2011-01-01

Digital Image Processing Based on Sparse Representation and Convex Programming

Carlos Andres Ramirez
University of Texas at El Paso, carlosrv19@gmail.com

DIGITAL IMAGE PROCESSING BASED ON SPARSE REPRESENTATION AND CONVEX PROGRAMMING

CARLOS ANDRES RAMIREZ VILLAMARIN

Program in Computational Science

APPROVED:

Miguel Argáez, Ph.D., Chair

Sergio Cabrera, Ph.D.

Osvaldo Mendez, Ph.D.

Benjamin C. Flores, Ph.D.
Acting Dean of the Graduate School

A mi madre y

en memoria de mi padre,

con amor.

DIGITAL IMAGE PROCESSING BASED ON SPARSE REPRESENTATION AND CONVEX PROGRAMMING

by

CARLOS ANDRES RAMIREZ VILLAMARIN

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Program in Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

December 2011

Acknowledgements

I would like to express my deep gratitude to my advisor Doctor Miguel Argáez, for his guidance, orientation, and most importantly, for giving me the opportunity to start pursuing my dream of becoming a scientist. He is a person of opportunities, he believed in me when I had first failed the TOEFL exam, when the only thing he knew about me was just that I loved mathematics. I am also profoundly indebted to Doctor Leticia Velázquez who has always been concerned with my growth as a researcher and as a person. She is also responsible for such a wonderful life-opportunity of studying in the U.S.

I am grateful with my committee members Doctor Osvaldo Mendez, who with admirable enthusiasm helped me to clarify some mathematical ideas, and specially with Doctor Sergio Cabrera who has always had an open door for me.

I would also like to thank my class mates Anibal, Paula and Reinaldo for their support, but most of all for their friendship. I specially thank Aline for her guidance in writing, and for her patient and inspiring energy.

Finally, I would like to thank the financial support provided by the ARL Grant No. W911NF-07-2-0027, that has made possible my graduate studies.

Abstract

Sparse representation models have been of central interest in recent years due to important achievements in computational harmonic analysis, such as wavelet transformations, and the most recent sampling theory, compressed sensing. Numerous applications based on sparse models have been studied in the last decade leading to promising results. These applications include areas in seismology, image processing, wireless sensor networks, computed tomography and magnetic resonance imaging just to mention a few.

In this work, we propose to extend such applications in the area of image processing, particularly for the image segmentation problem, and examine algorithms involved in sparse modeling from both theoretical and numerical perspectives. In particular, we focus on the Path Following Signal Recovery (PFSR) algorithm introduced by Argáez et al. in 2010, and the Fixed-Point Least-Squares Preconditioned Conjugate Gradient (FPLS_PCG) algorithm, presented for the first time in this thesis.

Numerical results are presented supporting our ideas in sparse modeling, specifically for solving the image denoising, image deblurring, image separation, and image inpainting problem.

Table of Contents

Chapter 1

Introduction

1.1 Overview

Providing a precise and meaningful description of a phenomenon or dataset is of central interest in the scientific research community. Inspired by this task, multi-resolution analysis led to the popularization of wavelets as a tool for representing certain types of signals in the early 90's. This effort was followed by a new area known as *sparse representation*, where a class of signals denoted as *compressible* are expressed as a linear combination of only a few elements of a certain basis. We say that a given signal-vector u has a sparse representation x if it can be expressed as

$$u = \Psi x,$$

where the number of nonzero elements of the vector x is by far less than its dimension, and Ψ is referred as the sparsifying matrix.

As it turns out, most natural images have a sparse representation in the wavelet domain[1], which has led to the development of new approaches for solving image processing problems exploiting compressibility as prior information (see for example [13, 17]).

The new sampling theory of compressed sensing has unified several insights about wavelets and sparse representation, benefiting several disciplines in sciences including image processing. Practical compressed sensing problems involve solving an optimization problem of the form

$$\min_x \|x\|_0 \ \text{ subject to } \ Ax = b, \tag{1.1}$$

[1]To be precise, most natural images have a nearly-sparse representation in the wavelet domain. Nearly-sparsity is defined in next chapter.

for decoding a sparse signal $x \in \mathbb{R}^n$ that has been significantly sub-sampled by a sampling matrix $A \in \mathbb{R}^{m \times n}$ with $m < n$. Here, $\|x\|_0$ counts the number of non-zero entries of the vector x.

Solving (1.1) amounts to finding the sparsest vector x such that $Ax = b$. Nevertheless, finding such a vector x is by nature combinatorial and generally an NP-hard problem [35]. The most popular strategy to overcome this difficulty is to replace the ℓ_0 norm with the ℓ_1 norm. This technique is known as *convex relaxation* since the ℓ_0 norm is relaxed with its closest convex function. In this view, and under some assumptions, problem (1.1) becomes

$$\min_x \|x\|_1 \quad \text{subject to} \quad Ax = b. \tag{1.2}$$

Problem (1.2) has proven to be successful at approximately solving (1.1) [5, 8, 9, 11]. Moreover, under some mild conditions, the optimal set of (1.2) is equal to the optimal set of (1.1) [4, 5, 9, 10].

This problem is known as ℓ_1 recovery problem, and plays a fundamental role in our research proposal since it has the same form of the sparse representation model.

1.2 Contribution

In recent years, some image processing problems have been successfully solved using the sparse representation model. A paper by Elad et al. published in 2010, summarizes most of these applications [15]. Nonetheless, little has been said about the possibility of applying the sparse representation model to the image segmentation problem. We believe that classical methods for segmenting images can benefit from the new theory of sparse representation when introducing the sparsity of the image as a prior information. On the other hand, because images generally lead to large scale problems when solving (1.2), efficient optimization techniques are of special interest.

Consequently, in this thesis we consider two main research directions. First, we aim to investigate and develop efficient algorithms for solving the ℓ_1 recovery problem. Second,

we look for extending the sparse representation model to solve image processing problems, particularly those involving image segmentation.

1.3 Outline

This thesis is organized as follows.

Chapter 2 presents the problem formulation, and some definitions related to the sparse representation theory.

Chapter 3 presents the methodology proposed for solving the central research problem, and describes some state-of-the-art algorithms.

Chapter 4 presents numerical experimentations that demonstrate the capability of the methodology proposed to solve the ℓ_1 recovery problem. In addition, image processing problems are solved with the sparse model, showing the viability of this methodology in imaging science.

Chapter 5 presents concluding remarks and discusses some thoughts on future research.

Chapter 2

Problem formulation

In this chapter, we present the central framework of the research proposal along with two particular problems to be investigated. Problem 1, defined in section 2.4.1, focuses on developing efficient algorithms for ℓ_1 signal recovery. Problem 2, defined in section 2.4.2, aims to investigate image processing applications using the sparse model, particularly for the image segmentation problem.

Before proceeding to formulate our research problem, we describe the basic concepts involved in sparse representation and its connection with image processing.

2.1 Notation and definitions

Throughout this thesis, we model signals as vectors. A signal x of length n is represented as a vector in $\mathbb{R}^n$, and its i-th component is referred as x_i, for $i = 1, \ldots, n$. Unless stated differently, we always assume x to be of length n. A discrete 2D image of size $\sqrt{n} \times \sqrt{n}$ is represented as a vector of size n produced by piling the columns of the image consecutively. With no loss of generality, we only consider square images whose sides length are a power of two. The support of a vector x is defined as

$$\operatorname{supp}(x) = \{i \mid x_i \neq 0\}.$$

We say that a signal x is ***sparse*** if it has only a few nonzero components. That is, $|\operatorname{supp}(x)| \ll n$, where $|\cdot|$ is the cardinal operator [1]. The energy of a vector x is measured with the ℓ_2 norm, that is, $\|x\|_2$. A signal x is said to be ***nearly sparse*** if most of its

[1] The symbol $|\cdot|$ also refers to the absolute value operator. The use of this notation will be clear from the context.

energy is concentrated in a small subset of indexes $I \subseteq \{1, \cdots, n\}$. A way to visualize this is by rearranging the entries of x in absolute value, from the largest to the smallest, and checking whether they decay rapidly.

2.2 Measure of sparsity

Several measures of sparsity have been proposed in the mathematical analysis community. A summary of the most important of such measures is presented below.

2.2.1 The ℓ_p norm

The ℓ_p norm is defined as

$$\|x\|_p^p = \sum_{i=1}^{n} |x_i|^p, \quad p \geq 1.$$

For $p < 1$, the ℓ_p norm is no longer a formal norm since it does not satisfy the triangular inequality. In the sparse representation literature, however, it is common to use the term "norm" to refer this family of quasi-norms. We shall also use this terminology keeping in mind this reservation. When $p \in (0, 1]$, the ℓ_p norm tends to produce lower values for sparse vectors, penalizing vectors that are dense. For this reason, the $\ell_{p\in(0,1]}$ norm is widely used for promoting and measuring sparsity. As an illustration, Figure 2.1 shows the behavior of the scalar function $|x|^p$ for $p = 0.1, 0.5, 1$, and 2. Notice for example that when $p = 0.1$, $|x|^p$ is 0 for $x = 0$, and approaches to 1 elsewhere penalizing nonzero entries.

2.2.2 The ℓ_0 norm

Figure 2.1 gives us an intuition about the behavior of $\|x\|_p^p$ for $p \in (0, 1]$. It is natural to ask for the limit of the ℓ_p norm as p goes to zero. This question gives rise to the definition of the ℓ_0 norm:

$$\|x\|_0 = \lim_{p\to 0} \|x\|_p^p = \lim_{p\to 0} \sum_{i=1}^{n} |x_i|^p = |\text{supp}(x)|.$$

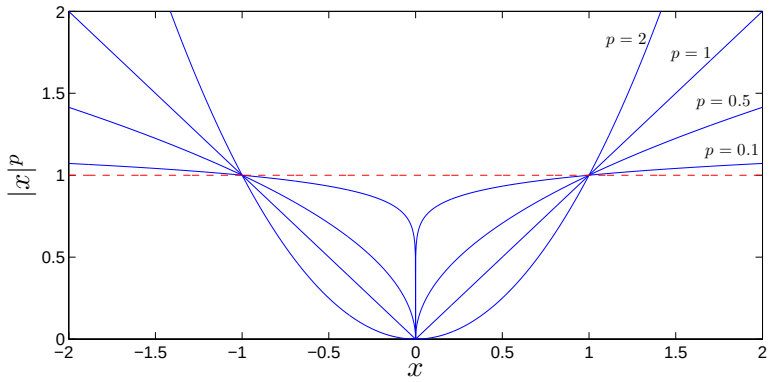

Figure 2.1: Behavior of the scalar function $|x|^p$ for different values of p approaching to zero.

Because $|\text{supp}(x)|$ is equivalent to counting the number of nonzero components in the vector x, the ℓ_0 norm is the most suitable measure of sparsity. Unfortunately, the function $\|x\|_0$ lacks of nice properties such as continuity and convexity, preventing standard numerical optimization techniques to be applied directly in models involving ℓ_0.

