ℓ based sparse representation

Yuan Liu

To cite this version:

Yuan Liu. ℓ based sparse representation. Machine Learning [cs.LG]. Normandie Université, 2019. English. NNT : 2019NORMIR22 . tel-02925022

HAL Id: tel-02925022
https://theses.hal.science/tel-02925022

Submitted on 28 Aug 2020

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité Informatique

Préparée au sein de INSA ROUEN NORMANDIE

Représentation parcimonieuse basée sur la norme ℓ_0

ℓ_0 based sparse representation

Présentée et soutenue par

YUAN LIU

Thèse soutenue publiquement le 25/11/2019
devant le jury composé de

M. Marcel MONGEAU,	Professeur de l'ENAC, Toulouse	Rapporteur
M. David BRIE,	Professeur de l'Université de Lorraine, Nancy	Rapporteur
Mme Ahlame DOUZAL,	Maître de Conférences (HDR) de l'Université Grenoble Alpes, Grenoble	Examinatrice
Mme Su RUAN,	Professeur de l'Université de Rouen, Normandie	Examinatrice
M. Stéphane CANU,	Professeur de l'INSA-Rouen, Normandie	Directeur de thèse
M. Paul HONEINE,	Professeur de l'Université de Rouen, Normandie	Codirecteur de thèse

Thèse dirigée par STÉPHANE CANU et PAUL HONEINE

INSA ROUEN NORMANDIE

MI∫IΣ

litis

Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mes recherches et qui m'ont aidée lors de la rédaction de ma thèse. Ils sont mes directeurs, mes collègues, et mes amis. Même avec mon français limité, je leur exprime mes grands mercis.

Je voudrais dans un premier temps remercier mes directeurs, monsieur Stéphane Canu, monsieur Paul Honeine et madame Su Ruan. Je remercie infiniment Stéphane. Chaque fois où j'ai rencontré des difficultés dans mes études et dans ma vie, il m'enseigne et m'aide à chercher la solution toujours avec patience. Il est savant dans le domaine scientifique et de la lecture qui est un bon exemple pour moi. Avec lui, j'apprends beaucoup. Je remercie sincèrement Paul. Il m'a aidé dès les premiers jours où je suis venue en France, à Troyes et à Rouen. Il est strict dans le travail et fait avancer fréquemment ma thèse. Grâce à lui, je peux résoudre plus facilement des difficultés et donc avancer bien dans ma thèse. Il est toujours souriant et patient. Je remercie beaucoup Su. Elle me propose des conseils pour l'amélioration des travaux surtout dans la part d'expérience et aussi des aides dans la vie. Elle est toujours gentille pour m'écouter et me donner des suggestions pour étudier mieux en France. En un mot, Je remercie tous mes directeurs de profiter de temps à m'aider dans ma thèse.

Je remercie également mes collègues, Linlin Jia, Imad Rida, Ismaila Seck, Cyprien, Rachel et Soufiane Belharbi. Ils m'aident à s'adapter dans la vie française. L'échange des idées avec eux me fait connaître les cultures diverses. En plus, comme on parle de ses propres travaux, on peut connaître des recherches de sujets différents et quelque fois, nous avons des inspirations grâce a la communication quotidienne.

Je tiens à remercier spécialement madame Brigitte et madame Sandra pour toutes leurs aides. Elles m'aident beaucoup à traiter des affaires de l'école et elles me font connaître le fonctionnement de l'école.

Je remercie mon copain Junhao Cao et ma famille qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

Enfin, je remercie infiniment CSC (Chinese Scholarship Council) qui m'offre la source financière. Avec ses supports, j'ai l'opportunité de faire mes recherches de thèse et finalement la réussir.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Résumé

Les méthodes parcimonieuses suscitent un vif intérêt pour l'estimation, de par leur capacité à proposer automatiquement un modèle interprétable ayant un très bon pouvoir prédictif. Ces méthodes se formulent comme des problèmes d'optimisation bi-critères où l'on cherche à obtenir un modèle minimisant simultanément un terme d'attache aux données et une pénalité visant à promouvoir la parcimonie via la sélection de caractéristiques pertinentes. Ce type de problème est aussi référencé sous le nom de codage parcimonieux.

Dans ce cadre, les données disponibles sont représentées par une matrice, considérée comme un dictionnaire, dont chaque colonne représente une caractéristique encore appelée atome. Le choix de ce dictionnaire joue un rôle très important et sa détermination est donc primordiale. Certaines études préconisent l'utilisation d'un dictionnaire prédéfini, par exemple à l'aide de polynômes ou d'ondelettes. Mais récemment, pour différentes applications, il s'est avéré plus efficace d'apprendre un dictionnaire dédié plutôt que d'utiliser un dictionnaire prédéfini. C'est le cas notamment en traitement du signal et des images, ainsi qu'en vision et en reconnaissance des formes.

Cette monographie traite de l'étude d'algorithmes d'apprentissage de dictionnaire, lorsque la parcimonie est favorisée par l'utilisation de la norme ℓ_0, c'est-à-dire par le contrôle explicite du nombre de caractéristiques à considérer. Elle est organisée en cinq chapitres. Le premier chapitre donne une brève introduction sur le contexte et les motivations du travail. Les contributions de la thèse et le plan du manuscrit sont également présentés.

Le deuxième chapitre porte sur la théorie de l'apprentissage de dictionnaire parcimonieux et notamment sur les problématiques d'optimisation bi-critères associées. Trois différents choix de pénalités pour le codage parcimonieux sont considérés : une mesure de comptage encore appelée pseudo-norme ℓ_0 (ou norme par abus de langage), la norme ℓ_1 et les normes ℓ_p avec $p > 1$ ou $0 < p < 1$. L'intérêt de la

norme ℓ_0 est qu'elle permet de compter et donc de contrôler explicitement le nombre de composantes d'un modèle. Ainsi, à performance de prédiction analogue, les modèles résultant de l'usage de la norme ℓ_0 sont plus parcimonieux que ceux obtenus en utilisant les autres normes ℓ_1 ou ℓ_p $(0 < p < 1$ et $1 < p)$. Cependant, les problèmes d'optimisation associés à l'usage de la norme ℓ_0 sont combinatoires, non convexes, non différentiables et NP-difficiles. Pour ces raisons, il sont généralement considérés comme difficiles à résoudre et ne passant pas à l'échelle.

Le troisième chapitre donne un état de l'art sur les méthodes permettant de résoudre le problème de codage parcimonieux pour l'apprentissage de dictionnaire. Il commence par présenter trois stratégies fréquemment considérées. La première est basée sur l'utilisation d'un algorithme glouton qui a donné les méthodes de poursuite et ses variantes (*matching pursuit* et *orthogonal matching pursuit*) pour obtenir une bonne approximation de la solution optimale globale du problème. La deuxième porte sur la relaxation de la norme ℓ_0 via l'utilisation de la norme ℓ_1 et les algorithmes associés comme le LASSO ou la « poursuite de base » (*Basis Pursuit*). La troisième utilise des méthodes du gradient, comme les méthodes de seuillage ou la méthode du gradient proximal.

Ce chapitre se poursuit par une présentation du problème d'apprentissage de dictionnaire qui consiste à estimer conjointement un dictionnaire pertinent et les coefficients pondérant ces atomes. Le problème d'optimisation associé est à la fois non convexe et NP-difficile. La manière classique d'aborder ce problème est d'utiliser une procédure de relaxation alternée en deux phases à la Gauss Seidel : 1) la phase de codage parcimonieux (*sparse coding*), qui consiste à estimer les coefficients de pondération en supposant le dictionnaire connu; 2) la phase de réactualisation de dictionnaire, qui consiste à estimer le dictionnaire en supposant cette fois les coefficients de pondération connus. La dernière partie de ce chapitre présente les principaux algorithmes existants permettant de résoudre le problème d'apprentissage de dictionnaire associé à la norme ℓ_0 : MOD (*Method of Optimal Direction*), K-SVD (*K- Singular Value Decomposition*) et SOUPDIL (*Sum of OUter Products Dictionary Learning*).

Le quatrième chapitre introduit une nouvelle méthode d'apprentissage de dictionnaire utilisant la norme ℓ_0. Comme le problème d'optimisation associé à cette norme ℓ_0 est NP-difficile, les stratégies développées dans la littérature sont basées sur des approximations ne permettant d'obtenir que des solutions locales. Notre idée est de développer une approche alternative permettant d'obtenir la solution globale du problème en un temps raisonnable. Pour ce faire, nous proposons de reformuler le problème du codage parcimonieux comme un programme quadratique mixte en nombres entiers (MIQP) et d'utiliser un logiciel d'optimisation, comme par exemple Gurobi ou Cplex, pour obtenir cet optimum global. La principale difficulté de cette approche étant le temps de calcul, nous proposons deux méthodes permettant de la surmonter : l'ajout de contraintes complémentaires accélérant la convergence de l'algorithme et son initialisation par un minimum local obtenu grâce à une méthode du premier ordre de type gradient proximal. La méthode ainsi obtenue, nommée MIQP accéléré (AcMIQP) permet de diminuer significativement les temps de calculs de la procédure d'optimisation et d'augmenter la taille des problèmes pouvant être ainsi traités. Notre méthode d'apprentissage de dictionnaire AcMIQP a été appliquée sur un problème de débruitage d'images démontrant sa faisabilité et sa pertinence. Les résultats montent que notre méthode est plus performante que les méthodes de références, comme le gradient proximal et l'algorithme K-SVD, et atteint des résultats comparables à ceux de SOUPDIL.

Un autre intérêt de la formulation quadratique mixte en nombres entiers est sa flexibilité. Il est facile d'introduire dans le problème d'optimisation MIQP de nouvelles contraintes pourvues qu'elles soient linéaires. Tirant profit de cet avantage, le cinquième chapitre traite de l'apprentissage de dictionnaire à faible cohérence, c'est-à-dire à faible corrélation entre ses atomes. Si, nous l'avons vu, la norme ℓ_0 permet le contrôle du niveau de parcimonie, elle ne dit rien en revanche sur la cohérence de ce dictionnaire. Or, plusieurs études théoriques ont montré la pertinence, pour un dictionnaire, d'avoir une faible cohérence. Les méthodes existantes qui s'attaquent à ce problème reposent sur la relaxation des contraintes, par exemple en ajoutant une étape de décorrélation à chaque itération.

Nous proposons une alternative originale, basée sur les travaux du chapitre précédent, visant à résoudre ce problème d'apprentissage de dictionnaire à faible cohérence en intégrant explicitement des contraintes favorisant l'indépendance des atomes. L'estimation du dictionnaire sous contraintes d'incohérence est abordée en combinant la méthode du lagrangien augmenté (ADMM) et la méthode du gradient proximal alterné étendu (*Extended Proximal Alternating Linearized Minimization*, EPALM), adaptée à des familles de problèmes non convexes. L'efficacité de la méthode AcMIQP+EPALM est démontrée sur une application de reconstruction d'image.

Finalement, nous conclurons nos travaux et donnerons des perspectives de notre recherche.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> “ *Scientists must use the simplest means of arriving at their results and exclude everything not perceived by the senses.* ”
>
> Ernst Mach

Sommaire

Today, people live in a world surrounded by diverse data. By improving smartphone technologies with cameras of higher and higher resolution or by alleviating user-generated content in the so-called data web such as Facebook and YouTube, large volume data of high dimension is produced every day. It makes a lot of demands on the storage capacity of the device, and moreover, to deal with these data, the high computation complexity is foreseen. However, it is noticed that in natural signals, namely audio, image, video, text and document, the number of information that makes sense is very small comparing to the whole signal, that is to say, the underlying model is sparse. The property of sparseness makes it possible to find a representation quite sparse replacing the original signal. As a result, the work on the sparse representation, rather than the signal, can greatly reduce computing complexity and decrease needs on storage capacity, which makes the research on sparse representation crucial with real benefits.

Sparse representation aims at representing a signal as linear combination, of which most coefficients are zeros, over a set of elementary signals where the set is called dictionary and the elementary signals are the atoms. The problem is often raised to find signals' regularity in signal processing or to extract images' features for tasks in pattern recognition [SMF15]. When the given dictionary is a basis, the problem involves solving a matrix equation, which is easy by introducing the matrix inverse. However, dictionaries with orthogonal atoms are not often able to take the most of sparse representations and signal reconstruction. Therefore, in most cases, the dictionary is learned as a redundant matrix, for which the resolution above is no more available. In fact, with a redundant dictionary, a solution to the problem may not be unique. Then, many problems will be raised: What is the condition for the uniqueness of the solution with sparsity constraint on the composition coefficients? How to handle the problem of simultaneously keeping the signal reconstructed to a great extent and adapting it to certain task? Even today, the studies on these problems are still challenging and open.

1.1 Evolution in sparse representation

During the past decades, sparse representation has attracted numerous research attentions. Originally, it is inspired by the decomposition of a studied signal over a well-defined basis in signal processing. In the beginning, the Fourier transform projects the signal in a basis formed by sine and cosine functions of different frequencies [BB86]. Thus, the signal defined in the time space can be analyzed in the frequency space. For the signal with simple composition in the frequency space, its coefficients before most basis functions are zeros, namely the representation of the signal in the frequency space is sparse. However, the Fourier analysis is merely on the frequency space, ignoring the information on time space. The natural signals such as music (audio) and video contain information both in time and frequency space. To deal with signals as music, a time-frequency dictionary by windowed Fourier transform is designed with success to analyze the variation of frequency over time, but it is limited to the signals with the same time and frequency resolution. The wavelets are thus proposed with the purpose of overcoming this limitation [Mal99].

The wavelet transform, considered to be time-frequency representation of sig-

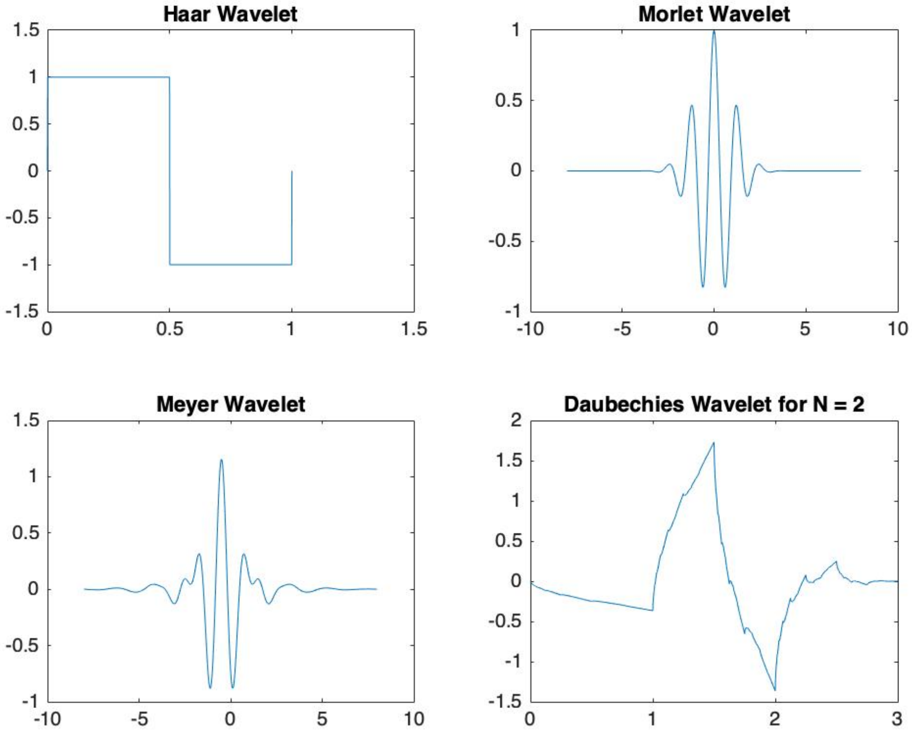


Figure 1.1 – Example of wavelet functions

nals [Gra95, Chu16], has played an important role in the development of sparse representation theory. Different from the Fourier transform defining the basis with known functions, the design of a basis is crucial in wavelet transform. To be precise, a prototype is defined by introducing "mother wavelet" $\Phi(x)$ [Mey95, CH97, Dau88] (Figure 1.1 shows 4 examples of a mother wavelet function). Then, an orthonormal bases $\Phi(x)_{a,b}$ is able to be constructed by dilation and scaling, with a and b being the scale and position parameters respectively. With specially constructed wavelet basis, the wavelet transform is used to analyze signals with much more complex structure [Lee96, RBE10, SCD02]. Specifically, transient structure and singularities are revealed through the signals' representation on a wavelet basis [Mal99], that is, the large-amplitude wavelet coefficient is detected in the surrounding of transient structure and singularities [LPM00]. In image processing, it means the edge and texture features [Uns95]. Based on its properties, the wavelet transform allows to create sparse representations of signals. However, the wavelet basis is predefined and signal-independent, which makes it necessary to construct an orthonormal wavelet basis adapted to each category of signals and the task at hand. For example, the Gabor wavelet models the impulse responses of the visual cortex, curvelet frames for edge detection, bandlet transform designed to adapt to the geometric image regularity, see chapter 5 and chapter 12 in [Mal99] for details. The wavelet transform for sparse representation is widely developed and applied with success in compressed sensing [Lee96], image denoising [CYV00a, CYV00b], multiresolution image representation [DV05], classification [Uns95, CK93], etc. Nevertheless, the sparse representation capacity of small dictionaries such as orthonormal wavelet basis is limited. Moreover, there is no off-the-shelf mathematical model for a predefined dictionary to represent any signal, such as natural images that contain for example textures.

In 1996, by studying the properties of receptive field of simple cells in mammalian primary visual cortex, Bruno Olshausen and David Field proposed a learning algorithm for sparse representation of a natural image, which allows to learn an overcomplete dictionary from the image at hand [OF96]. Since then, the overcomplete dictionary learning has become a hot topic in machine learning, as it allows to outperform predefined dictionaries like orthonormal wavelet bases.

Dictionary learning is defined as an optimization problem with respect to two vectors of optimization, sparse code (i.e., the coefficients of the linear model) and dictionary, which makes the problem harder to solve. Generally, researchers take the strategy of reaching the optimal solution by iteratively processing two alternating steps:

- Sparse Coding. It concerns the optimization problem with respect to sparse code by supposing the dictionary is already known. This sub-problem can be diversely formulated by introducing different functions for sparsity controlling, e.g. the ℓ_0-norm function which is even non-smooth and non-convex. For solving this problem, numerous algorithms are designed such as matching pursuit [MZ93] and basic pursuit [CD95].

- Dictionary Updating. This step deals with the optimization problem with respect to the dictionary while keeping the sparse code fixed. Unlike the sub-problem of sparse coding, the sub-problem of renewing the dictionary is often convex and differentiable. The algorithms representative for updating dictionary contains, the Method of Optimal Direction (MOD) [EAH99] and the coordinate descent algorithm with each atom updated by exploiting the singular value decomposition (SVD) [MBPS10, EA06].

The algorithms of dictionary learning are then developed by combining the algorithm of sparse coding and that of dictionary updating, like K-SVD [AEB$^+$06, EA06], proximal method [JMOB10], and online dictionary learning method [MBPS09]. Furthermore, for certain tasks, the specified dictionary can be leaned by adding an appropriate term of regularization. For instance, a discriminative dictionary is learned by proposing the regularization of classification error [ZL10, MBP11], and an incoherent dictionary is learned via the introduction of constraints on the off-diagonal entries of Gram matrix of dictionary [LDHL17].

The sparse representation by learning a dictionary has thus gained great success in the field of signal processing, image processing, pattern recognition, and computer vision [MCW05a, MBP$^+$14]. Specifically, sparse representation with the learned dictionary is successfully used to accomplish tasks like image denoising, deblurring, inpainting, face recognition, visual tracking, and classification (see [MBP$^+$14] and therein).

1.2 Contributions of the research

This monograph concentrates on the study of dictionary learning for sparse representations. More precisely, it corresponds to the optimization problem of minimizing the reconstruction error with limiting the sparsity of decomposition coefficients. As is known, originally, the sparsity is explicitly expressed by using the ℓ_0-norm. This function can exactly control the sparsity level, however, with a shortcoming of non-convexity and non-differentiability, which makes it the obstacle to overcome. The existing methods for solving this problem can be roughly grouped in two major classes: Greedy algorithms attains the solution by iteratively providing suboptimal solutions, such as matching pursuit (MP) and its variants [Tro04, BRF11], or gradient descent based algorithms such as Iterative Hard Thresholding (IHT) algorithm [GK09] or proximal method [BJQS14]. The second class corresponds to methods that relax the ℓ_0-norm by replacing it with the ℓ_1-norm, which is still nonsmooth but convex and continuous. This leads to a classical problem, often called the problem of LASSO [MY09] or Basis Pursuit [CD95] (technically, LASSO or Basis pursuit principle are the methods for solving the ℓ_1-norm based sparse representation problem). However, now in the research community, it is called directly LASSO problem or Basis pursuit problem [MBP$^+$14]). This relaxation simplifies the problem and makes it possible to transform the original problem into a certain standard optimization problem so that one can exploit optimization techniques such as the simplex method [NW06]. However, the ℓ_1-norm cannot always guarantee the required sparsity level [AGH18] and even the sparse representation is produced but with shrinkage [DDDM04].

When referring to the problem of dictionary learning, as aforementioned, it corresponds to two vectors of optimization and the strategy is to solve it by iteratively alternating two steps, sparse coding and dictionary updating. This will raise the difficulty. The purpose of our research is thus to find a method to learn the dictionary by solving exactly the ℓ_0-norm based problem. The main contributions of our work can be summarized as follows.

We work on the ℓ_0-norm constrained dictionary learning problem. For reaching the optimal solution, we take the frequently used strategy of iterative processing two alternating steps: sparse coding and dictionary updating. Specifically, in the step of sparse coding, we reformulate the original problem as a problem of mixed integer quadratic programming (MIQP) and solve it by the optimization method for MIQP problem, without using any greedy algorithm, or relaxation. The second step takes advantage of the coordinate descent algorithm (precisely, each atom is updated by SVD). Hence, it is the first time that the dictionary is learned with MIQP for sparse coding.

We propose two techniques for accelerating the algorithm of exact sparse coding. One is to offer an initialization when solving the MIQP problem. This value obtained by using the proximal method is considered to be an approximation of the optimal solution of the original problem. The other is to relax the problem by achieving a convex envelop of the region defined by all the constraints. With the help of these two acceleration methods, the MIQP is not limited to dealing with synthetic data problems but real data. For the demonstration, the accelerated MIQP (AcMIQP) is applied for denoising well-known images (*Barbara, Lena* etc). Furthermore, the results show the superiority of the dictionary learning algorithm with exact sparse coding method AcMIQP comparing to the methods of proximal and OMP in processing the image with high noise level.

Furthermore, we study incoherent dictionary learning. The coherence is regarded as one important quality of dictionary [Tro04]. The uniqueness of the solution of the problem of sparse representation is proved to rely on the coherence of dictionary [Ela10], and even some (greedy) algorithms can be precisely executed under the condition of incoherence of dictionary [Tro04]. Untill now, the methods for learning incoherent dictionary are either by adding a decorrelation step following the dictionary update, such as INK-SVD [MBP12], or by formulating the optimization problem by introducing a regularization term on the dictionary coherence [RLS09]. Unlike these methods, we seek to solve the exact incoherent dictionary learning problem, with explicit constraints on the dictionary and the ℓ_0-norm sparse code. The corresponding dictionary learning algorithm operates two steps, as in the classical problem, sparse coding and dictionary updating at each iteration. In this problem, the sparse coding is solved exactly with the AcMIQP method. For updating the dictionary, we exploit methods of the augmented Lagrangian and the proximal alternating linearized minimization. Moreover, the convergence of the algorithm is guaranteed.

The incoherent dictionary learning algorithm is used for estimating the relationship of image reconstruction with coherence of dictionary. It is worthy noting that with the proposed incoherent dictionary learning algorithm, the best results are obtained comparing with the methods INK-SVD [MBP12] and the incoherent dictionary learning algorithm by iterative projection and rotation (IPR) [DM13]. Moreover, the fact is proved, to our best knowledge, for the first time, that an appropriate higher incoherence favors the image reconstruction.

1.3 Outline of the monograph

The monograph focuses on the study of dictionary learning with ℓ_0-norm for sparsity promoting. The rest of the monograph is organized as follow.

Chapter 2 introduces the mathematical framework of the sparse model. We talk about the three main functions ℓ_0-norm, ℓ_1-norm and ℓ_p-norm ($0 < p < 1$), which are frequently used to control the sparsity. By analyzing the three functions in ability of sparsity controlling, convexity and influence on the accuracy of sparse code, we show the interest of studies on the ℓ_0-norm based sparse representation problems.

Chapter 3 presents the state-of-the-art algorithms for the sparse representation. These algorithms are divided into two parts: algorithms for sparse coding and those for dictionary learning. Specifically, we detail greedy algorithms, relaxation methods, gradient descent, proximal method and other optimization methods.

Chapter 4 develops a dictionary learning method in the same spirit of the K-SVD algorithm. The point of innovation is that an exact optimization technique named mixed integer programming with quadratic objective function (MIQP) is proposed for the sparse coding phase. Furthermore, two optimization techniques are introduced in MIQP to accelerate the algorithm, which makes the algorithm feasible in real image processing, such as image denoising.

Chapter 5 focuses on the study of coherence of dictionary, which is one of the dictionary's important properties with proved theoretical results. Our work studies explicitly the relationship between coherence of dictionary and image reconstruction performance. For this purpose, we study the dictionary learning problem with a predefined coherence level and an exact ℓ_0-norm promoting sparsity. Then, an augmented Lagrangian based algorithm combined with proximal alternating linearized minimization is proposed. The model is finally applied in image reconstruction experiments, whose results confirm the theoretical conclusion.

Chapter 6 gives the conclusion and provides some future work.

Publication

The main contributions presented in this monograph can also be found in our publications:

Journal Papers

- Liu, Yuan, Canu, Stéphane, Honeine, Paul and Ruan, Su, **Mixed integer programming for sparse coding: Application to image denoising.** IEEE Transactions on Computational Imaging, vol. 5, no. 3, pp. 354-365, Sept. 2019.

- Liu, Yuan, Canu, Stéphane, Honeine, Paul and Ruan, Su, **Incoherent dictionary learning via mixed-integer programming and hybrid augmented lagrangian.** Submitted to Digital Signal Processing, Sept. 2019.

Conference Papers

- Liu, Yuan, Canu, Stéphane, Honeine, Paul and Ruan, Su, **Une véritable approche ℓ_0 pour l'apprentissage de dictionnaire.** Actes du 26ème colloque du GRETSI en traitement du signal et des images, Juan Les Pins, 2017.

- Liu, Yuan, Canu, Stéphane, Honeine, Paul and Ruan, Su, **K-SVD with a real ℓ_0 optimization: Application to image denoising.** Proc. of the 28th IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Aalborg, 2018

- Liu, Yuan, Canu, Stéphane, Honeine, Paul and Ruan, Su, **Apprentissage de dictionnaire faiblement cohérent par programmation quadratique mixte.** Actes du 27ème colloque du GRETSI en traitement du signal et des images, Lille, 2019.

Notations

In this section, we give some mathematical notations used in this monograph.

The data is considered to be real-valued. Matrices are denoted by an uppercase letter, vectors by a lowercase letter in bold and scalars by a lowercase letter. For example, $\mathrm{X} \in \mathbb{R}^{m\times n}$ denotes a matrix with each column represented by the indexed vector $\mathbf{x}_i$ with the index $i \in \{1,\dots,n\}$. The j^{th} entry of $\mathbf{x}_i$ is written as $x_{i,j}$. A subset of integer is represented by $\mathrm{SI} \subset \{1,\dots,n\}$, then a matrix X_{SI} is a submatrix whose columns are formed by the columns of X indexed by all elements in SI. The complement of SI is represented by SI^c. We use X^T to represent the transpose of X. If $m = n$ and X is non-singular, the inverse of X is denoted by X^{-1}. More generally, for X with $m \neq n$, X has its pseudo-inverse X^+, also known as Moore–Penrose inverse. $tr(\mathrm{X})$ indicates the trace of a square matrix X, which is equal to the sum of all entries on its main diagonal. The Frobenius norm of the matrix X is expressed as

$$\|\mathrm{X}\|_\mathrm{F}^2 = \sum_{i,j} x_{i,j}^2.$$

For two matrices $\mathrm{A} \in \mathbb{R}^{m\times n}$ and $\mathrm{B} \in \mathbb{R}^{m\times n}$, the Frobenius inner product of the two matrices is written as

$$\langle \mathrm{A}, \mathrm{B} \rangle = tr(\mathrm{A}^\mathrm{T}\mathrm{B}).$$

Given a function of X, denoted by $f : \mathbb{R}^{n\times m} \to \mathbb{R}$, its gradient represented by $\nabla_\mathrm{X} f$ is a matrix of size $n \times m$ where the entry in position (i, j) is calculated via

$$(\nabla_\mathrm{X} f)_{i,j} = \frac{\partial f}{\partial x_{i,j}}.$$

Similarly, a vector $\mathbf{x} \in \mathbb{R}^m$ is composed by m elements, its i^{th} entry is denoted by x_i. The ℓ_0-norm of $\mathbf{x}$, represented by $\|\mathbf{x}\|_0$, indicates the number of nonzero elements in $\mathbf{x}$. The ℓ_1-norm of $\mathbf{x}$ is the sum of its entries' absolute value, denoted

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{m} |x_i|.$$

The ℓ_p-norm with is defined by

$$\|\mathbf{x}\|_p = \Big(\sum_{i=1}^{m} x_i^p\Big)^{\frac{1}{p}}.$$

Specially, when $p = 2$, it is called Euclidean norm. For a function of vectors, written as $f : \mathbb{R}^m \to \mathbb{R}$, the gradient of the function is denoted by $\nabla_{\mathbf{x}} f$. If the function f is not differentibale at $\mathbf{x}$, we calculate the Fréchet subdifferential of f, represented by $\partial f(\mathbf{x})$ and defined as follows.

Definition 1. *([Kru03]) Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper lower semicontinuous function. For each $x \in dom\, f$ (where $dom\, f\; = \{x \in \mathbb{R}^n \mid f(x) < +\infty\}$), the Fréchet (or regular) subdifferential of f at x is*

$$\partial f(x) = \Big\{x^* \in \mathbb{X} \mid \liminf_{z \to x} \tfrac{f(z) - f(x) - \langle x^*, z - x\rangle}{\|z - x\|} \geq 0\Big\}, \tag{1.1}$$

where $\mathbb{X}$ denotes the topological dual space of$\mathbb{R}^n$.

In linear algebra, the restricted isometry property (RIP) characterizes matrices that are nearly orthonormal. Supposing D a matrix of size $n \times m$ and k an integer, the restricted isometry constant (RIC) of the matrix D is defined as the minimal value $\delta_k \in (0, 1)$ that

$$(1 - \delta_k)\|\mathbf{x}\|_2^2 \leq \|\mathrm{D}\mathbf{x}\|_2^2 \leq (1 + \delta_k)\|\mathbf{x}\|_2^2, \quad \forall \mathbf{x} \in \mathbb{R}^m \text{ and } \|\mathbf{x}\|_0 \leq k.$$

If the RIC δ_k exists, we say that the matrix D satisfies the k-restricted isometry property.

Chapter 2

Mathematical Framework of Sparse Representation

Sommaire

Given a signal $\mathbf{x} \in \mathbb{R}^n$ and a matrix $\mathrm{D} \in \mathbb{R}^{n\times m}$, providing a sparse representation consists in finding the solution $\boldsymbol{\alpha} \in \mathbb{R}^m$ of the system of linear equation with a regularization term that promotes sparsity, that is

$$\arg\min_{\boldsymbol{\alpha}} \ \Omega(\boldsymbol{\alpha}) \quad \text{subject to} \quad \mathbf{x} = \mathrm{D}\boldsymbol{\alpha}, \tag{2.1}$$

where $\Omega(\boldsymbol{\alpha})$ measures the degree of sparsity of the solution $\boldsymbol{\alpha}$ where smaller values of $\mathrm{R}(\boldsymbol{\alpha})$ indicate more sparse solution.

In practice, the dictionary $\mathrm{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_i, \ldots, \mathbf{d}_m]$ is defined in the set $\mathbb{D} = \{\mathrm{D} \mid \|\mathbf{d}_i\|_2 \ = 1,\ \forall i = 1, \ldots, m\}$, which allows to avoid too large values in $\boldsymbol{\alpha}$. Moreover, one must assure that the matrix D can span the entire space $\mathbb{R}^n$ to be sure of the existence of the solution to problem (2.1). In spite of that, it is still difficult to give the condition defining the uniqueness of its solution.

The choice of the function $\Omega(\boldsymbol{\alpha})$ is diverse. Some favor functions explicitly promoting sparsity, however, making the problem hard to deal with. Others operate relaxation by taking advantage of convex function to make the problem easy to solve. For instance, the ℓ_p-norm functions, especially for $p = 0$ and $p = 1$, are frequently studied [XZJ13, Tib96]. Additionally, the family of functions such as $\mathrm{R}(\boldsymbol{\alpha}) = \sum(1 - \exp(|\alpha_i|))$, $\Omega(\boldsymbol{\alpha}) = \sum \log(1 + |\alpha_i|)$ and $\Omega(\boldsymbol{\alpha}) = |\alpha_i|/(1 + |\alpha_i|)$ are also used to promote sparsity [Ela10]. These will be introduced in this chapter.

2.1 The ℓ_0 based sparse representation

2.1.1 Introduction of ℓ_0-norm function

The ℓ_0-norm function is defined based on the concept of limit, which is:

$$\|\boldsymbol{\alpha}\|_0 = \lim_{p \to 0} \|\boldsymbol{\alpha}\|_p^p = \lim_{p \to 0} \sum |\alpha_i|^p = \mathrm{card}(\boldsymbol{\alpha}), \tag{2.2}$$

where $\mathrm{card}(\boldsymbol{\alpha})$ indicates the number of non-zero entries in the vector. However, despite the name of 'norm' is given, the ℓ_0-norm is not a strict 'norm' function , since it does not satisfy all the three conditions: 1) the triangle inequality, 2) absolute homogeneity and 3) being positive definite. Indeed, the ℓ_0-norm function, defined from the domain of vectors $\mathbb{R}^m$ to the naturals $\mathbb{N}$, obeys the triangle inequality, that is, for any arbitrary two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^m$, we have

$$\|\mathbf{u} + \mathbf{v}\|_0 \leq \|\mathbf{u}\|_0 + \|\mathbf{v}\|_0.$$

The third condition of positive definiteness is satisfied as well. For a vector $\mathbf{v} \in \mathbb{R}^m$, if the equation $\|\mathbf{v}\|_0 = 0$ holds, then it implies that $\mathbf{v} = \mathbf{0}$. For the homogeneity condition, it is easy to see that it is not satisfied, because the equation $\|a\mathbf{v}\|_0 = |a|\|\mathbf{v}\|_0$ is not true for any scalar number $a \in \mathbb{R}$ and vector $\mathbf{v} \in \mathbb{R}^m$.

Even the ℓ_0-norm function is misleading in a sense, it presents powerful capacity in promoting sparsity. Figure 2.1 illustrates some ℓ_p-norm functions by setting $p =$

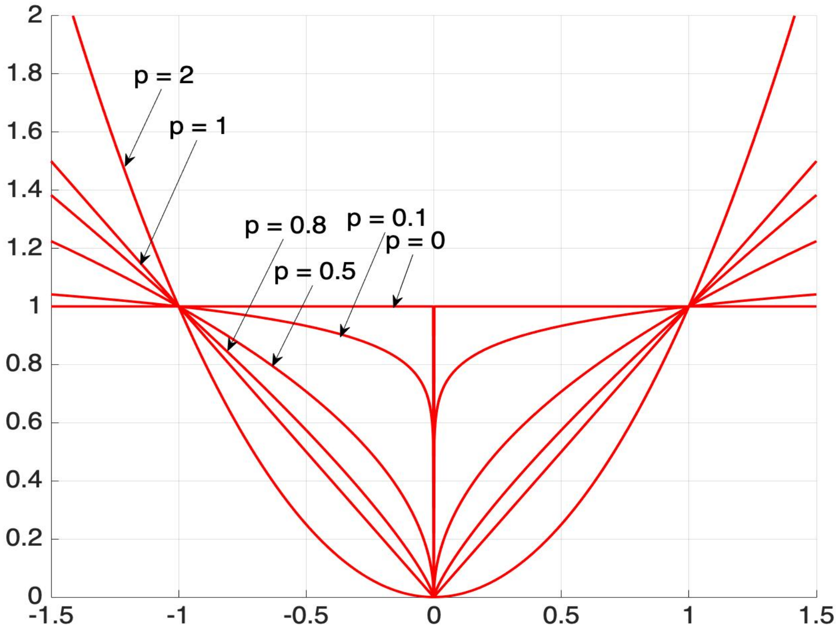


Figure 2.1 – Illustration of $\|\boldsymbol{\alpha}\|_p^p$ in one dimension with $p = 0,\ 0.1,\ 0.5,\ 0.8,\ 1,\ 2$

0, 0.1, 0.5, 0.8, 1, 2 respectively. It is obvious that, when $p > 1$, the function $\|\boldsymbol{\alpha}\|_p^p$ is convex and differentiable which makes the problem easy to solve, but loosing the sparsity. For $p = 1$, the function is convex but not differentiable at $\boldsymbol{\alpha} = \mathbf{0}$, making it of interest as discussed in Section 2.2 [Tib96, MBPS09, CD95]. When $0 < p < 1$, the function becomes non-convex but continuous, and the smaller the p is, the closer the norm will be to the ℓ_0-norm function. Therefore, an ℓ_p-norm function with $0 < p < 1$ can be used to approximate the ℓ_0-norm function [RSS10].

The ℓ_0-norm function, however, is non-convex and discontinuous. The optimization problem based on ℓ_0-norm is NP-hard [Tro04]. It is difficult to solve it exactly. On the other hand, the ℓ_0-norm explicitly measures the sparsity of the representation. Thus, for all ℓ_p-norm, the ℓ_0-norm shows the best performance in terms of sparsity control. Its proof is given in [Ela10] (see page 12 for details).

Despite the difficulty, a large number of researchers paid attention to the ℓ_0 based optimization problem [Nik13, AGH18, XZJ13]. The developments have been made both in theory and applications. Tropp analyzed greedy algorithms to solve ℓ_0 based sparse representation and summarized the sufficient condition for obtaining an optimal solution [Tro04]. Soubies et al. provided the sufficient and necessary condition for continuous function approximating ℓ_0-norm [SBFA17]. The ℓ_0

norm is, thus, utilized in image processing [XZJ13, EA06, MP06], signal processing [Mal99, LPM00], machine learning [BJQS14], etc.

2.1.2 The ℓ_0-norm function promoting sparsity

By setting the function $\Omega(\boldsymbol{\alpha}) = \|\boldsymbol{\alpha}\|_0$, the optimization problem (2.1) becomes:

$$\arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \mathbf{x} = \mathrm{D}\boldsymbol{\alpha}. \tag{2.3}$$

The constraint $\mathbf{x} = \mathrm{D}\boldsymbol{\alpha}$ gives a strict equality relation. In reality, caused by distortion or noise perturbation from the device or information loss during the transmission, $\mathbf{x}$ is not exactly the pure signal, but with some noise, which can be modeled:

$$\mathbf{x} = \mathrm{D}\boldsymbol{\alpha} + \mathbf{e}, \tag{2.4}$$

where $\mathbf{e} \in \mathbb{R}^m$ means the unfitness noise. Consequently, the problem (2.3) is reformulated by the quadratic inequality constraint rather than the equality constraint, that is,

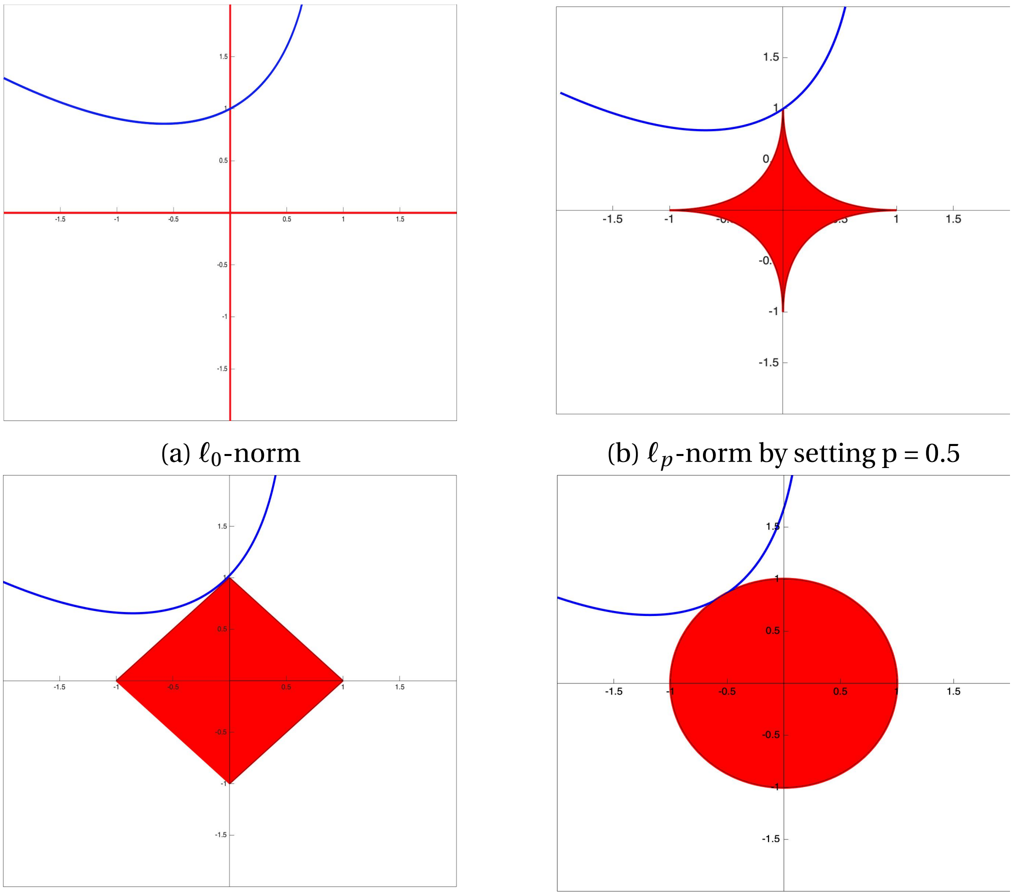
$$\arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{subject to } \|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 < \epsilon, \tag{2.5}$$

where ϵ is a parameter determining the tolerance of data fitting. Intrinsically, the value of ϵ depends on the noise level of the signals. This formulation that assumes knowing the noise level is usually used in image denoising [EA06], and seldom used in classification [FNZ+15]. Nevertheless, given a signal, ϵ seems hardly to be well estimated.

