Compressive Sensing Using ℓ_p Optimization

by

Jeevan Kumar Pant
B.E., Tribhuvan University, 1999
M.Sc.Eng., Tribhuvan University, 2003

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

Compressive Sensing Using ℓ_p Optimization

by

Jeevan Kumar Pant
B.E., Tribhuvan University, 1999
M.Sc.Eng., Tribhuvan University, 2003

Supervisory Committee

Dr. Andreas Antoniou, Co-Supervisor
(Department of Electrical and Computer Engineering)

Dr. Wu-Sheng Lu, Co-Supervisor
(Department of Electrical and Computer Engineering)

Dr. Dale D. Olesky, Outside Member
(Department of Computer Science)

Supervisory Committee

Dr. Andreas Antoniou, Co-Supervisor
(Department of Electrical and Computer Engineering)

Dr. Wu-Sheng Lu, Co-Supervisor
(Department of Electrical and Computer Engineering)

Dr. Dale D. Olesky, Outside Member
(Department of Computer Science)

ABSTRACT

Three problems in compressive sensing, namely, recovery of sparse signals from noise-free measurements, recovery of sparse signals from noisy measurements, and recovery of so called *block-sparse* signals from noisy measurements, are investigated.

In Chapter 2, the reconstruction of sparse signals from noise-free measurements is investigated and three algorithms are developed. The first and second algorithms minimize the approximate ℓ_0 and ℓ_p pseudonorms, respectively, in the null space of the measurement matrix using a sequential quasi-Newton algorithm. An efficient line search based on Banach's fixed-point theorem is developed and applied in the second algorithm. The third algorithm minimizes the approximate ℓ_p pseudonorm in the null space by using a sequential conjugate-gradient (CG) algorithm. Simulation results are presented which demonstrate that the proposed algorithms yield improved signal reconstruction performance relative to that of the iterative reweighted (IR), smoothed ℓ_0 (SL0), and ℓ_1-minimization based algorithms. They also require a reduced amount of computations relative to the IR and ℓ_1-minimization based algorithms. The ℓ_p-minimization based algorithms require less computation than the SL0 algorithm.

In Chapter 3, the reconstruction of sparse signals and images from noisy measurements is investigated. First, two algorithms for the reconstruction of signals are

developed by minimizing an ℓ_p-pseudonorm regularized squared error as the objective function using the sequential optimization procedure developed in Chapter 2. The first algorithm minimizes the objective function by taking steps along descent directions that are computed in the null space of the measurement matrix and its complement space. The second algorithm minimizes the objective function in the time domain by using a CG algorithm. Second, the well known total variation (TV) norm has been extended to a nonconvex version called the TV_p pseudonorm and an algorithm for the reconstruction of images is developed that involves minimizing a TV_p-pseudonorm regularized squared error using a sequential Fletcher-Reeves' CG algorithm. Simulation results are presented which demonstrate that the first two algorithms yield improved signal reconstruction performance relative to the IR, SL0, and ℓ_1-minimization based algorithms and require a reduced amount of computation relative to the IR and ℓ_1-minimization based algorithms. The TV_p-minimization based algorithm yields improved image reconstruction performance and a reduced amount of computation relative to Romberg's algorithm.

In Chapter 4, the reconstruction of so-called *block-sparse* signals is investigated. The $\ell_{2/1}$ norm is extended to a nonconvex version, called the $\ell_{2/p}$ pseudonorm, and an algorithm based on the minimization of an $\ell_{2/p}$-pseudonorm regularized squared error is developed. The minimization is carried out using a sequential Fletcher-Reeves' CG algorithm and the line search described in Chapter 2. A reweighting technique for the reduction of amount of computation and a method to use prior information about the locations of nonzero blocks for the improvement in signal reconstruction performance are also proposed. Simulation results are presented which demonstrate that the proposed algorithm yields improved reconstruction performance and requires a reduced amount of computation relative to the $\ell_{2/1}$-minimization based, block orthogonal matching pursuit, IR, and ℓ_1-minimization based algorithms.

Contents

List of Tables

List of Figures

List of Abbreviations

BFGS	Broyden-Fletcher-Godfarb-Shanno
BOMP	block orthogonal matching pursuit
BP	basis pursuit
BPDN	basis pursuit for denoising
CG	conjugate gradient
CS	compressive sensing
DCT	discrete-cosine transform
i.i.d.	independent identically distributed
IR	iterative reweighted
$\ell_{2/1}$-SOCP	$\ell_{2/1}$ second-order cone programming
$\ell_{2/p}$-RLS	$\ell_{2/p}$-regularized least-squares
$\ell_{2/p}$-RLS-WT	$\ell_{2/p}$-regularized least squares with weighting
ℓ_p-RLS	ℓ_p-regularized least-squares
ℓ_p-RLS-CG	ℓ_p-regularized least-squares conjugate gradient
NRAL0	nullspace reweighted approximate ℓ_0
PSNR	peak-signal-to-noise ratio
RIP	restricted isometry property
ROU	range of uncertainty

SL0	smoothed ℓ_0
SNR	signal-to-noise ratio
SOCP	second-order cone programming
SVD	singular-value decomposition
TV_p-RLS-CG	TV_p-regularized least-squares conjugate gradient
TV	total variation
UALP	unconstrained approximate ℓ_p
UALP-CG	unconstrained approximate ℓ_p conjugate gradient

ACKNOWLEDGEMENTS

I feel very fortunate to have had access to wonderful individuals who have provided their generous help and support during my doctoral studies.

I thank my supervisors Dr. Andreas Antoniou and Dr. Wu-Sheng Lu for their guidance, inspiration, and incredible support, for teaching me invaluable concepts from digital signal processing to optimization techniques, and for helping me to improve my technical writing skills. Their energy, motivation, dedication, and vast knowledge have stirred me towards conducting research in the new and exciting area of compressive sensing and also completing this dissertation.

I would like to thank Dr. Dale Olesky for teaching me invaluable concepts of numerical analysis, for serving as a member of my supervisory committee, and for supporting me.

I am also very grateful to the staff and faculty of the Department of Electrical and Computer Engineering for assisting me during my doctoral studies. Thank you Vicky, Moneca, Lynne, Janice, Dan, Monique, Steve, Kevin, and Erik for administrative and professional support and advice.

I would like to thank all the new friends I made in the last three and half years. It has always been wonderful and refreshing while spending time, eating delicious food, playing games, watching movies, and swimming with you.

Last but not the least, thank you Neeta for coming into my life, for being there for me, and for bearing with the pain of staying apart during my doctoral studies. I thank my parents, Dambar Pant and Chandra Pant, who have made many sacrifices for me to reach this stage; thank you mother for your deepest prayers for me. I would also like to remember and thank my siblings Kalpana, Ganga, and Ramesh and parents-in-law Ganesh Koirala and Shanta Koirala.

Jeevan Kumar Pant

DEDICATION

To my family

Chapter 1

Introduction

Although the concept of signal sparsity and ℓ_1-norm based recovery techniques can be traced back to the work of Logan [1] in 1965, Santosa and Symes [2] in 1986, and Donoho and Stark in 1989 [3], it is generally agreed that the foundation of the current state-of-the-art in *compressive sensing* (CS) theory, also known as *compressive sampling*, was laid by Candes, Romberg, and Tao, Donoho, and Candes and Tao in [4–6] in 2006. These papers together with several other papers [7–16] have inspired a burst of intensive research activity in CS [17–25] in the past six years.

In a nutshell, CS involves acquiring a signal of interest *indirectly* by collecting a *relatively small* number of its "projections" rather than evenly sampling it at the Nyquist rate [26], [27], which can be prohibitively high for broadband signals encountered in many applications. In a way, this new signal acquisition paradigm has fundamentally changed the traditional approach with which digital data are acquired. Before we proceed with the technical details, it should be stressed that although CS is not a universal sampling technique, it is nevertheless an effective method for data acquisition in situations such as the following [28]:

- The number of sensors is limited due to implementation constraints or cost.
- Measurements may be extremely expensive, e.g., in certain image processing applications that involve neutron scattering.
- The sensing process may be slow so that only a small number of measurements can be collected.
- Many inverse problems are such that the only way to acquire data is to use a measurement matrix and, consequently, such problems are most suitable for

CS.

- In some applications, the rate of sampling due to the conventional Nyquist theorem is too high to implement. For example, the current analog-to-digital conversion (ADC) technology based on uniform sampling in the time or spatial domain is limited to sampling rates of about 1 GHz.

The two key notions in the development of the current CS theory are *sparsity* and *incoherence*. Candes and Wakin [28] highlight the principle of sparsity in terms of the following facts:

- Sparsity expresses the idea that the information rate of a continuous-time signal may be much smaller than that suggested by its bandwidth.
- A discrete-time signal depends on a number of degrees of freedom which is relatively much smaller than its length.
- Many natural signals are sparse or compressible in the sense that they have sparse or approximately sparse representations when expressed in an approximate basis or dictionary.

Candes and Romberg [17] and Candes and Wakin [28] explain the notion of incoherence between a sensing system $\boldsymbol{P}$ and a sparsifying dictionary $\boldsymbol{D}$ by saying that the sensing vectors, i.e., some of the rows in $\boldsymbol{P}$, must be spread out in the $\boldsymbol{D}$ domain, just as a spike in the time domain is spread out in the frequency domain. In this regard, incoherence extends the duality between time and frequency.

1.1 What is Compressive Sensing?

1.1.1 *A Strategy for Sensing Data*

A discrete signal is said to be K-sparse (K-approximately-sparse) if it has only K nonzero components (K significant nonzero components). Let us now consider a discrete signal $\boldsymbol{u}$ which itself may or may not be sparse in the canonical basis but is sparse or approximately sparse in an appropriate dictionary $\boldsymbol{D} \in R^{N \times L}$, that is,

$$\boldsymbol{u} = \boldsymbol{D}\boldsymbol{x} \tag{1.1}$$

where $\boldsymbol{x}$ is a sparse or approximately sparse. A central idea in the CS theory is about how a discrete signal is acquired, which can be explained as follows. The acquisition of a signal $\boldsymbol{u}$ of length N is carried out by measuring M projections of $\boldsymbol{u}$ into sensing vectors also known as testing vectors $\{\boldsymbol{p}_i, i = 1, 2, \ldots, M\}$ in the form of $y_i = \boldsymbol{p}_i^T \boldsymbol{u}$ for $i = 1, 2, \ldots, M$. For improved sensing efficiency, a relatively much smaller number of measurements would be preferred, that is, M should be considerably smaller than N; hence the name *compressive sensing*. This data acquisition mechanism which marks a fundamental departure from the conventional data acquisition-compression-transmission-decompression framework is at the core of a CS system. The conventional framework collects a vast amount of data for the acquisition of a high-resolution signal or image and then essentially discards most of the data collected in the compression stage while in CS the data are measured in an compressed manner, and the much reduced amount of measurements is transmitted or stored economically. Every bit of the measurements is then utilized to recover the signal at the base station where sufficient computing resources are available to employ a reconstruction algorithm to recover the signal that was acquired at a rate much lower than the Nyquist rate. Using matrix notation, this sensing process is described as

$$\boldsymbol{y} = \hat{\boldsymbol{P}} \cdot \boldsymbol{u} \tag{1.2a}$$

where

$$\hat{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \\ \vdots \\ \boldsymbol{p}_M^T \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} \tag{1.2b}$$

To ensure that the projections are meaningful, it is often assumed that the lengths of the projection vectors $\{\boldsymbol{p}_i, i = 1, 2, \ldots, M\}$ are unity. Here are two questions that naturally arise from the signal model in (1.2):

(i) What type of matrix $\hat{\boldsymbol{P}}$ should one choose for the purpose of sensing?

(ii) How many measurements $\{y_i, i = 1, 2, \ldots, M\}$ should one collect so that these measurements will be sufficient to recover signal $\boldsymbol{u}$?

To address these questions explicitly, one needs the concept of *incoherence* between the sensing system represented by $\hat{\boldsymbol{P}}$ and the sparsifying system represented by $\boldsymbol{D}$.

1.1.2 *Coherence Between P and D*

To formally discuss the concept of incoherence, we take the view that $\hat{\boldsymbol{P}}$ is part of an orthonormal basis $\boldsymbol{P} \in R^{N\times N}$ as composed of its M rows. Furthermore, for simplicity of illustration, in what follows it is assumed that $\boldsymbol{D}$ is an orthonormal basis. The *coherence*, which is also known as *mutual coherence* in the literature, between $\boldsymbol{P}$ and $\boldsymbol{D}$ is defined as

$$\mu\left(\boldsymbol{P},\boldsymbol{D}\right)=\sqrt{N}\cdot\max_{1\leq k,j\leq N}\left|\boldsymbol{p}_k^T\boldsymbol{d}_j\right| \tag{1.3}$$

where $\boldsymbol{p}_k^T$ and $\boldsymbol{d}_j$ are the kth row and jth column of $\boldsymbol{P}$ and $\boldsymbol{D}$, respectively [28], [29]. Obviously μ measures the *largest correlation* between a basis vector in $\boldsymbol{P}$ and a basis vector in $\boldsymbol{D}$. It can be shown that the coherence measure μ in (1.3) is always in the range $[1,\sqrt{N}]$.

We are interested in basis pairs $\{\boldsymbol{P},\boldsymbol{D}\}$ with *small* coherence and in such a case, bases $\{\boldsymbol{P},\boldsymbol{D}\}$ are said to be *incoherent*. In the context of CS, the incoherence between the basis involved in signal sensing and the basis involved in a sparse representation of the signal is a crucial feature of an effective CS system as will be seen below. In this regard, an important question to address is what kind of matrices $\boldsymbol{P}$ and $\boldsymbol{D}$ should one use in order to achieve a small $\mu\left(\boldsymbol{P},\boldsymbol{D}\right)$?

If we let $\boldsymbol{V}=\sqrt{N}\cdot\boldsymbol{P}\boldsymbol{D}=\{v_{k,j}\}$, then the kth row of $\boldsymbol{V}$ is equal to $\sqrt{N}\cdot\boldsymbol{p}_k^T\boldsymbol{D}$ whose 2-norm is $\sqrt{N}$. Therefore, for a large μ, the entries of each row of $\boldsymbol{V}$ must be widely spread out, i.e., they should not be concentrated. For instance, a most concentrated row of $\boldsymbol{V}$ would look like $[0\ \cdots\ 0\ \pm\sqrt{N}\ 0\ \cdots\ 0]^T$ which gives $\mu=\sqrt{N}$ while a most spread out row of $\boldsymbol{V}$ would look like $[\pm 1\ \ \pm 1\ \ \cdots\ \ \pm 1]^T$; if every row of $\boldsymbol{V}$ assumes this form, we would have $\mu=1$. A similar claim can be made for the columns of $\boldsymbol{V}$. Furthermore, requiring that $\sqrt{N}\cdot\boldsymbol{p}_k^T\boldsymbol{D}=[\pm 1\ \ \pm 1\ \ \cdots\ \ \pm 1]^T$ for each row of $\boldsymbol{P}$ is the same as saying that each row of $\boldsymbol{P}$ must be spread out in the $\boldsymbol{D}$ domain. The same can also be said about $\boldsymbol{D}$: each column of $\boldsymbol{D}$ must be spread out in the $\boldsymbol{P}$ domain in order to have a small mutual coherence. Some examples of matrix pairs that have a small coherence are as follows:

- If $\boldsymbol{I}_N$ is the identity matrix of size $N\times N$ and $\boldsymbol{C}_N$ is a discrete-cosine transformation matrix, then $\mu\left(\boldsymbol{I}_N,\boldsymbol{C}_N^T\right)\leq\sqrt{2}$.
- If $\boldsymbol{F}_N$ is a Fourier transformation matrix, then $\mu\left(\boldsymbol{I}_N,\boldsymbol{F}_N^H\right)=1$, where $\boldsymbol{F}_N^H$ is the Hermitian of $\boldsymbol{F}_N$.

- If $\boldsymbol{D}$ is a fixed orthonormal basis and $\boldsymbol{P}$ is an orthonormal matrix with independent identically distributed (i.i.d.) entries, e.g., Gaussian or ± 1 binary entries, then $\mu(\boldsymbol{P}, \boldsymbol{D})$ is also very low [28].

1.2 State-of-the-Art

The reconstruction of signals in CS is founded on a sound theoretical framework. Below, we review some key theoretical results and some important state-of-the-art algorithms for the reconstruction of sparse signals in CS.

1.2.1 Fundamentals of CS

Below we review several known results in CS theory that pertain to noise-free and noisy data.

1.2.1.1 Compressive Sensing for Noise-Free Data

Theorem 1.2.1 ([17]). *Let $\boldsymbol{u} \in R^N$ and suppose that $\boldsymbol{u}$ is K-sparse in an orthonormal basis $\boldsymbol{D}$. Now select M measurements in the $\boldsymbol{P}$ domain uniformly at random via (1.2) such that the M sensing vectors $\{\boldsymbol{p}_i, i = 1, 2, \ldots, M\}$ are M rows uniformly randomly selected from matrix $\boldsymbol{P} \in R^{N \times N}$. If*

$$M \geq C \cdot \mu^2(\boldsymbol{P}, \boldsymbol{D}) \cdot K \cdot \log N \tag{1.4}$$

for some positive constant C, then signal $\boldsymbol{u}$ can be exactly reconstructed with a very high probability by using (1.1) where $\boldsymbol{x}$ is the solution of the convex minimization problem

$$\underset{\boldsymbol{x}}{minimize} \quad ||\boldsymbol{x}||_1 \tag{1.5a}$$

$$subject\ to: \quad \boldsymbol{\Phi x} = \boldsymbol{y} \tag{1.5b}$$

Matrix $\boldsymbol{\Phi} = \hat{\boldsymbol{P}}\boldsymbol{D}$, $\hat{\boldsymbol{P}}$ is given by (1.2b) and $||\boldsymbol{x}||_1$ is the ℓ_1 norm of $\boldsymbol{x}$ defined as $||\boldsymbol{x}||_1 = \sum_{i=1}^{N} |x_i|$.

Remarks (i) It is important to clarify the significance of the condition in (1.4) as it relates the key parameters in a CS platform, namely, the signal size N, the sparsity of the signal K (in domain $\boldsymbol{D}$), the number of measurements M that is sufficient to recover $\boldsymbol{x}$, and the incoherence between $\boldsymbol{P}$ and $\boldsymbol{D}$. Eq.

(1.4), in effect, states that one can use a number of measurements M that is substantially smaller than the signal's dimension N to reconstruct the signal if bases $\boldsymbol{P}$ and $\boldsymbol{D}$ are *incoherent*, i.e., $\mu(\boldsymbol{P}, \boldsymbol{D}) \approx 1$. Indeed, if $\mu(\boldsymbol{P}, \boldsymbol{D}) = 1$, (1.4) becomes

$$M \geq C \cdot K \cdot \log N$$

Based on the preceding principles, there is a *four-to-one* practical rule which says that for exact reconstruction, one needs about *four incoherent measurements per unknown nonzero term* in $\boldsymbol{u}$, i.e.,

$$M \geq 4K \tag{1.6}$$

regardless of the signal's dimension N [28]. We stress that to reach the condition in (1.4) or the 4-to-1 rule in (1.6), the low coherence is the key. In effect, if the two bases are not incoherent, say $\mu(\boldsymbol{P}, \boldsymbol{D}) = \sqrt{N}$, then (1.4) becomes

$$M \geq C \cdot N \cdot K \cdot \log N \tag{1.7}$$

Obviously, in this case no compressed sensing can be achieved.

(ii) The constraint $\boldsymbol{\Phi x} = \boldsymbol{y}$ in (1.5) with $\hat{\boldsymbol{P}} \in R^{M \times N}$ where $M < N$ and $\boldsymbol{D} \in R^{N \times N}$ is an *underdetermined* linear system and the problem in (1.5) is referred to as the *basis pursuit* (BP) problem in the literature.

1.2.1.2 Compressive Sensing for Noisy Data

An important feature of compressive sensing is that it is robust in the sense that small perturbations in the data cause small deviations in the reconstruction. Here small perturbations in the data refers either to signals that are not exactly sparse but nearly sparse or to the presence of noise in the sampling process. A signal model for CS that takes both of these issues into consideration is given by

$$\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{w} \tag{1.8a}$$

where $\boldsymbol{w}$ represents measurement noise and $\boldsymbol{\Phi}$ is a sensing matrix of size M by N given by

$$\boldsymbol{\Phi} = \boldsymbol{SPD} = \hat{\boldsymbol{P}}\boldsymbol{D} \tag{1.8b}$$

In (1.8b), $\boldsymbol{S}$ is a matrix of size M by N that selects M rows from matrix $\boldsymbol{P}$, thus $\boldsymbol{SP} = \hat{\boldsymbol{P}}$; that is $\hat{\boldsymbol{P}}$ is the same matrix introduced in (1.2). As expected, signal $\boldsymbol{x}$ in (1.8a) may be estimated from noisy measurement $\boldsymbol{y}$ by solving the constrained convex problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_1 \tag{1.9a}$$

$$\text{subject to:} \quad ||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2 \leq \upsilon \tag{1.9b}$$

where υ is a bound imposed on the amount of noise in the data. Suppose the measurement noise $\boldsymbol{w}$ is white with each component having zero mean and standard deviation σ, then for sufficiently large M we have $\upsilon \approx M\sigma^2$.

The robustness of current CS theory relies heavily on a notion called the *restricted isometry property* (RIP) which was first introduced and studied in [6] and [30]. Related results are also presented in [3], [4], and [31].

Definition For each integer $K = 1, 2, \ldots$ define the isometry constant δ_K of a matrix $\boldsymbol{\Phi}$ as the smallest number such that

$$(1 - \delta_K)||\boldsymbol{x}||_2^2 \leq ||\boldsymbol{\Phi x}||_2^2 \leq (1 + \delta_K)||\boldsymbol{x}||_2^2 \tag{1.10}$$

for all K-sparse vectors $\boldsymbol{x}$. We shall say, matrix $\boldsymbol{\Phi}$ obeys the RIP of order K if δ_K in (1.10) is less than one but not too close to one.

Several intuitive observations on RIP can be made, as follows, before presenting some relevant results:

(i) Suppose that $\boldsymbol{\Phi}$ is an orthogonal matrix, although we know that it cannot be so in practice, then the condition in (1.10) would hold for any vector $\boldsymbol{x}$ and any value of δ_K. Now, for a K-sparse $\boldsymbol{x}$, $\boldsymbol{\Phi x}$ is equal to $\boldsymbol{\Phi}_K \boldsymbol{x}_K$ where $\boldsymbol{\Phi}_K$ is a sub-matrix of $\boldsymbol{\Phi}$ consisting of those K columns of $\boldsymbol{\Phi}$ corresponding to the K nonzero entries of $\boldsymbol{x}$ and $\boldsymbol{x}_K$ is a vector $\boldsymbol{x}$ with its zero entries dropped (note that $||\boldsymbol{x}||_2 = ||\boldsymbol{x}_K||_2$). Therefore, the condition in (1.10) amounts to saying that all sub-matrices of K columns taken from $\boldsymbol{\Phi}$ are nearly orthogonal.

(ii) Obviously, the condition in (1.10) prevents any K-sparse $\boldsymbol{x}$ from being in the null space of $\boldsymbol{\Phi}$. This is important in the context of CS because if there exists a K-sparse $\boldsymbol{x}$ that belongs to the null space of $\boldsymbol{\Phi}$ then $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{w} = \boldsymbol{w}$ which means that the measurement obtained is totally useless as $\boldsymbol{y}$ contains no information whatsoever about sparse signal $\boldsymbol{x}$.

(iii) In the case where δ_{2K} is sufficiently smaller than one, then for all K-sparse vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, vector $\boldsymbol{x}_1 - \boldsymbol{x}_2$ is at most $2K$-sparse and hence we have

$$(1-\delta_{2K})\,||\boldsymbol{x}_1-\boldsymbol{x}_2||_2^2 \leq ||\boldsymbol{\Phi}\boldsymbol{x}_1-\boldsymbol{\Phi}\boldsymbol{x}_2||_2^2 \leq (1+\delta_{2K})\,||\boldsymbol{x}_1-\boldsymbol{x}_2||_2^2 \tag{1.11}$$

An immediate consequence of the condition in (1.11) for the noise-free case is that the measurement $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}$ of a K-sparse signal $\boldsymbol{x}$ completely characterizes the signal in the sense that if measurements $\boldsymbol{\Phi}\boldsymbol{x}_1$ and $\boldsymbol{\Phi}\boldsymbol{x}_2$ are identical, then signals $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ must be identical.

Theorem 1.2.2 ([17]). *For the noise-free case, assume that $\delta_{2K} < \sqrt{2}-1$. Then the solution $\boldsymbol{x}^*$ to the problem in (1.5) satisfies the inequalities*

$$||\boldsymbol{x}-\boldsymbol{x}^*||_2 \leq C_0 \cdot ||\boldsymbol{x}-\boldsymbol{x}_K||_1 / \sqrt{K} \tag{1.12a}$$

and

$$||\boldsymbol{x}-\boldsymbol{x}^*||_1 \leq C_0 \cdot ||\boldsymbol{x}-\boldsymbol{x}_K||_1 \tag{1.12b}$$

for some constant C_0, where $\boldsymbol{x}$ is the signal in the model of (1.8a) that one seeks to reconstruct and $\boldsymbol{x}_K$ is vector $\boldsymbol{x}$ with all but the largest K components set to zero.

Remark If $\boldsymbol{x}$ is K-sparse, then $\boldsymbol{x}_K = \boldsymbol{x}$ and hence the conditions in (1.12) imply that $\boldsymbol{x}^* = \boldsymbol{x}$, that is, the recovery is perfect. However, the power of Theorem 1.2.2 is that it deals with non-sparse signals as well. In particular, if $\boldsymbol{x}$ is not sparse but approximately K-sparse then $||\boldsymbol{x}-\boldsymbol{x}_K||_1$ is small and Theorem 1.2.2 implies that the solution $\boldsymbol{x}^*$ of the problem in (1.9) is expected to be a reconstruction of $\boldsymbol{x}$ with good accuracy.

Theorem 1.2.3 ([30]). *For the case of noisy measurements using the model in (1.8), assume that $\delta_{2K} < \sqrt{2}-1$. Then the solution $\boldsymbol{x}^*$ of the problem in (1.9) satisfies the inequality*

$$||\boldsymbol{x}-\boldsymbol{x}^*||_2 \leq C_0 \cdot ||\boldsymbol{x}-\boldsymbol{x}_K||_1 / \sqrt{K} + C_1\varepsilon \tag{1.13}$$

for some constants C_0 and C_1.

For instance, if $\delta_{2K} = 1/4$, then $C_0 \leq 5.5$ and $C_1 \leq 6$.

Remark Again, if $\boldsymbol{x}$ is K-sparse, then $\boldsymbol{x}_k = \boldsymbol{x}$, and the condition in (1.13) implies that $||\boldsymbol{x}-\boldsymbol{x}^*||_2 \leq C_1\varepsilon$. In other words, for sparse signals the solution accuracy

is determined by the noise level. Like Theorem 1.2.2, Theorem 1.2.3 also deals with non-sparse signals. When $\boldsymbol{x}$ is not sparse but approximately K-sparse, Theorem 1.2.3 implies that the accuracy of the solution $\boldsymbol{x}^*$ of the problem in (1.9) is determined by both the closeness of signal $\boldsymbol{x}$ to its K-sparse counterpart and the noise level.

1.2.2 Algorithms for CS

Several algorithms for reconstruction of sparse signals have been proposed in the past several years. Below, we present a review of the known algorithms that are based on ℓ_p-pseudonorm minimization with $p < 1$.

1.2.2.1 Signal reconstruction based on ℓ_p-pseudonorm minimization

We begin by defining the ℓ_0 pseudonorm as $||\boldsymbol{x}||_0 = \sum_{i=1}^{N} |x_i|^0$ which is equal to the number of nonzero components in $\boldsymbol{x}$. Hence, if we assume that signals $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are of the same length, then $\boldsymbol{x}_1$ is more sparse than $\boldsymbol{x}_2$ if $||\boldsymbol{x}_1||_0 < ||\boldsymbol{x}_2||_0$. It follows that the problem of reconstructing a sparse signal $\boldsymbol{x}$ from linear measurement $\boldsymbol{y} = \boldsymbol{\Phi x}$ can be formulated as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_0 \tag{1.14a}$$

$$\text{subject to:} \quad \boldsymbol{\Phi x} = \boldsymbol{y} \tag{1.14b}$$

If the measurement is corrupted by white noise $\boldsymbol{w}$, i.e., $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{w}$, then the signal reconstruction problem is modified to

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_0 \tag{1.15a}$$

$$\text{subject to:} \quad ||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2 \leq \upsilon \tag{1.15b}$$

where bound υ is related to the variance of noise $\boldsymbol{w}$. Unfortunately, the computational complexity of the problems in (1.14) and (1.15) grows exponentially with respect to the signal size, and becomes prohibitive even for signals of moderate sizes [32]. To deal with this complexity issue a technique has been proposed whereby the problems in (1.14) and (1.15) are converted to the problems in (1.5) and (1.9), respectively, by replacing function $||\boldsymbol{x}||_0$ by the convex function $||\boldsymbol{x}||_1$. In the literature, this course of action is called convex relaxation as it replaces a difficult nonconvex problem with an easy convex problem. However, exact signal reconstruction by solving the

ℓ_1-minimization problem in (1.5) is not guaranteed if the condition in (1.4) is not satisfied.

Alternative optimization methods proposed recently in [33, 34] have been shown to offer improved performance relative to the method based on (1.5), which is known as the BP method. In these contributions, it is demonstrated that more accurate signal reconstruction can be achieved by solving an ℓ_p-pseudonorm based minimization problem with $p < 1$. Specifically, the ℓ_p-pseudonorm minimization problem for the reconstruction of sparse signals whose measurements are free of noise corruption is given by

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_p^p \tag{1.16a}$$

$$\text{subject to:} \quad \boldsymbol{\Phi x} = \boldsymbol{y} \tag{1.16b}$$

where $0 \leq p < 1$ and $||\boldsymbol{x}||_p^p = \sum_{i=1}^{N} |x_i|^p$. For measurements contaminated by Gaussian white noise $\boldsymbol{w}$, i.e., $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{w}$, the ℓ_p-pseudonorm minimization problem is formulated as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_p^p \tag{1.17a}$$

$$\text{subject to:} \quad ||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2 \leq \upsilon \tag{1.17b}$$

Below, we illustrate why ℓ_p-pseudonorm minimization is more effective in determining a sparse signal relative to ℓ_1-norm and ℓ_2-norm minimizations.

Difference between $||\boldsymbol{x}||_0$, $||\boldsymbol{x}||_p$, and $||\boldsymbol{x}||_1$

The plots of functions $|x_i|^0$, $|x_i|$, and $|x_i|^p$ for $p = 0.3$ and 0.08 are shown in Fig. 1.1. As can be seen, both functions $|x_i|^{0.3}$ and $|x_i|^{0.08}$ are closer to function $|x_i|^0$ than the function $|x_i|$ is. Also, $|x_i|^{0.08}$ is closer to $|x_i|^0$ than $|x_i|^{0.3}$ is. Note that the ℓ_0 pseudonorm $||\boldsymbol{x}||_0$, the ℓ_p pseudonorm $||\boldsymbol{x}||_p$, and the ℓ_1 norm $||\boldsymbol{x}||_1$ consist of summations of functions $|x_i|^0$, $|x_i|^p$, and $|x_i|$, respectively, for $i = 1, 2, \ldots, N$. Therefore, from the plots in Fig. 1.1, one can infer that the ℓ_p pseudonorm with $0 \leq p < 1$ approximates the ℓ_0 pseudonorm more accurately than the ℓ_1 norm does.

Why ℓ_2 minimization fails?

Consider a signal $\boldsymbol{x}$ of length 3, i.e., $\boldsymbol{x} = [x_1 \ x_2 \ x_3]^T$, and measurement matrix $\boldsymbol{\Phi}$ of size 1×3. Equation $\boldsymbol{\Phi x} = \boldsymbol{y}$, in such case, can be plotted as a straight line as shown Fig. 1.2. Since all the points in the straight line are solutions of $\boldsymbol{\Phi x} = \boldsymbol{y}$, the straight

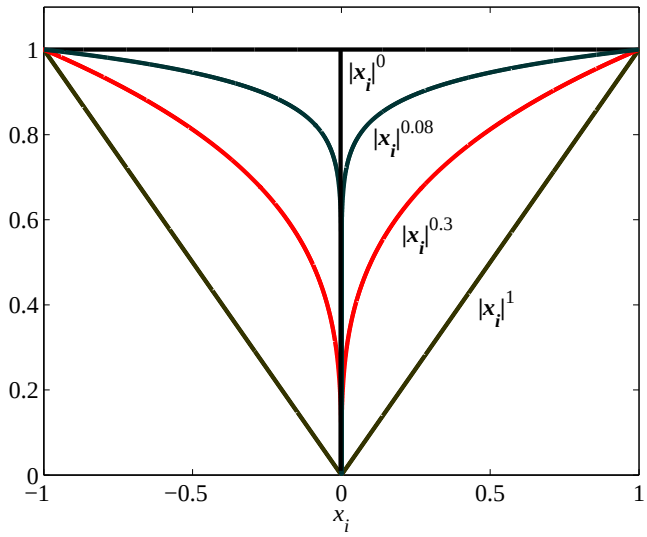


Figure 1.1: Plots of functions $|x_i|^0$, $|x_i|^{0.08}$, $|x_i|^{0.3}$, and $|x_i|$.

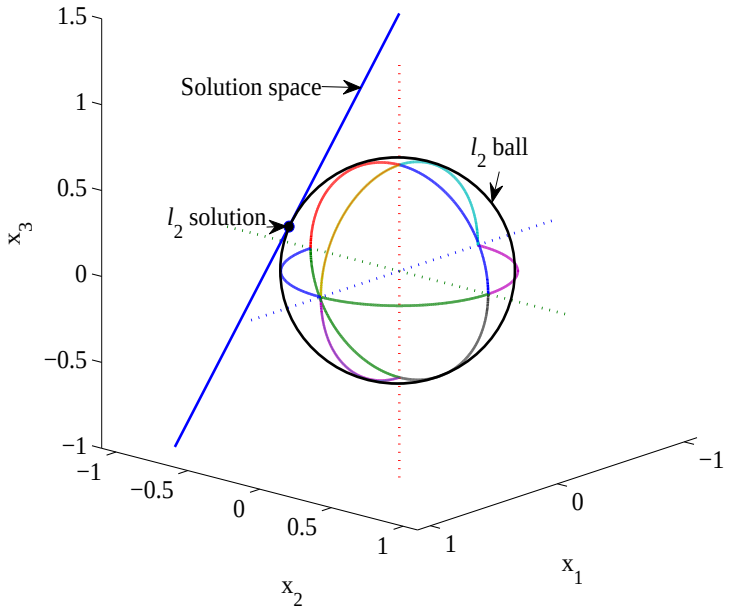
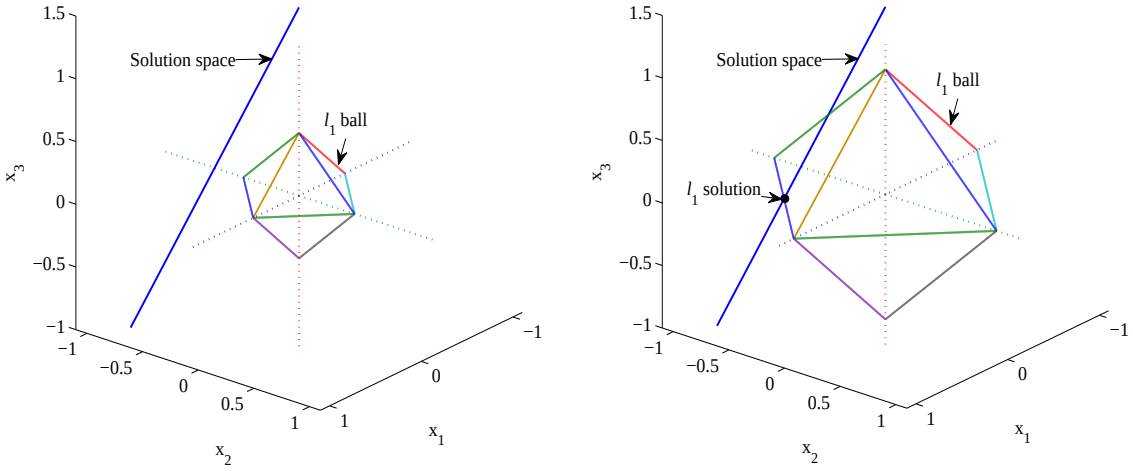


Figure 1.2: Solution space $\boldsymbol{\Phi x} = \boldsymbol{y}$ and ℓ_2 ball; the ℓ_2 solution is not sparse.

line is also referred to as the solution space. Fig. 1.2 also shows an ℓ_2 ball, which is the surface of a sphere given by $||\boldsymbol{x}||_2 = c$ where c is a positive scalar constant. In other words, all the points in the ℓ_2 ball have a constant ℓ_2 norm. As we increase r from a sufficiently small value, the size of the ℓ_2 ball increases and eventually touches the solution space. The point of touch where the solution space is tangent to the ℓ_2 ball is the minimum ℓ_2-norm solution and is also called the ℓ_2 solution. As can be noticed from the figure, all the three components of the ℓ_2 solution are nonzero, hence, the ℓ_2 solution is not sparse.