2.3 Sparsity in images

Previous work in the area of Computational Harmonic Analysis have shown that nearly-sparsity in the discrete cosine and wavelet domain can be used as a plausible image model [26, 27, 30]. In fact, the JPEG and JPEG-2000 encoding standards are based on the capability of the Discrete Cosine Transform (DCT) and Wavelet Transform (WT) to sparsely represent natural images, respectively [12]. It turns out that this is not an isolated case since natural images have redundant information coming from extensive nearly constant regions on them.

As an example, Figure 2.2 shows a typical natural image and a randomly generated image with the corresponding wavelet coefficients. The rearranged wavelet coefficients in absolute value are also presented. It can be seen that for the natural image, only a few wavelet coefficients capture almost all the energy of the image, whereas the energy of the randomly generated image is spread out along almost all the coefficients. This particular feature makes

of special interest the problem of finding sparse signals among many others that are not. This is the scenario that motivates our central problem presented in next section.

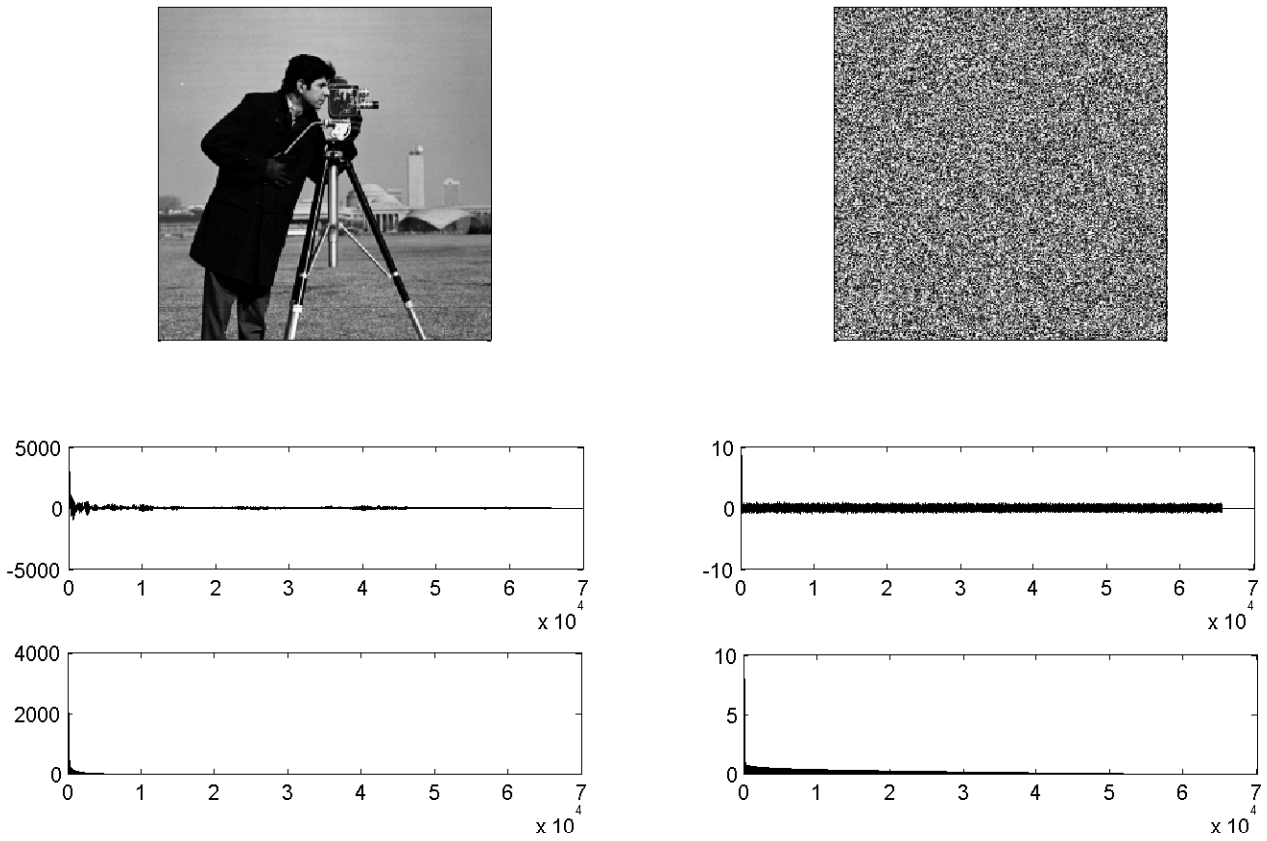

Figure 2.2: (Top left) a natural 256×256 image. (Middle left) wavelet-haar level 3 coefficients of the natural image. (Bottom left) rearranged wavelet coefficients of the natural image in absolute value. (Top right) a randomly generated 256×256 image. (Middle right) wavelet-haar level 3 coefficients of the randomly generated image. (Bottom right) rearranged wavelet coefficients of the randomly generated image in absolute value.

2.4 Problem formulation

We are interested in finding the sparsest solution of the underdetermined system of linear equations

$$Ax = b, \tag{2.1}$$

where $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and A is an $m \times n$ matrix with $m < n$. If we further assume that the matrix A has full rank, then the solution set of (2.1) has infinitely many elements from which we demand the sparsest one. This problem can be posed as

$$\min_x \|x\|_0 \;\text{ subject to }\; Ax = b. \tag{P_0}$$

Solving (P_0) is by nature combinatorial and generally an NP-hard problem [35]. To overcome this difficulty, the applied mathematical community uses the ℓ_1 norm which is the nearest convex function to $\|x\|_0$. Indeed, the use of the ℓ_1 norm for promoting sparse solutions has been first considered for geophysicists since approximately 1979 [36]. In this view, and under some assumptions, problem (P_0) becomes

$$\min_x \|x\|_1 \;\text{ subject to }\; Ax = b. \tag{P_1}$$

The main advantage of formulation (P_1) is that, unlike problem (P_0), it can be solved in polynomial time. The reason for this is that the ℓ_1 minimization problem can be converted into a linear programming problem which is proven to be solvable in polynomial time using interior-point methods [41].

Problem (P_1) has proven to be successful at approximately solving (P_0) [5, 8, 9, 11]. Moreover, under some mild conditions, the optimal set of (P_1) is equal to the optimal set of (P_0) [5, 9, 10, 4].

The equivalence between (P_1) and (P_0) has given rise to *convex relaxation* techniques, where the ℓ_0 norm is relaxed to its closest convex function ℓ_1, and convex optimization approaches can be considered. Furthermore, results obtained by Candès in what has been denominated *robust compressive sampling* [4], have shown that convex relaxation models

also approximate the ℓ_0 problem when the observed signal b is contaminated with additive noise ν whose energy is upper bounded by ϵ. That is,

$$\min_x \|x\|_1 \;\text{ subject to }\; Ax+\nu = b. \tag{P_1^ν}$$

This formulation, although rarely taken into account in the mathematical research community, is considered in [1] for designing a path-following signal recovery algorithm, where a conjugate gradient method is used to solve a sequence of resulting linear systems. In that approach, further studied in next chapter, the stopping criteria of the conjugate gradient algorithm is dictated by the level of noise ϵ, reducing dramatically the number of CG iterations for obtaining an optimal solution.

In contrast, the applied mathematical community considers the following formulation

$$\min_x \|x\|_1 \;\text{ subject to }\; \|Ax-b\|_2 \leq \epsilon, \tag{P_1^ϵ}$$

where ϵ is a measure of corruption of the observed signal as pointed out before. Notice that in the absence of noise, ϵ is equal to zero and both (P_1^ϵ) and (P_1^ν) becomes (P_1).

In order to solve problem (P_1^ϵ), we consider an equivalent formulation given by the following unconstrained optimization problem

$$\min_x \frac{1}{2}\|Ax-b\|_2^2 + \lambda\|x\|_1, \tag{P_1^λ}$$

for an appropriate penalization parameter $\lambda \in (0, \|A^T b\|_\infty)$ [33]. A proof of the equivalence between (P_1^ν), (P_1^ϵ) and (P_1^λ) is given in appendix A. The parameter λ is regarded as a penalization parameter that balances the optimal solution between sparsity and feasibility. Notice that if $\lambda \to 0^+$, (P_1^λ) turns into a simple least squares problem whose solution does not care about sparsity. On the other hand, if λ is too large, the least squares term in (P_1^λ) is insignificant and the solution tends to zero. One advantage of this formulation is that the inequality constraint in (P_1^ϵ) is incorporated in the objective function, yielding an unconstrained optimization problem. Nonetheless, the non-differentiability of $\|x\|_1$ still needs to be addressed in order to apply standard optimization techniques. To that end, we

use the relaxation $\|x\|_1 \approx \sum_{i=1}^{n} \sqrt{x_i^2 + \mu}$ for $\mu > 0$ introduced in [1] and studied in next chapter.

Formulation (P_1^λ) is the central problem in our research proposal, and inspires the two problems described below.

2.4.1 Developing efficient ℓ_1 recovery algorithms

Large scale problems that follow formulation (P_1^λ) demand efficient ℓ_1 recovery algorithms. We propose to investigate new alternatives to improve existing sparse recovery algorithms, and to develop new approaches to solve (P_1^λ). Specifically, we aim to enrich the Path Following Signal Recovery (PFSR) algorithm introduced by Argáez et al. in [1], and the Fixed Point Least Squares Preconditioned Conjugate Gradient (FPLS_PCG) algorithm introduced in this thesis.

2.4.2 Developing new image processing techniques for image segmentation

Image segmentation consists in partitioning an input image into several non-overlapping sub-images, that can be distinguished from each other because every single sub-image possesses certain properties. Currently used techniques such as Cannys algorithm, Prewitt masking or derivatives based algorithms do not take significant advantage of sparse representation.

We propose to investigate new techniques for solving the image segmentation problem, using the sparse representation model.

2.5 Discussion

We have shown in Figure 2.2 that natural images have a nearly-sparse representation in the wavelet domain [2]. This assertion when introduced as a prior in image modeling, leads us to

[2]However, the wavelet domain is not the only one with this property.

verify interesting results in image processing. For instance, measurement processes involved in image acquisition through convolution operations, are posed as linear projections

$$H\Psi x = b,$$

where $\Psi x = u$ is an original scene that has been degraded by a linear operator H, and b corresponds with the observed image. In order to recover the image u, we can include our prior that the image has a nearly-sparse representation in the Ψ domain. In this manner, we can recover the image u by solving the optimization problem

$$\min_x \|x\|_1 \text{ subject to } H\Psi x = b,$$

that amounts to demand the sparsest vector that satisfies the linear relation governing the acquisition process. The model above described, motivates the two research problems presented in this chapter. Primarily, we aim to develop efficient algorithms for solving the sparse recovery problem, and to extend traditional image processing techniques to the new theory of sparse representation.

Chapter 3

Methodology

In this chapter, we present a survey of some methodologies used to address problem (P_1^λ), and focus on the Path Following Signal Recovery (PFSR) algorithm introduced by Argáez et al. in [1] and the Fixed Point Least Squares Preconditioned Conjugate Gradient (FPLS_PCG) algorithm introduced in this thesis.

In the following, unless stated differently, we consider the matrix $A \in \mathbb{R}^{m\times n}$ has full rank, $m < n$, $b \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$.

