary and a separable cropped Wavelets dictionary.

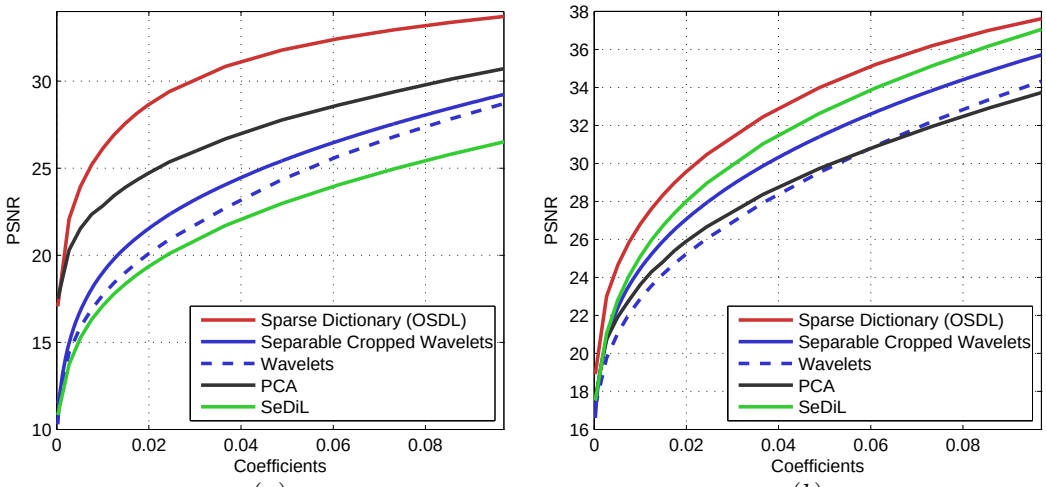


Figure 4.8: Experiment 5: a) Compression results (as in ratio of kept coefficients) by Wavelets, Cropped separable Wavelets, PCA, OSDL and SeDiL [HSK13] on aligned faces. b) Compression results for the "Cropped Labeled Faces in the Wild" database.

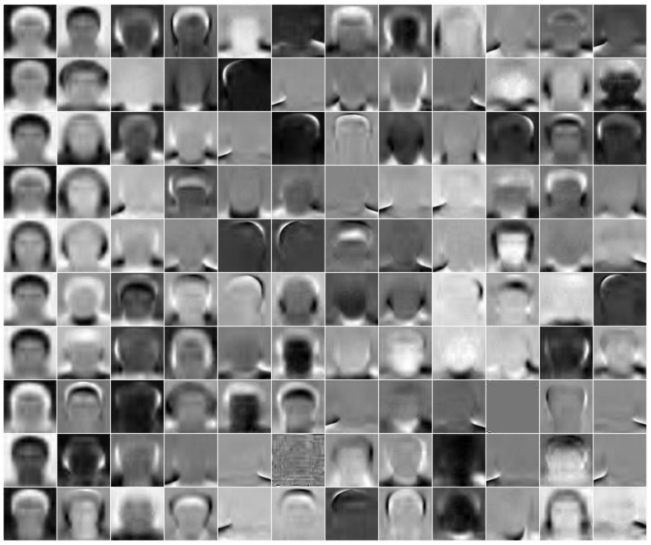


Figure 4.9: Subset of atoms from a sparse dictionary trained with OSDL on a database of aligned face images.

sparse model in representing fine details as the dimension of the signal grows. Also, the amount of training data is limited by the size of the image, having approximately 250,000 examples to train on. Once the dimension of the patches increases, the amount of training data might become a limiting factor in the denoising performance.

As a final word about this experiment, we note that treating all patches the same way (with the same patch size) is clearly not optimal. A multi-size patch approach has already been suggested in [LNDF12], though in the context of the Non-Local Means algorithm. The OSDL algorithm may be the right tool to bring multi-size patch processing to sparse representation-based algorithms, and this remains a topic of future work.

4.5.3 Adaptive Image Compression

Image compression is the task of reducing the amount of information needed to represent an image, such that it can be stored or transmitted efficiently. In a world where image resolution increases at a surprising rate, more efficient compression algorithms are always in demand. In this section, we do not attempt to provide a complete solution to this problem but rather show how our online sparse dictionaries approach could indeed aid a compression scheme.

Most (if not all) compression methods rely on sparsifying transforms. In particular, JPEG2000, one of the best performing and popular algorithms available, is based on the 2-D Wavelet transform. Dictionary learning has already been shown to be beneficial in this application. In [BE08], the authors trained several dictionaries for patches of size 15×15 on pre-aligned face pictures. These offline trained dictionaries were later used to compress images of the same type, by sparse coding the respective patches of each picture. The results reported in [BE08] surpass those by JPEG2000, showing the great potential of similar schemes.

In the experiment we are presenting here (Experiment 5), we go beyond the locally based compression scheme and propose to perform naive compression by just keeping a certain number

of coefficients through sparse coding, where each signal is the entire target image. To this end, we use the same data set as in [BE08] consisting of over 11,000 examples, and re-scaled them to a size of 64×64. We then train a sparse dictionary on these signals with OSDL, using the cropped Wavelets as the base dictionary for 15 iterations. For a fair comparison with other non-redundant dictionaries, in this case we chose the matrix $\mathbf{A}$ such that the final dictionary is non-redundant (a rectangular tall matrix). A word of caution should be said regarding the relatively small training data set. Even though we are training just over 4000 atoms on only 11,000 samples, these atoms are *only* 250-sparse. This provides a great reduction to the degrees of freedom during training. A subset of the obtained atoms can be seen in Fig. 4.9.

For completion, we include here the results obtained by the SeDiL algorithm [HSK13] (with the code provided by the authors and with the suggested parameters), which trains a separable dictionary consisting of 2 small dictionaries of size 64×128. Note that this implies a final dictionary which has a redundancy of 4, though the degrees of freedom are of course limited due to the separability imposed.

The results of this naive compression scheme are shown in Fig. 4.8a for a testing set (not included in the training). As we see, the obtained dictionary performs substantially better than Wavelets – on the order of 8 dB at a given coefficient count. Partially, the performance of our method is aided by the cropped Wavelets, which in themselves perform better than the regular 2-D Wavelet transform. However, the adaptability of the matrix $\mathbf{A}$ results in a much better compression-ratio. A substantial difference in performance is obtained after training with OSDL, even while the redundancy of the obtained dictionary is less (by about half) than the redundancy of its base-dictionary. The dictionary obtained by the SeDiL algorithm, on the other hand, has difficulties learning a completely separable dictionary for this dataset, in which the faces, despite being aligned, are difficult to approximate through separable atoms.

As one could observe from the dictionary obtained by our method, some atoms resemble PCA-like basis elements. Therefore we include the results by compressing the testing images with a PCA transform, obtained from the same training set – essentially, performing a dimensionality reduction. As one can see, the PCA results are indeed better than Wavelets due to the regular structure of the aligned faces, but they are still relatively far from the results achieved by OSDL [BDE09].

Lastly, we show that this naive compression scheme, based on the OSDL algorithm, does not rely on the regularity of the aligned faces in the previous database. To support this claim, we perform a similar experiment on images obtained for the "Cropped Labeled Faces in the Wild Database" [SL09]. This database includes images of subjects found on the web, and its *cropped* version consists of 64×64 images including only the face of the different subjects. These face images are in different positions, orientations, resolutions and illumination conditions. We trained a dictionary for this database, which consists of just over 13,000 examples, with the same parameter as in the previous case, and the compression is evaluated on a testing set not included in the training. An analogous training process was performed with SeDiL. As shown in Fig. 4.8b, the PCA results are now inferior, due to the lack of regularity of the images. The

separable dictionary provided by SeDiL performs better in this dataset, whose examples consists of truncated faces rather than heads, and which can be better represented by separable atoms. Yet, its representation power is compromised by its complete separability when compared to OSDL, with a 1 dB gap between the two.

4.5.4 Pursuing Universal Big Dictionaries

Dictionary learning has shown how to take advantage of sparse representations in specific domains, however dictionaries can also be trained for more general domains (i.e., natural images). For relatively small dimensions, several works have demonstrated that it is possible to train general dictionaries on patches extracted from non-specific natural images. Such general-purpose dictionaries have been used in many applications in image restoration, outperforming analytically-defined transforms.

Using our algorithm we want to tackle the training of such universal dictionaries for image patches of size 32×32, i.e., of dimension 1024. To this end, in this experiment we train a sparse dictionary with a total redundancy of 6: the cropped Wavelets dictionary introduces a redundancy of around 3, and the matrix $\mathbf{A}$ has a redundancy of 2. The atom sparsity was set to 250, and each example was coded with 60 non-zeros in the sparse coding stage. Training was done on 10 Million patches taken from natural images from the Berkeley Segmentation Dataset [MFTM01]. We run the OSDL algorithm for two data sweeps. For comparison, we trained a full (unconstrained) dictionary with ODL with the same redundancy, on the same database and with the same parameters.

We evaluate the quality of such a trained dictionary in an M-Term approximation experiment on 600 patches (or little images). Comparison is done with regular and separable cropped Wavelets (the last one being the base-dictionary of the double sparsity model, and as such the starting point of the training). We also want to compare our results with the approximation achieved by more sophisticated multi-scale transforms, such as Contourlets. Contourlets are a better suited multi-scale analysis for two dimensions, providing an optimal approximation rate for piece-wise smooth functions with discontinuities along twice differentiable curves [DV05]. This is a slightly redundant transform due to the Laplacian Pyramid used for the multi-scale decomposition (redundancy of 1.33). Note that, traditionally, hard-thresholding is used to obtain an M-term approximation, as implemented in the code made available by the authors. However, this is not optimal in the case of redundant dictionaries. We therefore construct an explicit Contourlet synthesis dictionary, and apply the same greedy pursuit we employ throughout the chapter. Thus we fully leverage the approximation power of this transform, making the comparison fair[8].

Moreover, and to provide a complete picture of the different transforms, we include also the

[8] Another option to consider is to use undecimated multi-scale transforms. The Undecimated Wavelet Transform (UDWT) [Mal08] and the Nonsubsampled Contourlet Transform (NSCT) [ER06] are shift-invariant versions of the Wavelet and Contourlet transforms, respectively, and are obtained by skipping the decimation step at each scale. This greater flexibility in representation, however, comes at the cost of a huge redundancy, which becomes a prohibiting factor in any pursuing scheme. A similar undecimated scheme could be proposed for the corresponding cropped transforms, however, but this is out of the scope of this work.

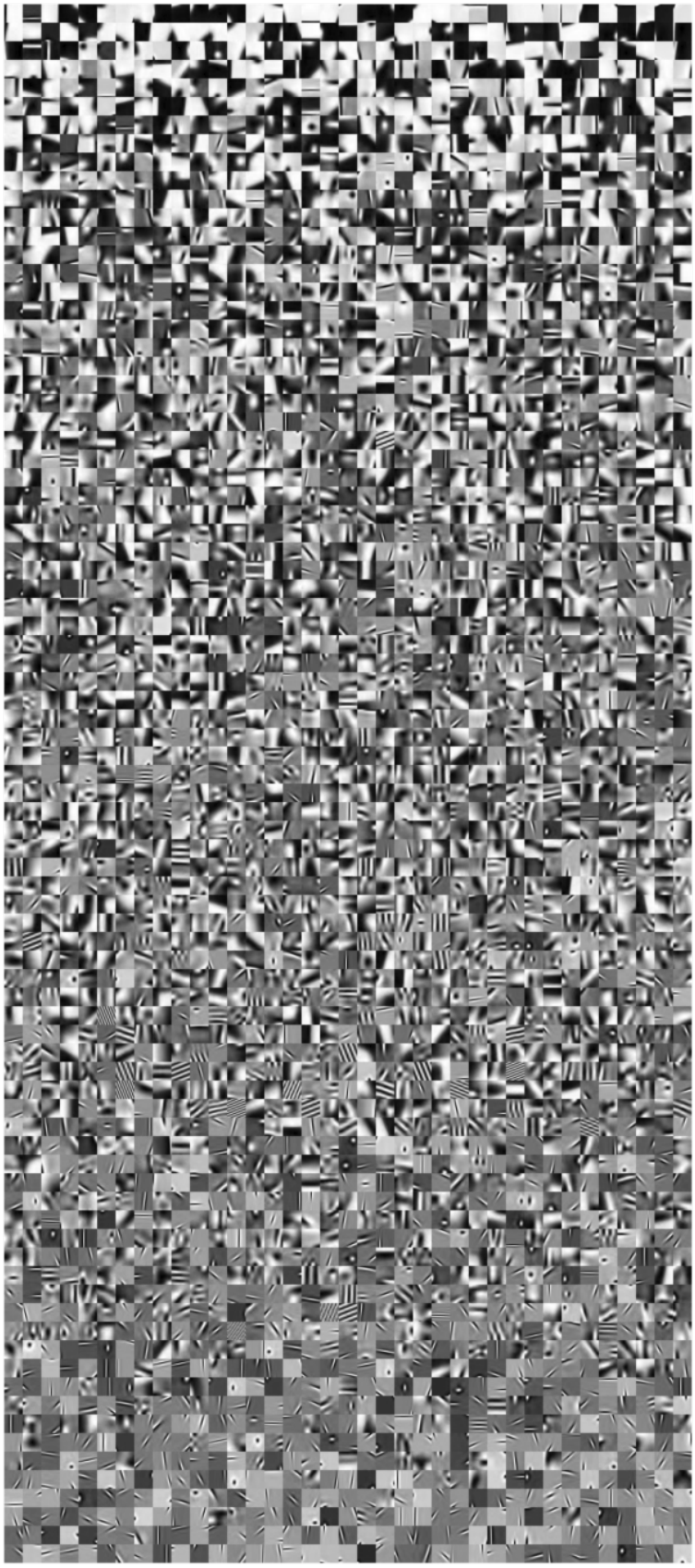


Figure 4.10: Experiment 6: Subset of the general (sparse) dictionary for patches of size 32×32 obtained with OSDL trained over 10 million patches from natural images.

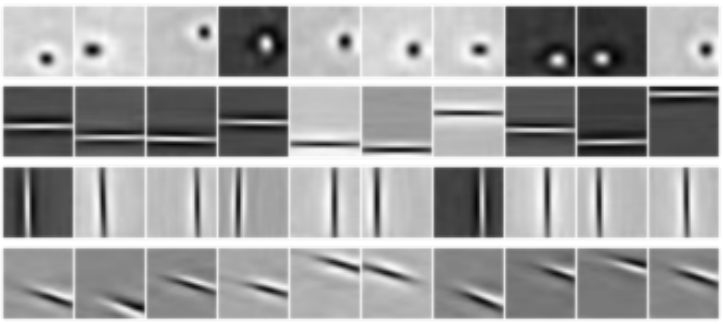


Figure 4.11: Experiment 6: Atoms of size 32×32 with recurring patterns at different locations.

results obtained for a *cropped* version of Contourlets. Since Contourlets are not separable we use a 2-D extension of our cropping procedure detailed in Section 4.3.1 to construct a cropped Contourlets synthesis dictionary. The lack of separability makes this dictionary considerably less efficient computationally. As in cropped Wavelets, we naturally obtain an even more redundant dictionary (redundancy factor of 5.3).

A subset of the obtained dictionary is shown in Fig. 4.10, where the atoms have been sorted according to their entropy. Very different types of atoms can be observed: from the piece-wise-constant-like atoms, to textures at different scales and edge-like atoms. It is interesting to see that Fourier type atoms, as well as Contourlet and Gabor-like atoms, naturally arise out of the training. In addition, such a dictionary obtains some flavor of shift invariance. As can be seen in Fig. 4.11, similar patterns may appear in different locations in different atoms. An analogous question could be posed regarding rotation invariance. Furthermore, we could consider enforcing these, or other, properties in the training. These, and many more questions, are part of on-going work.

The approximation results are shown in Fig. 4.12.a, where Contourlets can be seen to perform slightly better than Wavelets. The cropping of the atoms significantly enhances the results for both transforms, with a slight advantage for cropped Wavelets over cropped Contourlets. The Trainlets, obtained with OSDL, give the highest PSNR. Interestingly, the ODL algorithm by [MBPS10] performs slightly worse than the proposed OSDL, despite the vast database of examples. In addition, the learning (two epochs) with ODL took roughly 4.6 days, whereas the OSDL took approximately 2 days[9]. As we see, the sparse structure of the dictionary is not only beneficial in cases with limited training data (as in Experiment 1), but also in this big data scenario. We conjecture that this is due to the better guiding of the training process, helping to avoid local minima which an uncontrained dictionary might be prone to.

As a last experiment, we want to show that our scheme can be employed to train an adaptive dictionary for even higher dimensional signals. In Experiment 8, we perform a similar training with OSDL on patches (or images) of size 64×64, using an atom sparsity of 600. The cropped

[9]This experiment was run on a 64-bit operating system with an Intel Core i7 microprocessor, with 16 Gb of RAM, in Matlab.

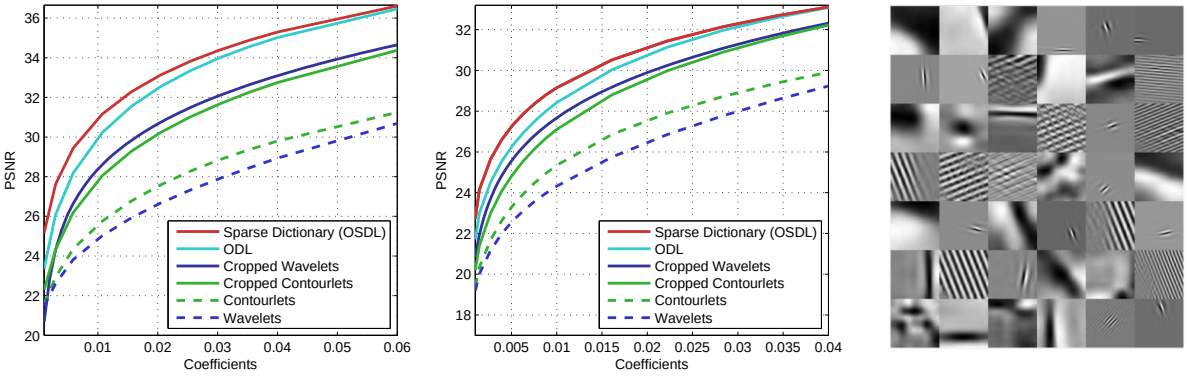


Figure 4.12: Experiment 7-8: a) M-term approximation of general image patches of size 32x32 for different methods. b) M-term approximation of general image patches of size 64x64 for different methods. c) Some atoms of size 64×64 from the dictionary trained with OSDL.

Wavelets dictionary has a redundancy of 2.44, and we set $\mathbf{A}$ to be square. For a fair comparison, and due to the extensive time involved in running ODL, we first ran ODL for 5 days, giving it sufficient time for convergence. During this time ODL accessed 3.8 million training examples. We then ran OSDL using the same examples[10].

As shown in Fig. 4.12.b, the relative performance of the different methods is similar to the previous case. Trainlets again gives the best approximation performance, giving a glimpse into the potential gains achievable when training can be effectively done at larger signal scales. It is not possible to show here the complete trained dictionary, but we do include some selected atoms from it in Fig. 4.12.c. We obtain many different types of atoms: from the very local curvelets-like atoms, to more global Fourier atoms, and more.

4.6 Large Face Image Inpainting

Image inpainting is a data completion problem that aims to recover – or fill in – missing information in an degraded image. These areas can be given by individual missing pixels distributed along the image, or by continuous regions resulting from scratches, foldings or other forms of degradation of old photographs. In the extreme case where the area to inpaint is relatively large (also known as *hole-filling*), this problem becomes challenging [GLM14].

This ill-posed problem, whose solution is often not even well-defined, has received considerable attention in recent years. Many inpainting approaches rely on Partial Differential Equations (PDF) [BSCB00, Tsc06], variational formulations [BBC$^+$01], exemplar-based methods [CPT04], sparsity-enforcing priors [XS10, Gul06] and combinations of them [BBCS10, LMG12]. Despite their efficient performance, all these works are restricted to either small areas or to the task of object removal, by propagating and filling-in a proper surrounding background.

[10]The provided code for ODL is not particularly well suited for cluster-processing (needed for this experiment), and so the times involved in this case should not be taken as an accurate run-time comparison.

Figure 4.13: Example of a inpainted image - left: Face image with missing eyes. Right: inpainted result obtained with the proposed approach.

Some problems, however, require a different approach. We shall focus in the specific problem of inpainting large areas of face images, like the case in Figure 4.13. As one could foresee, traditional patch-based methods will not be effective in recovering or estimating the missing data. Diffusion based and content propagation approaches will also find this problem too challenging. In fact, any method which seeks to inpaint the missing region by propagating information from the available image data will fail, as all these are oblivious to the fact that they are inpainting a face. This missing information needs to be provided in terms of a global model of the target image.

The task of obtaining an adaptive global model for high dimensional signals is a hard problem. Some attempts include manifold learning techniques, as in [Pey09], where the authors propose to learn an adaptable low-dimensional manifold for images. This work includes inpainting examples on synthetic and texture data, though it is still far from providing a practical solution for real world face images. The recent work in [PKD$^+$16], on the other hand, proposes the use of convolutional neural networks to train a global model to inpaint large holes in natural images. This network, however, was trained for general (street) images and it does not apply to our specific problem.

In this work, we propose to build such a global prior employing sparse representations modeling and dictionary learning. The problem of dictionary learning consists of adaptively learning a set of atoms which are able to represent real signals as sparsely as possible, and has been a popular topic in signal and image processing over the last decade [BDE09, MBP14]. However, due to the computational constraints that this problem entails, all learning methods are typically applied on small patches from the image and not on the image itself [AEB06, MBS09]. In other words, attempting to obtain such a global dictionary with traditional dictionary learning algorithms would be infeasible.

A novel work which has circumvented this problem is the recently proposed Trainlets

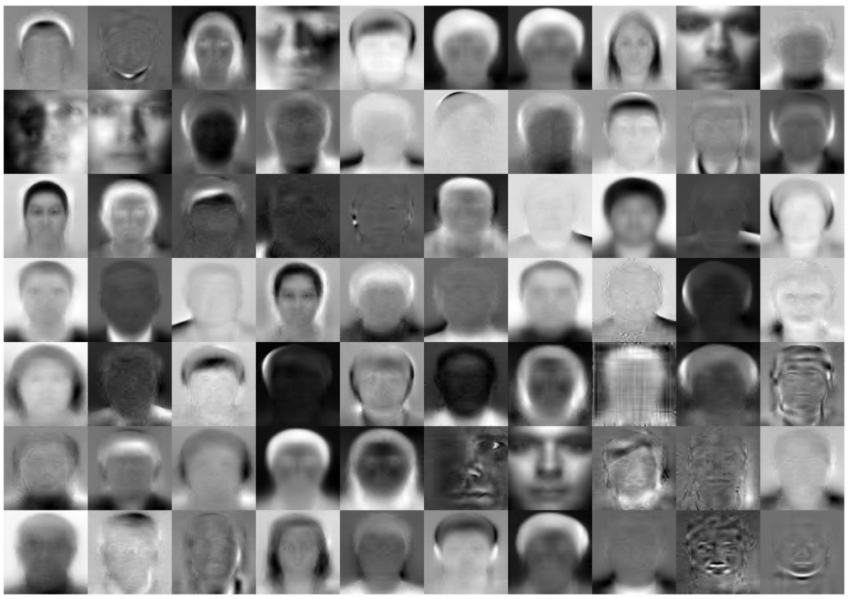


Figure 4.14: A subset of the obtained atoms by OSDL.

framework [SOZE16], where the authors proposed an Online Sparse Dictionary Learning (OSDL) algorithm that is able to obtain large adaptable atoms from natural images. Trainlets are built as linear (sparse) combinations of atoms from a fast and analytical dictionary, that of the novel Cropped Wavelets. This work [SOZE16] presented some initial results on sparse approximation of face images - indicating their effectiveness in modeling high dimensional data.

In this work we will formulate the inpainting task as an inverse problem regularized by a sparse prior under a global dictionary trained from publicly available datasets. Our results indicate that the proposed approach is able to synthesize missing information which is in a accordance with the global context of the image, yielding natural reconstructed faces.

In order to cope with the increase of training data, the work in [SOZE16] proposed a dictionary learning algorithm based on ideas from stochastic optimization [Bot98]. In a nutshell, the algorithm performs sparse coding of a mini-batch of training examples with (Sparse) OMP [RZE08], and then updates a subset of the dictionary atoms through a variation of the Normalized Iterative Hard Thresholding algorithm [BD10]. For completion, we present a summary of this method in Algorithm 4.2, and we refer the reader to [SOZE16] for further details.

Tackling the learning of a global model for face images in particular, we apply OSDL on a compendium of face images taken from different datasets, using the freely available code at the author's website. To increase the variability of the training data – and to obtain a more general model – we employ images taken from the Chinese Passport dataset used in [BE08] (both in its aligned and non aligned formats), the Chicago Faces Database [MCW15], the AT&T Faces Database[11], and the Cropped Yale Database [GBK01]. All images were rescaled to a size of 100×100 pixels, and employed *as is*; i.e., there was no coherent scaling or alignment

[11]Freely available from AT&T Laboratories Cambridge's website.

involved. All together, these amounted to a training set of roughly 19,000 images. OSDL took approximately 2 days to perform 40 data-sweeps[12]. We employed the Cropped Wavelets as the base dictionary (with Daubechies Wavelets with 4 vanishing moments), which has a redundancy of ≈ 1.7. The matrix $\mathbf{A}$ was chosen to be tall (under-complete), having 6,000 atoms in it. The atom sparsity was set to 1000; i.e., these are *only* $\approx 6\%$ sparse. We present some of the obtained atoms in Figure 4.14, where one can see that not only they resemble faces or face-features, but also the obtained variability between different sizes and configurations.

4.6.1 Inpainting Formulation

Once the global model has been obtained, we move to describe in detail the inpainting formulation. Consider the original image $\mathbf{y}_0 \in \mathbb{R}^n$ ($n = 10,000$), and a mask $\mathbf{M}$, given by a binary matrix of size $l \times n$, where $l = c \cdot n$. This way, c denotes the fraction of the pixels that have not been removed (and remain) from the degraded image given by $\mathbf{y} = \mathbf{M}\mathbf{y}_0$.

Given this degradation model, and leveraging the obtained dictionary $\mathbf{D}$, the inpainting inverse problem can be cast in terms of a pursuit by adding a sparse regularization term. Formally,

$$\min_{\boldsymbol{\gamma}} \|\boldsymbol{\gamma}\|_0 \text{ subject to } \|\mathbf{y} - \mathbf{M}\mathbf{D}\boldsymbol{\gamma}\|_2 \leq \epsilon.$$

This is nothing but a sparse coding problem with the incorporation of a degradation mask. Unlike the sparse coding stage in Algorithm 4.2, we now turn to a relaxation of this formulation moving from the ℓ_0 to the ℓ_1 norm. This way, we replace the problem above with the unconstrained optimization problem given by

$$\min_{\boldsymbol{\gamma}} \|\mathbf{y} - \mathbf{M}\mathbf{D}\boldsymbol{\gamma}\|_2 + \lambda\|\boldsymbol{\gamma}\|_1, \tag{4.8}$$

where λ is a the penalty parameter, compromising between the desired sparsity and the (masked) fidelity term. The shift from the ℓ_0 to the ℓ_1 norm is motivated by a practical aspect: in the inpainting problem, where one does not known a priori the number of non-zero elements needed to obtain a *good* reconstruction (or the equivalent ϵ threshold), it is easier to tune a penalty parameter λ. The number of non-zeros in $\boldsymbol{\gamma}$ might be larger than those employed during the training, therefore making a greedy pursuit time consuming. In addition, we have found this ℓ_1 approach to yield solutions that are smoother, resulting in more naturally-looking inpainted areas.

Due to the convexity of the problem in Equation (4.8), a variety of algorithms can be employed to find its solution. Iterative shrinkage algorithms are particularly well-suited for this kind of problems, and we employ FISTA as the specific solver [BT09a][13]. Our implementation of this method benefits from the relatively low-complexity of applying $\mathbf{D}$. Indeed, multiplying a

[12]We run our experiment on a Windows computer with an Intel Xeon E5 CPU, with 64 Gb of RAM running Windows 64 bits. However, no parallel processing was used, and memory consumption did not exceed 16 Gb.

[13]While we employ FISTA for the minimization of Equation (4.8), the learning algorithm (OSDL) still employs OMP for the Sparse Coding stage.

Figure 4.15: Inpainting of the image on the left column, for increasing values of λ (from left to right) in the range [0.05,50], with Trainlets.

vector by the dictionary (or its transpose) is never done explicitly. Instead, this is computed in terms of the product with the (very) sparse matrix $\mathbf{A}$ and the 1-dimensional wavelet dictionaries, which represent the separable operator $\mathbf{\Phi}$.

4.6.2 Results

For our experiments, we applied the method described in the previous section on a set of testing images, not included in the training set. In order to demonstrate the benefits of the proposed approach based on Trainlets, we compare with a number of other methods; namely: 1) the patch-propagation method of [XS10], which employs a sparse (patch) prior to inpaint the image, 2) a PCA (global) learned basis, and 3) the Separable Dictionary Learning (SEDIL) algorithm [HSK13], which also trains a global but separable dictionary. For this last method, we trained two 1-dimensional dictionaries of size 100×200 on the same training set, employing the code provided by the authors[14]. Note that both PCA and SEDIL obtain a set of global adaptive atoms by enforcing some constraints: orthogonality and separability, respectively.

The inpainting algorithm resulting from the minimization of Equation (4.8) depends on the parameter λ, and its value influences the quality of the final reconstruction. An example is presented in Figure 4.15, where we inpaint the image on the left with the proposed approach for increasing values of this parameter.

In our experiments, and for a legitimate comparison, we run each method for a series of values of this parameter and then selected the most plausible results for each method separately Note that the selection of the best (most plausible) result is somewhat subjective, for which we

[14]Note that this is a batch method, and we employed 2,000 iterations. Training with SEDIL took approximately 2.5 days, resulting in both dictionary learning algorithms running for about the same time.

have used our most fair judgment.

The comparison with [XS10], on the other hand, is not entirely fair: inpaiting methods based on patch propagation are not expected to perform well in this challenging problem, as they cannot inpaint elements (mouth, eyes, etc) that do not appear in the available image region. Yet, we include them for completion and in order to demonstrate the intrinsic need of a global model.

We present a subset of our results in Figure 4.16, and more examples can be found in the supplementary material. As expected, the local method of [XS10] provides results that are not in agreement with the global context. On the other hand, the performance of SEDIL is limited, while PCA sometimes manages to recover somewhat of a natural result. Still, the constraints imposed by both of these two methods appear to be too restrictive for this problem. As can be seen, Trainlets provide the best results – often making it hard to distinguish between the original and the synthetic inpainted image. Some cases are particularly interesting: in the third image, where the glare in the glasses occlude the left eye, our approach manages to restore it; in the fourth image, we inpaint an eye which was not originally there due to lighting conditions, still in a plausible manner. More interesting examples can be found in the supplementary material.

4.7 Chapter Conclusions

In this Chapter we have propose a different solution to the limitations of patch-based approaches: increase the patch size to become a global image. This approach requires a careful analysis of the dictionary model, as well as the training algorithm employed to adapt this model effectively. After these aspects have been considered, we have shown that benefit can be gained by increasing the small dimensional patches in traditional restoration applications, like image denoising. Interestingly, when searching for *universal* dictionaries, we found that our model proposes all kinds of atoms that resemble (to some extent) typical analytic constructions popular in approximation theory.

Our approach proves most successful when deployed for a particular class of images, as in the case of face images. In this scenario, one can tackle very challenging restoration problems – only attainable to very recent Deep Learning methods [PKD$^+$16]. An interesting observation is that once a good global model is at our disposal, there is no need for any extra algorithmic manipulation of the data: there is no symmetry, exemplar-based copying or other form of external regularization enforced in the reconstruction; this is naturally captured by the learning process. Exploring the ability of a similar approach in tackling other inverse problems is an interesting direction of research and part of ongoing work.

Finally, while the our method is very effective in modeling images from a similar class, employing this approach for the inpainting of large areas in natural images is unlikely to succeed, as learning a global model for such general cases is a significantly more challenging task. In this case, improvements on the learning algorithm (and the model) would be needed before attempting to solve this kind of inverse problem.

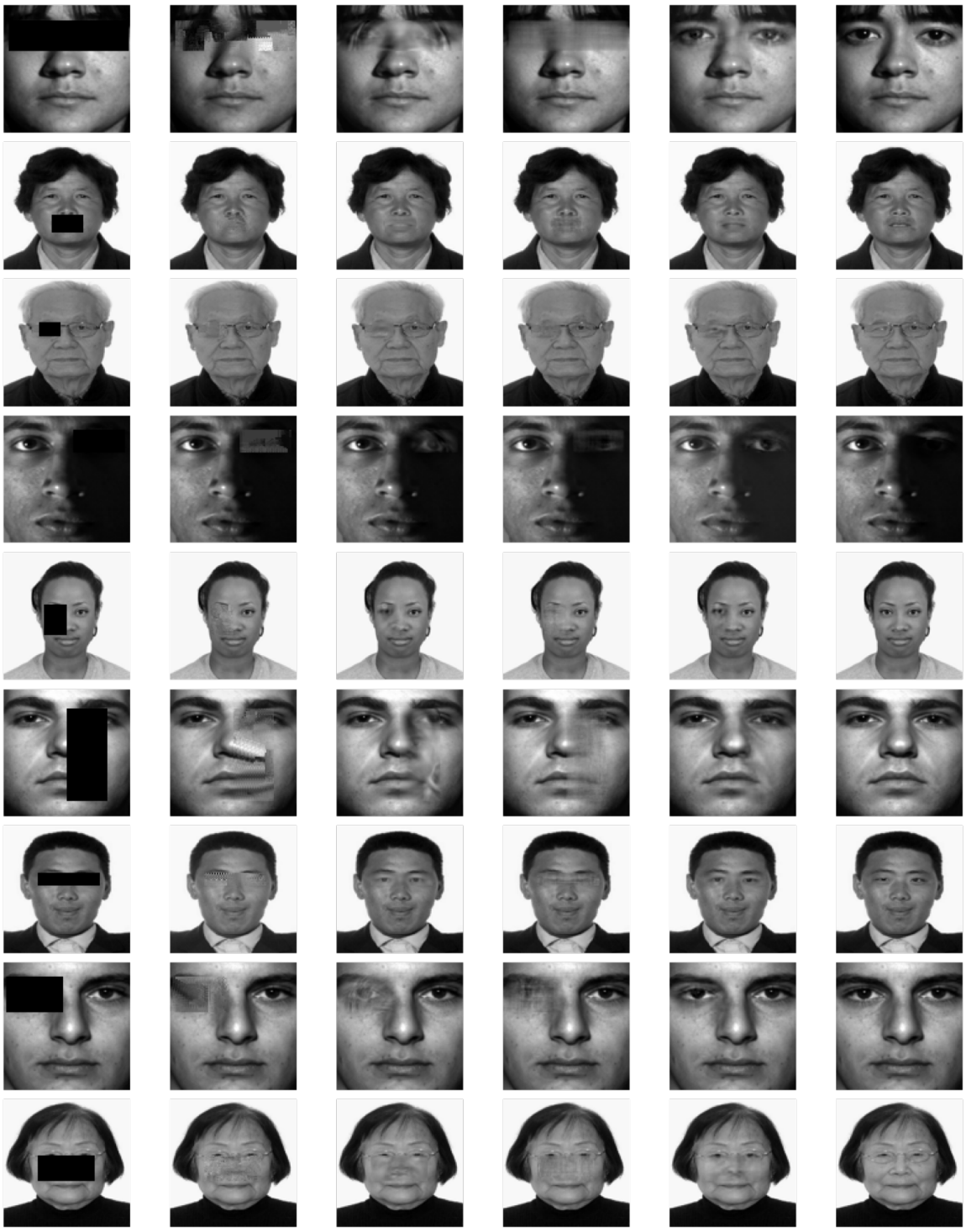


Figure 4.16: Inpainting results. From left to right: masked image, patch propagation [XS10], PCA, SEDIL [HSK13], Trainlets [SOZE16], and the original image.

4.8 Chapter Appendix

4.8.1 Further Inpainting Results

Figure 4.17: Inpainting results. From left to right: masked image, patch propagation [XS10], PCA, SEDIL [HSK13], Trainlets [SOZE16], and the original image.

Figure 4.18: Inpainting results. From left to right: masked image, patch propagation [XS10], PCA, SEDIL [HSK13], Trainlets [SOZE16], and the original image.

Chapter 5

Convolutional Sparse Coding

Chapter Abstract

The celebrated sparse representation model has led to remarkable results in various signal processing tasks in the last decade. However, despite its initial purpose of serving as a global prior for entire signals, it has been commonly used for modeling low dimensional patches due to the computational constraints it entails when deployed with learned dictionaries. A way around this problem has been recently proposed, adopting a convolutional sparse representation model. This approach assumes that the global dictionary is a concatenation of banded Circulant matrices. While several works have presented algorithmic solutions to the global pursuit problem under this new model, very few truly-effective guarantees are known for the success of such methods. In this chapter, we address the theoretical aspects of the convolutional sparse model providing the first meaningful answers to questions of uniqueness of solutions and success of pursuit algorithms, both greedy and convex relaxations, in ideal and noisy regimes. To this end, we generalize mathematical quantities, such as the ℓ_0 norm, mutual coherence, Spark and RIP to their counterparts in the convolutional setting, intrinsically capturing local measures of the global model. On the algorithmic side, we demonstrate how to solve the global pursuit problem by using simple local processing, thus offering a first of its kind bridge between global modeling of signals and their patch-based local treatment.

5.1 An Underlying Local-Global Model?

The previous chapter showed that if one is to propose a global sparse model for images, one must constrain the dictionary in order to ovoid the curse of dimensionality. We leveraged this observation in the context of the double sparsity model, showing that the dictionary learning problem can indeed be up-scaled. However, the Trainlets approach clearly has limitations that will prevent us from learning a model for *arbitrarily large* images.

An alternative is a constrained global model in which the signal is composed as a superposition of local atoms. The family of dictionaries giving rise to such signals is a concatenation of banded Circulant matrices, as depicted in Figure 5.1. This global model benefits from having a local shift invariant structure – a popular assumption in signal and image processing – suggesting an interesting connection to the above-mentioned local modeling.

When the dictionary $\mathbf{D}$ has this structure of a concatenation of banded Circulant matrices, the resulting pursuit problem is usually known as convolutional sparse coding [GRKN07]. Recently, several works have addressed the problem of using and training such a model in the context of image inpainting, super-resolution, and general image representation [BEL13, HHW15, KF14, Woh14, GZX+15]. These methods usually exploit an ADMM formulation [BPC+11] while operating in the Fourier domain in order to search for the sparse codes and train the dictionary involved. Several variations have been proposed for solving the pursuit problem, yet there has been no theoretical analysis of their success.

Assume a signal is created by multiplying a sparse vector by a convolutional dictionary. We will let the following set of questions guide our work and the results presented in this chapter:

1. Can we guarantee the uniqueness of such a global (convolutional) sparse vector?

2. Can global pursuit algorithms, such as the ones suggested in recent works, be guaranteed to find the true underlying sparse code, and if so, under which conditions?

3. Can we guarantee a stability of the sparse approximation problem, and a stability of corresponding pursuit methods in a noisy regime?

4. Can we solve the global pursuit by restricting the process to local pursuit operations?

A naïve approach to address such theoretical questions is to apply the fairly extensive results for sparse representation and compressed sensing to the above defined model [Ela10]. However, as we will show throughout this chapter, this strategy provides nearly useless results and bounds from a global perspective. Therefore, there exists a true need for a deeper and alternative analysis of the sparse coding problem in the convolutional case which would yield meaningful bounds.

In this chapter, we will demonstrate the futility of the ℓ_0-norm in providing meaningful bounds in the convolutional model. This, in turn, motivates us to propose a new localized measure – the $\ell_{0,\infty}$ norm. Based on it, we redefine our pursuit into a problem that operates locally while thinking globally. To analyze this problem, we extend useful concepts, such as

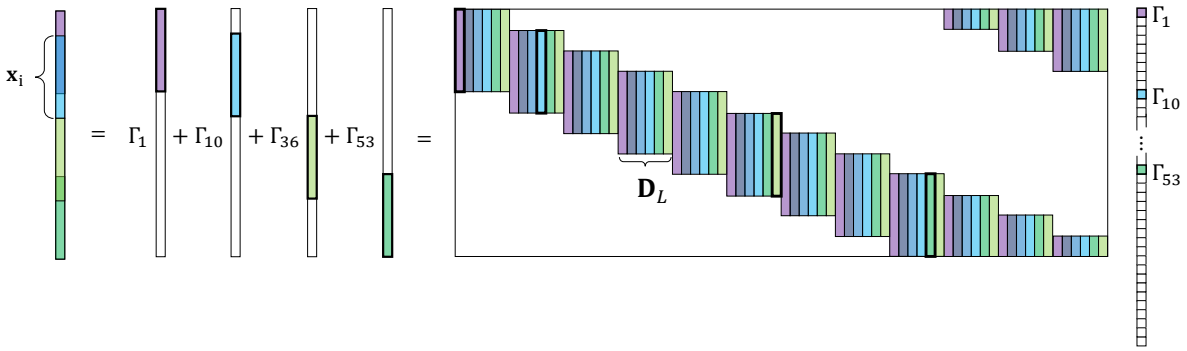


Figure 5.1: The convolutional model description, and its composition in terms of the local dictionary $\mathbf{D}_L$.

the Spark and mutual coherence, to the convolutional setting. We then provide claims for uniqueness of solutions and for the success of pursuit methods in the noiseless case, both for greedy algorithms and convex relaxations. Based on these theoretical foundations, we then extend our analysis to a more practical scenario, handling noisy data and model deviations. We generalize and tie past theoretical constructions, such as the Restricted Isometry Property (RIP) [CT05] and the Exact Recovery Condition (ERC) [Tro04], to the convolutional framework proving the stability of this model in this case as well.

Before diving in, a word about notation. The general approach of denoting vectors and matrices with bold characters will be maintained. However, and because the results presented in this chapter duel on the analysis of local and global vectors, we will employ the somewhat unorthodox choice of capital letters for global vectors and lowercase for local ones.

5.2 Preliminaries on CSC

Consider now the global dictionary to be a concatenation of m banded Circulant matrices[1], where each such matrix has a band of width $n \ll N$. As such, by simple permutation of its columns, such a dictionary consists of all shifted versions of a *local* dictionary $\mathbf{D}_L$ of size $n \times m$. This model is commonly known as Convolutional Sparse Representation [GRKN07,BL14,HHW15]. Hereafter, whenever we refer to the global dictionary $\mathbf{D}$, we assume it has this structure. Assume a signal $\mathbf{X}$ to be generated as $\mathbf{D}\boldsymbol{\Gamma}$. In Figure 5.1 we describe such a global signal, its corresponding dictionary that is of size $N \times mN$ and its sparse representation, of length mN. We note that $\boldsymbol{\Gamma}$ is built of N distinct and independent sparse parts, each of length m, which we will refer to as the local sparse vectors $\boldsymbol{\alpha}_i$.

Consider a sub-system of equations extracted from $\mathbf{X} = \mathbf{D}\boldsymbol{\Gamma}$ by multiplying this system by the patch extraction[2] operator $\mathbf{R}_i \in \mathbb{R}^{n \times N}$. The resulting system is $\mathbf{x}_i = \mathbf{R}_i\mathbf{X} = \mathbf{R}_i\mathbf{D}\boldsymbol{\Gamma}$,

[1] Each of these matrices is constructed by shifting a single column, supported on n subsequent entries, to all possible shifts. This choice of Circulant matrices comes to alleviate boundary problems.

[2] Denoting by $\mathbf{0}_{(a\times b)}$ a zeros matrix of size $a \times b$, and $\mathbf{I}_{(n\times n)}$ an identity matrix of size $n \times n$, then $\mathbf{R}_i = \left[\mathbf{0}_{(n\times(i-1))}, \mathbf{I}_{(n\times n)}, \mathbf{0}_{(n\times(N-i-n+1))}\right]$.

where $\mathbf{x}_i$ is a patch of length n extracted from $\mathbf{X}$ from location i. Observe that in the set of extracted rows, $\mathbf{R}_i\mathbf{D}$, there are only $(2n-1)m$ columns that are non-trivially zero. Define the operator $\mathbf{S}_i \in \mathbb{R}^{(2n-1)m \times mN}$ as a columns' selection operator[3], such that $\mathbf{R}_i\mathbf{D}\mathbf{S}_i^T$ preserves all the non-zero columns in $\mathbf{R}_i\mathbf{D}$. Thus, the subset of equations we get is essentially

$$\mathbf{x}_i = \mathbf{R}_i\mathbf{X} = \mathbf{R}_i\mathbf{D}\boldsymbol{\Gamma} = \mathbf{R}_i\mathbf{D}\mathbf{S}_i^T\mathbf{S}_i\boldsymbol{\Gamma}. \tag{5.1}$$

Definition 1. Given a global sparse vector $\boldsymbol{\Gamma}$, define $\boldsymbol{\gamma}_i = \mathbf{S}_i\boldsymbol{\Gamma}$ as its i^{th} stripe representation.

Note that a stripe $\boldsymbol{\gamma}_i$ can be also seen as a group of $2n-1$ adjacent local sparse vectors $\boldsymbol{\alpha}_j$ of length m from $\boldsymbol{\Gamma}$, centered at location $\boldsymbol{\alpha}_i$.

Definition 2. Consider a convolutional dictionary $\mathbf{D}$ defined by a local dictionary $\mathbf{D}_L$ of size $n \times m$. Define the stripe dictionary $\boldsymbol{\Omega}$ of size $n \times (2n-1)m$, as the one obtained by extracting n consecutive rows from $\mathbf{D}$, followed by the removal of its zero columns, namely $\boldsymbol{\Omega} = \mathbf{R}_i\mathbf{D}\mathbf{S}_i^T$.

Observe that $\boldsymbol{\Omega}$, depicted in Figure 5.2, is independent of i, being the same for all locations due to the union-of-Circulant-matrices structure of $\mathbf{D}$. In other words, the shift invariant property is satisfied for this model – all patches share the same stripe dictionary in their construction. Armed with the above two definitions, Equation (5.1) simply reads $\mathbf{x}_i = \boldsymbol{\Omega}\boldsymbol{\gamma}_i$.

From a different perspective, one can synthesize the signal $\mathbf{X}$ by considering $\mathbf{D}$ as a concatenation of N vertical stripes of size $N \times m$ (see Figure 5.1), where each can be represented as $\mathbf{R}_i^T\mathbf{D}_L$. In other words, the vertical stripe is constructed by taking the small and local dictionary $\mathbf{D}_L$ and positioning it in the i^{th} row. The same partitioning applies to $\boldsymbol{\Gamma}$, leading to the $\boldsymbol{\alpha}_i$ ingredients. Thus,

$$\mathbf{X} = \sum_i \mathbf{R}_i^T\mathbf{D}_L\boldsymbol{\alpha}_i.$$

Since $\boldsymbol{\alpha}_i$ play the role of local sparse vectors, $\mathbf{D}_L\boldsymbol{\alpha}_i$ are reconstructed patches (which are not the same as $\mathbf{x}_i = \boldsymbol{\Omega}\boldsymbol{\gamma}_i$), and the sum above proposes a patch averaging approach as practiced in several works [AEB06, ZW11, SE15]. This formulation provides another local interpretation of the convolutional model.

Yet a third interpretation of the very same signal construction can be suggested, in which the signal is seen as resulting from a sum of local/small atoms which appear in a small number of locations throughout the signal. This can be formally expressed as

$$\mathbf{X} = \sum_{i=1}^{m} \mathbf{d}_i * \mathbf{z}_i,$$

where the vectors $\mathbf{z}_i \in \mathbb{R}^N$ are sparse maps encoding the location and coefficients convolved with the i^{th} atom [GRKN07]. In our context, $\boldsymbol{\Gamma}$ is simply the interlaced concatenation of all $\mathbf{z}_i$.