Why ℓ_1 minimization works?

An ℓ_1 ball is defined by the set $\{\boldsymbol{x} : ||\boldsymbol{x}||_1 = c\}$ where c is a positive scalar constant. Fig. 1.3a shows the ℓ_1 ball with $c = 0.5$ and the solution space $\boldsymbol{\Phi x} = \boldsymbol{y}$. As we increase r, the size of the ℓ_1 ball increases. When $c = 1$, the ℓ_1 ball touches the solution space; the point of touch, which is the ℓ_1 solution, is marked by a black dot in Fig. 1.3b. As can be noticed, the components of the ℓ_1 solution are given by $x_1 \neq 0$,



(a) $||\boldsymbol{x}||_1 = 0.5$. (b) $||\boldsymbol{x}||_1 = 1$.

Figure 1.3: Solution space $\boldsymbol{\Phi x} = \boldsymbol{y}$ and ℓ_1 ball; the ℓ_1 solution is sparser than the ℓ_2 solution.

$x_2 \neq 0$, and $x_3 = 0$. Hence, the ℓ_1 solution is clearly sparser than the ℓ_2 solution.

Why ℓ_p minimization works better?

An ℓ_p ball is defined as a set of points $\{\boldsymbol{x} : ||\boldsymbol{x}||_p = c\}$ where c is a positive scalar constant. Fig. 1.4a shows an ℓ_p ball for $p = 0.5$ and $c = 0.5$, and the solution space. Recall from Fig. 1.3b that, for $c = 1$ the ℓ_1 ball touches the solution space yielding a 2-sparse solution. However, as shown in Fig. 1.4b, the ℓ_p ball does not touch the solution space for $c = 1$ and thus it does not have a 2-sparse solution. Fig. 1.4c shows that the ℓ_p ball touches the solution space for $c = 1.5$. One can see that the point of touch or the ℓ_p solution has only one nonzero component, i.e., $x_3 \neq 0$. Hence, the ℓ_p solution is sparser than the ℓ_1 solution.

1.2.2.2 Unconstrained formulation of signal reconstruction problems

An effective approach to deal with the convex constrained optimization problems in (1.5) and (1.9) is to convert each problem into a convex unconstrained problem of the form

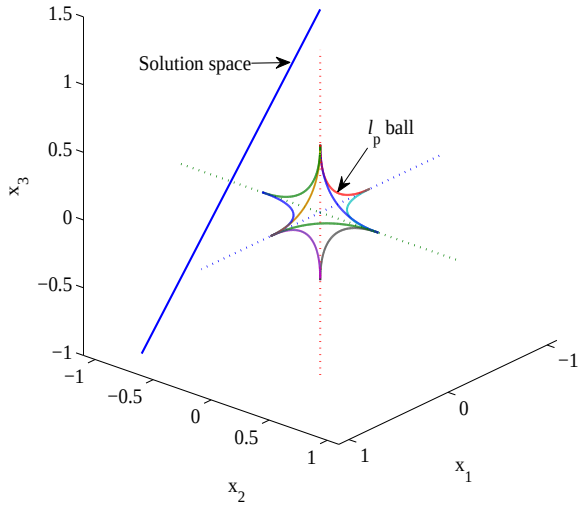
$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \tfrac{1}{2}\,||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2 + \lambda||\boldsymbol{x}||_1 \tag{1.18}$$

where $\lambda > 0$ is a regularization parameter. For the problem in (1.9), the λ in (1.18) is related to bound υ in (1.9b) and hence it depends on the variance of noise $\boldsymbol{w}$. For the problem in (1.5), the λ in (1.18) must be sufficiently small so that the solution of (1.18) satisfies the equality constraint in (1.5b). Because of the presence of term $\lambda||\boldsymbol{x}||_1$, the problem in (1.18) is a convex problem with a nonsmooth objective function. The problem has attracted a great deal of attention and, as a result, many efficient algorithms and software have been proposed. See [35] for an excellent survey of these algorithms.

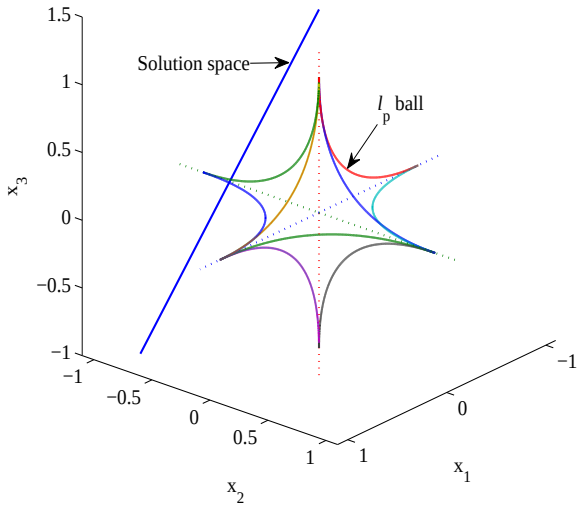
By analogy with the problems in (1.16) and (1.17) an ℓ_p-pseudonorm regularized least-squares optimization problem can be posed as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \tfrac{1}{2}\,||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2 + \lambda||\boldsymbol{x}||_p^p \tag{1.19}$$

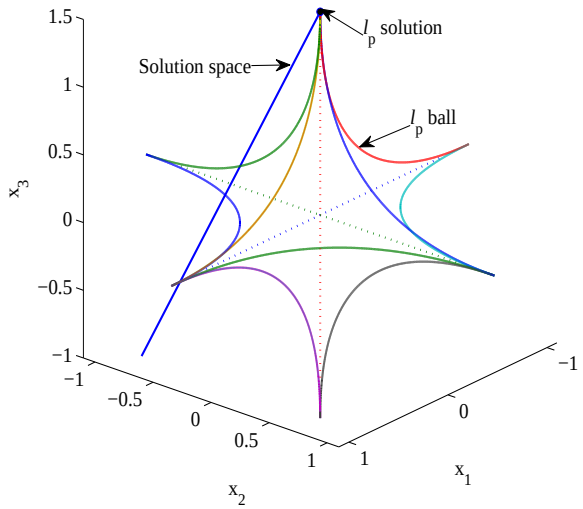
where $0 \leq p < 1$. On comparing the problem in (1.19) with that of (1.18), it is important to stress that the problem in (1.19) is both nonsmooth and nonconvex because of the presence of the term $||\boldsymbol{x}||_p^p$.



(a) $||\boldsymbol{x}||_p = 0.5$.



(b) $||\boldsymbol{x}||_p = 1$.



(c) $||\boldsymbol{x}||_p = 1.5$

Figure 1.4: Solution space $\boldsymbol{\Phi x} = \boldsymbol{y}$ and ℓ_p ball with $p = 0.5$; the minimum ℓ_p-pseudonorm solution is sparser than the minimum ℓ_1-norm solution.

1.2.2.3 Reconstruction of block-sparse signals

In conventional CS, the algorithms used to recover sparse signals do not take into account the block structure of the signal components, where the nonzero coefficients occur in clusters [7, 8, 28, 36–39].

Consider signal $\boldsymbol{x}$ of length N which is divisible by a positive integer d. We divide signal $\boldsymbol{x}$ into N/d blocks $\tilde{\boldsymbol{x}}_1$, $\tilde{\boldsymbol{x}}_2$, ..., $\tilde{\boldsymbol{x}}_{N/d}$ and denote $\boldsymbol{x}$ as

$$\boldsymbol{x} = \begin{bmatrix} \tilde{\boldsymbol{x}}_1^T & \tilde{\boldsymbol{x}}_2^T & \cdots & \tilde{\boldsymbol{x}}_{N/d}^T \end{bmatrix}^T \tag{1.20}$$

where

$$\tilde{\boldsymbol{x}}_i = \begin{bmatrix} x_{(i-1)d+1} & x_{(i-1)d+2} & \cdots & x_{(i-1)d+d} \end{bmatrix}^T$$

for $i = 1, 2, \ldots, N/d$ and x_i is the ith component of $\boldsymbol{x}$.

A signal $\boldsymbol{x}$ of the form in (1.20) is said to be K-block sparse if $\boldsymbol{x}$ has K nonzero blocks with $K \ll N/d$. Note that K-sparse in conventional CS is a special case of K-block-sparse where $d = 1$. Recently, it has been shown in [40], [41], [42] that improved performance for the reconstruction of K-block-sparse signals can be achieved by solving the $\ell_{2/1}$-norm minimization problem

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & ||\boldsymbol{x}||_{2/1} \\ \text{subject to:} \quad & \boldsymbol{\Phi x} = \boldsymbol{y} \end{aligned} \tag{1.21}$$

where $||\boldsymbol{x}||_{2/1}$ is the $\ell_{2/1}$ norm of $\boldsymbol{x}$ defined as

$$||\boldsymbol{x}||_{2/1} = \sum_{i=1}^{N/d} ||\tilde{\boldsymbol{x}}_i||_2 \tag{1.22}$$

In (1.22), $||\tilde{\boldsymbol{x}}_i||_2$ is the ℓ_2 norm of the ith block $\tilde{\boldsymbol{x}}_i$.

The problem in (1.21) can be recast as

$$\begin{aligned} \underset{\boldsymbol{x},\boldsymbol{t}}{\text{minimize}} \quad & \sum_{i=1}^{N/d} t_i \\ \text{subject to:} \quad & \boldsymbol{\Phi x} = \boldsymbol{y} \\ & ||\tilde{\boldsymbol{x}}_i||_2 \leq t_i \quad 1 \leq i \leq N/d \\ & 0 \leq t_i \qquad\quad 1 \leq i \leq N/d \end{aligned} \tag{1.23}$$

and can be solved using a second-order cone-programming (SOCP) solver [43]. An

algorithm which solves the problem in (1.23) will be referred to as the $\ell_{2/1}$-SOCP algorithm hereafter.

A so-called *greedy* algorithm called block orthogonal matching pursuit (BOMP) algorithm is presented in [42]. The BOMP algorithm initializes residual $\boldsymbol{r}_0 = \boldsymbol{y}$ and index set $\mathcal{I}$ to the empty set and the lth stage of the algorithm entails the following steps:

- Choose the block index k_l as $k_l = \underset{i}{\text{argmaximum}} \left|\left|\boldsymbol{\Phi}_i^T \boldsymbol{r}_{l-1}\right|\right|_2$ where $\boldsymbol{\Phi}_i$ is a matrix of size $M \times d$ whose columns are the columns of $\boldsymbol{\Phi}$ corresponding to the indices of the ith block $\tilde{\boldsymbol{x}}_i$.
- Set $\mathcal{I} = \mathcal{I} \cup \{(k_l - 1)d + 1, (k_l - 1)d + 2, \ldots, (k_l - 1)d + d\}$.
- Find solution $\boldsymbol{x}_l$ as a vector $\boldsymbol{x}$ that minimizes

$$||\boldsymbol{y} - \boldsymbol{\Phi}_{\mathcal{I}} \boldsymbol{x}_{\mathcal{I}}||_2$$

  where $\boldsymbol{\Phi}_{\mathcal{I}}$ is a matrix whose columns are the columns of $\boldsymbol{\Phi}$ corresponding to the indices $\mathcal{I}$ and $\boldsymbol{x}_{\mathcal{I}}$ is a vector obtained by retaining the components of $\boldsymbol{x}$ with indices $\mathcal{I}$.

- Update the residual as

$$\boldsymbol{r}_l = \boldsymbol{y} - \boldsymbol{\Phi}_{\mathcal{I}} \boldsymbol{x}_{\mathcal{I}}$$

When the algorithm stops, the value of $\boldsymbol{x}$ for the indices that were not included in $\mathcal{I}$ are set to zero.

1.3 Original Contributions

The goal of this work has been to develop new computational techniques for CS of sparse signals. The major contributions are as described below:

In Chapter 2, three algorithms for the reconstruction of sparse signals from noise-free measurements are presented. The first algorithm, called the null space reweighted approximate ℓ_0 (NRAL0) algorithm, is based on the minimization of an approximate ℓ_0 pseudonorm. The second algorithm, called unconstrained approximate ℓ_p (UALP) algorithm, and the third algorithm, called unconstrained approximate ℓ_p conjugate-gradient (UALP-CG) algorithm, are based on the minimization of an approximate ℓ_p

pseudonorm with $p < 1$. In each algorithm, the minimization is carried out in the null space of the measurement matrix. By doing so, the equality condition involved is automatically satisfied and the problem under consideration becomes unconstrained. This opens the door for the use of more efficient algorithms for the optimization. In addition, in the NRAL0 algorithm, a reweighting technique is incorporated in the approximate ℓ_0 norm so as to force the algorithm to reach the desired sparse solution faster. For the three algorithms, a sequential optimization procedure is used which helps to improve convergence to the desired optimal solution. An efficient quasi-Newton algorithm is applied for the optimization in the NRAL0 and UALP algorithms and the basic conjugate-gradient (CG) algorithm described on p. 149 of [44] is applied for the optimization in the UALP-CG algorithm. Reduced mathematical complexity is achieved in the UALP algorithm by using a new line-search based on Banach's fixed-point theorem [45] [46].

In Chapter 3, two algorithms for the reconstruction of sparse signals from noisy measurements and one algorithm for the reconstruction of sparse images from noisy measurements are developed. The first algorithm, called the ℓ_p-regularized least-squares (ℓ_p-RLS) algorithm, minimizes an ℓ_p-regularized squared error by taking steps along descent directions that are computed in the null space of the measurement matrix and its complement space. The line search is carried out using the technique proposed in Chapter 2. The second algorithm, called the ℓ_p-regularized least-squares conjugate-gradient (ℓ_p-RLS-CG) algorithm, minimizes an ℓ_p-regularized squared error in the time domain by using the CG algorithm. Next, the total variation (TV) norm is generalized to a nonconvex version, called the TV_p pseudonorm, and the third algorithm, called the TV_p-regularized least-squares conjugate-gradient (TV_p-RLS-CG) algorithm, is presented. The TV_p-RLS-CG algorithm minimizes a TV_p-pseudonorm regularized square error by using a sequential procedure where each optimization is solved by a CG algorithm known as Fletcher-Reeves' algorithm [44] in conjunction with the line search proposed in Chapter 2. A technique for using the ℓ_p-RLS-CG algorithm for reconstructing signals from noise-free measurements and a technique for using the ℓ_p-RLS-CG algorithm by determining an optimal value of the involved regularization parameter are also presented.

In Chapter 4, an algorithm for the reconstruction of block-sparse signals in the CS framework is developed. The algorithm, called the $\ell_{2/p}$-regularized least-squares ($\ell_{2/p}$-RLS) algorithm, is based on the minimization of an $\ell_{2/p}$-regularized squared error. The optimization problem involved is solved using a sequential procedure where each

optimization is solved using Fletcher-Reeves' CG algorithm. This algorithm, like some of the other algorithms, uses the new line search described in Chapter 2. Two extensions of the $\ell_{2/p}$-RLS algorithm, namely, a reweighting technique for reducing the amount of computation and a technique for incorporating partial information about the location of nonzero blocks, are also presented.

Chapter 5 summarizes the results and contributions made and concludes with a discussion on possible future research directions.

Chapter 2

Reconstruction of Sparse Signals from Noiseless Measurements by Using Nonconvex Optimization

2.1 Introduction

One of the most successful algorithms used to recover sparse signals in compressive sensing (CS) is an ℓ_1-norm-minimization based algorithm known as basis pursuit (BP) algorithm [4] [5] [6]. Alternative optimization based algorithms proposed recently in [33] and [34] are shown to offer improved performance relative to the BP algorithm. In these contributions, it is demonstrated that more accurate signal reconstruction can be achieved by solving an ℓ_p-pseudonorm based minimization problem with $p < 1$. A computationally efficient algorithm based on the minimization of an approximate smoothed ℓ_0 pseudonorm, known as *smoothed* ℓ_0 (SL0) algorithm was investigated in [47].

In this chapter, three algorithms for the reconstruction of sparse signals from noiseless measurements are presented. The first algorithm is based on minimizing a reweighted approximate ℓ_0 pseudonorm, the second is based on minimizing an approximate ℓ_p pseudonorm by using a quasi-Newton technique, and the third is based on minimizing an approximate ℓ_p pseudonorm using a conjugate-gradient technique.

2.2 Minimization of Reweighted Approximate ℓ_0 Pseudonorm

In this section, we propose to reconstruct a sparse signal $\boldsymbol{x}$ from its measurement $\boldsymbol{y} = \boldsymbol{\Phi x}$ by solving the optimization problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_{0,\sigma} \tag{2.1a}$$

$$\text{subject to:} \quad \boldsymbol{\Phi x} = \boldsymbol{y} \tag{2.1b}$$

where $||\boldsymbol{x}||_{0,\sigma}$ is an approximate ℓ_0 pseudonorm of signal $\boldsymbol{x}$ given by

$$||\boldsymbol{x}||_{0,\sigma} = \sum_{i=1}^{N} f_\sigma(x_i) \tag{2.2}$$

$$= \sum_{i=1}^{N} \left(1 - e^{-x_i^2/2\sigma^2}\right) \tag{2.3}$$

where $\sigma > 0$ is an approximation parameter. The approximate ℓ_0 pseudonorm $||\boldsymbol{x}||_{0,\sigma}$ consists of a summation of functions $\{f_\sigma(x_i)\}$, where $f_\sigma(x_i)$ is an approximation of $|x_i|^0$ with parameter σ. It follows from (2.2) and (2.3) that as $\sigma \to 0$,

$$\begin{aligned} f_\sigma(x_i) &\approx \begin{cases} 0 & \text{if} \quad |x_i| = 0 \\ 1 & \text{if} \quad |x_i| \neq 0 \end{cases} \\ &= |x_i|^0 \end{aligned} \tag{2.4}$$

for $i = 1, 2, \ldots, N$ where $0^0 = 0$ is assumed. As a result, we have

$$\lim_{\sigma \to 0} ||\boldsymbol{x}||_{0,\sigma} = ||\boldsymbol{x}||_0$$

Hence, the smaller the σ, the more accurate the approximation.

2.2.1 Working in the null space of $\boldsymbol{\Phi}$

It is well known that all solutions of $\boldsymbol{\Phi x} = \boldsymbol{y}$ can be parameterized as

$$\boldsymbol{x} = \boldsymbol{x}_s + \boldsymbol{V}_n \boldsymbol{\xi} \tag{2.5}$$

where $\boldsymbol{x}_s$ is a solution of $\boldsymbol{\Phi x} = \boldsymbol{y}$, $\boldsymbol{V}_n$ is a $N \times (N - M)$ matrix whose columns constitute an orthonormal basis of the null space of $\boldsymbol{\Phi}$, and $\boldsymbol{\xi}$ is a parameter vector of dimension $N - M$ [44]. Vector $\boldsymbol{x}_s$ and matrix $\boldsymbol{V}_n$ in (2.5) can be evaluated by using the singular-value decomposition (SVD) or, more efficiently, by using the QR decomposition of matrix $\boldsymbol{\Phi}$ [48], [44]. A description of the SVD and QR decomposition is given in Appendix A. Using (2.5), the constraint in (2.1b) is eliminated and the problem in (2.1) is reduced to

$$\underset{\boldsymbol{\xi}}{\text{minimize}} \quad F_\sigma(\boldsymbol{\xi}) = \sum_{i=1}^{N} \left(1 - e^{-[x_{si} + \boldsymbol{v}_i^T \boldsymbol{\xi}]^2/2\sigma^2}\right) \tag{2.6}$$

where $\boldsymbol{v}_i^T$ denotes the ith row of matrix $\boldsymbol{V}_n$. It follows from (2.4) and (2.5) that the function $F_\sigma(\boldsymbol{\xi})$ measures the ℓ_0 pseudonorm of $\boldsymbol{x}_s + \boldsymbol{V}_n\boldsymbol{\xi}$. Therefore, when the solution of the problem in (2.6), say, ξ^*, is used in (2.5), it would yield the sparsest signal $\boldsymbol{x}$.

As long as $\sigma > 0$, the objective function in (2.6) remains differentiable and its gradient can be obtained as

$$\hat{\boldsymbol{g}} = \frac{\boldsymbol{V}_n^T \boldsymbol{g}}{\sigma^2} \tag{2.7a}$$

where $\boldsymbol{g} = [g_1 \; g_2 \; \cdots \; g_N]^T$ with

$$g_i = \left[x_s(i) + \boldsymbol{v}_i^T \boldsymbol{\xi}\right] e^{-[x_s(i) + \boldsymbol{v}_i^T \boldsymbol{\xi}]^2/2\sigma^2} \tag{2.7b}$$

Evidently, working in the null space of $\boldsymbol{\Phi}$ through the parameterization in (2.5) facilitates the elimination of the constraints in (2.1b) and, furthermore, it reduces the problem size from N to $N - M$. In this way, unconstrained optimization methods that are more powerful than the steepest-descent method can be applied to improve the reconstruction performance, as will be shown below.

2.2.2 Reweighting the approximate ℓ_0 pseudonorm

Signal reconstruction based on the solution of the problem in (2.6) works well, but the technique can be considerably enhanced by incorporating a reweighting strategy. The reweighted unconstrained problem can be formulated as

$$\underset{\boldsymbol{\xi}}{\text{minimize}} \quad F_\sigma(\boldsymbol{\xi}) = \sum_{i=1}^{N} w_i \left(1 - e^{-[x_s(i) + \boldsymbol{v}_i^T \boldsymbol{\xi}]^2/2\sigma^2}\right) \tag{2.8}$$

where w_i are positive scalars that form a weight vector $\boldsymbol{w} = [w_1\ w_2\ \cdots\ w_N]$. Starting with an initial $\boldsymbol{w}^{(0)} = \boldsymbol{e}_N$, where $\boldsymbol{e}_N$ is the all-one vector of dimension N, in the $(k+1)$th iteration the weight vector is updated to $\boldsymbol{w}^{(k+1)}$ with its ith component given by

$$w_i^{(k+1)} = \frac{1}{|x_i^{(k)}| + \tau} \tag{2.9}$$

where $x_i^{(k)}$ denotes the ith component of vector $\boldsymbol{x}^{(k)}$ obtained in the kth iteration as $\boldsymbol{x}^{(k)} = \boldsymbol{x}_s + \boldsymbol{V}_n\boldsymbol{\xi}^{(k)}$, and τ is a small positive scalar which is used to prevent numerical instability when $|x_i^{(k)}|$ approaches zero. Evidently, for a small $|x_i^{(k)}|$ (2.9) yields a large weight $w_i^{(k+1)}$ and hence solving the problem in (2.8) tends to reduce $|x_i^{(k)}|$ further thus forcing a sparse solution. The gradient of the reweighted objective function in (2.8) is given by (2.7a) where the ith component of $\boldsymbol{g}$ is given by

$$g_i = w_i \left[x_s(i) + \boldsymbol{v}_i^T\boldsymbol{\xi}\right] e^{-[x_s(i)+\boldsymbol{v}_i^T\boldsymbol{\xi}]^2/2\sigma^2} \tag{2.10}$$

It should be mentioned that various reweighting techniques have been recently proposed in the literature, see, for example, [34], [49]. In the algorithms presented in [34] and [49], a sequence of sub-optimizations is carried out where reweighting is performed only once in each sub-optimization. In the algorithm proposed here, the reweighting in (2.9) is performed in every iteration of each sub-optimization.

2.2.3 Solving the optimization problem using a quasi-Newton method

It can be readily verified that the region where function $F_\sigma(\boldsymbol{\xi})$ in (2.8) is convex is closely related to the value of parameter σ: the greater the value of σ, the larger the convex region. On the other hand, for $F_\sigma(\boldsymbol{\xi})$ to well approximate the ℓ_0-pseudonorm of $\boldsymbol{x}$, σ must be sufficiently small. For this reason, the solution of the optimization problem in (2.8) is obtained using a relatively large $\sigma = \sigma_0$. This solution is then used as the initial point for minimizing $F_\sigma(\boldsymbol{\xi})$ with a reduced value of σ, say, $r\cdot\sigma$ with $r < 1$. This procedure is repeated until function $F_\sigma(\boldsymbol{\xi})$ with $\sigma \leq \sigma_J$ is minimized where σ_J is a prescribed sufficiently small value of σ. For a fixed value of σ, the problem in (2.8) is solved by using a quasi-Newton algorithm where an approximation of the inverse of the Hessian is obtained by using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update formula [44]. We note that applying a quasi-Newton algorithm is particularly

convenient in the present application because the gradient of the objective function can be efficiently evaluated using the closed-form formulas in (2.7a) and (2.7b). As demonstrated in our simulation studies (see Sec. 2.2.5), the application of the BFGS quasi-Newton algorithm to the problem in (2.8) yields improved solutions relative to those that can be obtained by using the steepest-decent algorithm.

2.2.4 Algorithm

The proposed method for reconstructing a sparse signal $\boldsymbol{x}$ using a measurement $\boldsymbol{y} = \boldsymbol{\Phi x}$ can now be implemented in terms of the algorithm in Table 2.1. This will be referred to hereafter as the *null-space reweighted approximate ℓ_0-pseudonorm* (NRAL0) algorithm.

Table 2.1: The Null-Space Reweighted Approximate ℓ_0-Pseudonorm Algorithm

Step 1
Input $\boldsymbol{\Phi}$, $\boldsymbol{x}_{\ell_2}$, σ_T, r, and τ, and ϵ.
Step 2
Set $\boldsymbol{\xi}^{(0)} = \boldsymbol{0}$, $\boldsymbol{w}^{(0)} = \boldsymbol{e}_N$, $\sigma = \max\|\boldsymbol{x}_{\ell_2}\| + \tau$, and $k = 0$.
Step 3
Perform the QR decomposition $\boldsymbol{\Phi}^T = \boldsymbol{QR}$ and construct $\boldsymbol{V}_n$ using the last $N - M$ columns of $\boldsymbol{Q}$.
Step 4
With $\boldsymbol{w} = \boldsymbol{w}^{(k)}$ and $\boldsymbol{\xi}^{(0)}$ as an initial point, apply the BFGS algorithm to solve the problem in (2.8), where reweighting with parameter ϵ is applied using (2.9) in each iteration. Denote the solution as $\boldsymbol{\xi}^{(k)}$.
Step 5
Compute $\boldsymbol{x}^{(k)} = \boldsymbol{x}_{\ell_2} + \boldsymbol{V}_n\boldsymbol{\xi}^{(k)}$ and update weight vector to $\boldsymbol{w}^{(k+1)}$ using (2.9).
Step 6
If $\sigma \leq \sigma_T$, stop and output $\boldsymbol{x}^{(k)}$ as the solution; otherwise, set $\boldsymbol{\xi}^{(0)} = \boldsymbol{\xi}^{(k)}$, $\sigma = r \cdot \sigma$, $k = k + 1$, and repeat from Step 4.

In the algorithm, vector $\boldsymbol{x}_s$ is chosen to be the least-squares solution $\boldsymbol{x}_{\ell_2}$ of $\boldsymbol{\Phi x} = \boldsymbol{y}$, namely, $\boldsymbol{x}_s = \boldsymbol{x}_{\ell_2} = \boldsymbol{\Phi}^T\left(\boldsymbol{\Phi\Phi}^T\right)^{-1}\boldsymbol{y}$. Concerning the initial value of parameter σ, we note that function $F_\sigma(\boldsymbol{\xi})$ remains convex in the region where the largest magnitude of the components of $\boldsymbol{x} = \boldsymbol{x}_{\ell_2} + \boldsymbol{V}_n\boldsymbol{\xi}$ is less than σ. Based on this, a reasonable initial value of σ can be chosen as $\sigma_0 = \max|\boldsymbol{x}_{\ell_2}| + \tau$ where τ is a small positive scalar. As the algorithm starts at the origin $\boldsymbol{\xi}^{(0)} = \boldsymbol{0}$, the above choice of σ_0 assures that the

optimization starts in a convex region of the parameter space. This greatly facilitates the convergence of the proposed algorithm.

2.2.5 Simulation results

The performance of the proposed NRAL0 algorithm was investigated by conducting two numerical experiments to compare its signal reconstruction performance and computational complexity with that of several competing algorithms.

In the first experiment, the signal length and number of measurements were set to $N = 256$ and $M = 100$, respectively. A total of 15 sparse signals with sparsity $K = 5q - 4$, $q = 1, 2, \ldots, 15$ were used. A K-sparse signal $\boldsymbol{x}$ was constructed as follows: (1) set $\boldsymbol{x}$ to a zero vector of length N; (2) generate a vector $\boldsymbol{u}$ of length K assuming that each component u_i is a random value drawn from a normal distribution $\mathcal{N}(0,1)$; (3) randomly select K indices from the set $\{1, 2, \ldots, N\}$, say $i_1, i_2, \ldots, i_K$, and set $x_{i_1} = u_1, x_{i_2} = u_2, \ldots, x_{i_K} = u_K$. The measurement matrix was assumed to be of size $M \times N$ and was generated by drawing its elements from $\mathcal{N}(0,1)$, followed by a normalization step to ensure that the ℓ_2-norm of each column is unity. The measurement was obtained as $\boldsymbol{y} = \boldsymbol{\Phi x}$. The performance of the iteratively reweighted (IR) algorithm [34] with $p = 0.1$ and $p = 0$, the SL0 algorithm [47], and the proposed NRAL0 algorithm with $\sigma_T = 10^{-4}$, $r = 1/3$, $\tau = 0.01$, and $\epsilon = 0.09$ was measured in terms of the number of perfect reconstructions over 100 runs. The results obtained are plotted in Figure 2.1. It can be observed that the NRAL0 algorithm outperforms the IR algorithm. On comparing the NRAL0 algorithm with the SL0 algorithm, we note that the two algorithms are comparable for K smaller than 40 but the NRAL0 algorithm performs better for K larger than 40. The mathematical complexity of the four algorithms was measured in terms of the average CPU time over 100 runs for typical instances with $M = N/2$ and $K = \text{round}(M/2.5)$ where N varies in the range between 128 and 512. The CPU time was measured on a PC laptop with a Intel T5750 2 GHz processor using MATLAB commands *tic* and *tac*, and the results are plotted in Figure 2.2. It is noted that the NRAL0 and SL0 algorithms are more efficient than the IR algorithm, and the complexity of the NRAL0 algorithm is slightly higher than that of the SL0 algorithm. The moderate increase in the mathematical complexity of the NRAL0 algorithm is primarily due to the fact that the objective function in (2.8) needs to be modified in each iteration using (2.9).

In the second experiment, the four algorithms were evaluated by using sparse

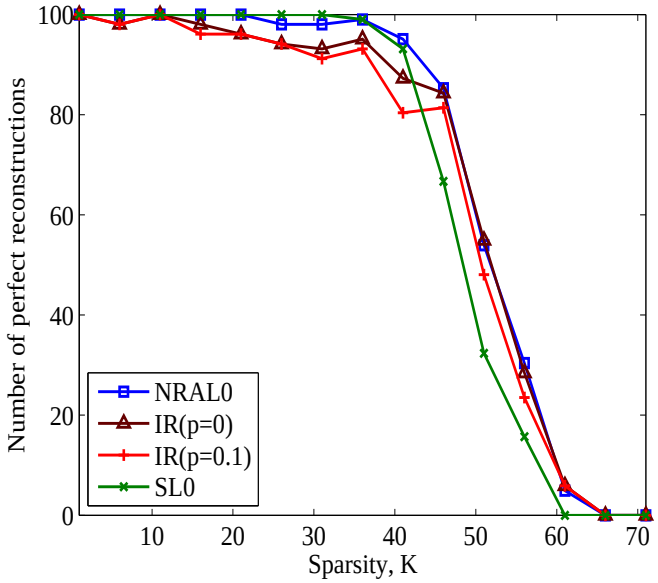


Figure 2.1: Number of perfect reconstructions by the NRAL0, IR, and SL0 algorithms over 100 runs with $N = 256$ and $M = 100$.

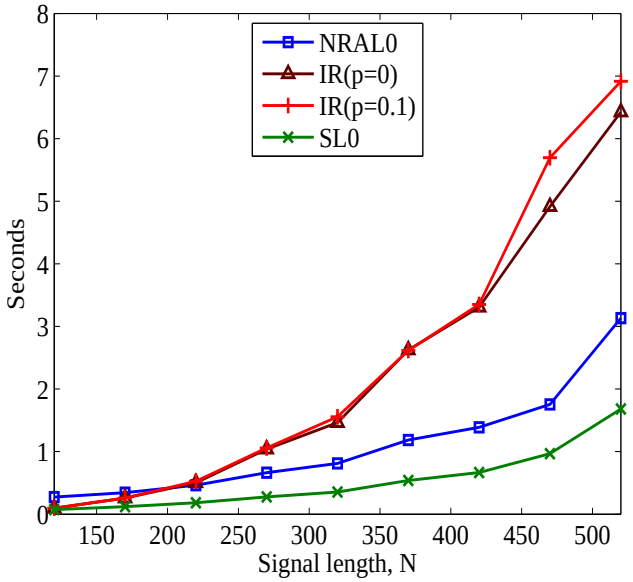


Figure 2.2: Average CPU time required by the NRAL0, IR, and SL0 algorithms over 100 runs with $M = N/2$ and $K = round(M/2.5)$.

signals with various values of N, M, and K so as to examine the algorithms' performance for signals of different lengths, measurement numbers, and sparsity levels. Specifically, the algorithms were first evaluated with $N = 512$ and $M = 200$ using

signals with sparsity $K = 70$, 90, and 110 and then with $N = 1024$ and $M = 400$, using signals with sparsity $K = 140$, 180, and 220. The results obtained are summarized in Table 2.4. It is observed that the performance of the NRAL0 algorithm is consistently better than that of the IR and SL0 algorithms in most cases.

Table 2.2: Number of Perfect Reconstructions for the IR, SL0, and NRAL0 algorithms for Various Values of N, M, and K over 100 Runs.

N/M	Algorithm	Number of perfect reconstructions		
		K=70	K=90	K=110
512/200	IR(p=0.1)	77	77	24
	IR(p=0)	85	67	21
	SL0	100	91	8
	NRAL0	100	96	28
		K=140	K=180	K=220
1024/400	IR(p=0.1)	65	49	16
	IR(p=0)	75	59	20
	SL0	100	94	2
	NRAL0	97	96	29

Typically the NRAL0 algorithm converges in a small number of iterations. As an example, Fig. 2.3 shows how the objective function $F_\sigma(\xi)$ in (2.6) converges in 18 iterations, where the parameters were set to $N = 256$, $M = 100$, $K = 40$, and $\sigma = 0.0218$.