3.1 l1_ls Algorithm

The l1_ls (ℓ_1 Least Squares) algorithm, introduced in 2007 by Kim et al. [25], rewrites formulation (P_1^λ) as

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}\|Ax-b\|_2^2 + \lambda \sum_{i=1}^{n} u_i \\ \text{subject to} \quad & -u_i \leq x_i \leq u_i, \end{aligned}$$

and solves it using logarithm barrier interior-point methods with inexact Newton's directions. The resulting Newton's system has the form

$$\begin{bmatrix} 2tA^TA + D_1 & D_2 \\ D_2 & D_1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = - \begin{bmatrix} 2tA^T(Ax-b) + h_1 \\ t\lambda \mathbb{1} - h_2 \end{bmatrix},$$

where D_1, $D_2 \in \mathbb{R}^{n\times n}$ are diagonal matrices that depend on x and u, and h_1, $h_2 \in \mathbb{R}^n$ are vectors also depending on x and u. This linear system is always positive definite and is solved using a preconditioned conjugate gradient PCG method.

The l1_ls algorithm is implemented in MATLAB and can be freely downloaded from `http://www.stanford.edu/ boyd/l1_ls/`.

3.2 GPSR Algorithm

The GPSR (Gradient Projection for Sparse Reconstruction) algorithm, introduced in 2007 by Figueiredo et al. [16], rewrites problem (P_1^λ) as an equivalent quadratic programming problem

$$\min_z \quad \frac{1}{2}\| [A \ -A] \, z - b\|_2^2 + \lambda c^T z$$
$$\text{subject to} \quad z \geq 0,$$

for an appropriate choice of c and z, and solves it using gradient projection methodologies.

The GPSR algorithm is implemented in MATLAB and can be freely downloaded from `http://www.lx.it.pt/ mtf/GPSR/.`

3.3 FPC Algorithm

The FPC (Fixed Point Continuation) algorithm, introduced in 2007 by Hale et. al [21], uses subgradient optimality conditions to characterize the optimal set of the problem

$$\min_x \ \frac{\mu}{2}\|Ax - b\|_2^2 + \|x\|_1,$$

which is equivalent to (P_1^λ) for an appropriate penalization parameter μ. This characterization leads to a fixed point formulation that is iteratively solved, and where the principal computational effort is in the calculation of matrix-vector multiplications at every step.

The FPC algorithm is implemented in MATLAB and can be freely downloaded from `http://www.caam.rice.edu/ optimization/L1/fpc/`

3.4 NESTA Algorithm

The NESTA (Nesterov's Algorithm) algorithm, introduced in 2010 by Becker et. al [2], considers formulation (P_1^ϵ) and applies the Nesterov's method for non-smooth convex optimization [31], where the ℓ_1 norm is approximated by

$$\|x\|_1 \approx \max_{u \in Q_d} u^T x - \mu \frac{1}{2}\|u\|_2^2$$

as $\mu \to 0$, and $Q_d = \{u : \ \|u\|_\infty \leq 1\}$.

The NESTA algorithm is implemented in MATLAB and can be freely downloaded from `http://www-stat.stanford.edu/~candes/nesta/`.

3.5 PFSR Algorithm

The PFSR (Path Following Signal Recovery) algorithm, introduced in 2010 by Argáez et al. [1], solves a sequence of relaxed subproblems of the form (P_1^ν) where the ℓ_1 norm term is approximated by a strictly convex multiquadric function that depends on a perturbed parameter $\mu > 0$. That is,

$$\begin{aligned} \min_x \quad & \sum_{i=1}^{n} \sqrt{x_i^2 + \mu} \\ \text{subject to} \quad & Ax + \nu = b. \end{aligned} \tag{3.1}$$

Problem (3.1) is strictly convex, and therefore has a unique global solution for a fixed μ. Additionally, unlike the original problem (P_1^ν), this new objective is continuously differentiable, so classical constrained optimization techniques can be applied. Moreover, if μ is regarded as a continuous parameter, the set containing the solution of all subproblems (3.1) defines a smooth curve called the central path that converges to an optimal solution of problem (P_1^ν) [1] (see Figure 3.1).

There are three fundamental aspects in the implementation of the PFSR algorithm: first, the characterization of the optimality set for each subproblem of the form (3.1), second, the

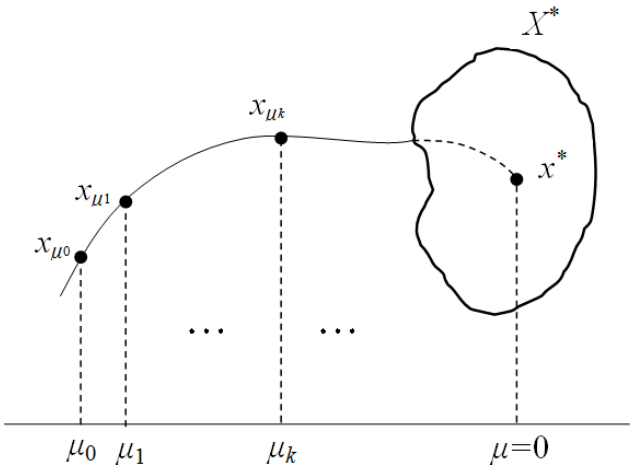

Figure 3.1: Central path. The solution x_{μ_k} of each subproblem (3.1) lies on a smooth curve called central path that converges to an optimal point x^* of the original problem $\min_x \{\|x\|_1 \mid Ax + \nu = b\}$.

equivalence between (3.1) as $\mu \to 0$ and (P_1^ν), and third, the way in which the regularization parameter μ is updated.

Characterization of the optimality set

The Lagrangian function associated with (3.1) is

$$l(x, y) = \sum_{i=1}^{n} \sqrt{x_i^2 + \mu} + (Ax + \nu - b)^T y,$$

where $y \in \mathbb{R}^m$ is the Lagrange multiplier for the equality constraint. Therefore, the KKT conditions are given by the following square system of nonlinear equations:

$$F_\mu(x, y) = \begin{bmatrix} D_\mu(x)^{-1/2} x + A^T y \\ Ax + \nu - b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{3.2}$$

where $D_\mu(x) = \text{diag}(x_i^2 + \mu)$ for $i = 1, 2, \ldots, n$, and $\mu > 0$.

It is important to point out that since each subproblem (3.1) is strictly convex, the KKT conditions are both sufficient and necessary. Consequently, a point (x, y) satisfying (3.2) is the only global minimizer of (3.1).

In order to solve (3.2), a fixed-point method is applied. To that effect, the set of nonlinear equations is written as the augmented system

$$\begin{bmatrix} D_\mu(x)^{-1/2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ b-\nu \end{bmatrix}, \tag{3.3}$$

and iteratively solved for a current $(x,y)_k$ until the difference between $(x,y)_k$ and $(x,y)_{k-1}$ satisfies a predefined tolerance. The linear system induced by fixing $(x,y)_{k-1}$ and solving for $(x,y)_k$ in (3.3), is solved by applying a conjugate gradient algorithm to the equivalent normal equations

$$\begin{cases} AD_\mu(x)^{1/2}A^T y &= \nu - b \\ x &= -D_\mu(x)^{1/2}A^T y \end{cases} \tag{3.4}$$

The structure exhibited in (3.4) allows the conjugate gradient to exploit fast matrix-vector multiplications associated with A and A^T. Moreover, the $m \times m$ matrix involved in this system is generally smaller than the one solved in the l1_ls algorithm. The price one has to pay is that the diagonal matrix $D_\mu(x)^{1/2}$ eventually deteriorates the condition number of the matrix $AD_\mu(x)^{1/2}A^T$ in (3.4).

Equivalence between (3.1) as $\mu \to 0$ and (P_1^ν)

The fixed point formulation (3.3) for a given μ, yields a solution $(x,y)_\mu$ which is said to lie on the central path [1]. As it turns out, the sequence of solutions $(x,y)_\mu$ as $\mu \to 0$ converges to an optimal solution of (P_1^ν) (See [1]).

Updating the regularization parameter μ

A necessary condition for x to be an optimal point of (P_1^ν) is that $\tilde{x}^T\tilde{z} = 0$, where $\tilde{x} = |x|$ and $\tilde{z} = \mathbb{1} - |A^T y|$ (See [1]). Here $|\cdot|$ is the absolute value operated component-wise on a vector, and y is the Lagrange multiplier associated with the equality constraint in (P_1^ν) [1]. Then, following the same idea of primal-dual methods we update the regularization parameter as

$$\mu = \sigma \frac{\tilde{x}^T\tilde{z}}{n},$$

where σ is a centering parameter in $(0, 1)$, and n is the size of the primal variable x.

Updating the regularization parameter in this manner allows us to systematically reduce μ, while achieving necessary conditions for the optimality of (P_1^ν).

The PFSR algorithm is implemented in MATLAB and can be freely downloaded from `http://www.math.utep.edu/Student/cramirez/`.

3.6 FPLS_PCG Algorithm

The FPLS_PCG (Fixed Point Least Squares Preconditioned Conjugate Gradient) algorithm, extends some ideas presented in the PFSR algorithm for approximating the absolute value with a continuously differentiable function. Specifically, the FPLS_PCG algorithm considers formulation (P_1^λ) where the ℓ_1 norm is relaxed by a sum of strictly convex multiquadric functions, and the optimal points are characterized by a fixed point equation.

Therefore, the strategy consists in solving a sequence of unconstrained problems of the form

$$\min_x \ f_\mu(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda \sum_{i=1}^{n} \sqrt{x_i^2 + \mu}, \tag{3.5}$$

as the regularization parameter μ tends to zero.

First order optimality conditions

The gradient of $f_\mu(x)$ is

$$\nabla f_\mu(x) = A^T Ax - A^T b + \lambda D_\mu^{-\frac{1}{2}}(x)x,$$

where $D_\mu(x) = \text{diag}\,(x_i^2 + \mu)$ for $i = 1, \cdots, n$.

The optimal solution is obtained by solving the following system of nonlinear equations

$$(A^T A + \lambda D_\mu^{-\frac{1}{2}}(x))x = A^T b. \tag{3.6}$$

Since the matrix $D_\mu^{-\frac{1}{2}}(x)$ is positive definite, it follows that $(A^TA + \lambda D_\mu^{-\frac{1}{2}}(x))$ is nonsingular and problem (3.6) can be cast as a fixed point problem of the form

$$x = \mathcal{F}_\mu(x), \tag{3.7}$$

where

$$\mathcal{F}_\mu(x) = (A^TA + \lambda D_\mu^{-\frac{1}{2}}(x))^{-1}A^Tb.$$

In order to solve (3.7), we proceed by taking an initial point $x\_$, and iteratively computing $x = \mathcal{F}_\mu(x\_)$ until two consecutive points x and $x\_$ are sufficiently close, as it is done in fixed point algorithms. To that end, we define a measure of closeness in the same manner as in [1, 28]. That is,

$$\frac{\|x - x\_\|_2}{1 + \|x\_\|_2} \leq \sqrt{\mu}.$$

This criterion allows us to solve the fixed point equation (3.7) with lower precision far away from the solution, avoiding unnecessary computational cost.

The preconditioned conjugate gradient

We propose to solve the linear equation

$$(A^TA + \lambda D_\mu^{-\frac{1}{2}}(x\_))x = A^Tb, \tag{3.8}$$

using a preconditioned conjugate gradient method, where $x\_$ is fixed.