[3] An analogous definition can be written for this operator as well.

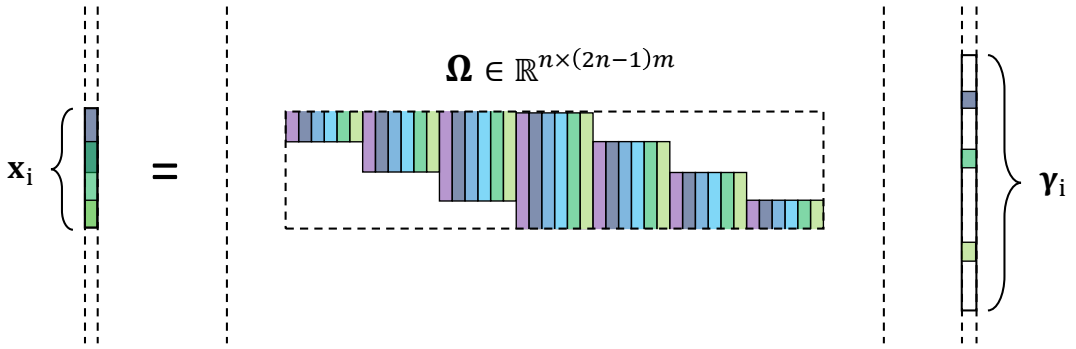


Figure 5.2: Stripe Dictionary

This model (adopting the last convolutional interpretation) has received growing attention in recent years in various applications. In [MSH08] a convolutional sparse coding framework was used for pattern detection in images and the analysis of instruments in music signals, while in [ZL15] it was used for the reconstruction of 3D trajectories. The problem of learning the local dictionary $\mathbf{D}_L$ was also studied in several works [ZKTF10, KSB+10, BL14, Woh14, HA15]. Different methods have been proposed for solving the convolutional sparse coding problem under an ℓ_1-norm penalty. Commonly, these methods rely on the ADMM algorithm [BPC+11], exploiting multiplications of vectors by the global dictionary in the Fourier domain in order to reduce the computational cost involved [BL14]. An alternative is the deployment of greedy algorithms of the Matching Pursuit family [MZ93], which suggest an ℓ_0 constraint on the global sparse vector. The reader is referred to the work of [Woh14] and references therein for a thorough discussion on these methods. In essence, all the above works are solutions to the minimization of a global pursuit under the convolutional structure. As a result, the theoretical results in our work will apply to these methods, providing guarantees for the recovery of the underlying sparse vectors.

5.3 From Global to Local Analysis

Consider a sparse vector $\mathbf{\Gamma}$ of size mN which represents a global (convolutional) signal. Assume further that this vector has a few $k \ll N$ non-zeros. If these were to be clustered together in a given stripe $\boldsymbol{\gamma}_i$, the local patch corresponding to this stripe would be very complex, and pursuit methods would likely fail in recovering it. On the contrary, if these k non-zeros are spread all throughout the vector $\mathbf{\Gamma}$, this would clearly imply much simpler local patches, facilitating their successful recovery. This simple example comes to show the futility of the traditional global ℓ_0-norm in assessing the success of convolutional pursuits, and it will be the pillar of our intuition throughout our work.

5.3.1 The $\ell_{0,\infty}$ Norm and the $P_{0,\infty}$ Problem

Let us now introduce a measure that will provide a local notion of sparsity within a global sparse vector.

Definition 3. Define the $\ell_{0,\infty}$ pseudo-norm of a global sparse vector $\mathbf{\Gamma}$ as

$$\|\mathbf{\Gamma}\|_{0,\infty} = \max_i \|\boldsymbol{\gamma}_i\|_0.$$

In words, this quantifies the number of non-zeros in the densest stripe $\boldsymbol{\gamma}_i$ of the global $\mathbf{\Gamma}$. This is equivalent to extracting all stripes from the global sparse vector $\mathbf{\Gamma}$, arranging them column-wise into a matrix $\mathbf{A}$ and applying the usual $\|\mathbf{A}\|_{0,\infty}$ norm – thus, the name. By constraining the $\ell_{0,\infty}$ norm to be low, we are essentially limiting all stripes $\boldsymbol{\gamma}_i$ to be sparse, and their corresponding patches $\mathbf{R}_i\mathbf{X}$ to have a sparse representation under a shift-invariant local dictionary $\mathbf{\Omega}$. This is one of the underlying assumptions in many signal and image processing algorithms. As for properties of this norm, similar to ℓ_0 case, in the $\ell_{0,\infty}$ the non-negativity and triangle inequality properties hold, while homogeneity does not.

Armed with the above definition, we now move to define the $P_{0,\infty}$ problem:

$$(P_{0,\infty}): \quad \min_{\mathbf{\Gamma}} \quad \|\mathbf{\Gamma}\|_{0,\infty} \ \text{ s.t. } \ \mathbf{D\Gamma} = \mathbf{X}.$$

When dealing with a global signal, instead of solving the P_0 problem (minimizing the ℓ_0 norm of Γ) as is commonly done, we aim to solve the above defined objective instead. The key difference is that we are not limiting the overall number of zeros in $\mathbf{\Gamma}$, but rather putting a restriction on its local density.

5.3.2 Global versus Local Bounds

As mentioned previously, theoretical bounds are often given in terms of the mutual coherence of the dictionary. In this respect, a lower bound on this value is much desired. In the case of the convolution sparse model, this value quantifies not only the correlation between the atoms in $\mathbf{D}_L$, but also the correlation between their shifts. Though in a different context, a bound for this value was derived in [Wel74], and it is given by

$$\mu(\mathbf{D}) \geq \sqrt{\frac{m-1}{m(2n-1)-1}}. \tag{5.2}$$

For a large value of m, one obtains that the best possible coherence is $\mu(\mathbf{D}) \approx \frac{1}{\sqrt{2n}}$. This implies that if we are to apply BP or OMP to recover the sparsest $\mathbf{\Gamma}$ that represents $\mathbf{X}$, the classical sparse approximation results [BDE09] would allow merely $O(\sqrt{n})$ non-zeros in **all** $\mathbf{\Gamma}$, for any N, no matter how long $\mathbf{X}$ is! As we shall see next, the situation is not as grave as it may seem, due to our migration from P_0 to $P_{0,\infty}$. Leveraging the previous definitions, we will provide global recovery guarantees that will have a local flavor, and the bounds will be given in terms of the

N	:	length of the global signal.
n	:	size of a local atom or a local signal patch.
m	:	number of unique local atoms (filters) or the number of Circulant matrices.
$\mathbf{X}$, $\mathbf{Y}$ and: $\mathbf{E}$		global signals of length N, where generally $\mathbf{Y} = \mathbf{X} + \mathbf{E}$.
$\mathbf{D}$	:	global dictionary of size $N \times mN$.
$\mathbf{\Gamma}$ and $\mathbf{\Delta}$	:	global sparse vectors of length mN.
Γ_i and Δ_i	:	the i^{th} entry in $\mathbf{\Gamma}$ and $\mathbf{\Delta}$, respectively.
$\mathbf{D}_L$	:	local dictionary of size $n \times m$.
$\mathbf{\Omega}$	:	stripe dictionary of size $n \times (2n-1)m$, which contains all possible shifts of $\mathbf{D}_L$.
$\boldsymbol{\alpha}_i$	:	local sparse code of size m.
$\boldsymbol{\gamma}_i$ and $\boldsymbol{\delta}_i$	:	a stripe of length $(2n-1)m$ extracted from the global vectors $\mathbf{\Gamma}$ and $\mathbf{\Delta}$, respectively.
$\boldsymbol{\gamma}_{i,s}$ and: $\boldsymbol{\delta}_{i,s}$		a local sparse vector of length m which corresponds to the s^{th} portion inside $\boldsymbol{\gamma}_i$ and $\boldsymbol{\delta}_i$, respectively.

Table 5.1: Summary of notations used throughout this chapter.

number of non-zeros in the densest stripe. This way, we will show that the guarantee conditions can be significantly enhanced to $O(\sqrt{n})$ non-zeros *locally* rather than *globally*.

5.4 Theoretical Study of Ideal Signals

As motivated in the previous section, the concerns of uniqueness, recovery guarantees and stability of sparse solutions in the convolutional case require special attention. We now formally address these questions by following the path taken in [Ela10], carefully generalizing each and every statement to the global-local model discussed here.

Before proceeding onto theoretical grounds, we briefly summarize, for the convenience of the reader, all notations used throughout this work in Table 5.1.

5.4.1 Uniqueness and Stripe-Spark

Just as it was initially done in the general sparse model, one might ponder about the uniqueness of the sparsest representation in terms of the $\ell_{0,\infty}$ norm. More precisely, does a unique solution to the $P_{0,\infty}$ problem exist? and under which circumstances? In order to answer these questions we shall first extend our mathematical tools, in particular the characterization of the dictionary, to the convolutional scenario.

In Chapter 2 we recalled the definition of the Spark of a general dictionary $\mathbf{D}$. In the same spirit, we propose the following:

Definition 4. Define the Stripe-Spark of a convolutional dictionary $\mathbf{D}$ as

$$\sigma_\infty(\mathbf{D}) = \min_{\mathbf{\Delta}} \quad \|\mathbf{\Delta}\|_{0,\infty} \;\; \text{s.t.} \;\; \mathbf{\Delta} \neq 0, \; \mathbf{D}\mathbf{\Delta} = 0.$$

In words, the Stripe-Spark is defined by the sparsest non-zero vector, in terms of the $\ell_{0,\infty}$ norm, in the null space of $\mathbf{D}$. Next, we use this definition in order to formulate an uncertainty and a uniqueness principle for the $P_{0,\infty}$ problem that emerges from it. The proof of this and the following theorems are described in detail in the Supplementary Material.

Theorem 5. (Uncertainty and uniqueness using Stripe-Spark): Let $\mathbf{D}$ be a convolutional dictionary. If a solution $\mathbf{\Gamma}$ obeys $\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\sigma_\infty$, then this is necessarily the global optimum for the $P_{0,\infty}$ problem for the signal $\mathbf{D\Gamma}$.

5.4.2 Lower Bounding the Stripe-Spark

In general, and similar to the Spark, calculating the Stripe-Spark is computationally intractable. Nevertheless, one can bound its value using the global mutual coherence defined in Section 2. Before presenting such bound, we formulate and prove a Lemma that will aid our analysis throughout this chapter.

Lemma 5.4.1. *Consider a convolutional dictionary $\mathbf{D}$, with mutual coherence $\mu(\mathbf{D})$, and a support $\mathcal{T}$ with $\ell_{0,\infty}$ norm[4] equal to k. Let $\mathbf{G}^{\mathcal{T}} = \mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}$, where $\mathbf{D}_{\mathcal{T}}$ is the matrix $\mathbf{D}$ restricted to the columns indicated by the support $\mathcal{T}$. Then, the eigenvalues of this Gram matrix, given by $\lambda_i\left(\mathbf{G}^{\mathcal{T}}\right)$, are bounded by*

$$1-(k-1)\mu(\mathbf{D}) \;\le\; \lambda_i\left(\mathbf{G}^{\mathcal{T}}\right) \;\le\; 1+(k-1)\mu(\mathbf{D}).$$

Proof. From Gerschgorin's theorem, the eigenvalues of the Gram matrix $\mathbf{G}^{\mathcal{T}}$ reside in the union of its Gerschgorin circles. The j^{th} circle, corresponding to the j^{th} row of $\mathbf{G}^{\mathcal{T}}$, is centered at the point $\mathbf{G}^{\mathcal{T}}_{j,j}$ (belonging to the Gram's diagonal) and its radius equals the sum of the absolute values of the off-diagonal entries; i.e., $\sum_{i,i\neq j}|\mathbf{G}^{\mathcal{T}}_{j,i}|$. Notice that both indices i,j correspond to atoms in the support $\mathcal{T}$. Because the atoms are normalized, $\forall\ j,\ \mathbf{G}^{\mathcal{T}}_{j,j}=1$, implying that all Gershgorin disks are centered at 1. Therefore, all eigenvalues reside inside the circle with the largest radius. Formally,

$$\left|\lambda_i\left(\mathbf{G}^{\mathcal{T}}\right)-1\right| \le \max_j \sum_{i,i\neq j}\left|\mathbf{G}^{\mathcal{T}}_{j,i}\right| = \max_j \sum_{\substack{i,i\neq j\\ i,j\in\mathcal{T}}}|\mathbf{d}_j^T\mathbf{d}_i|. \tag{5.3}$$

On the one hand, from the definition of the mutual coherence, the inner product between atoms that are close enough to overlap is bounded by $\mu(\mathbf{D})$. On the other hand, the product $\mathbf{d}_j^T\mathbf{d}_i$ is zero for atoms $\mathbf{d}_i$ too far from $\mathbf{d}_j$ (i.e., out of the stripe centered at the j^{th} atom). Therefore, we obtain:

$$\sum_{\substack{i,i\neq j\\ i,j\in\mathcal{T}}}|\mathbf{d}_j^T\mathbf{d}_i| \le (k-1)\ \mu(\mathbf{D}),$$

[4]Note that specifying the $\ell_{0,\infty}$ of a support rather than a sparse vector is a slight abuse of notation, that we will nevertheless use for the sake of simplicity.

where k is the maximal number of non-zero elements in a stripe, defined previously as the $\ell_{0,\infty}$ norm of $\mathcal{T}$. Note that we have subtracted 1 from k because we must omit the entry on the diagonal. Putting this back in Equation (5.3), we obtain

$$\left|\lambda_i\left(\mathbf{G}^{\mathcal{T}}\right)-1\right| \leq \max_j \sum_{\substack{i,i\neq j \\ i,j\in\mathcal{T}}} \left|\mathbf{d}_j^T\mathbf{d}_i\right| \leq \ (k-1)\ \mu(\mathbf{D}).$$

From this we obtain the desired claim. □

We now dive into the next theorem, whose proof relies on the above Lemma.

Theorem 6. (Lower bounding the Stripe-Spark via the mutual coherence): For a convolutional dictionary $\mathbf{D}$ with mutual coherence $\mu(\mathbf{D})$, the Stripe-Spark can be lower-bounded by

$$\sigma_\infty(\mathbf{D}) \geq 1+\frac{1}{\mu(\mathbf{D})}.$$

Using the above derived bound and the uniqueness based on the Stripe-Spark we can now formulate the following theorem:

Theorem 7. (Uniqueness using mutual coherence): Let $\mathbf{D}$ be a convolutional dictionary with mutual coherence $\mu(\mathbf{D})$. If a solution $\mathbf{\Gamma}$ obeys $\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}(1+\frac{1}{\mu(\mathbf{D})})$, then this is necessarily the sparsest (in terms of $\ell_{0,\infty}$ norm) solution to $P_{0,\infty}$ with the signal $\mathbf{D\Gamma}$.

The proof of this claim is rather trivial, noting that if $\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}(1+\frac{1}{\mu(\mathbf{D})})$, then necessarilly $\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\sigma_\infty$, and so from Theorem 5 $\mathbf{\Gamma}$ is unique.

At the end of Section 5.3 we mentioned that for $m \gg 1$, the classical analysis would allow an order of $O(\sqrt{n})$ non-zeros all over the vector $\mathbf{\Gamma}$, regardless of the length of the signal N. In light of the above theorem, in the convolutional case, the very same quantity of non-zeros is allowed locally per stripe, implying that the overall number of non-zeros in $\mathbf{\Gamma}$ grows linearly with the global dimension N.

5.4.3 Recovery Guarantees for Pursuit Methods

In this subsection, we attempt to solve the $P_{0,\infty}$ problem by employing two common, but very different, pursuit methods: the Orthogonal Matching Pursuit (OMP) and the Basis Pursuit (BP) – the reader is referred to [Ela10] for a detailed description of these formulations and respective algorithms. Leaving aside the computational burdens of running such algorithms, which will be addressed in the second part of this work, we now consider the theoretical aspects of their success.

Previous works [DE03, Tro04] have shown that both OMP and BP succeed in finding the sparsest solution to the P_0 problem if the cardinality of the representation is known a priori to be lower than $\frac{1}{2}(1+\frac{1}{\mu(\mathbf{D})})$. That is, we are guaranteed to recover the underlying solution as long as the *global sparsity* is less than a certain threshold. In light of the discussion in Section 5.3.2, these values are pessimistic in the convolutional setting. By migrating from P_0 to the $P_{0,\infty}$

problem, we show next that both algorithms are in fact capable of recovering the underlying solutions under far weaker assumptions.

Theorem 8. (Global OMP recovery guarantee using $\ell_{0,\infty}$ norm): Given the system of linear equations $\mathbf{X} = \mathbf{D}\boldsymbol{\Gamma}$, if a solution $\boldsymbol{\Gamma}$ exists satisfying

$$\|\boldsymbol{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right), \tag{5.4}$$

then OMP is guaranteed to recover it.

Note that if we assume $\|\boldsymbol{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$, according to our uniqueness theorem, the solution obtained by the OMP is the unique solution to the $P_{0,\infty}$ problem. Interestingly, under the same conditions the BP algorithm is guaranteed to succeed as well.

Theorem 9. (Global Basis Pursuit recovery guarantee using the $\ell_{0,\infty}$ norm): For the system of linear equations $\mathbf{D}\boldsymbol{\Gamma} = \mathbf{X}$, if a solution $\boldsymbol{\Gamma}$ exists obeying

$$\|\boldsymbol{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right),$$

then Basis Pursuit is guaranteed to recover it.

The recovery guarantees for both pursuit methods have now become *independent of the global signal dimension and sparsity*. Instead, the condition for success is given in terms of the *local* concentration of non-zeros of the global sparse vector. Moreover, the number of non-zeros allowed per stripe under the current bounds is in fact the same number previously allowed globally. As a remark, note that we have used these two algorithms in their natural form, being oblivious to the $\ell_{0,\infty}$ objective they are serving. Further work is required to develop OMP and BP versions that are aware of this specific goal, potentially benefiting from it.

5.4.4 Experiments

In this subsection we intend to provide numerical results that corroborate the above presented theoretical bounds. While doing so, we will shed light on the performance of the OMP and BP algorithms in practice, as compared to our previous analysis.

In [SNS14] an algorithm was proposed to construct a local dictionary such that all its aperiodic auto-correlations and cross-correlations are low. This, in our context, means that the algorithm attempts to minimize the mutual coherence of the dictionary $\mathbf{D}_L$ and all of its shifts, decreasing the global mutual coherence as a result. We use this algorithm to numerically build a dictionary consisting of two atoms ($m = 2$) with patch size $n = 64$. The theoretical lower bound on the $\mu(\mathbf{D})$ presented in Equation (5.2) under this setting is approximately 0.063, and we manage to obtain a mutual coherence of 0.09 using the aforementioned method. With these atoms we construct a convolutional dictionary with global atoms of length $N = 640$.

Once the dictionary is fixed, we generate sparse vectors with random supports of (global) cardinalities in the range $[1, 300]$. The non-zero entries are drawn from random independent

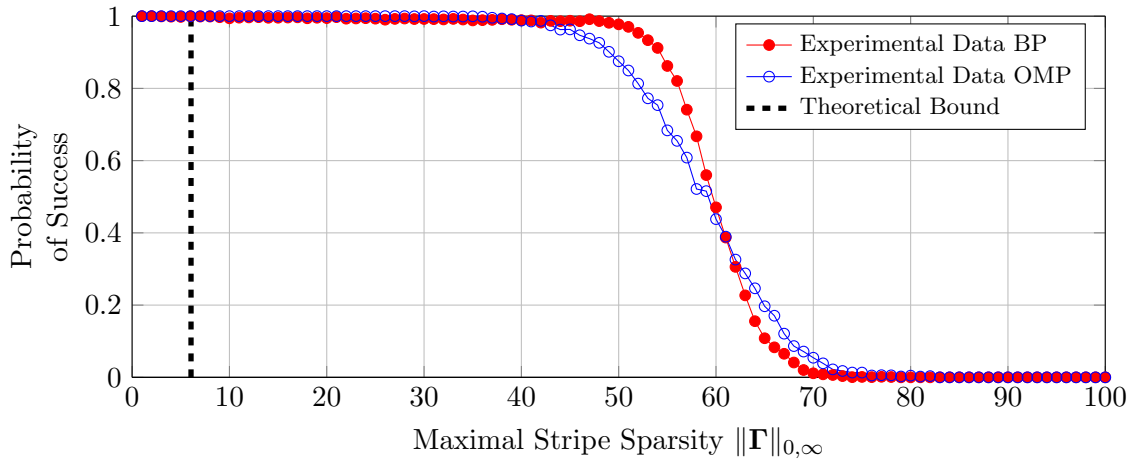


Figure 5.3: Probability of success of OMP and BP at recovering the true convolutional sparse code. The theoretical guarantee is presented on the same graph.

and identically-distributed Gaussians with mean equal to zero and variance equal to one. Given these sparse vectors, we compute their corresponding global signals and attempt to recover them using the global OMP and BP. We perform 500 experiments per each cardinality and present the probability of success as a function of the representation's $\ell_{0,\infty}$ norm. We define the success of the algorithm as the full recovery of the true sparse vector. The results for the experiment are presented in Figure 5.3. The theorems provided in the previous subsection guarantee the success of both OMP and BP as long as the $\|\mathbf{\Gamma}\|_{0,\infty} \leq 6$.

As can be seen from these results, the theoretical bound is far from being tight. However, in the traditional sparse representation model the corresponding bounds have the same loose flavor [BDE09]. This kind of results is in fact expected when using such a worst-case analysis. Tighter bounds could likely be obtained by a probabilistic study, which we leave for future work.

5.5 Shifted Mutual Coherence and Stripe Coherence

When considering the mutual coherence $\mu(\mathbf{D})$, one needs to look at the maximal correlation between every pair of atoms in the global dictionary. One should note, however, that atoms having a non-zero correlation must have overlapping supports, and $\mu(\mathbf{D})$ provides a bound for these values independently of the amount of overlap. One could go beyond this characterization of the convolutional dictionary by a single value and propose to bound all the inner products between atoms for a *given shift*. As a motivation, in several applications one can assume that signals are built from local atoms separated by some minimal lag, or shift. In radio communications, for example, such a situation appears when there exists a minimal time between consecutive transmissions on the same channel [HSL09]. In such cases, knowing how the correlation between the atoms depends on their shifts is fundamental for the design of the dictionary, its utilization and its theoretical analysis.

In this section we briefly explore this direction of analysis, introducing a stronger characterization of the convolutional dictionary, termed shifted mutual coherence. By being a considerably more informative measure than the standard mutual coherence, this will naturally lead to stronger bounds. We will only present the main points of these results here for the sake of brevity; the interested reader can find a more detailed discussion on this matter in the Supplementary Material.

Recall that $\mathbf{\Omega}$ is defined as a stripe extracted from the global dictionary $\mathbf{D}$. Consider the sub-system given by $\mathbf{x}_i = \mathbf{\Omega}\boldsymbol{\gamma}_i$, corresponding to the i^{th} patch in $\mathbf{X}$. Note that $\mathbf{\Omega}$ can be split into a set of $2n-1$ blocks of size $n \times m$, where each block is denoted by $\mathbf{\Omega}_s$, i.e.,

$$\mathbf{\Omega} = [\mathbf{\Omega}_{-n+1}, \ldots, \mathbf{\Omega}_{-1}, \mathbf{\Omega}_0, \mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_{n-1}],$$

as shown previously in Figure 5.2.

Definition 10. Define the shifted mutual coherence μ_s by

$$\mu_s = \max_{i,j} \quad |\langle \mathbf{d}_i^0, \mathbf{d}_j^s \rangle|,$$

where $\mathbf{d}_i^0$ is a column extracted from $\mathbf{\Omega}_0$, $\mathbf{d}_j^s$ is extracted from $\mathbf{\Omega}_s$, and we require[5] that $i \neq j$ if $s = 0$.

The above definition can be seen as a generalization of the mutual coherence for the shift-invariant (convolutional) local model presented in the beginning of this chapter. Indeed, μ_s characterizes $\mathbf{\Omega}$ just as $\mu(\mathbf{D})$ characterizes the coherence of a general dictionary. Note that if $s = 0$ the above definition boils down to the traditional mutual coherence of $\mathbf{D}_L$, i.e., $\mu_0 = \mu(\mathbf{D}_L)$. It is important to stress that the atoms used in the above definition *are normalized globally* according to $\mathbf{D}$ and not $\mathbf{\Omega}$. In the Supplementary Material we comment on several interesting properties of this measure.

Similar to $\mathbf{\Omega}$, $\boldsymbol{\gamma}_i$ can be split into a set of $2n-1$ vectors of length m, each denoted by $\boldsymbol{\gamma}_{i,s}$ and corresponding to $\mathbf{\Omega}_s$. In other words, $\boldsymbol{\gamma}_i = [\boldsymbol{\gamma}_{i,-n+1}^T, \ldots, \boldsymbol{\gamma}_{i,-1}^T, \boldsymbol{\gamma}_{i,0}^T, \boldsymbol{\gamma}_{i,1}^T, \ldots, \boldsymbol{\gamma}_{i,n-1}^T]^T$. Note that previously we denoted local sparse vectors of length m by $\boldsymbol{\alpha}_j$. Yet, we will also denote them by $\boldsymbol{\gamma}_{i,s}$ in order to emphasize the fact that they correspond to the s^{th} shift within $\boldsymbol{\gamma}_i$. Denote the number of non-zeros in $\boldsymbol{\gamma}_i$ as n_i. We can also write $n_i = \sum_{s=-n+1}^{n-1} n_{i,s}$, where $n_{i,s}$ is the number of non-zeros in each $\boldsymbol{\gamma}_{i,s}$. With these definitions, we can now propose the following measure.

Definition 11. Define the stripe coherence as

$$\zeta(\boldsymbol{\gamma}_i) = \sum_{s=-n+1}^{n-1} n_{i,s}\ \mu_s.$$

According to this definition, each stripe has a coherence given by the sum of its non-zeros weighted by the shifted mutual coherence. As a particular case, if all k non-zeros correspond

[5] The condition $i \neq j$ if $s = 0$ is necessary so as to avoid the inner product of an atom by itself.

to atoms in the center sub-dictionary, $\mathbf{D}_L$, this becomes $\mu_0 k$. Note that unlike the traditional mutual coherence, this new measure depends on the location of the non-zeros in $\mathbf{\Gamma}$ – it is a function of the support of the sparse vector, and not just of the dictionary. As such, it characterizes the correlation between the atoms participating in a given stripe. In what follows, we will use the notation ζ_i for $\zeta(\boldsymbol{\gamma}_i)$.

Having formalized these tighter constructions, we now leverage them to improve the previous results. Although these theorems are generally sharper, they are harder to grasp. We begin with a recovery guarantee for the OMP and BP algorithms, followed by a discussion on their implications.

Theorem 12. (Global OMP recovery guarantee using the stripe coherence): Given the system of linear equations $\mathbf{X} = \mathbf{D}\mathbf{\Gamma}$, if a solution $\mathbf{\Gamma}$ exists satisfying

$$\max_i \ \zeta_i = \max_i \sum_{s=-n+1}^{n-1} n_{i,s}\mu_s < \ \frac{1}{2}\left(1+\mu_0\right), \tag{5.5}$$

then OMP is guaranteed to recover it.

Theorem 13. (Global BP recovery guarantee using the stripe coherence): Given the system of linear equations $\mathbf{X} = \mathbf{D}\mathbf{\Gamma}$, if a solution $\mathbf{\Gamma}$ exists satisfying

$$\max_i \ \zeta_i = \max_i \sum_{s=-n+1}^{n-1} n_{i,s}\mu_s < \ \frac{1}{2}\left(1+\mu_0\right),$$

then Basis Pursuit is guaranteed to recover it.

The corresponding proofs are similar to their counterparts presented in the preceding section but require a more delicate analysis. We include the proof for the OMP variant in the Supplementary Material, and outline the main steps required to prove the BP version.

In order to provide an intuitive interpretation for these results, the above bounds can be tied to a concrete number of non-zeros per stripe. First, notice that requiring the maximal stripe coherence to be less than a certain threshold is equal to requiring the same for every stripe:

$$\forall i \quad \sum_{s=-n+1}^{n-1} n_{i,s}\mu_s < \ \frac{1}{2}\left(1+\mu_0\right).$$

Multiplying and dividing the left-hand side of the above inequality by n_i and rearranging the resulting expression, we obtain

$$\forall i \quad n_i < \ \frac{1}{2}\frac{1+\mu_0}{\sum_{s=-n+1}^{n-1}\frac{n_{i,s}}{n_i}\mu_s}.$$

Define $\bar{\mu}_i = \sum_{s=-n+1}^{n-1}\frac{n_{i,s}}{n_i}\mu_s$. Recall that $\sum_{s=-n+1}^{n-1}\frac{n_{i,s}}{n_i} = 1$ and as such $\bar{\mu}_i$ is simply the (weighted) average shifted mutual coherence in the i^{th} stripe. Putting this definition into the

above condition, the inequality becomes

$$\forall i \quad n_i < \frac{1}{2}\left(\frac{1}{\bar{\mu}_i} + \frac{\mu_0}{\bar{\mu}_i}\right).$$

Thus, the condition in (5.5) boils down to requiring the sparsity of all stripes to be less than a certain number. Naturally, this inequality resembles the one presented in the previous section for the OMP and BP guarantees. In the Supplementary Material we prove that under the assumption that $\mu(\mathbf{D}) = \mu_0$, the shifted mutual coherence condition is at least as strong as the original one.

5.6 From Global to Local Stability Analysis

One of the cardinal motivations for this work was a series of recent practical methods addressing the convolutional sparse coding problem; and in particular, the need for their theoretical foundation. However, our results are as of yet not directly applicable to these, as we have restricted our analysis to the ideal case of noiseless signals. This is the path we undertake in the following sections, exploring the question of whether the convolutional model remains stable in the presence of noise.

Assume a clean signal $\mathbf{X}$, which admits a sparse representation $\mathbf{\Gamma}$ in terms of the convolutional dictionary $\mathbf{D}$, is contaminated with noise $\mathbf{E}$ (of bounded energy, $\|\mathbf{E}\|_2 \leq \epsilon$) to create $\mathbf{Y} = \mathbf{D\Gamma} + \mathbf{E}$. Given this noisy signal, one could propose to recover the true representation $\mathbf{\Gamma}$, or a vector close to it, by solving the P_0^ϵ problem. In this context, as mentioned in the previous section, several theoretical guarantees have been proposed in the literature. As an example, consider the stability results presented in the seminal work of [DET06]. Therein, it was shown that assuming the total number of non-zeros in $\mathbf{\Gamma}$ is less than $\frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$, the distance between the solution to the P_0^ϵ problem, $\overline{\mathbf{\Gamma}}$, and the true sparse vector, $\mathbf{\Gamma}$, satisfies

$$\|\overline{\mathbf{\Gamma}} - \mathbf{\Gamma}\|_2^2 \leq \frac{4\epsilon^2}{1 - \mu(\mathbf{D})(2\|\mathbf{\Gamma}\|_0 - 1)}. \tag{5.6}$$

In the context of our convolutional setting, however, this result provides a weak bound as it constrains the total number of non-zeros to be below a certain threshold, which scales with the local filter size n.

We now re-define the P_0^ϵ problem into a different one, capturing the convolutional structure by relying on the $\ell_{0,\infty}$ norm instead. Consider the problem:

$$(P_{0,\infty}^\epsilon): \quad \min_{\mathbf{\Gamma}} \quad \|\mathbf{\Gamma}\|_{0,\infty} \quad \text{s.t.} \quad \|\mathbf{Y} - \mathbf{D\Gamma}\|_2 \leq \epsilon.$$

In words, given a noisy measurement $\mathbf{Y}$, we seek for the $\ell_{0,\infty}$-sparsest representation vector that explains this signal up to an ϵ error. In what follows, we address the theoretical aspects of this problem and, in particular, study the stability of its solutions and practical yet secured ways for retrieving them.

5.7 Theoretical Analysis of Corrupted Signals

5.7.1 Stability of the $P_{0,\infty}^{\epsilon}$ Problem

As expected, one cannot guarantee the uniqueness of the solution to the $P_{0,\infty}^{\epsilon}$ problem, as was done for the $P_{0,\infty}$. Instead, in this subsection we shall provide a stability claim that guarantees the found solution to be close to the underlying sparse vector that generated $\mathbf{Y}$. In order to provide such an analysis, we commence by arming ourselves with the necessary mathematical tools.

Definition 14. Let $\mathbf{D}$ be a convolutional dictionary. Consider all the sub matrices $\mathbf{D}_{\mathcal{T}}$, obtained by restricting the dictionary $\mathbf{D}$ to a support $\mathcal{T}$ with an $\ell_{0,\infty}$ norm equal to k. Define δ_k as the smallest quantity such that

$$\forall \boldsymbol{\Delta} \quad (1-\delta_k)\|\boldsymbol{\Delta}\|_2^2 \leq \|\mathbf{D}_{\mathcal{T}}\boldsymbol{\Delta}\|_2^2 \leq (1+\delta_k)\|\boldsymbol{\Delta}\|_2^2$$

holds true for any choice of the support. Then, $\mathbf{D}$ is said to satisfy k-SRIP (Stripe-RIP) with constant δ_k.

Given a matrix $\mathbf{D}$, similar to the Stripe-Spark, computing the SRIP is hard or practically impossible. Thus bounding it using the mutual coherence is of practical use.

Theorem 15. (Upper bounding the SRIP via the mutual coherence): For a convolutional dictionary $\mathbf{D}$ with global mutual coherence $\mu(\mathbf{D})$, the SRIP can be upper-bounded by

$$\delta_k \leq (k-1)\mu(\mathbf{D}).$$

Assume a sparse vector $\boldsymbol{\Gamma}$ is multiplied by $\mathbf{D}$ and then contaminated by a vector $\mathbf{E}$, generating the signal $\mathbf{Y} = \mathbf{D}\boldsymbol{\Gamma} + \mathbf{E}$, such that $\|\mathbf{Y} - \mathbf{D}\boldsymbol{\Gamma}\|_2^2 \leq \epsilon^2$. Suppose we solve the $P_{0,\infty}^{\epsilon}$ problem and obtain a solution $\hat{\boldsymbol{\Gamma}}$. How close is this solution to the original $\boldsymbol{\Gamma}$? The following theorem provides an answer to this question.

Theorem 16. (Stability of the solution to the $P_{0,\infty}^{\epsilon}$ problem): Consider a sparse vector $\boldsymbol{\Gamma}$ such that $\|\boldsymbol{\Gamma}\|_{0,\infty} = k < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$, and a convolutional dictionary $\mathbf{D}$ satisfying the SRIP property for $\ell_{0,\infty} = 2k$ with coefficient δ_{2k}. Then, the distance between the true sparse vector $\boldsymbol{\Gamma}$ and the solution to the $P_{0,\infty}^{\epsilon}$ problem $\hat{\boldsymbol{\Gamma}}$ is bounded by

$$\|\boldsymbol{\Gamma} - \hat{\boldsymbol{\Gamma}}\|_2^2 \leq \frac{4\epsilon^2}{1-\delta_{2k}} \leq \frac{4\epsilon^2}{1-(2k-1)\mu(\mathbf{D})}. \tag{5.7}$$

One should wonder if the new guarantee presents any advantage when compared to the bound based on the traditional RIP. Looking at the original stability claim for the global system, as discussed in Section 5.3, the reader should compare the assumptions on the sparse vector $\boldsymbol{\Gamma}$, as well as the obtained bounds on the distance between the estimates and the original vector.

The stability claim in the P_0^ϵ problem is valid under the condition

$$\|\mathbf{\Gamma}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right).$$

In contrast, the stability claim presented above holds whenever

$$\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right).$$

This allows for significantly more non-zeros in the global signal. Furthermore, as long as the above hold, and comparing Equations (5.6) and (5.25), we have that

$$\frac{4\epsilon^2}{1-(2\|\mathbf{\Gamma}\|_{0,\infty}-1)\mu(\mathbf{D})} \ll \frac{4\epsilon^2}{1-(2\|\mathbf{\Gamma}\|_0-1)\mu(\mathbf{D})},$$

since generally $\|\mathbf{\Gamma}\|_{0,\infty} \ll \|\mathbf{\Gamma}\|_0$. This inequality implies that the above developed bound is (usually much) lower than the traditional one. In other words, the bound on the distance to the true sparse vector is much tighter and far more informative under the $\ell_{0,\infty}$ setting.

5.7.2 Stability Guarantee of OMP

Hitherto, we have shown that the solution to the $P_{0,\infty}^\epsilon$ problem will be close to the true sparse vector $\mathbf{\Gamma}$. However, it is also important to know whether this solution can be approximated by pursuit algorithms. In this subsection, we address such a question for the OMP, extending the analysis presented to the noisy setting.

In [DET06], a claim was provided for the OMP, guaranteeing the recovery of the true support of the underlying solution if

$$\|\mathbf{\Gamma}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) - \frac{1}{\mu(\mathbf{D})} \cdot \frac{\epsilon}{|\Gamma_{min}|},$$

$|\Gamma_{min}|$ being the minimal absolute value of a (non-zero) coefficients in $\mathbf{\Gamma}$. This result comes to show the importance of both the sparsity of $\mathbf{\Gamma}$ and the signal-to-noise ratio, which relates to the term $\epsilon/|\Gamma_{min}|$. In the context of our convolutional setting, this result provides a weak bound for two different reasons. First, the above bound restricts the total number of non-zeros in the representation of the signal. From Section 5.4, it is natural to seek for an alternative condition for the success of this pursuit relying on the $\ell_{0,\infty}$ norm instead. Second, notice that the rightmost term in the above bound divides the global error energy by the minimal coefficient (in absolute value) in $\mathbf{\Gamma}$. In the convolutional scenario, the energy of the error ϵ is a *global* quantity, while the minimal coefficient $|\Gamma_{min}|$ is a *local* one – thus making this term enormous, and the corresponding bound nearly meaningless. As we show next, one can harness the inherent locality of the atoms in order to replace the global quantity in the numerator with a local one: ϵ_L.

Theorem 17. (Stable recovery of global OMP in the presence of noise): Suppose a clean signal $\mathbf{X}$ has a representation $\mathbf{D\Gamma}$, and that it is contaminated with noise $\mathbf{E}$ to create the signal

$\mathbf{Y} = \mathbf{X} + \mathbf{E}$, such that $\|\mathbf{Y} - \mathbf{X}\|_2 \leq \epsilon$. Denote by ϵ_L the highest energy of all n-dimensional local patches extracted from $\mathbf{E}$. Assume $\mathbf{\Gamma}$ satisfies

$$\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) - \frac{1}{\mu(\mathbf{D})} \cdot \frac{\epsilon_L}{|\Gamma_{min}|}, \tag{5.8}$$

where $|\Gamma_{min}|$ is the minimal entry in absolute value of the sparse vector $\mathbf{\Gamma}$. Denoting by $\mathbf{\Gamma}_{\text{OMP}}$ the solution obtained by running OMP for $\|\mathbf{\Gamma}\|_0$ iterations, we are guaranteed that

1. OMP will find the correct support;
2. $\|\mathbf{\Gamma}_{\text{OMP}} - \mathbf{\Gamma}\|_2^2 \leq \frac{\epsilon^2}{1-\mu(\mathbf{D})(\|\mathbf{\Gamma}\|_{0,\infty}-1)}$.

The proof of this theorem is presented in the Supplementary Material, and the derivations therein are based on the analysis presented in [DET06], generalizing the study to the convolutional setting. Note that we have assumed that the OMP algorithm runs for $\|\mathbf{\Gamma}\|_0$ iterations. We could also propose a different approach, however, using a stopping criterion based on the norm of the residual. Under such setting, the OMP would run until the energy of the global residual is less than the energy of the noise, given by ϵ^2.

5.7.3 Stability Guarantee of Basis Pursuit Denoising via ERC

A theoretical motivation behind relaxing the $\ell_{0,\infty}$ norm to the convex ℓ_1 was already established in Section 5.4, showing that if the former is low, the BP algorithm is guaranteed to succeed. When moving to the noisy regime, the BP is naturally extended to the Basis Pursuit DeNoising (BPDN) algorithm[6], which in its Lagrangian form is defined as follows

$$\min_{\mathbf{\Gamma}} \frac{1}{2}\|\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}\|_2^2 + \lambda\|\mathbf{\Gamma}\|_1. \tag{5.9}$$

Similar to how BP was shown to approximate the solution to the $P_{0,\infty}$ problem, in what follows we will prove that the BPDN manages to approximate the solution to the $P_{0,\infty}^{\epsilon}$ problem.

Assuming the ERC is met, the stability of BP was proven under various noise models and formulations in [Tro06]. By exploiting the convolutional structure used throughout our analysis, we now show that the ERC is met given that the $\ell_{0,\infty}$ norm is small, tying the aforementioned results to our story.

Theorem 18. (ERC in the convolutional sparse model): For a convolutional dictionary $\mathbf{D}$ with mutual coherence $\mu(\mathbf{D})$, the ERC condition is met for every support $\mathcal{T}$ that satisfies

$$\|\mathcal{T}\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right).$$

Based on this and the analysis presented in [Tro06], we present a stability claim for the Lagrangian formulation of the BP problem as stated in Equation (5.9).

[6]Note that an alternative to the BPDN extension is that of the Dantzig Selector algorithm. One can envision a similar analysis to the one presented here for this algorithm as well.

Theorem 19. (Stable recovery of global Basis Pursuit in the presence of noise): Suppose a clean signal $\mathbf{X}$ has a representation $\mathbf{D}\boldsymbol{\Gamma}$, and that it is contaminated with noise $\mathbf{E}$ to create the signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. Denote by ϵ_L the highest energy of all n-dimensional local patches extracted from $\mathbf{E}$. Assume $\boldsymbol{\Gamma}$ satisfies

$$\|\boldsymbol{\Gamma}\|_{0,\infty} \leq \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D})}\right). \tag{5.10}$$

Denoting by $\boldsymbol{\Gamma}_{\mathrm{BP}}$ the solution to the Lagrangian BP formulation with parameter $\lambda = 4\epsilon_L$, we are guaranteed that

1. The support of $\boldsymbol{\Gamma}_{\mathrm{BP}}$ is contained in that of $\boldsymbol{\Gamma}$.
2. $\|\boldsymbol{\Gamma}_{\mathrm{BP}} - \boldsymbol{\Gamma}\|_\infty < \frac{15}{2}\epsilon_L$.
3. In particular, the support of $\boldsymbol{\Gamma}_{\mathrm{BP}}$ contains every index i for which $|\Gamma_i| > \frac{15}{2}\epsilon_L$.
4. The minimizer of the problem, $\boldsymbol{\Gamma}_{\mathrm{BP}}$, is unique.

The proof for both of the above, inspired by the derivations in [Ela10] and [Tro06], are presented in the Supplementary Material.

The benefit of this over traditional claims is, once again, the replacement of the ℓ_0 with the $\ell_{0,\infty}$ norm. Moreover, this result bounds the difference between the entries in $\boldsymbol{\Gamma}_{\mathrm{BP}}$ and $\boldsymbol{\Gamma}$ in terms of a local quantity – the local noise level ϵ_L. As a consequence, all atoms with coefficients above this local measure are guaranteed to be recovered.

The implications of the above theorem are far-reaching as it provides a sound theoretical back-bone for all works that have addressed the convolutional BP problem in its Lagrangian form [BEL13, Woh16, BL14, HHW15, KF14]. In Section 5.8 we will propose two additional algorithms for solving the global BP efficiently by working locally, and these methods would benefit from this theoretical result as well. As a last comment, a different and perhaps more appropriate convex relaxation for the $\ell_{0,\infty}$ norm could be suggested, such as the $\ell_{1,\infty}$ norm. This, however, remains one of our future work challenges.

5.7.4 Experiments

Following the above analysis, we now provide a numerical experiment demonstrating the above obtained bounds. The global dictionary employed here is the same as the one used for the noiseless experiments in Section 5.4, with mutual coherence $\mu(\mathbf{D}) = 0.09$, local atoms of length $n = 64$ and global ones of size $N = 640$. We sample random sparse vectors with cardinality between 1 and 500, with entries drawn from a uniform distribution with range $[-a, a]$, for varying values of a. Given these vectors, we construct global signals and contaminate them with noise. The noise is sampled from a zero-mean unit-variance white Gaussian distribution, and then normalized such that $\|\mathbf{E}\|_2 = 0.1$.

In what follows, we will first center our attention on the bounds obtained for the OMP algorithm, and then proceed to the ones corresponding to the BP. Given the noisy signals, we run OMP with a sparsity constraint, obtaining $\boldsymbol{\Gamma}_{\mathrm{OMP}}$. For each realization of the global signal,

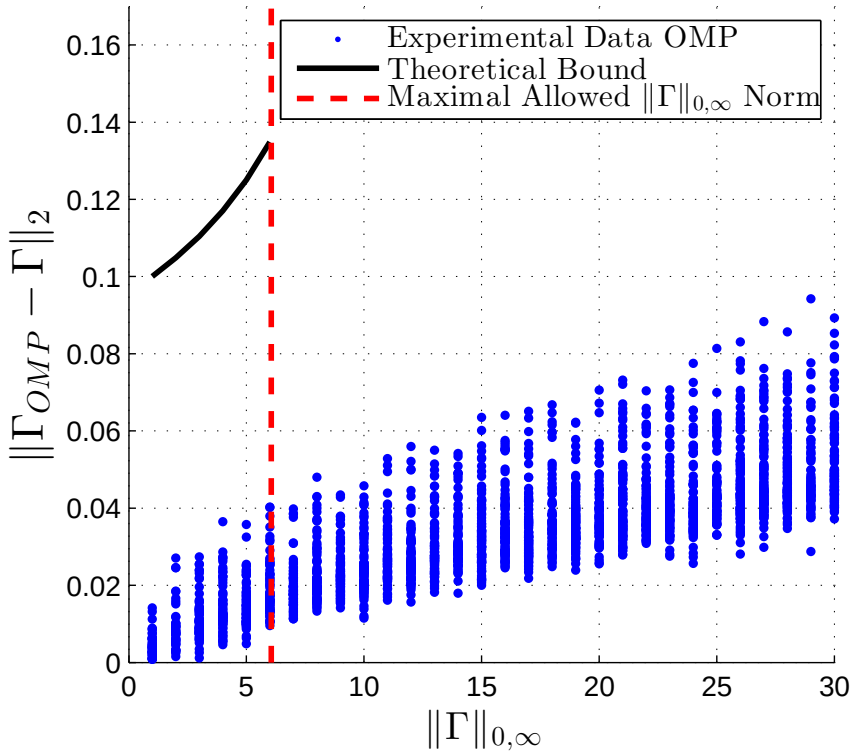
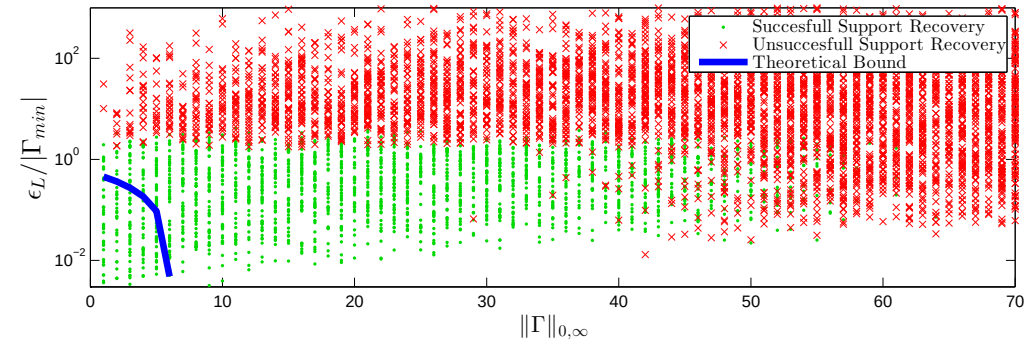


Figure 5.4: The distance $\|\mathbf{\Gamma}_{\text{OMP}} - \mathbf{\Gamma}\|_2$ as a function of the $\ell_{0,\infty}$ norm, and the corresponding theoretical bound.