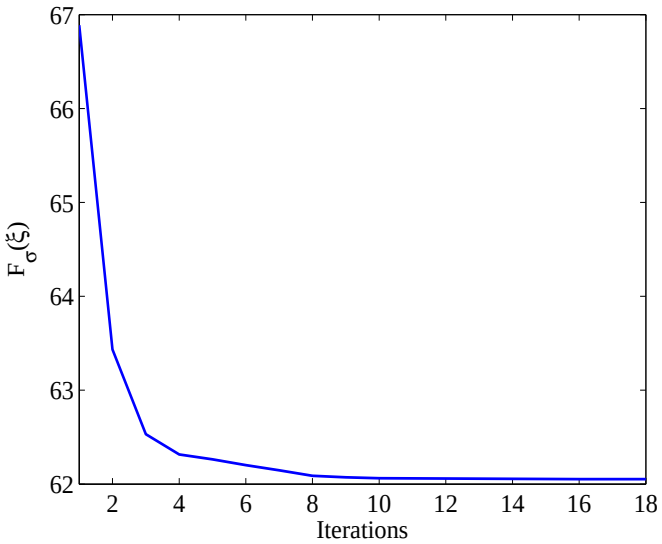


Figure 2.3: Function $F_\sigma(\boldsymbol{\xi})$ for $N = 256$, $M = 100$, $K = 40$, and $\sigma = 0.0218$.

2.3 Minimization of Approximate ℓ_p Pseudonorm Using a Quasi-Newton Algorithm

2.3.1 Problem formulation

Let us consider the approximate ℓ_p pseudonorm of $\boldsymbol{x}$, namely,

$$||\boldsymbol{x}||_{p,\epsilon}^p = \sum_{i=1}^{N} \left(x_i^2 + \epsilon^2\right)^{p/2} \tag{2.11}$$

where ϵ is a small approximation parameter and $p < 1$. Note that $||\boldsymbol{x}||_{p,\epsilon}^p \to ||\boldsymbol{x}||_p^p$ as $\epsilon \to 0$. Here we propose to reconstruct $\boldsymbol{x}$ by solving the constrained problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_{p,\epsilon}^p \tag{2.12a}$$

$$\text{subject to :} \quad \boldsymbol{\Phi x} = \boldsymbol{y} \tag{2.12b}$$

By using (2.5), the constraint in (2.12b) is eliminated and the problem at hand is converted into the unconstrained problem

$$\underset{\boldsymbol{\xi}}{\text{minimize}} \quad F_{p,\epsilon}(\boldsymbol{\xi}) = \sum_{i=1}^{N} \left[\left(x_{si} + \boldsymbol{v}_i^T \boldsymbol{\xi}\right)^2 + \epsilon^2\right]^{p/2} \tag{2.13}$$

Parameter ϵ plays two important roles in the proposed algorithm. First, the objective function $F_{p,\epsilon}(\boldsymbol{\xi})$ remains differentiable as long as ϵ is kept positive. In effect, for $\epsilon > 0$ the gradient of $F_{p,\epsilon}(\boldsymbol{\xi})$ is given by

$$\hat{\boldsymbol{g}} = p \cdot \boldsymbol{V}_n^T \cdot \boldsymbol{g} \tag{2.14}$$

where $\boldsymbol{g} = [g_1 \ g_2 \ \cdots \ g_N]^T$ and

$$g_i = \left[\left(x_{si} + \boldsymbol{v}_i^T \boldsymbol{\xi}\right)^2 + \epsilon^2\right]^{p/2-1} \left(x_{si} + \boldsymbol{v}_i^T \boldsymbol{\xi}\right) \tag{2.15}$$

Second, the region where function $F_{p,\epsilon}(\boldsymbol{\xi})$ in (2.13) is convex is controlled by ϵ: the greater the ϵ, the larger the region. To see this, note that the Hessian of $||\boldsymbol{x}||_{p,\epsilon}^p$ is a diagonal matrix given by

$$\boldsymbol{H} = \ \text{diag}\{h_{11}, \ h_{22}, \ \cdots, \ h_{NN}\} \tag{2.16}$$

where

$$h_{ii} = p(x_i^2 + \epsilon^2)^{p/2-1} \left[(p-1)x_i^2 + \epsilon^2\right] \tag{2.17}$$

Hence $||\boldsymbol{x}||_{p,\epsilon}^p$ is convex if and only if

$$|x_i| \leq \frac{\epsilon}{\sqrt{1-p}} \quad \text{for } 1 \leq i \leq N \tag{2.18}$$

Eq. (2.18) defines an N-dimensional hypercube whose volume is $\left(\frac{2\epsilon}{\sqrt{1-p}}\right)^N$. Therefore, for a fixed $p < 1$, the volume of the convex region in the $\boldsymbol{x}$ space is proportional to ϵ. Using $\boldsymbol{H}$ in (2.16), the Hessian of function $F_{p,\epsilon}(\boldsymbol{\xi})$ in (2.13) is found to be

$$\hat{\boldsymbol{H}} = \boldsymbol{V}_n^T \cdot \boldsymbol{H} \cdot \boldsymbol{V}_n$$

Hence $\hat{\boldsymbol{H}}$ is positive definite if $\boldsymbol{H}$ is positive definite. Consequently, we can use (2.18) and (2.5) to show that $F_{p,\epsilon}(\boldsymbol{\xi})$ is convex if

$$|\boldsymbol{v}_i^T \boldsymbol{\xi} + x_{si}| \leq \frac{\epsilon}{\sqrt{1-p}} \quad \text{for } 1 \leq i \leq N \tag{2.19}$$

From (2.19), we see that the size of the convex region of $F_{p,\epsilon}(\boldsymbol{\xi})$ is also proportional to the value of ϵ.

Based on the above analysis, we propose the following optimization technique for the problem in (2.13):

- First, we obtain the minimum ℓ_2-norm solution of $\boldsymbol{\Phi x} = \boldsymbol{y}$ and use it as the special solution $\boldsymbol{x}_s$ in (2.5). An initial value of ϵ is selected to satisfy the inequality

$$\epsilon \geq \sqrt{1-p} \cdot \max_{1 \leq i \leq N} |x_{si}| \tag{2.20}$$

  which will assure that

$$|x_{si}| \leq \frac{\epsilon}{\sqrt{1-p}} \quad \text{for } 1 \leq i \leq N \tag{2.21}$$

  It follows from (2.18) that such an $\boldsymbol{x}_s$ is in the convex region of $||\boldsymbol{x}||_{p,\epsilon}^p$. Now, (2.21) in conjunction with (2.19) implies that $\boldsymbol{\xi} = 0$ is in the convex region of $F_{p,\epsilon}(\boldsymbol{\xi})$. This justifies the choice of ϵ according to (2.20) and the use of $\boldsymbol{\xi} = 0$ as an initial point. A quasi-Newton algorithm such as the BFGS algorithm [44] [50] [51] is applied to minimize $F_{p,\epsilon}(\boldsymbol{\xi})$. The minimizer obtained is denoted as

$\boldsymbol{\xi}_1^*$.

- Next, the value of ϵ is reduced by a certain amount and the BFGS algorithm is applied again to minimize $F_{p,\epsilon}(\boldsymbol{\xi})$ with $\boldsymbol{\xi}_1^*$ as initial point. The minimizer obtained is denoted as $\boldsymbol{\xi}_2^*$; it is used as initial point in the next iteration.

- This procedure is repeated until the value of ϵ is reduced to a prescribed target value ϵ_J. The minimizer associated with $\epsilon = \epsilon_J$ is denoted by $\boldsymbol{\xi}_J^*$ and is used in (2.5) to reconstruct the sparse signal as $\boldsymbol{x}^* = \boldsymbol{x}_s + \boldsymbol{V}_n\boldsymbol{\xi}_J^*$.

The efficacy of the sequential optimization procedure described above is illustrated by using the following example.

Example: As an illustrative example, we consider a problem of recovering a sparse signal $\boldsymbol{x}$ of length two from measurement $\boldsymbol{\Phi x} = \boldsymbol{y}$, where $\boldsymbol{\Phi} = [0.1\ 0.06]$ and $y = 0.1$. The least-squares solution of $\boldsymbol{\Phi x} = \boldsymbol{y}$ is found to be $\boldsymbol{x}_s = [0.7353\ 0.4412]^T$ and the basis vector of the null space of $\boldsymbol{\Phi}$ is given by $\boldsymbol{v}_n = [-0.5145\ 0.8575]^T$. Using these values of $\boldsymbol{x}_s$ and $\boldsymbol{v}_n$, the objective function $F_{p,\epsilon}(\xi)$ in (2.13) can be expressed as

$$F_{p,\epsilon}(\xi) = \left[(0.7353 - 0.5145\xi)^2 + \epsilon^2\right]^{p/2} + \left[(0.4412 + 0.8575\xi)^2 + \epsilon^2\right]^{p/2}$$

where $-\infty < \xi < \infty$. Function $F_{p,\epsilon}(\xi)$ is plotted in Fig. 2.4 for $\xi \in [-1, 2]$ for $\epsilon = 0.8$, 0.5, 0.3, 0.15, 0.06, and 0. As can be seen in the plot for $\epsilon = 0$, the objective function has two minima at $\xi = -0.514$ and 1.43 where the minimum at $\xi = -0.514$ is the global minimizer. Obviously, a gradient based descent algorithm would converge to either minimum of function $F_{p,0}(\xi)$ depending on the location of the initial point. From Fig. 2.4, we see that if we use the minimizer of $F_{p,\epsilon}(\boldsymbol{\xi})$ for $\epsilon = 0.8$ as an initial point to minimize $F_{p,\epsilon}(\boldsymbol{\xi})$ for a slightly reduced value of ϵ, say $\epsilon = 0.5$, and use the minimizer as an initial point to minimize $F_{p,\epsilon}(\boldsymbol{\xi})$ for a smaller ϵ, say $\epsilon = 0.3$, and so on, the algorithm is likely to lead to the global solution of $F_{p,\epsilon}(\boldsymbol{\xi})$ for $\epsilon = 0$.

2.3.2 Line search based on Banach's fixed-point theorem

Given parameters p and ϵ, the kth iteration of the BFGS algorithm involves a step to solve the one-dimensional optimization problem

$$\underset{\alpha}{\text{minimize}} \quad f(\alpha) \tag{2.22}$$

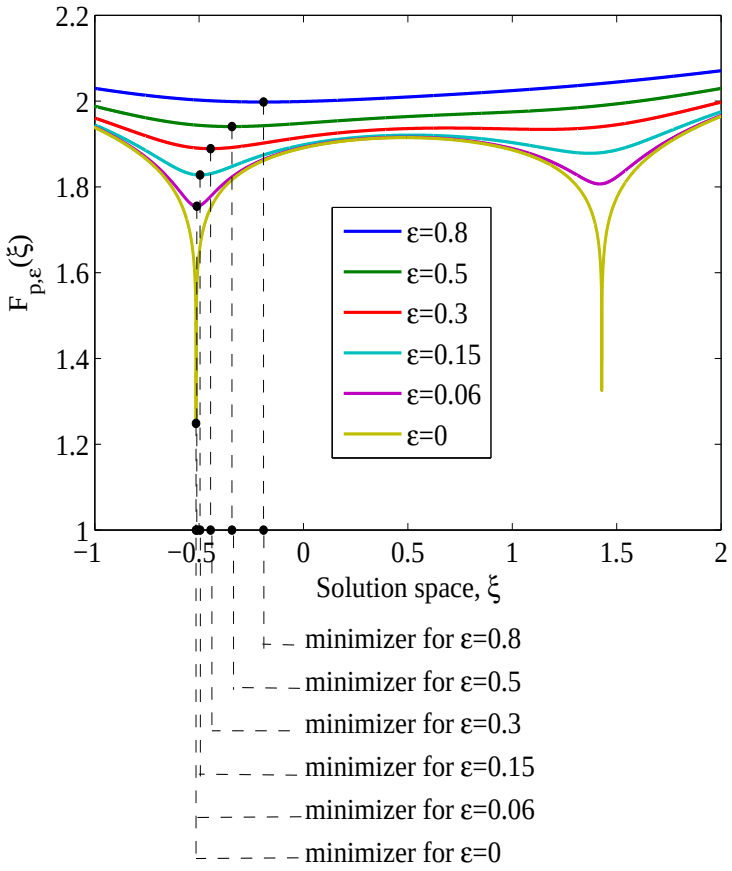


Figure 2.4: Value of function $F_{p,\epsilon}(\xi)$ in the solution space.

where

$$f(\alpha) = F_{p,\epsilon}(\boldsymbol{\xi}_k + \alpha \boldsymbol{d}_k)$$

Iterate $\boldsymbol{\xi}_k$ and search direction $\boldsymbol{d}_k$ are determined by using the BFGS algorithm. This is essentially a line search and its performance would have a great effect on the efficiency of the algorithm and the accuracy of the solution. Below we propose a new line search based on Banach's fixed-point theorem [45], which turns out to work very well for the optimization problem in (2.13). In the rest of this section, an α is said to be a *fixed point* for function $G(\alpha)$ if $\alpha = G(\alpha)$.

If $f(\alpha)$ in (2.22) is convex over a region of interest, R_a, and if it has a stationary point α^* in R_a, then α^* is a local minimizer of the problem in (2.22). The minimizer

α^* can be obtained by solving the equation

$$f'(\alpha) = 0 \tag{2.23}$$

where

$$\begin{aligned} f'(\alpha) &= \frac{dF_{p,\epsilon}(\boldsymbol{\xi}_k + \alpha \boldsymbol{d}_k)}{d\alpha} \\ &= p \sum_{i=1}^{N} \left[\gamma_i(\alpha, \epsilon)^{p/2-1} \cdot (x_i + \alpha v_i) \cdot v_i \right] \end{aligned} \tag{2.24}$$

and

$$\gamma_i(\alpha, \epsilon) = (x_i + \alpha v_i)^2 + \epsilon^2 \tag{2.25}$$

with $x_i = x_{si} + \boldsymbol{v}_i^T \boldsymbol{\xi}_k$ and $v_i = \boldsymbol{v}_i^T \boldsymbol{d}_k$.

Using (2.24), (2.23) can be written as

$$\alpha = G(\alpha) \tag{2.26}$$

where

$$G_\epsilon(\alpha) = -\frac{\sum_{i=1}^{N} x_i \cdot v_i \cdot \gamma_i(\alpha, \epsilon)^{p/2-1}}{\sum_{i=1}^{N} v_i^2 \cdot \gamma_i(\alpha, \epsilon)^{p/2-1}} \tag{2.27}$$

Therefore, finding a minimizer of $f(\alpha)$ amounts to finding a fixed point of function $G_\epsilon(\alpha)$. The well-known Banach's fixed-point theorem [45] states that if $G_\epsilon(\alpha)$ is a contraction mapping, i.e.,

$$|G_\epsilon(\alpha_1) - G_\epsilon(\alpha_2)| \leq \eta |\alpha_1 - \alpha_2| \quad \text{for any } \alpha_1, \alpha_2 \tag{2.28}$$

with $\eta < 1$, then there exists a fixed point α^* for function $G_\epsilon(\alpha)$ and α^* can be obtained as the limiting point of sequence $\{\alpha_l, l = 1, 2, \ldots\}$ which can be generated using the recursive relation

$$\alpha_{l+1} = G_\epsilon(\alpha_l) \quad \text{for } l = 1, 2, \ldots \tag{2.29}$$

Therefore, an approximate solution of (2.23) can be found by using a sufficient number of recursions of (2.29). It can be shown that $G_\epsilon(\alpha)$ satisfies the condition in (2.28) if

the magnitude of $G'_\epsilon(\alpha)$ is strictly bounded from above by unity, i.e., $|G'_\epsilon(\alpha)| \leq \eta < 1$ [45] [46]. For function $G_\epsilon(\alpha)$ in (2.27), theoretical verification of the condition imposed on $G'_\epsilon(\alpha)$ turns out to be difficult. Nevertheless, as far as the problem in (2.13) is concerned, the condition was found to be satisfied in our experiments.

2.3.3 Algorithm

The proposed unconstrained approximate ℓ_p-pseudonorm (UALP) based algorithm is described in Table 3.1. The input data for the algorithm include a value of $p < 1$, an initial value ϵ_1 satisfying (2.20), a target value ϵ_J, and the number of iterations T. The algorithm requires the minimum ℓ_2-norm solution $\boldsymbol{x}_{\ell_2}$ and matrix $\boldsymbol{V}_n$ (see (2.13) and (2.14)) which can be computed using the QR decomposition of $\boldsymbol{\Phi}^T$ (See Eq. (A.4) in Appendix A).

The $J-2$ values of ϵ for which the optimization formulated in (2.13) is carried out are set between the initial value ϵ_0 and target value ϵ_T as

$$\epsilon_i = e^{-\beta i} \quad \text{for } i = 2, 3, \ldots, T-1 \tag{2.30}$$

where $\beta = \log(\epsilon_1/\epsilon_T)/(T-1)$.

Table 2.3: UALP Algorithm

Step 1
Input: p, ϵ_1, ϵ_T, and T.
Set $\boldsymbol{\xi}^{(1)} = \boldsymbol{0}$.
Step 2
Compute ϵ_i for $i = 2, 3, \ldots, T-1$ using (2.30).
Step 3
Use (A.4) to compute $\boldsymbol{x}_{\ell_2}$ and $\boldsymbol{V}_n$.
Step 4
Repeat the following for $k = 1, \ldots, T$
i) Set $\epsilon = \epsilon_k$ and use $\boldsymbol{\xi}^{(k)}$ as an initial point. Apply the BFGS algorithm to solve the problem in (2.13). Denote the solution as $\boldsymbol{\xi}^{(k+1)}$.
Step 5
Set $\boldsymbol{\xi}^* = \boldsymbol{\xi}^{(T+1)}$, $\boldsymbol{x} = \boldsymbol{x}_{\ell_2} + \boldsymbol{V}_n \boldsymbol{\xi}^*$, and stop.

The line search based on Banach's fixed-point theorem (see Sec. 2.3.2) is used in Step 4 of the algorithm. Starting from an initial value $\alpha = 0$, α is iteratively computed using (2.29). The details of the line-search are summarized in Table 2.4.

Table 2.4: Line Search Based on Banach's Fixed-Point Theorem

Step 1
Input: $\boldsymbol{\xi}_k$, $\boldsymbol{x}_{\ell_2}$, $\boldsymbol{V}_n$, $\boldsymbol{d}_k$, δ_t, and ϵ_k.
Set $l = 1$, $\alpha_1 = 0$, and $\delta_\alpha = \delta_t + 1$.
Step 2
Repeat the following until $\delta_\alpha < \delta_t$
i) Compute $G_\epsilon(\alpha_l)$ using (2.27).
ii) Set $\alpha_{l+1} = G_\epsilon(\alpha_l)$.
iii) $\delta_\alpha = \alpha_{l+1} - \alpha_l$.
iv) $l = l + 1$.
Step 3
Set $\alpha = \alpha_l$ and stop.

2.3.4 Simulation results

The performance of the proposed UALP algorithm was investigated by conducting three numerical experiments.

In the first experiment, the signal length N and the number of measurements M were set to 256 and 100, respectively. A total of fifteen values of sparsity K were chosen from 1 to 71 with an increment of 5. A K-sparse signal was constructed by assigning K random values drawn from a normal distribution $\mathcal{N}(0, 1)$ to K randomly selected locations of the zero vector of length N. Measurement matrix $\boldsymbol{\Phi}$ of size $M \times N$ was constructed by drawing its elements from $\mathcal{N}(0, 1)$ followed by a normalization step where each column was normalized to the unit 2 norm. The measurement was obtained as $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}$. With $p = 0.1$, $\epsilon_1 = \sqrt{1-p} \cdot \min_{1 \le i \le N} |x_{si}|$, $\epsilon_J = 10^{-5}$, and $J = 9$, the UALP algorithm was applied and compared with the BP [52], iterative reweighted (IR) [34], smoothed ℓ_0 pseudonorm (SL0) [47], and the null-space reweighted approximate ℓ_0 pseudonorm (NRAL0) algorithms. In the IR algorithm p was set to 0.1. For each algorithm, the signal reconstruction was deemed perfect if the largest magnitude of the components of the reconstruction error vector $\hat{\boldsymbol{x}} - \boldsymbol{x}$ was less than 10^{-5} where $\hat{\boldsymbol{x}}$ and $\boldsymbol{x}$ are the reconstructed and test signals, respectively. For each value of K, the perfect reconstructions were counted over 100 runs. The results are plotted in Figure 2.5. It is observed that the performance of the UALP algorithm is better than that of the competing algorithms.

In the second experiment, the average CPU time required by the algorithms to converge was measured over 100 runs for typical instances with $M = N/2$ and $K = round(M/2.5)$ where N varied in the range between 128 to 512. The CPU time

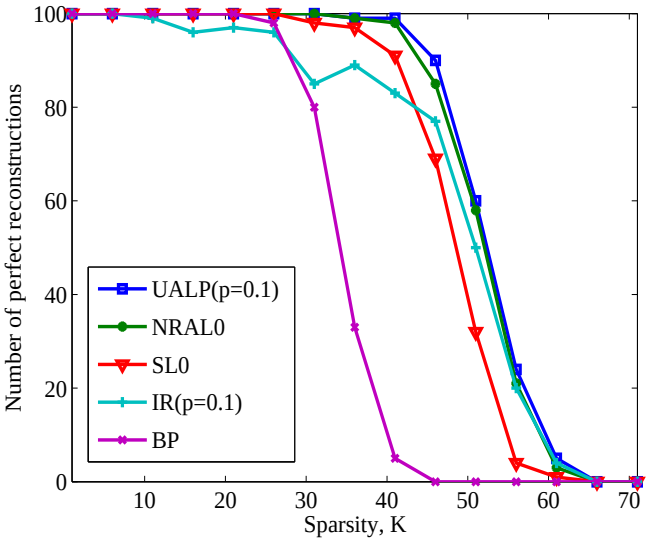


Figure 2.5: Number of perfect reconstructions for UALP, NRAL0, SL0, IR, and BP algorithms over 100 runs with $N = 256$, $M = 100$.

was measured using a PC desktop with Intel Core 2 CPU 6400 2.13 GHz processor using MATLAB command *cputime*. The results shown in Figure 2.6 indicate that the UALP algorithm requires the least amount of computation among the algorithms tested.

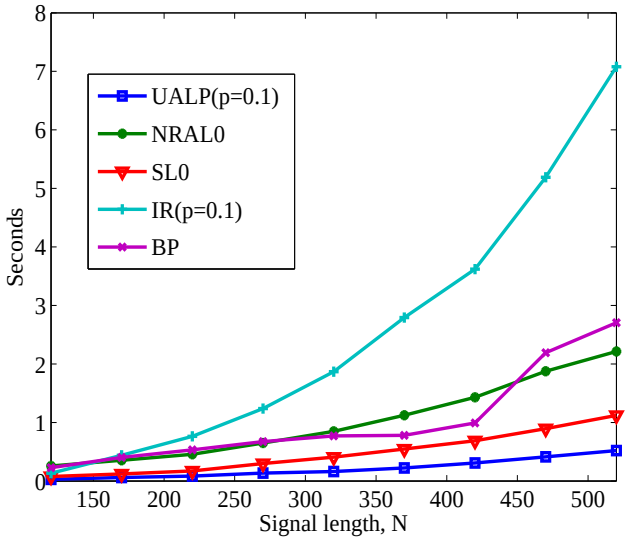


Figure 2.6: Average CPU time required for UALP, NRAL0, SL0, IR, and BP algorithms over 100 runs with $M = N/2$, $K = M/2.5$.

In the third experiment, the UALP algorithm was run for the same settings as for the first experiment with (a) the proposed line search based on Banach's fixed-point theorem and (b) Fletcher's inexact line search [44]. The reconstruction performance of the proposed algorithm for the two line searches was found to be the same as before, namely, the number of perfect reconstructions for the UALP algorithm shown in Fig. 2.5. Next, the UALP algorithm was run for the same settings as the second experiment with both line searches. The CPU times required in the two cases for $N = 120$, 220, 320, 420, and 520 are given in Table 2.5.

Table 2.5: CPU Time Required by the UALP Algorithm with the Proposed Line Search and Fletcher's Inexact Line Search, in Seconds

Signal length, N	120	220	320	420	520
Proposed line search	0.1218	0.2824	0.4394	0.6752	1.0678
Inexact line search	0.3844	0.5799	0.9009	1.4882	2.4240

It is noted that the amount of computation required by the UALP algorithm using the proposed line search based on Banach's fixed-point theorem is less than half of that required when the Fletcher's inexact line search is used. This demonstrates the crucial role of the proposed line search in reducing the numerical complexity of the UALP algorithm.

Typically, the UALP algorithm converges in a small number of iterations. As an example, Fig. 2.7 shows how the objective function $F_{p,\epsilon}(\xi)$ in (2.13) converges in 30 iterations, where the parameters were set to $N = 256$, $M = 100$, $K = 40$, and $\sigma = 0.0118$.

2.4 Minimization of Approximate ℓ_p Pseudonorm Using Conjugate-Gradient Algorithm

Recall that the introduction of parameter ϵ into function $F_{p,\epsilon}(\boldsymbol{\xi})$ makes it smooth and leads to a sequential optimization approach to solve the problem in (2.13). In this section, a conjugate-gradient algorithm is used to minimize function $F_{p,\epsilon}(\boldsymbol{\xi})$. The sequential optimization procedure used in the algorithm is outlined as follows.

- Select ϵ using (2.20), set $\boldsymbol{x}_s$ to the least-squares solution of $\boldsymbol{\Phi x} = \boldsymbol{y}$, initialize $\boldsymbol{\xi}$ to the zero vector of length $N - M$, and solve the problem in (2.13) using a conjugate-gradient technique. Denote the solution as $\boldsymbol{\xi}^*$.

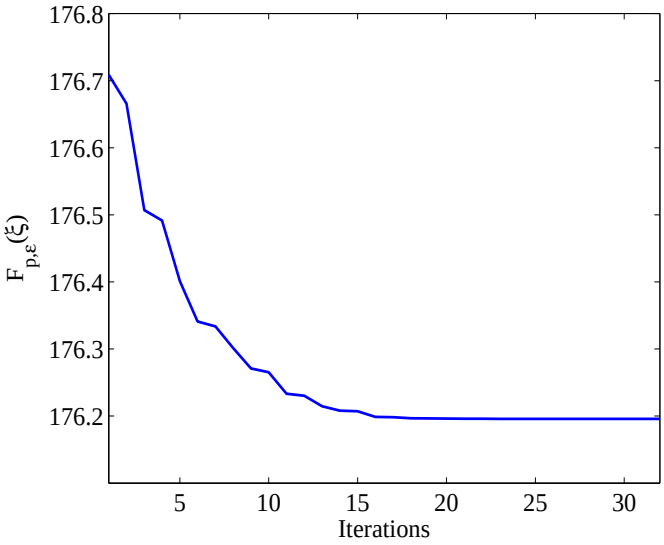


Figure 2.7: Function $F_{p,\epsilon}(\boldsymbol{\xi})$ for $N = 256$, $M = 100$, $K = 40$, and $\epsilon = 0.0118$.

- Update the solution $\boldsymbol{x}_s$ as $\boldsymbol{x}_s = \boldsymbol{x}_s + \boldsymbol{V}_n\boldsymbol{\xi}^*$, reduce the value of ϵ, set $\boldsymbol{\xi}$ to the zero vector, and solve the problem in (2.13) again.
- Continue this procedure until the problem in (2.13) is solved for a sufficiently small value of ϵ.
- Output $\boldsymbol{x}_s$ as the solution and stop.

In the kth iteration of the conjugate-gradient technique, iterate $\boldsymbol{\xi}_k$ is updated as

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_k + \alpha_k \boldsymbol{d}_k \tag{2.31}$$

for $k = 0, 1, \ldots, L-1$, where $\boldsymbol{\xi}_1 = \boldsymbol{0}$ and $\alpha_0 = 1$. The kth step size is computed using

$$\alpha_k = \frac{\alpha_n}{\alpha_d} \tag{2.32}$$

$$\alpha_n = \boldsymbol{g}_k^T \boldsymbol{g}_k \quad \text{and} \quad \alpha_d = \boldsymbol{d}_k^T \boldsymbol{H}_k \boldsymbol{d}_k \tag{2.33}$$

where $\boldsymbol{g}_k$ is the gradient vector computed using (2.14) and $\boldsymbol{H}_k$ is the Hessian matrix obtained using (2.3.1), respectively, with $\boldsymbol{x} = \boldsymbol{x}_k$.

The conjugate directions are computed as

$$\boldsymbol{d}_k = \begin{cases} -\boldsymbol{g}_0 & \text{for} \quad k = 0 \\ -\boldsymbol{g}_k + \beta_{k-1}\boldsymbol{d}_{k-1} & \text{for} \quad k = 1, 2, \ldots, L-1 \end{cases} \tag{2.34}$$

where

$$\beta_k = \frac{\beta_n}{\beta_d} \tag{2.35}$$

and

$$\beta_n = \boldsymbol{g}_{k+1}^T \boldsymbol{g}_{k+1} \quad \text{and} \quad \beta_d = \boldsymbol{g}_k^T \boldsymbol{g}_k \tag{2.36}$$

for $k = 0, 1, \ldots, L-2$.

2.4.1 Algorithm

The unconstrained approximate ℓ_p pseudonorm conjugate-gradient (UALP-CG) algorithm is summarized in Table 2.6. The algorithm takes the minimum ℓ_2 norm solution $\boldsymbol{x}_{\ell_2}$ of $\boldsymbol{\Phi x} = \boldsymbol{y}$ and matrix $\boldsymbol{V}_n$ as inputs which can be computed using the QR decomposition (see Appendix A).

The target value ϵ_T of ϵ, number of outer iterations T, and the parameters E_r and δ are supplied as input. The initial value ϵ_1 of ϵ is determined using (2.20) and a total of $T-2$ values lying between ϵ_1 and ϵ_T are computed by using (2.30).

The Hessian matrix $\hat{\boldsymbol{H}}_k$ which is computed using (2.16) in each step of the CG algorithm is made positive definite by setting the diagonal elements h_{ii} of matrix $\boldsymbol{H}$ in (2.16) as

$$h_{ii} = \begin{cases} h_{ii} & \text{if} \quad h_{ii} > \delta \\ \delta & \text{if} \quad h_{ii} \leq \delta \end{cases} \tag{2.37}$$

where δ is a small positive scalar.

The computation of the denominator term α_d in (2.32) is carried out in an efficient was as

$$\begin{aligned} \alpha_d &= \boldsymbol{d}_k^T \hat{\boldsymbol{H}}_k \boldsymbol{d} \\ &= \boldsymbol{d}_k^T \boldsymbol{V}_n^T \boldsymbol{H}^{1/2} \boldsymbol{H}^{1/2} \boldsymbol{V}_n \boldsymbol{d}_k \\ &= \left|\left| \boldsymbol{H}^{1/2} \boldsymbol{V}_n \boldsymbol{d}_k \right|\right|_2^2 \end{aligned} \tag{2.38}$$

where $\boldsymbol{H}^{1/2}$ is a diagonal matrix whose $\{i,i\}$th element is the square root of the

$\{i, i\}$th element of the matrix $\boldsymbol{H}$ in (2.16).

Table 2.6: The UALP-CG Algorithm

Step 1
Input p, T, ϵ_T, $\boldsymbol{\Phi}$, $\boldsymbol{y}$, E_t.
Step 2
Obtain vector $\boldsymbol{x}_{\ell_2}$ and matrix $\boldsymbol{V}_n$ using the QR decomposition.
Step 3
Set $\boldsymbol{x}_s = \boldsymbol{x}_{\ell_2}$.
Step 4
Compute: ϵ_1 using (2.20) and
ϵ_t for $t = 2, 3, \ldots, T-1$ using (2.30).
Step 5
For $t = 1, 2, \ldots, T$
i) Set $\epsilon = \epsilon_t$, $L_t = 3 + round(t/4)$.
ii) Set $k = 0$, $\boldsymbol{\xi}_0 = \boldsymbol{0}$, and $E_r = 10^{10}$.
iii) While $E_r > E_t$.
a) Compute $\boldsymbol{g}_k$ using (2.14).
b) Compute $\boldsymbol{d}_k$ using (2.34).
c) Compute α_k using (2.32) in conjunction with (2.38).
d) Compute $\boldsymbol{\xi}_{k+1}$ using (2.31).
e) Set $k = k + 1$.
f) Exit loop if $k > L_t$.
g) Compute $E_r = ||\alpha_k \boldsymbol{d}_k||_2$.
vii) Set $\boldsymbol{x}_s = \boldsymbol{x}_s + \boldsymbol{V}_n \boldsymbol{\xi}_k$.
Step 6
Output $\boldsymbol{x}^* = \boldsymbol{x}_s$ and stop.

2.4.2 Simulation results

The performance of the UALP-CG algorithm was investigated by carrying out two numerical experiments to compare its signal reconstruction performance and computational complexity with that of several competing algorithms.

In the first experiment, we set the signal length, N, to 512, the number of measurements, M, to 200, and vary the sparsity, K, from 1 to 121 with an increment of 5. A K-sparse signal was constructed as follows: i) a vector $\boldsymbol{x}$ of length N with all zero components was constructed, ii) a random vector of length K was constructed by drawing its components from $\mathcal{N}(0, 1)$, and iii) the components of the resulting vector were set to randomly chosen K locations of vector $\boldsymbol{x}$. Measurement matrix $\boldsymbol{\Phi}$ was constructed by drawing its elements from $\mathcal{N}(0, 1)$ followed by an orthonormalization step

ensuring that its rows are orthonormal to each other. The measurement was taken as $\boldsymbol{y} = \boldsymbol{\Phi x}$. The UALP-CG algorithm was run with $p = 0.1$, $\tau = 0.01$, $\epsilon_T = 1e-5$, $T = 80$, $E_t = 1e-25$, and $\delta = 1e-5$. It was applied to reconstruct signal $\boldsymbol{x}$ from $\boldsymbol{y}$. Reconstructions were deemed perfect when the maximum absolute valued error between the original $\boldsymbol{x}$ and reconstructed signal $\hat{\boldsymbol{x}}$ measured as $\underset{i}{\text{maximum}}\ |x_i - \hat{x}_i|$ was less than $1e-5$, where x_i and $\hat{x}_i$ are the ith components of vectors $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, respectively. The number of perfect reconstructions for the UALP-CG algorithm in 400 runs is compared with that for the BP, IR with $p = 0.1$, and SL0 algorithms in Figure 2.8a. As can be seen, the proposed UALP-CG algorithm outperformed other algorithms.

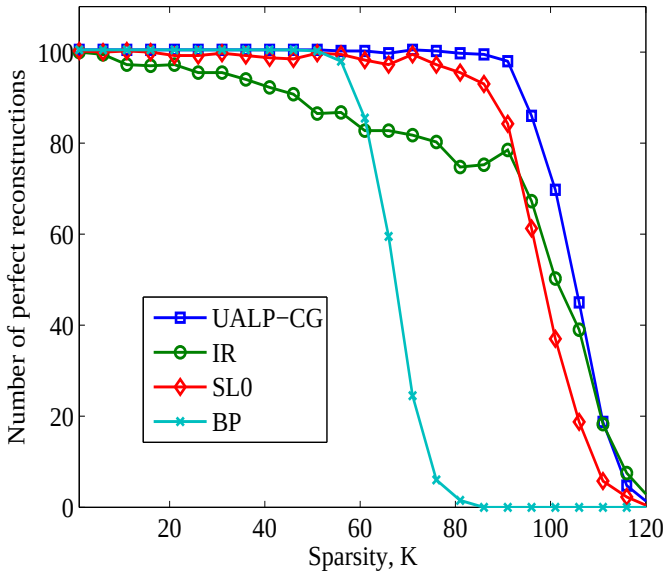
In the second experiment, the CPU time required by the UALP-CG, SL0, IR, and BP algorithms for $N = 200, 400, \ldots, 2000$ with $M = N/2$ and $K = round(M/2.5)$ were measured using Matlab function *cputime*. The average CPU time required by each algorithm over 100 runs is plotted in Figure 2.8b. We observe that the UALP-CG algorithm requires less CPU time than the other algorithms.