Since the matrix A is an underdetermined matrix with rank m, the number of zero eigenvalues of A^TA is exactly $n - m$. As a result, the matrix $(A^TA + \lambda D_\mu^{-\frac{1}{2}}(x\_))$ in (3.8) is likely to have eigenvalues close to zero whenever any element of $D_\mu^{-\frac{1}{2}}(x\_)$ is small.

To overcome this difficulty, we precondition the system (3.8) with

$$M = \frac{1}{\lambda}D_\mu^{\frac{1}{2}}(x\_), \tag{3.9}$$

moving all the eigenvalues above one. This is explained in the following property.

Proposition 1 *The resulting matrix* $\left(MA^TA + I\right)$ *after preconditioning the system (3.8) with* M *in (3.9), has all its eigenvalues greater or equal to one.*

Proof Let λ_{MA} be any eigenvalue of MA^TA. We show that $\lambda_{MA} \geq 0$, and therefore all eigenvalues of $(MA^TA + I)$ are greater or equal to one.

We proceed by contradiction. Assume $\lambda_{MA} < 0$. Since λ_{MA} is any eigenvalue of MA^TA, the matrix $(MA^TA - \lambda_{MA}I)$ is singular. Moreover, the matrix $M^{-1}(MA^TA - \lambda_{MA}I) = A^TA - \lambda_{MA}M^{-1}$ is also singular. Now, since by construction M^{-1} is positive definite, it follows that $A^TA - \lambda_{MA}M^{-1}$ is also positive definite, which leads to a contradiction. □

Furthermore, since the $n - m$ zero eigenvalues of A^TA are the same zero eigenvalues of MA^TA, it follows that $(MA^TA + I)$ has $n - m$ eigenvalues equal to one and m eigenvalues greater than one. Consequently, the PCG algorithm needs at most $m + 1$ iterations for convergence in infinite arithmetic.

The preconditioned conjugate gradient algorithm for solving the linear system (3.8) is presented in Algorithm 1.

The convergence of the conjugate gradient algorithm is dramatically improved when an efficient preconditioner is applied. In our case, we not only benefit from this fact, but also from the fast matrix-vector multiplication algorithm available for our preconditioner M. Therefore, the most expensive part of Algorithm 1 occurs in Steps 1b and 3a, where a matrix-vector multiplication is computed. This is an additional advantage, since for some applications such as compressed sensing and image processing, there exist fast matrix-vector algorithms for both A and A^T.

The algorithm

We present our FPLS_PCG algorithm in Algorithm 2. The FPLS_PCG algorithm is composed of three fundamental stages. First, the initial point is computed in Step 1. A commonly used initial point is the minimum energy norm or Moore-Penrose solution $x_- = A^T(AA^T)^{-1}b$. In some applications such as compressed sensing, the rows of A can be chosen to be orthogonal, making $x_- = A^Tb$ a more suitable and less expensive initial point. Second, Step 2 defines an outer loop which determines the number of values of μ in which the fixed point

Algorithm 1 Preconditioned Conjugate Gradient

Input Matrix A, vector b, current point $x\_$,
and the preconditioning diagonal matrix M.

Output Return x that solves $(A^TA + \lambda D_\mu^{-\frac{1}{2}}(x\_))x = A^Tb$

Parameters Tolerance ϵ, and maximum number of
iterations $cg\_maxiter$

Step 1. Initialization:
a. Set $x = 0$
b. Set $r_0 = A^Tb$, $r = r_0$, $r_p = Mr$
c. Set $d = r_p$, $\beta_n = r_p^T r$
Step 2. For $k = 1, ..., \ cg\_maxiter$
Step 3. Step length:
a. Set $q_1 = A^TAd$, $q_2 = \lambda D_\mu^{-\frac{1}{2}}(x\_)d$
b. Set $q = q_1 + q_2$
c. Compute: $\alpha_d = d^Tq$, $\alpha = \frac{\beta_n}{\alpha_d}$
Step 4. Update approximate solution:
$x = x + \alpha d$
Step 5. Update residual:
$r = r - \alpha q$
Step 6. Stopping criteria: If $\frac{\|r\|}{\|r_0\|} < \epsilon$ **end**
Step 7. Update preconditioned residual:
$r_p = Mr$
Step 8. Update direction:
a. Set $\beta_d = \beta_n$, $\beta_n = r_p^T r$, $\beta = \frac{\beta_n}{\beta_d}$
b. Set $d = r_p + \beta d$
Step 9. End-For

equation (3.6) is solved (Step 5). The values of μ, computed in Step 3, start with $\mu_1 = \frac{1}{2^{16}}$ and finish with $\mu_N = \frac{1}{2^{N+15}}$. Our numerical experimentation indicates that $N > 9$ suffices for convergence. The third stage of the FPLS_PCG algorithm is the inner loop in Steps 4 to 6. This inner loop corresponds with the fixed point iteration for a given value of μ. The algorithm remains in the inner loop whenever condition in Step 6 is satisfied, otherwise it exits from the inner loop, decreases the value of μ and performs the next outer loop iteration.

Convergence analysis

We show that solving a sequence of problems of the form (3.5) as $\mu \to 0$, is equivalent to solve problem (P_1^λ).

Theorem 1 *Any limit point of the sequence of minimizers* $\{x_{\mu_k}\}_{\mu_k \to 0}$ *of problem (3.5), is*

Algorithm 2 FPLS_PCG algorithm

Input Matrix A, and vector b.
Output Return $x = \arg\min \left\{ \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \right\}$
Parameters Penalization parameter λ, and maximum number of iterations N.

Step 1. Initialization:
Set $k = 1$ and $x\_$
Outer loop:
Step 2. While $k < N$
Step 3. Update $\mu = \frac{1}{2^{k+15}}$
Inner loop:
Step 4. Let $D_\mu(x\_) = \operatorname{diag}\left(\frac{1}{\sqrt{(x\_)_i^2 + \mu}}\right)$
Step 5. Solve for x : $\left(A^T A + \lambda D_\mu(x\_)\right) x = A^T b$
Step 6. if $\frac{\|x - x\_\|_2}{1 + \|x\_\|_2} > \sqrt{\mu}$
Let $x\_ = x$
go to Step 4 (Inner loop)
else
update $k = k + 1$
end-if
Step 7. end-While

contained in the optimal set of problem (P_1^λ).

Proof Choose an arbitrary sequence $\{\mu_k\}_k$ with $\mu_k \to 0$, and let

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1.$$

We start by pointing out that for each $\mu > 0$, the function $f_\mu(x)$ is strictly convex since

$$\nabla^2 f_\mu(x) = \operatorname{diag}\left(\frac{\lambda\mu}{(x_i^2 + \mu)^{3/2}}\right) + A^T A$$

is positive definite. Therefore for $\mu > 0$, the only stationary point x_μ is the unique global minimizer of $f_\mu(x)$.

Next we show that the family of global minimizers $\{x_\mu\}$ as $\mu \to 0$ is bounded. This is a

consequence of the bound

$$\begin{aligned}\lambda\sum_{i=1}^{n}|(x_\mu)_i| &\leq \lambda\sum_{i=1}^{n}\sqrt{(x_\mu)_i^2+\mu}\\ &\leq \frac{1}{2}\|Ax_\mu-b\|_2^2+\lambda\sum_{i=1}^{n}\sqrt{(x_\mu)_i^2+\mu}\\ &\leq \frac{1}{2}\|b\|_2^2+n\lambda\sqrt{\mu},\end{aligned}$$

where the last inequality follows because $f_\mu(x_\mu) \leq f_\mu(x)$ for all $x \in \mathbb{R}^n$, in particular for $x = 0$. Therefore,

$$\|x_\mu\|_1 \leq \frac{1}{2\lambda}\|b\|_2^2+n\sqrt{\mu}$$

and the family $\{x_\mu\}_{\mu\to 0}$ is bounded. Hence, there exists a convergent subsequence of $\{x_{\mu_k}\}_k$ which for simplicity will be denoted by $\{x_{\mu_k}\}_k$, and whose limit is written as

$$\bar{x} = \lim_{k\to\infty} x_{\mu_k}.$$

Now, since x_{μ_k} is the global minimizer of $f_{\mu_k}(x)$, then we have

$$\frac{1}{2}\|Ax_{\mu_k}-b\|_2^2+\lambda\sum_{i=1}^{n}\sqrt{(x_{\mu_k})_i^2+\mu_k} \leq \frac{1}{2}\|Ax^*-b\|_2^2+\lambda\sum_{i=1}^{n}\sqrt{(x^*)_i^2+\mu_k},$$

where $x^* = \arg\min f(x)$.

Taking the limit as $\mu_k \to 0$ in the above expression and by continuity yields

$$\frac{1}{2}\|A\bar{x}-b\|_2^2+\lambda\|\bar{x}\|_1 \leq \frac{1}{2}\|Ax^*-b\|_2^2+\lambda\|x^*\|_1.$$

Consequently, $f(\bar{x}) \leq f(x^*)$ and therefore $\bar{x}$ is a minimizer of $f(x)$. In other words, any limit point of the sequence of minimizers $\{x_{\mu_k}\}_{\mu_k\to 0}$ of problem (3.5), is contained in the optimal set of (P_1^λ). □

3.7 Discussion

This chapter presents a survey of the most popular methodologies used to solve our central problem (P_1^λ). Specifically, we describe the l1_ls, GPSR, FPC and NESTA algorithms which

have been referred in the literature as top solvers, and the PFSR algorithm introduced by Argáez et al. in [1]. Additionally, as part of our contribution, we present the FPLS_PCG algorithm as a new algorithm for ℓ_1 recovery. The next chapter validates numerically the proposed methods with encouraging results when compared with state-of-the-art algorithms.

Chapter 4

Numerical results

In this chapter, we present numerical results that support the methodology proposed for recovering sparse signals. Furthermore, image processing applications are exhibited demonstrating the viability of the sparse model in imaging sciences.

In our numerical tests, we include continuation steps for the FPC_AS, GPSR_BB and NESTA algorithms, since they rely on continuation strategies for accelerating convergence.

Experiments in Sections 4.1 and 4.2 involve a high computational burden, and are carried out in the Virgo machine at UTEP (two Intel Xeon 3.06 GHz processors with 4 GB of main memory) under Linux operating system. Image processing applications in Section 4.3 are carried out in a Dell Inspiron 1525 Intel Core 2 Duo at 2 GHz with 3 GB of main memory under Windows operating system.

4.1 Sparse recovery

In this first test, we illustrate the capability of the convex relaxation algorithms l1_ls, FPC_AS, GPSR_BB, PFSR, NESTA and FPLS_PCG, for recovering sparse signals. We consider a signal $x \in \mathbb{R}^{4096}$ with 80 spikes randomly located on the support of x, with amplitudes in the range $(-10, -2) \cup (2, 10)$, from $m = 1024$ noisy measurements. The matrix A consists of 1024 randomly chosen rows from the 4096×4096 Discrete Cosine Transform (DCT), and the noise is set according to a Gaussian distribution with zero mean and standard deviation of 0.01.

