Compressive Sensing Using $\ell_p$ 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

© Jeevan Kumar Pant, 2012
University of Victoria

All rights reserved. This dissertation may not be reproduced in whole or in part, by photocopying or other means, without the permission of the author.
Compressive Sensing Using $\ell_p$ 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)
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 $\ell_0$ and $\ell_p$ 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 $\ell_p$ 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 $\ell_0$ (SL0), and $\ell_1$-minimization based algorithms. They also require a reduced amount of computations relative to the IR and $\ell_1$-minimization based algorithms. The $\ell_p$-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 $\ell_p$-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 $\ell_1$-minimization based algorithms and require a reduced amount of computation relative to the IR and $\ell_1$-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 $\ell_1$-minimization based algorithms.
Contents

Supervisory Committee ii
Abstract iii
Table of Contents v
List of Tables ix
List of Figures x
List of Abbreviations xiv
Acknowledgements xvi
Dedication xvii

1 Introduction 1
  1.1 What is Compressive Sensing? . . . . . . . . . . . . . . . . . . . . . 2
    1.1.1 A Strategy for Sensing Data . . . . . . . . . . . . . . . . . . . 2
    1.1.2 Coherence Between $P$ and $D$ . . . . . . . . . . . . . . . . . 4
  1.2 State-of-the-Art . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
    1.2.1 Fundamentals of CS . . . . . . . . . . . . . . . . . . . . . . . 5
    1.2.2 Algorithms for CS . . . . . . . . . . . . . . . . . . . . . . . 9
  1.3 Original Contributions . . . . . . . . . . . . . . . . . . . . . . . . . 16

2 Reconstruction of Sparse Signals from Noiseless Measurements by
   Using Nonconvex Optimization 19
  2.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
  2.2 The NRAL0 Algorithm . . . . . . . . . . . . . . . . . . . . . . . . . 20
    2.2.1 Working in the null space of $\Phi$ . . . . . . . . . . . . . . . 20
3 Reconstruction of Sparse Signals from Noisy Measurements by Using Least-Squares Optimization

3.1 Introduction ........................................................................ 41
3.2 The $\ell_p$-RLS Algorithm .................................................. 42
  3.2.1 Problem formulation ..................................................... 43
  3.2.2 Computation of descent direction ................................. 44
  3.2.3 Line search ............................................................... 45
  3.2.4 Optimization ............................................................ 46
  3.2.5 Algorithm ............................................................... 46
  3.2.6 Simulation results ...................................................... 47
3.3 The $\ell_p$-RLS-CG Algorithm ............................................. 51
  3.3.1 Gradient and Hessian of function $F_{p,\epsilon}(x)$ ............... 51
  3.3.2 Optimization ............................................................ 51
  3.3.3 Use of conjugate-gradient algorithm ............................ 52
  3.3.4 Algorithm ............................................................... 53
  3.3.5 $\ell_p$-RLS-CG algorithm for noiseless measurements ...... 55
  3.3.6 Simulation results ...................................................... 57
3.4 Optimization of parameter $\lambda$ ........................................ 62
  3.4.1 Algorithm ............................................................... 62
  3.4.2 Experimental results ................................................ 63
3.5 The $\text{TV}_p$-RLS-CG Algorithm ........................................ 65
3.5.1 The total variation norm and its generalization 65
3.5.2 Problem formulation 68
3.5.3 Optimization 70
3.5.4 Line search 70
3.5.5 Algorithm 72
3.5.6 Simulation results 72
3.6 Conclusions 74

4 Reconstruction of Block-Sparse Signals by Using $\ell_{2/p}$-Regularized Least-Squares Optimization 77
4.1 Introduction 77
4.2 The $\ell_{2/p}$-RLS Algorithm 78
4.2.1 Problem formulation 78
4.2.2 Optimization 80
4.2.3 Use of Fletcher-Reeves’ algorithm 80
4.2.4 Line search 81
4.2.5 Algorithm 82
4.2.6 Simulation Results 82
4.3 Reweighting the $\ell_{2/p}$-RLS Algorithm 85
4.3.1 Problem formulation 86
4.3.2 Optimization 87
4.3.3 Algorithm 88
4.3.4 Simulation results 88
4.4 The $\ell_{2/p}$-RLS Algorithm with Prior Information 93
4.4.1 Simulation results 94
4.5 Conclusions 95

5 Conclusions and Future Directions 97
5.1 Conclusions 97
5.2 Future directions 99

Bibliography 101

Appendices 109

A Singular-Value and QR Decompositions 109
A.1 Singular-Value Decomposition 109
A.2 QR Decomposition ........................................ 110

B Derivation of Eqs. (3.14)-(3.16) ......................... 112
## List of Tables

Table 2.1 The Null-Space Reweighted Approximate $\ell_0$-Pseudonorm Algorithm ........................................... 23
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. 26
Table 2.3 UALP Algorithm ................................................................. 32
Table 2.4 Line Search Based on Banach’s Fixed-Point Theorem ........ 33
Table 2.5 CPU Time Required by the UALP Algorithm with the Proposed Line Search and Fletcher’s Inexact Line Search, in Seconds 35
Table 2.6 The UALP-CG Algorithm ................................................. 38
Table 3.1 $\ell_p$-RLS Algorithm ....................................................... 47
Table 3.2 $\ell_p$-RLS-CG Algorithm ................................................ 56
Table 3.3 $\ell_p$-RLS-CG Algorithm for Noiseless Measurements ........ 57
Table 3.4 $\ell_p$-RLS-CG Algorithm for Noiseless Measurements by Optimizing $\lambda$ .......................................................... 63
Table 3.5 $TV_p$-RLS-CG Algorithm .................................................. 73
Table 4.1 $\ell_{2/p}$-RLS Algorithm ..................................................... 83
Table 4.2 $\ell_{2/p}$-RLS-WT Algorithm ............................................. 89
Table 4.3 Relative error due to the $\ell_{2/p}$-RLS-WT and $\ell_{2/p}$-RLS algorithms 91
List of Figures

Figure 1.1 Plots of functions $|x_i|^0$, $|x_i|^{0.08}$, $|x_i|^{0.3}$, and $|x_i|$. ................................. 11
Figure 1.2 Solution space $\Phi x = y$ and $\ell_2$ ball; the $\ell_2$ solution is not sparse. .......................... 11
Figure 1.3 Solution space $\Phi x = y$ and $\ell_1$ ball; the $\ell_1$ solution is sparser than the $\ell_2$ solution. .......................................................... 12
(a) $\|x\|_1 = 0.5$. ................................................................. 12
(b) $\|x\|_1 = 1$. ................................................................. 12
Figure 1.4 Solution space $\Phi x = y$ and $\ell_p$ ball with $p = 0.5$; the minimum $\ell_p$-pseudonorm solution is sparser than the minimum $\ell_1$-norm solution. .................................................. 14
(a) $\|x\|_p = 0.5$. ................................................................. 14
(b) $\|x\|_p = 1$. ................................................................. 14
(c) $\|x\|_p = 1.5$ ................................................................. 14

Figure 2.1 Number of perfect reconstructions by the NRAL0, IR, and SL0 algorithms over 100 runs with $N = 256$ and $M = 100$. ............................. 25
Figure 2.2 Average CPU time required by the NRAL0, IR, and SL0 algorithms over 100 runs with $M = N/2$ and $K = \text{round}(M/2.5)$. .................. 25
Figure 2.3 Function $F_p(\xi)$ for $N = 256$, $M = 100$, $K = 40$, and $\sigma = 0.0218$. ............................. 26
Figure 2.4 Value of function $F_{p,\epsilon}(\xi)$ in the solution space. ......................... 30
Figure 2.5 Number of perfect reconstructions for UALP, NRAL0, SL0, IR, and BP algorithms over 100 runs with $N = 256$, $M = 100$. .......... 34
Figure 2.6 Average CPU time required for UALP, NRAL0, SL0, IR, and BP algorithms over 100 runs with $M = N/2$, $K = M/2.5$. .............. 34
Figure 2.7 Function $F_{p,\epsilon}(\xi)$ for $N = 256$, $M = 100$, $K = 40$, and $\epsilon = 0.0118$. ................. 36
Figure 2.8 Number of perfect reconstructions and average CPU time required for the UALP-CG, SL0, IR, and BP algorithms. ..................... 40
(a) Number of perfect reconstructions with $N = 512$ and $M = 200$ over 400 runs. .......................... 40
(b) Average CPU time with \( M = N/2 \) and \( K = \text{round}(M/2.5) \) over 100 runs. ................................................. 40

Figure 3.1 Percentage of recovered instances for the \( \ell_p \)-RLS, UALP, IR, SL0, and BPDN algorithms over 100 runs with \( N = 1024, M = 200 \). 48

Figure 3.2 Average CPU time required by the \( \ell_p \)-RLS, UALP, IR, SL0, and BPDN algorithms over 100 runs with \( M = N/2, K = M/2.5 \). 49

Figure 3.3 Function \( F_{p,\epsilon}(\phi, \xi) \) for \( N = 1024, M = 200, K = 60, \) and \( \epsilon = 0.01 \). ......................................................... 50

Figure 3.4 Contour of an \( \ell_2, \ell_p \) objective function for \( \Phi = [0.1 \ 0.06], y = 0.1, \lambda = 0.0007, \) and \( p = 0.1 \). ......................................................... 53

Figure 3.5 Contours of function \( F_{p,\epsilon}(x) \) for \( \epsilon = 1, 0.6, 0.2, 0.08, 0.04, \) and \( 0.008 \) with \( p = 0.1, \lambda = 0.0007, \Phi = [0.1 \ 0.06], \) and \( y = 0.1 \) .... 54

(a) \( \epsilon = 1 \) ......................................................... 54
(b) \( \epsilon = 0.6 \) ......................................................... 54
(c) \( \epsilon = 0.2 \) ......................................................... 54
(d) \( \epsilon = 0.08 \) ......................................................... 54
(e) \( \epsilon = 0.04 \) ......................................................... 54
(f) \( \epsilon = 0.008 \) ......................................................... 54

Figure 3.6 Reconstruction performance for the \( \ell_p \)-RLS-CG, SL0, IR, and BPDN algorithms for noisy measurements over 350 runs with \( N = 512, M = 200, \) and measurement noise \( \mathcal{N}(0, 0.01^2) \). .... 58

(a) Percentage of recovered instances .................................. 58
(b) Average signal-to-noise ratio ........................................ 58

Figure 3.7 Reconstruction performance for the \( \ell_p \)-RLS-CG, SL0, IR, and BPDN algorithms for noisy measurements over 250 runs with \( N = 512, M = 200, \) and measurement noise drawn from \( \mathcal{N}(0, 0.03) \). .... 59

Figure 3.8 Average CPU required by the \( \ell_p \)-RLS-CG, SL0, IR, and BPDN algorithms for noisy measurements over 100 runs with \( M = N/2, K = M/2.5 \). ................................................. 60

Figure 3.9 Number of perfect reconstructions and average CPU time required for the \( \ell_p \)-RLS-CG, SL0, IR, and BP algorithms. .... 61

(a) Number of perfect reconstructions over 400 runs ............ 61
(b) Average CPU time over 100 runs .................................. 61
Figure 3.10 Percentage of recovered instances for the $\ell_p$-RLS-CG algorithm with and without optimizing $\lambda$ and the SL0, IR, and BP algorithms over 100 runs with $N = 512$ and $M = 200$.

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

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

(a) Peak-signal-to-noise ratio
(b) CPU time

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.

(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 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 $N = 512, M = 100, and d = 8$.

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 $M = N/2, K = M/2.5d$.

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 $M = \text{round}(N/2), K = \text{round}(M/2.5d)$.

Figure 4.5 Relative error for the $\ell_2/p$-RLS-WT and $\ell_2/p$-RLS algorithms for $N = 1024, M = N/2, K = \text{round}(M/2.5d)$.

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 $d = 8$. .......................................................... 92

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 $d = 8$. .......................................................... 95
List of Abbreviations

BFGS  Broyden-Fletcher-Goldfarb-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
ℓ2/1-SOCP ℓ2/1 second-order cone programming
ℓ2/p-RLS ℓ2/p-regularized least-squares
ℓ2/p-RLS-WT ℓ2/p-regularized least squares with weighting
ℓp-RLS ℓp-regularized least-squares
ℓp-RLS-CG ℓp-regularized least-squares conjugate gradient
NRAL0 nullspace reweighted approximate ℓ0
PSNR  peak-signal-to-noise ratio
RIP   restricted isometry property
ROU   range of uncertainty
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL0</td>
<td>smoothed $\ell_0$</td>
</tr>
<tr>
<td>SNR</td>
<td>signal-to-noise ratio</td>
</tr>
<tr>
<td>SOCP</td>
<td>second-order cone programming</td>
</tr>
<tr>
<td>SVD</td>
<td>singular-value decomposition</td>
</tr>
<tr>
<td>$TV_p$-RLS-CG</td>
<td>$TV_p$-regularized least-squares conjugate gradient</td>
</tr>
<tr>
<td>TV</td>
<td>total variation</td>
</tr>
<tr>
<td>UALP</td>
<td>unconstrained approximate $\ell_p$</td>
</tr>
<tr>
<td>UALP-CG</td>
<td>unconstrained approximate $\ell_p$ conjugate gradient</td>
</tr>
</tbody>
</table>
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
Although the concept of signal sparsity and $\ell_1$-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
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 $P$ and a sparsifying dictionary $D$ by saying that the sensing vectors, i.e., some of the rows in $P$, must be spread out in the $D$ 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 $u$ which itself may or may not be sparse in the canonical basis but is sparse or approximately sparse in an appropriate dictionary $D \in \mathbb{R}^{N \times L}$, that is,

$$u = Dx$$

(1.1)
where $x$ 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 $u$ of length $N$ is carried out by measuring $M$ projections of $u$ into sensing vectors also known as testing vectors $\{p_i, i = 1, 2, \ldots, M\}$ in the form of $y_i = p_i^T u$ 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

$$y = \hat{P} \cdot u$$  \hspace{1cm} (1.2a)$$

where

$$\hat{P} = \begin{bmatrix}
p_1^T \\
p_2^T \\
\vdots \\
p_M^T
\end{bmatrix}, \quad u = \begin{bmatrix}
u_1 \\
u_2 \\
\vdots \\
u_N
\end{bmatrix}, \quad y = \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_M
\end{bmatrix}$$  \hspace{1cm} (1.2b)$$

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

(i) What type of matrix $\hat{P}$ 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 $u$?

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

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

$$\mu(P, D) = \sqrt{N} \cdot \max_{1 \leq k, j \leq N} |p_k^T d_j|$$

(1.3)

where $p_k^T$ and $d_j$ are the $k$th row and $j$th column of $P$ and $D$, respectively [28], [29]. Obviously $\mu$ measures the largest correlation between a basis vector in $P$ and a basis vector in $D$. It can be shown that the coherence measure $\mu$ in (1.3) is always in the range $[1, \sqrt{N}]$.

We are interested in basis pairs $\{P, D\}$ with small coherence and in such a case, bases $\{P, D\}$ 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 $P$ and $D$ should one use in order to achieve a small $\mu(P, D)$?