2.5 Conclusions

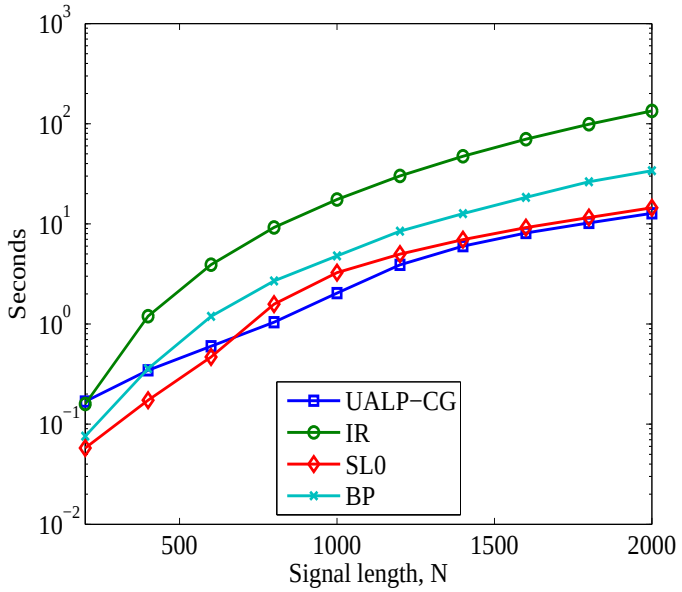
In this chapter, we have presented three algorithms for the reconstruction of sparse signals. First, the NRAL0 algorithm was developed by recasting a constrained optimization problem for the minimization of an approximate ℓ_0 pseudonorm to an unconstrained problem and applying a quasi-Newton algorithm to solve it. The constrained problem was converted to the unconstrained by working in the null space of the measurement matrix. In the quasi-Newton algorithm, the Hessian matrix was estimated by using the BFGS update formula [44] [50] [51]. A reweighting technique was also used to force the solution's sparsity.

Second, the UALP algorithm was developed by recasting a constrained optimization problem for the minimization of an approximate ℓ_p pseudonorm to an unconstrained problem and applying a quasi-Newton algorithm to solve it. The conversion of the problem from constrained to unconstrained was carried out by working in the null space. The resulting unconstrained optimization was carried out using a quasi-Newton algorithm where the Hessian matrix was estimated using the BFGS update formula. The step size used in each iteration of the optimization was computed by using a new line search based on Banach's fixed-point theorem [45] [46].

Third, the UALP-CG algorithm was developed by applying a conjugate-gradient



(a) Number of perfect reconstructions with $N = 512$ and $M = 200$ over 400 runs.



(b) Average CPU time with $M = N/2$ and $K = round(M/2.5)$ over 100 runs.

Figure 2.8: Number of perfect reconstructions and average CPU time required for the UALP-CG, SL0, IR, and BP algorithms.

technique to minimize the approximate ℓ_p pseudonorm in the null space of the measurement matrix. The step sizes used in the conjugate-gradient technique were com-

puted by using a simple closed-form formula.

Simulation results were presented which demonstrate that a) the NRAL0 algorithm yields improved signal reconstruction performance compared to the IR [34], SL0 [47], and BP [52] algorithms and requires a reduced amount of computation compared to the IR and BP algorithms. Compared to the SL0 algorithm, the NRAL0 algorithm requires slightly more computation; b) the UALP algorithm yields improved signal reconstruction performance and requires a reduced amount of computation relative to the IR, BP, NRAL0, and SL0 algorithms; and c) the UALP-CG algorithm yields improved signal reconstruction performance compared to the IR, SL0, and BP algorithms, and requires a reduced amount of computation compared to the IR and BP algorithms and slightly less computation compared to the SL0 algorithm for large sized data. Also, the new line search used in the UALP algorithm was shown to be numerically more efficient than Fletcher's inexact line search [44] [50] [51].

Chapter 3

Reconstruction of Sparse Signals from Noisy Measurements by Using Least-Squares Optimization

3.1 Introduction

The algorithms presented in Chapter 2 work well when measurements are noise free and signals are strictly sparse. However, we also need to deal with signals that are approximately sparse and measurements that are noisy. Such situations arise because a) signals are often not strictly sparse but approximately sparse, i.e., they have a few significant components and a large number of insignificant components and b) the devices used for the measurement do not have an accuracy of infinite precision and they add at least a small amount of noise in the measurements. An early and successful attempt to deal with signal reconstruction from noisy measurements is the ℓ_1-norm minimization based basis pursuit for denoising (BPDN) algorithm [52, 53].

In this chapter, three algorithms for the reconstruction of sparse signals from noisy and noiseless measurements are presented. The first algorithm is based on the minimization of an ℓ_p-pseudonorm regularized ℓ_2 error, with $p < 1$, by working in the null space of the measurement matrix and its complement space. In the second algorithm, the signal reconstruction problem is solved directly as an unconstrained problem using a conjugate-gradient (CG) algorithm; an algorithm for the optimization of the involved regularization parameter is also proposed. Finally, the well-known concept of total variation (TV) is generalized to a pth-power total variation with $p <$

1, denoted as TV_p, and the third algorithm of the chapter based on the minimization of a TV_p-regularized ℓ_2 objective function is presented.

3.2 ℓ_p-Regularized Least-Squares Optimization in Null and Complement Spaces

Recall from Sec. 1.2.1.2 in Chapter 1 that a CS measurement model that takes the measurement noise into account is given by

$$\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{w} \tag{3.1}$$

where $\boldsymbol{w}$ represents measurement noise. In such case, we seek to determine the sparsest $\boldsymbol{x}$ that would satisfy (3.1), which can be done by solving the optimization problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_{p,\epsilon}^{p} \tag{3.2a}$$

$$\text{subject to} \quad \boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{w} \tag{3.2b}$$

with $p < 1$, where $||\boldsymbol{x}||_{p,\epsilon}^{p}$ is an approximate ℓ_p pseudonorm of $\boldsymbol{x}$ defined as

$$||\boldsymbol{x}||_{p,\epsilon}^{p} = \sum_{i=1}^{N} (x_i^2 + \epsilon^2)^{p/2} \tag{3.3}$$

and ϵ is a small scalar.

If we assume that the components of vector $\boldsymbol{w}$ are independent identically distributed (i.i.d.) Gaussian random variables with zero mean and variance σ^2, then we have $||\boldsymbol{w}||_2^2 \approx M\sigma^2$ and the problem at hand can be formulated as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_{p,\epsilon}^{p} \tag{3.4a}$$

$$\text{subject to :} \quad ||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2 \leq \upsilon \tag{3.4b}$$

where υ is a constant that depends on the noise variance. This problem can be studied by converting the problem in (3.4) into an unconstrained problem as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad F_{p,\epsilon}(\boldsymbol{x}) = \frac{1}{2} ||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2 + \lambda ||\boldsymbol{x}||_{p,\epsilon}^{p} \tag{3.5}$$

where $\lambda > 0$ is a regularization parameter. In what follows, function $F_{p,\epsilon}(\boldsymbol{x})$ will be referred to as the $\ell_2, \ell_{p,\epsilon}$ objective function if $\epsilon > 0$ and the ℓ_2, ℓ_p objective function if $\epsilon = 0$.

3.2.1 Problem formulation

Let $\boldsymbol{\Phi} = \boldsymbol{U}\,[\boldsymbol{\Sigma}\ \boldsymbol{0}]\,\boldsymbol{V}^T$ be the singular-value decomposition (SVD) of $\boldsymbol{\Phi}$ (see Appendix A). Now let $\boldsymbol{V} = [\boldsymbol{V}_r\ \boldsymbol{V}_n]$ where the columns of matrix $\boldsymbol{V}_n$ span the null space of $\boldsymbol{\Phi}$ and the columns of $\boldsymbol{V}_r$ span the orthogonal complement of the null space. By using $\boldsymbol{V}_r$ and $\boldsymbol{V}_n$, we can express a signal $\boldsymbol{x}$ of length N as

$$\boldsymbol{x} = \boldsymbol{V}_r\boldsymbol{\phi} + \boldsymbol{V}_n\boldsymbol{\xi} \tag{3.6}$$

where $\boldsymbol{\phi}$ and $\boldsymbol{\xi}$ are vectors of length M and $N-M$, respectively. When measurement vector $\boldsymbol{y}$ is not corrupted by noise, vector $\boldsymbol{\phi}$ can be evaluated as

$$\boldsymbol{\phi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T\boldsymbol{y} \tag{3.7}$$

If measurement $\boldsymbol{y}$ is corrupted by noise, then vector $\boldsymbol{\phi}$ obtained from (3.7) is not optimal in general and we shall consider both $\boldsymbol{\phi}$ and $\boldsymbol{\xi}$ as independent variables. Using the SVD of $\boldsymbol{\Phi}$, we can simplify the ℓ_2 term in (3.5) as

$$\frac{1}{2}\,||\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}||_2^2 = \frac{1}{2}\,||\boldsymbol{\Sigma}\boldsymbol{\phi} - \tilde{\boldsymbol{y}}||_2^2 = \frac{1}{2}\sum_{i=1}^{M}(\sigma_i\phi_i - \tilde{y}_i)^2 \tag{3.8}$$

where σ_i is the ith singular value of $\boldsymbol{\Phi}$, ϕ_i is the ith component of vector $\boldsymbol{\phi}$, and $\tilde{y}_i$ is the ith component of vector $\tilde{\boldsymbol{y}} = \boldsymbol{U}^T\boldsymbol{y}$. Using (3.6) and (3.8), we can recast the optimization problem in (3.5) as

$$\underset{\boldsymbol{\phi},\boldsymbol{\xi}}{\text{minimize}} \quad F_{p,\epsilon}(\boldsymbol{\phi},\boldsymbol{\xi}) \tag{3.9}$$

where

$$F_{p,\epsilon}(\boldsymbol{\phi},\boldsymbol{\xi}) \;=\; \frac{1}{2}\,||\boldsymbol{\Sigma}\boldsymbol{\phi} - \tilde{\boldsymbol{y}}||_2^2 + \lambda\,||\boldsymbol{V}_r\boldsymbol{\phi} + \boldsymbol{V}_n\boldsymbol{\xi}||_{p,\epsilon}^p \tag{3.10}$$

with $\boldsymbol{x}$ given in (3.6).

Below, we propose an algorithm for the solution of the optimization problem in

(3.9).

3.2.2 Computation of descent direction

In the kth iteration of the proposed algorithm, signal $\boldsymbol{x}^{(k)}$ is updated as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)} \tag{3.11}$$

where

$$\boldsymbol{x}^{(k)} = \boldsymbol{V}_r \boldsymbol{\phi}^{(k)} + \boldsymbol{V}_n \boldsymbol{\xi}^{(k)} \tag{3.12a}$$

$$\boldsymbol{d}^{(k)} = \boldsymbol{V}_r \boldsymbol{d}_r^{(k)} + \boldsymbol{V}_n \boldsymbol{d}_n^{(k)} \tag{3.12b}$$

and $\alpha > 0$. The scalar α is determined using a line search (see Sec. 3.2.3) and the updating vectors $\boldsymbol{d}_r$ and $\boldsymbol{d}_n$ assume the forms

$$\boldsymbol{d}_r^{(k)} = \left[\delta_{r,1}^{(k)} \ \delta_{r,2}^{(k)} \ \cdots \ \delta_{r,M}^{(k)}\right]^T \tag{3.13a}$$

and

$$\boldsymbol{d}_n^{(k)} = \left[\delta_{n,1}^{(k)} \ \delta_{n,2}^{(k)} \ \cdots \ \delta_{n,N-M}^{(k)}\right]^T \tag{3.13b}$$

Vectors $\boldsymbol{d}_r^{(k)}$ and $\boldsymbol{d}_n^{(k)}$ in (3.13) are determined by minimizing the objective function $F_{p,\epsilon}(\boldsymbol{\phi}, \boldsymbol{\xi})$ along each of the directions defined by the column vectors of $[\boldsymbol{V}_r \ \boldsymbol{V}_n]$. In doing so, $\boldsymbol{d}_r^{(k)}$ and $\boldsymbol{d}_n^{(k)}$ become descent directions of $F_{p,\epsilon}$ and their components are found to be

$$\delta_{r,i}^{(k)} = -\frac{-\sigma_i u_i + \lambda p s_i}{\sigma_i^2 + \lambda p \beta_i} \tag{3.14a}$$

$$\delta_{n,i}^{(k)} = -\frac{s_i}{\beta_i} \tag{3.14b}$$

where $u_i = \tilde{y}_i - \sigma_i \phi_i$,

$$s_i = \sum_{j=1}^{N} x_j^{(k)} v_{ij} \gamma_j(\epsilon) \tag{3.15a}$$

$$\beta_i = \sum_{j=1}^{N} v_{ij}^2 \gamma_j(\epsilon) \tag{3.15b}$$

and

$$\gamma_j(\epsilon) = \left[\left(x_j^{(k)}\right)^2 + \epsilon^2\right]^{p/2-1} \tag{3.16}$$

In (3.15a) and (3.16), $x_j^{(k)}$ is the jth component of vector $\boldsymbol{x}^{(k)}$ and in (3.15a) and (3.15b), v_{ij} is the jth component of vector $\boldsymbol{v}_i$ where $\boldsymbol{v}_i$ is determined as follows: a) when s_i and β_i in (3.15) are computed for the use in (3.14a), the ith column of matrix $\boldsymbol{V}_r$ is selected as $\boldsymbol{v}_i$ and b) when s_i and β_i are computed for the use in (3.14b), the ith column of matrix $\boldsymbol{V}_n$ is selected as $\boldsymbol{v}_i$.

See Appendix B for the derivation of Eq. (3.14)-(3.16).

3.2.3 Line search

A line search step is required to determine the value of α in (3.11). In the kth iteration, this is done by minimizing function

$$F_{p,\epsilon}\left(\boldsymbol{\phi}^{(k)} + \alpha \boldsymbol{d}_r^{(k)}, \boldsymbol{\xi}^{(k)} + \alpha \boldsymbol{d}_n^{(k)}\right) \tag{3.17}$$

with respect to α. By applying Banach's fixed-point theorem [45] [46] to (3.17), the step size α is found to be

$$\alpha = -\frac{q_1 + \lambda p q_2}{q_3 + \lambda p q_4} \tag{3.18}$$

where

$$q_1 = \sum_{j=1}^{M} \left(\sigma_j \phi_j^{(k)} - \tilde{y}_j\right) \sigma_j d_{rj}^{(k)} \tag{3.19a}$$

$$q_2 = \sum_{j=1}^{N} x_j^{(k)} d_{vj}^{(k)} \gamma_j(\alpha, \epsilon) \tag{3.19b}$$

$$q_3 = \sum_{j=1}^{M} \left(\sigma_j d_{rj}^{(k)}\right)^2 \tag{3.19c}$$

$$q_4 = \sum_{j=1}^{N} \left(d_{vj}^{(k)}\right)^2 \gamma_j(\alpha, \epsilon) \tag{3.19d}$$

In (3.19), $\phi_j^{(k)}$, $d_{rj}^{(k)}$, $x_j^{(k)}$, and $d_{vj}^{(k)}$ are the jth components of $\boldsymbol{\phi}^{(k)}$, $\boldsymbol{d}_r^{(k)}$, $\boldsymbol{x}^{(k)}$, and $\boldsymbol{d}_v^{(k)}$, respectively, and

$$\gamma_j(\alpha, \epsilon) = \left[\left(x_j^{(k)} + \alpha d_{vj}^{(k)}\right)^2 + \epsilon^2\right]^{p/2-1}$$

Step size α can be obtained through a finite number of iterations by using the formula in (3.18).

3.2.4 Optimization

To solve the problem in (3.9) with a target value $\epsilon = \epsilon_J$ which is typically very small, we use a sequential optimization approach whereby a series of objective functions are minimized starting with a large value of ϵ and gradually decreasing ϵ to ϵ_J. The detailed steps of the sequential optimization are as follows:

- First, set ϵ to a large value, say, ϵ_1, typically in the range $0.5 \leq \epsilon_1 \leq 1$, and initialize $\boldsymbol{\phi}$ and $\boldsymbol{\xi}$ to the zero vectors of dimensions M and $N - M$, respectively.
- Solve the optimization problem in (3.9) by i) computing descent directions $\boldsymbol{d}_v$ and $\boldsymbol{d}_r$, ii) computing the step size α; and iii) updating solution $\boldsymbol{x}$ and coefficient vector $\boldsymbol{\phi}$.
- Reduce ϵ to a smaller value and again solve the problem in (3.9).
- Repeat this procedure until the specified target value, ϵ_J, is reached.
- Output $\boldsymbol{x}$ as the solution.

3.2.5 Algorithm

The proposed $\ell_{p,\epsilon}$-regularized least-squares (ℓ_p-RLS) algorithm for reconstructing sparse signals from compressed measurements is summarized in Table 3.1. The regularization parameter λ, number of iterations T, initial value ϵ_1, final value ϵ_T, and parameter p are supplied in Step 1. The algorithm uses the SVD to compute the singular values $\sigma_1, \sigma_2, \ldots, \sigma_M$ of $\boldsymbol{\Phi}$ and matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ whose columns are, respectively, the left and right singular vectors of $\boldsymbol{\Phi}$. The evaluation of the SVD is computationally demanding for measurement matrices of larger sizes. However, the computation can be performed offline and the resulting matrices can be stored and reused while reconstructing the signal.

A total of $(T - 2)$ values of ϵ lying between the initial value ϵ_1 and final value ϵ_T can be computed as

$$\epsilon_t = \epsilon_1 e^{-\beta(t-1)} \quad \text{for} \quad t = 2, 3, \ldots, T-1 \tag{3.20}$$

where $\beta = \log(\epsilon_1/\epsilon_T)/(T-1)$.

The computation of the step size using (3.18) in Step 4 requires vector $\boldsymbol{\phi}^{(k)}$ which is computed as

$$\boldsymbol{\phi}^{(k)} = \boldsymbol{V}_r^T \boldsymbol{x}^{(k)} \tag{3.21}$$

Table 3.1: ℓ_p-RLS Algorithm

Step 1
Input: λ, p, ϵ_1, ϵ_T, T, L, $\boldsymbol{\Phi}$, and $\boldsymbol{y}$.
Set $\boldsymbol{x}^{(1)} = \boldsymbol{0}$ and $k = 1$.
Step 2
Compute ϵ_j for $j = 2, 3, \ldots, T-1$ using (4.17).
Step 3
Compute the SVD of $\boldsymbol{\Phi}$ to obtain $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, $\boldsymbol{V}_r$, and $\boldsymbol{V}_n$.
Step 4
Repeat for $t = 1, 2, \ldots, T$
i) Set $\epsilon = \epsilon_t$.
ii) Repeat for $l = 1, 2, \ldots, L$
a) Use $\boldsymbol{x}^{(k)}$ as an initial value and compute $\boldsymbol{\phi}^{(k)}$ using Eq. (3.21).
b) Compute $\boldsymbol{d}^{(k)}$ using Eq. (3.12b).
c) Compute α using the line search based on Banach's fixed-point theorem using Eq. (3.18).
e) Compute $\boldsymbol{x}^{(k+1)}$ using Eq. (3.11).
f) Set $k = k + 1$.
Step 5
Set $\boldsymbol{x} = \boldsymbol{x}^{(k)}$ and stop.

3.2.6 Simulation results

Two experiments were performed to investigate the performance of the proposed algorithm.

In the first experiment, the signal length N and the number of measurements M were set to 1024 and 200, respectively. A total of eleven values of sparsity K were chosen from 1 to 101 with an increment of 10. A K-sparse signal $\boldsymbol{x}$ with energy value 100 was constructed as follows: i) a vector $\boldsymbol{x}$ of length N with all zero components was constructed, ii) a random vector of length K was constructed by drawing its components from a normal distribution $\mathcal{N}(0, 1)$ followed by a normalization step so that the ℓ_2 norm of the resulting vector is $\sqrt{100}$, and iii) the components of the resulting vector were set to randomly chosen K locations of vector $\boldsymbol{x}$. A measurement

matrix $\mathbf{\Phi}$ of size $M \times N$ was constructed by drawing its elements from $\mathcal{N}(0,1)$ followed by an orthonormalization step where the rows of $\mathbf{\Phi}$ were made orthonormal relative to each other. The measurement was obtained as $\boldsymbol{y} = \mathbf{\Phi}\boldsymbol{x} + \boldsymbol{w}$ where noise vector $\boldsymbol{w}$ was constructed by drawing its components from $\mathcal{N}(0, 0.01^2)$. The proposed ℓ_p-RLS algorithm was used to reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}$ with $p = 0.1$, $\lambda = 0.0008$, $\epsilon_1 = 0.8$, $\epsilon_J = 10^{-2}$, $J = 30$, and $L = 5$. The reconstruction performance of the ℓ_p-RLS algorithm was compared with that of the BPDN [52], unconstrained approximate ℓ_p (UALP) [54] with $p = 0.1$, iterative reweighted (IR) with $p = 0.1$ [34], and smoothed ℓ_0 pseudonorm (SL0) [47] algorithms. For each algorithm, the signal was deemed reconstructed if the signal-to-noise ratio value, measured as $20\log_{10}(||\boldsymbol{x}||_2/||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2)$, was greater than 27 dB where $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are the initial and reconstruted signals, respectively. The results are shown in Figure 3.1. As can be seen, the performance

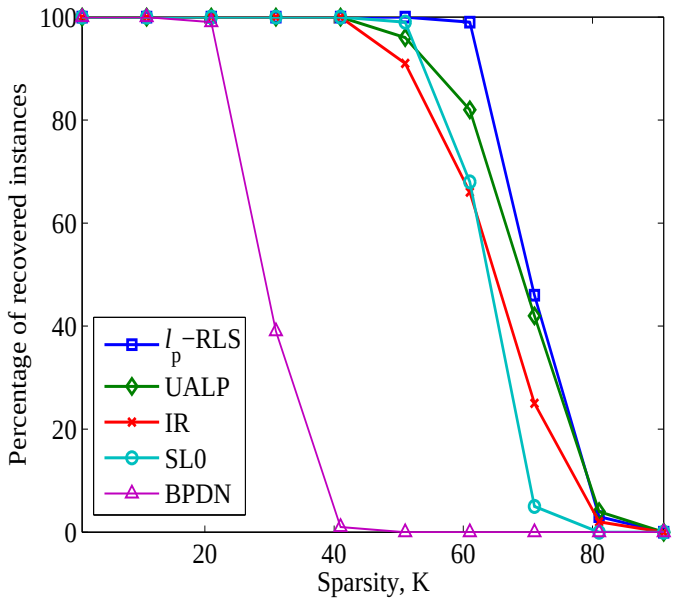


Figure 3.1: Percentage of recovered instances for the ℓ_p-RLS, UALP, IR, SL0, and BPDN algorithms over 100 runs with $N = 1024$, $M = 200$.

of the ℓ_p-RLS algorithm is better than that of the other algorithms.

In the second experiment, signal length N was varied in the range 128 to 512 where $M = N/2$ and $K = \text{round}(M/2.5)$. We constructed a measurement matrix $\mathbf{\Phi}$ and five K-sparse signals $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, $\boldsymbol{x}_4$, and $\boldsymbol{x}_5$ each with different values and locations of nonzero components. Five noisy measurements $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, $\boldsymbol{y}_3$, $\boldsymbol{y}_4$, and $\boldsymbol{y}_5$ were obtained by multiplying the sparse signals by $\mathbf{\Phi}$ and adding five different noise vectors

constructed by drawing their components from $\mathcal{N}(0, 0.01^2)$. The ℓ_p-RLS, UALP, IR, SL0, and BPDN algorithms were used to reconstruct signals from all five measurements and the CPU times required by the various algorithms to reconstruct the five signals were measured. For the proposed ℓ_p-RLS algorithm, the SVD was performed once and the resulting matrices were reused for all five signal reconstructions. For the UALP algorithm, the QR decomposition was performed once and the resulting matrices $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, $\boldsymbol{V}_r$, and $\boldsymbol{V}_n$ were reused for all five signal reconstructions. For the IR and SL0 algorithms, the pseudo-inverse of $\boldsymbol{\Phi}$ was computed only once as $\boldsymbol{\Phi}^T \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T\right)^{-1}$ and reused for all five signal reconstructions. The CPU times were measured using a PC desktop with Intel Core 2 CPU 6400 2.13 GHz processor using MATLAB command *cputime*. The results are shown in Figure 3.2. We observe that the ℓ_p-RLS

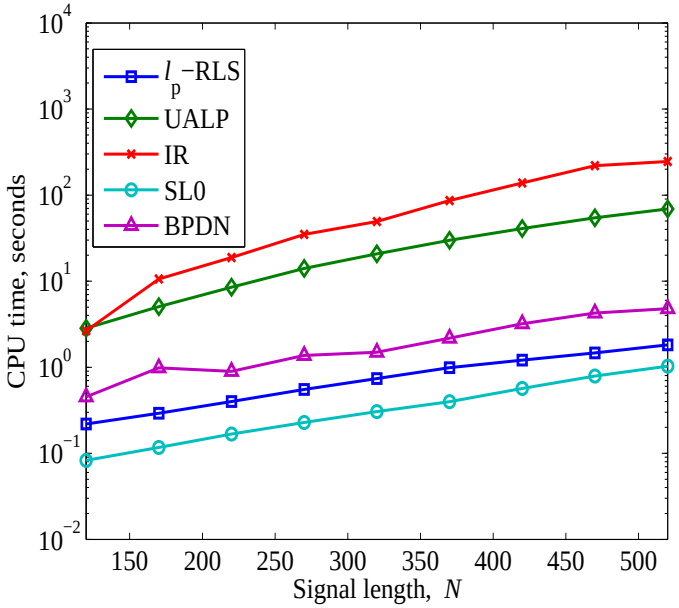


Figure 3.2: Average CPU time required by the ℓ_p-RLS, UALP, IR, SL0, and BPDN algorithms over 100 runs with $M = N/2$, $K = M/2.5$.

algorithm requires much less CPU time than the UALP and IR algorithms, slightly less than the BPDN algorithm, and slightly more than the SL0 algorithm.

Typically the ℓ_p-RLS algorithm converges in a small number of iterations. As an example, Fig. 3.3 shows how the objective function $F_{p,\epsilon}(\boldsymbol{\phi}, \boldsymbol{\xi})$ in (3.9) converges in 20 iterations, where the parameters were set to $N = 1024$, $M = 200$, $K = 60$, $\lambda = 0.0008$, and $\epsilon = 0.01$.

We should point out that the use of the SVD to compute matrices $\boldsymbol{U}$, $\boldsymbol{\Sigma}$, $\boldsymbol{V}_r$, and

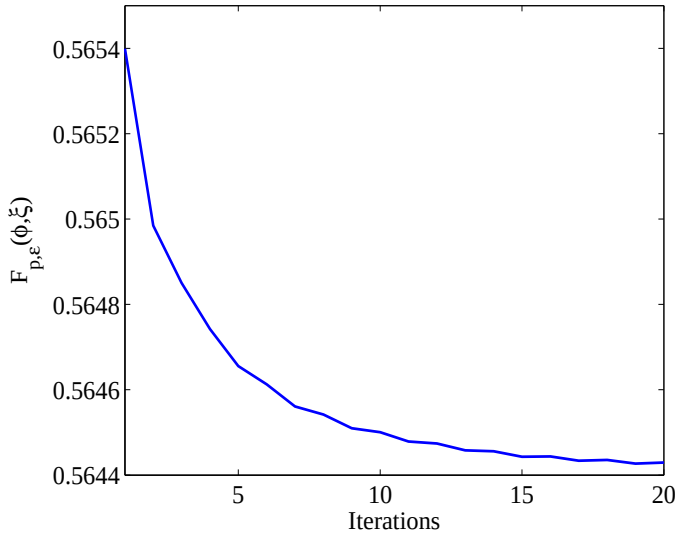


Figure 3.3: Function $F_{p,\epsilon}(\boldsymbol{\phi}, \boldsymbol{\xi})$ for $N = 1024$, $M = 200$, $K = 60$, and $\epsilon = 0.01$.

$\boldsymbol{V}_n$ in Step 3 of the algorithm is computationally expensive for data of moderate to large sizes, e.g., for measurement matrices of size greater than 1000×10000 for the case of a computer with a 2.4 GHz processor and 4 GB physical memory. Below we discuss three cases where the computational burden can be reduced or eliminated.

- In CS, a measurement matrix is usually reused for both sensing and reconstructing. In such applications, the SVD can be computed offline, and vector $\tilde{y}$, matrices $\boldsymbol{V}_r$ and $\boldsymbol{V}_n$, and singular values $\sigma_1, \sigma_2, \ldots, \sigma_M$ can be stored and reused for the reconstruction process.

- In some applications, the measurement matrix $\boldsymbol{\Phi}$ is constructed by selecting a number of rows of a random orthonormal matrix $\boldsymbol{R}$. In these applications, we can use $\boldsymbol{V}_r = \boldsymbol{\Phi}^T$ and $\boldsymbol{V}_n = \boldsymbol{\Psi}^T$ where $\boldsymbol{\Psi}$ is formed by using the remaining rows of $\boldsymbol{R}$; on the other hand, matrix $\boldsymbol{U}$ is the identity matrix and the singular values $\sigma_1, \sigma_2, \ldots, \sigma_M$ are all equal to unity.

- When measurements are taken as a set of samples of a standard transform of the signal such as the Fourier, DCT, or orthogonal wavelet transform, $\boldsymbol{W}$ can be taken to be the orthogonal transfrom matrix. In such cases, the measurement matrix $\boldsymbol{\Phi}$ is composed of a number of rows of $\boldsymbol{W}$. Consequently, we can assign $\boldsymbol{V}_r = \boldsymbol{\Phi}^T$ and $\boldsymbol{V}_n = \boldsymbol{\Psi}^T$ where $\boldsymbol{\Psi}$ is formed by using the remaining rows of $\boldsymbol{W}$. In these applications, matrix $\boldsymbol{U}$ is the identity matrix and the singular values

are all equal to unity.

3.3 ℓ_p-Regularized Least-Squares Optimization Using Conjugate-Gradient Algorithm

3.3.1 Gradient and Hessian of function $F_{p,\epsilon}(\boldsymbol{x})$

In this section, the CG algorithm described in [44] [55] is applied to solve the problem in (3.5). As discussed in Sec. 3.2.1, as long as $\epsilon > 0$ functions $||\boldsymbol{x}||_{p,\epsilon}^p$ and $F_{p,\epsilon}(\boldsymbol{x})$ are differentiable.

The gradient of $F_{p,\epsilon}(\boldsymbol{x})$ is given by

$$\boldsymbol{g} = \boldsymbol{\Phi}^T(\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}) + \lambda \boldsymbol{g}_{\ell_p} \tag{3.22}$$

with $\boldsymbol{g}_{\ell_p} = \begin{bmatrix} g_{\ell_p 1} & g_{\ell_p 2} & \cdots & g_{\ell_p N} \end{bmatrix}$ and

$$g_{\ell_p i} = p\left(x_i^2 + \epsilon^2\right)^{p/2-1} \cdot x_i \quad \text{for} \quad i = 1, 2, \ldots, N$$

The Hessian of function $F_{p,\epsilon}(\boldsymbol{x})$ can be determined by using

$$\boldsymbol{H} = \boldsymbol{\Phi}^T\boldsymbol{\Phi} + \lambda \boldsymbol{U} \tag{3.23}$$

where $\boldsymbol{U} = diag\{u_1,\ u_2,\ \ldots,\ u_N\}$ with

$$u_i = p\left(x_i^2 + \epsilon^2\right)^{p/2-2} \cdot \left((p-1)x_i^2 + \epsilon^2\right) \tag{3.24}$$

for $i = 1, 2, \ldots, N$.

3.3.2 Optimization

We begin by illustrating the role played by parameter ϵ and the efficiency of the sequential optimization for solving the problem in (3.5) using the following example.

Example: We consider a problem of recovering a sparse signal $\boldsymbol{x}$ of length two from measurement $\boldsymbol{\Phi}\boldsymbol{x} = \boldsymbol{y}$, where $\boldsymbol{\Phi} = [0.1\ 0.06]$ and $y = 0.1$. In what follows, we consider the case where $p = 0.1$ and $\lambda = 0.0007$ in which case function $F_{p,\epsilon}(\boldsymbol{x})$ assumes the

form

$$F_{0.1,\epsilon}(\boldsymbol{x}) = 0.5(0.1x_1 + 0.06x_2 - 0.1)^2 + 0.0007\left[\left(x_1^2 + \epsilon^2\right)^{0.05} + \left(x_2^2 + \epsilon^2\right)^{0.05}\right] \quad (3.25)$$

where x_1 and x_2 are the components of $\boldsymbol{x}$. If we let $\epsilon = 0$, then function $F_{0.1,\epsilon}(\boldsymbol{x})$ is reduced to

$$F_{0.1,0}(\boldsymbol{x}) = 0.5\left(0.1x_1 + 0.06x_2 - 0.1\right)^2 + 0.0007\left(|x_1|^{0.1} + |x_2|^{0.1}\right)$$

Function $F_{0.1,0}(\boldsymbol{x})$ is found to have two local minimizers at $\boldsymbol{x}^* = [0.9931\ 0.0001]^T$ and $\hat{\boldsymbol{x}} = [0.0001\ 1.6544]^T$ of which the first one is the global minimizer, as illustrated in the contour plot of Fig. 3.4. Fig. 3.5 illustrates how a sequential optimization procedure works in locating the global minimizer $\boldsymbol{x}^*$. The six plots in Fig. 3.5 show the contours of $F_{0.1,\epsilon}(\boldsymbol{x})$ with $\epsilon = 1$, 0.6, 0.2, 0.08, 0.04, and 0.008. In each plot, the global minimizer of $F_{0.1,0}(\boldsymbol{x})$ (red), the minimizer obtained by minimizing $F_{0.1,\epsilon}(\boldsymbol{x})$ with ϵ set to the preceding value (black), and the minimizer obtained by minimizing $F_{0.1,\epsilon}(\boldsymbol{x})$ with ϵ set to the current value and with the preceding minimizer as its initial point (blue) are shown. It is observed that although function $F_{0.1,\epsilon}(\boldsymbol{x})$ is nonconvex and possesses multiple minimizers, with an appropriately decreasing ϵ, the sequential optimization procedure generates a sequence of intermediate solution points that approach the desired global minimizer $\boldsymbol{x}^*$ of $F_{0.1,0}(\boldsymbol{x})$.

3.3.3 Use of conjugate-gradient algorithm

In the optimization procedure described above, the problem in (3.5) is solved for a set of values of ϵ. For each value of ϵ, we apply the CG algorithm described in [44] to the problem in (3.5). In the kth iteration, the iterate $\boldsymbol{x}_k$ is updated as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k \quad \text{for} \quad k = 0, 1, \ldots, L-1 \quad (3.26)$$

where $\boldsymbol{d}_0$, $\boldsymbol{d}_1$, ..., $\boldsymbol{d}_{L-1}$ are L conjugate directions and α_0, α_1, ..., α_{L-1} are L step sizes. The kth step size is computed using

$$\alpha_k = \frac{\alpha_n}{\alpha_d} \quad (3.27)$$

where

$$\alpha_n = \boldsymbol{g}_k^T \boldsymbol{g}_k \quad \text{and} \quad \alpha_d = \boldsymbol{d}_k^T \boldsymbol{H}_k \boldsymbol{d}_k \quad (3.28)$$

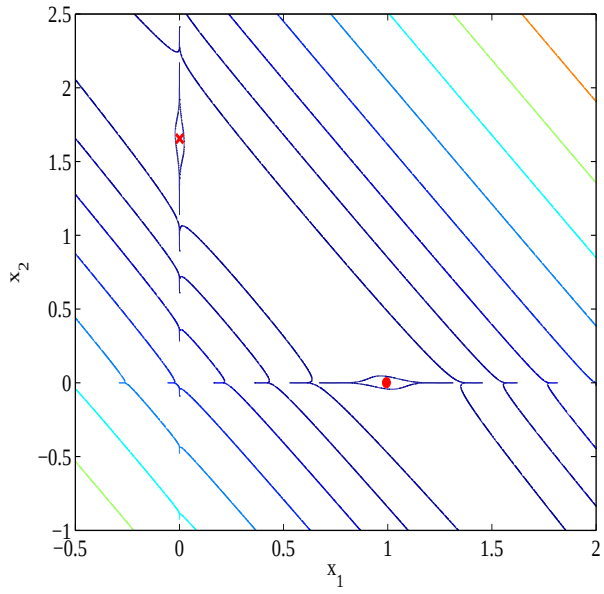


Figure 3.4: Contour of an ℓ_2, ℓ_p objective function for $\mathbf{\Phi} = [0.1\ 0.06]$, $y = 0.1$, $\lambda = 0.0007$, and $p = 0.1$.