Another formulation consists in minimizing the reconstruct error in the feasible region defined by the ℓ_0-norm constraint,

$$\arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 \quad \text{subject to } \|\boldsymbol{\alpha}\|_0 \leq k, \tag{2.6}$$

where $k \in \mathbb{N}$ and $k \geq 1$ denotes the sparsity level. By setting in advance the sparsity level, this formulation helps to represent the signal by the linear combination of less than k atoms in D. By exploiting this model, the signal can be sparsest represented, which can be proved by visualization the ℓ_p-norm based sparse model. As is shown in Figure 2.2, the intersection of the objective function's curve (the parabola in blue) and the region determined by the unit-radius ℓ_p-ball (here the region in red) locates the solution of the problem mentioned above with ℓ_0-norm and other three



(a) ℓ_0-norm

(b) ℓ_p-norm by setting p = 0.5

(c) ℓ_1-norm

(d) ℓ_2-norm

Figure 2.2 – Illustration in 2D (i.e., $\boldsymbol{\alpha} \in \mathbb{R}^2$) of the minimization of the quadratic reconstruction error (parabola in blue) in the feasible region defined by the ℓ_p-norm constraint (ℓ_p-ball region in red) with sparsity level k = 1.

ℓ_p-norm (i.e., $p = 0.5,\ 1,\ 2$) promoting sparsity. It concludes that ℓ_0-based sparse model can reveal always the sparse solution. However, for the ℓ_p-norm with $p > 0$, the optimal solution risks to be dense. In other words, the solution of the problem with ℓ_p-norm (p > 0) can be sparse only under certain conditions. By investigating k-sparsity model in (2.6), researchers have addressed problems like matrix factorization [PP12], image recovery [CYV00a, EA06], and feature extraction for classification [HA07, Uns95].

We reformulate sparse representation problem by exploiting the Lagrangian function of the optimization problem (2.6), therefore producing the regularized formulation,

$$\underset{\boldsymbol{\alpha}}{\arg\min}\, \tfrac{1}{2}\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_0, \tag{2.7}$$

where $\lambda > 0$ is a hyperparameter balancing the minimization of the reconstruction error and the maximization of sparsity. Nikolova has proved the relationship between the k-sparsity problem (2.6) and its regularized form (2.7) [Nik16]. Specifically, when the choice of k corresponds to a λ in an interval (bounds depending on the reconstruction error decrease with the representation's sparsity), then the optimal solutions to the two formulations become equivalent. Compared with the k-sparsity problem (2.6), the regularized problem (2.7) and its variants are more frequently applied in image processing [MP06] and machine learning [BJQS14, YLY12].

2.1.3 Some approximations of the ℓ_0-norm function

As aforementioned, the non-convexity and discontinuity of the ℓ_0-norm cause the corresponding optimization problems to be NP-hard. One feasible method is to approximate the optimal solution by a convergent series emerging from an iterative process. Proposed resolution methods include the greedy algorithm [MZ93] and the proximal method [BJQS14].

Another idea is inspired by reformulating the optimization problem with a smooth function that is not exactly the ℓ_0-norm, but can exceedingly approach it, that is

$$\|\boldsymbol{\alpha}\|_0 \leftarrow \sum \phi(\alpha_i), \tag{2.8}$$

where $\phi : \mathbb{R} \to \mathbb{R}^+$ is a smooth function that can measure if the entry α_i is zero or not. The most important is that the optimal solution of this problem must coincides with the one reached when applying the ℓ_0-norm. We talk about some representative reformulations of ℓ_0-norm in the following.

The simplest proposition is to approximate ℓ_0-norm by ℓ_p-norm. By this thinking, F. Rinaldi et al. proposed two concave formulations tuned by some hyperparameters [RSS10]. The first formulation is

$$\phi_\kappa(\alpha_i) = (\alpha_i + \kappa)^p, \tag{2.9}$$

where $0 < p < 1$ and $0 < \kappa$ are two hyperparameters. A variant of (2.9), is

$$\phi_\kappa(\alpha_i) = -(\alpha_i + \kappa)^{-p}. \tag{2.10}$$

These two functions are proved to be equivalent to the ℓ_0-norm function when p

and k tend to zero, which is based on the fact that

$$\lim_{p\to 0,\kappa\to 0}\sum(\alpha_i+\kappa)^p = \lim_{p\to 0,\kappa\to 0} -\sum(\alpha_i+\kappa)^{-p} = \|\boldsymbol{\alpha}\|_0.$$

Thus the solution of the original problem can be reached by solving the reformulated problem, by applying classical optimization techniques, e.g. Frank-Wolfe algorithm [RSS10].

In [MBZJ07], Mohimani et al. introduced a function of zero-mean Gaussian family to estimate the ℓ_0-norm. The expression of the function is

$$\phi_\sigma(\alpha_i) = 1 - \exp(-\alpha_i^2/2\sigma^2), \tag{2.11}$$

where σ denotes a bandwidth parameter. This function is smooth and differentiable, which simplifies the problem; however, the effect of σ becomes significant. To overcome this issue, the proposed algorithm decreases iteratively the value of σ, allowing to exclude the dependency of the solution on the bandwidth value.

The reformulations (2.9), (2.10) and (2.11) are established by approximating the ℓ_0-norm with smooth functions. Soubies et al. proposed another formulation to define a continuous non-smooth and non-convex function [SBFA17]. The function is a convex envelope of the original objective function in the optimization problem (2.7), which is created by applying twice the Legendre-Fenchel transformation. Hence, the penalty at each coordinate $\lambda\|\alpha_i\|_0$ can be estimated by the so-called continuous exact ℓ_0 penalty

$$\phi(\|\mathbf{d}_i\|_2,\lambda,\alpha_i) = \lambda - \frac{\|\mathbf{d}_i\|_2^2}{2}\Big(|\alpha_i| - \frac{\sqrt{2\lambda}}{\|\mathbf{d}_i\|_2}\Big)^2 \delta_{\left\{|\alpha_i|\le\frac{\sqrt{2\lambda}}{\|\mathbf{d}_i\|_2}\right\}}(\alpha_i), \tag{2.12}$$

where, given a set $\mathbb{C}$, $\delta_{\mathbb{C}}$ represents the indicator function

$$\delta_{\mathbb{C}}(\alpha) = \begin{cases} 1 & \text{if } \alpha\in\mathbb{C} \\ 0 & \text{else.} \end{cases} \tag{2.13}$$

This formulation retains the equivalence of the global minimizer with the original problem. Moreover, the equivalence between critical minimizers of the two formulations is proved in Theorems 4.5, 4.7 and 4.8 in [SBFA17].

2.1.4 Exact reformulation of ℓ_0-norm with complementary constraints

Very recently, a new strategy [SIBD11, FMP$^+$18] inspired by the work of [KM82] is emerging, where the ℓ_0-norm is reformulated by introducing complementary constraints. Specifically, this strategy consists in indicating if an entry of the vector $\boldsymbol{\alpha}$ is zero or not by an auxiliary variable $\mathbf{z} \in \mathbb{R}^m$ with the definition

$$z_i = 1 \Longleftrightarrow \alpha_i = 0 \quad \forall i = 1, \ldots, m \tag{2.14}$$

where z_i means the i^{th} entry of $\mathbf{z}$. Feng et al. presented two formulations in [FMP$^+$18]. The full-complementary formulation is

$$\begin{aligned} &\|\boldsymbol{\alpha}\|_0 = \sum_{i=1}^{m} (1 - z_i) \\ \text{under constraints} \quad &\begin{cases} \boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^- \\ \mathbf{z}^{\mathrm{T}}(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-) = 0 \\ \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^- \geq 0 \quad \text{and} \quad 0 \leq \mathbf{z} \leq \mathbf{1}_m, \end{cases} \end{aligned} \tag{2.15}$$

where $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ are the non-negative and non-positive parts of $\boldsymbol{\alpha}$ respectively. The second formulation, called half-complementary, takes the same strategy by only changing the above constraints by

$$\begin{cases} 0 \leq \mathbf{z} \leq \mathbf{1}_m \\ \mathbf{z} \circ \boldsymbol{\alpha} = 0, \end{cases} \tag{2.16}$$

where $\bullet \circ \bullet$ denotes the Hadamard (entrywise) product, and the above equality constraint corresponds to the logical relation (2.14). This formulation can be further reformulated by introducing a large value M, which is called big-M reformulation [FMP$^+$18]. The resulting optimization problem is a mixed integer programme (MIP), which is the main topic of this monograph and will be discussed with detail in the monograph.

2.1.5 Existence and uniqueness of the solution

Hereinabove, we know what the ℓ_0-norm function is, how the problem is formulated with ℓ_0-norm function in sparse representation and some reformulations of the ℓ_0-norm optimization problem. There is a precondition of all these discussions, namely the existence, and more strictly, the uniqueness, of the solution to the problem, which gives sense to the study.

The existence of a solution to the undetermined problem (2.1) is guaranteed under the condition that D spans the feasible solution space. For the optimization problem with regularization (2.7) or the other two constrained formulations (2.6) and (2.5), it is clear that we can find a vector $\hat{\boldsymbol{\alpha}}$ in the feasible region with an objective value, denoted by $\mathrm{F}(\hat{\boldsymbol{\alpha}})$, that satisfies $\mathrm{F}(\hat{\boldsymbol{\alpha}}) < \infty$. Besides, the objective functions are all bounded below. Hence, there must be a solution to these sparse problems. Nikolova stated in Theorem 4.4 in [Nik13] that not only the optimization problem (2.7) has a global optimal solution, but also the optimal solution is unique.

The uniqueness of the solution depends on a concept named 'Spark' of the dictionary, which is defined as the smallest number of columns from D that are linearly dependent. Theorem 2.4 in [Ela10] (or Theorem 1.2 in [DH01]) states the uniqueness of the solution to the equality constrained sparse model:

Theorem 2.1.1. ***Uniqueness-Spark (Theorem 2.4 [Ela10])*** *If there is a solution* $\hat{\boldsymbol{\alpha}}$ *to problem* (2.1) *with* $\|\hat{\boldsymbol{\alpha}}\|_0 < Spark(\mathrm{D})/2$*, then* $\hat{\boldsymbol{\alpha}}$ *must be the sparsest one.*

The above theorem offers the sparsity condition for the uniqueness of the optimal solution. In fact, the Spark of a matrix is quite difficult to obtain, which restricts the usage of the conclusion. Nikolova provided in [Nik13] the proof of uniqueness of the minimizer by defining the conditions on D and $\mathbf{x}$. Its conclusion for unique global minimizer is described in Theorem 5.6 in [Nik13] (For the completeness of the condition expression, the detail is not listed here).

2.2 The ℓ_1 based sparse representation

The ℓ_1-norm function is a convex and continuous function, and is the closest convex approximation of the ℓ_0-norm, which explicitly expresses the sparsity.

There is no evidence in providing the relationship between ℓ_0-norm and ℓ_1-norm in sparsity promoting. However, ℓ_1-norm is confirmed, in adequate conditions detailed in [Don06, CRT06, ZY06], in resulting in a sparse solution. Donoho proved in [DH01, Don06] that, when the representation is considerably sparse, the ℓ_1-norm can reach the same solution obtained by exploiting ℓ_0-norm. Therefore, it is frequently used as a relaxation of the ℓ_0-norm.

The sparse representation problem developed by the ℓ_1-norm has three formulations, just like the ℓ_0-norm problem. The sparsity constrained formulation is written as

$$\arg\min \frac{1}{2}\|\mathbf{x}-\mathrm{D}\boldsymbol{\alpha}\|_2^2 \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_1 \leq \lambda_k, \tag{2.17}$$

where λ_k controls the sparsity. The reconstruction error constrained formulation is expressed by

$$\arg\min \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \|\mathbf{x}-\mathrm{D}\boldsymbol{\alpha}\|_2^2 \leq \epsilon. \tag{2.18}$$

By adopting the ℓ_1 regularization term, the sparse model is developed of formulation

$$\arg\min \frac{1}{2}\|\mathbf{x}-\mathrm{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{2.19}$$

where $\lambda \geq 0$ is the parameter governing the trade-off between the data-fitting and sparsity criteria. This formulation is obtained from the Lagrangian function of the constrained formulation. The formulations (2.17) and (2.19) are equivalent and can yield the same optimal solution, where the relationship between λ_k and λ is data dependent.

The ℓ_1-norm optimization problem (2.18) is also called Basic Pursuit (BP) problem [CDS01]. More precisely, the BP principle aims to tackle the problem (2.18) by transforming it into a linear programming problem. The two equivalent (2.19) and (2.17) optimization problem are also called Least Absolute Shrinkage and Selection Operator (LASSO) problem [Tib96], which iteratively solves the quadratic optimization problem.

A number of variants of the problem have been widely studied. The Bayesian LASSO [PC08] solves the ordinary LASSO problem by applying Bayesian principle, which improves the computing complexity, and the parameter λ or λ_k can be derived via marginal maximum likelihood. The so-called group LASSO considers the structure of the sparsity [RNCR15, ZHY$^+$17]. The adaptive LASSO distributes adaptive weights for penalizing different coefficients in the ℓ_1-norm [Zou06]. In addition, considerable effort has been out on the algorithm design, resulting in methods such as the root-finding algorithm by searching for a convex and continuously differentiable curve that traces the optimal trade-off between the least-square fit and the ℓ_1-norm of the solution [BEFP08], and the greedy coordinate descent algorithm [WL08]. More recently, Arnold [AT16] and Markopoulos [MKCP17] have studied the efficiency of the LASSO problem.

With the solid theoretically development on the ℓ_1-norm optimization problem, it has been extensively used for image processing [MY09, LBRN07], classification [RNCR15], object tracking [ZHY+17], face recognition [WYG+08], and even in combination with graphical models [MB+06, AT16].

2.3 The ℓ_p based sparse representation

The ℓ_p-norm function with $p < 1$, as aforementioned, is closer to the ℓ_0-norm, than ℓ_1-norm. This function shows some similar properties to ℓ_0-norm such as its non-convexity. Besides, it is non-differentiable at 0. As a consequence, minimizing it is a hard problem to solve. That is why the ℓ_p-norm when $p < 1$ is less popular than the case where $p = 0$ and 1.

The optimization problem with ℓ_p-norm for penalty can be easily formulated

$$\arg\min \tfrac{1}{2}\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_p, \tag{2.20}$$

where the trade-off parameter λ functions the same as that in the ℓ_0-norm or ℓ_1-norm problems. This problem has obviously a solution. Furthermore, it is proved that the optimal solution of the ℓ_p-norm based problem is equivalent to that of the ℓ_0-norm based problem [FM11] under the condition that $\|\boldsymbol{\alpha}\|_0$ is upper bounded by

$$\mathrm{U}l = \frac{f_1(\mathrm{D})^p f_2(\mathrm{D})}{1 + f_1(\mathrm{D})^p},$$

where f_2 is the function indicating the spark of D, and f_1 is the minimum of the $f_2(\mathrm{D})^{th}$ descent ordered NULL(D) [MCW04, MCW05a].

The ℓ_p-norm is usually used as an approximation to the ℓ_0-norm to make the problem tractable. Nevertheless, it risks of heavy computation cost due to its non-convexity and non-differentiability. To overcome these difficulties, another approximation is needed to reach the solution of the problem, such as

$$\|\boldsymbol{\alpha}\|_p^p = \sum(|\alpha|^2 + \epsilon)^{p/2}, \tag{2.21}$$

where $\epsilon \geq 0$ is a smoothing parameter [MCW05a]. With this approximation replacing the ℓ_p-norm, the original optimization problem (2.20) can be solved by half-quadratic regularization method or iterative method (see [MCW05a, CG15] and therein for details).

Besides the ℓ_p-norm with $p < 1$, some other norms are used, such as the ℓ_{2-1}-

norm used for face recognition [SYGL14] and feature selection [NHCD10]. The joint use of several norms has been also considered, such as the combination of the ℓ_0 and ℓ_1 penalties [LW07].

2.4 Conclusion

This chapter introduced the mathematical framework of sparse representations. We presented the most-known norm functions for sparsity promoting: ℓ_0-norm, ℓ_1-norm and ℓ_p-norm with $0 < p < 1$. The characteristics of each norm function were discussed. Of particular interest is the ℓ_0-norm, which counts the number of non-zeros of a vector and thus can strictly control the sparsity. Nevertheless, its undeniable shortcoming is that the function is non-smooth and non-convex, making the problem NP-hard. The ℓ_1-norm is regarded as a relaxation of the ℓ_0-norm for the sparsity-promoting. Unlike the ℓ_0-norm, this function is convex and differentiable at all values except at the zero. The price to pay is the risk of missing the optimal sparsest solution. The ℓ_p-norm with $0 < p < 1$ is closer to the ℓ_0-norm function than the ℓ_1-norm, thus it has stronger sparsity control ability than the latter; however, it is a concave function. For each norm function, we detailed the formulations of the optimization problems of sparse representation and also gave a problem transformation survey.

Next chapter presents algorithms for sparse representation learning.

Chapter 3

Sparse Representation Algorithms

Sommaire

In the previous chapter, we presented the mathematical formulations of sparse representation. It is revealed that the problem can be defined by inducing the sparsity with the ℓ_0-norm, ℓ_1-norm or ℓ_p-norm (with $0 < p < 1$) functions. Owing to its power in controlling explicitly the number of non-zero coefficients, the ℓ_0-norm is theoretically the first choice to build sparse representations. However, the ℓ_0-norm is non-smooth and non-convex, which makes the corresponding optimization problem NP-hard. To overcome the shortcoming of the ℓ_0-norm, it is usually replace with the ℓ_1-norm, which is the closest convex norm function to ℓ_0-norm. Moreover, it is proved that, in some conditions (e.g. when the sparse code is quite sparse), the optimal solution of the ℓ_1-norm based optimization problem coincides with that of the ℓ_0-norm based problem.

This chapter presents the state-of-the-art algorithms for sparse representation learning. It considers two major problems: sparse coding and dictionary learning.

Sparse coding is the keystone in sparse representation learning. It consists in estimating the sparse code (i.e., the coefficients of the linear model) while the dictionary is already known. This optimization problem can be formulated by introducing sparsity-promoting constraints, mainly the ℓ_0-norm and the ℓ_1-norm defined in the previous chapter. Since the 1990's, extensive research efforts have been made to address to solve the sparse coding optimization problem. Section 3.2 in this chapter presents the prime algorithms for sparse coding, such as greedy algorithms [Tro04] whose major representative is the Matching Pursuit [MZ93], relaxation methods with Basis Pursuit [MXAP11] and LASSO [Tib96], Gradient Descent algorithms [KKK08] and Iterative Thresholding methods [Kow14].

In Sections 3.3 and 3.4, we will focus on the problems of dictionary learning. In complement to estimating the sparse code as in sparse coding problems, dictionary learning aims at estimating also the optimal dictionary, thus improving the performance of the sparse representation. The dictionary learning aims at finding jointly the optimal solution of the sparse code and of the dictionary. The resulting optimization problem is non-convex and NP-hard, and is more difficult than solving the single sparse coding problem. To overcome this issue, a good resolution strategy is to iteratively alternating two steps: sparse coding and dictionary updating. The sparse coding problem, as aforementioned, handles the estimation of the decomposition coefficients with a fixed dictionary. While fixing these coefficients, dictionary updating aims at estimating the optimal dictionary. In its general form, the resulting optimization problem is a convex problem, which can be easily solved by algorithms such as the Method of Optimal Direction [EAH99] and the Coordinate Descent algorithm applied in K-SVD [AEB$^+$06, MBPS09]. With appropriate sparse coding algorithm and dictionary update algorithm, the state-of-the-art algorithms for dictionary learning are introducing in the following, with the K-SVD algorithm [AEB$^+$06], the proximal method [JMOB10], and the Sum of OUter Products Dictionary Learning (SOUPDIL) algorithm [RNF17].

3.1 Sparse coding and dictionary learning

Given a matrix $\mathrm{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_\ell] \in \mathbb{R}^{n\times\ell}$ of ℓ signals of dimension n, and a (predefined or learned) dictionary dictionary $\mathrm{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_m] \in \mathbb{R}^{n\times m}$ the problem of sparse representation consists in finding a matrix $\mathrm{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_\ell] \in \mathbb{R}^{m\times\ell}$ of decomposition coefficients that satisfies $\mathrm{X} \approx \mathrm{DA}$. The columns of the dictionary, *i.e.*, $\mathbf{d}_j$ for $j = 1, \ldots, m$, are called atoms.

In signal or image data processing, the set of signals is typically larger than its dimension, namely $\ell \gg n$. The sparse representation can be obtained by solving the following optimization problem:

$$\min_{\boldsymbol{\alpha}_i \in \mathbb{R}^m} \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\tfrac{1}{2} \|\mathbf{x}_i - \mathrm{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \Omega(\boldsymbol{\alpha}_i) \right). \tag{3.1}$$

The first term $\frac{1}{2}\|\mathbf{x}_i - \mathrm{D}\boldsymbol{\alpha}_i\|_2^2$ is the reconstruction error with $\|\,.\,\|_2$ being the Euclidean norm. The second one includes the regularization term $\Omega(\boldsymbol{\alpha}_i)$ to enforce sparsity. The regularization parameter $\lambda > 0$ controls the trade-off between data fitting and sparsity of $\boldsymbol{\alpha}$. For the sake of clarity of this monograph, the reconstruction error is measured with the square loss; generalization to other loss functions, such as the logistic and hinge losses, is straightforward [ŞE13]. Generally, the regularization function Ω is associated to a norm that promotes sparsity and its formulation depends on the task at hand [BJQS14, AEB+06]. The natural definition of Ω to promote sparsity is, as mentioned in Chapter 2, the ℓ_0-norm.

With the assumption of independence of the signals, the optimization problem (3.1) can be divided into ℓ independent subproblems

$$\min_{\boldsymbol{\alpha}_i \in \mathbb{R}^m} \tfrac{1}{2} \|\mathbf{x}_i - \mathrm{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \Omega(\boldsymbol{\alpha}_i). \tag{3.2}$$

When the ℓ_0-norm is adopted to promote the sparsity, this problem becomes non-convex, non-smooth, and thus NP-hard, as analyzed in Chapter 2.

The dictionary plays a crucial role in sparse representation [Tro04, Ela10]. Nevertheless, there is no predefined dictionary that can satisfy all tasks in processing natural signals and images. For all these reasons, learning the dictionary from the data is a main building block in sparse representation. And in general, D is an over-complete dictionary, that is to say $n < m$, while the situation $n > m$ is allowed for some discrimination tasks [MPS+09]. To prevent the ℓ_2-norm of dictionary's atoms from being arbitrarily large which leads to arbitrarily small decomposition coefficient in A, the dictionary D is supposed without loss of generality, to satisfy

$$\mathbb{D} = \left\{ \mathrm{D} \in \mathbb{R}^{n \times m} \mid \|\mathbf{d}_j\|_2 \leq 1, \; \forall j = 1, \ldots, m \right\}. \tag{3.3}$$

Beyond these considerations, the dictionary learning problem can be written in the following general form,

$$\min_{\boldsymbol{\alpha}_i \in \mathbb{R}^m, \mathrm{D} \in \mathbb{D}} \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\tfrac{1}{2} \|\mathbf{x}_i - \mathrm{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \Omega(\boldsymbol{\alpha}_i) \right). \tag{3.4}$$

This problem of estimating simultaneously A and D is non-convex and belongs to NP-hard problems. It is often solved via an alternating strategy: 1) fixing D and finding the sparse coefficients A, this becomes the problem of sparse coding as aforementioned; 2) fixing A and searching for the solution D, this is the procedure of dictionary updating. Several popular algorithms for sparse coding and dictionary updating are described in the following, as well as the most known combinations of these algorithms.

3.2 Algorithms for sparse coding

The optimization problem of sparse coding (3.2) consists in finding the sparse representation $\boldsymbol{\alpha} \in \mathbb{R}^m$ of $\mathbf{x}$ over the dictionary D, which can be specifically formulated in different forms.

The k-sparse representation problem with maximal number of non-zero entries fixed is modeled by

$$\arg\min_{\boldsymbol{\alpha}} \tfrac{1}{2} \|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2, \quad \boldsymbol{\alpha} \in \mathbb{S}_k, \tag{3.5}$$

where

$$\mathbb{S}_k = \{\mathbf{v} \in \mathbb{R}^m \mid \|\mathbf{v}\|_0 \leq k\}$$

means the set of k-sparse vectors.

By considering the constraint on the reconstruction error, the sparsest representation can be obtained via the problem

$$\arg\min_{\boldsymbol{\alpha}} \; \Omega(\boldsymbol{\alpha}) \quad \text{subject to} \quad \|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 < \epsilon, \tag{3.6}$$

where $\Omega(\boldsymbol{\alpha})$ is a function of $\boldsymbol{\alpha}$ which measures the sparsity of the vector. These constrained optimization problems can be addressed by considering the minimization of the regularized reconstruction error,

$$\arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}), \tag{3.7}$$

where λ introduced previously balances the effects of data fitting and sparsity.

As aforementioned, these three formulations are used for different tasks and they can achieve the equivalent global optima under certain conditions. In contrast, the strategies for tackling the problems of different formulations are quite different. The ℓ_0-norm based problem is non-convex and non-continuous, which makes it hard to be resolved directly. To overcome these difficulties, two major strategies have been widely investigated. The first one use greedy algorithms, such as Matching Pursuit [MZ93], Orthogonal Matching Pursuit [PRK93], subspace pursuit [DM09], and even proximal method [PB14] to let it be tractable. The second strategy is relaxation, since the ℓ_1-norm based problem, also sometimes called LASSO problem, becomes convex which is solved by Basis Pursuit [CD95], by gradient descent algorithm [KY03], by homotopy algorithm [GG09] or by thresholding method [Mei07] and very recently, a novel strategy is emerging, by reformulating the problem as MIQP, which makes it possible to apply some optimization methods like the branch-and-bound method [AGH18, BNCM15] or relaxing the integer constraints to be continuous [BBF$^+$16] and so on. The two major strategies with the main algorithms for sparse coding are described in the following, while the novel strategy is presented in next chapter.

3.2.1 Greedy algorithms

A greedy algorithm solves an optimization problem by searching for the optimal solution of a subproblem at each step. For the sparse coding problem with ℓ_0-norm for sparsity promoting, it selects at each step an atom over which the residual has the maximal projection. Greedy algorithms provide a good sparse approximation [Tro04] rather than the optimal sparse solution. In this part, we present the most known greedy algorithms: matching pursuit and its variants.

Matching pursuit

Matching pursuit (MP) is proposed in [MZ93] to deal with the problem of signal decomposition. The algorithm is inspired by the fact that, giving a signal $\mathbf{x}$ and a dictionary D with each atom of unit norm, denoted by $\mathbf{d}_i$ with $i = 1, \ldots, m$, then the signal decomposition by orthogonal projection onto some selected atom $\mathbf{d}_{t_1}$ is given by

$$\mathbf{x} = \left(\mathbf{x}^{\mathrm{T}}\mathbf{d}_{t_1}\right)\mathbf{d}_{t_1} + \mathbf{r}_1, \tag{3.8}$$

where $\mathbf{r}_1$ is the representation residual orthogonal to $\mathbf{d}_{t_1}$,the latter being selected based on the residual. This leads to the result,

$$\|\mathbf{x}\|^2 = |\mathbf{x}^{\mathrm{T}}\mathbf{d}_{t_1}|^2 + \|\mathbf{r}_1\|^2. \tag{3.9}$$

Then it is conducted that by minimizing $\|\mathbf{r}_1\|^2$, namely maximizing $|\mathbf{x}^{\mathrm{T}}\mathbf{d}_{t_1}|$. The signal $\mathbf{x}$ can be maximally approximated in the space spanned by $\mathbf{d}_{t_1}$. By successively doing k times orthogonal projections on selected atoms of D, the signal can be approximated by

$$\mathbf{x} = \sum_{j=1}^{k} \mathbf{x}^{\mathrm{T}}\mathbf{d}_{t_j} + \mathbf{r}_k, \tag{3.10}$$

where the sequence t_j contains the indices of the selected atoms. With $\lim_{k\to\infty} \|\mathbf{r}_k\|^2 = 0$, the signal $\mathbf{x}$ can be approximated with high precision. It induces also the convergence of the algorithm.

The Matching Pursuit algorithm is exhibited in Algorithm 1. It is worthy to notice that MP algorithm guarantees in each iteration the orthogonality between the i^{th} selected atom $\mathbf{d}_{t_i}$ and the residual $\mathbf{r}_i$. Nevertheless, the space spanned by the selected atoms $[\mathbf{d}_{t_1}, \ldots, \mathbf{d}_{t_i}]$ after i approximations is not guaranteed to be orthogonal to $\mathbf{r}_i$. This risks of getting the sub-optimal solution in each step, which may cause large error after a finite number of steps [PRK93].

Orthogonal matching pursuit

The Orthogonal Matching Pursuit (OMP) algorithm [PRK93] is proposed to overcome the drawback of MP algorithm. Intrinsically, OMP inherits the spirit of MP, which selects an atom at each step to decrease the residual. Thus, OMP is still a greedy algorithm. However, unlike MP, OMP recalculates the composition in the space spanned by the selected atoms, which guarantees always the orthogonality of

Algorithm 1 Matching Pursuit Algorithm (MP)

```
Input: The given signal $\mathbf{x}$, the fixed dictionary D, the stop criterion of sparsity $k$, the
  stop criterion of reconstruct error $\epsilon$.
Output: Sparse approximation $\boldsymbol{\alpha}$
  function SPARSE CODING
      Initialization $i = 1$, the residual $\mathbf{r}_0 = \mathbf{x}$, the approximation $\boldsymbol{\alpha} = 0$ and the initial
  solution support $\mathrm{T}_0 = \emptyset$
      while $card(\mathrm{T}_{i-1}) < k+1$ or $\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 > \epsilon$ do
          Selecting the atom over which the residual $\mathbf{r}_{i-1}$ has the maximal magnitude
  of orthogonal projection
              $\hat{j} = \arg\max_{j=1,\ldots,m} |\langle \mathbf{r}_{i-1}, \mathbf{d}_j \rangle|;$
          Updating support $\mathrm{T}_i = \mathrm{T}_{i-1} \cup \hat{j}$
          Updating the residual after $i^{th}$ projection
              $\mathbf{r}_i = \mathbf{r}_{i-1} - \langle \mathbf{r}_{i-1}, \mathbf{d}_{\hat{j}} \rangle \mathbf{d}_{\hat{j}};$
          Updating the coefficient corresponding to the $\hat{j}^{th}$ atom
              $\boldsymbol{\alpha}(\hat{j}) = \langle \mathbf{r}_{i-1}, \mathbf{d}_{\hat{j}} \rangle;$
          $i = i + 1;$
      end while
  end function
```

representation residual and the spanned space.

The details of OMP algorithm are listed in Algorithm 2. Compared with MP algorithm, OMP algorithm carries out one more operation (3.11) to lead to the best approximation over the selected support [PRK93]. Tropp gave the theoretical support that, when the signal is sparse enough, OMP performs well in signal recovery and approximation [Tro04], even with random measurement matrix such as Gaussian and Bernoulli [TG07].

It is observed that OMP needs no more than k iterations to achieve the approximate solution. Nevertheless, the computing cost in atom identification and coefficient update (3.11) is unnegligible. A progressive Cholesky process is adopted to handle the problem (3.11) and some implementation skills are used to accelerate the algorithm [RZE08].

Furthermore, based on the OMP algorithm, some variants are proposed to reduce computation complexity by optimizing the identification step. The Generalized Orthogonal Matching Pursuit algorithm (gOMP) [WKS12] speeds up the com-

Algorithm 2 Orthogonal Matching Pursuit Algorithm (OMP)

```
Input: The signal for sparse coding $\mathbf{x}$, the known dictionary D, the stopping criterion of sparsity $k$ or that of reconstruct error $\epsilon$.
Output: The sparse representation $\boldsymbol{\alpha}$.
function SPARSE CODING
    Initialization $i = 1$, the residual $\mathbf{r}_0 = \mathbf{x}$, the solution support $\mathrm{T}_0 = \emptyset$, the sparse approximation $\boldsymbol{\alpha} = 0$
    while $i < k+1$ or $\|\mathbf{r}_{i-1}\|_2^2 > \epsilon$ do
        Finding the atom the highest correlated with the current residual, that is
            $t_i = \underset{j \in \{1,\dots,m\} \setminus \mathrm{T}_{i-1}}{\arg\max} |\langle \mathbf{r}_{i-1}, \mathbf{d}_j \rangle|;$
        where $j \in \{1,\dots,m\} \setminus \mathrm{T}_{i-1}$ means that $j \in \{1,\dots,m\}$ and $j \not\in \mathrm{T}_{i-1}$.
        Adding the new atom index into the support
            $\mathrm{T}_i = \mathrm{T}_{i-1} \cup t_i;$
        Updating coefficient vector
            $\boldsymbol{\alpha}_{\mathrm{T}_i} = (\mathrm{D}_{\mathrm{T}_i}^{\mathrm{T}} \mathrm{D}_{\mathrm{T}_i})^{-1} \mathrm{D}_{\mathrm{T}_i}^{\mathrm{T}} \mathbf{x};$    (3.11)
        Here we recall the notation of $\boldsymbol{\alpha}_{\mathrm{T}_i}$ and $\mathrm{D}_{\mathrm{T}_i}$ representing the sub-vector and submatrix indexed by the elements in the set $\mathrm{T}_i$.
        Updating residual
            $\mathbf{r}_i = \mathbf{x} - \mathrm{D}_{\mathrm{T}_i} \boldsymbol{\alpha}_{\mathrm{T}_i};$
        Increment $i = i + 1$
    end while
end function
```

putation by selecting more than one atom at each step. Through the method, a k-sparse signal can be approximated by less than k iterations. However, gOMP is not suitable for signals that are not strictly sparse.

The Regularized Orthogonal Matching Pursuit algorithm (ROMP), as is called, is regularized version of OMP [NV09]. ROMP takes the same strategy as in gOMP to recover sparse signals. At every iteration, all the atoms with comparable coordinates are selected. This process named regularization is realized by:

$$|\boldsymbol{\alpha}_i| \leq 2|\boldsymbol{\alpha}_j| \qquad \forall i, j \in \Lambda_s,$$

where Λ_s denotes a subset of support Λ at i^{th} iteration, which is formed by the k

atoms with largest correlation to the sparse signal. ROMP gains success in dealing with sparse signal reconstruction under the Restricted Isometry Condition (RIC). Even for natural signals, which are not strictly sparse, the algorithm shows stability in sparse approximation. However, the level of sparsity k can be missed with high probability [WKS12].

Similarly, Donoho et al. proposed in [DDTS06] the algorithm Stagewise Orthogonal Matching Pursuit (StOMP). In this algorithm, numerous atoms are determined at each stage by thresholding. The threshold b is defined as a function of the noise level σ, that is,

$$b = p\sigma,$$

where p is a predefined parameter. With adequate threshold setting, StOMP is proved to reach the exact sparse signal recovery as OMP but with higher speed. The algorithm is also suitable to deal with noisy signals. However, in this case, the results are not the one get by OMP.

OMP and its variants have been frequently used for sparse coding for dictionary learning [AEB+06, MBP12, Sch14]. Likewise, algorithms applying the same spirit of OMP are used in tasks such as classification [RS08, HA07, ZL10, MLB+08], face recognition [ZL10] and image denoising [EA06].

Compressive sampling matching pursuit (CoSaMP) and subspace pursuit (SP)

The Compressive Sampling Matching Pursuit (CoSaMP) [NT09] and the Subspace Pursuit (SP) [DM09] are two other sparse coding algorithms developed on OMP. Nevertheless, the two algorithms are inspired by the fact that the energy in each component of $\mathbf{u} = \mathrm{D}^{\mathrm{T}}\mathrm{D}\boldsymbol{\alpha}$ approximates the energy in the corresponding component of $\boldsymbol{\alpha}$, when D satisfies the RIC, namely the largest k elements in $\mathbf{u}$ point to the largest k entries in $\boldsymbol{\alpha}$. Thus, the k support of $\boldsymbol{\alpha}$ can be predicted by $\mathbf{u}$, which is called here proxy. Then, by iteratively regulating support to minimize the residual, the k-sparsity approximation or recovery problem can be solved.

The pseudocode of the CoSaMP algorithm is given in Algorithm 3. This algorithm, in contrast with gOMP, StOMP and ROMP, can guarantee that the obtained sparsity level is k. To this end, the rigorous condition of RIC on D needs to be established. Thus, signal-to-noise ratio (SNR) reduction occurs in each iteration, which induces the convergence of the algorithm. Moreover, based on analysis on SNR reduction, the number of iterations can be predicted. Even it is undeniable that the

Algorithm 3 Compressive Sampling Matching Pursuit (CoSaMP)

```
Input: A given signal $\mathbf{x}$, the fixed overcomplete dictionary D, maximal number of iteration $\mathrm{N}_{iter}$ , the sparsity level $k$.
Output: The sparse representation $\boldsymbol{\alpha}$.
function SPARSE CODING
    Initialization $i = 1$, the residual $\mathbf{r}_0 = \mathbf{x}$, the solution support $\mathrm{T}_0 = \emptyset$, the sparse approximation $\boldsymbol{\alpha} = 0$
    while $i < \mathrm{N}_{iter} + 1$ do
        Calculating $2k$ largest elements in the proxy $\mathbf{u} = \mathrm{D}^{\mathrm{T}}\mathbf{r}_{i-1}$, its support is thus obtained
            $$\mathrm{T} = \{t_1, \dots, t_{2k}\};$$
        where $t_i = \arg\max \mathbf{u}_{\{1,\dots,m\}\setminus\{t_1,\dots,t_{i-1}\}}$ pointing to the $i^{th}$ largest entry in $\mathbf{u}$.
        Merging T and the current support $\mathrm{T}_i$ to update support
            $$\mathrm{T}_i = \mathrm{T}_{i-1} \cup \mathrm{T};$$
        Updating projection value in the new support
            $$\boldsymbol{\alpha}_{\mathrm{T}_i} = (\mathrm{D}_{\mathrm{T}_i}^{\mathrm{T}}\mathrm{D}_{\mathrm{T}_i})^{-1}\mathrm{D}_{\mathrm{T}_i}^{\mathrm{T}}\mathbf{x}; \qquad (3.12)$$
        Pruning elements in support to keep only the $k$ atoms retaining the largest correlation with $\mathbf{x}$. The updated support is denoted by $\mathrm{T}_i$.
        Updating residual
            $$\mathbf{r}_i = \mathbf{x} - \mathrm{D}_{\mathrm{T}_i}\boldsymbol{\alpha}_{\mathrm{T}_i};$$
        Increment $i = i + 1$
    end while
end function
```

number of iterations depends on the properties of the signal. Algorithm 3 limits the iteration number to halt the loop. Besides, other stopping criteria have been presented, such as the reconstruct error [NT09, Apprendix].

The algorithm SP is nearly the same as CoSaMP except for the number of identified atoms [DM09]. This algorithm also well performs under RIC with adequate constant requirement. However, in SP, Dai and Milenkovic located only the k largest components in proxy, rather than $2k$ ones. With this improvement, new theoretical analysis is made in [DM09] to guarantee the residual reduction in each iteration and the convergence was proved with a limited number of iterations. In this case, the halting criterion can adopt the difference between the residuals before and after each iteration, that is to say, if $\|\mathbf{r}_i\|_2 > \|\mathbf{r}_{i-1}\|_2$, set $\mathrm{T}_i = \mathrm{T}_{i-1}$ and quit the iteration.

Other matching pursuit

There are other variants of the MP, such as Hierarchical Matching Pursuit (HMP) [BRF11], and (fast) Bayesian Matching Pursuit (BMP) [SPZ08, WYWW11, MAN13].

HMP was proposed by Bo et al. for high level image feature extraction [BRF13, BRF11]. In contrast with classical OMP algorithms, HMP is designed with three crucial factors: tree-structured dictionary for sparse coding with algorithm OMP, spatial pyramid max pooling and normalization. HMP contains multi-level and each level operates all the three above-mentioned processes. In this framework, the higher level uses the output feature in the last level as the input data for tree OMP, and its output passes to the next level for further learning. Image feature extracted by HMP is then applied in classification. The performance exhibited in [BRF11] proved its advantage even comparing with convolutional neural networks.