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

- If $I_N$ is the identity matrix of size $N \times N$ and $C_N$ is a discrete-cosine transformation matrix, then $\mu(I_N, C_N^T) \leq \sqrt{2}$.

- If $F_N$ is a Fourier transformation matrix, then $\mu(I_N, F_N^H) = 1$, where $F_N^H$ is the Hermitian of $F_N$. 
• If \( D \) is a fixed orthonormal basis and \( P \) is an orthonormal matrix with independent identically distributed (i.i.d.) entries, e.g., Gaussian or \( \pm 1 \) binary entries, then \( \mu(P, D) \) 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 \( u \in \mathbb{R}^N \) and suppose that \( u \) is \( K \)-sparse in an orthonormal basis \( D \). Now select \( M \) measurements in the \( P \) domain uniformly at random via (1.2) such that the \( M \) sensing vectors \( \{p_i, i = 1, 2, \ldots, M\} \) are \( M \) rows uniformly randomly selected from matrix \( P \in \mathbb{R}^{N \times N} \). If

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

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

\[
\begin{align*}
\text{minimize} \quad & ||x||_1 \tag{1.5a} \\
\text{subject to:} \quad & \Phi x = y \tag{1.5b}
\end{align*}
\]

Matrix \( \Phi = \hat{P}D \), \( \hat{P} \) is given by (1.2b) and \( ||x||_1 \) is the \( \ell_1 \) norm of \( x \) defined as

\[
||x||_1 = \sum_{i=1}^{N} |x_i|.
\]

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 \( D \)), the number of measurements \( M \) that is sufficient to recover \( x \), and the incoherence between \( P \) and \( D \). 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 $P$ and $D$ are *incoherent*, i.e., $\mu(P, D) \approx 1$. Indeed, if $\mu(P, D) = 1$, (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 $u$, i.e.,

$$M \geq 4K$$

(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 $\mu(P, D) = \sqrt{N}$, then (1.4) becomes

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

(1.7)

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

(ii) The constraint $\Phi x = y$ in (1.5) with $\hat{P} \in \mathbb{R}^{M \times N}$ where $M < N$ and $D \in \mathbb{R}^{N \times N}$ 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

$$y = \Phi x + w$$

(1.8a)

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

$$\Phi = SPD = \hat{P}D$$

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

$$\begin{align*}
\text{minimize} & \quad ||x||_1 \\
\text{subject to:} & \quad ||\Phi x - y||_2^2 \leq \nu
\end{align*}$$

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

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 $\delta_K$ of a matrix $\Phi$ as the smallest number such that

$$\begin{align*}
(1 - \delta_K)||x||_2^2 \leq ||\Phi x||_2^2 \leq (1 + \delta_K)||x||_2^2
\end{align*}$$

for all $K$-sparse vectors $x$. We shall say, matrix $\Phi$ obeys the RIP of order $K$ if $\delta_K$ 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 $\Phi$ 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 $x$ and any value of $\delta_K$. Now, for a $K$-sparse $x$, $\Phi x$ is equal to $\Phi_K x_K$ where $\Phi_K$ is a sub-matrix of $\Phi$ consisting of those $K$ columns of $\Phi$ corresponding to the $K$ nonzero entries of $x$ and $x_K$ is a vector $x$ with its zero entries dropped (note that $||x||_2 = ||x_K||_2$). Therefore, the condition in (1.10) amounts to saying that all sub-matrices of $K$ columns taken from $\Phi$ are nearly orthogonal.

(ii) Obviously, the condition in (1.10) prevents any $K$-sparse $x$ from being in the null space of $\Phi$. This is important in the context of CS because if there exists a $K$-sparse $x$ that belongs to the null space of $\Phi$ then $y = \Phi x + w = w$ which means that the measurement obtained is totally useless as $y$ contains no information whatsoever about sparse signal $x$. 
(iii) In the case where $\delta_{2K}$ is sufficiently smaller than one, then for all $K$-sparse vectors $x_1$ and $x_2$, vector $x_1 - x_2$ is at most $2K$-sparse and hence we have

$$(1 - \delta_{2K}) ||x_1 - x_2||^2_2 \leq ||\Phi x_1 - \Phi x_2||^2_2 \leq (1 + \delta_{2K}) ||x_1 - x_2||^2_2 \quad (1.11)$$

An immediate consequence of the condition in (1.11) for the noise-free case is that the measurement $y = \Phi x$ of a $K$-sparse signal $x$ completely characterizes the signal in the sense that if measurements $\Phi x_1$ and $\Phi x_2$ are identical, then signals $x_1$ and $x_2$ must be identical.

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

$$||x - x^*||_2 \leq C_0 \cdot ||x - x_K||_1 / \sqrt{K}$$

(1.12a)

and

$$||x - x^*||_1 \leq C_0 \cdot ||x - x_K||_1$$

(1.12b)

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

**Remark** If $x$ is $K$-sparse, then $x_K = x$, and hence the conditions in (1.12) imply that $x^* = x$, 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 $x$ is not sparse but approximately $K$-sparse then $||x - x_K||_1$ is small and Theorem 1.2.2 implies that the solution $x^*$ of the problem in (1.9) is expected to be a reconstruction of $x$ 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 $x^*$ of the problem in (1.9) satisfies the inequality

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

(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 $x$ is $K$-sparse, then $x_K = x$, and the condition in (1.13) implies that $||x - x^*||_2 \leq C_1 \varepsilon$. 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 \( x \) is not sparse but approximately \( K \)-sparse, Theorem 1.2.3 implies that the accuracy of the solution \( x^* \) of the problem in (1.9) is determined by both the closeness of signal \( x \) 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 \( \ell_p \)-pseudonorm minimization with \( p < 1 \).

1.2.2.1 Signal reconstruction based on \( \ell_p \)-pseudonorm minimization

We begin by defining the \( \ell_0 \) pseudonorm as \( ||x||_0 = \sum_{i=1}^{N} |x_i|^0 \) which is equal to the number of nonzero components in \( x \). Hence, if we assume that signals \( x_1 \) and \( x_2 \) are of the same length, then \( x_1 \) is more sparse than \( x_2 \) if \( ||x_1||_0 < ||x_2||_0 \). It follows that the problem of reconstructing a sparse signal \( x \) from linear measurement \( y = \Phi x \) can be formulated as

\[
\begin{align*}
\text{minimize} & \quad ||x||_0 \\
\text{subject to:} & \quad \Phi x = y
\end{align*}
\]  

(1.14a)

(1.14b)

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

\[
\begin{align*}
\text{minimize} & \quad ||x||_0 \\
\text{subject to:} & \quad ||\Phi x - y||_2^2 \leq \nu
\end{align*}
\]  

(1.15a)

(1.15b)

where bound \( \nu \) is related to the variance of noise \( w \). 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 \( ||x||_0 \) by the convex function \( ||x||_1 \). 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
\( \ell_1 \)-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 \( \ell_p \)-pseudonorm based minimization problem with \( p < 1 \). Specifically, the \( \ell_p \)-pseudonorm minimization problem for the reconstruction of sparse signals whose measurements are free of noise corruption is given by

\[
\begin{align*}
\text{minimize} & \quad \|x\|_p^p \\
\text{subject to:} & \quad \Phi x = y
\end{align*}
\tag{1.16a}
\]

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

\[
\begin{align*}
\text{minimize} & \quad \|x\|_p^p \\
\text{subject to:} & \quad \|\Phi x - y\|_2^2 \leq \nu
\end{align*}
\tag{1.17a}
\]

Below, we illustrate why \( \ell_p \)-pseudonorm minimization is more effective in determining a sparse signal relative to \( \ell_1 \)-norm and \( \ell_2 \)-norm minimizations.

**Difference between** \( \|x\|_0 \), \( \|x\|_p \), and \( \|x\|_1 \)

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 \( \ell_0 \) pseudonorm \( \|x\|_0 \), the \( \ell_p \) pseudonorm \( \|x\|_p \), and the \( \ell_1 \) norm \( \|x\|_1 \) 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 \( \ell_p \) pseudonorm with \( 0 \leq p < 1 \) approximates the \( \ell_0 \) pseudonorm more accurately than the \( \ell_1 \) norm does.

**Why \( \ell_2 \) minimization fails?**

Consider a signal \( x \) of length 3, i.e., \( x = [x_1 \ x_2 \ x_3]^T \), and measurement matrix \( \Phi \) of size \( 1 \times 3 \). Equation \( \Phi x = y \), 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 \( \Phi x = y \), 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 $\Phi x = y$ and $\ell_2$ ball; the $\ell_2$ solution is not sparse.
line is also referred to as the solution space. Fig. 1.2 also shows an $\ell_2$ ball, which is the surface of a sphere given by $||x||_2 = c$ where $c$ is a positive scalar constant. In other words, all the points in the $\ell_2$ ball have a constant $\ell_2$ norm. As we increase $r$ from a sufficiently small value, the size of the $\ell_2$ ball increases and eventually touches the solution space. The point of touch where the solution space is tangent to the $\ell_2$ ball is the minimum $\ell_2$-norm solution and is also called the $\ell_2$ solution. As can be noticed from the figure, all the three components of the $\ell_2$ solution are nonzero, hence, the $\ell_2$ solution is not sparse.

**Why $\ell_1$ minimization works?**

An $\ell_1$ ball is defined by the set $\{x : ||x||_1 = c\}$ where $c$ is a positive scalar constant. Fig. 1.3a shows the $\ell_1$ ball with $c = 0.5$ and the solution space $\Phi x = y$. As we increase $r$, the size of the $\ell_1$ ball increases. When $c = 1$, the $\ell_1$ ball touches the solution space; the point of touch, which is the $\ell_1$ solution, is marked by a black dot in Fig. 1.3b. As can be noticed, the components of the $\ell_1$ solution are given by $x_1 \neq 0$, $x_2 \neq 0$, and $x_3 = 0$. Hence, the $\ell_1$ solution is clearly sparser than the $\ell_2$ solution.

Figure 1.3: Solution space $\Phi x = y$ and $\ell_1$ ball; the $\ell_1$ solution is sparser than the $\ell_2$ solution.
Why \( \ell_p \) minimization works better?

An \( \ell_p \) ball is defined as a set of points \( \{x : \|x\|_p = c\} \) where \( c \) is a positive scalar constant. Fig. 1.4a shows an \( \ell_p \) ball for \( p = 0.5 \) and \( c = 0.5 \), and the solution space. Recall from Fig. 1.3b that, for \( c = 1 \) the \( \ell_1 \) ball touches the solution space yielding a 2-sparse solution. However, as shown in Fig. 1.4b, the \( \ell_p \) 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 \( \ell_p \) ball touches the solution space for \( c = 1.5 \). One can see that the point of touch or the \( \ell_p \) solution has only one nonzero component, i.e., \( x_3 \neq 0 \). Hence, the \( \ell_p \) solution is sparser than the \( \ell_1 \) 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

\[
\begin{align*}
\minimize_x \quad & \frac{1}{2} \| \Phi x - y \|_2^2 + \lambda \| x \|_1 \\
\end{align*}
\]

where \( \lambda > 0 \) is a regularization parameter. For the problem in (1.9), the \( \lambda \) in (1.18) is related to bound \( v \) in (1.9b) and hence it depends on the variance of noise \( u \). For the problem in (1.5), the \( \lambda \) 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 \( \lambda \| x \|_1 \), 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 \( \ell_p \)-pseudonorm regularized least-squares optimization problem can be posed as

\[
\begin{align*}
\minimize_x \quad & \frac{1}{2} \| \Phi x - y \|_2^2 + \lambda \| x \|_p^p \\
\end{align*}
\]

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 \( \| x \|_p^p \).
Figure 1.4: Solution space $\Phi x = y$ and $\ell_p$ ball with $p = 0.5$; the minimum $\ell_p$-pseudonorm solution is sparser than the minimum $\ell_1$-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 \( \mathbf{x} \) of length \( N \) which is divisible by a positive integer \( d \). We divide signal \( \mathbf{x} \) into \( N/d \) blocks \( \tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{N/d} \) and denote \( \mathbf{x} \) as

\[
\mathbf{x} = [\tilde{\mathbf{x}}_1^T \tilde{\mathbf{x}}_2^T \cdots \tilde{\mathbf{x}}_{N/d}^T]^T
\]

where

\[
\tilde{\mathbf{x}}_i = [x_{(i-1)d+1} \ x_{(i-1)d+2} \cdots x_{(i-1)d+d}]^T
\]

for \( i = 1, 2, \ldots, N/d \) and \( x_i \) is the \( i \)th component of \( \mathbf{x} \).

A signal \( \mathbf{x} \) of the form in (1.20) is said to be \( K \)-block sparse if \( \mathbf{x} \) 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

\[
\begin{align*}
\text{minimize} & \quad ||\mathbf{x}||_{2/1} \\
\text{subject to:} & \quad \Phi \mathbf{x} = \mathbf{y}
\end{align*}
\]

where \( ||\mathbf{x}||_{2/1} \) is the \( \ell_{2/1} \) norm of \( \mathbf{x} \) defined as

\[
||\mathbf{x}||_{2/1} = \sum_{i=1}^{N/d} ||\tilde{\mathbf{x}}_i||_2
\]

In (1.22), \( ||\tilde{\mathbf{x}}_i||_2 \) is the \( \ell_2 \) norm of the \( i \)th block \( \tilde{\mathbf{x}}_i \).

