Study on the variational models and dictionary learning

Tieyong Zeng

Thèse de Mathématiques Appliquées
pour obtenir le grade de
Docteur de l'Université Paris Nord

Études de Modèles Variationnels et Apprentissage de Dictionnaires

présentée et soutenue publiquement

par

ZENG Tieyong
LAGA/L2TI, Institut Galilée, Université Paris Nord

Directeurs de thèse

Alain Trouvé, François Malgouyres

Rapporteurs

Michael Kwok-Po, Jean-Luc Starck

Jury

Jean-Michel Morel	President du jury
Françoise Dibos	Examinateure
François Malgouyres	Directeur de thèse
Lionel Moisan	Examinateur
Jean-Luc Starck	Rapporteur
Alain Trouvé	Directeur de thèse

le 09 octobre, 2007

Remerciements

Ce travail est sous les directions de Monsieur François Malgouyres et Monsieur Alain Trouvé, que je voudrais remercier en première place pour m'avoir proposé ce sujet de recherche, et, pour tout leurs dynamismes et leurs compétences scientifiques qui m'ont permis de mener à bien cette étude.

Je remercie tous particulièrement Monsieur Jean-Luc Starck, Rechercheur au Commissariat à l'Energie Atomique, ainsi que Monsieur Michael Kwok-Po, Professeur du Département de Mathématiques, Hong Kong Baptist University, qui ont accepté de juger ce travail et d'en être les rapporteurs.

J'adresse mes remerciements pour leur présence dans ce jury de à Monsieur Jean-Michel Morel, Professeur du CMLA, ENS Cachan.

Je tiens également à remercier Madame Françoise Dibos, Professeur de l'Université Paris Nord, d'avoir accepté de participer au jury de cette thèse.

Je veux exprimer tous mes remerciements à Monsieur Lionel Moisan, Professeur de Mathématiques, Université Paris V, pour sa participation à cette commission d'examen.

Cette thèse a été effectuée au sein du Laboratoire Analyse, Géométrie et Applications et du Laboratoire de Traitement et Transport de l'Information dont je voudrais remercier tous les membres et particulièrement le directeur Daniel Barsky, le directeur Chen Ken et leurs secrétaires et ses informaticiens pour leur aide et leur accueil chaleureux.

Partie de cette thèse a été aussi faite au sein du Centre de Mathématiques et Leurs Applications, ENS Cachan. Je veux bien remercier tous les membres de CMLA.

Je remercie également Dr. Li Xiaolong, Dr. Chen Huayi, Dr. Yu Yong, Dr. Wang Jiaping, Wang Yizao, Yu Pin, Zhang Bo, Luo Bin pour partager avec moi leurs connaissances et leurs idées mathématiques qui sont très utiles pour cette thèse. Toute mon amitié à mes amies Chen Xi, Dr. Gao Bo, Gong Zheng, Dr. Jiang Donghua, Dr. Jiao Ying, Qin Botao, Shi Jiayi, Tan Xiaolu, Zheng Cengbo, Dr.Zhou Guodong.

Enfin, cette thèse est consacrée à Mlle. Zhou Feng pour tous les jours qu'on partageait, on partage et on partagera.

Résumé

Ce mémoire porte sur l'utilisation de dictionnaires en analyse et restauration d'images numériques. Nous nous sommes intéressés aux différents aspects mathématiques et pratiques de ce genre de méthodes: modélisation, analyse de propriétés de la solution d'un modèle, analyse numérique, apprentissage du dictionnaire et expérimentation.

Après le Chapitre 1, qui retrace les étapes les plus significatives de ce domaine, nous présentons dans le Chapitre 2 notre implémentation et les résultats que nous avons obtenus avec le modèle consistant à résoudre

$$\begin{cases} \min_w TV(w), \\ \text{sous les contraintes } |\langle w - v, \psi \rangle| \leq \tau, \forall \psi \in \mathcal{D} \end{cases} \tag{1}$$

pour $v \in \mathbb{R}^{N^2}$, une donnée initiale, $\tau > 0$, $TV(\cdot)$ la variation totale et un dictionnaire *invariant par translation* $\mathcal{D}$. Le dictionnaire est, en effet, construit comme toutes les translations d'un ensemble $\mathcal{F}_0$ d'éléments de $\mathbb{R}^{N^2}$ (des caractéristiques ou des patchs). L'implémentation de ce modèle avec ce genre de dictionnaire est nouvelle. (Les auteurs avaient jusque là considéré des dictionnaires de paquets d'ondelettes ou de curvelets.) La souplesse de la construction du dictionnaire a permis de conduire plusieurs expériences dont les enseignements sont rapportés dans les Chapitre 2 et 3.

Les expériences du Chapitre 2 confirment que, pour obtenir de bons résultats en débruitage avec le modèle ci-dessus, le dictionnaire doit bien représenter la courbure des textures. Ainsi, lorsque l'on utilise un dictionnaire de Gabor, il vaut mieux utiliser des filtres de Gabor dont le support est isotrope (ou presque isotrope). En effet, pour représenter la courbure d'une texture ayant une fréquence donnée et vivant sur un support Ω, il faut que le support, en espace, des filtres de Gabor permette un "pavage" avec peu d'éléments du support Ω. Dans la mesure où, pour une classe générale d'images, le support Ω est indépendant de la fréquence de la texture, le plus raisonnable est bien de choisir des filtres de Gabor dont le support est isotrope. Ceci est un argument fort en faveur des paquets d'ondelettes, qui permettent en plus d'avoir plusieurs tailles de supports en espace (pour une fréquence donnée) et pour lesquelles (1) peut être résolu rapidement.

Dans le Chapitre 3 nous présentons des expériences dans lesquels le dictionnaire contient les courbures de formes connues (des lettres). Le terme d'attache aux données du modèle (1) autorise l'apparition dans le résidu $w^* - v$ de toutes les structures, sauf des formes ayant servi à construire le dictionnaire. Ainsi, on s'attend à ce que les forment restent dans le résultat w^* et que les autres structures en soient absente. Nos expériences portent sur un problème de séparation de sources et confirment cette impression. L'image de départ contient des lettres (connues) sur un fond très structuré (une image). Nous montrons qu'il est possible, avec (1), d'obtenir une séparation raisonnable de ces structures. Enfin ce travail met bien en évidence que le dictionnaire $\mathcal{D}$ doit contenir la *courbure* des éléments que l'on cherche à préserver et non pas les éléments eux-mêmes, comme on pourrait le penser naïvement.

Le Chapitre 4 présente un travail dans lequel nous avons cherché à faire collaborer la méthode K-SVD avec le modèle (1). Notre idée de départ est d'utiliser le fait que quelques itérations de l'algorithme qu'il utilise pour résoudre (1) permettent de faire réapparatre des structures absentes de l'image servant à l'initialisation de l'algorithme (et dont la courbure est présente dans le dictionnaire). Nous appliquons donc quelques une de ces itérations au résultat de K-SVD et retrouvons bien les textures perdues. Ceci permet un gain visuel et en PSNR.

Dans le Chapitre 5, nous exposons un schéma numérique pour résoudre une variante du Basis Pursuit. Celle-ci consiste à appliquer un algorithme du point proximal à ce modèle. L'intérêt est de transformer un problème convexe non-différentiable en une suite (convergeant rapidement) de problèmes convexes très réguliers. Nous montrons la convergence théorique de l'algorithme. Celle-ci est confirmée par l'expérience. Cet algorithme permet d'améliorer considérablement la qualité (en terme de parcimonie) de la solution par rapport à l'état de l'art concernant la résolution pratique du Basis Pursuit. Nous nous espérons que cet algorithme devrait avoir un impact conséquent dans ce domaine en rapide développement.

Dans le Chapitre 6, nous adaptons aux cas d'un modèle variationnel, dont le terme régularisant est celui du Basis Pursuit et dont le terme d'attache aux données est celui du modèle (1), un résultat de D. Donoho (voir [55]). Ce résultat montre que, sous une condition liant le dictionnaire définissant le terme régularisant au dictionnaire définissant le terme d'attache aux données, il est possible d'étendre les résultats de D. Donoho aux modèles qui nous intéressent dans ce chapitre. Le résultat obtenu dit que, si la donnée initiale est très parcimonieuse, la solution du modèle est proche de sa décomposition la plus parcimonieuse. Ceci garantie la stabilité du modèle dans ce cadre et fait un lien entre régularisation l^1 et l^0, pour ce type d'attache aux données.

Le Chapitre 7 contient l'étude d'une variante du Matching Pursuit. Dans cette variante, nous proposons de réduire le produit scalaire avec l'élément le mieux corrélé au résidu, avant de modifier le résidu. Ceci pour une fonction de seuillage général. En utilisant des propriétés simples de ces fonctions de seuillage, nons montrons que l'algorithme ainsi obtenu converge vers la projection orthogonale de la donnée sur l'espace linéaire engendré par le dictionnaire (le tout modulo une approximation quantifiée par les caractéristiques de la fonction de seuillage). Enfin, sous une hypothèse faible sur la fonction de seuillage (par exemple le seuillage dur la satisfait), cet algorithme converge en un temps fini que l'on peut déduire des propriétés de la fonction de seuillage. Typiquement, cet algorithme peut-être utilisé pour faire les projections orthogonales dans l'algorithme "Orthogonal Matching Pursuit". Ceci n'a pas encore été fait.

Le Chapitre 8 explore enfin la problématique de l'apprentissage de dictionnaires. Le point de vue développé est de considerer cette problématique comme un problème d'estimation de paramètres dans une famille de modèles génératifs additifs. L'introduction de switchs aléatoires de Bernoulli activant ou désactivant chaque élément d'un dictionnaire invariant par translation à estimer en permet l'identification dans des conditions assez générales en particulier dans le cas où les coefficients sont gaussiens. En utilisant une technique d'EM variationel et d'approximation de la loi a posteriori par champ moyen, nous dérivons d'un principe d'estimation par maximum de vraisemblance un nouvel algorithme effectif d'apprentissage de dictionnaire que l'on peut apparenter pour certains aspects à l'algorithme K-SVD. Les résultats expérimentaux sur données synthétiques illustrent la possibilité d'une identification correcte d'un dictionnaire source et de plusieurs applications en décomposition d'images et en débruitage.

Abstract

Titre: Study on the variational models and dictionary learning

This dissertation is dedicated to the use of dictionaries in the image analysis and image restoration. We are interested in various mathematical and practical aspects of this kind of methods: modeling, analysis the solution to such model, numerical analysis, dictionary learning and experimentation.

After Chapter 1, which reviews the most significant works of this field, we present in Chapter 2 the implementation and results which we obtained by the model consisting in solving

$$\begin{cases} \min_w TV(w), \\ \text{subject to } |\langle w - v, \psi \rangle| \leq \tau, \forall \psi \in \mathcal{D} \end{cases} \tag{2}$$

for $v \in \mathbb{R}^{N^2}$, an initial image, $\tau > 0$, $TV(\cdot)$ the total variation and a *translation invariant* dictionary $\mathcal{D}$. Actually, the dictionary, is built as all the translations of a collection $\mathcal{F}_0$ of elements of $\mathbb{R}^{N^2}$ (of features or of the patches). The implementation of this model with this kind of dictionary is new. (The authors before this dissertation only considered the dictionaries of wavelet basis/packets or curvelets.) The flexibility of the construction of the dictionary leads to several experiments which we will report in chapter 2 and 3.

The experiments of Chapter 2 confirm that, to obtain good results of denoising with the above model, the dictionary must represent the curvature of textures well. Hence, when one uses Gabor dictionary, it is better to use Gabor filters whose supports are isotropic (or almost isotropic). Indeed, for represent the curvature of a texture with a given frequency and living on a fixed support Ω, it is necessary that the support, in space, of Gabor filters allows a paving with few elements for the support Ω. For a general class of images, the support Ω is independent of the frequency of texture, it is most reasonable to choose Gabor filters whose supports are isotropic. This is a strong argument in favor of the wavelet packets dictionary, which allows in addition to have several sizes of supports in space (for a given frequency) and for which (2) can be solved quickly.

In Chapter 3, we present the experiments in which the dictionary contains the curvatures of known forms (letters). The data-fidelity term of the model (2) authorizes the appearance in the residue $w^* - v$ of all the structures, except forms being used to build the dictionary. Thus, we can expect that these forms remain in the result w^* and that the other structures will disappear. Our experiments are carried on a problem of sources separation and confirm this impression. The starting image contains letters (known) on a very structured background (an image). We show that it is possible, with (2), to obtain a reasonable separation of these structures. Finally this work illustrates clearly that the dictionary $\mathcal{D}$ must contain the *curvature* of elements which we seek to preserve and not the elements themselves, as we might think this naively.

Chapter 4 presents a work in which we try to integrate the K-SVD method with the model (2). Our starting idea is to use the fact that some iterations of the algorithm which

we use to solve (2) allow to retrieve the lost structures from the image which we used as the initialization of the algorithm (and whose curvature is present in dictionary). We thus apply some of these iterations to the result of K-SVD and recover lost textures well. This allows a visual gain and an improvement of the $PSNR$.

In Chapter 5, we expose a numerical schema to solve a variant of Basis Pursuit. This consists to apply a proximal point algorithm to this model. The interest is to transform a non-differentiable convex problem to a sequence (quickly converging) of very regular convex problem. We show the theoretical convergence of the algorithm. This one is confirmed by the experiment. This algorithm allows to improve remarkably the quality (in term of sparseness) of the solution compared to the state-of-the-art concerning the practical resolution of Basis Pursuit. This algorithm should have a consequent impact in this rapidly developing field.

In chapter 6, we adapt to the cases of a variational model, whose regularization term is that of Basis Pursuit and whose data-fidelity term is that of the model (2), a result of D. Donoho. This result shows that, under a condition relating the dictionary defining the regularization term to the dictionary defining the data-fidelity term, it is possible to extend the results of D. Donoho to the models which interest us in this chapter. The obtained result says that, if the given data is very sparse, the solution of the model is close to its most sparse decomposition. This guarantee the stability of this model within this framework and establishes a link between l^1 and l^0 regularization, for this type of data-fidelity term.

Chapter 7 contains the study of a variant of Matching Pursuit. In this variant, we proposes to reduce the scalar product with the element best correlated with the residue, before modifying the residue. This is for a general threshold function. By using simple properties of these threshold functions, we show that the algorithm thus obtained converges towards the orthogonal projection of the data on linear space generated by the dictionary (the whole modulo an approximation quantified by the characteristics of the threshold function). Finally, under a weak assumption on the threshold function (for example the hard-threshold satisfies this assumption), this algorithm converges in a finite time which one can deduce from the properties of the threshold function. Typically, this algorithm might be useful to make the orthogonal projections in the algorithm Orthogonal Matching Pursuit. This we have not done yet.

Chapter 8 explores finally the dictionary learning problem. The developed point of view is to regard this problem as a parameter estimation problem in a family of additive generative models. The introduction of random on/off switches of Bernoulli activating or deactivating each element of a translation invariant dictionary to be estimated allows the identification under rather general conditions in particular if the coefficients are Gaussian. By using an EM variational technic and the approximation of the posteriori distribution by mean field, we derive from a estimation principle by maximum likelihood a new effective algorithm of dictionary learning which one can connect for certain aspects with algorithm K-SVD. The experimental results on synthetic data illustrate the possibility of a correct identification of a source dictionary and several applications in image decomposition and image denoising.

Contents

List of Figures

Introduction

The calculus of variations is a classical mathematical field that deals with functionals, as opposed to ordinary calculus which deals with functions. The first synthesis work of variation method goes back to Léonard Euler (1707 - 1783). Based on the seminal works of Pierre de Fermat (1601 - 1665), Jakob Bernoulli (1654 - 1705) and Johann Bernoulli (1667-1748), Euler developed the calculus of variations and proposed in 1743 the "principle of variation" and thus opened the pandora box with this fundamental work (see [1], [2]).

The computers appeared at the beginning of the 1960, and this promoted significantly the capability of human being to tackle very complex problems including digital image processing. From then on, the use of variational methods, typically Energy Minimization Method, grew throughout image processing.

A pioneer work applied the total variation to image restoration. It was first proposed by Rudin-Osher-Fatami in 1992. Ever since, the total variation became a classical tool in image processing. F.Malgouyres first combined the total variation idea with the wavelet soft-threshold idea of D.Donoho using the general framework of variational approaches. This leads to an efficient image restoration method and we call it the $TV - l^\infty$ model here. Throughout this dissertation, we will concentrate our efforts on the study of various variational models related to the $TV - l^\infty$ model with different dictionaries which play an important role in these models. The main framework of this dissertation is organized as following.

1. In Chapter 1, we will review the important methods which are closely to our works.

2. In Chapter 2, we will propose a translation-invariant dictionary approach for the $TV - l^\infty$ model.

3. In Chapter 3, in order to understand the mechanism of $TV - l^\infty$ model, we will examine the ad-hoc dictionary and dictionary with known features.

4. In Chapter 4, the $TV - l^\infty$ model will be used as a post-processing procedure for the K-SVD model and this will enhance the denoising performance.

5. In Chapter 5, we will propose an effective algorithm based on Uzawa and Armijo method to solve a variant of the Basis Pursuit problem. We will analyze its convergence and report some numerical results comparing with the existing algorithms.

6. In Chapter 6, we will investigate two sparse representations models in $\mathbb{R}^{N^2}$ and then propose a Soft-Threshold Matching Pursuit algorithm.

7. In Chapter 7, we propose a MP shrinkage method in Hilbert space. It is a generalization to Matching Pursuit and wavelet shrinkage. Hence, this builds a solid bridge between these two important fields of image processing.

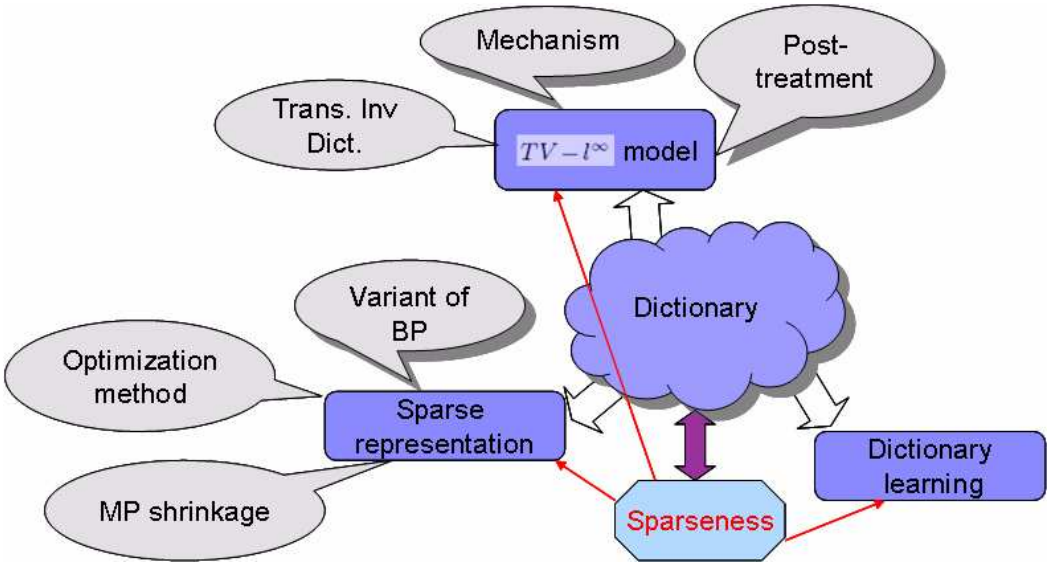

Figure 1: Idea of the dissertation

8. In Chapter 8, we will use MCMC and mean field method to search the typical patterns in a training set.

9. In Chapter 9, future works will be discussed.

Essentially, the concerns of this dissertation can be divided in three parts (see Figure 1): $TV-l^\infty$ model, sparse representation and dictionary learning. The first part contains three aspects: translation-invariant dictionary (Chapter 2), mechanism of the $TV-l^\infty$ model (Chapter 3) and a post-treatment for K-SVD model (Chapter 4); The second part also contains three aspects: a variant of Basis Pursuit (Chapter 5), two optimization models (Chapter 6) and MP shrinkage (Chapter 7); The third part is of dictionary learning (Chapter 8). The concept of dictionary acts as a central visible clue for this dissertation as it connects the three parts tightly. Moreover, dictionary and sparseness can be regarded a pairs of dural concepts. In fact, using redundant dictionary for representation implies various possible results and hence a sparse one might be interesting and in contrary, dictionary is the natural demand for sparse representation. Hence, sparseness is the hidden clue for this dissertation.

Acknowledgement

The most part of this dissertation was carried out in LAGA/L2TI while the author beneficed BDI-CNRS (2004-2007). Part of the work of Chapter 1 was done when the author carried the research internship supported by Alcatel Space. Part of this dissertation, especially Chapter 8 and the writing of this dissertation was finished in CMLA, ENS cachan.

Chapter 1

Preliminaries

1.1 Introduction

Digital image processing uses computer algorithms to perform information processing for which the input data is a digital image (digital photograph or frame of digital video). The output data is not necessarily a digital image, but can be, for instance, a set of features of the digital image.

Typical problems of digital image processing include but are not limited to geometric transformations such as enlargement, reduction, rotation, registration of two or more digital images, interpolation, segmentation of the image into regions, image editing and digital retouching, texture synthesis, classification, feature extraction, pattern recognition, projection and multi-scale image analysis.

As productions of digital images and various kinds of movies are often taken in some poor conditions, the requirement for efficient restoration techniques has grown rapidly. No matter how good a digital cameras is, an image improvement is always suitable to augment their range of action. In this chapter, we will concentrate our efforts on the problems related to image restoration.

Generally speaking, the digital image is encoded as a matrix of grey-level or color values. For each pair $(i, u(i))$, i is a point on the two-dimensional grid $\{0, \ldots, N-1\}^2$ and the pixel $u(i)$ is the value at position i. It is a real value for grey-level image or a triplet of real values for color image. For the sake of simplicity of notations and presentations of experiments, we shall only consider square 2D grey-level images and typically, we denote digital images by $u, v, w \in \mathbb{R}^{N^2}$ throughout this dissertation, unless explicitly specified.

1.2 Image restoration

The two main aspects which affect the image accuracy are the blur and the presence of noise. The blur is intrinsic to the image formation systems, as a digital image has only a finite number of samples and must approach the well-known Shannon-Nyquist sampling condition. As usual, we will use a linear operator H to model this blur. The second main image perturbation is the presence of noise. Mathematically, the observed image $v \in \mathbb{R}^{N^2}$is formed as:

$$v = Hu + b, \tag{1.1}$$

where H is the known linear operator from $\mathbb{R}^{N^2}$ to $\mathbb{R}^{N^2}$, $u \in \mathbb{R}^{N^2}$ is an ideal image and $b \in \mathbb{R}^{N^2}$ is a Gaussian white noise of standard variation σ.

The digital image restoration task, a classical inverse problem in image processing, consists in recovering the ideal image $u \in \mathbb{R}^{N^2}$ from the noisy, blurred image $v \in \mathbb{R}^{N^2}$ on the basis of a mathematical degradation model. Typical examples of image restoration include but are not limited to image denoising, image deblurring, image zooming, image inpainting and linear local contrast changes. In practice, combinations of these tasks are also of great importance. For example, one might want to denoise an image and in the meanwhile fill in some small parts of missing pixels.

Throughout this dissertation, we will either consider a general H or, for the sake of simplicity, only focus on image denoising, i.e. $H = Id$.

1.3 Recollection of image restoration/denoising

In this section, we will present a bibliography of the common image restoration/denosing methods. We would like to point out that it is far from a complete list containing all the various models proposed in this domain. A recent review of some significant works in the area of image denoising, including insights and potential future trends can be found in [3]. The authors of [4] compared several popular denoising methods and proposed a new one named *Non-Local means* with better visual effect on natural images. In fact, we only consider the models which are closely related to our current dissertation. They are:

- wavelet-shrinkage
- Total variation (TV) and Rudin-Osher-Fatemi (ROF) model
- $TV - l^\infty$
- NL-means
- K-SVD

1.3.1 Wavelet-shrinkage for denoising

The wavelet-shrinkage method is usually used for image denoising or image compression, so in this subsection, $H = Id$. This simple but useful method was first proposed by D.Dohono and I.Johnstone in [5] and has been extensively studied by many authors and is still a fruitful area of research in image processing. It is also closely related to the standard image compression method JPEG-2000.

Let $\mathcal{D} = (\psi_i)_{i\in I} \subset \mathbb{R}^{N^2}$ be a wavelet basis. To introduce the wavelet-shrinkage method, we define the soft-threshold function as, for any $t \in \mathbb{R}$

$$\rho_\tau(t) = \begin{cases} (|t| - \tau)sign(t) & \text{when } |t| \geq \tau \\ 0 & \text{otherwise,} \end{cases} \tag{1.2}$$

where τ is a fixed positive.

For a noisy image v, the wavelet-shrinkage method considers the following image

$$u = \sum_{i\in I} \rho_\tau(\langle v, \psi_i \rangle)\psi_i \tag{1.3}$$

as the denoised result. As most of the small wavelet coefficients of natural images are caused by noise, this method leads to a fairly good result (see [6]).

In order to anticipate the relation of wavelet-shrinkage with the upcoming method, we remark that if we denote

$$E(w) = \sum_{i \in I} f_l(|\langle w, \psi_i \rangle|),$$

where f_l are strictly increasing functions, the wavelet soft-shrinkage provides the solution to

$$\begin{cases} \min E(w) \\ \text{suject to } |\langle w - v, \psi_i \rangle| \leq \tau. \end{cases} \tag{1.4}$$

1.3.2 Rudin-Osher-Fatami Model

The total variation approach was initiated in [7] and was often considered as the most efficient method in the extensive works carried out by the group of Stanley Osher at UCLA and others. The greatest benefit of the total variation model is that it is very efficient for the restoration of edges present in natural images.

The basic idea of the ROF model is that the original image u has a simple geometric structure, corresponding to objects with smooth contours, or edges. The image is supposed to be smooth inside the objects but with some jumps across the boundaries.

A fundamental functional space to model these properties is the space of functions with bounded variation (BV). The space BV owns the feature of containing functions with discontinuities along lines which can represent edges in the natural image. We recall the definition of *total variation* for function of $\mathbb{R}^2$ here.

Definition 1 *Denote by Ω an open connected set of $\mathbb{R}^2$. $BV(\Omega)$ is the subspace of functions $u \in L^1(\Omega)$ s.t. the following quantity, named* ***Total Variation*** *of u, is finite:*

$$TV(u) = \sup \left\{ \int_\Omega u(x) \mathrm{div}(\zeta(x)) dx \mid \zeta \in C_c^1(\Omega, \mathbb{R}^2), \|\zeta\|_{L^\infty(\Omega)} \leq 1 \right\}. \tag{1.5}$$

When $u \in C^1(\Omega)$ (i.e. u is continuously differentiable over Ω), we have:

$$TV(u) = \int_\Omega |\nabla u(x)| dx.$$

Since most of the time, we work on the discrete images, we need to define a discrete total variation. Let's first introduce the discrete gradient operator. For a discrete image $u \in \mathbb{R}^{N^2}$, the discrete gradient ∇u is a vector of $(\mathbb{R}^2)^{N^2}$ defined by:

$$(\nabla u)_{i,j} = \left((\nabla u)^1_{i,j}, (\nabla u)^2_{i,j} \right), \text{ for all } 0 \leq i, j < N,$$

where

$$(\nabla u)^1_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } 0 \leq i < N-1 \\ u_{0,j} - u_{N-1,j} & \text{if } i = N-1, \end{cases} \tag{1.6}$$

and

$$(\nabla u)^2_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } 0 \leq j < N-1 \\ u_{i,0} - u_{i,N-1} & \text{if } j = N-1. \end{cases} \tag{1.7}$$

Then the discrete total variation of u is defined by:

$$TV(u) = \sum_{0 \leq i,j < N} \sqrt{\left((\nabla u)^1_{i,j} \right)^2 + \left((\nabla u)^2_{i,j} \right)^2}. \tag{1.8}$$

Given a degraded image $v \in \mathbb{R}^{N^2}$(see Eq.(1.1)), the authors of [7] proposed to recover the original image $u \in \mathbb{R}^{N^2}$ as the solution of the constrained minimization problem

$$\begin{cases} \min_{w \in \mathbb{R}^{N^2}} TV(w) \\ \text{suject to } \|Hw - v\|^2 \leq \tau^2. \end{cases} \tag{1.9}$$

The solution image u^* should be as regular as possible in the sense of the total variation, while the residue $Hu - v$ has a prescribed l^2-norm. The constraint of (1.9) prescribes the right variance to $Hu^* - v$ but does not guarantee that it is similar to a real Gaussian white noise, even when $H = Id$ (see thorough details in [8]). Reformatted with a Lagrange multiplier, the preceding problem is related to the following unconstrained problem

$$\min_{w \in \mathbb{R}^{N^2}} TV(w) + \lambda \|Hw - v\|^2, \tag{1.10}$$

where λ is the new parameter (which stands for a Lagrange multiplier).

Notice that (1.9) can be rewritten as:

$$\begin{cases} \min_{w \in \mathbb{R}^{N^2}} TV(w) \\ \text{suject to } |\langle Hw - v, \psi \rangle| \leq \tau, \forall \psi \in \mathcal{D}, \end{cases} \tag{1.11}$$

where we set $\mathcal{D} = \{\psi \in \mathbb{R}^{N^2}, \|\psi\|_2 = 1\}$.

1.3.3 The $TV - l^\infty$ model

Generalizing the R.O.F model and wavelet shrinkage, the author of [9] proposed the following unified variational framework,

$$\begin{cases} \min_{w \in \mathbb{R}^{N^2}} E(w) \\ \text{suject to } |\langle Hw - v, \psi_i \rangle| \leq \tau, \forall i \in I, \end{cases} \tag{1.12}$$

for a certain energy $E(w)$, a finite dictionary $\mathcal{D} = (\psi_i)_{i \in I} \subset \mathbb{R}^{N^2}$ and a positive parameter τ associated to the noise level. Notice that when $E = TV$, $\mathcal{D} = \{\psi \in \mathbb{R}^{N^2} | \|\psi\| = 1\}$, this is just the ROF model; when

$$E(w) = \sum_{i \in I} f_l(|\langle w, \psi_i \rangle|),$$

where f_l are strictly increasing functions and $\mathcal{D}$ is wavelet basis, this is just wavelet soft-shrinkage.

When we take $E = TV$, this leads to the so called $TV - l^\infty$ model which we will study in Chapter 2. Explicitly, the $TV - l^\infty$ model takes the following form

$$\begin{cases} \min_{w \in \mathbb{R}^{N^2}} TV(w) \\ \text{suject to } \|Hw - v\|_{\mathcal{D},\infty} \leq \tau, \end{cases} \tag{1.13}$$

where $\|.\|_{\mathcal{D},\infty}$ is defined by

$$\|u\|_{\mathcal{D},\infty} = \sup_{\psi \in \mathcal{D}} |\langle u, \psi \rangle|,$$

for a dictionary $\mathcal{D} \subset \mathbb{R}^{N^2}$ and the discrete total variation.

This model has, at least, been studied in [9, 10, 11, 12]. The contents of these papers are summarized in the introduction of [12].

In [9, 10, 11, 12], the authors only considered translation-dependant dictionaries and in most of the situation, this leads to a lack of translation-invariance of the restoration result. To overcome this problem, in Chapter 2, we will describe an implementation of the $TV - l^\infty$ model with a translation-invariant dictionary. We will also try to understand how to choose the dictionary, in order to improve the result of (1.13).

1.4 Sparse representation models

As redundant dictionaries are more flexible to incorporate prior information than a single orthogonal basis, the use of sparse representation models with redundant dictionary in image/signal processing is now a rapid growing research field.

1.4.1 Basis Pursuit

The Basis Pursuit model (see [13]) revoked great attention recently in image processing. For instance, it is used for compression, source separation (see [14]) and feature selection for classification (see [15]). In [16], using a Basis Pursuit model with Contourlet dictionary, the authors presented a satisfying denoising result which is comparable to Gaussian Scale Mixtures (GSM) approach of [17]. Many theoretical results have also been established supporting this model. Most of them aim at understanding the equivalence between the common Basis Pursuit model (see Eq.(1.14)) and the search for the sparsest decomposition (see, among others, [18, 19]). Other authors show that the Basis Pursuit model is an efficient way to simplify a complex data distribution (see [20, 21]). In [22], I.Daubechies and al. provided an alternative to linear and quadratic programming techniques via an iterative thresholding algorithm for this model. In [23], the authors proposed an iterative proximal thresholding algorithm to solve the Basis Pursuit model over orthonormal bases. The other paper devoted to the resolution of the usual Basis Pursuit Denoising are [24, 25, 26, 27, 28, 29, 30, 31]. We will detail the content of some of these papers.

The common model named Basis Pursuit takes the form

$$\min_{(\lambda_i)_{i\in I}\in\mathbb{R}^I} \|\sum_{i\in I}\lambda_i\psi_i - v\|^2 + \lambda\sum_{i\in I}|\lambda_i|, \tag{1.14}$$

for a finite dictionary $\mathcal{D} = (\psi_i)_{i\in I}$, $\lambda > 0$, a datum $v \in \mathbb{R}^{N^2}$ and the standard l^2 norm on $\mathbb{R}^{N^2}$, $\|.\|$.

Parallel Coordinate descent (PCD) Algorithm

In [28, 29], the authors proposed a parallel shrinkage approach for solving the Basis Pursuit model. Denoting

$$f((\lambda_i)_{i\in I}) = \|\sum_{i\in I}\lambda_i\psi_i - v\|^2 + \lambda\sum_{i\in I}|\lambda_i|,$$

the algorithm proposed in [28, 29] is described in Table 1.1.

We remark that in fact for all $i \in I$, we have:

$$d_i^k = (\arg\min_{t\in\mathbb{R}} f(\lambda^{k-1} + te_i) - \lambda_i^{i-1})$$

where $(e_i)_{i\in I}$ is the canonical basis of $\mathbb{R}^I$. The authors shows experimentally that the convergence of this parallel shrinkage method is satisfactory.

Iterative Thresholding (IT)

We also present an algorithms described in [22, 24, 25, 26]. This algorithm is easy to describe, given Table 1.1. When all the elements of the dictionary are normalised[1], the

[1]The normalisation is not necessary. It just simplifies the description of this algorithm once the PCD algorithm has been written.

- Initialize $(\lambda_i^0)_{i\in I}$.
- Repeat until convergence (loop in k)
  1. Compute d_i^k, for all $i \in I$:

$$d_i^k = \rho_{\frac{\lambda}{\|\psi_i\|^2}}\left(\lambda_i^k + \frac{1}{\|\psi_j\|_2^2}\langle v - \sum_{j\in I}\lambda_i^k\psi_j, \psi_i\rangle\right) - \lambda_i^k.$$

  2. Compute the optimal step :

$$t^k = \arg\min_{t\in\mathbb{R}} f\left((\lambda_i^k)_{i\in I} + t(d_i^k)_{i\in I}\right).$$

  3. Update λ^{k+1} :

$$\forall i \in I, \lambda_i^{k+1} = \lambda_i^k + t^k d_i^k.$$

Table 1.1: The algorithm, solving (1.14), described in [28].

algorithm in [22, 24, 25, 26] indeed corresponds to the one of Table 1.1 when we always choose $t^k = 1$. It therefore consist in an iterative thresholding.

In Chapter 5, we will propose a variant of the Basis Pursuit model and give a dual approach to solve this new model.

1.4.2 Matching Pursuit, OMP

Matching pursuit (MP) was first proposed in the image processing domain in [32] and [33]. Below we briefly present the basic ideas of MP: we are looking for a linear expansion approximating the analyzed signal/image $v \in \mathbb{R}^{N^2}$,

$$v \approx \sum_{i=1}^{M} \lambda_i \psi_i \tag{1.15}$$

in terms of atoms chosen from a large and redundant set (a dictionary $\mathcal{D} = (\psi_i)_{i\in I} \subset \mathbb{R}^{N^2}$ with $\|\psi_i\|_2 = 1, \forall i \in I$). The problem of choosing M atoms, which explain most of energy of a given image, is NP-hard, i.e. computationally intractable. MP provides a sub-optimal solution to this problem. It is obtained with the help of an iterative procedure. In the first step of the iterative algorithm we choose the atom which gives the largest scalar product with the image. The iterative procedure is repeated on the subsequent residue $R^n v$ (for details, see Table 1.2).

The procedure converges to v (see [32]):

$$v = \sum_{n=0}^{+\infty} \langle R^n v, \psi_{\gamma_n}\rangle \psi_{\gamma_n}, \tag{1.16}$$

and conserves the image's energy (see [32])

$$\|v\|^2 = \sum_{n=0}^{M-1} |\langle R^n v, \psi_{\gamma_n}\rangle|^2 + \|R^M v\|^2. \tag{1.17}$$

- Set $R^0 v = v$.
- Iterate (loop in n)
  1. Compute:
  $$\gamma_n = \arg\max_{i \in I} |\langle R^n v, \psi_i \rangle|.$$
  2. Sub-decompose:
  $$R^{n+1} v = R^n v - \langle R^n v, \psi_{\gamma_n} \rangle \psi_{\gamma_n}.$$

Table 1.2: MP algorithm

If we denote V the closed linear span of the vectors in $\mathcal{D}$, i.e.

$$V = Span\{\mathcal{D}\} \tag{1.18}$$

and W the orthogonal complement of V in $\mathbb{R}^{N^2}$, denoting also the orthogonal projector over V and W by P_V and P_W respectively, we have:

Theorem 2 *(Mallat and Zhang [32]) Let $v \in \mathbb{R}^{N^2}$. The residue $R^n v$ defined by the induction Eq.*(1.2) *satisfies*

$$\lim_{n \to +\infty} \|R^n v - P_W v\| = 0. \tag{1.19}$$

Hence

$$P_V v = \sum_{n=0}^{+\infty} \langle R^n v, \psi_{\gamma_n} \rangle \psi_{\gamma_n}, \tag{1.20}$$

and

$$\|P_V v\|^2 = \sum_{n=0}^{+\infty} |\langle R^n v, \psi_{\gamma_n} \rangle|^2. \tag{1.21}$$

Usually in applications (eg. image restoration or image compression), we take the M-first terms as the result u:

$$u = \sum_{n=0}^{M-1} \langle R^n v, \psi_{\gamma_n} \rangle \psi_{\gamma_n},$$

where M is predefined. Another possibility is to stop the process once the residue $R^M v$ attains a certain predefined level δ i.e.

$$\|R^M v\|_2 \leq \delta.$$

The so called Orthogonal Matching Pursuit (OMP) is slightly different from MP (see [34]). With more computations, this method attains a faster convergence approximation than MP. It is also an iterative procedure applied on the subsequent residue $R^n v$. The details of OMP are given in Table 1.3.

- Set $R^0 v = v$.
- Repeat until convergence (loop in n)
  1. Compute:
  $$\gamma_n = \arg\max_{i \in I} |\langle R^n v, \psi_i \rangle|,$$
  $$V^n = Span\{\psi_{\gamma_0}, \ldots, \psi_{\gamma_n}\}.$$
  2. Sub-decompose:
  $$R^{n+1} v = R^n v - P_{V^n} R^n v.$$

Table 1.3: OMP algorithm

1.5 Non-local algorithm and dictionary learning

Most of natural images contain a lot of redundant information. By this, we mean that every small window in a natural image is similar to many windows in the same image. More generally, the collection of small windows of same size in a natural image has a sparse representation over a certain dictionary. The image processing models which take advantage of this kind of redundancy information have better performance.

1.5.1 NL-means

The so called Non-Local means (NL-means) algorithm introduced in [4] can be given by a simple closed formula. Let u be defined in a bounded domain $\Omega \subset \mathbb{R}^2$, then

$$NL(u)(x) = \frac{1}{C(x)} \sum_y e^{-\frac{G_a * |u(x+\cdot) - v(y+\cdot)|^2(0)}{h^2}} u(y) \tag{1.22}$$

where $x \in \Omega$, G_a is a Gaussian kernel of standard variation a, h acts as a filtering parameter and $C(x) = \sum_z e^{-\frac{G_a * |u(x+\cdot) - v(z+\cdot)|^2(0)}{h^2}}$ is the normalization factor. In order to make Eq.(1.22) clear, we recall that

$$G_a * |u(x+\cdot) - v(y+\cdot)|^2(0) = \sum_t G_a(t) * |u(x+t) - v(y+t)|^2. \tag{1.23}$$

As NL-means incorporate non-local information, it is natural that this leads to a successful image restoration approach.

1.5.2 Dictionary learning

The approximation performances using redundant expansions rely strongly on choosing dictionaries adapted to images. For natural high-dimensional data, the statistical dependencies are, most of the time, not obvious. The data-driven learning of domain-specific overcomplete dictionaries is a recent popular problem in approximation theory.

In [35], the authors proposed to learn an environmentally adapted dictionary by developing algorithm which iterates between a representative set of sparse representations

found by variants of FOCUSS and an update of the dictionary using these sparse representations. In experiments with natural images, they showed that learned overcomplete dictionaries have higher coding efficiency than complete dictionaries; that is, images encoded with an overcomplete dictionary have both higher compression (fewer bits per pixel) and higher accuracy (lower mean square error).

In many situations, the basis elements are shift invariant, thus the learning process should try to find the best matching filters. In this regard, the authors of [36] presented an algorithm for learning iteratively generating functions that can be translated at all positions in the images to generate a highly redundant dictionary.

K-SVD

In [37], the authors presented a modification of the K-means clustering process, K-SVD algorithm to learn dictionary. This is an iterative method that alternates between sparse coding of the examples based on the current dictionary and a process of updating the dictionary atoms to better fit the data. In [38], the authors developed K-SVD into an image denoising method via sparse and redundant representations over the learned dictionary ([38]). After comparing with the leading denoising method using Gaussian Scale Mixtures approach in the wavelet domain of [17], it claims state-of-the-art denoising performance.

In this subsection, we will briefly review the main mathematical framework of K-SVD denoising method, as this is one of starting point of Chapter 4. First let the clean image u be written as a column vector of length N^2. Considering patches of size $\sqrt{n} \times \sqrt{n}$, we assume that all the patches in the clean image u admit a sparse representation in a certain basis. Addressing the denoising problem as a sparse decomposition technique for each patch leads to the following energy minimization problem:

$$\{\hat{\alpha_{i,j}}, \hat{\mathcal{D}}_0, u\} = \arg \min_{\mathcal{D}_0, \alpha_{ij}, w} \gamma \|w - v\|_2^2 + \sum_{i,j} \mu_{i,j} \|\alpha_{i,j}\|_0 + \sum_{i,j} \|\mathcal{D}_0 \alpha_{i,j} - R_{i,j} w\|_2^2, \tag{1.24}$$

where for $(i,j) \in \{0, 1, \ldots, N - \sqrt{n}\}^2$ fixed, $\|\alpha_{i,j}\|_0$ stands for the number of non-zero coefficients in the K-dimension column vector $\alpha_{i,j}$. In order to avoid confusing with the dictionary used in the post-processing process of Chapter 4, we use the symbols $\mathcal{D}_0, \hat{\mathcal{D}}_0$ to represent the matrices containing the dictionary.

In (1.24), u is the estimator of a hidden image and the dictionary $\hat{\mathcal{D}}_0 \in \mathbb{R}^{n \times K}$ is an estimator of the best dictionary which gives the sparsest representation of the patches associated to the restored image. The index (i,j) indicates the position of the patch in the image. The binary matrix $R_{i,j}$ of $n \times N^2$ extracts the square patch of size $\sqrt{n} \times \sqrt{n}$ at coordinate (i,j) from the image represented by a column vector w of size N^2. $(\mu_{i,j})$ are the hidden parameters which are implicitly fixed by the method.

The first term of (1.24) demands a proximity between u and v. The second and the third term both give the image prior. This regularization term assumes that every patch of a natural image has a sparse representation in $\hat{\mathcal{D}}_0$. The second term ensures the sparsest representation, and the third term forces the consistency of the decomposition.

The approximation method for solving (1.24) is presented in Table 1.4. When using this algorithm in Chapter 4, following with the work of [38], we assume that σ is known and we set $J = 10$, $C = 1.15$ (these values are experimentally tuned in [38], another more theoretical choice based on Rayleigh law, see [39] for C is 0.93).

Task: Denoise a given image v.
Parameters: n-block size, K-size of first dictionary, J-number of K-SVD iterations, C-noise gain, γ-Lagrange parameter.

1. Set $u = v$, $\mathcal{D}_0$=overcomplete DCT dictionary.

2. Repeat J times:

- Sparse Coding Stage:
  Use OMP pursuit algorithm to compute the representation vectors $\alpha_{i,j}$ for every fixed patch $R_{i,j}u$, by approximating the solution of

$$\min_{\alpha_{i,j}} \|\alpha_{i,j}\|_0 \text{ subject to } \|R_{ij}u - \mathcal{D}_0\alpha_{i,j}\|_2^2 \leq (C\sigma)^2.$$

- Dictionary Update Stage:
  For each column $l = 1, 2, \ldots, K$ in $\mathcal{D}_0$, update it by

  - Find the set of patches that use this atom, $\omega_l = \{(i,j)|\alpha_{i,j}(l) \neq 0\}$
  - Find every index $(i,j) \in \omega_l$, compute its representation error

$$e_{i,j}^l = R_{i,j}u_{i,j} - \sum_{m \neq l} d_m \alpha_{i,j}(m)$$

  - Set E_l as the matrix whose columns are $\{e_{i,j}^l\}_{(i,j)\in\omega_l}$
  - Apply SVD decomposition $E_l = U \triangle V^T$. Choose the updated dictionary column $\tilde{d}_l$ to be the first column of U. Update the coefficient values $\{\alpha_{i,j}(l)\}_{i,j} \in \omega_l$ to be entries of V multiplied by $\triangle(1,1)$.

3. Set:

$$u = (\gamma I + \sum_{i,j} R_{i,j}^T R_{i,j})^{-1}(\gamma v + \sum_{i,j} R_{i,j}^T \mathcal{D}_0 \alpha_{i,j})$$

Table 1.4: K-SVD algorithm for denoising. σ is known.

1.6 Image decomposition

The separation of a natural image into semantic parts plays an important role in applications such as image compression, image enhancement, image restoration, and computer vision. Recently, several pioneering papers suggested that such a separation can be achieved based on variational models ([8, 40]) or independent component analysis and sparsity ([41]).

The total variation plays an important role in the Rudin-Osher-Fatemi (ROF) model (see [7] or subsection 1.3.2). In [8], Yves Meyer has recently investigated this model and proposed another space G for oscillating patterns. This space G is very closely related to the dual space of BV. In this space, oscillating patterns have a small norm and this is very useful when we use an energy minimization process.

Thus in [8], the author proposed a G-norm model to replace the ROF model to decompose an image into a geometrical component and a textured component. Later, the authors of [40] proposed a decomposition model which splits an image into three components: the first one containing the structure of the image, the second one the texture of the image, and the third one the noise. This decomposition model relies on the use of three different semi-norms: the total variation for the geometrical component, a negative Sobolev norm for the texture, and a negative Besov norm for the noise.

In [41], combining the Basis Pursuit Denoising (BPDN) algorithm (see [13] or subsection 1.4.1) and the Total-Variation (TV) regularization scheme, the authors presented a method for separating images into texture and piecewise smooth (cartoon) parts, exploiting both the variational and the sparsity mechanisms.

The basic idea presented in [41], is the use of two appropriate dictionaries, one for the representation of textures, and the other for the piecewise-smooth content of natural scene. Both dictionaries are chosen such that they lead to sparse representations over one type of image-content (either texture or piecewise smooth). The use of the BPDN with the two amalgamated dictionaries leads to the desired separation, along with noise removal as a by-product. As the requirement to select proper dictionaries is generally hard, a TV regularization is employed to better control the separation process and to reduce ringing artifacts.

Chapter 2

Translation-invariant dictionary for $TV - l^\infty$ model

2.1 Introduction

This chapter is an extensive version of [42]. In this chapter, we will concentrate our efforts on the $TV - l^\infty$ model to get a restored image $u \in \mathbb{R}^{N^2}$ from a noisy image $v \in \mathbb{R}^{N^2}$ obtained by Eq.(1.1). The details of $TV - l^\infty$ model is given in Section 1.3.3.

The authors of [9, 10, 11, 12] only considered translation-dependent dictionaries such as wavelet or wavelet packet dictionaries and their restoration results lack of translation-invariance. Mathematically, for an image $u \in \mathbb{R}^{N^2}$, we can translate it by (i, j) (i.e. by applying the operator $T_{i,j}$):

$$T_{i,j}u(\cdot,\cdot) = u(\cdot - i, \cdot - j), 0 \leq i, j < N, \tag{2.1}$$

where we periodized the image by setting $u(m + N, n + N) = u(m, n)$.

Suppose O is the restoration operator, O is translation-invariant if and only if, for all $u \in \mathbb{R}^{N^2}$,

$$Ou = T_{-i,-j} \circ O \circ T_{i,j}u. \tag{2.2}$$

Translation-invariance is a natural requirement for image restoration. The result of the restoration of an object in an image should not depend on its location in the image.

The novelty of the current chapter is to provide an implementation of the $TV - l^\infty$ model with a translation-invariant dictionary. We emphasize that the main goal of this chapter is that we will try to understand how to choose the dictionary, in order to improve the results of (1.13). Instead of denoising itself, the mechanism behind of the model i.e. the role of the dictionary in the $TV - l^\infty$ model is more interesting for us. In fact, in this chapter and the upcoming chapter, we will try to answer, at least partially, the open question posed in [43]: for the $TV - l^\infty$ model, given a class of images and a degradation H, how should the dictionary $\mathcal{D}$ be designed, if one is to aim at optimal results?

2.2 The dictionary

2.2.1 From features to dictionary

In order to build the translation-invariant dictionary, we first consider a finite set

$$\mathcal{F}_0 = \{\Psi^k\}_{1 \leq k \leq r}$$

of elements in $\mathbb{R}^{N^2}$. In the remaining of the chapter, we refer to these elements as "features" and we refer to $\mathcal{F}_0$ as the "feature dictionary". Roughly speaking, the features need not to have small support. In the following of this chapter, we will consider both situations.

For any $k \in \{1, \ldots, r\}$ and any index $(i, j) \in \{0, \ldots, N-1\}^2$, we denote the translation of Ψ^k by

$$\Psi^{k,i,j} \triangleq T_{i,j}\Psi^k, \tag{2.3}$$

where $T_{i,j}$ is defined by (2.1).

We then consider the dictionary

$$\mathcal{D} = \{\Psi^{k,i,j}, \text{ for } 1 \leq k \leq r \text{ and } 0 \leq i, j < N\}.$$

In the remaining of the dissertation, we refer to $\mathcal{D}$ as "total dictionary". The total dictionary $\mathcal{D}$ is obviously translation invariant. Moreover, depending on the structure of feature dictionary $\mathcal{F}_0$, it might be rotation invariant, scale invariant,...

Before going on, we need to present two important operators for solving (1.13) with the above dictionary.

The first one calculates all the scalar products $\langle u, \psi\rangle_{\psi\in\mathcal{D}}$ for any $u \in \mathbb{R}^{N^2}$, we call it the decomposition; the other one computes $\sum_{\psi\in\mathcal{D}} \lambda_\psi \psi$ for any set of coefficients $(\lambda_\psi)_{\psi\in\mathcal{D}}$, this is the recomposition. If we calculate these operators in a straightforward manner, the complexity is $O(rN^4)$. Fortunately, as our dictionary is translation-invariant, we can provide a fast calculation method whose complexity is reduced to $O(rN^2 \log N)$.

2.2.2 The decomposition

The decomposition of $u \in \mathbb{R}^{N^2}$ provides the set of values

$$(\langle u, \Psi^{k,i,j}\rangle)_{0\leq i,j<N \text{ and } 1\leq k\leq r}.$$

Notice that, using (2.3), we have, for any $u \in \mathbb{R}^{N^2}$ and any feature $\Psi^k \in \mathcal{F}_0$,

$$\langle u, \Psi^{k,i,j}\rangle = \sum_{m,n=0}^{N-1} u_{m,n}\Psi^k_{m-i,n-j}.$$

So the set of values $(\langle u, \Psi_{k,i,j}\rangle)_{1\leq i,j<N}$, is just $u * \overline{\Psi^k}$, where $*$ stands for the convolution product and $\overline{\Psi^k}_{m,n} = \Psi^k_{-m,-n}$ (remember the images are periodized).

The decomposition can therefore be computed with one Fourier transform and r inverse Fourier transforms, if we memorize the Fourier transforms of the features. The details of the algorithm of decomposition are shown in Table 2.1.

2.2.3 The recomposition

Denoting $\Lambda = (\lambda^k_{i,j})_{0\leq i,j<N \text{ and } 1\leq k\leq r}$, the recomposition takes the following form

$$\pi : \Lambda \in \mathbb{R}^{rN^2} \to \sum_{k=1}^{r}\sum_{i,j=0}^{N-1} \lambda^k_{i,j}\Psi^{k,i,j} \in \mathbb{R}^{N^2}.$$

Using (2.3), we get

$$\pi(\Lambda) = \sum_{k=1}^{r} \lambda^k * \Psi^k,$$

where $\lambda^k = (\lambda^k_{i,j})_{0\leq i,j<N}$. This can be computed with r Fourier transforms and one inverse Fourier transform. The details of the algorithm of recomposition are shown in Table 2.2.

Task: Compute all the decomposition coefficients $\langle u, \Psi^{k,i,j}\rangle$
Remark: $(\mathcal{F}\overline{\Psi^k})_{1\le k\le r}$ have been already computed and stored

1. Compute $\mathcal{F}u$
2. For $k = 1$ to r
   - compute
   $$u^k = \mathcal{F}u \cdot \mathcal{F}\overline{\Psi^k}$$
   - compute
   $$(\langle u, \Psi^{k,i,j}\rangle)_{0\le i,j<N} = \mathcal{F}^{-1}u^k$$

Table 2.1: Details of decomposition algorithm: for input u, $(\mathcal{F}\overline{\Psi^k})_{1\le k\le r}$ and output $(\langle u, \Psi^{k,i,j}\rangle)_{0\le i,j<N,1\le k\le r}$. $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively.

Task: Compute the recomposition $\pi = \sum_{k=1}^{r}\sum_{i,j=0}^{N-1}\lambda_{i,j}^k\Psi^{k,i,j}$
Remark: $(\mathcal{F}\Psi^k)_{1\le k\le r}$ have already been computed and stored

1. Set $\hat{\pi} = 0$
2. For $k = 1$ to r
   - Compute $\mathcal{F}\lambda_{i,j}^k$
   - Compute
   $$w^k = \mathcal{F}\Psi^k \cdot \mathcal{F}\lambda_{i,j}^k$$
   - update
   $$\hat{\pi} \leftarrow \hat{\pi} + w^k$$
3. Compute $\pi = \mathcal{F}^{-1}\hat{\pi}$

Table 2.2: Details of decomposition algorithm: for input $(\mathcal{F}\Psi^k)_{1\le k\le r}$ and $\Lambda = (\lambda_{i,j}^k)_{1\le k\le r \text{ and } 0\le i,j<N} \in \mathbb{R}^{rN^2}$, output the recomposition result $\pi(\Lambda) = \sum_{k=1}^{r}\sum_{i,j=0}^{N-1}\lambda_{i,j}^k\Psi^{k,i,j}$. $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transform and its inverse, respectively.

2.3 Numerical aspects

The discrete total variation of an image $u \in \mathbb{R}^{N^2}$ is defined in (1.8). But in practice, we need the regularization technique which first appeared in [44]. Finally, what we use is:

$$TV(u) = \sum_{i,j=0}^{N-1} \sqrt{(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2 + \epsilon^2}, \tag{2.4}$$

where we let $u_{i,N} = u_{i,0}$, $u_{N,j} = u_{0,j}$ and ϵ is a very small positive (say $\epsilon = 0.001$).