BMP regards the sparse representation of a signal as a random variable that satisfies certain probabilistic distribution, rather than a deterministic one. For instance, in [SPZ08], an i.i.d. random variable $\boldsymbol{\alpha}$ is drawn from a Q-ary Gaussian mixture parameterized by $\mathbf{s} = [s_1, \dots, s_m]^{\mathrm{T}}$, that is, each entry α_i corresponds to a certain Gaussian distribution ($s_i = q$) with mean μ_q and variance σ_q^2,

$$\alpha_i \mid s_i = q \sim CN(\mu_q, \sigma_i^2).$$

With an appropriate Gaussian model (four models were introduced in [SPZ08]: zeros-means binary prior, nonzero-mean binary prior, zero-mean tenary prior and Q-ary circular prior), zero entries and non-zero entries in $\boldsymbol{\alpha}$ can be exactly modeled ($s_i = 0 \Rightarrow \alpha_i = 0$ and $s_i \neq 0 \Rightarrow \alpha_i \neq 0$). The probability $p(\mathbf{s}, \mathbf{x})$ is estimated by a variable $v(\mathbf{s}, \mathbf{x})$ called model selection metric, with $p(\mathbf{s}, \mathbf{x}) = e^{v(\mathbf{s}, \mathbf{x})}$ (see [SPZ08] for details). The determination of the support of $\mathbf{x}$ is thus transformed into a problem of finding the Gaussian mixture model with the largest probability. This problem is formulated as a tree search with each node representing a possible mixture $\mathbf{s}$, specifically, the p-level node contains the mixtures with $\|\mathbf{s}\|_0 = p$. By iteratively adding a non-zero entry in $\mathbf{s}$ via selecting the one with the largest probability, the Gaussian model can be finally reached. Then, the sparse code is computed by

$$\hat{\boldsymbol{\alpha}} = \sum p(\mathbf{s}, \mathbf{x}) \mathrm{E}(\boldsymbol{\alpha} \mid \mathbf{x}, \mathbf{s}),$$

where the expectation can be computed from the Gaussian model assumptions.

3.2.2 ℓ_p-norm relaxation

By replacing the ℓ_0-norm by another ℓ_p-norm for sparsity prompting, the optimization problems (3.5), (3.6) and (3.7) are transformed to become continuous, and even convex (for $p \geq 1$), and differentiable (for $p > 1$). The ℓ_1-norm, the nearest norm function to the ℓ_0-norm that is convex, attracted most attention [CDS01, MBPS09, Don06, MKCP17]. The resulting sparse representation problem based on ℓ_1-norm is often solved with the Basis Pursuit [CDS01] and LASSO method [Tib96].

Basis pursuit

The Basis Pursuit [MXAP11, CDS01] focuses on the linear constrained optimization problem (3.6). As mentioned in [CDS01], the Basis Pursuit is a principle rather than an algorithm. It reformulates the problem by separating the non-negative and the non-positive parts of $\boldsymbol{\alpha}$, namely $\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-$ with $\boldsymbol{\alpha}^+ \geq 0$ and $\boldsymbol{\alpha}^- \geq 0$. The optimization problem is thus written by

$$\arg\min \mathbf{c}^{\mathrm{T}}\mathbf{v} \quad \text{subject to} \quad [\mathrm{D}, -\mathrm{D}]\mathbf{v} = \mathbf{x}, \quad \mathbf{v} \geq 0, \tag{3.13}$$

where $\mathbf{c} = \mathbf{1}_{2m}$ is the vector of ones of size $2m$, and variable $\mathbf{v}$ of size $2m$ is composed by $(\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-)$. The problem (3.13) is obviously the standard form of linear programming. For denoising, the corresponding optimization problem, formulated with the error constraint (3.6) or regularization term (3.7) [CD95], can be reformulated as a problem of linear programming with quadratic constraints or quadratic programming respectively. In this way, the ℓ_1-norm based problem can be solved via optimization techniques, such as the simplex algorithm and the interior point method.

LASSO

LASSO provides a new point of view to solve the ℓ_1-norm constrained square error minimization problem (3.5), or its regularized problem (3.7) [Tib96]. The method of LASSO is realized by introducing an auxiliary matrix G whose rows contain the signs of the corresponding $\boldsymbol{\alpha}$. Consequently, the inequality constraint can be represented by the linear inequality $\mathrm{G}\boldsymbol{\alpha} \leq \lambda_k \mathbf{1}$. With considering the sign of a real value being either $+1$ or -1, there are totally 2^p kinds of possible structures. The number of rows of G is thus 2^p. By eliminating calculations on G of too big size, an iterative process

Algorithm 4 LASSO

```
Input: the signal $\mathbf{x}$, the dictionary D, the sparsity $\lambda_k$
Output: The sparse representation $\boldsymbol{\alpha}$.
function SPARSE CODING
    Initialization $\boldsymbol{\alpha}_0 = (\mathrm{D}^\mathrm{T}\mathrm{D})^{-1}\mathrm{D}^\mathrm{T}\mathbf{x}$, defining $\mathrm{E} = \{i \mid \mathbf{g}_i{}^\mathrm{T}\boldsymbol{\alpha} = \lambda_k\}$ where $\mathbf{g}_i$ is the $i^{th}$ row of G and setting initial $\mathrm{E} = \{i_0 \mid \mathbf{g}_{i_0} = \mathrm{sign}(\boldsymbol{\alpha}_0)\}$.
    do
        Finding solution $\hat{\boldsymbol{\alpha}}$ to the problem of minimization,
            $$\arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|^2 \quad \text{subject to} \quad \mathrm{G}_\mathrm{E}\boldsymbol{\alpha} \leq \lambda_k \mathbf{1}_{|\mathrm{E}|} \tag{3.14}$$
        updating $\mathrm{E} = \mathrm{E} \cup \{i \mid \mathbf{g}_i = \mathrm{sign}(\hat{\boldsymbol{\alpha}})\}$
    while $\|\boldsymbol{\alpha}_1\| > \lambda_k$
end function
```

is proposed in [Tib96]. The outline of the algorithm is summarized in Algorithm (4), where the optimization problem (3.14) is a quadratic programming with linear inequality constraints, which can be solved with optimization problem methods such as seeking the solution satisfying the Kuhn-Tucker conditions [Ber97].

In addition to reformulating the ℓ_1-norm regularized least square minimization problem as a classical optimization problem, this problem can also be solved by algorithms such as Least Angle Regression (LARS), homotopy algorithm or some gradient based algorithms [KKK08, GK09](gradient algorithms will be presented in the next subsection). The LARS algorithm updates the estimation iteratively along the least angle direction until all non-zero elements are found [EHJ+04].

The homotopy algorithms [MCW05b, GG09] are developed based on the fact that the solution to the regularized problem can be computed directly by letting the zero vector belong to the subdifferential of the objective function, which was presented in section 2 of [GG09], namely it exists a $\boldsymbol{d\alpha}$ that satisfies

$$\mathrm{D}^\mathrm{T}(\mathrm{D}\boldsymbol{\alpha} - \mathbf{x}) + \lambda\boldsymbol{d\alpha} = 0, \tag{3.15}$$

where $\boldsymbol{d\alpha}$ denotes the subdifferential of the objective function of the regularized problem, which takes the form

$$\boldsymbol{d\alpha} = \partial\|\boldsymbol{\alpha}\|_1 = \left\{\mathbf{v} \in \mathbb{R}^m \,\middle|\, \begin{cases} v_i = \mathrm{sign}(\alpha_i) & \text{if } |\alpha_i| > 0 \\ v_i \in [-1, 1] & \text{if } \alpha_i = 0. \end{cases}\right\}$$

Thus, with knowing the support and sign of the entries $\boldsymbol{\alpha}$, the problem parameterized by a certain λ can be easily solved. The Lagrangian method turns the constrained optimization problem into a regularized least-square minimization problem. This optimization problem lies then on the determination of the support and sign of $\boldsymbol{\alpha}$. Homotopy is the algorithm offering the way to find these two important information about $\boldsymbol{\alpha}$. In [MCW05b], the authors regarded the optimal solution to the problem as a function of λ, expressed by $\boldsymbol{\alpha}(\lambda)$. This function shows a piecewise linear continuity, and the support and sign of $\boldsymbol{\alpha}$ keep unchangeable in a range [GG09, MY12]. With this characteristic, the support and sign can be successively added or removed by getting the critical value of λ, namely the value that makes a non-zero entries of $\boldsymbol{\alpha}$ turn to zero and the subdifferential corresponding to zero entries of $\boldsymbol{\alpha}$ reaching the limit $\|\boldsymbol{d\alpha}\|_\infty = 1$. Thereupon, by decreasing λ from $\|\mathrm{D}^{\mathrm{T}}\mathbf{x}\|_\infty$ to zero, we get the optimal solution corresponding to each λ. Choosing the solution with the desirable number of zeros in α results in the final solution. The searching path is given in [GG09, MY12, PH07]. Garrigues and Ghaoui [GG09] improved the algorithm by rewriting the optimal solution of the problem as a function of an auxiliary variable t and parameter λ, namely $\boldsymbol{\alpha}(t, \lambda)$. Moreover, λ is not directly set but reached by an increasing sequence $\{\lambda_n\}$. With these notations, they design a path from $\boldsymbol{\alpha}$ at the i^{th} iteration $\boldsymbol{\alpha}^i = \boldsymbol{\alpha}(0, \lambda_i)$ to its value at iteration $i+1$ with $\boldsymbol{\alpha}^{i+1} = \boldsymbol{\alpha}(1, \lambda_{i+1})$, which is composed of two steps: 1) updating $\boldsymbol{\alpha}(0, \lambda_{i+1})$ from $\boldsymbol{\alpha}(0, \lambda_i)$ with the method mentioned above; 2) fixing λ and varying t from 0 to 1.

Relaxation formulations by ℓ_p-norm with $0 < p < 1$ are also studied in references [FM11, RSS10, SBFA17, ZMZ+13]. However, the corresponding optimization problem is not convex, which makes it hard to solve.

3.2.3 Gradient descent and iterative thresholding algorithms

Iterative Thresholding methods [Kow14, BYD07] often take advantage of thresholding gradient descent, which is used to solve sparse representation problems due to its fast convergence and theoretical sparse approximation guarantee [Fou11, YLZ18, GK09, KKK08]. It is noticed that Iterative Thresholding Algorithm can be adopted to solve ℓ_0-norm based optimization problems [BYD07, GK09, YLZ14], and ℓ_1-norm based optimization problems [FISI16, Mei07, WNF09, KXAH15].

Algorithm 5 Iterative Hard Thresholding Algorithm for ℓ_0 Regularization

```
Input: the signal $\mathbf{x}$, the dictionary D, the sparsity level $k$, the stepsize $\gamma$, the stopping criterion $\epsilon$.
Output: The sparse representation $\boldsymbol{\alpha}$.
function SPARSE CODING
    Initialization $i = 1$, the coefficient vector $\boldsymbol{\alpha}_0 = 0$.
    while $\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 > \epsilon$ do
        Gradient Descent to reduce error $\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2$,
            $\boldsymbol{\alpha}_m = \boldsymbol{\alpha}^{i-1} + \frac{1}{\gamma}\mathrm{D}^{\mathrm{T}}(\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}^{i-1}).$   (3.16)
        Sorting all entries of $|\boldsymbol{\alpha}|$ in descending order, that is, $|\alpha_{s(1)}| \geq \cdots \geq |\alpha_{s(m)}|$.
        Then, taking the $k^{th}$ largest one $|\alpha_{s(k)}|$ the threshold. The $\boldsymbol{\alpha}$ is thus updated by
            $\boldsymbol{\alpha}^i = \mathrm{T}_{|\alpha_{s(k)}|}(\boldsymbol{\alpha}),$
        where $\mathrm{T}_c(\boldsymbol{\alpha})$ means that all entries of $\boldsymbol{\alpha}$ larger than $c$ in magnitude will be retained while the others are set to 0.
        Increment $i = i + 1$
    end while
end function
```

Gradient descent with projection on constraint set

Iterative Hard Thresholding (IHT), also called Gradient Descent with Sparsification (GraDeS) in [GK09], creates a sequence of $\boldsymbol{\alpha}^i$ by iteratively reducing the square error along the gradient direction by a step of length $1/\gamma$ followed by retraining the largest k entries of $\boldsymbol{\alpha}$ by thresholding. Its framework is described in Algorithm 5.

The IHT algorithm is proved to produce sparse approximation whenever the signal is noisy or not, under the condition that the dictionary D satisfies the RIC with isometric constant $\delta_{2k} < 1/3$. Furthermore, the algorithm is proved to converge in a limited number of iterations by setting $\gamma = 1$ [BD08, BD09, HGT06] or $\gamma = 1 + \gamma_{2k}$ [GK09]. Moreover, the update step size γ is allowed to vary according to iteration [Fou11]. Foucart [Fou11] and Yuan et al [YLZ18] proposed a variant, Hard Thresholding Pursuit (HTP), which combines the IHT method with CoSaMP. At each iteration, following the thresholding step, HTP calculates the maximal projection in the space spanned by the support of $\boldsymbol{\alpha}^i$. Compared with IHT, HTP improves the performance on error minimizing and holds on the high-speed computation.

Besides, the IHT is intrinsically equivalent to the proximal method [BJQS14, PB14] when dealing with a k-sparsity constrained problem (details given in section 4.1.2), although these two methods are developed in two different frameworks.

Iterative thresholding algorithm for regularized sparse problem

For the regularized optimization problem, the solution depends on the value of trade-off parameter λ, which implies that the threshold in the algorithm is a function of λ, namely $s(\lambda)$. The resolution of this problem is done with the aid of a surrogate function [BYD07, BD08, DDDM04]. The surrogate function is obtained from reformulating the original problem by introducing an auxiliary variable $\mathbf{a} \in \mathbb{R}^m$ as follows

$$\operatorname*{arg\,min}_{\boldsymbol{\alpha},\mathbf{a}} \tfrac{1}{2}\|\mathbf{x}-\mathrm{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_p - \|\mathrm{D}\boldsymbol{\alpha}-\mathrm{D}\mathbf{a}\|_2^2 + \|\boldsymbol{\alpha}-\mathbf{a}\|_2^2, \tag{3.17}$$

where $\mathbf{a}$ is a variable closed to $\boldsymbol{\alpha}$. Developing the problem (3.17), we have

$$\operatorname*{arg\,min}_{\boldsymbol{\alpha},\mathbf{a}} \sum_j (\alpha_j^2 - 2\alpha_j(\mathbf{a}+\mathrm{D}^{\mathrm{T}}\mathbf{x}-\mathrm{D}^{\mathrm{T}}\mathrm{D}\mathbf{a})_j + \lambda|\alpha_j|^p) + \|\mathbf{x}\|^2 + \|\mathbf{a}\|^2 - \|\mathrm{D}\mathbf{a}\|^2. \tag{3.18}$$

When $p = 0$, the objective function is non-differentiable. Considering the problem separately for the cases $\alpha_j = 0$ and $\alpha_j \neq 0$, then, its solution is of the form

$$\boldsymbol{\alpha} = \mathrm{T}_{\lambda^{0.5}}(\mathbf{a}+\mathrm{D}^{\mathrm{T}}(\mathbf{x}-\mathrm{D}\mathbf{a})),$$

where $\mathrm{T}_{\lambda^{0.5}}$ is the thresholding operator defined as

$$\mathrm{T}_{\lambda^{0.5}}(\alpha) = \begin{cases} 0 & \text{if } |\alpha| \leq \lambda^{0.5} \\ \alpha & \text{if } |\alpha| > \lambda^{0.5}. \end{cases}$$

Hence, replacing $\mathbf{a}$ by $\boldsymbol{\alpha}^i$, the solution can be reached by a convergent sequence produced by using the IHT algorithm

$$\boldsymbol{\alpha}^{i+1} = \mathrm{T}_{\lambda^{0.5}}\big(\mathbf{a}^i + \mathrm{D}^{\mathrm{T}}(\mathbf{x}-\mathrm{D}\mathbf{a}^i)\big).$$

It is noticed that, in this algorithm, the descent step size is set to be 1 which could cause unstability. A sufficient condition for stability and convergence is having the eigenvalues of $\mathrm{I}-\mathrm{D}^{\mathrm{T}}\mathrm{D}$ strictly between 0 and 1 [BD08, BYD07].

When $p = 1$, the surrogate function becomes differentiable only if $\alpha_j \neq 0$. Then, except for $\alpha_j = 0$, letting the derivative of the function be equal to 0, the solution is expressed as

$$\alpha_j = \mathrm{TS}_{\lambda/2}(a_j + (\mathrm{D}^{\mathrm{T}}(\mathbf{x} - \mathrm{D}\mathbf{a}))_j),$$

where $\mathrm{TS}_{\lambda/2}$ is the soft thresholding operator with the definition of

$$\mathrm{TS}_{\lambda/2}(\alpha) = \begin{cases} \alpha - \lambda/2 & \text{if } \alpha \geq \lambda/2 \\ 0 & \text{if } |\alpha| < \lambda/2 \\ \alpha + \lambda/2 & \text{if } \alpha \leq -\lambda/2. \end{cases}$$

By replacing a_j by α_j^i, which is the solution obtained in the i^{th} iteration, and α_j by the updated solution α_j^{i+1} after $i+1$ iterations, the Iterative Soft Thresholding algorithm (IST) is obtained with

$$\alpha_j^{i+1} = \mathrm{TS}_{\lambda/2}(\alpha_j^i + (\mathrm{D}^{\mathrm{T}}(\mathbf{x} - \mathrm{D}\alpha^i))_j). \tag{3.19}$$

The proof of convergence can be found in [DDDM04].

By soft thresholding, the non-zero coefficients are shrunken, namely there is a bias in the coefficient estimation compared to the least-squares solution. Accordingly, relaxed LASSO estimator was proposed in [Mei07], which tunes λ by another parameter varying from 1 to 0. Fujiwara et al. [FISI16] improved the efficency of the IST algorithm by a predetermined step in which the necessary updated coefficients are selected and the unnecessary ones are pruned. Moreover, Two-Step Iterative Shrinkage/Thresholding Algorithm (TwIST) updates $\boldsymbol{\alpha}^{i+1}$ depending both on $\boldsymbol{\alpha}^i$ and $\boldsymbol{\alpha}^{i-1}$, which shows good performance on denoising and stability [BDF07].

Gradient descent algorithm

Due to the differentiability of the LASSO problem in all α_i, except where $\alpha_i = 0$, the gradient descent can be applied to the problem. However, as aforementioned, the step size and its direction are difficult to determine in order to keep the updated coefficient in the acceptance constraint region. In this section, we present the gradient LASSO proposed by Kim et al. in [KKK08], which is stable and convergent.

Similar to the iterative thresholding algorithm, the gradient descent LASSO is composed of two steps: addition step and deletion step. The optimization problem is transformed, by using the variable change $\boldsymbol{\omega} = \boldsymbol{\alpha}/\lambda$, into

$$\arg\min_{\boldsymbol{\omega}\in\mathbb{S}} \tfrac{1}{2}\|\mathbf{x} - \lambda \mathrm{D}\boldsymbol{\omega}\|_2^2,$$

where $\mathbb{S} = \{\mathbf{v} \in \mathbb{R}^m \mid \|\mathbf{v}\|_1 \leq 1\}$. The idea of the gradient LASSO is to find a path of descent $\boldsymbol{\omega}$ to cause reduction of the square error and guarantee $\boldsymbol{\omega}$ is always in the set $\mathbb{S}$, which is realized in [KKK08] with two steps: the first step decreases the square error and the second step corrects $\boldsymbol{\omega}$ in the set $\mathbb{S}$. The outline of the gradient descent LASSO is presented in Algorithm 6. Next, we detail the addition and deletion steps.

The addition step takes advantage of the coordinate gradient descent algorithm (CGD), namely at each iteration, an entry is updated. For instance, Kim et al. select the one with the largest gradient component

$$\hat{j} = \max_j |\nabla_{\omega_j} f(\boldsymbol{\omega})|,$$

where $f_{\boldsymbol{\omega}} = \frac{1}{2}\|\mathbf{x} - \lambda \mathrm{D}\boldsymbol{\omega}\|_2^2$. Then $\omega_{\hat{j}}$ is updated along the direction in $\mathbb{S}$,

$$\omega_{\hat{j}}^{i+1} = \omega_{\hat{j}}^{i} + \gamma(v - \omega_{\hat{j}}^{i}),$$

where $v = \mathrm{sign}(\nabla_{\omega_{\hat{j}}} f(\boldsymbol{\omega}))$ and $\gamma \in [0,1]$. It is noticed that the direction of descent is $\mathbf{v} - \boldsymbol{\omega}$ where $\mathbf{v}$ is the vector whose $\hat{j}^{th}$ component is v, namely $\mathbf{v} \in \mathbb{S}$. With assumption of $\boldsymbol{\omega}^i \in \mathbb{S}$ and γ varying from 0 to 1, each update is still in the set $\mathbb{S}$. The descent step γ is determined by minimizing

$$\hat{\gamma} = \arg\min_{\gamma\in[0,1]} f\big((1-\gamma)\boldsymbol{\omega} + \gamma\mathbf{v}\big).$$

The deletion step handles the problem by considering two possible cases: 1) the updated $\boldsymbol{\omega}^{i+1}$ does not achieve the optimal sparsity level, namely there is non-zero entries in $\boldsymbol{\omega}^{i+1}$ that ought to be zero in an optimal solution, 2) $\boldsymbol{\omega}^{i+1}$ has the desirable sparsity level but not optimal. The first case occurs when $\langle \nabla_{\boldsymbol{\omega}_{\mathrm{I}}^{i+1}} f, \boldsymbol{\theta}_{\mathbb{I}}\rangle < 0$ and $\|\boldsymbol{\omega}^{i+1}\|_1 = 1$, where $\mathbb{I}$ denotes the set of indices of the active entries $\mathbb{I} = \{j \mid \omega_j \neq 0\}$, where $\boldsymbol{\theta}$ is a vector that contains all signs of updated coefficient vector $\boldsymbol{\theta} = \mathrm{sign}(\boldsymbol{\omega}^{i+1})$. The notation of $\boldsymbol{\omega}_{\mathbb{I}}^{i+1}$, as a component of $\boldsymbol{\omega}^{i+1}$, is given in Chapter 1. Then the direction $\mathbf{v}$ was proved to be

$$\mathbf{v}_{\mathbb{I}} = -\nabla_{\boldsymbol{\omega}_{\mathbb{I}}^{i+1}} f + \frac{\langle \boldsymbol{\theta}_{\mathbb{I}}, \nabla_{\boldsymbol{\omega}_{\mathbb{I}}^{i+1}} f\rangle}{\|\boldsymbol{\omega}^{i+1}\|_0}\boldsymbol{\theta}_{\mathbb{I}}.$$

Algorithm 6 Gradient LASSO

```
Input: the signal $\mathbf{x}$ and the dictionary D, parameter $\lambda$, stopping criterion $\epsilon$.
Output: The sparse representation $\boldsymbol{\alpha}$.
function SPARSE CODING
    Initialization $i = 1$, the coefficient vector $\boldsymbol{\alpha}_0 = 0$
    while $\|\boldsymbol{\omega}^i - \boldsymbol{\omega}^{i-1}\|_2^2 > \epsilon$ do
        Addition step
        Iteratively updating $\omega_i$ by gradient descent algorithm
            $$\omega_j^m = (1-\gamma)\omega_j^i + \nu\gamma,$$
        where $\nu = -\mathrm{sign}(\nabla_{\omega_j}\|\mathbf{x} - \lambda\mathrm{D}\boldsymbol{\omega}\|_2^2)$ and $\hat{\gamma} = \arg\min_{\gamma\in[0,1]} f(\boldsymbol{\omega}^{i+1} + \gamma\mathbf{v})$, where $\mathbf{v} \in \mathbb{R}^m$ with the $j^{th}$ entry $\mathbf{v}_j = \nu$.
        Deletion step
        Finding the index set $\square$ which contains the indices of non-zero elements in $\boldsymbol{\omega}$.
        Computing derivative of function $f$ at the sub-vector $\boldsymbol{\omega}_\square$, namely $\nabla_{\boldsymbol{\omega}_\square^{i+1}} f$.
        if $\langle\nabla_{\boldsymbol{\omega}_\square^{i+1}} f, \boldsymbol{\theta}_\square\rangle < 0$ or $\|\boldsymbol{\omega}^{i+1}\|_1 = 1$ then
            $\mathbf{v}_\square = -\nabla_{\boldsymbol{\omega}_\square^{i+1}} f + \frac{\langle\boldsymbol{\theta}_\square, \nabla_{\boldsymbol{\omega}_\square^{i+1}} f\rangle}{\|\boldsymbol{\omega}^{i+1}\|_0}\boldsymbol{\theta}_\square$;
        else
            $\mathbf{v}_\square = -\nabla_{\boldsymbol{\omega}_\square^{i+1}} f$;
        end if
        Setting $\mathbf{v}$ the descent direction with all entries indexed by $\square$ the vector $\mathbf{v}_\square$.
        Updating $\boldsymbol{\omega}^{i+1} = \boldsymbol{\omega}^{i+1} + \hat{\gamma}\mathbf{v}$, where $\hat{\gamma} = \arg\min_{\gamma\in\mathrm{J}} f(\boldsymbol{\omega}^{i+1} + \gamma\mathbf{v})$
        Increment $i = i + 1$
    end while
end function
```

The second case is handled in the classical way of gradient descent, that is, $\mathbf{v}_\square = -\nabla_{\boldsymbol{\omega}_\square^{i+1}} f$. Then formulate $\mathbf{v}$ by assigning all entries indicated by $\square$ the corresponding value and the others to zeros. The step size is obtained by minimizing the objective function but is limited to the interval $\mathrm{J} = [0, \min\{-\omega_j^{i+1}/\nu_j \mid \omega_j^{i+1}\nu_j < 0,\ j \in \square\}]$. In this way, some coordinates of $\boldsymbol{\omega}^{i+1}$ would decrease to 0, namely some entries are deleted.

With a finite number of iterations, the gradient based LASSO is ensured to be convergent and with a bounded error [KKK08]. Furthermore, this algorithm has no RIC assumption but only requires the Lipschitz continuity of $\nabla_{\boldsymbol{\omega}} f$.

3.3 Algorithms for dictionary updating

The subproblem of dictionary updating, aims at finding the local optimal solution of the problem (3.4) while fixing the matrix of decomposition coefficients A of formulation

$$\min_{\mathrm{D}\in\mathbb{D}} \frac{1}{\ell}\sum_{i=1}^{\ell} \tfrac{1}{2}\|\mathbf{x}_i - \mathrm{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}_i). \tag{3.20}$$

By removing all the constant terms that are associated with the matrix of decomposition coefficients A, this subproblem can be reduced to the following optimization problem:

$$\min_{\mathrm{D}\in\mathbb{D}} \tfrac{1}{2}\|\mathrm{X} - \mathrm{DA}\|_{\mathrm{F}}^2, \tag{3.21}$$

where $\|\bullet\|_{\mathrm{F}}$ denotes the Frobenius norm. This problem is differentiable and convex, which is thus much easier to be solved. In this section, we present two algorithms for dictionary updating: the Method of Optimal Directions (MOD) [EAH99] and the method of Coordinate Descent based on SVD [EA06].

3.3.1 The method of optimal directions (MOD)

The MOD algorithm proposed by Engan et al [EAH99] updates the dictionary D by iteratively moving along the optimal direction that minimizes the representation residual.

Supposing that the sparse representation A of the signal matrix X is obtained, its representation residual can be expressed by

$$\begin{aligned} r(\mathrm{D}) &= \sum_i r_i(\mathrm{D}) \\ &= \sum_i \|\mathbf{x}_i - \mathrm{D}\boldsymbol{\alpha}_i\|_2^2 \\ &= \|\mathrm{X} - \mathrm{DA}\|_{\mathrm{F}}^2. \end{aligned} \tag{3.22}$$

The objective is to update the dictionary D to D + Δ, where the increment matrix Δ is obtained by minimizing this residual, namely

$$\underset{\Delta\in\mathbb{R}^{n\times m}}{\arg\min}\, r(\mathrm{D}+\Delta). \tag{3.23}$$

Moreover, $r(\mathrm{D}+\Delta) < r(\mathrm{D})$ would be satisfied to guarantee convergence of the algorithm. By computing the gradient of the above cost function and setting it to zero, we achieve the result

$$\Delta \mathrm{AA}^{\mathrm{T}} = (\mathrm{X} - \mathrm{DA})\mathrm{A}^{\mathrm{T}}.$$

Thus, D can be updated by $\mathrm{D} + \Delta$, namely $\mathrm{D}^{i-1} + (\mathrm{X} - \mathrm{D}^{i-1}\mathrm{A})\mathrm{A}^{\mathrm{T}}(\mathrm{AA}^{\mathrm{T}})^{-1}$, which can be simplified to $\mathrm{D}^{i} = \mathrm{XA}^{\mathrm{T}}(\mathrm{AA}^{\mathrm{T}})^{-1}$ [SE12].

The MOD algorithm is usually used for dictionary updating in sparse model researches [MLB+08], Moreover, its variant, Method of Optimal COherence-COnstrained Direction (MOCOD) is developed for incoherent dictionary learning [RLS09].

3.3.2 The algorithm of coordinate descent based on SVD

While the MOD algorithm updates the whole dictionary at each iteration, the Coordinate Descent algorithm based on Singular Value Decomposition (SVD) updates successively the atoms of the dictionary. In the following, we introduce the algorithm proposed in [AEB+06].

Let $\mathrm{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_m] \in \mathbb{R}^{l \times m}$ with $\mathrm{B} = \mathrm{A}^{\mathrm{T}}$, namely $\mathbf{b}_j$ denotes the j^{th} row of A. The optimization problem with respect to D is then formulated by

$$\begin{aligned}\hat{\mathrm{D}} &= \arg\min_{\mathrm{D}\in\mathbb{D}} \|\mathrm{X} - \mathrm{DA}\|_{\mathrm{F}}^{2} \\ &= \arg\min_{\mathrm{D}\in\mathbb{D}} \|(\mathrm{X} - \textstyle\sum_{j\neq i} \mathbf{d}_j \mathbf{b}_j^{\mathrm{T}}) - \mathbf{d}_i \mathbf{b}_i^{\mathrm{T}}\|_{\mathrm{F}}^{2},\end{aligned} \tag{3.24}$$

thus the matrix multiplication DA is represented by the sum of outer products $\sum_j \mathbf{d}_j \mathbf{b}_j^{\mathrm{T}}$. When updating $\mathbf{d}_i$ with fixing all the other atoms, for keeping the coefficient matrix in sparsity constraint, only the signals with j^{th} entry non-zero are considered. Let the set of these indices be denoted by $\mathbb{J}$, thus the set of the signals can be denoted by $\mathrm{X}_{\mathbb{J}}$. Supposing $\mathrm{E} = \mathrm{X}_{\mathbb{J}} - \sum_{j\neq i} \mathbf{d}_j \mathbf{b}_{j,\mathbb{J}}^{\mathrm{T}}$, the problem with respect to $\mathbf{d}_i$ is now

$$\hat{\mathbf{d}}_i = \underset{\|\mathbf{d}_i\|^2=1}{\arg\min} \|\mathrm{E} - \mathbf{d}_i \mathbf{b}_{i,\mathbb{J}}^{\mathrm{T}}\|_{\mathrm{F}}^{2}.$$

This problem aims at finding a normalized vector $\hat{\mathbf{d}}_i$ over which E achieves the largest projection. The SVD algorithm is able to find the closest rank-1 matrix (i.e., vector) that approximates E with great extent. Thus, $\mathbf{d}_i$ and $\mathbf{b}_{i,\mathbb{J}}$ will be simultaneously updated in this phase, this is unique in dictionary learning algorithm.

3.4 Algorithms for dictionary learning

In this section, we restrict the presentation to the classical dictionary learning, which has been extended to other tasks. Task-specified dictionaries are designed by reformulating the problem with a special regularization term. For instance, discriminative dictionaries are learned by introducing penalty on the classification error (e.g. measured by linear classifier [JLD13, ZL10], by logistic loss [MPS⁺09], or by Fisher criterion [HA07]). Low-rank dictionaries are obtained by learned problem with a regularization of the nuclear norm [ZJD13]. Incoherent dictionary learning is realized by adding a regularization term about the difference between $D^T D$ and the identity matrix of size $m \times m$, denoted by I_m [MBP12, TLZ⁺19, AFS15, BQJ14] (this problem will be discussed in detail in Chapter 5 of this monograph).

We will not present all the algorithms of dictionary learning for each special task but only the classical one, that is, the signal denoising or signal reconstruction problem. We rewrite the optimization problem of dictionary learning (3.4),

$$\underset{A \in \mathbb{R}^{m \times l}, D \in \mathbb{D}}{\arg\min} \ \frac{1}{2}\|X - DA\|_F^2 + \lambda \sum_i \Omega(\boldsymbol{\alpha}_i), \tag{3.25}$$

where the regularization $\Omega(\boldsymbol{\alpha}_i)$ can be $\|\boldsymbol{\alpha}_i\|_0$ or $\|\boldsymbol{\alpha}_i\|_1$, or any other sparse promoting penalty. Like the sparse coding problem, the above dictionary learning problem has also its constrained formulations, which are the sparsity constrained problem

$$\underset{A \in \mathbb{R}^{m \times l}, D \in \mathbb{D}}{\arg\min} \ \frac{1}{2}\|X - DA\|_F^2, \quad \boldsymbol{\alpha}_i \in \mathbb{S}_k \ \forall i = 1, \dots, l, \tag{3.26}$$

and the error constrained problem

$$\underset{A \in \mathbb{R}^{m \times l}, D \in \mathbb{D}}{\arg\min} \ \sum_i \Omega(\boldsymbol{\alpha}_i) \quad \text{subject to} \quad \|X - DA\|_F^2 < \epsilon. \tag{3.27}$$

The latter optimization problem, jointly on A and D, is NP-hard. It is difficult to solve this problem by directly using optimization techniques. Thus, we takes strategies of tackling the problem by two alternating steps: sparse coding, which is a problem with respect to A by fixing D, and dictionary updating, which fixes A and aims at computing D. The combination of sparse coding algorithms (in Section 3.2) and dictionary updating algorithm (in Section 3.3) allows to define dictionary learning algorithms. This section introduces the most frequently used algorithms

Algorithm 7 Algorithm of K-SVD

```
Input: Signals set X, sparsity level $k$, maximal number of iteration N.
Output: The sparse representation A and dictionary D.
Initialization The dictionary $D_0$.
function DICTIONARY LEARNING
    for $i = 1$ to N do
        Sparse Coding
        Computing sparse representation $\boldsymbol{\alpha}_j^i$ for each signal $\mathbf{x}_j$ with $j$ varying from 1 to $l$ by applying OMP algorithm.
        Dictionary updating
        $m$ = the number of column of $D_0$.
        for $j = 1$ to $m$ do
            Finding all signals with $j^{th}$ entry non-zero which is indexed by $\mathbb{J}$, denoted by $X_{\mathbb{J}}$
            Calculating $E = X_{\mathbb{J}} - \sum_{p \neq j} \mathbf{d}_p (\mathbf{b}^i)^T_{p,\mathbb{J}}$.
            Doing SVD decomposition of $E = USV^T$, then $\mathbf{d}_j^i = \mathbf{u}_1$ and $(\mathbf{b}^i)^T_{j,\mathbb{J}} = s(1,1)\mathbf{v}_1$.
        end for
    end for
end function
```

for dictionary learning.

3.4.1 K-SVD

The K-SVD algorithm is a representative dictionary learning algorithm that is proposed as a generalization of the k-means algorithm, but regarding a signal as a linear combination of several atoms [AEB+06].

Since it was proposed, efforts have been conducted on the improvement of implementation [RZE08] and some theoretical foundations [Sch14], which proved the certainty of reaching a local minimum under the condition of coefficient sufficient decay. Hence, the algorithm is widely applied with success in image denoising [EA06], face recognition [ZL10] and classification [JLD13].

K-SVD deals with the optimization problem (3.26) by iteratively alternating the two phases, sparse coding and dictionary updating as follows: it exploits OMP for sparse coding, and dictionary update is operated using the Coordinate Descent Method. While the MOD method updates the whole dictionary in each step, the K-SVD updates each atom of the dictionary successively, through the singular value decomposition (SVD) [AEB+06], as introduced in Section 3.3.2.

The complete K-SVD algorithm is shown in Algorithm 7. In practice, for speed consideration, the atoms of the dictionary are not updated in order but obeying a random selection rule.

Due to the great achievements of the K-SVD algorithm [Sch14, RZE08, MBP12, ZL10], some variants are developed on the algorithm. For example, the kernel K-SVD provides a nonlinear variant of K-SVD [GE16, VNPNC12, ZSJ+16, KD16]. The INK-SVD (for INcoherent K-SVD) allows to delete atoms by adding a decorrelation step following the dictionary update [MBP12] (See Chapter 5 for details).

3.4.2 Sum of outer products dictionary learning

The Sum of OUter Products Dictionary Learning (SOUPDIL) algorithm, recently proposed by Ravishankar et al [RNF17], is inspired by the K-SVD algorithm. Specifically, the sparsity regularized dictionary learning problem (3.25) is rewritten as an optimization problem with respect to the matrix B defined in the K-SVD algorithm, rather than the original sparse representation A, that is,

$$\underset{\mathrm{B}\in\mathbb{R}^{l\times m},\mathrm{D}\in\mathbb{D}}{\arg\min}\ \tfrac{1}{2}\|\mathrm{X}-\mathrm{DB}^{\mathrm{T}}\|_{\mathrm{F}}^{2}+\lambda\|\mathrm{B}\|_{0}, \tag{3.28}$$

where $\|\mathrm{B}\|_0$ corresponds to the number of non-zero elements in matrix B. Then applying the formulation of sum of outer products $\mathrm{DB}^{\mathrm{T}} = \sum_i \mathbf{d}_i\mathbf{b}_i^{\mathrm{T}}$, the problem is transformed into

$$\underset{\mathrm{B}\in\mathbb{R}^{l\times m},\mathrm{D}\in\mathbb{D}}{\arg\min}\ \tfrac{1}{2}\|\mathrm{X}-\sum_i \mathbf{d}_i\mathbf{b}_i^{\mathrm{T}}\|_{\mathrm{F}}^{2}+\lambda\sum_i\|\mathbf{b}_i\|_{0}. \tag{3.29}$$

It is worth noting that the problem restricts the total number of non-zero elements in all signal representations, in contrast with problem (3.25) that constrains only the sparsity of each signal representation. Thus, this formulation allows a flexible sparsity for different signals.

For solving the optimization problem (3.29), a block coordinate descent method is applied. In SOUPDIL, the number of inner iterations is m, which indicates the row size of dictionary D or that of matrix B. Thereby, in each iteration i, the i^{th} column of B, i.e., $\mathbf{b}_i$, and the i^{th} column of D, i.e., $\mathbf{d}_i$, are successively updated. Hereafter, we focus on the algorithm details in each iteration.

For sparse coding, consider the problem of estimating a column $\mathbf{b}_i$ of B, with D and all $\mathbf{b}_j$ with $j \neq i$ being fixed. Let $\mathrm{E}_i = \mathrm{X} - \sum_{j\neq i}, \mathbf{d}_j\mathbf{b}_j^{\mathrm{T}}$ and $\mathbb{B} = \{\mathbf{v} \in \mathbb{R}^l \mid \|\mathbf{v}\|_\infty < \mathrm{L}\}$ be the set of admissible solutions of $\mathbf{b}_i$, where L is defined in [RNF17] to avoid the

Algorithm 8 Sum of OUter Product Dictionary Learning (SOUPDIL)

```
Input: the input signals X, parameter $\lambda$, the upper bound L, maximal number of
    outer iterations N, the unit-norm vector $\mathbf{v}$ with the first entry one and the rest
    zeros.
Output: The transpose of sparse representation B and dictionary D.
function DICTIONARY LEARNING
    Initialization the dictionary $D_0$ and the matrix $B_0 = 0$
    for $t = 1$ to N do
        for $i = 1$ to $m$ do
            Sparse Coding
            Calculating $E_i$
                $E_i = X - \sum_{j<i} \mathbf{d}_j^t \mathbf{b}_j^{t\,\mathrm{T}} - \sum_{j>i} \mathbf{d}_j^{t-1} \mathbf{b}_j^{t-1\,\mathrm{T}};$
            Updating $\mathbf{b}_i^t$ by
                $\mathbf{b}_i = \min(|T_s({E_i}^{\mathrm{T}} \mathbf{d}_i^{t-1})|, L) \odot \mathrm{sign}(T_s({E_i^t}^{\mathrm{T}} \mathbf{d}_i^{t-1}));$
            Dictionary updating
            Computing $E_i \mathbf{b}_i^t$.
            Updating atom $\mathbf{d}_i^t$ by
                $\mathbf{d}_i = \begin{cases} \frac{E_i \mathbf{b}_i}{\|E_i \mathbf{b}_i\|_2} & \text{if } \mathbf{b}_i \neq 0 \\ \mathbf{v} & \text{else;} \end{cases}$
        end for
    end for
end function
```

non-coercive objective. This leads to the resulting optimization problem

$$\underset{\mathbf{b}\in\mathbb{B}}{\arg\min} \|E_i - \mathbf{d}_i \mathbf{b}_i^{\mathrm{T}}\|_F^2 + \lambda \|\mathbf{b}_i\|_0. \tag{3.30}$$

This problem can be handled by using the IHT algorithm with the threshold set to $s = \lambda$. The closed-form solution is thus

$$\mathbf{b}_i = \min(|T_s(E_i^{\mathrm{T}} \mathbf{d}_i)|, L) \odot \mathrm{sign}(T_s(E_i^{\mathrm{T}} \mathbf{d}_i)),$$

where $\odot$ indicates the element-wise multiplication.

For updating an atom $\mathbf{d}_i$ while fixing all the other variables, the problem becomes

$$\underset{\|\mathbf{d}_i\|^2=1}{\arg\min}\,\|\mathrm{E}_i - \mathbf{d}_i\mathbf{b}_i^{\mathrm{T}}\|_{\mathrm{F}}^2. \tag{3.31}$$

The solution is obtained by

$$\mathbf{d}_i = \begin{cases} \frac{\mathrm{E}_i\mathbf{b}_i}{\|\mathrm{E}_i\mathbf{b}_i\|_2} & \text{if } \mathbf{b}_i \neq 0 \\ \mathbf{v} & \text{else,} \end{cases}$$

where $\mathbf{v}$ is a unit-norm vector. In [RNF17], the first column of the identity matrix is used.

The SOUPDIL algorithm takes advantage of the manner of updating alternatively $\mathbf{d}_i$ and $\mathbf{b}_i$, which is proved to be convergent in a limited number of iterations. We summarize SOUPDIL in Algorithm 8.