The problem in (1.21) can be recast as

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{N/d} t_i \\
\text{subject to:} & \quad \Phi \mathbf{x} = \mathbf{y} \\
& \quad ||\tilde{\mathbf{x}}_i||_2 \leq t_i \quad 1 \leq i \leq N/d \\
& \quad 0 \leq t_i \quad 1 \leq i \leq N/d
\end{align*}
\]

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 $r_0 = y$ and index set $\mathcal{I}$ to the empty set and the $l$th stage of the algorithm entails the following steps:

- Choose the block index $k_l$ as $k_l = \text{argmax}\{\Phi_i^T r_{l-1}\}$ where $\Phi_i$ is a matrix of size $M \times d$ whose columns are the columns of $\Phi$ corresponding to the indices of the $i$th block $\tilde{x}_i$.
- Set $\mathcal{I} = \mathcal{I} \cup \{(k_l-1)d+1, (k_l-1)d+2, \ldots, (k_l-1)d+d\}$.
- Find solution $x_l$ as a vector $x$ that minimizes
  $$||y - \Phi_{\mathcal{I}} x_{\mathcal{I}}||_2$$
  where $\Phi_{\mathcal{I}}$ is a matrix whose columns are the columns of $\Phi$ corresponding to the indices $\mathcal{I}$ and $x_{\mathcal{I}}$ is a vector obtained by retaining the components of $x$ with indices $\mathcal{I}$.
- Update the residual as
  $$r_l = y - \Phi_{\mathcal{I}} x_{\mathcal{I}}$$

When the algorithm stops, the value of $x$ 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 $\ell_0$ (NRAL0) algorithm, is based on the minimization of an approximate $\ell_0$ pseudonorm. The second algorithm, called unconstrained approximate $\ell_p$ (UALP) algorithm, and the third algorithm, called unconstrained approximate $\ell_p$ conjugate-gradient (UALP-CG) algorithm, are based on the minimization of an approximate $\ell_p$
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 $\ell_0$ 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 $\ell_p$-regularized least-squares ($\ell_p$-RLS) algorithm, minimizes an $\ell_p$-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 $\ell_p$-regularized least-squares conjugate-gradient ($\ell_p$-RLS-CG) algorithm, minimizes an $\ell_p$-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 $\ell_p$-RLS-CG algorithm for reconstructing signals from noise-free measurements and a technique for using the $\ell_p$-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 $\ell_{2/\rho}$-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 $\ell_1$-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 $\ell_p$-pseudonorm based minimization problem with $p < 1$. A computationally efficient algorithm based on the minimization of an approximate smoothed $\ell_0$ pseudonorm, known as smoothed $\ell_0$ (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 $\ell_0$ pseudonorm, the second is based on minimizing an approximate $\ell_p$ pseudonorm by using a quasi-Newton technique, and the third is based on minimizing an approximate $\ell_p$ pseudonorm using a conjugate-gradient technique.
2.2 Minimization of Reweighted Approximate $\ell_0$ Pseudonorm

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

$$\begin{align*}
\text{minimize} & \quad ||x||_{0,\sigma} \\
\text{subject to:} & \quad \Phi x = y
\end{align*}$$

where $||x||_{0,\sigma}$ is an approximate $\ell_0$ pseudonorm of signal $x$ given by

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

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

where $\sigma > 0$ is an approximation parameter. The approximate $\ell_0$ pseudonorm $||x||_{0,\sigma}$ consists of a summation of functions $\{f_{\sigma}(x_i)\}$, where $f_{\sigma}(x_i)$ is an approximation of $|x_i|^0$ with parameter $\sigma$. It follows from (2.2) and (2.3) that as $\sigma \to 0$,

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

$$= |x_i|^0$$

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

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

Hence, the smaller the $\sigma$, the more accurate the approximation.

2.2.1 Working in the null space of $\Phi$

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

$$x = x_s + V_n \xi$$

(2.5)
where $x_s$ is a solution of $\Phi x = y$, $V_n$ is a $N \times (N - M)$ matrix whose columns constitute an orthonormal basis of the null space of $\Phi$, and $\xi$ is a parameter vector of dimension $N - M$ [44]. Vector $x_s$ and matrix $V_n$ in (2.5) can be evaluated by using the singular-value decomposition (SVD) or, more efficiently, by using the QR decomposition of matrix $\Phi$ [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

$$\min_{\xi} \quad F_\sigma(\xi) = \sum_{i=1}^{N} \left(1 - e^{-|x_{s(i)} + v_i^T \xi|^{2}/2\sigma^2}\right)$$

(2.6)

where $v_i^T$ denotes the $i$th row of matrix $V_n$. It follows from (2.4) and (2.5) that the function $F_\sigma(\xi)$ measures the $\ell_0$ pseudonorm of $x_s + V_n \xi$. Therefore, when the solution of the problem in (2.6), say, $\xi^*$, is used in (2.5), it would yield the sparsest signal $x$.

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

$$\hat{g} = \frac{V_n^T g}{\sigma^2}$$

(2.7a)

where $g = [g_1 \ g_2 \ \cdots \ g_N]^T$ with

$$g_i = [x_{s(i)} + v_i^T \xi] e^{-|x_{s(i)} + v_i^T \xi|^{2}/2\sigma^2}$$

(2.7b)

Evidently, working in the null space of $\Phi$ 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 $\ell_0$ 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

$$\min_{\xi} \quad F_\sigma(\xi) = \sum_{i=1}^{N} w_i \left(1 - e^{-|x_{s(i)} + v_i^T \xi|^{2}/2\sigma^2}\right)$$

(2.8)
where $w_i$ are positive scalars that form a weight vector $\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_N]$. Starting with an initial $\mathbf{w}^{(0)} = \mathbf{e}_N$, where $\mathbf{e}_N$ is the all-one vector of dimension $N$, in the $(k+1)$th iteration the weight vector is updated to $\mathbf{w}^{(k+1)}$ with its $i$th component given by

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

where $x_i^{(k)}$ denotes the $i$th component of vector $\mathbf{x}^{(k)}$ obtained in the $k$th iteration as $\mathbf{x}^{(k)} = \mathbf{x}_s + \mathbf{V}_n \xi^{(k)}$, and $\tau$ 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 $i$th component of $\mathbf{g}$ is given by

$$g_i = w_i \left[ x_s(i) + \mathbf{v}_i^T \xi \right] e^{-|x_s(i) + \mathbf{v}_i^T \xi|^2/2\sigma^2}$$  \hspace{1cm} (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 $F_\sigma(\xi)$ in (2.8) is convex is closely related to the value of parameter $\sigma$: the greater the value of $\sigma$, the larger the convex region. On the other hand, for $F_\sigma(\xi)$ to well approximate the $\ell_0$-pseudonorm of $\mathbf{x}$, $\sigma$ 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 $F_\sigma(\xi)$ with a reduced value of $\sigma$, say, $r \cdot \sigma$ with $r < 1$. This procedure is repeated until function $F_\sigma(\xi)$ with $\sigma \leq \sigma_J$ is minimized where $\sigma_J$ is a prescribed sufficiently small value of $\sigma$. For a fixed value of $\sigma$, 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 $x$ using a measurement $y = \Phi x$ 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 $\ell_0$-pseudonorm (NRAL0) algorithm.

Table 2.1: The Null-Space Reweighted Approximate $\ell_0$-Pseudonorm Algorithm

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Input $\Phi$, $x_{\ell_2}$, $\sigma_T$, $r$, and $\tau$, and $\epsilon$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Set $\xi^{(0)} = 0$, $w^{(0)} = e_N$, $\sigma = \max</td>
</tr>
<tr>
<td>Step 3</td>
<td>Perform the QR decomposition $\Phi^T = QR$ and construct $V_n$ using the last $N - M$ columns of $Q$.</td>
</tr>
<tr>
<td>Step 4</td>
<td>With $w = w^{(k)}$ and $\xi^{(0)}$ as an initial point, apply the BFGS algorithm to solve the problem in (2.8), where reweighting with parameter $\epsilon$ is applied using (2.9) in each iteration. Denote the solution as $\xi^{(k)}$.</td>
</tr>
<tr>
<td>Step 5</td>
<td>Compute $x^{(k)} = x_{\ell_2} + V_n \xi^{(k)}$ and update weight vector to $w^{(k+1)}$ using (2.9).</td>
</tr>
<tr>
<td>Step 6</td>
<td>If $\sigma \leq \sigma_T$, stop and output $x^{(k)}$ as the solution; otherwise, set $\xi^{(0)} = \xi^{(k)}$, $\sigma = r \cdot \sigma$, $k = k + 1$, and repeat from Step 4.</td>
</tr>
</tbody>
</table>

In the algorithm, vector $x_s$ is chosen to be the least-squares solution $x_{\ell_2}$ of $\Phi x = y$, namely, $x_s = x_{\ell_2} = \Phi^T (\Phi \Phi^T)^{-1} y$. Concerning the initial value of parameter $\sigma$, we note that function $F_\sigma(\xi)$ remains convex in the region where the largest magnitude of the components of $x = x_{\ell_2} + V_n \xi$ is less than $\sigma$. Based on this, a reasonable initial value of $\sigma$ can be chosen as $\sigma_0 = \max |x_{\ell_2}| + \tau$ where $\tau$ is a small positive scalar. As the algorithm starts at the origin $\xi^{(0)} = 0$, the above choice of $\sigma_0$ 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 $\mathbf{x}$ was constructed as follows: (1) set $\mathbf{x}$ to a zero vector of length $N$; (2) generate a vector $\mathbf{u}$ 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 $\ell_2$-norm of each column is unity. The measurement was obtained as $\mathbf{y} = \Phi \mathbf{x}$. 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 $K = \text{round}(M/2.5)$.

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.

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

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 $F_\sigma(\xi)$ for $N = 256$, $M = 100$, $K = 40$, and $\sigma = 0.0218$.](image)
2.3 Minimization of Approximate $\ell_p$ Pseudonorm Using a Quasi-Newton Algorithm

2.3.1 Problem formulation

Let us consider the approximate $\ell_p$ pseudonorm of $x$, namely,

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

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

$$\min_{x} \ | |x| |_{p,\epsilon}^p$$  \hfill (2.12a)

subject to : $\Phi x = y$  \hfill (2.12b)

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

$$\min_{\xi} \ F_{p,\epsilon}(\xi) = \sum_{i=1}^{N} \left[ (x_{si} + v_i^T \xi)^2 + \epsilon^2 \right]^{p/2}$$  \hfill (2.13)

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

$$\dot{g} = p \cdot V_n^T \cdot g$$  \hfill (2.14)

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

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

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

$$H = \text{diag}\{h_{11}, \ h_{22}, \ \cdots, \ h_{NN}\}$$  \hfill (2.16)
where
\[ h_{ii} = p(x_i^2 + \epsilon^2)^{p/2-1} \left[ (p - 1)x_i^2 + \epsilon^2 \right] \quad (2.17) \]

Hence \(||\mathbf{x}||^p_{p,\epsilon}\) is convex if and only if
\[ |x_i| \leq \frac{\epsilon}{\sqrt{1 - p}} \quad \text{for} \ 1 \leq i \leq N \quad (2.18) \]

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 \(\mathbf{x}\) space is proportional to \(\epsilon\).

Using \(H\) in (2.16), the Hessian of function \(F_{p,\epsilon}(\mathbf{\xi})\) in (2.13) is found to be
\[ \hat{H} = V_n^T \cdot H \cdot V_n \]

Hence \(\hat{H}\) is positive definite if \(H\) is positive definite. Consequently, we can use (2.18) and (2.5) to show that \(F_{p,\epsilon}(\mathbf{\xi})\) is convex if
\[ |v_i^T \mathbf{\xi} + x_{si}| \leq \frac{\epsilon}{\sqrt{1 - p}} \quad \text{for} \ 1 \leq i \leq N \quad (2.19) \]

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

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

- First, we obtain the minimum \(\ell_2\)-norm solution of \(\Phi \mathbf{x} = \mathbf{y}\) and use it as the special solution \(\mathbf{x}_s\) in (2.5). An initial value of \(\epsilon\) is selected to satisfy the inequality
\[ \epsilon \geq \sqrt{1 - p} \cdot \max_{1 \leq i \leq N} |x_{si}| \quad (2.20) \]

which will assure that
\[ |x_{si}| \leq \frac{\epsilon}{\sqrt{1 - p}} \quad \text{for} \ 1 \leq i \leq N \quad (2.21) \]

It follows from (2.18) that such an \(\mathbf{x}_s\) is in the convex region of \(||\mathbf{x}||^p_{p,\epsilon}\). Now, (2.21) in conjunction with (2.19) implies that \(\mathbf{\xi} = 0\) is in the convex region of \(F_{p,\epsilon}(\mathbf{\xi})\). This justifies the choice of \(\epsilon\) according to (2.20) and the use of \(\mathbf{\xi} = 0\) as an initial point. A quasi-Newton algorithm such as the BFGS algorithm [44] [50] [51] is applied to minimize \(F_{p,\epsilon}(\mathbf{\xi})\). The minimizer obtained is denoted as
Next, the value of $\epsilon$ is reduced by a certain amount and the BFGS algorithm is applied again to minimize $F_{p,\epsilon}(\xi)$ with $\xi_1^*$ as initial point. The minimizer obtained is denoted as $\xi_2^*$; it is used as initial point in the next iteration.

- This procedure is repeated until the value of $\epsilon$ is reduced to a prescribed target value $\epsilon_J$. The minimizer associated with $\epsilon = \epsilon_J$ is denoted by $\xi_J^*$ and is used in (2.5) to reconstruct the sparse signal as $x^* = x_s + V_n \xi_J^*$.

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 $x$ of length two from measurement $\Phi x = y$, where $\Phi = [0.1 \ 0.06]$ and $y = 0.1$. The least-squares solution of $\Phi x = y$ is found to be $x_s = [0.7353 \ 0.4412]^T$ and the basis vector of the null space of $\Phi$ is given by $v_n = [-0.5145 \ 0.8575]^T$. Using these values of $x_s$ and $v_n$, the objective function $F_{p,\epsilon}(\xi)$ in (2.13) can be expressed as

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

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 $F_{p,\epsilon}(\xi)$ for $\epsilon = 0.8$ as an initial point to minimize $F_{p,\epsilon}(\xi)$ for a slightly reduced value of $\epsilon$, say $\epsilon = 0.5$, and use the minimizer as an initial point to minimize $F_{p,\epsilon}(\xi)$ for a smaller $\epsilon$, say $\epsilon = 0.3$, and so on, the algorithm is likely to lead to the global solution of $F_{p,\epsilon}(\xi)$ for $\epsilon = 0$.

### 2.3.2 Line search based on Banach’s fixed-point theorem

Given parameters $p$ and $\epsilon$, the $k$th iteration of the BFGS algorithm involves a step to solve the one-dimensional optimization problem

$$\min_{\alpha} f(\alpha)$$

(2.22)
where

\[ f(\alpha) = F_{p, \epsilon}(\xi_k + \alpha d_k) \]

Iterate \( \xi_k \) and search direction \( d_k \) 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 \( \alpha \) 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 \( \alpha^* \) in \( R_a \), then \( \alpha^* \) is a local minimizer of the problem in (2.22). The minimizer
\( \alpha^* \) can be obtained by solving the equation

\[
f'(\alpha) = 0 \quad (2.23)
\]

where

\[
f'(\alpha) = \frac{dF_{p,\epsilon}(\xi_k + \alpha d_k)}{d\alpha}
= p \sum_{i=1}^{N} \left[ \gamma_i(\alpha, \epsilon)^{p/2-1} \cdot (x_i + \alpha v_i) \cdot v_i \right] \quad (2.24)
\]

and

\[
\gamma_i(\alpha, \epsilon) = (x_i + \alpha v_i)^2 + \epsilon^2 \quad (2.25)
\]

with \( x_i = x_{si} + v_i^T \xi_k \) and \( v_i = v_i^T d_k \).

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

\[
\alpha = G(\alpha) \quad (2.26)
\]

where

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

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.,

\[
|G_\epsilon(\alpha_1) - G_\epsilon(\alpha_2)| \leq \eta |\alpha_1 - \alpha_2| \quad \text{for any } \alpha_1, \alpha_2
\]

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

\[
\alpha_{l+1} = G_\epsilon(\alpha_l) \quad \text{for } l = 1, 2, \ldots
\]

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 \( \ell_p \)-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 \( \epsilon_1 \) satisfying (2.20), a target value \( \epsilon_T \), and the number of iterations \( T \). The algorithm requires the minimum \( \ell_2 \)-norm solution \( x_{\ell_2} \) and matrix \( V_n \) (see (2.13) and (2.14)) which can be computed using the QR decomposition of \( \Phi^T \) (see Eq. (A.4) in Appendix A).