We use a penalty method, in order to solve (1.13). More precisely, we minimize the unconstrained energy

$$E_\lambda(w) \triangleq TV(w) + \lambda \sum_{\Psi \in \mathcal{D}} \varphi_\tau(\langle Hw - v, \Psi \rangle), \tag{2.5}$$

with

$$\varphi_\tau(t) = (\sup(|t| - \tau, 0))^2,$$

and for a large number λ.

This optimization problem is solved by a steepest descent algorithm. In order to get such an algorithm, the main difficulty is to compute the gradient of (2.5). It takes the form

$$\nabla E_\lambda(w) = \nabla TV(w) + \lambda H^* \left(\sum_{\Psi \in \mathcal{D}} \varphi'_\tau(\langle Hw - v, \Psi \rangle)\Psi \right), \tag{2.6}$$

where φ'_τ is the derivative of φ_τ:

$$\varphi'_\tau(t) = \begin{cases} 2(t - \tau) & \text{if } t \geq \tau, \\ 0 & \text{if } |t| < \tau, \\ 2(t + \tau) & \text{otherwise.} \end{cases} \tag{2.7}$$

Compared to the soft thresholding of Eq.(1.2), we have:

$$\varphi'_\tau(t) = 2\rho_\tau(t). \tag{2.8}$$

In order to compute the gradient of the data fidelity term we need to compute the decomposition in $\mathcal{D}$ and a recomposition. These two operations are already detailed in the previous section.

We now give few details about the computation of $\nabla TV(w)$. It can easily be found in the literature (see, for instance,[44]) and can also be calculated directly. We have:

$$\nabla TV(w) = -\nabla \cdot \left(\frac{\nabla w}{\sqrt{|\nabla w|^2 + \epsilon^2}} \right). \tag{2.9}$$

The algorithm for fixed penalization parameter λ is detailed in Table 2.3. The main algorithm that we use to solve (1.13) is given in Table 2.4.

2.4 The Gabor dictionaries for $TV - l^\infty$ Model

From this section on, we will report on experiments where we use Gabor dictionaries in the $TV - l^\infty$ model. This allows many possible choices. Our conclusion is that the choice of the dictionary impact the restoration of textures with similar structures.

Task: Denoise a given image v by minimization Eq.(2.5)

1. initial u with certain method
2. repeat until convergence
   - calculate:
   $$\nabla E_\lambda(u) = \nabla TV(u) + \lambda H^* \left(\sum_{\Psi \in \mathcal{D}} \varphi'_\tau(\langle Hu - v, \Psi \rangle)\Psi \right)$$
   - find the optimal step by dichotomy method :
   $$t = \arg\min_{t \in \mathbb{R}^+} E_\lambda(u - t \cdot \nabla E_\lambda(u))$$
   - update u:
   $$u \leftarrow u - t \cdot \nabla E_\lambda(u)$$

Table 2.3: Penalization algorithm for fixed λ

Task: Denoise a given image v by solving the $TV - l^\infty$ model (Eq.(1.13))

1. set $(\lambda_k)_{k \in \mathbb{N}}, \lambda_k \to +\infty$
2. set u^{λ_0} with R.O.F method or noisy image
3. repeat for $k = 1$ to $+\infty$
   - Use $u^{\lambda_{k-1}}$ as the initial of u for algorithm of Table 2.3
   - Calculate result u of Table 2.3
   - Update:
   $$u^{\lambda_k} \leftarrow u$$
4. return the result
   $$u \leftarrow u^{\lambda_{+\infty}}$$

Table 2.4: Solving the $TV - l^\infty$ model by iterations of penalization processus

In order to make experiments with several kinds of dictionaries, we used the dictionaries made of Gabor functions as feature dictionary. The motivations for this choice are of two natures. First, as will be described in the next section, this allows many possibilities for frequency and spacial localization. Secondly, they are often used to describe textures and we believe that this kind of feature dictionary pursuits to better recover texture information.

The reason for this belief is that the Kuhn-Tucker equation satisfied by the solution u^* to (1.13) is:

$$\nabla TV(u^*) = \sum_{\psi \in \mathcal{D}} \lambda_\psi H^* \psi, \tag{2.10}$$

for some real numbers $(\lambda_\psi)_{\psi \in \mathcal{D}}$ where H^* is the adjoint operator of H (for the proof of (2.10), see Chapter 3). Moreover, if an element ψ is such that $\lambda_\psi \neq 0$, we know that $|\langle Hw - v, \psi \rangle| = \tau$. This means that, in order to solve (1.13), we had to erase, as much as possible, the information modeled by ψ (which is bad). So, for a good dictionary there should exist a sparse representation of $\nabla TV(u^*)$ in

$$H^*(\mathcal{D}) \triangleq \{H^*\psi | \psi \in \mathcal{D}\}.$$

When interpreted in the context of $BV([0, N-1]^2)$ (the space of bounded variation, see, for instance [8]), this means that the dictionary should give a good description of the dual of BV (at least for denoising). The latter is often considered for texture modeling (For definition of dual of BV and other details, see [8] and [45] and references therein).

Meanwhile, Gabor filters, generated by the dimension two Gabor functions, are known as multi-scale, multi-channel spatial-frequency and orientation selective filters. Psychophysiological experiments with Gabor filters for texture analysis have shown their remarkable similarity with the human visual system, i.e., Gabor filters could be conceived as hypothetical structures of neural receptive fields in the visual cortex of human beings (see [46]).

In image processing, Gabor filters are widely used for texture analysis since they contain oscillatory terms, and texture are typical oscillatory patterns (see [8]). It seems therefore natural to use Gabor dictionaries in (1.13).

Notice the above heuristic is confirmed by the experimental results described in Section 2.6: while we tested 12 different dictionaries, they all provide similar results on homogeneous zones and in the vicinity of edges. The only differences occur in textured zones.

2.5 Gabor filters

The considered features are Gabor filters of the form

$$g_{m,n}^{f,\theta} = C_0 e^{-\frac{x^2}{\sigma} - \frac{y^2}{\sigma'}} \cos(2\pi \frac{f\ x}{N}), \tag{2.11}$$

where $f, \theta \in \mathbb{R}$, $x = m\cos\theta + n\sin\theta$, $y = -m\sin\theta + n\cos\theta$, σ and σ' need to be chosen and C_0 is such that the l^2 norm of the features equals to 1. A typical Gabor filter and its Fourier transform are shown in Figure 2.1.

Knowing that the features take the form (2.11), we still need to determine the frequency and angular locations of elements of $\mathcal{F}_0$.

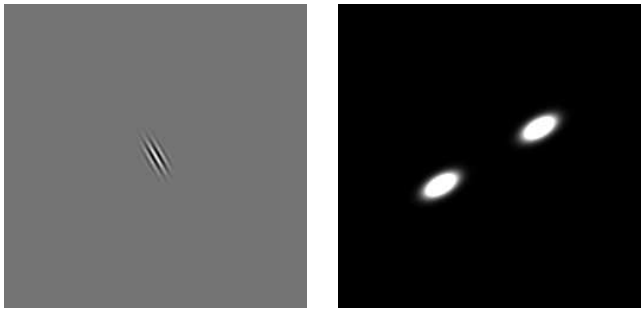

Figure 2.1: Example of Gabor filter. left: Gabor filter; right : Fourier transform of this filter.

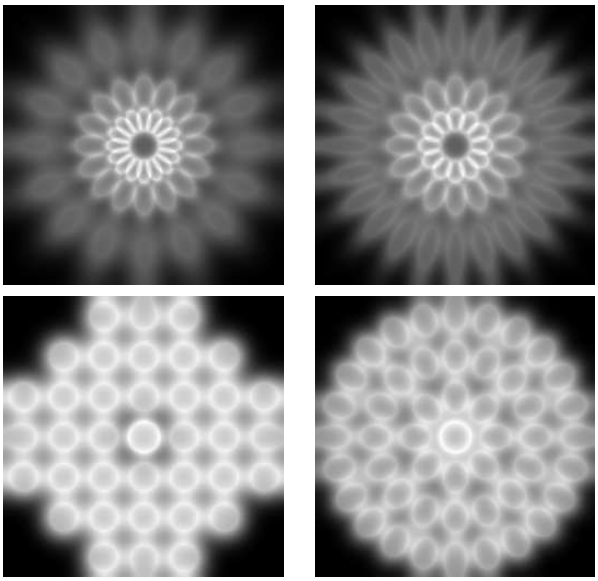

Figure 2.2: Sum of the Fourier transforms of the : up-left : Gabor I features; up-right : features with curvelet scaling; bottom-left : Gabor III features; bottom-right : Gabor II features.

Except for the features described in Section 2.5.4, we consider a finite set of frequencies $\{f_l\}_{0\leq l\leq F}$. We then split the frequency band characterized by f_l in A_l angular sections. For this band, we obtain A_l features

$$g^{f_l,\theta_a}, \tag{2.12}$$

where $\theta_a = \frac{2\pi a}{A_l}$, for $a \in \{0,\dots,A_l-1\}$.

Once these locations are fixed, we obtain a decomposition of the frequency plan. Then σ and σ' are chosen so that the Fourier transforms of the features cover the whole disk of center 0 and radius $\frac{N}{2}$. (Of course, we would gain in covering the whole Fourier domain.) Moreover, σ and σ' are fixed automatically (for details see Appendix) so that the Fourier transforms of any two features do not overlap too much . Notice that, given (2.12), there is no need to adapt the variances σ and σ' to the angular direction. We therefore have a bench of $(\sigma_l,\sigma'_l)_{0\leq l\leq F}$.

The sum of the Fourier transforms of the features described below are represented on Figure 2.2.

2.5.1 Features of type Gabor I

We call Gabor I features those filters built according to (2.12) where, for non-negative integers F and A, we take, for $l \in \{0, \dots, F\}$,

$$\begin{cases} f_l = 0 \text{ and } A_l = 1 & \text{, if } l = 0, \\ f_l = \frac{3}{8} 2^{l-F} \text{ and } A_l = A & \text{, otherwise.} \end{cases}$$

We then take, for $l \in \{0, \dots, F\}$,

$$(\sigma_l, \sigma_l') = \begin{cases} \left(C(\frac{2^F}{N})^2, C(\frac{2^F}{N})^2\right) & \text{, if } l = 0 \\ \left((C(\frac{42^F}{N2^l})^2, C(\frac{A_l}{2\pi f_l})^2\right) & \text{, otherwise,} \end{cases} \tag{2.13}$$

where C is given in the upcoming (2.17). For this schema, the feature dictionary contains

$$\sum_{l=1}^{F} A + 1 = FA + 1$$

filters corresponding to $2FA+1$ cells in the frequency plan. The sum of Fourier transform of filters of this schema (for $(F, A) = (3, 8)$) are shown in the up-left image of Figure 2.2.

2.5.2 Features of type Gabor II

For non-negative integers F and A, we take, for $l \in \{0, \dots, F\}$,

$$\begin{cases} f_l = 0 \text{ and } A_l = 1 & \text{, if } l = 0, \\ f_l = l\frac{N}{2F+1} \text{ and } A_l = lA & \text{, otherwise.} \end{cases}$$

The variances (σ_l, σ_l') equal

$$(\sigma_l, \sigma_l') = \begin{cases} (C(\frac{2F+1}{N})^2, C(\frac{2F+1}{N})^2) & \text{, if } l = 0 \\ (C(\frac{2F+1}{N})^2, C(\frac{A(1F+1)}{2\pi N})^2) & \text{, otherwise,} \end{cases}$$

where C is as in (2.13).

For this schema, the feature dictionary contains

$$\sum_{l=1}^{F} 2^{F-1} A + 1 = (2^F - 1)A + 1$$

filters corresponding to $(2^{F+1} - 2)A + 1$ cells in the frequency plan. The sum of Fourier transform of filters of this schema (for $(F, A) = (3, 4)$) is shown in the bottom-left image of Figure 2.2.

2.5.3 Features with a curvelet scaling

For details on the curvelet scaling, see [10] and references therein. For non-negative integers F and A, we take, for $l \in \{0, \dots, F\}$,

$$\begin{cases} f_l = 0 \text{ and } A_l = 1 & \text{, if } l = 0, \\ f_l = \frac{3N}{8} 2^{l-F} \text{ and } A_l = rd\left(A2^{\frac{l-F}{2}}\right) & \text{, otherwise,} \end{cases}$$

where $rd(t)$ is the closest integer to t.

The variances (σ_l, σ'_l) are determined according to (2.13).

For this schema, the feature dictionary contains

$$N_c = \sum_{l=1}^{F} rd\left(A2^{\frac{l-F}{2}}\right) + 1$$

filters corresponding to $2N_c + 1$ cells in the frequency plan. The sum of Fourier transform of filters of this schema (for $(F, A) = (3, 6)$) is shown in the bottom-right image of Figure 2.2.

2.5.4 Features of Gabor type III

This cosine dictionary, is similar to fully decomposed wavelet packet basis of a given depth. It has the advantage of being translation invariant.

For $F \in \mathbb{N}$, we consider the set of frequency locations

$$\mathcal{F}'_0 = \left\{ \left(i\frac{N}{2F}, j\frac{N}{2F} \right), \text{ with } (i,j) \in \mathcal{O}_F \text{ and } i^2 + j^2 \leq \frac{N^2}{4} \right\},$$

where

$$\mathcal{O}_F = \{(i,j) | i \in \{0, \ldots, F\} \text{ and } j \in \{-F, \ldots, F\}\}$$

if F is pair and

$$\mathcal{O}_F = \{(i,j) | i \in \{1, \ldots, F\} \text{ and } j \in \{-F, \ldots, F\} \text{ or } i = 0 \text{ and } j \in \{0, \ldots, F\}\}$$

if F is odd.

The set of features is then of the form

$$\mathcal{F}_0 = \left\{ e^{-\frac{n^2+m^2}{\sigma}} \cos(2\pi(f_x m + f_y n)), \text{ for } (f_x, f_y) \in \mathcal{F}'_0 \right\},$$

for $\sigma = C(\frac{2F+1}{N})^2$, where C is as in (2.13).

For this schema, the feature dictionary contains $\#\mathcal{F}'_0$ filters corresponding to $2\#\mathcal{F}'_0 - 1$ cells in the frequency plan. The sum of Fourier transform of filters of this schema (for $F = 7$) is shown in the bottom-left image of Figure 2.2.

2.6 Denoising experiments with Gabor dictionaries

We report on denoising experiments of the image "Barbara". The noise variance is $\sigma = 20$. The twelve dictionaries described in Table 2.5 have been tested. For each dictionary, we tuned the parameter τ (in (1.13)) in order to obtain good visual results. As the goal of this chapter is not the denoising performance itself but the understanding of the role of the dictionary, we do not compare the performance of our approach with other denoising method here. However, for the interested reader, we invite them to visit the online images by:

`http://www.math.univ-paris13.fr/~zeng/gabor/`

There we show the results of our twelve dictionaries approaches and the R.O.F model. Clearly, all our twelve approaches are much better than R.O.F model, especially in the region of textures.

type/size	small	medium	large
Gabor I, $(F, A) =$	(3,8)	(3,16)	(3,48)
Gabor II, $(F, A) =$	(3,4)	(5,4)	(8,4)
curvelet, $(F, A) =$	(3,6)	(3,10)	(3,32)
Gabor III, $F =$	7	11	18

Table 2.5: Parameters for the dictionary definitions. The features of small dictionaries are displayed on 2.2.

Figure 2.3: Barbara image. The most interesting zones are in white.

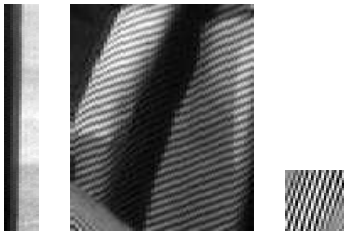

Figure 2.4: left: zone 1; center: zone 2; right : zone 3.

type/size	small	medium	large
Gabor I	27.2375	27.1484	27.1073
Gabor II	27.2617	27.1569	26.8859
curvelet	27.2239	27.1711	27. 0189
Gabor III	27.2449	27.1612	26.8798

Table 2.6: PSNR for zone 1.

type/size	small	medium	large
Gabor I	20.9255	21.443	21.619
Gabor II	22.986	23.3515	23.7118
curvelet	21.0472	21.6531	21.3748
Gabor	20.8129	22.9513	23.1407

Table 2.7: PSNR for zone 2.

In this section we focus on three regions of the images. They corresponds to the white zones on Figure 2.3. The zones are represented in Figure 2.4.

Zone 1 contains an edge. It seems that nearly all the dictionaries give the same kind of results (see Table 2.6). Also we see that there is a clear relation on the $PSNR$,

$$\text{small dictionary} \succeq \text{medium dictionary} \succeq \text{large dictionary}. \tag{2.14}$$

Above the relationship $\succeq$ means '*better than*'. This is not strange. Roughly speaking, since the more elements in the dictionary, the smaller cell of frequency plan, then the larger σ, σ' and the better similarity between the filter and the texture (Beware that when $\sigma, \sigma' \to 0$, the filter tends to a constant). But Zone 1 is an edge, so it is very reasonable that small size dictionary wins for this zone.

Zone 2 contains a texture whose orientation is not related to the shape of region where it lives. Gabor II features, whose spatial localization is almost isotropic, give the best results. Features with a curvelet scaling, whose spatial localization is strongly anisotropic and fits the texture patterns, give the worst. Figure 2.5 compares the result for curvelet (medium) dictionary and Gabor II (medium) dictionary, the result with Gabor II is much better than with curvelet.

On the other side, if we compare the result depends on the size of dictionary, we find the opposite of the relationship (2.14), except for the curvelet dictionary. This tells us that roughly speaking, Zone 2 is of high variance so this time large size dictionary wins.

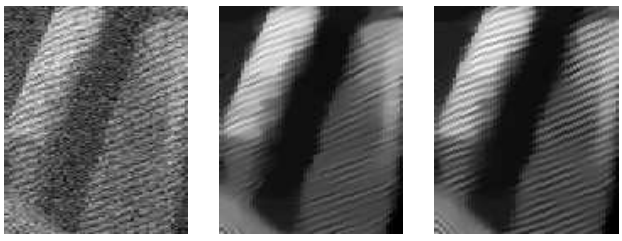

Figure 2.5: left: noisy zone 2; center: result for the medium "curvelet scaling" dictionary, $PSNR = 21.7$; right: result for the medium Gabor II dictionary, $PSNR = 23.4$.

type/size	small	medium	large
Gabor I	19.4346	19.113	21.0173
Gabor II	20.6871	20.0332	21.8354
curvelet	18.7523	21.0859	21.0625
Gabor III	20.4984	17.0148	20.4302

Table 2.8: PSNR for zone 3.

The situation of Zone 3 is more complex, as it seems that it contains a merge of high and low variation information, so there is no clear conclusion for the result depending on the size of the dictionary. On the other hand, Zone 3 contains a texture supported on an elongated region. Moreover, the pattern of the texture fits the shape of the region where it lives. Features with a curvelet scaling or Gabor II give better results than the other features. Our belief is that this region might be rare in natural images. From Table 3, we can see that this time the performances vary more. Visionally, we can barely see the difference between the images (for more clearly comparing, please see online results).

2.7 Conclusion

In this chapter, we have proposed a translation-invariant dictionary approach for the $TV - l^\infty$ model. The experiment confirmed that, to obtain good results of denoising with the $TV - l^\infty$ model, the dictionary must represent the textures (precisely, as we will show in the next chapter, it should be the curvature of textures) well. Hence, when we use the Gabor dictionary, it is better to use Gabor filters whose supports are isotropic (or almost isotropic). Indeed, for represent the texture with a given frequency and living on a fixed support Ω, it is necessary that the support, in space, of Gabor filters allows a "paving" with few elements for the support Ω. For a general class of images, the support Ω is independent of the frequency of texture, it is most reasonable to choose Gabor filters whose support are isotropic. This is a strong argument in favor of the wavelet packets dictionary, which allows in addition to have several sizes of supports in space (for a given frequency) and for which the $TV - l^\infty$ model can be solved quickly.

Appendix

How to determine σ, σ'

We explain how to σ, σ' after a decomposition of the frequency plan through an example. For another possibility of choosing these parameters for dictionary based on Gabor atoms, we refer the reader to [47]. We would like to point out that as the support of the Fourier transform of any Gabor filter is not compact, we can not expect an exact reconstruction over the frequency plan. We are happy if we can cover the frequency plan by the *main energy part* (roughly speaking, by this we mean the high-light part of Figure 2.2) of the Fourier transform of those Gabor filters.

Figure 2.6 shows the the decomposition of frequency plan for Gabor I (see Section 2.5.1) with $(F, A) = (4, 2)$. The frequency plan is thus divided into $2FA + 1 = 17$ cells: one circle in the center, 8 arc-trapezoidal in the first slice, 8 arc-trapezoidal in the

second slice. These 17 cells correspond to $FA + 1 = 9$ filters. The frequency center of a Gabor/Gaussian filter is the center of the cell, its frequency covers two opposite cells.

We determine σ, σ' in two steps.

The first step is, for each cell in the frequency plan, find a suitable ellipse to represent it. As the shapes of the cell are only of 3 types in all our frequency decomposition schemas: arc-trapezoidal, circular and square, our choice is to use an ellipse (neglecting rotation and position):

$$\frac{x^2}{(d/2)^2} + \frac{y^2}{(d'/2)^2} = 1, \tag{2.15}$$

where if the cell is a

1. arc-trapezoidal, we take d, d' the length of the center line and the middle arc of the arc-trapezoidal respectively;
2. circular, we take $d = d'$ as the diameter of the circle;
3. square, we take $d = d'$ as the length of the square.

We therefore take

$$d = d' = \frac{1}{4}N,$$

for the central cell of Figure 2.6, since it is a circular. For the 8 ellipses of the cells in the second slice shown in Figure 2.6, we should take

$$d = \frac{1}{4}N,\ d' = \frac{3\pi}{32}N.$$

The second step is to determine σ, σ' from d, d'. The values of σ, σ' are given by

$$\sigma = \frac{C}{d^2},\ \sigma' = \frac{C}{d'^2}, \tag{2.16}$$

with

$$C = \frac{4N^2 log(a^{-1})}{\pi^2}, \tag{2.17}$$

where a is a constant, in our experiments, we took $a = 0.15$. (The value of C is such that, once normalized at the frequency x', the Fourier transform of $e^{-\frac{x^2}{C(x')^{-2}}}$ equals to a. The value of a is tuned so that the overlaps of the Fourier transforms of the Gabor filters are reasonable (see Figure 2.2)).

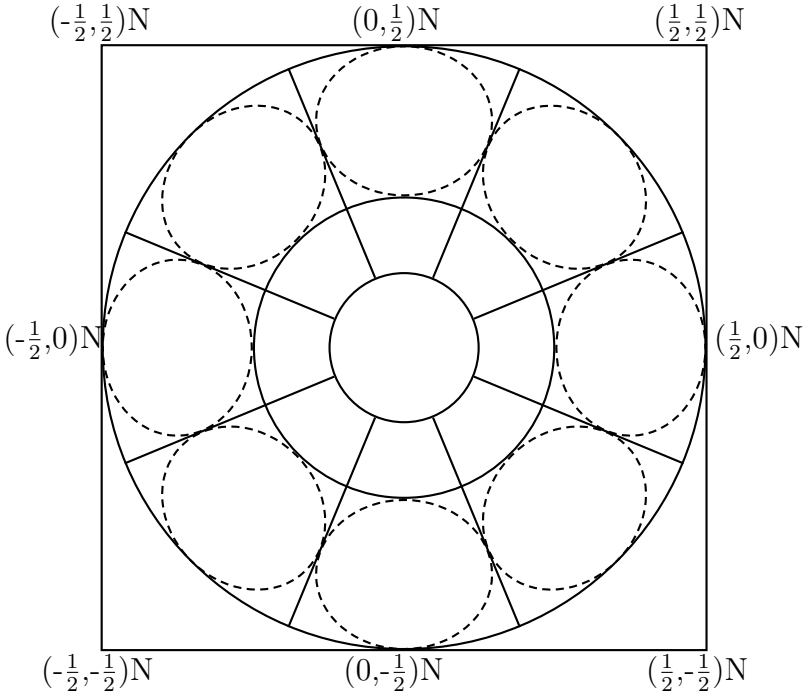

Figure 2.6: Decomposition the frequency plan and choice of σ, σ' to make the Fourier Transform of the Gabor filters cover the corresponding cells.

Chapter 3

Incorporate known features in the general $TV - l^\infty$ model

This chapter contains some theoretical analysis on the general $TV - l^\infty$ model. Then based on these theoretical results, we report denoising and separation experiments with known features.

In fact, we consider a more general model:

Model 1 *For a datum $v \in \mathbb{R}^{N^2}$, a known linear operator H of $\mathbb{R}^{N^2}$ to $\mathbb{R}^{N^2}$, a finite dictionary $\mathcal{D} = (\psi_i)_{i\in I}$ of elements of $\mathbb{R}^{N^2}$, a functional E convex and differentiable on $\mathbb{R}^{N^2}$ and $\tau > 0$, solve:*

$$\begin{cases} \textit{min } E(u) \\ \textit{subject to } \langle Hu - v, \psi_i \rangle \leq \tau, \textit{ for all } i \in I. \end{cases} \tag{3.1}$$

When $E = TV$, to distinguish with the $TV - l^\infty$ model, we refer to this model as the general $TV - l^\infty$ model. Obviously, in the case of $\mathcal{D}$ symmetric, the general $TV - l^\infty$ model is reduced to the $TV - l^\infty$ model.

3.1 Ad-hoc dictionary for $TV - l^\infty$ model

We recall some classical result of convex problem.

3.1.1 Preliminaries

Consider a convex optimization problem:

$$(Q) \begin{cases} \min & f(w), w \in \Omega \\ \text{subject to} & g_i(w) \leq 0, i = 1, \cdots, k, \\ & h_j(w) = 0, \ j = 1, \cdots, m \end{cases} \tag{3.2}$$

with the convex domain $\Omega \subset \mathbb{R}^n$, a convex $f \in C^1(\Omega)$ and some affine functions g_i and h_i, i.e.

$$h(w) = Aw - b$$

for some matrix A and vector b.

Definition 3 *The Lagrangian function of the problem* (Q) *is*

$$L(w,\alpha,\beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{j=1}^{m} \beta_j h_j(w), \tag{3.3}$$

where $\alpha_i \geq 0 (1 \leq i \leq k)$ *and* $\beta_j \in \mathbb{R} (1 \leq j \leq m)$.

Theorem 4 *(Kuhn-Tucker)The necessary and sufficient conditions for a normal point* $w^* \in \mathbb{R}^n$ *to be an optimum of Problem* Q *are the existence of* α^*, β^* *such that*

$$\begin{aligned} \frac{\partial L}{\partial w}(w^*,\alpha^*,\beta^*) &= 0 \\ \alpha_i^* g_i(w^*) &= 0,\ i = 1,\dots,k \\ h_j(w^*) &= 0,\ j = 1,\dots,m \\ \alpha_i^* &\geq 0,\ i = 1,\dots,k \end{aligned}$$

This theorem can be found in [48] (Theorem 5.21), another version of this theorem is Theorem 28.3 of [49]. Using Kuhn-Tucker theorem, we can easily prove the following Corollary.

Corollary 5 *Given an optimization problem with convex domain* $\Omega \subset \mathbb{R}^n$,

$$\begin{aligned} \textit{min} \quad & E(w), w \in \Omega \\ \textit{subject to} \quad & g_i(w) \leq 0, i = 1, \cdots, k, \end{aligned}$$

where the g_i *are affine functions, and* $E(w)$ *is of* $C^1(\Omega)$ *and convex. Then for any solution* u^* *of this problem, there exits* $(\lambda_1,\dots,\lambda_k) \geq 0$ *s.t.*

$$\nabla E(u^*) + \sum_{i=1}^{k} \lambda_i \nabla g_i(u^*) = 0. \tag{3.4}$$

3.1.2 Analysis on $TV - l^\infty$ model

Applying Corollary 5 to (3.1), we know that there exists Lagrangian parameters $(\lambda_i)_{i\in I} \in \mathbb{R}^{+I}$ such that:

$$\nabla E(u^*) + \sum_{i\in I} \lambda_i H^* \psi_i = 0, \tag{3.5}$$

where u^* is a solution to (3.1).

In particular, when $E = TV$, $\mathcal{D}$ is symmetric, we always have non-negative Lagrangian parameters $(\lambda_i^+)_{i\in I}, (\lambda_i^-)_{i\in I}$ such that:

$$\nabla TV(u^*) + \sum_{i\in I} (\lambda_i^+ - \lambda_i^-) H^* \psi_i = 0.$$

Hence if we denote $\lambda_i = -(\lambda_i^+ - \lambda_i^-), \forall i \in I$, we can obtain:

$$\nabla TV(u^*) = \sum_{i\in I} \lambda_i H^* \psi_i. \tag{3.6}$$

Since H is a linear operator, we can change (3.5) as:

$$-(H^*H)^+ H \nabla E(u) = \sum_{i\in I} \lambda_i \psi_i, \tag{3.7}$$

where $(H^*H)^+$ is the Moore-Penrose inverse of H^*H. This means that $-(H^*H)^+ H \nabla E(u)$ can be expressed by a linear sum of elements of $\mathcal{D}$.

Figure 3.1: Curvature of Lenna image

3.1.3 When $\mathcal{D}$ only contains one element

When $\mathcal{D} = \{\psi\}$ and we want to restore a known image, from (3.7), we find that a wise choice for ψ is (we force $\|\psi\|$=1):

$$\psi = -\frac{(H^*H)^+ H\nabla E(u)}{\|(H^*H)^+ H\nabla E(u)\|}. \tag{3.8}$$

More specially, when $H = Id$ and $E = TV$, since

$$\nabla TV(u) = -\nabla \cdot (\frac{\nabla u}{|\nabla u|}), \tag{3.9}$$

we should take (neglecting a normalization constant):

$$\psi = \nabla \cdot (\frac{\nabla u}{|\nabla u|}). \tag{3.10}$$

This is the curvature of u. We call this dictionary as the ad-hoc dictionary.

3.2 Experiments

3.2.1 Denoising experiments with the ad-hoc dictionary

We report the denoising tests with the ad-hoc dictionary. We add a gaussian additive noise of variation 20 to the famous Lenna image. The dictionary we used here is the curvature of Lenna image, it is shown in Figure 3.1. Beware that this is not a real image restoration experiment as we use the curvature of the ideal image. We use this only to demonstrate the ability of the ad-hoc dictionary.

Figure 3.2 shows the result of the general $TV-l^\infty$ with the ad-hoc dictionary and result of the ROF model. From this figure, we clearly see that the general $TV - l^\infty$ with this dictionary almost perfectly reconstruct the image. Not only the PSNR is very high, the visual effect are much better than the ROF model. The residue image is nearly a Gaussian noise and this is an important index to reflect the performance of the restoration(see [4]).

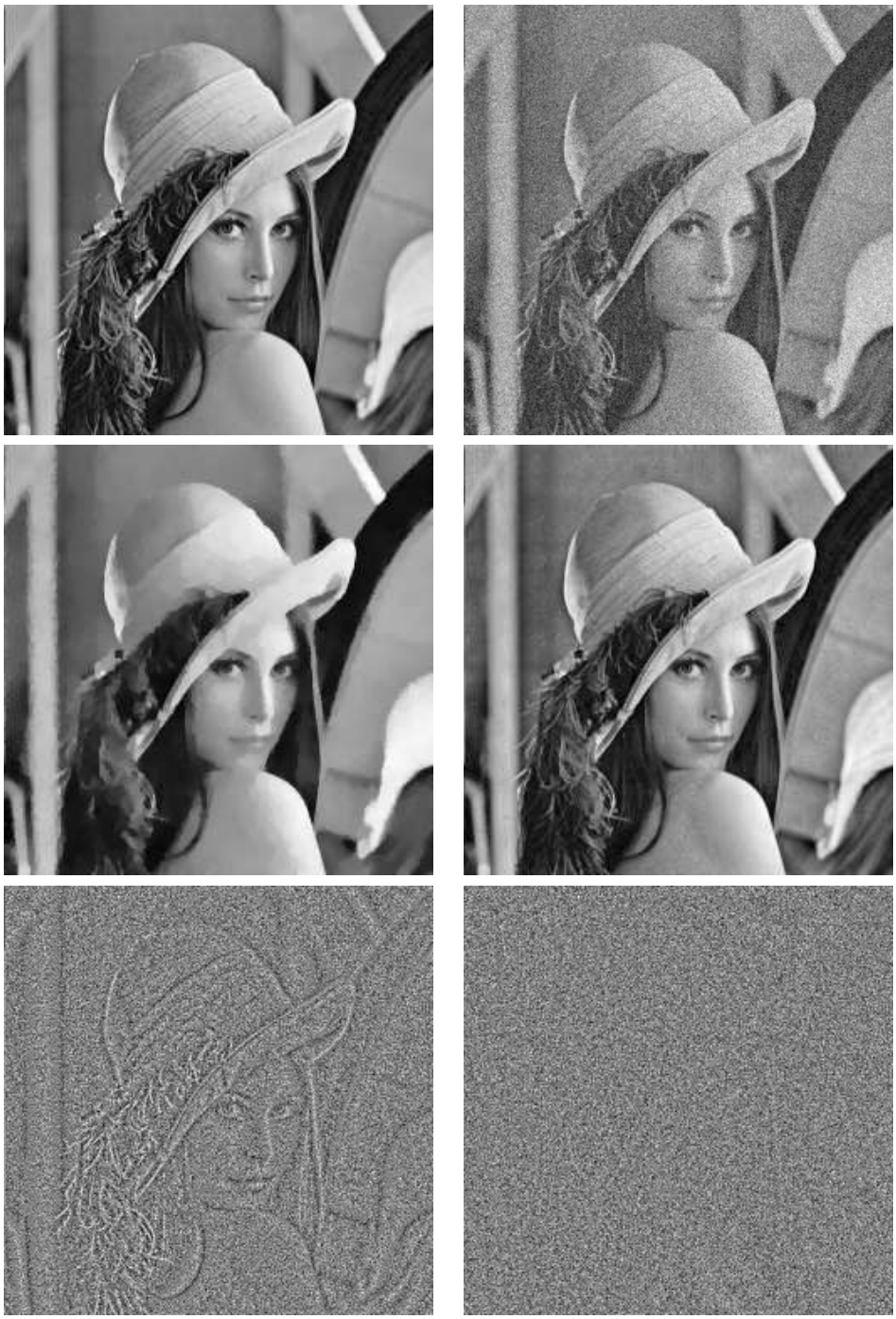

Figure 3.2: Denoising with the ad-hoc dictionary: original image (top-left), noisy image (top-right, $\sigma = 20, \text{PSNR} = 22.11$); result of the ROF model (middle-left, PSNR = 27.66), result of the general $TV - l^\infty$ with the ad-hoc dictionary(middle-right, PSNR = 34.93); residual of the ROF model (bottom-left), residual of the general $TV - l^\infty$ model (bottom-right)

Figure 3.3: Left: clean image; right: noisy image to decompose, it is obtained by adding 20% impulse noise on the left image

3.2.2 First application: image decomposition with known features

The above analysis and experiment illustrate that when we know the curvature of the ideal image, we can get a nearly perfect restoration result. But the problem is that the task of obtaining a nearly perfect curvature is equivalent to get the ideal image.

Fortunately, sometimes we have some prior information about the image. For instance, we may know that the ideal image contains some special structure and we are especially interested in extracting these structures. In this case we can still use the general $TV - l^\infty$ model together with a dictionary reflecting the prior information. We explain this process through an image decomposition example.

Suppose that we are interested in recognizing some letters in a noisy image (see Figure 3.3). We want to separate the image into two parts: one part containing the letters and one part containing the noise and the background information. We hope that the "letter part" contains more information corresponding to letters and less information on the background and noise. Typically, the letter part can be used in a pattern recognition process.

Now suppose that we know the letters. Then we can incorporate these information to construct a feature dictionary $\mathcal{F}_0$. Figure 3.4 displays the known letters and their curvature. We then obtain the total dictionary $\mathcal{D}$ by translating $\mathcal{F}_0$ on the plan (see Section 2.2.1 for details).

Using this total dictionary $\mathcal{D}$, the general $TV - l^\infty$ model provide a fairly good image decomposition result. Figure 3.5 displays the results and residuals of the general $TV - l^\infty$ model and the ROF model. Clearly we see that most of the letter information is contained in the letter part while most of background and noise information is in the residual part.

3.2.3 Second application: denoising with known features

Now we add a Gaussian noise of standard variation 20 to the left image of Figure 3.3. The noisy image is shown in top-right of Figure 3.7. We want to incorporate the prior provided by the known letters to denoise this image.

The feature dictionary contains two parts. The first part contains 9 filters: the curvatures of the letters which are shown in Figure 3.4. The second part contains 13 filters $\{d_1, \ldots, d_{13}\}$ which are from Daubechie-3 wavelet of level 4 and their opposites $\{-d_1, \ldots, -d_{13}\}$. Hence the size of the feature dictionary is $9 + 2 \times 13 = 35$. The 13 Daubechie-3 wavelet filters are shown in Figure 3.6.

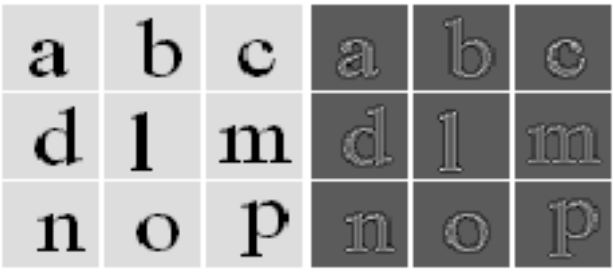

Figure 3.4: Left: ideal letter as prior information; right: basis elements to form the translate-invariant dictionary, it's curvature of the left part

Figure 3.5: Image decomposition results for right image of Figure 3.3. up-left: cartoon part of the ROF model; up-right: noisy-texture part of the ROF model; bottom-left: "letter part" of the general $TV - l^\infty$ model; bottom-right: background and noise part of the general $TV - l^\infty$ model.

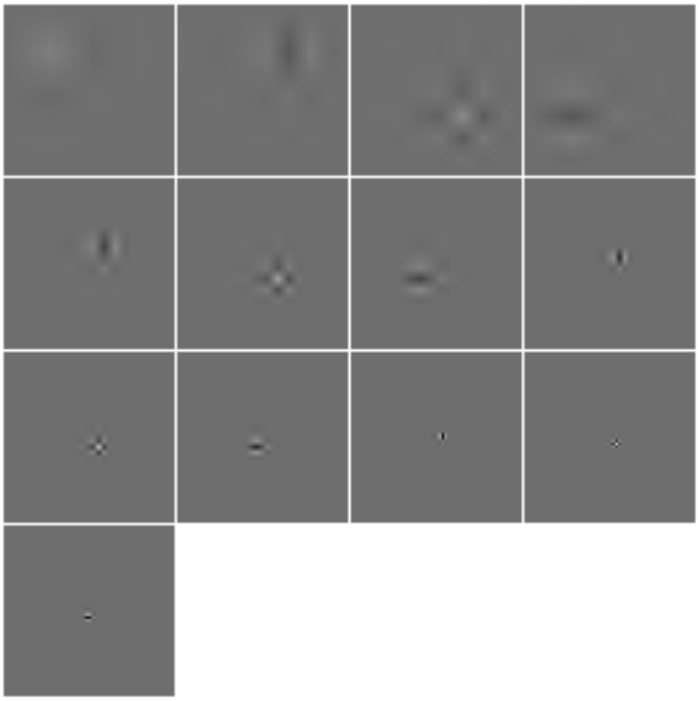

Figure 3.6: The Daubechie-3 wavelet filters.

The denoising results are shown in Figure 3.7. Clearly, with the known features we have a much better performance than the ROF model.

3.3 Discussion

When $H = Id$, if we neglect the interactive between features, we can conclude that a feature of the form $-\nabla TV(f)$ in the dictionary $\mathcal{D}$ will favor the appearance of the pattern f i.e. we have the mechanism:

$$\nabla \cdot \left(\frac{\nabla f}{|f|} \right) \leftrightsquigarrow f. \tag{3.11}$$

Thus if we aim at recovering a special pattern/structure f from the noisy image by using the general $TV - l^\infty$ model, we should add the feature $-\nabla TV(f)$ into the feature dictionary (when the position of this feature is not known) or total dictionary $\mathcal{D}$ (when it has a known position).

When the total dictionary $\mathcal{D}$ contains all the unit-norm vector of $\mathbb{R}^{N^2}$, the $TV - l^\infty$ is the ROF model (see Section 1.3.2). Various experiments have already shown that the ROF model is not good as $TV - l^\infty$ model with wavelet packets or Gabor dictionaries (see [50][42]). This illustrates that the construction of the total dictionary is not simply the union of all possible atoms. Actually, when $\mathcal{D}$ is of large size, we can not neglect the interaction between the elements of $\mathcal{D}$.

Rewriting Eq.(3.6) when $H = Id$, we have:

$$\nabla TV(u^*) = \sum_{i \in I} \lambda_i \psi_i.$$

We know that the solution of the $TV - l^\infty$ model is only involved with the active constraints (where $\lambda_i \neq 0$ and $\langle u^* - v, \psi_i \rangle = \pm\tau$). If the vector $(\lambda_i)_{i \in I}$ is sparse, this will reduce the

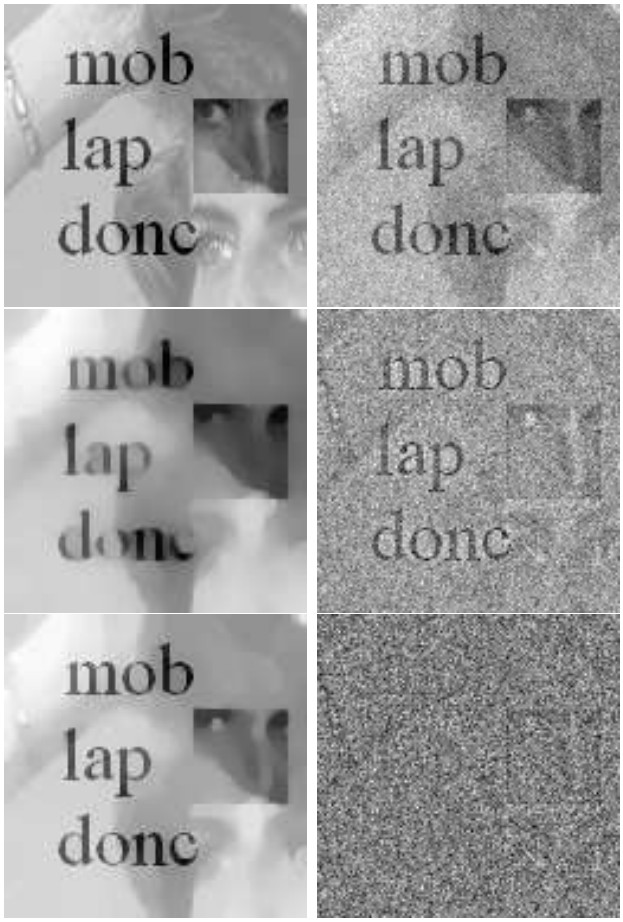

Figure 3.7: Image denoising with known features: top-left: original image, top-right: noisy image with $\sigma = 20, PSNR = 22.0801$; middle-left: denoise result of the ROF model, $PSNR = 24.5559$, middle-right: denoise residual of the ROF model; bottom-left: denoise result of the general $TV - l^\infty$ model, $PSNR = 31.1993$, bottom-right: residual of the general $TV - l^\infty$ model.

possibility of interaction between the atoms. The sparsest situation is that there is only one active atom, and this corresponds to the ad-hoc dictionary and we have already seen that the quality of restoration of this case is fairly good. This illustrates that we should choose a dictionary $\mathcal{D} = (\psi_i)_{i \in I}$ which can give a sparse representation for the curvature of the underlying ideal image.

As pointed out in the previous chapter, the authors of [43] proposed an important open problem: for the $TV - l^\infty$ model, given a class of images and a degradation H, how should the dictionary $\mathcal{D}$ be designed, if one is to aim at optimal results?

Our conclusion is that for $H = Id$, for a certain class $\mathcal{C}_0$ of images, in order to obtain ideal restoration result with the $TV - l^\infty$ model, we should take a dictionary $\mathcal{D}$ which gives sparse representation for the collection of curvature of $\mathcal{C}_0$:

$$\nabla TV(\mathcal{C}_0) \triangleq \{\nabla TV(f) | \forall f \in \mathcal{C}_0\}.$$

We leave the verification of this conclusion for future works. In Chapter 8, we will propose an Expectation-Maximum approach for learning the typical patterns from a certain class of images based some statistical model.

3.4 Conclusion

In this chapter, we presented the experiments in which the dictionary contains the curvatures of known forms (letters). The data-fidelity term of the $TV - l^\infty$ model authorizes the appearance in the residue $w^* - v$ of all the structures, except forms being used to build the dictionary. Thus, we can expect that these forms remain in the result w^* and that the other structures will disappear. Our experiments are carried on a problem of sources separation and confirm this impression. The starting image contains letters (known) on a very structured background (an image). We showed that it is possible, with the $TV - l^\infty$ model, to obtain a reasonable separation of these structures. Finally this work illustrated clearly that the dictionary $\mathcal{D}$ must contain the *curvature* of elements which we seek to preserve and not the elements themselves, as we might think this naively.

Chapter 4

The $TV - l^\infty$ post-processing for K-SVD

4.1 Introduction

In this chapter, we continue to consider the denoising problem (the special case of Eq.(1.1) when $H = Id$): an ideal image $u \in \mathbb{R}^{N^2}$ is observed in the presence of an additive zero-mean Gaussian white noise $b \in \mathbb{R}^{N^2}$ of standard deviation σ. Thus in this chapter, the observed image $v \in \mathbb{R}^{N^2}$ is obtained by:

$$v = u + b. \tag{4.1}$$

Recently, Michael Elad and Michal Aharon proposed an image denoising method via sparse and redundant representations over learned dictionaries (see [38] or Chapter 1). This leads to state-of-the-art denoising performance. The importance of this method is that it can recover most of the information in the noisy image while there is few wash-out effect (for instance, for the face region of the image Barbara). That is to say, it is able to avoid the shortcoming of most TV based denoising methods.

The drawback of this approach could be that it has some "checkboard" effect along edges and it sometimes still loses some texture information especially when the noise level σ is pretty high. It is well known that total variation model can avoid the checkboard effect and the $TV - l^\infty$ model with Gabor dictionary ([42]) has proved to be very effective for texture restoration. So in this chapter, we try to use the $TV - l^\infty$ model as a post-processing procedure for the K-SVD denoising model. Numerical results will show that the post-processing approach is quite effective and it can improve the visual quality of denoised images restored by the K-SVD method, in the meanwhile keeps or even augment the $PSNR$.

4.1.1 The $TV - l^\infty$ algorithm for Denoising

The details of the $TV - l^\infty$ model is presented in Chapter 2. We are interested in the penalization procedure for this model, i.e the algorithm presented in Table 2.3. For clarity, we call that step 2 of Table 2.3 as a TV-penalty procedure. Precisely, a TV-penalty procedure contains 3 steps: one calculation of the gradient, one search of the optimal step, one update of the image. Using this terminology, the algorithm of Table 2.3 can be rephrased as initialize with certain method and then repeat the TV-penalty procedure until convergence.

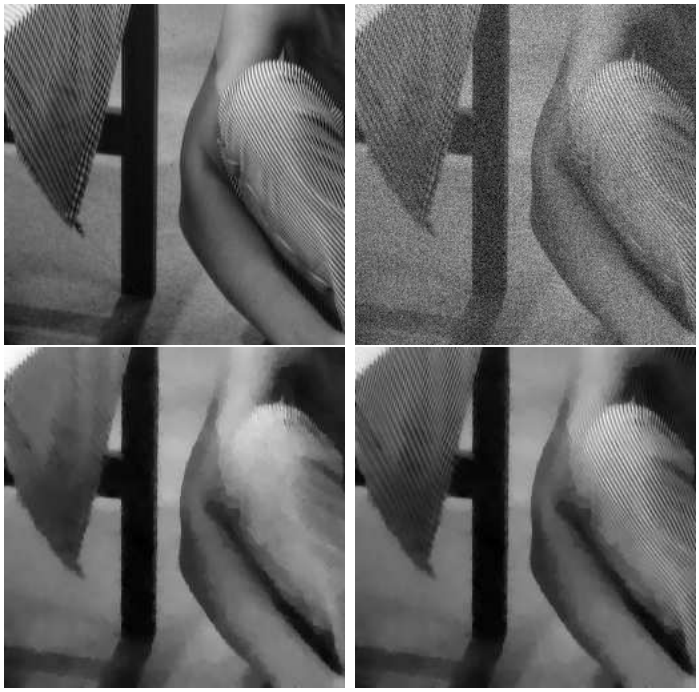

Figure 4.1: Fast construction of the $TV - l^\infty$ model with Gabor dictionary. up-left: a clean patch of image Barbara; up-right: noisy image of this clean patch by Gaussian noise of $\sigma = 20$; bottom-left: result of the ROF model; bottom-right: after one TV-penalty procedure from the ROF result

During the experiments of the $TV - l^\infty$ model, we observed that the TV-penalty procedure has strong reconstruct capability. More precisely, when λ is fixed and reasonably large, if we initialize for the penalization algorithm presented in Table 2.3 well, only one or two TV-penalty procedures are needed for reconstruct most of the lost information.

Figure 4.1 shows this fast reconstruction ability. The up-left is a clean patch of the Barbara image. A Gaussian noise with standard variation 20 was added to the clean image and the noisy image is shown in up-right of Figure 4.1. The bottom-left is the denoising result of the ROF model. We use this image to initialize for the $TV - l^\infty$ model with a certain Gabor dictionary. After only one TV-penalty procedure, we can reconstruct most of the lost texture information. The result is shown as bottom-right of Figure 4.1.

4.1.2 Post-processing approach

Comparing to the $TV - l^\infty$ model, K-SVD model gives higher $PSNR$ and has very few washout effect, especially for the face region of Barbara; comparing to K-SVD model, the $TV - l^\infty$ model has more chance to reduce the checkboard effect. Both methods can recover most part of the texture. Based on these observations, we propose a post-processing approach. In order to solve the denoising task, we first use the K-SVD method to get a fairly good restoration result and then we use this result as the initial for the $TV - l^\infty$ model. Beware that this time we only need to repeat k TV-penalty procedure with k very small.

Task: Denoise a given image v.
Parameters: λ-penalization parameter, ϵ-regularization for curvature, k-number of the TV-penalty procedure, τ- noise control parameter

1. initial u with result of K-SVD (Table 1.4)
2. Repeat k times:
   - Calculate direction of gradient:
   $$w = -\nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \epsilon^2}}\right) + \lambda \sum_{\psi \in \mathcal{D}} \varphi'_\tau(\langle u - v, \psi \rangle)\psi$$
   - Find the optimal step by dichotomy:
   $$s = \arg\min_{t \in \mathbb{R}^+} \left(TV(u - tw) + \lambda \sum_{\psi \in \mathcal{D}} \varphi_\tau(\langle (u - tw) - v, \psi \rangle) \right)$$
   - Update u:
   $$u = u - sw.$$

Table 4.1: General form of Post-processing algorithms.

4.2 Main algorithm

The details of our main algorithm is presented in Table 4.1, where φ' is defined in Eq.(2.7).

The typical choice of λ is between 10^4 and 10^6. $\tau = 3.5\sigma$. The choice of k is discussed in the experiments.

4.3 Experimental results

We report our experiments for $\sigma = 20$, 30 on Barbara and Lenna image. As in [38], we assume that σ is already known or could be estimated from elsewhere. We use Gabor dictionary (Gabor II of ([42]), large size) which contains 145 filters, as we think that it is globally better for restoration. The sum of FFT of all the filters of this dictionary is shown as Figure 4.2.

4.3.1 Noise level of $\sigma = 20$ for Barbara

Our first experiment is to denoise the Barbara image with noise level $\sigma = 20$, the $PSNR$ of the noisy image is 22.0977.

Figure 4.3 shows $TV(u)$ and $PSNR(u)$ with the iteration number k, in the main algorithm (see step 4 of Table 4.1). This Figure tells us that after about 2 or 3 times of the TV-penalty procedure, $TV(u)$ and $PSNR(u)$ both reach their maximums at the same time. So if we aim to get a higher $PSNR$ denoising, we should stop the iteration of the TV-penalty procedure once $TV(u)$ reaches its top.

For our experiment, the highest $PSNR$ is 30.9376, when $k = 2$, this is a slightly higher than K-SVD of Elad (30.8113) which claimed state-of-the-art denoising performance and much higher than the classical Rudin-Osher-Fatemi method (24.6759).

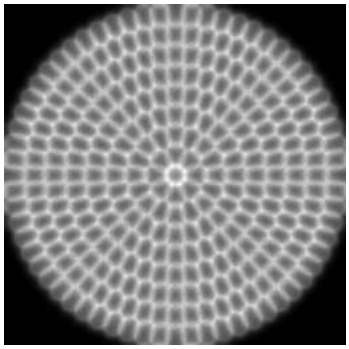

Figure 4.2: Sum of Fourier transforms of the 145 filters in the Gabor II dictionary (large size).

Meanwhile, if we want a better visual quality, we can continue the iteration of the TV-penalty procedure and set $k = 10$ to 15. Figure 4.4 displays a piece of the left-bottom part of the Barbara, with $k = 15$. The visual effect of the new approach is only slightly better than the K-SVD, in the texture region on the desk.

4.3.2 Noise level of $\sigma = 30$ for Barabara

In our experiment, when $\sigma = 30$ the $PSNR$ of the piece of the Barbara image is 18.5448. Figure 4.5 shows the result for the same piece as Figure 4.4. From this Figure we obviously see that our new approach performs better. Both Rudin-Osher-Fatemi method and K-SVD fail to recover the texture of the tablecloths, while our approach still can recover most of the information. And for the left part this piece, Rudin-Osher-Fatemi lost the texture information and K-SVD can recover some of this information. But our new approach recovers more information and the texture is still presented.

Globally,for this level of noise, the $PSNR$ of the noisy image and the result of Rudin-Osher-Fatemi,K-SVD, and our approach are respectively 18.5867, 24.0429, 28.5947, and 28.8376.

4.3.3 Noise level of $\sigma = 20$ for Lenna

We also report our experiment result on Lenna image of size 256×256. For the sake of display, we adopt a different version with the one reported in [38] where the size of Lenna image is 512×512. Beware that the similarly observations hold for larger version of Lenna.