In (3.28), $\boldsymbol{g}_k$ and $\boldsymbol{H}_k$ are the gradient vector obtained with $\boldsymbol{x} = \boldsymbol{x}_k$ using (3.22) and the Hessian matrix obtained with $\boldsymbol{x} = \boldsymbol{x}_k$ using (3.23), respectively.

The conjugate directions are computed as

$$\boldsymbol{d}_k = \begin{cases} -\boldsymbol{g}_0 & \text{for} \quad k = 0 \\ -\boldsymbol{g}_k + \beta_{k-1}\boldsymbol{d}_{k-1} & \text{for} \quad k = 1, 2, \ldots, L-1 \end{cases} \tag{3.29}$$

where

$$\beta_k = \frac{\beta_n}{\beta_d} \tag{3.30}$$

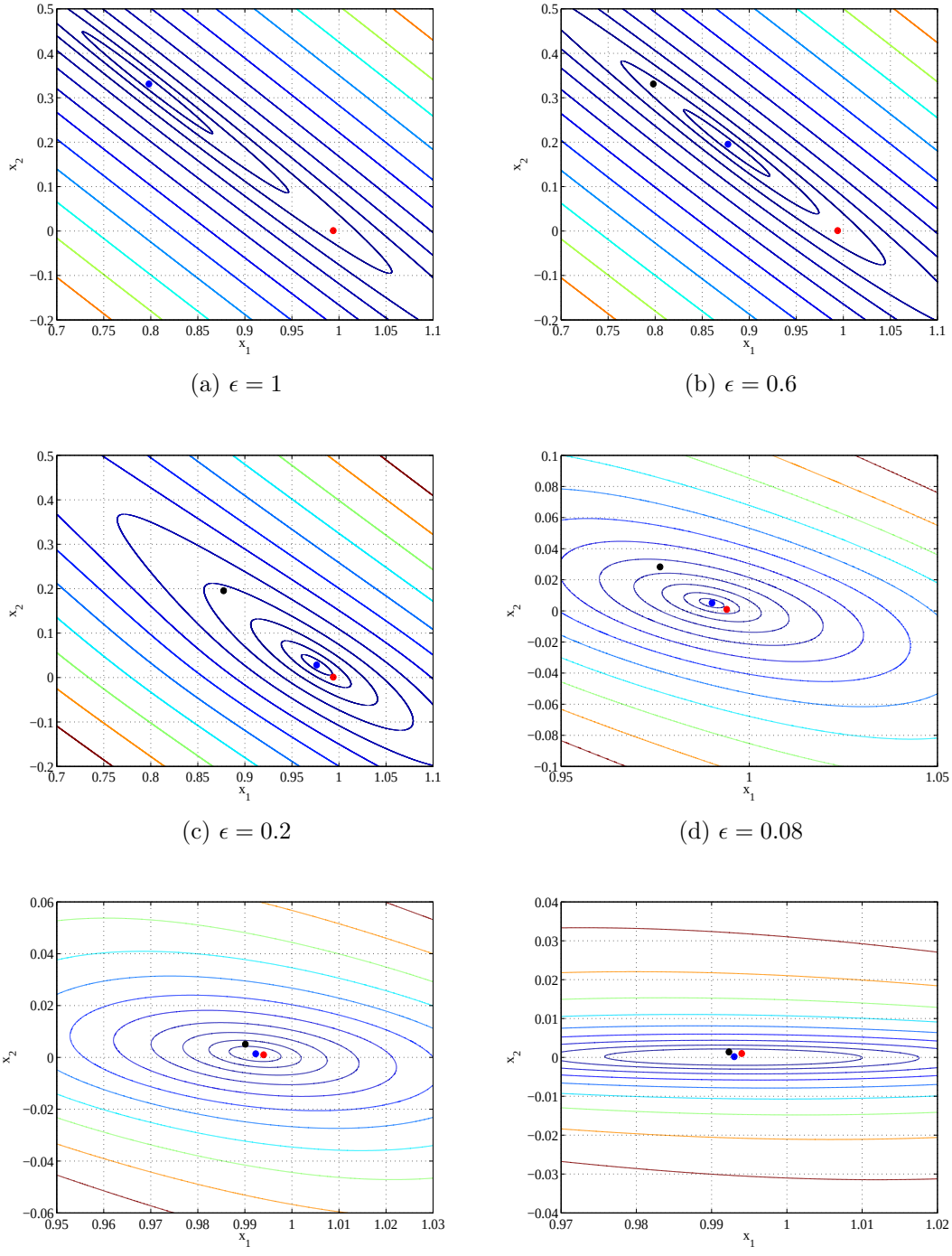
and

$$\beta_n = \boldsymbol{g}_{k+1}^T \boldsymbol{g}_{k+1} \quad \text{and} \quad \beta_d = \boldsymbol{g}_k^T \boldsymbol{g}_k \tag{3.31}$$

for $k = 0, 1, \ldots, L-2$.

3.3.4 Algorithm

The proposed ℓ_p-regularized least-squares conjugate-gradient (ℓ_p-RLS-CG) algorithm is summarized in Table 3.2. The algorithm takes parameter λ, initial and target values of ϵ, i.e. ϵ_1 and ϵ_T, number of outer iterations T, error threshold E_t, measurement



(a) $\epsilon = 1$

(b) $\epsilon = 0.6$

(c) $\epsilon = 0.2$

(d) $\epsilon = 0.08$

(e) $\epsilon = 0.04$

(f) $\epsilon = 0.008$

Figure 3.5: Contours of function $F_{p,\epsilon}(\boldsymbol{x})$ for $\epsilon = 1$, 0.6, 0.2, 0.08, 0.04, and 0.008 with $p = 0.1$, $\lambda = 0.0007$, $\boldsymbol{\Phi} = [0.1\ 0.06]$, and $y = 0.1$.

vector $\boldsymbol{y}$, and measurement matrix $\boldsymbol{\Phi}$ as input.

A total of $T-2$ values of ϵ lying between initial value ϵ_1 and target value ϵ_T are computed as

$$\epsilon_t = \epsilon_1 e^{-\beta(t-1)} \quad \text{for} \quad t = 2, 3, \ldots, T-1 \tag{3.32}$$

where $\beta = \log(\epsilon_1/\epsilon_T)/(T-1)$.

The Hessian matrix $\boldsymbol{H}_k$ computed in each step of the CG algorithm is made positive definite by setting the diagonal components of matrix $\boldsymbol{U}$ in (3.23) as

$$u_i = \begin{cases} u_i & \text{if } u_i > \delta \\ \delta & \text{if } u_i \leq \delta \end{cases} \tag{3.33}$$

where δ is a small positive scalar.

The computation of the denominator term α_d in (3.27) is carried out in an efficient way as

$$\begin{aligned} \alpha_d &= \boldsymbol{d}_k^T \boldsymbol{H} \boldsymbol{d}_k \\ &= \boldsymbol{d}_k^T \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{d}_k + \lambda \boldsymbol{d}_k^T \boldsymbol{U}^{1/2^T} \boldsymbol{U}^{1/2} \boldsymbol{d}_k \\ &= ||\boldsymbol{\Phi} \boldsymbol{d}_k||_2^2 + \lambda ||\boldsymbol{b}_k||_2^2 \end{aligned} \tag{3.34}$$

In (3.34), $\boldsymbol{U}^{1/2} = \text{diag}\left\{\sqrt{u_1}, \sqrt{u_2}, \ldots, \sqrt{u_N}\right\}$ where u_i is obtained using (3.24) in conjunction with (3.33) and $\boldsymbol{b}_k = [b_{k1}\ b_{k2}\ \cdots\ b_{kN}]^T$ where $b_{ki} = \sqrt{u_i} d_{ki}$ and d_{ki} is the ith component of $\boldsymbol{d}_k$.

3.3.5 ℓ_p-RLS-CG algorithm for noiseless measurements

In this section, we present a method for the reconstruction of sparse signals from noiseless measurements by using the ℓ_p-RLS-CG algorithm.

Recall from the discussion in Sec. 1.2.2.2 in Chapter 1 that, for noiseless measurements parameter λ in (3.5) must be sufficiently small so that the solution of (3.5) satisfies the equality constraint in (1.5b). Similarly, the ℓ_p-RLS-CG algorithm can be used with a sufficiently small λ for the reconstruction of signals from noiseless measurements. However, when the ℓ_p-RLS-CG algorithm is used with a small λ, it often yields a suboptimal solution. This is because with a small λ the gradient of the regularization term $\lambda ||\boldsymbol{x}||_{p,\epsilon}^p$ in the objective function becomes insignificant relative to that of the fidelity term $\frac{1}{2} ||\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}||_2^2$. Consequently, while carrying out the update

Table 3.2: ℓ_p-RLS-CG Algorithm

Step 1
Input p, T, ϵ_1, ϵ_T, λ, E_t, δ, $\boldsymbol{\Phi}$, and $\boldsymbol{y}$.
Set $\boldsymbol{x}_s = \boldsymbol{0}$.
Step 2
Compute ϵ_t for $t = 2, 3, \ldots, T-1$ using (3.32).
Step 3
Repeat for $t = 1, 2, \ldots, T$
i) Set $\epsilon = \epsilon_t$ and $L_t = 3 + \text{round}(t/4)$.
ii) Set $k = 0$, $\boldsymbol{x}_0 = \boldsymbol{x}_s$, and $E_r = 10^{10}$.
iii) While $E_r > E_t$.
a) Compute $\boldsymbol{g}_k$ using (3.22).
b) Compute $\boldsymbol{d}_k$ using (3.29).
c) Compute α_k using (3.27).
d) Compute $\boldsymbol{x}_{k+1}$ using (3.26).
e) Set $k = k + 1$.
f) Exit loop if $k > L_t$.
g) Compute $E_r = ||\alpha_k \boldsymbol{d}_k||_2$.
iv) Set $\boldsymbol{x}_s = \boldsymbol{x}_k$.
Step 4
Output $\boldsymbol{x}^* = \boldsymbol{x}_s$ and stop.

in (3.26), the step taken by the iterate along the descent direction of $\frac{1}{2}||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2$ is significantly longer than that taken along the descent direction of $\lambda||\boldsymbol{x}||_{p,\epsilon}^p$. We deal with this technical difficulty by initializing λ using a large value, say, λ_1, and decrease its value every time the value of ϵ is decreased. If we denote a sufficiently small target value of λ as λ_T, then a total of $(T-2)$ values of λ in between λ_1 and λ_T are computed as

$$\lambda_i = \lambda_1 e^{-\beta(i-1)} \quad \text{for} \quad i = 2, 3, \ldots, T-1 \tag{3.35}$$

where $\beta = \log(\lambda_1/\lambda_T)/(T-1)$.

The algorithm incorporating this technique is summarized in Table 3.3.

Table 3.3: ℓ_p-RLS-CG Algorithm for Noiseless Measurements

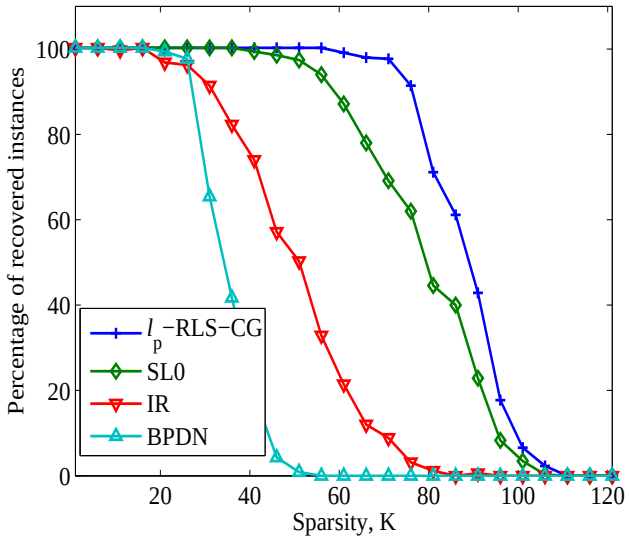
Step 1
Input p, T, ϵ_1, ϵ_T, λ_1, λ_T, E_t, δ, $\mathbf{\Phi}$, and $\boldsymbol{y}$.
Set $\boldsymbol{x}_s = \mathbf{0}$.
Step 2
Compute ϵ_t for $t = 2, 3, \ldots, T-1$ using (3.32).
Compute λ_t for $t = 2, 3, \ldots, T-1$ using (3.35).
Step 3
Repeat for $t = 1, 2, \ldots, T$
i) $\epsilon = \epsilon_t$, $\lambda = \lambda_t$, and $L_t = 3 + \text{round}(t/4)$.
$\vdots$
(Steps same as in Table 3.2)
$\vdots$
Step 4
Output $\boldsymbol{x}^* = \boldsymbol{x}_s$ and stop.

3.3.6 Simulation results

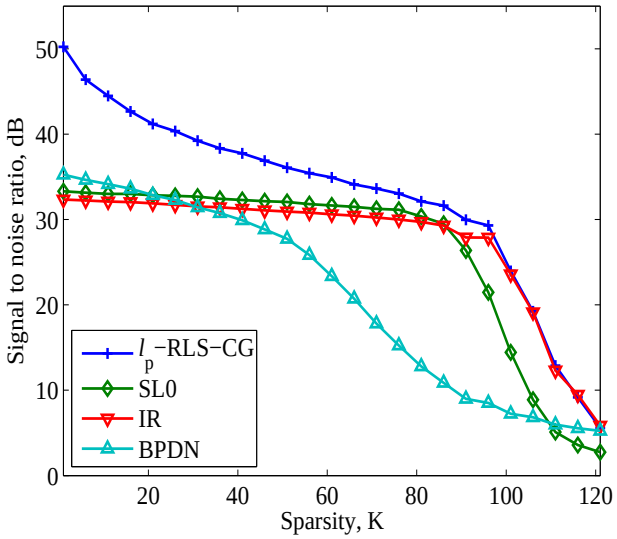
3.3.6.1 *Signal reconstruction from noisy measurements using ℓ_p-RLS-CG algorithm*

In the first experiment, we have set the signal length, N, to 512, the number of measurements, M, to 200, and varied the sparsity, K, from 1 to 121 with an increment of 5. A K-sparse signal with energy value 100 was constructed as follows: i) a vector $\boldsymbol{x}$ of length N with all zero components was constructed, ii) a random vector of length K was constructed by drawing its components from a normal distribution $\mathcal{N}(0, 1)$ followed by a normalization step so that the ℓ_2 norm of the resulting vector is $\sqrt{100}$, and iii) the components of the resulting vector were set to randomly chosen K locations of vector $\boldsymbol{x}$. A measurement matrix $\mathbf{\Phi}$ of size $M \times N$ was constructed by selecting its components from $\mathcal{N}(0, 1)$ followed by an orthonormalization step so that the rows of $\mathbf{\Phi}$ are orthonormal relative to each other. A noisy measurement was taken using $\boldsymbol{y} = \mathbf{\Phi}\boldsymbol{x} + \boldsymbol{w}$ where $\boldsymbol{w}$ is a measurement noise vector of length M whose components were drawn from $\mathcal{N}(0, 0.01^2)$. The ℓ_p-RLS-CG algorithm was applied with parameters $\epsilon_1 = 1$, $\epsilon_T = 1e-2$, $T = 80$, $p = 0.1$, $\lambda = 0.007$, $E_t = 1e-25$, and $\delta = 1e-5$. The signal-to-noise ratio (SNR) was measured as $20\log\left(||\boldsymbol{x}||_2/||\boldsymbol{x} - \hat{\boldsymbol{x}}||_2\right)$ where $\hat{\boldsymbol{x}}$ is the reconstructed signal. The signal was deemed to be reconstructed when the SNR was greater than 31 dB. The signal reconstruction performance of the iteratively reweighted (IR) algorithm [34] with $p = 0.1$, the smoothed ℓ_0 (SL0)

algorithm [47], the basis pursuit algorithm with denoising (BPDN) [52], and the proposed ℓ_p-RLS-CG algorithm was measured in terms of the percentages of recovered instances and the average SNR over 350 runs. The results are plotted in Figure 3.6. We observe from Figure 3.6a that the percentage of recovered instances for the



(a) Percentage of recovered instances



(b) Average signal-to-noise ratio

Figure 3.6: Reconstruction performance for the ℓ_p-RLS-CG, SL0, IR, and BPDN algorithms for noisy measurements over 350 runs with $N = 512$, $M = 200$, and measurement noise $\mathcal{N}(0, 0.01^2)$.

ℓ_p-RSL-CG algorithm is higher than that of the other algorithms with which it is being compared. Also, the average SNR performance of the ℓ_p-RLS-CG algorithm is superior to that of the other algorithms as can be observed in Figure 3.6b.

The first experiment was repeated by increasing the variance of the noise σ^2 from 0.01^2 to 0.03^2. Because of the increased noise level, a signal was deemed reconstructed when the SNR was greater than 20 dB. The percentages of recovered instances for the algorithms being compared are plotted in Fig. 3.7. As can be seen, the percentage of recovered instances for the ℓ_p-RLS-CG algorithm is much higher than that for the other algorithms.

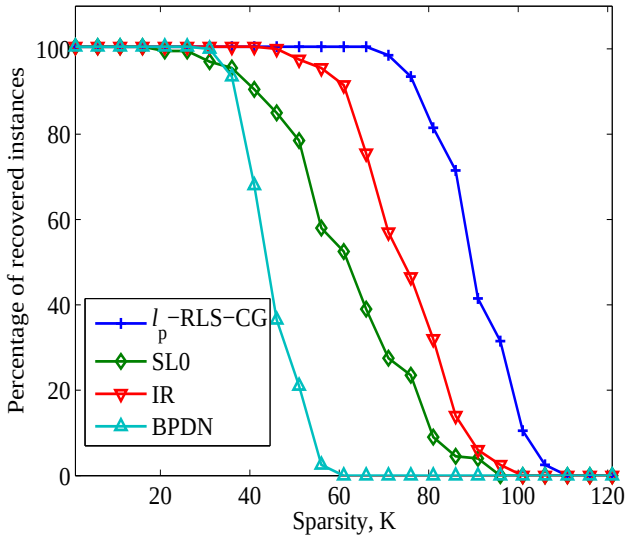


Figure 3.7: Reconstruction performance for the ℓ_p-RLS-CG, SL0, IR, and BPDN algorithms for noisy measurements over 250 runs with $N = 512$, $M = 200$, and measurement noise drawn from $\mathcal{N}(0, 0.03)$.

The second experiment was concerned with the computational complexity of the algorithms being compared. The signal length, N, was varied from 200 to 2000 with an increment of 200. The number of measurements, M, and sparsity, K, were set to $M = N/2$ and $K = \text{round}(M/2.5)$, respectively. Sparse signal $\boldsymbol{x}$ and measurement matrix $\boldsymbol{\Phi}$ were constructed, and the noisy measurement $\boldsymbol{y}$ was taken as in the first experiment. The ℓ_p-RLS-CG, SL0, IR, and BPDN algorithms were used to reconstruct signal $\boldsymbol{x}$ from measurement $\boldsymbol{y}$. The computational complexity of the five algorithms being compared was measured in terms of the average CPU time required over 100 runs. The CPU time was measured using a PC desktop with Intel Core 2 CPU 6400 2.13 GHz processor using MATLAB command *cputime*. The results are plotted in

Figure 3.8. It is observed that the CPU time required by the ℓ_p-RLS-CG algorithm is

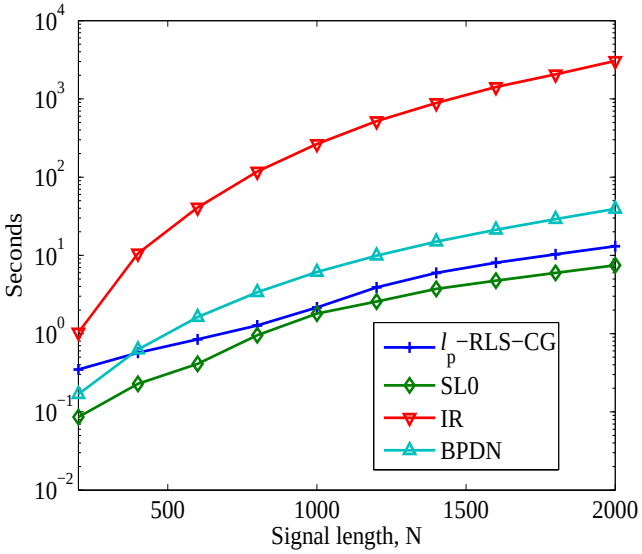


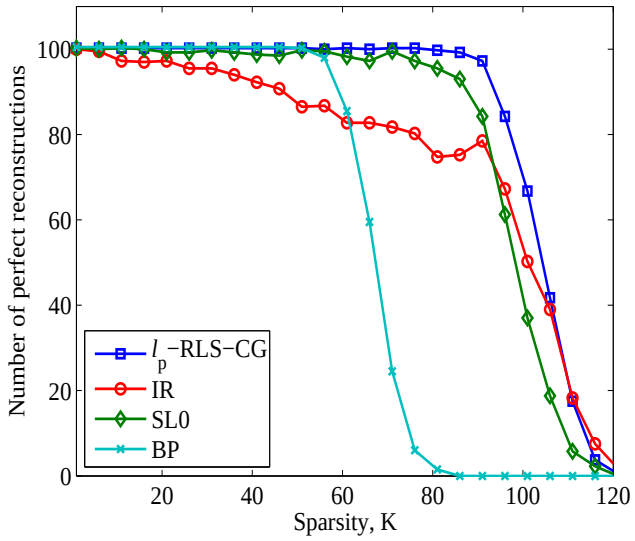
Figure 3.8: Average CPU required by the ℓ_p-RLS-CG, SL0, IR, and BPDN algorithms for noisy measurements over 100 runs with $M = N/2$, $K = M/2.5$.

less than that required by the IR and BPDN algorithms and slightly more than that required by the SL0 algorithm.

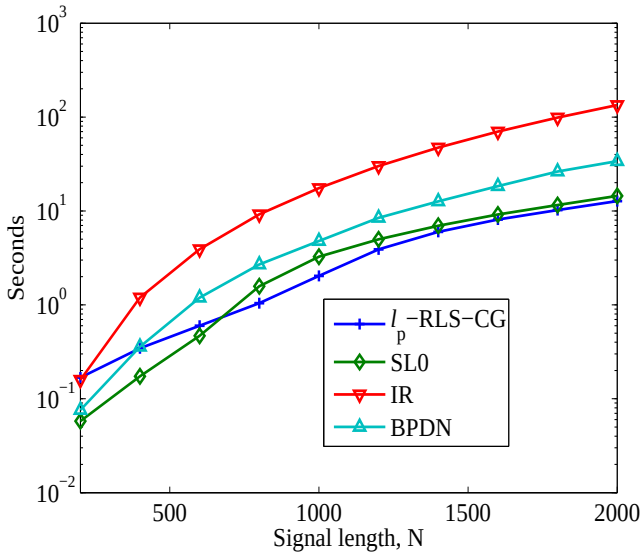
3.3.6.2 *Signal reconstruction from noise-free measurements using ℓ_p-RLS-CG algorithm*

In the third experiment, the signal length, N, the number of measurements, M, and the set of values of sparsity K were chosen as in the first experiment. A K-sparse $\boldsymbol{x}$ was constructed as follows: i) a vector $\boldsymbol{x}$ of length N with all zero components was constructed, ii) a random vector of length K was constructed by drawing its components from $\mathcal{N}(0, 1)$, and iii) the components of the resulting vector were set to randomly chosen K locations of vector $\boldsymbol{x}$. Measurement matrix $\boldsymbol{\Phi}$ was constructed and measurement $\boldsymbol{y}$ was taken as in the first experiment. The ℓ_p-RLS-CG algorithm was implemented using the procedure described in Sec. 3.3.5 with $\lambda_1 = 1e-5$ and $\lambda_T = 1e-15$; the other parameters were the same as in the first experiment. The ℓ_p-RLS-CG, SL0, IR with $p = 0.1$, and BP algorithm were used to reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}$. The maximum value of the absolute error between the true signal $\boldsymbol{x}$ and the reconstructed signal $\hat{\boldsymbol{x}}$ was measured as $\mathcal{E} = \max_i |x_i - \hat{x}_i|$ where x_i and $\hat{x}_i$ are the ith components of vectors $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, respectively. Reconstruction was considered perfect

if $\mathcal{E} < 1e-5$. The number of perfect reconstructions over 400 runs are plotted in Figure 3.9a. We observe that the ℓ_p-RLS-CG algorithm has performed better than



(a) Number of perfect reconstructions over 400 runs



(b) Average CPU time over 100 runs

Figure 3.9: Number of perfect reconstructions and average CPU time required for the ℓ_p-RLS-CG, SL0, IR, and BP algorithms.

other algorithms.

In the fourth experiment, all the algorithms under comparison were run with $M = N/2$ and $K = \text{round}(M/2.5)$ with $N = 200, 400, \ldots, 2000$. The amount of

computation required by each algorithm to reconstruct the signal was measured using MATLAB command *cputime*. The average CPU times required by the various algorithms over 100 runs are plotted in Figure 3.9b. Clearly, the computational complexity of the ℓ_p-RLS-CG algorithm is less than that required by other algorithms for large sized data, i.e., for $N > 1000$.

3.4 Optimization of parameter λ

The quality of the signal obtained by solving the problem in (3.5) is closely related to the value of parameter λ. In this section, we present a method for identifying a good value of λ for the signal model in (3.1) for the case where the components of noise $\boldsymbol{w}$ are i.i.d. Gaussian random variables with zero mean and variance σ^2. From (3.1), it follow that

$$||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2 = ||\boldsymbol{w}||_2^2 \approx M\sigma^2 \tag{3.36}$$

where M denotes the dimension of $\boldsymbol{w}$.

Let $[\lambda_l, \lambda_u]$ be an interval that contains the optimal value of λ. Since λ is always positive, we can set an initial interval $[\lambda_l, \lambda_u]$ with $\lambda_l = 0$ and λ_u sufficiently large. The length of the interval is regarded as a range of uncertainty (ROU) with respect to the optimal value of λ, which can be reduced by a bisection technique as follows. Let $\boldsymbol{x}_\lambda$ be the solution of the problem in (3.5) with $\lambda = (\lambda_l + \lambda_u)/2$. We evaluate $||\boldsymbol{\Phi x}_\lambda - \boldsymbol{y}||_2^2$ and use it to reduce the ROU based on (3.36) as

$$\begin{array}{lll} \text{set} & \lambda_u = \lambda & \text{if} \quad ||\boldsymbol{\Phi x}_\lambda - \boldsymbol{y}||_2^2 > M\sigma^2 \\ \text{set} & \lambda_l = \lambda & \text{if} \quad ||\boldsymbol{\Phi x}_\lambda - \boldsymbol{y}||_2^2 \leq M\sigma^2 \end{array} \tag{3.37}$$

In this way, interval $[\lambda_l, \lambda_u]$ is updated with its ROU reduced by 50%. This procedure is repeated until the ROU falls below a prescribed tolerance, and the midpoint of the last interval $[\lambda_l, \lambda_u]$ is taken to be an approximation of the optimal value of λ.

3.4.1 Algorithm

An algorithm based on the above analysis is summarized in Table 3.4. Step 2 of the algorithm runs the ℓ_p-RLS-CG algorithm with parameters p, T, ϵ_1, ϵ_T, λ_0, error threshold E_t, and δ and determines a sparse solution. Step 3 of the algorithm uses the parameters p, ϵ_T, λ_u, λ_l, inner error threshold E_t, outer error threshold E_{to}, the

maximum number of CG iterations L, and δ and determines an optimal value of parameter λ and the corresponding optimal solution denoted as $\boldsymbol{x}_\lambda$.

Table 3.4: ℓ_p-RLS-CG Algorithm for Noiseless Measurements by Optimizing λ

Step 1
Input p, T, ϵ_1, ϵ_T, λ_u, λ_l, $\boldsymbol{\Phi}$, $\boldsymbol{y}$, E_t, E_{to}, δ, and σ.
Set $\boldsymbol{x}_s = \boldsymbol{0}$, $E_{ro} = 10^{10}$.
Step 2
Repeat the following while $E_{ro} > E_{to}$.
i) Compute $\lambda = (\lambda_l + \lambda_u)/2$.
ii) Solve the problem in (3.5) using the ℓ_2-RLS-CG algorithm summarized in Table 3.2 with $\boldsymbol{x}_s$ as the initializer. Denote the resulting solution as $\boldsymbol{x}_\lambda$.
iii) Update λ_u or λ_l using (3.37).
iv) Compute $E_{ro} = \lambda_u - \lambda_l$.
v) Set $\boldsymbol{x}_s = \boldsymbol{x}_\lambda$.
Step 3
Compute $\lambda = (\lambda_l + \lambda_u)/2$.
Solve the problem in (3.5) using the ℓ_2-RLS-CG algorithm. Denote the resulting solution as $\boldsymbol{x}_\lambda$.
Step 4
Output $\boldsymbol{x}^* = \boldsymbol{x}_\lambda$ and stop.

3.4.2 Experimental results

The performance of the proposed technique for the optimization of parameter λ in the ℓ_p-RLS-CG algorithm was evaluated by using two numerical experiments as detailed below.

In the first experiment, the signal length and number of measurements were set to $N = 512$ and $M = 200$, and the value of sparsity K was varied from 1 to 121 with an increment of 5. A K-sparse signal of length N with energy value 100 was constructed by assigning K random values drawn from a normal distribution $\mathcal{N}(0, 1)$ to K randomly chosen locations of a zero vector of length N followed by a normalization step so that the ℓ_2 norm of the resulting vector is $\sqrt{100}$. A measurement matrix $\boldsymbol{\Phi}$ of size $M \times N$ was constructed by drawing its components from $\mathcal{N}(0, 1)$ followed by an orthonormalization step so that the rows of $\boldsymbol{\Phi}$ were orthonormal with respect to each other. A noisy measurement was taken using $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{w}$, where $\boldsymbol{w}$ is a measurement noise vector of length M whose components were drawn from $\mathcal{N}(0, 0.01^2)$. The ℓ_p-RLS-CG algorithm incorporating the optimization of λ was ap-

plied with parameters $p = 0.1$, $T = 80$, $\epsilon_1 = 1$, $\epsilon_T = 1e-2$, $\lambda_l = 0$, $\lambda_u = 0.005$, $E_t = 1e-25$, $E_{to} = 1e-4$, and $\delta = 1e-5$. The ℓ_p-RLS-CG algorithm was implemented with $p = 0.1$, $T = 80$, $\epsilon_1 = 1$, $\epsilon_T = 1e-2$, $\lambda = 0.007$, $E_t = 1e-25$, and $\delta = 1e-5$. The ℓ_p-RLS-CG incorporating the optimization of λ, ℓ_p-RLS-CG, SL0, IR, and BPDN algorithms were used to recover signal $\boldsymbol{x}$ from $\boldsymbol{y}$. The SNR was measured as $20\log(||\boldsymbol{x}||_2/||\boldsymbol{x}-\boldsymbol{x}^*||_2)$, where $\boldsymbol{x}$ and $\boldsymbol{x}^*$ are the original and reconstructed signals, respectively. The signal was deemed reconstructed when the SNR was greater than 31 dB. The percentages of recovered instances for the five algorithms are plotted in Fig. 3.10. The plots demonstrate that the proposed approach for the optimization of λ for the ℓ_p-RLS-CG algorithm yields improved performance. The ℓ_p-RLS-CG algorithm incorporating the optimization of λ took a total of six iterations in Step 2 (see Table 3.4) and the resulting optimal value of λ was $4.9219e-03$.

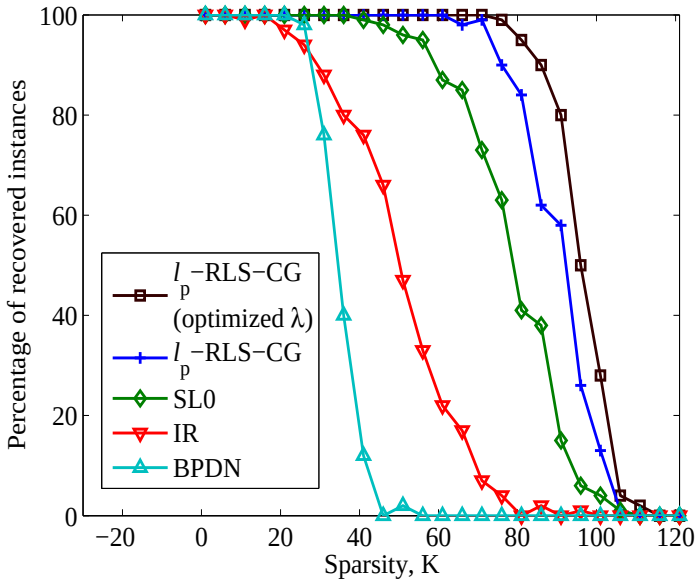


Figure 3.10: Percentage of recovered instances for the ℓ_p-RLS-CG algorithm with and without optimizing λ and the SL0, IR, and BP algorithms over 100 runs with $N = 512$ and $M = 200$.

In the second experiment, the signal length was set to $N = 200, 400, 600, \ldots, 2000$ and the number of measurements and sparsity levels were determined using $M = N/2$ and $K = \text{round}(M/2.5)$, respectively. All the algorithms under comparison were applied and the computational complexity of each algorithm was measured in terms of the average CPU time required over 100 runs. The results are plotted in Fig. 3.11. We observe that the CPU time required by the ℓ_p-RLS-CG algorithm with the proposed

method for the optimization of λ is less than that required by the IR algorithm and more than that required by the rest of the algorithms.

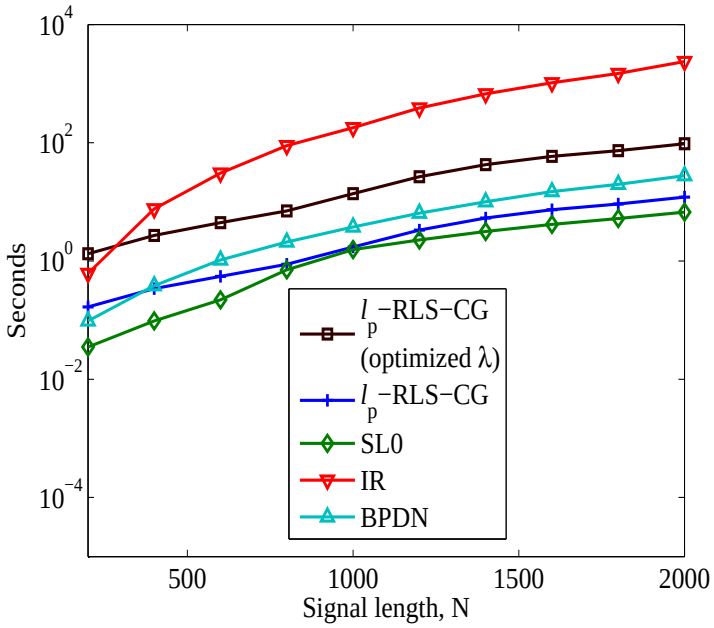


Figure 3.11: Average CPU time for the ℓ_p-RLS-CG algorithm with and without optimizing λ and the SL0, IR, and BP algorithms over 100 runs with $M = N/2, K = \text{round}(M/2.5)$.

3.5 Reconstruction of Images Using TV_p-Regularized Least-Squares and Conjugate-Gradient Algorithm

In this section, we generalize the concept of TV to a pth-power total variation with $p < 1$, namely, TV_p, and present an algorithm for the reconstruction of an image by minimizing a TV_p-regularized ℓ_2 objective function using Fletcher-Reeves' CG algorithm [44].

3.5.1 The total variation norm and its generalization

The concept of TV was first introduced in the celebrated paper by Rudin, Osher, and Fatemi (ROF) [56] where a technique for image denoising by minimizing the

so-called TV norm of an image was presented. The last two decades have witnessed the growth of increasing research interest on this effective methodology and a large volume of literature covering a variety of image and video processing algorithms has been produced [57] [58] [59] [60] [61].