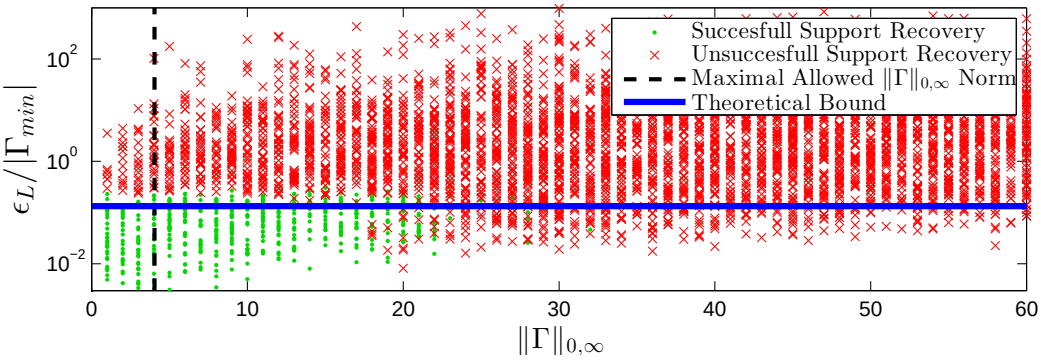
we compute the minimal entry (in absolute value) of the global sparse vector, $|\Gamma_{min}|$, and its $\ell_{0,\infty}$ norm. In addition, we compute the maximal local energy of the noise, ϵ_L, corresponding to the highest energy of a n-dimensional patch of $\mathbf{E}$.

Recall that the theorem in the previous subsection poses two claims: 1) the stability of the result in terms of $\|\mathbf{\Gamma}_{\text{OMP}} - \mathbf{\Gamma}\|_2$; and 2) the success in recovering the correct support. In Figure 5.4 we investigate the first of these points, presenting the distance between the estimated and the true sparse codes as a function of the $\ell_{0,\infty}$ norm of the original vector. As it is clear from the graph, the empirical distances are below the theoretical bound depicted in black, given by $\frac{\epsilon^2}{1-\mu(\mathbf{D})(\|\mathbf{\Gamma}\|_{0,\infty}-1)}$. According to the theorem's assumption, the sparse vector should satisfy $\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right) - \frac{1}{\mu(\mathbf{D})} \cdot \frac{\epsilon_L}{|\Gamma_{\min}|}$. The red dashed line delimits the area where this is met, with the exception that we omit the second term in the previous expression, as done previously in [DET06]. This disregards the condition on the $|\Gamma_{\min}|$ and ϵ_L (which depends on the realization). Yet, the empirical results remain stable.

In order to address the successful recovery of the support, we compute the ratio $\frac{\epsilon_L}{|\Gamma_{\min}|}$ for each realization in the experiment. In Figure 5.5(a), for each sample we denote by $\bullet$ or $\times$ the success or failure in recovering the support, respectively. Each point is plotted as a function of its $\ell_{0,\infty}$ norm and its corresponding ratio. The theoretical condition for the success of the OMP can be rewritten as $\frac{\epsilon_L}{|\Gamma_{min}|} < \frac{\mu(\mathbf{D})}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right) - \mu(\mathbf{D})\|\mathbf{\Gamma}\|_{0,\infty}$, presenting a bound on the ratio $\frac{\epsilon_L}{|\Gamma_{\min}|}$ as a function of the $\ell_{0,\infty}$ norm. This bound is depicted with a blue line, indicating that



(a) Orthogonal Matching Pursuit



(b) Basis Pursuit

Figure 5.5: The ratio $\epsilon_L/|\Gamma_{\min}|$ as a function of the $\ell_{0,\infty}$ norm, and the theoretical bound for the successful recovery of the support, for both the OMP (top) and BP (bottom) algorithms.

the empirical results agree with the theoretical claims.

One can also observe two distinct phase transitions in Figure 5.5(a). On the one hand, noting that the y axis can be interpreted as the inverse of the noise-to-signal ratio (in some sense), we see that once the noise level is too high, OMP fails in recovering the support[7]. On the other hand, similar to what was presented in the noiseless case, once the $\ell_{0,\infty}$ norm becomes too large, the algorithm is prone to fail in recovering the support.

We now shift to the empirical verification of the guarantees obtained for the BP. We employ the same dictionary as in the experiment above, and the signals are constructed in the same manner. We use the implementation of the LARS algorithm within the SPAMS package[8] in its Lagrangian formulation with the theoretically justified parameter $\lambda = 4\epsilon_L$, obtaining $\mathbf{\Gamma}_{\text{BP}}$. Once again, we compute the quantities: $|\Gamma_{min}|$, $\|\mathbf{\Gamma}\|_{0,\infty}$ and ϵ_L, and depict them in Figure 5.5(b).

Theorem 19 states that the ℓ_∞ distance between the BP solution and the true sparse vector is below $\frac{15}{2}\epsilon_L$. In Figure 5.6 we depict the ratio $\frac{\|\mathbf{\Gamma}_{\text{BP}}-\mathbf{\Gamma}\|_\infty}{\epsilon_L}$ for each realization, verifying it is indeed below $\frac{15}{2}$ as long as the $\ell_{0,\infty}$ norm is below $\frac{1}{3}\left(1+\frac{1}{\mu(\mathbf{D})}\right) \approx 4$. Next, we would like to corroborate the assertions regarding the recovery of the true support. To this end, note that the

[7]Note that the abrupt change in this phase-transition area is due to the log scale of the y axis.
[8]Freely available from http://spams-devel.gforge.inria.fr/.

theorem guarantees that all entries satisfying $|\Gamma_i| > \frac{15}{2}\epsilon_L$ shall be recovered by the BP algorithm. Alternatively, one can state that the complete support must be recovered as long as $\frac{\epsilon_L}{|\Gamma_{\min}|} < \frac{2}{15}$. To verify this claim, we plot this ratio for each realization as function of the $\ell_{0,\infty}$ norm in Figure 5.5(b), marking every point according to the success or failure of BP (in recovering the complete support). As evidenced in [Ela10], OMP seems to be far more accurate than the BP in recovering the true support. As one can see by comparing Figure 5.5(a) and 5.5(b), BP fails once the $\ell_{0,\infty}$ norm goes beyond 20, while OMP succeeds all the way until $\|\mathbf{\Gamma}\|_{0,\infty} = 40$.

5.8 From Global Pursuit to Local Processing

We now turn to analyze the practical aspects of solving the $P^{\epsilon}_{0,\infty}$ problem given the relationship $\mathbf{Y} = \mathbf{D\Gamma} + \mathbf{E}$. Motivated by the theoretical guarantees of success derived in the previous sections, the first naïve approach would be to employ global pursuit methods such as OMP and BP. However, these are computationally demanding as the dimensions of the convolutional dictionary are prohibitive for high values of N, the signal length.

As an alternative, one could attempt to solve the $P^{\epsilon}_{0,\infty}$ problem using a patch-based processing scheme. In this case, for example, one could suggest to solve a local and relatively cheaper pursuit for every patch in the signal (including overlaps) using the local dictionary $\mathbf{D}_L$. It is clear, however, that this approach will not work well under the convolutional model, because atoms used in overlapping patches are simply not present in $\mathbf{D}_L$. On the other hand, one could turn to employ $\mathbf{\Omega}$ as the *local* dictionary, but this is prone to fail in recovering the correct support of the atoms. To see this more clearly, note that there is no way to distinguish between any of the atoms having only one entry different than zero; i.e., those appearing on the extremes of $\mathbf{\Omega}$ in Figure 5.2.

As we can see, neither the naïve global approach, nor the simple patch-based processing, provide an effective strategy. Several questions arise from this discussion: Can we solve the global pursuit problem using local patch-based processing? Can the proposed algorithm rely merely on the low dimensional dictionaries $\mathbf{D}_L$ or $\mathbf{\Omega}$ while still fully solving the global problem? If so, in what form should the local patches communicate in order to achieve a global consensus? In what follows, we address these issues and provide practical and globally optimal answers.

5.8.1 Global to Local Through Bi-Level Consensus

When dealing with global problems which can be solved locally, a popular tool of choice is the Alternating Direction Method of Multipliers (ADMM) [BPC+11] in its consensus formulation. In this framework, a global objective can be decomposed into a set of local and distributed problems which attempt to reach a global agreement. We will show that this scheme can be effectively applied in the convolutional sparse coding context, providing an algorithm with a bi-level consensus interpretation.

The ADMM has been extensively used throughout the literature in convolutional sparse coding. However, as mentioned in the introduction, it has been usually applied in the Fourier

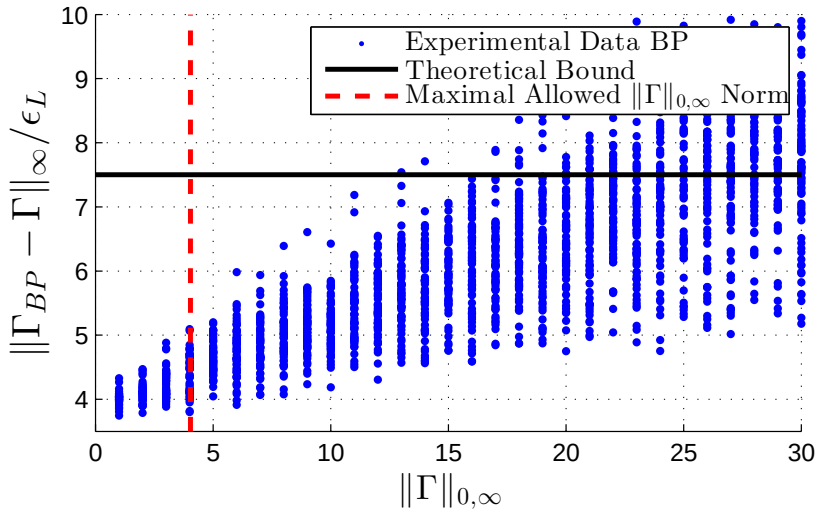


Figure 5.6: The distance $\|\mathbf{\Gamma}_{\text{BP}} - \mathbf{\Gamma}\|_\infty / \epsilon_L$ as a function of the $\ell_{0,\infty}$ norm, and the corresponding theoretical bound.

domain. As a result, the sense of locality is lost in these approaches and the connection to traditional (local) sparse coding is non-existent. On the contrary, the pursuit method we propose here is carried out in a localized fashion in the original domain, while still benefiting from the advantages of ADMM.

Recall the ℓ_1 relaxation of the global pursuit, given in Eq. (5.9). Note that the noiseless model is contained in this formulation as a particular case when λ tends to zero. Using the separability of the ℓ_1 norm, $\|\mathbf{\Gamma}\|_1 = \sum_i \|\boldsymbol{\alpha}_i\|_1$, where $\boldsymbol{\alpha}_i$ are $m-$dimensional local sparse vectors, as previously defined. In addition, using the fact that $\mathbf{R}_i\mathbf{D}\mathbf{\Gamma} = \mathbf{\Omega}\boldsymbol{\gamma}_i$, we apply a local decomposition on the first term as well. This results in

$$\min_{\{\boldsymbol{\alpha}_i\},\{\boldsymbol{\gamma}_i\}} \quad \frac{1}{2n}\sum_i \|\mathbf{R}_i\mathbf{Y} - \mathbf{\Omega}\boldsymbol{\gamma}_i\|_2^2 + \lambda \sum_i \|\boldsymbol{\alpha}_i\|_1,$$

where we have divided the first sum by the number of contributions per entry in the global signal, which is equal to the patch size n. Note that the above minimization is not equivalent to the original problem in Equation (5.9) since no explicit consensus is enforced between the local variables. Recall that the different $\boldsymbol{\gamma}_i$ overlap, and so we must enforce them to agree. In addition, $\boldsymbol{\alpha}_i$ should be constrained to be equal to the center of the corresponding $\boldsymbol{\gamma}_i$. Based on these observations, we modify the above problem by adding the appropriate constraints, obtaining

$$\begin{aligned}\min_{\{\boldsymbol{\alpha}_i\},\{\boldsymbol{\gamma}_i\},\mathbf{\Gamma}} \quad & \frac{1}{2n}\sum_i \|\mathbf{R}_i\mathbf{Y} - \mathbf{\Omega}\boldsymbol{\gamma}_i\|_2^2 + \lambda \sum_i \|\boldsymbol{\alpha}_i\|_1 \\ & \text{s.t.} \begin{cases} \mathbf{Q}\boldsymbol{\gamma}_i = \boldsymbol{\alpha}_i \\ \mathbf{S}_i\mathbf{\Gamma} = \boldsymbol{\gamma}_i \end{cases} \quad \forall i,\end{aligned}$$

where $\mathbf{Q}$ extracts the center m coefficients corresponding to $\boldsymbol{\alpha}_i$ from $\boldsymbol{\gamma}_i$, and $\mathbf{S}_i$ extracts the i^{th} stripe $\boldsymbol{\gamma}_i$ from $\mathbf{\Gamma}$.

Algorithm 5.1 Locally operating global pursuit via ADMM.

while *not converged* **do**

Local Thresholding: $\boldsymbol{\alpha}_i \leftarrow \min_{\boldsymbol{\alpha}} \lambda\|\boldsymbol{\alpha}\|_1 + \frac{\rho}{2}\|\mathbf{Q}\boldsymbol{\gamma}_i - \boldsymbol{\alpha} + \mathbf{u}_i\|_2^2$

Stripe Projection:

$$\boldsymbol{\gamma}_i \leftarrow \mathcal{M}^{-1}\left(\frac{1}{n}\boldsymbol{\Omega}^T\mathbf{R}_i\mathbf{Y} + \rho(\mathbf{S}_i\boldsymbol{\Gamma} + \bar{\mathbf{u}}_i) + \rho\mathbf{Q}^T(\boldsymbol{\alpha}_i - \mathbf{u}_i)\right),$$

where $\mathcal{M} = \rho\mathbf{Q}^T\mathbf{Q} + \frac{1}{n}\boldsymbol{\Omega}^T\boldsymbol{\Omega} + \rho\mathbf{I}$

Global Update:
$\boldsymbol{\Gamma} \leftarrow \left(\sum_i \mathbf{S}_i^T\mathbf{S}_i\right)^{-1}\sum_i \mathbf{S}_i^T(\boldsymbol{\gamma}_i - \bar{\mathbf{u}}_i)$

Dual Variables Update:
$\mathbf{u}_i \leftarrow \mathbf{u}_i + (\mathbf{Q}\boldsymbol{\gamma}_i - \boldsymbol{\alpha}_i) \quad \bar{\mathbf{u}}_i \leftarrow \bar{\mathbf{u}}_i + (\mathbf{S}_i\boldsymbol{\Gamma} - \boldsymbol{\gamma}_i)$

end

Defining $f_i(\boldsymbol{\gamma}_i) = \frac{1}{2n}\|\mathbf{R}_i\mathbf{Y} - \boldsymbol{\Omega}\boldsymbol{\gamma}_i\|_2^2$ and $g(\boldsymbol{\alpha}_i) = \lambda\|\boldsymbol{\alpha}_i\|_1$, the above problem can be minimized by employing the ADMM algorithm, as depicted in Algorithm 5.1. This is a two-level local-global consensus formulation: each m dimensional vector $\boldsymbol{\alpha}_i$ is enforced to agree with the center of its corresponding $(2n-1)m$ dimensional $\boldsymbol{\gamma}_i$, and in addition, all $\boldsymbol{\gamma}_i$ are required to agree with each other as to create a global $\boldsymbol{\Gamma}$. The above can be shown to be equivalent to the standard two-block ADMM formulation [BPC+11]. Each iteration of this method can be divided into four steps:

1. Local sparse coding that updates $\boldsymbol{\alpha}_i$ (for all i), which amounts to a simple soft thresholding operation.
2. Solution of a linear system of equations for updating $\boldsymbol{\gamma}_i$ (for all i), which boils down to a simple multiplication by a constant matrix.
3. Update of the global sparse vector $\boldsymbol{\Gamma}$, which aggregates the $\boldsymbol{\gamma}_i$ by averaging.
4. Update of the dual variables.

As can be seen, the ADMM provides a simple way of breaking the global pursuit into local operations. Moreover, the local coding step is just a projection problem onto the ℓ_1 ball, which can be solved through simple soft thresholding, implying that there is no complex pursuit involved.

Since we are in the ℓ_1 case, the function g is convex, as are the functions f_i. Therefore, the above is guaranteed to converge to the minimizer of the global BP problem. As a result, we benefit from the theoretical guarantees derived in previous sections. One could attempt, in addition, to enforce an ℓ_0 penalty instead of the ℓ_1 norm on the global sparse vector. Despite the fact that no convergence guarantees could be claimed under such formulation, the derivation of the algorithm remains practically the same, with the only exception that the soft thresholding is replaced by a hard one.

Algorithm 5.2 Global pursuit using local processing via iterative soft thresholding.

$\forall i \quad \mathbf{r}_i^0 = \mathbf{R}_i\mathbf{Y}, \quad \boldsymbol{\alpha}_i^0 = \mathbf{0}$;
k = 1 ;
while *not converged* **do**

Local Coding:
$\forall i \quad \boldsymbol{\alpha}_i^k = \mathcal{S}_{\lambda/c}\left(\boldsymbol{\alpha}_i^{k-1} + \frac{1}{c}\,\mathbf{D}_L^T\,\mathbf{r}_i^{k-1}\right)$

Computation of the Patch Averaging Aggregation:
$\widehat{\mathbf{X}}^k = \sum_i \mathbf{R}_i^T\mathbf{D}_L\boldsymbol{\alpha}_i^k$;

Update of the Residuals:
$\forall i \quad \mathbf{r}_i^k = \mathbf{R}_i\left(\mathbf{Y} - \widehat{\mathbf{X}}^k\right)$;

$k = k + 1$;
end

5.8.2 An Iterative Soft Thresholding Approach

While the above algorithm suggests a way to tackle the global problem in a local fashion, the matrix involved in the stripe projection stage, $\mathbf{Z}^{-1}$, is relatively large when compared to the dimensions of $\mathbf{D}_L$. As a consequence, the bi-level consensus introduces an extra layer of complexity to the algorithm. In what follows, we propose an alternative method based on the Iterative Soft Thresholding (IST) algorithm that relies solely on multiplications by $\mathbf{D}_L$ and features a simple intuitive interpretation and implementation. A similar approach for solving the convolutional sparse coding problem was suggested in [CPR13]. Our main concern here is to provide insights into local alternatives for the global sparse coding problem and their guarantees, whereas the work in [CPR13] focused on the optimizations aspects of this pursuit from an entirely global perspective.

Let us consider the IST algorithm [DDDM04] which minimizes the global objective in Equation (5.9), by iterating the following updates

$$\boldsymbol{\Gamma}^k = \mathcal{S}_{\lambda/c}\left(\boldsymbol{\Gamma}^{k-1} + \frac{1}{c}\mathbf{D}^T(\mathbf{Y} - \mathbf{D}\boldsymbol{\Gamma}^{k-1})\right),$$

where $\mathcal{S}$ applies an entry-wise soft thresholding operation with threshold λ/c. Interpreting the above as a projected gradient descent, the coefficient c relates to the gradient step size and should be set according to the maximal singular value of the matrix $\mathbf{D}$ in order to guarantee convergence [DDDM04].

The above algorithm might at first seem undesirable due to the multiplications of the residual $\mathbf{Y} - \mathbf{D}\boldsymbol{\Gamma}^{k-1}$ with the global dictionary $\mathbf{D}$. Yet, as we show in the Supplementary Material, such a multiplication does not need to be carried out explicitly due to the convolutional structure imposed on our dictionary. In fact, the above is mathematical equivalent to an algorithm that

performs local updates given by

$$\boldsymbol{\alpha}_i^k = \mathcal{S}_{\lambda/c}\left(\boldsymbol{\alpha}_i^{k-1} + \frac{1}{c}\,\mathbf{D}_L^T\,\mathbf{r}_i^{k-1}\right),$$

where $\mathbf{r}_i^k = \mathbf{R}_i(\mathbf{Y} - \mathbf{D}\boldsymbol{\Gamma}^{k-1})$ is a patch from the global residual. This scheme is depicted in Algorithm 5.2.

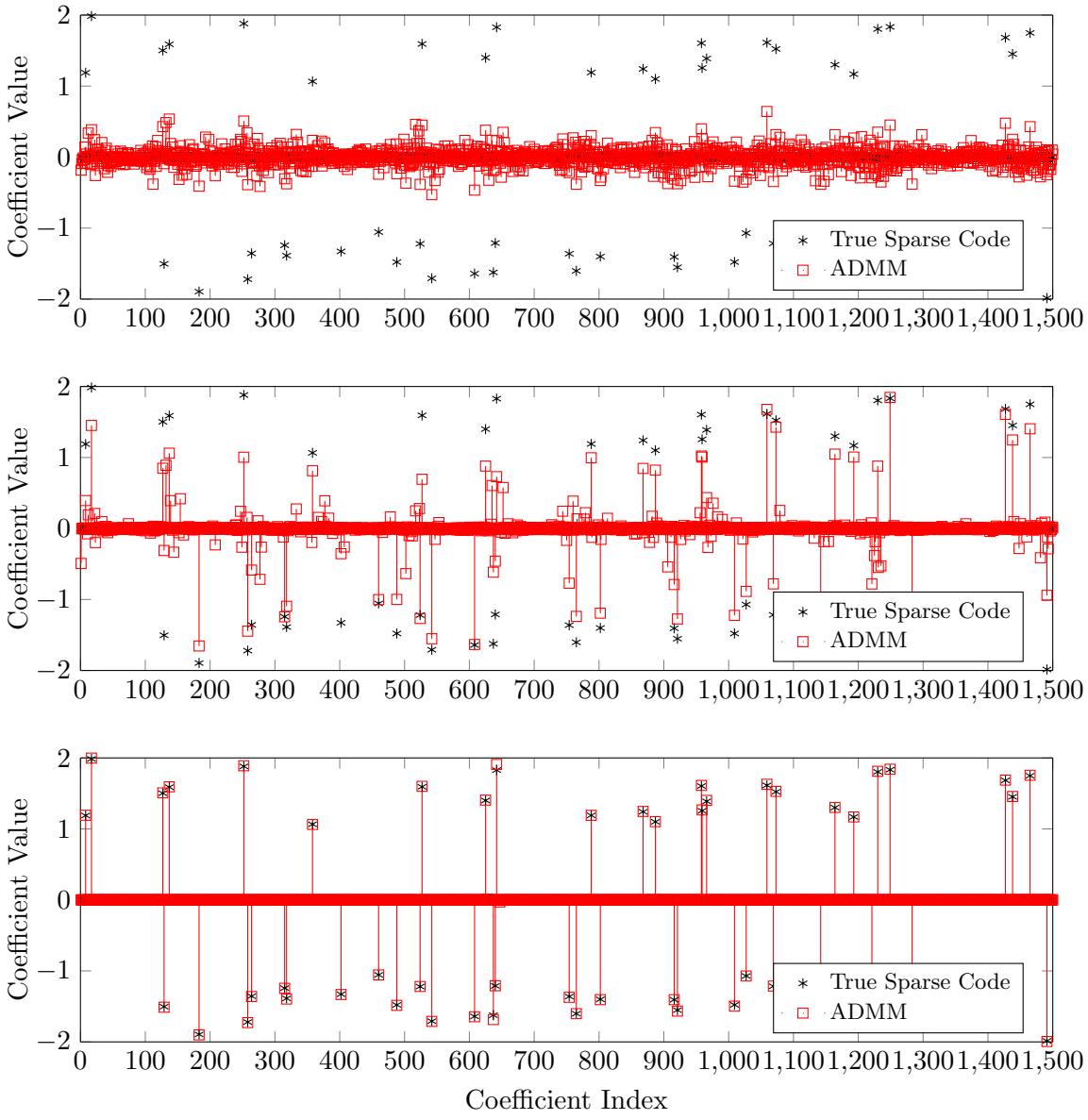


Figure 5.7: The sparse vector $\boldsymbol{\Gamma}$ after the global update stage in the ADMM algorithm at iterations 20 (top), 200 (middle) and 1000 (bottom). An ℓ_1 norm formulation was used for this experiment, in a noiseless setting.

From an optimization point of view, one can interpret each iteration of the above as a scatter and gather process: local residuals are first extracted and scattered to different nodes where

they undergo shrinkage operations, and the results are then gathered for the re-computation of the global residual. From an image processing point of view, this algorithm decomposes a signal into overlapping patches, *restores* these separately and then aggregates the result for the next iteration. Notably, this is very reminiscent of the patch averaging scheme, as described in the introduction, and it shows for the first time the relation between patch averaging and the convolutional sparse model. While the former processes every patch once and independently, the above algorithm indicates that one must iterate this process if one is to reach global consensus.

Assuming the step size is chosen appropriately, the above algorithm is also guaranteed to converge to the solution of the global BP. As such, our theoretical analysis holds in this case as well. Alternatively, one could attempt to employ an ℓ_0 approach, using a global iterative hard thresholding algorithm. In this case, however, there are no theoretical guarantees in terms of the $\ell_{0,\infty}$ norm. Still, we believe that a similar analysis to the one taken throughout this work could lead to such claims.

5.8.3 Experiments

Next, we proceed to provide empirical results for the above described methods. To this end, we take an undercomplete DCT dictionary of size 25×5, and use it as $\mathbf{D}_L$ in order to construct the global convolutional dictionary $\mathbf{D}$ for a signal of length $N = 300$. We then generate a random global sparse vector $\mathbf{\Gamma}$ with 50 non-zeros, with entries distributed uniformally in the range $[-2, -1] \cup [1, 2]$, creating the signal $\mathbf{X} = \mathbf{D}\mathbf{\Gamma}$.

We first employ the ADMM and IST algorithms in a noiseless scenario in order to minimize the global BP and find the underlying sparse vector. Since there is no noise added in this case, we decrease the penalty parameter λ progressively throughout the iterations, making this value tend to zero as suggested in the previous subsection. In Figure 5.7 we present the evolution of the estimated $\hat{\mathbf{\Gamma}}$ for the ADMM solver throughout the iterations, after the global update stage. Note how the algorithm progressively increases the consensus and eventually recovers the true sparse vector. Equivalent plots are obtained for the IST method, and these are therefore omitted.

To extend the experiment to the noisy case, we contaminate the previous signal with additive white Gaussian noise of different standard deviations: $\sigma = 0.02, 0.04, 0.06$. We then employ both local algorithms to solve the corresponding BPDN problems, and analyze the ℓ_2 distance between their estimated sparse vector and the true one, as a function of time. These results are depicted in Figure 5.8, where we include for completion the distance of the solution achieved by the global BP in the noisy cases. A few observations can be drawn from these results. Note that both algorithms converge to the solution of the global BP in all cases. In particular, the IST converges significantly faster than the ADMM method. Interestingly, despite the later requiring a smaller number of iterations to converge, these are relatively more expensive than those of the IST, which employs only multiplications by the small $\mathbf{D}_L$.

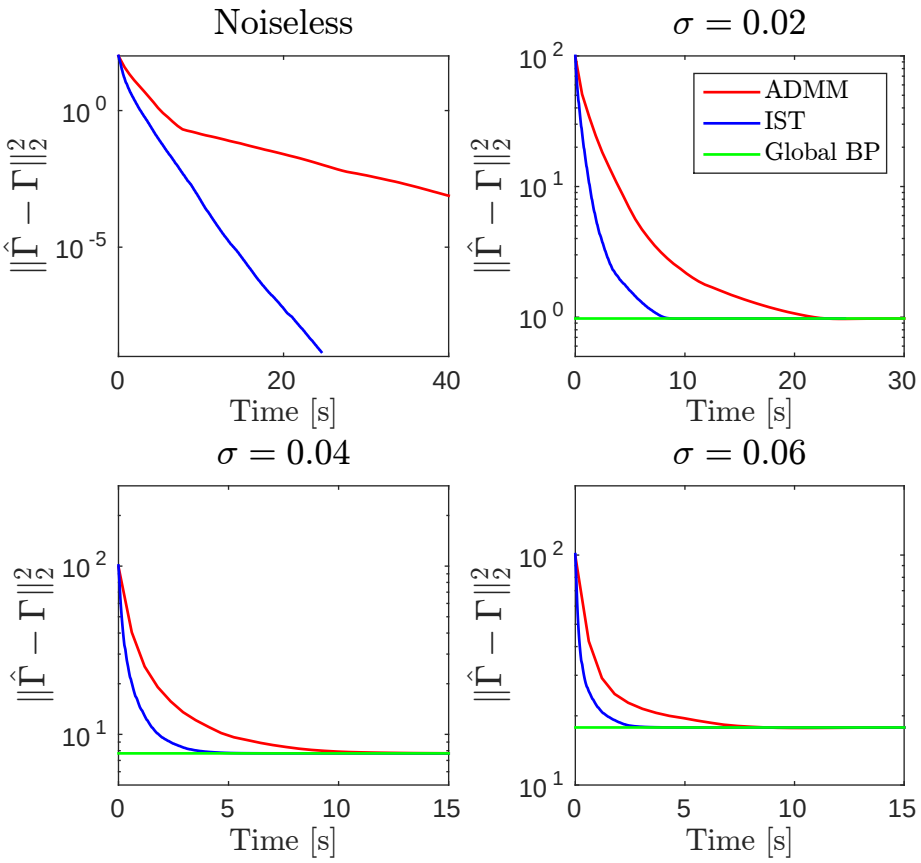


Figure 5.8: Distance between the estimate $\hat{\mathbf{\Gamma}}$ and the underlying solution $\mathbf{\Gamma}$ as a function of time for the IST and the ADMM algorithms compared to the solution obtained by solving the global BP.

5.9 Chapter Conclusion

In this chapter we have presented a formal analysis of the convolutional sparse representation model. In doing so, we have reformulated the objective of the global pursuit, introducing the $\ell_{0,\infty}$ norm and the corresponding $P_{0,\infty}$ problem, and proven the uniqueness of its solution. By migrating from the P_0 to the $P_{0,\infty}$ problem, we were able to provide meaningful guarantees for the success of popular algorithms in the noiseless case, improving on traditional bounds that were shown to be very pessimistic under the convolutional case. In order to achieve such results, we have generalized a series of concepts such as Spark and the mutual coherence to their counterparts in the convolutional setting.

Striding on the foundations paved in the first part of this work, we moved on to present a series of stability results for the convolutional sparse model in the presence of noise, providing guarantees for corresponding pursuit algorithms. These were possible due to our migration from the ℓ_0 to the $\ell_{0,\infty}$ norm, together with the generalization and utilization of concepts such as RIP and ERC. Seeking for a connection between traditional patch-based processing and the convolutional sparse model, we finally proposed two efficient methods that solve the global

pursuit while working locally.

We envision many possible directions of future work, and here we outline some of them:

- We could extend our study, which considers only worst-case scenarios, to an average-performance analysis. By assuming more information about the model, it might be possible to quantify the probability of success of pursuit methods in the convolutional case. Such results would close the gap between current bounds and empirical results.

- From an application point of view, we envision that interesting algorithms could be proposed to tackle real problems in signal and image processing while using the convolutional model. We note that while convolutional sparse coding has been applied to various problems, simple inverse problems such as denoising have not yet been properly addressed. We believe that the analysis presented in this work could facilitate the development of such algorithms by showing how to leverage on the subtleties of this model.

- Interestingly, even though we have declared the $P_{0,\infty}$ problem as our goal, at no point have we actually attempted to tackle it directly. What we have shown instead is that popular algorithms succeed in finding its solution. One could perhaps propose an algorithm specifically tailored for solving this problem – or its convex relaxation ($\ell_{1,\infty}$). Such a method might be beneficial from both a theoretical and a practical aspect.

All these points, and more, are matter of current research.

5.10 Chapter Appendix

5.10.1 On the $\ell_{0,\infty}$ Norm

Theorem 20. The triangle inequality holds for the $\ell_{0,\infty}$ norm.

Proof. Let $\mathbf{\Gamma}^1$ and $\mathbf{\Gamma}^2$ be two global sparse vectors. Denote the i^{th} stripe extracted from each as $\boldsymbol{\gamma}_i^1$ and $\boldsymbol{\gamma}_i^2$, respectively. Notice that

$$\begin{aligned}\|\mathbf{\Gamma}^1+\mathbf{\Gamma}^2\|_{0,\infty} &= \max_i \|\boldsymbol{\gamma}_i^1+\boldsymbol{\gamma}_i^2\|_0 \le \max_i \left(\|\boldsymbol{\gamma}_i^1\|_0+\|\boldsymbol{\gamma}_i^2\|_0\right) \\ &\le \max_i \|\boldsymbol{\gamma}_i^1\|_0 + \max_i \|\boldsymbol{\gamma}_i^2\|_0 = \|\mathbf{\Gamma}^1\|_{0,\infty}+\|\mathbf{\Gamma}^2\|_{0,\infty}.\end{aligned}$$

In the first inequality we have used the triangle inequality of the ℓ_0 norm. □

5.10.2 Theoretical Analysis of Ideal Signals

Theorem 5. (Uncertainty and uniqueness using Stripe-Spark): Let $\mathbf{D}$ be a convolutional dictionary with Stripe-Spark σ_∞. If a solution $\mathbf{\Gamma}$ obeys $\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\sigma_\infty$, then this is necessarily the global optimum for the $P_{0,\infty}$ problem for the signal $\mathbf{D\Gamma}$.

Proof. Let $\hat{\mathbf{\Gamma}} \neq \mathbf{\Gamma}$ be an alternative solution. Then $\mathbf{D}\left(\mathbf{\Gamma}-\hat{\mathbf{\Gamma}}\right) = 0$. By definition of the Stripe-Spark

$$\|\mathbf{\Gamma}-\hat{\mathbf{\Gamma}}\|_{0,\infty} \ge \sigma_\infty.$$

Using the triangle inequality of the $\ell_{0,\infty}$ norm,

$$\|\mathbf{\Gamma}\|_{0,\infty} + \|\hat{\mathbf{\Gamma}}\|_{0,\infty} \ge \|\mathbf{\Gamma}-\hat{\mathbf{\Gamma}}\|_{0,\infty} \ge \sigma_\infty.$$

This result poses an uncertainty principle for $\ell_{0,\infty}$ sparse solutions of the system $\mathbf{X} = \mathbf{D\Gamma}$, suggesting that if a very sparse solution is found, all alternative solutions must be much denser. Since $\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\sigma_\infty$, we must have that $\|\hat{\mathbf{\Gamma}}\|_{0,\infty} > \frac{1}{2}\sigma_\infty$, or in other words, every solution other than $\mathbf{\Gamma}$ has higher $\ell_{0,\infty}$ norm, thus making $\mathbf{\Gamma}$ the global solution for the $P_{0,\infty}$ problem. □

Theorem 6. (Lower bounding the Stripe-Spark via the mutual coherence): For a convolutional dictionary $\mathbf{D}$ with mutual coherence $\mu(\mathbf{D})$, the Stripe-Spark can be lower-bounded by

$$\sigma_\infty(\mathbf{D}) \ge 1 + \frac{1}{\mu(\mathbf{D})}.$$

Proof. Let $\mathbf{\Delta}$ be a vector such that $\mathbf{\Delta} \neq \mathbf{0}$ and $\mathbf{D\Delta} = \mathbf{0}$. Note that we can write

$$\mathbf{D}_\mathcal{T}\mathbf{\Delta}_\mathcal{T} = \mathbf{0}, \tag{5.11}$$

where $\mathbf{\Delta}_\mathcal{T}$ is the vector $\mathbf{\Delta}$ restricted to its support $\mathcal{T}$, and $\mathbf{D}_\mathcal{T}$ is the dictionary composed of the corresponding atoms. Consider now the Gram matrix, $\mathbf{G}^\mathcal{T} = \mathbf{D}_\mathcal{T}^T\mathbf{D}_\mathcal{T}$, which corresponds to a portion extracted from the global Gram matrix $\mathbf{D}^T\mathbf{D}$. The relation in Equation (5.11)

suggests that $\mathbf{D}_{\mathcal{T}}$ has a nullspace, which implies that its Gram matrix must have at least one eigenvalue equal to zero. Using Lemma 1, the lower bound on the eigenvalues of $\mathbf{G}^{\mathcal{T}}$ is given by $1-(k-1)\mu(\mathbf{D})$, where k is the $\ell_{0,\infty}$ norm of $\mathbf{\Delta}$. Therefore, we must have that $1-(k-1)\mu(\mathbf{D}) \leq 0$, or equally $k \geq 1+\frac{1}{\mu(\mathbf{D})}$. We conclude that a vector $\mathbf{\Delta}$, which is in the null-space of $\mathbf{D}$, must always have an $\ell_{0,\infty}$ norm of at least $1+\frac{1}{\mu(\mathbf{D})}$, and so the Stripe-Spark σ_∞ is also bounded by this number. □

Theorem 8. (Global OMP recovery guarantee using $\ell_{0,\infty}$ norm): Given the system of linear equations $\mathbf{X}=\mathbf{D}\mathbf{\Gamma}$, if a solution $\mathbf{\Gamma}$ exists satisfying

$$\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right), \tag{5.12}$$

then OMP is guaranteed to recover it.

Proof. Denoting by $\mathcal{T}$ the support of the solution $\mathbf{\Gamma}$, we can write

$$\mathbf{X}=\mathbf{D}\mathbf{\Gamma}=\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t. \tag{5.13}$$

Suppose, without loss of generality, that the sparsest solution has its largest coefficient (in absolute value) in Γ_i. For the first step of the OMP to choose one of the atoms in the support, we require

$$|\mathbf{d}_i^T\mathbf{X}| > \max_{j\notin\mathcal{T}}|\mathbf{d}_j^T\mathbf{X}|.$$

Substituting Equation (5.13) in this requirement we obtain

$$\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_i\right| > \max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j\right|. \tag{5.14}$$

Using the reverse triangle inequality, the assumption that the atoms are normalized, and that $|\Gamma_i| \geq |\Gamma_t|$, we construct a lower bound for the left hand side:

$$\begin{aligned}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_i\right| &\geq |\Gamma_i| - \sum_{t\in\mathcal{T},t\neq i}|\Gamma_t|\cdot|\mathbf{d}_t^T\mathbf{d}_i| \\ &\geq |\Gamma_i| - |\Gamma_i|\sum_{t\in\mathcal{T},t\neq i}|\mathbf{d}_t^T\mathbf{d}_i|.\end{aligned}$$

Consider the stripe which completely contains the i^{th} atom as shown in Figure 5.9. Notice that $\mathbf{d}_t^T\mathbf{d}_i$ is zero for every atom too far from $\mathbf{d}_i$ because the atoms do not overlap. Denoting the stripe which fully contains the i^{th} atom as $p(i)$ and its support as $\mathcal{T}_{p(i)}$, we can restrict the summation as:

$$\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_i\right| \geq |\Gamma_i| - |\Gamma_i|\sum_{t\in\mathcal{T}_{p(i)},t\neq i}|\mathbf{d}_t^T\mathbf{d}_i|. \tag{5.15}$$

We can bound the right side by using the number of non-zeros in the support $\mathcal{T}_{p(i)}$, denoted by

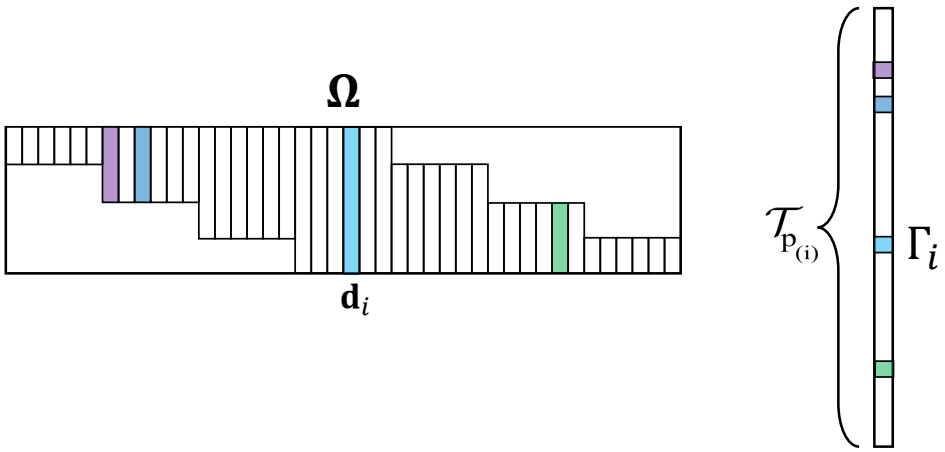


Figure 5.9: The $p_{(i)}$ stripe of atom $\mathbf{d}_i$.

$n_{p(i)}$, together with the definition of the mutual coherence, obtaining:

$$\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_i\right| \geq |\Gamma_i| - |\Gamma_i|\cdot(n_{p(i)}-1)\cdot\mu(\mathbf{D}).$$

Using the definition of the $\ell_{0,\infty}$ norm, we obtain

$$\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_i\right| \geq |\Gamma_i| - |\Gamma_i|\cdot(\|\mathbf{\Gamma}\|_{0,\infty}-1)\cdot\mu(\mathbf{D}).$$

Now, we construct an upper bound for the right hand side of Equation (5.14), using the triangle inequality and the fact that $|\Gamma_i|$ is the maximal value in the sparse vector:

$$\begin{aligned}\max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j\right| &\leq \max_{j\notin\mathcal{T}}\sum_{t\in\mathcal{T}}|\Gamma_t|\cdot|\mathbf{d}_t^T\mathbf{d}_j| \\ &\leq |\Gamma_i|\max_{j\notin\mathcal{T}}\sum_{t\in\mathcal{T}}|\mathbf{d}_t^T\mathbf{d}_j|.\end{aligned} \tag{5.16}$$

Relying on the same rational as above, we obtain:

$$\begin{aligned}\max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j\right| &\leq |\Gamma_i|\max_{j\notin\mathcal{T}}\sum_{t\in\mathcal{T}_{p(j)}}|\mathbf{d}_t^T\mathbf{d}_j| \\ \leq |\Gamma_i|\max_{j\notin\mathcal{T}}\ n_{p(j)}\cdot\mu(\mathbf{D}) &\leq |\Gamma_i|\cdot\|\mathbf{\Gamma}\|_{0,\infty}\cdot\mu(\mathbf{D}).\end{aligned}$$

Using both bounds, we get

$$\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_i\right| \geq |\Gamma_i| - |\Gamma_i|\cdot(\|\mathbf{\Gamma}\|_{0,\infty}-1)\cdot\mu(\mathbf{D})$$
$$> |\Gamma_i|\cdot\|\mathbf{\Gamma}\|_{0,\infty}\mu(\mathbf{D}) \geq \max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j\right|.$$

Thus,

$$1-(\|\mathbf{\Gamma}\|_{0,\infty}-1)\cdot\mu(\mathbf{D}) > \|\mathbf{\Gamma}\|_{0,\infty}\cdot\mu(\mathbf{D}).$$

From this we obtain the requirement stated in the theorem. Thus, this condition guarantees the success of the first OMP step, implying it will choose an atom inside the true support.

The next step in the OMP algorithm is an update of the residual. This is done by decreasing a term proportional to the chosen atom (or atoms within the correct support in subsequent iterations) from the signal. Thus, this residual is also a linear combination of the same atoms as the original signal. As a result, the $\ell_{0,\infty}$ norm of the residual's representation is less or equal than the one of the true sparse code $\mathbf{\Gamma}$. Using the same set of steps we obtain that the condition on the $\ell_{0,\infty}$ norm (5.12) guarantees that the algorithm chooses again an atom from the true support of the solution. Furthermore, the orthogonality enforced by the least-squares step guarantees that the same atom is never chosen twice. As a result, after $\|\mathbf{\Gamma}\|_0$ iterations the OMP will find all the atoms in the correct support, reaching a residual equal to zero. □

Theorem 9. (Global Basis Pursuit recovery guarantee using the $\ell_{0,\infty}$ norm): For the system of linear equations $\mathbf{D\Gamma}=\mathbf{X}$, if a solution $\mathbf{\Gamma}$ exists obeying

$$\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right),$$

then Basis Pursuit is guaranteed to recover it.

Proof. Define the following set

$$\mathbf{C} = \left\{\hat{\mathbf{\Gamma}} \;\middle|\; \begin{array}{cc} \hat{\mathbf{\Gamma}}\neq\mathbf{\Gamma}, & \mathbf{D}(\hat{\mathbf{\Gamma}}-\mathbf{\Gamma})=\mathbf{0} \\ \|\hat{\mathbf{\Gamma}}\|_1 \leq \|\mathbf{\Gamma}\|_1, & \|\hat{\mathbf{\Gamma}}\|_{0,\infty} > \|\mathbf{\Gamma}\|_{0,\infty} \end{array}\right\}.$$

This set contains all alternative solutions which have lower or equal ℓ_1 norm and higher $\|\cdot\|_{0,\infty}$ norm. If this set is non-empty, the solution of the basis pursuit is different from $\mathbf{\Gamma}$, implying failure. In view of our uniqueness result, and the condition posed in this theorem on the $\ell_{0,\infty}$ cardinality of $\mathbf{\Gamma}$, every solution $\hat{\mathbf{\Gamma}}$ which is not equal to $\mathbf{\Gamma}$ must have a higher $\|\cdot\|_{0,\infty}$ norm. Thus, we can omit the requirement $\|\hat{\mathbf{\Gamma}}\|_{0,\infty} > \|\mathbf{\Gamma}\|_{0,\infty}$ from $\mathbf{C}$.