3.4.3 Proximal method

The proximal method [PB14] is a tool for dealing with non-smooth, large-scale problems. Furthermore, Bolte et al proved that the proximal method can reach convergent result in non-convex problems [BST14]. Due to the advantages of the proximal method, it is used in image processing and machine learning [BJQS14, JMOB10, LDL15]. Indeed, the proximal method is also widely used in dictionary learning [JMOB10, CPR13], and moreover, it is proved to be globally convergent [BJQS14].

The dictionary learning problem can be rewritten in a generalized formulation of non-convex and non-smooth problem with respect to the two variables A and D

$$\underset{\mathrm{A},\mathrm{D}}{\arg\min}\, f(\mathrm{A}) + q(\mathrm{A},\mathrm{D}) + g(\mathrm{D}), \tag{3.32}$$

where $f = \|\mathrm{A}\|_0$ (or $f = \|\mathrm{A}\|_0 + \delta_{\mathbb{A}}(\mathrm{A})$) and $g(\mathrm{D}) = \delta_{\mathbb{D}}(\mathrm{D})$ are proper lower semi-continuous functions, and $q(\mathrm{A},\mathrm{D}) = \frac{1}{2}\|\mathrm{X} - \mathrm{DA}\|_{\mathrm{F}}^2$ is a differentiable function with a Lipschitz continuous with determined Lipschitz constants. In these expressions, the indicator functions are defined on the sets $\mathbb{A} = \{\mathrm{M} \in \mathbb{R}^{m\times l} \mid \|\mathrm{M}\|_{1,\infty} \leq c\}$ and $\mathbb{D} = \{\mathrm{D} \in \mathbb{R}^{n\times m} \mid \|\mathbf{d}_i\|_2 = 1, \forall i = 1,\ldots,m\}$.

The proximal method solves the optimization problem (3.32) by iteratively updating A and D. Specifically, the linearized proximal minimization problem with respect to A, when fixing D, is

$$\hat{\mathrm{A}} = \arg\min_{\mathrm{A}} f(\mathrm{A}) + \langle \mathrm{A} - \mathrm{A}^{'}, \nabla_{\mathrm{A}} q(\mathrm{A}^{'}, \mathrm{D}') \rangle + \tfrac{t_1}{2} \|\mathrm{A} - \mathrm{A}^{'}\|_{\mathrm{F}}^2, \tag{3.33}$$

where $\nabla_{\mathrm{A}} q(\mathrm{A}^{'}, \mathrm{D}')$ denotes the partial derivative with respect to q of A at point $(\mathrm{A}^{'}, \mathrm{D}^{'})$ and t_1 an appropriate step size. Similarly, the optimization problem with respect to D is

$$\hat{\mathrm{D}} = \arg\min_{\mathrm{D}} g(\mathrm{D}) + \langle \mathrm{D} - \mathrm{D}^{'}, \nabla_{\mathrm{D}} q(\mathrm{A}^{''}, \mathrm{D}') \rangle + \tfrac{t_2}{2} \|\mathrm{D} - \mathrm{D}^{'}\|_{\mathrm{F}}^2, \tag{3.34}$$

where $\nabla_{\mathrm{D}} q(\mathrm{A}^{''}, \mathrm{D}')$ is the partial derivative of q with respect to D at the point $(\mathrm{A}^{''}, \mathrm{D}')$ and t_2 the corresponds to the appropriately chosen step size. Thus, by replacing $\hat{\mathrm{A}}$, $\mathrm{A}^{'}$, $\mathrm{A}^{''}$, $\hat{\mathrm{D}}$, $\mathrm{D}^{'}$ by A^{t+1}, A^{t}, A^{t+1}, D^{t+1}, D^{t} respectively, processing iteratively the problems (3.33) and (3.34) produces a convergent sequence $\{(\mathrm{A}^t, \mathrm{D}^t)\}$ (see proof in [BJQS14]).

In fact the problem (3.33) can be solved by the IHT method and the solution of problem (3.34) can be reached by optimization methods such as the Lagrangian method [Ber97]. Specifically, Bao et al used a block coordinate descent algorithm to get the optimal D^{t+1}. Here, we provide directly the closed-form solutions of the optimization problems without the detail of calculation

$$\begin{cases} \mathrm{A}^{t+1} = \min\left(\mathrm{T}_{\sqrt{2\lambda/t_1}}(\mathrm{A}^t - \frac{1}{t_1}\nabla_{\mathrm{A}} q(\mathrm{A}^t, \mathrm{D}^t)), c\right) \\ \mathbf{d}_j^{t+1} = \mathbf{d}_{m_j} / \|\mathbf{d}_{m_j}\|_2, \forall j = 1, \ldots, m, \end{cases}$$

where $\mathbf{d}_{m_j}$ is calculated by a gradient descent, that is,

$$\mathbf{d}_{m_j} = \mathbf{d}_j^t - \tfrac{1}{t_2} \nabla_{\mathbf{d}_j} q(\mathrm{A}^{t+1}, \mathrm{D}_j^t),$$

where D_j^t is the dictionary in the $(t+1)^{th}$ iteration where all its atoms for $i \le j$ are updated atoms $\mathbf{d}_i^{t+1}$ and the other unchanged $\mathbf{d}_i^t$, namely $\mathrm{D}_j^t = [\mathbf{d}_1^{t+1}, \ldots, \mathbf{d}_{j-1}^{t+1}, \mathbf{d}_{j+1}^t, \ldots, \mathbf{d}_m^t]$.

Besides, the parameters determination is quite important in implementation of proximal method. In general, one should ensure that t_1 and t_2 are not larger than the Lipschitz constant of the gradient function with respect to A and $\mathbf{d}_j$, respectively [PB14]. Hence, for proximal method, it is the direction and descent step, which jointly ensure the reduction in the objective function. This is not the same with

Algorithm 9 Proximal Method for ℓ_0-norm based Dictionary Learning

```
Input: the input signals X, parameter $\lambda$, the step size parameter $t_1$ and $t_2$, the maximal number of outer iterations N.
Output: The sparse representation A and the dictionary D.
function DICTIONARY LEARNING
    Initialization the dictionary $D_0$ and the matrix $A_0 = 0$
    for $t = 1$ to N do
        Sparse Coding
        Calculating $A_t$
            $A^{t+1} = \min\left(T_{\sqrt{2\lambda/t_1}}(A^t - \frac{1}{t_1}\nabla_A q(A^t, D^t)), c\right);$
        Dictionary updating
        Updating iteratively atom $\mathbf{d}_i^t$ by
            $\mathbf{d}_{m_j} = \mathbf{d}_j^t - \frac{1}{t_2}\nabla_{\mathbf{d}_j} q(A^{t+1}, D_j^t);$
            $\mathbf{d}_j^{t+1} = \mathbf{d}_{m_j} / \|\mathbf{d}_{m_j}\|_2;$
    end for
end function
```

greedy and IHT algorithms, since they require a strict condition on the constant of the Restricted Isometry Condition (RIC).

3.5 Conclusion

This chapter presented the methods to solve the optimization problems of sparse coding and dictionary learning for sparse representation. Specifically, the dictionary learning problem, estimating jointly the sparse code and the dictionary, is NP-hard and intractable. The frequently exploited strategy is to iteratively process two alternating phases: sparse coding and dictionary updating. This optimization problem can thus be transformed into two tractable subproblems.

This chapter presented the state-of-the-art algorithms of sparse coding, of dictionary updating and also of dictionary learning. In Section 3.2, the sparse coding algorithms were discussed within three categories: 1) matching pursuit algorithms for dealing with ℓ_0 based problems, 2) relaxation methods by replacing ℓ_0 with ℓ_p, specially the case $p = 1$ addressed with Basis Pursuit and LASSO, and 3) thresholding algorithms that, in fact, cover both the first problem of ℓ_0 penalty or regularization, and the second one, namely LASSO problem and Basis Pursuit. In Section 3.3, the main dictionary updating algorithms were introduced, with the representative algorithm MOD and the Coordinate Descent algorithm based on SVD. In the last section, we presented the main algorithms of dictionary learning that combine the appropriate sparse coding algorithm and dictionary updating algorithm. The well-known algorithms are K-SVD, SOUPDIL, and proximal method.

In the next chapter, we will propose a novel strategy to address the exact ℓ_0-norm optimization problem for dictionary learning.

Chapter 4

Exact ℓ_0-norm for Dictionary Learning

Sommaire

The problem of dictionary learning plays an important role in sparse representation. This problem with quadratic objective function involving two optimisation variables, the sparse code A and the dictionary D, is intractable. To address this problem, one takes usually the strategy of iteratively alternating the two steps of sparse coding and dictionary updating. The subproblem corresponding to the dictionary updating is convex, which can be easily solved by algorithms such as MOD and gradient descent with SVD, as described in Section 3.3. Nevertheless, the subproblem of the sparse coding with the ℓ_0-norm constraint, due to the characteristics of the ℓ_0-norm function, is non-smooth, non-convex, non-differentiable and thus NP-hard [Ela10]. Generally, researchers take strategies like greedy algorithms [PRK93], relaxation optimization techniques (LASSO [MY12] and concave function ℓ_p with $0 < p < 1$ [RSS10]) or iteration shrinking/thresholding algorithms [YPXD09]. However, all these methods risk of missing the global optimal solution [BJQS14, ZXY+15].

In this chapter, we address the sparse coding with its original ℓ_0-norm formulation. To this end, we reformulate the sparse representation problem as a mixed integer programming (MIP) problem [BNCM15]. Thereby, the optimization problem with ℓ_0 constraint is transformed into an optimization problem that can be solved with off-the-shelf optimization software packages, such as CPLEX and Gurobi Optimizer. Moreover, we propose two acceleration techniques that allow to take on real datasets. Finally, we demonstrate the relevance of the proposed MIP based dictionary learning in well-known image denoising tasks.

4.1 Optimized algorithm for exact ℓ_0 penalty problem

This section focuses on the problem (3.26) with Ω the ℓ_0-norm function. We recall the problem that, given a set of l signals $\mathrm{X} = [\mathbf{x}_1, \dots, \mathbf{x}_\ell] \in \mathbb{R}^{n \times l}$ of length n and a given sparsity level k, sparse representation aims to find the optimal a linear combination over an overcomplete dictionary $\mathrm{D} = [\mathbf{d}_1, \dots, \mathbf{d}_m] \in \mathbb{D}$ where $\mathbb{D} = \{\mathrm{D} \in \mathbb{R}^{n \times m} \mid \|\mathbf{d}_i\|_2 = 1, \forall i = 1, \dots, m\}$ and sparse coefficient matrix $\mathrm{A} = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_\ell] \in \mathbb{R}^{m \times l}$, namely by solving the problem

$$\underset{\mathrm{A} \in \mathbb{R}^{m \times l}, \mathrm{D} \in \mathbb{D}}{\arg\min} \; \frac{1}{2}\|\mathrm{X} - \mathrm{DA}\|_\mathrm{F}^2, \quad \text{subject to} \quad \|\boldsymbol{\alpha}_i\|_0 \leq k \; \forall i = 1, \dots, l. \tag{4.1}$$

This optimization problem of estimating simultaneously the two variables A and D is NP-hard. A strategy is to solve A and D alternatingly. When fixing A, D can be effectively obtained by dictionary updating algorithms such as MOD. When fixing D, the optimization problem with respect to A is the sparse coding problem, which has convex objective functions but with a non-convex ℓ_0-norm constraint that makes the optimization problem NP-hard. For the resolution of this problem, methods that rely on a greedy algorithm [MZ93, PRK93, NT09], may produce a local optimal [Tro04, BJQS14], while relaxation formulations, such as with LASSO may sometimes achieve a solution not as sparse as that of the ℓ_0 constrained problem.

In this section, we will explore a novel approach to address the sparse coding problem with the ℓ_0-norm constraint. By reformulating the ℓ_0 constrained problem, it is transformed into a problem of MIP, which makes possible to exactly solve the ℓ_0

based problem.

4.1.1 Mixed integer programming (MIP)

Mixed Integer Programming (MIP) refers to optimization problems where some variables are restricted to be integer while others not. Commonly, MIP refers to optimization problems with linear objective function and under linear constraints, which is thus specially called Mixed Integer Linear Programming (MILP). When the optimization problem has a quadratic objective function, it is called Mixed Integer Quadratic Programming (MIQP). When even the constraints are quadratic, it is given the name Mixed Integer Quadratically Constrained Programming (MIQCP) problem. In the following, we firstly give an introduction of MIP. Then, we reformulate the problem of sparse coding as MIP. Finally, some advanced optimization methods for dealing with MIP problems will be presented.

Introduction of MIP

The history of MIP dates back to the 1940s [Bix12]. Its development relies on the algorithmic improvement on Linear Programming (LP) [Sol07, NW06], like the simplex method [NM65], barrier methods [GMS+86], and the computational capability of hardware. According to [Bix12], from 1988 to 2004, during these sixteen years, the improvement factor exceeds six orders of magnitude. This significant progress in LP solvers has been injected in solving MIP problems. This is possible thanks to techniques like cutting plans, which bridges the gap between MIP and LP [SS06]. Hence, MIP is becoming tractable in real-world problems [SDMFH01]. However, its application is still limited to small size problems [TS04, BE07].

The standard formulation of MIP problem with respect to a variable $\mathbf{v} \in \mathbb{R}^n$ is

$$\begin{aligned}
\operatorname*{arg\,min}_{\mathbf{v}} \quad & \mathbf{v}^{\mathrm{T}} \mathrm{Q} \mathbf{v} + \mathbf{c}^{\mathrm{T}} \mathbf{v} && (4.2)\\
\text{subject to} \quad & \mathrm{A}_{in} \mathbf{v} \le \mathbf{b}_{in} && (4.3)\\
& \mathbf{lb} \le \mathbf{v} \le \mathbf{ub} && (4.4)\\
& \mathbf{v}^{\mathrm{T}} \mathrm{Q}_i \mathbf{v} + \mathbf{c}_i^{\mathrm{T}} \mathbf{v} \le \mathbf{b}_i && (4.5)\\
& \mathbf{v}_{\mathbb{I}} \in \mathbb{Z}, && (4.6)
\end{aligned}$$

where $\mathrm{Q} \in \mathbb{R}^{n \times n}$ is a symmetric matrix and the vector $\mathbf{c} \in \mathbb{R}^n$ corresponds to the linear part of the objective function. The (4.3) ~ (4.6) are constraints. Specifically, (4.3) is a linear inequality constraint where A_{in} is a matrix of n columns with its rows depending on the number of linear inequalities, $\mathbf{b}_{in}$ and $\mathbf{b}_i$ are the right sides of the linear constraint and quadratic constraint respectively, and (4.4) defines the bounds for variable $\mathbf{v}$ where vectors $\mathbf{lb} \in \mathbb{R}^n$ and $\mathbf{ub} \in \mathbb{R}^n$ are respectively given lower bound and upper bound. The constraints determined by the two bounds are fundamental in the complexity sense; without bound constraints, the problem becomes undecidable [Jer73]. In contrast, by raising the lower bounds and reducing the upper bounds, the computation complexity can be easily decreased [NS04]. For (4.5), it represents the quadratic constraint where $\mathrm{Q}_i \in \mathbb{R}^{n\times n}$ and $\mathbf{c}_i \in \mathbb{R}^n$. If $\mathrm{Q}_i = 0$, the quadratic constraint (4.5) degenerates into a linear constraint. The last constraint (4.6) defines the type of each element in $\mathbf{v}$, namely $\mathbb{I}$ denotes a proper subset of $\{1,\ldots,n\}$ that identifies which entries of $\mathbf{v}$ are restricted to be integers.

Specially, if Q is semi-definite positive, the objective function in (4.2) becomes convex. It is noticed that if $\mathrm{Q}_i = 0$, the problem turns into a MIQP. Moreover, if Q is also equal to 0, the problem becomes a MILP.

Reformulation of sparse coding in MIP problem

The first attempt to tackle sparse coding as a MIP problem goes to Jokar and Pfetsch [JP08]. They proposed to use the method of branch-and-cut for getting the exact solution, which was regarded as the criterion to evaluate the performance of the heuristic methods mentioned in [JP08] such as the BP and the OMP. Even though this method is of high computational complexity, which limits its application to small-scale instances. Nevertheless, it is worthy noting that the exact solution can be used to verify the theoretical conclusion, namely when the optimal solution is sparse enough, it can be found by the heuristics, e.g. greedy algorithms. However, for the cases where the optimal solution has more non-zeros, the heuristics perform usually bad. Quite recently, Bourguignon et al reformulated in [BNCM15] the ℓ_0 based sparse coding problem as MIP problem, by replacing the logic relation with a *big-M* reformulation. Thus, MIP solver was applied in decomposition of small scale synthetic data with added noise [BNCM15]. In our work, MIP solver is used for sparse coding, which is further applied in dictionary learning [LCHR19]. Furthermore, this model is not limited to small optimization problems but applied in image processing thanks to two proposed acceleration techniques.

In this section, we focus on the phase of sparse coding of problem (4.1) which is the estimation of optimal A with fixing D. We recall that, by assuming independence of signals in X, the sparse coding problem can be regarded as l subproblems with respect to $\boldsymbol{\alpha}_i$ respectively. For expression simplification, hereafter, $\mathbf{x}_i$ and $\boldsymbol{\alpha}_i$ are expressed by $\mathbf{x}$ and $\boldsymbol{\alpha}$ respectively. Hence the problem is now

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^m}{\arg\min}\,\frac{1}{2}\|\mathbf{x}-\mathrm{D}\boldsymbol{\alpha}\|_2^2, \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_0 \leq k. \tag{4.7}$$

This constrained optimization problem can thus be reformulated, with all the entries of the sparse vector $\boldsymbol{\alpha}$ indicated by a binary variable $\mathbf{z} \in \{0,1\}^m$, which can be explained by the logical relation:

$$\begin{cases} \alpha_i = 0 & \text{if} \quad z_i = 0 \\ \alpha_i \neq 0 & \text{if} \quad z_i = 1 \end{cases}, \tag{4.8}$$

where z_i and α_i indicate the i-th entries of the vectors $\mathbf{z}$ and $\boldsymbol{\alpha}$ respectively, $i = 1,\ldots,m$.

Since such logical relation cannot be easily integrated into the objective function, we recast the sparsity condition into a linear inequality by introducing a sufficiently large value M > 0 ensuring that $\|\widehat{\boldsymbol{\alpha}}\|_\infty < \mathrm{M}$ for any desirable solution $\widehat{\boldsymbol{\alpha}}$, where $\|\cdot\|_\infty$ means the maximal magnitude of the vector's entries. A too large M value will result in an increased feasible region, which will make the problem less computational efficient. An appropriate value of M improves the performance. A method providing a lower value for M to obtain tight bounds is crucial in algorithm improvement.

Now the indicative function of $\mathbf{z}$ is ensured by satisfying the constraints:

$$-z_i\mathrm{M} \leq \alpha_i \leq z_i\mathrm{M}, \quad \forall i \in \{1,\ldots,m\}. \tag{4.9}$$

Then, the sparsity constraint $\|\boldsymbol{\alpha}\|_0 \leq \mathrm{T}$ can be depicted by $\mathbf{z}$ as:

$$\sum_{i=1}^{p} z_i \leq k. \tag{4.10}$$

As a consequence, the ℓ_0-based sparse coding problem can have a 'big-M' reformulation, that is, for a given M large enough:

$$\begin{aligned}\min_{\boldsymbol{\alpha}\in\mathbb{R}^m,\mathbf{z}\in\{0,1\}^m} \quad & \tfrac{1}{2}\|\mathbf{x}-\mathrm{D}\boldsymbol{\alpha}\|_2^2 \\ \text{subject to} \quad & \begin{cases} -\mathbf{z}\mathrm{M} \le \boldsymbol{\alpha} \le \mathbf{z}\mathrm{M} \\ \mathbf{1}_m^{\mathrm{T}}\mathbf{z} \le k, \end{cases}\end{aligned} \tag{4.11}$$

where $\mathbf{1}_m$ is the column vector of size m with all elements equal to one. In this formulation, the optimization variables $\boldsymbol{\alpha}$ and $\mathbf{z}$ are respectively continuous and integer vectors. Hence, the sparse coding problem is reformulated as a mixed-integer programming (MIP) problem.

The above reformulation (4.11) of the sparsity constrained problem (4.7) is valid, as proven by the following proposition.

Proposition 1. *The reformulated problem* (4.11) *is equivalent to its original problem* (4.7).

Proof. For proving the equivalence of the two problems, (4.11) and (4.7), we need to prove both the equivalence of the objective functions and of the feasible regions of $\boldsymbol{\alpha}$ defined by their constraints.

It is obviously that the two objective functions are identical. Thus, it is sufficient to prove the equivalence of the two feasible regions that are defined respectively by

$$\mathrm{A}_1 = \left\{\boldsymbol{\alpha} \in \mathbb{R}^m \mid \|\boldsymbol{\alpha}\|_0 \le k\right\},$$

and

$$\mathrm{A}_2 = \left\{\boldsymbol{\alpha} \in \mathbb{R}^m \mid \mathbf{z} \in [0,1]^m, -\mathbf{z}\mathrm{M} \le \boldsymbol{\alpha} \le \mathbf{z}\mathrm{M},\ \mathbf{1}_m^{\mathrm{T}}\mathbf{z} \le k\right\}.$$

First, we prove that $\mathrm{A}_1 \subseteq \mathrm{A}_2$. Supposing arbitrary $\mathbf{v} \in \mathrm{A}_1$, thus, we have $\|\mathbf{v}\|_0 \le k$. By considering the definition of $\mathbf{z}$, and there exists M sufficiently large and $\mathrm{M} \ge \|\mathbf{v}\|_\infty$, then it yields $-\mathbf{z}\mathrm{M} \le \mathbf{v} \le \mathbf{z}\mathrm{M}$ and $\mathbf{1}_m^{\mathrm{T}}\mathbf{z} \le k$. Hence, $\mathbf{v} \in \mathrm{A}_2$.

Then, we prove the reciprocal, namely $\mathrm{A}_2 \subseteq \mathrm{A}_1$. Let $\mathbf{v}$ be an arbitrary vector in A_2, then it satisfies the condition $-\mathbf{z}\mathrm{M} \le \mathbf{v} \le \mathbf{z}\mathrm{M}$ with $\mathbf{z} \in [0,1]^m$. It is easy to imply that, if $z_i = 0$, the corresponding entry v_i will be equal to 0. By considering the condition $\mathbf{1}_m^{\mathrm{T}}\mathbf{z} \le k$, we have that there are no more than k non-zero entries in $\mathbf{v}$. That is to say $\|\mathbf{v}\|_0 \le k$, namely $\mathbf{v} \in \mathrm{A}_1$.

Finally, by combining $\mathrm{A}_1 \subseteq \mathrm{A}_2$ and $\mathrm{A}_2 \subseteq \mathrm{A}_1$, we conclude that $\mathrm{A}_1 = \mathrm{A}_2$.

□

Moreover, our optimization problem defined in (4.11) with continuous and integer optimization variables, the objective function is quadratic and all the constraints are linear. Consequently, the sparse coding can be interpreted as a MIQP. In the following, we will write this optimization problem in the standard form expressed in (4.2) ~ (4.6) in order to use off-the-shelf solvers. Specifically, in the standard MIQP reformulation of sparse coding problem, $\mathbf{v}$ is obtained by combining the vectors $\boldsymbol{\alpha}$ and $\mathbf{z}$, that is, let

$$\mathbf{v} = (\boldsymbol{\alpha}^{\mathrm{T}}, \mathbf{z}^{\mathrm{T}})^{\mathrm{T}},$$

then the sparse coding problem is of form

$$\begin{aligned} \min_{\mathbf{v}} \quad & \tfrac{1}{2}\mathbf{v}^{\mathrm{T}}\mathrm{Q}\mathbf{v} + \mathbf{c}^{\mathrm{T}}\mathbf{v} \\ \text{subject to} \quad & \mathrm{A}_{in}\mathbf{v} \leq \mathbf{b}_{in} \\ & v_j \in \{0,1\} \quad \forall j \in \mathbb{I}, \end{aligned} \tag{4.12}$$

where the symmetric matrix Q, defining quadratic part of the objective function, of size $2m \times 2m$ is made up of four sub-matrices, namely

$$\mathrm{Q} = \begin{pmatrix} \mathrm{D}^{\mathrm{T}}\mathrm{D} & \mathbf{0}_{m,m} \\ \mathbf{0}_{m,m} & \mathbf{0}_{m,m} \end{pmatrix},$$

with $\mathbf{0}_{m,m}$ the zero matrix of size $m \times m$, and $\mathbf{c}$, defining the linear part in the objective function, is a column vector of size $2m$ that is defined by

$$\mathbf{c} = \begin{pmatrix} -\mathrm{D}^{\mathrm{T}}\mathbf{x} \\ \mathbf{0}_{m,1} \end{pmatrix}.$$

Now, consider the inequality constraint. The matrix of size $(2m+1)\times 2m$ in the linear inequality constraint is of the form

$$\mathrm{A}_{in} = \begin{pmatrix} -\mathrm{I}_m & -\mathrm{MI}_m \\ \mathrm{I}_m & \mathrm{MI}_m \\ \mathbf{0}_m^{\mathrm{T}} & \mathbf{1}_m^{\mathrm{T}} \end{pmatrix},$$

where I_m denotes the identity matrix of size $m \times m$, and the $(2m+1) \times 1$ column vector in the right-hand-size of the inequality is $\mathbf{b}_{in} = (\mathbf{0}_{2m}^{\mathrm{T}}, k)^{\mathrm{T}}$. Finally, the set $\mathbb{I}$ in

(4.12) indicates the integer components in the MIQP, namely

$$\mathbb{I} = \{m+1, m+2, \dots, 2m\}.$$

During the implementation, the variable's type is specialized as continuous or binary (namely variable's type is a parameter in MIQP construction) in the input for the solver at hand.

As an alternatve to interpreting the logical relation (4.8) by the *big-M* reformulation, the indicator function was recently proposed to deal with the logical relation problem, as explained next [BBF+16, BLTW15]. The logical relation (4.8) can be reformulated by introducing the indicator function as

$$\boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{1}_m - \mathbf{z}) = 0,$$

where we can find that, when $z_i = 0$, the entry α_i is forced to be 0 (thus the equality constraint is active), while $z_i = 1$ disables the constraint. Specifically, in this problem, the indicator function $\mathbf{z}$ switches the constraint in a "complementary" way. By adopting this interpretation, the sparse coding problem (4.7) is thus reformulated as

$$\begin{array}{cl} \displaystyle\min_{\boldsymbol{\alpha}\in\mathbb{R}^m, \mathbf{z}\in\{0,1\}^m} & \frac{1}{2}\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 \\ \text{subject to} & \begin{cases} \boldsymbol{\alpha}^{\mathrm{T}}(\mathbf{1}_m - \mathbf{z}) = 0 \\ \mathbf{1}_m^{\mathrm{T}}\mathbf{z} \leq k. \end{cases} \end{array} \tag{4.13}$$

This problem is an MIQP problem but with a nonlinear constraint. It is also called disjunctive programming [BLTW15]. Unlike the *big-M* reformulation of which the difficulty lies on the determination of an appropriate value for M, this reformulation can be solved by exploiting the lift-and-project cutting technique [FLT11]. For some special problems, for instance the SVM problem [BBF+16], the MIQP problem can degenerate into its continuous version. However, this is not the case for the sparse coding problem.

This reformulation risks to highly increase the computational complexity. In Section 4.1.2, we provide two techniques to accelerate the resolution of the above MIQP problem. The first one determines a relevant value of M, thus the *big-M* reformulation can be more efficiently implemented compared to that of disjunctive programming. The second method considers a convex envelop of the feasible continuous variables, which allows to solve the problem more efficiently. But before, we present recent advanced optimization methods for MIP.

Advanced optimization methods for MIP

This section gives an introduction to advanced optimization techniques for solving the MIP problem. By making clear the principle of the resolution of the MIP problem, it will help us in finding the way to accelerate the MIP solver. Thus, it offers some theoretical support of the inspirations and the motivations for the acceleration method we will propose in the following section.

Because of the optimization over both discrete and continuous variables, the MIP problems cannot be solved by common LP solvers, such as simplex method [NM65], dual-simplex method, interior point method [NW06] or barrier method [GMS+86]. An effective method is the branch-and-bound technique [BM91, BM97, BE07].

The branch-and-bound method processes the MIP problem by solving a sequence of relaxed LP problems. These relaxation subproblems are organized in a tree structure. Each node of the tree contains a subproblem. Then the solution searching process [VAN08, BM91] can be summarized as:

1) *Initialization:* Setting the parent node as an LP problem by relaxing all the integer variables to become continuous. The initial upper bound **ub** is set to $+\infty$. This problem can be solved by using a conventional LP solver, such as simplex method [VAN08] or interior point method [BM91].

2) *Stopping criterion testing:* If the active node set is empty, the obtained solution $\hat{\mathbf{v}}$ with $\hat{\mathbf{v}}_{\mathbb{I}} \in \mathbb{Z}$ is optimal and it yields the incumbent objective value which updates the upper bound **ub**. Otherwise, there is no feasible solution to the problem.

3) *Subproblem selection:* By an adequate searching method [LS99], e.g. depth first search strategy [BM91], the subproblem is successively selected and solved.

4) *Node deletion:* When the objective value of the subproblem has the minimal value **lb** > **ub** or there is no feasible solution produced, then this node will be deleted.

5) *Upper bound updating or branching:* When a feasible solution is found with its objective value lb, then if **lb** < **ub**, update **ub** = **lb** [GW13]. However, when

the optimal solution with $v_i,\ i \in \mathbb{I}$ is a fractional number, then branch the problem and continue to solve this subproblem.

The branch-and-bound method makes the MIP problem tractable. However, by only applying this method, the MIP problem is difficult to be solved efficiently. Hence, another optimization technique, cutting plans [MMWW02] is integrated with the branch-and-bound method, which is called branch-and-cut [Mit02, KG08]. The cutting plans method provides a way to relax the original problem [Mit02]. Suppose that the feasible region is a polyhedron defined by all linear inequality constraints. If such a convex hull of feasible region is found, then the problem will turn into an LP problem, which makes it much easier to be solved. The cutting plans offer the method for iteratively generating this convex hull. When the cutting plans method is used in branch-and-bound, it can be applied for all the problems or just a subproblem in certain nodes [KG08].

Branch and cut algorithm makes a theoretically improvement in MIP solver. Moreover, with development in hardware, the computational capability has been greatly improved [BBL14].

It is worthy noting that in MIP solver, the upper bound ub has an important effect on the computational efficiency. A samllest upper bound means fewer subproblems will be selected and solved. Besides, a good relaxation is crucial in approximating the convex hull of the feasible region. These two methods will be discussed in next section where we provide acceleration of MIQP solver for sparse coding.

For implementation of the MIQP solver, various optimization software packages can be explored. The most known are CPLEX and Gurobi Optimizer. IBM develops CPLEX[1] which integrates the latest MIP solvers to solve larger MILP problems [BBL14]. The recently developed Gurobi Optimizer[2] can have an equivalent performance to CPLEX while the latest releases get some improvements [Bix12]. The developed tools make it possible to apply MIQP into image processing, but by considering its computational complexity, some effort should be done to improve it, as described next.

[1] https://www.ibm.com/support/knowledgecenter/SSSA5P_12.9.0/ilog.odms.cplex.help/CPLEX/UsrMan/topics/discr_optim/mip_quadratic/02_introMIQP.html

[2] https://www.gurobi.com/products/gurobi-optimizer

4.1.2 Accelerated mixed integer programming (AcMIQP)

MIQP problem, as aforementioned, can be solved via bound and cut algorithm. However, the problem of computational cost has been raised, especially when dealing with real data, such as real images, even though the most advanced devices show powerful ability in computing. In this section, we propose a method of initialization and a relaxation technique for speeding up the MIQP based sparse coding algorithm.

Initialization by proximal method

A good initialization offers a warm start of the tree search, which will help to reduce the size of the problem. Furthermore, with good initial solution, we can get a tighter upper bound by calculating its corresponding objective value and an adequate value M which defines the scale of feasible region (a lower value of M means a smaller search region). Thus, the initialization method acts importantly in the acceleration of the MIQP solver.

In the following, the proximal method will be used for generating an initial solution [PB14, BJQS14], which is theoretically explicated by solving the problem,

$$\underset{\boldsymbol{\alpha}}{\arg\min} \tfrac{1}{2}\|\mathbf{x}-\mathrm{D}\boldsymbol{\alpha}\|_2^2+\delta_{\mathbb{S}_k}(\boldsymbol{\alpha}), \tag{4.14}$$

where $\delta_{\mathbb{S}}(\boldsymbol{\alpha})$ denotes the indicator function on the set $\mathbb{S}$ of k-sparse vectors, namely $\mathbb{S}_k = \{\mathbf{v} \in \mathbb{R}^m \mid \|\mathbf{v}\|_0 \le k\}$. The proximal method can be regarded intrinsically as finding the minimal upper bound of the objective function by first-order approximation, which is based on the fact that there exists a real value ρ that satisfies

$$\tfrac{1}{2}\|\mathbf{x}-\mathrm{D}\boldsymbol{\alpha}\|_2^2 \le \tfrac{1}{2}\|\mathbf{x}-\mathrm{D}\boldsymbol{\alpha}^i\|_2^2+\langle\boldsymbol{\alpha}-\boldsymbol{\alpha}^i,\mathrm{D}^{\mathrm{T}}(\mathrm{D}\boldsymbol{\alpha}^i-\mathbf{x})\rangle+\tfrac{1}{\rho}\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^i\|_2^2. \tag{4.15}$$

Hence, the solution to the problem (4.14) will be reached by a convergent sequence created by

$$\boldsymbol{\alpha}^{i+1} = \underset{\boldsymbol{\alpha}}{\arg\min}\langle\boldsymbol{\alpha}-\boldsymbol{\alpha}^i,\mathrm{D}^{\mathrm{T}}(\mathrm{D}\boldsymbol{\alpha}^i-\mathbf{x})\rangle+\tfrac{1}{\rho}\|\boldsymbol{\alpha}-\boldsymbol{\alpha}^i\|_2^2+\delta_{\mathbb{S}_k}(\boldsymbol{\alpha}), \tag{4.16}$$

We solve this problem by introducing the proximal operator [PB14] defined by

$$\operatorname{prox}_h^t(\mathbf{u}) = \underset{\mathbf{x}}{\arg\min}\Big(h(\mathbf{x})+\frac{t}{2}\|\mathbf{x}-\mathbf{u}\|^2\Big),$$

where h defines a proper and lower semi-continuous function, and $t > 0$ is a step size parameter. See [BST14] for more detail. In problem (4.16), $h(\mathbf{x})$ is the indicator function $\delta_{\mathbb{S}_k}$ to make sure that the feasible region is in the space $\mathbb{S}_k$ of k-sparsity. The proximal operator boils down to the projection onto the sparse space $\mathbb{S}_k$:

$$\mathrm{P}_{\mathbb{S}_k}(\mathbf{u}) = \underset{\mathbf{x}\in\mathbb{S}_k}{\arg\min}(\|\mathbf{x}-\mathbf{u}\|^2).$$

The solution of this problem can be easily obtained by keeping the k largest absolute value components of $\mathbf{u}$ and setting the other components to zero:

$$\mathrm{P}_{\mathbb{S}_k}(\mathbf{u}) = \begin{cases} u_j & \text{if } j \in \{(1), \dots, (k)\} \\ 0 & \text{otherwise} \end{cases},$$

where j is the index of the sequences that $|u_{(1)}| \geqslant |u_{(2)}| \geqslant \cdots \geqslant |u_{(p)}|$. Specifically, $\mathbf{u}$ in problem (4.16) can be obtained as

$$\mathbf{u} = \boldsymbol{\alpha}^i + \rho \mathrm{D}^{\mathrm{T}}(\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}^i).$$

Thus, by applying a proximal algorithm, the sparse representation problem can be solved through the iterative update process:

$$\boldsymbol{\alpha}^{k+1} \in \mathrm{P}_{\mathbb{S}_k}\Big(\boldsymbol{\alpha}^k + \rho \mathrm{D}^{\mathrm{T}}(\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}^i)\Big). \tag{4.17}$$

After a finite number of iterations n_{iter}, the $\boldsymbol{\alpha}^{n_{iter}}$ will be much closer to the optimal solution of the exact ℓ_0 problem.

By considering the definition of 'big-M', the constraints in the problem (4.11) related to M can be well determined by an approximation of the optimal solution. A simple method to determine an appropriate value for M can be:

$$\mathrm{M} = \eta \|\mathbf{x}^{n_{iter}}\|_\infty. \tag{4.18}$$

The much tighter bound defined by M and an approximate initialization allow to speed up the algorithm. Its performance will be discussed in the following section.

It is noticed that the proximal method for ℓ_0 constrained sparse coding problem has exactly the same operation in the IHT algorithm (Algorithm 5) introduced in Section 3.2, that is, the solution is reached by iteratively processing two alternating steps, gradient descent and projection. However, the two algorithms are built on different theoretically foundations. The IHT [GK09] assumes that the dictionary D satisfies the RIC with the isometric constant $\delta_{2k} < 1/3$. And the step size ρ depends on δ_{2k}. In contrast, the proximal method has no assumption on D, except that the gradient of the function $\frac{1}{2}\|\mathbf{x} - \mathrm{D}\boldsymbol{\alpha}\|_2^2$ with respect to $\boldsymbol{\alpha}$ is Lipchitz continuous with constant L. The step size in the proximal method cannot be larger than this constant.

Relaxation

The developments of the MIQP solvers have been following the progress in LP theory. The advanced-start capabilities of simplex algorithms in the branch-and-bound [vSG00] (or now more correctly, branch-and-cut [NS04]) search tree are well exploited by MIQP solvers. No matter which optimization technique is used, the search process remains the main time consumption factor. The searching time heavily relies on the feasible region determined by the constraints. Hence, the effort spent on getting a good formulation of the constraints do help to accelerate the resolution of the optimization problem.

Hoffman and Ralphs have proven in [HR13] that, if a feasible solution is obtained by a relaxation, then it must also be the optimal solution to the original problem. Especially, in the ideal case, if the convex envelope is found, a mixed integer programming will be transformed to the classical linear programming problem. However, it is an NP-hard problem to find the constraints defining the convex envelop [AOPT13]. The viable strategy is to create a convex envelop of the continuous variables

$$\mathbb{C} = \left\{ \boldsymbol{\alpha} \in \mathbb{R}^m \,\middle|\, \mathbf{z} \in \{0,1\}^m,\ \sum_{j=1}^{m} z_j \le k,\ |\boldsymbol{\alpha}_j| \le z_j k, \right\},$$

by adding the constraint about ℓ_1-norm and ℓ_∞-norm of $\boldsymbol{\alpha}$:

$$\left\{ \begin{array}{rcll} \sum\limits_{i=1}^{m} |\alpha_i| & < & k\mathrm{M} & \\ |\alpha_i| & < & \mathrm{M} & \forall i = 1, \dots, m. \end{array} \right.$$

The absolute values can be formulated as linear programs. To do so, we replace each unrestricted variable α_i, for $i = 1, \dots, m$, with the difference of two restricted variables,

$$\left\{ \begin{array}{l} \alpha_i = \alpha_i^+ - \alpha_i^- \\ \alpha_i^+, \alpha_i^- \geqslant 0, \end{array} \right.$$

namely in matrix form

$$\begin{cases} \boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^- \\ \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^- \geqslant 0. \end{cases}$$

Then, the absolute value of α_i in the above constraints can be represented in the linear program as:

$$|\alpha_i| = \alpha_i^+ + \alpha_i^- \qquad \forall i = 1, \ldots, m.$$

Thus, the constraints for MIQP can be summarized as:

$$\begin{cases} \sum_{i=1}^m & \alpha_i^+ + \alpha_i^- & < k\mathrm{M} \\ -\mathbf{z}\mathrm{M} < & \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^- & < \mathbf{z}\mathrm{M} \\ 0 \leq & \alpha_i^+, \alpha_i^- & < \mathrm{M} \\ & \mathbf{1}_p^{\mathrm{T}}\mathbf{z} & \leq k \quad . \end{cases} \tag{4.19}$$

With the new constraints, MIQP can be reformulated as the standard formulation by introducing as updated optimization variable $\mathbf{v} = (\boldsymbol{\alpha}^{+\mathrm{T}}, \boldsymbol{\alpha}^{-\mathrm{T}}, \mathbf{z}^{\mathrm{T}})^{\mathrm{T}}$. Accordingly, the model components Q, $\mathbf{c}$, A_{in}, $\mathbf{b}_{in}$, $\mathbf{l}$ and $\mathbf{u}$ are updated respectively as following: The matrix Q becomes the $3m \times 3m$ matrix

$$\mathrm{Q} = \begin{pmatrix} \mathrm{D}^{\mathrm{T}}\mathrm{D} & -\mathrm{D}^{\mathrm{T}}\mathrm{D} & \mathbf{0}_{m,m} \\ -\mathrm{D}^{\mathrm{T}}\mathrm{D} & \mathrm{D}^{\mathrm{T}}\mathrm{D} & \mathbf{0}_{m,m} \\ \mathbf{0}_{m,m} & \mathbf{0}_{m,m} & \mathbf{0}_{m,m}. \end{pmatrix}.$$

By this representation, we know Q is symmetric and semi-definite positive. It ensures the convexity of the objective function. The vector $\mathbf{c}$ becomes the vector of size $3m$

$$\mathbf{c} = \begin{pmatrix} -\mathrm{D}^{\mathrm{T}}\mathbf{x} \\ \mathrm{D}^{\mathrm{T}}\mathbf{x} \\ \mathbf{0}_{m,1}. \end{pmatrix}.$$

The linear constraint matrix $\mathrm{A}_{in} \in \mathbb{R}^{(2m+2)\times 3m}$ is now

$$\mathrm{A}_{in} = \begin{pmatrix} \mathbf{1}_m^{\mathrm{T}} & \mathbf{1}_m^{\mathrm{T}} & \mathbf{0}_m^{\mathrm{T}} \\ \mathbf{0}_m^{\mathrm{T}} & \mathbf{0}_m^{\mathrm{T}} & \mathbf{1}_m^{\mathrm{T}} \\ -\mathrm{I}_m & \mathrm{I}_m & -\mathrm{MI}_m \\ \mathrm{I}_m & -\mathrm{I}_m & -\mathrm{MI}_m. \end{pmatrix},$$

and the right-hand-side of the inequality constraint becomes

$$\mathbf{b}_{in} = \begin{pmatrix} k\mathrm{M} \\ k \\ \mathbf{0}_{2m}^{\mathrm{T}}. \end{pmatrix}.$$

The two bounds of the new variables $\mathbf{v}$ are now defined respectively as $\mathbf{u} = (\mathrm{M}\mathbf{1}_{2m}^{\mathrm{T}}, \mathbf{1}^{\mathrm{T}})^{\mathrm{T}}$ and $\mathbf{l} = \mathbf{0}_{3m}$, and

$$\mathbb{I} = \{2m+1, 2m+2, \ldots, 3m\}.$$

With the new formulation, the problem can be solved more efficiently.