Let $\boldsymbol{X}$ be a matrix of size $n_1 \times n_2$ whose components represent the pixels of a gray-scale image. In [59], two TV norms for $\boldsymbol{X}$ are defined as follows. For image $\boldsymbol{X}$, the discrete *isotropic* TV norm, denoted as $TV_I(\boldsymbol{X})$, and *anisotropic* TV norm, denoted as $TV_{l1}(\boldsymbol{X})$, are defined, respectively, as

$$\begin{aligned} TV_I(\boldsymbol{X}) &= \sum_{i=1}^{n_1-1}\sum_{j=1}^{n_2-1}\sqrt{(x_{i,j}-x_{i+1,j})^2+(x_{i,j}-x_{i,j+1})^2} \\ &\quad +\sum_{i=1}^{n_1-1}|x_{i,n_2}-x_{i+1,n_2}|+\sum_{j=1}^{n_2-1}|x_{n_1,j}-x_{n_1,j+1}| \end{aligned} \tag{3.38a}$$

and

$$\begin{aligned} TV_{l1}(\boldsymbol{X}) &= \sum_{i=1}^{n_1-1}\sum_{j=1}^{n_2-1}\left(|x_{i,j}-x_{i+1,j}|+|x_{i,j}-x_{i,j+1}|\right) \\ &\quad +\sum_{i=1}^{n_1-1}|x_{i,n_2}-x_{i+1,n_2}|+\sum_{j=1}^{n_2-1}|x_{n_1,j}-x_{n_1,j+1}| \end{aligned} \tag{3.38b}$$

where $x_{i,j}$ is the $\{i,j\}$th component of matrix $\boldsymbol{X}$. In what follows, we only consider the isotropic TV norm. For the sake of simplicity, we call it the TV norm of image $\boldsymbol{X}$ and denote it as $TV(\boldsymbol{X})$.

Consider the image model

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$$

where $\boldsymbol{Y}$ denotes the noisy measurement of image $\boldsymbol{X}$ and $\boldsymbol{W}$ is random noise. It is well known that the problem of denoising image $\boldsymbol{Y}$ can be treated effectively by solving the unconstrained convex problem

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \tfrac{1}{2}\,||\boldsymbol{X}-\boldsymbol{Y}||_F^2 + \lambda TV(\boldsymbol{X}) \tag{3.39}$$

where $\lambda > 0$ is a regularization parameter, and $||\cdot||_F$ denotes the Frobenius norm [56]. If we denote the vectorized $\boldsymbol{X}$ and $\boldsymbol{Y}$ in (3.39) by $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively, then (3.39)

becomes

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad \tfrac{1}{2}\,||\boldsymbol{x}-\boldsymbol{y}||_2^2 + \lambda TV(\boldsymbol{X}) \tag{3.40}$$

On comparing (3.40) with the ℓ_2, ℓ_1 optimization problem given in (1.18), namely,

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad \tfrac{1}{2}\,||\boldsymbol{\Phi x}-\boldsymbol{y}||_2^2 + \lambda||\boldsymbol{x}||_1 \tag{3.41}$$

we see the obvious similarities as well as differences between the two formulations. Encouraged by the improvement in signal reconstruction performance achieved by using the ℓ_p minimization in previous chapters, we propose to investigate possible extensions of the TV norm similar to our extension of the ℓ_1 norm to the ℓ_p pseudonorm. To this end, we introduce the TV_p pseudonorm of $\boldsymbol{X}$, $TV_p(\boldsymbol{X})$, as follows

$$\begin{aligned} TV_p^p(\boldsymbol{X}) \;=\; & \sum_{i=1}^{n_1-1}\sum_{j=1}^{n_2-1}\left[(x_{i,j}-x_{i+1,j})^2+(x_{i,j}-x_{i,j+1})^2\right]^{p/2} \\ & +\sum_{i=1}^{n_1-1}|x_{i,n_2}-x_{i+1,n_2}|^p \\ & +\sum_{j=1}^{n_2-1}|x_{n_1,j}-x_{n_1,j+1}|^p \end{aligned} \tag{3.42}$$

where $p \in (0,1]$. Note that $TV_p(\boldsymbol{X})$ reduces to $TV(\boldsymbol{X})$ when $p=1$.

Like the ℓ_p pseudonorm, $TV_p(\boldsymbol{X})$ is not differentiable. To facilitate the minimization of $TV_p^p(\boldsymbol{X})$, we consider a smoother version, namely,

$$\begin{aligned} TV_{p,\epsilon}^p(\boldsymbol{X}) \;=\; & \sum_{i=1}^{n_1-1}\sum_{j=1}^{n_2-1}\left[(x_{i,j}-x_{i+1,j})^2+(x_{i,j}-x_{i,j+1})^2+\epsilon^2\right]^{p/2} \\ & +\sum_{i=1}^{n_1-1}\left[(x_{i,n_2}-x_{i+1,n_2})^2+\epsilon^2\right]^{p/2}+\sum_{j=1}^{n_2-1}\left[(x_{n_1,j}-x_{n_1,j+1})^2+\epsilon^2\right]^{p/2} \end{aligned} \tag{3.43}$$

where ϵ is a small approximation parameter.

3.5.2 Problem formulation

Let a vector $\boldsymbol{x}$ of size $n_1 n_2$ be constructed as

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_{n_2} \end{bmatrix}$$

where $\boldsymbol{x}_i$ is the ith column of an image $\boldsymbol{X}$. Let a measurement vector $\boldsymbol{y}$ of size m be obtained as

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{w} \tag{3.44}$$

where $\boldsymbol{\Phi}$ is a measurement matrix of size $m \times n_1 n_2$ with $m < n_1 n_2$ and $\boldsymbol{w}$ is a noise vector. If we assume that the components of $\boldsymbol{w}$ are i.i.d. Gaussian random variables with zero mean and variance σ^2, then (3.44) implies that

$$||\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}||_2^2 = ||\boldsymbol{w}||_2^2 \approx m\sigma^2 \tag{3.45}$$

In image processing applications, matrix $\boldsymbol{\Phi}$ is often constructed by selecting its rows uniformly at random from standard transform matrices such as the Fourier, wavelet, or discrete-cosine transforms.

We propose to reconstruct image $\boldsymbol{X}$ from measurements $\boldsymbol{y}$ by solving the optimization problem

$$\underset{\boldsymbol{X}}{\text{minimize}} \quad F_{p,\epsilon}(\boldsymbol{X}) = ||\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}||_2^2 + \lambda \cdot TV_{p,\epsilon}^p(\boldsymbol{X}) \tag{3.46}$$

with a small ϵ, where $\lambda > 0$ is a regularization parameter and function $TV_{p,\epsilon}^p(\boldsymbol{X})$ is given in (3.43). Note that function $TV_{p,\epsilon}^p(\boldsymbol{X})$ is differentiable and so is function $F_{p,\epsilon}(\boldsymbol{X})$ in (3.46) as long as $\epsilon > 0$. In such a case, the gradient of $F_{p,\epsilon}(\boldsymbol{X})$ in vectorized form is given by

$$\boldsymbol{g} = \boldsymbol{\Phi}^T (\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}) + \lambda \boldsymbol{u} \tag{3.47}$$

where $\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_1^T & \boldsymbol{u}_2^T & \cdots & \boldsymbol{u}_{n_2}^T \end{bmatrix}^T$ and $\boldsymbol{u}_i$ is the ith column of the gradient matrix $\boldsymbol{U}$

given by

$$\boldsymbol{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n_2} \\ u_{21} & u_{22} & \cdots & u_{2n_2} \\ \vdots & \vdots & \ddots & \vdots \\ u_{n_1 1} & u_{n_1 2} & \cdots & u_{n_1 n_2} \end{bmatrix} \tag{3.48}$$

where

$$u_{ij} = \frac{\partial TV_{p,\epsilon}^{p}(\boldsymbol{X})}{\partial x_{ij}} \quad \text{for} \quad 1 \leq i \leq n_1 \text{ and } 1 \leq j \leq n_2 \tag{3.49}$$

Before we give the explicit expression for $u_{i,j}$, let $\boldsymbol{Q}^h$ and $\boldsymbol{Q}^v$ be matrices of sizes $n_1 \times (n_2 - 1)$ and $(n_1 - 1) \times n_2$, respectively, given by

$$\boldsymbol{Q}^h = \begin{bmatrix} \boldsymbol{q}_1^h & \boldsymbol{q}_2^h & \cdots & \boldsymbol{q}_{n_2}^h \end{bmatrix} \tag{3.50}$$

and

$$\boldsymbol{Q}^v = \begin{bmatrix} \boldsymbol{q}_1^v & \boldsymbol{q}_2^v & \cdots & \boldsymbol{q}_{n_1}^v \end{bmatrix}^T \tag{3.51}$$

where

$$\boldsymbol{q}_i^h = \boldsymbol{x}_i - \boldsymbol{x}_{i+1} \quad \text{for} \quad i = 1, 2, \ldots, n_2 - 1$$

$$\boldsymbol{q}_i^v = \boldsymbol{x}^i - \boldsymbol{x}^{i+1} \quad \text{for} \quad i = 1, 2, \ldots, n_1 - 1$$

and $\boldsymbol{x}_i$ and $(\boldsymbol{x}^i)^T$ denote the ith column and the ith row, respectively, of $\boldsymbol{X}$.

Again, let a matrix $\boldsymbol{Q}$ of size $n_1 \times n_2$ be constructed such that its $\{i, j\}$th component is given by

$$q_{i,j} = \begin{cases} \left[(q_{i,j}^v)^2 + (q_{i,j}^h)^2 + \epsilon^2\right]^{p/2-1} & \text{for} \quad 1 \leq i < n_1, 1 \leq j < n_2 \\ \left[(q_{i,j}^h)^2 + \epsilon^2\right]^{p/2-1} & \text{for} \quad i = n_1, 1 \leq j < n_2 \\ \left[(q_{i,j}^v)^2 + \epsilon^2\right]^{p/2-1} & \text{for} \quad 1 \leq i < n_1, j = n_2 \\ 0 & \text{for} \quad i = n_1, j = n_2 \end{cases}$$

where $q_{i,j}^v$ denotes the $\{i, j\}$th component of the matrix $\boldsymbol{Q}^v$ and $q_{i,j}^h$ denotes the $\{i, j\}$th

component of the matrix $\boldsymbol{Q}^h$. Then, $u_{i,j}$ in (3.49) can be computed using

$$u_{i,j} = \begin{cases} q_{i,j}\left(q^v_{i,j} + q^h_{i,j}\right) & \text{for} \quad i = 1, j = 1 \\ q_{i,j}\left(q^v_{i,j} + q^h_{i,j}\right) - q_{i,j-1}q^h_{i,j-1} & \text{for} \quad i = 1, 1 < j \leq n_2 \\ q_{i,j}\left(q^v_{i,j} + q^h_{i,j}\right) - q_{i-1,j}q^v_{i-1,j} & \text{for} \quad 1 < i \leq n_1, j = 1 \\ q_{i,j}\left(q^v_{i,j} + q^h_{i,j}\right) - q_{i,j-1}q^h_{i,j-1} - q_{i-1,j}q^v_{i-1,j} & \text{for} \quad 1 < i < n_1, 1 < j < n_2 \\ q_{i,j}q^h_{i,j} - q_{i,j-1}q^h_{i,j-1} - q_{i-1,j}q^v_{i-1,j} & \text{for} \quad i = n_1, 1 < j < n_2 \\ q_{i,j}q^v_{i,j} - q_{i,j-1}q^h_{i,j-1} - q_{i-1,j}q^v_{i-1,j} & \text{for} \quad 1 < i < n_1, j = n_2 \\ -q_{i,j-1}q^h_{i,j-1} - q_{i-1,j}q^v_{i-1,j} & \text{for} \quad i = n_1, j = n_2 \end{cases}$$

3.5.3 Optimization

As may be expected, the value of ϵ affects the behaviour of function $TV^p_{p,\epsilon}(\boldsymbol{X})$. Therefore, we propose to use a sequential optimization procedure similar to that in Sec. 3.3.2 to solve the problem in (3.46), in which (3.46) is solved for each value of ϵ using a finite number of iterations of Fletcher-Reeves' CG algorithm [44].

In the kth iteration of the CG algorithm, iterate $\boldsymbol{x}_k$ is updated as in (3.26). The conjugate directions $\{\boldsymbol{d}_k\}$ are computed using (3.29) where gradient $\boldsymbol{g}$ is computed using (3.47). The step sizes $\{\alpha_k\}$ are computed using a line-search described in the next section.

3.5.4 Line search

The step size α_k in (3.26) for the optimization problem in (3.46) can be determined by using a line search based on Banach's fixed-point theorem [45]. Let each of the kth iterate $\boldsymbol{x}_k$ and conjugate direction $\boldsymbol{d}_k$ be divided into n_2 blocks as $\boldsymbol{x}_k = \left[\boldsymbol{x}^T_{k1}\ \boldsymbol{x}^T_{k2}\ \cdots\ \boldsymbol{x}^T_{kn_2}\right]^T$ and $\boldsymbol{d}_k = \left[\boldsymbol{d}^T_{k1}\ \boldsymbol{d}^T_{k2}\ \cdots\ \boldsymbol{d}^T_{kn_2}\right]^T$ where $\boldsymbol{x}_{ki}$ and $\boldsymbol{d}_{ki}$ are the ith blocks of vectors $\boldsymbol{x}_k$ and $\boldsymbol{d}_k$ and let matrices $\boldsymbol{X}_k$ and $\boldsymbol{D}_k$ be constructed as $\boldsymbol{X}_k = [\boldsymbol{x}_{k1}\ \boldsymbol{x}_{k2}\ \cdots\ \boldsymbol{x}_{kn_2}]$ and $\boldsymbol{D}_k = [\boldsymbol{d}_{k1}\ \boldsymbol{d}_{k2}\ \cdots\ \boldsymbol{d}_{kn_2}]$. Under these circumstances, α_k can be obtained as

$$\alpha_k = \arg \underset{\alpha}{\text{minimum}}\ F_{p,\epsilon}(\boldsymbol{X}_k + \alpha \boldsymbol{D}_k)$$

By equating the derivative $dF_{p,\epsilon}(\boldsymbol{X}_k + \alpha \boldsymbol{D}_k)/d\alpha$ to zero, we obtain

$$\alpha = G(\alpha) \tag{3.52}$$

where

$$\begin{aligned} G(\alpha) &= -\frac{a+\lambda\cdot p\cdot b}{c+\lambda\cdot p\cdot d} \\ a &= \boldsymbol{d}_k^T\boldsymbol{\Phi}^T\left(\boldsymbol{\Phi}\boldsymbol{x}_k-\boldsymbol{y}\right) \\ b &= \sum_{i=1}^{n_1-1}\sum_{j=1}^{n_2-1}\hat{b}_{i,j}+\sum_{i=1}^{n_1-1}\hat{b}_{i,n_2}+\sum_{j=1}^{n_2-1}\hat{b}_{n_1,j} \\ c &= \boldsymbol{d}_k^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{d}_k \\ d &= \sum_{i=1}^{n_1-1}\sum_{j=1}^{n_2-1}\hat{d}_{i,j}+\sum_{i=1}^{n_1-1}\hat{d}_{i,n_2}+\sum_{j=1}^{n_2-1}\hat{d}_{n_1,j} \end{aligned} \tag{3.53}$$

In the above equations,

$$\begin{aligned} \hat{b}_{i,j} &= \gamma_{i,j}(\alpha)\left[q_{i,j}^v(d_{i,j}-d_{i+1,j})+q_{i,j}^h(d_{i,j}-d_{i,j+1})\right] \\ \hat{b}_{i,n_2} &= \gamma_{i,n_2}(\alpha)\cdot q_{i,n_2}^v(d_{i,n_2}-d_{i+1,n_2}) \\ \hat{b}_{n_1,j} &= \gamma_{n_1,j}(\alpha)\cdot q_{n_1,j}^h(d_{n_1,j}-d_{n_1,j+1}) \\ \hat{d}_{i,j} &= \gamma_{i,j}(\alpha)\cdot\left[(d_{i,j}-d_{i+1,j})^2+(d_{i,j}-d_{i,j+1})^2\right] \\ \hat{d}_{i,n_2} &= \gamma_{i,n_2}(\alpha)\cdot(d_{i,n_2}-d_{i+1,n_2})^2 \\ \hat{d}_{n_1,j} &= \gamma_{n_1,j}(\alpha)\cdot(d_{n_1,j}-d_{n_1,j+1})^2 \end{aligned}$$

At this point, $\gamma_{i,j}(\alpha)$ can be determined as follows:
For $1\le i<n_1$ and $1\le j<n_2$,

$$\begin{aligned} \gamma_{i,j}(\alpha) &= \Big\{\left[q_{i,j}^v+\alpha(d_{i,j}-d_{i+1,j})\right]^2 \\ &\quad +\left[q_{i,j}^h+\alpha(d_{i,j}-d_{i,j+1})\right]^2+\epsilon^2\Big\}^{p/2-1} \end{aligned}$$

For $1\le i<n_1$ and $j=n_2$,

$$\gamma_{i,j}(\alpha)=\Big\{\left[q_{i,j}^v+\alpha(d_{i,j}-d_{i+1,j})\right]^2+\epsilon^2\Big\}^{p/2-1}$$

For $i=n_1$ and $1\le j<n_2$,

$$\gamma_{i,j}(\alpha)=\Big\{\left[q_{i,j}^h+\alpha(d_{i,j}-d_{i,j+1})\right]^2+\epsilon^2\Big\}^{p/2-1}$$

In the above relations, $q_{i,j}^h$ and $q_{i,j}^v$ are the $\{i,j\}$th components of matrices $\boldsymbol{Q}^h$ and

$\boldsymbol{Q}_v$, respectively, defined in (3.50) and (3.51).

Finding step size α_k amounts to finding a fixed point of function $G_\epsilon(\alpha)$ which, according to Banach's fixed-point theorem [45], can be done by using the recursive relation

$$\alpha_{l+1} = G(\alpha_l) \quad \text{for} \quad l = 1, 2, \ldots \tag{3.54}$$

a sufficient number of times.

3.5.5 Algorithm

The proposed TV_p-regularized least-squares conjugate-gradient (TV_p-RLS-CG) algorithm is summarized in Table 3.5. Parameter p, the number of external iterations T, initial value ϵ_1 and target value ϵ_T of ϵ, initial value λ_1 and target value λ_T of λ, measurement matrix $\boldsymbol{\Phi}$, measurement vector $\boldsymbol{y}$, and error tolerance E_t are supplied as input. A total of $(T-2)$ values of ϵ in between ϵ_1 and ϵ_T are computed using (3.32) and a total of $(T-2)$ values of λ in between λ_1 and λ_T are computed as

$$\lambda_i = \lambda_1 e^{-\beta(i-1)} \quad \text{for} \quad i = 2, 3, \ldots, T-1 \tag{3.55}$$

where $\beta = \log(\lambda_1/\lambda_T)/(T-1)$.

3.5.6 Simulation results

The performance of the proposed TV_p-RLS-CG algorithm was tested through an experiment as detailed below.

In the experiment, an image of cameraman [62] of size 256×256 was vectorized into a vector $\boldsymbol{x}$ of size 65536. A total of eight measurements were chosen as $M = 10000$, 14000, 18000, 22000, 26000, 30000, 34000, and 38000. For each value of M, two values of sub-measurements, namely, M_1 and M_2, were chosen as $M_1 = \text{round}(M/21)$ and $M_2 = \text{round}(20M/21)$. The M_1 discrete-cosine transform (DCT) measurements and M_2 noiselet measurements of $\boldsymbol{x}$ were used as was done by Romberg for the CS of the image in [58]. The TV_p-RLS-CG algorithm was run with $\epsilon_1 = 256$, $\epsilon_T = 1e-2$, $\lambda_1 = 256$, $\lambda_T = 1e-4$, $T = 80$, and $p = 0.1$. The TV_p-RLS-CG and Romberg's algorithm [58] were used to recover the image from the DCT and noiselet measurements. The performance of the reconstruction of an image $\boldsymbol{X}$ was measured

Table 3.5: TV_p-RLS-CG Algorithm

Step 1
Input: p, T, ϵ_1, ϵ_T, λ_1, λ_T, $\boldsymbol{\Phi}$, $\boldsymbol{y}$, and E_t.
Set $\boldsymbol{x}_s = \mathbf{0}$.
Step 2
Compute ϵ_t for $t = 2, 3, \ldots, T-1$ using (3.32) and
λ_t for $t = 2, 3, \ldots, T-1$ using (3.55).
Step 3
Repeat the following for $t = 1, \ldots, T$
i) Set $\epsilon = \epsilon_t$, $\lambda = \lambda_t$, $L_t = 3 + \text{round}(t/4)$.
ii) Set $k = 0$, $\boldsymbol{x}_0 = \boldsymbol{x}_s$, $E_r = 10^{10}$.
iii) Repeat the following while $E_r > E_t$,
a) Compute $\boldsymbol{g}_k$ using (3.47).
b) Compute $\boldsymbol{d}_k$ using (3.29).
c) Compute α_k using (3.54).
d) Compute $\boldsymbol{x}_{k+1}$ using (3.26).
e) Set $k = k + 1$.
f) Exit loop if $k > L_t$.
g) Compute $E_r = ||\alpha_k \boldsymbol{d}_k||_2$.
iv) Set $\boldsymbol{x}_s = \boldsymbol{x}_k$.
Step 4
Output $\boldsymbol{x}^* = \boldsymbol{x}_s$ and stop.

in terms of the peak-signal-to-noise ratio (PSNR) which is defined as

$$PSNR = 20 \log \left(\frac{I_{MAX}}{\sqrt{MSE}} \right) dB \tag{3.56}$$

where $I_{MAX} = 2^b - 1$, b is the number of bits used to encode the components of $\boldsymbol{X}$, and MSE is the mean-square error which is defined as

$$MSE = \frac{1}{n_1 n_2} \left|\left| \boldsymbol{X} - \hat{\boldsymbol{X}} \right|\right|_F^2$$

Matrices $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ each of which is of size $n_1 \times n_2$ represent the original and recovered images, respectively, and $||\cdot||_F$ denotes the Frobenius norm. For the cameraman image used in this experiment we have $I_{MAX} = 255$, $n_1 = 256$, and $n_2 = 256$. The PSNRs for the two algorithms are plotted in Fig. 3.12a with respect to M. The plot indicates that the PSNR obtained with the TV_p-RLS-CG algorithm is better than that obtained with Romberg's algorithm. The CPU time required by the two algorithms to reconstruct the image were measured using MATLAB function *cputime*.

Fig. 3.12b shows the CPU time required by the two algorithms for values of M in the range 10000 to 38000. Clearly, the TV_p-RLS-CG algorithm is considerably faster than Romberg's algorithm.

Next, the TV_p-RLS-CG and Romberg's algorithms were used to recover the same image from 1000 DCT and 20000 noiselet measurements. The TV_p-RLS-CG algorithm was run with $p = 0.1$ and $p = 0.5$. The PSNRs obtained with Romberg's algorithm, the TV_p-RLS-CG algorithm with $p = 0.1$, and the TV_p-RLS-CG algorithm with $p = 0.5$ were found to be 32.16, 32.75, and 32.91 dB, respectively. Fig. 3.13 shows the original and reconstructed images. As can be seen, the images reconstructed with the TV_p-RLS-CG algorithm (bottom left and right panels) are visibly better than that obtained with the Romberg's algorithm (top right panel).

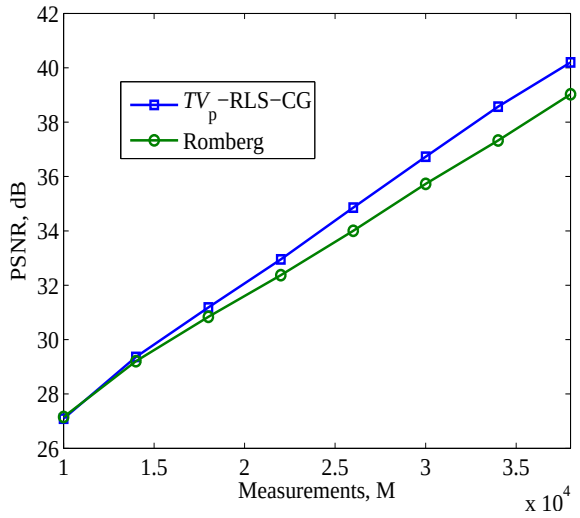
3.6 Conclusions

First, an ℓ_p-pseudonorm regularized least-squares algorithm for the reconstruction of sparse signals from noisy measurements was presented. The ℓ_p-RLS algorithm minimizes an ℓ_p-pseudonorm regularized ℓ_2 error in the nullspace of the measurement matrix and its complement space. The algorithm iteratively takes steps along the bases of the null space and the complement space. The step size is determined using a line search based on Banach's fixed-point theorem [45].

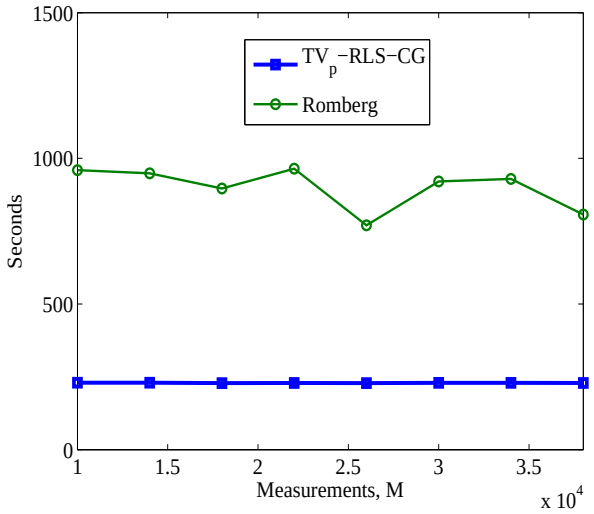
Second, an ℓ_p-pseudonorm regularized least-squares conjugate-gradient algorithm for the reconstruction of sparse signals from noisy and noiseless measurements has been presented. The ℓ_p-RLS-CG algorithm was developed by minimizing an ℓ_p-pseudonorm regularized ℓ_2 error using a sequential CG algorithm. The use of the CG algorithm renders the algorithm suitable for large-sized data. Two extensions of the ℓ_p-RLS-CG algorithm, namely, a) a technique for the application of ℓ_p-RLS-CG algorithm to noiseless data and b) a technique for the optimization of the regularization parameter involved in the ℓ_p-RLS-CG algorithm have also been proposed.

Third, a TV_p-pseudonorm regularized least-squares conjugate-gradient algorithm for the reconstruction of images from noiseless measurements has been presented. The TV_p-RLS-CG algorithm minimizes a TV_p-pseudonorm regularized ℓ_2 error by using the CG Fletcher-Reeves algorithm. The line search involved was carried out using a method based on Banach's fixed-point theorem.

Simulation results have been presented which demonstrate that a) the ℓ_p-RLS algorithm yields improved signal reconstruction performance compared to the UALP



(a) Peak-signal-to-noise ratio



(b) CPU time

Figure 3.12: Reconstruction performance of the TV_p-RLS-CG algorithm versus Romberg's algorithms.

(a) Original image

(b) Reconstructed using Romberg's algorithm

(c) Reconstructed using TV_p-RLS-CG algorithm with $p = 0.1$

(d) Reconstructed using TV_p-RLS-CG algorithm with $p = 0.5$

Figure 3.13: Original image and the images reconstructed by using the TV_p-RLS-CG with $p = 0.1$, $p = 0.5$ and Romberg's algorithms.

[54], IR [34], SL0 [47], and BPDN [52] algorithms and requires a reduced amount of computation compared to the UALP, IR, and BPDN algorithms. Compared to the SL0 algorithm, the ℓ_p-RLS algorithm requires slightly more computation; b) the ℓ_p-RLS-CG algorithm yields improved signal reconstruction performance for various noise levels compared to the SL0, IR, and BPDN algorithms and requires a reduced amount of computation compared to the IR and BPDN algorithms. Compared to the SL0 algorithm, the ℓ_p-RLS-CG algorithm requires slightly more computation. When applied for signal reconstruction from noiseless measurements, the ℓ_p-RLS-CG algorithm yields improved signal reconstruction performance and requires a reduced

amount of computation compared to the SL0, IR, and BPDN algorithms. Also, the proposed technique for the optimization of the regularization parameter in the ℓ_p-RLS-CG algorithm yields significantly improved signal reconstruction performance and requires almost the same amount computation as that required by the ℓ_p-RLS-CG algorithm; c) the TV_p-RLS-CG algorithm yields improved image reconstruction performance and requires a reduced amount of computation compared to a competing algorithm based on the TV norm. The image reconstructed using the TV_p-RLS-CG algorithm is visibly better than that reconstructed by the algorithm based on TV norm.

Chapter 4

Reconstruction of Block-Sparse Signals by Using $\ell_{2/p}$-Regularized Least-Squares Optimization

4.1 Introduction

In conventional CS, the algorithms used to recover sparse signals do not take into account the block structure of the nonzero coefficients of sparse signals where the nonzero coefficients occur in cluster [7, 8, 28, 36–39]. Signals with block-structured coefficients, also called block-sparse signals, are encountered in various applications such as in face recognition [63] [64], motion segmentation [65], processing of multiband signals [66] [66], and measurement of gene expression levels [67] (see [68] for more details). Recently, several authors have developed algorithms based on the minimization of the $\ell_{2/1}$ norm for recovering block-sparse signals where the involved optimization problem is solved by using a second-order cone-programming (SOCP) solver [40–42]. In [42], a block orthogonal matching pursuit (BOMP) algorithm is proposed as an extension of the orthogonal matching pursuit algorithm for block sparse signals [37].

Encouraged by the success of ℓ_p-minimization based algorithms for the reconstruction of sparse signals reported in Chapters 2 and 3, in this Chapter we generalize the $\ell_{2/1}$ norm used for the recovery of block-sparse signals to a pth power $\ell_{2/p}$ pseudonorm, and develop an $\ell_{2/p}$-pseudonorm regularized least-squares algorithm for the reconstruction of block-sparse signals. Two extensions of the proposed algorithm are also presented. First, a method for accelerating the convergence of the algorithm

by reweighting the $\ell_{2/p}$ pseudonorm is presented. Second, a method for using prior information about the locations of nonzero blocks for the reconstruction of block-sparse signals is presented.

4.2 $\ell_{2/p}$-Regularized Least-Squares Optimization Using Fletcher-Reeves' Algorithm

In this section, we present an algorithm for the reconstruction of block-sparse signals by minimizing an $\ell_{2/p}$-pseudonorm regularized ℓ_2 error using a conjugate-gradient (CG) algorithm known as Fletcher-Reeves' algorithm.

4.2.1 Problem formulation

Consider signal $\boldsymbol{x}$ of length N which is divisible by a positive integer d. Let us divide signal $\boldsymbol{x}$ into N/d blocks $\tilde{\boldsymbol{x}}_1$, $\tilde{\boldsymbol{x}}_2$, ..., $\tilde{\boldsymbol{x}}_{N/d}$ and denote $\boldsymbol{x}$ as

$$\boldsymbol{x} = \begin{bmatrix} \tilde{\boldsymbol{x}}_1^T & \tilde{\boldsymbol{x}}_2^T & \cdots & \tilde{\boldsymbol{x}}_{N/d}^T \end{bmatrix}^T \tag{4.1}$$

where

$$\tilde{\boldsymbol{x}}_i = \begin{bmatrix} x_{(i-1)d+1} & x_{(i-1)d+2} & \cdots & x_{(i-1)d+d} \end{bmatrix}^T$$

for $i = 1, 2, \ldots, N/2$ and x_i is the ith component of $\boldsymbol{x}$.

A signal $\boldsymbol{x}$ from (4.1) is said to be K-block sparse if $\boldsymbol{x}$ has K nonzero blocks with $K \ll N/d$. Note that the notation of K-sparse in conventional CS is a special case of K-block sparse with $d = 1$. Recently, it has been shown that improved performance for the reconstruction of K-block sparse signals can be achieved by solving the $\ell_{2/1}$-norm minimization problem

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & ||\boldsymbol{x}||_{2/1} \\ \text{subject to} \quad & \boldsymbol{\Phi x} = \boldsymbol{y} \end{aligned} \tag{4.2}$$

where $||\boldsymbol{x}||_{2/1}$ is the $\ell_{2/1}$ norm of $\boldsymbol{x}$ defined as

$$||\boldsymbol{x}||_{2/1} = \sum_{i=1}^{N/2} ||\tilde{\boldsymbol{x}}_i||_2$$

where $||\tilde{\boldsymbol{x}}_i||_2$ is the ℓ_2 norm of the ith block $\tilde{\boldsymbol{x}}_i$ [40], [41], [42].

Encouraged by the success of the ℓ_p minimization problems studied in Chapters 2 and 3 that minimize an approximate ℓ_p pseudonorm, we generalize the $\ell_{2/1}$ norm $||\boldsymbol{x}||_{2/1}$ to an approximate $\ell_{2/p}$ pseudonorm, denoted as $||\boldsymbol{x}||_{2/p,\epsilon}$, where

$$||\boldsymbol{x}||_{2/p,\epsilon}^p = \sum_{i=1}^{N/d} \left(||\tilde{\boldsymbol{x}}_i||_2^2 + \epsilon^2\right)^{p/2} \tag{4.3}$$

and ϵ is a small scalar. Moreover, we consider the CS signal measurement model that takes the measurement noise into account given as

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{w} \tag{4.4}$$

where $\boldsymbol{w}$ represents measurement noise.

Based on the above formulation, we seek to determine the K-block sparse $\boldsymbol{x}$ with the smallest value of K that would satisfy (4.4), which can be done by solving the optimization problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_{2/p,\epsilon}^p \tag{4.5a}$$

$$\text{subject to} \quad \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} + \boldsymbol{w} \tag{4.5b}$$

with $p < 1$. If we assume that the components of $\boldsymbol{w}$ are i.i.d. Gaussian random variables with zero mean and variance σ^2, then the problem at hand can be formulated as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad ||\boldsymbol{x}||_{2/p}^p \tag{4.6a}$$

$$\text{subject to} \quad ||\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}||_2^2 \leq \upsilon \tag{4.6b}$$

where υ is a constant that depends on the noise variance. This problem can be studied by converting the problem in (4.6) into an unconstrained problem as

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad F_{2/p,\epsilon}(\boldsymbol{x}) = \tfrac{1}{2}\,||\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}||_2^2 + \lambda||\boldsymbol{x}||_{2/p,\epsilon}^p \tag{4.7}$$

where $\lambda > 0$ is a regularization parameter. Function $||\boldsymbol{x}||_{2/p,\epsilon}^p$ with $p < 1$ and a small ϵ measures the inter-block sparsity of signal $\boldsymbol{x}$. Therefore, the optimization problem in (4.7) would yield a solution $\boldsymbol{x}$ with better inter-block sparsity relative to the solution of (4.2).

Note that function $||\boldsymbol{x}||_{2/p,\epsilon}^p$ remains differentiable and so is function $F_{2/p,\epsilon}(\boldsymbol{x})$ in

(4.7) as long as ϵ is kept positive. In effect, for $\epsilon > 0$ the gradient of $F_{2/p,\epsilon}(\boldsymbol{x})$ is given by

$$\boldsymbol{g} = \boldsymbol{\Phi}^T (\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}) + \lambda \boldsymbol{u} \tag{4.8}$$

where $\boldsymbol{u}$ denotes the gradient of $||\boldsymbol{x}||_{2/p,\epsilon}$ which assumes the form

$$\boldsymbol{u} = \left[\tilde{\boldsymbol{u}}_1^T \ \tilde{\boldsymbol{u}}_2^T \ \cdots \ \tilde{\boldsymbol{u}}_{N/d}^T\right]^T \tag{4.9}$$

where $\tilde{\boldsymbol{u}}_i$ is the ith block of $\boldsymbol{u}$. The $\{(i-1)d+j\}$th component of the ith block $\tilde{\boldsymbol{u}}_i$ of $\boldsymbol{u}$ is determined as

$$\tilde{u}_{(i-1)d+j} = p\left(||\tilde{\boldsymbol{x}}_i||_2^2 + \epsilon^2\right)^{p/2-1} x_{(i-1)d+j} \tag{4.10}$$

for $j = 1, 2, \ldots, d$ where $\tilde{\boldsymbol{x}}_i$ is the ith block of $\boldsymbol{x}$.