The \( J - 2 \) values of \( \epsilon \) for which the optimization formulated in (2.13) is carried out are set between the initial value \( \epsilon_0 \) and target value \( \epsilon_T \) as

\[
\epsilon_i = e^{-\beta i} \quad \text{for } i = 2, 3, \ldots, T - 1
\]

(2.30)

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

Table 2.3: UALP Algorithm

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Input: ( p, \epsilon_1, \epsilon_T, ) and ( T ). Set ( \xi^{(1)} = 0 ).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Compute ( \epsilon_i ) for ( i = 2, 3, \ldots, T - 1 ) using (2.30).</td>
</tr>
<tr>
<td>Step 3</td>
<td>Use (A.4) to compute ( x_{\ell_2} ) and ( V_n ).</td>
</tr>
</tbody>
</table>
| Step 4 | Repeat the following for \( k = 1, \ldots, T \):
| | i) Set \( \epsilon = \epsilon_k \) and use \( \xi^{(k)} \) as an initial point. Apply the BFGS algorithm to solve the problem in (2.13).
| | Denote the solution as \( \xi^{(k+1)} \). |
| Step 5 | Set \( \xi^* = \xi^{(T+1)} \), \( x = x_{\ell_2} + V_n \xi^* \), 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 \( \alpha = 0 \), \( \alpha \) 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

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input: $\xi_k, x_k, V_n, d_k, \delta_t,$ and $\epsilon_k$. Set $l = 1, \alpha_1 = 0,$ and $\delta_\alpha = \delta_t + 1$.</td>
</tr>
</tbody>
</table>
| 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) $l = l + 1$. |
| 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 $\Phi$ 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 $y = \Phi x$. With $p = 0.1, \epsilon_1 = \sqrt{1 - p} \cdot \min_{1 \leq i \leq N} |x_{si}|, \epsilon_J = 10^{-5},$ and $J = 9$, the UALP algorithm was applied and compared with the BP [52], iterative reweighted (IR) [34], smoothed $\ell_0$ pseudonorm (SL0) [47], and the null-space reweighted approximate $\ell_0$ 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 $\hat{x} - x$ was less than $10^{-5}$ where $\hat{x}$ and $x$ 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 $K = \text{round}(M/2.5)$ where $N$ varied in the range between 128 to 512. The CPU time...
was measured using a PC desktop with Intel Core 2 CPU 6400 2.13 GHz processor using MATLAB command \texttt{cputime}. The results shown in Figure 2.6 indicate that the UALP algorithm requires the least amount of computation among the algorithms tested.

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

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 \(N = 120, 220, 320, 420, \text{ and } 520\) 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

<table>
<thead>
<tr>
<th>Signal length, (N)</th>
<th>120</th>
<th>220</th>
<th>320</th>
<th>420</th>
<th>520</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed line search</td>
<td>0.1218</td>
<td>0.2824</td>
<td>0.4394</td>
<td>0.6752</td>
<td>1.0678</td>
</tr>
<tr>
<td>Inexact line search</td>
<td>0.3844</td>
<td>0.5799</td>
<td>0.9009</td>
<td>1.4882</td>
<td>2.4240</td>
</tr>
</tbody>
</table>

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 \(N = 256, M = 100, K = 40, \text{ and } \sigma = 0.0118\).

### 2.4 Minimization of Approximate \(\ell_p\) Pseudonorm Using Conjugate-Gradient Algorithm

Recall that the introduction of parameter \(\epsilon\) into function \(F_{p,\epsilon}(\xi)\) 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 \(F_{p,\epsilon}(\xi)\). The sequential optimization procedure used in the algorithm is outlined as follows.

- Select \(\epsilon\) using (2.20), set \(x_s\) to the least-squares solution of \(\Phi x = y\), initialize \(\xi\) to the zero vector of length \(N - M\), and solve the problem in (2.13) using a conjugate-gradient technique. Denote the solution as \(\xi^*\).
Figure 2.7: Function $F_{p,\epsilon}(\xi)$ for $N = 256$, $M = 100$, $K = 40$, and $\epsilon = 0.0118$.

- Update the solution $x_s$ as $x_s = x_s + V_n \xi^*$, reduce the value of $\epsilon$, set $\xi$ 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 $\epsilon$.
- Output $x_s$ as the solution and stop.

In the $k$th iteration of the conjugate-gradient technique, iterate $\xi_k$ is updated as

$$\xi_{k+1} = \xi_k + \alpha_k d_k$$

for $k = 0, 1, \ldots, L - 1$, where $\xi_1 = 0$ and $\alpha_0 = 1$. The $k$th step size is computed using

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

$$\alpha_n = g_k^T g_k \quad \text{and} \quad \alpha_d = d_k^T H_k d_k$$

where $g_k$ is the gradient vector computed using (2.14) and $H_k$ is the Hessian matrix obtained using (2.3.1), respectively, with $x = x_k$. 
The conjugate directions are computed as

\[
d_k = \begin{cases} 
-\mathbf{g}_0 & \text{for } k = 0 \\
-\mathbf{g}_k + \beta_{k-1}d_{k-1} & \text{for } k = 1, 2, \ldots, L - 1 
\end{cases}
\]  

(2.34)

where

\[
\beta_k = \frac{\beta_n}{\beta_d}
\]  

(2.35)

and

\[
\beta_n = \mathbf{g}_{k+1}^T \mathbf{g}_{k+1} \quad \text{and} \quad \beta_d = \mathbf{g}_k^T \mathbf{g}_k
\]  

(2.36)

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

### 2.4.1 Algorithm

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

The target value \( \epsilon_T \) of \( \epsilon \), number of outer iterations \( T \), and the parameters \( E_r \) and \( \delta \) are supplied as input. The initial value \( \epsilon_1 \) of \( \epsilon \) is determined using (2.20) and a total of \( T - 2 \) values lying between \( \epsilon_1 \) and \( \epsilon_T \) are computed by using (2.30).

The Hessian matrix \( \hat{\mathbf{H}}_k \) 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 \( \mathbf{H} \) in (2.16) as

\[
h_{ii} = \begin{cases} 
h_{ii} & \text{if } h_{ii} > \delta \\
\delta & \text{if } h_{ii} \leq \delta 
\end{cases}
\]  

(2.37)

where \( \delta \) is a small positive scalar.

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

\[
\alpha_d = d_k^T \hat{\mathbf{H}}_k d_k = d_k^T \mathbf{V}_n^T \mathbf{H}^{1/2} \mathbf{H}^{1/2} \mathbf{V}_n d_k = \left\| \mathbf{H}^{1/2} \mathbf{V}_n d_k \right\|_2^2
\]  

(2.38)

where \( \mathbf{H}^{1/2} \) is a diagonal matrix whose \( \{i,i\} \)th element is the square root of the
\{i, i\}^\text{th} element of the matrix \( H \) in (2.16).

Table 2.6: The UALP-CG Algorithm

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Input ( p, T, \epsilon_T, \Phi, y, E_t ).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Obtain vector ( x_{\ell_2} ) and matrix ( V_n ) using the QR decomposition.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Set ( x_s = x_{\ell_2} ).</td>
</tr>
<tr>
<td>Step 4</td>
<td>Compute: ( \epsilon_1 ) using (2.20) and ( \epsilon_t ) for ( t = 2, 3, \ldots, T - 1 ) using (2.30).</td>
</tr>
</tbody>
</table>
| Step 5     | For \( t = 1, 2, \ldots, T \)  
| i)     | Set \( \epsilon = \epsilon_t, L_t = 3 + \text{round}(t/4) \). |
| ii)    | Set \( k = 0, \xi_0 = 0, \text{and } E_r = 10^{10} \). |
| iii)   | While \( E_r > E_t \).  
| a)    | Compute \( g_k \) using (2.14). |
| b)     | Compute \( d_k \) using (2.34). |
| c)     | Compute \( \alpha_k \) using (2.32) in conjunction with (2.38). |
| d)     | Compute \( \xi_{k+1} \) using (2.31). |
| e)     | Set \( k = k + 1 \). |
| f)     | Exit loop if \( k > L_t \). |
| g)     | Compute \( E_r = ||\alpha_k d_k||_2 \). |
| vii)   | Set \( x_s = x_s + V_n \xi_k \). |

| Step 6     | Output \( x^* = x_s \) 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 \( x \) 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 \( x \). Measurement matrix \( \Phi \) 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 \( y = \Phi x \). 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 \( x \) from \( y \). Reconstructions were deemed perfect when the maximum absolute valued error between the original \( x \) and reconstructed signal \( \hat{x} \) measured as maximum \( |x_i - \hat{x}_i| \) was less than \( 1e^{-5} \), where \( x_i \) and \( \hat{x}_i \) are the \( i \)th components of vectors \( x \) and \( \hat{x} \), 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 \( K = \text{round}(M/2.5) \) were measured using Matlab function \( \text{cputime} \). 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 \( \ell_0 \) 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 \( \ell_p \) 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 $K = \text{round}(M/2.5)$ 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 $\ell_p$ 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 $\ell_1$-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 $\ell_p$-pseudonorm regularized $\ell_2$ 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 $p$th-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 $\ell_2$ objective function is presented.

### 3.2 $\ell_p$-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

$$y = \Phi x + w$$  \hspace{1cm} (3.1)

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

$$\min_{x} \ |x|_{p,\epsilon}^p \hspace{1cm} (3.2a)$$

subject to $y = \Phi x + w$  \hspace{1cm} (3.2b)

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

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

and $\epsilon$ is a small scalar.

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

$$\min_{x} \ |x|_{p,\epsilon}^p \hspace{1cm} (3.4a)$$

subject to $\ |\Phi x - y||_2^2 \leq \nu \hspace{1cm} (3.4b)$

where $\nu$ 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

$$\min_{x} \ F_{p,\epsilon}(x) = \frac{1}{2} |\Phi x - y||_2^2 + \lambda |x|_{p,\epsilon}^p \hspace{1cm} (3.5)$$
where $\lambda > 0$ is a regularization parameter. In what follows, function $F_{p,\epsilon}(x)$ will be referred to as the $\ell_2, \ell_{p,\epsilon}$ objective function if $\epsilon > 0$ and the $\ell_2, \ell_p$ objective function if $\epsilon = 0$.

3.2.1 Problem formulation

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

$$x = V_r \phi + V_n \xi$$  \hspace{1cm} (3.6)

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

$$\phi = \Sigma^{-1} U^T y$$  \hspace{1cm} (3.7)

If measurement $y$ is corrupted by noise, then vector $\phi$ obtained from (3.7) is not optimal in general and we shall consider both $\phi$ and $\xi$ as independent variables. Using the SVD of $\Phi$, we can simplify the $\ell_2$ term in (3.5) as

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

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

$$\underset{\phi, \xi}{\text{minimize}} \quad F_{p,\epsilon}(\phi, \xi)$$  \hspace{1cm} (3.9)

where

$$F_{p,\epsilon}(\phi, \xi) = \frac{1}{2} \| \Sigma \phi - \tilde{y} \|^2_2 + \lambda \| V_r \phi + V_n \xi \|^p_{p,\epsilon}$$  \hspace{1cm} (3.10)

with $x$ 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 \( k \)th iteration of the proposed algorithm, signal \( x^{(k)} \) is updated as

\[
x^{(k+1)} = x^{(k)} + \alpha d^{(k)}
\]

(3.11)

where

\[
x^{(k)} = V_r \phi^{(k)} + V_n \xi^{(k)}
\]

(3.12a)

\[
d^{(k)} = V_r d_r^{(k)} + V_n d_n^{(k)}
\]

(3.12b)

and \( \alpha > 0 \). The scalar \( \alpha \) is determined using a line search (see Sec. 3.2.3) and the updating vectors \( d_r \) and \( d_n \) assume the forms

\[
d_r^{(k)} = \begin{bmatrix} \delta_{r,1}^{(k)} & \delta_{r,2}^{(k)} & \cdots & \delta_{r,M}^{(k)} \end{bmatrix}^T
\]

(3.13a)

\[
d_n^{(k)} = \begin{bmatrix} \delta_{n,1}^{(k)} & \delta_{n,2}^{(k)} & \cdots & \delta_{n,N-M}^{(k)} \end{bmatrix}^T
\]

(3.13b)

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

\[
\delta_{r,i}^{(k)} = -\frac{\sigma_i u_i + \lambda p s_i}{\sigma_i^2 + \lambda p \beta_i}
\]

(3.14a)

\[
\delta_{n,i}^{(k)} = -\frac{s_i}{\beta_i}
\]

(3.14b)

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

\[
s_i = \sum_{j=1}^{N} x_{ij}^{(k)} v_{ij} \gamma_j(\epsilon)
\]

(3.15a)

\[
\beta_i = \sum_{j=1}^{N} v_{ij}^2 \gamma_j(\epsilon)
\]

(3.15b)
and
\[ \gamma_j(\epsilon) = \left[ \left( x_j^{(k)} \right)^2 + \epsilon^2 \right]^{p/2-1} \]  
(3.16)

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

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 \( \alpha \) in (3.11). In the \( k \)th iteration, this is done by minimizing function
\[ F_{\rho, \epsilon} \left( \phi^{(k)} + \alpha d_r^{(k)}, \xi^{(k)} + \alpha d_n^{(k)} \right) \]  
(3.17)

with respect to \( \alpha \). By applying Banach’s fixed-point theorem [45] [46] to (3.17), the step size \( \alpha \) is found to be
\[ \alpha = -\frac{q_1 + \lambda pq_2}{q_3 + \lambda pq_4} \]  
(3.18)

where
\[ q_1 = \sum_{j=1}^{M} \left( \sigma_j \phi_j^{(k)} - \bar{y}_j \right) \sigma_j d_{r_j}^{(k)} \]  
(3.19a)
\[ q_2 = \sum_{j=1}^{N} x_j^{(k)} d_{v_j}^{(k)} \gamma_j(\alpha, \epsilon) \]  
(3.19b)
\[ q_3 = \sum_{j=1}^{M} \left( \sigma_j d_r^{(k)} \right)^2 \]  
(3.19c)
\[ q_4 = \sum_{j=1}^{N} \left( d_{v_j}^{(k)} \right)^2 \gamma_j(\alpha, \epsilon) \]  
(3.19d)

In (3.19), \( \phi_j^{(k)} \), \( d_{r_j}^{(k)} \), \( x_j^{(k)} \), and \( d_{v_j}^{(k)} \) are the \( j \)th components of \( \phi^{(k)} \), \( d_r^{(k)} \), \( x^{(k)} \), and \( d_v^{(k)} \), respectively, and
\[ \gamma_j(\alpha, \epsilon) = \left[ \left( x_j^{(k)} + \alpha d_{v_j}^{(k)} \right)^2 + \epsilon^2 \right]^{p/2-1} \]
Step size $\alpha$ 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 $\epsilon$ and gradually decreasing $\epsilon$ to $\epsilon_J$. The detailed steps of the sequential optimization are as follows:

- First, set $\epsilon$ to a large value, say, $\epsilon_1$, typically in the range $0.5 \leq \epsilon_1 \leq 1$, and initialize $\phi$ and $\xi$ to the zero vectors of dimensions $M$ and $N - M$, respectively.