Figure 4.1 shows the recovered signals for each algorithm after one test. In each case, we observe that the approximate solution is sufficiently close to the original signal. Moreover,

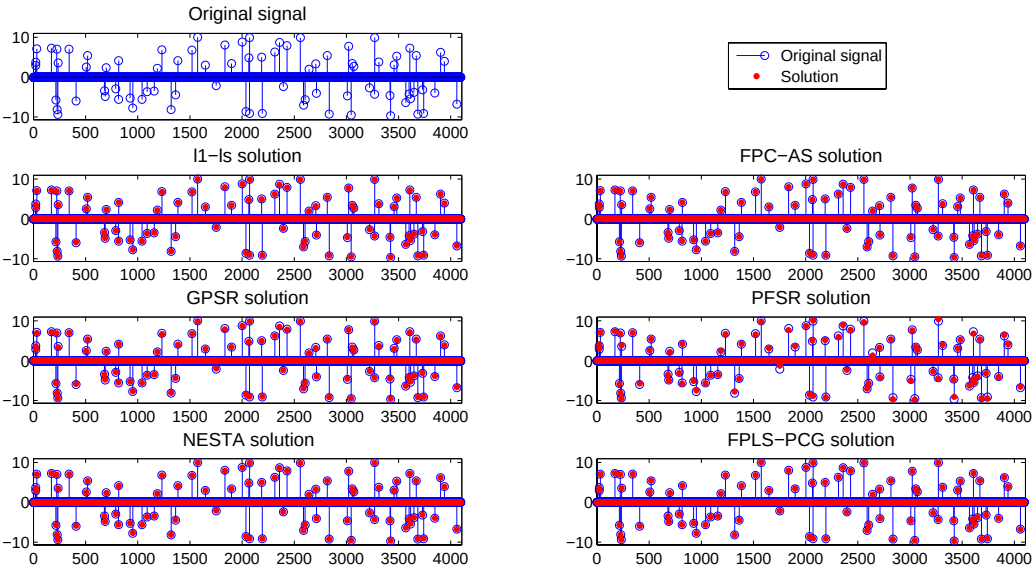

Figure 4.1: Sparse recovery example. The original signal is constructed by taking a zero vector $x \in \mathbb{R}^{4096}$ and randomly locating 80 spikes with amplitudes in the set $(-10,-2)\cup(2,10)$. All the ℓ_1 algorithms successfully recover the sparse signal.

this test is performed one hundred times with different random supports, recovering the signal in all the cases. The averaged results in terms of performance are summarized in Table 4.1 where the 2-norm error is defined as

$$\text{2-norm error} = \frac{\|x-\hat{x}\|_2}{\|x\|_2},$$

being x the true solution, and $\hat{x}$ the approximate solution obtained. In this experiment, the tolerance parameters of all the algorithms are adjusted so that a 2-norm error of approximately 1% is attained.

This first experiment is of dual aim. First, we want to illustrate the effectiveness of the convex relaxation algorithms for recovering sparse signals. In this respect, Table 4.1 gives evidence of a successful recovery in all the cases with comparable CPU times. Second, we aim to find a common point of reference for all the algorithms by adjusting the respective

Table 4.1: Average performance of sparse recovery algorithms after one hundred runs

Algorithm	2-norm error	CPU time [sec]
l1_ls	0.0101	1.1784
GPSR_BB (Continuation)	0.0114	0.5981
FPC_AS (Continuation)	0.0116	1.1134
NESTA (Continuation)	0.0093	0.6728
PFSR	0.0116	0.7830
FPLS_PCG	0.0098	0.7404

tolerance parameters. By doing this, we expect to achieve a fair experimental design for the next numerical test.

4.2 Scalability

In this experiment, we generate a sequence of synthetic problems to investigate how the problem size affects the performance of the algorithms when reconstructing sparse signals. We create a set of problems in the same fashion as in the previous section, with the difference that the non-zero components of x now take the values in the set $\{-1, +1\}$. The size n varies from $n = 2^{12}$ to $n = 2^{22}$. The total number of nonzero entries k is set as $k = \frac{5}{2^7}n$, and are randomly located forming the support of the signal x. The matrix A is constructed from m randomly chosen rows from the $n \times n$ DCT matrix with $m = \frac{n}{4}$. As before, the noise is set according to a Gaussian distribution with zero mean and standard deviation of 0.01. This experimentation follows a classical framework also considered in [16, 25], and [42].

Figures 4.2 and 4.3 show the results obtained in terms of the 2-norm error and CPU time respectively. From Figure 4.2, we observe that the 2-norm error scales preserving a nearly constant behavior in all the algorithms. In particular, our FPLS_PCG algorithm excels for reaching the best quality error. In Figure 4.3 we confirm that all the algorithms, except

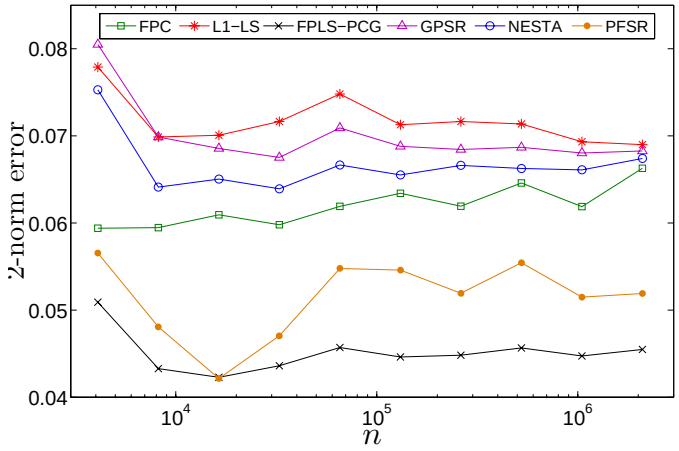

Figure 4.2: Scalability assessment for the 2-norm error.

perhaps the l1_ls algorithm that performs slower, maintain approximately the same order in terms of CPU time. Specifically, the GPSR algorithm preserves as the fastest until n = 16,384 where the FPLS_PCG algorithm surpasses it. The best times obtained in this test for large scale problems correspond with the algorithms FPLS_PCG, NESTA and PFSR.

4.3 Image processing applications

We now turn to analyze how the FPLS_PCG algorithm performs in real image processing problems. We consider four problems of major importance in the image processing community: the image denoising, the image deblurring, the image separation and the image inpainting problem.

In the image denoising and image deblurring problems, a Wiener filter algorithm is also considered as it is of standard reference for denoising and deblurring schemes.

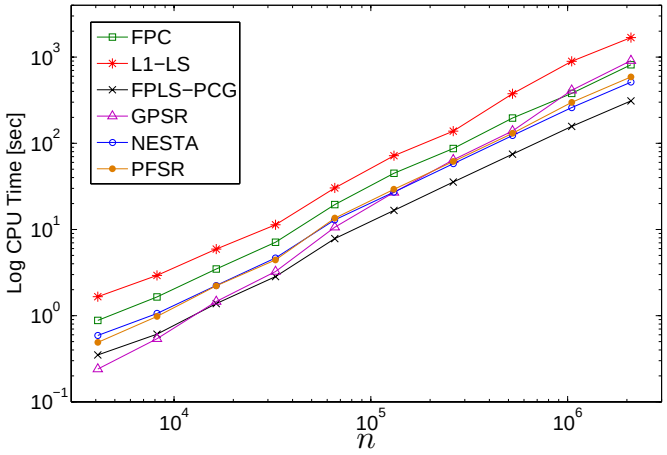

Figure 4.3: Scalability assessment for CPU time. (Time in seconds).

4.3.1 Image denoising

In the image denoising problem, we assume the original scene $u \in \mathbb{R}^n$ has been degraded by an additive noise ν, according to the model

$$b = u + \nu,$$

where b is the observed image. We consider the noise ν is unknown, spatial independent and follows a Gaussian distribution with zero mean and standard deviation σ. Therefore the energy of ν is upper bounded by $\|\nu\|_2 \leq \epsilon \approx \sigma\sqrt{n}$.

We assume the original image u has a sparse representation in the Ψ domain, that is, $u = \Psi x$. Therefore, the problem becomes in obtaining the image u^* that has the sparsest representation x^*, among all the images u that satisfy $\|u - b\|_2 \leq \epsilon$. This problem is posed as

$$\min_x \|x\|_1 \text{ subject to } \|\Psi x - b\|_2 \leq \epsilon. \tag{4.1}$$

Formulation (4.1) has the same form as (P_1^ϵ), and can be solved using ℓ_1 recovery algorithms. In particular, if the matrix Ψ is orthogonal, the FPLS_PCG algorithm greatly

benefits since the fixed point equation (3.6) becomes

$$\left(I + \lambda D_{\mu}^{-\frac{1}{2}}(x)\right) x = \Psi^T b,$$

that can be solved explicitly by

$$x_i = \frac{(\Psi^T b)_i}{1 + \frac{\lambda}{\sqrt{x_i^2+\mu}}},$$

for $i = 1, \cdots, n$, in just linear time.

A numerical example

We consider the standard 8-bit 512×512 test image *Lena*, that has been degraded by a Gaussian noise with zero mean and standard deviation of 10%. This problem is posed as formulation (4.1), where b is the degraded (noisy) image, Ψ is a wavelet Daubechies level 8 matrix, and $\epsilon \approx \sigma\sqrt{n} = 13,056$. Then we apply the FPLS_PCG algorithm for solving the image denoising problem. In this example, the degraded image has a 2-norm error of 0.0179. Table 4.2 shows the 2-norm error and CPU time achieved by the Wiener filter and the ℓ_1 based algorithms.

Table 4.2: PSNR and CPU time for the image denoising example

Algorithm	**2-norm error**	**CPU time [sec]**
Wiener Algorithm	0.0107	0.12
l1_ls	0.0128	14.30
GPSR_BB (Continuation)	0.0123	17.84
FPC_AS (Continuation)	0.0140	0.87
NESTA (Continuation)	0.0124	13.52
PFSR	0.0167	18.95
FPLS_PCG	0.0123	0.14

The results reported in this table reveal that all the algorithms decrease the 2-norm error with respect to the degraded image. In particular, the Wiener filter solution excels for

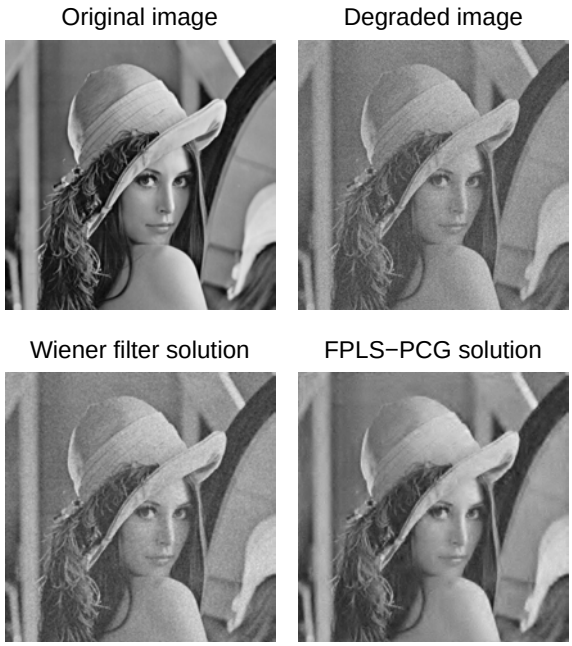

Figure 4.4: Image denoising example. (Top-left) Original image. (Top-right) The original image degraded by additive Gaussian noise of zero mean and standard deviation of 10%. (Bottom-left) Wiener filter solution. (Bottom-right) FPLS_PCG solution.

obtaining the best 2-norm error among all the algorithms, and our FPLS_PCG algorithm stands out for obtaining the best 2-norm error among the ℓ_1 based algorithms altogether with the GPSR_BB solver. In addition, as a second measure of quality, we present the restored images in Figures 4.4 and 4.5 for the Wiener filter and the FPLS_PCG algorithms. There is a clear distinction of the restored images in favor of our FPLS_PCG algorithm, even though the Wiener filter achieved a better 2-norm error. This is an example where a lower mean square error does not necessarily represent a better visual consistency. Finally, as we pointed out earlier, the formulation of the FPLS_PCG algorithm greatly benefits from the denoising model (4.1), making our algorithm the fastest among all the ℓ_1 based algorithms.