The clean Lenna is shown on the up-left in Fig.4.6. The TV of this image is 13.8457. With a Gaussian noise of standard variation 20, the TV and $PSNR$ of the noisy image is respectively 39.9117 and 22.0823. We show the noisy image as up-right of Fig.4.6. After K-SVD denoising method, a fairly good result is obtained, the TV and $PSNR$ of the restoration image is respectively 10.8710 and 30.4464. We then use our post-processing procedure, with $k = 1$ (again, the value of k is decided automatically), we obtain a new restoration image whose TV is 11.0179 and whose $PSNR$ is 30.4688. Hence, comparing to K-SVD result, the $PSNR$ of our new approach is only very slightly better, but the TV is more better and this hints us that our new approach recovers some structure lost in the K-SVD denosing method. The result of K-SVD and our new approach are shown as bottom-left and bottom-right in Fig.4.6. Clearly, our new approach recovers some texture

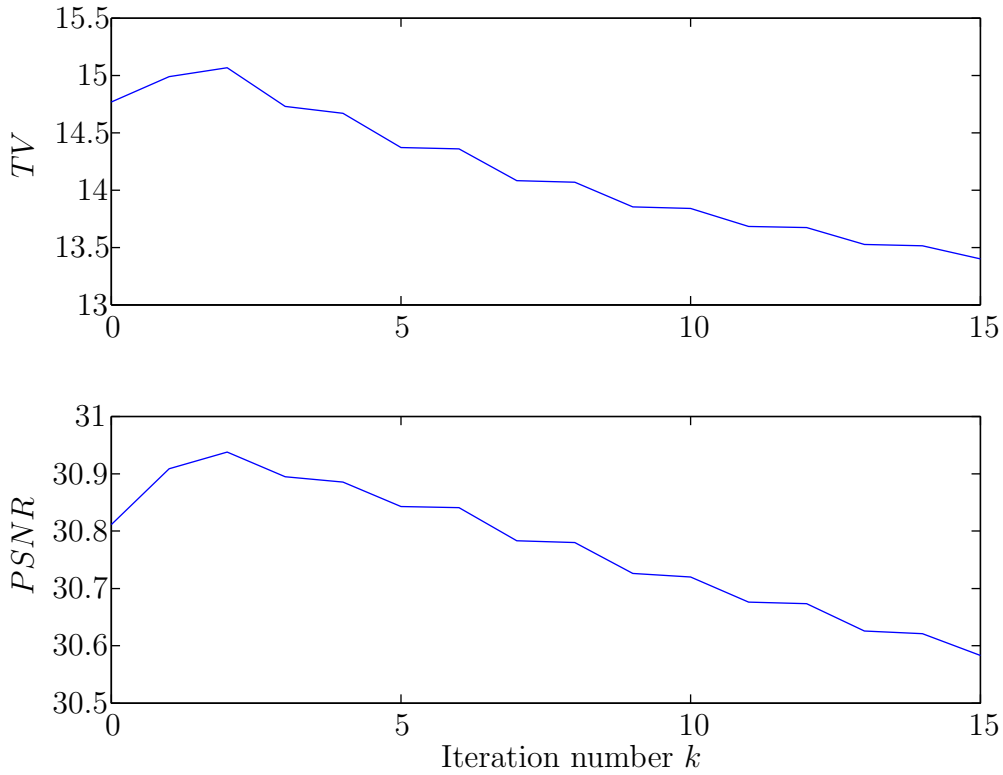

Figure 4.3: $TV(u)$ and $PSNR(u)$ as a function of the iteration number k (see Table 4.1). Note that $k = 0$ is result of K-SVD of Elad.

.

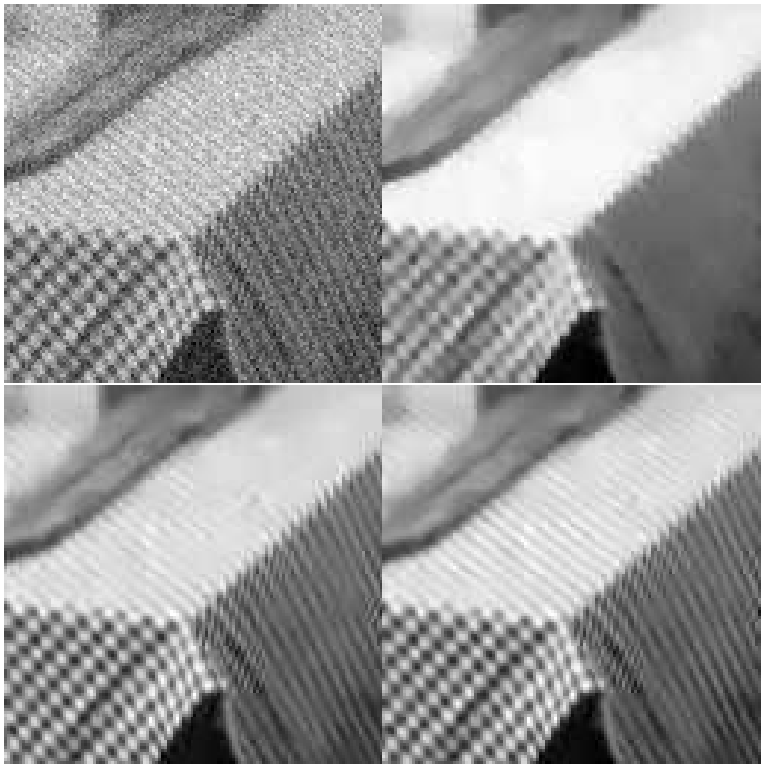

Figure 4.4: Denoising a 128×128 piece of Barbara. From left to right and from top to bottom: noisy image ($\sigma = 20$), PSNR 22.0896; Rudin-Osher-Fatemi, PSNR 24.2663); K-SVD, PSNR 28.9013; Our new approach,PSNR 29.1148.

on the region of the hat while these information are lost after the K-SVD denoising method.

4.4 Conclusion

In this chapter, we presented a work in which we try to integrate the K-SVD method with the $TV - l^\infty$ model. Our starting idea was to use the fact that some iterations of the algorithm which we use to solve the $TV - l^\infty$ model allow to reconstruct the lost structures from the image which we used as the initialization of the algorithm (and whose curvature is present in dictionary). We thus applied some of these iterations to the result of K-SVD and recovered the lost textures well. This allowed a visual gain and an improvement of the $PSNR$.

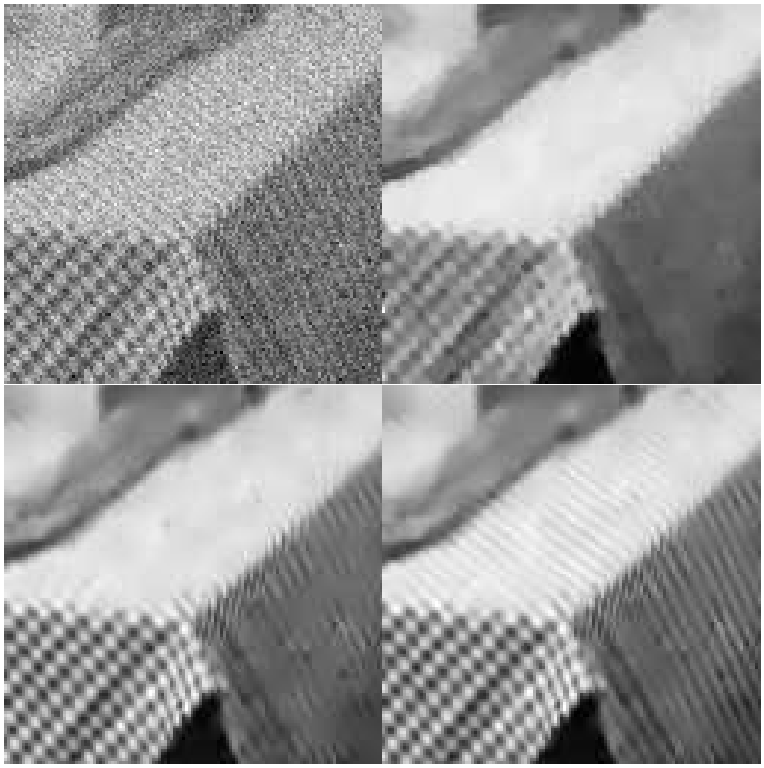

Figure 4.5: Denoising the same piece of Barbara as Figure 4.4. From left to right and from top to bottom: noisy ($\sigma = 30$), PSNR 18.5448; Rudin-Osher-Fatemi, PSNR 23.4331; K-SVD, PSNR 26.4467; Our new approach, PSNR 27.032.

Figure 4.6: Denoising Lenna (size 256×256). From left to right and from top to bottom: clean Lenna image, TV 13.8457; noisy image ($\sigma = 20$), TV 39.9117, $PSNR$ 22.0823; K-SVD result, TV 10.8710, $PSNR$ 30.4464; Our new approach, TV 11.0179, $PSNR$ 30.4688

Chapter 5

Proximal Point Algorithm for Non Negative Basis Pursuit model

This chapter develops an implementation of a Proximal Point Algorithm solving a Non Negative Basis Pursuit Denoising model. The variant imposes a constraint on the l^2 norm of the residual, instead of penalizing it. Thanks to the proximal regularisation (applied to the predual of the Non Negative Basis Pursuit Denoising model), we turn a constrained non differentiable convex problem into a small sequence of smooth concave maximization problems. By smooth, we mean that the functions which are maximized are differentiable and their gradient are Lipschitz.

The algorithm is easy to implement, easier to tune and more general than the algorithm found in the literature (it can be applied to the usual and the Non Negative Basis Pursuit and it does not make any assumption on the dictionary). We prove its convergence to an actual solution of the model and provide convergence rates.

Experiments on image approximation show that the algorithm is simultaneously faster and more accurate than the existing algorithms.

5.1 Introduction

5.1.1 From Basis Pursuit to the new variant

In its most recent form [19, 20], the Basis Pursuit functional is defined by a finite subset of $\mathcal{D} \subset \mathbb{R}^{N^2}$ (called dictionary) $(\psi_i)_{i\in I}$ and takes the form

$$\begin{cases} E(v) = \inf_{(\lambda_i)_{i\in I}} \sum_{i\in I} \lambda_i \\ \text{under the constraints } \lambda_i \geq 0, \forall i \in I, \\ \text{and } \sum_{i\in I} \lambda_i \psi_i = v, \end{cases}$$

for all $v \in \mathbb{R}^{N^2}$.

The ordinary Basis Pursuit model is presented as Eq.(1.14). It can be rewritten under the form

$$\min_{w\in\mathbb{R}^{N^2}} \|w - v\|^2 + \lambda E(w), \tag{5.1}$$

where E is defined with a symmetric dictionary $\mathcal{D}$ i.e. $\mathcal{D} = \{-\psi, \psi \in \mathcal{D}\}$.

The strength of the functional E is that its level sets are scaled versions of the convex hull of $(\psi_i)_{i\in I}$ (see [51, 20]). It is therefore possible to build a functional E that favor the apparition of specific structures; and we have a complete control on these structures. This functional can then be used in optimization problems designed for specific applications.

One drawback of the above functional E is that it favors both the apparition of ψ_i and $-\psi_i$. This might lead to a bad modeling of some structures which only appears with a given sign. For instance when dealing with images of text, the letters are always dark on a brighter background. If, in an approximation, an element of the dictionary representing a letter at a given location appears with a negative sign, it describes something which is not a letter and should be represented by elements of the dictionary devoted to the background. (This holds also for astronomical image, images of faces,...). This led some authors [19, 20] to study the Non Negative Basis Pursuit, where the above regularisation term E is replace by E_{nn} defined, for every $w \in \mathbb{R}^{N^2}$, by

$$\begin{cases} E_{nn}(w) = \min_{(\lambda_i)_{i\in I}} \sum_{i\in I} \lambda_i \\ \text{under the constraints } \lambda_i \geq 0, \forall i \in I, \\ \text{and } \sum_{i\in I} \lambda_i \psi_i = w, \end{cases}$$

for a dictionary $(\psi_i)_{i\in I}$. The level sets of E_{nn} are scaled versions of the convex hull of $(\psi_i)_{i\in I}$. Of course, if $(\psi_i)_{i\in I}$ is symmetric (for all $j \in I$, $-\psi_j \in (\psi_i)_{i\in I}$) one obtains a model similar to the usual Basis Pursuit Denoising model.

Another issue which we wanted to improve in (1.14) concerns the choice of the parameter λ. For practical applications, it is always preferable to solve the model under the form

$$\begin{cases} \min_{w\in\mathbb{R}^{N^2}} E_{nn}(w), \\ \text{under the constraints } \|w - v\| \leq \tau, \end{cases} \tag{5.2}$$

for a parameter $\tau > 0$. Indeed, τ can be tuned automatically, according to some prescribed precision (in approximation) or a known noise level (in denoising).

Notice that, as is well known, there is a correspondence between the parameter λ in (1.14) and the parameter τ in a model of the form (5.2). However, this correspondence depends on the initial data v. For instance, when solving (5.2), for $\lambda = 0.1$, with the translation invariant local cosine dictionary described in Section 5.3.1 and for the initial data "Barbara", "baboon" and "Lenna", the norm of the obtained residual are respectively 0.28, 0.40 and 0.30. If we run the same experiments with $\lambda = 200$, on noisy versions (an additive Gaussian noise of standard deviation 20) of those three images, we obtain respectively 24.06, 28.9, and 25.74. The discrepancy between those numbers does not occur when the model takes the form (5.2).

All these considerations led us to consider a Non Negative Basis Pursuit Denoising model taking the form

$$(D) \begin{cases} \min_{(\lambda_i)_{i\in I}} \sum_{i\in I} \lambda_i \\ \text{under the constraints } \lambda_i \geq 0, \forall i \in I, \\ \text{and } \|\sum_{i\in I} \lambda_i \psi_i - v\| \leq \tau, \end{cases}$$

for a dictionary $(\psi_i)_{i\in I}$, $\tau > 0$ and an initial datum $v \in \mathbb{R}^{N^2}$.

The purpose of the current chapter is to design an efficient algorithm for solving (D).

If $\|v\| \leq \tau$, obviously the unique solution to (D) is $\lambda_i = 0, \forall i \in I$. To avoid this trivial situation, in this chapter, we will always assume that:

$$\|v\| > \tau. \tag{5.3}$$

Throughout this chapter, we denote,

$$\mathbb{R}^{+I} = \{(\lambda_i)_{i\in I} \in \mathbb{R}^I, \forall i \in I, \lambda_i \geq 0\},$$

and we will assume also that the dictionary is such that

$$\forall w \in \mathbb{R}^{N^2}, \exists (\lambda_i)_{i\in I} \in \mathbb{R}^{+I} \text{ and } w = \sum_{i\in I} \lambda_i \psi_i,$$

or equivalently (see Proposition 6) that

$$\{w, \forall i \in I, \langle w, \psi_i \rangle \leq 1\} \text{ is bounded.}$$

When this hypothesis holds, (D) and the upcoming problem (P) have a solution.

Proposition 6 *Suppose $\mathcal{D} = (\psi_i)_{i\in I} \subset \mathbb{R}^{N^2}$. The following two assertions are equivalent:*

A.

$$\forall w \in \mathbb{R}^{N^2}, \exists (\lambda_i)_{i\in I} \in \mathbb{R}^{+I} \text{ and } w = \sum_{i\in I} \lambda_i \psi_i;$$

B.

$$\{w, \forall i \in I, \langle w, \psi_i \rangle \leq 1\} \text{ is bounded.}$$

Proof. see Appendix. □

5.1.2 Simple analysis on the solution to (D)

Proposition 7 *If the dictionary $\mathcal{D}$ satisfies the assertion of Proposition 6, then a solution $(\lambda_i)_{i\in I}$ to (D) exists and the value $\sum_{i\in I} \lambda_i \psi_i$ does not depend on the choice of $(\lambda_i)_{i\in I}$.*

Proof. As the assertion of Proposition 6 holds, we know that there exists $(\lambda_i^0)_{i\in I} \in \mathbb{R}^{+I}$ such that

$$v = \sum_{i\in I} \lambda_i^0 \psi_i.$$

Thus the feasible set is non-empty (at least it contains $(\lambda_i^0)_{i\in I}$), and if we denote

$$A_0 = \sum_{i\in I} \lambda_i^0,$$

then (D) is equivalent to:

$$(D_0) \begin{cases} \min_{(\lambda_i)_{i\in I}} \sum_{i\in I} \lambda_i \\ \text{under the constraints } 0 \leq \lambda_i \leq A_0, \forall i \in I, \\ \text{and } \|v - \sum_{i\in I} \lambda_i \psi_i\| \leq \tau. \end{cases}$$

The Problem (D_0) can be regarded as minimizing a continuous function over a non-empty compact set, so a solution to (D_0) (hence to (D)) exists.

Now if $\|v\| \leq \tau$, then it is obvious that the unique solution to (D) is $(\lambda_i)_{i\in I} = 0$. So we only need to consider the case $\|v\| > \tau$.

The first assertion is that for any solution $(\lambda_i)_{i\in I}$ to (D), the constraint $\|v - \sum_{i\in I} \lambda_i \psi_i\| \leq \tau$ must be active. Indeed, if this is not true, then the solution to (D) is also solution to

$$(D_1) \begin{cases} \min_{(\lambda_i)_{i\in I}} \sum_{i\in I} \lambda_i \\ \text{under the constraints } \lambda_i \geq 0, \forall i \in I, \end{cases}$$

i.e. $(\lambda_i)_{i\in I} = 0$ but this is impossible since $(\lambda_i)_{i\in I} = 0$ is not in the feasible set of (D). This implies that for any solution $(\lambda_i)_{i\in I}$ to (D), we always have:

$$\|v - \sum_{i\in I} \lambda_i \psi_i\| = \tau.$$

Now we want to prove the uniqueness of $\sum_{i\in I} \lambda_i \psi_i$. Suppose that $(\lambda_i^1)_{i\in I}, (\lambda_i^2)_{i\in I}$ are two different solutions to (D). Then we know that:

$$\sum_{i\in I} \lambda_i^1 = \sum_{i\in I} \lambda_i^2,$$

and

$$\|v - \sum_{i\in I} \lambda_i^1 \psi_i\| = \|v - \sum_{i\in I} \lambda_i^2 \psi_i\| = \tau. \tag{5.4}$$

Considering $\frac{1}{2}(\lambda_i^1 + \lambda_i)_{i\in I}$, from the fact that:

$$\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in \mathbb{R}^{N^2},$$

we know that $\frac{1}{2}(\lambda_i^1 + \lambda_i^2)_{i\in I}$ is in the feasible set of (D). Moreover since:

$$\sum_{i\in I} \frac{1}{2}(\lambda_i^1 + \lambda_i^2) = \sum_{i\in I} \lambda_i^1,$$

we know that $\frac{1}{2}(\lambda_i^1 + \lambda_i)_{i\in I}$ is also a solution to (D). Hence:

$$\|v - \sum_{i\in I} \frac{1}{2}(\lambda_i^1 + \lambda_i^2)\psi_i\| = \tau.$$

This lead to:

$$\|(v - \sum_{i\in I} \lambda_i^1 \psi_i) + (v - \sum_{i\in I} \lambda_i^2 \psi_i)\| = \|v - \sum_{i\in I} \lambda_i^1 \psi_i\| + \|v - \sum_{i\in I} \lambda_i^2 \psi_i\|.$$

Moreover, in Euclidian space$\|x+y\| = \|x\| + \|y\|$ holds if and only if there exits a positive β_0 such that $x = \beta_0 y$ or $y = \beta_0 x$. Without loss of generality, we assume that:

$$v - \sum_{i\in I} \lambda_i^1 \psi_i = \beta_0 (v - \sum_{i\in I} \lambda_i^2 \psi_i),$$

for $\beta_0 \geq 0$. Taking the norm on both sides of the above equation and using Eq.(5.4), we know that $\beta_0 = 1$ and then

$$\sum_{i\in I} \lambda_i^1 \psi_i = \sum_{i\in I} \lambda_i^2 \psi_i.$$

This finishes the proof. □

5.1.3 Sketch of this chapter

In section 5.2, we build the simplest version of our algorithm. The algorithm solves a problem (P) whose dual problem is (D). The problem (P) is stabilized by a Proximal regularization (see Section 5.2.2 and 5.2.3). Then, some calculations permits to obtain

closed form formulas for some of the necessary computations of the algorithms (see Section 5.2.4). They also guarantee its convergence (see Section 5.2.5). The simplest version of the algorithm is given in Section 5.2.6. It is easy to implement. Some variations around this algorithm are proposed in Section 5.2.7.

Then, some experiments are explained and commented in Section 5.3. The experiments are described in Section 5.3.1, the practical convergence of the proposed algorithms is studied in Section 5.3.2, as a bibliography on existing algorithms is made in Chapter 1, we compare our algorithms to the main existing algorithms. This comparison shows that our algorithms are far more accurate than the existing algorithms.

5.2 Building algorithms

We consider the optimization problem below and will show that the corresponding dual problem takes the form (D) above. As in the preceding section $v \in \mathbb{R}^{N^2}$ is the initial datum, $\mathcal{D} = (\psi_i)_{i\in I}$ is a finite subset of $\mathbb{R}^{N^2}$ (called dictionary) and $\tau > 0$.

$$(P)\begin{cases} \min_{w\in\mathbb{R}^{N^2}} \|w\| - \frac{1}{\tau}\langle w, v\rangle \\ \text{under the constraints } \forall i \in I, \langle w, \psi_i\rangle \leq 1. \end{cases}$$

5.2.1 Basic property of Problem (P)

We first present a simple lemma.

Lemma 8 *If the dictionary $\mathcal{D}$ satisfies the assertion of Proposition 6 and $v \in \mathbb{R}^{N^2}$ is nonzero, then*

$$\max_{i\in I}\langle v, \psi_i\rangle > 0.$$

Proof. Since the dictionary $\mathcal{D}$ satisfies the assertion of Proposition 6, there exists $(\lambda_i)_{i\in I} \in \mathbb{R}^{+I}$ such that:

$$v = \sum_{i\in I} \lambda_i \psi_i.$$

Thus:

$$\|v\|^2 = \langle v, v\rangle = \sum_{i\in I} \lambda_i \langle \psi_i, v\rangle \leq \max_{i\in I}\langle v, \psi_i\rangle \sum_{i\in I} \lambda_i.$$

This implies that:

$$\max_{i\in I}\langle v, \psi_i\rangle \geq \frac{\|v\|^2}{\sum_{i\in I}\lambda_i} > 0.$$

□

Proposition 9 *If the dictionary $\mathcal{D}$ satisfies the assertion of Proposition 6, then the solution to (P) exists and*

1. *if $\|v\| < \tau$, then the solution to (P) is unique and it is just $w = 0$;*
2. *if $\|v\| = \tau$, then the solutions to (P) are the segment $[0, \gamma_0 v]$ for an approximate positive number γ_0;*
3. *if $\|v\| > \tau$, then the solution to (P) is unique.*

Proof. When the assertion of Proposition 6 holds, then the feasible set of (D)

$$\{w \in \mathbb{R}^{N^2} | \forall i \in I, \langle w, \psi_i \rangle \leq 1\}$$

is non-empty (at least $w = 0$ is in this set) and bounded. Since $\|w\| - \frac{1}{\tau}\langle w, v\rangle$ is a continuous function, the minimum exists.

If $v = 0$, it is obvious that (P) has a unique solution $w = 0$. In the following we suppose that $v \neq 0$.

1. When $\|v\| < \tau$, for any $w \in \mathbb{R}^{N^2}$, we have

   $$\|w\| - \frac{1}{\tau}\langle w, v\rangle \geq \|w\|(1 - \frac{\|v\|}{\tau}) \geq 0.$$

   Thus the minimum of (P) is 0 and the only choice to attain this minimum is $w = 0$.

2. When $\|v\| = \tau$, we still have:

   $$\|w\| - \frac{1}{\tau}\langle w, v\rangle \geq \|w\|(1 - \frac{\|v\|}{\tau}) = 0.$$

   But this time, the equality holds if and only if:

   $$\langle w, v\rangle = \|w\|\|v\|.$$

   This tells us that $w = \gamma v$ where $\gamma \geq 0$. As γv should belong to the feasible set, we need:

   $$\langle \gamma v, \psi_i \rangle \leq 1, \forall i \in I.$$

   Using Lemma 8, we obtain,

   $$\gamma \leq \frac{1}{\max_{i \in I}\langle v, \psi_i \rangle}.$$

   Therefore,

   $$\gamma_0 \triangleq \frac{1}{\max_{i \in I}\langle v, \psi_i \rangle} > 0, \tag{5.5}$$

   is such that all the solutions to (P) are the segment $[0, \gamma_0 v]$.

3. Now let us consider the case: $\|v\| > \tau$.

   Our first assertion is that in this case, the minimum of (P) is strictly less than 0. In fact, consider $w = \gamma_0 v$ where γ_0 is defined in (5.5), then this point is in the feasible set of (P) and

   $$\|w\| - \frac{1}{\tau}\langle w, v\rangle = \gamma_0\|v\|(1 - \frac{\|v\|}{\tau}) < 0.$$

   Now suppose that there are two solutions $w_1, w_2 \in \mathbb{R}^{N^2}$ to (P). Since $\frac{1}{2}(w_1 + w_2)$ is also in the feasible set, we have:

   $$\begin{aligned} \frac{1}{2}\|w_1 + w_2\| - \frac{1}{2\tau}\langle w_1 + w_2, v\rangle &\geq \|w_1\| - \frac{1}{\tau}\langle w_1, v\rangle \\ &= \|w_2\| - \frac{1}{\tau}\langle w_2, v\rangle \\ &= \frac{1}{2}(\|w_1\| + \|w_2\|) - \frac{1}{2\tau}\langle w_1 + w_2, v\rangle. \end{aligned}$$

This lead to that:

$$\|w_1 + w_2\| \geq \|w_1\| + \|w_2\|.$$

Hence

$$\|w_1 + w_2\| = \|w_1\| + \|w_2\|,$$

and

$$w_1 = \beta w_2$$

where $\beta > 0$ (remember that w_1, w_2 are not zero).

Then we know that:

$$\|w_1\| - \frac{1}{\tau}\langle w_1, v\rangle = \beta(\|w_2\| - \frac{1}{\tau}\langle w_2, v\rangle).$$

But

$$\|w_1\| - \frac{1}{\tau}\langle w_1, v\rangle = \|w_2\| - \frac{1}{\tau}\langle w_2, v\rangle < 0,$$

thus we must have $\beta = 1$ and then $w_1 = w_2$. This finishes the proof of the uniqueness of the solution to (P), when $\|v\| > \tau$.

□

5.2.2 Dual formulation

The Lagrangian of the problem (P) is

$$L(w, (\lambda_i)_{i\in I}) = \|w\| - \frac{1}{\tau}\langle w, v\rangle + \sum_{i\in I} \lambda_i(\langle w, \psi_i\rangle - 1).$$

As usual (see Th. 28.3, pp 281, in [49]), the unique solution w^* to (P) is also the first argument of any saddle point $(w^*, (\lambda_i^*)_{i\in I})$ of the form

$$\min_{w\in\mathbb{R}^{N^2}} \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} L(w, (\lambda_i)_{i\in I}).$$

All along the chapter, we denote

$$\mathcal{S} = \{(\lambda_i^*)_{i\in I} \in \mathbb{R}^{+I}, (\lambda_i^*)_{i\in I} = \arg \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} L(w^*, (\lambda_i)_{i\in I})\}. \tag{5.6}$$

We know that $\mathcal{S} \neq \emptyset$ (see Cor. 28.2.1, pp. 278, in [49]) but cannot guarantee it is reduced to a single element.

Notice that, L is a saddle function (i.e. : convex in w and concave in $(\lambda_i)_{i\in I}$) which satisfies the hypotheses of Th. 37.6, pp. 397, in [49] ($L(., (\lambda_i)_{i\in I})$ and $-L(w, .)$ do not have any direction of recession). So, for any $(\lambda_i^*)_{i\in I} \in \mathcal{S}$, $(w^*, (\lambda_i^*)_{i\in I})$ is a saddle point of the form

$$\begin{aligned}
\min_{w\in\mathbb{R}^{N^2}} \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} L(w, (\lambda_i)_{i\in I}) &= \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} \min_{w\in\mathbb{R}^{N^2}} L(w, (\lambda_i)_{i\in I}) \\
&= \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} \min_{w\in\mathbb{R}^{N^2}} \left(\|w\| - \langle w, \frac{1}{\tau} v - \sum_{i\in I} \lambda_i \psi_i \rangle \right) - \sum_{i\in I} \lambda_i.
\end{aligned}$$

Finally, notice that, denoting $F(w) = \|w\|$, we have

$$\min_{w\in\mathbb{R}^{N^2}} \left(\|w\| - \langle w, \frac{1}{\tau} v - \sum_{i\in I} \lambda_i \psi_i \rangle \right) = \begin{cases} -\infty & \text{, if } v - \sum_{i\in I} \tau\lambda_i\psi_i \notin \tau\partial F(0) \\ 0 & \text{, otherwise.} \end{cases} \tag{5.7}$$

Also, we know that

$$\partial F(0) = \{w \in \mathbb{R}^{N^2}, \|w\| \leq 1\}. \tag{5.8}$$

So we finally know that any $(\lambda_i^*)_{i\in I} \in \mathcal{S}$ is solution to

$$\begin{cases} \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} & -\sum_{i\in I} \lambda_i \\ \text{under the constraint } & \|v - \sum_{i\in I} \tau\lambda_i\psi_i\| \leq \tau, \end{cases}$$

which, modulo a trivial multiplication by τ is precisely the problem (D) considered in the preceding section.

As a conclusion, the problem (D) can be solved by any algorithm solving (P) which also provides a Kuhn-Tucker vector $(\lambda_i^*)_{i\in I}$. The point is that, in fact, most algorithms solving (P) also provide such a $(\lambda_i^*)_{i\in I}$. We summarize this as:

Proposition 10 *Suppose that $\|v\| > \tau$ and the assertion of Proposition 6 holds. Denote w^* the unique solution to (P) and suppose that $(\lambda_i^*)_{i\in I}$ is any Kuhn-Tucker vector of (P). Then $(\tau\lambda_i^*)_{i\in I}$ is a solution to (D).*

In the following, we will only consider a small family of such algorithms. (Our motivation for considering this family will be clear after Section 5.2.4 and 5.2.5) This family is described in the next section.

5.2.3 Applying the Proximal Point Algorithm to (P)

We write

$$f\left((\lambda_i)_{i\in I}\right) = \min_{w\in\mathbb{R}^{N^2}} L(w, (\lambda_i)_{i\in I}).$$

As indicated in the previous section, (D) consists in maximizing f over $\mathbb{R}^{+I}$. Assuming that we know how to evaluate ∇f at any location $(\lambda_i)_{i\in I}$ such that $f\left((\lambda_i)_{i\in I}\right)$ is finite, we could in principle apply any gradient based algorithm to achieve that goal. A typical example is the Uzawa algorithm.

Now, at each iteration, the step size of such an algorithm will have to be such that f remains finite (see (5.7)). This will result in a slow and unstable algorithm.

In order to avoid this problem, we propose to stabilize (P) with a Proximal Point Algorithm (see [52, 53]).

We consider

$$g(w) = \|w\| - \frac{1}{\tau}\langle w, v\rangle + \mathbb{1}_{|C}(w), \tag{5.9}$$

with $C = \{w \in \mathbb{R}^{N^2}, \forall i \in I, \langle w, \psi_i\rangle \leq 1\}$ and $\mathbb{1}_{|C}(w)$ equal to 0, if $w \in C$, and to infinity otherwise. Minimizing g is equivalent to solving (P).

Using the Proximal Point Algorithm regularization, we consider, for any $u \in \mathbb{R}^{N^2}$,

$$g_{u,\alpha}(w) = \alpha\|w - u\|^2 + g(w),$$

and the algorithm

$$u^{m+1} = \text{argmin}_{w\in\mathbb{R}^{N^2}}\, g_{u^m,\alpha_m}(w), \tag{5.10}$$

for a given u^0 and a fixed nondecreasing sequence $(\alpha_m)_{m\in\mathbb{N}}$, with $\alpha_0 > 0$.

General results on Proximal Point Algorithm will guarantee that $(u^m)_{m\in\mathbb{N}}$ converges rapidly to the solution w^* to (P). (A more precise statement is given in Proposition 11).

In our dissertation, we need to go one step further and prove that this implies the convergence of the corresponding Kuhn-Tucker vectors. To do so, we exhibit those vectors and write

$$f_{u^m,\alpha_m}\left((\lambda_i)_{i\in I}\right) = \min_{w\in\mathbb{R}^{N^2}} L'(w,(\lambda_i)_{i\in I},u^m,\alpha_m), \tag{5.11}$$

with

$$L'(w,(\lambda_i)_{i\in I},u^m,\alpha_m) = \alpha_m\|w-u^m\|^2 + \|w\| - \langle w, \frac{1}{\tau}v - \sum_{i\in I}\lambda_i\psi_i\rangle - \sum_{i\in I}\lambda_i. \tag{5.12}$$

The family of algorithms which we consider in this chapter is described in Table 5.1. A discussion similar to the one of the preceding section guarantees that the sequence $(u^m)_{m\in\mathbb{N}}$ built by such an algorithm equals the one built with (5.10).

- Initialize u^0
- Repeat until convergence (loop in m)
  1. Use a gradient based algorithm for solving
     $$(\lambda_i^m)_{i\in I} \in \text{argmax}_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} f_{u^m,\alpha_m}\left((\lambda_i)_{i\in I}\right)$$
  2. Update $u^{m+1} = \text{argmin}_{w\in\mathbb{R}^{N^2}} L'(w,(\lambda_i^m)_{i\in I},u^m,\alpha_m)$.

Table 5.1: General form of the algorithms. The gradient based algorithm still needs to be specified.

Notice that, beside the decompositions and recompositions, the only difficulties in the implementation of the above algorithm are the computations of the gradient $\nabla f_{u^m,\alpha_m}$, in step 1, the resolution of the step 2 and, depending on the gradient based algorithm in step 1, the evaluation of $f_{u^m,\alpha_m}\left((\lambda_i)_{i\in I}\right)$.

Our interest for the algorithms above comes from the fact that, as will be shown in the next section, those computations can be performed exactly. Essentially, the cost of the evaluation of $\nabla f_{u^m,\alpha_m}$ is one decomposition and one recomposition in $(\psi_i)_{i\in I}$; the cost for computing $\text{argmin}_{w\in\mathbb{R}^{N^2}} L'(w,(\lambda_i^m)_{i\in I},u^m,\alpha_m)$ and for evaluating $f_{u^m,\alpha_m}\left((\lambda_i)_{i\in I}\right)$ is one recomposition in $(\psi_i)_{i\in I}$.

Moreover, we will show that $\nabla f_{u^m,\alpha_m}$ is Lipschitz and we will provide an upper bound of its Lipschitz constant (this bound can be computed numerically). This will guarantee the convergence of many gradient based algorithms considered in step 1.

Before, going into those details, let us first state the following proposition which guarantees that our "predual-primal" proximal approach actually provides an approximation of actual solutions to (P) and (D). It also guarantees that the loop in m of Table 5.1 converges rapidly and is short. Its proof is given in Appendix.

Proposition 11 *Assume $(\alpha_m)_{m\in\mathbb{N}}$ is a nondecreasing sequence with $\alpha_0 > 0$ and $\|v\| > \tau$ (if $\|v\| \le \tau$, 0 is a trivial solution to (P) and (D)), there exists $a > 0$ and $M > 0$ such that the sequences $(u^m)_{m\in\mathbb{N}}$ and $((\lambda_i^m)_{i\in I})_{m\in\mathbb{N}}$ defined in Table 5.1 satisfy*

1. $(u^m)_{m\in\mathbb{N}}$ *converges to the solution* w^* *of* (P). *Moreover, for* $c_m = \frac{a}{\sqrt{a^2+\alpha_m^2}} < 1$

$$\|u^{m+1} - w^*\| \leq c_m \|u^m - w^*\|, \forall m \geq M. \tag{5.13}$$

2. *For* $C_m = 2\alpha_m(c_m + 1) + 2\frac{c_m}{\|w^*\|}$ *and any* $(\lambda_i^*)_{i\in I} \in \mathcal{S}$, *where* $\mathcal{S}$ *is the optimal set of* (D),

$$\left\| \sum_{i\in I} \lambda_i^m \psi_i - \sum_{i\in I} \lambda_i^* \psi_i \right\| \leq C_m \|u^m - w^*\|, \forall m \geq M.$$

Using (5.13), *the right term converges to* 0.

3. $\lim_{m\to+\infty} d((\lambda_i^m)_{i\in I}, \mathcal{S}) = 0$ *where,*

$$d((\lambda_i^m)_{i\in I}, \mathcal{S}) = \min_{(\lambda_i^*)_{i\in I}\in\mathcal{S}} \|(\lambda_i^m - \lambda_i^*)_{i\in I}\|.$$

5.2.4 Exact resolution of step 2 and exact computation of $\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I})$ and $f_{u^m,\alpha_m}((\lambda_i)_{i\in I})$

First, as is usual with the gradient of functions defined as a minimum, many terms cancels out[1] and we will finally calculate $\nabla f_{u^m,\alpha_m}$:

Proposition 12 *The gradient of* f_{u^m,α_m} *can be calculated as:*

$$\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I}) = (\langle w^*, \psi_i\rangle - 1)_{i\in I},$$

where

$$w^* = \arg\min_{w\in\mathbb{R}^{N^2}} \alpha_m \|w - u^m\|^2 + \|w\| - \langle w, \frac{1}{\tau} v - \sum_{i\in I} \lambda_i \psi_i\rangle. \tag{5.14}$$

Proof. see Appendix. □

As a consequence, modulo a decomposition in $(\psi_i)_{i\in I}$, the computation of $\nabla f_{u^m,\alpha_m}$ and the resolution of step 2 boils down to the same problem : the resolution of (5.14).

Proposition 13 *Denote* $r = \sum_{i\in I} \lambda_i \psi_i - \frac{1}{\tau} v$, $u = u^m$, $\alpha = \alpha_m$. *Then the solution to* (5.14) w^* *is given by:*

$$w^* = \begin{cases} 0 & , \text{ if } \|2\alpha u - r\| \leq 1 \\ \frac{\|2\alpha u - r\| - 1}{2\alpha \|2\alpha u - r\|}(2\alpha u - r) & , \text{ otherwise.} \end{cases} \tag{5.15}$$

Proof. Using the notations of r, u, we need only consider the problem

$$w^* = \arg\min_{w\in\mathbb{R}^{N^2}} \alpha \|w - u\|^2 + \|w\| + \langle w, r\rangle,$$

where u and r are in $\mathbb{R}^{N^2}$.

[1] Notice that the differentiation is not that trivial since, in L', the optimal w depends on $(\lambda_i)_{i\in I}$. However, as is common with such max min problems, the term $\frac{\partial L'}{\partial w}$ equals zero and it cancels the terms $\frac{\partial w}{\partial \lambda_i}$ which appear in the calculation of $\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I})$. For an example of such a calculation, see the proof of Th. 9.3.3, in [54]

Let us begin with the situation where $\|w^*\| = 0$. Differentiating, we know that $2\alpha(w^* - u) + r \in \partial F(0)$, where $F(w) = \|w\|$. Using (5.8), we have

$$w^* = 0 \Rightarrow \|r - 2\alpha u\| \leq 1.$$

On the other hand, if we assume that $\|w^*\| \neq 0$, we know that

$$2\alpha(w^* - u) + \frac{w^*}{\|w^*\|} + r = 0.$$

This gives

$$\|w^*\|(2\alpha u - r) = (2\alpha\|w^*\| + 1)w^*. \tag{5.16}$$

Taking the norm of the above equality, we obtain

$$\|2\alpha u - r\| = 2\alpha\|w^*\| + 1, \tag{5.17}$$

which guaranties that $\|2\alpha u - r\| > 1$.

As a conclusion,

$$w^* = 0 \Leftrightarrow \|2\alpha u - r\| \leq 1,$$

and when $w^* \neq 0$, w^* can be computed, using (5.16) and (5.17), and is

$$w^* = \frac{\|2\alpha u - r\| - 1}{2\alpha\|2\alpha u - r\|}(2\alpha u - r)$$

We can rephrase this as

$$w^* = \begin{cases} 0 & \text{, if } \|2\alpha u - r\| \leq 1 \\ \frac{\|2\alpha u - r\| - 1}{2\alpha\|2\alpha u - r\|}(2\alpha u - r) & \text{, otherwise.} \end{cases}$$

□

As a conclusion, in the Step 1 of the algorithm described in Table 5.1, the gradient can be computed with :

$$\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I}) = (\langle w^*, \psi_i\rangle - 1)_{i\in I}\,, \tag{5.18}$$

where

$$w^* = \begin{cases} 0 & \text{, if } \|t\| \leq 1 \\ \frac{\|t\|-1}{2\alpha_m\|t\|}\ t & \text{, otherwise,} \end{cases} \tag{5.19}$$

with

$$t = 2\alpha_m u^m + \frac{v}{\tau} - \sum_{i\in I}\lambda_i\psi_i.$$

Moreover, the step 2 of the algorithm of Table 5.1 is solved by applying (5.19) at (λ_i^m).

5.2.5 Computing the Lipschitz constant of the energy gradient

Known results on the Moreau envelope permits to prove the following result. Notice that, at the expense of a longer proof, we could, using (5.18), obtain a similar bound.

Proposition 14 *For any* $(\lambda_i)_{i\in I}$ *and* $(\lambda'_i)_{i\in I}$ *in* $\mathbb{R}^{+I}$,

$$\|\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I}) - \nabla f_{u^m,\alpha_m}((\lambda'_i)_{i\in I})\| \leq C\|(\lambda_i - \lambda'_i)_{i\in I}\|,$$

with $C = \frac{\sqrt{M_1 M_2}}{\alpha_m}$, *with*

$$M_1 = \sum_{i\in I} \|\psi_i\|^2,$$

and M_2 *is the norm of the reconstruction operator :*

$$M_2 = \sup_{(\lambda_i)_{i\in I}\neq 0} \frac{\|\sum_{i\in I}\lambda_i\psi_i\|}{\|(\lambda_i)_{i\in I}\|}.$$

Proof. In order to obtain this result, we first rewrite

$$\begin{aligned} L'(w,(\lambda_i)_{i\in I},u^m,\alpha_m) &= \alpha_m\|w-u^m\|^2 + \|w\| - \langle w, \frac{v}{\tau} - \sum_{i\in I}\lambda_i\psi_i\rangle - \sum_{i\in I}\lambda_i \\ &= \alpha_m \left\| w - u^m - \frac{1}{2\alpha_m}\left(\frac{v}{\tau} - \sum_{i\in I}\lambda_i\psi_i\right)\right\|^2 + \|w\| \\ &\quad -\langle u^m, \frac{v}{\tau} - \sum_{i\in I}\lambda_i\psi_i\rangle - \frac{1}{4\alpha_m}\|\frac{v}{\tau} - \sum_{i\in I}\lambda_i\psi_i\|^2 - \sum_{i\in I}\lambda_i. \end{aligned}$$

So, for any $(\lambda_i)_{i\in I} \in \mathbb{R}^{+I}$,

$$\begin{aligned} f_{u^m,\alpha_m}((\lambda_i)_{i\in I}) &= \min_{w\in\mathbb{R}^{N^2}} L'(w,(\lambda_i)_{i\in I},u^m,\alpha_m) \\ &= e_{\alpha_m}\left(u^m + \frac{1}{2\alpha_m}\left(\frac{v}{\tau} - \sum_{i\in I}\lambda_i\psi_i\right)\right) \\ &\quad -\langle u^m, \frac{v}{\tau} - \sum_{i\in I}\lambda_i\psi_i\rangle - \frac{1}{4\alpha_m}\|\frac{v}{\tau} - \sum_{i\in I}\lambda_i\psi_i\|^2 - \sum_{i\in I}\lambda_i \end{aligned}$$

where, for any $t \in \mathbb{R}^{N^2}$,

$$e_{\alpha_m}(t) = \min_{w\in\mathbb{R}^{N^2}} \alpha_m\|w-t\|^2 + \|w\|$$

stands for the Moreau envelope of $\|.\|$.

As a consequence, for any $(\lambda_j)_{j\in I} \in \mathbb{R}^{+I}$,

$$\begin{aligned} \nabla f_{u^m,\alpha_m}((\lambda_j)_{j\in I}) = \Bigg(&-\frac{1}{2\alpha_m}\left\langle \nabla e_{\alpha_m}(u^m + \frac{1}{2\alpha_m}(\frac{v}{\tau} - \sum_{j\in I}\lambda_j\psi_j)), \psi_i\right\rangle \\ &+ \langle u^m, \psi_i\rangle + \frac{1}{2\alpha_m}\langle \frac{v}{\tau} - \sum_{j\in I}\lambda_j\psi_j, \psi_i\rangle - 1\Bigg)_{i\in I}. \end{aligned}$$

Moreover, as is common for the Moreau envelope (see the introduction of [55]),

$$\|\nabla e_{\alpha_m}(t) - \nabla e_{\alpha_m}(t')\| \leq 2\alpha_m\|t-t'\|.$$

So, for any $(\lambda_i)_{i\in I} \in \mathbb{R}^I$ and $(\lambda'_i)_{i\in I} \in \mathbb{R}^I$,

$$\begin{aligned}
&\|\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I}) - \nabla f_{u^m,\alpha_m}((\lambda'_i)_{i\in I})\|^2 \\
\leq\ & \sum_{i\in I}\left(\|\sum_{j\in I}\frac{\lambda'_j-\lambda_j}{2\alpha_m}\psi_j\|\|\psi_i\| + \frac{1}{2\alpha_m}\|\sum_{j\in I}(\lambda_j-\lambda'_j)\psi_j\|\|\psi_i\|\right)^2 \\
\leq\ & \frac{1}{\alpha_m^2}\|\sum_{j\in I}(\lambda_j-\lambda'_j)\psi_j\|^2\sum_{i\in I}\|\psi_i\|^2 \\
\leq\ & \frac{M_1^2M_2^2}{\alpha_m^2}\|(\lambda_j-\lambda'_j)_{j\in I}\|^2,
\end{aligned}$$

where M_1 and M_2 are given in the proposition. □

The above proposition is important since it guarantees that some gradient based algorithm with constant step size, used to solve the first step of Table 5.1, converges for some step size (see next sections). Together with Proposition 11, this ensures that the whole algorithm converges to the desired solution.

However, in order to chose the step size in these algorithms we need to have an estimate of the best possible constant Lipschitz constant. This can, of course be done experimentally be running the algorithm for several step-size, when all the other parameters are fixed.

A more flexible way to chose the step size is to use the formula expressing the bound C given in Proposition 14. With this regards, for most dictionaries, all its elements but M_2 are easy to calculate.

In order to estimate M_2, we use the upper bound (it is easy to obtain)

$$M_2 \leq \sqrt{\sum_{i\in I}\|\psi_i\|^2}. \tag{5.20}$$

Finally, as can easily be seen from (5.18) and (5.19), f_{u^m,α_m} does not satisfy any sort of ellipticity property. In particular, it is not elliptic. This rules out the guarantee of some well known properties (see [54]).

5.2.6 Uzawa version of the algorithm

In this section, we present the algorithm obtained when the gradient based algorithm used to solve the step 1 of the algorithm described in Table 5.1 is a simple projected gradient ascent with constant time step. The step 1 is then an Uzawa algorithm solving the dual of (P_{u^m}) (thus the name of the version). Given Proposition 14, we know (see [56], Cor. 2.1.2, pp. 70, and Th. 2.2.8, pp. 88) that it converges as soon as the time step is in the range $(0, \frac{2}{C})$, where C is given in Proposition 14. Moreover, the "best time step" is $\rho = \frac{1}{C}$.

We also know (see [56]) that, for $\rho = \frac{1}{C}$ and $u^m \in \mathbb{R}^{N^2}$, there exists a constant $C_1 > 0$ (which depends on the quality of the initialization) such that

$$f^*_{u^m,\alpha_m} - f_{u^m,\alpha_m}\left((\lambda_i^k)_{i\in I}\right) \leq C_1\frac{2C}{k+4},$$

where $(\lambda_i^k)_{i\in I}$ is the result at the k^{th} iteration of the algorithm and

$$f^*_{u^m,\alpha_m} = \max_{(\lambda_i)_{i\in I}\in\mathbb{R}^{+I}} f_{u^m,\alpha_m}\left((\lambda_i)_{i\in I}\right) \tag{5.21}$$

- Inputs : $\tau > 0$, the initial image $v \in \mathbb{R}^{N^2}$, a dictionary $(\psi_i)_{i\in i}$ and $(\alpha_m)_{m\in\mathbb{N}}$
- Output : the coordinates $(\lambda_i)_{i\in I}$
- The algorithm :
  - Initialize $(\lambda_i^0)_{i\in I}$, $u^0 \in \mathbb{R}^{N^2}$ and $\rho = \frac{1}{C}$.
  - Repeat until convergence (loop in m)
    - Repeat until convergence (loop in k)
      1. Compute $w^k = 2\alpha_m u^m - \sum_{i\in I} \lambda_i^k \psi_i + \frac{1}{\tau} v$
      2. if $(\|w^k\| \leq 1)$, set $w^k = 0$
         otherwise, set $w^k \leftarrow a\, w^k$, with $a = \frac{\|w^k\|-1}{2\alpha_m \|w^k\|}$
      3. Update λ^{k+1},

         $$\forall i \in I, \lambda_i^{k+1} = \max\left(0, \lambda_i^k + \rho(\langle w^k, \psi_i\rangle - 1)\right)$$

    - update $u^{m+1} = w^k$ and, for all $i \in I$, $\lambda_i^0 = \lambda_i^{k+1}$.
  - Compute $i \in I$, $\lambda_i = \tau\lambda_i^0$.

Table 5.2: Uzawa version of the algorithm : The step 1 of the algorithm described in Table 5.1 is solved by a projected gradient descent with constant step size.

The final algorithm is described in Table 5.2.

The details on the initialization are given in Section 5.2.7. The constant C was estimated using Proposition 14 and (5.20). This gives :

$$C = \frac{\left(\sum_{i\in I} \|\psi_i\|^2\right)^{\frac{3}{2}}}{\alpha_m}. \tag{5.22}$$

5.2.7 Details and variants of the algorithm

This section contains some details on the use of the above algorithm when solving the usual Basis Pursuit Denoising model (instead of the Non-Negative Basis Pursuit Denoising), the initialization and the stopping criterion of the algorithm.

Also, there exists many gradient based algorithms for solving the step 1 in Table 5.1. In addition to the projected gradient algorithm with constant step described in the above section, we have implemented two other versions. Those versions are described in this section.

Symmetric and partly symmetric dictionaries

The algorithm presented so far solves a Non Negative Basis Pursuit Denoising model. We would like to emphasize that when the dictionary is symmetric (i.e. $\exists J \subset I$, such that $(\psi_i)_{i\in I} = (\psi_j)_{j\in J} \cup (-\psi_j)_{j\in J}$) or partly symmetric (i.e. $\exists J$ and $J' \subset I$, such that $(\psi_i)_{i\in I} = (\psi_j)_{j\in J'} \cup (\psi_j)_{j\in J} \cup (-\psi_j)_{j\in J}$), this generalization is not made at any expense.

For simplicity, let us consider a symmetric dictionary $(\psi_i)_{i\in I} = (\psi_j)_{j\in J} \cup (-\psi_j)_{j\in J}$. When applied to coordinates $(\lambda_j^+)_{j\in J} \cup (\lambda_j^-)_{j\in J} \in \mathbb{R}^{2J}$, the reconstruction operator in

$(\psi_i)_{i\in I}$ is

$$\sum_{i\in J}(\lambda_j^+ - \lambda_j^-)\psi_j.$$

This is just the reconstruction in $(\psi_j)_{j\in J}$, applied to $(\lambda_j^+ - \lambda_j^-)_{j\in J}$.

Similarly, the decomposition of any $w \in \mathbb{R}^{N^2}$, in $(\psi_i)_{i\in I}$, is

$$(\langle w, \psi_j\rangle)_{j\in J} \cup (-\langle w, \psi_j\rangle)_{j\in J},$$

and only requires to decompose $w \in \mathbb{R}^{N^2}$ in $(\psi_j)_{j\in J}$.

As a conclusion, the cost for applying a decomposition or a recomposition operator in $(\psi_j)_{j\in J} \cup (-\psi_j)_{j\in J}$ is essentially the same as the cost for applying the corresponding operators in $(\psi_j)_{j\in J}$.

A more serious issue is that the algorithm might converge more slowly, because it needs time to set a coordinate (for instance) λ_i^- to 0 although $\lambda_i^+ > 0$. In order to assess the extent of this problem, we evaluated

$$R = 100\frac{\#\{j \in J, \lambda_j^+ > 0 \text{ and } \lambda_j^- > 0\}}{\#J}$$

for a symmetric dictionary, along the iterative process (# denotes the cardinal of a set). The order of magnitude of the worse value we found was $R \approx 0.1$ and it always rapidly decays to 0. This suggests that it is not a practical problem.

However, when this occurs, we also observed that adding the "projection"

$$\forall j \in J, (\lambda_j^+, \lambda_j^-) \leftarrow \begin{cases} (\lambda_j^+ - \lambda_j^-, 0) & \text{, if } \lambda_j^+ \geq \lambda_j^- \\ (0, \lambda_j^- - \lambda_j^+) & \text{, otherwise,} \end{cases}$$

as a fourth step, in the algorithm of Table 5.2, slightly improves the convergence. Notice that this "projection" obviously increases f_{u^m,α_m} (the objective function which is maximized). We have no theoretical proof of convergence with this "projection", but we do neither anticipate, nor have experimentally observed, any convergence problem when using this "projection".

Although it does not seem to be a mandatory step, all the experiments conducted in Section 5.3 use this "projection".

The initialization

In the algorithm of Table 5.2, we need to initialize $(\lambda_i^0)_{i\in I}$ and $u^0 \in \mathbb{R}^{N^2}$.

We have not studied the initialization of $(\lambda_i^0)_{i\in I}$. There are indeed many possibilities for this initialization and we leave this study for a future work. We therefore simply use

$$\lambda_i^0 = 0, \text{ for all } i \in I.$$

Concerning the initialization of u^0, we tried two possibilities:

- the simplest : $u^0 = 0$
- the most efficient : First observe that $(u^m)_{m\in\mathbb{N}}$ converges to the solution w^* to (P). Therefore, u^0 should be close to w^*. Let us approximate w^*, given an estimate $(\lambda_i^0)_{i\in I}$ of a solution to (D).

If $(\lambda_i^0)_{i\in I}$ is properly initialized, we know that

$$\frac{w^*}{\|w^*\|} - \frac{v}{\tau} + \sum_{i\in I} \lambda_i^0 \psi_i \approx 0.$$

Moreover, since w^* solves (P) and $\|v\| > \tau$ we have

$$\max_{i\in I} \langle w^*, \psi_i \rangle = 1.$$

So, we have

$$w^* \approx \frac{1}{\max_{i\in I} \langle w', \psi_i \rangle} w', \tag{5.23}$$

with

$$w' = \frac{v}{\tau} - \sum_{i\in I} \lambda_i^0 \psi_i.$$

All these considerations leads to the idea of initializing u^0 at the approximate value of w^* given by (5.23).

Experimentally, the latter initialization procedure consistently provides a better initialization than $u^0 = 0$, even when $(\lambda_i^0)_{i\in I}$ is far from the actual solution to (D). Of course, the advantage of this initialization is more striking when $(\lambda_i^0)_{i\in I}$ is close to the actual solution to (D).

In all the experiments which are presented in Section 5.3, we use the initialization defined by (5.23).

Stopping criteria

Although it is a source of improvements of the algorithm, this an aspect we have not really studied. The stopping criteria used in the experiments are :

- for the loop in k : The loop continues while :

  $$\| \sum_{i\in I} (\lambda_i^k - \lambda_i^{k-1}) \psi_i \| > 0.01 \text{ and } k < 50.$$

  In practice, during the first iterations of the loop in m, the used stopping criterion is $k \geq 50$. After that $\| \sum_{i\in I} (\lambda_i^k - \lambda_i^{k-1}) \psi_i \| \leq 0.01$ is used and the number of iteration in k rapidly equals 1.

  Notice with this regard that a better stopping criterion could be deduced from conditions B or B', in [52], pp. 880. It would indeed provide better theoretical guarantees of convergence.

- for the loop in m : In order to study the ability of the algorithm to converge, we simply use the stopping criterion : continue the loop in m while

  $$\text{the number of decomposition/recomposition} \leq 3000.$$

  A better stopping criterion should be used if one wants to avoid useless iterations.

  Notice that the transition between m and $m+1$ is sometimes visible on the curves of Section 5.3. The "singularities" of those curves correspond indeed to such a transition.

Armijo Rule Along the Projection Arc

We also implemented a version of the algorithm where the gradient based algorithm used to solve step 1 of Table 5.1 is the "Armijo Rule Along the Projection Arc" described in [57], Section 2.3.1, pp. 230.

In short, the principle of this algorithm (for maximization) is to define

$$(\lambda_i(\rho))_{i\in I} = \mathbf{sup}\left((\lambda_i^k)_{i\in I} + \rho\nabla f_{u^m,\alpha_m}((\lambda_i^k)_{i\in I}), 0\right),$$

where we denote, for any $(\lambda_i)_{i\in I} \in \mathbb{R}^I$,

$$\mathbf{sup}((\lambda_i)_{i\in I}, 0) = (\sup(\lambda_i, 0))_{i\in I}.$$

The algorithm uses the update

$$(\lambda_i^{k+1})_{i\in I} = (\lambda_i(\rho^k))_{i\in I},$$

where $\rho^k = \beta^m\rho_0$, for $\beta \in (0,1)$, a fixed $\rho_0 > 0$ and for the first nonnegative integer m such that

$$f_{u^m,\alpha_m}((\lambda_i(\beta^m\rho_0))_{i\in I}) - f_{u^m,\alpha_m}((\lambda_i^k)_{i\in I}) \geq \sigma\langle\nabla f_{u^m,\alpha_m}((\lambda_i^k)_{i\in I}), (\lambda_i(\beta^m\rho_0) - \lambda_i^k)_{i\in I}\rangle,$$

for $\sigma \in (0,1)$.