By defining $\mathbf{\Delta}=\hat{\mathbf{\Gamma}}-\mathbf{\Gamma}$, we obtain a shifted version of the set,

$$\mathbf{C}_s = \left\{\mathbf{\Delta} \;\middle|\; \begin{array}{c} \mathbf{\Delta}\neq\mathbf{0}, \quad \mathbf{D\Delta}=\mathbf{0} \\ \mathbf{0} \geq \|\mathbf{\Delta}+\mathbf{\Gamma}\|_1 - \|\mathbf{\Gamma}\|_1 \end{array}\right\}.$$

In what follows, we will enlarge the set $\mathbf{C}_s$ and prove that it remains empty even after this expansion. Since $\mathbf{D}\boldsymbol{\Delta} = \mathbf{0}$, then $\mathbf{D}^T\mathbf{D}\boldsymbol{\Delta} = \mathbf{0}$. By subtracting $\boldsymbol{\Delta}$ from both sides, we obtain

$$-\boldsymbol{\Delta} = (\mathbf{D}^T\mathbf{D} - \mathbf{I})\boldsymbol{\Delta}. \tag{5.17}$$

Taking an entry-wise absolute value on both sides, we obtain

$$|\boldsymbol{\Delta}| = |(\mathbf{D}^T\mathbf{D} - \mathbf{I})\boldsymbol{\Delta}| \leq |\mathbf{D}^T\mathbf{D} - \mathbf{I}| \cdot |\boldsymbol{\Delta}|, \tag{5.18}$$

where we have applied the triangle inequality to the multiplication of the i^{th} row of $(\mathbf{D}^T\mathbf{D} - \mathbf{I})$ by the vector $\boldsymbol{\Delta}$. Note that in the convolutional case $\mathbf{D}^T\mathbf{D}$ is zero for inner products of atoms which do not overlap. Furthermore, the i^{th} row of $\mathbf{D}^T\mathbf{D}$ is non-zero only in the indices which correspond to the stripe that fully contains the i^{th} atom, and these non-zero entries can be bounded by $\mu(\mathbf{D})$. Thus, extracting the i^{th} row from the above equation gives

$$|\Delta_i| \leq \mu(\mathbf{D}) \left(\|\boldsymbol{\delta}_{p(i)}\|_1 - |\Delta_i| \right),$$

where $p(i)$ is the stripe centered around the i^{th} atom and $\boldsymbol{\delta}_{p(i)}$ is the corresponding sparse vector of length $(2n-1)m$ extracted from $\boldsymbol{\Delta}$, as can be seen in Figure 5.10. This can be written as

$$|\Delta_i| \leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D}) + 1} \|\boldsymbol{\delta}_{p(i)}\|_1.$$

The above expression is a relaxation of the equality in Equation (5.17), since each entry Δ_i is no longer constrained to a specific value, but rather bounded from below and above. Therefore, by putting the above into $\mathbf{C}_s$, we obtain a larger set $\mathbf{C}_s^1$:

$$\mathbf{C}_s \subseteq \mathbf{C}_s^1 = \left\{ \boldsymbol{\Delta} \;\middle|\; \begin{array}{c} \boldsymbol{\Delta} \neq \mathbf{0}, \quad \mathbf{0} \geq \|\boldsymbol{\Delta} + \boldsymbol{\Gamma}\|_1 - \|\boldsymbol{\Gamma}\|_1 \\ |\Delta_i| \leq \dfrac{\mu(\mathbf{D})}{\mu(\mathbf{D}) + 1} \|\boldsymbol{\delta}_{p(i)}\|_1, \quad \forall i \end{array} \right\}.$$

Next, let us examine the second requirement

$$\begin{aligned} \mathbf{0} \geq & \|\boldsymbol{\Delta} + \boldsymbol{\Gamma}\|_1 - \|\boldsymbol{\Gamma}\|_1 \\ = & \sum_{i \in \mathcal{T}(\boldsymbol{\Gamma})} (|\Delta_i + \Gamma_i| - |\Gamma_i|) + \sum_{i \notin \mathcal{T}(\boldsymbol{\Gamma})} |\Delta_i|, \end{aligned} \tag{5.19}$$

where, as before, $\mathcal{T}(\boldsymbol{\Gamma})$ denotes the support of $\boldsymbol{\Gamma}$. Using the reverse triangle inequality, $|a+b|-|b| \geq -|a|$, we obtain

$$\begin{aligned} \mathbf{0} \geq & \sum_{i \in \mathcal{T}(\boldsymbol{\Gamma})} (|\Delta_i + \Gamma_i| - |\Gamma_i|) + \sum_{i \notin \mathcal{T}(\boldsymbol{\Gamma})} |\Delta_i| \\ \geq & \sum_{i \in \mathcal{T}(\boldsymbol{\Gamma})} -|\Delta_i| + \sum_{i \notin \mathcal{T}(\boldsymbol{\Gamma})} |\Delta_i| = \|\boldsymbol{\Delta}\|_1 - 2\mathbf{1}_{\mathcal{T}(\boldsymbol{\Gamma})}^T |\boldsymbol{\Delta}|, \end{aligned} \tag{5.20}$$

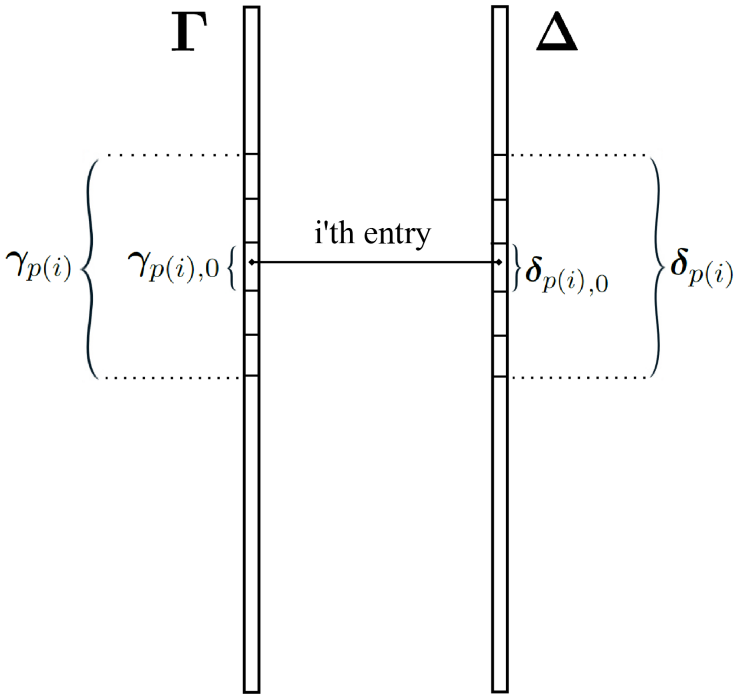


Figure 5.10: On the left we have the global sparse vector $\mathbf{\Gamma}$, a stripe $\boldsymbol{\gamma}_{p(i)}$ (centered around the i^{th} atom) extracted from it, and the center of this stripe $\boldsymbol{\gamma}_{p(i),0}$. The length of the stripe $\boldsymbol{\gamma}_{p(i)}$ is $(2n-1)m$ and the length of $\boldsymbol{\gamma}_{p(i),0}$ is m. On the right we have the corresponding global vector $\mathbf{\Delta}$. Notice that if we were to consider the $i+1$ entry instead of the i^{th}, the vector corresponding to $\boldsymbol{\delta}_{p(i)}$ would not change because the atoms i and $i+1$ are fully overlapping.

where the vector $\mathbf{1}_{\mathcal{T}(\mathbf{\Gamma})}$ contains ones in the entries corresponding to the support of $\mathbf{\Gamma}$ and zeros elsewhere. Note that every vector satisfying Equation (5.19) will necessarily satisfy Equation (5.20). Therefore, by relaxing this constraint in $\mathbf{C}_s^1$, we obtain a larger set $\mathbf{C}_s^2$

$$\mathbf{C}_s^1 \subseteq \mathbf{C}_s^2 = \left\{ \mathbf{\Delta} \;\middle|\; \begin{array}{c} \mathbf{\Delta} \neq \mathbf{0}, \quad \mathbf{0} \geq \|\mathbf{\Delta}\|_1 - 2\mathbf{1}_{\mathcal{T}(\mathbf{\Gamma})}^T |\mathbf{\Delta}| \\ |\Delta_i| \leq \dfrac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \|\boldsymbol{\delta}_{p(i)}\|_1, \quad \forall i \end{array} \right\}.$$

Next, we will show the above defined set is empty for a small-enough support. We begin by summing the inequalities $|\Delta_i| \leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \|\boldsymbol{\delta}_{p(i)}\|_1$ over the support of $\boldsymbol{\gamma}_{p(i),0}$. Recall that $\boldsymbol{\gamma}_{p(i)}$ is defined to be a stripe of length $(2n-1)m$ extracted from the global representation vector and $\boldsymbol{\gamma}_{p(i),0}$ corresponds to the central m coefficients in the $p(i)$ stripe. Also, note that $\boldsymbol{\delta}_{p(i)}$ is equal for all the entries inside the support of $\boldsymbol{\gamma}_{p(i),0}$. Since all the atoms inside the support of $\boldsymbol{\gamma}_{p(i),0}$ are fully overlapping, $\boldsymbol{\delta}_{p(i)}$ does not change, as explained in Figure 5.10. Thus, we obtain

$$\mathbf{1}_{\mathcal{T}(\boldsymbol{\gamma}_{p(i),0})}^T |\mathbf{\Delta}| \leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \cdot \|\boldsymbol{\gamma}_{p(i),0}\|_0 \cdot \|\boldsymbol{\delta}_{p(i)}\|_1.$$

Summing over all different $p(i)$ we obtain

$$\mathbf{1}^T_{\mathcal{T}(\mathbf{\Gamma})}|\mathbf{\Delta}| \leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \sum_k \|\boldsymbol{\gamma}_{k,0}\|_0 \cdot \|\boldsymbol{\delta}_k\|_1. \tag{5.21}$$

Notice that in the sum above we multiply the ℓ_0-norm of the *local sparse vector* $\boldsymbol{\gamma}_{k,0}$ by the ℓ_1 norm of the *stripe* $\boldsymbol{\delta}_k$. In what follows, we will show that, instead, we could multiply the ℓ_0-norm of the *stripe* $\boldsymbol{\gamma}_k$ by the ℓ_1 norm of the *local sparse vector* $\boldsymbol{\delta}_{k,0}$, thus changing the order between the two. As a result, we will obtain the following inequality:

$$\mathbf{1}^T_{\mathcal{T}(\mathbf{\Gamma})}|\mathbf{\Delta}| \leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \sum_k \|\boldsymbol{\gamma}_{k}\|_0 \cdot \|\boldsymbol{\delta}_{k,0}\|_1.$$

Returning to Equation (5.21), we begin by decomposing the ℓ_1 norm of the stripe $\boldsymbol{\delta}_k$ into all possible shifts ($m-$dimensional chunks) and pushing the sum outside, obtaining:

$$\begin{aligned}
\mathbf{1}^T_{\mathcal{T}(\mathbf{\Gamma})}|\mathbf{\Delta}| &\leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \sum_k \|\boldsymbol{\gamma}_{k,0}\|_0 \cdot \|\boldsymbol{\delta}_k\|_1 \\
&= \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \sum_k \left(\|\boldsymbol{\gamma}_{k,0}\|_0 \sum_{j=k-n+1}^{k+n-1} \|\boldsymbol{\delta}_{j,0}\|_1 \right) \\
&= \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \sum_k \sum_{j=k-n+1}^{k+n-1} \|\boldsymbol{\gamma}_{k,0}\|_0 \|\boldsymbol{\delta}_{j,0}\|_1.
\end{aligned} \tag{5.22}$$

Define a banded matrix $\mathbf{A}$ (with a band of width $2n-1$) such that $\mathbf{A}_{k,j} = \|\boldsymbol{\gamma}_{k,0}\|_0 \cdot \|\boldsymbol{\delta}_{j,0}\|_1$, where $k-n+1 \leq j \leq k+n-1$. Notice that the summation in (5.22) is equal to the sum of all entries in this matrix, where the first sum considers all its rows k while the second sum considers all its columns j (the second sum is restricted to the non-zero band). Instead, this interpretation suggests that we could first sum over all the columns j, and only then sum over all the rows k which are inside the band. As a result, we obtain that

$$\begin{aligned}
\mathbf{1}^T_{\mathcal{T}(\mathbf{\Gamma})}|\mathbf{\Delta}| &\leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \sum_k \sum_{j=k-n+1}^{k+n-1} \|\boldsymbol{\gamma}_{k,0}\|_0 \cdot \|\boldsymbol{\delta}_{j,0}\|_1 \\
&= \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \sum_j \sum_{k=j-n+1}^{j+n-1} \|\boldsymbol{\gamma}_{k,0}\|_0 \cdot \|\boldsymbol{\delta}_{j,0}\|_1 \\
&= \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \sum_j \left(\|\boldsymbol{\delta}_{j,0}\|_1 \sum_{k=j-n+1}^{j+n-1} \|\boldsymbol{\gamma}_{k,0}\|_0 \right).
\end{aligned}$$

Summing over all possible shifts we obtain the ℓ_0-norm of the stripe $\boldsymbol{\gamma}_j$; i.e.,

$$\mathbf{1}^T_{\mathcal{T}(\mathbf{\Gamma})}|\mathbf{\Delta}| \leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1} \sum_j \|\boldsymbol{\delta}_{j,0}\|_1 \cdot \|\boldsymbol{\gamma}_{j}\|_0.$$

Using the definition of $\|\cdot\|_{0,\infty}$

$$\begin{aligned}\mathbf{1}_{\mathcal{T}(\mathbf{\Gamma})}^T|\mathbf{\Delta}| &\leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1}\sum_j \|\boldsymbol{\delta}_{j,0}\|_1 \cdot \|\boldsymbol{\gamma}_j\|_0 \\ &\leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1}\sum_j \|\boldsymbol{\delta}_{j,0}\|_1 \cdot \|\mathbf{\Gamma}\|_{0,\infty} \\ &\leq \frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1}\cdot \|\mathbf{\Delta}\|_1 \cdot \|\mathbf{\Gamma}\|_{0,\infty}. \end{aligned} \tag{5.23}$$

For the set $\mathbf{C}_s^2$ to be non-empty, there must exist a $\mathbf{\Delta}$ which satisfies

$$\begin{aligned}\mathbf{0} \geq& \|\mathbf{\Delta}\|_1 - 2\mathbf{1}_{\mathcal{T}(\mathbf{\Gamma})}^T|\mathbf{\Delta}| \\ \geq& \|\mathbf{\Delta}\|_1 - 2\frac{\mu(\mathbf{D})}{\mu(\mathbf{D})+1}\cdot \|\mathbf{\Delta}\|_1 \cdot \|\mathbf{\Gamma}\|_{0,\infty},\end{aligned}$$

where the first and second inequalities are given in (5.20) and (5.23), respectively. Rearranging the above we obtain $\|\mathbf{\Gamma}\|_{0,\infty} \geq \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$. However, we have assumed that $\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$ and thus the previous inequality is not satisfied. As a result, the set we have defined is empty, implying that BP leads to the desired solution. □

5.10.3 On the Shifted Mutual Coherence and Stripe Coherence

Definition 10: Define the shifted mutual coherence μ_s by

$$\mu_s = \max_{i,j} \quad |\langle \mathbf{d}_i^0, \mathbf{d}_j^s \rangle|,$$

where $\mathbf{d}_i^0$ is a column extracted from $\mathbf{\Omega}_0$, $\mathbf{d}_j^s$ is extracted from $\mathbf{\Omega}_s$, and we require[9] that $i \neq j$ if $s = 0$.

The shifted mutual coherence exhibits some interesting properties:

1. μ_s is symmetric with respect to the shift s, i.e. $\mu_s = \mu_{-s}$.
2. Its maximum over all shifts equals the global mutual coherence of the convolutional dictionary: $\mu(\mathbf{D}) = \max_s \mu_s$.
3. The mutual coherence of the local dictionary is bounded by that of the global one: $\mu(\mathbf{D}_L) = \mu_0 \leq \max_s \mu_s = \mu(\mathbf{D})$.

We now briefly remind the definition of the maximal stripe coherence, as we will make use of it throughout the rest of this section. Given a vector $\mathbf{\Gamma}$, recall that the stripe coherence is defined as $\zeta(\boldsymbol{\gamma}_i) = \sum_{s=-n+1}^{n-1} n_{i,s}\ \mu_s$, where $n_{i,s}$ is the number of non-zeros in the s^{th} shift of $\boldsymbol{\gamma}_i$, taken from $\mathbf{\Gamma}$. The reader might ponder how the maximal stripe coherence might be computed. Let us now define the vector $\mathbf{v}$ which contains in its i^{th} entry the number $n_{i,0}$. Using this definition,

[9] The condition $i \neq j$ if $s = 0$ is necessary so as to avoid the inner product of an atom by itself.

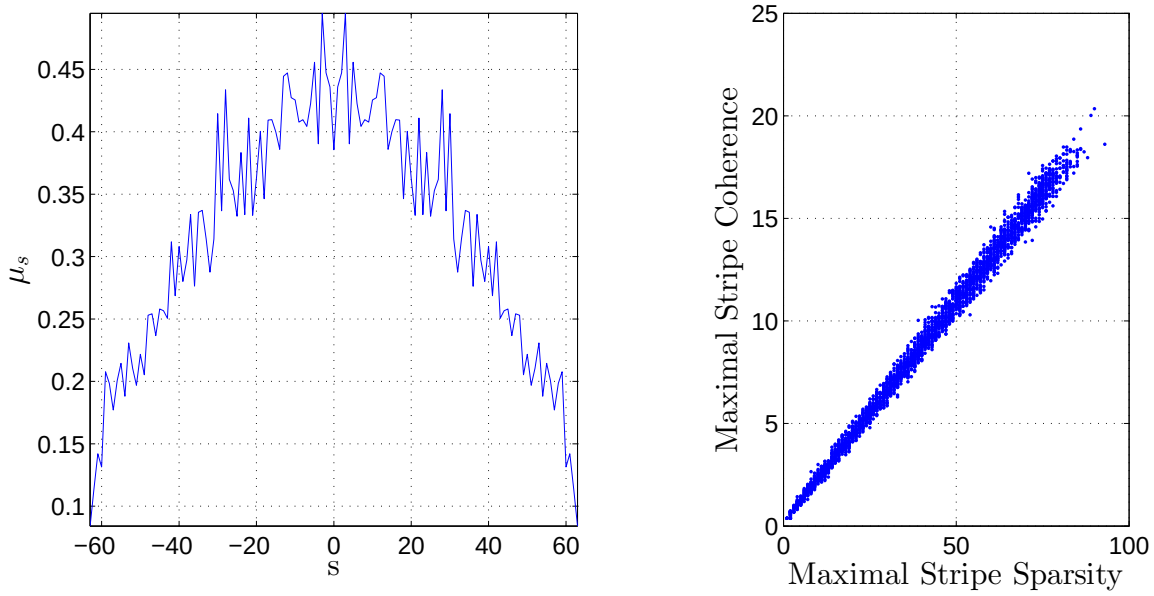


Figure 5.11: Left: the shifted mutual coherence as function of the shift. The larger the shift between the atoms, the lower μ_s is expected to be. Right: the maximal stripe coherence as a function of the $\ell_{0,\infty}$ norm, for random realizations of global sparse vectors.

the coherence of every stripe can be calculated efficiently by convolving the vector $\mathbf{v}$ with the vector of the shifted mutual coherences $[\mu_{-n+1}, \ldots, \mu_{-1}, \mu_0, \mu_1, \ldots, \mu_{n-1}]$.

Next, we provide an experiment in order to illustrate the shifted mutual coherence. To this end, we generate a random local dictionary with $m = 8$ atoms of length $n = 64$ and afterwards normalize its columns. We then construct a convolutional dictionary which contains global atoms of length $N = 640$. We exhibit the shifted mutual coherences for this dictionary in Figure 5.11(a).

Given this dictionary, we generate sparse vectors with random supports of cardinalities in the range $[1, 300]$. For each sparse vector we compute its $\ell_{0,\infty}$ norm by searching for the densest stripe, and its maximal stripe coherence using the convolution mentioned above. In Figure 5.11(b) we illustrate the connection between the $\ell_{0,\infty}$ norm and the maximal stripe coherence for this set of sparse vectors. As expected, the $\ell_{0,\infty}$ norm and the maximal stripe coherence are highly correlated. Although the theorems based on the stripe coherence are sharper, they are harder to comprehend. In this experiment we attempted to alleviate this by showing an intuitive connection between the two.

We now present a theorem relating the stripe coherences of related sparse vectors.

Theorem 21. Let $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$ be two global sparse vectors such that the support of $\mathbf{\Gamma}_1$ is contained in the support of $\mathbf{\Gamma}_2$. Then the maximal stripe coherence of $\mathbf{\Gamma}_1$ is less or equal than that of $\mathbf{\Gamma}_2$.

Proof. Denote by $\boldsymbol{\gamma}_i^1$ and $\boldsymbol{\gamma}_i^2$ the i^{th} stripe extracted from $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$, respectively. Also, denote by $n_{i,s}^1$ and $n_{i,s}^2$ the number of non-zeros in the s^{th} shift of $\boldsymbol{\gamma}_i^1$ and $\boldsymbol{\gamma}_i^2$, respectively. Since the

support of $\mathbf{\Gamma}_1$ is contained in the support of $\mathbf{\Gamma}_2$, we have that $\forall i, s \quad n_{i,s}^1 \leq n_{i,s}^2$. As a result, we have that

$$\max_i \sum_{s=-n+1}^{n-1} n_{i,s}^1 \mu_s \leq \max_i \sum_{s=-n+1}^{n-1} n_{i,s}^2 \mu_s.$$

The left-hand side of the above inequality is the maximal stripe coherence of $\mathbf{\Gamma}_1$, while the right-hand side is the corresponding one for $\mathbf{\Gamma}_2$, proving the hypothesis. □

Theorem 12. (Global OMP recovery guarantee using the stripe coherence): Given the system of linear equations $\mathbf{X} = \mathbf{D\Gamma}$, if a solution $\mathbf{\Gamma}$ exists satisfying

$$\max_i \ \zeta_i = \max_i \sum_{s=-n+1}^{n-1} n_{i,s}\mu_s < \ \frac{1}{2}\left(1 + \mu_0\right),$$

then OMP is guaranteed to recover it.

Proof. The first steps of this proof are exactly those derived in proving Theorem 8, and are thus omitted for the sake brevity. Recall that in order for the first step of OMP to succeed, we require

$$\left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_i\right| > \max_{j\notin\mathcal{T}} \left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_j\right|. \tag{5.24}$$

Lower bounding the left hand side of the above inequality, we can write

$$\left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_i\right| \geq |\Gamma_i| - |\Gamma_i| \sum_{t\in\mathcal{T}_{p(i)}, t\neq i} |\mathbf{d}_t^T \mathbf{d}_i|,$$

as stated previously in Equation (5.15). Instead of summing over the support $\mathcal{T}_{p(i)}$, we can sum over all the supports $\mathcal{T}_{p(i),s}$, which correspond to all possible shifts. We can then write

$$\left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_i\right| \geq |\Gamma_i| - |\Gamma_i| \sum_{s=-n+1}^{n-1} \sum_{\substack{t\in\mathcal{T}_{p(i),s} \\ t\neq i}} |\mathbf{d}_t^T \mathbf{d}_i|.$$

We can bound the right term by using the number of non-zeros in each sub-support $\mathcal{T}_{p(i),s}$, denoted by $n_{p(i),s}$, together with the corresponding shifted mutual coherence μ_s. Also, we can disregard the constraint $t \neq i$ in the above summation by subtracting an extra μ_0 term, obtaining:

$$\left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_i\right| \geq |\Gamma_i| - |\Gamma_i| \left(\sum_{s=-n+1}^{n-1} \mu_s n_{p(i),s} - \mu_0\right).$$

Bounding the above by the maximal stripe coherence, we obtain

$$\left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_i\right| \geq |\Gamma_i| - |\Gamma_i| \left(\max_k \sum_{s=-n+1}^{n-1} \mu_s n_{k,s} - \mu_0\right).$$

In order to upper bound the right hand side of Equation (5.24) we follow the steps leading to Equation (5.16), resulting in

$$\max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j\right| \leq |\Gamma_i|\max_{j\notin\mathcal{T}}\sum_{t\in\mathcal{T}_{p(j)}}|\mathbf{d}_t^T\mathbf{d}_j|.$$

Using a similar decomposition of the support and the definition of the shifted mutual coherence, we have

$$\begin{aligned}\max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j\right| &\leq |\Gamma_i|\max_{j\notin\mathcal{T}}\sum_{s=-n+1}^{n-1}\sum_{t\in\mathcal{T}_{p(j),s}}|\mathbf{d}_t^T\mathbf{d}_j| \\ &\leq |\Gamma_i|\max_{j\notin\mathcal{T}}\sum_{s=-n+1}^{n-1}\mu_s n_{p(j),s}.\end{aligned}$$

Once again bounding this expression by the maximal stripe coherence, we obtain

$$\max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j\right| \leq |\Gamma_i|\cdot\max_{k}\sum_{s=-n+1}^{n-1}\mu_s n_{k,s}.$$

Using both bounds, we have that

$$\begin{aligned}\left|\sum_{t\in\mathcal{T}}\mathbf{\Gamma}_t\mathbf{d}_t^T\mathbf{d}_i\right| &\geq |\Gamma_i| - |\Gamma_i|\left(\max_{k}\sum_{s=-n+1}^{n-1}\mu_s n_{k,s} - \mu_0\right) \\ &> |\Gamma_i|\cdot\max_{k}\sum_{s=-n+1}^{n-1}\mu_s n_{k,s} \\ &\geq \max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j\right|.\end{aligned}$$

Thus,

$$1-\max_{k}\sum_{s=-n+1}^{n-1}\mu_s n_{k,s}+\mu_0 \;>\; \max_{k}\sum_{s=-n+1}^{n-1}\mu_s n_{k,s}.$$

Finally, we obtain

$$\max_{k}\;\zeta_k = \max_{k}\sum_{s=-n+1}^{n-1}\mu_s n_{k,s} \;<\; \frac{1}{2}\left(1+\mu_0\right),$$

which is the requirement stated in the theorem. Thus, this condition guarantees the success of the first OMP step, implying it will choose an atom inside the true support $\mathcal{T}$.

The next step in the OMP algorithm is an update of the residual. This is done by decreasing a term proportional to the chosen atom (or atoms within the correct support in subsequent iterations) from the signal. Thus, the support of this residual is contained within the support of the true signal. As a result, according to the previous theorem, the maximal stripe coherence corresponding to the residual is less or equal to the one of the true sparse code $\mathbf{\Gamma}$. Using the

same set of steps we obtain that the condition on the maximal stripe coherence guarantees that the algorithm chooses again an atom from the true support of the solution. Furthermore, the orthogonality enforced by the least-squares step guarantees that the same atom is never chosen twice. As a result, after $\|\mathbf{\Gamma}\|_0$ iterations the OMP will find all the atoms in the correct support, reaching a residual equal to zero. □

Theorem 13. (Global BP recovery guarantee using the stripe coherence): Given the system of linear equations $\mathbf{X} = \mathbf{D\Gamma}$, if a solution $\mathbf{\Gamma}$ exists satisfying

$$\max_i \ \zeta_i = \max_i \sum_{s=-n+1}^{n-1} n_{i,s}\mu_s < \ \frac{1}{2}\left(1+\mu_0\right),$$

then Basis Pursuit is guaranteed to recover it.

The proof of this claim is similar to that of Theorem 9, while using ideas from Theorem 12. In particular, one should repeat the steps leading to Equation (5.18). Then, using the definition of the shifted mutual coherence, one can have

$$|\Delta_i| \leq \sum_{s=-n+1}^{n-1} \mu_s \|\boldsymbol{\delta}_{p(i),s}\|_1 - \mu_0|\Delta_i|.$$

Summing over the support of $\boldsymbol{\gamma}_{p(i),0}$

$$\mathbf{1}^T_{\mathcal{T}(\boldsymbol{\gamma}_{p(i),0})}|\boldsymbol{\Delta}| \leq \sum_{s=-n+1}^{n-1} \frac{\mu_s}{\mu_0+1} \ \|\boldsymbol{\delta}_{p(i),s}\|_1 \ \|\boldsymbol{\gamma}_{p(i),0}\|_0.$$

Summing over all different $p(i)$ we obtain

$$\mathbf{1}^T_{\mathcal{T}(\mathbf{\Gamma})}|\boldsymbol{\Delta}| \leq \sum_k \sum_{s=-n+1}^{n-1} \frac{\mu_s}{\mu_0+1} \ \|\boldsymbol{\delta}_{k,s}\|_1 \ \|\boldsymbol{\gamma}_{k,0}\|_0.$$

Changing the order of summation, just as done in proving Theorem 9, we get

$$\mathbf{1}^T_{\mathcal{T}(\mathbf{\Gamma})}|\boldsymbol{\Delta}| \leq \sum_k \sum_{s=-n+1}^{n-1} \frac{\mu_s}{\mu_0+1} \ \|\boldsymbol{\delta}_{k,0}\|_1 \ \|\boldsymbol{\gamma}_{k,s}\|_0.$$

Rearranging the above,

$$\|\boldsymbol{\Delta}\|_1 \leq \sum_k \frac{\|\boldsymbol{\delta}_{k,0}\|_1}{\mu_0+1} \sum_{s=-n+1}^{n-1} \mu_s \|\boldsymbol{\gamma}_{k,s}\|_0,$$

or equally,

$$\|\boldsymbol{\Delta}\|_1 \leq \frac{\|\boldsymbol{\Delta}\|_1}{\mu_0+1} \sum_{s=-n+1}^{n-1} \mu_s \|\boldsymbol{\gamma}_{k,s}\|_0.$$

Using the definition of the stripe coherence and recalling that $n_{i,s} = \|\boldsymbol{\gamma}_{k,s}\|_0$, one gets

$$\|\boldsymbol{\Delta}\|_1 \leq \frac{\|\boldsymbol{\Delta}\|_1}{\mu_0 + 1} \max_i \sum_{s=-n+1}^{n-1} n_{i,s}\mu_s.$$

From the above equation, the rest of the proof follows naturally.

We have provided two theorems for the success of the OMP algorithm. Before concluding, we aim to show that assuming $\mu(\mathbf{D}) = \mu_0$, the guarantee based on the stripe coherence is at least as strong as the one based on the $\ell_{0,\infty}$ norm. Assume the recovery condition using the $\ell_{0,\infty}$ norm is met and as such $\|\boldsymbol{\Gamma}\|_{0,\infty} = \max_i \ n_i < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$, where n_i is equal to $\|\boldsymbol{\gamma}_i\|_0$. Multiplying both sides by $\mu(\mathbf{D})$ we obtain $\max_i \ n_i \cdot \mu(\mathbf{D}) < \frac{1}{2}(1 + \mu(\mathbf{D}))$. Using the above inequality and the properties:

$$1) \sum_{s=-n+1}^{n-1} n_{i,s} = n_i, \qquad 2) \ \forall s \quad \mu_s \leq \mu(\mathbf{D}),$$

we have that

$$\begin{aligned}\max_i \sum_{s=-n+1}^{n-1} n_{i,s}\mu_s &\leq \max_i \sum_{s=-n+1}^{n-1} n_{i,s}\mu(\mathbf{D}) \\ &= \max_i \ n_i \cdot \mu(\mathbf{D}) < \frac{1}{2}(1 + \mu(\mathbf{D}))\,.\end{aligned}$$

Thus, we obtain that

$$\max_i \sum_{s=-n+1}^{n-1} n_{i,s}\mu_s < \frac{1}{2}(1 + \mu(\mathbf{D})) = \frac{1}{2}(1 + \mu_0)\,,$$

where we have used our assumption that $\mu(\mathbf{D}) = \mu_0$. We conclude that if the recovery condition based on the $\ell_{0,\infty}$ norm is met, then so is the one based on the stripe coherence. As a result, the condition based on the stripe coherence is at least as strong as the one based on the $\ell_{0,\infty}$ norm.

As a final note, we mention that assuming $\mu(\mathbf{D}) = \max_s \mu_s = \mu_0$ is in fact a reasonable assumption. Recall that in order to compute μ_s we evaluate inner products between atoms which are s indexes shifted from each other. As a result, the higher the shift s is, the less overlap the atoms have, and the less μ_s is expected to be. Thus, we expect the value μ_0 to be the largest or close to it in most cases.

5.10.4 Theoretical Analysis of Corrupted Signals

Theorem 15. (Upper bounding the SRIP via the mutual coherence): For a convolutional dictionary $\mathbf{D}$ with global mutual coherence $\mu(\mathbf{D})$, the SRIP can be upper-bounded by

$$\delta_k \leq (k-1)\mu(\mathbf{D}).$$

Proof. Consider the sub-dictionary $\mathbf{D}_{\mathcal{T}}$, obtained by restricting the columns of $\mathbf{D}$ to a support $\mathcal{T}$ with $\ell_{0,\infty}$ norm equal to k. Lemma 1 states that the eigenvalues of the Gram matrix $\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}$ are bounded by

$$1-(k-1)\mu(\mathbf{D}) \leq \lambda_i(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}) \leq 1+(k-1)\mu(\mathbf{D}).$$

Now, for every $\boldsymbol{\Delta}$ we have that

$$\begin{aligned}(1-(k-1)\mu(\mathbf{D}))\|\boldsymbol{\Delta}\|_2^2 \leq & \lambda_{min}(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}})\|\boldsymbol{\Delta}\|_2^2 \\ \leq & \|\mathbf{D}_{\mathcal{T}}\boldsymbol{\Delta}\|_2^2 \leq \lambda_{max}(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}})\|\boldsymbol{\Delta}\|_2^2 \\ \leq & (1+(k-1)\mu(\mathbf{D}))\|\boldsymbol{\Delta}\|_2^2,\end{aligned}$$

where λ_{max} and λ_{min} are the maximal and minimal eigenvalues, respectively. As a result, we obtain that $\delta_k \leq (k-1)\mu(\mathbf{D})$. □

Theorem 16. (Stability of the solution to the $P_{0,\infty}^{\epsilon}$ problem): Consider a sparse vector $\boldsymbol{\Gamma}$ such that $\|\boldsymbol{\Gamma}\|_{0,\infty} = k < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$, and a convolutional dictionary $\mathbf{D}$ satisfying the SRIP property for $\ell_{0,\infty} = 2k$ with coefficient δ_{2k}. Then, the distance between the true sparse vector $\boldsymbol{\Gamma}$ and the solution to the $P_{0,\infty}^{\epsilon}$ problem $\hat{\boldsymbol{\Gamma}}$ is bounded by

$$\|\boldsymbol{\Gamma}-\hat{\boldsymbol{\Gamma}}\|_2^2 \leq \frac{4\epsilon^2}{1-\delta_{2k}} \leq \frac{4\epsilon^2}{1-(2k-1)\mu(\mathbf{D})}. \tag{5.25}$$

Proof. The solution to the $P_{0,\infty}^{\epsilon}$ problem satisfies $\|\mathbf{Y}-\mathbf{D}\hat{\boldsymbol{\Gamma}}\|_2^2 \leq \epsilon^2$, and it must also satisfy $\|\hat{\boldsymbol{\Gamma}}\|_{0,\infty} \leq \|\boldsymbol{\Gamma}\|_{0,\infty}$ (since $\hat{\boldsymbol{\Gamma}}$ is the solution with the minimal $\ell_{0,\infty}$ norm). Defining $\boldsymbol{\Delta} = \boldsymbol{\Gamma}-\hat{\boldsymbol{\Gamma}}$, using the triangle inequality, we have that $\|\mathbf{D}\boldsymbol{\Delta}\|_2^2 = \|\mathbf{D}\boldsymbol{\Gamma}-\mathbf{Y}+\mathbf{Y}-\mathbf{D}\hat{\boldsymbol{\Gamma}}\|_2^2 \leq 4\epsilon^2$. Furthermore, since the $\ell_{0,\infty}$ norm satisfies the triangle inequality as well, we have that $\|\boldsymbol{\Delta}\|_{0,\infty} = \|\boldsymbol{\Gamma}-\hat{\boldsymbol{\Gamma}}\|_{0,\infty} \leq \|\boldsymbol{\Gamma}\|_{0,\infty} + \|\hat{\boldsymbol{\Gamma}}\|_{0,\infty} \leq 2k$. Using the SRIP of $\mathbf{D}$, we have that

$$(1-\delta_{2k})\|\boldsymbol{\Delta}\|_2^2 \leq \|\mathbf{D}\boldsymbol{\Delta}\|_2^2 \leq 4\epsilon^2,$$

where in the first inequality we have used the lower bound provided by the definition of the SRIP. Finally, we obtain the following stability claim:

$$\|\boldsymbol{\Delta}\|_2^2 = \|\boldsymbol{\Gamma}-\hat{\boldsymbol{\Gamma}}\|_2^2 \leq \frac{4\epsilon^2}{1-\delta_{2k}}.$$

Using our bound of the SRIP in terms of the mutual coherence, we obtain that

$$\|\boldsymbol{\Delta}\|_2^2 = \|\boldsymbol{\Gamma}-\hat{\boldsymbol{\Gamma}}\|_2^2 \leq \frac{4\epsilon^2}{1-\delta_{2k}} \leq \frac{4\epsilon^2}{1-(2k-1)\mu(\mathbf{D})}.$$

For the last inequality to hold, we have assumed $k = \|\boldsymbol{\Gamma}\|_{0,\infty} < \frac{1}{2}(1+\frac{1}{\mu(\mathbf{D})})$.

Theorem 17. (Stable recovery of global OMP in the presence of noise): Suppose a clean signal $\mathbf{X}$ has a representation $\mathbf{D}\boldsymbol{\Gamma}$, and that it is contaminated with noise $\mathbf{E}$ to create the signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$, such that $\|\mathbf{Y} - \mathbf{X}\|_2 \leq \epsilon$. Denote by ϵ_L the highest energy of all n-dimensional local patches extracted from $\mathbf{E}$. Assume $\boldsymbol{\Gamma}$ satisfies

$$\|\boldsymbol{\Gamma}\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) - \frac{1}{\mu(\mathbf{D})} \cdot \frac{\epsilon_L}{|\Gamma_{min}|}, \tag{5.26}$$

where $|\Gamma_{min}|$ is the minimal entry in absolute value of the sparse vector $\boldsymbol{\Gamma}$. Denoting by $\boldsymbol{\Gamma}_{\text{OMP}}$ the solution obtained by running OMP for $\|\boldsymbol{\Gamma}\|_0$ iterations, we are guaranteed that

[a)]

1. OMP will find the correct support; And,
2. $\|\boldsymbol{\Gamma}_{\text{OMP}} - \boldsymbol{\Gamma}\|_2^2 \leq \frac{\epsilon^2}{1-\mu(\|\boldsymbol{\Gamma}\|_{0,\infty}-1)}$.

Proof. We shall first prove that the first step of OMP succeeds in recovering an element from the correct support. Denoting by $\mathcal{T}$ the support of $\boldsymbol{\Gamma}$, we can write

$$\mathbf{Y} = \mathbf{D}\boldsymbol{\Gamma} + \mathbf{E} = \sum_{t \in \mathcal{T}} \Gamma_t \mathbf{d}_t + \mathbf{E}. \tag{5.27}$$

Suppose that $\boldsymbol{\Gamma}$ has its largest coefficient in absolute value in Γ_i. For the first step of OMP to choose one of the atoms in the support, we require

$$|\mathbf{d}_i^T \mathbf{Y}| > \max_{j \notin \mathcal{T}} |\mathbf{d}_j^T \mathbf{Y}|.$$

Substituting Equation (5.27) in this requirement we obtain

$$\left|\sum_{t \in \mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_i + \mathbf{E}^T \mathbf{d}_i\right| > \max_{j \notin \mathcal{T}} \left|\sum_{t \in \mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_j + \mathbf{E}^T \mathbf{d}_j\right|. \tag{5.28}$$

Using the reverse triangle inequality we can construct a lower bound for the left hand side:

$$\left|\sum_{t \in \mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_i + \mathbf{E}^T \mathbf{d}_i\right| \geq \left|\sum_{t \in \mathcal{T}} \Gamma_t \mathbf{d}_t^T \mathbf{d}_i\right| - \left|\mathbf{E}^T \mathbf{d}_i\right|.$$

Our next step is to bound the absolute value of the inner product of the noise and the atom $\mathbf{d}_i$. A naïve approach, based on the Cauchy-Schwarz inequality and the normalization of the atoms, would be to bound the inner product as $|\mathbf{E}^T \mathbf{d}_i| \leq \|\mathbf{E}\|_2 \cdot \|\mathbf{d}_i\|_2 \leq \epsilon$. However, such bound would disregard the local nature of the atoms. Due to their limited support we have that $\mathbf{d}_i = \mathbf{R}_i^T \mathbf{R}_i \mathbf{d}_i$ where, as previously defined, $\mathbf{R}_i$ extracts a n-dimensional patch from a N-dimensional signal. Based on this observation, we have that

$$|\mathbf{E}^T \mathbf{d}_i| = |\mathbf{E}^T \mathbf{R}_i^T \mathbf{R}_i \mathbf{d}_i| \leq \|\mathbf{R}_i \mathbf{E}\|_2 \cdot \|\mathbf{d}_i\|_2 \leq \epsilon_L,$$

where we have used the fact that $\|\mathbf{R}_i\mathbf{E}\|_2 \leq \epsilon_{_L}\ \forall\ i$. By exploiting the locality of the atom, together with the assumption regarding the maximal local energy of the noise, we are able to obtain a much tighter bound, because $\epsilon_{_L} \ll \epsilon$ in general. As a result, we obtain

$$\left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T\mathbf{d}_i + \mathbf{E}^T\mathbf{d}_i\right| \geq \left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T\mathbf{d}_i\right| - \epsilon_{_L}.$$

Using the reverse triangle inequality, the normalization of the atoms and the fact that $|\Gamma_i| \geq |\Gamma_t|$, we obtain

$$\begin{aligned}\left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T\mathbf{d}_i + \mathbf{E}^T\mathbf{d}_i\right| &\geq |\Gamma_i| - \sum_{t\in\mathcal{T}, t\neq i} |\Gamma_t|\cdot|\mathbf{d}_t^T\mathbf{d}_i| - \epsilon_{_L} \\ &\geq |\Gamma_i| - |\Gamma_i| \sum_{t\in\mathcal{T}, t\neq i} |\mathbf{d}_t^T\mathbf{d}_i| - \epsilon_{_L}.\end{aligned}$$

Notice that $\mathbf{d}_t^T\mathbf{d}_i$ is zero for every atom too far from $\mathbf{d}_i$ because the atoms do not overlap. Denoting the stripe which fully contains the i^{th} atom as $p(i)$ and its support as $\mathcal{T}_{p(i)}$, we can restrict the summation as:

$$\left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T\mathbf{d}_i + \mathbf{E}^T\mathbf{d}_i\right| \geq |\Gamma_i| - |\Gamma_i| \sum_{\substack{t\in\mathcal{T}_{p(i)},\\ t\neq i}} |\mathbf{d}_t^T\mathbf{d}_i| - \epsilon_{_L}.$$

Denoting by $n_{p(i)}$ the number of non-zeros in the support $\mathcal{T}_{p(i)}$ and using the definition of the mutual coherence we obtain:

$$\begin{aligned}\left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T\mathbf{d}_i + \mathbf{E}^T\mathbf{d}_i\right| &\geq |\Gamma_i| - |\Gamma_i|(n_{p(i)}-1)\mu(\mathbf{D}) - \epsilon_{_L} \\ &\geq |\Gamma_i| - |\Gamma_i|(\|\mathbf{\Gamma}\|_{0,\infty}-1)\mu(\mathbf{D}) - \epsilon_{_L}.\end{aligned}$$

In the last inequality we have used the definition of the $\ell_{0,\infty}$ norm.

Now, we construct an upper bound for the right hand side of equation (5.28), once again using the triangle inequality and the fact that $|\mathbf{E}^T\mathbf{d}_j| \leq \epsilon_{_L}$:

$$\max_{j\notin\mathcal{T}} \left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T\mathbf{d}_j + \mathbf{E}^T\mathbf{d}_j\right| \leq \max_{j\notin\mathcal{T}} \left|\sum_{t\in\mathcal{T}} \Gamma_t \mathbf{d}_t^T\mathbf{d}_j\right| + \epsilon_{_L}.$$

Using the same rationale as before we get

$$\begin{aligned}\max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j+\mathbf{E}^T\mathbf{d}_j\right| &\le |\Gamma_i|\max_{j\notin\mathcal{T}}\sum_{t\in\mathcal{T}}|\mathbf{d}_t^T\mathbf{d}_j|+\epsilon_L\\ &\le |\Gamma_i|\max_{j\notin\mathcal{T}}\sum_{t\in\mathcal{T}_{p(j)}}|\mathbf{d}_t^T\mathbf{d}_j|+\epsilon_L\\ &\le |\Gamma_i|\cdot\|\mathbf{\Gamma}\|_{0,\infty}\cdot\mu(\mathbf{D})+\epsilon_L.\end{aligned}$$

Using both bounds, we obtain

$$\begin{aligned}&\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_i+\mathbf{E}^T\mathbf{d}_i\right|\ge|\Gamma_i|-|\Gamma_i|(\|\mathbf{\Gamma}\|_{0,\infty}-1)\mu(\mathbf{D})-\epsilon_L\\ \ge&|\Gamma_i|\cdot\|\mathbf{\Gamma}\|_{0,\infty}\mu(\mathbf{D})+\epsilon_L\ge\max_{j\notin\mathcal{T}}\left|\sum_{t\in\mathcal{T}}\Gamma_t\mathbf{d}_t^T\mathbf{d}_j+\mathbf{E}^T\mathbf{d}_j\right|.\end{aligned}$$

From this, it follows that

$$\|\mathbf{\Gamma}\|_{0,\infty}\le\frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)-\frac{1}{\mu(\mathbf{D})}\cdot\frac{\epsilon_L}{|\Gamma_i|}. \tag{5.29}$$

Note that the theorem's hypothesis assumes that the above holds for $|\Gamma_{\min}|$ instead of $|\Gamma_i|$. However, because $|\Gamma_i|\ge|\Gamma_{min}|$, this condition holds for every i. Therefore, Equation (5.29) holds and we conclude that the first step of OMP succeeds.

Next, we address the success of subsequent iterations of the OMP. Define the sparse vector obtained after $k<\|\mathbf{\Gamma}\|_0$ iterations as $\mathbf{\Lambda}^k$, and denote its support by $\mathcal{T}^k$. Assuming that the algorithm identified correct atoms (i.e., has so far succeeded), $\mathcal{T}^k=\text{supp}\{\mathbf{\Lambda}_k\}\subset\text{supp}\{\mathbf{\Gamma}\}$. The next step in the algorithm is the update of the residual. This is done by decreasing a term proportional to the chosen atoms from the signal; i.e.,

$$\mathbf{Y}^k=\mathbf{Y}-\sum_{i\in\mathcal{T}^k}\mathbf{d}_i\Lambda_i^k.$$

Moreover, $\mathbf{Y}^k$ can be seen as containing a clean signal $\mathbf{X}^k$ and the noise component $\mathbf{E}$, where

$$\mathbf{X}^k=\mathbf{X}-\sum_{i\in\mathcal{T}^k}\mathbf{d}_i\Lambda_i^k=\mathbf{D}\mathbf{\Gamma}^k.$$

The objective is then to recover the support of the sparse vector corresponding to $\mathbf{X}^k$, $\mathbf{\Gamma}^k$, defined as[10]

$$\Gamma_i^k=\begin{cases}\Gamma_i-\Lambda_i^k & \text{if } i\in\mathcal{T}^k\\ \Gamma_i & \text{if } i\notin\mathcal{T}^k.\end{cases} \tag{5.30}$$

[10]Note that if $k=0$, $\mathbf{X}_0=\mathbf{X}$, $\mathbf{Y}_0=\mathbf{Y}$, and $\mathbf{\Gamma}_0=\mathbf{\Gamma}$.

Note that $\text{supp}\{\mathbf{\Gamma}^k\} \subseteq \text{supp}\{\mathbf{\Gamma}\}$ and so

$$\|\mathbf{\Gamma}^k\|_{0,\infty} \leq \|\mathbf{\Gamma}\|_{0,\infty}. \tag{5.31}$$

In words, the $\ell_{0,\infty}$ norm of the underlying solution of $\mathbf{X}^k$ does not increase as the iterations proceed. Note that this representation is also unique in light of the uniqueness theorem presented in part I. From the above definitions, we have that

$$\begin{aligned}\mathbf{Y}^k - \mathbf{X}^k &= \mathbf{Y} - \sum_{i\in\mathcal{T}^k} \mathbf{d}_i \Lambda_i^k - \mathbf{X} + \sum_{i\in\mathcal{T}^k} \mathbf{d}_i \mathbf{\Lambda}_i^k \\ &= \mathbf{Y} - \mathbf{X} = \mathbf{E}.\end{aligned}$$

Hence, the noise level is preserved, both locally and globally; both ϵ and ϵ_L remain the same.

Note that $\mathbf{\Gamma}^k$ differs from $\mathbf{\Gamma}$ in at most k places, following Equation (5.30) and that $|\mathcal{T}^k| = k$. As such, $\|\mathbf{\Gamma}^k\|_\infty$ is greater or equal than the $(k+1)^{th}$ largest element in absolute value in $\mathbf{\Gamma}$. This implies that $\|\mathbf{\Gamma}^k\|_\infty \geq |\Gamma_{\min}|$. Finally, we obtain that

$$\begin{aligned}\|\mathbf{\Gamma}^k\|_{0,\infty} \leq \|\mathbf{\Gamma}\|_{0,\infty} &< \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) - \frac{1}{\mu(\mathbf{D})} \cdot \frac{\epsilon_L}{|\Gamma_{min}|} \\ &\leq \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) - \frac{1}{\mu(\mathbf{D})} \cdot \frac{\epsilon_L}{\|\mathbf{\Gamma}^k\|_\infty}.\end{aligned}$$

The first inequality is due to (5.31), the second is the assumption in (6.10) and the third was just obtained above. Thus,

$$\|\mathbf{\Gamma}^k\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) - \frac{1}{\mu(\mathbf{D})} \cdot \frac{\epsilon_L}{\|\mathbf{\Gamma}^k\|_\infty}.$$

Similar to the first iteration, the above inequality together with the fact that the noise level is preserved, guarantees the success of the next iteration of the OMP algorithm. From this follows that the algorithm is guaranteed to recover the true support after $\|\mathbf{\Gamma}\|_0$ iterations.