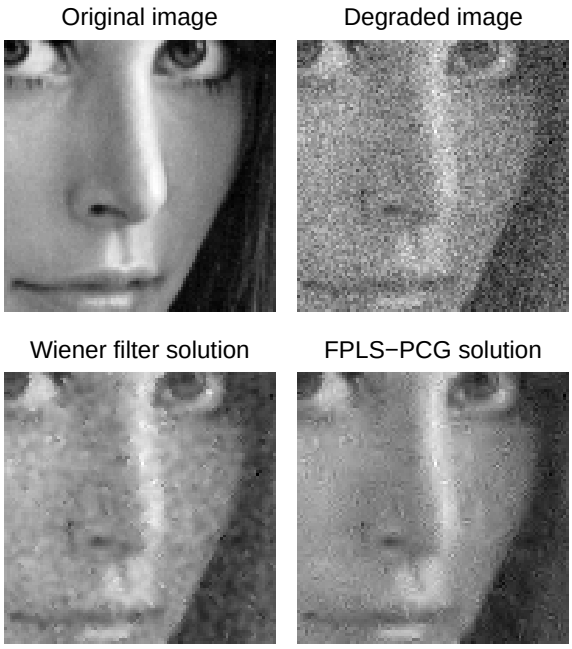

Figure 4.5: Image denoising example. Zoom of Figure 4.4.

4.3.2 Image deblurring

In the image deblurring problem, we assume the original scene $u \in \mathbb{R}^n$ has been degraded by a linear operator H followed by an additive noise ν. That is

$$b = Hu + \nu,$$

where b is the observed image. As before, the noise is unknown, spatial independent and follows a Gaussian distribution with zero mean and standard deviation σ. The linear operator $H \in \mathbb{R}^{n \times n}$ is generally assumed to be poorly conditioned [22].

Following the same idea as in the previous section, we aim to obtain an approximation u^* to the original scene, by imposing the sparsity of $x = \Psi^{-1}u$ as a prior. Therefore, we formulate the following optimization problem

$$\min_x \|x\|_1 \text{ subject to } \|H\Psi x - b\|_2 \leq \epsilon, \tag{4.2}$$

that once again can be solved using an ℓ_1 recovery algorithm.

A numerical example

We solve the image deblurring problem also considered in [16]. The data set for this problem is available in the sparco library [39] as Problem 701 [1]. The blur operator H consists of a 15×15 kernel with values being $1/(i^2 + j^2 + 1)$ for $-7 \leq i, j \leq 7$, normalized to have a unit sum. The additive noise ν is white zero-mean Gaussian noise with standard deviation 1%, where the image pixels are in the range [0,255]. The original scene is the standard 256×256 *cameraman* test image. As before, we apply the Wiener filter and the FPLS_PCG algorithms to restore the degraded image. In this example, the vector b is the degraded (blurred and noisy) image, Ψ is a wavelet Haar level 8 matrix, and $\epsilon = \sigma\sqrt{n} = 653$. The corresponding 2-norm error for the degraded image is 0.0371. After applying the Wiener filter and the FPLS_PCG algorithms, the restored images reached a 2-norm error of 0.0171 and 0.0159 respectively, improving in both cases the image quality. The CPU time reported in this experiment is about 1 second for the Wiener filter, and 8 seconds for the FPLS_PCG. The deblurred images are presented in Figure 4.6, exhibiting a satisfactory solution for both algorithms.

4.3.3 Image separation

In the image separation problem, we assume the observed image is the superposition of two or more images of different nature. In the case where the image is composed of two superposed images, we have the following model

$$b = u_1 + u_2 + \nu, \tag{4.3}$$

where b is the observed image, and ν is an additive noise vector of upper bounded energy. In this model, the images u_1 and u_2 are assumed to be morphologically distinct. We further assume that there exist two sparsifying transformations Ψ_1 and Ψ_2 where u_1 and u_2 have a

[1] `http://www.cs.ubc.ca/labs/scl/sparco/index.php/`

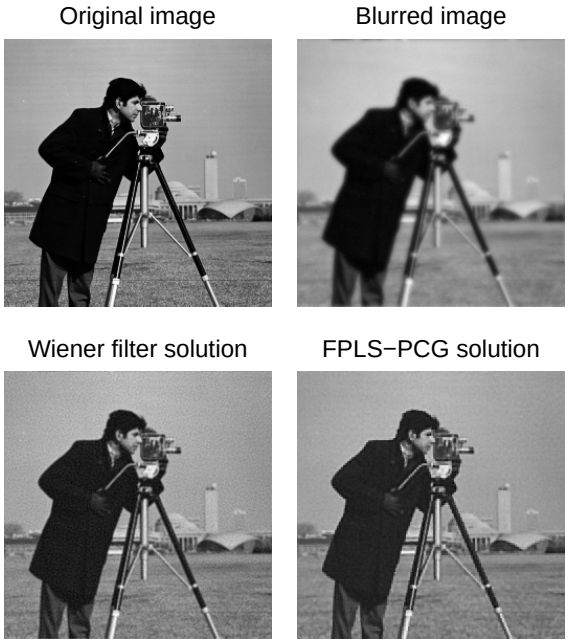

Figure 4.6: Image deblurring example. (Top-left) Original image. (Top-right) The original image degraded by a blur operator consisting in a kernel with values being $1/(i^2 + j^2 + 1)$ for $-7 \leq i, j \leq 7$, normalized to have a unit sum, and additive Gaussian noise with zero mean and standard deviation of 1%. (Bottom-left) Wiener filter solution. (Bottom-right) FPLS_PCG solution.

sparse representation x_1 and x_2, respectively. Consequently, model (4.3) becomes

$$b = \Psi_1 x_1 + \Psi_2 x_2 + \nu.$$

In order to find the images u_1 and u_2, we find the sparsest vector $[x_1 \; x_2]^T$ that satisfies the model (4.3). That is,

$$\min_{(x_1,x_2)} \|x_1\|_1 + \|x_2\|_1 \quad \text{subject to} \quad \begin{bmatrix} \Psi_1 & \Psi_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \nu = b.$$

The equation above has the form (P_1^ν) and can be solved using an ℓ_1 recovery algorithm.

A numerical example

We consider an 8-bit 512×512 image that has been synthetically constructed as the superposition of two images u_1 and u_2. In this example, u_1 is the natural image *Lena* and u_2 is a texture image artificially generated. A Gaussian noise ν of zero mean and standard deviation of 1% is also added. Because natural images have a sparse representation in the wavelet domain, we choose Ψ_1 to be a wavelet Daubechies transformation. On the other hand, because texture images are sparsely represented with discrete cosine coefficients, we choose Ψ_2 to be the DCT transform. The 2-norm error for the degraded image b, considering u_1 as the target scene and u_2 as a corruptive image, is 0.5304.

The FPLS_PCG algorithm is applied to recover the sparse vector $[x_1\ x_2]^T$, and the original images are computed as $u_1 = \Psi_1 x_1$ and $u_2 = \Psi_2 x_2$ respectively. From Figure 4.7 we observe that a successful image separation can be attained following the sparse representation model and morphological diversity. In this experiment, the CPU time required for the separation process was about a minute, obtaining a 2-norm error of 0.1384.

4.3.4 Image Inpainting

In the image inpainting problem, the observed data $b \in \mathbb{R}^m$ is the resulting image after an original scene $u \in \mathbb{R}^n$ is subjected to a loss data process. The degradation process associated with the missing information can be modeled as an underdetermined linear system

$$b = Hu + \nu,$$

where $H \in \mathbb{R}^{m \times n}$ is constructed by removing from the $n \times n$ identity matrix, the $n - m$ rows associated with the missing data; and ν is additive Gaussian noise of bounded energy.

Unlike the image deblurring problem where H is square and poorly conditioned, here H is associated with an underdetermined system which is consistent. In order to recover the original image u, we proceed in a manner reminiscent of compressed sensing. That is, from incomplete measurements we aim to recover an approximation u^* of the original object by

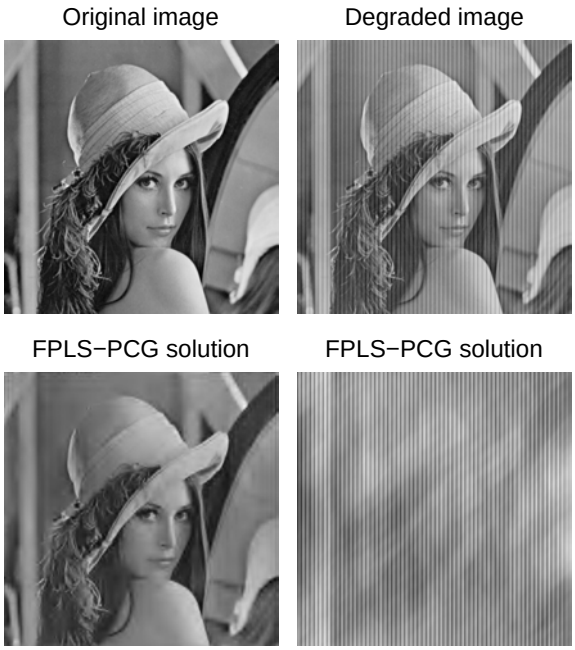

Figure 4.7: Image separation example. (Top-left) An original natural image. (Top-right) The original natural image degraded by superposing a texture image of similar power. (Bottom) The recovered natural image after a separation via sparse representation, and the texture extracted from the original image.

imposing sparsity of $x = \Psi^{-1}u$ as a prior. Therefore, we formulate the following optimization problem

$$\min_x \frac{1}{2}\|H\Psi x - b\|_2^2 + \lambda\|x\|_1, \tag{4.4}$$

that can be solved using an ℓ_1 recovery algorithm.