In the context of our problem, the drawback of this algorithm is that each test of a new value m requires one evaluation of $f_{u^m,\alpha_m}((\lambda_i(\beta^m\rho_0))_{i\in I})$. This evaluation is made using (5.11) and requires one recomposition in $(\psi_i)_{i\in I}$. In order to avoid too many evaluations we restricted our search to $m \in \{0,1,2\}$. It avoids useless computations when, close to convergence, the step size tends to be small. We have not proved that such a variation converges. However, we anticipate no difficulty in this regard, as long as $\beta^2\rho_0 < \frac{2}{C}$, where C is the constant in Proposition 14.

In the experiments using this version of the algorithm, we take $\sigma = \frac{1}{4}$, $\rho_0 = \frac{4}{C}$ and $\beta = \frac{1}{2}$. Doing so, we only test the steps $\frac{4}{C}$, $\frac{2}{C}$ and $\frac{1}{C}$. This is the best set of parameters we found.

Nesterov version of the algorithm

We also implemented a version of the algorithm where the gradient based algorithm used to solve step 1 of Table 5.1 is the Nesterov Algorithm described [56], Section 2.2.4, pp.90.

Despite theoretical qualities[2], this algorithm suffers from instabilities during the first iterations which make it slower than the Uzawa version above. Specialists who saw our results was not surprised.

We do not report any further on this implementation.

5.3 Experimental results

In Section 5.3.1, we give all the details on the experimental data and the quantities which will be used to assess the quality of the algorithms.

[2]The convergence of this algorithm is in $\frac{1}{k^2}$ (see [56], Th. 2.2.3, pp. 80). This improves the $\frac{1}{k}$ performance of our projected gradient algorithm (see Section 5.2.6).

We display in Section 5.3.2 some experiments on the convergence of the algorithms presented in this chapter. In particular they emphasize on the influence of $(\alpha_m)_{m\in\mathbb{N}}$ on the convergence speed of the algorithms.

In Section 5.3.3, we describe the existing algorithms solving the Basis Pursuit Denoising model. In particular, we describe in details the algorithm proposed by M. Elad, in [28], and the one proposed by I. Daubechies M. Defrise and C. De Mol, in [22]. We also comment other existing algorithms.

Finally, in Section 5.3.4, we compare the different implementations (our Proximal Point Algorithm, the one of Daubechies-Defrise-De Mol, called IT, and the one of Elad, which is called PCD) of the Basis Pursuit.

5.3.1 Experiments description

All the experiments are made with the same dictionary : a translation invariant discrete local cosine dictionary. It consists in all the translations of the 64 small images displayed on Figure 5.1. The small image at the "frequency location" $(\xi,\eta)\in\{0,\dots,7\}^2$ is

$$\phi^{\xi,\eta}_{m,n}=\frac{1}{\sqrt{C_{\xi,\eta}}}\begin{cases}\cos(\frac{\xi(2m+1)\pi}{16})\cos(\frac{\eta(2n+1)\pi}{16}) & \text{, if } (m,n)\in\{0,\dots,7\}^2,\\ 0 & \text{, if } (m,n)\notin\{0,\dots,7\}^2,\end{cases}$$

with

$$C_{\xi,\eta}=\sum_{m.n=0}^{7}\left(\cos(\frac{\xi(2m+1)\pi}{16})\cos(\frac{\eta(2n+1)\pi}{16})\right)^2.$$

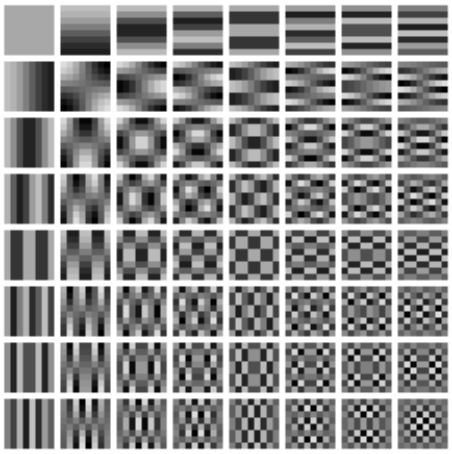

Figure 5.1: Small images defining the translation invariant discrete local cosine dictionary.

The dictionary is also symmetrized and we finally obtain

$$(\psi_i)_{i\in I}=(\psi_j)_{j\in J}\cup(-\psi_j)_{j\in J},$$

where

$$(\psi_j)_{j\in J}=\left\{\tau_{m,n}(\phi^{\xi,\eta}),\text{ for }(\xi,\eta)\in\{0,\dots 7\}^2\text{ and }(m,n)\in\{0,\dots,N-1\}^2\right\},$$

for $\tau_{m,n}$, the translation of an image by the vector (m,n).

Doing so, we obtain a model which can also be solved by any algorithm solving the usual Basis Pursuit Denoising (as opposed to the Non Negative Basis Pursuit Denoising).

This will allow comparisons. Moreover, the dictionary is particularly large and we can expect the resolution of the Basis Pursuit Denoising to be particularly difficult.

The decompositions and recompositions which are needed in the algorithms are computed with Fast Fourier Transforms, as is explained in [42] or Chapter 1.

To assess the quality of a decomposition $(\lambda_i)_{i\in I} = (\lambda_j^+)_{j\in J} \cup (\lambda_j^-)_{j\in J}$ approximating an image $v \in \mathbb{R}^{N^2}$, we consider three quantities :

$$l^0 = \frac{100}{\#J}\#\{j \in J, \lambda_j^+ \neq 0 \text{ or } \lambda_j^- \neq 0\}, \tag{5.24}$$

where, again, # denotes the cardinal of a set.

Similarly, we consider

$$l^1 = \frac{1}{\#J}\sum_{j\in J} |\lambda_j^+ - \lambda_j^-|, \tag{5.25}$$

and

$$l^2 = \left| \| \sum_{i\in I} \lambda_i \psi_i - v\| - \tau \right|, \tag{5.26}$$

where, τ is the parameter in (D) and, for any $u \in \mathbb{R}^{N^2}$,

$$\|u\| = \sqrt{\frac{1}{N^2}\sum_{i,j=0}^{N-1} u_{i,j}^2}.$$

Those are the quantities evaluated by the curves displayed on Figure 5.3, 5.4, 5.5,...

Since most of the computational time is spent in computing the decomposition and recomposition in $(\psi_i)_{i\in I}$, the unit of the x-axis of all the curves presented in the paper corresponds to one decomposition and one recomposition in $(\psi_i)_{i\in I}$. Notice this also corresponds to the computational effort for computing two recompositions. This allows keeping the same x-axis unit for the Armijo version of the algorithm.

In the experiments, we take v equal to an extracted part of the image Barbara (see Figure 5.2).

Figure 5.2: Image extracted from the image Barbara. It is used for the input v in all the experiments.

Finally, the experiments are for

- $\tau = 0.0254$ which corresponds to the l^2 norm of the residual when applying the IT algorithm (see Section 5.3.3), for solving (5.27), with $\lambda = 0.1$ and for v : the image on Figure 5.2. (The l^2 norm of the residual obtained by solving the same problem with the PCD algorithm is larger.)

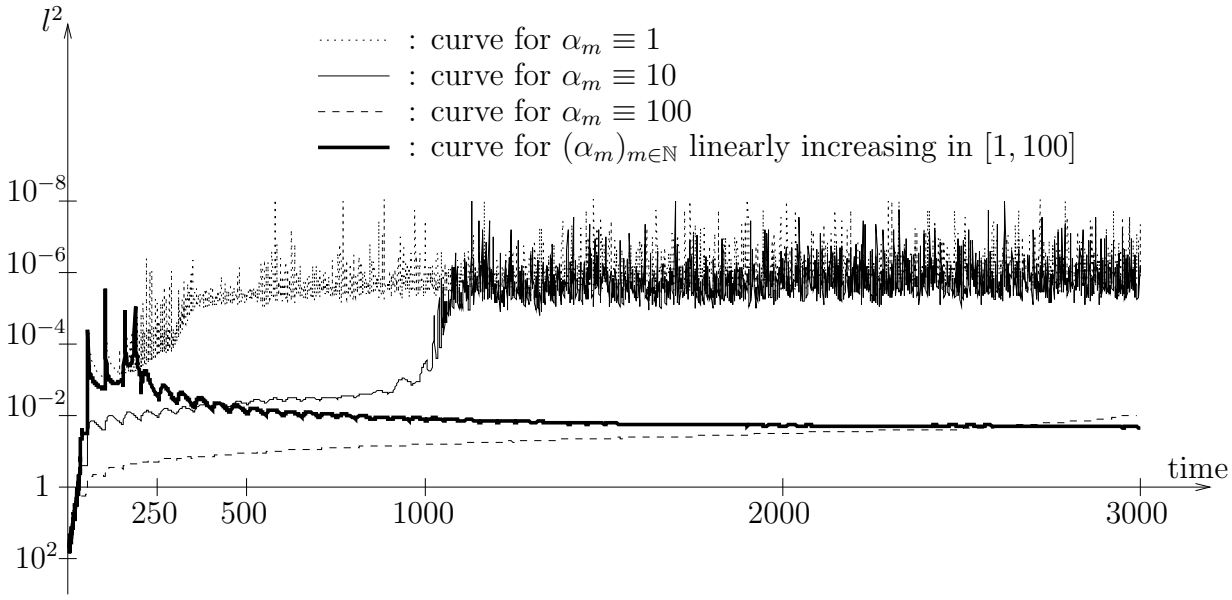

Figure 5.3: l^2 curves for Uzawa algorithm, for $\tau = 0.0254$: The drawn curves give the criterion l^2 (see (5.26)), as a function of the number of decomposition/recomposition, for $\alpha_m \equiv 1$, $\alpha_m \equiv 10$, $\alpha_m \equiv 100$ and $(\alpha_m)_{m\in\mathbb{N}}$ linearly increasing from 1 to 100. The final norms of the residual are respectively 0.0254, 0.0254, 0.0364 and 0.0497.

- $\tau = 15.29$ which corresponds to the l^2 norm of the residual when applying the PCD algorithm (see Section 5.3.3), for solving (5.27), with $\lambda = 200$ and for v : the image on Figure 5.2. (The l^2 norm of the residual obtained by solving the same problem with the IT algorithm is larger.)

Notice that $\tau = 0.0254$ is a very difficult situation since, with such a large dictionary, the choice of the non-zero coordinates is very ambiguous and the decomposition is not extremely sparse. The values $\tau = 15.29$ correspond to much simpler situation.

5.3.2 Practical convergence of the Proximal Point Algorithm and influence of $(\alpha_m)_{m\in\mathbb{N}}$

As can be seen in the preceding sections, beside the parameters of the problem $(\psi_i)_{i\in I}$, τ and v, the only parameter of the algorithm is $(\alpha_m)_{m\in\mathbb{N}}$ (see (5.10)). Our first experiments therefore aim at understanding its role on the convergence properties of the algorithms. We study 4 possibilities : $\alpha_m \equiv 1$, $\alpha_m \equiv 10$, $\alpha_m \equiv 100$ and $(\alpha_m)_{m\in\mathbb{N}}$ linearly varying between 1 and 100.

In fact, it plays the same role in both the Uzawa and the "Armijo" implementation. So, we only display the curves for the Uzawa version of the algorithm. All the curves which we comment and display in this section concern experiments with the image on Figure 5.2, $\tau = 0.0254$ and the dictionary described in Section 5.3.1.

The first issue we would like to address is the convergence of l^2. In theory, it should converge to 0. This is actually the case in our experiments. We display on Figure 5.3 the curves representing l^2 as a function of computational cost, for the 3000 first decomposition/recompositions. Those curves use a logarithmic scale. A precision of 10^{-6} is all we can expect, since this corresponds to the precision of the float numbers used in the implementation of the algorithm.

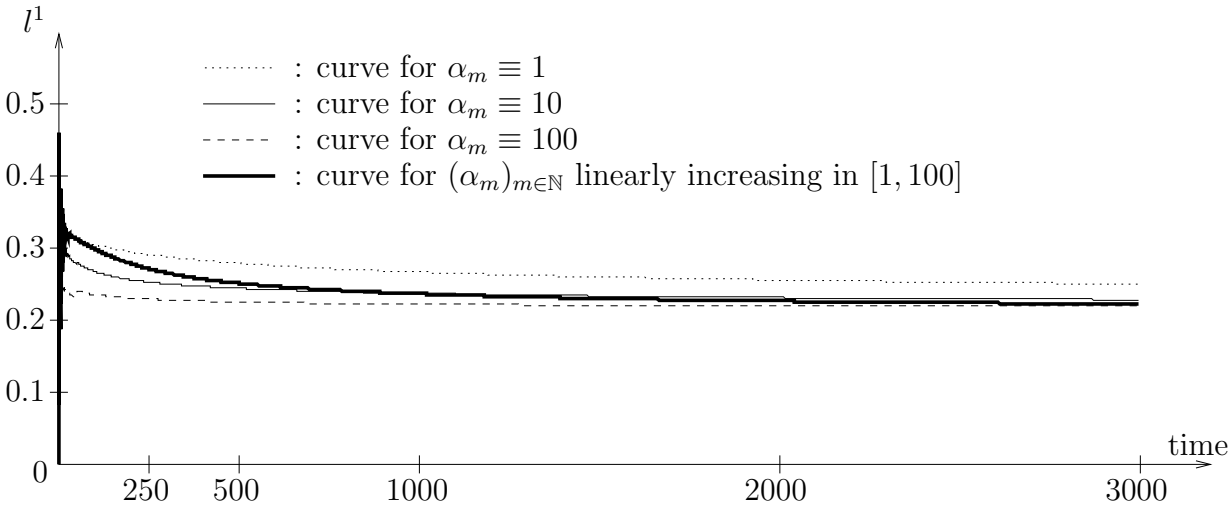

Figure 5.4: l^1 curves for Uzawa algorithm, for $\tau = 0.0254$: The drawn curves give the criterion l^1 (see (5.25)), as a function of the number of decomposition/recomposition, for $\alpha_m \equiv 1$, $\alpha_m \equiv 10$, $\alpha_m \equiv 100$ and $(\alpha_m)_{m\in\mathbb{N}}$ linearly increasing from 1 to 100. The final values of these curves are respectively 0.250, 0.228, 0.219 and 0.222.

We see on Figure 5.3 (and this was confirmed in many other experiments for both the Uzawa and the "Armijo" versions of the algorithm) that, as far as the l^2 criterion is concerned, small values in $(\alpha_m)_{m\in\mathbb{N}}$ are preferable. Also, when compared to a constant α_m, an increasing $(\alpha_m)_{m\in\mathbb{N}}$ does not seem to improve the convergence. The l^2 value for a given value of α_m does not seem to be influenced a lot by the results of the preceding iterations.

We display on Figure 5.4 the curves representing the quantity l^1 as a function of the number of decomposition/recompositions. Again, those curves are representative of many other experiments confirming the same statement : As far as the l^1 criterion is concerned, large values in $(\alpha_m)_{m\in\mathbb{N}}$ are preferable. Notice that adding more iterations does not permit to improve the result as much as a change of $(\alpha_m)_{m\in\mathbb{N}}$. Again, the l^1 value for a given value of $(\alpha_m)_{m\in\mathbb{N}}$ is not influenced a lot by the results of the preceding iterations.

The quantity l^0 is also of a particular interest, since people usually use the Basis Pursuit Denoising model to obtain a result which is sparsely represented in the dictionary $(\psi_i)_{i\in I}$. We display on Figure 5.5 the curves representing the quantity l^0, as a function of the number of decomposition/recomposition. These curves are, of course, very much correlated to those concerning the l^1 criterion. We get the conclusions : As far as the l^0 criterion is concerned, large values in $(\alpha_m)_{m\in\mathbb{N}}$ are preferable. Again, adding more iterations does not permit to improve the result as much as a change in $(\alpha_m)_{m\in\mathbb{N}}$. This change is not influenced a lot by the results of the preceding iterations.

As a conclusion, $(\alpha_m)_{m\in\mathbb{N}}$ is a numerical parameter. A strategy which consists in increasing α_m, with m, does not permit to keep the benefit of the first iterations (where the l^2 criterion was good). However, since a change in α_m is not influenced a lot by results of the preceding iterations, a strategy consisting in

- increasing α_m, if the l^2 norm of the residual is close to τ.
- shrinking α_m, if the l^2 norm of the residual is far from τ.

is a possible way to get rid of this parameter.

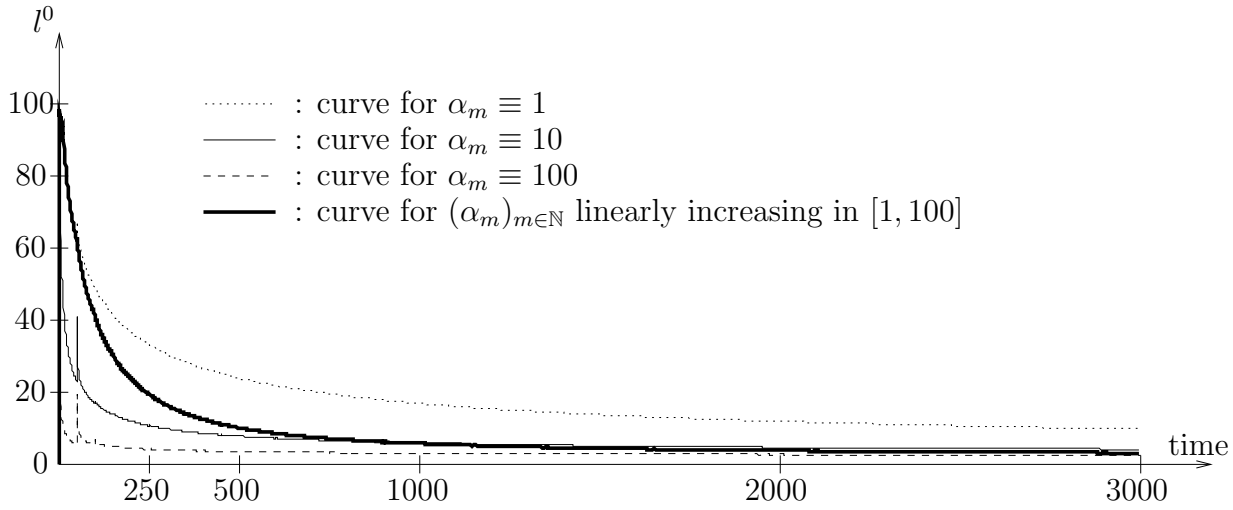

Figure 5.5: l^0 curves for Uzawa algorithm, for $\tau = 0.0254$: The drawn curves give the criterion l^0 (see (5.24)), as a function of the number of decomposition/recomposition, for $\alpha_m \equiv 1$, $\alpha_m \equiv 10$, $\alpha_m \equiv 100$ and $(\alpha_m)_{m\in\mathbb{N}}$ linearly increasing from 1 to 100. The final values of these curves are respectively 9.58, 3.97, 2.38 and 2.94.

5.3.3 Existing algorithms for solving the Basis Pursuit Denoising model

As already mentioned in the introduction, there are surprisingly few algorithms for solving the Basis Pursuit Denoising model. The literature on the subject is currently rapidly growing though. All those we found [22, 24, 26, 25, 27, 28, 29, 30, 31] deal with the model under its form :

$$\min_{(\lambda_i)_{i\in I}\in\mathbb{R}^I} \|\sum_{i\in I}\lambda_i\psi_i - v\|^2 + \lambda\sum_{i\in I}|\lambda_i|. \tag{5.27}$$

Let us denote, for all $(\lambda_i)_{i\in I} \in \mathbb{R}^I$

$$f\left((\lambda_i)_{i\in I}\right) = \|\sum_{i\in I}\lambda_i\psi_i - v\|^2 + \lambda\sum_{i\in I}|\lambda_i|,$$

and, for $\sigma > 0$ and $t \in \mathbb{R}$,

$$S_\sigma(t) = \begin{cases} t - \frac{\sigma}{2} & \text{, if } t \geq \frac{\sigma}{2} \\ 0 & \text{, if } |t| < \frac{\sigma}{2} \\ t + \frac{\sigma}{2} & \text{, if } t \leq -\frac{\sigma}{2}. \end{cases}$$

Parallel Coordinate descent (PCD) Algorithm

Recall that the PCD algorithm proposed in [28] is described in Table 1.1 (see Chapter 1). This is one of the existing algorithm to which we will compare our results.

Iterative Thresholding (IT)

This IT algorithm (described in Table 1.1, see Chapter 1))is proved to converge as soon as the norm of the reconstruction operator is strictly smaller than 1. Taking the notation of Proposition 14, we write

$$M_2 < 1.$$

Of course, it can be applied to any dictionary $(\psi_i)_{i\in I}$, since, for any $\beta > 0$:

$$(\lambda_i^*)_{i\in I} \in \operatorname{argmin}_{(\lambda_i)_{i\in I}\in\mathbb{R}^I} \| \textstyle\sum_{i\in I} \lambda_i\psi_i - v\|^2 + \lambda \sum_{i\in I} |\lambda_i| \tag{5.28}$$

$$\iff (\beta\lambda_i^*)_{i\in I} \in \operatorname{argmin}_{(\lambda_i)_{i\in I}\in\mathbb{R}^I} \| \textstyle\sum_{i\in I} \lambda_i\frac{\psi_i}{\beta} - v\|^2 + \frac{\lambda}{\beta} \sum_{i\in I} |\lambda_i| \tag{5.29}$$

So one can solve (5.29), for β such that $\frac{M_2}{\beta} < 1$, and multiply the obtained solution by $\frac{1}{\beta}$. This provides a solution to (5.28).

In practice, we used the upper bound provided by (5.20) (which can be computed numerically) and chose β such that

$$\frac{\sqrt{\sum_{i\in I} \|\psi_i\|^2}}{\beta} = c,$$

with $c = 0.999$. This value 0.999 might seem arbitrary, since any $c \in (0, 1)$ guarantees convergence. However, we found experimentally that a larger c leads to a better convergence. We do not report any further on the tuning of c.

Our Proximal Point Algorithm under the light of IT

The loop in k of the algorithm is very similar to the soft thresholding of IT. In particular, when applied to a symmetric dictionary, the step 3 is a soft thresholding. However, w^k is not exactly the residual which is found in IT. It indeed contains a "stabilization term" $2\alpha_m u^m$ and its norm is slightly modified.

Notice, by the way, that our Proximal Point Algorithms are not homogeneous in $(\psi_i)_{i\in I}$. So an extra parameter similar to c (or β) (see the above description of IT) might permit to improve the convergence of our Proximal Point Algorithms. We have not tested this possibility.

What remains

Beside a small regularization, the main innovation, in [29], is to replace t^k by a $M + 1$-dimensional vector. Its computation is then performed by minimizing f over

$$\operatorname{Span}\left(d^k \cup ((\lambda_i^{k-m})_{i\in I} - (\lambda_i^{k-1-m})_{i\in I})_{m\in\{1,\dots M\}}\right).$$

Although this obviously improves the convergence results, we have not implemented this algorithm. It seems indeed to provide only a relatively small improvement when compared to the algorithm described in Table 1.1 (see [29]). This improvement is made at the price of an important effort in the implementation of the algorithm. Experimentally, they found in [29] that the best value for M is 1.

In [13], the authors propose an interior point method. (A better description is given in [27].) We have not implemented it, since it is not guaranteed to converge.

The BCR algorithm introduced in [27] only applies when the dictionary $(\psi_i)_{i\in I}$ is a union of orthonormal bases (its extension to a union of orthogonal bases is straightforward). It indeed uses the fact that the soft-thresholding operator provides an exact resolution of (5.27) when the dictionary is an orthonormal basis. Moreover, in order to obtain a reasonable algorithm, a fast transform needs to be available for all the bases contained in $(\psi_i)_{i\in I}$. It is therefore a very specialised algorithm and we have not implemented it.

Finally, the algorithm proposed in [30, 31] is very elegant and has the advantage of being exact. However, it does require, at each iteration, the inversion of a matrix. The size of this matrix goes to the number of non-zero coordinates of the result. This restricts its use to applications where this number remains very small.

5.3.4 Comparison of the algorithms

We display on Figure 5.6, 5.7, 5.8, 5.9, 5.11, 5.12 and 5.13 the curves corresponding to a comparison between the Uzawa and the Armijo versions of our algorithm (see Table 5.2), the PCD algorithm described in Table 1.1 and the IT algorithm described in Section 5.3.3.

Concerning the choice of the parameters, the purpose of our paper is obviously not to answer the question : How to fix λ in the model (5.27)? So our only choice is to follow the steps:

- Run the PCD algorithm and IT algorithm for a given value λ.
- Compute τ : the smallest l^2 norm of the residual amongst those obtained by the PCD and IT algorithm.
- Run Uzawa and Armijo versions of the algorithm for this τ.

This results in an unfair comparison favoring the PCD algorithm or the IT algorithm, depending on which leads to the smallest norm.

We display on Figure 5.6, 5.7 and 5.8 the l^2, l^1 and l^0 criterion as a function of the number of decomposition/recomposition. This experiment is made for $\lambda = 0.1$, in (5.27), which corresponds to $\tau = 0.0254$, when using IT. Notice that for our Proximal Point Algorithm the curves corresponds to $\alpha_m \equiv 10$. A better l^0 and l^1 convergence is achieved with $\alpha_m \equiv 100$ and a better l^2 convergence is achieved with $\alpha_m \equiv 1$ (see the curves of Section 5.3.2).

Concerning the comparison between the Uzawa and the Armijo versions of the algorithm, we find that they have very similar convergence ability. The Uzawa version seems to converge a little faster, but this conclusion might be false on some other experiments or with better parameters for the Armijo version of the algorithm. Given the additional difficulty in the implementation of the Armijo version, we do not recommend it.

Concerning the convergence of the l^2 criterion, PCD and IT are much faster at obtaining a fair approximation. It is not clear whether we would find the same result when λ is tuned in order to reach a given precision level τ. This would clearly depend on the strategy used to achieve this goal. Notice also that PCD and IT are slower at getting a very good convergence to 0. In particular, IT and PCD do not converge to the same error.

The convergence of the l^1 and l^0 criterion are in favor of our Proximal Point Algorithm implementation of the Basis Pursuit Denoising (see Figure 5.7 and 5.8). In particular, almost none of the coordinates are canceled by the PCD implementation of the Basis Pursuit Denoising, our implementation has less than 3.9% non-zero coordinates (after the 3000 iterations). To compare the IT and our Proximal Point Algorithm implementation, remember that the curves on Figure 5.7 and 5.8 needs to be red "horizontally". For instance, in order to obtain 20% of non-zero coordinates, it takes less than a hundred decomposition/recomposition with a Proximal Point Algorithm and around one thousand with IT. Similarly the Proximal Point Algorithm reaches $l^0 = 11.38$ after 215 decomposition/recompositions only, when IT needs 3000.

Finally, concerning the PCD algorithm, we observe (this is corroborated by many other experiments) that, modulo negligible changes, it stops evolving after few iterations (say 20).

The recurrent question concerning those l^0 statistics concerns the instability of the l^0 criterion. Indeed, the addition of the tinny noise on a very sparse $(\lambda_i)_{i\in I}$ leads to $l^0 = 100$.

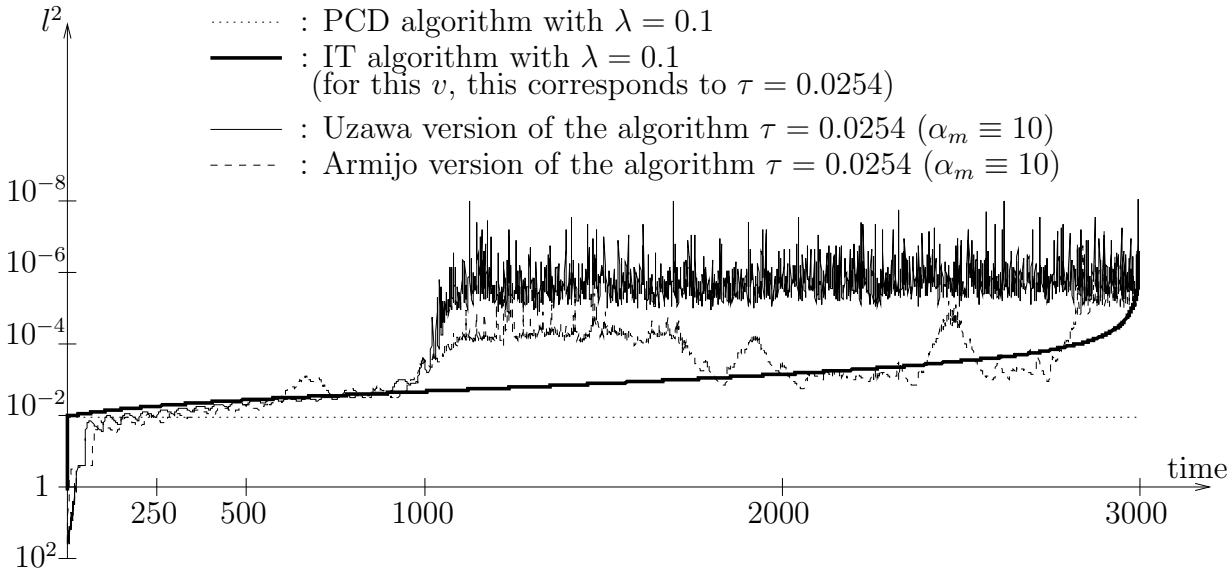

Figure 5.6: Comparison of l^2 curves : The drawn curves give the criterion l^2 (see (5.26)), as a function of the number of decomposition/recomposition, for the PCD Algorithm (see Table 1.1), IT algorithm (see Section 5.3.3), the Uzawa and Armijo versions of the Proximal Point Algorithm (see Table 5.2 and Section 5.2.7). The final norm of the residual are respectively : 0.0348, 0.0254, 0.0254 and 0.0254.

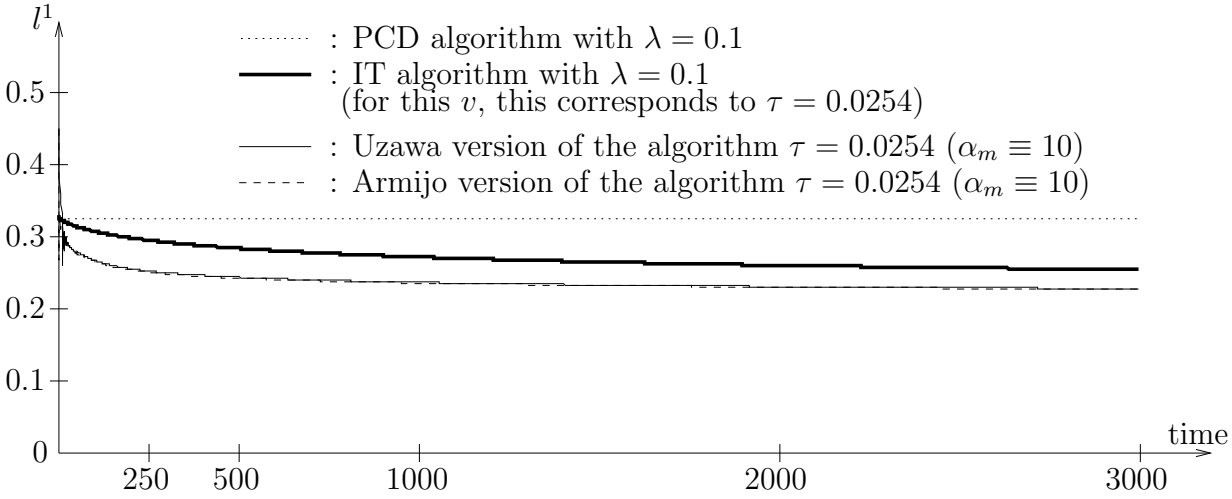

Figure 5.7: Comparison of l^1 curves : The drawn curves give the criterion l^1 (see (5.25)), as a function of the number of decomposition/recomposition, for the PCD Algorithm (see Table 1.1), IT algorithm (see Section 5.3.3), the Uzawa and Armijo versions of the Proximal Point Algorithm (see Table 5.2 and Section 5.2.7). The final values are respectively : 0.326, 0.254, 0.228 and 0.227.

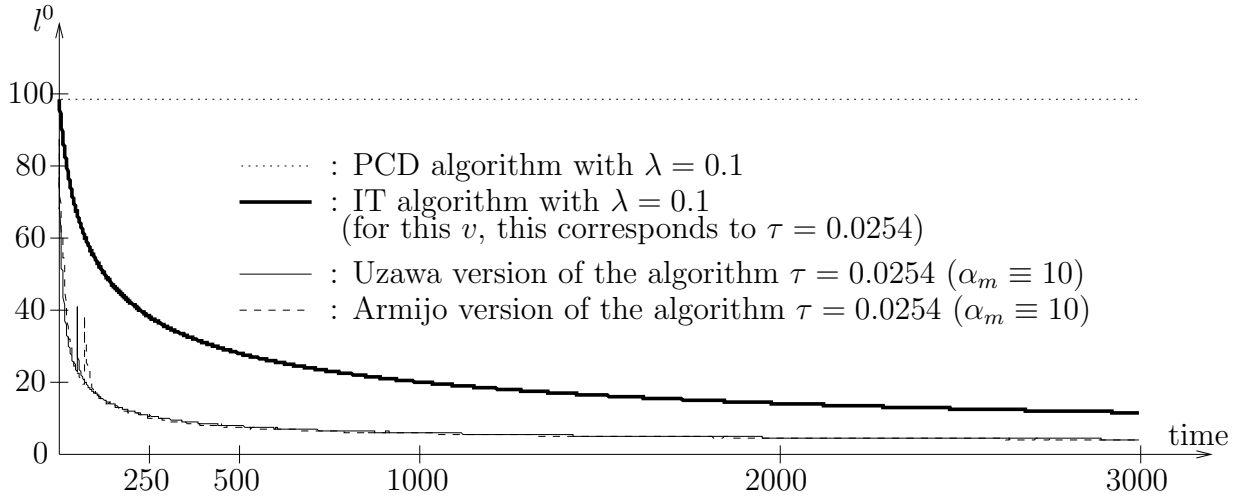

Figure 5.8: Comparison of l^0 curves : The drawn curves give the criterion l^0 (see (5.24)), as a function of the number of decomposition/recomposition, for the PCD Algorithm (see Table 1.1), IT algorithm (see Section 5.3.3), the Uzawa and Armijo versions of the Proximal Point Algorithm (see Table 5.2 and Section 5.2.7). The final values are respectively : 98.78, 11.38, 3.97 and 3.88.

In order to illustrate that the solutions provided by the IT and PCD algorithm actually differ from ours, we applied a hard-thresholding on those solutions. At each threshold corresponds new statistics l^0 and l^2. When the threshold varies we obtain the curves drawn on Figure 5.9. In order to obtain a solution as sparse as with the Proximal Point Algorithm (i.e. $l^0 = 3.97$), we need to set the threshold to a value such that we obtain $l^2 \approx 9.2$ with PCD and $l^2 \approx 4$ with IT. So, clearly, the l^0 curves are meaningful.

This is confirmed by the images of the coordinates displayed on Figure 5.10.

We also display the curves corresponding to the same experiment for $\lambda = 200$, in (5.27), this corresponds to $\tau = 15.29$ with the PCD algorithm, for the small image displayed on Figure 5.2. Although the situation is completely different (the problem is much simpler), we can draw, from these curves (see Figure 5.11, 5.12 and 5.13), exactly the same conclusions as in the previous case. Again, those curves need to be red "horizontally". For instance, the Uzawa version of the Proximal Point Algorithm reaches $l^0 = 0.66$ after 724 decomposition/recompositions and the norm of its residual is smaller than the on with the IT algorithm. In comparison the IT algorithm needs 3000 decomposition/recomposition to obtain $l^0 = 0.66$. The PCD algorithm never reaches $l^0 = 0.66$.

To conclude with these curves, notice that the Uzawa version of our algorithm reaches a fair level of convergence after few hundreds of decomposition/recomposition in the dictionary $(\psi_i)_{i \in I}$.

5.4 Conclusion

In this chapter, we exposed a numerical schema to solve a variant of Basis Pursuit. This consists to apply a proximal point algorithm to this model. The interest is to transform a non-differentiable convex problem to a sequence (quickly converging) of very regular convex problem. We showed the theoretical convergence of the algorithm. This one was confirmed by the experiment. This algorithm allows to improve remarkably the quality (in term of sparseness) of the solution compared to the state-of-the-art concerning the practical resolution of Basis Pursuit. This algorithm should have a consequent impact in

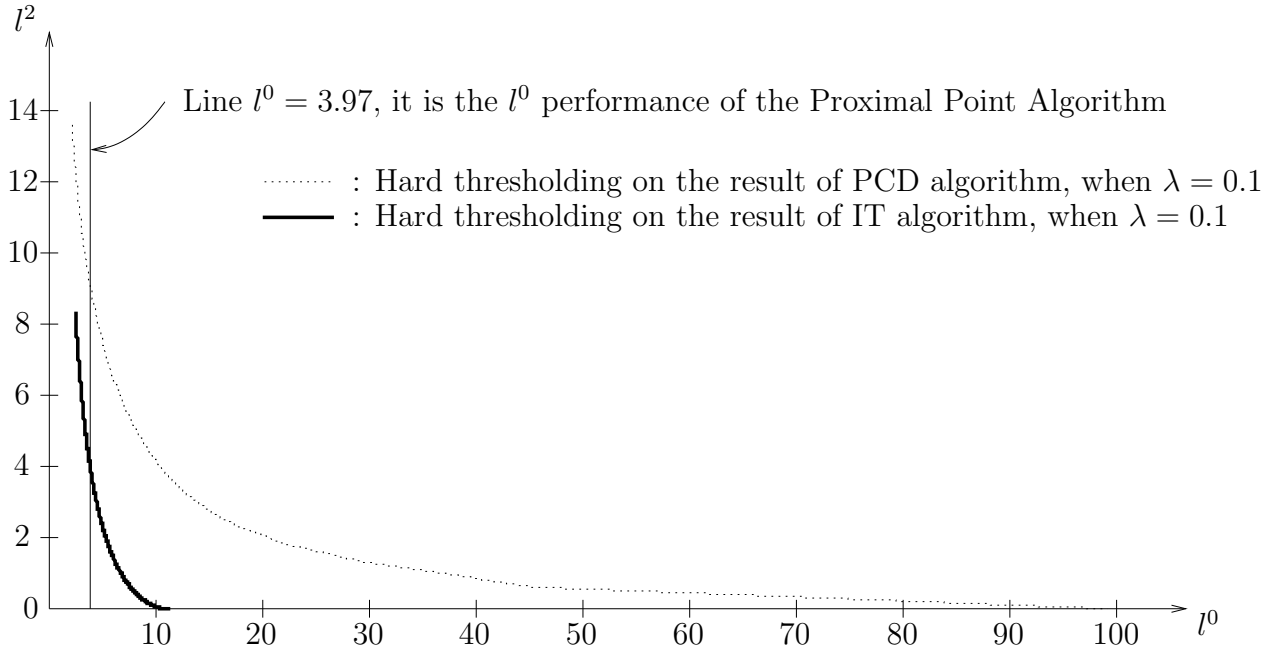

Figure 5.9: Post processing on PCD and IT algorithms: When applying a hard thresholding on the result of an algorithm, we obtain a decomposition which is represented by a point in the (l^0, l^2) plane. When the threshold varies, we obtain a curve. The curves displayed on the figure are obtained by applying this process to the result of the IT and the PCD algorithms. In order to achieve the l^0 performance of our Proximal Point Algorithm, the thresholds need to be such that $l^2 \approx 4$, with IT algorithm and $l^2 \approx 9.2$, with PCD algorithm.

this rapidly developing field.

Appendix

Proof of Proposition 6

Proof.

- ($\Leftarrow$) Denote
$$C = \{w | w = \sum_{i \in I} \lambda_i \psi_i, \forall i \in I, \lambda_i \geq 0\}.$$
Since $\#\mathcal{D}$ is finite, we know that C is a close convex set. So now suppose that there exists a $w \in \mathbb{R}^{N^2}$, but $w \notin C$. Let w^* be the projection of w on C, then $w^* \in C$ and
$$w^* = \arg\min_{v \in C} \|w - v\|^2.$$
As $w^* \in C$, there exists $(\lambda_i)_{i \in I}, \forall i \in I, \lambda_i \geq 0$ and
$$w^* = \sum_{i \in I} \lambda_i \psi_i.$$
Then for any fixed $j \in I$, $\forall d_j \geq 0$, we always have:
$$\|w - \sum_{i \in I} \lambda_i \psi_i\|^2 \leq \|w - \sum_{i \in I} \lambda_i \psi_i - d_j \psi_j\|^2.$$

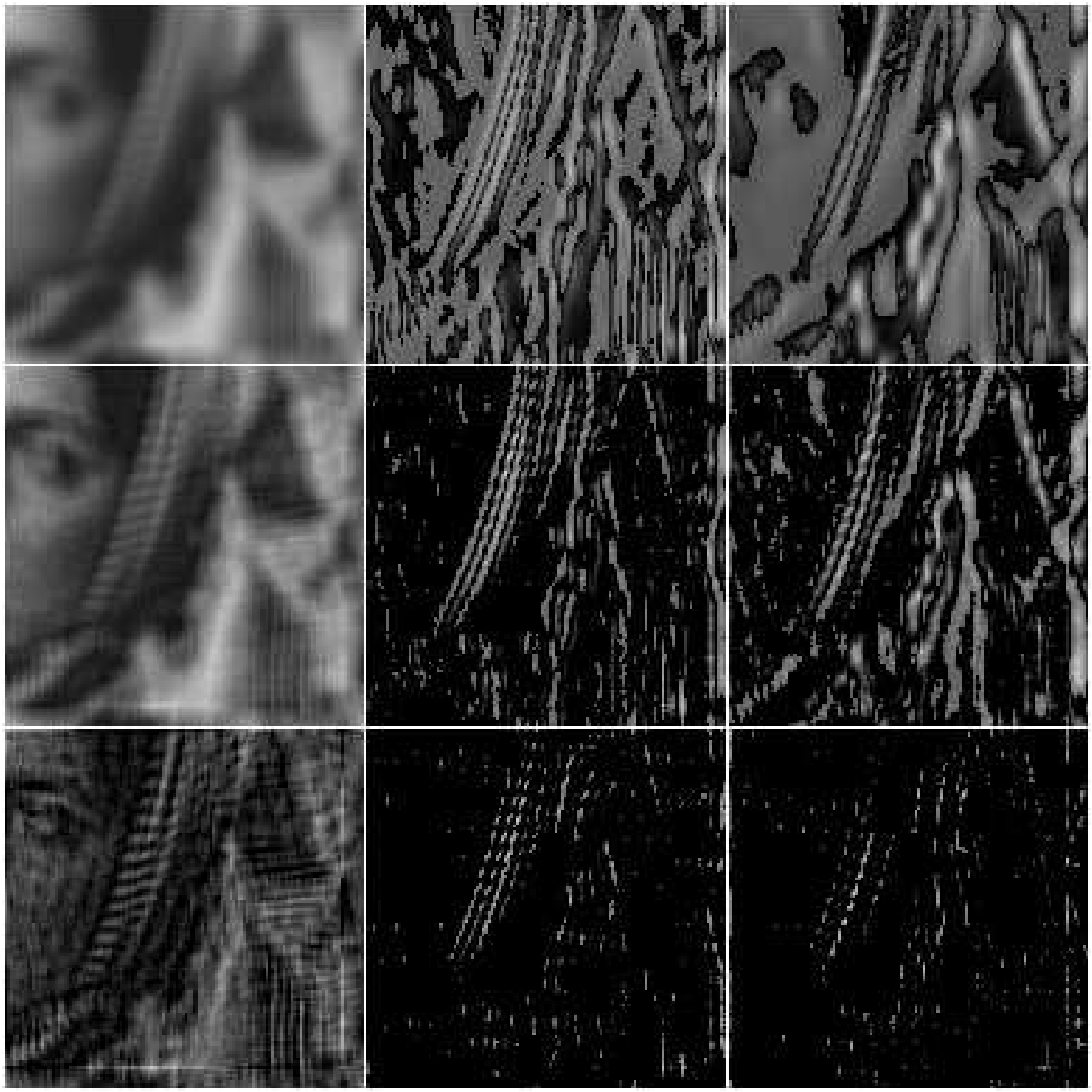

Figure 5.10: Experiment with $\tau = 0.0254$ (i.e. $\lambda = 0.1$) : Absolute values of the coordinates along the three directions (defined by the small images of the dictionary represented on Figure 5.1) along which the Proximal Point Algorithm has most non-zero coordinates. (For a good display, the coordinates are rescaled to have the same range.) The corresponding small images correspond to $(\xi, \eta) = (0, 0)$, $(0, 2)$ and $(0, 1)$. Top raw : for PCD algorithm; Middle raw : for IT; Bottom raw : for the Proximal Point Algorithm.

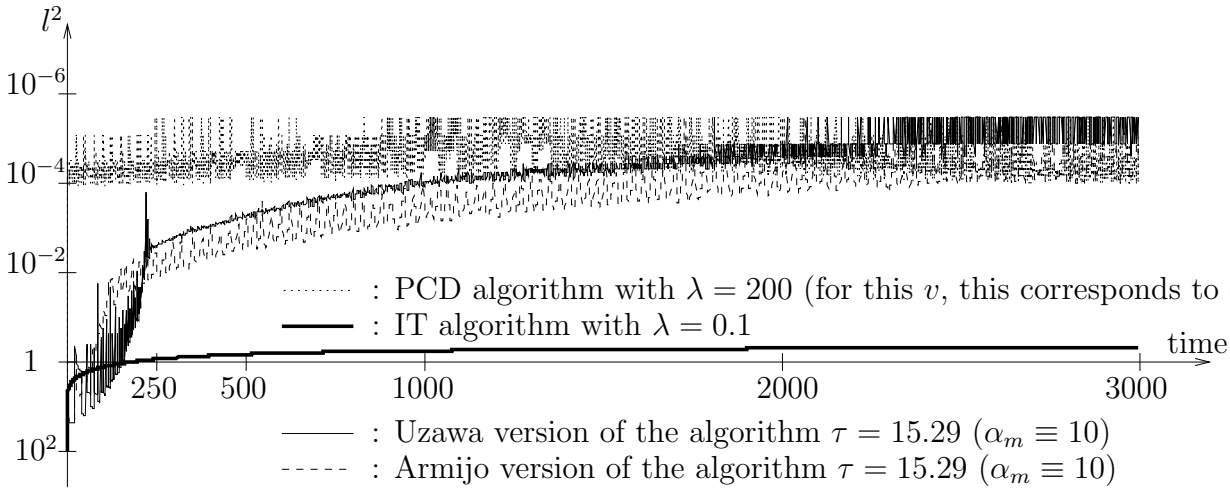

Figure 5.11: Comparison of l^2 curves : The drawn curves give the criterion l^2 (see (5.26)), as a function of the number of decomposition/recomposition, for the PCD Algorithm (see Table 1.1), IT algorithm (see Section 5.3.3), the Uzawa and Armijo versions of the Proximal Point Algorithm (see Table 5.2 and Section 5.2.7). The final norm of the residual are respectively : 15.29, 15.72, 15.29 and 15.29.

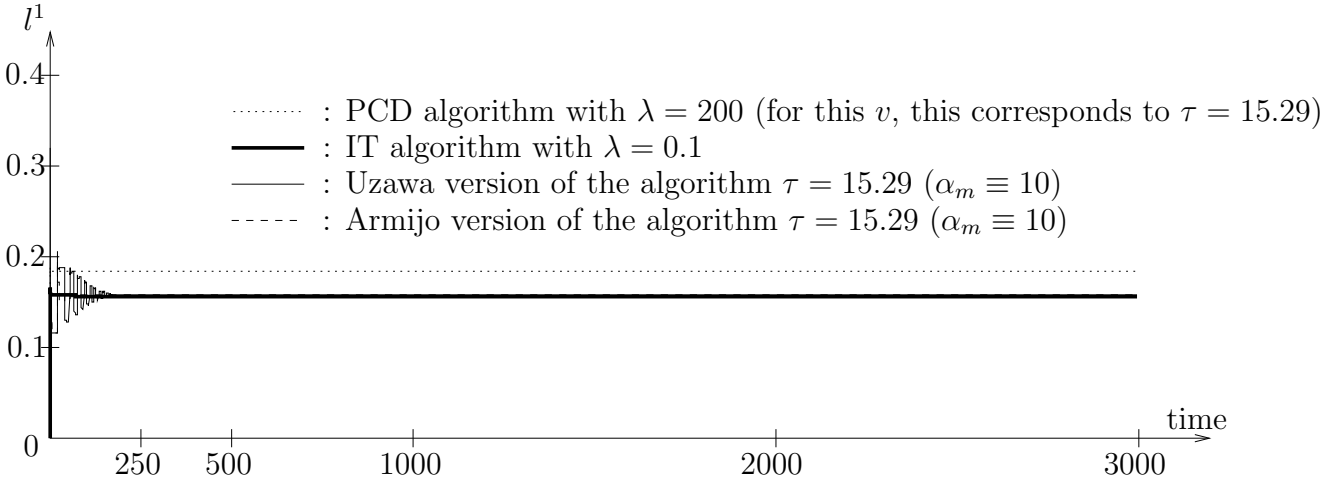

Figure 5.12: Comparison of l^1 curves : The drawn curves give the criterion l^1 (see (5.25)), as a function of the number of decomposition/recomposition, for the PCD Algorithm (see Table 1.1), IT algorithm (see Section 5.3.3), the Uzawa and Armijo versions of the Proximal Point Algorithm (see Table 5.2 and Section 5.2.7). The final values are respectively : 0.188, 0.159, 0.16 and 0.16.

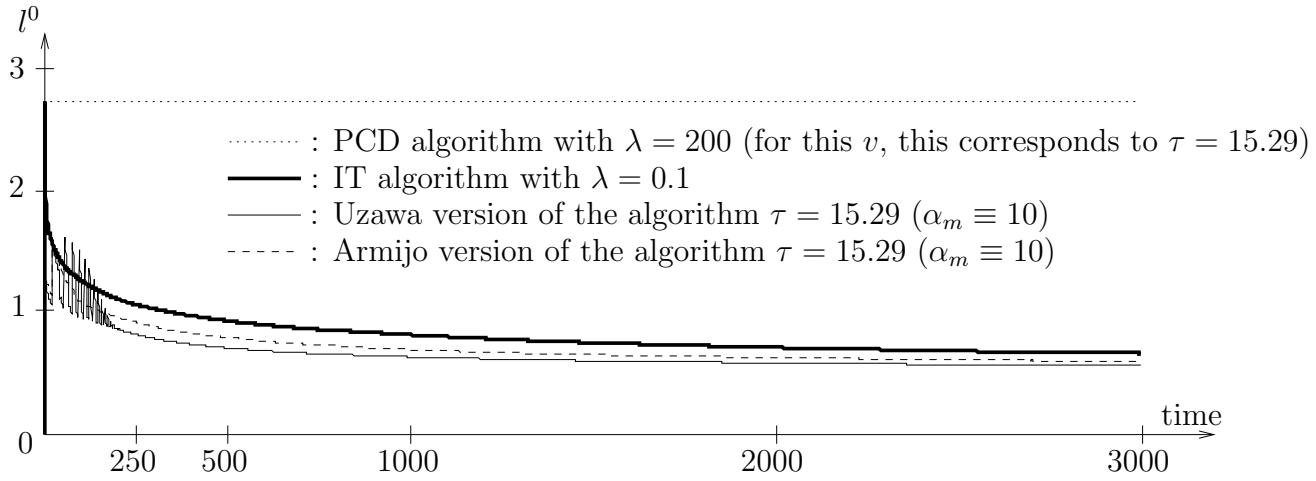

Figure 5.13: Comparison of l^0 curves : The drawn curves give the criterion l^0 (see (5.24)), as a function of the number of decomposition/recomposition, for the PCD Algorithm (see Table 1.1), IT algorithm (see Section 5.3.3), the Uzawa and Armijo versions of the Proximal Point Algorithm (see Table 5.2 and Section 5.2.7). The final values are respectively: 2.74, 0.660, 0.558 and 0.595.

So we know that $u = w - \sum_{i\in I} \lambda_i \psi_i$ is such that $u \neq 0$ and:

$$\langle u, \psi_j \rangle \leq \frac{1}{2} d_j \|\psi_j\|^2.$$

Let $d_j \to 0+$, we get

$$\langle u, \psi_j \rangle \leq 0.$$

So $\forall \gamma > 0$, we have $\gamma u \in \{w, \forall i \in I, \langle w, \psi_i \rangle \leq 1\}$. This leads a to contradiction since the latter set is bounded. This contraction tells us $C = \mathbb{R}^{N^2}$ and this finish the first part of the proof.

- ($\Rightarrow$) Set

$$W = \{w | w = \sum_i \lambda_i \psi_i, \lambda_i \geq 0, \sum_i \lambda_i \leq 1\}.$$

Then W is a non-empty convex of $\mathbb{R}^{N^2}$. Now we want to prove that 0 is an interior point of W. In fact, we know that for any $w \neq 0$, we can find a positive r_w such that:

$$\forall r \leq r_w, rw \in W.$$

Thus for any basis $(e_1, \ldots, e_N)$ of $\mathbb{R}^{N^2}$, we can find $r_0 = \min\{r_{e_1}, \ldots, r_{e_N}\} > 0$,

$$\forall r \leq r_0, \forall i \in I, re_i \in W.$$

As W is convex, we know that for any $(\lambda_i)_{i\in I}$ such that $\lambda_i \geq 0, \sum_i \lambda_i \leq r_0$, we have $\sum_{i\in I} \lambda_i e_i \in W$. Thus 0 is an interior point of W and then there exists a $r_1 > 0$ such that:

$$\forall w, \text{if } \|w\| \leq r_1, \text{then } w \in W.$$

Now for any $w \in \{w, \forall i \in I, \langle w, \psi_i \rangle \leq 1\}$, $w \neq 0$, if we let

$$w^* = \frac{r_1}{\|w\|} w,$$

then $\|w^*\| = r_1$ and $w^* \in W$. So there exists $(\lambda_i)_{i\in I}$ such that:

$$w^* = \sum_{i\in I} \lambda_i \psi_i, \text{ and } \forall i \in I, \lambda_i \geq 0, \text{ and} \sum_{i\in I} \lambda_i \leq 1.$$

Then we know that $\forall j \in I$,

$$\langle \sum_i \lambda_i \psi_i, \lambda_j \psi_j \rangle = \langle w^*, \lambda_j \psi_j \rangle \leq \lambda_j \max_i \langle w^*, \psi_i \rangle.$$

So,

$$\langle \sum_i \lambda_i \psi_i, \sum_j \lambda_j \psi_j \rangle \leq \sum_j \lambda_j \max_i \langle w^*, \psi_i \rangle,$$

and

$$\max_i \langle w^*, \psi_i \rangle \geq \frac{r_1^2}{\sum \lambda_j} \geq r_1^2.$$

Since $w^* = \frac{w}{\|w\|}$, we finally obtain

$$\|w\| \leq \frac{1}{r_1} \max_i \langle w, \psi_i \rangle \leq \frac{1}{r_1}.$$

This concludes the proof. □

Appendix : proof of Proposition 11

Proof of the first statement

The first statement is a direct application of the Theorem 2 in [52]. In order to apply this theorem, we need to show that the sequence $(u^m)_{m\in\mathbb{N}}$ is bounded and that ∂g^{-1} is Lipschitz continuous at 0 (see [52]), for g defined by (5.9).