Finally, we move to prove the second claim. In its last iteration OMP solves the following problem:

$$\mathbf{\Gamma}_{OMP} = \arg\min_{\mathbf{\Delta}} \|\mathbf{D}_\mathcal{T}\mathbf{\Delta} - \mathbf{Y}\|_2^2,$$

where $\mathbf{D}_\mathcal{T}$ is the convolutional dictionary restricted to the support $\mathcal{T}$ of the true sparse code $\mathbf{\Gamma}$. Denoting $\mathbf{\Gamma}_\mathcal{T}$ the (dense) vector corresponding to those atoms, the solution to the above problem is simply given by

$$\begin{aligned}\mathbf{\Gamma}_{OMP} &= \mathbf{D}_\mathcal{T}^\dagger \mathbf{Y} = \mathbf{D}_\mathcal{T}^\dagger (\mathbf{D}\mathbf{\Gamma} + \mathbf{E}) \\ &= \mathbf{D}_\mathcal{T}^\dagger (\mathbf{D}_\mathcal{T}\mathbf{\Gamma}_\mathcal{T} + \mathbf{E}) = \mathbf{\Gamma}_\mathcal{T} + \mathbf{D}_\mathcal{T}^\dagger \mathbf{E},\end{aligned}$$

where we have denoted by $\mathbf{D}_{\mathcal{T}}^{\dagger}$ the Moore-Penrose pseudoinverse of the sub-dictionary $\mathbf{D}_{\mathcal{T}}$. Thus,

$$\begin{aligned}\|\mathbf{\Gamma}_{OMP}-\mathbf{\Gamma}_{\mathcal{T}}\|_2^2 &= \|\mathbf{D}_{\mathcal{T}}^{\dagger}\mathbf{E}\|_2^2 \le \|\mathbf{D}_{\mathcal{T}}^{\dagger}\|_2^2\cdot\|\mathbf{E}\|_2^2 \\ &= \frac{1}{\lambda_{\min}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)}\|\mathbf{E}\|_2^2 \le \frac{\epsilon^2}{1-\mu(\mathbf{D})(\|\mathbf{\Gamma}\|_{0,\infty}-1)}.\end{aligned}$$

In the last inequality we have used the bound on the eigenvalues of $\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}$ derived in Lemma 1.

Theorem 18. (ERC in the convolutional sparse model): For a convolutional dictionary $\mathbf{D}$ with mutual coherence $\mu(\mathbf{D})$, the ERC condition is met for every support $\mathcal{T}$ that satisfies

$$\|\mathcal{T}\|_{0,\infty} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right).$$

Proof. For the ERC to be satisfied, we must require that, for every $i\notin\mathcal{T}$,

$$\|\mathbf{D}_{\mathcal{T}}^{\dagger}\mathbf{d}_i\|_1 = \left\|\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{d}_i\right\|_1 < 1.$$

Using properties of induced norms, we have that

$$\left\|\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{d}_i\right\|_1 \le \left\|\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\right\|_1 \left\|\mathbf{D}_{\mathcal{T}}^T\mathbf{d}_i\right\|_1. \tag{5.32}$$

Using the definition of the mutual coherence, it is easy to see that the absolute value of the entries in the vector $\mathbf{D}_{\mathcal{T}}^T\mathbf{d}_i$ are bounded by $\mu(\mathbf{D})$. Moreover, due to the locality of the atoms, the number of non-zero inner products with the atom $\mathbf{d}_i$ is equal to the number of atoms in $\mathcal{T}$ that overlap with it. This number can, in turn, be bounded by the maximal number of non-zeros in a stripe from $\mathcal{T}$, i.e., its $\ell_{0,\infty}$ norm, denoted by k. Therefore, $\left\|\mathbf{D}_{\mathcal{T}}^T\mathbf{d}_i\right\|_1 \le k\mu(\mathbf{D})$.

Addressing now the first term in Equation (5.32), note that

$$\left\|\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\right\|_1 = \left\|\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\right\|_\infty, \tag{5.33}$$

since the induced ℓ_1 and ℓ_∞ norms are equal for symmetric matrices. Next, using the Ahlberg-Nilson-Varah bound and similar steps to those presented in Lemma 1, we have that

$$\left\|\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\right\|_\infty \le \frac{1}{1-(k-1)\mu(\mathbf{D})}. \tag{5.34}$$

In order for this to hold, we must require the Gram $\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}$ to be diagonally dominant, which is satisfied if $1-(k-1)\mu(\mathbf{D})>0$. This is indeed the case, as follows from the assumption on the $\ell_{0,\infty}$ norm of $\mathcal{T}$. Plugging the above into Equation (5.32), we obtain

$$\begin{aligned}\left\|\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{d}_i\right\|_1 &\le \left\|\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\right\|_1\left\|\mathbf{D}_{\mathcal{T}}^T\mathbf{d}_i\right\|_1 \\ &\le \frac{k\mu(\mathbf{D})}{1-(k-1)\mu(\mathbf{D})}.\end{aligned} \tag{5.35}$$

Our assumption that $k < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$ implies that the above term is less than one, thus showing the ERC is satisfied for all supports $\mathcal{T}$ that satisfy $\|\mathcal{T}\|_{0,\infty} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$. □

Theorem 19. (Stable recovery of global Basis Pursuit in the presence of noise): Suppose a clean signal $\mathbf{X}$ has a representation $\mathbf{D\Gamma}$, and that it is contaminated with noise $\mathbf{E}$ to create the signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. Denote by ϵ_L the highest energy of all n-dimensional local patches extracted from $\mathbf{E}$. Assume $\mathbf{\Gamma}$ satisfies

$$\|\mathbf{\Gamma}\|_{0,\infty} \leq \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D})}\right). \tag{5.36}$$

Denoting by $\mathbf{\Gamma}_{\mathrm{BP}}$ the solution to the Lagrangian BP formulation with parameter $\lambda = 4\epsilon_L$, we are guaranteed that

1. The support of $\mathbf{\Gamma}_{\mathrm{BP}}$ is contained in that of $\mathbf{\Gamma}$.
2. $\|\mathbf{\Gamma}_{\mathrm{BP}} - \mathbf{\Gamma}\|_\infty < \frac{15}{2}\epsilon_L$.
3. In particular, the support of $\mathbf{\Gamma}_{\mathrm{BP}}$ contains every index i for which $|\Gamma_i| > \frac{15}{2}\epsilon_L$.
4. The minimizer of the problem, $\mathbf{\Gamma}_{\mathrm{BP}}$, is unique.

We first state and prove a Lemma that will become of use while proving the stability result of BP.

Lemma 5.10.1. *Suppose a clean signal $\mathbf{X}$ has a representation $\mathbf{D\Gamma}$, and that it is contaminated with noise $\mathbf{E}$ to create the signal $\mathbf{Y} = \mathbf{X}+\mathbf{E}$. Denote by ϵ_L the highest energy of all n-dimensional local patches extracted from $\mathbf{E}$. Assume that*

$$\|\mathbf{\Gamma}\|_{0,\infty} \leq \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right). \tag{5.37}$$

Denoting by $\mathbf{X}_{LS}$ the best ℓ_2 approximation of $\mathbf{Y}$ over the support $\mathcal{T}$, we have that[11]

$$\|\mathbf{D}^T(\mathbf{Y} - \mathbf{X}_{LS})\|_\infty \leq 2\epsilon_L.$$

Proof. Using the expression for the least squares solution (and assuming that $\mathbf{D}_\mathcal{T}$ has full-column rank), we have that

$$\begin{aligned}\mathbf{D}_\mathcal{T}^T(\mathbf{Y} - \mathbf{X}_{\mathrm{LS}}) &= \mathbf{D}_\mathcal{T}^T\left(\mathbf{Y} - \mathbf{D}_\mathcal{T}\left(\mathbf{D}_\mathcal{T}^T\mathbf{D}_\mathcal{T}\right)^{-1}\mathbf{D}_\mathcal{T}^T\mathbf{Y}\right)\\ &= \left(\mathbf{D}_\mathcal{T}^T - \mathbf{D}_\mathcal{T}^T\mathbf{D}_\mathcal{T}\left(\mathbf{D}_\mathcal{T}^T\mathbf{D}_\mathcal{T}\right)^{-1}\mathbf{D}_\mathcal{T}^T\right)\mathbf{Y} = \mathbf{0}.\end{aligned}$$

This shows that all inner products between atoms inside $\mathcal{T}$ and the vector $\mathbf{Y} - \mathbf{X}_{\mathrm{LS}}$ are zero, and thus $\|\mathbf{D}_{\overline{\mathcal{T}}}^T(\mathbf{Y} - \mathbf{X}_{\mathrm{LS}})\|_\infty = \|\mathbf{D}^T(\mathbf{Y} - \mathbf{X}_{\mathrm{LS}})\|_\infty$. We have denoted by $\overline{\mathcal{T}}$ the complement to

[11] We suspect that, perhaps under further assumptions, the constant in this bound can be improved from 2 to 1. This is motivated by the fact that the bound in [Tro06], for the traditional sparse model, is $1 \cdot \epsilon$ – where ϵ is the global noise level.

the support, containing all atoms not found in $\mathcal{T}$, and by $\mathbf{D}_{\overline{\mathcal{T}}}$ the corresponding dictionary. Denoting by $\mathbf{\Gamma}_{\mathcal{T}}$ the vector $\mathbf{\Gamma}$ restricted to its support, and expressing $\mathbf{X}_{\text{LS}}$ and $\mathbf{Y}$ conveniently, we obtain

$$\begin{aligned}
&\|\mathbf{D}_{\overline{\mathcal{T}}}^T(\mathbf{Y}-\mathbf{X}_{\text{LS}})\|_\infty \\
=&\|\mathbf{D}_{\overline{\mathcal{T}}}^T\left(\mathbf{I}-\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\right)\mathbf{Y}\|_\infty \\
=&\|\mathbf{D}_{\overline{\mathcal{T}}}^T\left(\mathbf{I}-\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\right)(\mathbf{D}_{\mathcal{T}}\mathbf{\Gamma}_{\mathcal{T}}+\mathbf{E})\|_\infty.
\end{aligned}$$

It is easy to verify that

$$\left(\mathbf{I}-\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\right)\mathbf{D}_{\mathcal{T}}\mathbf{\Gamma}_{\mathcal{T}}=\mathbf{0}.$$

Plugging this into the above, we have that

$$\|\mathbf{D}_{\overline{\mathcal{T}}}^T(\mathbf{Y}-\mathbf{X}_{\text{LS}})\|_\infty=\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\left(\mathbf{I}-\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\right)\mathbf{E}\right\|_\infty.$$

Using the triangle inequality for the ℓ_∞ norm, we obtain

$$\begin{aligned}
&\|\mathbf{D}_{\overline{\mathcal{T}}}^T(\mathbf{Y}-\mathbf{X}_{\text{LS}})\|_\infty \\
=&\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{E}-\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{E}\right\|_\infty \\
\leq&\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{E}\right\|_\infty+\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{E}\right\|_\infty.
\end{aligned}\tag{5.38}$$

In what follows, we will bound both terms in the above expression with ϵ_L. First, due to the limited support of the atoms, $\mathbf{d}_i=\mathbf{R}_i^T\mathbf{R}_i\mathbf{d}_i$, where $\mathbf{R}_i$ extracts the i^{th} local patch from the global signal, as previously defined. Thus,

$$\begin{aligned}
\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{E}\right\|_\infty=\max_{i\in\overline{\mathcal{T}}}|\mathbf{d}_i^T\mathbf{E}|&=\max_{i\in\overline{\mathcal{T}}}|\mathbf{d}_i^T\mathbf{R}_i^T\mathbf{R}_i\mathbf{E}| \\
&\leq\max_{i\in\overline{\mathcal{T}}}\|\mathbf{R}_i\mathbf{d}_i\|_2\cdot\|\mathbf{R}_i\mathbf{E}\|_2\leq\epsilon_L,
\end{aligned}\tag{5.39}$$

where we have used the Cauchy-Schwarz inequality, the normalization of the atoms and the fact that $\|\mathbf{R}_i\mathbf{E}\|_2\leq\epsilon_L\ \forall\ i$. Next, we move to the second term in Equation (5.38). Using the definition of the induced ℓ_∞ norm, and the bound $\|\mathbf{D}_{\mathcal{T}}^T\mathbf{E}\|_\infty\leq\epsilon_L$, we have that

$$\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{E}\right\|_\infty\leq\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\right\|_\infty\epsilon_L.$$

Recall that the induced infinity norm of a matrix is equal to the maximal ℓ_1 norm of its rows. Notice that a row in the above matrix can be written as $\mathbf{d}_i^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}$, where $i\in\overline{\mathcal{T}}$. Then,

$$\begin{aligned}
&\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{E}\right\|_\infty \\
\leq&\max_{i\in\overline{\mathcal{T}}}\left\|\mathbf{d}_i^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\right\|_1\cdot\epsilon_L.
\end{aligned}$$

Using the definition of induced ℓ_1 norm and Equation (5.33) and (5.34), we obtain that

$$\begin{aligned}
&\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{E}\right\|_\infty \\
&\leq \max_{i\in\overline{\mathcal{T}}}\left\|\mathbf{d}_i^T\mathbf{D}_{\mathcal{T}}\right\|_1 \cdot \left\|\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\right\|_1 \cdot \epsilon_L \\
&\leq \max_{i\in\overline{\mathcal{T}}}\left\|\mathbf{d}_i^T\mathbf{D}_{\mathcal{T}}\right\|_1 \cdot \frac{1}{1-(k-1)\mu(\mathbf{D})} \cdot \epsilon_L,
\end{aligned}$$

where we have denoted by k the $\ell_{0,\infty}$ norm of $\mathcal{T}$. Notice that due to the limited support of the atoms, the vector $\mathbf{d}_i^T\mathbf{D}_{\mathcal{T}}$ has at most k non-zeros entries. Additionally, each of these is bounded in absolute value by the mutual coherence of the dictionary. Therefore, $\|\mathbf{d}_i^T\mathbf{D}_{\mathcal{T}}\|_1 \leq k\mu(\mathbf{D})$ (note that $i \notin \mathcal{T}$). Plugging this into the above equation, we obtain

$$\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{E}\right\|_\infty \leq \frac{k\mu(\mathbf{D})}{1-(k-1)\mu(\mathbf{D})} \cdot \epsilon_L.$$

Rearranging our assumption in Equation (5.37), we get $\frac{k\mu(\mathbf{D})}{1-(k-1)\mu(\mathbf{D})} \leq 1$. Therefore, the above becomes

$$\left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{E}\right\|_\infty \leq \epsilon_L. \tag{5.40}$$

Finally, plugging Equation (5.39) and (5.40) into Equation (5.38), we conclude that

$$\begin{aligned}
&\|\mathbf{D}_{\overline{\mathcal{T}}}^T(\mathbf{Y}-\mathbf{X}_{\text{LS}})\|_\infty \\
&\leq \left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{E}\right\|_\infty + \left\|\mathbf{D}_{\overline{\mathcal{T}}}^T\mathbf{D}_{\mathcal{T}}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)^{-1}\mathbf{D}_{\mathcal{T}}^T\mathbf{E}\right\|_\infty \\
&\leq \epsilon_L + \epsilon_L = 2\epsilon_L.
\end{aligned}$$

□

For completeness, and before moving to the proof of the stability of BP, we now reproduce Theorem 8 from [Tro06].

Theorem 22. (Tropp): Suppose a clean signal $\mathbf{X}$ has a representation $\mathbf{D}\boldsymbol{\Gamma}$, and that it is contaminated with noise $\mathbf{E}$ to create the signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. Assume further that $\mathbf{Y}$ is a signal whose best ℓ_2 approximation over the support of $\boldsymbol{\Gamma}$, denoted by $\mathcal{T}$, is given by $\mathbf{X}_{\text{LS}}$, and that $\mathbf{X}_{\text{LS}} = \mathbf{D}\boldsymbol{\Gamma}_{\text{LS}}$. Moreover, consider $\boldsymbol{\Gamma}_{\text{BP}}$ to be the solution to the Lagrangian BP formulation with parameter λ. If the following conditions are satisfied:

1. The ERC is met with constant $\theta \geq 0$ for the support $\mathcal{T}$; And
2. $\|\mathbf{D}^T(\mathbf{Y}-\mathbf{X}_{\text{LS}})\|_\infty \leq \lambda\theta$,

then the following hold:

1. The support of $\boldsymbol{\Gamma}_{\text{BP}}$ is contained in that of $\boldsymbol{\Gamma}$.

2. $\|\mathbf{\Gamma}_{\mathrm{BP}} - \mathbf{\Gamma}_{\mathrm{LS}}\|_\infty \leq \lambda \left\| \left(\mathbf{D}_\mathcal{T}^T \mathbf{D}_\mathcal{T}\right)^{-1} \right\|_\infty$.

3. In particular, the support of $\mathbf{\Gamma}_{\mathrm{BP}}$ contains every index i for which $|\mathbf{\Gamma}_{\mathrm{LS}\,i}| > \lambda \left\| \left(\mathbf{D}_\mathcal{T}^T \mathbf{D}_\mathcal{T}\right)^{-1} \right\|_\infty$.

4. The minimizer of the problem, $\mathbf{\Gamma}_{\mathrm{BP}}$, is unique.

Armed with these, we now proceed to proving Theorem 19.

Proof. In this proof we shall show that Theorem 22 can be reformulated in terms of the $\ell_{0,\infty}$ norm and the mutual coherence of $\mathbf{D}$, thus adapting it to the convolutional setting. Our strategy will be first to restrict its conditions (1) and (2), and then to derive from its theses the desired claims.

To this end, we begin by converting the assumption on the ERC into another one relying on the $\ell_{0,\infty}$ norm. This can be readily done using Theorem 18, which states that the ERC is met assuming the $\ell_{0,\infty}$ norm of the support is less than $\frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$ – a condition that is indeed satisfied due to our assumption in Equation (5.36). Next, we move to assumption (2) in Theorem 22. We can lower bound the ERC constant θ by employing the inequality in (5.35), thus obtaining

$$\theta = 1 - \max_{i \notin \mathcal{T}} \|\mathbf{D}_\mathcal{T}^\dagger \mathbf{d}_i\|_1 \geq 1 - \frac{\|\mathbf{\Gamma}\|_{0,\infty}\mu(\mathbf{D})}{1 - (\|\mathbf{\Gamma}\|_{0,\infty} - 1)\mu(\mathbf{D})}.$$

Using the assumption that $\|\mathbf{\Gamma}\|_{0,\infty} \leq \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$, as stated in Equation (5.36), the above can be simplified into

$$\theta = 1 - \max_{i \notin \mathcal{T}} \|\mathbf{D}_\mathcal{T}^\dagger \mathbf{d}_i\|_1 \geq \frac{1}{2}. \tag{5.41}$$

Bringing now the fact that $\lambda = 4\epsilon_L$, as assumed in our Theorem, and using the just obtained inequality (5.41), condition (2) must hold since

$$\|\mathbf{D}^T(\mathbf{Y} - \mathbf{X}_{\mathrm{LS}})\|_\infty \leq 2\epsilon_L \leq \theta\lambda.$$

Note that the leftmost inequality is Lemma (5.10.1), and the implication here is that $\lambda \geq 4\epsilon_L$.

Thus far, we have addressed the conditions in Theorem 22, showing that they hold in our convolutional setting. In the remainder of this proof we shall expand on its results, in particular point 2 and 3. We can upper bound the term $\left\| \left(\mathbf{D}_\mathcal{T}^T \mathbf{D}_\mathcal{T}\right)^{-1} \right\|_\infty$ using Equation (5.34), obtaining

$$\left\| \left(\mathbf{D}_\mathcal{T}^T \mathbf{D}_\mathcal{T}\right)^{-1} \right\|_\infty \leq \frac{1}{1 - (\|\mathbf{\Gamma}\|_{0,\infty} - 1)\mu(\mathbf{D})}. \tag{5.42}$$

Using once again the assumption that $\|\mathbf{\Gamma}\|_{0,\infty} \leq \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D})}\right)$, we have that $\|\mathbf{\Gamma}\|_{0,\infty} < \frac{1}{3}\left(3 + \frac{1}{\mu(\mathbf{D})}\right)$. From this last inequality, we get $(\|\mathbf{\Gamma}\|_{0,\infty} - 1)\,\mu(\mathbf{D}) < \frac{1}{3}$. Thus, it follows that

$$\frac{1}{1 - (\|\mathbf{\Gamma}\|_{0,\infty} - 1)\mu(\mathbf{D})} < \frac{3}{2}.$$

Based on the above inequality, and Equation (5.42), we get

$$\left\| \left(\mathbf{D}_{\mathcal{T}}^T \mathbf{D}_{\mathcal{T}} \right)^{-1} \right\|_\infty < \frac{3}{2}. \tag{5.43}$$

Plugging this into the second result in Tropp's theorem, together with the above fixed λ, we obtain that

$$\|\mathbf{\Gamma}_{\mathrm{BP}} - \mathbf{\Gamma}_{\mathrm{LS}}\|_\infty \leq \lambda \left\| \left(\mathbf{D}_{\mathcal{T}}^T \mathbf{D}_{\mathcal{T}} \right)^{-1} \right\|_\infty < 4\epsilon_L \cdot \frac{3}{2} = 6\epsilon_L. \tag{5.44}$$

On the other hand, looking at the distance to the real $\mathbf{\Gamma}$,

$$\begin{aligned} \|\mathbf{\Gamma}_{\mathrm{LS}} - \mathbf{\Gamma}\|_\infty &= \| \left(\mathbf{D}_{\mathcal{T}}^T \mathbf{D}_{\mathcal{T}} \right)^{-1} \mathbf{D}_{\mathcal{T}}^T \left(\mathbf{Y} - \mathbf{X} \right) \|_\infty \\ &\leq \| \left(\mathbf{D}_{\mathcal{T}}^T \mathbf{D}_{\mathcal{T}} \right)^{-1} \|_\infty \cdot \|\mathbf{D}_{\mathcal{T}}^T \mathbf{E}\|_\infty < \frac{3}{2}\epsilon_L. \end{aligned} \tag{5.45}$$

For the first inequality we have used the definition of the induced ℓ_∞ norm, and the second one follows from (5.43) and a similar derivation to that in (5.39). Finally, using triangle inequality and Equations (5.45) and (5.44) we obtain

$$\|\mathbf{\Gamma}_{\mathrm{BP}} - \mathbf{\Gamma}\|_\infty \leq \|\mathbf{\Gamma}_{\mathrm{BP}} - \mathbf{\Gamma}_{\mathrm{LS}}\|_\infty + \|\mathbf{\Gamma}_{\mathrm{LS}} - \mathbf{\Gamma}\|_\infty < \frac{15}{2}\epsilon_L.$$

The third result in the theorem follows immediately from the above.

5.10.5 Global Pursuit Through Local Processing

Let us consider the Iterative Soft Thresholding algorithm which minimizes the global BP problem by iterating the following updates

$$\mathbf{\Gamma}^k = \mathcal{S}_{\lambda/c} \left(\mathbf{\Gamma}^{k-1} + \frac{1}{c} \mathbf{D}^T (\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}^{k-1}) \right),$$

where $\mathcal{S}$ applies an entry-wise soft thresholding operation with threshold λ/c. Defining as $\mathbf{P}_i$ the operator which extracts the i^{th} $m-$dimensional vector from $\mathbf{\Gamma}$, we can break the above algorithm into local updates by

$$\mathbf{P}_i \mathbf{\Gamma}^k = \mathcal{S}_{\lambda/c} \left(\mathbf{P}_i \mathbf{\Gamma}^{k-1} + \frac{1}{c} \mathbf{P}_i \mathbf{D}^T (\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}^{k-1}) \right).$$

As a first observation, the matrix $\mathbf{P}_i \mathbf{D}^T$, which is of size $m \times N$, is in-fact $\mathbf{D}_L^T$ padded with zeros. As a consequence, the above can be rewritten as follows:

$$\mathbf{P}_i \mathbf{\Gamma}^k = \mathcal{S}_{\lambda/c} \left(\mathbf{P}_i \mathbf{\Gamma}^{k-1} + \frac{1}{c} \mathbf{P}_i \mathbf{D}^T \mathbf{R}_i^T \mathbf{R}_i (\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}^{k-1}) \right),$$

where we have used $\mathbf{R}_i$ as the operator which extracts the i^{th} $n-$dimensional patch from an $N-$dimensional global signal. The operator $\mathbf{P}_i$ extracts m rows from $\mathbf{D}^T$, while $\mathbf{R}_i^T$ extracts its

non-zero columns. Therefore, $\mathbf{P}_i\mathbf{D}^T\mathbf{R}_i^T = \mathbf{D}_L^T$, and so we can write

$$\mathbf{P}_i\mathbf{\Gamma}^k = \mathcal{S}_{\lambda/c}\left(\mathbf{P}_i\mathbf{\Gamma}^{k-1} + \frac{1}{c}\mathbf{D}_L^T\mathbf{R}_i(\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}^{k-1})\right).$$

Noting that $\boldsymbol{\alpha}_i^k = \mathbf{P}_i\mathbf{\Gamma}^k$ is the i^{th} local sparse code, and defining $\mathbf{r}_i^k = \mathbf{R}_i(\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}^{k-1})$ as the corresponding patch-residual at iteration k, we obtain our final update (for every patch)

$$\boldsymbol{\alpha}_i^k = \mathcal{S}_{\lambda/c}\left(\boldsymbol{\alpha}_i^{k-1} + \frac{1}{c}\ \mathbf{D}_L^T\ \mathbf{r}_i^{k-1}\right).$$

We summarize the above derivations in the main corpus of the chapter.

Chapter 6

Multi-Layer Convolutional Sparse Modeling

Chapter Summary

The recently proposed Multi-Layer Convolutional Sparse Coding (ML-CSC) model, consisting of a cascade of convolutional sparse layers, provides a new interpretation of Convolutional Neural Networks (CNNs). Under this framework, the computation of the forward pass in a CNN is equivalent to a pursuit algorithm aiming to estimate the nested sparse representation vectors – or feature maps – from a given input signal. Despite having served as a pivotal connection between CNNs and sparse modeling, a deeper understanding of the ML-CSC is still lacking: there are no pursuit algorithms that can serve this model exactly, nor are there conditions to guarantee a non-empty model. While one can easily obtain signals that *approximately* satisfy the ML-CSC constraints, it remains unclear how to simply sample from the model and, more importantly, how one can train the convolutional filters from real data.

In this chapter, we propose a sound pursuit algorithm for the ML-CSC model by adopting a projection approach. We provide new and improved bounds on the stability of the solution of such pursuit and we analyze different practical alternatives to implement this in practice. We show that the training of the filters is essential to allow for non-trivial signals in the model, and we derive an online algorithm to learn the dictionaries from real data, effectively resulting in cascaded sparse convolutional layers. Last, but not least, we demonstrate the applicability of the ML-CSC model for several applications in an unsupervised setting, providing competitive results. The work condensed in this chapter represents a bridge between matrix factorization, sparse dictionary learning and sparse auto-encoders, and we analyze these connections in detail.

6.1 Convolutional Sparse Coding and Deep Learning

As we have reviewed in Chapter 2, sparse representation modeling brought about the idea that natural signals can be (well) described as a linear combination of only a few building blocks or components, known as atoms [BDE09]. The popularity of this model has grown dramatically, partially because of the (often state-of-the-art) algorithms that have emerged from it are backed by elegant theoretical results and sound understanding.

Neural networks, on the other hand, were introduced around forty years ago and were shown to provide powerful classification algorithms through a series of function compositions [LBD+90, RHW+88]. It was not until the last half-decade, however, that through a series of incremental modifications these methods were boosted to become the state-of-the-art machine learning tools for a wide range of problems, and across many different fields [LBH15]. For the most part, the development of new variants of deep convolutional neural networks (CNNs) has been driven by trial-and-error strategies and a considerable amount of intuition.

Withal, a few research groups have begun providing theoretical justifications and analysis strategies for CNNs from very different perspectives. For instance, by employing wavelet filters instead of adaptive ones, the work by Bruna and Mallat [BM13] demonstrated how *scattering networks* represent shift invariant analysis operators that are robust to deformations (in a Lipschitz-continuous sense). The work in [GSB15] showed that deep neural networks preserve the metric structure of the data, under Gaussian weights assumption. In [CSS16], the authors proposed a hierarchical tensor factorization analysis model to analyze deep CNNs. Fascinating connections between sparse modeling and CNN have also been proposed. In [GL10], a neural network architecture was shown to be able to learn iterative shrinkage operators, essentially *unrolling* the iterations of a sparse pursuit. Building on this interpretation, the work in [XWG+16] further showed that CNNs can in fact improve the performance of sparse recovery algorithms.

A precise connection between sparse modeling and CNNs was recently presented in [PRE16], and its contribution is centered in defining the Multi-Layer Convolutional Sparse Coding (ML-CSC) model. When deploying this model to real signals, compromises were made in way that each layer is only *approximately* explained by the following one. With this relaxation in the pursuit of the convolutional representations, the main observation of this work is that the inference stage of CNNs – nothing but the forward-pass – can be interpreted as a very crude pursuit algorithm seeking for unique sparse representations. This is a useful perspective as it provides a precise optimization objective which, it turns out, CNNs attempt to minimize.

The work in [PRE16] further proposed improved pursuits for approximating the sparse representations of the network, or feature maps, such as the Layered Basis Pursuit algorithm. Nonetheless, as we will show later, neither this nor the forward pass serve the ML-CSC model exactly, as they do not provide signals that comply with the model assumptions. In addition, the theoretical guarantees accompanying these layered approaches suffer from bounds that become looser with the network's depth. The lack of a suitable pursuit, in turn, obscures how to properly sample from the ML-CSC model, and how to train the model's dictionaries from real data.

In this work we undertake a fresh study of the ML-CSC and of pursuit algorithms for signals

in this model. Our contributions will be guided by addressing the following questions:

1. Given proper convolutional dictionaries, how can one project[1] signals onto the ML-CSC model?

2. When will the model allow for *any* signal to be expressed in terms of nested sparse representations? In other words, is the model empty?

3. What conditions should the convolutional dictionaries satisfy? and how can we adapt or learn them to represent real-world signals?

4. How is the learning of the ML-CSC model related to traditional CNN and dictionary learning algorithms?

5. What kind of performance can be expected from this model?

Before proceeding, it is worth noting that the model we analyze in this work is related to several recent contributions, both in the realm of sparse representations and deep-learning. On the one hand, the ML-CSC model is tightly connected to dictionary learning approaches, in particular to those leveraging different structures or constraints in the construction of such dictionary. A very partial list of these works include the Chasing Butterflies approach [LMG15], fast transform learning [CMTD15], Trainlets [SOZE16], among several others. On the other hand, and because of the unsupervised flavor of the learning algorithm, our work shares connections to sparse auto-encoders [Ng11], and in particular to the k-sparse [MF13] and winner-take-all versions [MF15].

Lastly, because the analyzis in this chapter will not be focused on local vectors or structures, we will return to the traditional notation of employing uppercase notation for matrices exclusively, and lowercase for vectors.

6.2 Preliminaries on ML-CSC

The Multi-Layer Convolutional Sparse Coding (ML-CSC) model is a natural extension of the CSC described above, as it assumes that a signal can be expressed by sparse representations at different layers in terms of nested convolutional filters. Suppose $\mathbf{x} = \mathbf{D}_1\boldsymbol{\gamma}_1$, for a convolutional dictionary $\mathbf{D}_1 \in \mathbb{R}^{N\times Nm_1}$ and an $\ell_{0,\infty}$-sparse representation $\boldsymbol{\gamma}_1 \in \mathbb{R}^{Nm_1}$. One can cascade this model by imposing a similar assumption on the representation $\boldsymbol{\gamma}_1$, i.e., $\boldsymbol{\gamma}_1 = \mathbf{D}_2\boldsymbol{\gamma}_2$, for a corresponding convolutional dictionary $\mathbf{D}_2 \in \mathbb{R}^{Nm_1\times Nm_2}$ with m_2 local filters and a $\ell_{0,\infty}$-sparse $\boldsymbol{\gamma}_2$, as depicted in Figure 6.1. In this case, $\mathbf{D}_2$ is a also a convolutional dictionary with local filters skipping m_1 entries at a time[2] – as there are m_1 *channels* in the representation $\boldsymbol{\gamma}_1$.

[1] By projection, we refer to the task of getting the closest signal to the one given that obeys the model assumptions.

[2] This construction provides operators that are convolutional in the space domain, but not in the channel domain – just as for CNNs.

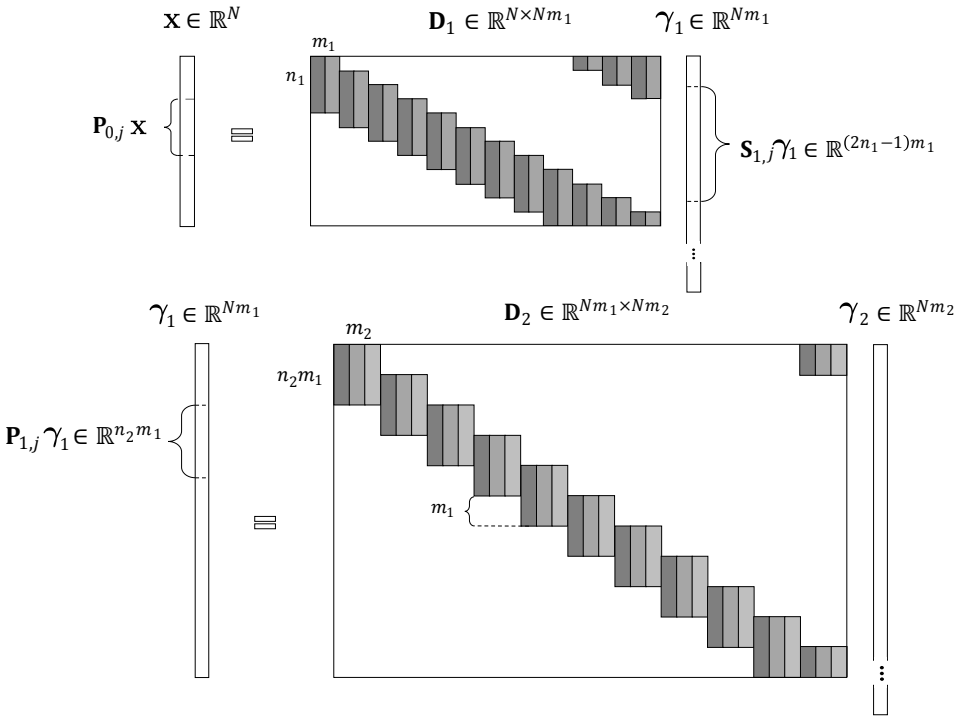


Figure 6.1: The CSC model (top), and its ML-CSC extension by imposing a similar model on $\boldsymbol{\gamma}_1$ (bottom). From a local perspective, a patch from the signal, $\mathbf{P}_{0,j}\mathbf{x}$ has a corresponding sparse stripe given by $\mathbf{S}_{1,j}\boldsymbol{\gamma}_1$. An analogous decomposition can be stated for a patch from the signal $\boldsymbol{\gamma}_1$, represented by $\mathbf{P}_{1,j}\boldsymbol{\gamma}_1$.

Because of this multi-layer structure, vector $\boldsymbol{\gamma}_1$ can be viewed both as a sparse representation (in the context of $\mathbf{x} = \mathbf{D}_1\boldsymbol{\gamma}_1$) or as a signal (in the context of $\boldsymbol{\gamma}_1 = \mathbf{D}_2\boldsymbol{\gamma}_2$). Thus, one one can refer to both its stripes (looking backwards to patches from $\mathbf{x}$) or its patches (looking forward, corresponding to stripes of $\boldsymbol{\gamma}_2$). In this way, when analyzing the ML-CSC model we will not only employ the $\ell_{0,\infty}$ norm as defined above, but we will also leverage its *patch* counterpart, where the maximum is taken over all patches from the sparse vector by means of a patch extractor operator $\mathbf{P}_i$. In order to make their difference explicit, we will denote them as $\|\boldsymbol{\gamma}\|_{0,\infty}^s$ and $\|\boldsymbol{\gamma}\|_{0,\infty}^p$ for stripes and patches, respectively. In addition, we will employ the $\ell_{2,\infty}$ norm version, naturally defined as $\|\boldsymbol{\gamma}\|_{2,\infty}^s = \max_i \|\mathbf{S}_i\boldsymbol{\gamma}\|_2$, and analogously for patches.

We now formalize the model definition:

Definition 23. ML-CSC model:
Given a set of convolutional dictionaries $\{\mathbf{D}_i\}_{i=1}^L$ of appropriate dimensions, a signal $\mathbf{x} \in \mathbb{R}^N$

admits a representation in terms of the ML-CSC model if

$$\begin{aligned}
\mathbf{x} &= \mathbf{D}_1\boldsymbol{\gamma}_1, \quad \|\boldsymbol{\gamma}_1\|_{0,\infty}^{s} \leq \lambda_1, \\
\boldsymbol{\gamma}_1 &= \mathbf{D}_2\boldsymbol{\gamma}_2, \quad \|\boldsymbol{\gamma}_2\|_{0,\infty}^{s} \leq \lambda_2, \\
&\vdots \\
\boldsymbol{\gamma}_{L-1} &= \mathbf{D}_L\boldsymbol{\gamma}_L, \quad \|\boldsymbol{\gamma}_L\|_{0,\infty}^{s} \leq \lambda_L.
\end{aligned}$$

We will refer to the set of signals satisfying the ML-CSC model assumptions with parameter $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_L]$, as the set $\mathcal{M}_{\boldsymbol{\lambda}}$. In addition, when referring to a signal $\mathbf{x} \in \mathcal{M}_{\boldsymbol{\lambda}}$, we will often denote it by $\mathbf{x}(\boldsymbol{\gamma}_i)$ to emphasize its decomposition in terms of the nested representations $\{\boldsymbol{\gamma}_i\}_{i=1}^{L}$.

Note that $\mathbf{x} \in \mathcal{M}_{\boldsymbol{\lambda}}$ can also be expressed as $\mathbf{x} = \mathbf{D}_1\mathbf{D}_2 \ldots \mathbf{D}_L\boldsymbol{\gamma}_L$. For the purpose of the following derivations, define $\mathbf{D}^{(i)}$ to be the *effective* dictionary at the i^{th} level, i.e., $\mathbf{D}^{(i)} = \mathbf{D}_1\mathbf{D}_2 \ldots \mathbf{D}_i$. This way, one can concisely write

$$\mathbf{x} = \mathbf{D}^{(L)}\boldsymbol{\gamma}_L,$$

where $\mathbf{D}^{(L)}$ is the L-layers Convolutional Dictionary. Generally, we have that $\mathbf{x} = \mathbf{D}^{(i)}\boldsymbol{\gamma}_i,\ 1 \leq i \leq L$.

Interestingly, the ML-CSC can be interpreted as a special case of a CSC model: one that enforces a very specific structure on the intermediate representations. We make this statement precise in the following Lemma:

Lemma 6.2.1. *Given the ML-CSC model described by the set of convolutional dictionaries $\{\mathbf{D}_i\}_{i=1}^{L}$, with filters of spatial dimensions n_i and channels m_i, any dictionary $\mathbf{D}^{(i)} = \mathbf{D}_1\mathbf{D}_2 \ldots \mathbf{D}_i$ is a convolutional dictionary with m_i local atoms of dimension $n_i^{eff} = \sum_{j=1}^{i} n_j - (i-1)$. In other words, the ML-CSC model is a structured global convolutional model.*

The proof of this lemma is rather straight forward, and we include it in Appendix 6.7.1. Note that what was denoted as the effective dimension at the i^{th} layer is nothing else than what is known in the deep learning community as the *receptive field* of a filter at layer i. Here, we have made this concept precise in the context of the ML-CSC model.

As it was presented, the convolutional model assumes that every n-dimensional atom is located at every possible location, which implies that the filter is shifted with strides of $s = 1$. An alternative, which effectively reduces the redundancy of the resulting dictionary, is to consider a stride greater than one. In such case, the resulting dictionary is of size $N \times Nm_1/s$ for one dimensional signals, and $N \times Nm_1/s^2$ for images. This construction, popular in the CNN community, does not alter the effective size of the filters but rather decreases the length of each stripe by a factor of s in each dimension. In the limit, when $s = n_1$, one effectively considers non-overlapping blocks and the stripe will be of length[3] m_1 - the number of local filters.

[3]When $s = n_1$, the system is no longer shift-invariant, but rather invariant with a shift of n samples.

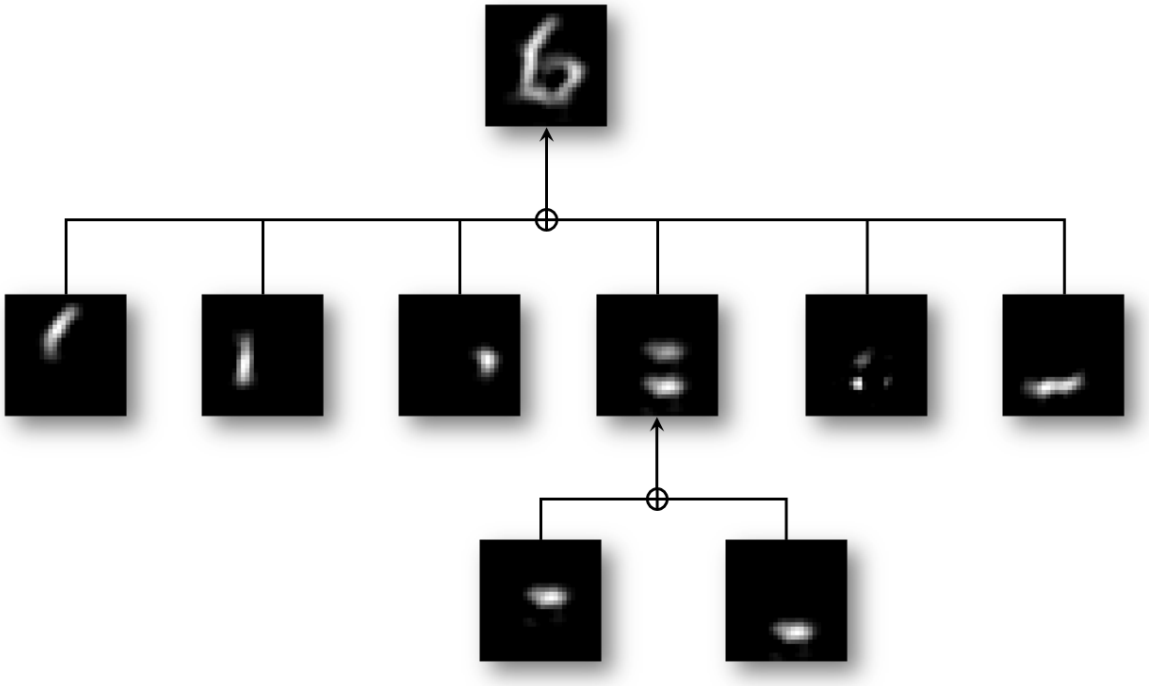


Figure 6.2: From atoms to molecules: Illustration of the ML-CSC model for a number 6. Two local convolutional atoms (bottom row) are combined to create slightly more complex structures – molecules – at the second level, which are then combined to create the global atom representing, in this case, a digit. Note that even though the atoms are local (with small support) and convolutional, we depict them in their respective locations within the global structure. Refer to the main body (Section 6.2) for a detailed description of this decomposition.

Naturally, one can also employ $s > 1$ for any of the multiple layers of the ML-CSC model. We will consider $s = 1$ for all layers in our derivations for simplicity.

The ML-CSC imposes a unique structure on the global dictionary $\mathbf{D}^{(L)}$, as it provides a multi-layer linear composition of simpler structures. In other words, $\mathbf{D}_1$ contains (small) local n_1-dimensional atoms. The product $\mathbf{D}_1\mathbf{D}_2$ contains in each of its columns a linear combination of atoms from $\mathbf{D}_1$, merging them to create molecules. Further layers continue to create more complex constructions out of the simpler convolutional building blocks. We depict an example of such decomposition in Figure 6.2 for a 3^{rd}-layer convolutional atom of the digit "6". While the question of how to obtain such dictionaries will be addressed later on, let us make this illustration concrete: consider this atom to be given by $\mathbf{x}_0 = \mathbf{D}_1\mathbf{D}_2\mathbf{d}_3$, where $\mathbf{d}_3$ is sparse, producing the upper-most image $\mathbf{x}_0$. Denoting by $\mathcal{T}(\mathbf{d}_3) = Supp(\mathbf{d}_3)$, this atom can be equally expressed as

$$\mathbf{x}_0 = \mathbf{D}^{(2)}\mathbf{d}_3 = \sum_{j \in \mathcal{T}(\mathbf{d}_3)} \mathbf{d}_j^{(2)} d_3^j.$$

In words, the effective atom is composed of *a few* elements from the effective dictionary $\mathbf{D}^{(2)}$. These are the building blocks depicted in the middle of Figure 6.2. Likewise, we now focus on the fourth of such atoms, $\mathbf{d}_{j_4}^{(2)} = \mathbf{D}_1\mathbf{d}_{2,j_4}$. In this particular case, $\|\mathbf{d}_{2,j_4}\|_0 = 2$. Indicating these

two non-zeros elements by i_1 and i_2, we can express:

$$\mathbf{d}_{j_4}^{(2)} = \mathbf{d}_{i_1}^{(1)} d_{2,j_1}^{i_1} + \mathbf{d}_{i_2}^{(1)} d_{2,j_1}^{i_2}.$$

These two atoms from $\mathbf{D}_1$ are precisely those appearing in the bottom of the decomposition.

6.2.1 Pursuit in the noisy setting

Real signals might contain noise or deviations from the above idealistic model assumption, preventing us from enforcing the above model exactly. Consider the scenario of acquiring a signal $\mathbf{y} = \mathbf{x} + \mathbf{v}$, where $\mathbf{x} \in \mathcal{M}_{\boldsymbol{\lambda}}$ and $\mathbf{v}$ is a nuisance vector of bounded energy, $\|\mathbf{v}\|_2 \leq \mathcal{E}_0$. In this setting, the objective is to estimate all the representations $\boldsymbol{\gamma}_i$ which explain the measurements $\mathbf{y}$ up to an error of $\mathcal{E}_0$. This pursuit problem – searching for sparse convolutional features under the ML-CSC model – can be formulated in a number of different ways depending on the model deviations assumed at each layer. In its most general form, this pursuit is represented by the Deep Coding Problem ($\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$), as introduced in [PRE16]:

Definition 24. $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ Problem:
For a global signal $\mathbf{y}$, a set of convolutional dictionaries $\{\mathbf{D}_i\}_{i=1}^L$, and vectors $\boldsymbol{\lambda}$ and $\mathcal{E}$, the deep coding problem $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ is defined as:

$$\begin{aligned}
(\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}): \quad \text{find} \quad \{\boldsymbol{\gamma}_i\}_{i=1}^L \quad \text{s.t.} \quad & \|\mathbf{y} - \mathbf{D}_1\boldsymbol{\gamma}_1\|_2 \leq \mathcal{E}_0, && \|\boldsymbol{\gamma}_1\|_{0,\infty}^s \leq \lambda_1 \\
& \|\boldsymbol{\gamma}_1 - \mathbf{D}_2\boldsymbol{\gamma}_2\|_2 \leq \mathcal{E}_1, && \|\boldsymbol{\gamma}_2\|_{0,\infty}^s \leq \lambda_2 \\
& \qquad\qquad \vdots && \qquad\qquad \vdots \\
& \|\boldsymbol{\gamma}_{L-1} - \mathbf{D}_L\boldsymbol{\gamma}_L\|_2 \leq \mathcal{E}_{L-1}, && \|\boldsymbol{\gamma}_L\|_{0,\infty}^s \leq \lambda_L,
\end{aligned}$$

where the scalars λ_i and $\mathcal{E}_i$ are the i^{th} entries of $\boldsymbol{\lambda}$ and $\mathcal{E}$, respectively.