A numerical example

We consider the standard 8-bit 256×256 test image *cameraman*, that has been degraded with several scratches as shown in Figure 4.8 (left), and Gaussian noise with zero mean and standard deviation of 1%. We aim to recover the original image by solving the optimization problem (4.4), assuming that the degraded pixels correspond with missing information. In

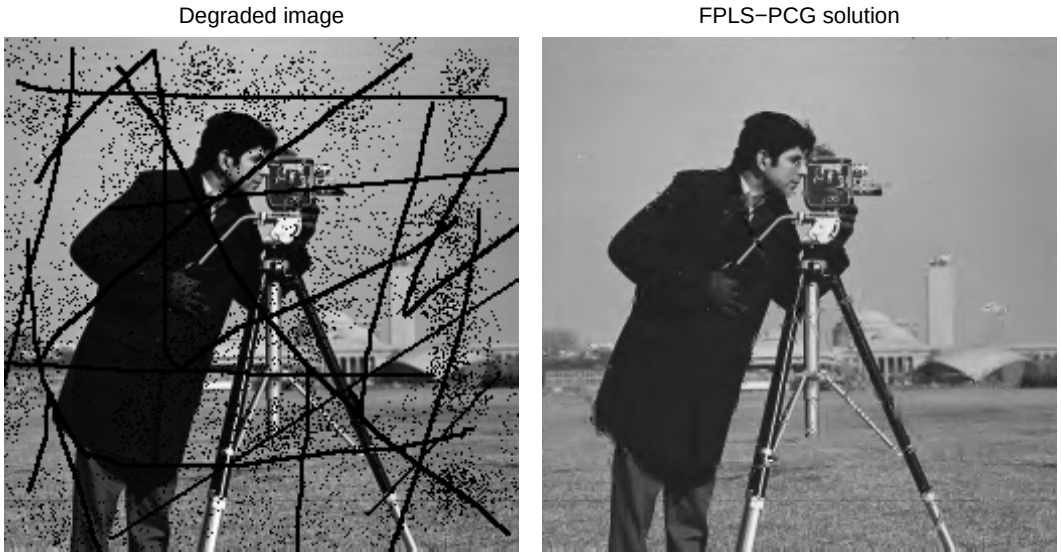

Figure 4.8: Image inpainting example. (left) A degraded image. (right) The recovered (inpainted) image via sparse representation. A 2-norm error of 0.0192 is obtained after a 2-norm error of 0.2220 of the degraded image.

this example, b is the observed and degraded image, Ψ is the wavelet Daubechies transformation, λ is fixed to 0.01, and H is constructed as discussed above. The percentage of degraded (or missing) pixels is 18.71% corresponding to a 2-norm error of 0.2220.

The FPLS_PCG algorithm is applied for solving (4.4), and the results are presented in Figure 4.8 (right). A 2-norm error of 0.0192 is obtained after reconstruction, indicating a successful inpainting process under the scheme of sparse representation. The CPU time invested in the inpainting process was about 22 seconds.

4.4 Discussion

This chapter provides numerical evidence that supports the methodologies proposed in Chapter 3. First, we demonstrate the capabilities of the PFSR and FPLS_PCG algorithms for recovering sparse signals with a performance comparable to state-of-the-art solvers. Indeed,

based on these preliminary results, the FPLS_PCG algorithm stands out for achieving a remarkably better 2-norm error without compromising computational efficiency. Second, we successfully apply the sparse representation model for solving image processing problems, specifically for image denoising, image deblurring, image separation, and image inpainting.

The numerical results presented in this chapter motivate us to extend the proposed methodology to other image processing problems, particularly those related to image segmentation.

Chapter 5

Concluding remarks and future work

In this thesis, we have concentrated on two fundamental research directions. First, how to efficiently solve optimization problems involved in the sparse representation model, and second, how to apply the sparse representation model to solve image processing problems.

In the first research direction, we have presented state-of-the-art algorithms for solving the ℓ_1 recovery problem. Moreover, we have shown a competitive performance of the PFSR and FPLS_PCG algorithms for recovering sparse signals compared with the l1_ls, GPSR, FPC and NESTA algorithms referred as top solvers in the literature.

In the second research direction, we have shown preliminary results that demonstrate the viability of the sparse representation model for solving image processing problems. Specifically, we have solved the image denoising, image deblurring, image separation and image inpainting problems, showing competitive results with respect to classical approaches such as Wiener filtering.

In the future, our efforts will be focused on the following aspects:

Numerically The FPLS_PCG algorithm can be improved by implementing an adaptive strategy for choosing the penalization parameter λ. In addition, a High Performance Computer (HPC) version of the FPLS_PCG algorithm can be implemented for very large scale problems.

Applications The image segmentation problem is of our special attention, since little has been investigated about the possibility of applying the sparse representation model to segmentation schemes. We believe that incorporating sparsity as a prior in existing segmentation techniques will lead us to interesting results. In particular, we will fo-

cus on applications involving seismic images, where the main objective is to identify structures associated to presence of hydrocarbons in time-migrated data.

On the other hand, there is a potential application of image inpainting in the area of gap-filling for micrometeorological analysis, where rejected or lost data needs to be estimated for a better decision making process.

References

[1] M. Argáez, C. Ramirez, R. Sanchez, "An ℓ_1 algorithm for underdetermined systems and applications", IEEE proceedings of the North American Fuzzy Information Processing Society annual meeting 2011, 2011.

[2] S. Becker, J. Bobin, and E. J. Candès. "NESTA: a fast and accurate first-order method for sparse recovery". SIAM J. on Imaging Sciences, vol. 4, No. 1, pp. 1-39. 2010.

[3] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge, U.K.: Cambridge University Press, 2004.

[4] E. Candès, "Compressive sampling", *Proceedings of the International Congress of Mathematicians*, Madrid, Spain, 2006.

[5] E. Candès, J. Romberg and T. Tao, "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information", *IEEE Transactions on Information Theory*, Vol. 52, pp. 489-509, 2006.

[6] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization", *IEEE Signal Processing Letters*, Vol. 14, pp. 707-710, 2007.

[7] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing", *in IEEE International Conference on Acoustics Speech, and Signal Processing*, pp. 3869-3872, 2008.

[8] S. Chen, D. Donoho, M. Saunders, "Atomic Decomposition by Basis Pursuit", *SIAM Review*, Vol. 43 No 1, pp. 129-159, 2001.

[9] D. Donoho, X. Huo, "Uncertainty principles and ideal atomic decomposition", *IEEE Transactions on Information Theory* Vol 47, pp. 2845-2862, 2001.

[10] D. Donoho, "Compressed sensing", *IEEE Transactions on Information Theory*, Vol. 52, No. 4, pp. 1289-1306, 2006.

[11] D. Donoho, "For most large underdetermined systems of linear equations, the minimal ℓ_1 solution is also the sparsest solution", *Communication on pure and applied Mathematics*, Vol. 59, No. 7, pp. 907934, 2006.

[12] D. Donoho, M. Vetterli, R. DeVore, I. Daubechies, "Data compression and harmonic analysis", *IEEE Transaction on Information Theory* Vol. 44, pp. 2435-2476, 1998.

[13] D. Donoho, M. E. Raimondo, "A fast wavelet algorithm for image deblurring", *Australian New Zealand Ind. Appl. Math. J.*, Vol. 46, pp.C29-C46, 2005.

[14] M. Elad, Sparse and redundant representations, Springer 2010.

[15] M. Elad, M. Figueiredo, Ma Yi, "On the Role of Sparse and Redundant Representations in Image Processing", *Proceedings of the IEEE*, Vol 98, No. 6, pp. 972-982, 2010.

[16] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems", *IEEE Selected topics in signal processing*, Vol. 1, No. 4, pp. 586-597, 2007.

[17] M. Figueiredo, R. Nowak, "An EM algorithm for wavelet-based image restoration", *IEEE Transactions on image processing*, Vol. 12, No. 8, pp. 906-916, 2003.

[18] D. Fong, M. Saunders, "LSMR: An iterative algorithm for sparse least-squares problems", Submitted to SIAM Journal in Scientific Computing. 2010.

[19] R. C. Gonzalez, R. E. Woods, Digital Image Processing, Prentice Hall, 2008.

[20] P. Guillen, F. Martinez-de-Pinson, R. Sanchez, M. Argáez, L. Velázquez, "Characterization of Subcortical Structures during Deep Brain Stimulation utilizing Support Vector Machines", *IEEE 33rd Annual International Conference Proceedings of the Engineering in Medicine and Biology Society*, pp 7949 - 7952. August 2011.

[21] E. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for ℓ_1-regularized minimization with applications to compresses sensing", Technical Report TR07-07, Department of Computational and Applied Mathematics, Rice University, Houston, TX, USA, 2007.

[22] P. C. Hansen, J. G. Nagy, D. P. O'Leary, "Deblurring Images: matrices, spectra and filtering", Fundam. Algorithms 3, SIAM, Philadelphia, 2006.

[23] X. Huo, "Sparse image representation via combined transform", PhD Thesis, Stanford University, 1999.

[24] S. Jokar, M. Pfetsch, "Exact and approximate sparse solutions of underdetermined linear equations", *SIAM Journal on Scientific Computing*, Vol. 31, No. 1, pp. 23-44, 2008.

[25] S. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinvesky, "An interior-point method for large-scale ℓ_1-regularized least squares", *IEEE Selected topics in signal processing*, Vol. 1, No. 4, pp. 606-617, 2007.

[26] S. Mallat, "Theory of multiresolution signal decomposition: The wavelet representation", *IEEE Transaction on pattern analysis and machine intelligence.*, Vol. 11, pp. 674693, 1989.

[27] S. Mallat, A Wavelet Tour of Signal Processing, San Diego, CA: Academic, 1998.

[28] C. Miosso, R. Von-Borries, M. Argáez, L. Velázquez, C. Quintero, C. Potes, "Compressed sensing reconstruction with prior information using penalized reweighted normal equations", *IEEE Transactions on Signal Processing*, Vol. 52, No. 4, pp. 1289-1306, 2009.

[29] M. Motwani, M. Gadiya, R. Motwani, and F. C. Harris, Jr. "A Survey of Image Denoising Techniques", *in Proceedings of GSPx 2004*, Santa Clara Convention Center, Santa Clara, CA, 2004.

[30] P. Moulin and J. Liu, "Analysis of multiresolution image denoising schemes using generalized-Gaussian and complexity priors", *IEEE Transactions on information theory*, Vol. 45, pp. 909919, 1999.

[31] Y. Nesterov, "Smooth minimization of non-smooth functions", Mathematical programming, vol. 103, No. 1, pp. 127-152. 2005.

[32] F. A. Potra and S. J. Wright, "Interior-point methods", *Journal of computational and applied mathematics*, Vol. 124, pp. 281-302, 2000.

[33] R. T Rockafellar, Convex Analysis, Princeton. NJ : Princeton Univ. Press; 1970.

[34] R. Sanchez, M. Argaez, P. Guillen, "Sparse Representation via l1-minimization for Underdetermined Systems in Classification of Tumors with Gene Expression Data", *IEEE 33rd Annual International Conference Proceedings of the Engineering in Medicine and Biology Society*, pp 3362-3366. August 2011.

[35] B. K. Natarajan, "Sparse approximate solutions to linear systems", *SIAM Journal on computing*, Vol. 24, pp. 227-234, 1995.

[36] H. L. Taylor, S. C. Banks, J. F. McCoy. "Deconvolution with the ℓ_1 norm", *Geophysics*, Vol. 44, pp. 39-52, 1979.