- Solve the optimization problem in (3.9) by i) computing descent directions $d_v$ and $d_r$, ii) computing the step size $\alpha$; and iii) updating solution $x$ and coefficient vector $\phi$.

- Reduce $\epsilon$ to a smaller value and again solve the problem in (3.9).

- Repeat this procedure until the specified target value, $\epsilon_J$, is reached.

- Output $x$ as the solution.

### 3.2.5 Algorithm

The proposed $\ell_p,\epsilon$-regularized least-squares ($\ell_p$-RLS) algorithm for reconstructing sparse signals from compressed measurements is summarized in Table 3.1. The regularization parameter $\lambda$, number of iterations $T$, initial value $\epsilon_1$, final value $\epsilon_T$, 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 $\Phi$ and matrices $U$ and $V$ whose columns are, respectively, the left and right singular vectors of $\Phi$. 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 $\epsilon$ lying between the initial value $\epsilon_1$ and final value $\epsilon_T$ 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 \( \phi^{(k)} \) which is computed as

\[
\phi^{(k)} = V_r^T x^{(k)}
\] (3.21)

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Input: ( \lambda, p, \epsilon_1, \epsilon_T, T, L, \Phi, ) and ( y ). Set ( x^{(1)} = 0 ) and ( k = 1 ).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Compute ( \epsilon_j ) for ( j = 2, 3, \ldots, T - 1 ) using (4.17).</td>
</tr>
<tr>
<td>Step 3</td>
<td>Compute the SVD of ( \Phi ) to obtain ( U, \Sigma, V_r, ) and ( V_n ).</td>
</tr>
</tbody>
</table>
| Step 4 | Repeat for \( t = 1, 2, \ldots, T \)  
   i) Set \( \epsilon = \epsilon_t \).  
   ii) Repeat for \( l = 1, 2, \ldots, L \)  
   a) Use \( x^{(k)} \) as an initial value and compute \( \phi^{(k)} \) using Eq. (3.21).  
   b) Compute \( d^{(k)} \) using Eq. (3.12b).  
   c) Compute \( \alpha \) using the line search based on Banach’s fixed-point theorem using Eq. (3.18).  
   e) Compute \( x^{(k+1)} \) using Eq. (3.11).  
   f) Set \( k = k + 1 \). |
| Step 5 | Set \( x = x^{(k)} \) 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 \( x \) with energy value 100 was constructed as follows: i) a vector \( x \) 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 \( \ell_2 \) 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 \( x \). A measurement
matrix $\Phi$ 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 $\Phi$ were made orthonormal relative to each other. The measurement was obtained as $y = \Phi x + w$ where noise vector $w$ was constructed by drawing its components from $\mathcal{N}(0, 0.01^2)$. The proposed $\ell_p$-RLS algorithm was used to reconstruct $x$ from $y$ 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 $\ell_p$-RLS algorithm was compared with that of the BPDN [52], unconstrained approximate $\ell_p$ (UALP) [54] with $p = 0.1$, iterative reweighted (IR) with $p = 0.1$ [34], and smoothed $\ell_0$ pseudonorm (SL0) [47] algorithms. For each algorithm, the signal was deemed reconstructed if the signal-to-noise ratio value, measured as $20 \log_{10} (||x||_2/||x - \hat{x}||_2)$, was greater than 27 dB where $x$ and $\hat{x}$ are the initial and reconstructed signals, respectively. The results are shown in Figure 3.1. As can be seen, the performance of the $\ell_p$-RLS algorithm is better than that of the other algorithms.

![Figure 3.1](image_url)

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

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 $\Phi$ and five $K$-sparse signals $x_1$, $x_2$, $x_3$, $x_4$, and $x_5$ each with different values and locations of nonzero components. Five noisy measurements $y_1$, $y_2$, $y_3$, $y_4$, and $y_5$ were obtained by multiplying the sparse signals by $\Phi$ and adding five different noise vectors.
constructed by drawing their components from $\mathcal{N}(0, 0.01^2)$. The $\ell_p$-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 $\ell_p$-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 $U$, $\Sigma$, $V_r$, and $V_n$ were reused for all five signal reconstructions. For the IR and SL0 algorithms, the pseudo-inverse of $\Phi$ was computed only once as $\Phi^T (\Phi \Phi^T)^{-1}$ 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 $\ell_p$-RLS

![Figure 3.2: Average CPU time required by the $\ell_p$-RLS, UALP, IR, SL0, and BPDN algorithms over 100 runs with $M = N/2$, $K = M/2.5$.](image)

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 $\ell_p$-RLS algorithm converges in a small number of iterations. As an example, Fig. 3.3 shows how the objective function $F_{p, \epsilon}(\phi, \xi)$ 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 $U$, $\Sigma$, $V_r$, and
\textbf{V} 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 $V_r$ and $V_n$, 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 $\Phi$ is constructed by selecting a number of rows of a random orthonormal matrix $R$. In these applications, we can use $V_r = \Phi^T$ and $V_n = \Psi^T$ where $\Psi$ is formed by using the remaining rows of $R$; on the other hand, matrix $U$ 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, $W$ can be taken to be the orthogonal transfrom matrix. In such cases, the measurement matrix $\Phi$ is composed of a number of rows of $W$. Consequently, we can assign $V_r = \Phi^T$ and $V_n = \Psi^T$ where $\Psi$ is formed by using the remaining rows of $W$. In these applications, matrix $U$ is the identity matrix and the singular values...
are all equal to unity.

### 3.3 $\ell_p$-Regularized Least-Squares Optimization Using Conjugate-Gradient Algorithm

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

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 $||x||_{p,\epsilon}^p$ and $F_{p,\epsilon}(x)$ are differentiable.

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

$$
g = \Phi^T (\Phi x - y) + \lambda g_{\ell_p}$$  \hspace{1cm} (3.22)

with $g_{\ell_p} = [g_{\ell_p,1} \ g_{\ell_p,2} \ \cdots \ g_{\ell_p,N}]$ and

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

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

$$H = \Phi^T \Phi + \lambda U$$ \hspace{1cm} (3.23)

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

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

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

#### 3.3.2 Optimization

We begin by illustrating the role played by parameter $\epsilon$ 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 $x$ of length two from measurement $\Phi x = y$, where $\Phi = [0.1 \ 0.06]$ and $y = 0.1$. In what follows, we consider the case where $p = 0.1$ and $\lambda = 0.0007$ in which case function $F_{p,\epsilon}(x)$ assumes the
form
\[
F_{0.1,\epsilon}(\mathbf{x}) = 0.5(0.1x_1 + 0.06x_2 - 0.1)^2 + 0.0007 \left[(x_1^2 + \epsilon^2)^{0.05} + (x_2^2 + \epsilon^2)^{0.05}\right] \tag{3.25}
\]
where \(x_1\) and \(x_2\) are the components of \(\mathbf{x}\). If we let \(\epsilon = 0\), then function \(F_{0.1,\epsilon}(\mathbf{x})\) is reduced to
\[
F_{0.1,0}(\mathbf{x}) = 0.5(0.1x_1 + 0.06x_2 - 0.1)^2 + 0.0007 (|x_1|^{0.1} + |x_2|^{0.1})
\]
Function \(F_{0.1,0}(\mathbf{x})\) is found to have two local minimizers at \(\mathbf{x}^* = [0.9931 \ 0.0001]^T\) and \(\hat{\mathbf{x}} = [0.0001 \ 1.6544]^T\) 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 \(\mathbf{x}^*\). The six plots in Fig. 3.5 show the contours of \(F_{0.1,\epsilon}(\mathbf{x})\) with \(\epsilon = 1, 0.6, 0.2, 0.08, 0.04, \) and 0.008. In each plot, the global minimizer of \(F_{0.1,0}(\mathbf{x})\) (red), the minimizer obtained by minimizing \(F_{0.1,\epsilon}(\mathbf{x})\) with \(\epsilon\) set to the preceding value (black), and the minimizer obtained by minimizing \(F_{0.1,\epsilon}(\mathbf{x})\) with \(\epsilon\) set to the current value and with the preceding minimizer as its initial point (blue) are shown. It is observed that although function \(F_{0.1,\epsilon}(\mathbf{x})\) is nonconvex and possesses multiple minimizers, with an appropriately decreasing \(\epsilon\), the sequential optimization procedure generates a sequence of intermediate solution points that approach the desired global minimizer \(\mathbf{x}^*\) of \(F_{0.1,0}(\mathbf{x})\).

### 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 \(\epsilon\). For each value of \(\epsilon\), we apply the CG algorithm described in [44] to the problem in (3.5). In the \(k\)th iteration, the iterate \(\mathbf{x}_k\) is updated as
\[
\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k \quad \text{for} \quad k = 0, 1, \ldots, L - 1 \tag{3.26}
\]
where \(\mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{L-1}\) are \(L\) conjugate directions and \(\alpha_0, \alpha_1, \ldots, \alpha_{L-1}\) are \(L\) step sizes. The \(k\)th step size is computed using
\[
\alpha_k = \frac{\alpha_n}{\alpha_d} \tag{3.27}
\]
where
\[
\alpha_n = \mathbf{g}_k^T \mathbf{g}_k \quad \text{and} \quad \alpha_d = \mathbf{d}_k^T \mathbf{H}_k \mathbf{d}_k \tag{3.28}
\]
In (3.28), $\mathbf{g}_k$ and $\mathbf{H}_k$ are the gradient vector obtained with $\mathbf{x} = \mathbf{x}_k$ using (3.22) and the Hessian matrix obtained with $\mathbf{x} = \mathbf{x}_k$ using (3.23), respectively.

The conjugate directions are computed as

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

(3.29)

where

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

(3.30)

and

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

(3.31)

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

### 3.3.4 Algorithm

The proposed $\ell_p$-regularized least-squares conjugate-gradient ($\ell_p$-RLS-CG) algorithm is summarized in Table 3.2. The algorithm takes parameter $\lambda$, initial and target values of $\epsilon$, i.e. $\epsilon_1$ and $\epsilon_T$, number of outer iterations $T$, error threshold $E_t$, measurement
Figure 3.5: Contours of function $F_{p,\epsilon}(x)$ for $\epsilon = 1$, 0.6, 0.2, 0.08, 0.04, and 0.008 with $p = 0.1$, $\lambda = 0.0007$, $\Phi = [0.1 0.06]$, and $y = 0.1$. 
vector \( \mathbf{y} \), and measurement matrix \( \Phi \) as input.

A total of \( T - 2 \) values of \( \epsilon \) lying between initial value \( \epsilon_1 \) and target value \( \epsilon_T \) are computed as

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

(3.32)

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

The Hessian matrix \( H_k \) computed in each step of the CG algorithm is made positive definite by setting the diagonal components of matrix \( U \) in (3.23) as

\[
u_i = \begin{cases} 
  u_i & \text{if } u_i > \delta \\
  \delta & \text{if } u_i \leq \delta
\end{cases}
\]

(3.33)

where \( \delta \) is a small positive scalar.

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

\[
\alpha_d = d_k^T H d_k = d_k^T \Phi^T \Phi d_k + \lambda d_k^T U^{1/2} U \lambda^{1/2} d_k
= \| \Phi d_k \|^2 + \lambda \| \mathbf{b}_k \|^2 
\]

(3.34)

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

### 3.3.5 \( \ell_p \)-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 \( \ell_p \)-RLS-CG algorithm.

Recall from the discussion in Sec. 1.2.2.2 in Chapter 1 that, for noiseless measurements parameter \( \lambda \) in (3.5) must be sufficiently small so that the solution of (3.5) satisfies the equality constraint in (1.5b). Similarly, the \( \ell_p \)-RLS-CG algorithm can be used with a sufficiently small \( \lambda \) for the reconstruction of signals from noiseless measurements. However, when the \( \ell_p \)-RLS-CG algorithm is used with a small \( \lambda \), it often yields a suboptimal solution. This is because with a small \( \lambda \) the gradient of the regularization term \( \lambda \| \mathbf{x} \|_{p,\epsilon} \) in the objective function becomes insignificant relative to that of the fidelity term \( \frac{1}{2} \| \Phi \mathbf{x} - \mathbf{y} \|_2^2 \). Consequently, while carrying out the update...
Table 3.2: $\ell_p$-RLS-CG Algorithm

**Step 1**
Input $p$, $T$, $\epsilon_1$, $\epsilon_T$, $\lambda$, $E_t$, $\delta$, $\Phi$, and $y$.
Set $x_s = 0$.

**Step 2**
Compute $\epsilon_t$ 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 $k = 0$, $x_0 = x_s$, and $E_r = 10^{10}$.

iii) While $E_r > E_t$.

a) Compute $g_k$ using (3.22).

b) Compute $d_k$ using (3.29).

   c) Compute $\alpha_k$ using (3.27).

   d) Compute $x_{k+1}$ using (3.26).

   e) Set $k = k + 1$.

   f) Exit loop if $k > L_t$.

   g) Compute $E_r = \|\alpha_k d_k\|_2$.

iv) Set $x_s = x_k$.

**Step 4**
Output $x^* = x_s$ and stop.

in (3.26), the step taken by the iterate along the descent direction of $\frac{1}{2}\|\Phi x - y\|_2^2$ is significantly longer than that taken along the descent direction of $\lambda\|x\|_{p,\epsilon}^p$. We deal with this technical difficulty by initializing $\lambda$ using a large value, say, $\lambda_1$, and decrease its value every time the value of $\epsilon$ is decreased. If we denote a sufficiently small target value of $\lambda$ as $\lambda_T$, then a total of $(T - 2)$ values of $\lambda$ in between $\lambda_1$ and $\lambda_T$ are computed as

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

(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: $\ell_p$-RLS-CG Algorithm for Noiseless Measurements

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Input $p$, $T$, $\epsilon_1$, $\epsilon_T$, $\lambda_1$, $\lambda_T$, $E_t$, $\delta$, $\Phi$, and $y$. Set $x_s = 0$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Compute $\epsilon_t$ for $t = 2, 3, \ldots, T - 1$ using (3.32). Compute $\lambda_t$ for $t = 2, 3, \ldots, T - 1$ using (3.35).</td>
</tr>
<tr>
<td>Step 3</td>
<td>Repeat for $t = 1, 2, \ldots, T$</td>
</tr>
<tr>
<td></td>
<td>i) $\epsilon = \epsilon_t$, $\lambda = \lambda_t$, and $L_t = 3 + \text{round}(t/4)$.</td>
</tr>
<tr>
<td></td>
<td>$\vdots$</td>
</tr>
<tr>
<td></td>
<td>(Steps same as in Table 3.2)</td>
</tr>
<tr>
<td></td>
<td>$\vdots$</td>
</tr>
<tr>
<td>Step 4</td>
<td>Output $x^* = x_s$ and stop.</td>
</tr>
</tbody>
</table>

3.3.6 Simulation results

3.3.6.1 Signal reconstruction from noisy measurements using $\ell_p$-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 $x$ 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 $\ell_2$ 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 $x$. A measurement matrix $\Phi$ 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 $\Phi$ are orthonormal relative to each other. A noisy measurement was taken using $y = \Phi x + w$ where $w$ is a measurement noise vector of length $M$ whose components were drawn from $\mathcal{N}(0, 0.01^2)$. The $\ell_p$-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 $20 \log (\|x\|_2/\|x - \hat{x}\|_2)$ where $\hat{x}$ 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 $\ell_0$ (SL0)
algorithm [47], the basis pursuit algorithm with denoising (BPDN) [52], and the proposed $\ell_p$-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

![Figure 3.6a: Percentage of recovered instances](image)

![Figure 3.6b: Average signal-to-noise ratio](image)

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

The \( \ell_p \)-RSL-CG algorithm is higher than that of the other algorithms with which it is being compared. Also, the average SNR performance of the \( \ell_p \)-RSL-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 \( \sigma^2 \) 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 \( \ell_p \)-RSL-CG algorithm is much higher than that for the other algorithms.