The solution to this problem was shown to be stable in terms of a bound on the ℓ_2-distance between the estimated representations $\hat{\boldsymbol{\gamma}}_i$ and the true ones $\boldsymbol{\gamma}_i$. These results depend on the characterization of the dictionaries through their mutual coherence, $\mu(\mathbf{D})$, which measures the maximal normalized correlation between atoms in the dictionary. Formally, assuming the atoms are normalized as $\|\mathbf{d}_i\|_2 = 1 \; \forall i$, this measure is defined as

$$\mu(\mathbf{D}) = \max_{i \neq j} |\mathbf{d}_i^T \mathbf{d}_j|.$$

Relying on this measure, Theorem 5 in [PRE16] shows that given a signal $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ contaminated with noise of known energy $\mathcal{E}_0^2$, if the representations satisfy the sparsity constraint

$$\|\boldsymbol{\gamma}_i\|_{0,\infty}^s < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right),$$

then the solution to the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ given by $\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^L$ satisfies

$$\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 \leq 4{\mathcal{E}_0}^2 \prod_{j=1}^{i} \frac{4^{i-1}}{1-(2\|\boldsymbol{\gamma}_j\|_{0,\infty}^s - 1)\mu(\mathbf{D}_j)}.$$

In the particular instance of the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ where $\mathcal{E}_i = 0$ for $1 \leq i \leq L-1$, the above bound can be made tighter by a factor of 4^{i-1} while preserving the same form.

These results are encouraging, as they show for the first time stability guarantees for a problem for which the forward pass provides an approximate solution. More precisely, if the above model deviations are considered to be greater than zero ($\mathcal{E}_i > 0$) several layer-wise algorithms, including the forward pass of CNNs, provide approximations to the solution of this problem [PRE16].

Two remarks should be noted about the above stability result:

1. The bound increases with the number of layers or the depth of the network. This is a direct consequence of the layer-wise relaxation in the above pursuit, which causes these discrepancies to accumulate over the layers.

2. Given the underlying signal $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$, with representations $\{\boldsymbol{\gamma}_i\}_{i=1}^L$, this problem searches for their corresponding estimates $\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^L$. However, because at each layer $\|\hat{\boldsymbol{\gamma}}_{i-1} - \mathbf{D}_i\hat{\boldsymbol{\gamma}}_i\|_2 > 0$, this problem *does not* provide representations for a signal in the model. In other words, $\hat{\mathbf{x}} \neq \mathbf{D}_1\hat{\boldsymbol{\gamma}}_1$, $\hat{\boldsymbol{\gamma}}_1 \neq \mathbf{D}_2\hat{\boldsymbol{\gamma}}_2$, and generally $\hat{\mathbf{x}} \notin \mathcal{M}_{\boldsymbol{\lambda}}$.

6.3 A Projection Alternative

In this section we provide an alternative approach to the problem of estimating the underlying representations $\boldsymbol{\gamma}_i$ under the same noisy scenario of $\mathbf{y} = \mathbf{x}(\boldsymbol{\gamma}_i) + \mathbf{v}$. In particular, we are interested in projecting the measurements $\mathbf{y}$ onto the set $\mathcal{M}_{\boldsymbol{\lambda}}$. Consider the following projection problem:

Definition 25. ML-CSC Projection $\mathbf{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$:
For a signal $\mathbf{y}$ and a set of convolutional dictionaries $\{\mathbf{D}_i\}_{i=1}^L$, define the Multi-Layer Convolutional Sparse Coding projection as:

$$(\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}): \quad \min_{\{\boldsymbol{\gamma}_i\}_{i=1}^L} \|\mathbf{y} - \mathbf{x}(\boldsymbol{\gamma}_i)\|_2 \quad \text{s.t.} \quad \mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}. \tag{6.1}$$

Note that this problem differs from the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ counterpart in that we seek for a signal close to $\mathbf{y}$, whose representations $\boldsymbol{\gamma}_i$ give rise to $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$. This is more demanding (less general) than the formulation in the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$. Put differently, the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem can be considered as a special case of the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ where model deviations are allowed only at the outer-most level. From this perspective, the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ is an instance of the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ for which $\mathcal{E}_i = 0$ for $i \geq 1$. Recall that the theoretical analysis of the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ problem indicated that the error thresholds should increase with the layers. Here, the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem suggests a completely different approach.

6.3.1 Stability of the projection $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$

Given $\mathbf{y} = \mathbf{x}(\boldsymbol{\gamma}_i) + \mathbf{v}$, one can seek for the underlying representations $\boldsymbol{\gamma}_i$ through either the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ or $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem. In light of the above discussion and the known stability result for the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ problem, how close will the solution of the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem be from the true set of representations? The answer is provided through the following result.

Theorem 26. Stability of the solution to the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem:
Suppose $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$ is observed through $\mathbf{y} = \mathbf{x} + \mathbf{v}$, where $\mathbf{v}$ is a bounded noise vector, $\|\mathbf{v}\|_2 \leq \mathcal{E}_0$, and $\|\boldsymbol{\gamma}_i\|_{0,\infty}^s = \lambda_i < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}^{(i)})}\right)$, for $1 \leq i \leq L$. Consider the set $\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^L$ to be the solution of the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem. Then,

$$\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 \leq \frac{4\mathcal{E}_0^2}{1 - (2\|\boldsymbol{\gamma}_i\|_{0,\infty}^s - 1)\mu(\mathbf{D}^{(i)})}.$$

Before presenting the proof of this claim, we note a few remarks on this result:

1. The obtained bound for every layer depends only on the sparsity of the representation and on the mutual coherence of the effective dictionary for that layer, $\mathbf{D}^{(i)}$. This allows us to provide a bound which is not cumulative across the layers – it does not grow with the depth of the network.

2. Unlike the stability result for the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ problem, the assumptions on the sparse vectors $\boldsymbol{\gamma}_i$ are given in terms of the mutual coherence of the effective dictionaries $\mathbf{D}^{(i)}$. Interestingly enough, we will see in the experimental section that one can in fact have that $\mu(\mathbf{D}^{(i-1)}) > \mu(\mathbf{D}^{(i)})$ in practice; i.e., the effective dictionary becomes incoherent as it becomes deeper. On the other hand, the deeper layers correspond to higher abstractions levels, and the corresponding representations are indeed expected to be sparser.

3. While the conditions imposed on the sparse vectors $\boldsymbol{\gamma}_i$ might seem prohibitive, one should remember that this follows from a worst case analysis. Moreover, one can effectively construct analytic nested convolutional dictionaries with small coherence measures, as shown in [PRE16].

We now prove the stability result.

Proof. Denote the solution to the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem by $\hat{\mathbf{x}}$; i.e., $\hat{\mathbf{x}} = \mathbf{D}^{(i)}\hat{\boldsymbol{\gamma}}_i$. Given that the original signal $\mathbf{x}$ satisfies $\|\mathbf{y} - \mathbf{x}\|_2 \leq \mathcal{E}_0$, the solution to the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem, $\hat{\mathbf{x}}$ must satisfy

$$\|\mathbf{y} - \hat{\mathbf{x}}\|_2 \leq \|\mathbf{y} - \mathbf{x}\|_2 \leq \mathcal{E}_0,$$

as this is the signal which provides the shortest ℓ_2 (data-fidelity) distance from $\mathbf{y}$. Note that because $\hat{\mathbf{x}}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$, we can have that $\hat{\mathbf{x}} = \mathbf{D}^{(i)}\hat{\boldsymbol{\gamma}}_i$, $\forall\ 1 \leq i \leq L$. Recalling Lemma 6.7.1, the product $\mathbf{D}_1\mathbf{D}_2\ldots\mathbf{D}_i$ is a convolutional dictionary. In addition, we have required that

$\|\hat{\boldsymbol{\gamma}}_i\|_{0,\infty}^s \leq \lambda_i < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}^{(i)})}\right)$. Therefore, from the same arguments presented in [PSE17b], it follows that

$$\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 \leq \frac{4\mathcal{E}_0^2}{1 - (2\|\boldsymbol{\gamma}_i\|_{0,\infty}^s - 1)\mu(\mathbf{D}^{(i)})}.$$

Interestingly, one can also formulate bounds for the stability of the solution, i.e. $\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2$, which are the tightest for the inner-most layer, and then increase as one moves to shallower layers – precisely the opposite behavior of the solution to the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ problem. This result, however, provides bounds that are generally looser than the one presented in the above theorem, and so we defer this to Appendix 6.7.2.

6.3.2 Pursuit Algorithms

Both the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ and $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problems seek for underlying representations $\hat{\boldsymbol{\gamma}}_i$ which explain – under different assumptions – the measurements $\mathbf{y}$. The next natural question is how and to what accuracy one can retrieve the solutions of those respective problems. In other words, how does one solve these problems *in practice*?

As shown in [PRE16], one can approximate the solution to the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ in a layer-wise manner, solving for the sparse representations $\hat{\boldsymbol{\gamma}}_i$ progressively from $i = 1, \ldots, L$. Surprisingly, the Forward Pass of a CNN *is* one such algorithm, and it provides an approximate solution of this problem. Better alternatives were also proposed, such as the Layered BP algorithm, where each representation $\hat{\boldsymbol{\gamma}}_i$ is sparse coded (in a Basis Pursuit formulation) given the previous representation $\hat{\boldsymbol{\gamma}}_{i-1}$ and dictionary $\mathbf{D}_i$. As solutions to the $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ problem, naturally, these algorithms inherit the layer-wise relaxation referred above, which causes the theoretical bounds to increase as a function of the layers or network depth.

Moving to the variation proposed in this work, how can one solve the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem in practice? Applying the above layer-wise pursuit is clearly not an option, since after obtaining a necessarily distorted estimate $\hat{\boldsymbol{\gamma}}_1$ we cannot proceed with equalities for the next layers, as $\boldsymbol{\gamma}_1$ does not necessarily have a perfectly sparse representation with respect to $\mathbf{D}_2$. Herein we present a simple approach based on a global sparse coding solver which yields a stable solution.

Algorithm 6.1 ML-CSC Pursuit

Input: $\mathbf{y}, \{\mathbf{D}_i\}, k$

$\hat{\boldsymbol{\gamma}}_L \leftarrow \text{Pursuit}(\mathbf{y}, \mathbf{D}^{(L)}, k)$

for $j = L, \ldots, 1$ **do**
  $\hat{\boldsymbol{\gamma}}_{j-1} \leftarrow \mathbf{D}_j \hat{\boldsymbol{\gamma}}_j$

end

return $\{\hat{\boldsymbol{\gamma}}_i\}$

Consider Algorithm 6.1. This approach circumvents the problem of sparse coding the intermediate features while guaranteeing their exact expression in terms of the following layer. This is done by first running a Pursuit for the deepest representation through an algorithm

which provides an approximate solution to the following problem:

$$\min_{\boldsymbol{\gamma}} \|\mathbf{y} - \mathbf{D}^{(L)}\boldsymbol{\gamma}\|_2^2 \;\text{ s.t. }\; \|\boldsymbol{\gamma}\|_{0,\infty}^{s} \leq k. \tag{6.2}$$

In a setting where a signal $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ has been corrupted with noise of known energy $\mathcal{E}_0^2$, one could reformulate this problem by analogously minimizing over the $\ell_{0,\infty}$ norm of $\boldsymbol{\gamma}$ subject to the constraint $\|\mathbf{y} - \mathbf{D}^{(L)}\boldsymbol{\gamma}\|_2^2 \leq \mathcal{E}_0^2$. We employ the formulation in (6.2), however, as this preserves the structure of our projection formulation.

Once the deepest representation has been estimated, we proceed by obtaining the remaining ones by simply applying their definition, thus assuring that $\hat{\mathbf{x}} = \mathbf{D}^{(i)}\hat{\boldsymbol{\gamma}}_i \in \mathcal{M}_{\lambda}$. While this might seem like a dull strategy, we will see in the next section that, if the measurements $\mathbf{y}$ are close enough to a signal in the model, Algorithm 6.1 indeed provides stable estimates $\hat{\boldsymbol{\gamma}}_i$. In fact, the resulting stability bounds will be shown to be generally tighter than those existing for the layer-wise pursuit alternative. Moreover, as we will later see in the Results section, this approach can effectively be harnessed in practice in a real-data scenario.

6.3.3 Stability Guarantees for Pursuit Algorithms

Given a signal $\mathbf{y} = \mathbf{x}(\boldsymbol{\gamma}_i) + \mathbf{v}$, and the respective solution of the ML-CSC Pursuit in Algorithm 6.1, how close will the estimated $\hat{\boldsymbol{\gamma}}_i$ be to the original representations $\boldsymbol{\gamma}_i$? These bounds will clearly depend on the specific Pursuit algorithm employed to obtain $\hat{\boldsymbol{\gamma}}_L$. In what follows, we will present two stability guarantees that arise from solving this sparse coding problem under two different strategies: a greedy and a convex relaxation approach. Before presenting these results, however, we shall need to state two elements that will become necessary for our derivations.

The first one is a property that relates to the propagation of the support, or non-zeros, across the layers. Given the support of a sparse vector $\mathcal{T} = Supp(\boldsymbol{\gamma})$, consider dictionary $\mathbf{D}_{\mathcal{T}}$ as the matrix containing only the columns indicated by $\mathcal{T}$. Define $\|\mathbf{D}_{\mathcal{T}}\|_{\infty}^{0} = \sum_{i=1}^{n} \|\mathcal{R}_i\mathbf{D}_{\mathcal{T}}\|_{\infty}^{0}$, where $\mathcal{R}_i$ extracts the i^{th} *row* of the matrix on its right-hand side. In words, $\|\mathbf{D}_{\mathcal{T}}\|_{\infty}^{0}$ simply counts the number of non-zero rows of $\mathbf{D}_{\mathcal{T}}$. With it, we now define the following property:

Definition 27. Non Vanishing Support (N.V.S.):
A sparse vector $\boldsymbol{\gamma}$ with support $\mathcal{T}$ satisfies the the N.V.S property for a given dictionary $\mathbf{D}$ if

$$\|\mathbf{D}\boldsymbol{\gamma}\|_0 = \|\mathbf{D}_{\mathcal{T}}\|_{\infty}^{0}.$$

Intuitively, the above property implies that the entries in $\boldsymbol{\gamma}$ will not cause two or more atoms to be combined in such a way that (any entry of) their supports cancel each other. Notice that this is a very natural assumption to make. Because our derivations will follow a worse-case and deterministic analysis, however, we will need this property to formulate recovery guarantees for pursuit algorithms. Alternatively, one could assume the non-zero entries from $\boldsymbol{\gamma}$ to be Gaussian distributed, and in this case the N.V.S. property holds almost surely.

A direct consequence of the above property is that of maximal cardinality of representations. If $\boldsymbol{\gamma}$ satisfies the the N.V.S property for a dictionary $\mathbf{D}$, and $\bar{\boldsymbol{\gamma}}$ is another sparse vector with equal support (i.e., $Supp(\boldsymbol{\gamma}) = Supp(\bar{\boldsymbol{\gamma}})$), then necessarily $Supp(\mathbf{D}\bar{\boldsymbol{\gamma}}) \subseteq Supp(\mathbf{D}\boldsymbol{\gamma})$, and thus $\|\mathbf{D}\boldsymbol{\gamma}\|_0 \geq \|\mathbf{D}\bar{\boldsymbol{\gamma}}\|_0$. This follows from the fact that the number of non-zeros in $\mathbf{D}\bar{\boldsymbol{\gamma}}$ cannot be greater than the sum of non-zero rows from the set of atoms, $\mathbf{D}_{\mathcal{T}}$.

The second element concerns the local stability of the Stripe-RIP, the convolutional version of the Restricted Isometric Property [CT05]. As defined in [PSE17b], a convolutional dictionary $\mathbf{D}$ satisfies the Stripe-RIP condition with constant δ_k if, for every $\boldsymbol{\gamma}$ such that $\|\boldsymbol{\gamma}\|_{0,\infty}^{s} = k$,

$$(1-\delta_k)\|\boldsymbol{\gamma}\|_2^2 \leq \|\mathbf{D}\boldsymbol{\gamma}\|_2^2 \leq (1+\delta_k)\|\boldsymbol{\gamma}\|_2^2. \tag{6.3}$$

The S-RIP bounds the maximal change in (global) energy of a $\ell_{0,\infty}$-sparse vector when multiplied by a convolutional dictionary. We would like to establish an equivalent property but in a local sense. Recall the $\|\mathbf{x}\|_{2,\infty}^{p}$ norm, given by the maximal norm of a *patch* from $\mathbf{x}$, i.e. $\|\mathbf{x}\|_{2,\infty}^{p} = \max_i \|\mathbf{P}_i\mathbf{x}\|_2$. Analogously, one can consider $\|\boldsymbol{\gamma}\|_{2,\infty}^{s} = \max_i \|\mathbf{S}_i\boldsymbol{\gamma}\|_2$ to be the maximal norm of a *stripe* from $\boldsymbol{\gamma}$.

Now, is $\|\mathbf{D}\boldsymbol{\gamma}\|_{2,\infty}^{p}$ nearly isometric? The (partially affirmative) answer is given in the form of the following Lemma, which we prove in Appendix 6.7.3.

Lemma 6.3.1. *Local one-sided near isometry property:*
If $\mathbf{D}$ *is a convolutional dictionary satisfying the Stripe-RIP condition in* (6.3) *with constant* δ_k*, then*

$$\|\mathbf{D}\boldsymbol{\gamma}\|_{2,\infty}^{2,p} \leq (1+\delta_k)\ \|\boldsymbol{\gamma}\|_{2,\infty}^{2,s}.$$

This result is worthy in its own right, as it shows for the first time that not only the CSC model is globally stable for $\ell_{0,\infty}$-sparse signals, but that one can also bound the change in energy in a local sense (in terms of the $\ell_{2,\infty}$ norm). On the other hand, this property states nothing unexpected: if the CSC model is fully described by a shift-invariant local model, then its properties should be able to be characterized in a local manner as well. Lastly, while the above Lemma only refers to the upper bound of $\|\mathbf{D}\boldsymbol{\gamma}\|_{2,\infty}^{2,p}$, we conjecture that an analogous lower bound can be shown to hold as well. The respective proof is more involved, however, and a matter of current work.

With these elements, we can now move to the stability of the solutions provided by Algorithm 6.1:

Theorem 28. Stable recovery of the Multi-Layer Pursuit Algorithm in the convex relaxation case:
Suppose a signal $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$ is contaminated with locally-bounded noise $\mathbf{v}$, resulting in $\mathbf{y} = \mathbf{x} + \mathbf{v}$, $\|\mathbf{v}\|_{2,\infty}^{p} \leq \epsilon_0$. Assume that all representations $\boldsymbol{\gamma}_i$ satisfy the N.V.S. property for the respective dictionaries $\mathbf{D}_i$, and that $\|\boldsymbol{\gamma}_i\|_{0,\infty}^{s} = \lambda_i < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right)$, for $1 \leq i \leq L$ and

$\|\boldsymbol{\gamma}_L\|_{0,\infty}^s = \lambda_L \leq \frac{1}{3}\left(1+\frac{1}{\mu(\mathbf{D}^{(L)})}\right)$. Consider solving the Pursuit stage in Algorithm 6.1 as

$$\hat{\boldsymbol{\gamma}}_L = \arg\min_{\boldsymbol{\gamma}} \|\mathbf{y} + \mathbf{D}^{(L)}\boldsymbol{\gamma}\|_2^2 + \zeta_L\|\boldsymbol{\gamma}\|_1,$$

for $\zeta_L = 4\epsilon_0$, and set $\hat{\boldsymbol{\gamma}}_{i-1} = \mathbf{D}_i\hat{\boldsymbol{\gamma}}_i$, $i = L,\ldots,1$. Then, for every $1 \leq i \leq L$ layer,

1. $Supp(\hat{\boldsymbol{\gamma}}_i) \subseteq Supp(\boldsymbol{\gamma}_i)$,
2. $\|\hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i\|_{2,\infty}^p \leq \epsilon_L \prod_{j=i+1}^{L} \sqrt{\frac{3c_j}{2}}$,

where $\epsilon_L = \frac{15}{2}\ \epsilon_0\sqrt{\|\boldsymbol{\gamma}_L\|_{0,\infty}^p}$ is the error at the last layer, and c_j is a coefficient that depends on the ratio between the local dimensions of the layers, $c_j = \left\lceil \frac{2n_{j-1}-1}{n_j} \right\rceil$.

Theorem 29. Stable recovery of the Multi-Layer Pursuit Algorithm in the greedy case: Suppose a signal $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$ is contaminated with energy-bounded noise $\mathbf{v}$, such that $\mathbf{y} = \mathbf{x}+\mathbf{v}$, $\|\mathbf{y}-\mathbf{x}\|_2 \leq \mathcal{E}_0$, and $\epsilon_0 = \|\mathbf{v}\|_{2,\infty}^{\mathbf{P}}$. Assume that all representations $\boldsymbol{\gamma}_i$ satisfy the N.V.S. property for the respective dictionaries $\mathbf{D}_i$, with $\|\boldsymbol{\gamma}_i\|_{0,\infty}^s = \lambda_i < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right)$, for $1 \leq i \leq L$, and

$$\|\boldsymbol{\gamma}_L\|_{0,\infty}^s < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}^{(L)})}\right) - \frac{1}{\mu(\mathbf{D}^{(L)})}\cdot\frac{\epsilon_0}{|\gamma_L^{min}|}, \tag{6.4}$$

where γ_L^{min} is the minimal entry in the support of $\boldsymbol{\gamma}_L$. Consider approximating the solution to the Pursuit step in Algorithm 6.1 by running Orthogonal Matching Pursuit for $\|\boldsymbol{\gamma}_L\|_0$ iterations. Then

1. $Supp(\hat{\boldsymbol{\gamma}}_i) \subseteq Supp(\boldsymbol{\gamma}_i)$,
2. $\|\hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i\|_2^2 \leq \frac{\mathcal{E}_0^2}{1-\mu(\mathbf{D}^{(L)})(\|\boldsymbol{\gamma}_L\|_{0,\infty}^s-1)}\left(\frac{3}{2}\right)^{L-i}$.

The proofs of both Theorems 28 and 29 are included in Appendix 6.7.4 and 6.7.4, respectively. The coefficient c_j refers to the ratio between the filter dimensions at consecutive layers, and assuming $n_i \approx n_{i+1}$ (which indeed happens in practice), this coefficient is roughly 2. Importantly, and unlike the bounds provided for the layer-wise pursuit algorithm, the recovery guarantees are the tightest for the inner-most layer, and the bound increases slightly towards shallower representations. The relaxation to the ℓ_1 norm, in the case of the BP formulation, provides local error bounds, while the guarantees for the greedy version, in its OMP implementation, yield a global bound on the representation error.

These results show the flavor of theoretical claims that can be obtained for the proposed ML-CSC Pursuit Algorithm. By employing similar derivations to those detailed in the respective proofs, one could – in principle – provide recovery claims for other versions of this method, by employing other sparse coding strategies.

Algorithm 6.2 ML-CSC Projection Algorithm

Init: $\mathbf{x}^* = \mathbf{0}$ **for** $k = 1 : \lambda_L$ **do**
  $\hat{\boldsymbol{\gamma}}_L \leftarrow \text{OMP}(\mathbf{y}, \mathbf{D}^{(L)}, k)$ **for** $j = L : -1 : 1$ **do**
    $\hat{\boldsymbol{\gamma}}_{j-1} \leftarrow \mathbf{D}_j \hat{\boldsymbol{\gamma}}_j$
  end
  if $\|\hat{\boldsymbol{\gamma}}_i\|_{0,\infty}^s > \lambda_i$ *for any* $1 \leq i < L$ **then**
    break
  end
  else
    $\mathbf{x}^* \leftarrow \mathbf{D}^{(i)} \hat{\boldsymbol{\gamma}}_i$
  end
end
return $\mathbf{x}^*$

6.3.4 Projecting General Signals

In the most general case, i.e. removing the assumption that $\mathbf{y}$ is close enough to a signal in the model, Algorithm 6.1 by itself might not solve $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$. Consider we are given a general signal $\mathbf{y}$ and a model $\mathcal{M}_{\boldsymbol{\lambda}}$, and we run the ML-CSC Pursuit with $k = \lambda_L$ obtaining a set of representations $\{\hat{\boldsymbol{\gamma}}_j\}$. Clearly $\|\hat{\boldsymbol{\gamma}}_L\|_{0,\infty}^s \leq \lambda_L$. Yet, nothing guarantees that $\|\hat{\boldsymbol{\gamma}}_i\|_{0,\infty}^s \leq \lambda_i$ for $i < L$. In other words, in order to solve $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ one must guarantee that all sparsity constraints are satisfied.

Algorithm 6.2 progressively recovers sparse representations to provide a projection for any general signal $\mathbf{y}$. The solution is initialized with the zero vector, and then the OMP algorithm is applied with a progressively larger $\ell_{0,\infty}$ constraint on the deepest representation[4], from 1 to λ_L. The only modification required to run the OMP in this setting is to check at every iteration the value of $\|\hat{\boldsymbol{\gamma}}_L\|_{0,\infty}^s$, and to stop accordingly. At each step, given the estimated $\hat{\boldsymbol{\gamma}}_L$, the intermediate features and their $\ell_{0,\infty}$ norms, are computed. If all sparsity constraints are satisfied, then the algorithm proceeds. If, on the other hand, any of the constraints is violated, the previously computed $\mathbf{x}^*$ is reported as the solution.

This algorithm can be shown to be a greedy approximation to an optimal projection algorithm, under certain assumptions, and we now provide a sketch of the proof of this claim. Consider the first iteration of the above method, where $k = 1$. If OMP succeeds in providing the closest $\hat{\boldsymbol{\gamma}}_L$ subject to the respective constraint, i.e. providing the solution to

$$\min_{\boldsymbol{\gamma}} \|\mathbf{y} - \mathbf{D}^{(L)}\boldsymbol{\gamma}\|_2^2 \text{ s.t. } \|\boldsymbol{\gamma}\|_{0,\infty}^s \leq 1,$$

and if $\|\hat{\boldsymbol{\gamma}}_i\|_{0,\infty}^s \leq \lambda_i$ for every i, then this solution effectively provides the closest signal to $\mathbf{y}$ in the model defined by $\boldsymbol{\lambda} = [\lambda_1, \ldots, 1]$. If $\lambda_L = 1$, we are done. Otherwise, if $\lambda_L > 1$, we might increase the number of non-zeros in $\hat{\boldsymbol{\gamma}}_L$, while decreasing the ℓ_2 distance to $\mathbf{y}$. This is done by continuing to the next iteration: running again OMP with the constraint $\|\hat{\boldsymbol{\gamma}}_L\|_{0,\infty}^s \leq 2$, and obtaining the respective $\hat{\boldsymbol{\gamma}}_i$.

[4] Instead of repeating the pursuit from scratch at every iteration, one might-warm start the OMP algorithm by employing current estimate, $\hat{\boldsymbol{\gamma}}_L$, as initial condition so that only new non-zeros are added.

At any k^{th} iteration, due to the nature of the OMP algorithm, $Supp(\hat{\boldsymbol{\gamma}}_L^{k-1}) \subseteq Supp(\hat{\boldsymbol{\gamma}}_L^k)$. If all estimates $\hat{\boldsymbol{\gamma}}_i$ satisfy the N.V.S. property for the respective dictionaries $\mathbf{D}_i$, then the sparsity of each $\hat{\boldsymbol{\gamma}}_i$ is non-decreasing through the iterations, $\|\hat{\boldsymbol{\gamma}}_i^{k-1}\|_{0,\infty}^s \leq \|\hat{\boldsymbol{\gamma}}_i^k\|_{0,\infty}^s$. For this reason, if an estimate $\hat{\boldsymbol{\gamma}}_L^k$ is obtained such that any of the corresponding $\ell_{0,\infty}$ constraints is violated, then necessarily one constraint will be violated at the next (or any future) iteration. Therefore, the algorithm outputs the signal corresponding to the iteration before one of the constraints was violated. A complete optimal (combinatorial) algorithm would need to retrace its steps and replace the last non-zero added to $\hat{\boldsymbol{\gamma}}_L^k$ by the second best option, and then evaluate if all constraints are met for this selection of the support. This process should be iterated, and Algorithm 6.2 provides a greedy approximation to this idea.

As a final comment on this subject, while Algorithms 6.1 and 6.2 were presented separately, they are indeed related and one could envision combining them into a single method. The distinction between them was motivated by making the derivations of our theoretical analysis easier to grasp. Nevertheless, stating further theoretical claims without the assumption of the signal $\mathbf{y}$ being close to an underlying $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$ is non-trivial, and we defer a further analysis of this case for future work.

6.3.5 Summary - Pursuit for the ML-CSC

Before proceeding, let us briefly summarize what we have introduced so far. We have defined a projection problem, $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$, seeking for the closest signal in the model $\mathcal{M}_{\boldsymbol{\lambda}}$ to the measurements $\mathbf{y}$. We have shown that if the measurements $\mathbf{y}$ are close enough to a signal in the model, i.e. $\mathbf{y} = \mathbf{x}(\boldsymbol{\gamma}_i) + \mathbf{v}$, with bounded noise $\mathbf{v}$, then the ML-CSC Pursuit in Algorithm 6.1 manages to obtain approximate solutions that are not far from these representations, by deploying either the OMP or the BP algorithms. In particular, the support of the estimated sparse vectors is guaranteed to be a subset of the correct support, and so all $\hat{\boldsymbol{\gamma}}_i$ satisfy the model constraints. In doing so we have introduced the N.V.S. property, and we have proven that the CSC and ML-CSC models are locally stable. Lastly, if no prior information is known about the signal $\mathbf{y}$, we have proposed an OMP-inspired algorithm that finds the closest signal $\mathbf{x}(\boldsymbol{\gamma}_i)$ to any measurements $\mathbf{y}$ by gradually increasing the support of all representations $\hat{\boldsymbol{\gamma}}_i$ while guaranteeing that the model constraints are satisfied.

We now move to the next major difficulty with the ML-CSC model: studying the convolutional filters and the need to learn its parameters.

6.4 Learning the model

6.4.1 Preliminaries

The entire analysis presented so far relies on the assumption of the existence of proper dictionaries $\mathbf{D}_i$ allowing for corresponding *nested sparse features* $\boldsymbol{\gamma}_i$. Clearly, the ability to obtain such representations greatly depends on the design and properties of these dictionaries.

While in the traditional sparse modeling scenario certain analytically-defined dictionaries (such as the Discrete Cosine Transform) often perform well in practice, in the ML-CSC case it is hard to propose an off-the-shelf construction which would allow for any meaningful decompositions. To see this more clearly, consider obtaining $\hat{\boldsymbol{\gamma}}_L$ with Algorithm 6.1 removing all other assumptions on the dictionaries $\mathbf{D}_i$. In this case, nothing will prevent $\hat{\boldsymbol{\gamma}}_{L-1} = \mathbf{D}_L\hat{\boldsymbol{\gamma}}_L$ from being dense. While one could argue that this is an artifact of the presented algorithm (for instance, for not explicitly enforcing both representations to be sparse), nothing guarantees that *any* collection of dictionaries would allow for any signal with nested sparse components $\boldsymbol{\gamma}_i$. In other words, how do we know if the model represented by $\{\mathbf{D}_i\}_{i=1}^L$ is not empty?

To illustrate this important point, consider the case where $\mathbf{D}_i$ are random – a popular construction in other sparsity-related applications. In this case, every atom from the dictionary $\mathbf{D}_L$ will be a random variable $\mathbf{d}_L^j \sim \mathcal{N}(\mathbf{0}, \sigma_L^2\mathbf{I})$. In this case, one can indeed construct $\boldsymbol{\gamma}_L$, with $\|\boldsymbol{\gamma}_L\|_{0,\infty}^s \leq 2$, such that *every entry* from $\boldsymbol{\gamma}_{L-1} = \mathbf{D}_L\boldsymbol{\gamma}_L$ will be a random variable $\gamma_{L-1}^j \sim \mathcal{N}(0, \sigma_L^2)$, $\forall\, j$. Thus, $\Pr\left(\gamma_{L-1}^j = 0\right) = 0$. As we see, there will not exist any sparse (or dense, for that matter) $\boldsymbol{\gamma}_L$ which will create a sparse $\boldsymbol{\gamma}_{L-1}$. In other words, for this choice of dictionaries, the ML-CSC model is empty.

6.4.2 Sparse Dictionaries

From the discussion above one can conclude that one of the key components of the ML-CSC model is sparse dictionaries: if both $\boldsymbol{\gamma}_L$ and $\boldsymbol{\gamma}_{L-1} = \mathbf{D}_L\boldsymbol{\gamma}_L$ are sparse, then atoms in $\mathbf{D}$ must indeed contain only a few non-zeros. We make this observation concrete in the following lemma.

Lemma 6.4.1. *Dictionary Sparsity Condition*
Consider the ML-CSC model $\mathcal{M}_{\boldsymbol{\lambda}}$ described by the the dictionaries $\{\mathbf{D}_i\}_{i=1}^L$ and the layer-wise $\ell_{0,\infty}$-sparsity levels $\lambda_1, \lambda_2, \ldots, \lambda_L$. Given $\boldsymbol{\gamma}_L : \|\boldsymbol{\gamma}_L\|_{0,\infty}^s \leq \lambda_L$ and constants $c_i = \left\lceil \frac{2n_{i-1}-1}{n_i} \right\rceil$, the signal $\mathbf{x} = \mathbf{D}^{(L)}\boldsymbol{\gamma}_L \in \mathcal{M}_{\boldsymbol{\lambda}}$ if

$$\|\mathbf{D}_i\|_0 \leq \frac{\lambda_{i-1}}{\lambda_i c_i}, \quad \forall\ 1 < i \leq L.$$

The simple proof of this Lemma is included in Appendix 6.7.5. Notably, while this claim does not tell us if a certain model is empty, it does guarantee that if the dictionaries satisfy a given sparsity constraint, one can simply sample from the model by drawing the inner-most representations such that $\|\boldsymbol{\gamma}_L\|_{0,\infty}^s \leq \lambda_L$. One question remains: how do we train such dictionaries from real data?

6.4.3 Learning Formulation

One can understand from the previous discussion that there is no hope in solving the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem for real signals without also addressing the learning of dictionaries $\mathbf{D}_i$ that would allow for the respective representations. To this end, considering the scenario where one is given a collection of K training signals, $\{\mathbf{y}^k\}_{k=1}^K$, we upgrade the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem to a learning setting in

the following way:

$$\min_{\{\boldsymbol{\gamma}_i^k\},\{\mathbf{D}_i\}} \quad \sum_{k=1}^{K} \|\mathbf{y}^k - \mathbf{x}^k(\boldsymbol{\gamma}_i^k, \mathbf{D}_i)\|_2^2 \quad \text{s.t.} \begin{cases} \mathbf{x}^k \in \mathcal{M}_{\boldsymbol{\lambda}}, \\ \|\mathbf{d}_i^j\|_2 = 1, \forall\ i, j \end{cases} \tag{6.5}$$

We have included the constraint of every dictionary atom, of every level, to have a unit norm to prevent arbitrarily small coefficients in the representations $\boldsymbol{\gamma}_i^k$. This formulation, while complete, is difficult to address directly. To begin with, the constraints on the representations $\boldsymbol{\gamma}_i$ are coupled, just as in the pursuit problem discussed in the previous section. In addition, the sparse representations now also depend on the variables $\mathbf{D}_i$. In what follows, we provide a relaxation of this cost function that will result in a simple learning algorithm.

The problem above can also be understood from the perspective of minimizing the number of non-zeros in the representations at every layer, subject to an error threshold – a typical reformulation of sparse coding problems. Our main observation arise from the fact that, since $\boldsymbol{\gamma}_{L-1}$ is function of both $\mathbf{D}_L$ and $\boldsymbol{\gamma}_L$, one can upper-bound the number of non-zeros in $\boldsymbol{\gamma}_{L-1}$ by that of $\boldsymbol{\gamma}_L$. More precisely,

$$\|\boldsymbol{\gamma}_{L-1}\|_{0,\infty}^s \leq c_L \|\mathbf{D}_L\|_0 \|\boldsymbol{\gamma}_L\|_{0,\infty}^s,$$

where c_L is a constant[5]. Therefore, instead of minimizing the number of non-zeros in $\boldsymbol{\gamma}_{L-1}$, we can address the minimization of its upper bound by minimizing both $\|\boldsymbol{\gamma}_L\|_{0,\infty}^s$ and $\|\mathbf{D}_L\|_0$. This argument can be extended to any layer, and we can generally write

$$\|\boldsymbol{\gamma}_i\|_{0,\infty}^s \ \leq \ c \prod_{j=i+1}^{L} \|\mathbf{D}_j\|_0 \|\boldsymbol{\gamma}_L\|_{0,\infty}^s.$$

In this way, minimizing the sparsity of any i^{th} representation can be done implicitly by minimizing the sparsity of the last layer *and* the number of non-zeros in the dictionaries from layer $(i+1)$ to L. Put differently, the sparsity of the intermediate convolutional dictionaries serve as proxies for the sparsity of the respective representation vectors. Following this observation, we now recast the problem in Equation (6.5) into the following Multi-Layer Convolutional Dictionary Learning Problem:

$$\min_{\{\boldsymbol{\gamma}_L^k\},\{\mathbf{D}_i\}} \sum_{k=1}^{K} \|\mathbf{y}^k - \mathbf{D}_1\mathbf{D}_2 \dots \mathbf{D}_L\boldsymbol{\gamma}_L^k\|_2^2 + \sum_{i=2}^{L} \zeta_i \|\mathbf{D}_i\|_0 \quad \text{s.t.} \begin{cases} \|\boldsymbol{\gamma}_L^k\|_{0,\infty}^s \leq \lambda_L, \\ \|\mathbf{d}_i^j\|_2 = 1, \forall\ i, j \end{cases} \tag{6.6}$$

Under this formulation, this problem seeks for sparse representations $\boldsymbol{\gamma}_L^k$ for each example $\mathbf{y}^k$, while forcing the intermediate convolutional dictionaries (from layer 2 to L) to be sparse. The reconstructed signal, $\mathbf{x} = \mathbf{D}_1\boldsymbol{\gamma}_1$, is not expected to be sparse, and so there is no reason to enforce this property on $\mathbf{D}_1$. Note that there is now only one sparse coding process involved –

[5]From [PRE16], we have that $\|\boldsymbol{\gamma}_{L-1}\|_{0,\infty}^p \leq \|\mathbf{D}_L\|_0 \|\boldsymbol{\gamma}_L\|_{0,\infty}^s$. From here, and denoting by c_L the upper-bound on the number of patches in a stripe from $\boldsymbol{\gamma}_{L-1}$ given by $c_L = \left\lceil \frac{2n_{L-1}-1}{n_L} \right\rceil$, we can obtain a bound to $\|\boldsymbol{\gamma}_{L-1}\|_{0,\infty}^s$.

that of $\boldsymbol{\gamma}_L^k$ – while the intermediate representations are never computed explicitly. Recalling the theoretical results from the previous section, this is in fact convenient as one only has to estimate the representation for which the recovery bound is the tightest.

Following the theoretical guarantees presented in Section 6.3, one can alternatively replace the $\ell_{0,\infty}$ constraint on the deepest representation by a convex ℓ_1 alternative. The resulting formulation resembles the lasso formulation of the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem, for which we have presented theoretical guarantees in Theorem 28. In addition, we replace the constraint on the ℓ_2 of the dictionary atoms by an appropriate penalty term, recasting the above problem into a simpler (unconstrained) form:

$$\min_{\{\boldsymbol{\gamma}_L^k\},\{\mathbf{D}_i\}} \sum_{k=1}^{K} \|\mathbf{y}^k - \mathbf{D}_1\mathbf{D}_2\dots\mathbf{D}_L\boldsymbol{\gamma}_L^k\|_2^2 + \iota\sum_{i=1}^{L}\|\mathbf{D}_i\|_F^2 + \sum_{i=2}^{L}\zeta_i\|\mathbf{D}_i\|_0 + \lambda\|\boldsymbol{\gamma}_L^k\|_1. \tag{6.7}$$

The problem in Equation (6.7) is highly non-convex, due to the ℓ_0 terms and the product of the factors. In what follows, we present an online alternating minimization algorithm, based on stochastic gradient descent, which seeks for the deepest representation $\boldsymbol{\gamma}_L$ and then progressively updates the layer-wise convolutional dictionaries.

For each incoming sample $\mathbf{y}^k$ (or potentially, a mini-batch), we will first seek for its deepest representation $\boldsymbol{\gamma}_L^k$ considering the dictionaries fixed. This is nothing but the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem in (6.1), which was analyzed in detail in the previous sections, and its solution will be approximated through iterative shrinkage algorithms. In particular, we employ an efficient implementation of the FISTA algorithm [BT09b]. Also, one should keep in mind that while representing each dictionary by $\mathbf{D}_i$ is convenient in terms of notation, these matrices are never computed explicitly – which would be prohibitive. Instead, these dictionaries (or their transpose) are applied effectively through convolution operators, common in the deep learning community. In addition, these operators are expected to be very efficient to apply due to their high sparsity, and one could in principle benefit from specific libraries to boost performance in this case, such as the one in [LWF+15].

Given the obtained $\boldsymbol{\gamma}_L^k$, we then seek to update the respective dictionaries. As it is posed – with a global ℓ_0 norm over each dictionary – this is nothing but a generalized pursuit as well. Therefore, for each dictionary $\mathbf{D}_i$, we minimize the function in Problem (6.7) by applying T iterations of projected gradient descent. This is done by computing the gradient of the ℓ_2 terms in Problem (6.7) (call it $f(\mathbf{D}_i)$) with respect to a each dictionary $\mathbf{D}_i$ (i.e., $\nabla f(\mathbf{D}_i)$), making a gradient step and then applying a hard-thresholding operation, $\mathcal{H}_{\zeta_i}(\cdot)$, depending on the parameter ζ_i. This is simply an instance of the Iterative Hard Thresholding algorithm [BD08]. In addition, the computation of $\nabla f(\mathbf{D}_i)$ involves only multiplications the convolutional dictionaries for the different layers. The overall algorithm is depicted in Algorithm 6.3, and we will expand on further implementation details in the results section.

The parameters of the models involve the ℓ_1 penalty of the deepest representation, i.e. λ, and the parameter for each dictionary, ζ_i. The first parameter can be set manually or determined so as to obtain a given given representation error. On the other hand, the dictionary-wise ζ_i

Algorithm 6.3 Multi-Layer Convolutional Dictionary Learning

Data: Training samples $\{\mathbf{y}_k\}_{k=1}^K$, initial convolutional dictionaries $\{\mathbf{D}_i\}_{i=1}^L$
for $k = 1, \ldots, K$ **do**
  Draw $\mathbf{y}_k$ at random
  Sparse Coding: $\boldsymbol{\gamma}_L \leftarrow \arg\min_{\boldsymbol{\gamma}} \|\mathbf{y}_k - \mathbf{D}^{(L)}\boldsymbol{\gamma}\|_2 + \lambda\|\boldsymbol{\gamma}\|_1$
  Update Dictonaries:
  for $i = L, \ldots, 1$ **do**
    for t = *1,...,T* **do**
      $\mathbf{D}_i^{t+1} \leftarrow \mathcal{H}_{\zeta_i}\left[\mathbf{D}_i^t - \eta\nabla f(\mathbf{D}_i^t)\right]$
    end
  end
  for t = *1,...,T* **do**
    $\mathbf{D}_1^{t+1} \leftarrow \mathbf{D}_1^t - \eta\nabla f(\mathbf{D}_1^t)$
  end
end

parameters are less intuitive to establish, and the question of how to set these values for a given learning scenario remains a subject of current research. Nevertheless, we will show in the experimental section that setting these manually results in effective constructions.

As a final comment, note this approach can also be employed to minimize Problem (6.6) by introducing minor modifications: In the sparse coding stage, the Lasso is replaced by a $\ell_{0,\infty}$ pursuit, which can be tackled with a greedy alternative as the OMP (as described in Theorem 29) or by an Iterative Hard Thresholding alternative [BD08]. In addition, one could consider employing the ℓ_1 norm as a surrogate for the ℓ_0 penalty imposed on the dictionaries. In this case, their update can still be performed by the same projected gradient descent approach, though replacing the hard thresholding with its soft counterpart.

6.4.4 Connection to related works

Naturally, the proposed algorithm has tight connections to several recent dictionary learning approaches. For instance, our learning formulation is closely related to the Chasing Butterflies work [LMG15], and our resulting algorithm is very similar of the PALM method employed by the authors, initially proposed in [BST14]. However, their approach is designed for general (not convolutional) multi-level dictionaries, and the algorithm is particularly targeted to lower semicontinuous functions. The inspiring work of in [CMTD15], on the other hand, proposed a learning approach where the dictionary is expressed as a cascade of convolutional filters with sparse kernels, and they effectively showed how this approach can be used to approximate large-dimensional analytic atoms such as those from wavelets and curvelets. Finally, as our proposed approach effectively learns a sparse dictionary, we share some similarities with the double-sparsity work from [RZE10]. In particular, in its Trainlets version [SOZE16], the authors proposed to learn a dictionary as a sparse combination of cropped wavelets atoms. From the previous comment on the work from [CMTD15], this could also potentially be expressed as a

product of sparse convolutional atoms.

What is the connection between this learning formulation and that of deep convolutional networks? Recalling the analysis presented in [PRE16], the Forward Pass is nothing but a layered non-negative thresholding algorithm, the simplest form of a pursuit for the ML-CSC model with layer-wise deviations. Therefore, if the pursuit for $\hat{\boldsymbol{\gamma}}_L$ in our setting is solved with such an algorithm, then the problem in (6.7) *implements a convolutional neural network with only one RELU operator at the last layer, with sparse-enforcing penalties on the filters.* Moreover, due the data-fidelity term in our formulation, the proposed optimization problem provides nothing but a convolutional sparse autoencoder. As such, our work is related to the extensive literature in this topic. For instance, in [Ng11], sparsity is enforced in the hidden activation layer by employing a penalty term proportional to the KL divergence between the hidden unit marginals and a target sparsity probability. Other related works include the k-sparse autoencoders [MF13], where the hidden layer is constrained to having exactly k non-zeros. In practice, this boils down to a simple $k-$hard thresholding step of the hidden activation, and the neuron weights are updated with gradient descent. In this respect, our work can be thought of a generalization of this work, where the pursuit algorithm is more sophisticated than a simple thresholding operation, and where the filters are composed by a cascade of sparse convolutional filters. More recently, the work in [MF15] proposed the *winner-take-all* autoencoders. In a nutshell, these are non-symmetric autoencoders having a few convolutional layers (with ReLu non-linearities) as the encoder, and a simple linear decoder. Sparsity is enforced in what the authors refer to as "spatial" and a "lifetime" sparsity.

Finally, and due to the fact that our formulation effectively provides a convolutional network with sparse kernels, our approach is reminiscent of works attempting to sparsify the filters in deep learning models. For instance, the work in [LWF+15] showed that the weights of learned deep convolutional networks can be sparsified without considerable degradation of classification accuracy. Nevertheless, one should perpend the fact that these works are motivated merely by cheaper and faster implementations, whereas our model is intrinsically built by theoretically justified sparse kernels. We do not attempt to compare our approach to such sparsifying methods at this stage, and we defer this to future work.

6.5 Experiments

We now provide experimental results to demonstrate several aspects of the ML-CSC model. As a case-study, we consider the MNIST dataset [LBBH98]. We define our model as consisting of 3 convolutional layers: the first one contains 32 local filters of size 7×7 (with a stride of 2), the second one consists of 128 filters of dimensions $5 \times 5 \times 32$ (with a stride of 1), and the last one contains 1024 filters of dimensions $7 \times 7 \times 128$. At the third layer, the effective size of the atoms is 28 – representing an entire digit.