Performance of the AcMIQP

In this section, we will give some experimental evaluations of the performance of the proposed algorithm AcMIQP. The experiments are carried out on synthetic data in order to assess the AcMIQP sparse coding algorithm. The classical MIQP solver, proximal method and OMP are chosen for comparison. By analysis on the accuracy of sparse representation and the computational complexity, the advantage of the proposed AcMIQP becomes remarkable, which provides the support to use AcMIQP in the more complex problem of dictionary learning and further apply it in image processing.

For evaluating the AcMIQP solver in solving the sparse coding problem (4.7), a sparse matrix $\mathrm{A} \in \mathbb{R}^{128\times 10000}$ is created with a column-wise maximum sparsity level of 6, and the dictionary $\mathrm{D} \in \mathbb{R}^{64\times 128}$, which is column-wise normalized matrix randomly generated from a Gaussian distribution. With the available matrices A and D, the training data X is finally produced by the following equation

$$\mathrm{X} = \mathrm{DA} + \kappa \mathrm{E}, \tag{4.20}$$

where E is a randomly generated zero-mean white Gaussian noise matrix and κ a parameter controlling the noise level, set to $\kappa = 0.01$ in the experiments.

For statistical purpose, data of size 10000 is divided into 100 units. For each unit $\mathrm{X}_i \in \mathbb{R}^{64\times 100}$, a sparse code matrix A_i is estimated. This allows to provide the median, the 5th and 95th percentiles.

The two MIQP sparse coding algorithms, with and without acceleration, are

Table 4.1 – Computational time and accuracy results (100-batch median, 5th and 95th percentiles) on synthetic data [LCHR19].

	Method Results	OMP	Proximal	MIQP	AcMIQP
Computational time	P_5	0.019	0.023	2002.5	286.98
	median	0.020	0.024	2002.6	415.34
	P_{95}	0.032	0.032	2002.7	543.90
Reconstruction error $\|X_i - D\hat{A}_i\|_F^2$	P_5	14.00	13.20	2.36	**1.58**
	median	15.90	15.14	2.84	**2.74**
	P_{95}	17.45	16.19	**2.86**	3.51
Sparse coding error $\|\hat{A}_i - A_i\|_F$	P_5	39.17	34.68	5.41	**3.60**
	median	44.84	42.33	6.45	**6.06**
	P_{95}	51.13	46.40	**6.46**	7.99
Position accuracy of non-zero elements (%)	P_5	98.16	98.42	99.73	**99.94**
	median	98.52	98.62	99.74	**99.97**
	P_{95}	98.81	98.98	99.78	**99.98**

compared to the OMP and proximal methods. The performances of the sparse coding methods are evaluated with three criteria: the difference between A_i and the estimated $\hat{A}_i$, *i.e.*, $\|A_i - \hat{A}_i\|_F$, the reconstruction error, *i.e.*, $\|X_i - D\hat{A}_i\|_F^2$, and the percentage of zero and non-zero elements of the sparse code being found in the right positions.

Table 4.1 presents the computational time and results, in terms of the reconstruction error, the accuracy of the sparse coding estimation and the percentage of number of zero and non-zero elements being recovered in the right position. It shows that the errors obtained by MIQP are far less than that the ones of OMP and proximal methods. Furthermore, the introduction of optimization techniques (including initialization and relaxation) for acceleration has a little effect on the accuracy, while the computational cost is reduced by a factor of 5. In addition, more non-zero elements are found in the right positions. These advantages make AcMIQP of great interest to be used as a sparse coding algorithm and, in conjunction with a dictionary updating rule, as a dictionary learning algorithm. In spite of the overall strength of MIQP, its Achilles' heel is the excessive computational complexity, making it difficult to use for large-scale problems. However, as aforementioned, the proposed acceleration opens the possibility to apply the MIQP-based dictionary learning algorithm on large-scale problems, such as in image denoising. Note that, in practice, to get an improvement over the proximal method, there is no need to run the optimization until the global minimum. Whatever the computing budget is allocated, the AcMIQP formulation allows to use it to improve the results.

4.1.3 Dictionary learning with AcMIQP for sparse coding

In the previous section, the proposed AcMIQP algorithm was used with success in solving sparse coding problem. The experiments on synthetic data showed the feasibility of the AcMIQP algorithm in processing data with high dimension. In this section, AcMIQP will further be applied in a more sophisticated task: dictionary learning, namely the problem (4.1) given at the beginning of Section 3.4. To this end, the problem is addressed by iteratively processing two alternating steps: sparse coding and dictionary updating.

In the sparse coding phase, the problem (4.7) is reformulated as an MIQP problem. By exploiting the AcMIQP algorithm, the sparse coefficient matrix can be estimated at each iteration. Let A^q be the estimated coefficient matrix at the q^{th} iteration. For updating the dictionary, the problem with respect to D becomes convex, that is, at the q^{th} iteration, there is,

$$\mathrm{D}^q = \underset{\mathrm{D}\in\mathbb{D}}{\arg\min}\, \tfrac{1}{2}\|\mathrm{X} - \mathrm{D}\mathrm{A}^q\|_\mathrm{F}^2.$$

By writing $\mathrm{D}\mathrm{A}^q$ in the form of sum of outer product (3.24), we solve this problem via a coordinate descent algorithm. More precisely, for updating each atom, the method of SVD is applied.(See Section 3.3 for details). Simultaneously, this process updates all the non-zero coefficients in the i^{th} row of A^q. Therefore, the proposed dictionary learning with AcMIQP for sparse coding can be regarded as an exact ℓ_0-norm resolution of the K-SVD algorithm.

The summary of this method is illustrated in Algorithm 10. The proof of convergence of the algorithm is out of scope due to the difficulty in the analysis of the used MIQP solver. However, it is guaranteed that at each step, with sparse coding and with dictionary updating, the objective value reduces.

Algorithm 10 Dictionary learning algorithm via AcMIQP.

Input: Signals for training X, target sparsity k, step size ρ for updating approximate A by proximal method, coefficient η for optimizing M, number of iteration for dictionary learning N_d and for proximal method N_p

Output: Optimized dictionary D and sparse coefficient matrix A.

```
Initialization of the dictionary $D^0$ and the coefficients matrix $A^0 = 0$,
for $q = 1$ to $N_d$ do
    Initializing X by proximal method:
    $A^{q-1,0} = A^{q-1}$
    for $p = 1$ to $N_p$ do
        Updating sparse approximation via
        $$A^{q-1,p} = \mathrm{prox}_{\delta_{\mathbb{S}_k}}\Big(A^{q-1,p-1} - \rho\big((D^{q-1})^T D^{q-1} A^{q-1,p-1} - (D^{q-1})^T X\big)\Big)$$
    end for
    $A^{q-1} = A^{q-1,N_p}$
    $M = \eta \|A\|_{1,\infty}$,
    Optimization of A via MIQP solver:
    for $i = 1, \dots, l$ do
        Initialization of $\boldsymbol{\alpha}_i^+$, $\boldsymbol{\alpha}_i^-$, $\mathbf{z}_i$ and then $\mathbf{v}$,
        $\boldsymbol{\alpha}_i^+ = \max(0, \boldsymbol{\alpha}_i^{q-1})$
        $\boldsymbol{\alpha}_i^- = \max(0, -\boldsymbol{\alpha}_i^{q-1})$
        $\mathbf{z}_i = \mathrm{abs}(\mathrm{sign}(\boldsymbol{\alpha}_i^{q-1}))$
        $\mathbf{v} = (\boldsymbol{\alpha}_i^{+T}, \boldsymbol{\alpha}_i^{-T}, \mathbf{z}_i^T)^T$
        Solving MIQP problem
        $$\begin{aligned} \min_{\mathbf{v}} \quad & \tfrac{1}{2}\mathbf{v}^T Q \mathbf{v} + \mathbf{c}^T \mathbf{v} \\ \text{subject to} \quad & A_{in}\mathbf{v} \le \mathbf{b}_{in}, \\ & \mathbf{l} \le \mathbf{v} \le \mathbf{u}, \\ & v_j \in \{0,1\}, \quad \forall j \in \mathbb{I}. \end{aligned}$$
        Computing the solution of $\boldsymbol{\alpha}_i^q = \mathbf{v}[1:m] - \mathbf{v}[m+1:2m]$
    end for
    Update $D^q$ with the coordinate descent algorithm (introduced in Algorithm 7)
end for
```

4.2 Image denoising based on the exact ℓ_0

In the previous section, we introduced the AcMIQP algorithm for solving problem based on exact ℓ_0-norm. Furthermore, a dictionary learning method by applying AcMIQP algorithm for sparse coding was proposed. We showed that the good performance in the signal recovery and that the learned dictionary is of good quality. In this section, we address image processing with real well-known natural images, such as Barbara and Lena.

Sparse representation is proved to be a good model in image processing, including image denoising [EA06], image inpainting [MSE08], and image deblurring [DZSW11]. The performance of sparse model on application of image processing becomes an important criterion for evaluation of dictionary learned by solving specialized sparse representation problem. Thus, hereafter, the proposed AcMIQP for exact ℓ_0 based dictionary learning will be used for image denoising on well-known images.

4.2.1 Sparse representation for image denoising

In a natural image, spatial smooth structures appear much more frequently than highly non-smooth and discontinuous structures. This fact acts as a proof of sparseness in image representation [EFM10]. Hence, image can be formulated as a linear combination of some few atoms of dictionary. However, because of imperfection in imaging device, poor illumination and information loss during signal transmission, the observed image is contaminated by noise [Ela10]. Thus, the application of image denoising is of great interest, and moreover, being an inverse problem, it provides an evidence for testing image processing algorithms and techniques, with a particular interest in demonstrating the relevance of sparse representations.

Considering an image $\mathbf{x}$ contaminated by a white Gaussian noise with zero-mean and standard deviation σ, the observed image can be expressed by

$$\mathbf{y} = \mathbf{x} + \mathbf{e},$$

where $\mathbf{y}$ indicates the image of observation and $\mathbf{e}$ the Gaussian noise. When applying sparse representation model, the image $\mathbf{x}$ can be modeled as a linear combination of only a few atoms in a dictionary D, namely $\mathbf{x} = \mathrm{D}\boldsymbol{\alpha}$ with $\|\boldsymbol{\alpha}\|_0 \leq k$ where $\boldsymbol{\alpha}$ denotes the sparse coefficient vector and D can be explained as "atomic" images (which is interpret as periodic-table in field of chemistry [EFM10]). By using this relation, the observed image $\mathbf{y}$ can be modeled by

$$\mathbf{y} = \mathrm{D}\boldsymbol{\alpha} + \mathbf{e}.$$

With knowledge on the energy bound of noise, namely $\|\mathbf{e}\|_2 \leq \epsilon$, where ϵ is defined by $\epsilon = 1.15n\sigma^2$ in [EFM10], the problem of image denoising can be formulated as

$$\arg\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \quad \text{subject to } \|\mathbf{y} - \mathrm{D}\boldsymbol{\alpha}\|_2^2 \leq \epsilon. \tag{4.21}$$

Therefore, this problem consists in finding the sparsest representation of the underlying image $\mathbf{x}$ under the condition that the reconstructed one obtained by $\mathrm{D}\boldsymbol{\alpha}$ has no more distance than ϵ with the noisy observation $\mathbf{y}$. In fact, the representation is quite sparse, namely $\|\boldsymbol{\alpha}\|_0 \leq k$. In this way, the image will be component-wise denoised which is proved to be more efficient [HHO99].

Alternatively, one can formulate the image denoising problem by adopting the Lagrangian function associated to the constrained problem (4.22), that is,

$$\arg\min_{\boldsymbol{\alpha}} \lambda \|\boldsymbol{\alpha}\|_0 + \tfrac{1}{2}\|\mathbf{y} - \mathrm{D}\boldsymbol{\alpha}\|_2^2, \tag{4.22}$$

where λ is the parameter balancing the data-fitting and the sparsity of $\boldsymbol{\alpha}$. The bigger λ is, the sparser the representation $\boldsymbol{\alpha}$ is, which may remove some useful structural information of the image, in addition to noise. As a result, an adequate λ setting is important for achieving good performance in image denoising.

Besides, the corresponding sparsity constrained problem can be also reformulated as

$$\arg\min_{\boldsymbol{\alpha}, \|\mathbf{y}-\mathrm{D}\boldsymbol{\alpha}\|_2^2 \leq \epsilon} \ \tfrac{1}{2}\|\mathbf{y} - \mathrm{D}\boldsymbol{\alpha}\|_2^2, \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_0 \leq k. \tag{4.23}$$

where the sparsity level k should be explicitly prefixed by the user.

Problems (4.21), (4.23) and (4.22) are exactly the problems of sparse representation with ℓ_0 promoting sparsity. When the dictionary D is given, these problems can be solved via algorithms introduced in Section 3.2. If the optimal value $\hat{\boldsymbol{\alpha}}$ is obtained, then, the underlying image can be constructed directly by

$$\hat{\mathbf{x}} = \mathrm{D}\hat{\boldsymbol{\alpha}}. \tag{4.24}$$

In the above discussion, the dictionary D is assumed to be known. However, in practice, the determination of D plays an important role in image denoising [ZXY+15]. At the beginning of the emergence of sparse representations, dictionaries were predefined. Some special designed wavelet transforms [CYV00a] are used for forming a dictionary, for instance, Gabor wavelet [Lee96] contains diverse shifts of all entries in an orthogonal wavelet which will produce a shift-invariant representation; or curvelet [SCD02] is used for extracting discontinuous structures of an image. The application of those dictionaries gained significant improvement in image denoising (see [EFM10] and therein). Then, with the theoretical development in dictionary learning [MBPS09, AEB+06, BJQS14], data-driven dictionaries [EA06] have been extensively used for image denoising.

Learned dictionary processes an image not as a whole but on small patches (8×8 is often chosen to conduct the experiments). Then by vectorization of all the pixels in the patches, the matrix of signals $\mathrm{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_i, \dots, \mathbf{y}_\ell]$ is produced. Then, taking Y as the input signals, the optimal sparse representation A and dictionary D can be learned as mentioned in Section 3.4. Each underlying patch can be recovered by

$$\mathrm{X} = \mathrm{DA}. \tag{4.25}$$

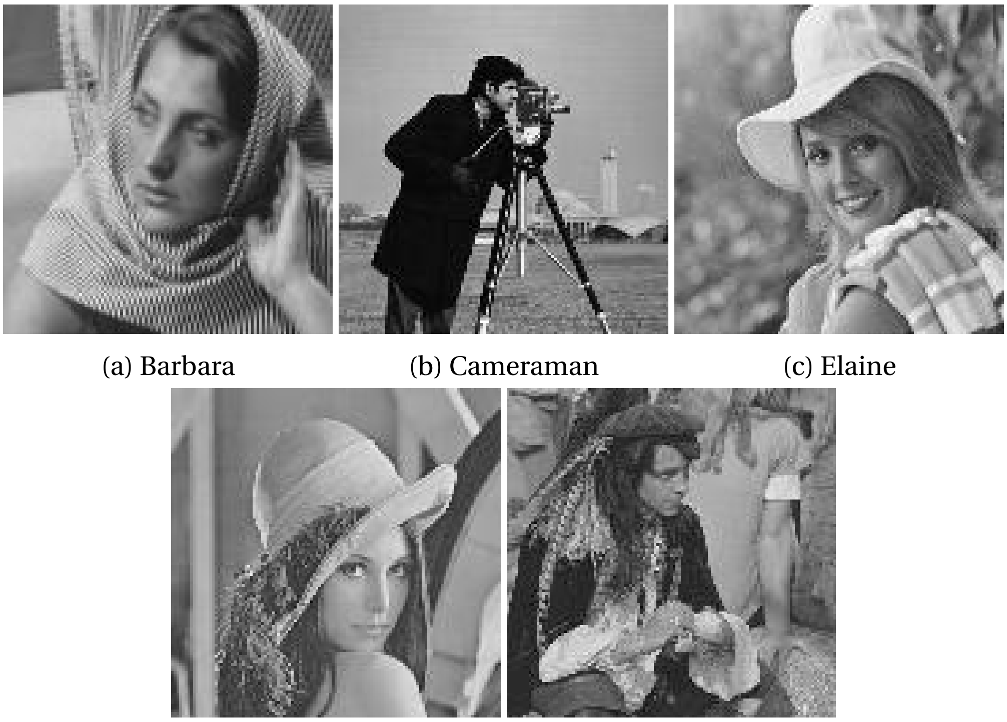
Then by averaging the pixels value, the whole image can be reconstructed.

The above method reconstructs a whole image by removing noise in each patch. Moreover, Elad et al [EA06] proposed a method for handling directly the entire image, by formulating the problem of image denoising as

$$\underset{\mathrm{A,X,D}}{\arg\min}\ \lambda\|\mathrm{X}-\mathrm{Y}\|_2^2 + \sum_i \omega_i \|\boldsymbol{\alpha}\|_0 + \sum_i \|\mathrm{D}\boldsymbol{\alpha}_i - \mathrm{R}_i\mathrm{X}\|_2^2, \tag{4.26}$$

where the first term $\|\mathrm{X}-\mathrm{Y}\|_2^2$ measures the proximity of the observed image Y and its reconstructed one X, and R_i is the matrix that projects the whole image to the i^{th} patch. This problem is solved by firstly applying the K-SVD algorithm to learn an optimal dictionary D. Then, with the known D, the method alternatively solves the problem with respect to A with fixed X, and with respect to X with fixed A. The closed-form solution of X is given by

$$\mathrm{X} = \Big(\lambda \mathrm{I} + \sum_i \mathrm{R}_i^{\mathrm{T}}\mathrm{R}_i\Big)^{-1}\Big(\lambda \mathrm{Y} + \sum_i \mathrm{R}_i^{\mathrm{T}}\mathrm{D}\boldsymbol{\alpha}_i\Big), \tag{4.27}$$



(a) Barbara (b) Cameraman (c) Elaine

(d) Lena (e) Man

Figure 4.1 – Examples in the USC-SIPI Image Database

where λ is set to $30/\sigma$ as recommended in [EA06].

In addition to the K-SVD method, sparse representation on image denoising has also been addressed by methods like BM3D [DE07, KD09], and it has been recently surpassed by convolutional neural networks [ZZC$^+$17]. However, the experiments on image denoising still provide a good assessment on sparse representation algorithms [Buc14].

In this section, we choose segments of natural images in the USC-SIPI Image Database [3] for experiments. The dataset contains five frequently used images in signal and image processing, as presented in Figure 4.1. The images are of size 121 × 121. The images in all experiments are corrupted with an additive zero-mean white Gaussian noise.

The experiments are conducted with the Gurobi solver and the parameters settings of Gurobi are: TimeLimit 50 and IterationLimit 500. For initialization with the proximal method, the number of iterations is set to 200. The coefficient in (4.18) to decrease M is set to $\eta = 2.5$. To assess the quality of denoising an image Y, we

[3] http://sipi.usc.edu/database/database.php?volume=misc

consider the peak signal-to-noise ratio (PSNR), namely

$$\mathrm{PSNR} = -10\log\frac{\|\mathrm{X}-\mathrm{Y}\|^2}{255^2},$$

where X denotes the reconstructed image.

4.2.2 Large-scale dictionary learning

This part considers the set of high-quality images in order to construct a unique global dictionary that will serve to denoise every image. More than $\ell \approx 1.6 \times 10^4$ overlapping patches of size $n = 8 \times 8$ from the images are extracted to get a single training dataset denoted Y. The number of atoms is set to $p = 256$ and the sparsity level is $k = 20$ (these parameters are determined by preliminary experiments and corroborated by other studies, such as [EA06]).

The dictionary learning algorithm with AcMIQP for sparse coding and coordinate descent algorithm for dictionary updating (Algorithm 10) is executed for 30 iterations to learn the global dictionary. This number of iterations is more than enough for convergence, as illustrated in Figure 4.2. This figure illustrates the learning curve, namely the evolution of the objective value at each iteration. It shows how the AcMIQP based dictionary learning algorithm converges faster than the other methods, namely K-SVD with OMP, proximal method and SOUPDIL algorithm. Moreover, the limit objective value (here indicating the value obtained after 30 iterations, which is proved to be more than sufficient for converging) of the AcMIQP based algorithm is the smallest comparing to the other methods. Consequently, we can conclude that AcMIQP can yield more exact solution with few iterations.

To measure the quality of the dictionaries, we consider the correlation, measured with the inner product between each pair of atoms of dictionary, thus measuring how much two atoms in the dictionary are similar. This fundamental information allows to define more powerful measures, such as the coherence and Babel function [Tro04, Hon15a]. The coherence measure of a given dictionary, defined by the maximum absolute inner product between two distinct atoms, provides strong insights on the capacity of the dictionary to recover sparse signals. For instance, it is shown in [Tro04] that a μ-coherence dictionary can recover a k-sparse signal if $\mu < \frac{1}{2k-1}$. It is well known that the OMP algorithm (e.g. K-SVD) often provides dictionaries with high coherence, and most atoms are highly correlated. To over-

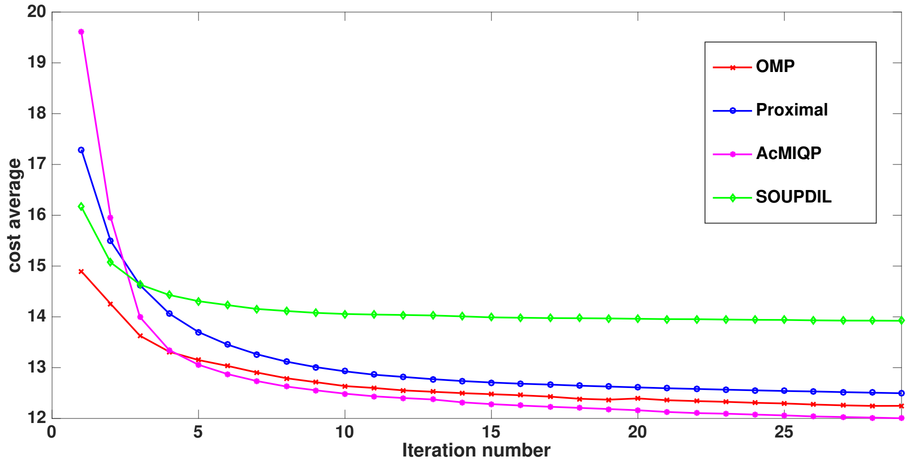


Figure 4.2 – Convergence of the proposed algorithm and the comparison with K-SVD using OMP, proximal method and SOUPDIL

come this issue, several strategies have been proposed to provide more incoherent dictionaries. This issue will be addressed in the next chapter.

Figure 4.3 provides the histogram of the correlation between each pair of atoms of the learned dictionaries, for each of the four methods under investigation. It is observed that the distribution of the correlations of the obtained dictionaries can be roughly ordered as follows

$$\text{SOUPDIL} \prec \text{AcMIQP} \prec \text{Proximal method} \prec \text{K-SVD}.$$

For the purpose of studying the properties of the learned dictionary, the sparsity-constrained formulation (4.23) and the error-constrained formulation (4.21) are respectively investigated to learn the sparse code for reconstruction. The sparsity-constrained formulation (4.23) defines a sparse coding problem with a predefined sparsity parameter k. Considering the error-constrained optimization problem (4.21), it is easy to make OMP satisfy the sparsity constraint by measuring the reconstruction error each time after adding a non-zero entry [EA06]; The proximal method will search for the Pareto optimal when the sparsity level varies [SIDB15]; AcMIQP keeps all the signals in the constraint based on the decided sparsity of initialization obtained by the proximal method. As recommended in [EA06], $\epsilon = c\,n\,\sigma^2$ with $c = 1.15$ and a maximum sparsity parameter k_m (usually the same value as k) set to assure the sparse level.

In order to understand the influence of the noise level on the results of the proposed method, we consider additive Gaussian noise of different standard deviations

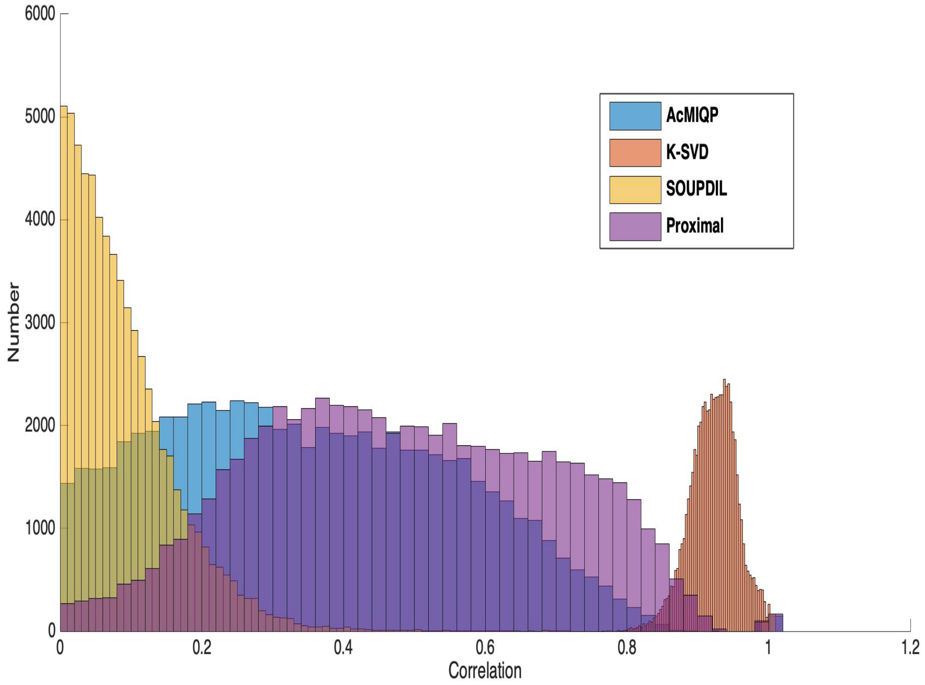


Figure 4.3 – Histogram of the correlation of the learned dictionaries

($\sigma = 10, 20, 50$ in the experiments). The reconstruction accuracy is given in Table 4.2 in terms of the PSNR. These results to evaluate sparse coding show that the sparsity-constrained formulation (4.23) always outperforms the error-constrained formulation (4.21) when $\sigma = 10$ (with an average improvement of 2.73dB) and $\sigma = 20$ (with an average improvement of 1.95dB). At high noise level with $\sigma = 50$, their performances are comparable.

In tests, we compare the following methods: sparse coding using OMP, proximal method and AcMIQP, and dictionary updating with coordinate descent algorithm introduced in the K-SVD algorithm. The performance of the reconstruction accuracy is given in Table 4.3. These results show that the proposed method outperforms OMP and proximal methods in a high noise level. For the large-scale (global) dictionary learning setting, MIQP provides important enhancements, with an average improvement of 1.79 with regard to the proximal method, and 3.73 with regard to the OMP algorithm. It is worthy noting that the enhancement is significant since, the parameters were optimized for OMP, as recommended in [EA06]. According to the experiments, we can conclude that the AcMIQP can reconstruct the signal more accurately with a high tolerance to noise.

Table 4.2 – Accuracy of the denoising in terms of the PSNR in the large-scale (global) dictionary learning, for each of the five images at several noise levels, comparing the sparsity-constrained formulation (4.23) and the error-constrained formulation (4.21)(the higher, the better)

Image	Sparse coding formulation	σ =10	σ =20	σ = 50
Barbara	error-constrained	24.71	23.75	20.79
	sparsity-constrained	26.77	25.24	20.14
Cameraman	error-constrained	24.93	23.90	20.16
	sparsity-constrained	27.70	25.75	20.19
Elaine	error-constrained	26.78	25.64	21.57
	sparsity-constrained	29.87	27.81	21.14
Lena	error-constrained	26.05	24.98	21.22
	sparsity-constrained	28.83	26.93	20.92
Man	error-constrained	24.67	23.68	20.80
	sparsity-constrained	27.60	25.97	20.10
Average	error-constrained	25.43	24.39	20.91
	sparsity-constrained	28.15	21.15	20.49

4.2.3 Adapted dictionary learning

In this part, the dictionary is trained on the corrupted image under scrutiny, and then used to denoise it; the dictionary is then "adapted" to the image at hand. As in the last section, the signal matrix is created in the same way using overlapping patches. For each corrupted image $\widetilde{X}$, an adapted dictionary is trained on it and then used for denoising the same image.

All three methods, OMP, proximal method and SOUPDIL, are compared with the proposed AcMIQP based dictionary learning method. Moreover, we consider also a variant of K-SVD with OMP, where the signals are pre-centered (subtracting the image mean) prior to learning the dictionary [MBP$^+$14]; connections between centered and uncentered data are studied in [Hon16]. In the experiments, SOUPDIL is

Table 4.3 – Accuracy of the reconstruction in terms of the PSNR (the higher, the better)

Image	Method	Large-scale dictionary learning, reconstruction with (4.25)	Large-scale dictionary learning, reconstruction with (4.27)
Barbara	OMP	19.71	20.03
	Prox	20.71	21.03
	AcMIQP	22.73	22.73
Cameraman	OMP	19.46	19.78
	Prox	21.11	21.43
	AcMIQP	22.30	22.62
Elaine	OMP	19.73	20.05
	Prox	22.87	23.19
	AcMIQP	24.20	24.52
Lena	OMP	19.79	20.11
	Prox	22.12	22.44
	AcMIQP	24.20	24.52
Man	OMP	19.68	20.00
	Prox	21.26	21.58
	AcMIQP	23.59	23.91
Average	OMP	19.674	19.994
	Prox	21.614	21.934
	AcMIQP	23.404	23.724

implemented using the original Matlab code provided by its authors and available here[4]. For the other three methods, the experiments details and the parameter settings are given in the following. When dealing with noisy data in the training phase, the knowledge about the noise level σ is used for restricting the reconstruction error, as shown in the constraint in the optimization problem (4.23) and the parameter setting $\epsilon = cn\sigma^2$ with $c = 1.15$. These values, optimized for OMP in [EA06], are used here for both proximal method and AcMIQP, thus putting our method in a less favorable situation.

In this part, the error-constrained optimization problem (4.23) is used for sparse coding. The method of realization is described in the large-scale dictionary learning. In order to ensure the sparsity of the signals, the upper bound k is set to 20

[4] https://gitlab.eecs.umich.edu/fessler/soupdil_dinokat

Table 4.4 – Denoising results in the adapted dictionary learning setting, for each of the five images, as well as the average results (the higher, the better)

Image	Method	PSNR
Barbara	OMP	22.04
	OMP (pre-centering)	21.97
	proximal	22.54
	SOUPDIL	22.30
	AcMIQP	**22.59**
Cameramen	OMP	22.54
	OMP (pre-centering)	22.63
	proximal	22.49
	SOUPDIL	**22.79**
	AcMIQP	22.58
Elaine	OMP	23.00
	OMP (pre-centering)	22.91
	proximal	23.29
	SOUPDIL	**23.43**
	AcMIQP	23.39
Lena	OMP	22.48
	OMP (pre-centering)	22.51
	proximal	23.08
	SOUPDIL	**23.20**
	AcMIQP	23.09
Man	OMP	21.23
	OMP (pre-centering)	21.32
	proximal	21.70
	SOUPDIL	21.67
	MIQP	**21.86**

for the proximal and AcMIQP methods, as in the first setting. By fixing the dictionary updating method to coordinate descent algorithm (Algorithm 7, SVD method is used for updating each atom in the dictionary) in all the methods, this allows to have a fairly comparable setting to analyze and compare the performance of the sparse coding methods. The number of atoms is set to $p = 256$ for OMP, as suggested in [EA06] where extensive experiments were conducted. The number of atoms for the proximal method is set to $p = 65$, which is obtained from a set of 14 candidate values $\{50, 55, 60, 65, \ldots, 110, 150, 200, 256, 300\}$ that encloses the most used values in the literature. The same value is used for AcMIQP, which is a less favorable situation for our method. The total number of iterations is still 30 for the two-step sparse coding and dictionary updating. The SOUPDIL method uses the same parameter setting as recommended in [RNF17] after extensive experimental analysis.

With the learned dictionaries, the same reconstruction model (4.27) is used for obtaining the denoised image. Table 4.4 gives the denoising accuracy in terms of PSNR by using the three aforementioned dictionary learning methods. We notice that the influence of data pre-centering is not always positive. It is observable that MIQP can outperform the K-SVD and proximal methods almost in all cases. On average over all five images, the proposed method carries out an improvement of 0.45 with respect to OMP, and 0.08 with respect to the proximal method. These improvements are important since, on one hand, PSNR is a logarithmic-scale measure and, on the other hand, the parameters were optimized for OMP (*e.g.* ϵ, $c = 1.15$, $p = 256$) and for the proximal method ($p = 65$). Even compared with the state-of-the-art dictionary learning algorithm SOUPDIL, MIQP has comparable performance.

4.2.4 Analysis on the computational complexity

In spite of the great performance of the proposed AcMIQP method on all images and compared to all the other methods, it has high computational complexity in implementation. Because we have different sizes of the training data in each setting (global dictionary learning and adapted dictionary training for each image), the training time is not comparable. In the following, we focus on the average time of a single image. While the OMP algorithm and the proximal method require only a couple of minutes for completing the dictionary learning, AcMIQP needs about one hour. See also Table 4.1 for results obtained on synthetic data. However, recent advances in MIQP solvers allow to reduce this gap.

Indeed, while the computational complexity remains the Achilles' heel of such methods, great improvements are being carried out these days on MIQP solvers. For instance, the new Gurobi Optimizer v8.1 (released in October 2018) is 2.8 times faster overall on MIQP problems, than v8.0 (released in May 2018), which is more than 220% faster than the one used in this thesis (v7.0 released in October 2016). Moreover, new advances in solvers are exploiting more and more the modern architectures and multi-core processors. Finally, currently available off-the-shelf solvers, such as Gurobi and CPLEX, do not have GPU implementations, which could also provide important computational improvements.

4.3 Conclusion

In this chapter, we proposed an exact optimization method AcMIQP for the exact ℓ_0 based sparse coding. Thanks to recent advances in linear programming techniques, as well as more powerful hardware, the speed of solving MIQP problems has been greatly improved. Furthermore, by introducing additive constraints and an appropriate proximal initialization, it was proved that it is feasible to use MIQP for sparse coding with the proposed AcMIQP method. In conjunction with a dictionary update, such as coordinate descent method used in K-SVD, we proposed a dictionary learning algorithm with exact ℓ_0 based sparsity. Though, the AcMIQP method had much more time complexity in implementation comparing with the approximate methods, the feasibility of the method was proved for large-scale well-known images. Moreover, the image denoising experiments showed the advantage of the proposed AcMIQP method. Furthermore, the high noise-tolerance of our method was demonstrated on both the large-scale and the adapted dictionary learning settings.

Therefore, we have demonstrated that the exact ℓ_0 optimization problem in dictionary learning can be solved for image processing, working on real images. While having good performance amelioration, the Achilles' heel of the proposed method is the computational complexity. However, great improvements are being carried out these days on MIQP solvers, with more than 220% speed enhancement in a single year (*e.g.* Gurobi Optimizer v8.0 versus v7.0).

In the following chapter, we will study the problem of dictionary learning by considering learning incoherent dictionary. For handling the resulting constrained optimization problem, we will exploit the classic strategy of iteratively alternating two steps, sparse coding and dictionary updating. With the similarity of optimization problem formulation in sparse coding, the proposed AcMIQP algorithm will still be used. When updating the dictionary, a new algorithm will be proposed to address the incoherence constraint.

Chapter 5

Incoherent Dictionary Learning

Sommaire

The coherence of a dictionary corresponds to the largest correlation between its atoms (*e.g.* null coherence for dictionaries of orthogonal elements). Beyond being elementary and very simple to compute, the coherence is intimately related to the sparsity level and the relevance of the resulting sparse representation. For example, the uniqueness of solution in the sparse representation problem lies on the condition of incoherence of the dictionary [GJB15, Tro04, DH01]. Moreover, the coherence was used as a criterion in evaluation of the dictionary learned based on the proposed AcMIQP algorithm (in Section 4.1.2). In sparse representation, the coherence measure is crucial and acts as a fundamental measure to characterize a dictionary.

Indeed, several theoretical studies have demonstrated the prominence of having incoherent dictionaries, namely dictionaries having a low coherence measure [Hon15b, Hon15a, Tro04]. Incoherent dictionary learning, as an extension of the generic dictionary learning, aims at minimizing the reconstruction error by imposing simultaneously the sparsity on the coefficients and the incoherence of the dictionary. For this purpose, several incoherent dictionary learning algorithms have been proposed, within three major strategies: either adding a decorrelation step after dictionary updating at each iteration, such as INK-SVD and related algorithms [MBP12, DM13], or introducing an additional regularization on the coherence in the optimization problem [RLS09, BQJ14].

In this chapter, we formulate the dictionary learning problem with an explicit constraint on the coherence. Thus, we consider solving an optimization problem with quadratic objective function and quadratic constraints based on the ℓ_0-norm. As the generic dictionary learning, the solution can be reached by iteratively processing two alternating processes: sparse coding and dictionary updating. For sparse coding, we use the proposed AcMIQP algorithm. For the dictionary updating, we propose a novel resolution method that combines the alternating direction method of multipliers (ADMM) and the method of extended proximal alternating linearized minimization. The relevance of the proposed incoherent dictionary learning method is demonstrated with experiments on real data, compared to well-known methods. Moreover, we provide a theoretical analysis on the convergence.

5.1 Coherence and sparse representation

Sparse representations require finding the sparest code of the given signal over a determined dictionary. The value of the coherence of the dictionary has a significant effect in the resolution of sparse coding and in the estimation of the dictionary, as well as the evaluation of its quality. The coherence is thus obviously a major tool for the dictionary analysis and synthesize in sparse representations.

5.1.1 Definition of coherence

The coherence is defined as the greatest correlation, in absolute value, between two distinct atoms of the dictionary under scrutiny. When dealing with unit-norm atoms, the coherence of a given dictionary $\mathrm{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_m] \in \mathbb{R}^{n\times m}$, is defined as ,

$$\mu = \max_{i\neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle| = \max_{i\neq j} |\mathbf{d}_i^{\mathrm{T}} \mathbf{d}_j|. \tag{5.1}$$

For a dictionary, the coherence is a fundamental quality of assessment which estimates how much two atoms in dictionary are correlated. Specially, when $\mu = 0$, the dictionary become orthogonal. For redundant representations, the dictionary is overcomplete with $m > n$, then, the coherence will be bounded by

$$\mu \geq \sqrt{\tfrac{m-n}{n(m-1)}}. \tag{5.2}$$

The limit of dictionary coherence is met when the dictionary is an optimal Grassmannian frame [WGL17, SWDS15]. Consider for example, a dictionary of 256 atoms living in a 64 dimensional space, its coherence cannot be less than 0.108, i.e., the angle between any two distinct atoms of the dictionary cannot be bigger than 83.8°.

Letting $\mathrm{G} = \mathrm{D}^{\mathrm{T}}\mathrm{D}$ be the Gram matrix of D, then the coherence of D is obtained by searching the maximal absolute value of off-diagonal elements in G, namely $\mu = \sup_{i\neq j} |\mathrm{G}_{ij}|$. The coherence of the dictionary is also related to the RIC defined in Chapter 1. The relationship is addressed in [Hon15a] with

$$\delta_k = (m-1)\mu.$$

The coherence considers only the extreme correlation of atoms rather the global situation. For avoiding this shortcoming, another definition called *cumulative coherence* (or *babel function*) is created [Tro04]

$$\mu_1(g) = \max_{|\mathbb{J}|=g} \max_{i\notin \mathbb{J}} \sum_{\mathbb{J}} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle|, \tag{5.3}$$

for any integer g, and where $\mathbb{J}$ is a subset of $\{1, \ldots, m\}$. This function is non-decreasing. When the function varies little with g increasing, the dictionary is more likely incoherent. For a given dictionary, its coherence μ and cumulative coherence $\mu_1(g)$ satisfy the relation $\mu_1(g) \leq g\mu$.