The first assertion directly follows from Theorem 1, in [52], and the fact that (P) has a solution. We therefore know that the sequence $(u^m)_{m\in\mathbb{N}}$ is bounded.

The Lipschitz continuity of ∂g^{-1} at 0, will then follow from Proposition 7, in [52]. We are indeed going to show that, when $\|v\| > \tau$, the function g satisfies the second statement (named b) of Proposition 7, in [52].

First, notice that, when $\|v\| > \tau$, the minimizer w^* of g (i.e. the solution to (P)) is unique. The proof of this statement is straightforward and is detailed in [**?**].

We still need to show that

$$\liminf_{w\to w^*} \frac{g(w) - g(w^*)}{\|w - w^*\|^2} > 0. \tag{5.30}$$

In order to prove this last statement, let us first remark that, since $\|v\| > \tau$, $w^* \neq 0$. Therefore, $w \to \|w\| - \frac{1}{\tau}\langle w, v\rangle$ is infinitely differentiable at w^*. The second order Lagrange serie holds and, for all $w \in C$,

$$g(w) = g(w^*) + \langle \frac{w^*}{\|w^*\|} - \frac{v}{\tau}, w - w^* \rangle + \frac{1}{2\|w^*\|^3}(\|w-w^*\|^2\|w^*\|^2 - \langle w^*, w-w^*\rangle^2) + o(\|w-w^*\|^2). \tag{5.31}$$

Also, since w^* solves (P), there exists a Kuhn-Tucker vector $(\lambda_i)_{i\in I}$ such that (see [49], Th. 28.3, pp. 281)

$$\forall i \in I, \lambda_i \geq 0 \text{ and } \lambda_i(\langle w^*, \psi_i\rangle - 1) = 0 \tag{5.32}$$

and

$$\frac{w^*}{\|w^*\|} - \frac{v}{\tau} = -\sum_{i\in I} \lambda_i \psi_i. \tag{5.33}$$

Notice that, since $\|v\| > \tau$, (5.33) guarantees that there exists $i_0 \in I$ such that $\lambda_{i_0} > 0$. Notice then that, from (5.32), for any $i \in I$ such that $\lambda_i > 0$, $\langle w^*, \psi_i\rangle = 1$ and $\lambda_i\langle \psi_i, w^* - w\rangle \geq 0$, for all $w \in C$. Thus

$$\begin{aligned}
\langle \frac{w^*}{\|w^*\|} - \frac{v}{\tau}, w - w^*\rangle &= \sum_{i\in I} \lambda_i \langle \psi_i, w^* - w\rangle \\
&\geq \lambda_{i_0}\langle \psi_{i_0}, w^* - w\rangle \qquad (5.34)\\
&\geq 0. \qquad (5.35)
\end{aligned}$$

Notice that for any $w \in C$ there exists $\beta \in \mathbb{R}$ and $r \in \mathbb{R}^{N^2}$ such that $w = (1-\beta)w^* + r$, with $\langle r, w^*\rangle = 0$. Moreover, β and r are unique. Let us denote

$$E = \{w = (1-\beta)w^* + r \in C, \text{ for } \beta \geq 0, \langle r, w^*\rangle = 0 \text{ and } \|r\| \leq \frac{\beta}{2\|\psi_{i_0}\|}\}.$$

We deduce from (5.31) and (5.34) that, for all $w \in E$,

$$\begin{aligned}
g(w) - g(w^*) &\geq \lambda_{i_0}(\beta - \langle r, \psi_{i_0}\rangle) + o(\|w - w^*\|) \\
&\geq \lambda_{i_0}\frac{\beta}{2} + o(\|w - w^*\|).
\end{aligned}$$

Moreover, for $w \in E$,

$$\beta^2\|w^*\|^2 \leq \|w - w^*\|^2 = \beta^2\|w^*\|^2 + \|r\|^2 \leq (\|w^*\|^2 + \frac{1}{4\|\psi_{i_0}\|^2})\beta^2$$

So

$$\liminf_{\substack{w \to w^* \\ w \in E}} \frac{g(w) - g(w^*)}{\|w - w^*\|^2} = +\infty. \tag{5.36}$$

If $w \in C \setminus E$, we deduce from (5.31) and (5.35) that

$$g(w) - g(w^*) \geq \frac{1}{2\|w^*\|^3}(\|w - w^*\|^2\|w^*\|^2 - \langle w^*, w - w^*\rangle^2) + o(\|w - w^*\|^2).$$

Decomposing again $w = (1-\beta)w^* + r$, with $\langle r, w^*\rangle = 0$, we obtain

$$\begin{aligned}
g(w) - g(w^*) &\geq \frac{1}{2\|w^*\|^3}((\beta^2\|w^*\|^2 + \|r\|^2)\|w^*\|^2 - \beta^2\|w^*\|^4) + o(\|w - w^*\|^2) \\
&\geq \frac{\|r\|^2}{2\|w^*\|} + o(\|w - w^*\|^2). \qquad (5.37)
\end{aligned}$$

For $w \notin E$, we either have $\beta < 0$ or $\|r\| > \frac{\beta}{2\|\psi_{i_0}\|} \geq 0$. Now, if $\beta < 0$,

$$\begin{aligned}
1 &\geq \langle w, \psi_{i_0}\rangle \\
&\geq (1 - \beta) + \langle r, \psi_{i_0}\rangle.
\end{aligned}$$

So

$$\langle r, \psi_{i_0} \rangle \leq \beta$$

and

$$0 < -\beta \leq \langle -r, \psi_{i_0} \rangle \leq \|\psi_{i_0}\| \|r\|.$$

This implies that $\beta^2 \leq \|\psi_{i_0}\|^2 \|r\|^2$.

We finally obtain that, whenever $w \in C \setminus E$,

$$\beta^2 \leq 4\|\psi_{i_0}\|^2 \|r\|^2$$

and

$$\|r\|^2 \leq \|w - w^*\|^2 = \beta^2 \|w^*\|^2 + \|r\|^2 \leq (4\|\psi_{i_0}\|^2 \|w^*\|^2 + 1)\|r\|^2.$$

Together with (5.37), this guarantees that

$$\liminf_{\substack{w \to w^* \\ w \in C \setminus E}} \frac{g(w) - g(w^*)}{\|w - w^*\|^2} > 0.$$

Together with (5.36), this guarantees that (5.30) holds.

Proof of the second statement

Again, since $\|v\| > \tau$, $w^* \neq 0$. So, for any $(\lambda_i^*)_{i \in I} \in \mathcal{S}$,

$$\frac{w^*}{\|w^*\|} - \frac{1}{\tau} v + \sum_{i \in I} \lambda_i^* \psi_i = 0.$$

Since u^{m+1} converges to w^*, for m large enough, u^{m+1} cannot be zero. Given the definition of u^{m+1}, we know that

$$2\alpha_m (u^{m+1} - u^m) + \frac{u^{m+1}}{\|u^{m+1}\|} - \frac{1}{\tau} v + \sum_{i \in I} \lambda_i^m \psi_i = 0.$$

We finally obtain

$$\sum_{i \in I} (\lambda_i^* - \lambda_i^m) \psi_i = 2\alpha_m (u^{m+1} - u^m) + \frac{u^{m+1}}{\|u^{m+1}\|} - \frac{w^*}{\|w^*\|},$$

from which we obtain

$$\begin{aligned}
& \| \sum_{i \in I} (\lambda_i^* - \lambda_i^m) \psi_i \| \\
\leq \; & 2\alpha_m (\|u^{m+1} - w^*\| + \|u^m - w^*\|) + \left\| \frac{\|w^*\| u^{m+1} - \|u^{m+1}\| w^*}{\|w^*\| \|u^{m+1}\|} \right\| \\
\leq \; & 2\alpha_m \left(\frac{a}{\sqrt{a^2 + \alpha_m^2}} + 1\right) \|u^m - w^*\| + \left\| \frac{(\|w^*\| - \|u^{m+1}\|) u^{m+1} - \|u^{m+1}\| (w^* - u^{m+1})}{\|w^*\| \|u^{m+1}\|} \right\| \\
\leq \; & 2\alpha_m \left(\frac{a}{\sqrt{a^2 + \alpha_m^2}} + 1\right) \|u^m - w^*\| + \frac{\|w^*\| - \|u^{m+1}\|}{\|w^*\|} + \frac{\|w^* - u^{m+1}\|}{\|w^*\|} \\
\leq \; & \left[2\alpha_m \left(\frac{a}{\sqrt{a^2 + \alpha_m^2}} + 1\right) + \frac{2a}{\sqrt{a^2 + \alpha_m^2} \|w^*\|} \right] \|u^m - w^*\|.
\end{aligned}$$

Proof of the third statement

In order to establish the last statement of Proposition 11, we are going to show that $\left((\lambda_i^{m+1})_{i\in I}\right)_{m\in\mathbb{N}}$ is bounded in $\mathbb{R}^I$ and that any converging sequence extracted from $\left((\lambda_i^{m+1})_{i\in I}\right)_{m\in\mathbb{N}}$ converges to an element in $\mathcal{S}$.

Let us first remark that because of the definition u^{m+1} and $(\lambda_i^{m+1})_{i\in I}$, we have for any $(\lambda_i^*)_{i\in I} \in \mathcal{S}$ (as for any element of $\mathbb{R}^{+I}$),

$$L'(u^{m+1}, (\lambda_i^{m+1})_{i\in I}, u^m, \alpha_m) \geq L'(u^{m+1}, (\lambda_i^*)_{i\in I}, u^m, \alpha_m).$$

Using the definition of L', we obtain

$$\langle u^{m+1}, \sum_{i\in I} \lambda_i^{m+1}\psi_i\rangle - \sum_{i\in I}\lambda_i^{m+1} \geq \langle u^{m+1}, \sum_{i\in I}\lambda_i^*\psi_i\rangle - \sum_{i\in I}\lambda_i^*.$$

So,

$$\sum_{i\in I}\lambda_i^{m+1} \leq \sum_{i\in I}\lambda_i^* + \|u^{m+1}\| \left\|\sum_{i\in I}(\lambda_i^* - \lambda_i^{m+1})\psi_i\right\|. \tag{5.38}$$

Since $\lim_{m\to+\infty} u^m = w^*$ and $\lim_{m\to+\infty}\sum_{i\in I}(\lambda_i^* - \lambda_i^{m+1})\psi_i = 0$, we are sure that there exists $B > 0$, such that, for all $m \in \mathbb{N}$,

$$\sum_{i\in I}\lambda_i^m \leq B.$$

Let $(\overline{\lambda}_i)_{i\in I}$ be an accumulation point of $((\lambda_i^m)_{i\in I})_{m\in\mathbb{N}}$, we obtain, using (5.38),

$$\sum_{i\in I}\overline{\lambda}_i \leq \sum_{i\in I}\lambda_i^*.$$

Now, since $\lim_{m\to+\infty}\sum_{i\in I}\lambda_i^m\psi_i = \sum_{i\in I}\lambda_i^*\psi_i$, we obviously have

$$\sum_{i\in I}\overline{\lambda}_i\psi_i = \sum_{i\in I}\lambda_i^*\psi_i.$$

Using the fact that $(\lambda_i^*)_{i\in I}$ solves (D), we finally have

$$\sum_{i\in I}\overline{\lambda}_i = \sum_{i\in I}\lambda_i^*,$$

which implies $(\overline{\lambda}_i)_{i\in I} \in \mathcal{S}$.

This concludes the proof.

Proof of Proposition 12

Proof. Using (5.11),(5.12), we know that

$$f_{u^m,\alpha_m}((\lambda_i)_{i\in I}) = L'(w^*, (\lambda_i)_{i\in I}, u^m),$$

where

$$\begin{aligned} w^* &= \arg\min L'(w, (\lambda_i)_{i\in I}, u^m, \alpha_m) \\ &= \arg\min_{w\in\mathbb{R}^{N^2}} \alpha_m\|w - u^m\|^2 + \|w\| - \langle w, \frac{1}{\tau}v - \sum_{i\in I}\lambda_i\psi_i\rangle. \end{aligned}$$

When $(\lambda_i)_{i\in I}$ is fixed, w^* uniquely exists (L' is coercive and strictly convex). Hence it can be regarded as a function on $(\lambda_i)_{i\in I}$,

$$w^* = w^*((\lambda_i)_{i\in I}).$$

The explicitly formula of w^* on $(\lambda_i)_{i\in I}$ is given by (5.15).

Denote

$$x((\lambda_i)_{i\in I}) = 2\alpha_m u^m - \sum_{i\in I}\lambda_i\psi_i + \frac{v}{\tau}. \tag{5.39}$$

Using (5.15), we have:

$$w^*(\lambda_i)_{i\in I} = \begin{cases} \frac{1}{2\alpha_m}(x - \frac{x}{\|x\|}) & , \text{if } \|x\| \geq 1 \\ 0 & , \text{if } \|x\| \leq 1. \end{cases} \tag{5.40}$$

Beware that (5.40) is well defined for $\|x\| = 1$. Hence, when $\|x\| \geq 1$, $\|w^*\| = \frac{1}{2\alpha_m}(\|x\|-1)$. Thus we have,

$$f_{u^m,\alpha_m}((\lambda_i)_{i\in I}) = \begin{cases} \alpha_m\|w^* - u^m\|^2 + \frac{1}{2\alpha_m}(\|x\| - 1) - \langle w^*, x - 2\alpha_m u^m\rangle - \sum_{i\in I}\lambda_i & , \text{if } \|x\| \geq 1 \\ \alpha_m\|u^m\|^2 - \sum_{i\in I}\lambda_i & , \text{if } \|x\| \leq 1, \end{cases} \tag{5.41}$$

where x, w^* is given by (5.39), (5.40). Beware that (5.41) is well defined for $\|x\| = 1$ since $\|x\| = 1$ implies that $w^* = 0$.

Now for $(\lambda_i)_{i\in I}$ fixed, considering a vector $(\delta_i)_{i\in I}$, we want to compute the one-side directional derivative,

$$\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I}; (\delta_i)_{i\in I}) \triangleq \lim_{t\to 0^+} \frac{f_{u^m,\alpha_m}((\lambda_i)_{i\in I} + t(\delta_i)_{i\in I}) - f_{u^m,\alpha_m}((\lambda_i)_{i\in I})}{t}. \tag{5.42}$$

Indeed, we will prove that:

$$\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I}; (\delta_i)_{i\in I}) = \sum_{i\in I}(\langle w^*, \psi_i\rangle - 1)\delta_i. \tag{5.43}$$

Denote

$$\psi_0 = -\sum_{i\in I}\delta_i\psi_i. \tag{5.44}$$

Then we know that when $t_0 > 0$ is small enough, one of the following assertions must hold:

1. $[x, x + t_0\psi_0] \subset \{\xi \in \mathbb{R}^{N^2} | \|\xi\| \leq 1\}$;
2. $[x, x + t_0\psi_0] \subset \{\xi \in \mathbb{R}^{N^2} | \|\xi\| \geq 1\}$.

Indeed, this fact is obvious for $\|x\| > 1$ or $\|x\| < 1$. When $\|x\| = 1$, if the second assertion is not true, then there must exist a $t_0 > 0$ such that $\|x + t_0\psi_0\| < 1$. Since $\{\xi \in \mathbb{R}^{N^2} | \|\xi\| \leq 1\}$ is convex, the segment $[x, x + t_0\psi_0] \subset \{\xi \in \mathbb{R}^{N^2} | \|\xi\| \leq 1\}$. ie. the first assertion is true.

When the first assertion occurs, we must have $\|x\| \leq 1$, and then $w^* = 0$. Using (5.42) and the second formula of (5.41), we have:

$$\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I}; (\delta_i)_{i\in I}) = -\sum_{i\in I}\delta_i. \tag{5.45}$$

Since $w^* = 0$, (5.43) is true for this case.

Now suppose that the second assertion occurs, then $\|x\| \geq 1$. Since $\|x\| \geq 1$, the one side directional derivative of $w^*, \|x\|, x$ for direction ψ_0 all exist. Indeed, when $\|x\| > 0$,

$$\begin{aligned}
\frac{\partial}{\partial x}(\|x\|;\psi_0) &\triangleq \lim_{t\to 0^+} \frac{\|x+t\psi_0\| - \|x\|}{t} \\
&= \lim_{t\to 0^+} \frac{\|x+t\psi_0\|^2 - \|x\|^2}{t(\|x+t\psi_0\| + \|x\|)} \\
&= \lim_{t\to 0^+} \frac{2\langle x,\psi_0\rangle + t\|\psi_0\|^2}{\|x+t\psi_0\| + \|x\|} \\
&= \langle \frac{x}{\|x\|}, \psi_0\rangle.
\end{aligned}$$

Similarly, when $\|x\| \geq 1$, using (5.40), we know that:

$$\frac{\partial}{\partial x}(w^*;\psi_0) = \frac{1}{2\alpha_m}\psi_0,$$

and for all $x \in \mathbb{R}^{N^2}$,

$$\frac{\partial}{\partial x}(x;\psi_0) = \psi_0.$$

When t varies from 0 to t_0, $x((\lambda_i + t\delta_i)_{i\in I})$ varies linearly from x to $x + t_0\psi_0$. Hence, the computation of (5.42) is only involved with the one side directional derivative for direction ψ_0. Moreover, denoting

$$\Xi \triangleq \nabla x((\lambda_i)_{i\in I};(\delta_i)_{i\in I}),$$

then using (5.39), we have,

$$\Xi = -\sum_{i\in I}\delta_i\psi_i.$$

The chain rule on (5.43) (for details of directional chain rule for locally Lipschitz functions, see Lemma 5.13 of [58] or Lemma 2.2 of [59]) leads to,

$$\begin{aligned}
&\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I};(\delta_i)_{i\in I}) \\
=\ & 2\alpha_m\langle w^* - u^m, \frac{\partial}{\partial x}(w^*;\psi_0)\rangle\Xi + \frac{1}{2\alpha_m}\langle\frac{x}{\|x\|},\psi_0\rangle\Xi \\
& -\langle w^*,\psi_0\rangle - \frac{1}{2\alpha_m}\langle\psi_0, x - 2\alpha_m u^m\rangle\Xi - \sum_{i\in I}\delta_i \\
=\ & \frac{1}{2\alpha_m}(2\alpha_m w^* + \frac{x}{\|x\|} - x)\Xi - \langle w^*,\psi_0\rangle - \sum_{i\in I}\delta_i \\
=\ & -\langle w^*,\psi_0\rangle - \sum_{i\in I}\delta_i \text{ (using (5.40) for } \|x\| \geq 1) \\
=\ & \sum_{i\in I}(\langle w^*,\psi_i\rangle - 1)\delta_i \text{ (using Eq.(5.44))}.
\end{aligned}$$

Overall, (5.43) is always true and it can be rewritten as:

$$\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I};(\delta_i)_{i\in I}) = \langle(\langle w^*,\psi_i\rangle - 1)_{i\in I},(\delta_i)_{i\in I}\rangle.$$

Hence, $f_{u^m,\alpha_m}((\lambda_i)_{i\in I})$ is differential at any point $(\lambda_i)_{i\in I} \in \mathbb{R}^I$, and

$$\nabla f_{u^m,\alpha_m}((\lambda_i)_{i\in I}) = (\langle w^*, \psi_i\rangle - 1)_{i\in I}.$$

□

Chapter 6

Sparse representation in $\mathbb{R}^{N^2}$

In this chapter, we concentrate our efforts on sparse representation theory in $\mathbb{R}^{N^2}$. In Section 6.1, we first introduce a $\mathcal{D}'$-functional where $\mathcal{D}'$ is a finite dictionary of unit-norm entries in $\mathbb{R}^{N^2}$ and then using this functional, we define two sparse representation models named $(D.P_0, \tau)$, $(D.P_1, \tau)$. In Section 6.2, we present some stability results on these models. Then in Section 6.3, we introduce a fast algorithm to approximate $(D.P_1, \tau)$ for the case $\mathcal{D} = \mathcal{D}'$ and prove its convergence. Some numerical experiments on this fast algorithm are presented in Section 6.4.

6.1 Preliminaries

Before presenting the sparse representation models, let us introduce a definition.

6.1.1 $\mathcal{D}'$-functional

Definition 15 *Suppose $\mathcal{D}'$ is a finite dictionary of unit-normalized entries in $\mathbb{R}^{N^2}$. Then for any $f \in \mathbb{R}^{N^2}$, we can define the $\mathcal{D}'$-functional as:*

$$\|f\|_* = \max\left(\sup_{\psi\in\mathcal{D}'} \langle f, \psi\rangle, 0\right). \tag{6.1}$$

Proposition 16 *The $\mathcal{D}'$-functional satisfies:*

1. $\|f\|_* \geq 0, \forall f \in \mathbb{R}^{N^2}$;
2. $\|\alpha f\|_* = \alpha\|f\|_*, \forall \alpha > 0, \forall f \in \mathbb{R}^{N^2}$;
3. $\|f_1 + f_2\|_* \leq \|f_1\|_* + \|f_2\|_*, \forall f_1, f_2 \in \mathbb{R}^{N^2}$;
4. $\|f\|_* \leq \|f\|_2, \forall f \in \mathbb{R}^{N^2}$.

Proof. We only verify the triangle inequality, the others are trivial.
For any $f_1, f_2 \in \mathbb{R}^{N^2}, \psi \in \mathcal{D}'$, we have:

$$\langle f_1 + f_2, \psi\rangle = \langle f_1, \psi\rangle + \langle f_2, \psi\rangle.$$

Taking the sup and max over the right side we have,

$$\langle f_1 + f_2, \psi\rangle \leq \max\left(\sup_{\psi\in\mathcal{D}'} \langle f_1, \psi\rangle, 0\right) + \max\left(\sup_{\psi\in\mathcal{D}'} \langle f_2, \psi\rangle, 0\right).$$

Taking the max over the left side we have:

$$\sup_{\psi\in\mathcal{D}'}\langle f_1+f_2,\psi\rangle \le \max\left(\sup_{\psi\in\mathcal{D}'}\langle f_1,\psi\rangle,0\right)+\max\left(\sup_{\psi\in\mathcal{D}'}\langle f_2,\psi\rangle,0\right).$$

Thus, since the right side is non-negative,

$$\max\left(\sup_{\psi\in\mathcal{D}'}\langle f_1+f_2,\psi\rangle,0\right) \le \max\left(\sup_{\psi\in\mathcal{D}'}\langle f_1,\psi\rangle,0\right)+\max\left(\sup_{\psi\in\mathcal{D}'}\langle f_2,\psi\rangle,0\right).$$

This finishes the proof of

$$\|f_1+f_2\|_* \le \|f_1\|_*+\|f_2\|_*.$$

□

Proposition 17 *If $\mathcal{D}'=(\psi_i)_{i\in I}$ is symmetric and satisfies*

$$\{w\in\mathbb{R}^{N^2}|\forall i\in I,\langle w,\psi_i\rangle\le 1\} \text{ is bounded,}$$

then $\|\cdot\|_*$, as defined in (6.1), is a norm of $\mathbb{R}^{N^2}$.

Proof. When $\mathcal{D}'$ is symmetric, Eq.(6.1) can be rewritten as:

$$\|f\|_*=\sup_{\psi\in\mathcal{D}'}|\langle f,\psi\rangle|,$$

and we know that, for any $\alpha\in\mathbb{R}$,

$$\|\alpha f\|_*=|\alpha|\|f\|_*.$$

Together with the triangle inequality, this shows that $\|\cdot\|_*$ is a semi-norm. Now for any $f\in\mathbb{R}^{N^2}$, if $\|f\|_*=0$, then for any $\gamma\in\mathbb{R}$, we still has $\|\gamma f\|_*=0$. But

$$\{w\in\mathbb{R}^{N^2}|\forall i\in I,\langle w,\psi_i\rangle\le 1\} \text{ is bounded,}$$

and γf is in this set. So the only choice is that $f=0$. This shows that $\|\cdot\|$ is a norm.

□

6.1.2 Sparse representation models

Suppose that the dictionary $\mathcal{D}=(\varphi_i)_{i\in I}$ is a finite family of unit-norm vectors in $\mathbb{R}^{N^2}$. For $v\in\mathbb{R}^{N^2}$, we consider the problem of finding the sparsest possible representation in the dictionary $\mathcal{D}$. As a measure of sparsity of a vector $(\lambda)_{i\in I}$, we take the so-called l^0 norm $\|(\lambda)_{i\in I}\|_0$, which is simply the number of non-zero elements in $(\lambda_i)_{i\in I}$. The sparsest representation is then the solution to the optimization problem

$$(P_0):\min_{(\lambda_i)_{i\in I}}\|(\lambda)_{i\in I}\|_0 \text{ subject to } v=\sum_{i\in I}\lambda_i\varphi_i. \tag{6.2}$$

As the original Problem (P_0) is unrealistic to compute, the classical approach, Basis Pursuit (see Chapter 1) is to approximate (P_0) by replacing the l^0-norm with an l^1 norm:

$$(P_1): \min_{(\lambda_i)_{i\in I}} \sum_{i\in I} |\lambda_i| \text{ subject to } v = \sum_{i\in I} \lambda_i \varphi_i. \tag{6.3}$$

This can be cast as a Linear Program (LP), for which various exact solutions or approximations have already been discussed in literature. The BP is known to give highly sparse solutions to problem known to have such sparse solutions (see [13, 60]). It has been shown that it could, in some specific cases, outperform the greedy Matching Pursuit approach in generating sparse solutions (see [13, 60]).

6.1.3 Presence of noise

In most practical situations it is not sensible to assume that the available data v obey precise equality $v = \sum_{i\in I} \lambda_i \varphi_i$ with a sparse representation $(\lambda_i)_{i\in I}$. A more plausible scenario assumes *sparse approximate representation*: that there is an ideal noiseless signal/image u with a sparse representation $u = \sum_{i\in I} \mu_i \varphi_i$ with $\|(\mu)_{i\in I}\|_0$ small, but we can observe only a noisy version $v = u + b$, where $\|b\|_2 \leq \epsilon$.

We can adapt to this noisy setting by modifying (P_0),(P_1) to include a noise allowance and a non-negative direction. In order to do so, we consider another dictionary $\mathcal{D}'$ (this defines the functional $\|\cdot\|_*$, as in (6.1)) and the optimization problems:

$$(D.P_{0,\tau}): \begin{cases} \text{minimize} & \|(\lambda)_{i\in I}\|_0 \\ \text{subject to} & \|v - \sum \lambda_i \varphi_i\|_* \leq \tau \\ & \lambda_i \geq 0, \forall i \in I, \end{cases} \tag{6.4}$$

and

$$(D.P_{1,\tau}): \begin{cases} \text{minimize} & \sum \lambda_i \\ \text{subject to} & \|v - \sum \lambda_i \varphi_i\|_* \leq \tau \\ & \lambda_i \geq 0, \forall i \in I. \end{cases} \tag{6.5}$$

These two models will be studied in this chapter. In these models, every element φ in $\mathcal{D}$ is a possible pattern/atom for the recomposition of the original image u. $\mathcal{D}'$ represents those interesting directions and plays the same role as the dictionary in the $TV - l^\infty$ model (see Chapter 2 and 3).

6.2 Stability Results

The concept of *mutual coherence* of the dictionary $\mathcal{D}$, which appeared in [60] and references therein, plays an important role in the stability result.

Definition 18 *Assuming $\mathcal{D} = (\varphi_i)_{i\in I}$ is such that for all $i \in I$, $\|\varphi_i\| = 1$, we define the mutual coherence as:*

$$\nu = \nu(\mathcal{D}) = \max_{k,j\in I, k\neq j} |\langle \varphi_k, \varphi_j \rangle|. \tag{6.6}$$

We also need an important Lemma (Lemma 2.9 of [60]).

Lemma 19 *Given an s-by s symmetric matrix H with diagonal entries equal to one and off-diagonal entries not larger than ν in amplitude, the smallest eigenvalue of H is at least $1 - \nu(s-1)$.*

6.2.1 Stability of $(D.P_{0,\tau})$

Theorem 20 *Let the dictionary $\mathcal{D} = (\varphi_i)_{i\in I}$ has mutual coherence $\nu(\mathcal{D})$ and there exists a constant $c_0 = c_0(\mathcal{D}, \mathcal{D}')$ such that for any subset $\Gamma \subset \mathcal{D}$, we always have:*

$$\frac{1}{\#\Gamma}\sum_{\varphi\in\Gamma}\langle u,\varphi\rangle^2 \le c_0\|u\|_*^2 \tag{6.7}$$

where $\|\cdot\|_*$ is defined in (6.1)*. If some representation of the noiseless image $u = \sum_{i\in I}\mu_i\varphi_i$ satisfies*

$$N_0 = \|(\mu_i)_{i\in I}\|_0 \le (1/\nu + 1)/2, \tag{6.8}$$

and

$$\|v - \sum_{i\in I}\mu_i\varphi_i\|_2 \le \epsilon,$$

then $(\mu_i)_{i\in I}$ is the unique sparsest representation of u; moreover, when $\tau \ge \epsilon$, the result $\hat{\lambda}_{0,\tau,\epsilon}$ of $(D.P_{0,\tau})$ applied at the noisy data v approximates $(\mu_i)_{i\in I}$:

$$\|\hat{\lambda}_{0,\tau,\epsilon} - (\mu_i)_{i\in I}\|_2 \le \frac{\sqrt{2c_0N_0}(\epsilon+\tau)}{1-\nu(2N_0-1)}.$$

Proof. The first part is just the same as Th.2.1 of [60]. For the second part, let $\chi = \hat{\lambda}_{0,\tau,\epsilon} - (\mu_i)_{i\in I}$, we know that:

$$\|\sum_{i\in I}\chi_i\varphi_i\|_* \le \epsilon + \tau.$$

But χ only has at most $2N_0$ non-zero entries (remember that $\|(\mu_i)_{i\in I}\| = N_0$ and $(\mu_i)_{i\in I}$ is in the feasible set of $(D.P_{0,\tau})$ since $\tau \ge \epsilon$, hence $\|(\hat{\lambda}_{0,\tau,\epsilon})\| \le \|(\mu_i)_{i\in I}\| = N_0$), so there is only at most $2N_0$ entries of $\mathcal{D}$ involved. For simplicity, we suppose those are

$$\mathcal{D}_s = \{\varphi_1, \ldots, \varphi_s\},$$

and $\chi_s = (\chi(1), \ldots, \chi(s))^T$ where $s \le 2N_0$. So we have:

$$\|\sum_{k=1}^{s}\chi_s(k)\varphi_k\|_* \le \epsilon + \tau.$$

Using Eq.(6.7), we know that:

$$\sum_{i=1}^{s}\langle\sum_{k=1}^{s}\chi_s(k)\varphi_k, \varphi_i\rangle^2 \le c_0 s(\epsilon+\tau)^2.$$

Denote $G_s = (\langle\varphi_k, \varphi_j\rangle)_(1 \le k, j \le s)$, then the above inequality can be rewritten as:

$$\chi_s^T G_s^T G_s \chi_s \le c_0 s(\epsilon+\tau)^2.$$

Using Lemma 19, we have

$$\chi_s^T G_s^T G_s \chi_s \ge \chi_s^T\chi_s \cdot \sigma_{min}^2\{G_s\} \ge \|\chi_s\|_2^2(1-\nu(s-1))^2 \ge \|\chi_s\|_2^2(1-\nu(2N_0-1))^2,$$

where the last inequality is based on the fact $s \le 2N_0$ and (6.8).

So we have proved:

$$\|\chi\|_2 = \|\chi_s\|_2 \le \frac{\sqrt{c_0 s}(\epsilon+\tau)}{1-\nu(N_0-1)} \le \frac{\sqrt{2c_0N_0}(\epsilon+\tau)}{1-\nu(2N_0-1)}.$$

□

Remark. When $\mathcal{D} \bigcup \{-\psi, \psi \in \mathcal{D}\} \subset \mathcal{D}'$, (6.7) holds for $c_0 = 1$. Indeed, we have in this case:

$$|\langle u, \varphi \rangle| \leq \|u\|_*, \forall \varphi \in \mathcal{D}.$$

Hence, for arbitrary $\Gamma \subset \mathcal{D}$, we have:

$$\sum_{\varphi \in \Gamma} \langle u, \varphi \rangle^2 \leq \#\Gamma \cdot \|u\|_*^2.$$

6.2.2 Stability of $(D.P_{1,\tau})$

We want to prove a stability result similar to the work of Donoho ([60]). Suppose we are given a signal $v = u + b$ where b is an additive noise, known to satisfy $\|b\| \leq \epsilon$. We apply $(D.P_{1,\tau})$ with the dictionaries $\mathcal{D}, \mathcal{D}'$ and τ to this image (not necessarily with $\tau = \epsilon$) i.e. we solve $(D.P_{1,\tau})$ and obtain a solution $\hat{\lambda}_{1,\tau,\epsilon}$. We study its deviation from the ideal representation $u = \sum_{i \in I} \mu_i \varphi_i$.

Theorem 21 *Let the dictionary $\mathcal{D} = (\varphi_i)_{i \in I}$ has mutual coherence $\nu(\mathcal{D})$ and there exists a constant $\gamma_0 = \gamma_0(\mathcal{D}, \mathcal{D}')$ such that for any $(\xi_i)_{i \in I} \in \mathbb{R}^I$, we have:*

$$\| \sum_{i \in I} \xi_i \varphi_i \|^2 \leq \gamma_0 \cdot \sum_{i \in I} |\xi_i| \cdot \| \sum_{i \in I} \xi_i \varphi_i \|_*. \tag{6.9}$$

If some representation of the noiseless image $u = \sum_{i \in I} \mu_i \varphi_i$ satisfies

$$N_0 = \|(\mu)_{i \in I}\|_0 > (1/\nu + 1)/4, \tag{6.10}$$

and

$$\|v - \sum_{i \in I} \mu_i \varphi_i\|_2 \leq \epsilon,$$

then this is the unique sparsest representation of u; moreover, when $\tau \geq \epsilon$, the result $\hat{\lambda}_{1,\tau,\epsilon}$ of $(D.P_{1,\tau})$ applied at the noisy data v approximates $(\mu_i)_{i \in I}$:

$$\|\hat{\lambda}_{1,\tau,\epsilon} - \mu\|_2 \leq \frac{2\sqrt{N_0}\gamma_0(\epsilon + \tau)}{1 - \nu(4N_0 - 1)}.$$

Proof. First, the assertion that $(\mu_i)_{i \in I}$ is the unique sparsest representation follows from Th.2.1 of Donoho (see [60]) and the fact that $\frac{1+\nu}{4\nu} < \frac{1+\nu}{2\nu}$. From now on, we denote $\mu = (\mu_i)_{i \in I}$.

Second, the stability bound can be posed as the solution to an optimization problem of the form:

$$\max_{\mu, b} \|\hat{\lambda} - \mu\|_2 \text{ subject to } \left\{ \begin{array}{c} \hat{\lambda} = \arg\min_{\lambda \in \mathbb{R}^I} \|\lambda\|_1 \text{ subject to } \|v - \sum_{i \in I} \lambda_i \varphi_i\|_* \leq \tau \\ v = \sum_{i \in I} \mu_i \varphi_i + b, \|b\|_2 \leq \epsilon, \|(\mu_i)_{i \in I}\|_0 \leq N_0. \end{array} \right\}. \tag{6.11}$$

In words, we consider all the representation vectors μ of bounded support, and all possible realizations of bounded noise, and we ask for the largest error between the ideal sparse decomposition and its reconstruction from noisy data. Defining $\vartheta = \mu - \lambda$, and similarly $\chi = \hat{\lambda} - \mu$, we can rewrite the above problem as:

$$\max_{\mu,b} \|\chi\|_2 \text{ subject to } \left\{ \begin{array}{c} \chi = \arg\min_\vartheta \|\mu - \vartheta\|_1 \text{ subject to } \|b + \sum_{i\in I} \vartheta_i \varphi_i\|_* \le \tau \\ \|b\|_2 \le \epsilon, \|(\mu_i)_{i\in I}\|_0 \le N_0. \end{array} \right\}, \tag{6.12}$$

We will estimate an upper bound of the maximum of (6.12) by a sequence of relaxations, each one expanding the feasible set and increasing the maximal value. To begin, note that if χ is the minimizer of $\|\mu - \vartheta\|_1$ under these constraints, then relaxing the constraints to all χ satisfying $\|\mu - \chi\| \le \|\mu\|$ expands the feasible set. Attention here we used the fact $\tau \ge \epsilon$, so $\vartheta = 0$ is in the feasible set. Thus, we consider:

$$\left\{ \chi \;\middle|\; \begin{array}{c} \|\mu - \chi\|_1 \le \|\mu\| \text{ and } \|b + \sum_{i\in I} \chi_i \varphi_i\|_* \le \tau \\ \|b\|_2 \le \epsilon, \#\mathcal{S} \le N_0. \end{array} \right\}. \tag{6.13}$$

We now expand this set by exploiting the relation

$$\|\mu - \chi\|_1 - \|\mu\|_1 \ge \|\chi\|_1 - 2\sum_{k\in\mathcal{S}} |\chi(k)|,$$

where $\mathcal{S}$ is the support of the non-zeros in μ with complement $\mathcal{S}^c$, and we used $|a-b|-|a| \ge |a|-|b|-|a| = -|b|$. Therefore, we get a further increase in value by replacing the feasible set in (6.13) with

$$\max_{\chi,\mathcal{S},b} \|\chi\|_2 \text{ subject to } \left\{ \begin{array}{c} \|\chi\|_1 \le 2\sum_{k\in\mathcal{S}} |\chi(k)| \text{ and } \|b + \sum_{i\in I} \chi_i \varphi_i\|_* \le \tau \\ \|b\|_2 \le \epsilon, \#\mathcal{S} \le N_0. \end{array} \right\}, \tag{6.14}$$

We next simplify our analysis by eliminating the noise vector b, using

$$\{\chi| \quad \exists b, \|b + \sum_{i\in I} \chi_i \varphi_i\|_* \le \tau \text{ and } \|b\|_2 \le \epsilon\} \subset \{\chi| \quad \|\sum_{i\in I} \chi_i \varphi_i\|_* \le \tau + \epsilon\}. \tag{6.15}$$

Expanding the feasible set of (6.14) with this observation gives

$$\max_{\chi,\mathcal{S}} \|\chi\|_2 \text{ subject to } \left\{ \begin{array}{c} \|\chi\|_1 \le 2\sum_{k\in\mathcal{S}} |\chi(k)|, \quad \|\sum_{i\in I} \chi_i \varphi_i\|_* \le \Delta \\ \#\mathcal{S} \le N_0. \end{array} \right\}, \tag{6.16}$$

where we introduced $\Delta = \epsilon + \tau$. Now using (6.9) we have,

$$\|\sum_{i\in I} \chi_i \varphi_i\|_* \le \Delta \Rightarrow \|\sum_{i\in I} \chi_i \varphi_i\|_2^2 \le \gamma_0 \|\chi\|_1 \Delta.$$

Using this fact, we can expand the feasible set to:

$$\max_{\chi,\mathcal{S}} \|\chi\|_2 \text{ subject to } \left\{ \begin{array}{c} \|\chi\|_1 \le 2\sum_{k\in\mathcal{S}} |\chi(k)|, \quad \|\sum_{i\in I} \chi_i \varphi_i\|_2^2 \le \gamma_0 \|\chi\|_1 \Delta \\ \#\mathcal{S} \le N_0. \end{array} \right\}. \tag{6.17}$$

The constraints $\|\sum_{i\in I} \chi_i \varphi_i\|_2^2 \le \gamma_0 \|\chi\|_1 \Delta$ is still not posed in terms of the absolute values in the vector χ, complicating the analysis; we now relax this constraint using incoherence of $\mathcal{D}$. Again the Gram matrix is $G = (\langle \varphi_i, \varphi_j \rangle)_{i,j\in I}$, and the mutual coherence is the maximal off-diagonal amplitude: $\nu = \max_{i\neq j} |G(i,j)|$. Let $abs(\chi)$ be the vector $(|\chi_i|)_{i\in I}$. We also use a similar notation for matrices. Also, let 1_c be the $\#I$-by-$\#I$ matrix whose entries are all equal to one, Id the $\#I$-by-$\#I$ identity matrix. The constraint

$$\|\sum_{i\in I} \chi_i \varphi_i\|_2^2 = \chi^T G \chi \le \gamma_0 \|\chi\|_1 \Delta,$$

can be relaxed to:

$$\begin{aligned}\gamma_0\|\chi\|_1\Delta \geq \chi^T G\chi &= \|\chi\|^2 + \chi^T(G-Id)\chi \\ &\geq \|\chi\|_2^2 - abs(\chi)^T\ abs(G-Id)\ abs(\chi) \\ &\geq \|\chi\|_2^2 - \nu\cdot abs(\chi)^T\ abs(1_c - Id)\ abs(\chi) \\ &= (1+\nu)\|\chi\|_2^2 - \nu\|\chi\|_1^2.\end{aligned}$$

Using this fact, the maximum in (6.17) is upper-bounded by the value

$$\max_{\chi,\mathcal{S}} \|\chi\|_2 \text{ subject to } \left\{ \begin{array}{c} \|\chi\|_1 \leq 2\sum_{k\in\mathcal{S}}|\chi(k)| \\ (1+\nu)\|\chi\|_2^2 - \nu\|\chi\|_1^2 \leq \gamma_0\|\chi\|_1\Delta \\ \#\mathcal{S} \leq N_0 \end{array} \right\}. \tag{6.18}$$

This problem is invariant under permutations of the entries in χ which preserve membership in $\mathcal{S}$ and $\mathcal{S}^c$. It is also invariant under relabeling of coordinates. So assume that all non-zeros in μ are concentrated in the initial slots of the vector, i.e. that $\mathcal{S} = \{0,\ldots,N_0-1\}$.

Putting $\chi = (\chi_0,\chi_1)$ where χ_0 contains the N_0 first entries in χ and χ_1 contains the remaining $\#I - N_0$ entries of χ, we obviously have

$$\|\chi\|_2^2 = \|\chi_0\|_2^2 + \|\chi_1\|_2^2$$

and

$$\|\chi\|_1 = \|\chi_0\|_1 + \|\chi_1\|_1.$$

The l^1-norm on $\mathbb{R}^k$ dominates the l^2 norm and is dominated by $\sqrt{k}$ times the l^2 norm. Thus

$$\|\chi_0\|_1 \geq \|\chi_0\|_2 \geq \frac{\|\chi_0\|_1}{\sqrt{N_0}},$$

$$\|\chi_1\|_1 \geq \|\chi_1\|_2 \geq \frac{\|\chi_1\|_1}{\sqrt{\#I - N_0}}.$$

We define

$$A = \|\chi_0\|_1, B = \|\chi_1\|_1, c_0 = \left(\frac{\|\chi_0\|_2}{\|\chi_0\|_1}\right)^2, c_1 = \left(\frac{\|\chi_1\|_2}{\|\chi_1\|_1}\right)^2. \tag{6.19}$$

Returning to the problem given in (6.18), and using our notations, we obtain a further reduction, from an optimization problem on $\mathbb{R}^{\#I}$ to an optimization problem on $(A,B,c_0,c_1) \in \mathbb{R}^4$:

$$\max\sqrt{c_0A^2 + c_1B^2} \text{ subject to } \left\{ \begin{array}{c} A > B \\ (1+\nu)(c_0A^2 + c_1B^2) - \nu(A+B)^2 \leq (A+B)\gamma_0\Delta \\ A,B \geq 0, \frac{1}{N_0} \leq c_0 \leq 1, 0 < c_1 \leq 1 \end{array} \right\}. \tag{6.20}$$

We further define $B = \rho A$, where $0 \leq \rho < 1$ and rewrite (6.20) as,

$$\max A\sqrt{c_0 + \rho^2 c_1} \text{ subject to } \left\{ \begin{array}{c} (1+\nu)\frac{c_0+\rho^2c_1}{1+\rho}A - \nu(1+\rho)A \leq \gamma_0\Delta \\ A \geq 0, \frac{1}{N_0} \leq c_0 \leq 1, 0 < c_1 \leq 1, 0 \leq \rho < 1 \end{array} \right\}. \tag{6.21}$$

Define $\xi = (1+\rho)/\sqrt{c_0+\rho^2c_1}$. Then $\frac{1}{\sqrt{2}} \leq \xi \leq 2\sqrt{N_0}$ over the region (6.21). Setting $V = A\sqrt{c_0+\rho^2c_1}$, the first constraint defining that region takes the form

$$[(1+\nu)\frac{1}{\xi} - \nu\xi]V \leq \gamma_0\Delta. \tag{6.22}$$

Our hypothesis (6.10) guaranties that,

$$(1+\nu) - \nu\xi^2 \geq 1 - \nu(4N_0 - 1) > 0. \tag{6.23}$$

Hence,

$$V \leq \frac{\xi\gamma_0\Delta}{(1+\nu) - \nu\xi^2} \leq \frac{2\sqrt{N_0}\gamma_0\Delta}{1 - \nu(4N_0 - 1)}. \tag{6.24}$$

□

Remark When $\mathcal{D}\bigcup\{-\varphi, \varphi \in \mathcal{D}\} \subset \mathcal{D}'$, (6.9) holds with $\gamma_0 = 1$. Indeed, under this condition, $\forall(\xi_i)_{i\in I} \in \mathbb{R}^{\#I}$, we have:

$$|\langle\sum_{i\in I}\xi_i\varphi_i, \varphi_k\rangle| \leq \|\sum_{i\in I}\xi_i\varphi_i\|_*, \forall k \in I.$$

Hence,

$$\langle\sum_{i\in I}\xi_i\varphi_i, \xi_k\varphi_k\rangle \leq |\xi_k| \cdot \langle\sum_{i\in I}\xi_i\varphi_i, \varphi_k\rangle| \leq |\xi_k| \cdot \|\sum_{i\in I}\xi_i\varphi_i\|_*, \forall k \in I.$$

Taking the sum over $k \in I$ leading to (6.9), with $\gamma_0 = 1$.

6.3 Soft-Threshold Matching Pursuit

In this section, we want to develop a fast algorithm to approximate the solution of $(D.P_{1,\tau})$ when $\mathcal{D}' = \mathcal{D}$. We consider $\mathcal{D} = \mathcal{D}' = (\psi_i)_{i\in I}$ and suppose that $\forall i \in I, \|\psi_i\| = 1$. Beware that we do not force that $\mathcal{D}$ is symmetric in this section. We will detail this case in an upcoming chapter.

We consider the following problem:

$$(P'') : \begin{cases} \min & \sum\lambda_i \\ \text{subject to} & \langle v - \sum_{i\in I}\lambda_i\psi_i, \psi\rangle \leq \tau, \text{for } \psi \in \mathcal{D} \\ & \lambda_i \geq 0, \text{for any } i \in I. \end{cases} \tag{6.25}$$

We propose a Soft-Threshold Matching Pursuit (STMP) scheme to approximation its solution. This STMP algorithm is an iterative greedy process that decomposes the function $v \in \mathbb{R}^{N^2}$, using the dictionary $\mathcal{D} = (\psi_i)_{i\in I} \subset \mathbb{R}^{N^2}$. Recall that for all $i \in I$, ψ_i is a function with unit norm.

Our STMP algorithm builds iteratively some coordinates in $(\psi_i)_{i\in I}$.

Set $\lambda_i = 0$ for all $i \in I$ and $R^0v = v$.

Repeat(loop for $k \geq 0$):

- Search $\gamma_k \in I$ such that:
$$\gamma_k = \arg\max_{i\in I}\langle R^kv, \psi_i\rangle.$$
- Update λ_{γ_k} by:
$$\lambda_{\gamma_k} \leftarrow \lambda_{\gamma_k} + \max(\langle R^kv, \psi_{\gamma_k}\rangle - \tau, 0).$$

- Update $R^{k+1}v$ by:

$$R^{k+1}v \leftarrow v - \sum_{i\in I} \lambda_i \psi_i.$$

Repeat n times and denote the result $(\lambda_i)_{i\in I}$ as $(\lambda_i^n)_{i\in I}$.

The goal of this STMP algorithm is to find a feasible solution $(\lambda_i)_{i\in I}$ for Problem (P'') and in the mean time to keep $\sum_{i\in I}\lambda_i$ be as small as possible.

We can prove that:

Theorem 22 *When $n \to +\infty$,*

$$\sum_{i\in I} \lambda_{i\in I}^n \psi_i$$

converges.

Proof. See Appendix. □

Theorem 23 *Denote F_0 the feasible set of Problem (P'') and*

$$C_0 = \{w \in \mathbb{R}^{N^2} | \exists (\mu_i)_{i\in I} \in F_0, w = \sum_{i\in I} \mu_i \psi_i\}.$$

Denote

$$u^* = \lim_{n\to+\infty} \sum_{i\in I} \lambda_i^n \psi_i.$$

If F_0 is non-vide, then:

$$u^* \in C_0.$$

Proof. See Appendix. □

This theorem shows that if Problem (P'') is feasible, then the result of STMP algorithm converges to a point in the feasible set. This can be used as an initial point to the penalization algorithm to solve Problem (P'').

6.4 Experiments

6.4.1 STMP for approximation

We use this algorithm to approximate the top-right image of Figure 6.1. We consider a translation-invariant dictionary built on the features presented in Figure 6.2 (their mean is zero and they are normalized).

Figure 6.1 shows the results of STMP. The middle-left and bottom-left image are $\sum_{n=0}^{+\infty} s_n \psi_{\gamma_n}$ for $\tau = 50$ and 100 respectively. The middle-right and bottom-right image are the residues $R^{+\infty}v$. This figure illustrates that, when τ augments, more information is left in the residual image.

The left image of Figure 6.3 shows s_n as a function of n, for $\tau = 50$ and $\tau = 100$. From this image we clearly see that in both cases, $(s_n)_{n\in\mathbb{N}}$ rapidly decreases to zero. The right image of Figure 6.3 shows $s_n^{50} - s_n^{100}$ where s_n^t means the coefficient s_n, for $\tau = t$. Clearly, the coefficients s_n, for $\tau = 100$, is below those for $\tau = 50$.

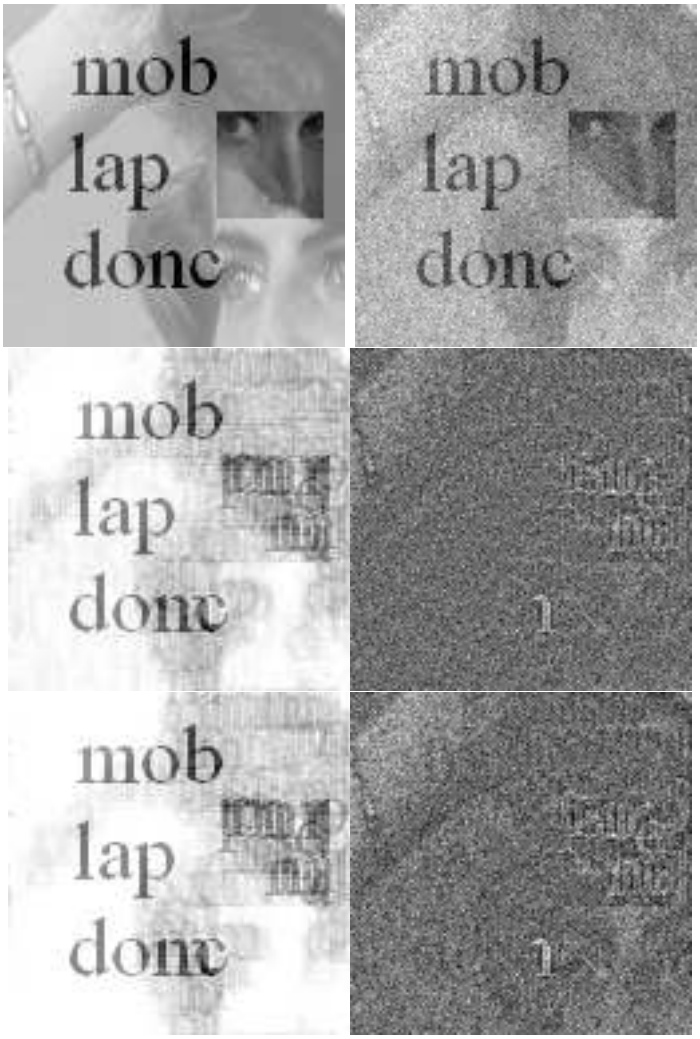

Figure 6.1: Results of STMP: top-left: original image, top-right: noisy image with $\sigma = 20$; middle-left: result for $\tau = 50$, middle-right: residue for $\tau = 50$; bottom-left: result for $\tau = 100$, bottom-right: residue for $\tau = 100$.

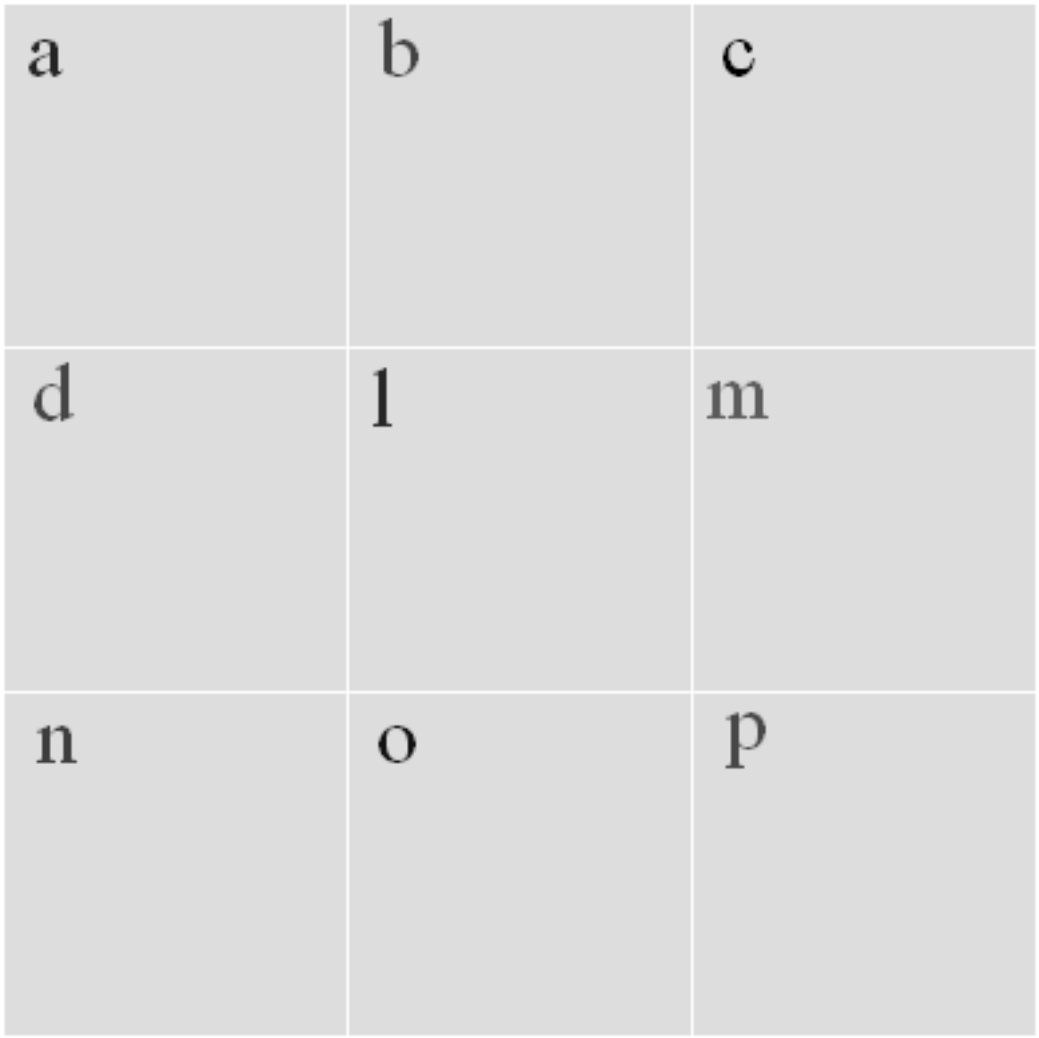

Figure 6.2: Features to build the translation-invariant dictionary

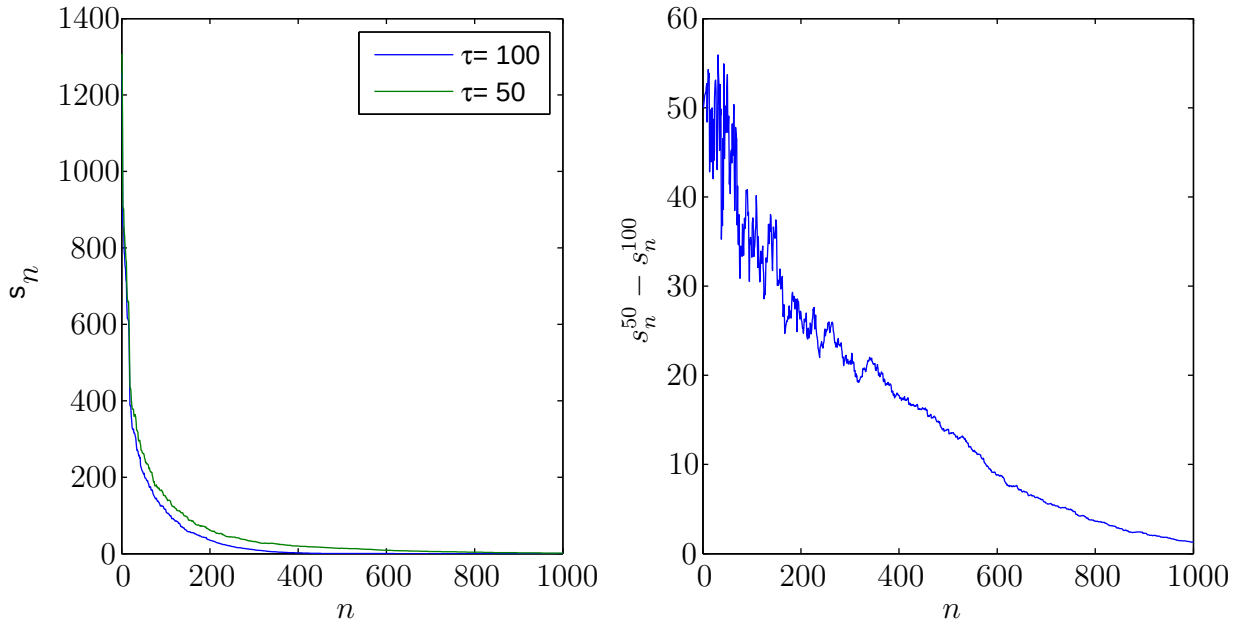

Figure 6.3: STMP: left: s_n as a function on n, for $\tau = 100$ and $\tau = 50$; right: $s_n^{50} - s_n^{100}$ where s_n^t stands for the coefficients s_n, for $\tau = t$.