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

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 \( \mathbf{x} \) and measurement matrix \( \mathbf{\Phi} \) were constructed, and the noisy measurement \( \mathbf{y} \) was taken as in the first experiment. The \( \ell_p \)-RSL-CG, SL0, IR, and BPDN algorithms were used to reconstruct signal \( \mathbf{x} \) from measurement \( \mathbf{y} \). 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 \textit{cputime}. The results are plotted in
Figure 3.8. It is observed that the CPU time required by the $\ell_p$-RLS-CG algorithm is

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 $\ell_p$-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 $\mathbf{x}$ was constructed as follows: i) a vector $\mathbf{x}$ 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 $\mathbf{x}$. Measurement matrix $\Phi$ was constructed and measurement $\mathbf{y}$ was taken as in the first experiment. The $\ell_p$-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 $\ell_p$-RLS-CG, SL0, IR with $p = 0.1$, and BP algorithm were used to reconstruct $\mathbf{x}$ from $\mathbf{y}$. The maximum value of the absolute error between the true signal $\mathbf{x}$ and the reconstructed signal $\hat{\mathbf{x}}$ was measured as $E = \max \{|x_i - \hat{x}_i|\}$ where $x_i$ and $\hat{x}_i$ are the $i$th components of vectors $\mathbf{x}$ and $\hat{\mathbf{x}}$, 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 $\ell_p$-RLS-CG algorithm has performed better than 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

![Figure 3.9: Number of perfect reconstructions and average CPU time required for the $\ell_p$-RLS-CG, SL0, IR, and BP algorithms.](image-url)
computation required by each algorithm to reconstruct the signal was measured using MATLAB command \textit{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 $\ell_p$-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 $\lambda$

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

$$\|\Phi x - y\|_2^2 = \|w\|_2^2 \approx M \sigma^2$$

(3.36)

where $M$ denotes the dimension of $w$.

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

$$\begin{align*}
\text{set } \lambda_u &= \lambda \text{ if } \|\Phi x_\lambda - y\|_2^2 > M \sigma^2 \\
\text{set } \lambda_l &= \lambda \text{ if } \|\Phi x_\lambda - y\|_2^2 \leq M \sigma^2
\end{align*}$$

(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 $\lambda$.

3.4.1 Algorithm

An algorithm based on the above analysis is summarized in Table 3.4. Step 2 of the algorithm runs the $\ell_p$-RLS-CG algorithm with parameters $p$, $T$, $\epsilon_1$, $\epsilon_T$, $\lambda_0$, error threshold $E_t$, and $\delta$ and determines a sparse solution. Step 3 of the algorithm uses the parameters $p$, $\epsilon_T$, $\lambda_u$, $\lambda_l$, inner error threshold $E_{t_i}$, outer error threshold $E_{to}$, the
maximum number of CG iterations $L$, and $\delta$ and determines an optimal value of parameter $\lambda$ and the corresponding optimal solution denoted as $\bm{x}_\lambda$.

Table 3.4: $\ell_p$-RLS-CG Algorithm for Noiseless Measurements by Optimizing $\lambda$

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Input $p, T, \varepsilon_1, \varepsilon_T, \lambda_u, \lambda_l, \Phi, \mathbf{y}, E_\ell, E_{to}, \delta, \sigma$. Set $\mathbf{x}<em>s = \mathbf{0}$, $E</em>{ro} = 10^{10}$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Repeat the following while $E_{ro} &gt; E_{to}$.</td>
</tr>
<tr>
<td>i)</td>
<td>Compute $\lambda = (\lambda_l + \lambda_u)/2$.</td>
</tr>
<tr>
<td>ii)</td>
<td>Solve the problem in (3.5) using the $\ell_2$-RLS-CG algorithm summarized in Table 3.2 with $\mathbf{x}<em>s$ as the initializer. Denote the resulting solution as $\mathbf{x}</em>\lambda$.</td>
</tr>
<tr>
<td>iii)</td>
<td>Update $\lambda_u$ or $\lambda_l$ using (3.37).</td>
</tr>
<tr>
<td>iv)</td>
<td>Compute $E_{ro} = \lambda_u - \lambda_l$.</td>
</tr>
<tr>
<td>v)</td>
<td>Set $\mathbf{x}<em>s = \mathbf{x}</em>\lambda$.</td>
</tr>
<tr>
<td>Step 3</td>
<td>Compute $\lambda = (\lambda_l + \lambda_u)/2$.</td>
</tr>
<tr>
<td></td>
<td>Solve the problem in (3.5) using the $\ell_2$-RLS-CG algorithm. Denote the resulting solution as $\mathbf{x}_\lambda$.</td>
</tr>
<tr>
<td>Step 4</td>
<td>Output $\mathbf{x}^* = \mathbf{x}_\lambda$ and stop.</td>
</tr>
</tbody>
</table>

3.4.2 Experimental results

The performance of the proposed technique for the optimization of parameter $\lambda$ in the $\ell_p$-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 $\ell_2$ norm of the resulting vector is $\sqrt{100}$. A measurement matrix $\Phi$ 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 $\Phi$ were orthonormal with respect to each other. A noisy measurement was taken using $\mathbf{y} = \Phi \mathbf{x} + \mathbf{w}$, where $\mathbf{w}$ is a measurement noise vector of length $M$ whose components were drawn from $\mathcal{N}(0, 0.01^2)$. The $\ell_p$-RLS-CG algorithm incorporating the optimization of $\lambda$ was ap-
plied with parameters $p = 0.1$, $T = 80$, $\epsilon_1 = 1$, $\epsilon_T = 1e-2$, $\lambda_l = 0$, $\lambda_u = 0.005$, $E_t = 1e-25$, $E_{io} = 1e-4$, and $\delta = 1e-5$. The $\ell_p$-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 $\ell_p$-RLS-CG incorporating the optimization of $\lambda$, $\ell_p$-RLS-CG, SL0, IR, and BPDN algorithms were used to recover signal $x$ from $y$. The SNR was measured as $20 \log(||x||_2 / ||x - x^*||_2)$, where $x$ and $x^*$ 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 $\lambda$ for the $\ell_p$-RLS-CG algorithm yields improved performance. The $\ell_p$-RLS-CG algorithm incorporating the optimization of $\lambda$ took a total of six iterations in Step 2 (see Table 3.4) and the resulting optimal value of $\lambda$ was $4.9219e-03$.

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 $\ell_p$-RLS-CG algorithm with the proposed
method for the optimization of $\lambda$ 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 $\ell_p$-RLS-CG algorithm with and without optimizing $\lambda$ 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 $p$th-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 $\ell_2$ 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 $X$ 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 $X$ are defined as follows. For image $X$, the discrete isotropic TV norm, denoted as $TV_I(X)$, and anisotropic TV norm, denoted as $TV_{II}(X)$, are defined, respectively, as

$$TV_I(X) = \sum_{i=1}^{n_1-1} \sum_{j=1}^{n_2-1} \sqrt{(x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2}$$

$$+ \sum_{i=1}^{n_1-1} |x_{i,n_2} - x_{i+1,n_2}| + \sum_{j=1}^{n_2-1} |x_{n_1,j} - x_{n_1,j+1}|$$

and

$$TV_{II}(X) = \sum_{i=1}^{n_1-1} \sum_{j=1}^{n_2-1} \frac{1}{2} (|x_{i,j} - x_{i+1,j}| + |x_{i,j} - x_{i,j+1}|)$$

$$+ \sum_{i=1}^{n_1-1} |x_{i,n_2} - x_{i+1,n_2}| + \sum_{j=1}^{n_2-1} |x_{n_1,j} - x_{n_1,j+1}|$$

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

Consider the image model

$$Y = X + W$$

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

$$\min_X \frac{1}{2} ||X - Y||_F^2 + \lambda TV(X)$$

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

$$\min_{\mathbf{X}} \frac{1}{2} ||\mathbf{x} - \mathbf{y}||_2^2 + \lambda TV(\mathbf{X})$$  (3.40)

On comparing (3.40) with the $\ell_2, \ell_1$ optimization problem given in (1.18), namely,

$$\min_{\mathbf{x}} \frac{1}{2} ||\Phi \mathbf{x} - \mathbf{y}||_2^2 + \lambda ||\mathbf{x}||_1$$  (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 $\ell_p$ minimization in previous chapters, we propose to investigate possible extensions of the TV norm similar to our extension of the $\ell_1$ norm to the $\ell_p$ pseudonorm.

To this end, we introduce the $TV_p$ pseudonorm of $\mathbf{X}$, $TV_p(\mathbf{X})$, as follows

$$TV_p^p(\mathbf{X}) = \sum_{i=1}^{n_1-1} \sum_{j=1}^{n_2-1} \left[ (x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2 \right]^{p/2}$$

$$+ \sum_{i=1}^{n_1-1} |x_{i,n_2} - x_{i+1,n_2}|^p$$

$$+ \sum_{j=1}^{n_2-1} |x_{n_1,j} - x_{n_1,j+1}|^p$$  (3.42)

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

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

$$TV_{p,\epsilon}^p(\mathbf{X}) = \sum_{i=1}^{n_1-1} \sum_{j=1}^{n_2-1} \left[ (x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2 + \epsilon^2 \right]^{p/2}$$

$$+ \sum_{i=1}^{n_1-1} \left[ (x_{i,n_2} - x_{i+1,n_2})^2 + \epsilon^2 \right]^{p/2} + \sum_{j=1}^{n_2-1} \left[ (x_{n_1,j} - x_{n_1,j+1})^2 + \epsilon^2 \right]^{p/2}$$  (3.43)

where $\epsilon$ is a small approximation parameter.
3.5.2 Problem formulation

Let a vector \( \mathbf{x} \) of size \( n_1 n_2 \) be constructed as

\[
\mathbf{x} = \begin{bmatrix}
\mathbf{x}_1 \\
\mathbf{x}_2 \\
\vdots \\
\mathbf{x}_{n_2}
\end{bmatrix}
\]

where \( \mathbf{x}_i \) is the \( i \)th column of an image \( \mathbf{X} \). Let a measurement vector \( \mathbf{y} \) of size \( m \) be obtained as

\[
\mathbf{y} = \Phi \mathbf{x} + \mathbf{w}
\]

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

\[
\| \Phi \mathbf{x} - \mathbf{y} \|_2^2 = \| \mathbf{w} \|_2^2 \approx m \sigma^2
\]

(3.45)

In image processing applications, matrix \( \Phi \) 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 \( \mathbf{X} \) from measurements \( \mathbf{y} \) by solving the optimization problem

\[
\min_{\mathbf{X}} \quad F_{p,\epsilon}(\mathbf{X}) = \| \Phi \mathbf{x} - \mathbf{y} \|_2^2 + \lambda \cdot TV_{p,\epsilon}(\mathbf{X})
\]

(3.46)

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

\[
\mathbf{g} = \Phi^T (\Phi \mathbf{x} - \mathbf{y}) + \lambda \mathbf{u}
\]

(3.47)

where \( \mathbf{u} = [\mathbf{u}_1^T \mathbf{u}_2^T \cdots \mathbf{u}_{n_2}^T]^T \) and \( \mathbf{u}_i \) is the \( i \)th column of the gradient matrix \( \mathbf{U} \).
given by

\[
U = \begin{bmatrix}
    u_{11} & u_{12} & \cdots & u_{1n_2} \\
    u_{21} & u_{22} & \cdots & u_{2n_2} \\
    \vdots & \vdots & \ddots & \vdots \\
    u_{n_11} & u_{n_12} & \cdots & u_{n_1n_2}
\end{bmatrix}
\]  

(3.48)

where

\[
u_{ij} = \frac{\partial TV_{p,e}(X)}{\partial x_{ij}} \quad \text{for } 1 \leq i \leq n_1 \text{ and } 1 \leq j \leq n_2
\]  

(3.49)

Before we give the explicit expression for \(u_{i,j}\), let \(Q_h\) and \(Q_v\) be matrices of sizes \(n_1 \times (n_2 - 1)\) and \((n_1 - 1) \times n_2\), respectively, given by

\[
Q_h = [q^h_1 \, q^h_2 \, \cdots \, q^h_{n_2}]
\]  

(3.50)

and

\[
Q_v = [q_v^1 \, q_v^2 \, \cdots \, q_v^{n_1}]^T
\]  

(3.51)

where

\[
q^h_i = x_i - x_{i+1} \quad \text{for } i = 1, 2, \ldots, n_2 - 1
\]

\[
q_v^i = x^i - x^{i+1} \quad \text{for } i = 1, 2, \ldots, n_1 - 1
\]

and \(x_i\) and \((x^i)^T\) denote the \(i\)th column and the \(i\)th row, respectively, of \(X\).

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

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

where \(q_{i,j}^v\) denotes the \(\{i, j\}\)th component of the matrix \(Q^v\) and \(q_{i,j}^h\) denotes the \(\{i, j\}\)th
component of the matrix $Q^h$. Then, $u_{i,j}$ in (3.49) can be computed using

$$u_{i,j} = \begin{cases} 
  q_{i,j} (q_{i,j}^v + q_{i,j}^h) & \text{for } i = 1, j = 1 \\
  q_{i,j} (q_{i,j}^v + q_{i,j}^h) - q_{i,j-1} q_{i,j-1}^h & \text{for } i = 1, 1 < j \leq n_2 \\
  q_{i,j} (q_{i,j}^v + q_{i,j}^h) - q_{i-1,j} q_{i-1,j}^v & \text{for } 1 < i \leq n_1, j = 1 \\
  q_{i,j} q_{i,j}^h - q_{i-1,j} q_{i-1,j}^h & \text{for } 1 < i < n_1, 1 < j < n_2 \\
  q_{i,j} q_{i,j}^h - q_{i-1,j} q_{i-1,j}^v & \text{for } 1 < i < n_1, j = n_2 \\
  q_{i,j} q_{i,j}^h - q_{i-1,j} q_{i-1,j}^v & \text{for } i = n_1, 1 < j < n_2 \\
  q_{i,j} q_{i,j}^h - q_{i-1,j} q_{i-1,j}^v & \text{for } i = n_1, j = n_2
\end{cases}$$

3.5.3 Optimization

As may be expected, the value of $\epsilon$ affects the behaviour of function $TV_{p,\epsilon}^p(X)$. 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 $\epsilon$ using a finite number of iterations of Fletcher-Reeves’ CG algorithm [44].