5.1.2 Coherence of dictionary in sparse representations

The dictionary coherence has a great effect in sparse representations. First of all, the uniqueness of solution to the problem of sparse representation can be ensured by satisfying a condition on μ, that is,

Theorem 5.1.1. *(Uniqueness-Coherence [Ela10]) If a problem of signal estimation* $\mathbf{x} = \mathrm{D}\boldsymbol{\alpha}$ *has a solution* $\boldsymbol{\alpha}$ *with* $\|\boldsymbol{\alpha}\|_0 < \frac{1}{2}(1+\frac{1}{\mu})$, *then this solution is necessarily the sparsest possible.*

This theorem can be regarded as an inference of Theorem 2.1.1 by considering the relation between spark(D) and μ(D),

$$\mathrm{spark}(\mathrm{D}) \geq 1 + \frac{1}{\mu(\mathrm{D})}.$$

Via Theorem 5.1.1, the uniqueness of the solution to the problem (4.7) is ensured when the sparsity obeys that $k < \frac{1}{2}(1+\frac{1}{\mu})$. Similarly, the uniqueness of the solution can also be guaranteed by the cumulative coherence of D by considering the relationship

$$\mathrm{spark}(\mathrm{D}) \leq \min_{1\leq g\leq m}\{g \mid \mu_1(g-1) \geq 1\}.$$

The importance of the coherence measure to characterize dictionaries has been demonstrated in several works [Hon15b, Hon15a, Tro04]. For example, it is proven in [Tro04] that orthogonal matching pursuit and basis pursuit can correctly recover the signal under the condition

$$\max_{i\notin\mathbb{J}} \|\mathrm{D}_{\mathbb{J}}^{\mathrm{T}}\mathbf{d}_i\| < 1.$$

Or if the signal is not strictly sparse, it can be approximated with a $k-$sparse vector under the condition that $k \leq \frac{1}{3\mu}$ or $\mu_1(k) \leq \frac{1}{3}$. Furthermore, Gribonval et al give the cumulative coherence condition to make sure the signal with some noise can be exactly recovered (see Proposition 3 in [GJB15]). Although these conditions may not be the same for whichever the sparse coding methods are, the importance of incoherent dictionary learning is undoubted.

More often, the low coherence between atoms or sub-blocks of dictionary makes the dictionary gain the capability of discrimination in classification [WJY$^+$16, TLZ$^+$19, LLF14]. In fact, some other discriminative dictionary learning via sparse model can be regarded as implicitly learning a dictionary with low coherence

[ZL10, MLB+08, MPS+09]. In this sense, incoherent dictionary learning attracts lots of attentions. Great achievement on learning incoherent dictionary is gained and there is still much space for further developments.

5.2 Methods of learning incoherent dictionary

The classical dictionary learning problem, with given matrix $\mathrm{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_\ell] \in \mathbb{R}^{n\times\ell}$ of ℓ signals of dimension n, aims at finding a sparse representation of the coefficient matrix A for a decomposition of the form X = DA, where the matrix $\mathrm{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_\ell] \in \mathbb{R}^{m\times\ell}$ containing the decomposition coefficients is sparse, and the matrix $\mathrm{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_m] \in \mathbb{R}^{n\times m}$ is the dictionary with each column called atom. The optimization problem is written as

$$\begin{aligned} \min_{\mathrm{D}\in\mathbb{C}, \boldsymbol{\alpha}_i \in \mathbb{R}^p} \quad & \frac{1}{\ell}\sum_{i=1}^{\ell} \tfrac{1}{2}\|\mathbf{x}_i - \mathrm{D}\boldsymbol{\alpha}_i\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{\alpha}_i\|_0 \le k, \quad i = 1, \ldots, \ell, \end{aligned} \tag{5.4}$$

where the dictionary D is restricted in the constraint

$$\mathbb{C} = \left\{ \mathrm{D} \in \mathbb{R}^{n\times m} \mid \mathbf{d}_j^{\mathrm{T}}\mathbf{d}_j = 1, \, \forall j = 1, \ldots, p \right\},$$

in order to prevent the ℓ_2-norm of dictionary's atoms from being arbitrarily large, which leads to arbitrarily small decomposition coefficients in X. However, by this problem formulation, the resulting dictionary does not guarantee excellent performance, because its atoms can be arbitrarily correlated. Thus, to learn dictionary with low coherence is of great interest.

For learning incoherent dictionary, two main strategies have been considered: 1) Adding a decorrelation step following the phase of dictionary updating at each iteration. This strategy is applied in dictionary learning algorithms such as INK-SVD and the incoherent dictionary learning algorithm via iteratively projection and rotation (IPR) [MBP12, DM13]. 2) The second strategy learns an incoherent dictionary by introducing the regularization term $\|\mathrm{D}^{\mathrm{T}}\mathrm{D} - \mathrm{I}_m\|_{\mathrm{F}}^2$, where I_m is the identity matrix of size $m \times m$ [RLS09, BQJ14]. The normalization is realized by adding the regularization term $\sum_{i=1}^{p}(\|\mathbf{d}_i\|^2 - 1)^2$ [RLS09] or by a normalization step following the dictionary updating [BQJ14]. In this chapter, we investigate a new strategy that models the problem by proposing the explicit constraints on the coherence of the dictionary. By this method, the problem of incoherent dictionary learning becomes a constrained optimization problem with quadratic objective function and quadratic constraints.

5.2.1 Dictionary learning with additive decorrelation

The conventional dictionary learning problem (5.4) is solved, via iteratively processing two alternating steps, sparse coding and dictionary updating as described in Chapter 3. A first approach to decorrelate was proposed in the implementation[1] of the conventional K-SVD (Algorithm 7), where the authors removed the atoms that are highly correlated with each other. However, the resulting algorithm does not ensure an optimal reconstruction performance.

For overcoming the difficulties, the leading method INK-SVD [MBP12] and incoherent dictionary learning algorithm via iterative projections and rotations [DM13] are developed from the well-known K-SVD algorithm. The two algorithms take the same strategy that adds a supplementary step following the phase of dictionary updating at each iteration. These two algorithms are described in the following.

The INK-SVD is defined as follows. After obtaining the updated dictionary D^t at the t^{th} iteration, Mailhé et al proposed in [MBP12] to find the closest dictionary to D^t with unit-norm atoms and a coherence below a predefined threshold μ_c. The problem is formulated as

$$\widetilde{\mathrm{D}^t} = \underset{\mathrm{D}\in\mathbb{D}_\mu}{\arg\min} \|\mathrm{D} - \mathrm{D}^t\|_\mathrm{F}^2,$$

where $\mathbb{D}_\mu = \{\mathrm{D} \in \mathbb{R}^{n\times m} \mid \mu(\mathrm{D}) \leq \mu_c, \|\mathbf{d}_i\|_2 = 1, \forall i = 1,\dots,m\}$. To find the optimal solution, an iterative algorithm is proposed by identifying the sub-dictionary (in the same spirit as K-SVD) and decorrelating pairs of atoms with a greedy algorithm. More precisely, at each iteration, a pair of atoms $(\mathbf{d}_i, \mathbf{d}_j)$ with inner product larger than μ_c is selected and rotated symmetrically with respect to their mean vector until the angle between them reaches θ, with $\cos(\theta) = \mu_c$. The dictionary $\widetilde{\mathrm{D}^t}$ can be reached within a finite number of iterations. The algorithm of decorrelation is summarized in Algorithm 11.

The iterative projections and rotation (IPR), proposed by Barchiesi et al in [DM13], optimizes the INK-SVD algorithm by considering simultaneously the minimization of the residual error of sparse approximation when learning the dictionary

[1] http://www.cs.technion.ac.il/ ronrubin/software.html

Algorithm 11 Algorithm of decorrelation in INK-SVD [MBP12]

```
Input: Initial dictionary D and required coherence $\mu_c$.
Output: Approximate dictionary $\widetilde{\mathrm{D}}$ with maximal inner product of two atoms equal to $\mu_c$.
function DECORRELATION
    $t = 1$
    while $\max_{i \neq j} |\langle \mathbf{d}_i, \mathbf{d}_j \rangle| > \epsilon$ do
        Selection of a pair of atoms $(\mathbf{d}_i, \mathbf{d}_j) = \arg\max |\mathrm{D}^{\mathrm{T}}\mathrm{D} - \mathrm{I}_m|$;
        Decorrelation of the two atoms by rotating them until angle between them reaches $\theta$ with $\cos(\theta) = \mu_c$.
        Increment $i = i + 1$
    end while
end function
```

with a fixed target coherence level, that is,

$$\arg\min_{\mathrm{D} \in \mathbb{D}} \|\mathrm{X} - \mathrm{DA}\|_{\mathrm{F}}^2 \quad \text{subject to} \quad \mu(\mathrm{D}) \leq \mu_c.$$

The resolution of this problem consists of two steps:

- Decorrelation of atoms by projection method.
  In this step, the Gram matrix $\mathrm{G} = \mathrm{D}^{\mathrm{T}}\mathrm{D}$ is considered, rather than processed directly with D. For obtaining the dictionary D with the required low coherence, it is necessary to project G in the space defined by two constraints, named in [DM13] the structural constraint

$$\mathbb{G}_{\mu_c} = \{\mathrm{G} \in \mathbb{R}^{m \times m} \mid \mathrm{G}^{\mathrm{T}} = \mathrm{G},\ \mathrm{diag}(\mathrm{G}) = \mathbf{1}_m,\ \max_{i \neq j} g_{ij} \leq \mu_c\}, \tag{5.5}$$

  and the spectral constraint

$$\mathbb{G}_{\lambda} = \{\mathrm{G} \in \mathbb{R}^{m \times m} \mid \mathrm{G}^{\mathrm{T}} = \mathrm{G},\ \mathrm{eig}(\mathrm{G}) \geq 0,\ \mathrm{rank}(\mathrm{G}) \leq n\}. \tag{5.6}$$

  For a given G, the projection on $\mathbb{G}_{\mu_c}$ can be easily gained by the operations:

$$g_{ij} = \begin{cases} 1 & \text{if } i = j \\ \mathrm{sign}(g_{ij}) \min(|g_{ij}|, \mu_c) & \text{if } i \neq j. \end{cases}$$

  For getting the projection on $\mathbb{G}_{\lambda}$, it computes the eigenvalues $\lambda_i,\ i = 1, \ldots, m$, by eigendecomposition of $\mathrm{G} = \mathrm{V}\Delta\mathrm{V}^{\mathrm{T}}$ where D is a diagonal matrix formed by the eigvenvalues of G and V formed by the corresponding eigenvectors, and then, retains the largest n ones with ensuring their positivity and setting to zero the remaining ones. However, after projection on $\mathbb{G}_\lambda$, the incoherence of D cannot be guaranteed any more. Hence, the projection on the intersection of the two constraint spaces is not reached by only one projection, but by an iterative process.

When the Gram matrix G belongs to this intersection, namely satisfying $\mathrm{G} \in \mathbb{G}_{\mu_c} \cap \mathbb{G}_\lambda$, then D can be recovered by

$$\mathrm{D} = \Delta^{\frac{1}{2}} \mathrm{V}^{\mathrm{T}}. \tag{5.7}$$

- Dictionary rotation for minimizing the reconstruction error while keeping the dictionary with the required coherence level.
  The dictionary rotation is realized by introducing an orthogonal transform matrix W, by which the coherence quality is still held due to the fact that $(\mathrm{WD})^{\mathrm{T}}(\mathrm{WD}) = \mathrm{G}$. Consequently, the reconstruction error minimizing problem is turned into an optimization problem with respect to W, that is,

  $$\widetilde{\mathrm{W}} = \arg\min_{\mathrm{W} \in \mathbb{N}} \|\mathrm{X} - \mathrm{WDA}\|_{\mathrm{F}}^2,$$

  where $\mathbb{N}$ denotes the set of orthogonal matrices of size $m \times m$. This problem has a closed-form solution

  $$\widetilde{\mathrm{W}} = \mathrm{SU}^{\mathrm{T}},$$

  where U and S are the left and right unitary matrices in the singular value decomposition (SVD) of $\mathrm{DAX}^{\mathrm{T}}$, namely $\mathrm{DAX}^{\mathrm{T}} = \mathrm{U}\Sigma\mathrm{S}^{\mathrm{T}}$.

5.2.2 Regularized incoherent dictionary learning

Methods from the second strategy seek to learn an incoherent dictionary by minimizing a regularized objective function, where the regularization term constrains the coherence. We describe in the following the most-known methods. The problem is formulated in [RLS09] as

$$\arg\min_{\mathrm{D} \in \mathbb{R}^{n \times m}, \mathrm{A} \in \mathbb{R}^{m \times l}} \|\mathrm{X} - \mathrm{DA}\|_{\mathrm{F}}^2 + \lambda_1 \sum_j \sum_i \log(|\alpha_{ij}| + \beta) + \lambda_2 \|\mathrm{D}^{\mathrm{T}}\mathrm{D} - \mathrm{I}_m\|_{\mathrm{F}}^2 + \lambda_3 \sum_k (\mathbf{d}_k^2 - 1)^2, \tag{5.8}$$

where the second term denotes the sparsity-promoting. This formulation is obtained by using a universal models [RLS09]. The β is the parameter regulating the universal model. The coefficients λ_1, λ_2 and λ_3 are trade-off parameters. However, here we focus on the step of dictionary updating. The optimization problem with respect to D is of the form

$$\operatorname*{arg\,min}_{\mathrm{D}\in\mathbb{R}^{n\times m}} \|\mathrm{X}-\mathrm{DA}\|_{\mathrm{F}}^2 + \lambda_2 \|\mathrm{D}^{\mathrm{T}}\mathrm{D}-\mathrm{I}_m\|_{\mathrm{F}}^2 + \lambda_3 \sum_k (\mathbf{d}_k^2 - 1)^2,$$

where the second term measures the correlation of two distinct atoms in D, which are the off-diagonal elements of $\mathrm{D}^{\mathrm{T}}\mathrm{D}$, and the last term makes sure that columns of D are of norm close to 1. This optimization problem can be solved with the method of optimal coherence-constraint directions (MOCOD) [RLS09], inspired from the method of optimal direction (MOD), which updates D by

$$\mathrm{D}^{t+1} = \big(\mathrm{X}(\mathrm{A}^{t+1})^{\mathrm{T}} + 2(\lambda_2+\lambda_3)\mathrm{D}^t\big)\Big(\mathrm{A}^{t+1}(\mathrm{A}^{t+1})^{\mathrm{T}} + 2\lambda_2(\mathrm{D}^t)^{\mathrm{T}}\mathrm{D}^t + 2\lambda_3\mathrm{diag}((\mathrm{D}^t)^{\mathrm{T}}\mathrm{D}^t)\Big)^{-1}.$$

The MOCOD method is proved to outperform the MOD method in image reconstruction.

In [BQJ14], the incoherent dictionary learning problem is formulated by introducing only the coherence regularization, namely the Frobenius norm of the difference of Gram matrix and identity,

$$\operatorname*{arg\,min}_{\mathrm{D}\in\mathbb{D},\mathrm{A}\in\mathbb{R}^{m\times l}} \tfrac{1}{2}\|\mathrm{X}-\mathrm{DA}\|_{\mathrm{F}}^2 + \lambda_1\|\mathrm{A}\|_0 + \tfrac{\lambda_2}{2}\|\mathrm{D}^{\mathrm{T}}\mathrm{D}-\mathrm{I}_m\|_{\mathrm{F}}^2.$$

The incoherent dictionary is learned via a hybrid alternating proximal method, of which the dictionary is updated atom-by-atom using the operation

$$\mathbf{d}_j^{t+1} = \mathrm{prox}_{\lambda_j^t}^{\delta_{\mathbb{D}}}\big(\mathbf{d}^t - \tfrac{1}{\lambda_j^t}\nabla_{\mathbf{d}_j}\mathrm{Q}(\mathrm{A}^t,\mathrm{D}_j^t)\big),$$

where $\mathrm{Q}(\mathrm{A},\mathrm{D}) = \frac{1}{2}\|\mathrm{X}-\mathrm{DA}\|_{\mathrm{F}}^2 + \frac{\lambda_2}{2}\|\mathrm{D}^{\mathrm{T}}\mathrm{D}-\mathrm{I}_m\|_{\mathrm{F}}^2$ and $\mathrm{D}_j^t = [\mathbf{d}_1^{t+1},\ldots,\mathbf{d}_{j-1}^{t+1},\mathbf{d}_j^t,\ldots,\mathbf{d}_m^t]$, and then the dictionary is normalized after the dictionary updating at each iteration.

Similarly, Abolghasemi et al [AFS15] tackled the problem with the same coherence regularization as in [BQJ14]. However, they proposed an incoherent dictionary learning algorithm with dictionary updating by a gradient descent method. In addition, the coherence regularization was also measured by the sum of ℓ_1-norm of every two different atoms [LHDL16, LDHL17]. Moreover, for some tasks such as classification, the Fisher criterion [LLF14] can be regarded as a coherence regularization, which makes the sub-dictionary of different class coherent. Incoherent dictionary learning algorithms of the second strategy achieve good performance in data reconstruction [AFS15, LDHL17], classification [BQJ14, LLZ12, LHT$^+$18] and object recognition [LLF14, ZJD13]. However, they suffer from a major issue: it is not possible to constraint exactly the coherence level to a fixed value, because the relation between it and the regularization trade-off parameter is unknown.

5.2.3 Towards incoherence-constrained dictionary learning

The third strategy considers the simplest way to formulate the problem, by adding the constraints of coherence and unit norm of the dictionary elements into the generic dictionary learning problem. The coherence of the dictionary is constrained with the inequality

$$|\mathbf{d}_p^{\mathrm{T}}\mathbf{d}_q| \leq \mu_c, \quad \forall p, q \in \{1, 2, \ldots, m\},\ p \neq q, \tag{5.9}$$

where μ_c is the predefined coherence level. The unit norm of the dictionary's atoms is modeled by the equality

$$\mathbf{d}_p^{\mathrm{T}}\mathbf{d}_p = 1 \quad \forall p = 1, 2, \ldots, m. \tag{5.10}$$

Thus, the problem of incoherent dictionary learning can be resumed as a constrained optimization problem with quadratic objective function and quadratic constraints

$$\begin{aligned}
&\min_{\mathrm{D}\in\mathbb{R}^{n\times m}, \boldsymbol{\alpha}_i\in\mathbb{R}^m} \frac{1}{l}\sum_{i=1}^{l} \tfrac{1}{2}\|\mathbf{x}_i - \mathrm{D}\boldsymbol{\alpha}_i\|_2^2 \\
&\text{subject to} \begin{cases} |\mathbf{d}_p^{\mathrm{T}}\mathbf{d}_q| \leq \mu_c,\ \forall p, q \in \{1, 2, \ldots, m\}, q \neq p \\ \mathbf{d}_p^{\mathrm{T}}\mathbf{d}_p = 1, \quad p = 1, \ldots, m \\ \|\mathbf{x}_i\|_0 \leq k, \quad i = 1, \ldots, l. \end{cases}
\end{aligned} \tag{5.11}$$

The problem of estimating simultaneously A and D is non-convex and non-smooth because of the sparsity-prompting ℓ_0-norm and the constraints. To the best of our knowledge, there is no work on incoherent dictionary learning by solving the problem with explicit constraints on dictionary coherence and its unit norm. To solve this constrained optimization problem, we take advantage of recent developments in optimization problem with orthogonality constraints, with the augmented Lagrangian method and the alternating proximal minimization method.

The optimization problem with orthogonality constraints has been recently addressed in physics [UWY$^+$15], mathematics [LCWM09] and information science [CJY16]. The Lagrangian multiplier method [Ber99] is frequently used to deal with such a problem [LCWM09, UWY$^+$15]. However, it is not always easy to solve the Lagrangian function by satisfying the first order optimal condition. In [UWY$^+$15], the Kohn-Sham problem was reformulated by the Lagrangian multiplier method, and the proximal gradient method was then proposed to solve the Lagrange function. Moreover, it was proven that the algorithm has good convergence property. The orthogonality constrained optimization problem were also solved via the augmented Lagrangian method [CJY16, ZZCL17]. Compared with the Lagrangian method, the penalty method shows more stability [Ber99]. However, the reformulated problem can be non-convex and non-smooth, which makes the problem hard to tackle. In [CJY16], the alternating proximal method was combined with the augmented Lagrangian method and the existence of the sub-sequence to a KKT point was proven. The new proposed algorithm was then applied in compressed mode for variational problems in physics, illustrating the effectiveness and efficiency of the algorithm. In [ZZCL17], an extended proximal alternating linearized minimization method was introduced to solve the Lagrangian function, and its convergence was proven based on the theory of the Kurdyka-Łojasiewicz inequality property [ABRS10].

5.3 Exact ℓ_0 based incoherent dictionary learning

This section gives a method to solving the exact ℓ_0 based incoherent dictionary learning problem (5.11). Similar to all other dictionary learning algorithms presented in Section 3.4, we learn the incoherent dictionary via iteratively processing two alternate steps: sparse coding and dictionary learning. The sparse coding problem of (5.11) is the ℓ_0-norm constrained problem, which is exactly the same as in the generic sparse coding (4.7). Therefore, we use the AcMIQP algorithm proposed in Section 4.1.2 to solve exactly the ℓ_0 constrained sparse coding problem. With a fixed sparse code, the problem with respect to the dictionary D becomes a non-convex constrained optimization problem. For solving this problem, the augmented Lagrange method and proximal alternating linearized minimization method are used. The convergence of the proposed algorithm is analyzed in the following.

5.3.1 Augmented Lagrangian method

The augmented Lagrangian method is used in this part to reformulate the incoherence-constrained dictionary learning problem. The optimization problem with respect to D is:

$$\begin{aligned} \min_{\mathrm{D}\in\mathbb{R}^{n\times m}} \quad & \frac{1}{l}\sum_{i=1}^{l}\left(\tfrac{1}{2}\|\mathbf{x}_i-\mathrm{D}\boldsymbol{\alpha}_i\|_2^2\right) \\ \text{subject to} \quad & \begin{cases} |\mathbf{d}_p^{\mathrm{T}}\mathbf{d}_q|\le\mu_c,\ \forall p,q\in\{1,2,\dots,m\},\ p\neq q \\ \mathbf{d}_p^{\mathrm{T}}\mathbf{d}_p=1, \quad p=1,\dots,m. \end{cases} \end{aligned} \tag{5.12}$$

By introducing a new variable $\mathrm{G}\in\mathbb{R}^{m\times m}$ that satisfies the identity

$$\mathrm{G}=\mathrm{D}^{\mathrm{T}}\mathrm{D},$$

the problem can be written in the form[2]

$$\begin{aligned} \min_{\mathrm{D}\in\mathbb{R}^{n\times m},\mathrm{G}\in\mathbb{R}^{m\times m}} \quad & \tfrac{1}{2}\|\mathrm{X}-\mathrm{DA}\|_{\mathrm{F}}^2 \\ \text{subject to} \quad & \begin{cases} \mathrm{G}=\mathrm{D}^{\mathrm{T}}\mathrm{D}, \quad \mathrm{G}\in\mathbb{S}_{\mathrm{G}} \\ \mathbf{d}_p^{\mathrm{T}}\mathbf{d}_p=1, \quad p=1,\dots,m, \end{cases} \end{aligned} \tag{5.13}$$

where

$$\mathbb{S}_{\mathrm{G}}=\left\{\mathrm{G}\in\mathbb{R}^{m\times m}\ \middle|\ |g_{ij}|\le\mu_c, i,j=\{1,2,\dots,m\}, i\neq j\right\}.$$

Let $\delta_{\mathbb{S}_{\mathrm{G}}}(\mathrm{G})$ be the indicator function on this set, namely

$$\delta_{\mathbb{S}_{\mathrm{G}}}(\mathrm{G})=\begin{cases} 0, & \text{if } \mathrm{G}\in\mathbb{S}_{\mathrm{G}} \\ +\infty, & \text{otherwise.} \end{cases} \tag{5.14}$$

[2] Another formulation can be proposed, by putting the unit-norm constraint in the space definition [WGY10, YY13]. However, this formulation suffers from low convergence. For this reason, we consider the optimization problem (5.13) where the explicit unit-norm constraint will be addressed using the augmented Lagrangian method.

The constrained optimization problem can be solved by considering the augmented Lagrangian function:

$$\mathrm{L}_{(c1,c2)}(\mathrm{D},\mathrm{G},\boldsymbol{\lambda},\mathrm{H}) = \frac{1}{2}\|\mathrm{X}-\mathrm{DA}\|_\mathrm{F}^2 + \sum_{p=1}^{m}\lambda_p(\mathbf{d}_p^\mathrm{T}\mathbf{d}_p - 1) + \frac{c_1}{2}\sum_{p=1}^{m}(\mathbf{d}_p^\mathrm{T}\mathbf{d}_p - 1)^2 \\ + \mathrm{tr}(\mathrm{H}(\mathrm{G}-\mathrm{D}^\mathrm{T}\mathrm{D})) + \frac{c_2}{2}\|\mathrm{G}-\mathrm{D}^\mathrm{T}\mathrm{D}\|_\mathrm{F}^2 + \delta_{\mathbb{S}_\mathrm{G}}(\mathrm{G}), \quad (5.15)$$

where $\boldsymbol{\lambda} = [\lambda_1,\ldots,\lambda_m]$ and H are respectively the vector and matrix associated to the equality constraints on the diagonal of $\mathrm{D}^\mathrm{T}\mathrm{D}$ and on G, c_1 and c_2 are the positive penalty parameters (the augmentation). When these parameters grow into infinity, the optimal solution of the original problem (5.13) can be reached.

Therefore, the optimization problem becomes:

$$\min_{\mathrm{D}\in\mathbb{R}^{n\times m},\mathrm{G}\in\mathbb{R}^{m\times m}} \mathrm{L}_{(c1,c2)}(\mathrm{D},\mathrm{G},\boldsymbol{\lambda},\mathrm{H}). \quad (5.16)$$

It is not the standard augmented Lagrangian method (where the objective function is convex and has only one term, in most case, the constraints are closed convex set). While our problem is non-convex and non-smooth, it is still reasonable to consider the inexact ADMM framework [Ber99]. The resulting algorithm is illustrated in Algorithm 12.

As presented in Algorithm 12, the inexact augmented Lagrangian method operates in three alternating steps: In Step 1, the primal variables are computed by solving, as explained in next section, the optimization problem,

$$(\mathrm{D}^i,\mathrm{G}^i) = \arg\min_{\mathrm{D},\mathrm{G}} \mathrm{L}_{(c_1^i,c_2^i)}(\mathrm{D},\mathrm{G},\boldsymbol{\lambda}^i,\mathrm{H}^i), \quad (5.17)$$

where D^i, G^i $\boldsymbol{\lambda}^i$ and H^i are the values in the i-th iteration; In Step 2, the Lagrangian multipliers $\boldsymbol{\lambda}$ and H are updated; And in Step 3, the penalty parameters c_1 and c_2 are increased. It is proven that the two parameters c_1 and c_2 can stay much smaller to solve such optimization problem [Ber99].

5.3.2 Proximal alternating linearized minimization

The problem (5.17) is a non-convex and non-smooth optimization problem. It is unsolvable by satisfying the Karush-Kuhn-Tucker (KKT) conditions. We propose to use an alternating strategy to address this optimization problem. The optimal ma-

Algorithm 12 The inexact ADMM framework for solving (5.16)

```
Input: The training data X and coefficient matrix A, the stop criteria ($\epsilon$, $N_{iter}$).
Output: The optimal solution $D^*$
  function DICTIONARYUPDATING
    Initialization the parameters ($\boldsymbol{\lambda}^0$, $D^0$, $c_1^0$, $H^0$, $G^0$, $c_2^0$, $\rho_1$, $\rho_2$)
    for all $i = 0$ to $N_{iter} - 1$ do
      1. Computing the optimal solution $(D^i, G^i)$:
           $(D^i, G^i) = \arg\min L_{(c_1^i, c_2^i)}(D, G, \boldsymbol{\lambda}^i, H^i).$
      2. Updating the Lagrangian multiplier $(\boldsymbol{\lambda}^i, H^i)$:
           $\begin{cases} \boldsymbol{\lambda}^{i+1} = \boldsymbol{\lambda}^i + c_1^i(\mathrm{diag}((D^i)^T D^i) - \mathbf{1}); \\ H^{i+1} = H^i + c_2^i(G^i - (D^i)^T D^i). \end{cases}$
      3. Updating the penalty parameters $(c_1^i, c_2^i)$:
           $\begin{cases} c_1^{i+1} = \rho_1 c_1^i; \\ c_2^{i+1} = \rho_2 c_2^i. \end{cases}$
      if $\max|(\mathbf{d}_p^i)^T \mathbf{d}_p^i - 1| \leq \epsilon$ and $\max|G^i - (D^i)^T D^i| \leq \epsilon$ then
        return,
      end if
    end for
  end function
```

trices of D and G are obtained by alternating the gradient descent method and the proximal method, which can be regarded as the special case of the extended proximal alternating linearized minimization (EPALM) [ZZCL17].

To investigate the EPALM method, we rewrite the objective function in problem (5.17) in the form of three additive parts:

$$L_{(c_1^i, c_2^i)}(D, G, \boldsymbol{\lambda}^i, H^i) = f(D) + h(D, G) + g(G), \tag{5.18}$$

with the definition of:

$$\begin{cases} f(D) &= \frac{1}{2}\|X - DA\|_F^2 + \sum_{p=1}^{m} \lambda_p^i(\text{å}_p^T \mathbf{d}_p - 1) + \frac{c_1^i}{2}\sum_{p=1}^{m}(\mathbf{d}_p^T \mathbf{d}_p - 1)^2 \\ h(D, G) &= \mathrm{tr}\big(H^i(G - D^T D)\big) + \frac{c_2^i}{2}\|G - D^T D\|_F^2 \\ g(G) &= \delta_{\mathbb{S}_G}(G). \end{cases}$$

The problem is well defined with $f : \mathbb{R}^{n\times m} \to (-\infty,+\infty]$, $h : \mathbb{R}^{n\times m} \times \mathbb{R}^{m\times m} \to (-\infty,+\infty]$ being a $\mathbb{C}^1$ function (i.e., continuously differentiable), and $\inf f(\mathrm{D}) > -\infty$, $\inf h(\mathrm{D},\mathrm{G}) > -\infty$, $g : \mathbb{R}^{p\times p} \to [0,\infty]$ a proper and lower semicontinuous function. In addition, for guaranteeing the convergence of the EPALM method, the following assumptions should be satisfied:

(i) The functions $\mathrm{D} \to h(\mathrm{D},\mathrm{G})$ and $\mathrm{G} \to h(\mathrm{D},\mathrm{G})$ have their gradients globally Lipschitz continuous with moduli $\mathrm{L_D}$ and $\mathrm{L_G}$, respectively. In other words, the partial gradients of h with respect to D and G verify the property:

$$\begin{cases} \|\nabla_{\mathrm{D}} h(\mathrm{D},\mathrm{G}) - \nabla_{\mathrm{D}} h(\bar{\mathrm{D}},\mathrm{G})\|_{\mathrm{F}} \le \mathrm{L_D}\|\mathrm{D}-\bar{\mathrm{D}}\|_{\mathrm{F}} \\ \|\nabla_{\mathrm{G}} h(\mathrm{D},\mathrm{G}) - \nabla_{\mathrm{G}} h(\mathrm{D},\bar{\mathrm{G}})\|_{\mathrm{F}} \le \mathrm{L_G}\,\|\mathrm{G}-\bar{\mathrm{G}}\|_{\mathrm{F}}, \end{cases} \tag{5.19}$$

for all $(\mathrm{D},\bar{\mathrm{D}})$ and $(\mathrm{G},\bar{\mathrm{G}})$.

(ii) $\mathrm{L}_{(c_1^i,c_2^i)}(\mathrm{D},\mathrm{G},\boldsymbol{\lambda}^i,\mathrm{H}^i)$ satisfies the Kurdyka-Łojasiewicz inequality [ZZCL17]. The proof is given in Section 5.3.5.

The problem (5.17) can now be solved by alternating the optimization problems with respect to D and G, respectively:

$$\begin{cases} \mathrm{D}^{i,j} = \underset{\mathrm{D}\in\mathbb{R}^{n\times m}}{\arg\min}\, f(\mathrm{D}) + h(\mathrm{D}^{i,j-1},\mathrm{G}^{i,j-1}) \\ \qquad\qquad +\mathrm{tr}\big((\mathrm{D}-\mathrm{D}^{i,j-1})^{\mathrm{T}}\nabla_{\mathrm{D}} h(\mathrm{D}^{i,j-1},\mathrm{G}^{i,j-1})\big) \\ \qquad\qquad +\frac{\tilde{t}_1}{2}\|\mathrm{D}-\mathrm{D}^{i,j-1}\|_{\mathrm{F}}^2 \\ \mathrm{G}^{i,j} = \underset{\mathrm{G}\in\mathbb{R}^{m\times m}}{\arg\min}\, g(\mathrm{G}) + h(\mathrm{D}^{i,j},\mathrm{G}^{i,j-1}) \\ \qquad\qquad +\langle \mathrm{G}-\mathrm{G}^{i,j-1},\nabla_{\mathrm{G}} h(\mathrm{D}^{i,j},\mathrm{G}^{i,j-1})\rangle \\ \qquad\qquad +\frac{t_2}{2}\|\mathrm{G}-\mathrm{G}^{i,j-1}\|_{\mathrm{F}}^2, \end{cases} \tag{5.20}$$

where $\langle \mathrm{M}_1,\mathrm{M}_2\rangle = \mathrm{tr}(\mathrm{M}_1^{\mathrm{T}}\mathrm{M}_2)$ is defined as the scalar product in the matrix space $\mathbb{M}_n(\mathbb{R})$, $\tilde{t}_1$ and t_2 are the coefficients associated respectively to the second order approximation terms. By considering that the function f is differentiable, the problem of estimating D can be expressed as:

$$
\begin{aligned}
\mathrm{D}^{i,j} = \underset{\mathrm{D}\in\mathbb{R}^{n\times m}}{\arg\min}\, & f(\mathrm{D}^{i,j-1}) + h(\mathrm{D}^{i,j-1},\mathrm{G}^{i,j-1}) \\
& + \operatorname{tr}\big((\mathrm{D}-\mathrm{D}^{i,j-1})^{\mathrm{T}}\nabla_{\mathrm{D}}(f(\mathrm{D}^{i,j-1}) + h(\mathrm{D}^{i,j-1},\mathrm{G}^{i,j-1}))\big) \\
& + \frac{t_1}{2}\|\mathrm{D}-\mathrm{D}^{i,j-1}\|_{\mathrm{F}}^2.
\end{aligned}
\tag{5.21}
$$

To solve this optimization problem, it is easy to update D by the method of gradient descent. As for the problem of estimating G in (5.20), the proximal method is applicable. By combining both steps, the solution of D and G can be achieved by the following process:

$$
\begin{cases}
\mathrm{D}^{i,j} = \mathrm{D}^{i,j-1} - \frac{1}{t_1}\nabla_{\mathrm{D}}\big(f(\mathrm{D}^{i,j-1}) + h(\mathrm{D}^{i,j-1},\mathrm{G}^{i,j-1})\big) \\
\mathrm{G}^{i,j} = \operatorname{prox}_{\frac{1}{t_2}g}\big(\mathrm{G}^{i,j-1} - \frac{1}{t_2}\nabla_{\mathrm{G}}(h(\mathrm{D}^{i,j},\mathrm{G}^{i,j-1}))\big),
\end{cases}
\tag{5.22}
$$

where the notation prox_{uf} denotes the proximal operator, defined in Section 4.1.2, of the scaled function uf (also called the proximal operator of g with parameter u), namely

$$
\operatorname{prox}_{\frac{1}{t_2}f}(\mathbf{v}) = \underset{\mathbf{x}}{\arg\min}\, f(\mathbf{x}) + \tfrac{1}{2t_2}\|\mathbf{x}-\mathbf{v}\|^2.
\tag{5.23}
$$

The partial derivatives of the three parts of (5.18) are:

$$
\begin{cases}
\nabla_{\mathrm{D}} f(\mathrm{D}^{i,j-1}) & = -(\mathrm{Y}-\mathrm{D}^{i,j-1}\mathrm{X})\mathrm{X}^{\mathrm{T}} + 2\mathrm{D}^{i,j-1}\operatorname{diag}(\boldsymbol{\lambda^i}) + 2c_1^i\mathrm{D}^{i,j-1}\operatorname{diag}(\mathbf{v^{j-1}}) \\
\nabla_{\mathrm{D}} h(\mathrm{D}^{i,j-1},\mathrm{G}^{i,j-1}) & = -\mathrm{D}^{i,j-1}(\mathrm{H}^i + (\mathrm{H}^i)^{\mathrm{T}}) - c_2^i\mathrm{D}^{i,j-1}\big(\mathrm{G}^{i,j-1} + (\mathrm{G}^{i,j-1})^{\mathrm{T}} - 2(\mathrm{D}^{i,j-1})^{\mathrm{T}}\mathrm{D}^{i,j-1}\big) \\
\nabla_{\mathrm{G}} h(\mathrm{D}^{i,j},\mathrm{G}^{i,j-1}) & = \mathrm{H}^i + c_2^i(\mathrm{G}^{i,j-1} - (\mathrm{D}^{i,j})^{\mathrm{T}}\mathrm{D}^{i,j}),
\end{cases}
\tag{5.24}
$$

where $\mathbf{v^{j-1}} = \operatorname{diag}\big((\mathrm{D}^{i,j-1})^{\mathrm{T}}\mathrm{D}^{i,j-1} - \mathrm{I}_m\big)$ denotes the vector with the entries valued by the diagonal of the matrix $(\mathrm{D}^{i,j-1})^{\mathrm{T}}\mathrm{D}^{i,j-1} - \mathrm{I}_p$. The two expressions $\operatorname{diag}(\boldsymbol{\lambda^i})$ and $\operatorname{diag}(\mathbf{v^{j-1}})$ return the matrix with the diagonal filled by the elements in the vectors $\boldsymbol{\lambda^i}$ and $\mathbf{v^{j-1}}$, respectively. Moreover, $\mathrm{G}^{i,j}$ can be computed as:

$$
\mathrm{G}^{i,j}(i_x,i_y) =
\begin{cases}
\widetilde{\mathrm{G}}(i_x,i_y) & \text{if } |\widetilde{\mathrm{G}}(i_x,i_y)| \le \mu_c \\
\operatorname{sign}(\widetilde{\mathrm{G}}(i_x,i_y))\mu_c & \text{otherwise,}
\end{cases}
\tag{5.25}
$$

where $\widetilde{\mathrm{G}} = \mathrm{G}^{i,j-1} - \frac{1}{t_2}\nabla_{\mathrm{G}}\big(h(\mathrm{D}^{i,j},\mathrm{G}^{i,j-1})\big)$, $\operatorname{sign}(\widetilde{\mathrm{G}}(i_x,i_y))$ is the sign of the $\widetilde{\mathrm{G}}(i_x,i_y)$, i.e.,

$+1$ if $\widetilde{\mathrm{G}}(i_x, i_y) \geq 0$ and -1 otherwise. In these expressions, i_x and i_y are respectively the row and column indices of the matrix $\widetilde{\mathrm{G}}$, with $i_x, i_y = 1, 2, \ldots, m$.

The proposed EPALM method for estimating D and G is summarized in Algorithm 13. For the completeness of the algorithm, we provide next a convergence analysis in terms of the subdifferential of the objective function in (5.17), as well as the choice of the parameters.

A necessary but not sufficient condition for $x \in \mathbb{R}^n$ to be a minimizer of f is $0 \in \partial f(x)$. Back to our optimization problem, the subdifferential of $\mathrm{L}_{(c_1^i, c_2^i)}(\mathrm{D}, \mathrm{G}, \boldsymbol{\lambda}^i, \mathrm{H}^i)$ at $(\mathrm{D}^i, \mathrm{G}^i)$, denoted by $\Theta^i = (\Theta^i_{\mathrm{D}}, \Theta^i_{\mathrm{G}})$ and expressed as

$$\Theta^i = \partial \mathrm{L}_{(c_1^i, c_2^i)}(\mathrm{D}^i, \mathrm{G}^i, \boldsymbol{\lambda}^i, \mathrm{H}^i),$$

can be computed directly and the result can be written in form of:

$$\begin{cases} \Theta^i_{\mathrm{D}} = \nabla_{\mathrm{D}} f(\mathrm{D}^i) + \nabla_{\mathrm{D}} h(\mathrm{D}^i, \mathrm{G}^i) \\ \Theta^i_{\mathrm{G}} = t_2(\mathrm{G}^{i-1} - \mathrm{G}^i). \end{cases} \tag{5.26}$$

Thus a solution of our optimization problem can be found when $\|\Theta^i\|_\infty \to 0$. According to the formulation of Θ^i in (5.26), D^i is exactly the local optimal solution of the subproblem with respect to D, and the sequence $\mathrm{G}^{i,j}$ is convergent, since $\|\mathrm{G}^i - \mathrm{G}^{i-1}\|_{\mathrm{F}} \to 0$.

Besides, it is noticed that to guarantee that every bounded sequence generated by the proposed method converges to a critical point of $\mathrm{L}_{(c_1^i, c_2^i)}(\mathrm{D}^i, \mathrm{G}^i, \boldsymbol{\lambda}^i, \mathrm{H}^i)$, the parameters c_1, c_2 and the stepsizes t_1, t_2 need to be appropriately chosen. The following can be noted:

- The initial positive penalty parameter c_1^0 and c_2^0 should be carefully chosen to avoid ill-conditioning, i.e., they should satisfy the second-order sufficient condition:

  $$\nabla^2_{\mathrm{DD}} \mathrm{L}(\mathrm{D}^i, \mathrm{G}^i, \boldsymbol{\lambda}^{\boldsymbol{i}}, \mathrm{H}^i) > 0.$$

  Due to the complexity of the derivative of a matrix function with respect to a matrix (the derivative of the function with respect to each element of the matrix being a matrix), we do not give the detail here.

- The convergence of the algorithm requires that the descent stepsize, i.e., $\frac{1}{t_1}$ and $\frac{1}{t_2}$, should not be too large, satisfying $t_1 > \mathrm{L_D}$ and $t_2 > \mathrm{L_G}$.