6.4.2 STMP for image decomposition

The feature dictionary contains two parts. The first part contains 9 filters: the letters which are shown in Figure 6.2. The second part contains 13 filters $\{d_1, \ldots, d_{13}\}$ which corresponds to the Daubechie-3 wavelet basis of level 4 and their opposites. (see Figure 3.6). Hence the size of the feature dictionary is $9 + 2 \times 13 = 35$. We use this feature dictionary to build a translation-invariant dictionary $\mathcal{D}$. Using this $\mathcal{D}$, we try to represent a noisy image v which is shown as top-right of Figure 6.4.

Figure 6.4 shows the STMP for $\tau = 0$. The top-right is the noisy image. The middle-left is the reconstructed image

$$\sum_{i \in I} \lambda_i^n \psi_i$$

with $n = 4000, \tau = 0$. Since neither the wavelet filters nor the letter filters are good for representation of noise, the residual (middle-right) contains most of the noise. The letter part and the background part are shown as bottom-left, bottom-right of Figure 6.4.

Figure 6.5 shows the STMP for $\tau = 100$. The top-left is the reconstructed image which contains less information than middle-left of Figure 6.4. The residual image is in top-right. It contains more information than the middle-right of Figure 6.4. The letter part(bottom-left) is cleaner than the bottom-left of Figure 6.4 as it contains less information. The background part (bottom-right) also contains less noise when compared to bottom-right of Figure 6.4.

6.5 Conclusion

In this chapter, we adapted to the cases of a variational model, whose regularization term is that of Basis Pursuit and whose data-fidelity term is that of the $TV - l^\infty$ model, a result

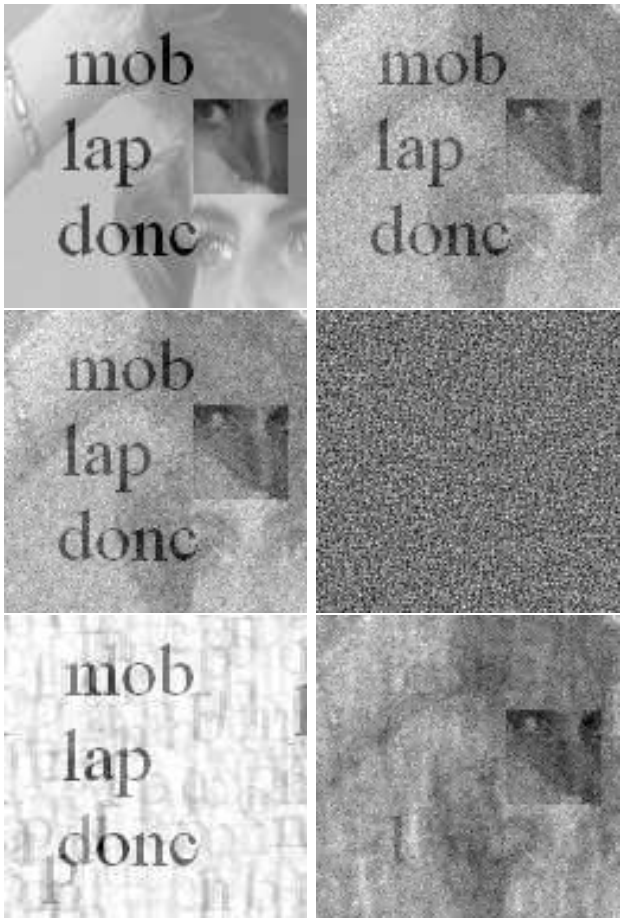

Figure 6.4: STMP for $\tau = 0$: top-left: original image, top-right: noisy image with $\sigma = 20$; middle-left: STMP with $\tau = 0$ and $n = 4000$, middle-right: residual image corresponding to the middle-left image; bottom-left: the letter part; bottom-right: the background part.

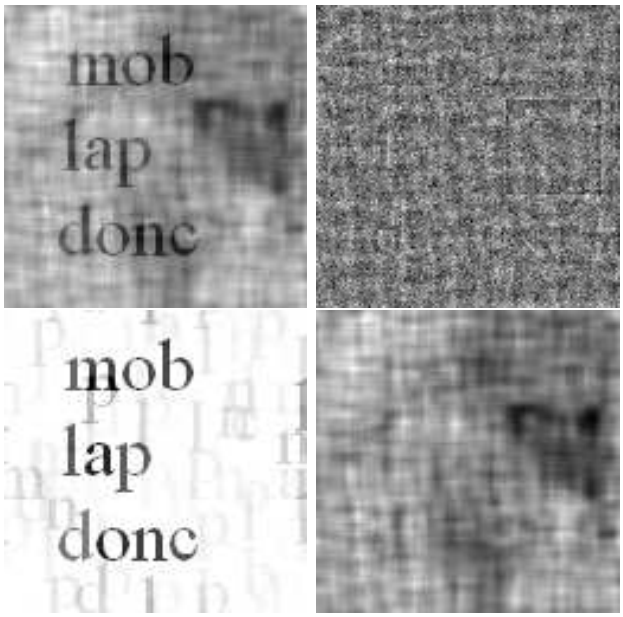

Figure 6.5: STMP for $\tau = 100$: top-left: the result image; top-right: the residual; bottom-left: the letter part; bottom-right: the background part.

of D. Donoho. This result showed that, under a condition relating the dictionary defining the regularization term to the dictionary defining the data-fidelity term, it is possible to extend the results of D. Donoho to the models which interest us in this chapter. The obtained result says that, if the given data is very sparse, the solution of the model is close to its most sparse decomposition. This guarantee the stability of this model within this framework and establishes a link between l^1 and l^0 regularization, for this type of data-fidelity term.

Appendix

Proof of Theorem 22

Proof. Denote $M_n = \langle R^n v, \psi_{\gamma_n} \rangle$ and $s_n = \max(M_n - \tau, 0)$. Thus

$$R^0 v = v$$

and

$$\gamma_n = \arg\max_{i \in I} \langle R^n v, \psi_i \rangle,$$

$$R^n v = R^{n+1} v + \max(M_n - \tau, 0)\psi_{\gamma_n}.$$

As for any $n \in \mathbb{N}$:

$$\sum_{i \in I} \lambda_i^n \psi_i = \sum_{i=0}^{n-1} s_i \psi_{\gamma_i},$$

we need to prove the convergence of

$$\sum_{n=0}^{+\infty} s_n \psi_{\gamma_n}.$$

By recurrence, we can prove that,

$$\|v\|^2 = \sum_{n=0}^{M-1} \Big(s_n^2 + 2 s_n (M_n - s_n) \Big) + \|R^M v\|^2. \tag{6.26}$$

As $s_n \geq 0, M_n \geq s_n$, we know that:

$$\sum_{n=0}^{+\infty} s_n^2 < +\infty.$$

Then using Eq.(6.26) again,

$$\sum_{n=0}^{+\infty} s_n M_n < +\infty.$$

But $\tau s_n \leq s_n M_n$, thus:

$$\sum_{n=0}^{+\infty} s_n \leq \frac{1}{\tau} \sum_{n=0}^{+\infty} s_n M_n < +\infty.$$

Then easily we know that: $(\sum_{n=0}^{m-1} s_n \psi_{\gamma_n})_{m \in \mathbb{N}}$ is a Cauchy sequence and then $\sum_{n=0}^{m} s_n \psi_{\gamma_n}$ exists.

□

Proof of Theorem 23

Still using the Notations in the Proof of Theorem 22.
Suppose that $u^* \notin C_0$. Thus there exists a $\psi \in \mathcal{D}$ such that,

$$\langle v - u^*, \psi \rangle = \tau + \delta_0,$$

where $\delta_0 > 0$.
Since $\sum_{i=0}^{+\infty} s_i \psi_{\gamma_i}$ exists, there exist N_0 such that $\forall m \geq N_0$, we have:

$$\| \sum_{i=m}^{+\infty} s_i \psi_{\gamma_i} \| \leq \frac{\delta_0}{2}.$$

Hence,

$$\begin{aligned} \langle v - \sum_{i=0}^{m-1} s_i \psi_{\gamma_i}, \psi \rangle &= \langle v - u^*, \psi \rangle - \langle \sum_{i=m}^{+\infty} s_i \psi_{\gamma_i}, \psi \rangle \\ &\geq \tau + \delta_0 - \| \sum_{i=m}^{+\infty} s_i \psi_{\gamma_i} \| \\ &\geq \tau + \frac{1}{2}\delta_0. \end{aligned}$$

It implies that $\forall m \geq N_0$:

$$M_m \geq \tau + \frac{1}{2}\delta_0.$$

Thus,

$$s_m \geq \frac{1}{2}\delta_0, \ \forall m \geq N_0.$$

This contradicts, $\lim_{n \to +\infty} s_n = 0$. Thus $u^* \in C_0$.

Chapter 7

MP shrinkage in Hilbert space

This chapter contains a MP shrinkage approach with general dictionary. As this chapter is closely related to the work of [32], we adopt the view of that paper for the convenience of the readers who are familiar with this work. Hence, we use the notation of the Hilbert space (eg. $\mathcal{H} = L^2(\mathbb{R}^2)$ or $\mathcal{H} = \mathbb{R}^{N^2}$). Beware that in the experimental part, we use the space $\mathbb{R}^{N^2}$, as usual.

We reaffirm that all the dictionaries that we consider in this chapter is still of finite size. Hence, the dictionary is not a redundant basis when the underlying Hilbert space $\mathcal{H}$ is of infinite dimension.

7.1 General shrinkage function

Before presenting the MP shrinkage method, let us introduce a family of shrinkage function.

Definition 24 *A function $\theta(\cdot) : \mathbb{R} \to \mathbb{R}$ is called **a general shrinkage function** if and only if it satisfies:*

1. $\theta(0) = 0$;
2. $\theta(\cdot)$ *is nondecreasing i.e*
$$\theta(x) \leq \theta(y), \forall x \leq y, x, y \in \mathbb{R};$$
3. $\theta(\cdot)$ *is a shrinkage i.e*
$$|\theta(x)| \leq |x|, \forall x \in \mathbb{R}.$$

We have the following proposition.

Proposition 25 *For any general shrinkage function $\theta\cdot$, we have:*

$$\theta(x)(x - \theta(x)) \geq 0, \forall x \in \mathbb{R}. \tag{7.1}$$

Proof. When $x \geq 0$, using the definition, we know that:

$$0 \leq \theta(x) \leq x.$$

Hence, $\theta(x)(x - \theta(x)) \geq 0$. The similar discussion holds for $x \leq 0$. □

Definition 26 *The gap of a general shrinkage function $\theta(\cdot)$ is defined as:*

$$r_0 = \sup\{r \in \mathbb{R}^+ | \theta(x) > 0 \Rightarrow \theta^2(x) + 2\theta(x)(x - \theta(x)) \geq r^2\}. \tag{7.2}$$

If the gap $r_0 > 0$, the function is called gap shrinkage function and if the gap $r_0 = 0$, the function is called non-gap shrinkage function.

Definition 27 *A general shrinkage function $\theta(\cdot)$ is τ-controlled if one of the following is satisfied:*

- $\tau > 0$ *and*
$$|x| \leq \tau \Rightarrow \theta(x) = 0; \tag{7.3}$$
- $\tau = 0$ *and there exists a constant* $0 < c_0 \leq 1$ *such that:*
$$c_0|x| \leq |\theta(x)| \leq |x|. \tag{7.4}$$

Definition 28 *A τ-controlled shrinkage function $\theta(\cdot)$ is strictly τ-controlled and if one of the following is satisfied:*

- $\tau > 0$ *and*
$$\theta(x) \to 0 \Rightarrow d(x, (-\tau, \tau)) \to 0, \tag{7.5}$$
i.e.
$$\forall \epsilon > 0, \exists \delta > 0, \forall x, |\theta(x)| \leq \delta \Rightarrow d(x, (-\tau, \tau)) \leq \epsilon, \tag{7.6}$$
where $d(x, (-\tau, \tau)) = \inf_{y \in (-\tau, \tau)} |x - y|$;
- $\tau = 0$.

Example

1. For $\tau > 0$, the soft-threshold function $\rho_\tau(\cdot)$ defined in Eq.(1.2) is a non-gap shrinkage function i.e the gap $r_0(\rho_\tau) = 0$. This function is strictly τ-controlled.
2. For $\tau > 0$, the hard-threshold function defined as following,

$$h_\tau(t) = \begin{cases} t & \text{if } |t| \geq \tau \\ 0 & \text{otherwise,} \end{cases} \tag{7.7}$$

is a gap shrinkage function with gap $r_0(h_\tau) = \tau$. This function is strictly τ-controlled.
3. The identity function defined as:

$$i(t) = t, \forall t \in \mathbb{R}, \tag{7.8}$$

is a non-gap shrinkage function. This function is strictly 0-controlled.
4. For $\tau > 0$, the Non-Negative Garrote threshold function(see [61]) defined as:

$$\delta_\tau^G(t) = t\left(1 - \frac{\tau^2}{t^2}\right)^+, \tag{7.9}$$

is strictly τ-controlled, non-gap.
5. For $0 < \tau_1 < \tau_2$, the firm shrinkage function(see [62]) defined as:

$$\delta_{\tau_1,\tau_2}(t) = \begin{cases} 0 & |t| \le \tau_1 \\ sign(t)\frac{\tau_2(|t|-\tau_1)}{\tau_2-\tau_1} & \tau_1 < |t| < \tau_2 \\ t & |t| \ge \tau_2, \end{cases} \tag{7.10}$$

is strictly τ_1-controlled, non-gap.

6. For $p \in \mathbb{N}$, $\tau > 0$, the generalized threshold function(see [63]) defined as:

$$\delta_\tau^p(t) = t - tI(|t| \le \tau) - \frac{\tau^p}{t^{p-1}} I(|t| > \tau)(sign(t)^p), \tag{7.11}$$

is strictly τ-controlled shrinkage function, non-gap when $p < +\infty$. When $p = 1$, it is soft-threshold function; when $p = +\infty$, it is hard-threshold function, with gap τ. When $p = 2$ it is actually Non-Negative Garrote threshold function.

All these shrinkage functions are strictly τ-controlled with some $\tau \ge 0$, this implies that in fact this is a very general class of shrinkage function.

7.2 MP shrinkage in Hilbert space

Let $\mathcal{H}$ be a Hilbert space and $v \in \mathcal{H}$. We define a dictionary as a family $\mathcal{D} = (\psi_i)_{i\in I}$ of vectors in $\mathcal{H}$, such that $\forall i \in I$, $\|\psi_i\|_2 = 1$. We assume that v contains some noise. We aim to find a linear expansion approximating the analyzed signal/image v in the presence of noise. The MP algorithm (see Chapter 1) is widely used to generate an adaptive representation for this image v. The wavelet shrinkage (see Chapter 1) is also widely used to denoise images. In this section, we will propose an algorithm (called MP shrinkage) which combines these two important algorithms.

7.2.1 The details of MP shrinkage algorithm

Throughout this chapter, we always assume that $t \mapsto \theta(t)$ of $\mathbb{R} \mapsto \mathbb{R}$ is a general shrinkage function.

The MP shrinkage method is defined recursively. Recall that $v \in \mathcal{H}$ fixed. Let $R^0 v = v$. We suppose that we have computed the n-th order residue $R^n v$ for $n \ge 0$. We choose an elements $\psi_{\gamma_n} \in \mathcal{D}$ which best matches the residue $R^n v$:

$$\gamma_n = \arg\max_{i\in I} |\langle R^n v, \psi_i\rangle|.$$

The residue $R^n v$ is sub-decomposed into

$$R^n v = \theta(\langle R^n v, \psi_{\gamma_n}\rangle)\psi_{\gamma_n} + R^{n+1} v$$

which defines the residue at the order $n + 1$.

More clearly, the details of MP shrinkage algorithm are given in Table 7.1. When θ is the identity function, this is just the MP; when $\mathcal{D}$ is the wavelet basis and θ is hard/soft threshold function, this is Donoho's wavelet shrinkage.

Remark. If in the step 2 of Table 7.1, the choice of the atom is made according to:

find an atom ψ_{γ_n} satisfies:

$$|\langle \psi_{\gamma_n}, R^n v\rangle| \ge \alpha \max_{i\in I} |\langle R^n v, \psi_i\rangle|,$$

where $0 < \alpha \le 1$ is a predefined constant. Then almost all of the theoretical results of this section still hold. We do not present the details here.

Task: MP shrinkage algorithm for fixed $\tau \geq 0$

1. initialize $R^0 v = v$
2. repeat for $n = 0, \ldots, +\infty$
   - find the best atom ψ_{γ_n} by:
   $$\gamma_n = \arg\max_{i \in I} |\langle R^n v, \psi_i \rangle|,$$
   - sub-decompose the residue $R^n v$ into
   $$R^n v = s_n \psi_{\gamma_n} + R^{n+1} v$$
   where $M_n = \langle R^n v, \psi_{\gamma_n} \rangle$, $s_n = \theta(M_n)$.
3. take:
   $$u = \sum_{n=0}^{+\infty} s_n \psi_{\gamma_n}$$

Table 7.1: Details of MP shrinkage algorithm

7.2.2 Theoretical aspects on MP shrinkage

Now for MP shrinkage the energy conservation corresponding to (1.17) of MP becomes:

Proposition 29 *For MP shrinkage, we have:*

$$\|v\|^2 = \sum_{n=0}^{M-1} \left(s_n^2 + 2 s_n (M_n - s_n) \right) + \|R^M v\|^2, \tag{7.12}$$

$$\|v\|^2 \geq \sum_{n=0}^{M-1} s_n^2 + \|R^M v\|^2, \tag{7.13}$$

$$\sum_{n=0}^{+\infty} s_n^2 < +\infty, \tag{7.14}$$

where $M_n = \langle R^n v, \psi_{\gamma_n} \rangle$, $s_n = \theta(M_n)$ is defined in Table 7.1.

Proof. A straightforward calculation gives,

$$\begin{aligned} s_n \langle \psi_{\gamma_n}, R^{n+1} v \rangle &= s_n \left(\langle \psi_{\gamma_n}, R^n v \rangle - s_n \langle \psi_{\gamma_n}, \psi_{\gamma_n} \rangle \right) \\ &= s^n (M_n - s_n) \\ &\geq 0 \end{aligned}$$

where the last inequality is from Eq.(7.1).
We can therefore deduce from

$$R^n v = s_n \psi_{\gamma_n} + R^{n+1} v,$$

that

$$\|R^n v\|^2 = \|R^{n+1} v\|^2 + 2 s^n (M_n - s_n) + s_n^2 \geq \|R^{n+1} v\|^2 + s_n^2.$$

Using $R^0 v = v$, we then obtained (7.12), (7.13) by a recursion. (7.14) directly follows from (7.12). □

Theorem 30 *For MP shrinkage, there exists a sequence* $(g_m)_{m\in\mathbb{N}}$ *of* $\mathbb{N}$ *such that:*

$$\sum_{n=0}^{g_m-1} s_n \psi_{\gamma_n}$$

converges when m *tends to* $+\infty$.

Proof. From Proposition 29, we know that:

$$\forall n \in \mathbb{N}, \|R^n v\| \leq \|v\|.$$

As $R^n v$ is in the finite-dimension subspace $v + Span\{\mathcal{D}\}$, thus a convergent subsequence can be extracted from $(R^n v)_{n\in\mathbb{N}}$. Since:

$$\sum_{n=0}^{m-1} s_n \psi_{\gamma_n} = v - R^m v,$$

a convergent subsequence can be extracted from $(\sum_{n=0}^{m-1} s_n \psi_{\gamma_n})_{n\in\mathbb{N}}$. □

Theorem 31 *If the shrinkage function* $\theta(\cdot)$ *is* τ*-controlled with* $\tau \geq 0$ *then*

$$\sum_{n=0}^{m-1} s_n \psi_{\gamma_n}$$

converges when m *tends to* $+\infty$.

Proof. If $\tau = 0$, then this situation contains MP as special case. We can modify the proof of Theorem 2 in [32] to adapt this kind of generation and prove that $\sum_{n=0}^{m-1} s_n \psi_{\gamma_n}$ converges. We leave the details as Appendix.

Now we suppose that $\tau > 0$ and $\theta(\cdot)$ is τ-controlled. If $\sum_{n=0}^{+\infty} s_n \psi_{\gamma_n}$ contains only finite number of terms, it obviously converges. Let us suppose that it has infinitely matching terms. This guaranties that:

$$|M_n| > \tau, \forall n \in \mathbb{N}.$$

Using (7.12), (7.13) we know that:

$$\sum_{n=0}^{+\infty} s_n M_n < +\infty.$$

As s_n, M_n have the same sign, we have:

$$\sum_{n=0}^{+\infty} |s_n| \cdot |M_n| < +\infty.$$

But $|s_n| \cdot |M_n| \geq \tau |s_n|$, we obtain:

$$\sum_{n=0}^{+\infty} |s_n| < +\infty.$$

As for any $N_1 \leq N_2$, we have:

$$\|R^{N_1}v - R^{N_2}v\| = \|\sum_{n=N_1}^{N_2-1} s_n\psi_{\gamma_n}\| \leq \sum_{n=N_1}^{N_2-1} |s_n|.$$

Thus $(R^m v)_{m\in\mathbb{N}}$ is a cauchy sequence, and $(R^m v)_{m\in\mathbb{N}}$ converges. This tell us that: $(\sum_{n=0}^{m-1} s_n\psi_{\gamma_n})_{m\in\mathbb{N}}$ converges as m tends to $+\infty$.

□

This theorem tell us when $\theta(\cdot)$ is τ-controlled, then $\sum_{n=0}^{+\infty} s_n\psi_{\gamma_n}$ exists, thus $(R^m v)_{m\in\mathbb{N}}$ converges as $m \to +\infty$, we denote $R^{+\infty}v$ its limit. Before saying where $R^{+\infty}v$ is, we present a corollary.

Corollary 32 *For MP shrinkage,*

1. *when $n \to +\infty$, s_n tends to 0, i.e.*

$$\lim_{n\to+\infty} s_n = 0;$$

2. *if $\theta(\cdot)$ is strictly τ-controlled with $\tau > 0$, then*

$$\lim_{n\to+\infty} d(M_n, -(\tau,\tau)) = 0.$$

Proof. From (7.14) of Proposition(29), we know that

$$\sum_{n=0}^{+\infty} s_n^2 < +\infty.$$

Thus we have:

$$\lim_{n\to+\infty} s_n = 0.$$

Now, for the second part. For any fixed $\epsilon > 0$, as $\theta(\cdot)$ is strictly τ-controlled, we know that there exists a $\delta > 0$ such that:

$$\forall x, |\theta(x)| \leq \delta \Rightarrow d(x, (-\tau,\tau)) \leq \epsilon.$$

But $\lim_{n\to+\infty} s_n = 0$, so for this δ, there exits a N_0, such that $\forall n \geq N_0$, we have $|s_n| \leq \delta$, then using the fact: $s_n = \theta(M_n)$, we know:

$$d(M_n, (-\tau,\tau)) \leq \epsilon.$$

This complete the proof. □

Now, we denote V the closed linear span of the vectors in $\mathcal{D}$, i.e.

$$V = Span\{\mathcal{D}\} \tag{7.15}$$

and W the orthogonal complement of V in $\mathcal{H}$. The orthogonal projectors over V and W are denoted by P_V and P_W respectively.

Theorem 33 *For MP shrinkage, if $\theta(\cdot)$ is strictly τ-controlled with $\tau \geq 0$, and we denote the convex*

$$C = \{g \in \mathcal{H} | \ |\langle g, \psi \rangle| \leq 1, \forall \psi \in \mathcal{D}\},$$

then

$$\lim_{n \to +\infty} d(R^n v, P_W v + \tau C) = 0.$$

Hence,

$$R^{+\infty} v \in (P_W v + \tau C) \bigcap (v + Span\{\mathcal{D}\}),$$

and

$$\sum_{n=0}^{+\infty} s_n \psi_{\gamma_n} \in (P_V v + \tau C) \bigcap Span\{\mathcal{D}\}.$$

Proof. When $\tau = 0$, it is nearly the same as the proof of MP in [32]. We leave the details in Appendix.

We now suppose that $\tau > 0$. Using the Corollary 32, we know that :

$$\lim_{n \to +\infty} d(M_n, (-\tau, \tau)) = 0. \tag{7.16}$$

As

$$R^M v = v - \sum_{n=0}^{M-1} s_n \psi_{\gamma_n},$$

and as the projection operator is linear, we know that for all M:

$$P_W(R^M v) = P_W(v), \quad P_V(R^M v) = P_V(v) - \sum_{n=0}^{M-1} s_n \psi_{\gamma_n}.$$

As

$$\forall i \in I, |\langle R^n v, \psi_i \rangle| \leq M_n,$$

and $P_V(\psi_i) = \psi_i, \forall i \in I$, we have:

$$\forall i \in I, |\langle P_V(R^n v), \psi_i \rangle| \leq M_n.$$

Denoting

$$C_0 = C \bigcap Span\{\mathcal{D}\},$$

we know that C_0 is compact and:

$$\rho_0 = \max_{g \in C_0} \|g\|_2 < +\infty.$$

If $|M_n| \leq \tau$, then $P_V(R^n v) \in C_0$, then $d(P_V(R^n v), \tau C_0) = 0$. If $|M_n| > \tau$, since 0 is the inside of τC_0 and $P_V(R^n v) \in C_0$ is outside of τC_0, the segment $[0, P_V(R^n v)]$ must intersect with the border of τC_0. Denote the intersection by

$$w_n^* \triangleq \frac{\tau}{M_n} P_V(R^n v).$$

Then we know that:

$$d(P_V(R^n v), \tau C_0) \leq \|P_V(R^n v) - w_n^*\| \leq \frac{|M_n| - \tau}{\tau} \|w_n^*\| \leq (|M_n| - \tau)\rho_0.$$

In both cases we have:

$$d(P_V(R^n v), \tau C_0) \leq \max(|M_n| - \tau, 0)\rho_0.$$

As $C_0 \subset C$, thus

$$d(P_V(R^n v), \tau C) \leq \max(|M_n| - \tau, 0)\rho_0.$$

Using (7.16), we have:

$$\lim_{n \to +\infty} d(P_V(R^n v), \tau C) = 0.$$

Since $R^n v = P_V(R^n v) + P_W v$, we have:

$$\lim_{n \to +\infty} d(R^n v, P_W v + \tau C) = 0.$$

The remaining part of the theorem is easy to deduced.

□

During the proof Theorem 31 we saw that when $\theta(\cdot)$ is τ-controlled with $\tau > 0$, then,

$$\sum_{n=0}^{\infty} |s_n| < +\infty.$$

Theorem 34 *If the shrinkage function $\theta(\cdot)$ is τ-controlled with $\tau > 0$, then*

$$\sum_{n=0}^{+\infty} |s_n| \leq \frac{\|v\|^2 - |R^{+\infty} v\|^2}{\tau}.$$

Proof. Using (7.13), we know that for any $M \in \mathbb{N}$,

$$\sum_{n=0}^{M-1} s_n^2 \leq \|v\|^2 - |R^M v\|^2$$

From (7.13), we can deduced that:

$$\begin{aligned} \sum_{n=0}^{M-1} 2 s_n M_n &= \|v\|^2 + \sum_{n=0}^{M-1} s_n^2 - \|R^M v\|^2 \\ &\leq 2(\|v\|^2 - |R^M v\|^2). \end{aligned}$$

Thus:

$$\sum_{n=0}^{M-1} s_n M_n \leq \|v\|^2 - |R^M v\|^2$$

Now since s_n, M_n have the same sign, $s_n M_n = |s_n||M_n|$. Since $\theta(\cdot)$ is τ-controlled,

$$s_n M_n \geq \tau |s_n|.$$

Thus we know that, for all $M > 0$,

$$\sum_{n=0}^{M-1} |s_n| \leq \frac{\|v\|^2 - \|R^M v\|^2}{\tau}.$$

Letting M go to infinity, we have,

$$\sum_{n=0}^{+\infty} |s_n| \leq \frac{\|v\|^2 - |R^{+\infty}v\|^2}{\tau}.$$

□

Remark. This theorem tells us that we can control $\sum_{n=0}^{+\infty} |s_n|$ when $\theta(\cdot)$ is τ-controlled. We would like to point out some important facts. Firstly, τ-controlled is not a very strong condition. For example, both soft and hard-threshold satisfy it. Secondly, for all $i \in I$, for all $M > 0$, if we denote,

$$\lambda_i \triangleq \sum_{n=0}^{+\infty} s_{\gamma_n} 1_{\{\gamma_n = i\}},$$

then λ_i exists and

$$\sum_{n=0}^{+\infty} s_n \psi_{\gamma_n} = \sum_{i \in I} \lambda_i \psi_i.$$

Moreover,

$$\sum_{i \in I} |\lambda_i| \leq \sum_{n=0}^{+\infty} |s_n|.$$

It is well-known that the control of l^p-norm for $0 < p \leq 1$ implies sparsity, our theorem gives a theoretical guarantee of sparseness and it is controlled by parameter τ. Generally speaking, the larger τ, the smaller l^1-norm, the sparser the solution. As MP could be regarded as a special case of $\tau = 0$ with soft-threshold, our general MP shrinkage should be more appropriate than MP when searching for a sparse representation of v in the presence of noise. Comparing to the soft/hard wavelet shrinkage, our MP shrinkage can be used with a more general dictionary $\mathcal{D}$.

Now again using Proposition 29, we want to prove that if we deal with a gap shrinkage function, MP shrinkage stops automatically.

Theorem 35 *For any gap shrinkage function $\theta(\cdot)$ with gap $r_0 > 0$, then the MP shrinkage algorithm will stop after at most*

$$M = \lfloor \frac{\|v\|^2}{r_0^2} \rfloor$$

iterations, where $\lfloor \cdot \rfloor$ denotes the floor function.

Proof. If the MP shrinkage algorithm does not stop after $M = \lfloor \frac{\|v\|^2}{\tau^2} \rfloor$ iterations, that means that for all $n = 0, \ldots, M$, we have:

$$s_n^2 + 2s_n(M_n - s_n) \geq r_0^2,$$

where $M_n = \langle R^n v, \psi_{\gamma_n} \rangle$, $s_n = \theta(M_n)$ and γ_0 is the gap of $\theta(\cdot)$.
From (7.12), we know that:

$$\|v\|^2 \geq \sum_{n=0}^{M} \Big(s_n^2 + 2s_n(M_n - s_n) \Big) \geq (M+1) r_0^2.$$

Thus

$$M \leq \frac{\|v\|^2}{r_0^2} - 1.$$

Since M is an integer, we have:

$$M \leq \lfloor \frac{\|v\|^2}{r_0^2} \rfloor - 1.$$

This is contradict to $M = \lfloor \frac{\|v\|^2}{r_0^2} \rfloor$.

□

Since τ-control ensures the convergence of the MP shrinkage, from now on, we assume that this condition holds. Analyzing on the MP shrinkage algorithm, we know that when $\theta(\cdot)$ is τ-controlled, then for any $v \in \mathcal{H}$, the result $\sum_{n=0}^{+\infty} s_n \psi_{\gamma_n}$ after MP shrinkage exists and is unique (to be rigorous, we need define how to chose γ_n when $\gamma_n = \arg\max_{i \in I} |\langle R^n v, \psi_i \rangle|$ is not unique, normally we chose smallest allowed γ_n. Since this does not affect our conclusions, we neglect this detail), so $\sum_{n=0}^{+\infty} s_n \psi_{\gamma_n}$ can be regarded as a function of $v \in \mathcal{H}$. We call this function as MP shrinkage function/operator and denote it by

$$\mathcal{M}_\tau(v) = \sum_{n=0}^{+\infty} s_n \psi_{\gamma_n}. \tag{7.17}$$

Recall that V is the closed linear span of the vectors in $\mathcal{D}$, and W the orthogonal complement of V in $\mathcal{H}$. The orthogonal projectors over V and W are respectively written as P_V and P_W.

Theorem 36 *If $\theta(\cdot)$ is strictly τ-controlled, then for any $v \in \mathcal{H}$ fixed, when $\tau \to 0^+$, $\mathcal{M}_\tau(v)$ converges to the projection $P_V(v)$. More precisely, we have:*

$$\mathcal{M}_\tau(v) = P_V(v) + O(\tau) \quad (\textit{when } \tau \to 0^+) \tag{7.18}$$

Proof. From Theorem 33, we know that:

$$\mathcal{M}_\tau(v) - P_V(v) \in \tau C \bigcap V,$$

where

$$C = \{g \in \mathcal{H} | |\langle g, \psi \rangle| \leq 1, \forall \psi \in \mathcal{D}\}.$$

Since $C \bigcap V$ is compact (see Proposition 6), there exists a $\rho > 0$ such that for all $g \in C \bigcap V$, $\|g\| \leq \rho$.

Hence,

$$\|\mathcal{M}_\tau(v) - P_V(v)\|_2 \leq \rho\tau.$$

□

7.3 Experiment

In this section, we want to compare the performance of MP with MP shrinkage. We want to approximate a noisy image v shown as Figure 7.1 (right side) which is a noisy version (Gaussian white noise of standard variation 20) of Figure 7.1 (left side). The size of both images are 128×128. The $PSNR$ of the noisy image is 22.2215.

Figure 7.1: Test image of MP shrinkage: left: clean image; right: noisy image of Gaussian white noise of variation 20, this is the image v that we used in MP shrinkage

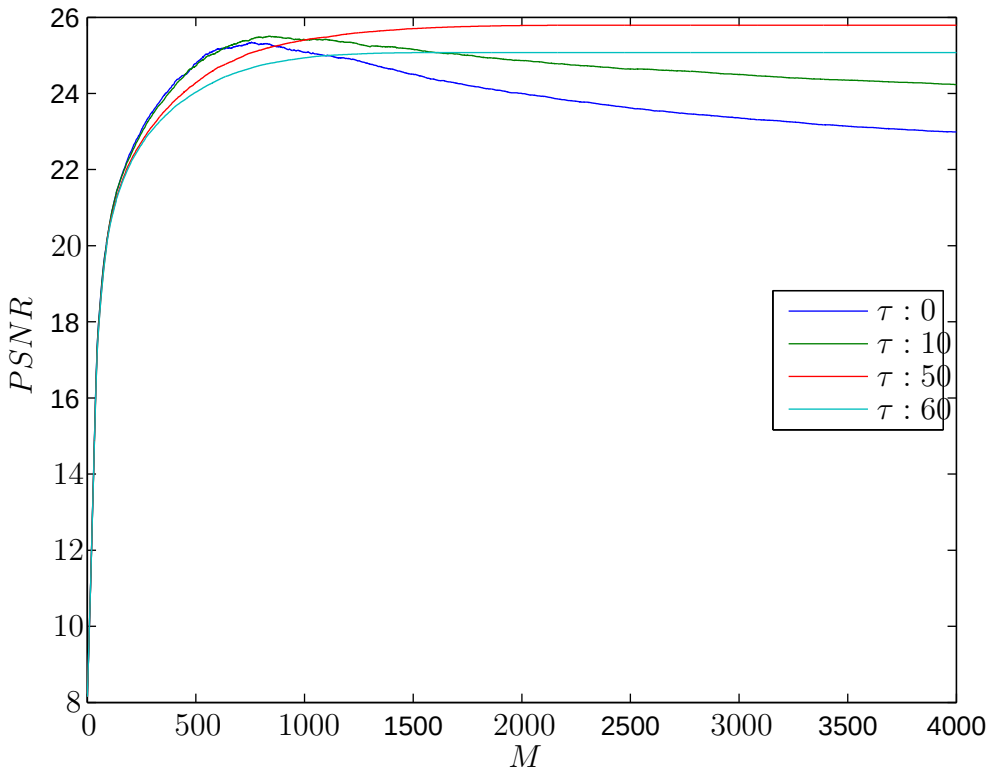

Figure 7.2: $PSNR$ for the M terms reconstructed image of MP shrinkage, with soft-threshold function on various τ: blue line: τ=0; other lines: $\tau = 10, 50, 60$.

The feature dictionary we used is still the 13 filters of Daubechies-3 wavelet of level 4. The image of these 13 filters are shown in Figure 3.6.

We use these features to construct the translation-invariant dictionary. We test various τ for MP shrinkage with soft-threshold function. When $\tau = 0$, this method is just MP.

Figure 7.2 shows the $PSNR$ between

$$\sum_{n=0}^{M-1} s_n \psi_{\gamma_n}, \tag{7.19}$$

and the clean image for $M = 1$ to 4000. Beware that in (7.19), both s_n, γ_n depend on τ.

Figure 7.2 shows that when M is small (say $M < 200$), all the τ values perform similarly. When M is greater than 1000, then it seems that MP shrinkage with τ positive is better than MP. In this experiment, for our case, $\tau = 50$ provides best result.

7.4 Conclusion

This chapter contains the study of a variant of Matching Pursuit. In this variant, we proposed to reduce the scalar product with the element best correlated with the residue, before modifying the residue. This is for a general threshold function. By using simple properties of these threshold functions, we showed that the algorithm thus obtained converges towards the orthogonal projection of the data on linear space generated by the dictionary (the whole modulo an approximation quantified by the characteristics of the threshold function). Finally, under a weak assumption on the threshold function (for example the hard-threshold satisfies this assumption), this algorithm converges in a finite time which one can deduce from the properties of the threshold function. Typically, this algorithm might be useful to make the orthogonal projections in the algorithm Orthogonal Matching Pursuit. This we have not done yet.

Appendix

Proof of Theorem 31, Theorem 33 for $\tau = 0$

This section gives an adaptation to MP of Jones's proof (see [64]) for the convergence of projection pursuit regressions and the proof of Theorem 1 of [32]. Throughout this proof, we suppose that the shrinkage function $\theta(\cdot)$ for the MP shrinkage is 0-controlled with c_0 (see Eq.(7.4)).

Lemma 37 *Let $h_n = s_n \psi_{\gamma_n}$. For any $n \geq 0$ and $m \geq 0$*

$$|\langle h_m, R^n v \rangle| \leq \frac{1}{c_0} \|h_m\| \|h_n\|$$

.

Proof. Using Eq.(7.4), we have:

$$\begin{aligned}
|\langle h_m, R^n v\rangle| &= |s_m| \cdot |\langle \psi_{\gamma_m}, R^n v\rangle| \\
&\leq |s_m| \cdot |\langle \psi_{\gamma_n}, R^n v\rangle| \\
&\leq |s_m| \cdot \frac{1}{c_0} |\theta(\langle \psi_{\gamma_n}, R^n v\rangle)| \\
&= \frac{1}{c_0} |s_m| \cdot |s_n| \\
&= \frac{1}{c_0} \|h_m\| \|h_n\|.
\end{aligned}$$

□

Lemma 38 *If $(x_n)_{n\in\mathbb{N}}$ is a positive sequence such that $\sum_{n=0}^{+\infty} x_n^2 < +\infty$, then*

$$\liminf_{n\to+\infty} x_n \sum_{k=0}^{n} x_k = 0.$$

Proof. For any $\epsilon > 0$, we choose n such that $\sum_{k=n+1}^{+\infty} x_n^2 \leq \frac{\epsilon}{2}$. Since $\lim_{k\to\infty} x_k = 0$, we can choose $k > n+1$ large enough such that $x_k \sum_{i=0}^{n} x_i \leq \frac{\epsilon}{2}$. Let x_j be the smallest element in $\{x_{n+1}, \ldots, x_k\}$. Then

$$\begin{aligned}
x_j \sum_{k=0}^{j} x_k &= x_j \sum_{k=0}^{n} x_k + x_j \sum_{k=n+1}^{j} x_k \\
&\leq \frac{\epsilon}{2} + \sum_{k=n+1}^{j} x_k^2 \\
&\leq \epsilon.
\end{aligned}$$

□

To prove Theorem 31 for $\tau = 0$, we prove that the sequence $(R^n v)_{n\in\mathbb{N}}$ is a Cauchy sequence. Let $N_1 > N_2 \geq 0$. Beware that for all $w_1, w_2 \in \mathcal{H}$, we have:

$$\|w_1 - w_2\|^2 = \|w_1\|^2 - \|w_2\|^2 - 2\langle w_2, w_1 - w_2\rangle \leq \|w_1\|^2 - \|w_2\|^2 + 2|\langle w_2, w_1 - w_2\rangle|.$$

This fact and Lemma 37 imply that:

$$\begin{aligned}
\|R^{N_1} v - R^{N_2} v\|^2 &\leq \|R^{N_1} v\|^2 - \|R^{N_2} v\|^2 + 2|\langle R^{N_2} v, \sum_{n=N_1}^{N_2-1} h_n\rangle| \\
&\leq \|R^{N_1} v\|^2 - \|R^{N_2} v\|^2 + \frac{2}{c_0} \|h_{N_2}\| \sum_{n=N_1}^{N_2-1} \|h_n\|. \qquad (7.20)
\end{aligned}$$

Using (7.12) and Proposition 29, we know that the sequence $(\|R^n v\|)_{n\in\mathbb{N}}$ is monotonically decreasing and thus converges to some value R_∞. Let $\epsilon > 0$, there exits $K > 0$ such that for all $m > K$, $\|R^m v\|^2 \leq R_\infty^2 + \epsilon^2$. Let $p > 0$. We want to estimate $\|R^m v - R^{m+p} v\|$,

for $m > K$. Using (7.14), we know that $\sum_{n=0}^{+\infty} \|h_n\|^2 = \sum_{n=0}^{+\infty} s_n^2 < +\infty$. Hence Lemma 38 implies that there exists $q > m + p$ such that

$$\|h_q\| \sum_{n=0}^{q} \|h_n\| \leq \epsilon^2.$$

We can decompose

$$\|R^m v - R^{m+p} v\| \leq \|R^m v - R^q v\| + \|R^{m+p} v - R^q v\|.$$

Eq.(7.20) for $N_1 = m$ and $N_2 = q$ implies

$$\|R^m v - R^q v\|^2 \leq \epsilon^2 + \frac{2}{c_0}\epsilon^2.$$

Similarly,

$$\|R^{m+p} v - R^q v\|^2 \leq \epsilon^2 + \frac{2}{c_0}\epsilon^2.$$

Hence,

$$\|R^m v - R^{m+p} v\| \leq \epsilon\sqrt{2(1 + 2/c_0)},$$

which proves that $(R^n v)_{n\in\mathbb{N}}$ is a Cauchy sequence. This completes the proof of Theorem 31 for $\tau = 0$.

In order to prove the Theorem 33 for $\tau = 0$, we denote

$$R^{+\infty} v = \lim_{n\to+\infty} R^n v.$$

As $\lim_{n\to+\infty} |s_n| = 0$ (using Proposition 29), and $|s_n| = |\theta(\langle R^n v, \psi_{\gamma_n}\rangle)| \geq c_0 |\langle R^n v, \psi_{\gamma_n}\rangle|$ ($\theta(\cdot)$ is 0-controlled), we have:

$$\lim_{n\to+\infty} |\langle R^n v, \psi_{\gamma_n}\rangle| = 0.$$

Hence, for all $i \in I$, we have:

$$\lim_{n\to+\infty} |\langle R^n v, \psi_i\rangle| = 0.$$

Thus $|\langle R^{+\infty} v, \psi_i\rangle| = 0, \forall i \in I$. This implies that $R^{+\infty} v \in W$. Since,

$$v = \sum_{n=0}^{+\infty} s_n \psi_{\gamma_n} + R^{+\infty} v$$

and $\sum_{n=0}^{+\infty} s_n \psi_{\gamma_n} \in V$, we derive that

$$P_V v = \sum_{n=0}^{+\infty} s_n \psi_{\gamma_n},$$

and

$$P_W v = R^{+\infty} v.$$

This completes the proof of Theorem 33 for $\tau = 0$.

Chapter 8

Statistical approach for dictionary learning

In this chapter, we focus on the dictionary learning problem by itself. We do not follow our previous variational route but for a while, we consider a more probabilistic modeling point of view based on probabilistic generative models. Introducing simple additive stochastic image models with Bernoulli on/off flags, we cast the dictionary learning problem into a rigorous statistical framework. This allows a principle derivation of a new dictionary learning algorithm based on maximum likelihood and mean field approximation algorithms.

8.1 A simple probabilistic generative model

Denote $\mathcal{Y} = \mathbb{R}^{\Lambda}$ the space of digital image (more precisely, in this chapter, $\mathcal{Y}$ would be a space of small size image). In the following, the support Λ will always be the discrete tore $(\mathbb{Z}/n\mathbb{Z})^2$ as we will consider the translation over the plan.

The point of view adopted here is the framework of generative models. This makes it possible to state the dictionary learning problem as a statistical parameters estimation problem. According to this regard, we represent the database as a realization of a family of i.i.d random variables whose probability distribution is defined through a generative mechanism. This mechanism depends on a parameter $\theta \in \Theta$ which will be estimated by maximum likelihood estimation method.

Let us begin with the case of a single observed image defined as the realization $y = Y(\omega)$ of the random variable $Y : \Omega \to \mathcal{Y}$. To define the distribution of Y, we introduce a dictionary of basic patterns, which is represented by a family $\Phi = (\phi_h)_{1 \leq h \leq q}$ of elements in $\mathcal{Y}$. To obtain a dictionary invariant by translation, for all $s \in \Lambda$, we denote $\phi_{h,s}$ the translation of ϕ_h by s, i.e.

$$\phi_{h,s}(t) = \phi_h(t - s)\,.$$

The basic model for Y is an additive model defined by:

$$Y \triangleq \sum_{h,s} B_{h,s} X_{h,s} \phi_{h,s} + \sigma\epsilon, \tag{8.1}$$

where:

- $B = (B_{h,s})_{1 \leq h \leq q, s \in \Lambda}$ is a family of independent Bernoulli variables acting as on/off flags activating or deactivating a pattern $\phi_{h,s}$ in the dictionary, such that for each

h, $(B_{h,s})_{s\in\Lambda}$ is i.i.d and distributed according to $\mathcal{B}_{p_h}$, the Bernoulli distribution with parameter p_h;

- $X = (X_{h,s})_{1\leq h\leq q, s\in\Lambda}$ is a family of independent random coefficients, such that for each h, $(X_{h,s})_{s\in\Lambda}$ is i.i.d;
- ϵ is a additive independent white noise i.e. $\epsilon \sim \mathcal{N}(0, \mathrm{Id}_\Lambda)$.

From this generic probabilistic structure, one can derive several specific statistical models depending on the chosen distributions for X. We will investigate in this chapter basically two different models.

8.1.1 The Bernoulli-Exponential model (BEM)

For this model, we assume that the $(X_{h,s})_{s\in\Lambda}$ are distributed according to the exponential distribution $\mathcal{E}_{\lambda_h}$ with parameter λ_h. The parameters of the model are defined by:

$$\theta = (\Phi, (\lambda_h)_{1\leq h\leq q}, (p_h)_{1\leq h\leq q}, \sigma, q), \tag{8.2}$$

which we seek to estimate. To avoid trivial problems of identifiability of the model, we suppose that the elements of the dictionary are normalized to 1: $\forall 1 \leq h \leq q, \|\phi_h\|_2 = 1$. (In fact, since $uX_{h,s} \sim \mathcal{E}_{\lambda_h/u}$, the change $(\lambda_h, \phi_h) \to (\lambda_h/u, \phi_h/u)$ does not change the result on Y). Hence, throughout the chapter, the parameter space Θ_{BEM} for the Bernoulli-Exponential model is defined as:

$$\Theta_{\mathrm{BEM}} = \{(\Phi, (\lambda_h)_{1\leq h\leq q}, (p_h)_{1\leq h\leq q}, \sigma, q) | q \in \mathbb{N}_*, \\ \forall 1 \leq h \leq q, \|\phi_h\| = 1, \lambda_h > 0, 0 < p_h < 1 \text{ and } \sigma > 0\}.$$

Now, using the hidden variables $B_{h,s}, X_{h,s}$, we can write the complete likelihood:

$$L_\theta(X, B, Y) = e^{-|Y-DZ|^2/(2\sigma^2)}(2\pi\sigma^2)^{-|\Lambda|/2} \prod_{h,s} p_h^{B_{h,s}}(1-p_h)^{1-B_{h,s}} \lambda_h e^{-\lambda_h X_{h,s}}, \tag{8.3}$$

where $Z = (X_{h,s}B_{h,s})_{1\leq h\leq q, s\in\Lambda}$ and D is the matrix obtained by concatenation of the column vectors formed by the elements of the dictionary. The log-likelihood can be written as:

$$l_\theta(X, B, Y) = -\frac{|Y-DZ|^2}{2\sigma^2} - \sum_{h,s} [\lambda_h X_{h,s} + r_h B_{h,s}] + C_\theta, \tag{8.4}$$

where $r_h = \log(\frac{1-p_h}{p_h})$ and $C_\theta \triangleq |\Lambda| [-\log(2\pi\sigma^2)/2 + \log(\lambda_h) + \log(1-p_h)]$.

We remark that the minimization of $-l_\theta(X, B, Y)$ on X, B leads to a variational problem which combines the term L^2 and the mixed regularization of L^1 and L^0:

$$\begin{cases} \min \frac{|Y-DZ|^2}{2\sigma^2} + \sum_{h,s} [\lambda_h X_{h,s} + r_h B_{h,s}] \\ \\ X_{h,s} \geq 0, \ B_{h,s} \in \{0, 1\} \ \text{ for all } h, s. \end{cases} \tag{8.5}$$

Remark: We remark here, by this probabilistic interpretation, that the parameters λ_h at p_h associated respectively with the L^1 and L^0 regularization terms have very different significance: the parameter p_h stands for the appearance probability for a pixel of a

given element of dictionary while λ_h controls the intensity of the multiplicative coefficient applied to this element (in fact the reverse of this intensity since $E(X_{h,s}) = 1/\lambda_h$) *conditionally with the fact that this element is active*. A model with a regularization purely L^1 as for basis pursuit, makes the parameter λ_h play a double role in only one parameter. From a statistical modeling point of view, this is problematic: when one element of a dictionary rarely appears but with a strong intensity or when an element often appears but with a low intensity we are brought to choose the same value of λ_h whereas these are two situations very different. In some sense, a purely L^1 regularization is intrinsically ambiguous.

If we no longer have only one image but a finite family $Y \triangleq (Y^1, \cdots, Y^N)$, we choose to extend the model by independence. By denoting $X \triangleq (X^1, \cdots, X^N)$ and $B \triangleq (B^1, \cdots, B^N)$, we obtain the log-likelihood

$$l_{N,\theta}(X, B, Y) = \sum_{k=1}^{N} \{-\frac{|Y^k - DZ^k|^2}{2\sigma^2} - \sum_{h,s} \left[\lambda_h X_{h,s}^k + r_h B_{h,s}^k\right] + C_\theta\}. \tag{8.6}$$

8.1.2 The Bernoulli-Gaussian model (BGM)

In BEM, the random coefficients are constrained to positive values. In the following second model, we relax this constraint and we assume that the $(X_{h,s})_{s\in\Lambda}$ are distributed according to a common gaussian distribution $\mathcal{N}(0, \sigma_h^2)$. Here, the parameter θ becomes $\theta = (\Phi, (\sigma_h^2)_{1\leq h\leq q}, (p_h)_{1\leq h\leq q}, \sigma^2) \in \Theta_{\text{BGM}}$ with new parameter space:

$$\Theta_{\text{BGM}} = \{(\Phi, (\sigma_h^2)_{1\leq h\leq q}, (p_h)_{1\leq h\leq q}, \sigma^2, q) \mid q \in \mathbb{N}_*, \\ \forall 1 \leq h \leq q, \|\phi_h\|_2 = 1, \sigma_h > 0, 0 < p_h < 1 \text{ and } \sigma > 0\}.$$

We compute, with the same notation as for the BEM, the log-likelihood for a finite family $Y \triangleq (Y^1, \cdots, Y^N)$:

$$l_{N,\theta}(X, B, Y) = \sum_{k=1}^{N} \{-\frac{|Y^k - DZ^k|^2}{2\sigma^2} - \sum_{h,s} \left[\frac{1}{2}\left(\frac{X_{h,s}^k}{\sigma_h}\right)^2 + r_h B_{h,s}^k\right] + C_\theta\}, \tag{8.7}$$

where $r_h = \log(\frac{1-p_h}{p_h})$ and $C_\theta \triangleq |\Lambda|\,[-\log(2\pi\sigma) + \log(1 - p_h)]$.

One can notice that the maximization of the log-likelihood leads to a variational $L^2 - L^0$ problem so that the Bernoulli-Exponential model and the Bernoulli-Gaussian model covers two easily interpretable variational setting.

8.2 Identifiability issues

The first interesting statistical question is to address the identifiability issues of our models i.e. can we distinguish the distribution on Y given by two different parameters θ and $\theta' \in \Theta$? We will give a positive answer under weak conditions in both cases. This part is slightly technical and the reader can skip it at the first reading and come back later if interested.

8.2.1 Identifiability of the BEM

Obviously, a permutation on the indexes h does not change the overall distribution so that we should define an appropriate equivalence relation on Θ_{BEM}. Two parameters

$\theta, \theta' \in \Theta_{\text{BEM}}$ are said to be equivalent if and only if (denoted by $\theta \sim \theta'$):

$$q = q'$$

and there exits a permutation π on $\{1, \ldots, q\} \times \Lambda$ such that $\forall (h, s)$,

$$\phi_{h,s} = \phi'_{h',s'}, \lambda_h = \lambda'_h, p_h = p'_h, \sigma = \sigma',$$

where $(h', s') = \pi(h, s)$.