Training is performed with Algorithm 6.3, using a mini-batch of 100 samples per iteration. For the Sparse Coding stage, we leverage an efficient implementation of the FISTA [BT09b] algorithm, and we adjust the penalty parameter λ to obtain roughly 15 non-zeros in the deepest

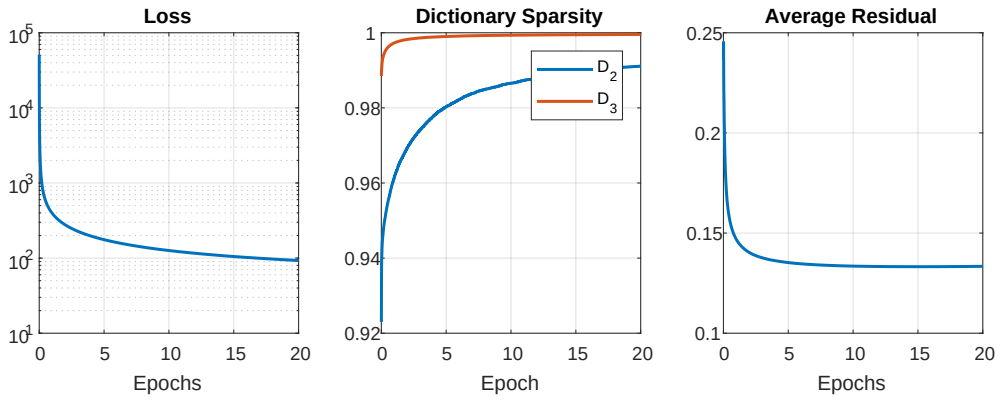


Figure 6.3: Evolution of the Loss function, sparsity of the convolutional dictionaries and average residual norm during training on the MNIST dataset.

representation $\boldsymbol{\gamma}_3$. The ζ_i parameters, the penalty parameters for the dictionaries sparsity levels, are set manually for simplicity. In addition, and as it is commonly done in various Gradient Descent methods, we employ a momentum term for the update of the dictionaries $\mathbf{D}_i$ within the projected gradient descent step in Algorithm 6.3, and set its memory parameter to 0.9. The step size is set to 1, the update dictionary iterations is set as $T = 1$, $\iota = 0.001$, and we run the algorithm for 20 epochs, which takes approximately 30 minutes. Our implementation uses the Matconvnet library, which leverages efficient functions for GPU[6].

We depict the evolution of the Loss function during training in Figure 6.3, as well as the sparsity of the second and third dictionaries and the average residual norm. The resulting model is depicted in Figure 6.4. One can see how the first layer is composed of very simple small-dimensional edges or blobs. The second dictionary, $\mathbf{D}_2$, is effectively 99% sparse, and its non-zeros combine a few atoms from $\mathbf{D}_1$ in order to create slightly more complex edges, as the ones in the effective dictionary $\mathbf{D}^{(2)}$. Lastly, $\mathbf{D}_3$ is 99.8% sparse, and it combines atoms from $\mathbf{D}^{(2)}$ in order to provide atoms that resemble different kinds (or parts) of digits. These final global atoms are nothing but a linear combination of local small edges by means of convolutional sparse kernels.

Interestingly, we have observed that the mutual coherence of the effective dictionaries do not necessarily increase with the layers, and they often decrease with the depth. While this measure relates to worst-case analysis conditions and do not mean much in the context of practical performance, one can see that the effective dictionary indeed becomes less correlated as the depth increases. This is intuitive, as very simple edges – and at every location – are expected to show large inner products, larger than the correlation of two more complex number-like structures. This effect can be partially explained by the dictionary redundancy: having 32 local filters in $\mathbf{D}_1$ (even while using a stride of 2) implies a 8-fold redundancy in the effective dictionary at this level. This redundancy decreases with the depth (at this least for the current construction),

[6]All experiments are run on a 16 i7 cores Windows station with a NVIDIA GTX 1080 Ti.

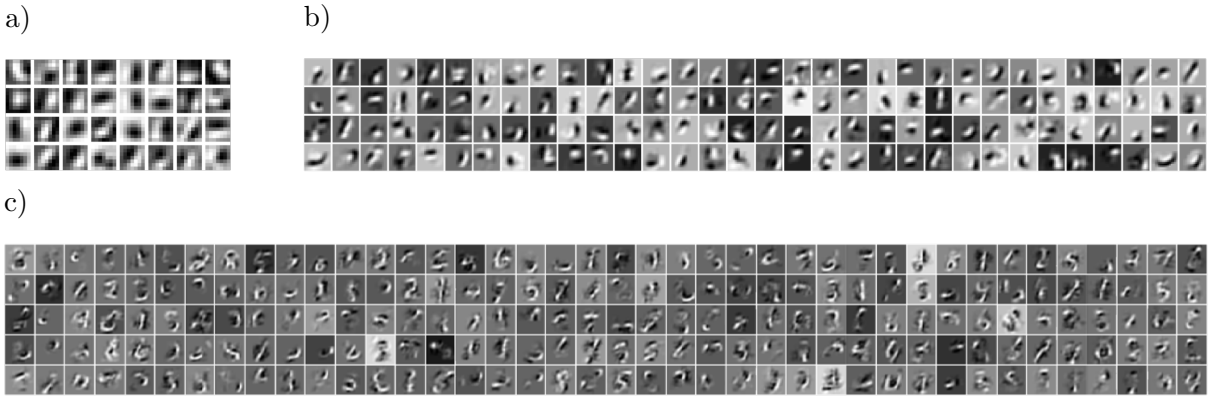


Figure 6.4: ML-CSC model trained on the MNIST dataset. a) The local filters of the dictionary $\mathbf{D}_1$. b) The local filters of the effective dictionary $\mathbf{D}^{(2)} = \mathbf{D}_1\mathbf{D}_2$. c) Some of the 1024 local atoms of the effective dictionary $\mathbf{D}^{(3)}$ which, because of the dimensions of the filters and the strides, are global atoms of size 28×28.

and at the third layer one has *merely* 1024 atoms (redundancy of about 1.3, since the signal dimension is 28^2).

We can also find the multi-layer representation for real images – essentially solving the projection problem $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$. In Figure 6.5, we depict the multi-layer features $\boldsymbol{\gamma}_i$, $i = 1, 2, 3$, obtained with the Algorithm 6.1, that approximate an image $\mathbf{y}$ (not included in the training set). Note that all the representations are notably sparse thanks to the very high sparsity of the dictionaries $\mathbf{D}_2$ and $\mathbf{D}_3$. These decompositions (any of them) provide a sparse decomposition of the number 3 at different scales, resulting in an approximation $\hat{\mathbf{x}}$. Naturally, the quality of the approximation can be improved by increasing the cardinality of the representations.

6.5.1 Sparse Recovery

The first experiment we explore is that of recovering sparse vectors from corrupted measurements, in which we will compare the presented ML-CSC Pursuit with the Layered approach from [PRE16]. For the sake of completion and understanding, we will first carry this experiment in a synthetic setting and then on projected real digits, leveraging the dictionaries obtained in the beginning of this section.

We begin by constructing a 3 layers "non-convolutional" [7] model for signals of length 200, with the dictionaries having 250, 300, and 350 atoms, respectively. The first dictionary is constructed as a random matrix, whereas the remaining ones are composed of sparse atoms with random supports and a sparsity of 99%. Finally, 500 representations are sampled by drawing sparse vectors $\boldsymbol{\gamma}_L$, with a target sample sparsity k and normally distributed coefficients. We generate the signals as $\mathbf{x} = \mathbf{D}^{(i)}\boldsymbol{\gamma}_i$, and then corrupt them with Gaussian noise ($\sigma = 0.02$)

[7] The non-convolutional case is still a ML-CSC model, in which the signal dimension is the same as the length of the atoms n, and with a stride of the same magnitude n. We choose this setting for the synthetic experiment to somewhat favor the results of the layered pursuit approach.

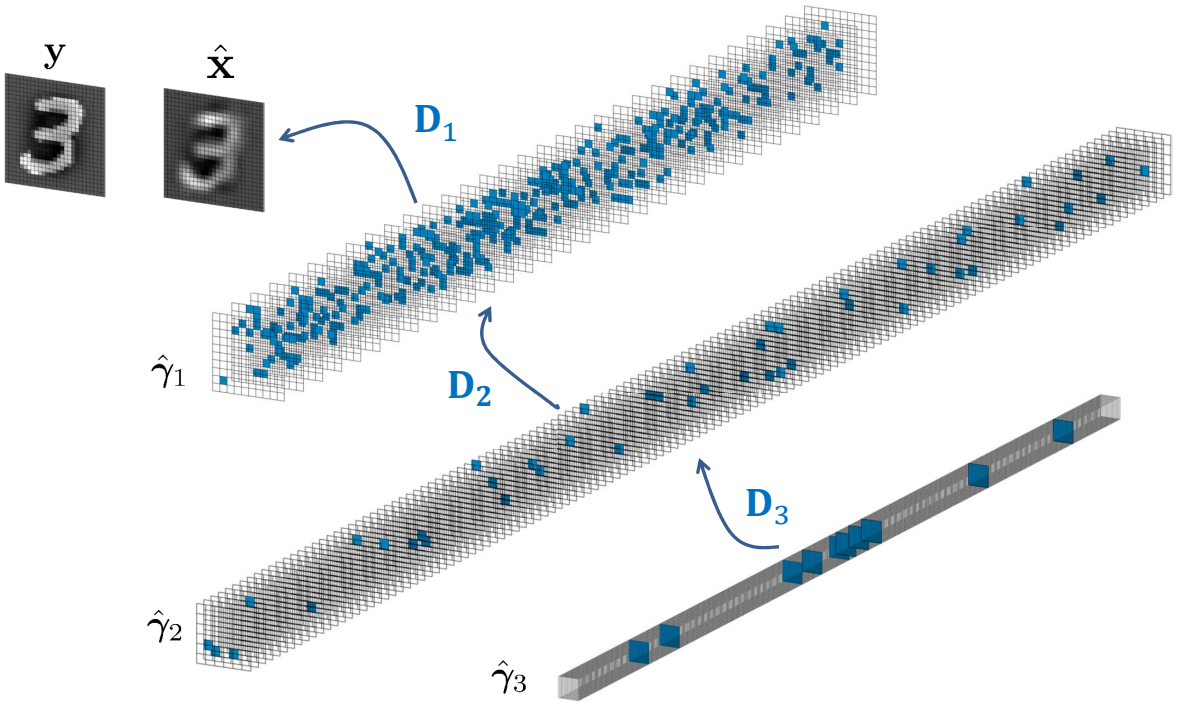


Figure 6.5: Decompositions of an image from MNIST in terms of its nested sparse features $\boldsymbol{\gamma}_i$ and multi-layer convolutional dictionaries $\mathbf{D}_i$.

obtaining the measurements $\mathbf{y} = \mathbf{x}(\boldsymbol{\gamma}_i) + \mathbf{v}$.

In order to evaluate our projection approach, we run Algorithm 6.1 employing the Subspace Pursuit algorithm [DM09] for the sparse coding step, with the oracle target cardinality k. Recall that once the deepest representations $\hat{\boldsymbol{\gamma}}_L$ have been obtained, the inner ones are simply computed as $\hat{\boldsymbol{\gamma}}_{i-1} = \mathbf{D}_i \hat{\boldsymbol{\gamma}}_i$. In the layered approach from [PRE16], on the other hand, the pursuit of the representations progresses sequentially: first running a pursuit for $\hat{\boldsymbol{\gamma}}_1$, then employing this estimate to run another pursuit for $\hat{\boldsymbol{\gamma}}_2$, etc. In the same spirit, we employ Subspace Pursuit layer by layer, employing the oracle cardinality of the representation at each stage. The results are presented in Figure 6.6: at the top we depict the relative ℓ_2 error of the recovered representations ($\|\hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i\|_2 / \|\boldsymbol{\gamma}_i\|_2$) and, at the bottom, the normalized intersection of the supports [Ela10], both as a function of the sample cardinality k and the layer depth.

The projection algorithm manages to retrieve the representations $\hat{\boldsymbol{\gamma}}_i$ more accurately than the layered pursuit, as evidenced by the ℓ_2 error and the support recovery. The main reason behind the difficulty of the layer-by-layer approach is that the entire process relies on the correct recovery of the first layer representations, $\hat{\boldsymbol{\gamma}}_1$. If these are not properly estimated (as evidenced by the bottom-left graph), there is little hope for the recovery of the deeper ones. In addition, these representations $\boldsymbol{\gamma}_1$ are the least sparse ones, and so they are expected to be the most challenging ones to recover. The projection alternative, on the other hand, relies on the estimation of the deepest $\hat{\boldsymbol{\gamma}}_L$, which are very sparse. Once these are estimated, the remaining

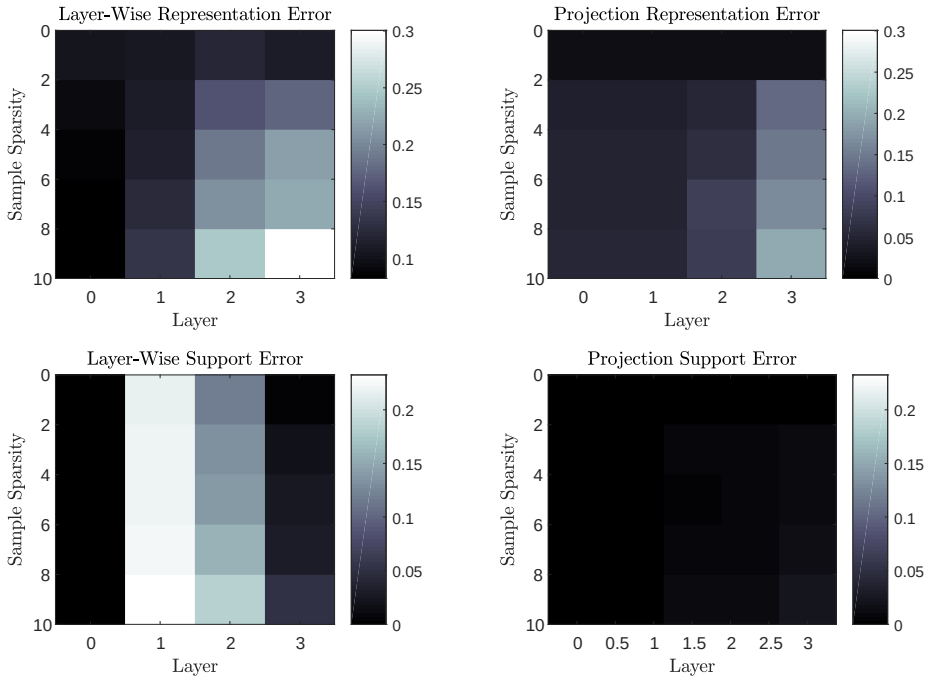


Figure 6.6: Recovery of representations from noisy synthetic signals. Top: normalized ℓ_2 error between the estimated and the true representations. Bottom: normalized intersection between the estimated and the true support of the representations.

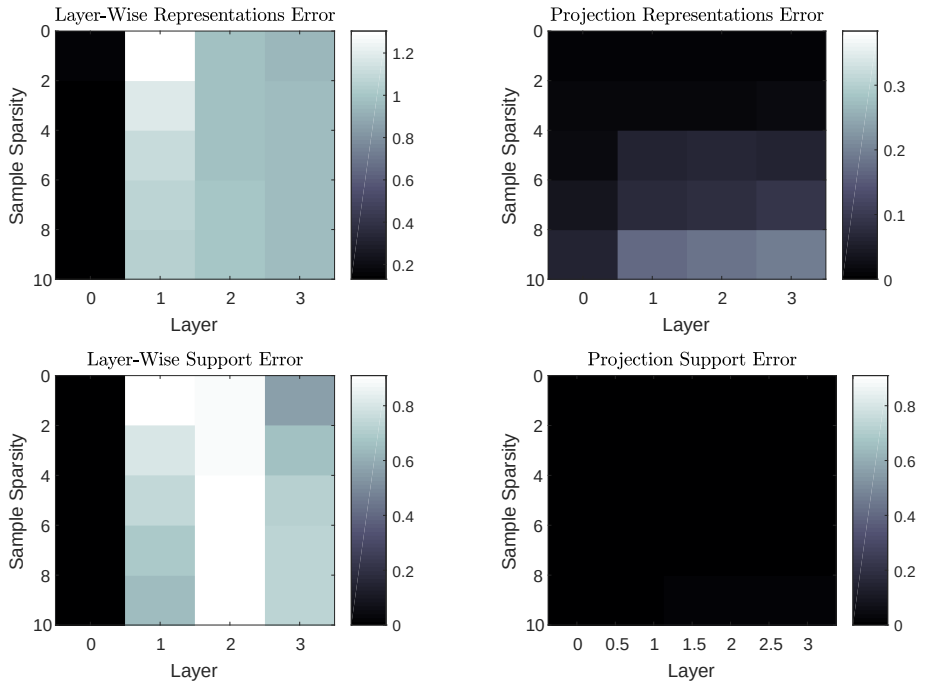


Figure 6.7: Recovery of representations from noisy MNIST digits. Top: normalized ℓ_2 error between the estimated and the true representations. Bottom: normalized intersection between the estimated and the true support of the representations.

ones are simply computed by propagating them to the shallower layers. Following our analysis in the Section 6.3.3, if the support of $\hat{\boldsymbol{\gamma}}_L$ is estimated correctly, so will be the support of the remaining representations $\hat{\boldsymbol{\gamma}}_i$.

We now turn to deploy the 3 layer convolutional dictionaries for real digits obtained previously. To this end we take 500 digits from the MNIST dataset and project them on the trained model, essentially running Algorithm 6.1 and obtaining the representations $\boldsymbol{\gamma}_i$. We then create the noisy measurements as $\mathbf{y} = \mathbf{D}^{(i)}\boldsymbol{\gamma}_i + \mathbf{v}$, where $\mathbf{v}$ is Gaussian noise with $\sigma = 0.02$, providing nothing but noisy digits. We then repeat both pursuit approaches seeking to estimate the underlying representations, obtaining the results reported in Figure 6.7.

Clearly, this represents a significantly more challenging scenario for the layered approach, which recovers only a small fraction of the correct support of the sparse vectors. The projection algorithm, on the other hand, provides accurate estimations with negligible mistakes in the estimated supports, and very low ℓ_2 error. Note that the ℓ_2 error has little significance for the Layered approach, as this algorithm does not manage to find the true supports. The reason for the significant deterioration in the performance of the Layered algorithm is that this method actually finds alternative representations $\hat{\boldsymbol{\gamma}}_1$, of the same sparsity, providing a lower fidelity term than the projection counterpart for the first layer. However, these estimates $\hat{\boldsymbol{\gamma}}_1$ do not necessarily provide a signal in the model, which causes further errors when estimating $\hat{\boldsymbol{\gamma}}_2$.

6.5.2 Sparse Approximation

A straight forward application for unsupervised learned model is that of approximation: how well can one approximate or reconstruct a signal given only a few k non-zero values from some representation? In this subsection, we study the performance of the ML-CSC model for this task while comparing with related methods, and we present the results in Figure 6.8. The model is trained on $60K$ training examples, and the M-term approximation is measured on the remaining $10K$ testing samples. All of the models are designed with 1K hidden units (or atoms).

Given the close connection of the ML-CSC model to sparse auto-encoders, we present the results obtained by approximating the signals with sparse autoencoders [Ng11] and k-sparse autoencoders [MF13]. In particular, the work in [Ng11] trains sparse auto-encoders by penalizing the KL divergence between the activation distribution of the hidden neurons and that of a binomial distribution with a certain target activation rate. As such, the resulting activations are never truly sparse. For this reason, since the M-term approximation is computed by picking the highest entries in the hidden neurons and setting the remaining ones to zero, this method exhibits a considerable representation error.

K-sparse auto-encoders perform significantly better, though they are sensitive to the number of non-zeros used during training. Indeed, if the model is trained with 25 non-zeros per sample, the model performs well for a similar range of cardinalities. Despite this sensitivity on training, their performance is remarkable considering the simplicity of the pursuit involved: the reconstruction is done by computing $\hat{\mathbf{x}} = \mathbf{W}\hat{\boldsymbol{\gamma}}_k + \mathrm{b}'$, where $\hat{\boldsymbol{\gamma}}_k$ is a k-sparse activation (or feature) obtained by hard thresholding as $\hat{\boldsymbol{\gamma}}_k = H_k\left[\mathbf{W}^T\mathbf{y} + \mathrm{b}\right]$, and where b and b$'$ are biases

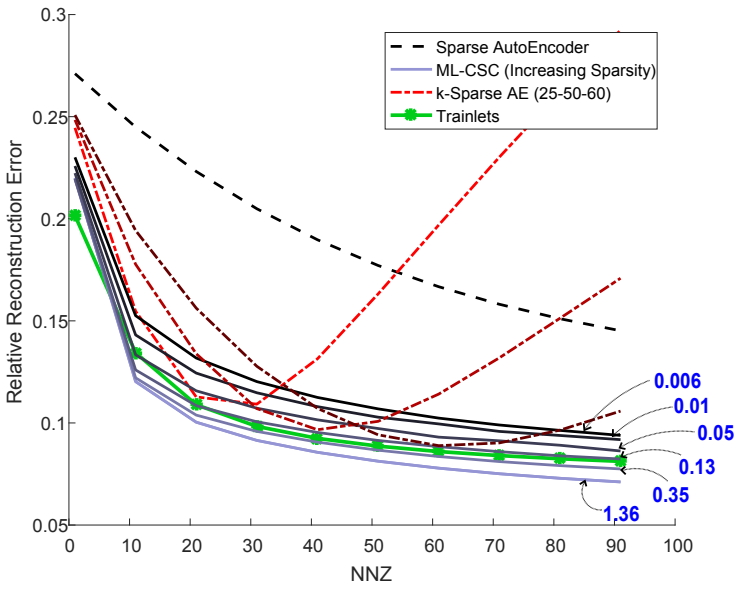


Figure 6.8: M-term approximation as a function of non-zero coefficients (NNZ) for MNIST digits, comparing sparse autoencoders [Ng11], k-sparse autoencoders [MF13], trainlets (OSDL) [SOZE16], and the proposed ML-CSC for models with different filter sparsity levels. The relative number of parameters is depicted in NavyBlue.

vectors. Note that while a convolutional multi-layer version of this family of autoencoders was proposed in [MF15], these constructions are trained in stacked manner – i.e., training the first layer independently, then training the second one to represent the features of the first layer while introducing pooling operations, and so forth. In this manner, each layer is trained to represent the (pooled) features from the previous layer, but the entire architecture cannot be trivially employed for comparison in this problem.

Regarding the ML-CSC, we trained 6 different models by enforcing 6 different levels of sparsity in the convolutional filters (i.e., different values of the parameters ζ_i in Algorithm 6.3), with a fixed target sparsity of $k = 10$ non-zeros. The sparse coding of the inner-most $\hat{\gamma}_3$ was done with the Iterative Hard Thresholding algorithm, in order to guarantee an exact number of non-zeros. The numbers pointing at the different models indicate the relative amount of parameters in the model, where 1 corresponds to $28^2 \times 1K$ parameters required in a standard autoencoder (this is also the number of parameters in the sparse-autoencoders and k-sparse autoencoders, without counting the biases). As one can see, the larger the number of parameters, the lower the representation error the model is able to provide. In particular, the ML-CSC yields slightly better representation error than that of k-sparse autoencoders, for a wide range of non-zero values (without the need to train different models for each one) and *with 1 and 2 orders of magnitude less parameters*.

Since the training of the ML-CSC model can also be understood as a dictionary learning algorithm, we compare here with the state-of-the-art method of [SOZE16]. For this case, we

Method	Classification Error
Stacked Denoising Autoencoder (3 layers) [VLL$^+$10]	1.28%
k-Sparse Autoencoder (1K units) [MF13]	1.35%
Shallow WTA Autoencoder (2K units) [MF15]	1.20%
Stacked WTA Autoencoder (2K units) [MF15]	1.11%
ML-CSC (1K units) - 2nd Layer Rep.	1.30%
ML-CSC (2K units) - 2nd&3rd Layer Rep.	1.15%

Table 6.1: Unsupervised classification results on MNIST.

trained 1K trainlet atoms with the OSDL algorithm. Note that this comparison is interesting, as OSDL also provides sparse atoms with reduced number of parameters. For the sake of comparison, we employed an atom-sparsity that results in 13% of parameters relative to the total model size (just as one of the trained ML-CSC models), and the sparse coding was done also with the IHT algorithm. Notably, the performance of this relatively sophisticated dictionary learning method, which leverages the representation power of a cropped wavelets base dictionary, is only slightly superior to the proposed ML-CSC.

6.5.3 Unsupervised Classification

Unsupervised trained models are usually employed as feature extractors, and a popular way to assess the quality of such features is to train a linear classifier on them for a certain classification task. To this end, we train a model with 3 layers, each containing: 16 (5×5) atoms, 64 ($5\times5\times16$) atoms and 1024 atoms of dimension $5\times5\times64$ (stride of 2) on 60K training samples from MNIST. Just as for the previous model, the global sparse coding is performed with FISTA and a target (average) sparsity of 25 non-zeros. Once trained, we compute the representations $\hat{\gamma}_i$ with an elastic net formulation and non-negativity constraints, before fitting a simple linear classifier on the obtained features. Employing an elastic-net formulation (by including an ℓ_2 regularization parameter, in addition to the ℓ_1 norm) results in slightly denser representations, with improved classification performance. Similarly, the non-negativity constraint significantly facilitates the classification by linear classifiers. We compare our results with similar methods under the same experimental setup, and we depict the results in Table 6.1, reporting the classification error on the 10K testing samples.

Recall that within the ML-CSC model, all features $\boldsymbol{\gamma}_i$ have a very clear meaning: they provide a sparse representation at a different layer. We can leverage this multi-layer decomposition in a very natural way within this unsupervised classification framework. We detail the classification performance achieved by our model in two different scenarios: on the first one we employ the 1K-dimensional features corresponding to the second layer of the ML-CSC model, obtaining better performance than the equivalent k-sparse autoencoder. In the second case, we add to the previous features the 1K-dimensional features from the third layer, resulting in a classification error of 1.15%, comparable to the Stacked Winner Take All (WTA) autoencoder (with the same number of neurons).

Lastly, it is worth mentioning that a stacked version of convolutional WTA autoencoder [MF15] achieve a classification error of 0.48, which provide significantly better results. However, note that this model is trained with a 2-stage process (training the layers separately) involving significant pooling operations between the features at different layers. More importantly, the features computed by this model are 51,200-dimensional (more than an order of magnitude larger than in the other models) and thus cannot be directly compared to the results reporter by our method. In principle, similar stacked-constructions that employ pooling could be built for our model as well, and this remains as part of ongoing work.

6.6 Chapter Conclusion

We have carefully revisited the ML-CSC model and explored the problem of projecting a signal onto it. In doing so, we have provided new theoretical bounds for the solution of this problem as well as stability results for practical algorithms, both greedy and convex. The search for signals within the model led us to propose a simple, yet effective, learning formulation adapting the dictionaries across the different layers to represent natural images. In particular, we employed the dictionary sparsity as a proxi for the sparsity of the inner representations, which effectively yields a model consisting of cascade of sparse convolutional filters. We demonstrated the proposed approach by learning the model on the MNIST dataset, and studied several practical applications. The experimental results show that the ML-CSC can indeed provide significant expressiveness with a very small number of model parameters.

Several question remain open: how should the model be modified to incorporate pooling operations between the layers? what consequences, both theoretical and practical, would this have? How should one recast the learning problem in order to address supervised and semi-supervised learning scenarios? Lastly, we envisage that the analysis provided in this work will empower the development of better practical and theoretical tools not only for structured dictionary learning approaches, but to the field of deep learning and machine learning in general.

6.7 Chapter Appendix

6.7.1 Properties of the ML-CSC model

Lemma 6.2.1: Given the ML-CSC model described by the set of convolutional dictionaries $\{\mathbf{D}_i\}_{i=1}^{L}$, with filters of spatial dimensions n_i and channels m_i, any dictionary $\mathbf{D}^{(i)} = \mathbf{D}_1\mathbf{D}_2 \dots \mathbf{D}_i$ is a convolutional dictionary with m_i local atoms of dimension $n_i^{\text{eff}} = \sum_{j=1}^{i} n_j - (i-1)$. In other words, the ML-CSC model is a structured global convolutional model.

Proof. A convolutional dictionary is formally defined as the concatenation of banded circulant matrices. Consider $\mathbf{D}_1 = \left[\mathbf{C}_1^{(1)}, \mathbf{C}_2^{(1)}, \dots, \mathbf{C}_{m_1}^{(1)}\right]$, where each circulant $\mathbf{C}_i^{(1)} \in \mathbb{R}^{N\times N}$. Likewise, one can express $\mathbf{D}_2 = \left[\mathbf{C}_1^{(2)}, \mathbf{C}_2^{(2)}, \dots, \mathbf{C}_{m_2}^{(2)}\right]$, where $\mathbf{C}_i^{(2)} \in \mathbb{R}^{Nm_1\times N}$. Then,

$$\mathbf{D}^{(2)} = \mathbf{D}_1\mathbf{D}_2 = \left[\mathbf{D}_1\mathbf{C}_1^{(2)}, \mathbf{D}_1\mathbf{C}_2^{(2)}, \dots, \mathbf{D}_1\mathbf{C}_{m_2}^{(2)}\right].$$

Each term $\mathbf{D}_1\mathbf{C}_i^{(2)}$ is the product of a concatenation of banded circulant matrices and a banded circulant matrix. Because the atoms in each $\mathbf{C}_i^{(2)}$ have a stride of m_1 (the number of filters in $\mathbf{D}_1$) each of these products is in itself a banded circulant matrix. This is illustrated in Figure 6.9, where it becomes clear that the first atom in $\mathbf{C}_1^{(2)}$ (of length n_2m_1) linearly combines atoms from the first n_2 blocks of m_1 filters in $\mathbf{D}_1$ (in this case $n_2 = 2$). These block are simply the unique set of filters shifted at every position. The second column in $\mathbf{C}_1^{(2)}$ will do the same for the next set n_2 blocks, starting from the second one, etc.

From the above discussion, $\mathbf{D}_1\mathbf{C}_1^{(2)}$ results in a banded circulant matrix of dimension $N \times N$. In particular, the band of this matrix is given by the dimension of the filters in the first dictionary (n_1) plus the number of blocks combined by $\mathbf{C}_1^{(2)}$ minus one. In other words, the effective dimension of the filters in $\mathbf{D}_1\mathbf{C}_1^{(2)}$ is given by $n_2^{\text{eff}} = n_2 + n_1 - 1$.

The effective dictionary $\mathbf{D}^{(2)} = \mathbf{D}_1\mathbf{D}_2$ is simply a concatenation of m_2 such banded circulant matrices. In other words, $\mathbf{D}^{(2)}$ is a convolutional dictionary with filters of dimension n_2^{eff}. The same analysis can be done for the effective dictionary at every layer, $\mathbf{D}^{(i)}$, resulting in an effective dimension of $n_i^{\text{eff}} = n_i + n_{i-1}^{\text{eff}} - 1$, and so $n_L^{\text{eff}} = \sum_{i=1}^{L} n_i - (L-1)$.

Finally, note that $\mathbf{D}^{(i)}$ has Nm_i columns, and thus there will be m_i local filters in the effective CSC model. □

6.7.2 Another stability result for the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem

Theorem 30. (Stability of the solution to the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem):
Suppose $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$ is observed through $\mathbf{y} = \mathbf{x} + \mathbf{v}$, where $\mathbf{v}$ is a bounded noise vector, $\|\mathbf{v}\|_2 \leq \mathcal{E}_0$, and [8] $\|\boldsymbol{\gamma}_i\|_{0,\infty}^{s} = \lambda_i < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right)$, for $1 \leq i \leq L$. Consider the set $\{\hat{\boldsymbol{\gamma}}_i\}_{i=1}^{L}$ to be

[8] The assumption that $\|\boldsymbol{\gamma}_i\|_{0,\infty}^{s} = \lambda_i$ can be relaxed to $\|\boldsymbol{\gamma}_i\|_{0,\infty}^{s} \leq \lambda_i$, with slight modifications of the result.

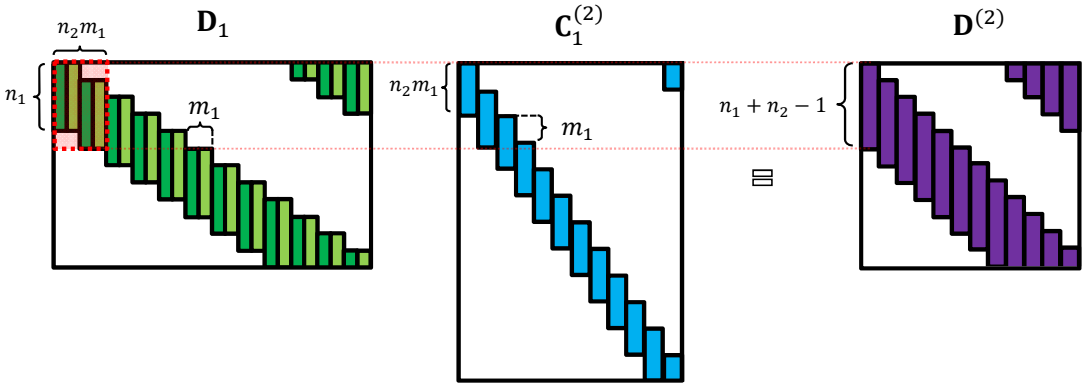


Figure 6.9: Illustration of a convolutional dictionary $\mathbf{D}_1$ multiplied by one of the circulat matrices from $\mathbf{D}_2$, in this case $\mathbf{C}_1^{(2)}$.

the solution of the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem. If $\|\boldsymbol{\gamma}_L\|_{0,\infty}^s < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}^{(L)})}\right)$ then

$$\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 \leq \frac{4\mathcal{E}_0^2}{1-(2\|\boldsymbol{\gamma}_L\|_{0,\infty}^s-1)\mu(\mathbf{D}^{(L)})}\prod_{j=i+1}^{L}\left[1+(2\|\boldsymbol{\gamma}_j\|_{0,\infty}^s-1)\mu(\mathbf{D}_j)\right].$$

Proof. Given that the original signal $\mathbf{x}$ satisfies $\|\mathbf{y}-\mathbf{x}\|_2 \leq \mathcal{E}_0$, the solution to the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem, $\hat{\mathbf{x}}$ must satisfy

$$\|\mathbf{y}-\hat{\mathbf{x}}\|_2 \leq \|\mathbf{y}-\mathbf{x}\|_2 \leq \mathcal{E}_0,$$

as this is the signal which provides a lowest ℓ_2 (data-fidelity) term. In addition, $\|\hat{\boldsymbol{\gamma}}_L\|_{0,\infty}^s = \lambda_L < \frac{1}{2}(1+\frac{1}{\mu(\mathbf{D}^{(L)})})$. Therefore, from the same arguments presented in [PSE17b], it follows that

$$\|\boldsymbol{\gamma}_L - \hat{\boldsymbol{\gamma}}_L\|_2^2 \leq \frac{4\mathcal{E}_0^2}{1-(2\|\boldsymbol{\gamma}_L\|_{0,\infty}^s-1)\mu(\mathbf{D}^{(L)})} = \mathcal{E}_L^2.$$

Because the solution $\hat{\mathbf{x}}(\{\hat{\boldsymbol{\gamma}}_i\}) \in \mathcal{M}_{\boldsymbol{\lambda}}$, then $\hat{\boldsymbol{\gamma}}_{L-1} = \mathbf{D}_L\hat{\boldsymbol{\gamma}}_L$. Therefore

$$\|\boldsymbol{\gamma}_{L-1} - \hat{\boldsymbol{\gamma}}_{L-1}\|_2^2 = \|\mathbf{D}_L(\boldsymbol{\gamma}_L - \hat{\boldsymbol{\gamma}}_L)\|_2^2 \leq (1+\delta_{2k})\|\boldsymbol{\gamma}_L - \hat{\boldsymbol{\gamma}}_L\|_2^2,$$

where δ_{2k} is the S-RIP of $\mathbf{D}_L$ with constant $2k = 2\|\boldsymbol{\gamma}_L\|_{0,\infty}^s$. This follows from the triangle inequality of the $\ell_{0,\infty}$ norm and the fact that, because $\hat{\boldsymbol{\gamma}}_L$ is a solution to the $\mathcal{P}_{\mathcal{M}_{\boldsymbol{\lambda}}}$ problem, $\|\hat{\boldsymbol{\gamma}}_L\|_{0,\infty}^s \leq \lambda_L = \|\boldsymbol{\gamma}_L\|_{0,\infty}^s$. The S-RIP can in turn be bounded with the mutual coherence [PSE17b] as $\delta_k \leq (k-1)\mu(\mathbf{D}_L)$, from which one obtains

$$\|\boldsymbol{\gamma}_{L-1} - \hat{\boldsymbol{\gamma}}_{L-1}\|_2^2 \leq \mathcal{E}_L^2\ (1+(2\|\boldsymbol{\gamma}_L\|_{0,\infty}^s-1)\mu(\mathbf{D}_L)).$$

From similar arguments, extending this to an arbitrary i^{th} layer,

$$\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 \leq \mathcal{E}_L^2 \prod_{j=i+1}^{L} (1 + (2\|\boldsymbol{\gamma}_j\|_{0,\infty}^s - 1)\mu(\mathbf{D}_j)).$$

For the sake of simplicity, one can relax the above bounds further obtaining that, subject to the assumptions in Theorem 30,

$$\|\boldsymbol{\gamma}_i - \hat{\boldsymbol{\gamma}}_i\|_2^2 \leq \mathcal{E}_L^2 \, 2^{(L-i)}.$$

This follows simply by employing the fact that $\|\boldsymbol{\gamma}_i\|_{0,\infty}^s < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right)$.

6.7.3 Local stability of the S-RIP

Lemma 6.3.1: Local one-sided near isometry:
If $\mathbf{D}$ is a convolutional dictionary satisfying the Stripe-RIP condition with constant δ_k, then

$$\|\mathbf{D}\boldsymbol{\gamma}\|_{2,\infty}^{2,p} \leq (1 + \delta_k) \, \|\boldsymbol{\gamma}\|_{2,\infty}^{2,s}$$

Proof. Consider the patch-extraction operator $\mathbf{P}_i$ from the signal $\mathbf{x} = \mathbf{D}\boldsymbol{\gamma}$, and $\mathbf{S}_i$ the operator that extracts the corresponding stripe from $\boldsymbol{\gamma}$ such that $\mathbf{P}_i\mathbf{x} = \boldsymbol{\Omega}\mathbf{S}_i\boldsymbol{\gamma}$, where $\boldsymbol{\Omega}$ is a local stripe dictionary [PSE17b]. Denote the i^{th} stripe by $\mathbf{s}_i = \mathbf{S}_i\boldsymbol{\gamma}$. Furthermore, denote by $\bar{\mathbf{S}}_i$ the operator that *extracts the support* of $\mathbf{s}_i$ from $\boldsymbol{\gamma}$. Clearly, $\mathbf{x} = \mathbf{D}\bar{\mathbf{S}}_i^T\bar{\mathbf{S}}_i\boldsymbol{\gamma}$. Note that $\|\mathbf{P}_i\|_2 = \|\mathbf{S}_i\|_2 = 1$. Then,

$$\begin{aligned}
\|\mathbf{D}\boldsymbol{\gamma}\|_{2,\infty}^p &= \max_i \|\mathbf{P}_i\mathbf{D}\bar{\mathbf{S}}_i^T\bar{\mathbf{S}}_i\boldsymbol{\gamma}\|_2 \\
&\leq \max_i \|\mathbf{P}_i\|_2 \, \|\mathbf{D}\bar{\mathbf{S}}_i^T\bar{\mathbf{S}}_i\boldsymbol{\gamma}\|_2 \\
&\leq \max_i \|\mathbf{D}\bar{\mathbf{S}}_i^T\|_2 \|\bar{\mathbf{S}}_i\boldsymbol{\gamma}\|_2 \\
&\leq \max_i \|\mathbf{D}\bar{\mathbf{S}}_i^T\|_2 \, \max_j \|\bar{\mathbf{S}}_j\boldsymbol{\gamma}\|_2.
\end{aligned}$$

Note that

$$\max_j \|\bar{\mathbf{S}}_j\boldsymbol{\gamma}\|_2 = \max_j \|\mathbf{S}_j\boldsymbol{\gamma}\|_2 = \|\boldsymbol{\gamma}\|_{2,\infty}^s,$$

as the non-zero entries in $\bar{\mathbf{S}}_j\boldsymbol{\gamma}$ and $\mathbf{S}_j\boldsymbol{\gamma}$ are the same. On the other hand, denoting by $\lambda_{max}(\cdot)$ the maximal eigenvalue of the matrix in its argument, $\|\mathbf{D}\bar{\mathbf{S}}_i^T\|_2 = \sqrt{\lambda_{max}\left(\bar{\mathbf{S}}_i\mathbf{D}^T\mathbf{D}\bar{\mathbf{S}}_i^T\right)}$, and if $\mathcal{T} = Supp(\boldsymbol{\gamma})$,

$$\lambda_{max}\left(\bar{\mathbf{S}}_i\mathbf{D}^T\mathbf{D}\bar{\mathbf{S}}_i^T\right) \leq \lambda_{max}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right), \tag{6.8}$$

because[9] $\bar{\mathbf{S}}_i\mathbf{D}^T\mathbf{D}\bar{\mathbf{S}}_i^T$ is a principal sub-matrix of $\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}$. Thus, also $\|\mathbf{D}\bar{\mathbf{S}}_i^T\|_2 \leq \|\mathbf{D}_{\mathcal{T}}\|_2$.

[9]The inequality in (6.8) can be shown by considering the equivalent expression $\lambda_{max}\left(\mathbf{S}_i\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\mathbf{S}_i^T\right)$, where the matrix $\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}$ is real and symmetric, and the matrix $\mathbf{S}_i$ is semi-orthogonal; i.e. $\mathbf{S}_i\mathbf{S}_i^T = \mathbf{I}$. Thus, from Poincaré Separation Theorem, $\lambda_{min}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right) \leq \lambda\left(\mathbf{S}_i\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\tau}\mathbf{S}_i^T\right) \leq \lambda_{max}\left(\mathbf{D}_{\mathcal{T}}^T\mathbf{D}_{\mathcal{T}}\right)$.

The Stripe-RIP condition, as in Equation (6.3), provides a bound on the square of the singular values of $\mathbf{D}_\mathcal{T}$. Indeed, $\|\mathbf{D}_\mathcal{T}\|_2^2 \leq (1+\delta_k)$, for every $\mathcal{T} : \|\mathcal{T}\|_{0,\infty}^s = k$. Including these in the above one obtains the desired claim:

$$\|\mathbf{D}\boldsymbol{\gamma}\|_{2,\infty}^p \leq \max_i \|\mathbf{D}\bar{\mathbf{S}}_i^T\|_2 \max_j \|\bar{\mathbf{S}}_j\boldsymbol{\gamma}\|_2 \leq \sqrt{1+\delta_k}\|\boldsymbol{\gamma}\|_{2,\infty}^s.$$

6.7.4 Recovery guarantees for pursuit algorithms

Convex relaxation case

Theorem 28. Stable recovery of the Multi-Layer Pursuit Algorithm in the convex relaxation case:
Suppose a signal $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$ is contaminated with locally-bounded noise $\mathbf{v}$, resulting in $\mathbf{y} = \mathbf{x} + \mathbf{v}$, $\|\mathbf{v}\|_{2,\infty}^p \leq \epsilon_0$. Assume that all representations $\boldsymbol{\gamma}_i$ satisfy the N.V.S. property for the respective dictionaries $\mathbf{D}_i$, and that $\|\boldsymbol{\gamma}_i\|_{0,\infty}^s = \lambda_i < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\right)$, for $1 \leq i \leq L$ and $\|\boldsymbol{\gamma}_L\|_{0,\infty}^s = \lambda_L \leq \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D}^{(L)})}\right)$. Consider solving the Pursuit stage in Algorithm 6.1 as

$$\hat{\boldsymbol{\gamma}}_L = \arg\min_{\boldsymbol{\gamma}} \|\mathbf{y} + \mathbf{D}^{(L)}\boldsymbol{\gamma}\|_2^2 + \zeta_L\|\boldsymbol{\gamma}\|_1,$$

for $\zeta_L = 4\epsilon_0$, and set $\hat{\boldsymbol{\gamma}}_{i-1} = \mathbf{D}_i\hat{\boldsymbol{\gamma}}_i$, $i = L, \ldots, 1$. Then, for every $1 \leq i \leq L$ layer,

1. $Supp(\hat{\boldsymbol{\gamma}}_i) \subseteq Supp(\boldsymbol{\gamma}_i)$,
2. $\|\hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i\|_{2,\infty}^p \leq \epsilon_L \prod_{j=i+1}^{L} \sqrt{\frac{3c_j}{2}}$,

where $\epsilon_L = \frac{15}{2}\,\epsilon_0\sqrt{\|\boldsymbol{\gamma}_j\|_{0,\infty}^p}$ is the error at the last layer, and c_j is a coefficient that depends on the ratio between the local dimensions of the layers, $c_j = \left\lceil \frac{2n_{j-1}-1}{n_j} \right\rceil$.

Proof. Denote $\boldsymbol{\Delta}_i = \hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i$. From [PSE17b] (Theorem 19), the solution $\hat{\boldsymbol{\gamma}}_L$ will satisfy:

1. $\mathbf{S}(\hat{\boldsymbol{\gamma}}_L) \subseteq \mathbf{S}(\boldsymbol{\gamma}_L)$; and
2. $\|\boldsymbol{\Delta}_L\|_\infty \leq \frac{15}{2}\,\epsilon_0$. □

As shown in [PRE16], given the ℓ_∞ bound of the representation error, we can bound its $\ell_{2,\infty}$ norm as well, obtaining

$$\|\boldsymbol{\Delta}_L\|_{2,\infty}^p \leq \|\boldsymbol{\Delta}_L\|_\infty\sqrt{\|\boldsymbol{\Delta}_L\|_{0,\infty}^p} \leq \frac{15}{2}\,\epsilon_0\sqrt{\|\boldsymbol{\gamma}_L\|_{0,\infty}^p}, \tag{6.9}$$

because, since $\mathbf{S}(\hat{\boldsymbol{\gamma}}_L) \subset \mathbf{S}(\boldsymbol{\gamma}_L)$, $\|\boldsymbol{\Delta}_L\|_{0,\infty}^s \leq \|\boldsymbol{\gamma}_L\|_{0,\infty}^s$. Define $\epsilon_L = \frac{15}{2}\,\epsilon_0\sqrt{\|\boldsymbol{\gamma}_L\|_{0,\infty}^p}$.

Recall that the N.V.S. property states that the entries in $\boldsymbol{\gamma}$ will no cause the support of the atoms in $\mathbf{D}$ cancel each other; i.e., $\|\mathbf{D}\boldsymbol{\gamma}\|_0 = \|\mathbf{D}_\mathcal{T}\|_\infty^0$ (Definition 27). In other words, this

provides a bound on the cardinality of the vector resulting from the multiplication of $\mathbf{D}$ with any sparse vector with support $\mathcal{T}$. Concretely, if $\boldsymbol{\gamma}$ satisfies the N.V.S., then $\|\mathbf{D}\boldsymbol{\gamma}\|_0 \geq \|\mathbf{D}\hat{\boldsymbol{\gamma}}\|_0$.

Consider now the estimate at the $L-1$ layer, obtained as $\hat{\boldsymbol{\gamma}}_{L-1} = \mathbf{D}_L\hat{\boldsymbol{\gamma}}_L$. Because $\boldsymbol{\gamma}_L$ satisfies the N.V.S. property, and $\mathbf{S}(\hat{\boldsymbol{\gamma}}_L) \subseteq \mathbf{S}(\boldsymbol{\gamma}_L)$, then $\|\hat{\boldsymbol{\gamma}}_{L-1}\|_0 \leq \|\boldsymbol{\gamma}_{L-1}\|_0$, and more so $\mathbf{S}(\hat{\boldsymbol{\gamma}}_{L-1}) \subseteq \mathbf{S}(\boldsymbol{\gamma}_{L-1})$.