Algorithm 13 EPALM algorithm for solving subproblem (5.17)

```
Input: The training data X and coefficient matrix A, the parameters ($\boldsymbol{\lambda}^{i}$, $H^i$, $c_1^i$ and $c_2^i$, the step size $t_1$ and $t_2$), the stop criteria ($\epsilon^i$, $N^i_{iter}$, the subdifferential $\Theta^i$ of the $L_{(c_1^i,c_2^i)}(D^i, G^i, \boldsymbol{\lambda}^i, H^i)$)
Output: The solution $D^i$ and $G^i$
  function EPALM
      Initialization $j = 0$, $D^{i,0} = D^{i-1}$,
      $G^{i,0} = (D^{i,0})^T D^{i,0}$,
      $G^{i,0}(i_x, i_y) = \mathrm{sign}(G^{i,0}(i_x, i_y)) \min(|G^{i,0}(i_x, i_y)|, \mu_c)$,
      $\Theta^i = \Theta^i(D^{i,0}, G^{i,0})$.
      while $j < N^i_{iter}$ and $\Theta^i > \epsilon^i$ do
          1. Updating $D^{i,j}$ by computing:
             $$D^{i,j} = D^{i,j-1} - \frac{1}{t_1}\nabla_D\big(f(D^{i,j-1}) + h(D^{i,j-1}, G^{i,j-1})\big).$$
          2. Computing:
             $$\widetilde{G} = G^{i,j-1} - \frac{1}{t_2}\nabla_G\big(h(D^{i,j}, G^{i,j-1})\big).$$
          3. Projecting the $\widetilde{G}$ in the space $\mathbb{S}_{\mathbb{G}}$:
             $$G^{i,j}(i_x, i_y) = \begin{cases} \widetilde{G}(i_x, i_y) & \text{if } |\widetilde{G}(i_x, i_y)| \le \mu_c; \\ \mathrm{sign}(\widetilde{G}(i_x, i_y))\mu_c & \text{otherwise.} \end{cases}$$
          4. Calculating the subdifferential $\Theta^i(D^{i,j}, G^{i,j})$.
          5. $j = j + 1$.
      end while
  end function
```

Proposition 2. *To sum up, a sequence* $(D^{i,j}, G^{i,j})$, $j \in \mathbb{N}$ *is generated by using the proposed method, then the following conditions will be satisfied:*

- *When* $j \to \infty$, $\|\Theta^i(D^{i,j}, G^{i,j})\|_\infty \to 0$.

- *The sequence* $(D^{i,j}, G^{i,j})$ *for* $j \in \mathbb{N}$ *has a finite length, that is,*

$$\sum_{j=1}^{\infty} \|(D^{i,j+1}, G^{i,j+1}) - (D^{i,j}, G^{i,j})\|_F < \infty. \tag{5.27}$$

5.3.3 Experiments to assess the dictionary updating algorithm

To assess the performance of the dictionary updating algorithm, it is tested on synthetic data. Specifically, a sparse matrix $A \in \mathbb{R}^{20\times100}$ with the maximal column-wise sparsity level 3 is manually created. A dictionary $D \in \mathbb{R}^{5\times20}$ is generated from the IPR incoherent dictionary learning algorithm [DM13] on an arbitrary image, with the coherence parameter set to 0.6; The obtained dictionary has a coherence computed by (5.1) of 0.608. Then, in each test, the set of signals X can be generated with

$$X = DA + \omega E, \tag{5.28}$$

where the second term in the right-hand-side corresponds to the unfitness noise, where E a white Gaussian zero-mean matrix with a noise level set to $\omega = 0.1$.

To provide an overall evaluation of the proposed algorithm, several coherence parameter values $\mu_c = \{0.5, 0.55, 0.6, 0.7, 0.8, 0.9, 1\}$ are used. It is worth noting that values below 0.4 cannot be reached due to geometric constraints, namely the coherence of an overcomplete dictionary of size $n \times m$ is bounded by (5.2). The other parameters values are set as follows: For Algorithm 12, we set: ADMM algorithm, the maximal outer iteration number $N_{iter} = 50$, the coefficient to update the penalty parameter $\rho_1 = \rho_2 = 1.5$, the stopping criterion $\epsilon = 0.01$; For Algorithm 13, we set: EPALM algorithm, the maximal inner iteration number $N^i_{iter} = 1000$, the stopping criterion $\epsilon^i = \epsilon^0 = 0.01$. For each coherence value, five independent Monte Carlo simulations are conducted.

We analyze the algorithm through the accuracy $\|\widetilde{D} - D^*\|_F$ and the objective function value $\frac{1}{2}\|X - \widetilde{D}A\|_F^2$, as well as the computing time, where $\widetilde{D}$ is the output of the algorithm and D^* is the known optimal solution. Moreover, we also study the iteration numbers with different coherence parameter settings. The results are listed in Table 5.1. These results show that, with the decrease of the coherent parameter μ_c, more iterations are needed to converge, and thus more time. On the other hand, as the coherence parameter increases, the stopping criteria $\max(|\mathbf{d}_p^T\mathbf{d}_p - 1|) \leq 0.01$ and $\max|G - D^TD| \leq 0.01$ can be easily satisfied. It is observable that when $\mu_c = 0.6$, which is the closest value to the coherence of the target dictionary D^* (i.e., $\mu^* = 0.607$), the results have the greatest accuracy of 0.058. For the other values of μ_c, the results remain consistent but with a deduced accuracy. This is easy to understand since, for $\mu_c > \mu^*$, the optimal solution D^* is in the feasible region, which should also be the output of the algorithm. But, influenced by the

Table 5.1 – Accuracy results and computing time on synthetic data

Coherence parameter μ_c	0.5	0.55	0.6	0.7	0.8	0.9	1.0
Initial objective function value: $\frac{1}{2}\|X - D_0A\|_F^2$	9523	9592	9318	9643	9340	9483	9446
Final objective function value: $\frac{1}{2}\|X - \widetilde{D}A\|_F^2$	216.84	91.80	2.01	1.89	1.89	1.89	1.89
Accuracy: $\|\widetilde{D} - D^*\|_F$	1.480	1.060	0.058	0.100	0.100	0.100	0.100
Outer iteration number: N_{iter} in Algorithm 12	50	22	9	4	4	4	4
Inner iteration number: N_{iter}^i in Algorithm 13	972	877	692	312	319	319	318
$\max\|\mathbf{d}_p^T\mathbf{d}_p - 1\|$	0.015	0.001	0.003	0.007	0.007	0.007	0.007
$\max\|G - D^TD\|$	0.052	0.0057	0.0084	10^{-8}	10^{-8}	10^{-8}	10^{-8}
Computing time	155.78	47.31	16.33	3.38	3.42	3.41	3.39

noise, the output of the algorithm cannot be exactly D^*. For this reason, the objective function for $\mu_c > \mu^*$ are always lower than that when $\mu_c = 0.6$. However, when $\mu_c < \mu^*$, the situation is totally different, because μ^* is out of the feasible region. Therefore, a solution that satisfies the coherence constraint can be found, but the price to pay is an increase of the objective function, as well as the computational cost to converge. Consequently, by appropriately choosing the coherent parameter, an incoherent dictionary can be produced by this algorithm. Moreover, the smaller the target coherent parameter is, the greater the computational complexity will be.

5.3.4 Incoherent dictionary learning algorithm

We get an incoherent dictionary learning method by combining the dictionary updating method introduced in the previous section with a sparse coding method, such as the proximal method or the AcMIQP introduced in Chapter 4. By considering AcMIQP, we get an incoherent dictionary learning algorithm with an exact ℓ_0 optimization. The outline is illustrated in Algorithm 14. In the reminder of this section, we examine the relevance of two combinations: prox+EPALM, which combines the proximal method with the hybrid algorithm of ADMM and EPALM, and AcMIQP+EPALM, which combines AcMIQP with the proposed hybrid algorithm.

Algorithm 14 Exact ℓ_0 based incoherent dictionary learning algorithm (5.17)

```
Input: The training data X, number of iterations N, all parameters and stropping
  criterion needed in inexact ADMM algorithm (Algorithm 12) and EPALM algo-
  rithm (Algorithms 13).
Output: The solution D and A
  function INCOHERENT DICTIONARY LEARNING
    Initialisation of dictionary $D_0$ and sparse representation $A = 0$,
    for $i = 1$ to N do
      1. Sparse coding with AcMIQP algorithm.
      2. Dictionary updating by ADMM with EPALM (Algorithms 12-13).
    end for
  end function
```

In the following, the dictionary learning algorithm is evaluated on the segment of image *Barbara* presented in Figure 4.1. The overlapping patches of size 8×8 (namely a signal is a vector of size 64) form the set of signals X. With the signals, a dictionary D is learned by using the proposed method (the hybrid algorithm of the ADMM and EPALM for dictionary updating, and either the proximal method or the AcMIQP for sparse coding) and compared to the other two comparative incoherent dictionary learning algorithms, INK-SVD and IPR. When both D and X are known, the sparse code A can be easily obtained using a sparse coding method, namely proximal method and AcMIQP for our algorithm, OMP algorithm for the other two methods. Then, the reconstructed image is obtained by doing the matrix multiplication $\widetilde{X} = DA$. Consequently, we compare their performance by computing the peak signal-to-noise ratio (PSNR). The MIQP problem is solved by the software Gurobi Optimizer 8.1.0. The parameter settings of Gurobi are fixed as the default values except that the time limit is set to 0.5 seconds and the maximal iteration number to 1000. The initialization of MIQP is given by running the proximal method by setting the maximal iteration number to 200. When only the proximal method is used to sparse code, the iteration number is 1000. The number of atoms is set to $p = 256$ and the sparsity level $k = 20$ (the active atoms is less than 8%). The iteration number for learning a dictionary is set to 30, which is sufficient for the algorithms to converge, as shown in Figure 5.2. For the other two comparative methods, the parameter values are chosen as in the original papers [MBP12, DM13].

Figure 5.1 presents the convergence property of the algorithms with the coherent parameter set to $\mu_c = 0.6$, which corresponds to having angles between any two atoms greater than 53°. It is observable that the dictionary algorithm with MIQP for

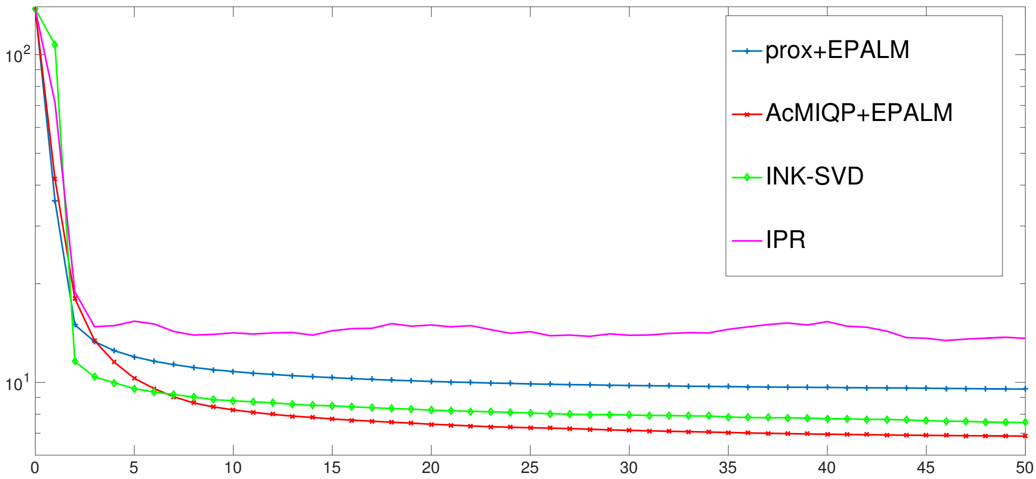


Figure 5.1 – The convergence of the exact ℓ_0 based incoherent dictionary learning algorithm (AcMIQP+EPALM) and its comparison to the prox+EPALM, the INK-SVD and IPR algorithms

sparse coding and EPALM for dictionary updating has the fastest convergence and the value of limit is the smallest. It is worth pointing out that 30 iterations is sufficient for the algorithms to converge, even though the IPR algorithm shows some convergence unstability.

Figure 5.2 shows the distribution of the absolute inner product between each two atoms in the learned dictionary. Combined with the statistics given in Table 5.2, we notice that independently of the used sparse coding algorithm, the proposed method can achieve a dictionary with almost the target coherence parameter value, which is not the case of IPR. The proximal method combined with EPALM provides the smallest absolute average, which is an important property related to the so-called Babel function whose theory is well established [Hon15b, Hon15a, Tro04]. However, this algorithm cannot beat the one with MIQP for sparse coding in terms of variance. The INK-SVD algorithm outputs as well a dictionary with almost the target coherence value, but with a higher variance. Nevertheless, INK-SVD updates the dictionary without considering the reconstruction error (see next paragraph). For the IPR algorithm, the target coherence parameter value cannot be obtained even though it shows the least variance. Considering the distribution of absolute inner products between each two atoms in the learned dictionary, as illustrated in Figure 5.2, it is hard to tell which of the proximal method or the MIQP is better in combination with the proposed algorithm.

To analyze the reconstruction errors, we study seven coherence values $\mu_c = \{0.996, 0.966, 0.866, 0.707, 0.500, 0.259, 0.122\}$, i.e., the angle between any two atoms

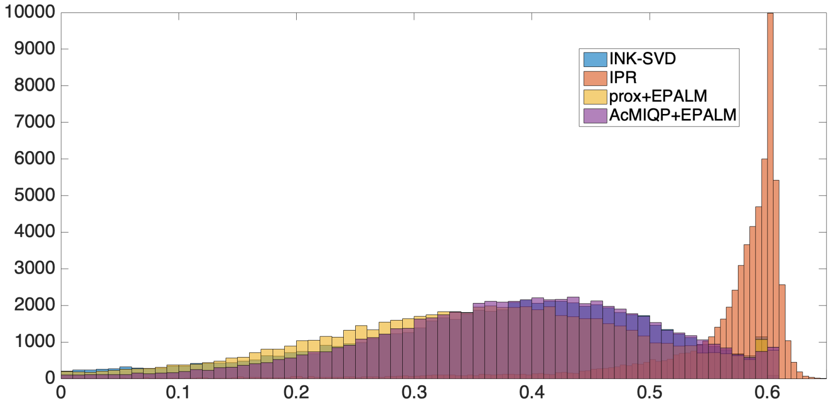


Figure 5.2 – The distribution of the coherence between each two atoms of the proposed algorithms and its comparison to the INK-SVD and IPR algorithms

Table 5.2 – Statistics on the resulting dictionary

	μ	Average of $\{\lvert\mathbf{d}_i^{\mathrm{T}}\mathbf{d}_j\rvert \mid i \neq j\}$	Variance of $\{\lvert\mathbf{d}_i^{\mathrm{T}}\mathbf{d}_j\rvert \mid i \neq j\}$
INK-SVD	0.601	0.368	0.0177
IPR	0.711	0.557	0.0073
Prox+EPALM	0.608	0.352	0.0176
AcMIQP+EPALM	0.609	0.382	0.0146

is bigger than $\theta_c = \{5°, 15°, 30°, 45°, 60°, 75°, 83°\}$. Table 5.3 and Figure 5.3 present the reconstruction errors for each of these target values. For the incoherent dictionary learning with the proposed method of ADMM and EPALM, the reconstruction performance improves with the coherence of dictionary decreasing, with the best results when $\mu_c = \cos(45°)$ with proximal method for sparse coding and $\mu_c = \cos(30°)$ with AcMIQP; afterwards, the reconstruction performance begins to decrease. This is different from the results of INK-SVD and IPR algorithms whose performances monotonically decrease with the coherence (i.e., the incoherence of the dictionary is increasing). Hence, our algorithm increases the dictionary incoherence without the risk of loss in reconstruction accuracy. Furthermore, our method proves that an appropriate incoherent dictionary helps to improve the performance. However, one point should be noticed, incoherent dictionary learning algorithm with AcMIQP for sparse coding has the highest computing complexity comparing to the other meth-

Table 5.3 – The reconstruction errors in PSNR (in dB) by using the dictionary with different coherence parameter values $\mu_c = \cos(\theta_c)$

	Largest angle θ_c between two atoms						
	5°	15°	30°	45°	60°	75°	83°
INK-SVD	36.46	36.56	36.26	34,83	34.04	30.15	-
IPR	36.82	36.57	35.72	31.51	30.60	27.84	-
Prox+EPALM	27.40	27.42	28.06	29.80	29.31	29.75	22.97
AcMIQP+EPALM	**37.60**	**37.26**	**38.89**	**38.55**	**36.52**	**35.31**	**33.97**

ods. It is remarked that there is no result of INK-SVD or IPR when $\theta_c = 83°$. That is caused by the non-convergence of the two algorithms when dealing with some high coherence values. By contrast, the hybrid algorithm of ADMM and EPALM presents good performance in stability.

Figure 5.4 presents the best results of image reconstruction obtained respectively by algorithms of INK-SVD, IPR, prox+EPALM and AcMIQP+EPALM by setting the coherence parameter respectively to $\mu_c = 0.996, 0.996, 0.707, 0.866$. Patches are recovered by $\widetilde{X} = DA$ and then by averaging pixel value, the final reconstructed image is obtained. It is observable that the combination of AcMIQP for sparse coding and the proposed algorithm for dictionary update outperforms the other methods by maintaining most of the details in the image.

5.3.5 Convergence analysis

In this part, we give the proof of convergence of the proposed algorithm for incoherent dictionary learning. As aforementioned, the proposed algorithm aims at tackling the constrained optimization problem by transforming the problem into an unconstrained optimization problem via the augmented Lagrangian method. In each iteration of the augmented Lagrangian method, the minimization problem with respect to the primal variables is solved by the EPALM algorithm. Thus, for proving the convergence of the algorithm, we need to prove the convergence of the augmented Lagrangian method and that of the EPALM.

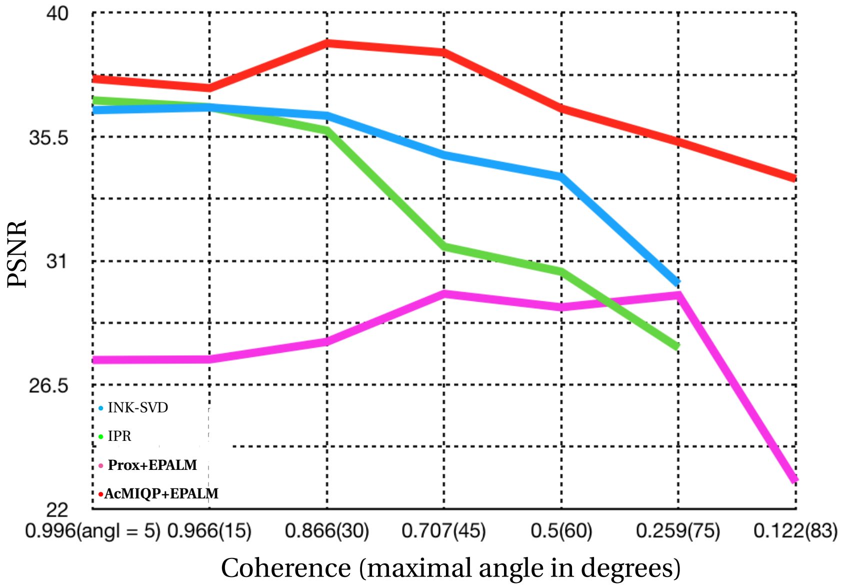


Figure 5.3 – The reconstruction errors in PSNR for each method at each target coherence value

Convergence of the augmented Lagrangian method

Before proceeding and for completeness, we give here the convergence of the augmented Lagrangian method [Ber99]. Consider the general expression of an equality constrained problem:

$$\begin{array}{ll} \min & q(x) \\ \text{subject to} & p(x) = 0, \quad \forall x \in \mathbb{X}, \end{array} \tag{5.29}$$

where $\mathbb{X}$ is a closed set, and q and p are continuous functions in $\mathbb{X}$.

Proposition 3 (Proposition 4.2.1 in [Ber99])**.** *Assume q and p are continuous functions, $\mathbb{X}$ is a closed set, and the constraint set $\{x \in \mathbb{X} \mid p(x) = 0\}$ is nonempty. For $k = 0, 1, \cdots$, let x^k be a global minimum of the optimization problem*

$$\min_{x \in \mathbb{X}} \mathrm{L}_{c^k}(x, \lambda^k), \tag{5.30}$$

where λ^k is bounded, $0 < c^k < c^{k+1}$ for all k, and $c^k \to \infty$. Then every limit point of the sequence $\{x^k\}$ is a global minimum of the original problem (5.29).

Furthermore, according to the Proposition 4.2.2 in [Ber99], the limit of the sequence $\{\lambda^k\}$ can be reached by iteratively updating λ^k through $\widetilde{\lambda}^k = \lambda^k + c^k p(x^k)$,

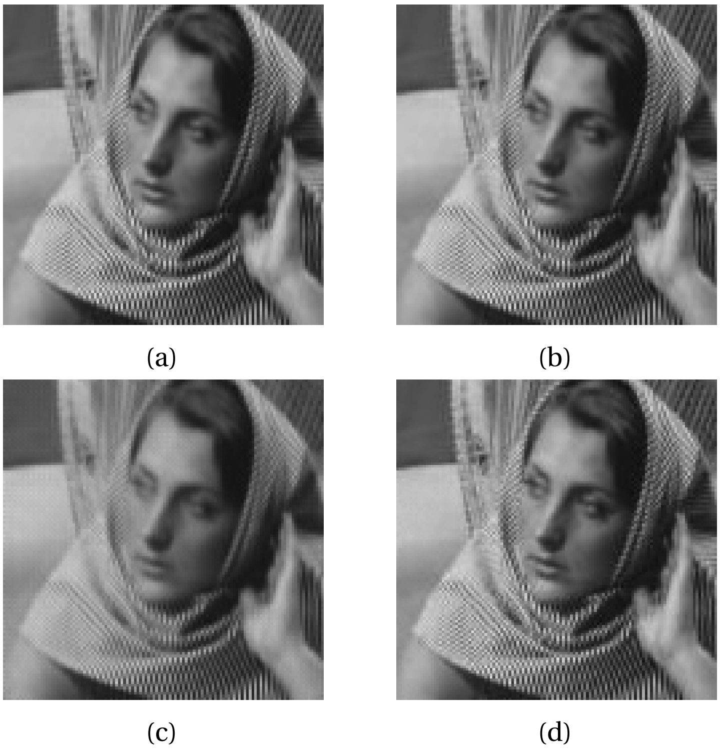


Figure 5.4 – Reconstructed images produced respectively by algorithms (a) INK-SVD, (b) IPR, (c) Prox+EPALM and (d) AcMIQP+EPALM

and $\lim_{k\to\infty}\lambda^k + c^k p(x^k) = \lambda^*$, where λ^* is the optimal value of the dual variable. Then, x^* is the solution of the problem (5.30) when λ^k here is λ^*. Setting the first order derivative $\partial(q+\lambda^* p)(x)$ to zero, the solution x^* can be reached.

Convergence of the EPALM algorithm

The analysis of the convergence of the EPALM algorithm is based on the Kurdyka-Łojasiewicz (KL) equality. We begin by introducing its definition [ABS13]:

Definition 2. *(Kurdyka-Łojasiewicz function)*

(a) The function $f:\mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ *is said to have the Kurdyka-Łojasiewicz property at* $x^* \in \operatorname{dom}\partial f$ *if there exist* $\eta \in (0,+\infty]$, *a neighborhood* U *of* x^* *and a continuous concave function* $\phi:[0,\eta)\to\mathbb{R}_+$ *such that:*

(i) $\phi(0)=0$*;*

(ii) ϕ *is* $\mathbb{C}^1$ *on* $(0,\eta)$*;*

(iii) *for all* $s \in (0, \eta)$, $\phi'(s) > 0$*;*

(iv) *for all* $x \in U \cap [f(x^*) < f < f(x^*) + \eta]$*, the Kurdyka-Łojasiewicz inequality holds:*

$$\phi'(f(x) - f(x^*))dist(0, \partial f(x)) \geq 1. \tag{5.31}$$

(b) *The proper lower semicontinuous functions that satisfy the Kurdyka-Łojasiewicz inequality at each point of* dom ∂f *are called KL functions.*

We now study the convergence property of the algorithm, that is, the convergence of the sequence generated by the proposed algorithm in this chapter. We will prove that, with a and b two fixed positive value, the proposed algorithm generates a sequence $\{x^k\}_{k\in\mathbb{N}}$ that satisfies the following conditions:

H1. *(Sufficient decrease condition).* For each $k \in \mathbb{N}$,

$$f(x^{k+1}) + a\|x^{k+1} - x^k\|^2 \leq f(x^k);$$

H2. *(Relative error condition).* For each $k \in \mathbb{N}$, there exists $w^{k+1} \in \partial f(x^{k+1})$ such that

$$\|w^{k+1}\| \leq b\|x^{k+1} - x^k\|;$$

H3. *(Continuity condition).* There exist a subsequence $\{x^{k_j}\}_{j\in\mathbb{N}}$ and $\tilde{x}$ such that

$$x^{k_j} \to \tilde{x} \text{ and } f(x^{k_j}) \to f(\tilde{x}), \qquad \text{when } j \to \infty. \tag{5.32}$$

Then the following theorem (Theorem 2.9 in [ABS13]) will be used to prove the convergence of the proposed algorithm.

Theorem 5.3.1. *(Convergence to a critical point)*
Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *be a proper lower semicontinuous function. Consider a sequence* $\{x^k\}_{k\in\mathbb{N}}$ *that satisfies the conditions **H1, H2, H3**.*

If f *has the Kurdyka-Łojasiewicz property at the cluster point* $\tilde{x}$ *specified in **H3**, then the sequence* $\{x^k\}_{k\in\mathbb{N}}$ *converges to* $\bar{x}$*, and* $\bar{x} \to \tilde{x}$ *as k goes to infinity, and* $\bar{x}$ *is a critical point of* f*.*

Moreover the sequence $\{x^k\}_{k\in\mathbb{N}}$ *has a finite length, i.e.,*

$$\sum_{k=0}^{+\infty} \|x^{k+1} - x^k\| < +\infty.$$

In the following, we begin with the proof of satisfaction of assumption on functions.

Proposition 4. *The objective function* $\mathrm{L}_{(c_1^i,c_2^i)}(\mathrm{D},\mathrm{G},\boldsymbol{\lambda}^i,\mathrm{H}^i)$ *is a KL function.*

Proof. The objective function $\mathrm{L}_{(c_1^i,c_2^i)}(\mathrm{D},\mathrm{G},\boldsymbol{\lambda}^i,\mathrm{H}^i)$ can be written in form of (5.18), namely $f(\mathrm{D})+h(\mathrm{D},\mathrm{G})+g(\mathrm{G})$. According to [ABRS10] and therein, it is easy to prove that f and h are KL functions. Moreover, g is also a KL function because it is the indicator function of a semi-algebraic set. Hence, the sum of the KL functions, i.e., $\mathrm{L}_{(c_1^i,c_2^i)}(\mathrm{D},\mathrm{G},\boldsymbol{\lambda}^i,\mathrm{H}^i)$, is a KL function. □

Proposition 5. *In problem* (5.29)*, if* $q(x)$ *is a proper semicontinuous function in a closed set* $\mathbb{X}$ *and* $p(x)$ *is a proper lower continuous function in* $\mathbb{X}$*, then the augmented Lagrangian function* $\mathrm{L}_{c^k}(x,\lambda^k)$ *is a proper lower semicontinuous function.*

Proof. Firstly, if $p(x)$ is a continuous function in $\mathbb{X}$, then $\{x \mid p(x) < \infty\} = \mathbb{X}$. Moreover, $q(x)$ is a proper function in $\mathbb{X}$. $\mathrm{L}_{c^k}(x,\lambda^k) = q(x)+\lambda^k p(x)+\frac{c^k}{2}p^2(x)$ is consequently a proper function in $\mathbb{X}$.

Secondly, it is evident that if $p(x)$ is a continuous function in $\mathbb{X}$, then $\lambda^k p(x)$ and $\frac{c^k}{2}p^2(x)$ are continuous functions in $\mathbb{X}$. The sum of a semicontinuous function in $\mathbb{X}$, the function $q(x)$ and a continuous function $\lambda^k p(x)+\frac{c^k}{2}p^2(x)$, is still a semicontinuous function, i.e., $\mathrm{L}_{c^k}(x,\lambda^k)$ is a semicontinuous function.

Finally, p and q are both lower-bounded functions in $\mathbb{X}$, that is, $\forall x \in \mathbb{X}$, $p(x) > -\infty$ and $q(x) > -\infty$. $\lambda^k p(x)+\frac{c^k}{2}p^2(x)$ is a convex function because $c^k > 0$, then, $\lambda^k p(x)+\frac{c^k}{2}p^2(x) > -\infty$, $\forall x \in \mathbb{X}$. Hence, $\mathrm{L}_{c^k}(x,\lambda^k)$ is the sum of two lower-bounded functions in $\mathbb{X}$.

Therefore, $\mathrm{L}_{c^k}(x,\lambda^k)$ is a proper lower semicontinuous function. □

In our optimization problem, $q(x)$ is a proper lower semicontinuous function dedicating here to an indicator term, and $p(x)$ represents all the constraints that can be linear functions and quadratic functions, and which are all proper lower continuous functions. By applying the above Proposition 5, we can deduce that the augmented Lagrangian function in our problem is a proper lower semicontinuous function.

Now, to prove the convergence of the algorithm, we still need to prove that the generated sequence $(\mathrm{D}^{i,j},\mathrm{G}^{i,j})$ satisfies the conditions **H1**, **H2** and **H3**. The sequence is generated from the process

$$\begin{cases} \mathrm{D}^{i,j+1} = \arg\min_{\mathrm{D}\in\mathbb{R}^{n\times m}} \mathrm{P}_1(\mathrm{D}) \\ \mathrm{G}^{i,j+1} = \arg\min_{\mathrm{G}\in\mathbb{R}^{m\times m}} \mathrm{P}_2(\mathrm{G}), \end{cases} \tag{5.33}$$

where the functions P_1 and P_2 are defined as:

$$\begin{cases} \mathrm{P}_1(\mathrm{D}) = f(\mathrm{D}^{i,j}) + h(\mathrm{D}^{i,j},\mathrm{G}^{i,j}) + \mathrm{tr}\big((\mathrm{D}-\mathrm{D}^{i,j})^{\mathrm{T}}\nabla_{\mathrm{D}}(f(\mathrm{D}^{i,j}) + h(\mathrm{D}^{i,j},\mathrm{G}^{i,j}))\big) + \frac{t_1}{2}\|\mathrm{D}-\mathrm{D}^{i,j}\|_{\mathrm{F}}^2 \\ \mathrm{P}_2(\mathrm{G}) = g(\mathrm{G}) + h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j}) + \mathrm{tr}\big((\mathrm{G}-\mathrm{G}^{i,j})^{\mathrm{T}}\nabla_{\mathrm{G}} h(\mathrm{D}^{i,j},\mathrm{G}^{i,j})\big) + \frac{t_2}{2}\|\mathrm{G}-\mathrm{G}^{i,j}\|_{\mathrm{F}}^2. \end{cases} \tag{5.34}$$

Proposition 6. *The process* P_1 *produces a sequence* $\{\mathrm{D}^{i,j}\}$ *that respects the conditions* ***H1, H2*** *and* ***H3****.*

Proof. The three functions $\nabla f : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$, $\nabla_{\mathrm{D}} h : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ and $\nabla_{\mathrm{G}} h : \mathbb{R}^{m\times m} \to \mathbb{R}^{m\times m}$ are all Lipchitz continuous functions on their own domain. Then, there exists a Lipchitz constant $\mathrm{L}_1 = \mathrm{L} + \mathrm{L}_{\mathrm{D}}$, where L is the Lipchitz constant for the function ∇f and L_{D} defined in (5.19), that is

$$\begin{aligned} f(\mathrm{D}^{i,j+1}) + h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j}) \leq f(\mathrm{D}^{i,j}) + h(\mathrm{D}^{i,j},\mathrm{G}^{i,j}) + \frac{\mathrm{L}_1}{2}\|\mathrm{D}^{i,j+1}-\mathrm{D}^{i,j}\|_{\mathrm{F}}^2 \\ +\mathrm{tr}\big((\mathrm{D}^{i,j+1}-\mathrm{D}^{(}i,j))^{\mathrm{T}}\nabla_{\mathrm{D}}(f(\mathrm{D}^{i,j}) + h(\mathrm{D}^{i,j},\mathrm{G}^{i,j})\big). \end{aligned} \tag{5.35}$$

The minimization of the optimization problems (5.33) requires that

$$\mathrm{tr}\big((\mathrm{D}-\mathrm{D}^{i,j})^{\mathrm{T}}\nabla_{\mathrm{D}}(f(\mathrm{D}^{i,j}) + h(\mathrm{D}^{i,j},\mathrm{G}^{i,j})\big) + \tfrac{t_1}{2}\|\mathrm{D}-\mathrm{D}^{i,j}\|_{\mathrm{F}}^2 \leq 0, \tag{5.36}$$

which assures descent in the objective function. By combining inequality (5.36) and the inequality (5.35), we obtain the following result:

$$f(\mathrm{D}^{i,j+1}) + h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j}) + \tfrac{t_1-\mathrm{L}_1}{2}\|\mathrm{D}^{i,j+1}-\mathrm{D}^{i,j}\|_{\mathrm{F}}^2 \leq f(\mathrm{D}^{i,j}) + h(\mathrm{D}^{i,j},\mathrm{G}^{i,j}). \tag{5.37}$$

The satisfaction of condition **H1** can be easily proven by choosing a t_1 greater than the Lipchitz constant L_1.

We now begin to prove the condition **H2**. A large-enough b can be found such that,

$$\|\nabla_{\mathrm{D}}(f(\mathrm{D}^{i,j}) + h(\mathrm{D}^{i,j},\mathrm{G}^{i,j})\| \leq b\|\mathrm{D}^{k+1}-\mathrm{D}^{k+1}\|. \tag{5.38}$$

By considering the Lipchitz continuity of the function $\mathrm{D} \to \nabla(f(\mathrm{D}) + h(\mathrm{D},\mathrm{G}))$ and applying the triangle inequality, the following is deduced:

$$\begin{aligned}\|\nabla_{\mathrm{D}} f(\mathrm{D}^{i,j+1}) + \nabla_{\mathrm{D}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\| \leq & \|\nabla_{\mathrm{D}}\big(f(\mathrm{D}^{i,j}) - h(\mathrm{D}^{i,j},\mathrm{G}^{i,j})\big)\| \\ & + \|\nabla_{\mathrm{D}}\big(f(\mathrm{D}^{i,j+1}) + h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\big) \\ & - \nabla_{\mathrm{D}}\big(f(\mathrm{D}^{i,j}) - h(\mathrm{D}^{i,j},\mathrm{G}^{i,j})\big)\| \\ \leq & (b+\mathrm{L}_1)\|\mathrm{D}^{i,j+1} - \mathrm{D}^{i,j}\|, \end{aligned} \tag{5.39}$$

which is the relative error condition **H2**.

The continuity condition **H3** is satisfied because of the continuity of the functions f and h with respect to D. □

Proposition 7. *The process* P_2 *produces a sequence* $\{\mathrm{G}^{i,j}\}$ *having the properties introduced in conditions **H1, H2** and **H3**.*

Proof. The minimization of the second subproblem in (5.33) assures that,

$$g(\mathrm{G}^{i,j+1}) + \mathrm{tr}\big((\mathrm{G}^{i,j+1} - \mathrm{G}^{i,j})^{\mathrm{T}}\nabla_{\mathrm{G}} h(\mathrm{D}^{i,j},\mathrm{G}^{i,j})\big) + \frac{t_2}{2}\|\mathrm{G}^{i,j+1} - \mathrm{G}^{i,j}\|_{\mathrm{F}}^2 \leq g(\mathrm{G}^{i,j}). \tag{5.40}$$

The function $\mathrm{G} \rightarrow h(\mathrm{D},\mathrm{G})$ is a L_{D}-Lipchitz continuous function. Here, for simplification, let $\mathrm{L}_2 = \mathrm{L}_{\mathrm{G}}$. Thus, the inequality (5.40) becomes

$$g(\mathrm{G}^{i,j+1}) + \frac{-\mathrm{L}_2 + t_2}{2}\|\mathrm{G}^{i,j+1} - \mathrm{G}^{i,j}\|_{\mathrm{F}}^2 \leq g(\mathrm{G}^{i,j}). \tag{5.41}$$

When $t_2 > \mathrm{L}_2$, the condition **H1** is satisfied.

We prove the satisfaction of condition **H2** by using its first order necessary condition:

$$\partial g(\mathrm{G}^{i,j+1}) + t_2(\mathrm{G}^{i,j+1} - \mathrm{G}^{i,j} + \tfrac{1}{t_2}\nabla_{\mathrm{G}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})) = 0. \tag{5.42}$$

Equivalently:

$$\partial g(\mathrm{G}^{i,j+1}) + \nabla_{\mathrm{G}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j}) = -t_2(\mathrm{G}^{i,j+1} - \mathrm{G}^{i,j}). \tag{5.43}$$

Taking the norm on both sides, the following equality holds:

$$\|\partial g(\mathrm{G}^{i,j+1}) + \nabla_{\mathrm{G}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\| = t_2\|\mathrm{G}^{i,j+1} - \mathrm{G}^{i,j}\|. \tag{5.44}$$

Then by applying the triangle inequality, the condition **H2** can be proven:

$$\begin{aligned}\|\partial g(\mathrm{G}^{i,j+1})+\nabla_{\mathrm{G}}h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1})\| \leq & \|\partial g(\mathrm{G}^{i,j+1})+\nabla_{\mathrm{G}}h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\| \\ & +\|\nabla_{\mathrm{G}}h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1})-\nabla_{\mathrm{G}}h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\| \\ \leq & (t_2+\mathrm{L}_2)\|\mathrm{G}^{i,j+1}-\mathrm{G}^{i,j}\|. \end{aligned} \tag{5.45}$$

H3 is satisfied for the continuous function h and the semicontinuous function g in $\mathbb{S}_{\mathrm{G}}$. □

Proposition 8. *The iterative process* P_1 *and* P_2 *produces a sequence* $\{(\mathrm{D}^{i,j},\mathrm{G}^{i,j})\}$ *that satisfies the conditions **H1**, **H2** and **H3**.*

Proof. The Lipchitz continuity of the gradient of $\mathrm{G}\to h(\mathrm{D},\mathrm{G})$ and the inequality (5.45) infer that there exists an $\mathrm{L}'<0$ that verifies

$$h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1})-h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\leq \mathrm{L}'\|\mathrm{G}^{i,j+1}-\mathrm{G}^{i,j}\|_{\mathrm{F}}^2. \tag{5.46}$$

By summing the inequalities (5.37) and (5.41), we get:

$$\begin{aligned} f(\mathrm{D}^{i,j+1})+h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})+g(\mathrm{G}^{i,j+1})+\tfrac{t_1-\mathrm{L}_1}{2}\|\mathrm{D}^{i,j+1}-\mathrm{D}^{i,j}\|_{\mathrm{F}}^2+\tfrac{t_2-\mathrm{L}_2}{2}\|\mathrm{G}^{i,j+1}-\mathrm{G}^{i,j}\|_{\mathrm{F}}^2 \\ \leq f(\mathrm{D}^{i,j})+h(\mathrm{D}^{i,j},\mathrm{G}^{i,j})+g(\mathrm{G}^{i,j}). \end{aligned} \tag{5.47}$$

Using the result of (5.46), the inequality becomes:

$$\begin{aligned} \mathrm{L}_{c_1^i,c_2^i}(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1},\boldsymbol{\lambda}^i,\mathrm{H}^i)+\tfrac{t_1-\mathrm{L}_1}{2}\|\mathrm{D}^{i,j+1}-\mathrm{D}^{i,j}\|_{\mathrm{F}}^2+\tfrac{t_2-\mathrm{L}_2-2\mathrm{L}'}{2}\|\mathrm{G}^{i,j+1}-\mathrm{G}^{i,j}\|_{\mathrm{F}}^2 \\ \leq \mathrm{L}_{c_1^i,c_2^i}(\mathrm{D}^{i,j},\mathrm{G}^{i,j},\boldsymbol{\lambda}^i,\mathrm{H}^i). \end{aligned} \tag{5.48}$$

Setting $a=\min\left(\frac{t_1-\mathrm{L}_1}{2},\frac{t_2-\mathrm{L}_2-2\mathrm{L}'}{2}\right)$, we obtain

$$\mathrm{L}_{c_1^i,c_2^i}(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1},\boldsymbol{\lambda}^i,\mathrm{H}^i)+a\|(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1})-(\mathrm{D}^{i,j},\mathrm{G}^{i,j})\|_{\mathrm{F}}^2\leq \mathrm{L}_{c_1^i,c_2^i}(\mathrm{D}^{i,j},\mathrm{G}^{i,j},\boldsymbol{\lambda}^i,\mathrm{H}^i). \tag{5.49}$$

Thus, the sequence $\left\{(\mathrm{D}^{i,j},\mathrm{G}^{i,j})_{j\in\mathbb{N}}\right\}$ satisfies the condition **H1**.