In the $k$th iteration of the CG algorithm, iterate $x_k$ is updated as in (3.26). The conjugate directions $\{d_k\}$ are computed using (3.29) where gradient $g$ 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 $\alpha_k$ 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 $k$th iterate $x_k$ and conjugate direction $d_k$ be divided into $n_2$ blocks as $x_k = \begin{bmatrix} x_{k1}^T & x_{k2}^T & \cdots & x_{kn_2}^T \end{bmatrix}^T$ and $d_k = \begin{bmatrix} d_{k1}^T & d_{k2}^T & \cdots & d_{kn_2}^T \end{bmatrix}^T$ where $x_{ki}$ and $d_{ki}$ are the $i$th blocks of vectors $x_k$ and $d_k$ and let matrices $X_k$ and $D_k$ be constructed as $X_k = \begin{bmatrix} x_{k1} & x_{k2} & \cdots & x_{kn_2} \end{bmatrix}$ and $D_k = \begin{bmatrix} d_{k1} & d_{k2} & \cdots & d_{kn_2} \end{bmatrix}$. Under these circumstances, $\alpha_k$ can be obtained as

$$\alpha_k = \arg \min_{\alpha} F_{p,\epsilon}(X_k + \alpha D_k)$$

By equating the derivative $dF_{p,\epsilon}(X_k + \alpha D_k)/d\alpha$ to zero, we obtain

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

(3.52)
where

\[
G(\alpha) = \frac{-a + \lambda \cdot p \cdot b}{c + \lambda \cdot p \cdot d} \tag{3.53}
\]

\[
a = d_k^T \Phi^T (\Phi x_k - y)
\]

\[
b = \sum_{i=1}^{n_1-1} \sum_{j=1}^{n_2-1} \hat{b}_{i,j} + \sum_{i=1}^{n_1-1} \hat{b}_{i,n_2} + \sum_{j=1}^{n_2-1} \hat{b}_{n_1,j}
\]

\[
c = d_k^T \Phi^T \Phi d_k
\]

\[
d = \sum_{i=1}^{n_1-1} \sum_{j=1}^{n_2-1} \hat{d}_{i,j} + \sum_{i=1}^{n_1-1} \hat{d}_{i,n_2} + \sum_{j=1}^{n_2-1} \hat{d}_{n_1,j}
\]

In the above equations,

\[
\hat{b}_{i,j} = \gamma_{i,j}(\alpha) [q^v_{i,j}(d_{i,j} - d_{i+1,j}) + q^h_{i,j}(d_{i,j} - d_{i,j+1})]
\]

\[
\hat{b}_{i,n_2} = \gamma_{i,n_2}(\alpha) \cdot q^v_{i,n_2}(d_{i,n_2} - d_{i+1,n_2})
\]

\[
\hat{b}_{n_1,j} = \gamma_{n_1,j}(\alpha) \cdot q^h_{n_1,j}(d_{n_1,j} - d_{n_1,j+1})
\]

\[
\hat{d}_{i,j} = \gamma_{i,j}(\alpha) \cdot [(d_{i,j} - d_{i,j+1})^2 + (d_{i,j} - d_{i,j+1})^2]
\]

\[
\hat{d}_{i,n_2} = \gamma_{i,n_2}(\alpha) \cdot (d_{i,n_2} - d_{i+1,n_2})^2
\]

\[
\hat{d}_{n_1,j} = \gamma_{n_1,j}(\alpha) \cdot (d_{n_1,j} - d_{n_1,j+1})^2
\]

At this point, \(\gamma_{i,j}(\alpha)\) can be determined as follows:

For \(1 \leq i < n_1\) and \(1 \leq j < n_2\),

\[
\gamma_{i,j}(\alpha) = \left\{ \left[ q^v_{i,j} + \alpha(d_{i,j} - d_{i+1,j}) \right]^2 + \left[ q^h_{i,j} + \alpha(d_{i,j} - d_{i,j+1}) \right]^2 + \epsilon^2 \right\}^{p/2-1}
\]

For \(1 \leq i < n_1\) and \(j = n_2\),

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

For \(i = n_1\) and \(1 \leq j < n_2\),

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

In the above relations, \(q^h_{i,j}\) and \(q^v_{i,j}\) are the \(i,j\)th components of matrices \(Q^h\) and
Finding step size $\alpha_k$ 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

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

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 $\epsilon_1$ and target value $\epsilon_T$ of $\epsilon$, initial value $\lambda_1$ and target value $\lambda_T$ of $\lambda$, measurement matrix $\Phi$, measurement vector $y$, and error tolerance $E_\epsilon$ are supplied as input. A total of $(T - 2)$ values of $\epsilon$ in between $\epsilon_1$ and $\epsilon_T$ are computed using (3.32) and a total of $(T - 2)$ values of $\lambda$ in between $\lambda_1$ and $\lambda_T$ are computed as

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

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 $256 \times 256$ was vectorized into a vector $x$ 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 $x$ 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 $X$ was measured.
Table 3.5: TV\textsubscript{p}-RLS-CG Algorithm

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Input: ( p, T, \epsilon_1, \epsilon_T, \lambda_1, \lambda_T, \Phi, y, ) and ( E_t. ) Set ( x_s = 0. )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Compute ( \epsilon_t ) for ( t = 2, 3, \ldots, T - 1 ) using (3.32) and ( \lambda_t ) for ( t = 2, 3, \ldots, T - 1 ) using (3.55).</td>
</tr>
</tbody>
</table>
| Step 3 | Repeat the following for \( t = 1, \ldots, T \)  
  i) Set \( \epsilon_t = \epsilon_t, \lambda_t = \lambda_t, L_t = 3 + \text{round}(t/4). \)  
  ii) Set \( k = 0, x_0 = x_s, E_r = 10^{10}. \)  
  iii) Repeat the following while \( E_r > E_t, \)  
      a) Compute \( g_k \) using (3.47).  
      b) Compute \( d_k \) using (3.29).  
      c) Compute \( a_k \) using (3.54).  
      d) Compute \( x_{k+1} \) using (3.26).  
      e) Set \( k = k + 1. \)  
      f) Exit loop if \( k > L_t. \)  
      g) Compute \( E_r = \|a_k d_k\|_2. \)  
  iv) Set \( x_s = x_k. \) |
| Step 4 | Output \( x^* = x_s \) 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 \( X, \) and MSE is the mean-square error which is defined as

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

Matrices \( X \) and \( \hat{X} \) 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\textsubscript{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 \texttt{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 $\ell_p$-pseudonorm regularized least-squares algorithm for the reconstruction of sparse signals from noisy measurements was presented. The $\ell_p$-RLS algorithm minimizes an $\ell_p$-pseudonorm regularized $\ell_2$ 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 $\ell_p$-pseudonorm regularized least-squares conjugate-gradient algorithm for the reconstruction of sparse signals from noisy and noiseless measurements has been presented. The $\ell_p$-RLS-CG algorithm was developed by minimizing an $\ell_p$-pseudonorm regularized $\ell_2$ error using a sequential CG algorithm. The use of the CG algorithm renders the algorithm suitable for large-sized data. Two extensions of the $\ell_p$-RLS-CG algorithm, namely, a) a technique for the application of $\ell_p$-RLS-CG algorithm to noiseless data and b) a technique for the optimization of the regularization parameter involved in the $\ell_p$-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 $\ell_2$ 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 $\ell_p$-RLS algorithm yields improved signal reconstruction performance compared to the UALP
Figure 3.12: Reconstruction performance of the $TV_p$-RLS-CG algorithm versus Romberg’s algorithms.
Figure 3.13: Original image and the images reconstructed by using the $TV_p$-RLS-CG algorithm with $p = 0.1,$ $p = 0.5$ and Romberg’s algorithms.

When applied for signal reconstruction from noiseless measurements, the $\ell_p$-RLS-CG algorithm yields improved signal reconstruction performance and requires a reduced amount of computation compared to the IR and BPDN algorithms. Compared to the SL0 algorithm, the $\ell_p$-RLS-CG algorithm requires slightly more computation.
amount of computation compared to the SL0, IR, and BPDN algorithms. Also, the proposed technique for the optimization of the regularization parameter in the $\ell_p$-RLS-CG algorithm yields significantly improved signal reconstruction performance and requires almost the same amount computation as that required by the $\ell_p$-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 $\ell_p$-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 $\ell_2$ error using a conjugate-gradient (CG) algorithm known as Fletcher-Reeves’ algorithm.

#### 4.2.1 Problem formulation

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

$$x = [\tilde{x}_1^T \tilde{x}_2^T \cdots \tilde{x}_{N/d}^T]^T$$

where

$$\tilde{x}_i = [x_{(i-1)d+1} x_{(i-1)d+2} \cdots x_{(i-1)d+d}]^T$$

for $i = 1, 2, \ldots, N/2$ and $x_i$ is the $i$th component of $x$.

A signal $x$ from (4.1) is said to be $K$-block sparse if $x$ 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

$$\begin{align*}
\text{minimize} & \quad ||x||_{2/1} \\
\text{subject to} & \quad \Phi x = y
\end{align*}$$

(4.2)

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

$$||x||_{2/1} = \sum_{i=1}^{N/2} ||\tilde{x}_i||_2$$
where $||\tilde{x}_i||_2$ is the $\ell_2$ norm of the $i$th block $\tilde{x}_i$ [40], [41], [42].

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

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

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

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

where $w$ represents measurement noise.

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

$$
\begin{align*}
\text{minimize} & \quad ||x||^p_{2/p,\epsilon} \\
\text{subject to} & \quad y = \Phi x + w 
\end{align*} \tag{4.5a}
$$

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

$$
\begin{align*}
\text{minimize} & \quad ||x||^p_{2/p} \\
\text{subject to} & \quad ||\Phi x - y||_2^2 \leq \nu 
\end{align*} \tag{4.6a}
$$

where $\nu$ 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

$$
\begin{align*}
\text{minimize} & \quad F_{2/p,\epsilon}(x) = \frac{1}{2} ||\Phi x - y||_2^2 + \lambda ||x||^p_{2/p,\epsilon} 
\end{align*} \tag{4.7}
$$

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

Note that function $||x||^p_{2/p,\epsilon}$ remains differentiable and so is function $F_{2/p,\epsilon}(x)$ in
(4.7) as long as $\epsilon$ is kept positive. In effect, for $\epsilon > 0$ the gradient of $F_{2/p,\epsilon}(x)$ is given by

$$g = \Phi^T (\Phi x - y) + \lambda u$$

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

$$u = [\tilde{u}_1^T \tilde{u}_2^T \cdots \tilde{u}_{N/d}^T]^T$$

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

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

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

### 4.2.2 Optimization

Parameter $\epsilon$ in (4.3) has the effect of smoothing function $||x||_{2/p,\epsilon}$ and thereby making function $F_{2/p,\epsilon}(x)$ 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 $\epsilon$, the region over which function $F_{2/p,\epsilon}(x)$ is convex is large, where a minimizer of $F_{2/p,\epsilon}(x)$ can be easily located. On the other hand, the global minimizer of function $F_{2/p,\epsilon}(x)$ is an accurate approximation of the optimal solution of the problem in (4.7) for a sufficiently small value of $\epsilon$. We, therefore, solve the problem in (4.7) for a sequence of decreasing values of $\epsilon$ 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 $\epsilon$. For each value of $\epsilon$, 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 $F_{2/p,\epsilon}(x)$. In the $k$th
iteration, iterate $x_k$ is updated as

$$x_{k+1} = x_k + \alpha_k d_k \quad \text{for} \quad k = 0, 1, \ldots, L - 1 \quad (4.11)$$

where $d_0, d_1, \ldots, d_{L-1}$ are $L$ conjugate directions and $\alpha_0, \alpha_1, \ldots, \alpha_{L-1}$ are $L$ step sizes computed using the line-search technique described in Sec. 4.2.4. The conjugate directions are computed as

$$d_k = \begin{cases} -g_0 & \text{for} \quad k = 0 \\ -g_k + \beta_{k-1} d_{k-1} & \text{for} \quad k = 1, 2, \ldots, L - 1 \end{cases} \quad (4.12)$$

where $g_k$ is the gradient computed by using $x = x_k$ in (4.8) and $\beta_k = \beta_n/\beta_d$ where

$$\beta_n = g_{k+1}^T g_{k+1} \quad \text{and} \quad \beta_d = g_k^T g_k$$

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

### 4.2.4 Line search

Given parameters $p, \epsilon, \text{and} \lambda$, the step size $\alpha_k$ in Fletcher-Reeves’ algorithm is obtained by solving the one-dimensional optimization problem

$$\min_{\alpha} F_{2/p, \epsilon} (x_k + \alpha d_k) \quad (4.13)$$

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

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

In $G(\alpha), \tilde{x}_{ki}$ and $\tilde{d}_{ki}$ are the $i$th blocks of vectors $x_k$ and $d_k$, respectively, and

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

Finding step size $\alpha_k$ 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


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

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 $\epsilon_1$ and target value $\epsilon_T$ of $\epsilon$, and parameter $\lambda$ are supplied as input.

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

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

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

For noise-free measurement $y$, a large initial value $\lambda_1$ and a small target value $\lambda_T$ are supplied as input instead of $\lambda$. A total of $T - 2$ values of $\lambda$ are chosen as

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

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 $x$ 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 $\Phi$ 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 $\Phi$ were made orthonormal relative to each other. The measurement was obtained as $y = \Phi x$. 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: $p$, $T$, $\epsilon_1$, $\epsilon_T$, $\Phi$, $y$, $E_t$,
\[ \lambda \text{ if measurement } y \text{ is noisy, and} \]
\[ \lambda_1 \text{ and } \lambda_T \text{ if measurement } y \text{ is noiseless.} \]
Set $x_s = 0$.

**Step 2**
Compute $\epsilon_t$ for $t = 2, 3, \ldots, T - 1$ using (4.17) and
\[ \lambda_t \text{ for } t = 2, 3, \ldots, T - 1 \text{ using (4.18)} \]
if measurement $y$ is noiseless.