On the other hand, recalling Lemma 6.7.3 and denoting by δ_{λ_L} the Stripe-RIP constant of the $\mathbf{D}_L$ dictionary, and because $\|\boldsymbol{\Delta}_L\|_{0,\infty}^s \leq \|\boldsymbol{\gamma}_L\|_{0,\infty}^s \leq \lambda_L$,

$$\|\boldsymbol{\Delta}_{L-1}\|_{2,\infty}^{2,p} = \|\mathbf{D}_L\boldsymbol{\Delta}_L\|_{2,\infty}^{2,p} \leq (1+\delta_{\lambda_L})\|\boldsymbol{\Delta}_L\|_{2,\infty}^{2,s}.$$

Notice that by employing the above Lemma, we have bounded the **patch-wise** $\ell_{2,\infty}$ norm of $\boldsymbol{\Delta}_{L-1}$ in terms of the **stripe-wise** $\ell_{2,\infty}$ of $\boldsymbol{\Delta}_L$. Recalling the derivation from [PRE16] (Section 7.1), at each i^{th} layer, a stripe includes up to $(2n_{i-1}-1)/n_i$ patches. Define $c_i = \left\lceil \frac{2n_{i-1}-1}{n_i} \right\rceil$. From this, one can bound the square of the ℓ_2 norm of a stripe with the norm of the maximal patch within it - this is true for every stripe, and in particular for the stripe with the maximal norm. This implies that $\|\boldsymbol{\Delta}_L\|_{2,\infty}^{2,s} \leq c_L\|\boldsymbol{\Delta}_L\|_{2,\infty}^{2,p}$. Then,

$$\|\boldsymbol{\Delta}_{L-1}\|_{2,\infty}^{2,p} \leq (1+\delta_k)\|\boldsymbol{\Delta}_L\|_{2,\infty}^{2,s} \leq (1+\delta_{\lambda_L})c_L\|\boldsymbol{\Delta}_L\|_{2,\infty}^{2,p}.$$

Employing the result in Eq. (6.9),

$$\|\boldsymbol{\Delta}_{L-1}\|_{2,\infty}^{2,p} \leq (1+\delta_k)c_L\|\boldsymbol{\Delta}_L\|_{2,\infty}^{2,p} \leq (1+\delta_k)\ c_L\ \epsilon_L^2.$$

We can further bound the Stripe-RIP constant by $\delta_k \leq (k-1)\mu(\mathbf{D})$ [PSE17b], obtaining

$$\|\boldsymbol{\Delta}_{L-1}\|_{2,\infty}^{2,p} \leq (1+(\|\boldsymbol{\gamma}_L\|_{0,\infty}^s - 1)\mu(\mathbf{D}_L))\ \epsilon_L^2\ c_L.$$

Iterating this analysis for the remaining layers yields

$$\|\hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i\|_{2,\infty}^{2,p} \leq \epsilon_L^2 \prod_{j=i+1}^{L} c_j\ (1+(\|\boldsymbol{\gamma}_j\|_{0,\infty}^s - 1)\mu(\mathbf{D}_j)).$$

This general result can be relaxed for the sake of simplicity. Indeed, considering that $\|\boldsymbol{\gamma}_i\|_{0,\infty}^s < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right)$, for $1 \leq i \leq L$,

$$1+(\|\boldsymbol{\gamma}_j\|_{0,\infty}^s - 1)\mu(\mathbf{D}_j) < 3/2,$$

and so

$$\|\hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i\|_{2,\infty}^{p} \leq \epsilon_L \prod_{j=i+1}^{L} \sqrt{\frac{3c_j}{2}}$$

Greedy case

Theorem 29. Stable recovery of the Multi-Layer Pursuit Algorithm in the greedy case: Suppose a signal $\mathbf{x}(\boldsymbol{\gamma}_i) \in \mathcal{M}_{\boldsymbol{\lambda}}$ is contaminated with energy-bounded noise $\mathbf{v}$, such that $\mathbf{y} = \mathbf{x}+\mathbf{v}$, $\|\mathbf{y}-\mathbf{x}\|_2 \leq \mathcal{E}_0$, and $\epsilon_0 = \|\mathbf{v}\|_{2,\infty}^{\mathbf{P}}$. Assume that all representations $\boldsymbol{\gamma}_i$ satisfy the N.V.S. property for the respective dictionaries $\mathbf{D}_i$, with $\|\boldsymbol{\gamma}_i\|_{0,\infty}^s = \lambda_i < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right)$, for $1 \leq i \leq L$, and

$$\|\boldsymbol{\gamma}_L\|_{0,\infty}^s < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}^{(L)})}\right) - \frac{1}{\mu(\mathbf{D}^{(L)})} \cdot \frac{\epsilon_0}{|\gamma_L^{min}|}, \tag{6.10}$$

where γ_L^{min} is the minimal entry in the support of $\boldsymbol{\gamma}_L$. Consider approximating the solution to the Pursuit step in Algorithm 6.1 by running Orthogonal Matching Pursuit for $\|\boldsymbol{\gamma}_L\|_0$ iterations. Then

1. $Supp(\hat{\boldsymbol{\gamma}}_i) \subseteq Supp(\boldsymbol{\gamma}_i)$,
2. $\|\hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i\|_2^2 \leq \frac{\mathcal{E}_0^2}{1-\mu(\mathbf{D}^{(L)})(\|\boldsymbol{\gamma}_L\|_{0,\infty}^s-1)}\left(\frac{3}{2}\right)^{L-i}$.

Proof. Given that $\boldsymbol{\gamma}_L$ satisfies Equation (6.10), from [PSE17b] (Theorem 17) one obtains that

$$\|\hat{\boldsymbol{\gamma}}_L - \boldsymbol{\gamma}_L\|_2^2 \leq \frac{\mathcal{E}_0^2}{1-\mu(\mathbf{D}^{(L)})(\|\boldsymbol{\gamma}_L\|_{0,\infty}^s-1)}.$$

Moreover, if the OMP algorithm is run for $\|\boldsymbol{\gamma}_L\|_0$ iterations, then all the non-zero entries are recovered, i.e., $Supp(\hat{\boldsymbol{\gamma}}_L) = Supp(\boldsymbol{\gamma}_L)$. Therefore, $\|\hat{\boldsymbol{\gamma}}_L - \boldsymbol{\gamma}_L\|_{0,\infty}^s \leq \|\boldsymbol{\gamma}_L\|_{0,\infty}^s = \lambda_L$.

Now, let $\hat{\boldsymbol{\gamma}}_{L-1} = \mathbf{D}_L\hat{\boldsymbol{\gamma}}_L$. Regarding the support of $\hat{\boldsymbol{\gamma}}_{L-1}$, because $\boldsymbol{\gamma}_L$ satisfies the N.V.S. property, $\|\hat{\boldsymbol{\gamma}}_{L-1}\|_0 \leq \|\boldsymbol{\gamma}_{L-1}\|_0$. More so, all entries in $\hat{\boldsymbol{\gamma}}_{L-1}$ will correspond to non-zero entries in $\boldsymbol{\gamma}_{L-1}$. In other words,

$$Supp(\hat{\boldsymbol{\gamma}}_{L-1}) \subseteq Supp(\boldsymbol{\gamma}_{L-1}).$$

Consider now the error at the $L-1$ layer, $\|\boldsymbol{\gamma}_{L-1} - \hat{\boldsymbol{\gamma}}_{L-1}\|_2^2$. Since $\|\boldsymbol{\gamma}_{L-1} - \hat{\boldsymbol{\gamma}}_{L-1}\|_{0,\infty}^s \leq \|\boldsymbol{\gamma}_{L-1}\|_{0,\infty}^s$, we can bound this error in terms of the Stripe RIP:

$$\|\boldsymbol{\gamma}_{L-1} - \hat{\boldsymbol{\gamma}}_{L-1}\|_2^2 = \|\mathbf{D}_L(\boldsymbol{\gamma}_L - \hat{\boldsymbol{\gamma}}_L)\|_2^2 \leq (1+\delta_{\lambda_L})\|\boldsymbol{\gamma}_L - \hat{\boldsymbol{\gamma}}_L\|_2^2,$$

We can further bound the SRIP constant as $\delta_k \leq (k-1)\mu(\mathbf{D})$, from which one obtains

$$\|\hat{\boldsymbol{\gamma}}_{L-1} - \boldsymbol{\gamma}_{L-1}\|_2^2 \leq \frac{\mathcal{E}_0^2}{1-\mu(\mathbf{D}^{(L)})(\|\boldsymbol{\gamma}_L\|_{0,\infty}^s-1)}(1+(\|\boldsymbol{\gamma}_L\|_{0,\infty}^s-1)\mu(\mathbf{D}_L)).$$

From similar arguments, one obtains analogous claims for any i^{th} layer; i.e.,

$$\|\hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i\|_2^2 \leq \frac{\mathcal{E}_0^2}{1-\mu(\mathbf{D}^{(L)})(\|\boldsymbol{\gamma}_L\|_{0,\infty}^s-1)} \prod_{j=i+1}^{L}(1+(\|\boldsymbol{\gamma}_j\|_{0,\infty}^s-1)\mu(\mathbf{D}_j)).$$

This bound can be further relaxed for the sake of simplicity. Because $\|\boldsymbol{\gamma}_i\|_{0,\infty}^s < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right)$,

for $1 \leq i \leq L$, then $(1 + (\|\boldsymbol{\gamma}_L\|_{0,\infty}^s - 1)\mu(\mathbf{D}_L)) < 3/2$, and so

$$\|\hat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i\|_2^2 \leq \frac{\mathcal{E}_0^2}{1 - \mu(\mathbf{D}^{(L)})(\|\boldsymbol{\gamma}_L\|_{0,\infty}^s - 1)} \left(\frac{3}{2}\right)^{L-i}.$$

6.7.5 Sparse Dictionaries

Lemma 6.4.1: Dictionary Sparsity Condition
Consider the ML-CSC model $\mathcal{M}_{\boldsymbol{\lambda}}$ described by the the dictionaries $\{\mathbf{D}_1\}_{i=1}^L$ and the layer-wise $\ell_{0,\infty}$-sparsity levels $\lambda_1, \lambda_2, \ldots, \lambda_L$. Given $\boldsymbol{\gamma}_L : \|\boldsymbol{\gamma}_L\|_{0,\infty}^s \leq \lambda_L$ and constants $c_i = \left\lceil \frac{2n_{i-1}-1}{n_i} \right\rceil$, the signal $\mathbf{x} = \mathbf{D}^{(L)}\boldsymbol{\gamma}_L \in \mathcal{M}_{\boldsymbol{\lambda}}$ if

$$\|\mathbf{D}_i\|_0 \leq \frac{\lambda_{i-1}}{\lambda_i c_i}, \quad \forall\ 1 < i \leq L.$$

Proof. This lemma can be proven simply by considering that the patch-wise $\ell_{0,\infty}$ of the representation $\boldsymbol{\gamma}_{L-1}$ can be bounded by $\|\boldsymbol{\gamma}_{L-1}\|_{0,\infty}^p \leq \|\mathbf{D}_L\|_0 \|\boldsymbol{\gamma}_L\|_{0,\infty}^s$. Thus, if $\|\mathbf{D}_L\|_0 \leq \lambda_{L-1}/\lambda_L$ and $\|\boldsymbol{\gamma}_L\|_{0,\infty}^s \leq \lambda_L$, then $\|\boldsymbol{\gamma}_{L-1}\|_{0,\infty}^p \leq \lambda_{L-1}$. Recalling the argument in [PRE16] (Section 7.1), a stripe from the i^{th} layer includes up to $c_i = \lceil (2n_{i-1} - 1)/n_i \rceil$ patches. Therefore, $\|\boldsymbol{\gamma}_{L-1}\|_{0,\infty}^s \leq c_L \|\boldsymbol{\gamma}_{L-1}\|_{0,\infty}^p$, and so $\boldsymbol{\gamma}_{L-1}$ will satisfy its corresponding sparsity constraint if $\|\mathbf{D}_L\|_0 \leq \lambda_{L-1}/(c_L \lambda_L)$. Iterating this argument for the remaining layers proves the above lemma.

Chapter 7

Conclusion

In the Introduction, right after commenting on the limitations of local-based sparse modeling approaches, we stated three questions that were to guide the development of this work. Let us now review where each of these points has taken us:

1. *How can one solve, or at least alleviate, the artifacts that result from employing a local sparse paradigm when restoring global images?*

   A central aspect of local sparse processing (any many other local restorations methods) is that each local patch is estimated while disregarding its global context. Based on this observation, we proposed a multi-scale approach that provides local estimates but of different effective sizes, allowing to consider progressively more global information and resulting in a remarkable improvement in visual quality. We further explored the concept of the Expected Patch Log Likelihood, which states that the prior (or model) should be enforced on patches from the reconstructed image, and not just on the intermediate ones. Recognizing that this decreases the lack of agreement between overlapping patches, we borrowed this idea and formulated it in terms of a sparse enforcing prior, resulting in an improved restoration algorithm. We further explored global regularization techniques by leveraging the Laplacian obtained from a popular denoising operator. In this case as well, we corroborated that this global force improves the performance of patch-based approaches.

2. *How can one deploy sparse-enforcing ideas to treat global signals or images?*

   As an alternative to solving the issues of patch-based approaches, we explored the path of learning global sparse models for natural images. For this to feasible, one needs the model to be computationally efficient to apply. In addition, adding constraints to the model allows to lessen the curse of dimensionality and facilitates learning. This led us to employ the double sparsity model, not without first boosting its representation power by proposing a new cropped wavelets dictionary that avoids edge effects. By suggesting an online learning algorithm, we demonstrated that (i) performance can be gained by employing larger patches given a suitable model and learning algorithm, (ii) the proposed approach scales

and performs better than competing methods when addressing the dictionary learning problem in higher dimensions, and (iii) when focusing on a particular class of images, our approach can provide solutions to problems that local approaches simply can not address.

3. *What is the global model imposed on signals while working under a local-sparsity framework?*

   This question led us to the field of Convolutional Sparse Coding, enforcing a global structure while assuring a local and shift-invariant sparse model – only to find that traditional results in sparse representation theory are simply not applicable to this case! We thus proposed a novel theoretical analysis based on a local sparsity measure, enabling us to provide guarantees for the pursuit problems in the convolutional setting as well as claims for practical (and popular) algorithms that approximate the solution to these problems. This local treatment of global pursuits was not only fruitful in terms of theoretical results, but it also enabled us to propose sparse coding and learning algorithms leveraging local convolutional entities – slices – yielding better and faster methods.

After having explored these questions, and motivated by the acute connection between deep learning and CSC, we undertook the study of the Multi-Layer Convolutional Sparse Coding model. Interestingly, this last work contains ingredients from all the points above: it provides local estimates that contribute coherently to increasingly larger structures, at different scales, in a convolutional manner and resulting in a global but constrained sparse model. The proposed projection approached allowed us to undertake the pursuit of signals in the model, and to introduce the first known method to train the ML-CSC model from real data.

7.1 Open Questions

All the above works, while providing solutions to the initial issues that motivated them, also raised further questions and working directions that remain unexplored. Before concluding, we comment on some of these points.

From the results presented in Chapter 3, one can understand that different content in natural images should receive a different restoration treatment. This can be clearly seen by considering the denoising problem with patches of increasing sizes: while large and smooth regions (like lakes and sky areas) are finely recovered by employing large patches (either with Trainlets or the Fused K-SVD algorithm), this approach generally comes with a decrease in the capabilities of representing fine details or texture. From this observation it naturally follows one could benefit from employing, for example, different patch sizes in different image regions. We now have a better undertanding of how one should treat larger patches in natural images, so how could one device an algorithm that optimally combines different sized patches and adaptively employs certain scales that allow for the best restoration for each region? Such a algorithm, not only for image denoising for any restoration task, would clearly push the performance of current approaches.

The work presented in Chapter 4 showed that dictionary learning can be employed to higher dimensional signals. In particular, this approach is mostly useful when addressing signals belonging to the same class – as demonstrated for the problem of face image inpainting. There are many other problems in image and signal processing that could benefit from a global model even for a reduced type of images. Image compression is one such application, and compression of face images bears special importance due to the plethora of visual comunication tools in social media. It would be interesting and promising to consider algorithms that would leverage these global models to suggest novel and better compression algorithms.

All the analysis that we presented in Chapter 5 (and 6, for that matter) is based on worst-case assumptions, and it is therefore pesimistic in the imposed conditions. Extending this analysis to a probabilistic setting, in which one could guarantee recovery of sparse vectors with high probability under more permissive conditions would be interesting and (most importantly) useful from a practical point of view. Moreover, noting that the CSC framework addressed specially the pursuit of global images, one can foresee extending the theory of Compressed Sensing to the convolutional setting based on the same local analysis presenting in our work. This path is likely to extend known bounds and improve the performance of current sensing systems, and the applications that come with them.

The last chapter, which studies the ML-CSC model, is perhaps the one that represents the most interesting working directions. Deep CNNs where shown to be tightly connected to sparse representations, essentially performing a pursuit for the sparse features maps at every forward pass iteration. However, what is the truly underlying connection between sparsity and general machine learning? While we have leveraged the importance of sparse representations in learning generative synthesis models, to what extent do other problems, such as classification, depend on sparsity? In fact, recalling the last experiment for the ML-CSC model for unsupervised classification, the computed features were not very sparse. How do current models and theories explain this phenomenon?

The $\text{DCP}_{\boldsymbol{\lambda}}^{\mathcal{E}}$ problem formulation is the most general way to pose the pursuit of the multi-layer representations. By restricting ourselves to the projection formulation, we have made the study of the ML-CSC model simpler and better posed, though departing from the algorithms and models related to deep learning. For instance, one of the arguably most important properties of modern deep networks is their invariance to different sources of nuisance in the input data. The ML-CSC is a formal synthesis model, which is all but invariant: it is *covariant*! While this is a desirable property in some applications (like detection), it might be detrimental in others (like classification). How could we generalize the ML-CSC model to incorporate, in a principled way, invariance to shift, deformation, or even color? This would compel the study of radically new, and likely more flexible, generative models, which will certainly expand the practical capabilities of sparse modeling methods in signal and image processing.

Bibliography

[AE08] Michal Aharon and Michael Elad. Sparse and Redundant Modeling of Image Content Using an Image-Signature-Dictionary. *SIAM Journal on Imaging Sciences*, 1(3):228–247, 2008.

[AEB06] M. Aharon, M. Elad, and A. M. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Trans. on Signal Process.*, 54(11):4311–4322, 2006.

[AF13] M.S.C. Almeida and M.A.T. Figueiredo. Frame-based image deblurring with unknown boundary conditions using the alternating direction method of multipliers. In *IEEE International Conference on Image Processing (ICIP)*, pages 582–585, Sept 2013.

[BB08] L Bottou and Olivier Bousquet. The Tradeoffs of Large Scale Learning. *Artificial Intelligence*, 20:161–168, 2008.

[BBC+01] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *Image Processing, IEEE Transactions on*, 10(8):1200–1211, 2001.

[BBCS10] Aurélie Bugeau, Marcelo Bertalmío, Vicent Caselles, and Guillermo Sapiro. A comprehensive framework for image inpainting. *Image Processing, IEEE Transactions on*, 19(10):2634–2645, 2010.

[BCM05] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. *Conference on Computer Vision and Pattern Recognition, CVPR IEEE.*, pages 60–65, 2005.

[BD08] T. Blumensath and Mike E. Davies. Iterative Thresholding for Sparse Approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, September 2008.

[BD10] Thomas Blumensath and Mike E. Davies. Normalized iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal on Selected Topics in Signal Processing*, 4(2):298–309, 2010.

[BDE09] A. M. Bruckstein, D. L. Donoho, and M. Elad. From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. *SIAM Review.*, 51(1):34–81, February 2009.

[BE08] Ori Bryt and Michael Elad. Compression of facial images using the K-SVD algorithm. *J. Vis. Commun. Image Represent.*, 19(4):270–282, May 2008.

[BEL13] Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast Convolutional Sparse Coding. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, (2):391–398, June 2013.

[BL14] Hilton Bristow and Simon Lucey. Optimization Methods for Convolutional Sparse Coding. Technical report, June 2014.

[BM13] Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[BMBP11] L. Benoît, J. Mairal, F. Bach, and J. Ponce. Sparse image representation with epitomes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[Bot98] Léon Bottou. Online algorithms and stochastic approximations. In *Online Learning and Neural Networks*. Cambridge University Press, 1998. revised, Oct 2012.

[BPC+11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[BSCB00] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424. ACM Press/Addison-Wesley Publishing Co., 2000.

[BSH12] Harold C. Burger, Christian J. Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with BM3D? *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, 2012.

[BST14] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

[BT09a] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[BT09b] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[CD00] Emmanuel J. Candes and David L. Donoho. Curvelets, multiresolution representation, and scaling laws. In *Proc. SPIE*, volume 4119, pages 1–12, 2000.

[CDS01] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Review*, 43(1):129–159, 2001.

[CDV93] Albert Cohen, Ingrid Daubechies, and Pierre Vial. Wavelet bases on the interval and fast algorithms. *Journal of Applied and Computational Harmonic Analysis*, 1(12):54–81, 1993.

[CLMW11] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58:11:1–11:37, 2011.

[CMTD15] O Chabiron, F Malgouyres, Jy Tourneret, and N Dobigeon. Toward Fast Transform Learning. *International Journal of Computer Vision*, pages 1–28, 2015.

[CPR13] Rakesh Chalasani, Jose C Principe, and Naveen Ramakrishnan. A fast proximal method for convolutional sparse coding. In *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pages 1–5. IEEE, 2013.

[CPT04] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *Image Processing, IEEE Transactions on*, 13(9):1200–1212, 2004.

[CSS16] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 698–728, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[CT05] Emmanuel J Candes and Terence Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.

[DDDM04] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

[DE03] David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[DET06] D.L. Donoho, M. Elad, and V.N. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *Information Theory, IEEE Transactions on*, 52(1):6–18, Jan 2006.

[DFKE06] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising with block-matching and 3D filtering. *Proc. SPIE-IS&T Electron. Imaging*, 6064:1–12, 2006.

[DFKE07] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Trans. on Image Process.*, 16(8):2080–2095, January 2007.

[DM09] Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, 2009.

[DMA97] Geoff Davis, Stephane Mallat, and Marco Avellaneda. Adaptive greedy approximations. *Constructive approximation*, 13(1):57–98, 1997.

[DV02] Minh N. Do and Martin Vetterli. Contourlets: a directional multiresolution image representation. In *ICIP*, 2002.

[DV05] Minh N Do and Martin Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Trans. Image Process.*, 14(12):2091–2106, 2005.

[DZSW11] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Trans. on Image Process.*, 20(7):1838–1857, 2011.

[EA06] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745, December 2006.

[EAH00] Kjersti Engan, Sven Ole Aase, and John Hakon Husoy. Multi-frame Compression: Theory and Design. *Signal Processing*, 80:2121–2140, 2000.

[EHJ+04] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[Ela06] Michael Elad. Why Simple Shrinkage Is Still Relevant for Redundant Representations? *IEEE Transactions on Information Theory*, 52(12):5559–5569, December 2006.

[Ela10] Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[ELB08] Abderrahim Elmoataz, Olivier Lezoray, and Sébastien Bougleux. Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE Transactions on Image Processing*, 17(7):1047–1060, 2008.

[EMR07] Michael Elad, Peyman Milanfar, and Ron Rubinstein. Analysis versus synthesis in signal priors. *Inverse Problems*, 23:947–968, 2007.

[ER06] Ramin Eslami and Hayder Radha. Translation-invariant contourlet transform and its application to image denoising. *IEEE Trans. Image Process.*, 15(11):3362–3374, 2006.

[GBK01] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[GL10] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 399–406, 2010.

[GLM14] Christine Guillemot and Olivier Le Meur. Image inpainting: Overview and recent advances. *Signal Processing Magazine, IEEE*, 31(1):127–144, 2014.

[GN03] Rémi Gribonval and Morten Nielsen. Sparse representations in unions of bases. *Information Theory, IEEE Transactions on*, 49(12):3320–3325, 2003.

[GO07] Guy Gilboa and Stanley Osher. Nonlocal linear image regularization and supervised segmentation. *Multiscale Modeling & Simulation*, 6(2):595–630, 2007.

[GR97] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. *IEEE Trans. Signal Process.*, 45(3):600–616, March 1997.

[GRKN07] Roger Grosse, Rajat Raina, Helen Kwong, and Andrew Y Ng. Shift-Invariant Sparse Coding for Audio Classification. In *Uncertainty in Artificial Intelligence*, 2007.

[GSB15] Raja Giryes, Guillermo Sapiro, and Alex M Bronstein. Deep neural networks with random gaussian weights: A universal classification strategy. *CoRR, abs/1504.08291*, 2015.

[Gul06] Onur G Guleryuz. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part ii: adaptive algorithms. *Image Processing, IEEE Transactions on*, 15(3):555–571, 2006.

[GZX+15] Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1823–1831, 2015.

[HA15] Furong Huang and Animashree Anandkumar. Convolutional dictionary learning through tensor factorization. *arXiv preprint arXiv:1506.03509*, 2015.

[Haw03] S Hawking. *On the Shoulders of Giants Running Press.* Philadelphia-London, 2003.

[HHW15] Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. Fast and flexible convolutional sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 5135–5143. IEEE, 2015.

[HSK13] Simon Hawe, Matthias Seibert, and Martin Kleinsteuber. Separable dictionary learning. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 438–445, 2013.

[HSL09] Hao He, Petre Stoica, and Jian Li. Designing unimodular sequence sets with good correlations - including an application to mimo radar. *IEEE Transactions on Signal Processing*, 57(11):4391–4405, 2009.

[Jai79] Anil K. Jain. A sinusoidal family of unitary transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:356–365, 1979.

[JLD13] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent k-svd: Learning a discriminative dictionary for recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2651–2664, 2013.

[K+12] Gitta Kutyniok et al. *Shearlets: Multiscale analysis for multivariate data*. Springer Science & Business Media, 2012.

[KF14] Bailey Kong and Charless C Fowlkes. Fast convolutional sparse coding (fcsc). *Department of Computer Science, University of California, Irvine, Tech. Rep*, 2014.

[KM14] Amin Kheradmand and Peyman Milanfar. A general framework for regularized, similarity-based image restoration. *IEEE Transactions on Image Processing*, 23(12):5136–5151, 2014.

[KSB+10] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann L Cun. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pages 1090–1098, 2010.

[LBBH98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[LBD+90] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[LBH15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[LBM13] M. Lebrun, A. Buades, and J. M. Morel. Implementation of the "Non-Local Bayes" (NL-Bayes) Image Denoising Algorithm. *Image Processing On Line*, 3(3):1–42, 2013.

[LMG12] Olivier Le Meur and Christine Guillemot. Super-resolution-based inpainting. In *Computer Vision–ECCV 2012*, pages 554–567. Springer, 2012.

[LMG15] Luc Le Magoarou and Rémi Gribonval. Chasing butterflies: In search of efficient dictionaries. In *IEEE Int. Conf. Acoust. Speech, Signal Process*, April 2015.

[LNDF12] Anat Levin, Boaz Nadler, Fredo Durand, and William T Freeman. Patch Complexity, Finite Pixel Correlations and Optimal Denoising. In *European Conference on Computer Vision (ECCV)*, 2012.

[LWF+15] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 806–814, 2015.

[Mal08] Stphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition, 2008.

[MBP14] Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*, 2014.

[MBPS09] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Dictionary Learning for Sparse Coding. In *Int. Conference on Machine Learning*, 2009.

[MBPS10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online Learning for Matrix Factorization and Sparse Coding. *J. Mach. Learn. Res.*, 11:19–60, 2010.

[MBS09] J. Mairal, F. Bach, and G. Sapiro. Non-local Sparse Models for Image Restoration. *IEEE International Conference on Computer Vision.*, 2:2272–2279, 2009.

[MCW15] Debbie S Ma, Joshua Correll, and Bernd Wittenbrink. The chicago face database: A free stimulus set of faces and norming data. *Behavior research methods*, 47(4):1122–1135, 2015.

[MESM08] J. Mairal, M. Elad, G. Sapiro, and S. Member. Sparse Representation for Color Image Restoration. *IEEE Transactions of Image Processing*, 17(1):53–69, 2008.

[MF13] Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

[MF15] Alireza Makhzani and Brendan J Frey. Winner-take-all autoencoders. In *Advances in Neural Information Processing Systems*, pages 2791–2799, 2015.

[MFTM01] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[Mil13] Peyman Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *Signal Processing Magazine, IEEE*, 30(1):106–128, 2013.

[MS14] Francois G Meyer and Xilin Shen. Perturbation of the eigenvectors of the graph laplacian: Application to image denoising. *Applied and Computational Harmonic Analysis*, 36(2):326–334, 2014.

[MSE07] J. Mairal, G. Sapiro, and M. Elad. Multiscale Sparse Image Representation with Learned Dictionaries. *Int. Conf. Image Process.*, 3:105–108, 2007.

[MSH08] Morten Mørup, Mikkel N Schmidt, and Lars K Hansen. Shift invariant sparse coding of image and music data. *Submitted to Journal of Machine Learning Research*, 2008.

[MZ93] S. Mallat and Z. Zhang. Matching Pursuits With Time-Frequency Dictionaries. *IEEE Trans. Signal Process.*, 41(12):3397–3415, 1993.

[Ng11] Andrew Ng. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[oCD] National Oceanic of Commerce and Atmospheric Administration U.S. Department. NOAA Photo Library.

[OLE11] Boaz Ophir, Michael Lustig, and Michael Elad. Multi-Scale Dictionary Learning Using Wavelets. *IEEE J. Sel. Top. Signal Process.*, 5(5):1014–1024, September 2011.

[PCCP14] Vishal M Patel, Yi-Chen Chen, Rama Chellappa, and P Jonathon Phillips. Dictionaries for image and video-based face recognition. *JOSA A*, 31(5):1090–1103, 2014.

[Pey09] Gabriel Peyré. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113(2):249–260, 2009.

[PKD+16] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. 2016.

[PRE16] Vardan Papyan, Yaniv Romano, and Michael Elad. Convolutional neural networks analyzed via convolutional sparse coding. *To appear in JMLR. arXiv preprint arXiv:1607.08194*, 2016.

[PRE17] Vardan Papyan, Yaniv Romano, and Michael Elad. Convolutional neural networks analyzed via convolutional sparse coding. *Journal of Machine Learning Research*, 18(83):1–52, 2017.

[PRK93] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. *Asilomar Conf. Signals, Syst. Comput. IEEE.*, pages 40–44, 1993.

[PSE17a] Vardan Papyan, Jeremias Sulam, and Michael Elad. Working locally thinking globally: Theoretical guarantees for convolutional sparse coding. *IEEE Transactions on Signal Processing*, 65(21):5687–5701, 2017.

[PSE17b] Vardan Papyan, Jeremias Sulam, and Michael Elad. Working locally thinking globally: Theoretical guarantees for convolutional sparse coding. *To appear in IEEE Transactions of signal processing. Preprint arXiv:1607.02009*, 2017.

[PSWS03] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Transactions on Image Processing.*, 12(11):1338–51, January 2003.

[RB09] S. Roth and M. J. Black. Fields of Experts. *International Journal of Computer Vision.*, 82(2):205–229, January 2009.

[RB13a] Saiprasad Ravishankar and Yoram Bresler. Learning doubly sparse transforms for images. *IEEE Trans. Image Process.*, 22(12):4598–4612, 2013.

[RB13b] Saiprasad Ravishankar and Yoram Bresler. Learning Sparsifying Transforms. *IEEE Trans. Signal Process.*, 61(5):61801, 2013.

[RBE10] Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *IEEE Proceedings - Special Issue on Applications of Sparse Representation & Compressive Sensing*, 98(6):1045–1057, 2010.

[RE14] Ron Rubinstein and Michael Elad. Dictionary Learning for Analysis-Synthesis Thresholding. *IEEE Trans. on Signal Process.*, 62(22):5962–5972, 2014.

[RE15] Yaniv Romano and Michael Elad. Boosting of Image Denoising Algorithms. *SIAM Journal on Imaging Sciences*, 8(2):1187–1219, 2015.

[Ree05] S.J. Reeves. Fast image restoration without boundary artifacts. *IEEE Trans. Image Process.*, 14(10):1448–1453, Oct 2005.

[RH11] Samuel Rathmanner and Marcus Hutter. A philosophical treatise of universal induction. *Entropy*, 13(6):1076–1136, 2011.

[RHW+88] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[RPE14] Yaniv Romano, Matan Protter, and Michael Elad. Single image interpolation via adaptive nonlocal sparsity-based modeling. *IEEE Trans. on Image Process.*, 23(7):3085–3098, 2014.

[RWB15] Saiprasad Ravishankar, Bihan Wen, and Yoram Bresler. Online Sparsifying Transform Learning - Part I: Algorithms. *IEEE Journal of Selected Topics in Signal Processing*, 9(4):625–636, 2015.

[RZE08] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit. *Technion - Computer Science Department - Technical Report.*, pages 1–15, 2008.

[RZE10] R. Rubinstein, M. Zibulevsky, and M. Elad. Double Sparsity : Learning Sparse Dictionaries for Sparse Signal Approximation. *IEEE Trans. Signal Process.*, 58(3):1553–1564, 2010.

[SE15] Jeremias Sulam and Michael Elad. Expected patch log likelihood with a sparse prior. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Lecture Notes in Computer Science, pages 99–111. Springer International Publishing, 2015.

[SE16] Jeremias Sulam and Michael Elad. Large inpainting of face images with trainlets. *IEEE Signal Processing Letters*, 23(12):1839–1843, 2016.

[SL09] Conrad Sanderson and Brian C. Lovell. Multi-region probabilistic histograms for robust and scalable identity inference. *Lecture Notes in Computer Science*, pages 199–208, 2009.

[SLE15] Wen-Ze Shao, Hai-Bo Li, and Michael Elad. Bi-l 0-l 2-norm regularization for blind motion deblurring. *Journal of Visual Communication and Image Representation*, 33:42–59, 2015.

[SNS14] Mojtaba Soltanalian, Mohammad Mahdi Naghsh, and Petre Stoica. Approaching peak correlation bounds via alternating projections. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 5317–5312. IEEE, 2014.

[SO02] P. Sallee and B. A. Olshausen. Learning Sparse Multiscale Image Representations. *Adv. Neural Neural Inf. Process. Syst.*, 15:1327–1334, 2002.

[Sob15] Elliott Sober. *Ockham's razors*. Cambridge University Press, 2015.

[SOE14] J. Sulam, B. Ophir, and M. Elad. Image Denoising Through Multi-Scale Learnt Dictionaries. In *IEEE International Conference on Image Processing*, pages 808 – 812, 2014.

[SOZE16] Jeremias Sulam, Boaz Ophir, Michael Zibulevsky, and Michael Elad. Trainlets: Dictionary learning in high dimensions. *IEEE Transactions on Signal Processing*, 64(12):3180–3193, 2016.

[SPC14] Ashish Shrivastava, Vishal M Patel, and Rama Chellappa. Multiple kernel learning for sparse representation-based classification. *IEEE Transactions on Image Processing*, 23(7):3013–3024, 2014.

[SPRE17] Jeremias Sulam, Vardan Papyan, Yaniv Romano, and Michael Elad. Multi-layer convolutional sparse modeling: Pursuit and dictionary learning. *arXiv preprint arXiv:1708.08705*, 2017.

[SRE16] Jeremias Sulam, Yaniv Romano, and Michael Elad. Gaussian mixture diffusion. In *Science of Electrical Engineering (ICSEE), IEEE International Conference on the*, pages 1–5. IEEE, 2016.

[TM98] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[Tro04] J.A. Tropp. Greed is Good: Algorithmic Results for Sparse Approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[Tro06] J. A. Tropp. Just Relax : Convex Programming Methods for Identifying Sparse Signals in Noise. *IEEE Transactions on In*, 52(3):1030–1051, 2006.

[Tsc06] David Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *International Journal of Computer Vision*, 68(1):65–82, 2006.

[VLL+10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(Dec):3371–3408, 2010.

[WB09] Z. Wang and A. C. Bovik. Mean Squared Error : Love It or Leave It? *IEEE Signal Process. Mag.*, 26(January):98–117, 2009.

[WBSS04] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–12, April 2004.

[Wel74] Lloyd R Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *Information Theory, IEEE Transactions on*, 20(3):397–399, 1974.

[WF07] Yair Weiss and William T. Freeman. What makes a good model of natural images? In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2007.

[Woh14] Brendt Wohlberg. Efficient convolutional sparse coding. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7173–7177. IEEE, 2014.

[Woh16] Brendt Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing*, 25(1):301–315, January 2016.

[XS10] Zongben Xu and Jian Sun. Image inpainting by patch propagation using patch sparsity. *Image Processing, IEEE Transactions on*, 19(5):1153–1165, 2010.

[XWG+16] Bo Xin, Yizhou Wang, Wen Gao, David Wipf, and Baoyuan Wang. Maximal sparsity with deep networks? In *Advances in Neural Information Processing Systems*, pages 4340–4348, 2016.

[YD09] Mehrdad Yaghoobi and E. Davies, Mike. Compressible dictionary learning for fast sparse approximations. In *IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 662–665, August 2009.

[YQR13] N. Yu, T. Qiu, and F. Ren. Denoising for Multiple Image Copies through Joint Sparse Representation. *J. Math. Imaging Vis.*, 45:46–54, 2013.

[YWHM10] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE Trans. on Image Process.*, 19(11):2861–2873, 2010.

[ZKTF10] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

[ZL15] Yingying Zhu and Simon Lucey. Convolutional sparse coding for trajectory reconstruction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):529–540, 2015.

[ZM00] Ying Zhao and David Malah. Improved segmentation and extrapolation for block-based shape-adaptive image coding. In *Proc. Vision Interface*, pages 388–394, 2000.

[ZW11] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. *2011 International Conference on Computer Vision, ICCV.*, pages 479–486, November 2011.

בחלקו האחרון של חיבור זה נעמוד על הקשר של מודל הדלילות הקונבולוציוני ללמידה עמוקה. ננתח גרסה מרובת שכבות של מודל הדלילות, זו מבוססת על הרכבה של מודל הדלילות הקונבולוציוני לכמה שכבות עומק. נציע אלגוריתם שביכולתו למצוא את סט הייצוגים הדלילים של סיגנל השייך למודל הרב־השכבתי הדליל. אלגוריתם זה מציע להטיל את האות הנתון על המודל הנ"ל, וזה נהנה מחסמי־יציבות טובים יותר בהשוואה לאלגוריתמים מתחרים. בנוסף לכך, נפתח שיטה לאימון מילוני הקונבולוציה המרכיבים את המודל המדובר, אלו מתאימים לשכבות העומק השונות. נדגים את השיטה המוצעת על אותות אמיתיים וביישומים מגוונים.

חיבור זה מתאר דרכים שונות להתמודדות עם בעיות גלובליות, תוך מינוף הגישה לעיבוד מקומי של טלאים שבה מככב מודל הייצוגים הדלילים. התוצרים של מחקר זה הם מגוון של אלגוריתמים חדשים, פתרונות מעשיים לבעיות שונות, מודלים חדשניים ותוצאות תאורטיות אשר, אני מקווה ומאמין, יעצימו ויעשירו את הדור הבא של מודלים מתמטיים לאותות.

תקציר

מזה שנים רבות מודלים מתמטיים של אותות הם גורם מרכזי באבולוציה של אלגוריתמים חדשים ומתקדמים יותר בתחום של עיבוד אותות ותמונות ולמידת מכונה. מחקר זה עוסק במודל שנקרא ייצוגים דלילים, הנשען על האמונה שניתן להציג אותות טבעיים כקומבינציה לינארית של מעט איברי בסיס (מכאן הדלילות), הנקראים אטומים, הנלקחים מתוך מטריצה הנקראת מילון. בעשור האחרון עבודות רבות עסקו בבעיה של מציאת סט דליל של אטומים אשר מייצג באופן מיטבי אות נתון, ובשיטות שונות להתאים את המילון (ולכן גם את המודל) למגוון רחב של אותות. בעיית אימון המילון העצימה את מודל זה והובילה לתוצאות מרהיבות בתחומים רבים, החל מבעיות שחזור בעיבוד אותות ותמונות, דרך משימות מורכבות כמו זיהוי, סיווג ובעיות נוספות בתחום למידת מכונה.

בשל אילוצים חישוביים וקשיים שמקורם בבעיית אימון המילון, בבואנו לעבד אותות מממד גבוה לרוב השימוש במודל הייצוגים הדלילים מתבצע על טלאים קטנים הנלקחים מתוך האות הגלובלי. לגישה זו יש יתרונות רבים, הרי זו דרך יעילה במיוחד לקרב את פתרונן של בעיות גלובליות דרך התמודדות עם תתי בעיות מממד נמוך באופן משמעותי. יחד עם זאת, הפתרון של בעיות מקומיות בלבד מתבטא באי התאמות בין השערוכים של טלאים הקשורים זה לזה. אנו קוראים לתופעה זו "הפער הלוקאלי־גלובלי", וזה בא לידי ביטוי הן בפן המעשי והן בפן התאורטי. בחלקו הראשון של מחקר זה אנו מציעים דרכים מגוונות להתמודדות עם פער זה, על ידי שימוש בכלים של עיבוד רב־רזולוציוני ושיטות רגולריזציה גלובליות כדוגמת EPLL ורגולריזצית לפלסיאן. בהמשכה של העבודה אנו עוקפים את הצורך בעיבוד מקומי של טלאים והחסרונות הכרוכים בכך, על ידי פיתוח של שיטה ללמוד מילון לאותות מממד גבוה. לצורך כך, אנו נשען על וריאציה של מודל הדלילות המסורתי, אשר נקראת מודל הדלילות הכפול, וזה משתמש במילון cropped wavelets. גישה חדשנית זו תוביל אותנו לאלגוריתם לאימון מילון הנקרא Trainlets המסוגל לאמן אטומים מממד גבוה המותאמים לאותות קלט נתונים. גישה זו לא רק מובילה לביצועים מהטובים בעולם בפתרון בעיית אימון המילון, אלא גם מאפשרת להתמודד עם משימות שהיו חסומות בפננו עקב היכולות המוגבלות של השיטות המקומיות המסוגלות ללמוד אטומים מממד נמוך בלבד.

בחלקו השני של מחקר זה, אנו עוסקים במודל הייצוגים הדלילים הקונבולוציוני ונראה שהוא התשובה (המפתיעה במידת מה) לפער הלוקאלי־גלובלי. מודל זה הוצע בשנים האחרונות, אך הובא כשהוא אינו מגובה באנליזה תאורטית. אנו נרחיב חלק נכבד מהכלים התאורטיים שפותחו בהקשר של מודל הייצוגים הדלילים הקלאסי למקרה הקונבולוציוני, ונוכיח את יחידות הפתרון, יציבות הבעיה ואת שחזורו המוצלח של האות, וכל אלו תוך שימוש במדד מקומי של דלילות. מצד אחד, התאוריה המוצעת מצדיקה שפע של עבודות אשר מתבססות על מודל זה. מצד שני, הגישה שלנו מובילה לפיתוח של אלגוריתמים חדשים למציאת סט האטומים המייצגים אות נתון ולימוד המילון, כאשר כל זאת מתבצע על ידי פעולות מקומיות על טלאים, תוך הבטחה לשרת נאמנה את המודל הגלובלי המדובר.

כמה מרתק הוא החיפוש אחר תגליות חדשות. הוא תומך ומסייע לי בצורה בלתי רגילה, וזה מעורר השראה לעבוד איתו. לעולם אהיה אסיר תודה עבור כל אלו.

אני רוצה להודות לחברי, הקרובים ואלו הרחוקים, הותיקים ואלו מהעת האחרונה, על כל מה שהקנו לי בעצתם מלאת המחשבה ובתמיכתם. תודות מיוחדת לאלו שזכיתי לשתף פעולה עימם וללמוד מהם: בועז, חויאר, ורדן, יניב, דימה - אני מקווה שנוכל להמשיך לעבוד יחד. אני רוצה להודות לחברי הפקולטה והמעבדה שתמיכתם לאורך השנים האחרונות עזרה לי במיוחד: מיכאל ציבולבסקי, רונן טלמון, עירד יבנה, יאנה כץ, נדב טולדו, תום פלני ואנה קליינר: חשיבות עבודתכם לא תסולא בפז לכל אלו שזכו וזוכים לעבוד כאן.

תודות מיוחדות לבת הזוג האוהבת שלי, היידי, אשר תמכה בי ברגעים קשים וחגגה עימי את אלו השמחים יותר. הכרתי אותך הודות לדוקטורט זה, וזוהי רק תחילת הדרך של המסע המרגש, המרתק ומלא ההרפתקאות שלנו. ברצוני להודות לחברי ממעבדת LSyDNL (UNER), על הדרכתם, השראתם והשפעתם על צעדי הראשונים באקדמיה בכלל וכחוקר בפרט.

אחרונים חביבים, אני רוצה להביע את תודתי המיוחדת למשפחתי: להורי, אלברטו ומרגה, אשר תמיד עודדו אותי לחפש אחר תשובות, לחקור ולהבין, ולאחי, אריאל, אשר תמיד היה חכם מספיק להזכיר לי לא להפסיק לחייך ולהנות מהדרך. המחקר הזה מוקדם לכולכם.

אני מודה לטכניון על התמיכה הכספית הנדיבה בהשתלמותי.

המחקר בוצע בהנחייתו של פרופסור מיכאל אלעד, בפקולטה למדעי המחשב.

חלק מן התוצאות בחיבור זה פורסמו כמאמרים מאת המחבר ושותפיו למחקר בכנסים ובכתבי־עת במהלך תקופת מחקר הדוקטורט של המחבר, אשר גרסאותיהם העדכניות ביותר הינן:

List of Publications

- **Jeremias Sulam**, Boaz Ophir and Michael Elad, *Image denoising through multi-scale learnt dictionaries.* 2014 IEEE International Conference onImage Processing (ICIP).
- **Jeremias Sulam** and Michael Elad, *Expected patch log likelihood with a sparse prior.* Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science, 2015
- Javier Turek, **Jeremias Sulam**, Michael Elad and Irad Yavneh. *Fusion of ultrasound harmonic imaging with clutter removal using sparse signal separation.* 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
- **Jeremias Sulam**, Boaz Ophir, Michael Zibulevsky and Michael Elad. *Trainlets: Dictionary learning in high dimensions.* IEEE Transactions on Signal Processing, 2016.
- **Jeremias Sulam***, Yaniv Romano* and Michael Elad, *Gaussian Mixture Diffusion.* IEEE International Conference on the Science of Electrical Engineering (ICSEE), 2016.
- Vardan Papyan*, **Jeremias Sulam*** and Michael Elad. *Working Locally Thinking Globally: Theoretical Guarantees for Convolutional Sparse Coding.* IEEE Transactions on Signal Processing, 2017.
- **Jeremias Sulam**, Yaniv Romano and Ronen Talmon, *Dynamical system classification with diffusion embedding for ECG-based person identification.* Signal Processing, 2017.
- Vardan Papyan, Yaniv Romano, **Jeremias Sulam** and Michael Elad, *Convolutional Dictionary Learning via Local Processing.* IEEE International Conference on Computer Vision (ICCV) 2017.
- **Jeremias Sulam**, Vardan Papyan, Yaniv Romano and Michael Elad, *Multi-Layer Convolutional Sparse Modeling: Pursuit and Dictionary Learning*, Submitted, 2017.

Note: * denotes equal contribution.

תודות

לולא אנשים רבים ויקרים לי מקבץ עבודות זה לא היה אפשרי, ועל כך אני אסיר תודה.

ראשית, אני רוצה להודות ממעמקי ליבי למנחה שלי, פרופ׳ מיכאל אלעד. מארק ואן דורן כתב, "אומנות ההוראה היא האומנות לסייע בגילוי תגליות." מיקי הוא מורה מדהים: תמיד עוזר, מסביר, מייעץ, ומראה עד

מתיאור גלובלי ללוקאלי של מודלים מבוססי דלילות

חיבור על מחקר

לשם מילוי חלקי של הדרישות לקבלת התואר
דוקטור לפילוסופיה

ג'רמיאס סולם

הוגש לסנט הטכניון – מכון טכנולוגי לישראל
ה' בחשון תשע"ח חיפה נובמבר 2017

מתיאור גלובלי ללוקאלי של מודלים מבוססי דלילות

ג'רמיאס סולם