To prove the condition **H2**, it is necessary to compute the subdifferential of the function (5.18) with respect to the pair of matrix variables $(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1})$, which is denoted by $\partial\mathrm{L}_{(c_1^i,c_2^i)}(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1},\boldsymbol{\lambda}^i,\mathrm{H}^i)$. With the results obtained in (5.39) and

(5.45), we use again the triangle inequality, then

$$\begin{aligned}
\|\partial \mathrm{L}_{(c_1^i,c_2^i)}(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1},\boldsymbol{\lambda}^i,\mathrm{H}^i)\| &= \|\nabla f(\mathrm{D}^{i,j+1}) + \nabla h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1}) + \partial g(\mathrm{G}^{i,j+1})\| \\
&\leq \|\nabla_{\mathrm{D}} f(\mathrm{D}^{i,j+1}) + \nabla_{\mathrm{D}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1})\| \\
&\quad + \|\partial g(\mathrm{G}^{i,j+1}) + \nabla_{\mathrm{G}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1})\| \\
&\leq \|\nabla_{\mathrm{D}} f(\mathrm{D}^{i,j+1}) + \nabla_{\mathrm{D}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\| \\
&\quad + \|\partial g(\mathrm{G}^{i,j+1}) + \nabla_{\mathrm{G}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1})\| \\
&\quad + \|\nabla_{\mathrm{D}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1}) - \nabla_{\mathrm{D}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\|.
\end{aligned}$$

Using the expressions of the partial derivatives in (5.24), then the following inequality holds

$$\begin{aligned}
\|\nabla_{\mathrm{D}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1}) - \nabla_{\mathrm{D}} h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\| &\leq \|\nabla h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1}) - \nabla h(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j})\| \\
&\leq \mathrm{L}\|(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1}) - (\mathrm{D}^{i,j},\mathrm{G}^{i,j})\|,
\end{aligned}$$

where L is the Lipchitz constant of the function h. Combining the inequalities (5.39) and (5.44), the condition **H2** of the global sequence $(\mathrm{D}^{i,j},\mathrm{G}^{i,j})_{j\in\mathbb{N}}$ is obtained

$$\begin{aligned}
\|\partial \mathrm{L}_{(c_1^i,c_2^i)}(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1},\boldsymbol{\lambda}^i,\mathrm{H}^i)\| \leq &(\mathrm{L}_1 + b)\|\mathrm{D}^{i,j+1} - \mathrm{D}^{i,j}\| + (t_2 + \mathrm{L}_2)\|\mathrm{G}^{i,j+1} - \mathrm{G}^{i,j}\| \\
&+ \mathrm{L}\|(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1}) - (\mathrm{D}^{i,j},\mathrm{G}^{i,j})\|.
\end{aligned}$$

Let $t = \max(\mathrm{L}_1 + b + \mathrm{L}, \mathrm{L}_2 + t_2 + \mathrm{L})$, then

$$\|\partial \mathrm{L}_{(c_1^i,c_2^i)}(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1},\boldsymbol{\lambda}^i,\mathrm{H}^i)\| \leq t\|(\mathrm{D}^{i,j+1},\mathrm{G}^{i,j+1}) - (\mathrm{D}^{i,j},\mathrm{G}^{i,j})\|.$$

The condition **H3** is straightforward by considering the continuity of the function.

□

5.4 Conclusion

This section investigated the exact incoherent dictionary learning problem, where the constraint of coherence is explicitly added to formulate the ℓ_0-norm based constrained optimisation problem. To solve this constrained optimization problem, we introduced the new dictionary update algorithm EPALM that combines the proximal alternating minimization method and augmented Lagrangian method are introduced. This algorithm was used for dictionary learning together with a sparse coding algorithm, such as the proximal method and AcMIQP. We showed firstly the feasibility of the algorithm on synthetic data, examining the performance of the dictionary learning independently of the sparse coding algorithm. And then, the incoherent dictionary algorithm was used for real image reconstruction. We studied the statistics of the resulting dictionary, and the reconstruction performance for a large set of target coherence parameters. It was proven that the combination of EPALM for dictionary updating and MIQP for sparse coding always outperformed the other methods in terms of the reconstruction results. The relevance of having an incoherent dictionary was also demonstrated.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This monograph focused on the optimization problem of sparse representation. The problem was formulated as the minimization of the mean square error with a constraint on the sparsity of the decomposition coefficients. Generally, the sparsity of the decomposition coefficients can be promoted by the ℓ_p-norm function for $0 \leq p \leq 1$, specially, $p = 0$ or 1. The ℓ_0-norm can explicitly count the number of non-zero coefficients, nevertheless with the characteristics of non-smoothness and non-convexity, which make the problem NP-hard. The nearest convex norm function to ℓ_0-norm is ℓ_1-norm, but is not differentiable at 0. Additionally, it can achieve the optimal solution under some conditions, such as the signal is sufficiently sparse [Tro04]. However, the resolution of the ℓ_1-norm based optimization problem risks in missing the optimal sparse solution [BNCM15]. Hence, the resolution of ℓ_0-norm based sparse representation problems is of great interest, which was the main topic in this monograph.

The generic problem of sparse representation aims at estimating the decomposition coefficients with the predefined dictionary. This is the so-called sparse coding problem. Moreover, since the quality of the dictionary greatly affects the performance of the sparse representation, researchers have been increasingly interested in data-driven dictionary learning, instead of a predefined dictionary. Thus the dictionary learning problem emerged and began to play a significant role in sparse representation. The dictionary learning estimates jointly the sparse code and the dictionary. This problem is still NP-hard and much more difficult than the single problem of sparse coding. Nevertheless, this optimization problem can be solved by the strategy of iteratively alternating two steps: sparse coding and dictionary updating. In this monograph, we studied the problem of sparse coding and dictionary learning optimization problems based on the exact ℓ_0-norm.

Chapter 3 presented the state-of-the-art methods for sparse coding and dictionary learning. For tackling the problem of sparse coding, three major classes of methods were proposed. The greedy methods, such as (Orthogonal) Matching Pursuit and its variants, solve the problem by iteratively finding the local optimal solution of the ℓ_0-norm optimization problem. Another class is relaxation methods that replace the ℓ_0-norm by the ℓ_1-norm to make the problem convex and tractable, such as Basis Pursuit and LASSO. The last class brings together gradient descent methods and iterative thresholding methods, such as Iterative Soft Thresholding and Iterative Hard Thresholding (IHT). However, these algorithms risk in missing the global optimal solution.

In Chapter 4, we reformulated the sparse coding problem by introducing an auxiliary variable to indicate if the corresponding coefficient is zero or not. Then, the subproblem of sparse coding was transformed to a mixed integer quadratic programming (MIQP). This method can exactly solve the problem by exploiting optimization strategies such as the branch-and-cut. The optimizer solver Gurobi was used for the implementation. Moreover, it is worthy noting that this algorithm had no assumption on the dictionary, which is unlike the IHT or greedy algorithms that need to satisfy conditions such as the RIC. Furthermore, two optimization techniques were proposed to accelerate the MIQP for sparse coding. The first one offered an appropriate initialization. The method of proximal method was used here for producing the initial estimation, since it is considered to be a good approximation of the optimal solution, and thus the searching time was greatly reduced. The second technique was the method of relaxation with additive constraints that consider the convex envelop of the continuous variables. With these two acceleration techniques, the proposed Accelerated MIQP (AcMIQP) was investigated for sparse coding and further for handling the dictionary learning problem. AcMIQP allowed to break the limitation of previous work where the MIQP was only applied on small-scale synthetic data. The proposed algorithm can be used to deal with classical well-known data in signal and image processing. In our experiments, AcMIQP was used in image reconstruction and denoising. More precisely, when dealing with different formulations of the sparse representation problem, experiments on AcMIQP showed that the sparsity constrained based model reached the best results comparing to the error constrained one. Besides, by combining AcMIQP for sparse coding with the coordinate descent algorithm based on SVD for dictionary updating, the resulting dictionary learning method proved its advantage in image recovery with high-level noise, compared to state-of-the-art methods K-SVD, proximal method, and SOUPDIL.

In Chapter 5, we studied the problem of incoherent dictionary learning, since incoherent dictionaries bring in great improvement in sparse representation. In general, this optimization problem can be tackled with two strategies. The first introduces an additive decorrelation step following the step of dictionary updating at each iteration, such as the algorithm INK-SVD. The other reformulates the classical dictionary learning problem by introducing a term of regularization that measures the dictionary coherence, such as the incoherent dictionary learning by proximal method. However, the additive decorrelation method did not show the improved performance with more incoherent dictionary, and the regularized formulation cannot explicitly measure the relationship of performance with the coherence of dictionary varying. In this monograph, we proposed a third strategy, formulating the dictionary learning problem with explicit constraints on the coherence. We proposed a novel method to solve the resulting optimization problem, using the strategy of iteratively alternating sparse coding and dictionary updating. The sparse coding was solved by applying the AcMIQP method. For dictionary updating, we faced an optimization problem with quadratic inequality constraints. For dealing with this problem, we firstly took advantage of Alternating Direction Method of Multipliers (ADMM). Then, with the method of augmented Lagrangian method, the optimization problem was transformed into a unconstrained optimization problem. However, it is a non-convex problem. To overcome this difficulty, the proximal alternating linearized minimization was exploited. This algorithm was used in the application of image reconstruction. It proved for the first time, to our best knowledge, the accuracy improvement of image recovery by appropriate reduction of coherence of the dictionary.

6.2 Future Work

The study of the exact ℓ_0-norm optimization is a promising research topic in sparse representation. In its genesis with the work of Jokar and Pfetsch in [JP08], the exact solution of the sparse coding problem was used to verify some propositions and to provide a criterion to evaluate the relevance of existing greedy and relaxation algorithms. More recently, the resolution of the exact ℓ_0-norm sparse coding problem was studied by Bourguignon et al. in [BNCM15], demonstrating that it can tackle a small-scaled problem with synthetic data. In this monograph, we proposed the algorithm AcMIQP which makes it possible to achieve the exact optimal solution and to be applied to real well-known signal and image processing. Besides sparse coding, dictionary learning was proposed with AcMIQP. Therefore, in the future, the exact ℓ_0-norm could be used to deal with more complex problems. With the exact sparse results, the performance of the other algorithms could be evaluated from one more dimension. Hence, the theoretical conclusion could be verified more rigorously.

Moreover, we proposed a flexible model that is easy to extend. In this monograph, we extended the classical dictionary learning problem to the incoherent dictionary learning problem by adding the constraint of the dictionary. Similarly, for the task of classification, the constraint on the classifier may be added. For solving this problem, one needs just to add one more step to update the classifier following dictionary updating. Besides, if there is more demands on the decomposition coefficients or on the dictionary (e.g. the positivity or low-rank of the dictionary), the formulation of the problem is simple, namely, adding the corresponding constraints. Even for solving these novel problems, the strategy is to iteratively alternating the steps of finding the solution of one variable with fixing all the others. For each subproblem, it may be a classical optimization problem or much more complex, which needs many steps to be transformed into a classical optimization problem. Then, the optimizer solver Cplex or Gurobi could be used to solve the problem, or one could exploit some methods that are less generic and more specific to the sparse representation task at hand.

As already pointed, the shortcoming of the exact ℓ_0-norm method is its high computational complexity. Fortunately, we can benefit from the fact that the optimization theory is well developing and the computational capability of the machine is highly improving. Of particular interest, one may take advantage of the GPU and operate parallel computing. It can thus be predicted that the research on the acceleration of the exact ℓ_0-norm can be an important tendency.

Bibliography

[ABRS10] Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

[ABS13] Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1-2):91–129, 2013.

[AEB$^+$06] Michal Aharon, Michael Elad, Alfred Bruckstein, et al. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311, 2006.

[AFS15] Vahid Abolghasemi, Saideh Ferdowsi, and Saeid Sanei. Fast and incoherent dictionary learning algorithms with application to fMRI. *Signal, Image and Video Processing*, 9(1):147–158, 2015.

[AGH18] Alper Atamturk, Andres Gomez, and Shaoning Han. Sparse and smooth signal estimation: Convexification of l0 formulations. *arXiv preprint arXiv:1811.02655*, 2018.

[AOPT13] Amir Ali Ahmadi, Alex Olshevsky, Pablo A Parrilo, and John N Tsitsiklis. NP-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical Programming*, 137(1-2):453–476, 2013.

[AT16] Taylor B Arnold and Ryan J Tibshirani. Efficient implementations of the generalized lasso dual path algorithm. *Journal of Computational and Graphical Statistics*, 25(1):1–27, 2016.

[BB86] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, 1986.

[BBF$^+$16] Pietro Belotti, Pierre Bonami, Matteo Fischetti, Andrea Lodi, Michele Monaci, Amaya Nogales-Gómez, and Domenico Salvagnin. On handling indicator constraints in mixed integer programming. *Computational Optimization and Applications*, 65(3):545–566, 2016.

[BBL14] Christian Bliek1ú, Pierre Bonami, and Andrea Lodi. Solving mixed-integer quadratic programming problems with IBM-CPLEX: A progress report. In *Proceedings of the twenty-sixth RAMP symposium*, pages 16–17, 2014.

[BD08] Thomas Blumensath and Mike E Davies. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5-6):629–654, 2008.

[BD09] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[BDF07] José M Bioucas-Dias and Mário AT Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image processing*, 16(12):2992–3004, 2007.

[BE07] Alain Billionnet and Sourour Elloumi. Using a mixed integer quadratic programming solver for the unconstrained quadratic 0-1 problem. *Mathematical Programming*, 109(1):55–68, 2007.

[BEFP08] Van Den Berg, Ewout, Friedlander, and Michael P. Probing the Pareto frontier for basis pursuit solutions. *SIAM Journal on Scientific Computing*, 31(2):890–912, 2008.

[Ber97] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

[Ber99] Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

[Bix12] Robert E Bixby. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, pages 107–121, 2012.

[BJQS14] Chenglong Bao, Hui Ji, Yuhui Quan, and Zuowei Shen. L0 norm based dictionary learning by proximal methods with global convergence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3858–3865, 2014.

[BLTW15] Pierre Bonami, Andrea Lodi, Andrea Tramontani, and Sven Wiese. On mathematical programming with indicator constraints. *Mathematical programming*, 151(1):191–223, 2015.

[BM91] Brian Borchers and John E Mitchell. Using an interior point method in a branch and bound algorithm for integer programming. Technical report, 195, Mathematical Sciences, Rensselaer Polytechnic, 1991.

[BM97] Brian Borchers and John E Mitchell. A computational comparison of branch and bound and outer approximation algorithms for 0–1 mixed integer nonlinear programs. *Computers & Operations Research*, 24(8):699–701, 1997.

[BNCM15] Sébastien Bourguignon, Jordan Ninin, Hervé Carfantan, and Marcel Mongeau. Exact sparse approximation problems via mixed-integer programming: Formulations and computational performance. *IEEE Transactions on Signal Processing*, 64(6):1405–1419, 2015.

[BQJ14] Chenglong Bao, Yuhui Quan, and Hui Ji. A convergent incoherent dictionary learning algorithm for sparse coding. In *European Conference on Computer Vision*, pages 302–316. Springer, 2014.

[BRF11] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Hierarchical matching pursuit for image classification: Architecture and fast algorithms. In *Advances in neural information processing systems*, pages 2115–2123, 2011.

[BRF13] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Multipath sparse coding using hierarchical matching pursuit. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 660–667, 2013.

[BST14] Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.

[Buc14] Ioan Buciu. Image denoising via sparse data representation: A comparative study. In *Fundamentals of Electrical Engineering (ISFEE), 2014 International Symposium on*, pages 1–4. IEEE, 2014.

[BYD07] Thomas Blumensath, Mehrdad Yaghoobi, and Mike E Davies. Iterative hard thresholding and l0 regularisation. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 3, pages III–877. IEEE, 2007.

[CD95] Scott Chen and David L Donoho. Examples of basis pursuit. In *Wavelet Applications in Signal and Image Processing III*, volume 2569, pages 564–575. International Society for Optics and Photonics, 1995.

[CDS01] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

[CG15] Laming Chen and Yuantao Gu. Local and global optimality of lp minimization for sparse recovery. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3596–3600. IEEE, 2015.

[CH97] CF Chen and CH Hsiao. Haar wavelet method for solving lumped and distributed-parameter systems. *IEE Proceedings-Control Theory and Applications*, 144(1):87–94, 1997.

[Chu16] Charles K Chui. *An introduction to wavelets.* Elsevier, 2016.

[CJY16] Weiqiang Chen, Hui Ji, and Yanfei You. An augmented Lagrangian method for 1-regularized optimization problems with orthogonality constraints. *SIAM Journal on Scientific Computing*, 38(4):B570–B592, 2016.

[CK93] Tianhorng Chang and C-C Jay Kuo. Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on image processing*, 2(4):429–441, 1993.

[CPR13] Rakesh Chalasani, Jose C Principe, and Naveen Ramakrishnan. A fast proximal method for convolutional sparse coding. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–5. IEEE, 2013.

[CRT06] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[CYV00a] S Grace Chang, Bin Yu, and Martin Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE transactions on image processing*, 9(9):1532–1546, 2000.

[CYV00b] S Grace Chang, Bin Yu, and Martin Vetterli. Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Transactions on image Processing*, 9(9):1522–1531, 2000.

[Dau88] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7):909–996, 1988.

[DDDM04] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

[DDTS06] David Leigh Donoho, Iddo Drori, Yaakov Tsaig, and Jean-Luc Starck. *Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit*. Department of Statistics, Stanford University, 2006.

[DE07] Foi Alessandro Katkovnik Vladimir Dabov, Kostadin and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[DH01] David L Donoho and Xiaoming Huo. Uncertainty principles and ideal atomic decomposition. *IEEE transactions on information theory*, 47(7):2845–2862, 2001.

[DM09] Wei Dai and Olgica Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE transactions on Information Theory*, 55(5):2230–2249, 2009.

[DM13] Barchiesi Daniele and D. Plumbley Mark. Learning incoherent dictionaries for sparse approximation using iterative projections and rotations. *IEEE Transactions on Signal Processing*, 61(8):2055–2065, April 2013.

[Don06] David L Donoho. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(6):797–829, 2006.

[DV05] Minh N Do and Martin Vetterli. The contourlet transform: An efficient directional multiresolution image representation. *IEEE Transactions on image processing*, 14(12):2091–2106, 2005.

[DZSW11] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011.

[EA06] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[EAH99] Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 5, pages 2443–2446. IEEE, 1999.

[EFM10] Michael Elad, Mario AT Figueiredo, and Yi Ma. On the role of sparse and redundant representations in image processing. *Proceedings of the IEEE*, 98(6):972–982, 2010.

[EHJ+04] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[Ela10] Michael Elad. *Sparse and redundant representations: From theory to applications in signal and image processing.* Springer Science & Business Media, 2010.

[FISI16] Yasuhiro Fujiwara, Yasutoshi Ida, Hiroaki Shiokawa, and Sotetsu Iwamura. Fast lasso algorithm via selective coordinate descent. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[FLT11] Matteo Fischetti, Andrea Lodi, and Andrea Tramontani. On the separation of disjunctive cuts. *Mathematical Programming*, 128(1-2):205–230, 2011.

[FM11] GM Fung and OL Mangasarian. Equivalence of minimal l0-and lp-norm solutions of linear equalities, inequalities and linear programs for sufficiently small p. *Journal of optimization theory and applications*, 151(1):1–10, 2011.

[FMP+18] Mingbin Feng, John E. Mitchell, Jong-Shi Pang, Xin Shen, and Andreas Wächter. Complementarity formulations of l0-norm optimization. *Pacific Journal of Optimization*, 14(2):273–305, 2018.

[FNZ+15] Zizhu Fan, Ming Ni, Qi Zhu, Chengli Sun, and Lipan Kang. L0-norm sparse representation based on modified genetic algorithm for face recognition. *Journal of Visual Communication and Image Representation*, 28:15–20, 2015.

[Fou11] Simon Foucart. Hard thresholding pursuit: An algorithm for compressive sensing. *SIAM Journal on Numerical Analysis*, 49(6):2543–2563, 2011.

[GE16] Alona Golts and Michael Elad. Linearized kernel dictionary learning. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):726–739, 2016.

[GG09] Pierre Garrigues and Laurent E Ghaoui. An homotopy algorithm for the lasso with online observations. In *Advances in neural information processing systems*, pages 489–496, 2009.

[GJB15] Rémi Gribonval, Rodolphe Jenatton, and Francis Bach. Sparse and spurious: Dictionary learning with noise and outliers. *IEEE Transactions on Information Theory*, 61(11):6298–6319, 2015.

[GK09] Rahul Garg and Rohit Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In *Proceedings of the 26th International Confer- ence on Machine Learning*, volume 9, pages 337–344, 2009.

[GMS$^+$86] Philip E Gill, Walter Murray, Michael A Saunders, John A Tomlin, and Margaret H Wright. On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method. *Mathematical programming*, 36(2):183–209, 1986.

[Gra95] Amara Graps. An introduction to wavelets. *IEEE computational science and engineering*, 2(2):50–61, 1995.

[GW13] Ambros M Gleixner and Stefan Weltge. Learning and propagating Lagrangian variable bounds for mixed-integer nonlinear programming. In *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*, pages 355–361. Springer, 2013.

[HA07] Ke Huang and Selin Aviyente. Sparse representation for signal classification. In *Advances in neural information processing systems*, pages 609–616, 2007.

[HGT06] Kyle K Herrity, Anna C Gilbert, and Joel A Tropp. Sparse approximation via iterative thresholding. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 3, pages III–III. IEEE, 2006.

[HHO99] Aapo Hyvärinen, Patrik Hoyer, and Erkki Oja. Image denoising by sparse code shrinkage. In *Intelligent Signal Processing*. Citeseer, 1999.

[Hon15a] Paul Honeine. Analyzing sparse dictionaries for online learning with kernels. *IEEE Transactions on Signal Processing*, 63(23):6343–6353, 2015.

[Hon15b] Paul Honeine. Approximation errors of online sparsification criteria. *IEEE Transactions on Signal Processing*, 63(17):4700 – 4709, September 2015.

[Hon16] Paul Honeine. An eigenanalysis of data centering in machine learning. *CoRR*, abs/1407.2904:1–13, March 2016.

[HR13] Karla L Hoffman and Ted K Ralphs. Integer and combinatorial optimization. *Encyclopedia of Operations Research and Management Science*, pages 771–783, 2013.

[Jer73] Robert C Jeroslow. There cannot be any algorithm for integer programming with quadratic constraints. *Operations Research*, 21(1):221–224, 1973.

[JLD13] Zhuolin Jiang, Zhe Lin, and Larry S Davis. Label consistent K-SVD: Learning a discriminative dictionary for recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2651–2664, 2013.

[JMOB10] Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and Francis R Bach. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, volume 1, page 2. Citeseer, 2010.

[JP08] Sadegh Jokar and Marc E Pfetsch. Exact and approximate sparse solutions of underdetermined linear equations. *SIAM Journal on Scientific Computing*, 31(1):23–44, 2008.

[KD09] Vladimir Katkovnik Karen Egiazarian Kostadin Dabov, Alessandro Foi. BM3D image denoising with shape-adaptive principal component analysis. In *Workshop on Signal Processing Adaptive Sparse Structed Representations (SPARS), 2009*, 2009.

[KD16] Kuldeep Kumar and Christian Desrosiers. A sparse coding approach for the efficient representation and segmentation of white matter fibers. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, pages 915–919. IEEE, 2016.

[KG08] Ramkumar Karuppiah and Ignacio E Grossmann. A Lagrangean based branch-and-cut algorithm for global optimization of nonconvex mixed-integer nonlinear programs with decomposable structures. *Journal of global optimization*, 41(2):163–186, 2008.

[KKK08] Jinseog Kim, Yuwon Kim, and Yongdai Kim. A gradient-based optimization algorithm for lasso. *Journal of Computational and Graphical Statistics*, 17(4):994–1009, 2008.

[KM82] J Kormylo and J Mendel. Maximum likelihood detection and estimation of Bernoulli-Gaussian processes. *IEEE transactions on information theory*, 28(3):482–488, 1982.

[Kow14] Matthieu Kowalski. Thresholding rules and iterative shrinkage/thresholding algorithm: A convergence study. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4151–4155. IEEE, 2014.

[Kru03] A Ya Kruger. On Fréchet subdifferentials. *Journal of Mathematical Sciences*, 116(3):3325–3358, 2003.

[KXAH15] M Amin Khajehnejad, Weiyu Xu, A Salman Avestimehr, and Babak Hassibi. Improving the thresholds of sparse recovery: An analysis of a two-step reweighted basis pursuit algorithm. *IEEE Transactions on Information Theory*, 61(9):5116–5128, 2015.

[KY03] Harold Kushner and G George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

[LBRN07] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2007.

[LCHR19] Yuan Liu, Stéphane Canu, Paul Honeine, and Su Ruan. Mixed integer programming for sparse coding: Application to image denoising. *IEEE Transactions on Computational Imaging*, 5(3):354–365, Sep. 2019.

[LCWM09] Zhouchen Lin, Minming Chen, Leqin Wu, and Yi Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Coordinated Science Laboratory Report no. UILU-ENG-09-2215, DC-247*, 2009.

[LDHL17] Zhenni Li, Shuxue Ding, Takafumi Hayashi, and Yujie Li. Incoherent dictionary learning with log-regularizer based on proximal operators. *Digital Signal Processing*, 63:86–99, 2017.

[LDL15] Zhenni Li, Shuxue Ding, and Yujie Li. A fast algorithm for learning overcomplete dictionary for sparse representation based on proximal operators. *Neural computation*, 27(9):1951–1982, 2015.

[Lee96] Tai Sing Lee. Image representation using 2D Gabor wavelets. *IEEE Transactions on pattern analysis and machine intelligence*, 18(10):959–971, 1996.

[LHDL16] Zhenni Li, Takafumi Hayashi, Shuxue Ding, and Xiang Li. An efficient algorithm for incoherent analysis dictionary learning based on proximal operator. In *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pages 003546–003549. IEEE, 2016.

[LHT$^+$18] Huafeng Li, Xiaoge He, Dapeng Tao, Yuanyan Tang, and Ruxin Wang. Joint medical image fusion, denoising and enhancement via discriminative low-rank sparse dictionaries learning. *Pattern Recognition*, 79:130–146, 2018.

[LLF14] Liangyue Li, Sheng Li, and Yun Fu. Learning low-rank and discriminative dictionary for image classification. *Image and Vision Computing*, 32(10):814–823, 2014.

[LLZ12] Tong Lin, Shi Liu, and Hongbin Zha. Incoherent dictionary learning for sparse representation. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1237–1240. IEEE, 2012.

[LPM00] Erwan Le Pennec and Stéphane Mallat. Image compression with geometrical wavelets. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 1, pages 661–664. IEEE, Sep. 2000.

[LS99] Jeff T Linderoth and Martin WP Savelsbergh. A computational study of search strategies for mixed integer programming. *INFORMS Journal on Computing*, 11(2):173–187, 1999.

[LW07] Yufeng Liu and Yichao Wu. Variable selection via a combination of the l0 and l1 penalties. *Journal of Computational and Graphical Statistics*, 16(4):782–798, 2007.

[Mal99] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[MAN13] Mudassir Masood and Tareq Y Al-Naffouri. Sparse reconstruction using distribution agnostic Bayesian matching pursuit. *IEEE Transactions on Signal Processing*, 61(21):5298–5309, 2013.

[MB$^+$06] Nicolai Meinshausen, Peter Bühlmann, et al. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, 34(3):1436–1462, 2006.

[MBP11] Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):791–804, 2011.

[MBP12] Boris Mailhé, Daniele Barchiesi, and Mark D Plumbley. INK-SVD: Learning incoherent dictionaries for sparse representations. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3573–3576. IEEE, 2012.

[MBP$^+$14] Julien Mairal, Francis Bach, Jean Ponce, et al. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2-3):85–283, 2014.

[MBPS09] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.

[MBPS10] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

[MBZJ07] G Hosein Mohimani, Massoud Babaie-Zadeh, and Christian Jutten. Fast sparse representation based on smoothed l0 norm. In *International Conference on Independent Component Analysis and Signal Separation*, pages 389–396. Springer, 2007.

[MCW04] Dmitry M Malioutov, Müjdat Cetin, and Alan S Willsky. Optimal sparse representations in general overcomplete bases. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages ii–793. IEEE, 2004.

[MCW05a] Dmitry Malioutov, Müjdat Cetin, and Alan S Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE transactions on signal processing*, 53(8):3010–3022, 2005.

[MCW05b] Dmitry M Malioutov, Müjdat Cetin, and Alan S Willsky. Homotopy continuation for sparse signal representation. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–733. IEEE, 2005.

[Mei07] Nicolai Meinshausen. Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1):374–393, 2007.

[Mey95] Yves Meyer. *Wavelets and operators*, volume 1. Cambridge university press, 1995.

[Mit02] John E Mitchell. Branch-and-cut algorithms for combinatorial optimization problems. *Handbook of applied optimization*, 1:65–77, 2002.

[MKCP17] Panos P Markopoulos, Sandipan Kundu, Shubham Chamadia, and Dimitris A Pados. Efficient l1-norm principal-component analysis via bit flipping. *IEEE Transactions on Signal Processing*, 65(16):4252–4264, 2017.

[MLB$^+$08] Julien Mairal, Marius Leordeanu, Francis Bach, Martial Hebert, and Jean Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. In *European Conference on Computer Vision*, pages 43–56. Springer, 2008.

[MMWW02] Hugues Marchand, Alexander Martin, Robert Weismantel, and Laurence Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 123(1-3):397–446, 2002.

[MP06] Luis Mancera and Javier Portilla. L0-norm-based sparse representation through alternate projections. In *2006 International Conference on Image Processing*, pages 2089–2092. IEEE, 2006.

[MPS$^+$09] Julien Mairal, Jean Ponce, Guillermo Sapiro, Andrew Zisserman, and Francis R Bach. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009.

[MSE08] Julien Mairal, Guillermo Sapiro, and Michael Elad. Learning multiscale sparse representations for image and video restoration. *Multiscale Modeling & Simulation*, 7(1):214–241, 2008.

[MXAP11] João FC Mota, João MF Xavier, Pedro MQ Aguiar, and Markus Puschel. Distributed basis pursuit. *IEEE Transactions on Signal Processing*, 60(4):1942–1956, 2011.

[MY09] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The annals of statistics*, 37(1):246–270, 2009.

[MY12] Julien Mairal and Bin Yu. Complexity analysis of the lasso regularization path. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1835–1842. Omnipress, 2012.

[MZ93] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[NHCD10] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. Efficient and robust feature selection via joint l2, 1-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821, 2010.

[Nik13] Mila Nikolova. Description of the minimizers of least squares regularized with l0-norm: Uniqueness of the global minimizer. *SIAM Journal on Imaging Sciences*, 6(2):904–937, 2013.

[Nik16] Mila Nikolova. Relationship between the optimal solutions of least squares regularized with l0-norm and constrained by k-sparsity. *Applied and Computational Harmonic Analysis*, 41(1):237–265, 2016.

[NM65] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[NS04] Arnold Neumaier and Oleg Shcherbina. Safe bounds in linear and mixed-integer linear programming. *Mathematical Programming*, 99(2):283–296, 2004.

[NT09] Deanna Needell and Joel A Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009.

[NV09] Deanna Needell and Roman Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of computational mathematics*, 9(3):317–334, 2009.

[NW06] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer, 2006.

[OF96] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.

[PB14] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[PC08] Trevor Park and George Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[PH07] Mee Young Park and Trevor Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.

[PP12] Robert Peharz and Franz Pernkopf. Sparse nonnegative matrix factorization with l0-constraints. *Neurocomputing*, 80:38–46, 2012.

[PRK93] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

[RBE10] Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.

[RLS09] Ignacio Ramírez, Federico Lecumberry, and Guillermo Sapiro. Sparse modeling with universal priors and learned incoherent dictionaries. In *University of Minnesota. Institute for Mathematics and Its Applications*, 2009.

[RNCR15] Nikhil Rao, Robert Nowak, Christopher Cox, and Timothy Rogers. Classification with the sparse group lasso. *IEEE Transactions on Signal Processing*, 64(2):448–463, 2015.

[RNF17] Saiprasad Ravishankar, Raj Rao Nadakuditi, and Jeffrey A Fessler. Efficient sum of outer products dictionary learning (soup-dil) and its application to inverse problems. *IEEE transactions on computational imaging*, 3(4):694–709, 2017.

[RS08] Fernando Rodriguez and Guillermo Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries. Technical report, MINNESOTA UNIV MINNEAPOLIS, 2008.

[RSS10] Francesco Rinaldi, Fabio Schoen, and Marco Sciandrone. Concave programming for minimizing the zero-norm over polyhedral sets. *Computational Optimization and Applications*, 46(3):467–486, 2010.

[RZE08] Ron Rubinstein, Michael Zibulevsky, and Michael Elad. Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. Technical report, Computer Science Department, Technion, 2008.

[SBFA17] Emmanuel Soubies, Laure Blanc-Féraud, and Gilles Aubert. A unified view of exact continuous penalties for l2-l0 minimization. *SIAM Journal on Optimization*, 27(3):2034–2060, 2017.

[SCD02] Jean-Luc Starck, Emmanuel J Candès, and David L Donoho. The curvelet transform for image denoising. *IEEE Transactions on image processing*, 11(6):670–684, 2002.

[Sch14] Karin Schnass. On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD. *Applied and Computational Harmonic Analysis*, 37(3):464–491, 2014.

[SDMFH01] Tom Schouwenaars, Bart De Moor, Eric Feron, and Jonathan How. Mixed integer programming for multi-vehicle path planning. In *2001 European Control Conference (ECC)*, pages 2603–2608. IEEE, 2001.

[SE12] Leslie N Smith and Michael Elad. Improving dictionary learning: Multiple dictionary updates and coefficient reuse. *IEEE Signal Processing Letters*, 20(1):79–82, 2012.

[ŞE13] Mehmet Umut Şen and Hakan Erdogan. Linear classifier combination and selection using group sparse regularization and hinge loss. *Pattern Recognition Letters*, 34(3):265–274, 2013.

[SIBD11] Charles Soussen, Jérôme Idier, David Brie, and Junbo Duan. From Bernoulli–Gaussian deconvolution to sparse signal restoration. *IEEE Transactions on Signal Processing*, 59(10):4572–4584, 2011.

[SIDB15] Charles Soussen, Jérome Idier, Junbo Duan, and David Brie. Homotopy based algorithms for L0-regularized least-squares. *IEEE Transactions on Signal Processing*, 63(13):3301–3316, 2015.

[SMF15] Jean-Luc Starck, Fionn Murtagh, and Jalal Fadili. *Sparse image and signal processing: Wavelets and related geometric multiscale analysis*. Cambridge university press, 2015.

[Sol07] Daniel Solow. Linear and nonlinear programming. *Wiley Encyclopedia of Computer Science and Engineering*, 2007.

[SPZ08] Philip Schniter, Lee C Potter, and Justin Ziniel. Fast Bayesian matching pursuit. In *2008 Information Theory and Applications Workshop*, pages 326–333. IEEE, 2008.

[SS06] Suvrajeet Sen and Hanif D Sherali. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Mathematical Programming*, 106(2):203–223, 2006.

[SWDS15] Rim Slama, Hazem Wannous, Mohamed Daoudi, and Anuj Srivastava. Accurate 3D action recognition using learning on the Grassmann manifold. *Pattern Recognition*, 48(2):556–567, 2015.

[SYGL14] Xiaoshuang Shi, Yujiu Yang, Zhenhua Guo, and Zhihui Lai. Face recognition by sparse discriminant analysis via joint l2, 1-norm minimization. *Pattern Recognition*, 47(7):2447–2453, 2014.

[TG07] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[TLZ$^+$19] Hongzhong Tang, Xiao Li, Xiaogang Zhang, Dongbo Zhang, Lizhen Mao, and Ting Liu. Coherence-regularized discriminative dictionary learning for histopathological image classification. *Signal, Image and Video Processing*, 13(5):923–931, 2019.

[Tro04] Joel A Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242, 2004.

[TS04] Mohit Tawarmalani and Nikolaos V Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical programming*, 99(3):563–591, 2004.

[Uns95] Michael Unser. Texture classification and segmentation using wavelet frames. *IEEE Transactions on image processing*, 4(11):1549–1560, 1995.

[UWY+15] Michael Ulbrich, Zaiwen Wen, Chao Yang, Dennis Klockner, and Zhaosong Lu. A proximal gradient method for ensemble density functional theory. *SIAM Journal on Scientific Computing*, 37(4):A1975–A2002, 2015.

[VAN08] Juan Pablo Vielma, Shabbir Ahmed, and George L Nemhauser. A lifted linear programming branch-and-bound algorithm for mixed-integer conic quadratic programs. *INFORMS Journal on Computing*, 20(3):438–450, 2008.

[VNPNC12] Hien Van Nguyen, Vishal M Patel, Nasser M Nasrabadi, and Rama Chellappa. Kernel dictionary learning. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2021–2024. IEEE, 2012.

[vSG00] Oskar von Stryk and Markus Glocker. Decomposition of mixed-integer optimal control problems using branch and bound and sparse direct collocation. *The 4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems, Dortmund*, September 18-19:99–104, 2000.

[WGL17] Qiong Wang, Junbin Gao, and Hong Li. Grassmannian manifold optimization assisted sparse spectral clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5258–5266, 2017.

[WGY10] Zaiwen Wen, Donald Goldfarb, and Wotao Yin. Alternating direction augmented Lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2(3-4):203–230, 2010.

[WJY+16] Fei Wu, Xiao-Yuan Jing, Xinge You, Dong Yue, Ruimin Hu, and Jing-Yu Yang. Multi-view low-rank dictionary learning for image classification. *Pattern Recognition*, 50:143–154, 2016.

[WKS12] Jian Wang, Seokbeop Kwon, and Byonghyo Shim. Generalized orthogonal matching pursuit. *IEEE Transactions on signal processing*, 60(12):6202–6216, 2012.

[WL08] Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244, 2008.

[WNF09] Stephen J Wright, Robert D Nowak, and Mário AT Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[WYG$^+$08] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2008.

[WYWW11] Min Wang, Shuyuan Yang, Yanyan Wan, and Jing Wang. High resolution radar imaging based on compressed sensing and fast Bayesian matching pursuit. In *2011 International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping*, pages 1–5. IEEE, 2011.

[XZJ13] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1107–1114, 2013.

[YLY12] XiaoTong Yuan, Xiaobai Liu, and Shuicheng Yan. Visual classification with multitask joint sparse representation. *IEEE Transactions on Image Processing*, 21(10):4349–4360, 2012.

[YLZ14] Xiaotong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *International Conference on Machine Learning*, pages 127–135, 2014.

[YLZ18] Xiao-Tong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit. *Journal of Machine Learning Research*, 18:1–43, 2018.

[YPXD09] JingYu Yang, YiGang Peng, WenLi Xu, and QiongHai Dai. Ways to sparse representation: An overview. *Science in China series F: information sciences*, 52(4):695–703, 2009.

[YY13] Junfeng Yang and Xiaoming Yuan. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of computation*, 82(281):301–329, 2013.

[ZHY+17] Yun Zhou, Jianghong Han, Xiaohui Yuan, Zhenchun Wei, and Richang Hong. Inverse sparse group lasso model for robust object tracking. *IEEE Transactions on Multimedia*, 19(8):1798–1810, 2017.

[ZJD13] Yangmuzi Zhang, Zhuolin Jiang, and Larry S Davis. Learning structured low-rank representations for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 676–683, 2013.

[ZL10] Qiang Zhang and Baoxin Li. Discriminative K-SVD for dictionary learning in face recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2691–2698. IEEE, 2010.

[ZMZ+13] Wangmeng Zuo, Deyu Meng, Lei Zhang, Xiangchu Feng, and David Zhang. A generalized iterated shrinkage algorithm for non-convex sparse coding. In *Proceedings of the IEEE international conference on computer vision*, pages 217–224, 2013.

[Zou06] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

[ZSJ+16] Guoqing Zhang, Huaijiang Sun, Zexuan Ji, Guiyu Xia, Lei Feng, and Quansen Sun. Kernel dictionary learning based discriminant analysis. *Journal of Visual Communication and Image Representation*, 40:470–484, 2016.

[ZXY+15] Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang. A survey of sparse representation: Algorithms and applications. *IEEE access*, 3:490–530, 2015.

[ZY06] Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006.

[ZZC+17] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[ZZCL17] Hong Zhu, Xiaowei Zhang, Delin Chu, and Li-Zhi Liao. Nonconvex and nonsmooth optimization with generalized orthogonality constraints: An approximate augmented Lagrangian method. *Journal of Scientific Computing*, 72(1):331–372, 2017.

Résumé

Cette monographie traite du problème d'apprentissage de dictionnaire parcimonieux associé à la pseudo-norme ℓ_0. Ce problème est classiquement traité par une procédure de relaxation alternée itérative en deux phases : un codage parcimonieux (*sparse coding*) et une réactualisation du dictionnaire. Cependant, le problème d'optimisation associé à ce codage parcimonieux s'avère être non convexe et NP-difficile, ce qui a justifié la recherche de relaxations et d'algorithmes gloutons pour obtenir une bonne approximation de la solution globale du problème. A l'inverse, nous reformulons le problème comme un programme quadratique mixte en nombres entiers (MIQP) permettant d'obtenir l'optimum global du problème. La principale difficulté de cette approche étant le temps de calcul, nous proposons deux méthodes (la relaxation par l'ajout de contraintes complémentaires et l'initialisation par la méthode du gradient proximal) permettant de le réduire. Cet algorithme est baptisé MIQP accéléré (AcMIQP). L'application de AcMIQP à un problème de débruitage d'images démontre sa faisabilité et ses bonnes performances. Nous proposons ensuite d'améliorer cet algorithme en y intégrant des contraintes visant à promouvoir l'indépendance des atomes du dictionnaire sélectionné. Pour traiter ce problème à l'aide de AcMIQP, la phase de réactualisation du dictionnaire sous contraintes est adaptée en combinant la méthode du lagrangien augmenté (ADMM) et la méthode *Extended Proximal Alternating Linearized Minimization* (EPALM). L'efficacité de cette approche AcMIQP+EPALM est démontrée sur un problème de reconstruction d'image.

Abstract

In this monograph, we study the exact ℓ_0 based sparse representation problem. For the classical dictionary learning problem, the solution is obtained by iteratively processing two steps: sparse coding and dictionary updating. However, even the problem associated with sparse coding is non-convex and NP-hard. The method for solving this is to reformulate the problem as mixed integer quadratic programming (MIQP). Then by introducing two optimization techniques, initialization by proximal method and relaxation with augmented contraints, the algorithm is greatly speed up (which is thus called AcMIQP) and applied in image denoising, which shows the good performance. Moreover, the classical problem is extended to learn an incoherent dictionary. For dealing with this problem, AcMIQP or proximal method is used for sparse coding. As for dictionary updating, augmented Lagrangian method (ADMM) and extended proximal alternating linearized minimizing method are combined. This exact ℓ_0 based incoherent dictionary learning is applied in image recovery, which illustrates the improved performance with a lower coherence.