4.2.2 Optimization

Parameter ϵ in (4.3) has the effect of smoothing function $||\boldsymbol{x}||_{2/p,\epsilon}$ and thereby making function $F_{2/p,\epsilon}(\boldsymbol{x})$ differentiable. As a result, a gradient descent based algorithm can be applied to solve the problem in (4.7). Furthermore, for a large value of ϵ, the region over which function $F_{2/p,\epsilon}(\boldsymbol{x})$ is convex is large, where a minimizer of $F_{2/p,\epsilon}(\boldsymbol{x})$ can be easily located. On the other hand, the global minimizer of function $F_{2/p,\epsilon}(\boldsymbol{x})$ is an accurate approximation of the optimal solution of the problem in (4.7) for a sufficiently small value of ϵ. We, therefore, solve the problem in (4.7) for a sequence of decreasing values of ϵ where the solution of one optimization is used to initialize the next optimization. This approach was found to work well on locating the desired optimal solution of the problem in (4.7).

4.2.3 Use of Fletcher-Reeves' algorithm

In the optimization procedure described above, the problem in (4.7) is solved for a set of values ϵ. For each value of ϵ, we propose to use a finite number of iterations of Fletcher-Reeves' algorithm [44] to minimize the objective function. Fletcher-Reeves' algorithm belongs to the class of conjugate-gradient methods where search directions are conjugate directions computed based on the gradient of $F_{2/p,\epsilon}(\boldsymbol{x})$. In the kth

iteration, iterate $\boldsymbol{x}_k$ is updated as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k \quad \text{for} \quad k = 0, 1, \ldots, L-1 \tag{4.11}$$

where $\boldsymbol{d}_0$, $\boldsymbol{d}_1$, ..., $\boldsymbol{d}_{L-1}$ are L conjugate directions and α_0, α_1, ..., α_{L-1} are L step sizes computed using the line-search technique described in Sec. 4.2.4. The conjugate directions are computed as

$$\boldsymbol{d}_k = \begin{cases} -\boldsymbol{g}_0 & \text{for} \quad k = 0 \\ -\boldsymbol{g}_k + \beta_{k-1}\boldsymbol{d}_{k-1} & \text{for} \quad k = 1, 2, \ldots, L-1 \end{cases} \tag{4.12}$$

where $\boldsymbol{g}_k$ is the gradient computed by using $\boldsymbol{x} = \boldsymbol{x}_k$ in (4.8) and $\beta_k = \beta_n/\beta_d$ where

$$\beta_n = \boldsymbol{g}_{k+1}^T \boldsymbol{g}_{k+1} \quad \text{and} \quad \beta_d = \boldsymbol{g}_k^T \boldsymbol{g}_k$$

for $k = 0, 1, \ldots, L-1$.

4.2.4 Line search

Given parameters p, ϵ, and λ, the step size α_k in Fletcher-Reeves' algorithm is obtained by solving the one-dimensional optimization problem

$$\underset{\alpha}{\text{minimize}} \quad F_{2/p,\epsilon}\left(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k\right) \tag{4.13}$$

By setting the derivative $dF_{2/p,\epsilon}\left(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k\right)/d\alpha$ to zero, we obtain an equation of the form $\alpha = G_\epsilon(\alpha)$ where

$$G_\epsilon(\alpha) = -\frac{\boldsymbol{d}_k^T \boldsymbol{\Phi}^T \left(\boldsymbol{\Phi}\boldsymbol{x}_k - \boldsymbol{y}\right) + \lambda \cdot p \cdot \sum_{i=1}^{N/d} \gamma_i \cdot (\tilde{\boldsymbol{x}}_{ki}^T \tilde{\boldsymbol{d}}_{ki})}{||\boldsymbol{\Phi}\boldsymbol{d}_k||_2^2 + \lambda \cdot p \cdot \sum_{i=1}^{N/d} \gamma_i \cdot (\tilde{\boldsymbol{d}}_{ki}^T \tilde{\boldsymbol{d}}_{ki})} \tag{4.14}$$

In $G(\alpha)$, $\tilde{\boldsymbol{x}}_{ki}$ and $\tilde{\boldsymbol{d}}_{ki}$ are the ith blocks of vectors $\boldsymbol{x}_k$ and $\boldsymbol{d}_k$, respectively, and

$$\gamma_i = \left(||\tilde{\boldsymbol{x}}_i + \alpha \tilde{\boldsymbol{d}}_i||_2^2 + \epsilon^2\right)^{p/2-1} \quad \text{for} \quad i = 1, 2, \ldots, N/d \tag{4.15}$$

Finding step size α_k amounts to finding a fixed point of function $G_\epsilon(\alpha)$ which, according to Banach's fixed-point theorem [45] [46], can be done by using the recursive

relation

$$\alpha_{l+1} = G_\epsilon(\alpha_l) \quad \text{for} \quad l = 1, 2, \ldots \tag{4.16}$$

a sufficient number of times.

4.2.5 Algorithm

The proposed $\ell_{2/p}$-regularized least-squares ($\ell_{2/p}$-RLS) algorithm is summarized in Table 4.1. Parameter p, the number of iterations T, the length of block d, the initial value ϵ_1 and target value ϵ_T of ϵ, and parameter λ are supplied as input.

A total of $T-2$ values of ϵ, for which the optimization in (4.7) is carried out, are chosen as

$$\epsilon_t = \epsilon_1 e^{-\beta(t-1)} \quad \text{for } t = 2, 3, \ldots, T-1 \tag{4.17}$$

where $\beta = \log(\epsilon_1/\epsilon_T)/(T-1)$. The initial conjugate direction is set to $-\boldsymbol{g}$ for each value of ϵ.

For noise-free measurement $\boldsymbol{y}$, a large initial value λ_1 and a small target value λ_T are supplied as input instead of λ. A total of $T-2$ values of λ are chosen as

$$\lambda_t = \lambda_1 e^{-\sigma(t-1)} \quad \text{for } t = 2, 3, \ldots, T-1 \tag{4.18}$$

where $\sigma = \log(\lambda_1/\lambda_T)/(T-1)$.

4.2.6 Simulation Results

To demonstrate the effectiveness of the proposed method, we have carried out two experiments, as detailed below.

In the first experiment, the signal length, N, number of measurements, M, and block length, d, were set to 512, 100, and 8, respectively. A total of sixteen block-sparsity levels $K = 1, 2, \ldots, 16$ were chosen. A K-block sparse signal $\boldsymbol{x}$ was constructed by assigning random values drawn from a normal distribution $\mathcal{N}(0, 1)$ to all the components of K randomly selected blocks of the zero vector of length N. Measurement matrix $\boldsymbol{\Phi}$ of size $M \times N$ was constructed by drawing its elements from $\mathcal{N}(0, 1)$ followed by an orthonormalization step where the rows of $\boldsymbol{\Phi}$ were made orthonormal relative to each other. The measurement was obtained as $\boldsymbol{y} = \boldsymbol{\Phi x}$. With $p = 0.1$, $T = 80$, $\epsilon_1 = 1$, $\epsilon_T = 1e-5$, $\lambda_1 = 1$, $\lambda_T = 1e-10$, and $E_t = 1e-25$, the $\ell_{2/p}$-RLS algorithm was applied and compared with the $\ell_{2/1}$-norm minimization

Table 4.1: $\ell_{2/p}$-RLS Algorithm

Step 1
Input: p, T, ϵ_1, ϵ_T, $\boldsymbol{\Phi}$, $\boldsymbol{y}$, E_t,
λ if measurement $\boldsymbol{y}$ is noisy, and
λ_1 and λ_T if measurement $\boldsymbol{y}$ is noiseless.
Set $\boldsymbol{x}_s = \mathbf{0}$.
Step 2
Compute ϵ_t for $t = 2, 3, \ldots, T-1$ using (4.17) and
λ_t for $t = 2, 3, \ldots, T-1$ using (4.18)
if measurement $\boldsymbol{y}$ is noiseless.
Step 3
For $t = 1, \ldots, T$
i) Set $\epsilon = \epsilon_t$, $L_t = 3 + \text{round}(t/4)$.
ii) If measurement is noiseless, set $\lambda = \lambda_t$.
iii) Set $k = 0$, $\boldsymbol{x}_0 = \boldsymbol{x}_s$, $E_r = 10^{10}$.
iv) Repeat the following while $E_r > E_t$,
a) Compute $\boldsymbol{g}_k$ using (4.8).
b) Compute $\boldsymbol{d}_k$ using (4.12).
c) Compute α_k using (4.16).
d) Compute $\boldsymbol{x}_{k+1}$ using (4.11).
e) Set $k = k + 1$.
f) Exit loop if $k > L_t$.
g) Compute $E_r = ||\alpha_k \boldsymbol{d}_k||_2$.
v) Set $\boldsymbol{x}_s = \boldsymbol{x}_k$.
Step 4
Output $\boldsymbol{x}^* = \boldsymbol{x}_s$ and stop.

using $\ell_{2/1}$ SOCP [69], the iterative reweighted (IR) with $p = 0.1$ [34], the smoothed ℓ_0 pseudonorm (SL0) [47], the BOMP [42], and the BP [52] algorithms. Reconstruction was deemed successful if the maximum absolute error between the original signal, $\boldsymbol{x}$, and recovered signal, $\hat{\boldsymbol{x}}$, measured as $\max_{1 \le i \le N} |x_i - \hat{x}_i|$ was smaller than 0.09, where x_i and $\hat{x}_i$ are the ith components of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, respectively. The percentage of the number of successful reconstructions over 100 runs is plotted in Figure 4.1. It is observed that the performance of the $\ell_{2/p}$-RLS algorithm is significantly better than that of the other algorithms.

In the second experiment, the number of measurements N, block length d, and block-sparsity level K were set to $N = 512$, $d = 8$, and $K = 6$, respectively. A total of nineteen measurements M were selected from 40 to 220 with an increment of 10. A K-block sparse signal and a measurement matrix were constructed, and the measurement was obtained using the procedure used in the first experiment. Figure

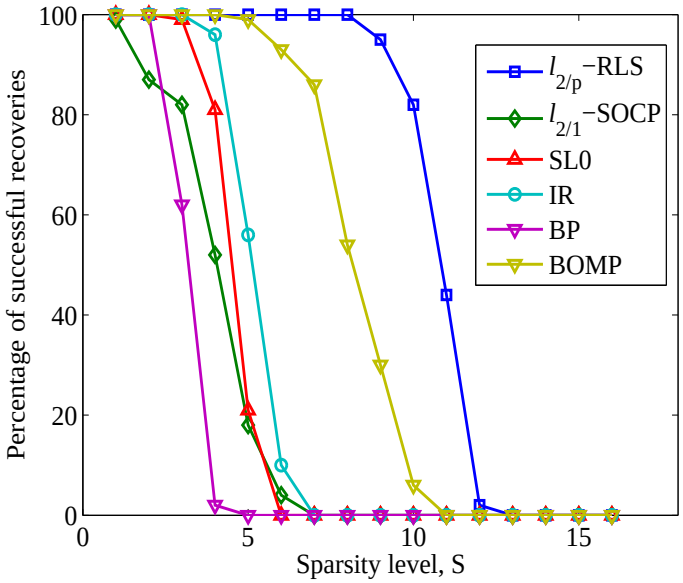


Figure 4.1: Percentage of recovered instances for the $\ell_{2/p}$-RLS ($p = 0.1$), $\ell_{2/1}$-SOCP, SL0, IR ($p = 0.1$), BP, and BOMP algorithms over 100 runs with $N = 512$, $M = 100$, and $d = 8$.

4.2 shows the percentage of recovered instances over 100 runs as a function of the number of measurements. Clearly the $\ell_{2/p}$-RLS algorithm has outperformed the other algorithms.

In the third experiment, the average CPU time required by the algorithms to converge was measured over 100 runs for typical instances with $M = \text{round}(N/2)$, $K = \text{round}(M/2.5d)$, and $d = 8$ for $N = 128$, 256, 512, 1024, 2048, 4096, and 8192. The CPU time was measured on a PC desktop with Intel Core 2 CPU E6850 3.00 GHz processor. The CPU times for the six algorithms are plotted in Fig. 4.3 for values of N in the range 123 to 1024. The CPU times for the $\ell_{2/p}$-RLS, SL0, BP, and BOMP algorithms for values of N in the range 1024 to 8192 are plotted in Fig. 4.4. As can be seen, for $N > 5000$ the proposed $\ell_{2/p}$-RLS algorithm requires the least amount of computation among the algorithms tested.

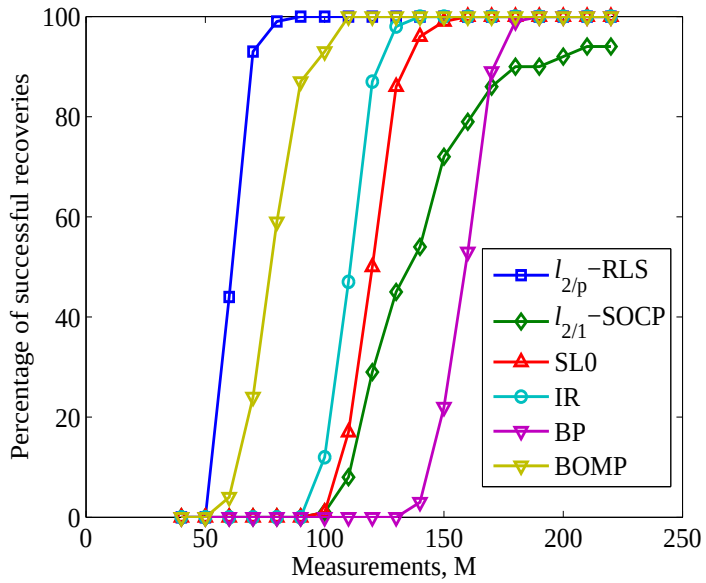


Figure 4.2: Percentage of recovered instances for the $\ell_{2/p}$-RLS ($p = 0.1$), $\ell_{2/1}$-SOCP, SL0, IR ($p = 0.1$), BP, and BOMP algorithms over 100 runs with $N = 512$, $K = 6$, and $d = 8$.

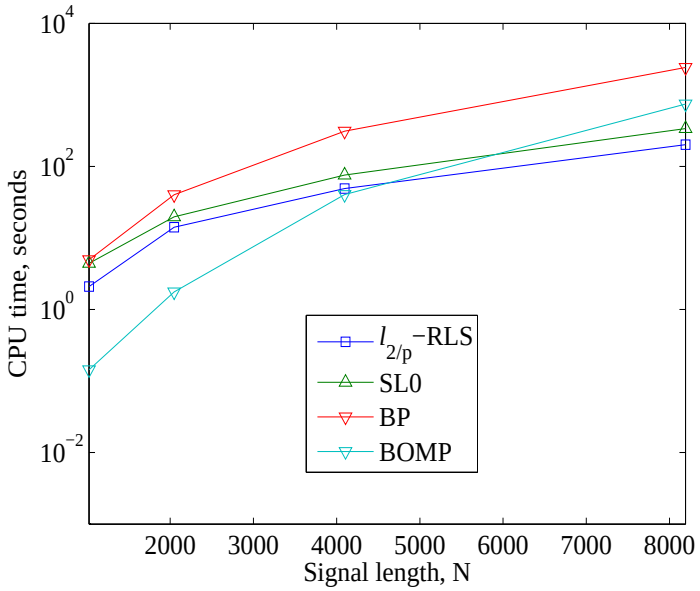


Figure 4.3: Average CPU time required for the $\ell_{2/p}$-RLS ($p = 0.1$), SL0, BP, and BOMP algorithms over 100 runs with $M = N/2$, $K = M/2.5d$.

4.3 A Reweighting Technique for Complexity Reduction

In this section, we present a method of iteratively reweighting function $||\boldsymbol{x}||^p_{2/p,\epsilon}$ in (4.7) so as to reduce the computational complexity of the $\ell_{2/p}$-RLS algorithm pre-

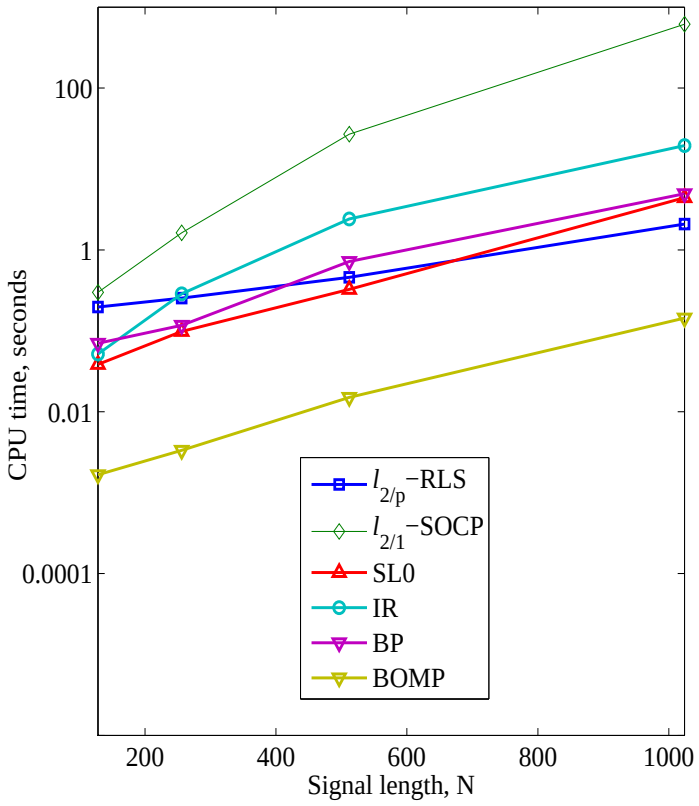


Figure 4.4: Average CPU time required for the $\ell_{2/p}$-RLS ($p = 0.1$), $\ell_{2/1}$-SOCP, SL0, IR ($p = 0.1$), BP, and BOMP algorithms over 100 runs with $M = \text{round}(N/2)$, $K = \text{round}(M/2.5d)$.

sented in Sec. 4.2.

4.3.1 Problem formulation

Consider a weighted, approximate $\ell_{2/p}$ pseudonorm of $\boldsymbol{x}$ given by

$$||\boldsymbol{x}||^p_{2/p,\epsilon,\boldsymbol{w}} = \sum_{i=1}^{N/d} w_i \left(||\tilde{\boldsymbol{x}}_i||_2^2 + \epsilon^2\right)^{p/2} \tag{4.19}$$

where w_i is the weight for the ith term in the summation and $\boldsymbol{w} = \begin{bmatrix} w_1 & w_2 & \cdots & w_{N/d} \end{bmatrix}^T$ is a weight vector.

We propose to reconstruct signal $\boldsymbol{x}$ by solving the optimization problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad F_{2/p,\epsilon,\boldsymbol{w}}(\boldsymbol{x}) = \tfrac{1}{2}\,||\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}||_2^2 + \lambda||\boldsymbol{x}||_{2/p,\epsilon,\boldsymbol{w}}^p \tag{4.20}$$

where $\lambda > 0$ is a regularization parameter. The objective function $F_{2/p,\epsilon,\boldsymbol{w}}(\boldsymbol{x})$ in (4.20) remains differentiable and gradient $\boldsymbol{g}$ can be computed using (4.8) where the $\{(i-1)d+j\}$th component of vector $\boldsymbol{u}$ lying in the ith block $\tilde{\boldsymbol{u}}_i$ is determined as

$$\tilde{u}_{(i-1)d+j} = pw_i \left(||\tilde{\boldsymbol{x}}_i||_2^2 + \epsilon^2\right)^{p/2-1} x_{(i-1)d+j} \tag{4.21}$$

for $j = 1, 2, \ldots, d$ where $\tilde{\boldsymbol{x}}_i$ is the ith block of $\boldsymbol{x}$.

Note that function $||\boldsymbol{x}||_{2/p,\epsilon,\boldsymbol{w}}^p$ in (4.19) is different from function $||\boldsymbol{x}||_{2/p,\epsilon}^p$ in (4.7) in that the behaviour of $||\boldsymbol{x}||_{2/p,\epsilon,\boldsymbol{w}}^p$ depends on the values of the components of the weight vector $\boldsymbol{w}$. As a consequence, the minimization of function $F_{2/p,\epsilon,\boldsymbol{w}}(\boldsymbol{x})$ in (4.20) can be accelerated by using appropriate values for the components of $\boldsymbol{w}$. Below, a method for the optimization of function $F_{2/p,\epsilon,\boldsymbol{w}}(\boldsymbol{w})$ is presented and a method for the estimation of weight vector $\boldsymbol{w}$ is discussed.

4.3.2 Optimization

The following can be inferred from the discussion in Sec. 4.2.1: a) with $\epsilon > 0$, objective function $F_{2/p,\epsilon,\boldsymbol{w}}(\boldsymbol{x})$ in (4.20) becomes differentiable, b) for a sufficiently large value of ϵ, the region over which function $F_{2/p,\epsilon,\boldsymbol{w}}(\boldsymbol{x})$ is convex is large, where a minimizer of $F_{2/p,\epsilon,\boldsymbol{w}}(\boldsymbol{x})$ can be easily located, and c) the global minimizer of function $F_{2/p,\epsilon,\boldsymbol{w}}(\boldsymbol{x})$ is an accurate approximation of the optimal solution of the problem in (4.20) for an appropriate weight vector $\boldsymbol{w}$ and a sufficiently small value of ϵ. In this situation, we propose to solve the problem in (4.20) using the sequential optimization procedure discussed in Sec. 4.2.1. In the proposed approach, the problem in (4.20) is solved sequentially for $\epsilon = \epsilon_1, \epsilon_2, \ldots, \epsilon_T$ where $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_T$ and ϵ_T is a sufficiently small value. Let us call the optimization to solve the problem in (4.20) for $\epsilon = \epsilon_t$ the tth sub-optimization. The tth sub-optimization for $t = 1, 2, \ldots, T$ is carried out using the procedure as described below.

First, the initializer, say, $\boldsymbol{x}_0$, is set as

$$\boldsymbol{x}_0 = \begin{cases} \boldsymbol{0} & \text{if} \quad t = 1 \\ \boldsymbol{x}_s & \text{if} \quad t = 2, 3, \ldots, T \end{cases} \tag{4.22}$$

where vector $\boldsymbol{x}_s$ for the tth sub-optimization is the solution obtained from the $(t-1)$th sub-optimization. The weight vector, say, $\boldsymbol{w}$, for the tth sub-optimization is constructed by computing its ith component as

$$w_i = \begin{cases} 1 & \text{if } t = 1 \\ 1/\left(||\tilde{\boldsymbol{x}}_{0i}||_2 + \delta\right) & \text{if } t = 2, 3, \ldots, T \end{cases} \tag{4.23}$$

for $i = 1, 2, \ldots, N/d$, where $\tilde{\boldsymbol{x}}_{0i}$ is the ith block of the initializer $\boldsymbol{x}_0$. In (4.23), δ is a small positive scalar which helps to keep w_i to a finite value when $||\tilde{\boldsymbol{x}}_{0i}||_2$ is very small or zero.

Next, the tth sub-optimization is carried out using Fletcher-Reeves' CG algorithm. The steps for Fletcher-Reeves' algorithm given in Sec. 4.2.3 are used to solve the problem in (4.20) with two modifications. First, the gradient vectors $\{\boldsymbol{g}_k\}$ used in (4.12) are computed using (4.8) where vector $\boldsymbol{u}$ is computed using (4.21). Second, the step sizes $\{\alpha_k\}$ in (4.11) are computed by using (4.16) where function $G(\alpha)$ is given by

$$G(\alpha) = -\frac{\boldsymbol{d}_k^T \boldsymbol{\Phi}^T \left(\boldsymbol{\Phi}\boldsymbol{x}_k - \boldsymbol{y}\right) + \lambda \cdot p \cdot \sum_{i=1}^{N/d} w_i \cdot \gamma_i \cdot (\tilde{\boldsymbol{x}}_{ki}^T \tilde{\boldsymbol{d}}_{ki})}{||\boldsymbol{\Phi}\boldsymbol{d}_k||_2^2 + \lambda \cdot p \cdot \sum_{i=1}^{N/d} w_i \cdot \gamma_i \cdot (\tilde{\boldsymbol{d}}_{ki}^T \tilde{\boldsymbol{d}}_{ki})} \tag{4.24}$$

where γ_i is as given in (4.15).

4.3.3 Algorithm

The $\ell_{2/p}$-regularized least squares with weighting ($\ell_{2/p}$-RLS-WT) algorithm based on above analysis is shown in Table 4.2.

4.3.4 Simulation results

The convergence, computational complexity, and signal reconstruction performance of the proposed $\ell_{2/p}$-RLS-WT algorithm were compared with that of the $\ell_{2/p}$-RLS algorithm in the context of noiseless measurements by conducting two numerical experiments as detailed below.

In the first experiment, the signal length, N, and block length, d, were set to $N = 1024$ and $d = 8$. The number of measurements, M, and block-sparsity level, K, were computed as $M = N/2$ and $K = \text{round}(M/2.5d)$. A K-block sparse signal $\boldsymbol{x}$ was

Table 4.2: $\ell_{2/p}$-RLS-WT Algorithm

Step 1
Input: p, T, ϵ_1, ϵ_T, $\boldsymbol{\Phi}$, $\boldsymbol{y}$, E_t,
λ if measurement $\boldsymbol{y}$ is noisy, and
λ_1 and λ_T if measurement $\boldsymbol{y}$ is noiseless.
Set $\boldsymbol{x}_s = \mathbf{0}$.
Step 2
Compute ϵ_t for $t = 2, 3, \ldots, T-1$ using (4.17) and
λ_t for $t = 2, 3, \ldots, T-1$ using (4.18)
if measurement $\boldsymbol{y}$ is noiseless.
Step 3
For $t = 1, \ldots, T$
i) Set $\epsilon = \epsilon_t$, $L_t = 3 + \text{round}(t/4)$.
ii) If measurement is noiseless, set $\lambda = \lambda_t$.
iii) Set $k = 0$, $\boldsymbol{x}_0 = \boldsymbol{x}_s$, $E_r = 10^{10}$.
iv) Compute $\boldsymbol{w}$ using (4.23).
v) Repeat the following while $E_r > E_t$,
a) Compute $\boldsymbol{g}_k$ using (4.8) in conjunction with (4.21).
b) Compute $\boldsymbol{d}_k$ using (4.12).
c) Compute α_k using (4.16) in conjunction with (4.24).
d) Compute $\boldsymbol{x}_{k+1}$ using (4.11).
e) Set $k = k + 1$.
f) Exit loop if $k > L_t$.
g) Compute $E_r = ||\alpha_k \boldsymbol{d}_k||_2$.
vi) Set $\boldsymbol{x}_s = \boldsymbol{x}_k$.
Step 4
Output $\boldsymbol{x}^* = \boldsymbol{x}_s$ and stop.

constructed by assigning random values drawn from a normal distribution $\mathcal{N}(0, 1)$ to K randomly selected blocks of a zero vector of length N. Measurement matrix $\boldsymbol{\Phi}$ of size $M \times N$ was constructed by drawing its elements from $\mathcal{N}(0, 1)$ followed by an orthonormalization step where the rows of $\boldsymbol{\Phi}$ were made orthonormal relative to each other. The noiseless measurement was obtained as $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}$. With $p = 0.1$, $T = 80$, $\epsilon_1 = 1$, $\epsilon_T = 1e-5$, $\lambda_1 = 1$, $\lambda_T = 1e-10$, and $E_t = 1e-25$, the $\ell_{2/p}$-RLS and $\ell_{2/p}$-RLS-WT algorithms were applied and compared. A relative error was measured at the end of each sub-optimization as $||\boldsymbol{x}_s - \boldsymbol{x}_0||_2 \,/\, ||\boldsymbol{x}_0||_2$ where $\boldsymbol{x}_0$ and $\boldsymbol{x}_s$ are the initializer and solution, respectively, of the sub-optimization. The relative errors for the two algorithms are plotted in Fig. 4.5 with respect to the iteration numbers. It is observed that the $\ell_{2/p}$-RLS-WT algorithm yields smaller relative error than the $\ell_{2/p}$-RLS algorithm in fewer iterations. Table 4.3 gives the relative error for the two

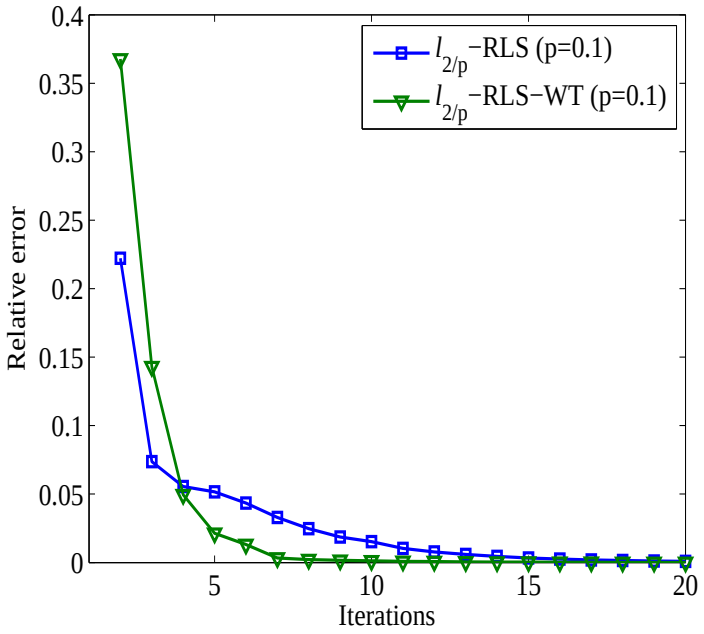


Figure 4.5: Relative error for the $\ell_{2/p}$-RLS-WT and $\ell_{2/p}$-RLS algorithms for $N = 1024$, $M = N/2$, $K = \text{round}(M/2.5d)$.

algorithms for 80 iterations. As can be noticed, the relative error for the $\ell_{2/p}$-RLS-WT algorithm is always smaller than that of the $\ell_{2/p}$-RLS algorithm.

In the second experiment, the signal length, N, was varied as $N = 128$, 256, 512, 1024, 2048, 4096, 8192 and the number of measurements, M, the block-sparsity level, K, and the block length, d, were set to $M = N/2$, $K = \text{round}(M/2.5)$, $d = 8$. A K-block-sparse signal $\boldsymbol{x}$ and a measurement matrix $\boldsymbol{\Phi}$ were constructed, and measurement $\boldsymbol{y}$ was taken as in the first experiment. The $\ell_{2/p}$-RLS and $\ell_{2/p}$-RLS-WT algorithms were used to reconstruct signal $\boldsymbol{x}$ from $\boldsymbol{y}$ for all the values of N. The amount of computation required by the two algorithms was measured using MATLAB function *cputime* over 100 runs. The average CPU times for the two algorithms are plotted in Fig. 4.6 which shows that the amount of computation required by the $\ell_{2/p}$-RLS-WT algorithm is less than that required by the $\ell_{2/p}$-RLS algorithm. We also note that the amount of computational saving achieved by using the $\ell_{2/p}$-RLS-WT algorithm increases with the value of N.

In the third experiment, the signal length, N, number of measurements, M, and block length, d, were set to $N = 512$, $M = 100$, $d = 8$. A total of sixteen block-sparsity levels $K = 1, 2, \ldots, 16$ were chosen. A K-block sparse signal $\boldsymbol{x}$ and a mea-

Table 4.3: Relative error due to the $\ell_{2/p}$-RLS-WT and $\ell_{2/p}$-RLS algorithms

Iteration number	Relative error	
	$\ell_{2/p}$-RLS-WT	$\ell_{2/p}$-RLS
1	Infinity	Infinity
5	$2.1308e-002$	$5.1633e-002$
10	$1.2431e-003$	$1.5240e-002$
15	$2.7641e-004$	$3.3704e-003$
20	$6.6884e-005$	$8.2619e-004$
25	$1.6160e-005$	$2.0585e-004$
30	$3.9139e-006$	$5.1161e-005$
35	$9.4723e-007$	$1.2735e-005$
40	$2.3021e-007$	$3.1756e-006$
45	$5.6014e-008$	$7.9197e-007$
50	$1.3651e-008$	$1.9764e-007$
55	$3.3303e-009$	$4.9314e-008$
60	$8.1364e-010$	$1.2310e-008$
65	$1.9904e-010$	$3.0737e-009$
70	$4.8749e-011$	$7.6765e-010$
75	$1.1954e-011$	$1.9175e-010$
80	$2.9348e-012$	$4.7907e-011$

surement matrix $\boldsymbol{\Phi}$ were constructed and measurement $\boldsymbol{y}$ was taken as in the first experiment. Both the $\ell_{2/p}$-RLS-WT and $\ell_{2/p}$-RLS algorithms were used to reconstruct $\boldsymbol{x}$ from $\boldsymbol{y}$. Reconstruction was deemed successful if the maximum absolute error between the original signal $\boldsymbol{x}$ and the recovered signal $\hat{\boldsymbol{x}}$ measured as $\max_{1\leq i\leq N}|x_i - \hat{x}_i|$ was smaller than 0.09, where x_i and $\hat{x}_i$ are the ith components of $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, respectively. The percentages of successful recoveries are plotted in Fig. 4.7 which shows that the reconstruction performance of the $\ell_{2/p}$-RLS-WT algorithm is slightly worse than that of the $\ell_{2/p}$-RLS algorithm. This indicates that the improvement in the convergence of the $\ell_{2/p}$-RLS-WT algorithm has been achieved with a small loss in signal reconstruction performance.

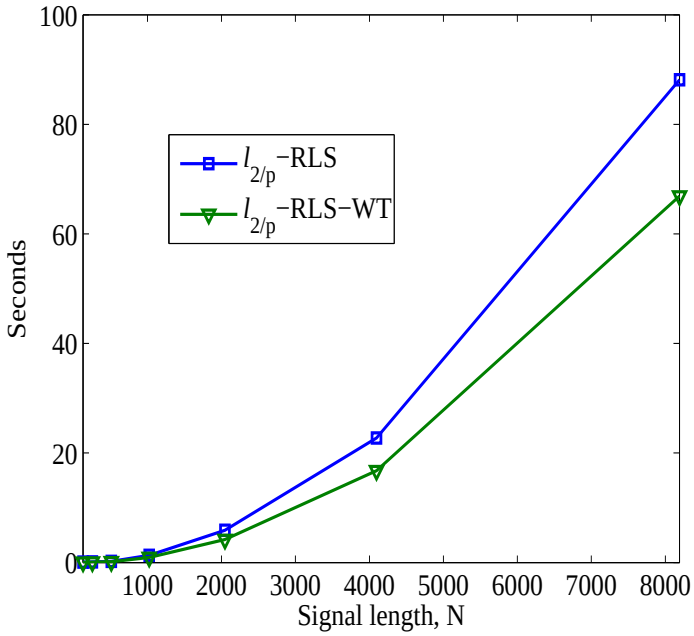


Figure 4.6: Average CPU time required for the $\ell_{2/p}$-RLS-WT and $\ell_{2/p}$-RLS algorithms for $M = N/2$, $K = \text{round}(M/2.5d)$.

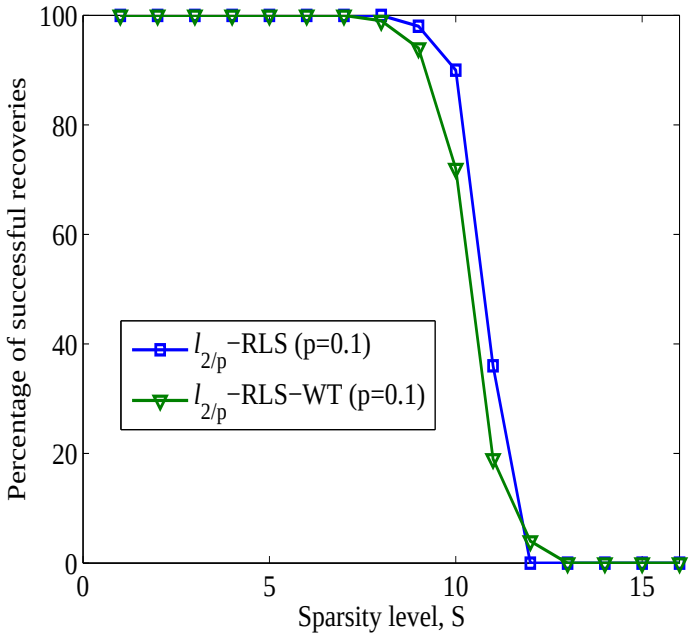


Figure 4.7: Percentage of recovered instances for the $\ell_{2/p}$-RLS-WT and $\ell_{2/p}$-RLS algorithms over 100 runs with $N = 512$, $M = 100$, and $d = 8$.