**Step 3**
For $t = 1, \ldots, T$
\[ \text{i) Set } \epsilon = \epsilon_t, \quad L_t = 3 + \text{round}(t/4). \]
\[ \text{ii) If measurement is noiseless, set } \lambda = \lambda_t. \]
\[ \text{iii) Set } k = 0, \quad x_0 = x_s, \quad E_r = 10^{10}. \]
\[ \text{iv) Repeat the following while } E_r > E_t, \]
\[ \text{a) Compute } g_k \text{ using (4.8).} \]
\[ \text{b) Compute } d_k \text{ using (4.12).} \]
\[ \text{c) Compute } \alpha_k \text{ using (4.16).} \]
\[ \text{d) Compute } x_{k+1} \text{ using (4.11).} \]
\[ \text{e) Set } k = k + 1. \]
\[ \text{f) Exit loop if } k > L_t. \]
\[ \text{g) Compute } E_r = ||\alpha_k d_k||_2. \]
\[ \text{v) Set } x_s = x_k. \]

**Step 4**
Output $x^* = x_s$ and stop.

using $\ell_{2/1}$ SOCP [69], the iterative reweighted (IR) with $p = 0.1$ [34], the smoothed $\ell_0$
pseudonorm (SL0) [47], the BOMP [42], and the BP [52] algorithms. Reconstruction
was deemed successful if the maximum absolute error between the original signal, $x$,
and recovered signal, $\hat{x}$, 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 $i$th components of $x$ and $\hat{x}$, respectively. The percentage of the num-
ber 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 $N = 128, 256, 512, 1024, 2048, 4096, \text{and } 8192$. 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.
4.3 A Reweighting Technique for Complexity Reduction

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

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

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

where $w_i$ is the weight for the $i$th term in the summation and $\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_{N/d}]^T$ is a weight vector.
We propose to reconstruct signal $x$ by solving the optimization problem

$$\minimize_x F_{2/p,\epsilon,w}(x) = \frac{1}{2} ||\Phi x - y||^2_2 + \lambda ||x||^p_{2/p,\epsilon,w}$$ (4.20)

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

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

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

Note that function $||x||^p_{2/p,\epsilon,w}$ in (4.19) is different from function $||x||^p_{2/p,\epsilon}$ in (4.7) in that the behaviour of $||x||^p_{2/p,\epsilon,w}$ depends on the values of the components of the weight vector $w$. As a consequence, the minimization of function $F_{2/p,\epsilon,w}(x)$ in (4.20) can be accelerated by using appropriate values for the components of $w$. Below, a method for the optimization of function $F_{2/p,\epsilon,w}(w)$ is presented and a method for the estimation of weight vector $w$ 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 $F_{2/p,\epsilon,w}(x)$ in (4.20) becomes differentiable, 
b) for a sufficiently large value of $\epsilon$, the region over which function $F_{2/p,\epsilon,w}(x)$ is convex is large, where a minimizer of $F_{2/p,\epsilon,w}(x)$ can be easily located, and 
c) the global minimizer of function $F_{2/p,\epsilon,w}(x)$ is an accurate approximation of the optimal solution of the problem in (4.20) for an appropriate weight vector $w$ and a sufficiently small value of $\epsilon$. 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 $\epsilon_T$ is a sufficiently small value. Let us call the optimization to solve the problem in (4.20) for $\epsilon = \epsilon_t$ the $t$th sub-optimization. The $t$th sub-optimization for $t = 1, 2, \ldots, T$ is carried out using the procedure as described below.

First, the initializer, say, $x_0$, is set as

$$x_0 = \begin{cases} 
0 & \text{if } t = 1 \\
x_s & \text{if } t = 2, 3, \ldots, T
\end{cases}$$ (4.22)
where vector $x_s$ for the $t$th sub-optimization is the solution obtained from the $(t - 1)$th sub-optimization. The weight vector, say, $w$, for the $t$th sub-optimization is constructed by computing its $i$th component as

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

for $i = 1, 2, \ldots, N/d$, where $\tilde{x}_{0i}$ is the $i$th block of the initializer $x_0$. In (4.23), $\delta$ is a small positive scalar which helps to keep $w_i$ to a finite value when $||\tilde{x}_{0i}||_2$ is very small or zero.

Next, the $t$th 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 $\{g_k\}$ used in (4.12) are computed using (4.8) where vector $u$ 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

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

where $\gamma_i$ 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 $x$ was
### Table 4.2: $\ell_{2/p}$-RLS-WT Algorithm

| Step 1 | Input: $p$, $T$, $\epsilon_1$, $\epsilon_T$, $\Phi$, $y$, $E_t$,
|        | $\lambda$ if measurement $y$ is noisy, and
|        | $\lambda_1$ and $\lambda_T$ if measurement $y$ is noiseless.
|        | Set $x_s = 0$. |
| Step 2 | Compute $\epsilon_t$ for $t = 2, 3, \ldots, T - 1$ using (4.17) and
|        | $\lambda_t$ for $t = 2, 3, \ldots, T - 1$ using (4.18) if measurement $y$ 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 $k = 0$, $x_0 = x_s$, $E_r = 10^{10}$.
|        | iv) Compute $w$ using (4.23).
|        | v) Repeat the following while $E_r > E_t$,
|        | a) Compute $g_k$ using (4.8) in conjunction with (4.21).
|        | b) Compute $d_k$ using (4.12).
|        | c) Compute $\alpha_k$ using (4.16) in conjunction with (4.24).
|        | d) Compute $x_{k+1}$ using (4.11).
|        | e) Set $k = k + 1$.
|        | f) Exit loop if $k > L_t$.
|        | g) Compute $E_r = ||\alpha_k d_k||^2$.
|        | vi) Set $x_s = x_k$. |
| Step 4 | Output $x^* = x_s$ 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 $\Phi$ 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 $\Phi$ were made orthonormal relative to each other. The noiseless measurement was obtained as $y = \Phi x$. 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 and $\ell_{2/p}$-RLS-WT algorithms were applied and compared. A relative error was measured at the end of each sub-optimization as $||x_s - x_0||_2 / ||x_0||_2$ where $x_0$ and $x_s$ 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 $\mathbf{x}$ and a measurement matrix $\mathbf{Φ}$ were constructed, and measurement $\mathbf{y}$ was taken as in the first experiment. The $\ell_{2/p}$-RLS and $\ell_{2/p}$-RLS-WT algorithms were used to reconstruct signal $\mathbf{x}$ from $\mathbf{y}$ for all the values of $N$. The amount of computation required by the two algorithms was measured using MATLAB function $\text{cputime}$ 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 $\mathbf{x}$ and a mea-
Table 4.3: Relative error due to the $\ell_{2/p}$-RLS-WT and $\ell_{2/p}$-RLS algorithms

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

Measurement matrix $\Phi$ were constructed and measurement $y$ was taken as in the first experiment. Both the $\ell_{2/p}$-RLS-WT and $\ell_{2/p}$-RLS algorithms were used to reconstruct $x$ from $y$. Reconstruction was deemed successful if the maximum absolute error between the original signal $x$ and the recovered signal $\hat{x}$ 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 $i$th components of $x$ and $\hat{x}$, 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 $S \subset \{1, 2, \ldots, N/d\}$ be the set of indices of blocks of $x$ where the coefficients are nonzero. Assume that partial information about the locations of the nonzero blocks of $x$ is available, i.e., a subset $\hat{S}$ of $S$ is known a priori. In such a case, the minimization of function $||x||_{2/p, \epsilon}^p$ in (4.3) can be reduced to the minimization of function $||x||_{p, \epsilon, s}^p$ where

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

(4.25)

Under this circumstance, we propose to reconstruct signal $x$ by solving the problem

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

(4.26)

The objective function $F_{p, \epsilon, s}(x)$ 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 $F_{p, \epsilon, s}(x)$ is given by

$$g = \Phi^T (\Phi x - y) + \lambda g_s$$

(4.27)

where the $((i-1)d + j)$th component of vector $g_s$ lying in its $i$th block is given by

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

(4.28)

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

As can be noticed, the role of parameter $\epsilon$ in the objective function $F_{p, \epsilon, s}(x)$ in (4.26) is similar to that in function $F_{2/p, \epsilon}(x)$ 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 $\alpha_k$ in (4.11) is computed using a line search based in Banach’s fixed-point theorem which solves the optimization problem

$$\minimize_{\alpha} F_{p,\epsilon,s}(x_k + \alpha d_k)$$

(4.29)

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

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

where

$$G(\alpha) = -\frac{d_k^T (\Phi x_k - y) + \lambda p \sum_{i \notin S} \gamma_i \cdot (\tilde{x}_{ki}^T \tilde{d}_{ki})}{||\Phi d_k||^2 + \lambda p \sum_{i \notin S} \gamma_i \cdot (\tilde{d}_{ki}^T \tilde{d}_{ki})}$$

(4.30)

Parameter $\gamma_i$ is given in (4.15) and $\tilde{x}_{ki}$ and $\tilde{d}_{ki}$ are the $i$th blocks of $x_k$ and $d_k$, 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 $\ell_p$-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 $g_k$ in Step 3.iv.a in Table 4.1 is computed using (4.27) and b) the step size $\alpha_k$ 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 $\ell_p$-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 $x$ 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 $\Phi$ 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 $\Phi$ are made orthonormal to each other. The noiseless measurement was taken as $y = \Phi x$. 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 $\ell_p$-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 maximum $|x_i - \hat{x}_i|$ was less than 0.09, where $x_i$ and $\hat{x}_i$ are the $i$th components of the original signal $x$ and the reconstructed signal $\hat{x}$, 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 $p$th 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 $\ell_0$ (NRAL0), unconstrained approximate $\ell_p$ (UALP), and unconstrained approximate $\ell_p$ conjugate-gradient (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 $\ell_0$ pseudonorm whereas the UALP and UALP-CG algorithms are based on the minimization of an approximate $\ell_p$ 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
rewighted (IR), smoothed $\ell_0$ (SL0), and $\ell_1$-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 $\ell_1$ 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 $\ell_1$-minimization based algorithms.

In Chapter 3, two algorithms, namely, the $\ell_p$-regularized least-squares ($\ell_p$-RLS) and $\ell_p$-regularized least-squares conjugate-gradient ($\ell_p$-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 $\ell_p$-RLS and $\ell_p$-RLS-CG algorithms are based on minimizing an $\ell_p$-regularized $\ell_2$ error where the involved optimization problem is unconstrained. The $\ell_p$-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 $\ell_p$-RLS-CG algorithm solves the same problem in the time domain by using a sequential CG algorithm. A method for applying the $\ell_p$-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 $\ell_2$ error. The minimization is carried out by using a sequential Fletcher-Reeves’ CG technique. Improvement in the performances of the proposed $\ell_p$-RLS, $\ell_p$-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 $\ell_p$-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 $\ell_2$ 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 \( \ell_p \)-optimization based algorithms presented offer improved performance relative to \( \ell_1 \)-optimization based algorithms. However, indepth theoretical studies of the proposed algorithms, especially in terms of convergence properties and the relation of convergence to parameters \( p, \epsilon, \) and \( \lambda \), are not available. Studies along these lines would provide more insights on \( \ell_p \)-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 \( \ell_1 \)-optimization based algorithm.
(b) In biomedical signal processing, electrocardiogram (ECG) and electroencephalogram (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 $\ell_0$ pseudonorm or soft thresholding based on minimizing the $\ell_1$ norm. It would be interesting to investigate iterative shrinkage based methods using $\ell_p$-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


Appendix A

Singular-Value and QR Decompositions

A.1 Singular-Value Decomposition

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

$$\Phi = U S V^T$$  \hspace{1cm} (A.1)

where $U \in \mathbb{R}^{M \times M}$ and $V \in \mathbb{R}^{N \times N}$ are orthogonal matrices,

$$S = [\Sigma \ 0]$$

with

$$\Sigma = \text{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_M\}$$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_M > 0,$$

In the above equations, $0$ denotes the zero matrix of size $M \times (N - M)$, $\sigma_1, \sigma_2, \ldots, \sigma_M$ are the nonzero singular values of $\Phi$, and the columns of $U$ are the left singular vectors of $\Phi$ corresponding to the singular values \cite{89} \cite{48}. The matrix $V$ in (A.1) can be expressed as $V = [V_r \ V_n]$ where $V_n$ is composed of the last $N - M$ columns of $V$ which span the null space of $\Phi$, while $V_r$ is composed of first $M$ columns of $V$, 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 $\Phi$. 
Consider equation $\Phi x = y$ with $M < N$, which is an underdetermined system of linear equations. All the solutions of $\Phi x = y$ can be parameterized as

$$x = x_s + V_n \xi$$  \hspace{1cm} (A.2)

where $x_s$ is a special solution of $\Phi x = y$. In many applications, the least-squares solution is used as $x_s$ which can be computed as

$$x_{\ell_2} = \Phi^\dagger y$$

where $\Phi^\dagger$ is the Moore-Penrose pseudoinverse of $\Phi$ given by

$$\Phi^\dagger = V \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & \end{bmatrix} U^T$$

with $\Sigma^{-1} = \text{diag}\{1/\sigma_1, 1/\sigma_2, \ldots, 1/\sigma_M\}$.

### A.2 QR Decomposition

Given a matrix $\Phi \in \mathbb{R}^{M \times N}$ with $M < N$, the QR decomposition of $\Phi^T$ is given by

$$\Phi^T = QR$$  \hspace{1cm} (A.3)

where $Q \in \mathbb{R}^{N \times N}$ is an orthogonal matrix and

$$R = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}$$

with $\hat{R}$ an upper triangular square matrix [89] [48].

If we express matrix $Q$ as $Q = [Q_1 \; Q_2]$ where $Q_1 \in \mathbb{R}^{N \times M}$ and $Q_2 \in \mathbb{R}^{N \times (N-M)}$, it can be shown that the columns of $Q_2$ and $Q_1$ span the null space of $\Phi$ and its orthogonal complement space, respectively. The QR decomposition can be used to efficiently compute the least-squares solution $x_{\ell_2}$ of $\Phi x = y$ and the basis of the null space $V_n$ of $\Phi$ as

$$x_{\ell_2} = Q_1 \hat{R}^{-T} y, \quad V_n = Q_2,$$  \hspace{1cm} (A.4)
respectively.
Appendix B

Derivation of Eqs. (3.14)-(3.16)

Suppose the vectors $\phi$ and $\xi$ in (3.6) are fixed and that $e_i$ is the $i$th column of the identity matrix of size $M \times M$. At point $\phi$, a line search along direction $e_i$ is carried out by solving the one-dimensional optimization problem

$$\text{minimize } F_{p,\epsilon}(\phi + \delta e_i, \xi)$$

(B.1)

where

$$F_{p,\epsilon}(\phi + \delta e_i, \xi) = \frac{1}{2} ||\Sigma (\phi + \delta e_i) - \tilde{y}||_2^2 + \lambda ||x + \delta v||_p$$

$$= \frac{1}{2} [\sigma_i (\phi_i + \delta) - \tilde{y}_i]^2$$

$$+ \lambda \sum_{j=1}^{N} [(x_j + \delta v_{ij})^2 + \epsilon^2]^{p/2},$$

(B.2)

$v_i$ is the $i$th column of matrix $V_r$, and $x_j$ and $v_{ij}$ are the $j$th components of vectors $x$ and $v_i$, respectively. By equating the derivative $dF_{p,\epsilon}(\phi + \delta e_i, \xi)/d\delta$ to zero, we obtain

$$\delta = \frac{-\sigma_i \tilde{y}_i + \sigma_i^2 \phi_i + \lambda p \sum_{j=1}^{n} \gamma_j(\epsilon) x_j v_{ij}}{\sigma_i^2 + \lambda p \sum_{j=1}^{N} \gamma_j(\epsilon) v_{ij}^2}$$

(B.3)

where

$$\gamma_j(\epsilon) = [(x_j + \delta v_{ij})^2 + \epsilon^2]^{p/2 - 1}$$

(B.4)
Note that (B.3) can be solved for $\delta$ 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 $\delta$. 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 $\xi$ 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 $\sigma_i^2$ in the denominator of (B.3) can be removed. Consequently, (3.14b) can be obtained using (B.3).