[37] J. Tropp, "Greed is good: Algorithmic results for sparse approximation", *IEEE Transactions on Information Theory*, Vol. 50, No. 10, pp. 22312242, 2004.

[38] J. Tropp, S. Wright, "Computational Methods for Sparse Solution of Linear Inverse Problems", *Proceedings of the IEEE*, Vol.98, No.6, pp.948-958, 2010.

[39] E. van den Berg, M. P. Friedlander, G. Hennenfent, F. J. Herrmann, R. Saab, O. Yilmaz, "Sparco: A Testing Framework for Sparse Reconstruction", Technical Report TR-2007-20, University of British Columbia, Vancouver, 2007.

[40] W. Yin, Y. Zhang, "Extracting salient features from less data via ℓ_1-minimization", *SIAG/Optimization Views and News* Vol. 9, 2008.

[41] S. Wright, Primal-dual Interior-point Methods, 1997, SIAM, Philadelphia.

[42] S. Wright, R. Nowak, M. Figueiredo, "Sparse reconstruction by separable approximation", *in IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008*, Las Vegas, NV, 2008.

[43] M. Zibulevsky, M. Elad, "L1-L2 Optimization in Signal and Image Processing", *IEEE Signal Processing Magazine* Vol. 27 No. 3, 2010.

Appendix A

Equivalences

We start by showing that problem (P_1^ν) is equivalent to (P_1^ϵ). Then we show that problems (P_1^ϵ) and (P_1^λ) are equivalent for an appropriate choice of ϵ and λ. To that end, we characterize the optimal points of both problems using sub-differential calculus and the KKT conditions.

A.0.1 Equivalence between (P_1^ν) and (P_1^ϵ)

Suppose that x^* solves problem (P_1^ν). Then we have that $-\nu = Ax^* - b$ and therefore $\|Ax^* - b\| = \|\nu\|$. Now, since ν is bounded by ϵ, it follows that $\|Ax^* - b\| = \|\nu\| \leq \epsilon$ and x^* also solves (P_1^ϵ).

Conversely, if x^* solves (P_1^ϵ), we have that $\|Ax^* - b\| \leq \epsilon$. Choosing $-\nu = Ax^* - b$, we easily verify that x^* also solves (P_1^ν).

A.0.2 Optimality conditions for problem (P_1^ϵ)

Problem (P_1^ϵ) is equivalent to the following constrained optimization problem

$$\begin{aligned} \min_x \ & \|x\|_1 && (P_1^{\epsilon r}) \\ \text{s.t. } & Ax - b = r \\ & \|r\|_2^2 \leq \epsilon^2. \end{aligned}$$

The Lagrangian associated with $(P_1^{\epsilon r})$ is

$$L(x, r, y, z) = \|x\|_1 - (r + b - Ax)^T y - (\epsilon^2 - \|r\|_2^2)z,$$

where $(x, r) \in \mathbb{R}^{n+m}$ are the primal variables, and $y \in \mathbb{R}^m$ and $z \in \mathbb{R}$ are the Lagrange multipliers associated with the equality and inequality constraints respectively.

For the sake of completeness, we recall that for any real valued function u, $\partial_x u$ stands for the subdifferential of u with respect to x. Accordingly, the KKT conditions for $(P_1^{\epsilon r})$ are

(i)	$-A^T y \in \partial_x \|x\|_1$	Since $0 \in \partial_x L(x, r, y, z)$
(ii)	$y = 2zr$	Since $0 \in \partial_r L(x, r, y, z)$
(iii)	$Ax - b = r$	Equality constraint
(iv)	$\epsilon^2 - \|r\|_2^2 \geq 0$	Inequality constraint
(v)	$(\epsilon^2 - \|r\|_2^2)z = 0$	Complementarity condition
(vi)	$z \geq 0$	Non negativity of Lagrange multiplier associated with the inequality constraint.

A.0.3 Optimality conditions for problem (P_1^{λ})

Problem (P_1^{λ}) is equivalent to the following constrained optimization problem

$$\min_x \frac{1}{2}\|e\|_2^2 + \lambda\|x\|_1 \qquad (P_1^{\lambda e})$$
$$\text{s.t. } e = Ax - b.$$

The Lagrangian associated with $(P_1^{\lambda e})$ is

$$L(x, e, \bar{y}) = \frac{1}{2}\|e\|_2^2 + \lambda\|x\|_1 - (e - Ax + b)^T \bar{y},$$

where $(x, e) \in \mathbb{R}^{n+m}$ are the primal variables and $\bar{y} \in \mathbb{R}^m$ is the Lagrange multiplier associated with the equality constraint. Then the KKT conditions are

(i)	$-\frac{1}{\lambda}A^T \bar{y} \in \partial_x \|x\|_1$	Since $0 \in \partial_x L(x, e, \bar{y})$
(ii)	$e = \bar{y}$	Since $0 \in \partial_e L(x, e, \bar{y})$
(iii)	$e = Ax - b.$	Equality constraint

A.0.4 Equivalence between (P_1^{ϵ}) and (P_1^{λ})

We show that an optimal solution of $(P_1^{\epsilon r})$ also satisfies the KKT conditions of $(P_1^{\lambda e})$. This point will also be an optimal solution of $(P_1^{\lambda e})$, since $(P_1^{\lambda e})$ is a convex problem and the KKT

conditions are sufficient. The same argument is used to show that an optimal solution of $(P_1^{\lambda e})$ is also an optimal solution of $(P_1^{\epsilon r})$.

We start by showing that the set of optimal points of $(P_1^{\epsilon r})$ is contained in the set of optimal points of $(P_1^{\lambda e})$. Let (x^*, r^*) be an optimal point of $(P_1^{\epsilon r})$ with the Lagrange multipliers (y^*, z^*). First, we consider the case when $z^* > 0$.

From KKT condition (i) for $(P_1^{\epsilon r})$ it follows that

$$-A^T y^* \in \partial_x \|x^*\|_1.$$

Set $\lambda = \frac{1}{2z^*}$ and choose the vector $\bar{y} = \lambda y^*$. Then we have

$$\boxed{\text{(a)} \quad -\tfrac{1}{\lambda} A^T \bar{y} \in \partial_x \|x^*\|_1.}$$

From KKT condition (ii) for $(P_1^{\epsilon r})$ it follows that

$$y^* = 2z^* r^*.$$

Choosing $e = r^*$ in conjuction with λ and $\bar{y}$ we have

$$\boxed{\text{(b)} \quad \bar{y} = e.}$$

From KKT condition (iii) for $(P_1^{\epsilon r})$ it follows that that

$$r^* = Ax^* - b.$$

Now since $e = r^*$, we have

$$\boxed{\text{(c)} \quad e = Ax^* - b.}$$

From (a), (b), (c) and the convexity of $(P_1^{\lambda e})$, we conclude that x^* also solves problem $(P_1^{\lambda e})$.

If $z^* = 0$, then from KKT condition (ii) for $(P_1^{\epsilon r})$, y^* is zero and KKT condition (i) for $(P_1^{\epsilon r})$ turns into

$$0 \in \partial_x \|x^*\|_1.$$

This implies that $x^* = \arg\min \{\|x\|_1\}$, that is, $x^* = 0$. Thus, it is sufficient to show that there exists a choice for λ such that $x^* = 0$ solves problem $(P_1^{\lambda e})$. This is accomplished by choosing $\bar{y} = e = -b$ and $\lambda \geq \|A^T b\|_\infty$, since for this choice the KKT conditions (i) to (iii) for $(P_1^{\lambda e})$ are satisfied. To see this, recall that

$$\partial_x |x| = \begin{cases} 1 & x > 0 \\ [-1, 1] & x = 0 \\ -1 & x < 0, \end{cases}$$

and that $\frac{(A^T b)_i}{\lambda} \in [-1, 1]$ for all $i = 1, \cdots, n$, provided $\lambda \geq \|A^T b\|_\infty$.

Based on previous discussion, we conclude that the only case of practical interest (without involving the trivial solution) is when $z^* > 0$. As a result, and from KKT condition (v) for $(P_1^{\epsilon r})$, it follows that the best residual one can obtain at the solution is the level of noise, that is $\|r\|_2 = \epsilon$.

We now show that the set of optimal points of $(P_1^{\lambda e})$ is contained in the set of optimal points of $(P_1^{\epsilon r})$. Let (x^*, e^*) be an optimal point of $(P_1^{\lambda e})$ with the Lagrange multiplier $\bar{y}^*$.

From KKT condition (i) for $(P_1^{\lambda e})$ it follows that

$$-\frac{1}{\lambda} A^T \bar{y}^* \in \partial_x \|x^*\|_1.$$

Choose the vector $y = \frac{1}{\lambda}\bar{y}^*$. Then we have

$$\boxed{\text{(d)} \quad -A^T y \in \partial_x \|x^*\|_1.}$$

From KKT condition (ii) for $(P_1^{\lambda e})$ we have $\bar{y}^* = e^*$. Choosing $z = \frac{1}{2\lambda}$ and $r = e^*$, it follows

$$\boxed{\text{(e)} \quad y = 2zr.}$$

Moreover,

$$\begin{aligned} y - 2zr &= 0 \\ r^T y - (2z) r^T r &= 0 \\ \lambda r^T y - \|r\|_2^2 &= 0. \end{aligned}$$

The choice $\epsilon^2 = \lambda r^T y$ yields

$$\boxed{\text{(f)} \quad \epsilon^2 - \|r\|_2^2 = 0.}$$

Finally, from KKT condition (iii) for $(P_1^{\lambda e})$ we have $e^* = Ax^* - b$. Hence,

$$\boxed{\text{(g)} \quad r = Ax^* - b.}$$

From (d), (e), (f), (g) and the convexity of $(P_1^{\epsilon r})$, we conclude that x^* also solves $(P_1^{\epsilon r})$.

Curriculum Vita

Carlos Ramirez was born in Cali, Colombia. After graduating from the Lacordaire high school in 1998, he had the honor to serve his country for one year in the Colombian army, as demanded by law. He entered the Javeriana-Cali University in the Fall of 1999 to study Electronic Engineering, where early was awarded with undergraduate Teaching Assistantships in the areas of Fundamental Mathematics, Calculus, Differential Equations, Discrete Mathematics, Signals and Systems, and Electromagnetic Theory due to his high grades and academic achievements. During his undergraduate studies, he joined the mathematical club "Ramanujan" and actively participated in Mathematical Olympiads across his Country, obtaining an honor mention in the seventh Colombian Mathematics Olympiads, 2003. In the same year, he was awarded a honor distinction given by the Educational Department of Colombia for obtaining one of the best ten scores in the nation-wide undergraduate students standardized official tests, ECAES 2003. In the Fall 2004, he graduated with the highest honor (Cum Laude) for academic and human excellence.

After receiving his degree in Engineering, Carlos worked for four years in Carvajal S.A, Bogotá, as Coordinator of Technology. During this time, he coordinated the national operation for the document and printing solution business in the company, gaining important experience in the industry. In 2008, he decided to continue with his studies, particularly in the area of mathematics.

In Fall 2008, Carlos started his graduate studies at The University of Texas at El Paso, UTEP, in the Computational Science Program.

Permanent address: 3500 Sun Bowl Dr No 86.

El Paso, Texas 79902