4.4 Reconstruction of Block-Sparse Signals Using Prior Information

In this section, we present a technique that uses partial information about the locations of the nonzero blocks to improve the reconstruction performance of the $\ell_{2/p}$-RLS algorithm described in Sec. 4.2.

Let $\mathcal{S} \subset \{1, 2, \ldots, \mathrm{N/d}\}$ be the set of indices of blocks of $\boldsymbol{x}$ where the coefficients are nonzero. Assume that partial information about the locations of the nonzero blocks of $\boldsymbol{x}$ is available, i.e., a subset $\hat{\mathcal{S}}$ of $\mathcal{S}$ is known *a priori*. In such a case, the minimization of function $||\boldsymbol{x}||_{2/p,\epsilon}^p$ in (4.3) can be reduced to the minimization of function $||\boldsymbol{x}||_{p,\epsilon,s}^p$ where

$$||\boldsymbol{x}||_{p,\epsilon,s}^p = \sum_{\substack{i=1 \\ i \notin \hat{\mathcal{S}}}}^{N/d} \left(||\tilde{\boldsymbol{x}}_i||_2^2 + \epsilon^2\right)^{p/2} \tag{4.25}$$

Under this circumstance, we propose to reconstruct signal $\boldsymbol{x}$ by solving the problem

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad F_{p,\epsilon,s}(\boldsymbol{x}) = \tfrac{1}{2}\,||\boldsymbol{\Phi x} - \boldsymbol{y}||_2^2 + \lambda ||\boldsymbol{x}||_{p,\epsilon,s}^p \tag{4.26}$$

The objective function $F_{p,\epsilon,s}(\boldsymbol{x})$ in (4.26) is differentiable as long as $\epsilon > 0$ and a gradient descent based algorithm can be applied for its optimization. The gradient of function $F_{p,\epsilon,s}(\boldsymbol{x})$ is given by

$$\boldsymbol{g} = \boldsymbol{\Phi}^T (\boldsymbol{\Phi x} - \boldsymbol{y}) + \lambda \boldsymbol{g}_s \tag{4.27}$$

where the $\{(i-1)d + j\}$th component of vector $\boldsymbol{g}_s$ lying in its ith block is given by

$$g_{s,(i-1)d+j} = \begin{cases} p\left(||\boldsymbol{x}_i||_2^2 + \epsilon^2\right)^{p/2-1} x_{(i-1)d+j} & \text{if} \quad i \notin \hat{\mathcal{S}} \\ 0 & \text{if} \quad i \in \hat{\mathcal{S}} \end{cases} \tag{4.28}$$

for $j = 1, 2, \ldots, d$ where $\tilde{\boldsymbol{x}}_i$ is the ith block of $\boldsymbol{x}$.

As can be noticed, the role of parameter ϵ in the objective function $F_{p,\epsilon,s}(\boldsymbol{x})$ in (4.26) is similar to that in function $F_{2/p,\epsilon}(\boldsymbol{x})$ in (4.7). Therefore, the sequential optimization procedure described in Sec. 4.2.1 is used in conjunction with Fletcher-Reeves' CG algorithm described in Sec. 4.2.3 to solve the problem in (4.26) with two modifications. First, the gradient computed using (4.27) is used in (4.12) to compute

the conjugate directions. Second, the step size α_k in (4.11) is computed using a line search based in Banach's fixed-point theorem which solves the optimization problem

$$\underset{\alpha}{\text{minimize}} \quad F_{p,\epsilon,s}(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k) \tag{4.29}$$

By equating the derivative $dF_{p,\epsilon,s}(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)/d\alpha$ to zero followed by a minor rearrangement, we obtain the relation

$$\alpha = G(\alpha)$$

where

$$G(\alpha) = -\frac{\boldsymbol{d}_k^T(\boldsymbol{\Phi}\boldsymbol{x}_k - \boldsymbol{y}) + \lambda p \sum\limits_{i \notin \hat{\mathcal{S}}} \gamma_i \cdot (\tilde{\boldsymbol{x}}_{ki}^T \tilde{\boldsymbol{d}}_{ki})}{||\boldsymbol{\Phi}\boldsymbol{d}_k||_2^2 + \lambda p \sum\limits_{i \notin \hat{\mathcal{S}}} \gamma_i \cdot (\tilde{\boldsymbol{d}}_{ki}^T \tilde{\boldsymbol{d}}_{ki})} \tag{4.30}$$

Parameter γ_i is given in (4.15) and $\tilde{\boldsymbol{x}}_{ki}$ and $\tilde{\boldsymbol{d}}_{ki}$ are the ith blocks of $\boldsymbol{x}_k$ and $\boldsymbol{d}_k$, respectively.

From what we have done in Sec. 4.2.4, it follows that the step sizes $\{\alpha_k\}$ in (4.11) are computed using (4.16) where function $G(\alpha)$ is given by (4.30).

The ℓ_p-RLS algorithm with the proposed extension for the use of partial prior information about the locations of the nonzero blocks of the signals can be implemented as shown in Table 4.1 with two modifications, namely, a) the gradient vector $\boldsymbol{g}_k$ in Step 3.iv.a in Table 4.1 is computed using (4.27) and b) the step size α_k in Step 3.iv.c is computed using (4.16) where the function $G(\alpha)$ is evaluated using (4.30).

4.4.1 Simulation results

The performance of the ℓ_p-RLS algorithm with the proposed extension for the use of a partial prior information about the locations of nonzero blocks was evaluated through a numerical experiment as detailed below.

The signal length, N, number of measurements, M, and block length, d, were set to $N = 512$, $M = 144$, and $d = 8$. Six block-sparsity levels were chosen as $K = 13, 14, 15, 16, 17, 18$. A K-block sparse signal $\boldsymbol{x}$ was constructed by assigning random values drawn from the Gaussian distribution $\mathcal{N}(0, 1)$ to a total of K randomly selected blocks of the zero vector of length N. Three levels of prior information were selected as $K_1 = \text{minimum } \{K, 6\}$, $K_2 = \text{minimum}\{K, 16\}$, and $K_3 = \text{minimum}\{K, 18\}$. A measurement matrix $\boldsymbol{\Phi}$ of size $M \times N$ was constructed by drawing its elements

from $\mathcal{N}(0,1)$ followed by a orthonormalization step whereby the rows of $\boldsymbol{\Phi}$ are made orthonormal to each other. The noiseless measurement was taken as $\boldsymbol{y} = \boldsymbol{\Phi x}$. The $\ell_{2/p}$-RLS algorithm was run *without* and *with* prior information about the locations of K_1, K_2, and K_3 nonzero blocks. The parameters for the ℓ_p-RLS algorithm were set to $p = 0.1$, $T = 80$, $\epsilon_1 = 1$, $\epsilon_T = 1e-5$, $\lambda_1 = 1$ $\lambda_T = 1e-10$, $E_t = 1e-25$ and L_t in Step 3.i in Table 4.1 was computed as $L_t = 10 + \text{round}(t/4)$. A signal was assumed recovered if the maximum absolute valued error measured as $\underset{i}{\text{maximum}}\, |x_i - \hat{x}_i|$ was less than 0.09, where x_i and $\hat{x}_i$ are the ith components of the original signal $\boldsymbol{x}$ and the reconstructed signal $\hat{\boldsymbol{x}}$, respectively. The percentages of successful recoveries in the four cases are shown in Fig. 4.8. It is observed that the percentage of recovered instances by the proposed method increases with the increase in prior information about the number of nonzero blocks of signal.

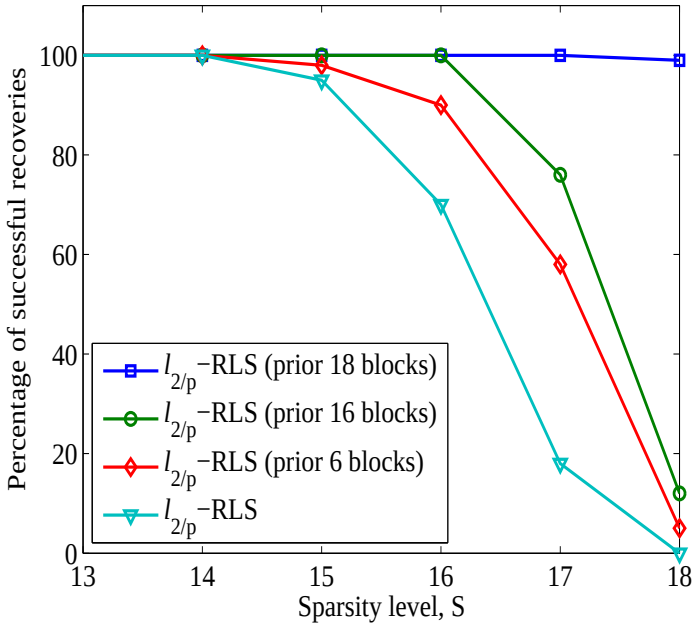


Figure 4.8: Percentage of recovered instances for the $\ell_{2/p}$-RLS algorithm without and with prior information about the locations of 6, 16, and 18 nonzero blocks over 100 runs with $N = 512$, $M = 144$, and $d = 8$.

4.5 Conclusions

The $\ell_{2/p}$-RLS algorithm for the reconstruction of block-sparse signals and two extensions have been presented.

First, the $\ell_{2/p}$-RLS algorithm was developed by generalizing the widely used $\ell_{2/1}$ norm to a pth powered $\ell_{2/p}$ pseudonorm, casting the signal reconstruction problem as an $\ell_{2/p}$-regularized least squares optimization problem, and solving the resulting optimization problem by using a sequential optimization procedure in conjunction with Fletcher-Reeves' CG algorithm. The line-search involved in Fletcher-Reeves' algorithm was carried out by using an algorithm based on Banach's fixed-point theorem.

Second, an extension of the $\ell_{2/p}$-RLS algorithm for reducing its computational complexity by reweighting the $\ell_{2/p}$ pseudonorm has been presented. The use of reweighting has helped to force the block-sparsity in the solution and thereby to reduce the amount of computation.

Third, an extension of the $\ell_{2/p}$-RLS algorithm for the use of partial information about the locations of the nonzero blocks has been presented. The use of prior information has helped to improve the signal reconstruction performance of the $\ell_{2/p}$-RLS algorithm.

Simulation results have been presented which demonstrate that a) the $\ell_{2/p}$-RLS algorithm yields improved signal reconstruction performance and requires a reduced amount of computation for large-sized data relative to several competing algorithms, b) the proposed reweighting technique helps the $\ell_{2/p}$-RLS algorithm to converge to a solution faster, and c) the proposed technique for the use of prior information helps to improve the signal reconstruction performance of the $\ell_{2/p}$-RLS algorithm.

Chapter 5

Conclusions and Future Directions

Below we draw conclusions about the research done and point out several future directions.

5.1 Conclusions

Three major problems in compressive sensing (CS) have been investigated in Chapter 2, Chapter 3, and Chapter 4.

In Chapter 2, three algorithms, namely, the nullspace reweighted approximate ℓ_0 (NRAL0), unconstrained approximate ℓ_p (UALP), and unconstrained approximate ℓ_p conjugate-gradiet (UALP-CG) algorithms, for the reconstruction of sparse signals from noise-free measurements have been presented. The NRAL0 algorithm is based on the minimization of an approximate ℓ_0 pseudonorm whereas the UALP and UALP-CG algorithms are based on the minimization of an approximate ℓ_p pseudonorm. All three algorithms work in the null space of the measurement matrix and thus they satisfy the equality constraint of the involved optimization problems. The optimization in the NRAL0 and UALP algorithms is carried out using a sequential version of the quasi-Newton algorithm in conjunction with the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update formula and the optimization in the UALP-CG algorithm is carried out by using a sequential conjugate-gradient (CG) technique. The step size for the NRAL0 algorithm is computed using Fletcher's line search. For the UALP algorithm, a new computationally efficient line search based on Banach's fixed-point theorem has been developed. The NRAL0, UALP, and UALP-CG algorithms have been found to offer improved signal reconstruction performance relative to that of the iterative

reweighted (IR), smoothed ℓ_0 (SL0), and ℓ_1-minimization based algorithms. The amount of computation required by the NRAL0 algorithm has been found to be less than that required by the IR and ℓ_1 minimization based algorithms and slightly more than that required by the SL0 algorithm. Both the UALP and UALP-CG algorithms are computationally more efficient than the IR, SL0, and ℓ_1-minimization based algorithms.

In Chapter 3, two algorithms, namely, the ℓ_p-regularized least-squares (ℓ_p-RLS) and ℓ_p-regularized least-squares conjugate-gradient (ℓ_p-RLS-CG) algorithms, for the reconstruction of sparse signals from noisy measurements and one algorithm, namely, the TV_p-regularized least-squares conjugate-gradient (TV_p-RLS-CG) algorithm, for the reconstruction of sparse images from noisy measurements have been presented. Both the ℓ_p-RLS and ℓ_p-RLS-CG algorithms are based on minimizing an ℓ_p-regularized ℓ_2 error where the involved optimization problem is unconstrained. The ℓ_p-RLS algorithm solves the optimization problem by taking descent steps in the null space of the measurement matrix and its complement space. The line search is carried out by using a technique based on Banach's fixed-point theorem. The ℓ_p-RLS-CG algorithm solves the same problem in the time domain by using a sequential CG algorithm. A method for applying the ℓ_p-RLS-CG algorithm for signal reconstruction from noiseless measurements has also been presented. Next, the TV norm has been generalized to its nonconvex version which is called the TV_p pseudonorm. Then, the TV_p-RLS-CG algorithm has been developed by minimizing a TV_p-pseudonorm regularized ℓ_2 error. The minimization is carried out by using a sequential Fletcher-Reeves' CG technique. Improvement in the performances of the proposed ℓ_p-RLS, ℓ_p-RLS-CG, and TV_p-RLS-CG algorithms relative to several competing algorithms has been demonstrated through several numerical experiments. Also, a technique for using the ℓ_p-RLS-CG algorithm by optimizing the involved regularization parameter has been proposed and improvement in the signal reconstruction performance by the proposed technique has been shown using numerical experiments.

In Chapter 4, an algorithm for the reconstruction of block sparse signals, namely, the $\ell_{2/p}$-regularized least-squares ($\ell_{2/p}$-RLS) algorithm has been presented. The $\ell_{2/p}$-RLS algorithm has been developed by generalizing the $\ell_{2/1}$ norm to the $\ell_{2/p}$ pseudonorm for block sparse signals and minimizing an $\ell_{2/p}$-pseudonorm regularized ℓ_2 error by using a sequential Fletcher-Reeve's CG algorithm. Two extensions of the $\ell_{2/p}$-RLS algorithm have also been presented, namely, a reweighting technique for reducing the amount of computation and a technique for incorporating partial information about

the location of nonzero blocks. The signal reconstruction performances and computational complexity of the proposed algorithm and its extensions have been evaluated through numerical experiments.

Due to the complexity of the problems at hand, it would be very difficult to say very much about the global optimality of the solutions. An exhaustive search would be out of the question in view of the size of the problems. The best that can be said, based on our simulations, would be that the methods would yield very acceptable solutions in many practical situations.

5.2 Future directions

There are several interesting open problems that are closely related to the work presented.

1. The ℓ_p-optimization based algorithms presented offer improved performance relative to ℓ_1-optimization based algorithms. However, indepth theoretical studies of the proposed algorithms, especially in terms of convergence properties and the relation of convergence to parameters p, ϵ, and λ, are not available. Studies along these lines would provide more insights on ℓ_p-optimization based algorithms and pave the way to the development of more efficient algorithms for sparse signal reconstruction. The same can also be said for the TV_p-optimization and $\ell_{2/p}$-optimization based algorithms presented in Chapters 3 and 4, respectively.

2. CS could be effectively applied to many areas and the application of the proposed algorithms should be explored. Some of the possible applications are briefly discussed below:

   (a) Magnetic resonance imaging (MRI) is a medical imaging technique that can be used for assessing brain disease, spinal disorder, cardiac function, and musculoskeletal damage [70]. The sparsity in the images and the slow data-acquisition process associated with MRI render the CS technique a natural fit for MRI [71]. Recently, a CS technique for MRI [72] based on solving a nonconvex optimization problem has been shown to offer improved reconstruction performance relative to the ℓ_1-optimization based algorithm.

(b) In biomedical signal processing, electrocardiogram (ECG) and electroencephelogram (EEG) data are processed and used for the automated analysis of the heart and brain activities, respectively [73]. Recently, ECG and EEG data have been shown to exhibit sparse representation with respect to appropriate bases and CS techniques have been applied to recover ECG and EEG signals from small datasets [74–76].

(c) There are many other areas where CS can effectively be applied, see [77] for more information.

3. Bayesian learning has been an effective technique for solving inverse problems in the areas of computer vision and machine learning. Recently, this technique has also been applied for the reconstruction of sparse signals in CS [78, 79]. It would be of interest to investigate the applicability of the concepts developed in this dissertation for Bayesian learning.

4. Iterative shrinkage based methods for the reconstruction of sparse signals have been known for their good computational efficiency and have recently been applied for the reconstruction of sparse signals [57, 80, 81]. State-of-the-art iterative shrinkage based algorithms employ either hard thresholding based on minimizing the ℓ_0 pseudonorm or soft thresholding based on minimizing the ℓ_1 norm. It would be interesting to investigate iterative shrinkage based methods using ℓ_p-pseudonorm minimization with $0 < p < 1$.

5. Dictionary learning is a technique for determining a set of vectors called 'atoms' so that signals or images can be represented as linear combinations of small numbers of atoms. This technique has recently attracted increased attention from the computer vision and machine learning communities [82–88]. It would be interesting to investigate extensions of the algorithms presented for dictionary learning.

Bibliography

[1] B. F. Logan, *Properties of High-Pass Signals*. Ph.D. Thesis, Columbia University, 1965.

[2] F. Santosa and W. W. Symes, "Linear inversion of band-limited reflection seismograms," *SIAM J. Sci. Stat. Comput*, vol. 7, no. 4, pp. 1307–1330, 1986.

[3] D. L. Donoho and P. B. Stark, "Uncertainty principles and signal recovery," *SIAM J. Appl. Math.*, vol. 49, pp. 906–931, Jun. 1989.

[4] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, pp. 489–509, Feb. 2006.

[5] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, Apr. 2006.

[6] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies," *IEEE Trans. Inf. Theory*, vol. 52, pp. 5406–5425, Dec. 2006.

[7] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[8] E. Candes and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Annals of Statistics*, vol. 35, pp. 2313–2351, 2007.

[9] D. L. Donoho and J. Tanner, "Exponential bounds implying construction of compressed sensing matrices, error-correcting codes, and neighborly polytopes by random sampling," *IEEE Trans. Inf. Theory*, vol. 56, pp. 2002–2016, Mar. 2010.

[10] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Process.*, vol. 47, pp. 187–200, Jan. 1999.

[11] I. F. Gorodnitsky, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Trans. Signal Process.*, vol. 45, pp. 600–616, Mar. 1997.

[12] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Jour. Royal Statist. Society Ser.*, vol. 58, no. 1, pp. 267–288, 1996.

[13] M. Elad and A. M. Bruckstein, "A generalized uncertainty principle and sparse representation in pairs of bases," *IEEE Trans. Inf. Theory*, vol. 48, pp. 2558–2567, Sept. 2002.

[14] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization," *PNAS*, vol. 100, pp. 2197–2202, Mar. 2003.

[15] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, pp. 3320–3325, Dec. 2003.

[16] J.-J. Fuchs, "On sparse representations in arbitrary redundant bases," *IEEE Trans. Inf. Theory*, vol. 50, pp. 1341–1344, Jun. 2004.

[17] E. J. Candes and J. Romberg, "Sparsity and incoherence in compressive sensing," *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007.

[18] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Sig. Proc. Magazine*, pp. 83–91, Mar. 2008.

[19] E. J. Candes, M. Rudelson, T. Tao, and R. Vershynin, "Error correction via linear programming," *IEEE Symposium on Foundations of Comp. Sci.*, pp. 668–681, Oct. 2005.

[20] J. L. Paredes, G. R. Arce, and Z. Wang, "Ultra-wideband compressed sensing: Channel estimation," *IEEE Jour. Selected Topics Signal Process.*, vol. 1, pp. 383–395, Oct. 2007.

[21] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, "Signal processing with compressive measurements," *IEEE Jour. Selected Topics Signal Process.*, vol. 4, pp. 445–460, Apr. 2010.

[22] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse reconstruction perspective for source localization with sensor arrays," *IEEE Trans. Signal Process.*, vol. 53, pp. 3010–3022, Aug. 2005.

[23] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. Commun.*, vol. 50, pp. 374–377, Mar. 2002.

[24] Y. Elrich, A. Gordon, M. Brand, G. J. Hannon, and P. P. Mitra, "Compressed genotyping," *IEEE Trans. Inf. Theory*, vol. 56, pp. 706–723, Feb. 2010.

[25] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, pp. 2231–2242, Oct. 2004.

[26] H. Nyquist, "Certain topics in telegraph transmission theory," *Trans. Amer. Inst. Electr. Eng.*, vol. 47, pp. 617–644, Apr. 1928.

[27] C. E. Shannon, "Communication in the presence of noise," *Proceed. IRE*, vol. 37, pp. 10–21, Jan. 1949.

[28] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, pp. 21–30, Mar. 2008.

[29] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Trans. Inf. Theory*, vol. 47, pp. 2845–2862, Nov. 2001.

[30] E. J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Signal Process.*, vol. 51, pp. 4203–4215, Dec. 2005.

[31] E. Candes and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions," *Foundations of Computational Mathematics*, vol. 6, no. 2, 2006.

[32] B. K. Natrajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.

[33] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 770–710, 2007.

[34] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," *IEEE Inter. Conf. Acoustics, Speech, Signal Processs*, pp. 3869–3872, 2008.

[35] M. Zibuleusky and M. Elad, "L1-l2 optimization in signal and image processing," *IEEE Signal Process. Mag.*, pp. 76–88, May. 2010.

[36] R. G. Baraniuk, "Compressive sensing," *IEEE Signal Process. Mag.*, vol. 24, pp. 118–120, Jul. 2007.

[37] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, pp. 4655–4666, Dec. 2007.

[38] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *App. Comp. Harmonic Anal.*, vol. 26, pp. 301–321, May. 2009.

[39] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Jour. Select. Topics Signal Process.*, vol. 1, pp. 586–597, Dec. 2007.

[40] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *IEEE Trans. Signal Process.*, vol. 57, pp. 3075–3085, Aug. 2009.

[41] Y. C. Eldar and M. Mishali, "Robust recovery of signals from a structured union of subspaces," *IEEE Trans. Inf. Theory*, vol. 55, pp. 5302–5216, Nov. 2009.

[42] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: uncertainty relations and efficient recovery," *IEEE Trans. Inf. Theory*, vol. 58, pp. 3042–3054, Jun. 2010.

[43] M. Lobo, L. Vanderberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, pp. 193–228, 1998.

[44] A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*. Springer, 2006.

[45] R. P. Agarwal, M. Meehan, and D. O'regan, *Fixed Point Theory and Applications*. Cambridge University Press, 2001.

[46] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*. SIAM, 1995.

[47] H. Mohimani, M. Babie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed ℓ_0-norm," *IEEE Trans. Signal Process.*, vol. 57, pp. 289–301, Jan. 2009.

[48] G. H. Golub and C. F. V. Loan, *Matrix Computations*. The Johns Hopkins University Press, 3rd ed., 1996.

[49] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, 2008.

[50] R. Fletcher, *Practical Methods of Optimization, 2nd ed.* Wiley, New York, 1987.

[51] R. Fletcher, *Practical Methods of Optimization, vol. 1.* Wiley, New York, 1980.

[52] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientif. Comput.*, vol. 20, no. 1, pp. 33–61, 1999.

[53] S. S. Chen, *Basis Pursuit*. Ph.D. Thesis, Stanford Univerisity, 1995.

[54] J. K. Pant, W.-S. Lu, and A. Antoniou, "Reconstruction of sparse signals by minimizing a re-weighted approximate ℓ_0-norm in the null space of the measurement matrix," *IEEE Inter. Midwest Symp. on Circuits-Syst*, pp. 430–433, 2010.

[55] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Natl. Bur. Stand.*, vol. 49, pp. 409–436, 1952.

[56] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D.*, vol. 60, pp. 259–268, 1992.

[57] A. Beck and M. Teboulle, "Fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[58] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, pp. 14–20, Mar. 2008.

[59] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, pp. 2419–2434, Nov. 2009.

[60] UCLA, "Computational and applied mathematics reports, a web site maintained by the math dept. UCLA," Dec. 2011.

[61] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[62] Imageprocessing.com, "URL: `http://www.imageprocessingplace.com/root\_files\_V3/image\_databases.htm`," [Online; accessed 25-January-2012].

[63] R. Basri and D. Jacobs, "Lambertian reflectances and linear subspaces," *IEEE Trans. Pattern Analys. Mach. Intellig.*, vol. 25, pp. 398–390, Feb. 2003.

[64] A. Y. Yang, J. Wright, Y. Ma, and S. Sastry, "Feature selection in face recognition: A sparse representation perspective," *UC Berkeley Tech. Report*, Aug. 2007.

[65] R. Vidal and Y. Ma, "A unified algebraic approach to 2-D and 3-D motion segmentation and estimation," *Jour. Math. Imag. Vision*, vol. 25, pp. 403–421, Oct. 2006.

[66] M. Mishali and Y. C. Eldar, "From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals," *IEEE Trans. Signal Process.*, vol. 4, pp. 375–391, Apr. 2010.

[67] F. Parvaresh, H. Vikalo, S. Misra, and B. Hassibi, "Recovering sparse signals using sparse measurement matrices in compressed DNA microarrays," *IEEE Jour. Select. Topics Signal Process.*, vol. 2, pp. 275–285, Jun. 2008.

[68] K. Rosenblum, L. Zelnik-Manor, and Y. C. Eldar, "Dictionary optimization for block-sparse representations," *to appear in IEEE Trans. Signal Process.*, 2011.

[69] M. Mishali and Y. C. Eldar, "Blind multiband signal reconstruction: Compressed sensing for analog signals," *IEEE Trans. Signal Process.*, vol. 57, pp. 993–1009, Mar. 2009.

[70] A. Webb, *Introduction to Biomedical Imaging.* IEEE Press Series in Biomed. Engg., 2003.

[71] D. L. D. Michael Lustig, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI," *IEEE Signal Process. Magazine*, pp. 72–82, Mar. 2008.

[72] R. Chartrand, "Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data," *IEEE Inter. Symp. Biomedical Imaging*, pp. 262–265, 2009.

[73] E. N. Bruce, *Biomedical Signal Processing and Signal Modeling.* Wiley Series in Telecom. and Signal Process., 2001.

[74] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Trans. Biomed. Engg.*, vol. 58, no. 9, pp. 2456–2466, 2011.

[75] S. Aviyente, "Compressed sensing framework for EEG compression," *Proc. IEEE Workshop Stat. Signal Process.*, pp. 181–184, Aug. 2007.

[76] S. Senay, L. F. Chaparro, M. Sun, and R. J. Sclabassi, "Compressive sensing and random filtering of EEG signals using slepian basis," *EUSIPCO*, Aug. 2008.

[77] "Rice university compressive sensing resources," URL: `http://dsp.rice.edu/cs`,[Online; accessed 25-January-2012].

[78] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Inter. Conf. Image Process.*, pp. III–105–III–108, 2007.

[79] L. He and L. Carin, "Exploiting structure in wavelet-based bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 57, pp. 3488–3497, Sept. 2009.

[80] M. Elad, "Why simple shrinkage is still relevant for redundant representations?," *IEEE Trans. Inf. Theory*, vol. 52, pp. 5559–5569, Dec. 2006.

[81] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, pp. 4800–4810, July 2009.

[82] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.

[83] K. Kreutz-Delgado, J. Murray, B. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comput.*, vol. 15, no. 2, pp. 349–396, 2003.

[84] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Jour. Machine Learning Res.*, vol. 11, pp. 19–60, 2010.

[85] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[86] M. Yaghoobi, L. Daudget, and M. E. Daves, "Parametric dictionary design for sparse coding," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4800–4810, 2009.

[87] I. Tosic and P. Frossard, "Dictionary learning," *IEEE Signal Process. Magazine*, pp. 27–38, 2011.

[88] J. Mairal, G. Spairo, and M. Elad, "Multiscale sparse image representation with learned dictionaries," *IEEE Inter. Conf. Image Process.*, pp. III–105–III–108, 2007.

[89] G. W. Stewart, *Introduction to Matrix Computations.* Academic Press, 1973.

Appendix A

Singular-Value and QR Decompositions

A.1 Singular-Value Decomposition

Given a matrix $\mathbf{\Phi} \in R^{M \times N}$ where $M < N$ and given that $\mathbf{\Phi}$ is of rank M, the *singular-value decomposition* (SVD) of $\mathbf{\Phi}$ is given by

$$\mathbf{\Phi} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{A.1}$$

where $\boldsymbol{U} \in R^{M \times M}$ and $\boldsymbol{V} \in R^{N \times N}$ are orthogonal matrices,

$$\boldsymbol{S} = [\mathbf{\Sigma}\ \mathbf{0}]$$

with

$$\mathbf{\Sigma} = \text{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_M\}$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M > 0,$$

In the above equations, $\mathbf{0}$ denotes the zero matrix of size $M \times (N - M)$, $\sigma_1, \sigma_2, \ldots, \sigma_M$ are the nonzero singular values of $\mathbf{\Phi}$, and the columns of $\boldsymbol{U}$ are the left singular vectors of $\mathbf{\Phi}$ corresponding to the singular values [89] [48]. The matrix $\boldsymbol{V}$ in (A.1) can be expressed as $\boldsymbol{V} = [\boldsymbol{V}_r\ \boldsymbol{V}_n]$ where $\boldsymbol{V}_n$ is composed of the last $N - M$ columns of $\boldsymbol{V}$ which span the null space of $\mathbf{\Phi}$, while $\boldsymbol{V}_r$ is composed of first M columns of $\boldsymbol{V}$, which span the orthogonal complement of the null space. In the literature, this orthogonal complement space is also known as the range or row space of $\mathbf{\Phi}$.

Consider equation $\boldsymbol{\Phi x} = \boldsymbol{y}$ with $M < N$, which is an underdetermined system of linear equations. All the solutions of $\boldsymbol{\Phi x} = \boldsymbol{y}$ can be parameterized as

$$\boldsymbol{x} = \boldsymbol{x}_s + \boldsymbol{V}_n \boldsymbol{\xi} \tag{A.2}$$

where $\boldsymbol{x}_s$ is a special solution of $\boldsymbol{\Phi x} = \boldsymbol{y}$. In many applications, the least-squares solution is used as $\boldsymbol{x}_s$ which can be computed as

$$\boldsymbol{x}_{\ell_2} = \boldsymbol{\Phi}^{\dagger} \boldsymbol{y}$$

where $\boldsymbol{\Phi}^{\dagger}$ is the Moore-Penrose pseudoinverse of $\boldsymbol{\Phi}$ given by

$$\boldsymbol{\Phi}^{\dagger} = \boldsymbol{V} \begin{bmatrix} \boldsymbol{\Sigma}^{-1} \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{U}^T$$

with $\boldsymbol{\Sigma}^{-1} = \text{diag}\{1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_M\}$.

A.2 QR Decomposition

Given a matrix $\boldsymbol{\Phi} \in R^{M \times N}$ with $M < N$, the QR decomposition of $\boldsymbol{\Phi}^T$ is given by

$$\boldsymbol{\Phi}^T = \boldsymbol{Q}\boldsymbol{R} \tag{A.3}$$

where $\boldsymbol{Q} \in R^{N \times N}$ is an orthogonal matrix and

$$\boldsymbol{R} = \begin{bmatrix} \hat{\boldsymbol{R}} \\ \boldsymbol{0} \end{bmatrix}$$

with $\hat{\boldsymbol{R}}$ an upper triangular square matrix [89] [48].

If we express matrix $\boldsymbol{Q}$ as $\boldsymbol{Q} = [\boldsymbol{Q}_1 \ \boldsymbol{Q}_2]$ where $\boldsymbol{Q}_1 \in R^{N \times M}$ and $\boldsymbol{Q}_2 \in R^{N \times (N-M)}$, it can be shown that the columns of $\boldsymbol{Q}_2$ and $\boldsymbol{Q}_1$ span the null space of $\boldsymbol{\Phi}$ and its orthogonal complement space, respectively. The QR decomposition can be used to efficiently compute the least-squares solution $\boldsymbol{x}_{\ell_2}$ of $\boldsymbol{\Phi x} = \boldsymbol{y}$ and the basis of the null space $\boldsymbol{V}_n$ of $\boldsymbol{\Phi}$ as

$$\boldsymbol{x}_{\ell_2} = \boldsymbol{Q}_1 \hat{\boldsymbol{R}}^{-T} \boldsymbol{y}, \quad \boldsymbol{V}_n = \boldsymbol{Q}_2, \tag{A.4}$$

respectively.

Appendix B

Derivation of Eqs. (3.14)-(3.16)

Suppose the vectors $\boldsymbol{\phi}$ and $\boldsymbol{\xi}$ in (3.6) are fixed and that $\boldsymbol{e}_i$ is the ith column of the identity matrix of size $M \times M$. At point $\boldsymbol{\phi}$, a line search along direction $\boldsymbol{e}_i$ is carried out by solving the one-dimensional optimization problem

$$\underset{\delta}{\text{minimize}} \quad F_{p,\epsilon}(\boldsymbol{\phi} + \delta \boldsymbol{e}_i, \boldsymbol{\xi}) \tag{B.1}$$

where

$$\begin{aligned} F_{p,\epsilon}(\boldsymbol{\phi} + \delta \boldsymbol{e}_i, \boldsymbol{\xi}) &= \frac{1}{2} \left|\left|\boldsymbol{\Sigma}(\boldsymbol{\phi} + \delta \boldsymbol{e}_i) - \tilde{\boldsymbol{y}}\right|\right|_2^2 + \lambda \left|\left|\boldsymbol{x} + \delta \boldsymbol{v}_i\right|\right|_{p,\epsilon}^p \\ &= \frac{1}{2} \left[\sigma_i(\phi_i + \delta) - \tilde{y}_i\right]^2 \\ &\quad + \lambda \sum_{j=1}^{N} \left[(x_j + \delta v_{ij})^2 + \epsilon^2\right]^{p/2}, \end{aligned} \tag{B.2}$$

$\boldsymbol{v}_i$ is the ith column of matrix $\boldsymbol{V}_r$, and x_j and v_{ij} are the jth components of vectors $\boldsymbol{x}$ and $\boldsymbol{v}_i$, respectively. By equating the derivative $dF_{p,\epsilon}(\boldsymbol{\phi} + \delta \boldsymbol{e}_i, \boldsymbol{\xi})/d\delta$ to zero, we obtain

$$\delta = -\frac{-\sigma_i \tilde{y}_i + \sigma_i^2 \phi_i + \lambda p \sum_{j=1}^{n} \gamma_j(\epsilon) x_j v_{ij}}{\sigma_i^2 + \lambda p \sum_{j=1}^{N} \gamma_j(\epsilon) v_{ij}^2} \tag{B.3}$$

where

$$\gamma_j(\epsilon) = \left[(x_j + \delta v_{ij})^2 + \epsilon^2\right]^{p/2-1} \tag{B.4}$$

Note that (B.3) can be solved for δ iteratively. In practice, a descent step can be determined in one iteration by setting $\delta = 0$ in the right-hand side of (B.3) to compute δ. With $\delta = 0$, (B.4) is simplified to (3.16) and (3.14a) follows from (B.3).

For the descent directions along each component of vector $\boldsymbol{\xi}$ in (3.10), the fidelity term becomes a constant. As a result, $-\sigma_i \tilde{y}_i$ and $\sigma_i^2 \phi_i$ in the numerator and σ_i^2 in the denominator of (B.3) can be removed. Consequently, (3.14b) can be obtained using (B.3).