Proposition 39 *If $(\phi_{h,s})_{1 \le h \le q, s \in \Lambda}$ are different from each other, then the model (8.1) is identifiable on $\Theta_{BEM}/\sim$.*

Proof. We only need to prove the identifiability in the more general case

$$Y = \sum_h B_h X_h \phi_h + \sigma \epsilon, \tag{8.8}$$

where B_h is of $\mathcal{B}_{p_h}$, X_h is of $\mathcal{E}_{\lambda_h}$, ϵ is of $\mathcal{N}(0, \mathrm{Id})$ and all ϕ_h are different from each other.

The characteristic function of Eq.(8.8) is:

$$\begin{aligned} \phi_Y(\xi) &\triangleq E(e^{i\langle \xi, Y \rangle}) \\ &= \Pi_h E(e^{iB_h X_h \langle \xi, \phi_h \rangle}) \\ &= \Pi_h [p_h (1 - \frac{i\langle \xi, \phi_h \rangle}{\lambda_h})^{-1} + (1 - p_h)] \cdot e^{-\frac{\sigma^2}{2}|\xi|^2}. \end{aligned}$$

As the convergence of $e^{-\frac{\sigma^2}{2}|\xi|^2}$ is faster than any polynomial, let $\xi \to +\infty$, we can determine σ uniquely. In fact, suppose that there are two decompositions satisfying:

$$\Pi_{h=1}^{q} [p_h (1 - \frac{i\langle \xi, \phi_h \rangle}{\lambda_h})^{-1} + (1 - p_h)] \cdot e^{-\frac{\sigma^2}{2}|\xi|^2} = \Pi_{h=1}^{q'} [p'_h (1 - \frac{i\langle \xi, \phi'_h \rangle}{\lambda'_h})^{-1} + (1 - p'_h)] \cdot e^{-\frac{\sigma'^2}{2}|\xi|^2}. \tag{8.9}$$

If $\sigma' \neq \sigma$, then without loss of generality, we can assume that $\sigma' < \sigma$. Rewriting (8.9), we have:

$$e^{\frac{(\sigma^2 - \sigma'^2)}{2}|\xi|^2} = \frac{\Pi_{h=1}^{q} [p_h (1 - \frac{i\langle \xi, \phi_h \rangle}{\lambda_h})^{-1} + (1 - p_h)]}{\Pi_{h=1}^{q'} [p'_h (1 - \frac{i\langle \xi, \phi'_h \rangle}{\lambda'_h})^{-1} + (1 - p'_h)]}. \tag{8.10}$$

Let $\xi \to +\infty$ for both sides of (8.10), the right side is finite while the left side is infinite. Thus we must have $\sigma = \sigma'$.

Now we can only consider,

$$\begin{aligned} \widetilde{\phi_Y}(\xi) &\triangleq \phi_Y(-i\xi) \cdot e^{\frac{\sigma^2}{2}|\xi|^2} \\ &= \Pi_{h=1}^{q} [p_h (1 - \frac{\langle \xi, \phi_h \rangle}{\lambda_h})^{-1} + (1 - p_h)] \\ &= \Pi_{h=1}^{q} (1 - p_h) \Pi_{h=1}^{q} \frac{\langle \xi, \phi_h \rangle - \frac{\lambda_h}{1 - p_h}}{\langle \xi, \phi_h \rangle - \lambda_h}. \end{aligned}$$

Suppose that there are two compositions such that:

$$\Pi_{h=1}^{q}(1-p_h)\Pi_{h=1}^{q}\frac{\langle\xi,\phi_h\rangle-\frac{\lambda_h}{1-p_h}}{\langle\xi,\phi_h\rangle-\lambda_h}=\Pi_{h=1}^{q}(1-p'_h)\Pi_{h=1}^{q'}\frac{\langle\xi,\phi'_h\rangle-\frac{\lambda'_h}{1-p'_h}}{\langle\xi,\phi'_h\rangle-\lambda'_h}.$$

Rewriting the above equation, we have:

$$\begin{aligned}\Pi_{h=1}^{q}(1-p_h)\Pi_{h=1}^{q}[\langle\xi,\phi_h\rangle-\frac{\lambda_h}{1-p_h}]\Pi_{h=1}^{q'}[\langle\xi,\phi'_h\rangle-\lambda'_h]\\ =\Pi_{h=1}^{q}(1-p'_h)\Pi_{h=1}^{q}[\langle\xi,\phi_h\rangle-\lambda_h]\Pi_{h=1}^{q'}[\langle\xi,\phi'_h\rangle-\frac{\lambda'_h}{1-p'_h}]\end{aligned}\quad(8.11)$$

Notice that both sides of (8.11) can be regarded as non-zero elements of the multivariable polynomial ring $\mathbb{C}[\xi]\triangleq\mathbb{C}[\xi_1,\ldots,\xi_{|\Lambda|}]$. Since $\mathbb{C}$ is a field, $\mathbb{C}[\xi]$ is a unique factorization domain and obviously, the polynomial $(\langle\xi,\phi_h\rangle-A)$ with $\|\psi\|\neq 0$ and A constant is an irreducible element in $\mathbb{C}[\xi]$, Eq.(8.11) should be equal term by term except a constant multiplier. Moreover, all these multipliers are equal to 1. In fact, suppose that

$$\langle\xi,\psi_1\rangle-A_1=c_0(\langle\xi,\psi_2\rangle-A_2)\quad(8.12)$$

where ψ_1,ψ_2 could be any member of

$$\{\phi_1,\ldots,\phi_q,\phi'_1,\ldots,\phi'_{q'}\},$$

and A_1,A_2 could be any one of

$$\{(\lambda_h)_{1\leq h\leq q},(\frac{\lambda_h}{1-p_h})_{1\leq h\leq q},(\lambda'_h)_{1\leq h\leq q'},(\frac{\lambda'_h}{1-p'_h})_{1\leq h\leq q'}\}.$$

Comparing the coefficients of (8.12), we know that:

$$\psi_1=c_0\psi_2,A_1=c_0A_2.$$

Since $A_1,A_2>0$, we must have $c_0>0$. Moreover, $\|\psi_1\|^2=c_0^2\|\psi_2\|^2$. So $c_0=1$ as $\|\psi_1\|=\|\psi_2\|=1$. Hence, $\psi_1=\psi_2$.

Now we will prove the Lemma by induction on $\max(q,q')$.

When $\max(q,q')=1$, observing (8.11), we know that $q=q'=1$. The only possibility for (8.11) to hold is that:

$$\langle\xi,\phi_1\rangle-\frac{\lambda_1}{1-p_1}=\langle\xi,\phi'_1\rangle-\frac{\lambda'_1}{1-p'_1},$$

and

$$\langle\xi,\phi_1\rangle-\lambda_1=\langle\xi,\phi'_1\rangle-\lambda'_1.$$

Hence

$$\phi_1=\phi'_1,\lambda_1=\lambda'_1,p_1=p'_1.$$

The lemma is true for this situation.

Suppose that for $\max(q,q')-1$, the lemma is true.

For $\max(q,q')$, consider ϕ_q. As all the $\phi_h(1\leq h\leq q-1)$ are different with ϕ_q and $\frac{\lambda_q}{1-p_q}\neq\lambda_q$), the only possible term of the right side of Eq.(8.11) corresponding to $\langle\xi,\phi_q\rangle-\frac{\lambda_q}{1-p_q}$ should be from $\phi'_1,\ldots,\phi'_q$. Say it is:

$$\langle\xi,\phi_q\rangle-\frac{\lambda_q}{1-p_q}=\langle\xi,\phi'_q\rangle-\frac{\lambda'_q}{1-p'_q}.$$

Hence,

$$\phi_q = \phi'_q, \frac{\lambda_q}{1-p_q} = \frac{\lambda'_q}{1-p'_q}.$$

Now considering $\langle \xi, \phi'_q \rangle - \lambda'_q$ and $\langle \xi, \phi_q \rangle - \lambda_q$ of Eq.(8.11), these two terms have to be the same as there is no possibility to provide this opportunity from $\{\phi_1, \ldots, \phi_q\}$ or $\{\phi'_1, \ldots, \phi'_q\}$ (in other case, this will lead to some $h, 1 \leq h < q$ such that $\phi_h = \phi_q$ or $\phi'_h = \phi'_q$). Hence:

$$\langle \xi, \phi'_q \rangle - \lambda'_q = \langle \xi, \phi_q \rangle - \lambda_q.$$

Thus

$$\phi_q = \phi'_q, \lambda_q = \lambda'_q, \phi_q = \phi'_q.$$

Dividing $[\langle \xi, \phi_q \rangle - \frac{\lambda_q}{1-p_q}][\langle \xi, \phi_q \rangle - \lambda_q]$ from both sides of Eq.(8.11) leads to situation of $\max(q, q') - 1$. By induction, the lemma is true for any q, q'. □

Remark When ϕ is degenerate, then this model is non-identifiable. In fact, a simple counterexample is: $q = 2$, $\phi_1 = \phi_2$, $(\lambda_1, p_1) = (\frac{1}{4}, \frac{1}{2}), (\lambda_2, p_2) = (\frac{1}{6}, \frac{1}{2})$, $\sigma = 1$. Then $(\lambda_1, p_1) = (\frac{1}{4}, \frac{1}{4}), (\lambda_2, p_2) = (\frac{1}{6}, \frac{2}{3})$, $\sigma = 1$ generate the same Y. Indeed, in both situations, their characteristic functions of Y are:

$$\frac{1}{4} \cdot \frac{i\langle \xi, \psi_1 \rangle - \frac{1}{2}}{i\langle \xi, \psi_1 \rangle - \frac{1}{4}} \cdot \frac{i\langle \xi, \psi_1 \rangle - \frac{1}{3}}{i\langle \xi, \psi_1 \rangle - \frac{1}{6}} \cdot e^{-\frac{1}{2}|\xi|^2}.$$

As random vectors are equal in distribution if and only if their characteristic functions are equal (see lemma 2.15 of [65]), both cases generate the same Y.

Remark When ϕ is degenerate, then the collection of all the possible patterns is still identifiable i.e. the following set is uniquely determined by Y:

$$\{\phi_1, \ldots, \phi_q\}.$$

In fact, in this case, Eq.(8.11) still holds. Hence for any $\phi \in \{\phi_1, \ldots, \phi_q\}$ fixed, taking out all the term relative to ϕ, we have:

$$\Pi_{h:\phi_h=\phi} \frac{\langle \xi, \phi_h \rangle - \frac{\lambda_h}{1-p_h}}{\langle \xi, \phi_h \rangle - \lambda_h} = \Pi_{h:\phi'_h=\phi} \frac{\langle \xi, \phi'_h \rangle - \frac{\lambda'_h}{1-p'_h}}{\langle \xi, \phi'_h \rangle - \lambda'_h}. \tag{8.13}$$

As

$$\Pi_{h:\phi_h=\phi} \frac{\frac{\lambda_h}{1-p_h}}{\lambda_h} = \Pi_{h:\phi_h=\phi} \frac{1}{1-p_h} > 1,$$

the left side of (8.13) will not be reduced to a constant. Thus the right side of (8.13) must have some term i.e. $\{h | \phi'_h = \phi\}$ is non-empty, thus $\phi \in \{\phi'_1, \ldots, \phi'_{q'}\}$. When ϕ passes over all the elements of $\{\phi_1, \ldots, \phi_q\}$, we have:

$$\{\phi_1, \ldots, \phi_q\} \subset \{\phi'_1, \ldots, \phi'_{q'}\}.$$

And similarly for the other direction. Hence,

$$\{\phi_1, \ldots, \phi_q\} = \{\phi'_1, \ldots, \phi'_{q'}\}.$$

8.2.2 Identifiability of the BGM

In this section, we will prove that the BGM is also identifiable under a similarly equivalence relation in Θ_{BGM}. The two elements of Θ_{BGM} have a equivalence relation (also denoting by $\sim$):

$$(\Phi, (\sigma_h^2)_{1\leq h\leq q}, (p_h)_{1\leq h\leq q}, \sigma^2, q) \sim (\Phi', (\sigma'^2_h)_{1\leq h\leq q'}, (p'_h)_{1\leq h\leq q'}, \sigma'^2, q')$$

if and only if $q = q'$ and there exits a permutation π on $\{1, \ldots, q\} \times \Lambda$, a family $\varepsilon = (\varepsilon_h)_{1\leq h\leq q} \in \{-1, 1\}^q$ such that:

$$\phi_{h,s} = \varepsilon_h \phi'_{h',s'}, \sigma_h^2 = \sigma'^2_{h'}, p_h = p'_{h'}, \sigma^2 = \sigma'^2$$

where $(h', s') = \pi(h, s)$.

Whenever the dictionary $(\phi_h)_{1\leq h\leq q}$ is degenerate or non-degenerate, this new model is always identifiable. We need a lemma.

Lemma 40 *Suppose that*

$$\Pi_{h=1}^q \left(p_h e^{a_h z} + (1 - p_h)\right) = \Pi_{h=1}^{q'} \left(p'_h e^{a'_h z} + (1 - p'_h)\right), \forall z \in (0, 1), \tag{8.14}$$

where q, q' are positive integers, $a_h > 0, 0 < p_h < 1, \forall 1 \leq h \leq q$ and $a'_h > 0, 0 < p'_h < 1, \forall 1 \leq h \leq q'$. Then $q = q'$ and except a permutation we have:

$$a_h = a'_h, p_h = p'_h, \forall 1 \leq h \leq q.$$

Proof. Both sides of (8.14) are entire functions, thus extending analytically to the total plan $z \in \mathbb{C}$, we have:

$$\Pi_{h=1}^q [p_h e^{a_h z} + (1 - p_h)] = \Pi_{h=1}^{q'} [p'_h e^{a'_h z} + (1 - p'_h)], \forall z \in \mathbb{C}. \tag{8.15}$$

We prove the lemma by induction on $\max(q, q')$.
When $\max(q, q') = 1$, Obviously, the lemma is true.
Suppose that for $\max(q, q') - 1$, the lemma is true. For $\max(q, q')$, without loss of generality, we can assume that:

$$a_q = \max\{a_1, \ldots, a_q, a'_1, \ldots, a'_{q'}\}.$$

Note that $a_q > 0$ and if we denote $i = \sqrt{-1}$, then we know that

$$z = \frac{1}{a_q}\left(\log\frac{1 - p_q}{p_q} + \pi i\right),$$

is a complex root of the left side of (8.15). So it exits a $h, 1 \leq h \leq q'$ such that:

$$\frac{1}{a_q}\left(\log\frac{1 - p_q}{p_q} + \pi i\right) = \frac{1}{a'_h}\left(\log\frac{1 - p'_h}{p'_h} + (2k + 1)\pi i\right), \tag{8.16}$$

where k is some integer. Comparing the real part of (8.16), we have:

$$\frac{1}{a_q}\log\frac{1 - p_q}{p_q} = \frac{1}{a'_h}\log\frac{1 - p'_h}{p'_h}.$$

Comparing the imaginary part of (8.16) we know that:

$$\frac{1}{a_q} = (2k + 1) \cdot \frac{1}{a'_h}.$$

Thus k is non-negative and

$$\frac{1}{a_q} = (2k+1) \cdot \frac{1}{a'_h} \geq \frac{1}{a'_h} \geq \frac{1}{a_q}.$$

Hence, $a_q = a'_h$. Re-comparing the real part of (8.16) we know that:

$$p_q = p'_h.$$

Dividing both sides of (8.15) by $(p_q e^{a_q z} + (1 - p_q))$ leads to the situation of $\max(q, q') - 1$. By induction we know that the lemma is always true for any q, q'.

□

Proposition 41 *The model* (8.1) *is identifiable on* $\theta = (\Phi, (\sigma_h^2)_{1 \leq h \leq q}, (p_h)_{1 \leq h \leq q}, \sigma^2)$ *with* $(B_{h,s})_{1 \leq h \leq q, s \in \Lambda}$ *of* $\mathcal{B}_{p_h}$, $(X_{h,s})_{1 \leq h \leq q, s \in \Lambda}$ *of* $\mathcal{N}(0, \sigma_h^2)$ *and* ϵ *of* $\mathcal{N}(0, Id)$ *in the sense of* $\Theta_{BGM}/\sim$.

Proof. We only need to prove the more general case:

$$Y = \sum_h B_h X_h \phi_h + \sigma\epsilon \tag{8.17}$$

where B_h is of $\mathcal{B}_{p_h}$, X_h is of $\mathcal{N}(0, \sigma_h^2)$, ϵ is of $\mathcal{N}(0, Id)$ is identifiable modulo a permutation on $\{1, \ldots, q\}$ and a sign on $\phi_h (1 \leq h \leq q)$. Similarly to the proof of Prop.39, we have:

$$\begin{aligned} \phi_Y(\xi) &\triangleq E(e^{-i\langle Y, \xi \rangle}) \\ &= \Pi_h [p_h e^{-\frac{1}{2}\sigma_h^2 \langle \xi, \phi_h \rangle^2} + (1 - p_h)] e^{-\frac{1}{2}\sigma^2 |\xi|^2}. \end{aligned}$$

Still reasoning as the proof of Prop.39 and using the fact that $\forall 1 \leq h \leq q, 0 < p_h < 1$ we know that σ^2 is determined by $\phi_Y(\xi)$. Hence if we consider:

$$\widetilde{\phi_Y}(\xi) \triangleq \phi_Y(\xi) e^{-\frac{1}{2}\sigma^2 |\xi|^2},$$

we have:

$$\widetilde{\phi_Y}(t\xi) = \Pi_h [p_h e^{-\frac{1}{2}t^2 \sigma_h^2 \langle \xi, \phi_h \rangle^2} + (1 - p_h)].$$

We only need to prove the decomposition of the above equation is uniquely. Suppose that there are two decompositions such that:

$$\Pi_{h=1}^{q} [p_h e^{-\frac{1}{2}t^2 \sigma_h^2 \langle \xi, \phi_h \rangle^2} + (1 - p_h)] = \Pi_{h=1}^{q'} [p'_h e^{-\frac{1}{2}t^2 {\sigma'_h}^2 \langle \xi, \phi'_h \rangle^2} + (1 - p'_h)],$$

where ϕ_h, ϕ'_h are all unit-norm, q, q' are positive integers, $\sigma_i > 0, p_h \in (0, 1), \forall 1 \leq h \leq q$ and $\sigma'_h > 0, p'_h \in (0, 1), \forall 1 \leq h \leq q'$.

Replacing $-\frac{1}{2}t^2$ by z and extending analytical to the entire plan $z \in \mathbb{C}$, we have:

$$\Pi_{h=1}^{q} [p_h e^{z\sigma_h^2 \langle \xi, \phi_h \rangle^2} + (1 - p_h)] = \Pi_{h=1}^{q'} [p'_h e^{z{\sigma'_h}^2 \langle \xi, \phi'_h \rangle^2} + (1 - p'_h)], \forall z \in \mathbb{C} \tag{8.18}$$

Without loss of generality, we assume that:

$$\sigma_q = \max\{\sigma_1, \ldots, \sigma_q, \sigma'_1, \ldots, \sigma'_{q'}\}.$$

Taking $\xi = \phi_q$ in Eq.(8.18) and using Lemma 40, we know that there exists a $h \in \{1, \ldots, q'\}$ such that:

$$\sigma_q^2 \langle \phi_q, \phi_q \rangle^2 = {\sigma'_h}^2 \langle \phi_q, \phi'_h \rangle^2, \frac{1-p_q}{p_q} = \frac{1-p'_h}{p'_h}.$$

Hence,

$$\sigma_q^2 = \sigma_q^2 \langle \phi_q, \phi_q \rangle^2 = {\sigma'_h}^2 \langle \phi_q, \phi'_h \rangle^2 \leq {\sigma'_h}^2 \leq \sigma_q^2.$$

Then we must have:

$$\sigma_q = \sigma'_h, \phi_q = \pm \phi'_h, p_q = p'_h.$$

Dividing $\sigma_q^2 \langle \xi, \phi_q \rangle^2 + \frac{1-p_q}{p_q}$ from both sides of Eq. (8.18) and using the induction method we know that except a perturbation and except a sign on ϕ_h, ϕ'_h we have,

$$q = q'$$

and

$$\sigma_h = \sigma'_h, p_h = p'_h, \phi_h = \phi'_h, \forall 1 \leq h \leq q.$$

This completes the proof. □

8.3 From likelihood to MCMC

The maximum likelihood approach for the estimation problem is basically defined as the computation of:

$$\hat{\theta} = \arg\max_{\theta \in \Theta} L_{N,\theta}(Y) \text{ with } L_{N,\theta}(Y) = \sum_B \int_{X \geq 0} L_{N,\theta}(X, B, Y) dX. \tag{8.19}$$

8.3.1 The MCMC-EM approach

The resolution of (8.19) requires to use an EM (Expectation-Minimization) type algorithm [66] which is built on the following principle: let $\rho(dx, dB)$ be dominated measure associated with the likelihood (8.3) (i.e. the product of the measure of counting for B and Lebesgue measure restricted to subspace $X \geq 0$ for X). We then have,

$$l_{N,\theta}(Y) \triangleq \log \left(\int L_{N,\theta}(x, b, Y) d\rho \right) = \max_{\mu} \left(\int l_{N,\theta}(x, b, Y) d\mu - K(\mu, \rho) \right) \tag{8.20}$$

where μ is an arbitrary distribution for the couple (X, B) and $K(\mu, \rho)$ is the Kullback-Leibler divergence. The max is reached for $\mu = \mu_{\theta,Y}$ equal to the posterior distribution (X, B) knowning Y i.e.

$$\frac{d\mu_{\theta,Y}}{d\rho}(x, b) = L_{N,\theta}(x, b, Y) / Z_{N,\theta,Y} \tag{8.21}$$

with $Z_{N,\theta,Y} = \int L_{N,\theta}(x', b', Y) \rho(dx', db')$ (for details, see [67]). By alternating maximization on μ and on θ of (8.20), we are led to the following algorithm:

$$\hat{\theta}_{l+1} = \arg\max_{\theta} \int l_{N,\theta}(x, b, Y) d\mu_{\theta_l,Y}(dx, db). \tag{8.22}$$

Parameter: NEmIter (number of iter. of'EM), M (nb of iteration of MCMC)
input θ_0;
for l=1 NEmIter **do**
  for k=1 N **do**
    Generate a sequence $(X(m), B(m))_{1\leq m\leq M}$ according to kernel Q_{θ_l}.
  end for
  Compute $\theta_{l+1} = \arg\max_\theta \sum_{m=1}^{M} l_{N,\theta_l}(X(m), B(m), Y)$
end for
return θ_{NEmIter};

Table 8.1: pseudo-code version of algorithm MCMC-EM.

The integral of complete log-likelihood of the hidden variables following the posterior distribution θ_l (so called (E)-step, the (M)-step corresponding to the optimization in θ of (8.22)) cannot be given in closed form. A first approach would be to replace the posterior distribution by a Dirac mass on the maximum of a posteriori $x_{*,l}, b_{*,l}$ of $\mu_{\theta_l,Y}$ which leads to more or less a resolution of (8.5) by alternating maximization on θ and (X, b). However, it is known that usually this approach is not consistent (for instance, see [68]). Moreover, the calculation of the maximum on (X, b) with θ fixed is not easy because of the presence of a L^0 term. One alternative is to approximate the posterior distribution thanks to a MCMC (Monte Carlo Markov Chain) [69]. The algorithm generates, according to a Markov chain $(X(m), B(m))_{m\geq 1}$ associated to a Markov kernel Q_{θ_l} which admits $\mu_{\theta_l,Y}$ as the invariant measure and for this Markov chain we have,

$$\frac{1}{M}\sum_{m=1}^{M} l_{N,\theta_l}(X(m), B(m), Y) \to \int l_{N,\theta}(x, b, Y) d\mu_{\theta_l,Y}(dx, db). \tag{8.23}$$

The pseudo-code version of algorithm MCMC-EM (presented as Table 8.1) thus is obtained. It is necessary to detail here two important aspects: the choice of the kernel Q_θ and the method of carrying out the stage of maximization on θ.

8.3.2 MCMC dynamic

It is obviously a crucial point. Let us note that the independence of the observations (Y^k) in the database already makes that on the posteriori distribution, the parts $(X^k, B^k)_{1\leq k\leq N}$ of the hidden variables are independent among themselves. In this case, we can choose separable Markov kernels on various images. We thus reduce to the study of a kernel $Q^k_{\theta_l}$ on (X^k, B^k). We choose for the moment to explore kernels obtained by the composition of elementary kernels acting on restricted parts A of the co-ordinates. Thus let us introduce that $E = \{1, \cdots, q\} \times \Lambda$ is the space of the indices (h, s) and denote $W^k_{h,s} = (X^k_{h,s}, B^k_{h,s})$. The elementary kernels that we used are defined by,

$$Q^k_{\theta_l,A}(w, d\tilde{w}^k_A) = \mu^k_{\theta_l,Y^k}(d\tilde{w}^k_A | w^k_{E\setminus A}) \tag{8.24}$$

where μ^k_{θ,Y^k} indicates the posteriori distribution of W^k knowing Y^k for the parameter θ. However, the simulation of the kernel is not direct (because of the normalization constants to be computed and the mixture of several continuous conditional distributions for the values of the discrete variables). But we can use a acceptance-rejection method [70, 71] with intrumental distribution π^k_A under which, the variables $(X^k_{h,s})_{(h,s)\in A}$ follow the prior

Input: w, A, k
Set: flag $\leftarrow 1$.
while (flag) **do**
  Simulate $\tilde{W}_A^k$ according to the distribution π_A^k
  Calculate $\alpha = e^{-H_A^k(w,\tilde{W}_A)+H_A^k(w)}$.
  if rand(1)$<\alpha$ **then**
    set: flag $\leftarrow 0$
  end if
end while
return $\tilde{W}_A^k$;

Table 8.2: Acceptance-rejection method

distribution of θ_l and $(B_{h,s}^k)_{(h,s)\in A}$ are i.i.d. Bernoulli variables of parameter p_c (in the following we will often use $p_c = 1/2$)[1].

Suppose $H_A^k(w,\tilde{w}_A) \triangleq -\log\left\{\frac{d\mu_{\theta_l,Y}^k}{d\pi_A^k}(\tilde{w}_A^k|w_{E\setminus A}^k)\right\}$. For $\theta_l = \theta$ and $p_c = 1/2$, we have:

$$H_A^k(w,\tilde{w}_A) = \frac{|Y^k - D_{E\setminus A}z_{E\setminus A}^k - D_A\tilde{z}_A^k|^2}{2\sigma^2} + \sum_A r_h\tilde{b}_{h,s}^k + \text{Cte} \tag{8.25}$$

where D_B indicates the restriction of D on the indices B. By introducing the projection Y_A^k of Y^k on space $D_{E\setminus A}z_{E\setminus A}^k + D_A\mathbb{R}^A$, and $x_{A,*}^k$ the co-ordinates of Y_A^k on $\mathbb{R}^A$ of this projection, we have

$$H_A^k(w,\tilde{w}_A) = \frac{|D_A(\tilde{z}_A^k - x_{A,*}^k)|^2}{2\sigma^2} + \sum_A r_h\tilde{b}_{h,s}^k + \text{Cte}. \tag{8.26}$$

For w fixed, we now obtain a problem in $(\mathbb{R}_+ \times \{0,1\})^A$ for which we can calculate the minimum $H_A^k(W)$.

For the calculation of $H_A^k(w)$, we calculate for all $C \subset A$,

$$x_{A,C}^k = \underset{u_C^k\in\mathbb{R}^C}{\arg\min}\,|D_C u_C^k - D_A x_{A,*}^k|^2 \tag{8.27}$$

and we denote $C_h = \{s \in \Lambda \mid (h,s) \in C \text{ and } x_{A,C_{|(h,s)}}^k > 0\}$. As

$$H_A^k(w) = \min_{C\subset A, x_{A,C}^k\geq 0}\left\{\frac{|D_C x_{A,C}^k - D_A x_{A,*}^k)|^2}{2\sigma^2} + \sum_{1\leq h\leq q} r_h|C_h|\right\} + \text{Cte}\,. \tag{8.28}$$

We thus deduce a method of obtaining $H_A^k(w)$: for small A, say $\#A = 2$, we can take all possible C ($2^{\#A}$ choices) and then compute Eq.(8.27) directly. The desired kernel $Q_{\theta_l}^k$ is defined as the composition the elementary local kernels $Q_{\theta_l,A}^k$ for an appropriate sweeping of the set of indexes.

[1]It seems strange not to choose the prior distribution of the current parameter θ_l but that leads in the usual cases where the proportion of active flags is small to propose in a very preferential way the value 0 for Bernoulli. When an element of the dictionary must be presented so that the candidate is not rejected, this will cause to simulate an unnecessarily large number of candidate before the first acceptance.

8.3.3 The update of θ

Now, let us look at the update of θ. Since,

$$\theta_{l+1} = \arg\max_{\theta} \sum_{m=1}^{M} l_{N,\theta_l}(X(m), B(m), Y), \tag{8.29}$$

Calculating in a straightforward manner, we have,

$$\begin{cases} p_h &= \sum_{k,m,s} B_{h,s}^{k,m}/(NMn^2) \\ \frac{1}{\lambda_h} &= \sum_{k,m,s} X_{h,s}^{k,m}/(NMn^2) \\ \sigma &= \sqrt{\sum_{k,m} \left(|Y^k - \Phi B^{k,m} \otimes X^{k,m}|^2\right)/(NMn^2)} \\ \Phi &= \arg\min_{\Phi} \sum_{k,m} |Y^k - \Phi B^{k,m} \otimes X^{k,m}|^2. \end{cases} \tag{8.30}$$

The details of the derivation the last equation is presented in the upcoming section.

8.4 Mean field approach

It is well known that the MCMC-EM algorithm or more sophisticated stochastic approximation versions like MCMC-SAEM [72], despite good convergence properties are unfortunately slow as soon as the number of hidden variable is large. In the following part, we derive a speedup version based on mean field approximation in variational EM [73].

8.4.1 Mean field derivation

The main idea of variational EM methods is to replace the untractable computation of the posterior distribution $\mu_{\theta,Y^k}(dx^k, db^k)$ in the (E) step by computable approximation in a given family of distribution. In the mean field approach, we seek to approximate the posterior distribution by a product distribution on B^k and X^k. In this chapter, we will derive the mean field approximation for the Bernoulli-Gaussian model (BGM) for which mean-field equations are simpler.

For this let us consider $\mathcal{M}'$ the collection of the distributions ν under which the variables $(B_{h,s}^k)_{h,s}$ and $(X_{h,s}^k)_{h,s}$ are independent and for all (h,s), $B_{h,s}^k$ follows a Bernoulli distribution of parameter $q_{h,s}$ and $X_{h,s}^k$ follows a distribution $\mathcal{N}(m_{h,s}, \sigma_{h,s}^2)$. We pose then:

$$\hat{\nu}_k = \inf_{\nu \in \mathcal{M}'} K(\nu, \mu_{\theta,Y^k}). \tag{8.31}$$

The idea is that $\hat{\nu}_k$ is an approximation which can be calculated fairly easily by fixed point method (we will see it later) and can be plugged into algorithm of EM[2].

In fact, if we denote $y = Y^k$ (we also omit the superscript k in the following but we should note that we consider not only the case of a single observed image here), then calculating directly, we have:

[2] a basic fact is that $\int l_\theta(x^k, b^k, Y^k)d\nu - K(\nu, \rho^k) = -K(\nu, \mu_{\theta,Y^k}) + \text{Cte}$ so that the maximization of the term of left on $\mathcal{M}'$ is equivalent to the minimization of $K(\nu, \mu_{\theta,Y^k})$

$$K(\nu, \mu_{\theta,Y^k}) = \sum_{h,s} \left\{ \log(\frac{q_{h,s}}{p_h}) q_{h,s} + \log(\frac{1-q_{h,s}}{1-p_h})(1-q_{h,s}) + \frac{1}{2}\left(\frac{\sigma_{h,s}^2 + m_{h,s}^2}{\sigma_h^2} - \log(\sigma_{h,s}^2) \right) \right\} + \frac{1}{2\sigma^2}\nu(|y - DZ|^2) + \text{Cte} \quad (8.32)$$

We remark easily that now $K(\nu, \mu_{\theta,Y^k})$ is convex for each coordinate of

$$\left((q_{h,s})_{h,s}, (m_{h,s})_{h,s}, (\sigma_{h,s}^2)_{h,s} \right).$$

Moreover the derivative of $\log(\frac{q_{h,s}}{p_h}) q_{h,s} + \log(\frac{1-q_{h,s}}{1-p_h})(1-q_{h,s})$ is $-\infty$ at $q_{h,s} = 0$ and $+\infty$ at $q_{h,s} = 1$, we thus easily know that the minimum exists and is reached for $q_{h,s} \in]0,1[$ and $\sigma_{h,s}^2 > 0$.

8.4.2 Fixed point equation

Using the fact that $|\phi_{h,s}|^2 = 1$, we have

$$\nu(|y - DZ|^2) = |y - D\nu(Z)|^2 + \sum_{h,s} V_\nu(Z_{h,s}), \quad (8.33)$$

with $V_\nu(Z_{h,s}) = m_{h,s}^2 q_{h,s}(1-q_{h,s}) + q_{h,s}\sigma_{h,s}^2$. Then we know that

$$\frac{\partial}{\partial q_{h,s}} \nu(|y - DZ|^2) = -2\langle y - D\nu(Z), \phi_{h,s} m_{h,s} \rangle + m_{h,s}^2(1 - 2q_{h,s}) + \sigma_{h,s}^2 . \quad (8.34)$$

Since

$$\frac{\partial}{\partial q_{h,s}} \left\{ \log(\frac{q_{h,s}}{p_h}) q_{h,s} + \log(\frac{1-q_{h,s}}{1-p_h})(1-q_{h,s}) \right\} = \log\left(\frac{q_{h,s}(1-p_h)}{(1-q_{h,s})p_h} \right), \quad (8.35)$$

the equation $\frac{\partial}{\partial q_{h,s}} K(\nu, \mu_{\theta,Y^k}) = 0$ leads to

$$q_{h,s} = \frac{p_h}{p_h + (1-p_h)\exp(\frac{-2\langle y - D\nu(Z), \phi_{h,s} m_{h,s} \rangle + m_{h,s}^2(1-2q_{h,s}) + \sigma_{h,s}^2}{2\sigma^2})} . \quad (8.36)$$

The same reason, since

$$\frac{\partial}{\partial m_{h,s}} \nu(|y - DZ|^2) = -2\langle y - D\nu(Z), \phi_{h,s} q_{h,s} \rangle + 2m_{h,s} q_{h,s}(1-q_{h,s}) = -2(\langle y - D\nu(Z), \phi_{h,s} q_{h,s} \rangle + m_{h,s} q_{h,s}^2) + 2m_{h,s} q_{h,s} , \quad (8.37)$$

the equation $\frac{\partial}{\partial m_{h,s}} K(\nu, \mu_{\theta,Y^k}) = 0$ leads to

$$m_{h,s} = \frac{\langle y - D\nu(Z), \phi_{h,s} q_{h,s} \rangle + m_{h,s} q_{h,s}^2}{\frac{\sigma^2}{\sigma_h^2} + q_{h,s}} . \quad (8.38)$$

Finally, since

$$\frac{\partial}{\partial \sigma_{h,s}^2} \nu(|y - DZ|^2) = q_{h,s} \quad (8.39)$$

we obtain

$$\sigma_{h,s}^2 = \frac{1}{\frac{1}{\sigma_h^2} + \frac{q_{h,s}}{\sigma^2}}\,. \tag{8.40}$$

Combining (8.36), (8.38) et (8.40), we obtain the fixed point equation

$$\left\{\begin{array}{rcl} q_{h,s} & = & \frac{p_h}{p_h+(1-p_h)\exp(\frac{-2\langle y-D\nu(Z),\phi_{h,s}m_{h,s}\rangle+m_{h,s}^2(1-2q_{h,s})+\sigma_{h,s}^2}{2\sigma^2})} \\ m_{h,s} & = & \frac{\langle y-D\nu(Z),\phi_{h,s}q_{h,s}\rangle+m_{h,s}q_{h,s}^2}{\frac{\sigma^2}{\sigma_h^2}+q_{h,s}} \\ \sigma_{h,s}^2 & = & \frac{1}{\frac{1}{\sigma_h^2}+\frac{q_{h,s}}{\sigma^2}} \end{array}\right. \tag{8.41}$$

When the mapping underlying the fixed point equation is a contraction, (8.41) can be solved by fixed point method, precisely, by iterating from a starting solution[3]. The update in the fixed point can be made in a fast way since $\nu(Z) = q \otimes m$, thus

$$D\nu(Z)(s) = \sum_{1\le h\le q} (\phi_h \star (q_h \otimes m_h))(s)$$

and denoting $a = y - D\nu(Z)$, we have,

$$\langle y - D\nu(Z), \phi_{h,s}\rangle = (a \star \check{\phi}_h)(s)$$

where $\check{f}(s) = f(-s)$ is the symmetry of f. Using fast Fourier transform, the update is of complexity $O(qn^2\log(N))$ times the iteration number required to have convergence towards the fixed point.

Now, let us look at the update of θ. We reintroduce the superscript k again and denote $\hat{\nu}_k$ the solution of (8.41). By using the approximation for $\theta = \theta_l$ of $\mu_{\theta_l,Y}$ defined by

$$\hat{\nu} = \otimes_{k=1}^n \hat{\nu}_k, \tag{8.42}$$

we have

$$\theta_{l+1} = \arg\max_{\theta\in\Theta_{\mathrm{BGM}}} \int l_{N,\theta}(x,b,Y)d\hat{\nu}. \tag{8.43}$$

Computing directly, we know that:

$$\left\{\begin{array}{rcl} p_h & = & \sum_{k,s} q_{h,s}^k/(Nn^2) \\ \sigma_h^2 & = & \sum_{k,s}\left((\sigma_{h,s}^k)^2 + (m_{h,s}^k)^2\right)/(Nn^2) \\ \sigma & = & \sqrt{\sum_k\left(|Y^k - \Phi q^k \otimes m^k|^2 + \sum_{h,s}(m_{h,s}^k)^2 q_{h,s}^k(1-q_{h,s}^k) + q_{h,s}^k(\sigma_{h,s}^k)^2\right)/(Nn^2)} \\ \Phi & = & \arg\min_\Phi \sum_k |Y^k - \Phi q^k \otimes m^k|^2. \end{array}\right. \tag{8.44}$$

The last equation can not be solved in closed form since the elements of Φ are on the sphere. A method to tackle this problem is to solve it without constraints on the norm

[3] a difficult problem is to know when the application is a contraction. You have no general answer about this point but the practice shows that this is not always the case. Alternative strategies will be given below.

and then project the elements on the sphere by normalization. Then for the problem without constraints, by denoting $z_h^k(s) \triangleq q_{h,s}^k m_{h,s}^k$, we have:

$$\sum_{k=1}^{N} \tilde{z}_h^k \star (Y^k - \sum_{h'=1}^{q} z_{h'}^k \star \phi_{h'}) = 0, \ \forall 1 \leq h \leq q. \tag{8.45}$$

This is a linear equations system which can be solved easily on with the Fast Fourier Transform by writing:

$$M(s)(\mathcal{F}\Phi)(s) = U(s), \ \forall s \in \Lambda \tag{8.46}$$

where $M(s)$ is the complex matrix $q \times q$ and $U(s)$ the complex vector defined by

$$M_{h,h'}(s) = \sum_{k=1}^{N} \overline{(\mathcal{F}z_h^k)}(s)(\mathcal{F}z_{h'}^k)(s) \text{ et } U_h(s) = \sum_{k=1}^{N} \overline{(\mathcal{F}z_h^k)}(s)(\mathcal{F}Y)(s) \tag{8.47}$$

and $(\mathcal{F}\Phi)(s)$ is the complex vector defined by

$$(\mathcal{F}\Phi)_h(s) = (\mathcal{F}\phi_h)(s)\,. \tag{8.48}$$

Consequently we are brought back to reverse n^2 system $q \times q$ for a complexity of $O(q^2n^2 + qn^2\log(n))$. Once obtained $\mathcal{F}\Phi$, we can recover the new dictionary by inverting the Fourier transform and by re-normalization the vectors to bring them back on the sphere.

8.4.3 Presence of background

Suppose that we are interested in the situation where a weak background is presented in the image Y. Now our model is changed from Eq. (8.1) to

$$Y = \sum_{h,s} B_{h,s} X_{h,s} \phi_{h,s} + \sum_{h'} X_{h'} G_{h'} + \sigma\epsilon\,, \tag{8.49}$$

where $X_{h'}$ $(1 \leq h' \leq q')$ is of $\mathcal{N}(0, \sigma_{h'}^2)$ and $G_{h'}$ are typical functions to represent the background. To simplify the model, we assume that $G_{h'}$ are known and fixed and can be estimated by other statistical model elsewhere. We also assume that $\|G_{h'}\|_2 = 1$. A trivial choice is to let $q' = 1$ and $G_1 = \frac{1}{\sqrt{|\Lambda|}} 1_\Lambda$ where 1_Λ is the index function on Λ. The parameters of our new model are defined as:

$$\theta = (\Phi, (\sigma_h^2)_{1\leq h\leq q}, (p_h)_{1\leq h\leq q}, \sigma^2, (\sigma_{h'}^2)_{1\leq h'\leq q'}) \in \Theta_{\text{BGM}}, \tag{8.50}$$

which are to be estimated.

Similarly, consider $\mathcal{M}'$ the collection of distribution ν under which the variable $(B_{h,s}^k)_{h,s}$ and $(X_{h,s}^k)_{h,s}$, $X_{h'}$ are independent. For all (h, s, h'), $B_{h,s}^k$ follows a Bernoulli distribution of parameter $q_{h,s}$, $X_{h,s}^k$ follows the distribution $\mathcal{N}(m_{h,s}, \sigma_{h,s}^2)$ and $X_{h'}$ follows the distribution $\mathcal{N}(m_{h',0}, \sigma_{h',0}^2)$. We pose

$$\hat{\nu}_k = \inf_{\nu\in\mathcal{M}'} K(\nu, \mu_{\theta,Y^k}) \tag{8.51}$$

Similarly to Eq.(8.32), denoting $y = Y^k$, the Kullback-Leibler distance is:

$$K(\nu,\mu_{\theta,Y^k}) =$$
$$\sum_{h,s}\left\{\log(\frac{q_{h,s}}{p_h})q_{h,s}+\log(\frac{1-q_{h,s}}{1-p_h})(1-q_{h,s})+\frac{1}{2}\left(\frac{\sigma_{h,s}^2+m_{h,s}^2}{\sigma_h^2}-\log(\sigma_{h,s}^2)\right)\right\}$$
$$+\sum_{h'}\frac{1}{2}\left(\frac{\sigma_{h',0}^2+m_{h',0}^2}{\sigma_{h'}^2}-\log(\sigma_{h',0}^2)\right)+\frac{1}{2\sigma^2}\nu(|y-DZ-\sum_{h'}X_{h'}G_{h'}|^2)+\text{Cte.} \quad (8.52)$$

Here again $K(\nu,\mu_{\theta,Y^k})$ is convex for each coordinate of

$$\left((q_{h,s})_{h,s},(m_{h,s})_{h,s},(\sigma_{h,s}^2)_{h,s},(\sigma_{h',0}^2)_{h'},(m_{h',0})_{h'}\right).$$

Since $K(\nu,\mu_{\theta,Y^k})$ is lower bounded, the minimum exists.

Using the fact that $|\psi_{h,s}|^2=1$ and $|G_{h'}|^2=1$, we have

$$\nu(|y-DZ-\sum_{h'}X_{h'}G_{h'}|^2) \quad (8.53)$$
$$= |y-D\nu(Z)-\sum_{h'}\nu(X_{h'})G_{h'}|^2+\sum_{h,s}V_\nu(Z_{h,s})+\sum_{h'}V_\nu(X_{h'}) \quad (8.54)$$
$$= |y-D\nu(Z)-\sum_{h'}m_{h',0}G_{h'}|^2+\sum_{h,s}V_\nu(Z_{h,s})+\sum_{h'}\sigma_{h',0}^2, \quad (8.55)$$

where $V_\nu(Z_{h,s})=m_{h,s}^2q_{h,s}(1-q_{h,s})+q_{h,s}\sigma_{h,s}^2$. Hence, we know that

$$\frac{\partial}{\partial q_{h,s}}\nu(|y-DZ-\sum_{h'}X_{h'}G_{h'}|^2)=-2\langle y-D\nu(Z)-m_{h',0}G_{h'},\phi_{h,s}m_{h,s}\rangle+m_{h,s}^2(1-2q_{h,s})+\sigma_{h,s}^2\,. \quad (8.56)$$

The equation $\frac{\partial}{\partial q_{h,s}}K(\nu,\mu_{\theta,Y^k})=0$ gives

$$q_{h,s}=\frac{p_h}{p_h+(1-p_h)\exp(\frac{-2\langle y-D\nu(Z)-\sum_{h'}m_{h',0}G_{h'},\phi_{h,s}m_{h,s}\rangle+m_{h,s}^2(1-2q_{h,s})+\sigma_{h,s}^2}{2\sigma^2})}\,. \quad (8.57)$$

Now the equation of fixed point Eq.(8.41) is changed into:

$$\begin{cases} q_{h,s} &= \frac{p_h}{p_h+(1-p_h)\exp(\frac{-2\langle y-D\nu(Z)-\sum_{h'}m_{h',0}G_{h'},\phi_{h,s}m_{h,s}\rangle+m_{h,s}^2(1-2q_{h,s})+\sigma_{h,s}^2}{2\sigma^2})} \\ m_{h,s} &= \frac{\langle y-D\nu(Z)-\sum_{h'}m_{h',0}G_{h'},\phi_{h,s}q_{h,s}\rangle+m_{h,s}q_{h,s}^2}{\frac{\sigma^2}{\sigma_h^2}+q_{h,s}} \\ \sigma_{h,s}^2 &= \frac{\sigma^2\sigma_h^2}{\sigma^2+\sigma_h^2q_{h,s}} \\ m_{h',0} &= \frac{\sigma_{h'}^2}{\sigma^2+\sigma_{h'}^2}\langle y-D\nu(Z)-\sum_{h''\neq h'}m_{h'',0}G_{h''},G_{h'}\rangle \\ \sigma_{h',0}^2 &= \frac{\sigma^2\sigma_{h'}^2}{\sigma^2+\sigma_{h'}^2}\,. \end{cases} \quad (8.58)$$

The complete likelihood for the model with background is,

$$L_\theta(X,B,X',Y)=e^{-|Y-DZ-\sum_{h'}X_{h'}G_{h'}|^2/(2\sigma^2)}(2\pi\sigma^2)^{-|\Lambda|/2}\prod_{h,s}p_h^{B_{h,s}}(1-p_h)^{1-B_{h,s}}$$
$$\prod_{h,s}\frac{1}{\sqrt{2\pi\sigma_h^2}}e^{-X_{h,s}^2/(2\pi\sigma_h^2)}\prod_{h'}\frac{1}{\sqrt{2\pi\sigma_{h'}^2}}e^{-X_{h'}^2/(2\pi\sigma_{h'}^2)}, \quad (8.59)$$

thus the *log*-likelihood turns into:

$$\begin{aligned} l_\theta(X,B,X',Y) = &-\frac{1}{2\sigma^2}|Y-DZ-\sum_{h'}X_{h'}G_{h'}|^2-\frac{|\Lambda|}{2}\log(2\pi\sigma^2)\\ &+\sum_{h,s}B_{h,s}\log p_h+(1-B_{h,s})\log(1-p_h)-\frac{1}{2}\log(2\pi\sigma_h^2)-\frac{X_{h,s}^2}{2\sigma_h^2}\\ &+\sum_{h'}-\frac{1}{2}\log(2\pi\sigma_{h'}^2)-\frac{X_{h'}^2}{2\sigma_{h'}^2} \end{aligned} \tag{8.60}$$

We need update θ as:

$$\theta_{l+1}=\arg\min_{\theta\in\Theta_{\mathrm{BGM}}}\sum_{k=1}^{N}\int l_\theta(x,b,x',Y^k)d\hat{\nu} \tag{8.61}$$

where

$$\hat{\nu}=\otimes_{k=1}^{n}\hat{\nu_k}. \tag{8.62}$$

Calculating directly, the update of θ, Eq.(8.44) is changed to:

$$\left\{\begin{array}{rcl} p_h &=& \sum_{k,s}q_{h,s}^k/(Nn^2)\\ \sigma_h^2 &=& \sum_{k,s}\left((\sigma_{h,s}^k)^2+(m_{h,s}^k)^2\right)/(Nn^2)\\ \sigma^2 &=& \sum_k\Big\{|Y^k-\Phi q^k\otimes m^k-\sum_{h'}m_{h',0}^kG_{h'}|^2+\sum_{h,s}(m_{h,s}^k)^2q_{h,s}^k(1-q_{h,s}^k)\\ &+& q_{h,s}^k(\sigma_{h,s}^k)^2+\sum_{h'}(\sigma_{h',0}^k)^2\Big\}/(Nn^2)\\ \sigma_{h'}^2 &=& \sum_k\left((\sigma_{h',0}^k)^2+(m_{h',0}^k)^2\right)/(N)\\ \Phi &=& \arg\min_\Phi\sum_k|Y^k-\Phi q^k\otimes m^k-\sum_{h'}m_{h,0'}^kG_{h'}|^2. \end{array}\right. \tag{8.63}$$

Notice that, if we want to learn the background $G=(G_{h'})_{1\le h'\le q'}$, we can use:

$$G=\arg\min_G\sum_k|Y^k-\Phi q^k\otimes m^k-\sum_{h'}m_{h',0}^kG_{h'}|^2 \tag{8.64}$$

That's to say,

$$G_{h'}=\frac{1}{1+m_{h',0}^2}\sum_k\{Y^k-\Phi q^k\otimes m^k-\sum_{h''\neq h'}m_{h'',0}^kG_{h''}\}^2 \tag{8.65}$$

and then normalization $G_{h'}$ to let $\|G_{h'}\|=1$.

When $q'=1$, the solution is of closed form. When $q'>1$, we need iterate on Eq.(8.65).

8.5 Numerical aspects

When the initial value for the parameter

$$\theta_0 = (\Phi, (\sigma_h^2)_{1\leq h\leq q}, (p_h)_{1\leq h\leq q}, \sigma, (\sigma_{h'}^2)_{1\leq h'\leq q'}) \in \Theta_{\mathrm{BGM}} \tag{8.66}$$

is reasonable (say, near the "real" parameter) and for each θ_l, the initial values $(q_{h,s}, m_{h,s}, \sigma_{h,s}^2)$ is in the region where the map Eq.(8.58) is a contraction, our EM method with mean field is usually converging. As these conditions are not easy to guarantee, we need some technics to promote the possibility of convergence.

8.5.1 Grids for fixed point equation

Before going on, we need state some simple facts for the fix point equation Eq.(8.58). Since

$$D\nu(Z)(s) = \sum_{1\leq h\leq q} (\phi_h \star (q_h \otimes m_h))(s),$$

we know that for (h, s) fixed, we have:

$$\begin{aligned}
\langle y - D\nu(Z) - \sum_{h'} m_{h',0} G_{h'}, \phi_{h,s}\rangle &= -\langle D\nu(Z), \phi_{h,s}\rangle + A_1 \\
&= -\langle \sum_{h',s'} q_{h',s'} m_{h',s'} \phi_{h',s'}, \phi_{h,s}\rangle + A_2 \\
&= q_{h,s} m_{h,s} \langle \phi_{h,s}, \phi_{h,s}\rangle + A_3 \\
&= q_{h,s} m_{h,s} + A_3,
\end{aligned}$$

where A_1, A_2, A_3 are constants for $m_{h,s}, q_{h,s}$.

Using this fact, from Eq.(8.58) we know that, for q, s fixed, the update of $q_{h,s}$ does not depend on $q_{h,s}$ itself. This means that if all the other values are fixed and we only change $q_{h,s}$, the update formula is nothing but just the unique minimum point of $q_{h,s}$ for $K(\nu, \mu_{\theta,Y^k})$. This observation also holds for other parameter $(\sigma_{h,s}^2)_{h,s}, (\sigma_{h',0}^2)_{h'}, (m_{h',0})_{h'}$. Hence, if we update all these parameter sequentially, we are always on the direction of decreasing $K(\nu, \mu_{\theta,Y^k})$. Thus we can find the minimum point of $K(\nu, \mu_{\theta,Y^k})$ by sequential method.

Now the problem is that sequential update may be time consuming. We propose a grids method to approximate the sequential updating. For fixed integer r, let

$$\Gamma_{i,j} = \{(\mathbb{Z}/r\mathbb{Z})^2 + (i, j)\} \bigcap \Lambda, 1 \leq i, j \leq r, \tag{8.67}$$

$$\Gamma_r = \{\Gamma_{i,j} | 1 \leq i, j \leq r\}. \tag{8.68}$$

Then Γ_r is a partition of Λ. Now we do not update all the $s \in \Lambda$ at the same time as this might cause convergence problem. Instead, we update $s \in \Gamma_{i,j}$ sequentially i.e. we first update all the $s \in \Gamma_{1,1}$, then all the $s \in \Gamma_{1,2}$ and so on. When r is large enough (say $r > c$, where c is the size of support compact of all the ϕ_h), convergence is guaranteed. In practice, we need not r be so large, $r = c/2$ is also reasonable choice. When (i, j) runs over $1 \leq i, j \leq r$, $\Gamma_{i,j}$ will cover all the $s \in \Lambda$ one time, after this, we can update the other parameter $(m_{h',0})_{1\leq h'}, (\sigma_{h',0}^2)_{1\leq h'}$ one time. If the initiation of parameters is reasonable, the convergence is almost sure.

In practice, if we think that the grids method is still time consuming, we can carry out the grids method and update totally alternatively, e.g. 5 times update totally then 1 time grids method.

8.5.2 Thresholding to get sparse elements

Our first technic is to threshold each element of dictionary ϕ_h after the inverse Fourier transform on the result of Eq.(8.48):

$$\phi_h(s) = \begin{cases} \phi_h(s) & \text{if } |\phi_h(s)| \geq \eta \cdot \max_s |\phi_h(s)| \\ 0 & \text{else} \end{cases} \tag{8.69}$$

where η tends from a small positive (say 0.1) to 0 along the EM iteration. The reason for this technic is that we prefer the sparse element and we want to build the larger values of ϕ_h firstly during the EM procedure.

8.5.3 Support compact

Our second technic is to use the constraint that the support of ϕ_h is compact. This is reasonable for us because we are more interested in looking for the typical patterns in the image Y^k. As we have placed ϕ_h in every position $s \in \Lambda$, we can suppose that the support of ϕ_h is much smaller than Λ. So suppose that the support of ϕ_h is $\Lambda_h \subset \Lambda$, we can update ϕ_h after Eq.(8.63) as:

$$\phi_h(s) = 1_{\Lambda_h}(s)\phi_h(s), \forall s \in \Lambda, \tag{8.70}$$

where again 1_{Λ_h} is index function and typical choice of Λ_h is

$$\Lambda_h = [0, c] \times [0, c] \tag{8.71}$$

where c is known and much smaller than n.

8.5.4 Initialization of parameters

The initializaion of parameters is of great importance. We present here as two parts.

- Dictionary Φ
  For each $y \in \mathcal{Y}$, it has $(n - c + 1)^2$ patches of size c^2. We can choose the patch which has maximum l^2-norm, normalization this patch, and then extend it to size of n^2 by filling in zero. We set the result as ϕ_y. Randomly choose q differen images in the training set: $(y^h)_{1 \leq h \leq q} \in \mathcal{Y}$, and then we can set the initiation of Φ as:

$$\Phi = (\phi_{y^h})_{1 \leq h \leq q}.$$

- $z_{h,s}, m_{h',0}$
  When n is small, say $n \leq 20$, combining all the above technics with other reasonable parameters, the convergence is almost sure. When n is larger than 20, the convergence is not so easy, usual parameters setting can not guarantee the convergence. But when we observe carefully the situation of failure of convergence, we find that it is caused by the fact during iteration, most part of image $y = Y^k$ in the training set are not represented sufficiently. This hints us that

$$|y - D\nu(Z) - \sum_{h'} m_{h',0} G_{h'}|^2$$

(it is essential part of Eq.(8.53) and key part of Eq.(8.52)) should be reasonable small. To achieve this goal, as

$$\begin{aligned} D\nu(Z)(s) &= \sum_{1\leq h\leq q} (\phi_h \star (q_h \otimes m_h))(s) \\ &= \sum_{h,s} q_{hs} m_{hs} \phi_{h,s} \\ &= \sum_{h,s} z_{hs} \phi_{h,s}, \end{aligned}$$

and all of $(\phi_{h,s})_{h,s}, (G_{h'})_{h'}$ are normalized, we can matching pursuit y in the total dictionary

$$((\phi_{h,s})_{h,s}, (G_{h'})_{h'})_{1\leq h\leq q, s\in\Lambda, 1\leq h'\leq q'}$$

and then set $z_{h,s}, m_{h',0}$ as the corresponding coefficients.

With all the above technics of convergence, our tests show that even $n \geq 100$, our codes is stable and can provide a reasonable solution under the condition that the size of training set N is reasonable large though the choose of N is beyond the scope of our current chapter.

8.5.5 Force the appearance probability of $(\phi_h)_{1\leq h\leq q}$

When the data does not obey the probability models very well, through our experiments we find that it is useful to force all the (p_h) to the same value. So sometimes, through a flag FEqProb, the update of θ, Eq.(8.62)

$$p_h = \sum_{k,s} q_{h,s}^k / (Nn^2)$$

is changed to

$$p_h = \sum_{k,s,h} q_{h,s}^k / (qNn^2).$$

Beware that in all of our experiments, the FEqProb will not be turned on except in the experiment of learning typical patterns from natural image (see Section 8.7.5) where we set FEqProb=1. When q is fixed, this flag can force the learning processus to learn q atoms, i.e there is no atom who is always unchanged ever since the initialization. We remark that this flag might by discarded in the final version of the dissertation.

8.5.6 Details of mean field algorithm

Mixing all these aspects, we present our mean field algorithm as Table 8.3.

8.6 Experiments on MCMC

In all the experiments for MCMC, the training set always contains 10 images of size 15×15. They are generated as the additive composition of the translation of 3 basic atoms (see Figure 8.1) at random position over Λ, together with a Gaussian noise of standard variation $\sigma = 0.1$. Figure 8.2 shows the training set. The real dictionary contains $q = 3$ atoms (see Figure 8.1) and we want to test what happens if set q correctly ($q = 3$) and wrongly ($q = 1$).

Parameter: NEmIter (number of iter. of'EM), M (nb of iteration of Mean Field), N (size of training set), r (for Grids), η (to threshold dictionary), FEqProb (flag to force p_h to same level), N_c (to control the use of Grid), c (for compact support)
input $\theta_0 = (\Phi, (\sigma_h^2)_{1\le h\le q}, (p_h)_{1\le h\le q}, \sigma^2, (\sigma_{h'}^2)_{1\le h'\le q'})$
prepare Generate Grids Γ_r, Γ_1 by Eq.(8.68)
for $k = 1$ to N **do**
  generate: $\Xi_k = (m_{h,s}, q_{h,s}, m_{h',0}, \sigma_{h,s}^2, \sigma_{h',0}^2)$ (by MP method)
end for
for $l = 1$ to NEmIter **do**
  if $l = 1$ or N_c dividing NEmIter **then**
    set Grid Γ^l as Γ_r
  else
    set Grid Γ^l as Γ_1
  end if
  for $k = 1$ to N **do**
    initial: $(m_{h,s}, q_{h,s}, m_{h',0}, \sigma_{h,s}^2, \sigma_{h',0}^2)$ with Ξ_k
    for $m = 1$ to M **do**
      update $(m_{h,s}, q_{h,s}, m_{h',0}, \sigma_{h,s}^2, \sigma_{h',0}^2)$ by fixed point equation (Eq. 8.58) with Grid Γ^l
    end for
  end for
  Update θ_{l+1} by Eq. (8.63)
  Threshold $\Phi = (\phi_h)_{1\le h\le q}$ by Eq.(8.69) and then decrease η
  Project $\Phi = (\phi_h)_{1\le h\le q}$ on the support compact by Eq.(8.70) and then normalization to 1
  if FEqProb=1 **then**
    set all $(p_h)_{1\le h\le q}$ to $\frac{1}{q}\sum_h p_h$
  end if
end for
return θ_{NEmIter}

Table 8.3: Mean Field-EM Method.

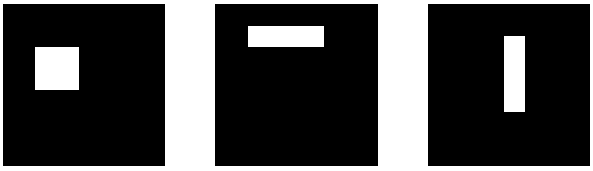

Figure 8.1: Real dictionary: left: 4×4 square; middle: 2×7 rectangle; right: 7×2 rectangle.

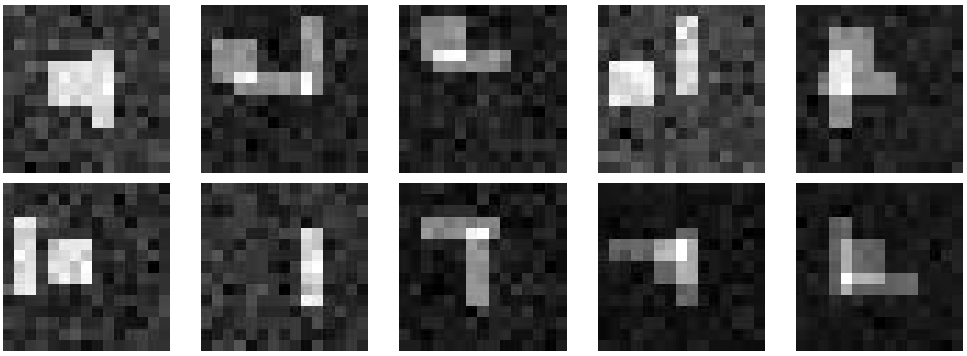

Figure 8.2: Training set for simple structure experiments of MCMC and of mean field. The noise level is 0.1.

8.6.1 MCMC for simple structure with $q = 3$

We fixed $q = 3$ in this experiment. After MCMC, the learned dictionary is shown in Figure 8.3. The reconstruction image

$$\sum_{h,s} B_{h,s} X_{h,s} \phi_{h,s}$$

is shown in Figure 8.4. We can see that MCMC can learn the dictionary properly.

8.6.2 MCMC for simple structure with $q = 1$

In this experiment, we set $q = 1$. The learned dictionary and reconstruction images given by the MCMC algortihm are shown in Figure 8.5 and Figure 8.6 respectively. From these two figures we can see that this time, since q is set too small, we are obliged to use a special atom: the common part of the real atoms to represent the images in the training set and this kind of representation is a little worse comparing to $q = 3$. The estimated noise is $\sigma = 0.12$, so interestingly, it seems that the noise level is properly estimated.

Figure 8.3: Learned dictionary by MCMC with $q = 3$.

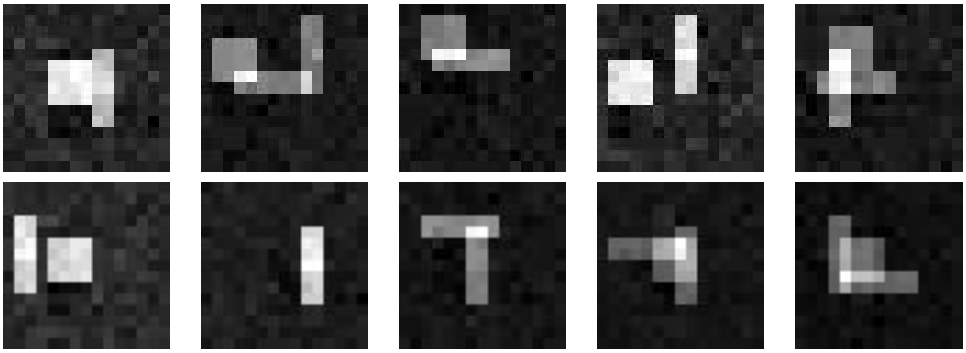

Figure 8.4: Reconstruction image by MCMC for $q = 3$.

Figure 8.5: Learned dictionary by MCMC with $q = 1$.

When q set properly, the MCMC approach works robustly and efficiently. Unfortunately, the computation burden of this approach is high. Hence, we illustrate now the mean field approach which is much faster than MCMC.

8.7 Experiments for mean field

8.7.1 Experiments on simple structure by mean field

In this experiment, the image of our training set is the same as the experiments for MCMC (see Section 8.6). After the completion of the mean field algorithm, the learned dictionary is shown as Figure 8.7. The reconstruction images are shown in Figure 8.8. The estimated noise standard deviation is 0.0995 (the real $\sigma = 0.1$). Hence, the performance of mean field is comparative to MCMC by various aspects: the result of the dictionary, the reconstruction images, the estimation of noise. Moreover, mean field method is much faster than MCMC.

8.7.2 Mean field for learning 5 numbers

In this experiment, we try to learn $q = 5$ basic atoms from a training set containing $N = 100$ images of size 20×20. The real atoms are shown in top image of Figure 8.10 and typical examples in the training set are shown in Figure 8.9. The noise level is 0.1.

After learning, the learned dictionary is shown as bottom image of Figure 8.10. Some reconstruction images are shown bottom in Figure 8.9.

8.7.3 Mean field for 10 numbers

When $q = 10$, the problem is very difficult. However, our result is also good. We use a training set of $N = 100$ images, the size is also 20×20. Typical examples are shown in

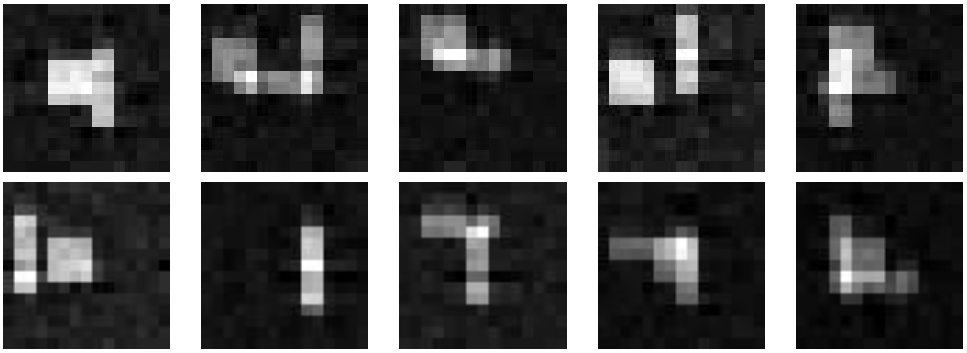

Figure 8.6: Reconstruction image of MCMC for $q = 1$.

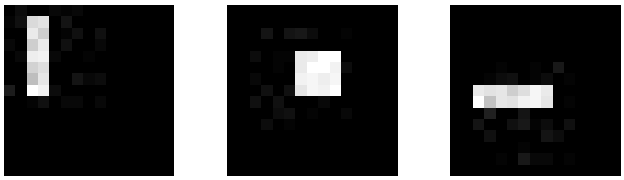

Figure 8.7: Learned dictionary for simple structure experiment by mean field.

the Figure 8.11, the standard deviation is 0.1.

The real dictionary is shown as the top 10 images in Figure 8.12. After learning, the learned dictionary is shown as the bottom 10 images of Figure 8.12.

8.7.4 Analysis on $q_{h,s}, z_{h,s}$

Now we study a little bit the distribution of the hidden variables through a simple example. The training set contains 9 images of size 15×15 (one of them, denoted by Y^k, is shown as the top-left image of Figure 8.13). Beware that we have translated this atoms to the corner, in order to make the reader judges the position of occurring atoms more clearly, the upcoming $q_{h,s}, z_{h,s}$ are also translated accordingly. The real dictionary is shown in Figure 8.1. The noise level is 0.1.

After completion of the mean field algorithm, all the reconstruction images are shown in the top-right image of Figure 8.13 (Every 15×15 patch in appropriate position is a reconstruction image). One of the 3 learning atoms, denoted by ϕ_h is shown in the top-middle of Figure 8.13.

The $(q^k_{h,s})_{s\in\Lambda}$ is displayed in bottom-left of Figure 8.13. The values of $(q^k_{h,s})$ for two points $s = (3, 3)$, $(5, 6)$ are nearly 1.0. Except these two points, the values of $(q^k_{h,s})$ for all the other points are nearly zero. And we have,

$$\sum_{s\in\Lambda} q^k_{h,s} = 2.0011.$$

Roughly speaking, this means that there are two atoms relative to ϕ_h appearing in Y^k. The histogram of $q^k_{h,s}$ is shown in the bottom-right image of Figure 8.13. This image clearly shows that except a very small number (actually, 2) of s, all the other $q^k_{h,s}$ is almost zero.

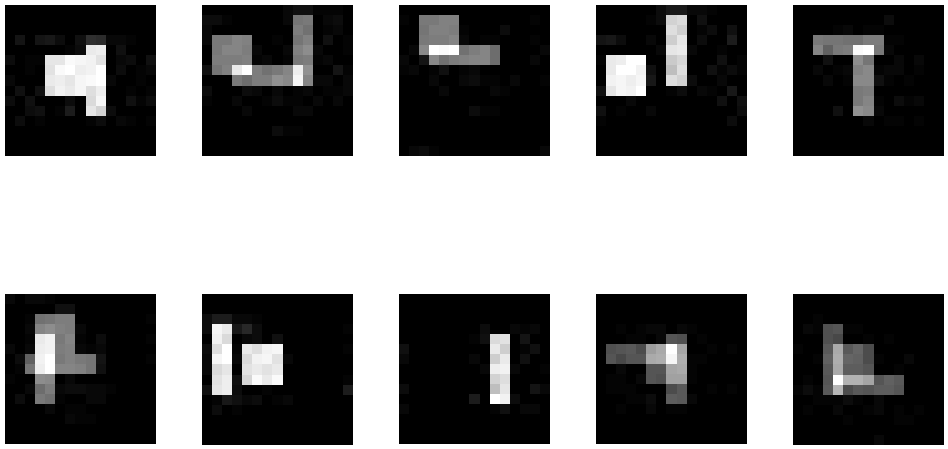

Figure 8.8: Reconstruction image for simple structure experiment by mean field.

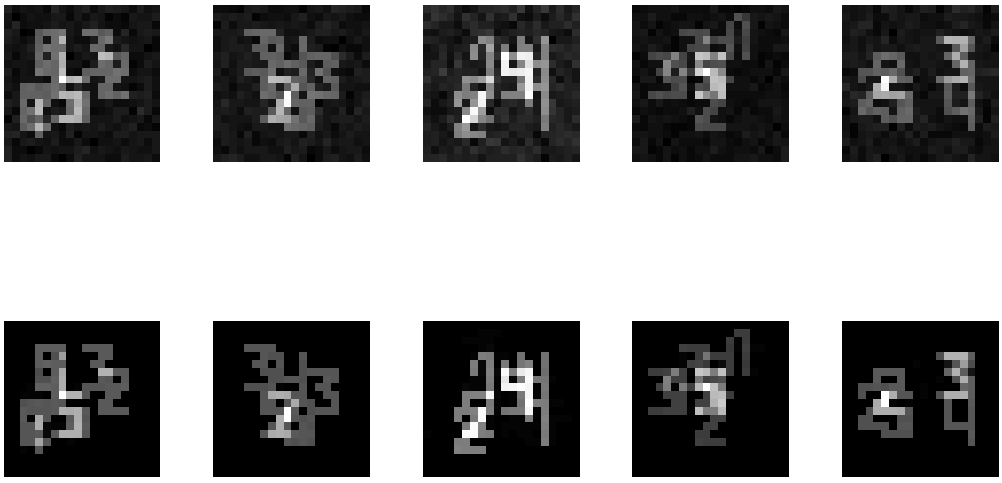

Figure 8.9: Top: typical images of training set for $q = 5$; bottom, reconstruction images.

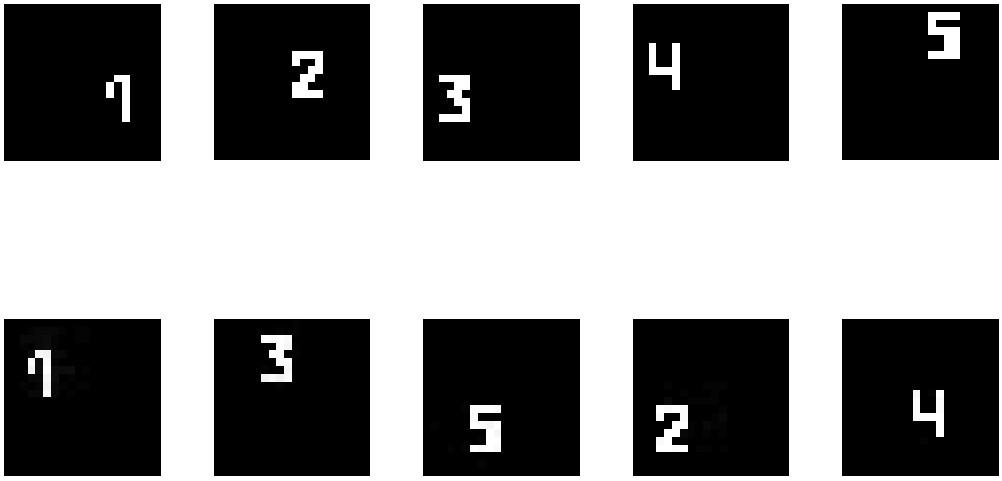

Figure 8.10: Real dictionary and the learning dictionary (support part) of $q = 5$.

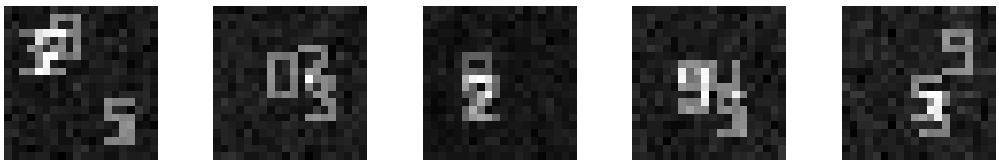

Figure 8.11: Typical images of training set for $q = 10$

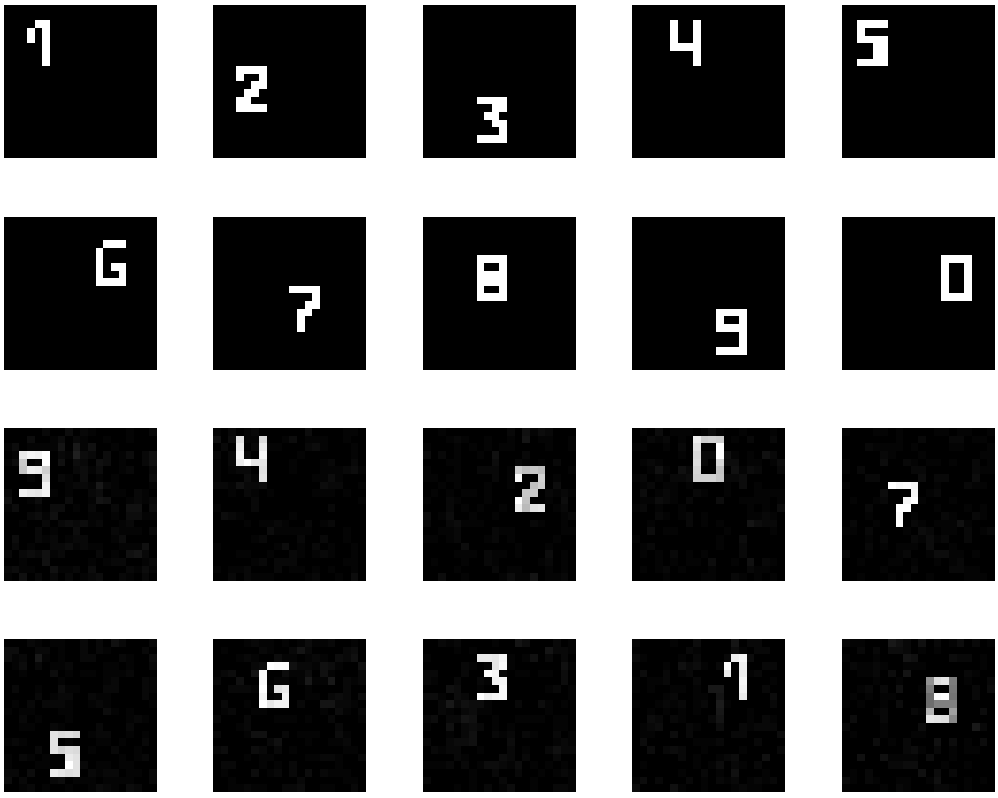

Figure 8.12: Real dictionary (top 10 images) and learned dictionary (bottom 10 images) for $q = 10$.

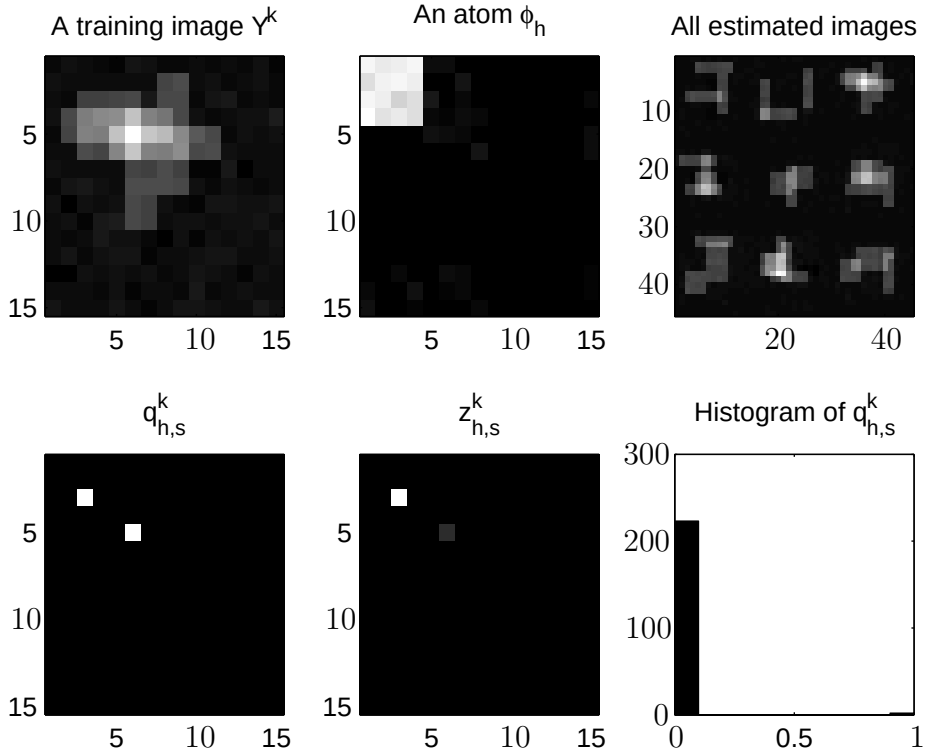

Figure 8.13: Analysis for hidden variables

Now for $z^k_{h,s}$, we have

$$z^k_{h,s} = q^k_{h,s} m^k_{h,s}.$$

Except

$$z^k_{h,s=(3,3)} = 0.9922, z^k_{h,s=(5,6)} = 0.6675,$$

the other values of $z^k_{h,s}$ are nearly zero. Hence, we have a correct detection of the ϕ_h at position $s = (3, 3)$. But for position $s = (5, 6)$, this is not very clear. It could be regarded as a false alarm though the pattern ϕ_h does occur in that position. But a more probable interpretation is that this ϕ_h is formed by the union of 2 rectangles of 2×7, 2 rectangles of 7×2.

8.7.5 Learning patterns from natural image

In this experiment, we use a piece of Barbara image. The original image is shown as the left image of Figure 8.15. The size of this image is 64×64 and it is the only element in the training set. We want to learn $q = 8$ typical patterns appearing in this image.

The initial dictionary is shown in the left of Figure 8.14. These 8 atoms are the diagonal elements of Figure 8.14 (DCT dictionary). Since these 8 atoms contains various frequency information, we think that this may be a good choice for representation . We set the flag FEqProb of the main algorithm (Table 8.3) as true to force the appearance of 8 elements.

After the mean field algorithm, the learned dictionary is shown as right image of Figure

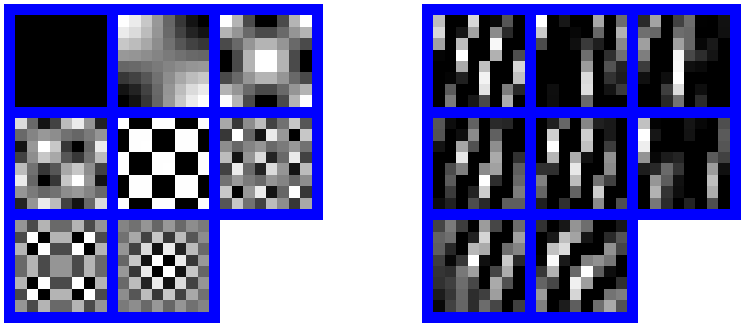

Figure 8.14: Dictionaries: left: initial dictionary; right: learned dictionary from the left image of Figure 8.15.

Figure 8.15: Left: original image; MP result(1141 terms) with special DCT dictionary, $PSNR = 21.4602$; reconstruction image of mean field (containing 1141 terms), $PSNR = 25.7237$.

8.14. The reconstruction image (there is no background for the experiment presented here)

$$\sum_{h=1}^{8} \sum_{s \in \Lambda} m_{h,s} q_{h,s} \phi_{h,s}$$

is shown as the right image of Figure 8.15. This reconstruction image has 1141 no zero-terms (more precisely, if the absolute value of $m_{h,s}q_{h,s}$ is greater than 0.000001, we say that $m_{h,s}q_{h,s}\phi_{h,s}$ is a non-zero term).

The $PSNR$ of the reconstruction image to the original image (left image of Figure 8.15) is 25.7237. The middle image of Figure 8.15 is the 1141-terms matching pursuit result of the original image with the dictionary displayed in the left image of Figure 8.14. The $PSNR$ of this image is 21.4602. From Figure 8.15 and 8.14, we clearly see that the typical patterns (texture) of the original are learned via our mean field approach.

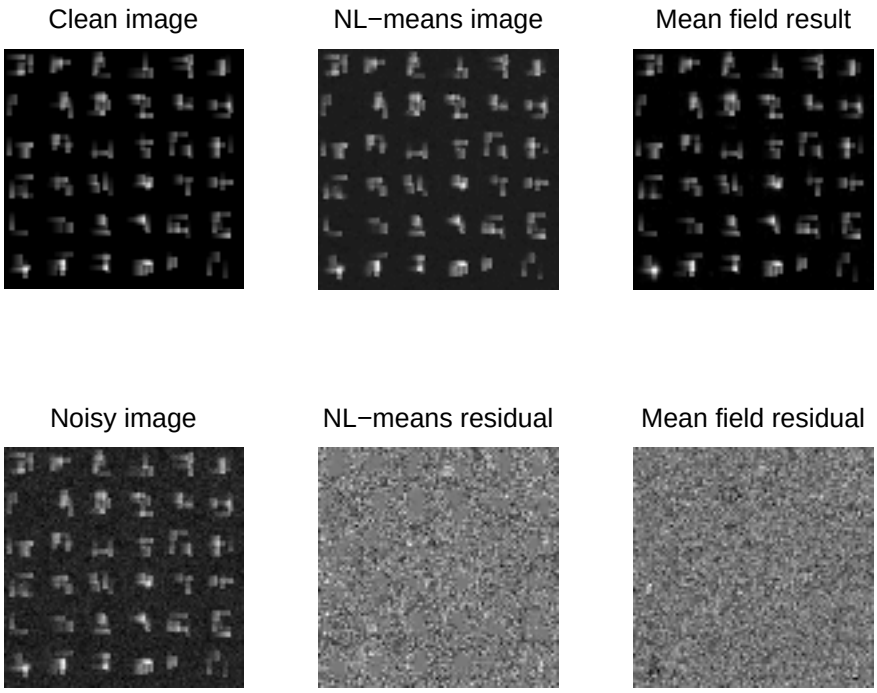

Figure 8.16: Denoising performances: top-left, clean image; bottom-left, noisy image ($\sigma = 0.2$, $PSNR = 27.5296$); top-middle: NL-means denoising result ($PSNR = 33.6332$); bottom-middle, NL-means denoising residual; top-right: result of mean field approach ($PSNR = 34.7421$), bottom-right: mean field residual

8.7.6 Experiment: by-product of denoising

In this experiment, the training set contains 36 small images of 15×15. We merge these images to form a 90×90 image. Figure 8.16 demonstrates the result. The top-left is the clean image, the top-right is Gaussian noisy image with standard deviation $\sigma = 0.2$. The $PSNR$ of this image is 27.5296; The middle-left is the result of NL-means denoising which is obtained by various tests to best choice of parameters for this method. The $PSNR$ of this image is 33.6332, the middle-right is the residual for NL-means denoising result; The bottom-left is the result of mean field approach with $PSNR$ equal to 34.7421, the bottom-right is the residual for the mean field method.

In order to see the difference, we zoom out a small patch ($\{[1, 15] \times [1, 15]\}$) of the Figure 8.16 to Figure 8.17. From the latter figure, we clearly see that the NL-means is fairly good for the background but its denoising performance on the structure itself is not as good as our mean field approach.

Moreover, we can learn the atoms which are served to represent the images in the training set via mean field approach. The top 3 images of Figure 8.18 shows the atoms in the true dictionary. After mean field, we can learn these atoms. The learned result are shown as the bottom 3 images of Figure 8.18.

The true σ in this experiment is 0.2, the learned σ is 0.2150.

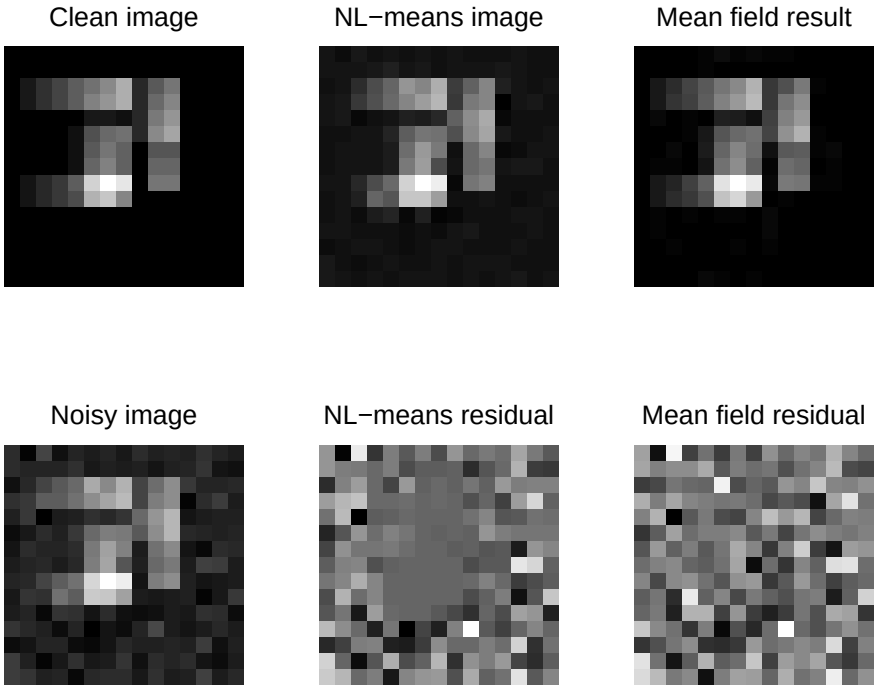

Figure 8.17: Zoom on a zone of Figure 8.16: top-left, clean image; bottom-left, noisy image ($\sigma = 0.2$, $PSNR = 24.3946$); top-middle: NL-means denoising result ($PSNR = 29.5700$); bottom-middle, NL-means denoising residual; top-right: result of mean field approach ($PSNR = 34.4510$), bottom-right: mean field residual

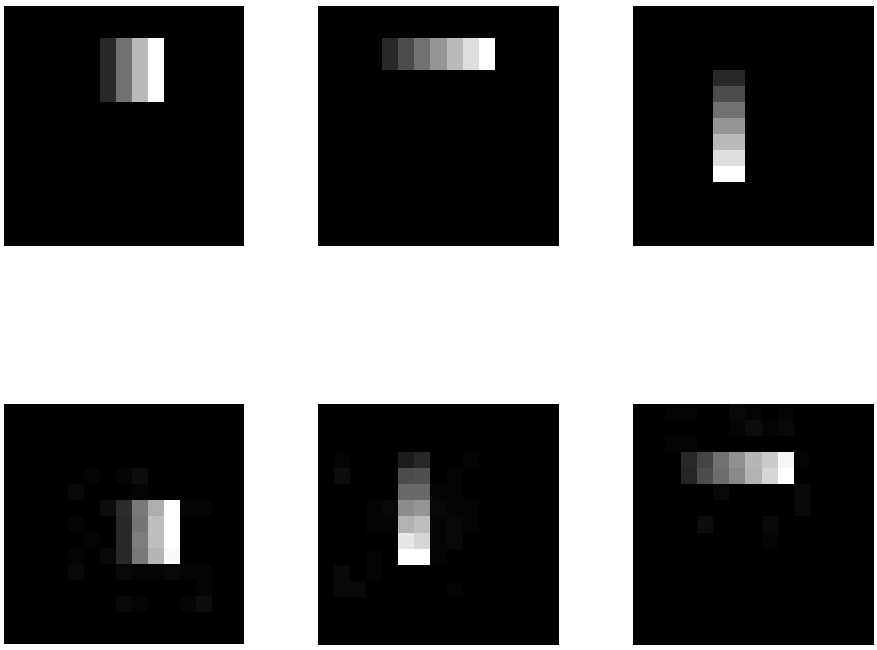

Figure 8.18: Dictionary: top: the 3 atoms of the true dictionary; the 3 atoms of learned dictionary.

8.8 Conclusion

This chapter explored finally the dictionary learning problem. The developed point of view is to regard this problem as a parameter estimation problem in a family of additive generative models. The introduction of random on/off switches of Bernoulli activating or deactivating each element of a translation invariant dictionary to be estimated allows the identification under rather general conditions in particular if the coefficients are Gaussian. By using an EM variational technic and the approximation of the posteriori distribution by mean field, we derived from a estimation principle by maximum likelihood a new effective algorithm of dictionary learning which one can connect for certain aspects with algorithm K-SVD. The experimental results on synthetic data illustrated the possibility of a correct identification of a source dictionary and several applications in image decomposition and image denoising.

Chapter 9

Conclusion and Discussion

We have concentrated our efforts on various aspects of some important variational models which use dictionary in the same time. The contributions of this dissertation can be divided into three parts: $TV - l^\infty$ model, sparse representation and dictionary learning. The concept of dictionary and sparseness connects tightly all these three parts.

9.1 Part I: $TV - l^\infty$ model

The first part contains the study on the $TV - l^\infty$ model. We considered this model on three aspects (Chapter 2, 3, 4).

In Chapter 2, we have proposed 12 Gabor dictionaries for the $TV - l^\infty$ model. All these dictionaries are translation-invariant. The main conclusion for this chapter is: to obtain good results of denoising with this model, the dictionary must represent the curvature of textures well. Hence, when we use Gabor dictionary, it is better to use Gabor filters whose supports are isotropic (or almost isotropic). In fact, for represent the curvature of a texture with a given frequency and living on a fixed support Ω, it is necessary that the support, in space, of Gabor filters allows a paving with few elements for the support Ω. For a general class of images, the support Ω is independent with the frequency of texture, it is reasonable to choose Gaobr filters whose supports are isotropic. This is a strong argument in favor of the wavelet packets dictionary, which allows in addition to have several sizes of supports in space (for a given frequency) for which the $TV - l^\infty$ model can be solved quickly.

In Chapter 3, in order to understand further the mechanism of $TV - l^\infty$ model, we presented the experiments where the dictionary contains the curvatures of known forms (letters). The data-fidelity term of the model authorizes the appearance in the residue $w^* - v$ of all the structures, except the forms being used to build the dictionary. Thus, we can expect that these forms remain in the result w^* and the other structures will disappear. Our experiments are carried on a problem of sources separation and confirm this impression. The starting image contains letters (known) on a very structured background (an image). We showed that it is possible, with the $TV - l^\infty$ model, to obtain a reasonable separation of these structures. Finally this work illustrated clearly that the dictionary $\mathcal{D}$ must contain the *curvature* of elements which we seek to preserve and not the elements themselves, as we might think naively. Moreover, the discussion of this chapter also implied that, in order to decrease the interactive effects among the atoms in the dictionary, the representation of the dictionary for the curvature of the underlying image should be sparse. This point of view links to the second part of the dissertation: sparse

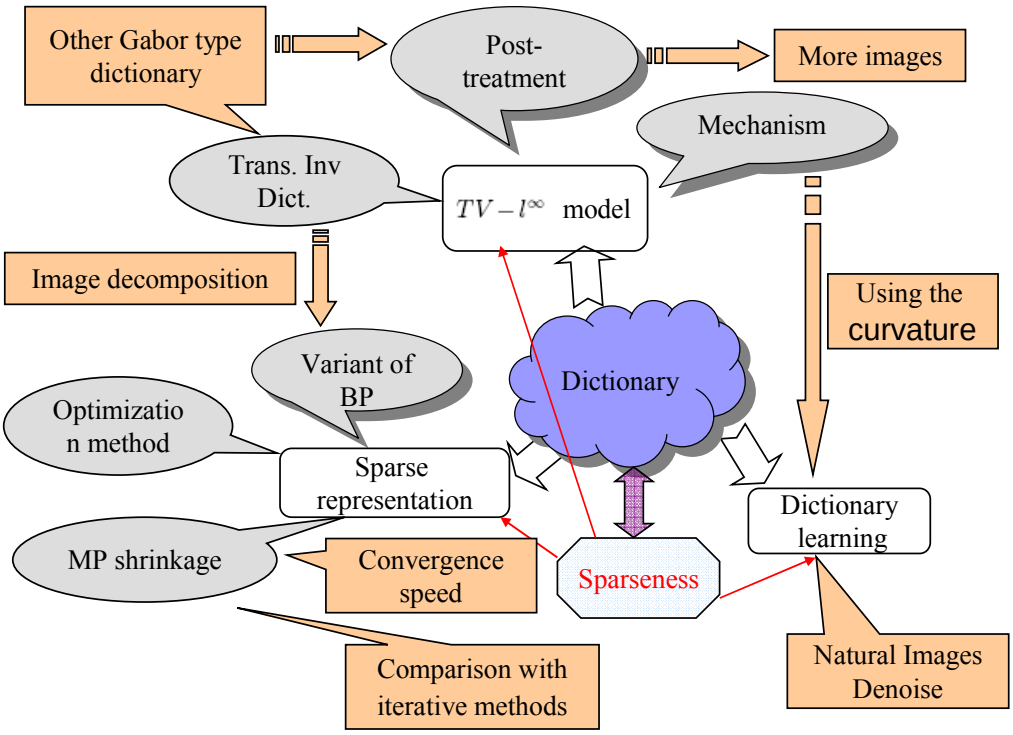

Figure 9.1: Idea of the dissertation

representation.

Chapter 4 presented a work in which we tried to integrate the K-SVD method with the $TV - l^\infty$ model. Our starting idea was to use the fact that some iterations of the algorithm which we use to solve the $TV-l^\infty$ model allow to reconstruct the lost structures from the image which we used as the initialization of the algorithm (and whose curvature is present in dictionary). We thus applied some of these iterations to the result of K-SVD and recovered lost textures well. This allows a visual gain and an improvement of the $PSNR$.

9.2 Part II: sparse representation

[1]

The second part of this dissertation is devoted to the problem of searching a sparse representation for a given image v (in presence of noise) in a certain dictionary fixed $\mathcal{D}$. It also can divided into 3 aspects (Chapter 5, 6, 7).

In Chapter 5, we exposed a numerical schema to solve a variant of Basis Pursuit. This consists to apply a proximal point algorithm to the new variant model. The interest is to transform a non-differentiable convex problem to a sequence (quickly converging) of very regular convex problem. We showed the theoretical convergence of the algorithm. This one was confirmed by the experiment. This algorithm allows to improve remarkably the quality (in term of sparseness) of the solution compared to the state-of-the-art concerning the practical resolution of Basis Pursuit. This algorithm might have a consequent impact in this rapidly developing field.

In chapter 6, we adapted to the cases of a variational model, whose regularization term is that of Basis Pursuit and whose data-fidelity term is that of the $TV-l^\infty$ model, a result of D. Donoho. This result showed that, under a condition relating the dictionary defining

the regularization term to the dictionary defining the data-fidelity term, it is possible to extend the results of D. Donoho to the models which interest us in this chapter. The obtained result says that, if the given data is very sparse, the solution of the model is close to its most sparse decomposition. This guarantee the stability of this model within this framework and establishes a link between l^1 and l^0 regularization, for this type of data-fidelity term.

Chapter 7 contains the study of a variant of Matching Pursuit. In this variant, we proposed to reduce the scalar product with the element best correlated with the residue, before modifying the residue. This is for a general threshold function. By using simple properties of these threshold functions, we showed that the algorithm thus obtained converges towards the orthogonal projection of the data on linear space generated by the dictionary (the whole modulo an approximation quantified by the characteristics of the threshold function). Finally, under a weak assumption on the threshold function (for example the hard-threshold satisfies this assumption), this algorithm converges in a finite time which one can deduce from the properties of the threshold function. Typically, this algorithm might be useful to make the orthogonal projections in the algorithm "Orthogonal Matching Pursuit". This we have not done yet.

9.3 Part III: dictionary learning

The third part of this dissertation is to explore finally the dictionary learning problem (Chapter 8).

The developed point of view of Chapter 8 is to regard this problem as a parameter estimation problem in a family of additive generative models. The introduction of random on/off switches of Bernoulli activating or deactivating each element of a translation invariant dictionary to be estimated allows the identification under rather general conditions in particular if the coefficients are Gaussian. By using an EM variational technic and the approximation of the posteriori distribution by mean field, we derived from a estimation principle by maximum likelihood a new effective algorithm of dictionary learning which one can connect for certain aspects with algorithm K-SVD. The experimental results on synthetic data illustrated the possibility of a correct identification of a source dictionary and several applications in image decomposition and image denoising.

9.4 Future works

There are many possibilities for the future works. For instance:

- Other Gabor-type dictionary for the $TV - l^\infty$ model
  For example, the author of [47] studied an anisotropic Gabor dictionary based on a generating function:
  $$g(x,y) = \frac{2}{\sqrt{3\pi}}(4x^2 - 2)e^{-(x^2+y^2)},$$
  where $[x, y]$ is the vector of discrete image coordinates and $\|g\| = 1$. The choice of the Gaussian envelope was motivated by the optimal joint spatial and frequency localization of this kernel. The authors there showed that when using for coding by Matching Pursuit, the anisotropic atoms are more efficient than the isotropic version (see Figure 7.16 of [47]). We think that it might be interesting to adopt

similar version of Gabor dictionary for the $TV - l^\infty$ model and then we can use this kind of dictionary for the post-treatment of K-SVD on more images.

- Using the Basis Pursuit variant model for image processing
  For example, for a noisy image v, we can serve certain Gabor dictionary for the Basis Pursuit model to get a sparse representation. Then we regard the part represented by the low-pass filters as cartoon part and take the high-pass filters part as texture part. This might lead an efficient method to decompose the image as cartoon part, texture part or even a noisy part.

- Continue to study on the MP shrinkage
  We can study the MP shrinkage in a more general way and study the speed of convergence and other important proprieties. For instance, the authors of [74] proved the exponential convergence of Matching Pursuit in quasi-incoherent dictionaries. For MP shrinkage, similar results are also expected as the parameter τ strictly positif might imply a faster convergence. The comparison of MP shrinkage with the iterative methods (see for instance, [22]) is also very interesting.

- Continue the study of statistical models.
  For instance, we might prove the consistence of the EM-MCMC method. The mean field method or other variational approach to solve the BEM model will also be interesting. Learning some typical patterns from the curvature of natural image and then using these patterns to construct a dictionary for the $TV - l^\infty$ model might also be a possible choice.

Overall, the whole idea of this dissertation is displayed in Figure 9.1.

Bibliography

[1] I.M. Gelfand and S.V Fomin, *Calculus of Variations*, Dover Publ., 2000.

[2] J. Jost and X. Li-Jost, *Calculus of Variations*, Cambridge University Press, 1998.

[3] Rakhi Motwani Mukesh Motwani, Mukesh Gadiya and Jr Frederick C. Harris, Eds., *A Survey of Image Denoising Techniques*, Santa Clara Convention Center, Santa Clara, CA, September 27-30 2004. GSPx 2004.

[4] T. Buades, B. Coll, and J.M. Morel, "A review of denoising algorithms, with a new one," *SIAM, Multiscale modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[5] D.L. Donoho and I.M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[6] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, Boston, 1998.

[7] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[8] Y. Meyer, *Oscillating paterns in image processing and in some nonlinear evolution equation*, AMS, Boston, MA, USA, 2001, The Fifteenth Dean Jacqueline B. Lewis Memorial Lectures.

[9] F.Malgouyres, "Minimizing the total variation under a model convex constraint for image restoration," *IEEE, trans. On Image Processing,*, vol. 11(12)), pp. 1450–1456, Dec.2002.

[10] E.Candes and F.Guo, "New multiscale transforms, minimun total variation synthesis: application to edge regularization in image compression," *Signal Processing*, pp. 82(11):1519–1543, 2002.

[11] F. Malgouyres, "Mathematical analysis of a model which combines total variation and wavelet for image restoration," *Journal of information processes*, vol. 2, no. 1, pp. 1–10, 2002, available at http://www.math.univ-paris13.fr/∼malgouy.

[12] S.Lintner and F.Malgouyres, "Solving a variational image restoration model which involves l^∞ contraints.," *Inverse Problem*, vol. 20(3), pp. 815–831, June 2004.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.

[14] J-L. Starck, M. Elad, and D.L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE, Trans. on Image Processing*, vol. 14, no. 10, pp. 1570–1582, October 2005.

[15] M. Brown and N.P. Costen, "Exploratory basis pursuit classification," *Pattern Recognition Letters*, vol. 26, pp. 1907–1915, 2005.

[16] B. Matalon, M. Elad, and M. Zibulevsky, "Image denoising with the contourlet transform," in *Proceedings of SPARSE'05*, Rennes, France, November 2005.

[17] M. J. Wainwright J. Portilla, V. Strela and E. P. Simoncelli, "Image enoising using scale mixtures of gaussians in the wavelet domain," *IEEE Trans. Image Process*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

[18] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representation in the presence of noise," *IEEE, Trans. on Information Theory*, vol. 52, pp. 6–18, 2006.

[19] D. Donoho and J. Tanner, "Sparse nonegative solution of underdetermined linear equations by linear programming," Tech. Rep. 2005-06, Stanford University, April 2005.

[20] F. Malgouyres, "Rank related properties for basis pursuit and total variation regularization," Tech. Rep. ccsd-00020801, CCSD, March 2006, Accepted to Signal Processing (minor modifications).

[21] F. Malgouyres, "Projecting onto a polytope simplifies data distributions," Tech. Rep. 2006-1, University Paris 13, January 2006.

[22] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problem with sparsity constraint," *Communication on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, Aug. 2004.

[23] P. L. Combettes and J.-C. Pesquet, "Proximal thresholding algorithm for minimization over orthonormal bases," *SIAM Journal on Optimization*, to appear.

[24] J. Bect, L. Blanc-Féraud, G. Aubert, and A. Chambolle, "A l1-unified variational framework for image restoration," *Lecture notes in Computer Science*, 2004, Proc. ECCV 2004.

[25] M.Figueiredo and R.Nowak, "A bound optimization approach to wavelet-based image deconvolution," in *ICIP*, 2005, vol. 2, pp. 782–785.

[26] M. Figueiredo and R. Nowak, "An em algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, 2003.

[27] S.Sardy, A.G. Bruce, and P.Tseng, "Block coordinate relaxation methods for nonparametric wavelet denoising," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 361–379, June 2000.

[28] M. Elad, "Why simple shrinkage is still relevant for redundant transforms," *IEEE, Trans. on Information theory*, vol. 52, no. 12, pp. 5559–5569, Dec. 2006.

[29] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Journal on Applied and Computational Harmonic Analysis*, 2007, To appear.

[30] S. Maria and J.J. Fuchs, "Application of the global matched filter to stap data: an efficient algorithmic approach," in *Proceedings of ICASSP 2006*, Toulouse, France, May 2006, vol. 4, pp. 1013–1016.

[31] D.Donoho and Y.Tsaig, "Fast solution of l1 - norm minimization problems when the solution may be sparse," Tech. Rep. 2006-18, Stanford University, dept of statistics, Oct. 2006.

[32] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE, Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, December 1993.

[33] S. Qian, D. Chen, and K.Chen, "Signal approximation via data-adaptive normalized gaussian function and its applications for speech processing," in *ICASSP-1992*, March 23-26 1992, pp. 141–144.

[34] Pati Y. C., Rezaiifar R., and Krishnaprasad P. S, "Orthogonal matching pursuit : Recursive function approximation with applications to wavelet decomposition," in *Proc. of 27th Asimolar Conf. on Signals, Systems and Computers*, Los Alamitos, 1993.

[35] Kreutz-Delgado, Murray JF K, Rao BD, Engan K, Lee TW, and Sejnowski TJ, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, no. 2, pp. 349–396, Feb 2003.

[36] P. Jost, P. Vandergheynst, S. Lesage, and R. Gribonval, "Motif : an efficient algorithm for learning translation invariant dictionaries," in *Int. Conf. Acoust. Speech Signal Process. (ICASSP'06)*, Toulouse, France, May 2006, IEEE.

[37] M. Aharon, M. Elad, and A. Bruckstein, "The k-svd, an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE, Trans. on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[38] M. Elad and M. Aharon, "Image denoising via sparse and redundant representation over learned dictionnary," *IEEE, Trans. on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[39] J.Mairal, M.Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. on Image Processing*, to appear.

[40] J.F. Aujol, G. Aubert, L. Blanc-Ferraud, and A. Chambolle, "Image decomposition into a bounded variation component and an oscillating component," *Journal of Mathematical Imaging and Vision*, vol. 22, no. 1, pp. 71–88, Jan. 2005.

[41] J.-L. Starck, M. Elad, and D.L. Donoho, "Image decomposition via the combination of sparse representation and a variational approach.," *IEEE Trans. on Image Processing*, vol. 14(10), pp. 1570–1582, 2005.

[42] T. Zeng and F. Malgouyres, "Using gabor dictionaries in a $tv - l^\infty$ model, for denoising," in *Proceedings of ICASSP 2006*, Toulouse, France, May 2006, vol. 2, pp. 865–868.

[43] S. Lintner and F. Malgouyres, "Solving a variational image restoration model which involves l^∞ contraints," *Inverse Problem*, vol. 20, no. 3, pp. 815–831, June 2004.

[44] R.Acar and C.Vogel, "Analysis of bounded variation methods for ill-posed problems," *Inverse Problems,*, vol. vol.10, pp. pp.1217–1229, 1994.

[45] J.F. Aujol, G. Gilboa, and S Osher, "Structure-texture image decomposition - modeling, algorithms and parameter section," CAM report 05-10, UCLA, 2005.

[46] M Turner, "Texture discrimination by gabor functions," *Biological Cybernetics*, pp. 71–82, 55(1986).

[47] Rosa Maria, *Sparse image approximation with application to flexible image coding*, Ph.D. thesis, École Polytechnique Fédérale de Laussanne, THÈSE No.3284(2005).

[48] J.S-Taylor N.Cristianini, *Support Vector Machines and other kernel-based learning methods*, Cambridge University Press, 2000.

[49] R.T. Rockafellar, *Convex analysis*, Princeton University Press, 1970.

[50] F.Malgouyres, "Increase in the resolution of digital images: Variational theory and applications," *Ph.D. thesis, Ecole Normale Supérieure de Cachan, Cachan, France,*, 2000.

[51] D. Donoho, "Neighborly polytopes and sparse solution of underdetermined linear equations," Tech. Rep. 2005-04, Dept of Statistics, Stanford University, January 2005.

[52] R.T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM, J. Control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[53] O.Güler, "On the convergence of the proximal point algorithm for convex minimization," *SIAM, J. Control and optimization*, vol. 29, 1991.

[54] P. Ciarlet, *Introduction to numerical linear algebra and optimisation*, Cambridge University Press, 1989.

[55] C.Lemarechal and C.Sagastizabal, "Practial aspects of the moreau-yoshida regularization 1 : theoretical properties," *SIAM, J. optimization*, 1997.

[56] Y. Nesterov, *Introductory lectures on convex optimization : A basic course*, Kluwer Academic Publishers, 2004.

[57] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, second edition, 2003.

[58] J.M.Borwein and W.B.Moors, "Essentially strictly differentiable lipschitz functions," Tech. Rep., CECM, 95-029.

[59] J.M.Borwein and W.B.Moors, "A chain rule for essentially strictly differentiable lipschitz functions," Tech. Rep., CECM, 96-057.

[60] David L.Donoho, Michael Elad, and Vladimir Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *Information Theory, IEEE Transactions on*, vol. 52, no. 1, pp. 6–18, Jan.2006, http://www-stat.stanford.edu/ donoho/Reports/2004/StableSparse-Donoho-etal.pdf.

[61] Hong ye Gao, "Wavelet shrinkage denoising using the non-negative garrote," *Journal of Computational and Graphical Statistics*, vol. 7, no. 4, pp. 469–488, 1998.

[62] H.-Y. Gao and A. G. Bruce, "Waveshrink with firm shrinkage," *Statistica Sinica*, pp. 855–874, 7(1997).

[63] ZHI-DONG ZHAO, "Wavelet shrinkage denoising by generalized threshold function," in *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, 18-21 August 2005.

[64] P.J. Huber, "Projection pursuit," *Ann. Statist.*, vol. 13, no. 435-525, 1985.

[65] A.W. van der Vaart, *Asymptotic Statistics(Cambridge Series in Statistical and Probabilistic Mathematics) (Paperback)*, Cambridge Press, 2000.

[66] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, B*, vol. 39, pp. 1–38, 1977.

[67] N. D. Lawrence, *Variational Inference in Probabilistic Models*, Ph.D. thesis, Computer Laboratory, University of Cambridge, New Museums Site, Pembroke Street, Cambridge, CB2 3QG,U.K., 2000.

[68] Stéphanie Allassonnière, Yali Amit, and Alain Trouvé, "Towards a coherent statistical framework for dense deformable template estimation," *Journal of the Royal Statistical Society, Series B*, vol. 69, no. 1, pp. 3–29, 2007.

[69] G. C. G. Wei and M. A. Tanner, "A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms," *Journal of the American Statistical Association*, vol. 85, no. 411, pp. 699–704, 1990.

[70] Luc Devroye, *Nonuniform Random Variate Generation*, Springer Verlag, 1986.

[71] C. P. Robert and G. Cassela, *Monte Carlo Statistical Method*, Springer Verlag, 2004, second edition.

[72] Estelle Kuhn and Marc Lavielle, "Coupling a stochastic approximation version of EM with an MCMC procedure," *ESAIM PS*, vol. 8, pp. 115–131, 2004.

[73] Z. Jordan, M.I. Ghahramani, T.S. Jaakkola, and L.K. Saul, *Learning in Graphical Models*, chapter An introduction to variational methods for graphical models, pp. 105–162, Kluwer, 1998.

[74] Rmi Gribonval and Pierre Vandergheynst, "On the exponential convergence of matching pursuit in quasi-incoherent dictionaries," *IEEE Trans. Information Theory*, vol. 52, no. 1, pp. 255–261, Jan. 2006.