Estimation of Sparse Distributions

Md. Mashud Hyder
M.Sc. (Computer Science and Engineering)

A thesis submitted in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy

School of Electrical Engineering
and Computer Science

THE UNIVERSITY OF NEWCASTLE
AUSTRALIA

January, 2012
Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying subject to the provisions of the Copyright Act 1968.

I hereby certify that the work embodied in this thesis contains published papers/scholarly work of which I am a joint author. I have included as part of the thesis a written statement, endorsed by my supervisor, attesting to my contribution to the joint publications/scholarly work.

Md. Mashud Hyder
January, 2012

Kaushik Mahata
January, 2012
Acknowledgments

I performed the thesis work under the supervision of Dr. Kaushik Mahata. I am grateful to him for his constant guidance, helpful suggestions and valuable assistance. He helped me by giving his invaluable time throughout the thesis work. His encouragement and motivations helped me to overcome all difficulties.

I am profoundly grateful to my family, especially my wife, who has always expressed me her unconditional support and love.
# Contents

Acknowledgments ................................................................. v
Table of Contents ............................................................. vii
List of Figures ................................................................. xii
List of Tables ................................................................. xx
Abstract .................................................................................. xxi

1 Introduction ........................................................................... 1
   1.1 Sparse Distribution ......................................................... 1
   1.2 Decoding via Sparsity ...................................................... 2
   1.3 Illustrative Examples ....................................................... 3
   1.4 Thesis Outline ............................................................... 7
   1.5 Contributions ............................................................... 12
   1.6 Topics for Future Research ............................................. 13

1 Sparse Signal Recovery Algorithms ...................................... 15

2 Single Measurement Vector Problem .................................. 17
   2.1 Introduction ............................................................... 17
      2.1.1 Orthogonal Matching Pursuit .................................. 18
      2.1.2 Basis Pursuit ........................................................ 19
      2.1.3 FOCUSS .............................................................. 20
      2.1.4 Contributions and Motivation .................................. 22
   2.2 Smoothed ℓ₀ (SL₀) cost function and its minimization ....... 23
      2.2.1 Analysis .............................................................. 24
2.2.2 Optimization strategies ........................................ 30
2.3 Experiments ......................................................... 35
  2.3.1 Signal recovery from noise-free measurements ............... 35
  2.3.2 Recovery of power-law decaying signals ..................... 38
  2.3.3 Approximate Signal Recovery in Noisy Environment .......... 41
  2.3.4 Application of ISL0 in single pixel imaging ................. 46
2.4 Conclusion ......................................................... 46

3 Structured Sparse Signal Recovery 50
  3.1 Introduction ...................................................... 50
    3.1.1 Contributions and Motivation ................................ 51
  3.2 Multiple Measurement Vectors Problem ......................... 51
    3.2.1 Background .................................................. 51
    3.2.2 Joint $\ell_{2,0}$ Approximation Algorithm (JLZA) .......... 53
  3.3 Block Sparse Signals .......................................... 60
    3.3.1 Background .................................................. 60
    3.3.2 Block ISL0 .................................................. 61
  3.4 ISL0 for Complex Valued Signals ................................ 62
    3.4.1 Complex Valued Multiple Measurement Vectors Problem .... 62
  3.5 Experiments ...................................................... 63
    3.5.1 Simulations with Multiple Measurement Vectors ............ 64
    3.5.2 Block sparse problem .................................... 69
3.6 Conclusion ......................................................... 71

4 Sparse Signal Recovery with Prior Information 76
  4.1 Introduction ...................................................... 76
    4.1.1 Contributions and Motivation ................................ 77
  4.2 Problem definition and related work .......................... 77
  4.3 Maximum a posteriori estimation (MAP) ......................... 79
    4.3.1 MAP framework .............................................. 79
    4.3.2 Solution strategy ........................................... 81
    4.3.3 Minimizing $\varphi_j$ ...................................... 82
    4.3.4 Numerical algorithm for solving MAP optimization ....... 83
CONTENTS

4.4 PMAP in noisy environment ........................................ 84
4.5 Experimental results .................................................. 86
  4.5.1 Simulation using artificial data set .............................. 86
  4.5.2 Simulations using real data ....................................... 91
4.6 Conclusion ............................................................ 95

II Applications .......................................................... 99

5 Direction-of-Arrival Estimation ....................................... 101
  5.1 Introduction .......................................................... 101
    5.1.1 Contributions ................................................... 102
  5.2 Problem formulation .................................................. 102
    5.2.1 DOA Estimation as a Joint-Sparse Recovery Problem ...... 103
    5.2.2 Results on the Joint-Sparse Recovery Problem ............ 104
  5.3 $\ell_1$-SVD ........................................................... 105
  5.4 DOA estimation using JLZA .......................................... 106
    5.4.1 Narrowband DOA estimation .................................. 106
    5.4.2 Broadband DOA estimation ................................... 107
  5.5 Simulation Results .................................................... 108
    5.5.1 Narrowband signal .............................................. 108
    5.5.2 Broadband signal ............................................... 117
  5.6 Conclusion ........................................................... 118

6 Spectrum Analysis from Irregularly Sampled Data ................. 123
  6.1 Introduction .......................................................... 123
    6.1.1 Contributions ................................................... 124
  6.2 Spectrum Analysis of Single Signal ................................ 124
    6.2.1 Sparse Representation ......................................... 124
    6.2.2 SparSpec ........................................................ 126
    6.2.3 An approach Based on $\ell_0$ Approximation ............... 127
  6.3 Coherent Spectrum Analysis ......................................... 129
    6.3.1 Background ...................................................... 129
6.3.2 Joint Sparse Model .............................................. 129
6.3.3 Estimation of $X$ .................................................. 131
6.4 Simulation Results ............................................... 133
  6.4.1 Spectrum Estimation of Single Signal ...................... 133
  6.4.2 Spectrum Estimation of Coherent Signals ................... 136
6.5 Conclusion .......................................................... 141

7 Target Detection using MIMO Radar System 145
  7.1 Introduction ...................................................... 145
  7.2 Problem Formulation ............................................. 146
  7.3 Joint Sparse Representation .................................... 148
  7.4 Target Detection using Capon Based Estimator ............... 149
  7.5 Target Detection using JLZA .................................... 149
  7.6 Simulation Results .............................................. 149

8 Range-Doppler Imaging 154
  8.1 Introduction ...................................................... 154
    8.1.1 Contributions and Motivation ............................. 155
  8.2 Problem Formulation ............................................. 155
  8.3 Reconstruction Approaches .................................... 160
    8.3.1 $\ell_1$ Optimization Approach ............................. 160
    8.3.2 $\ell_0$ Minimization Approach ............................. 161
  8.4 Simulation Results .............................................. 162
    8.4.1 Effect of Noise ............................................. 163
    8.4.2 Large Number of Targets ................................... 168
    8.4.3 Performance in Presence of Land Clutter ............... 169
  8.5 Conclusion ........................................................ 172

9 Target Detection in MTI Radar 173
  9.1 Introduction ...................................................... 173
    9.1.1 Background .................................................. 173
    9.1.2 Contributions and Motivation ............................. 175
  9.2 Problem Formulation ............................................. 176
## CONTENTS

9.3 Proposed Reconstruction Approach ........................................ 177  
  9.3.1 Step 1 - Approximate Localization of the Clutter Spread ........ 177  
  9.3.2 Step 2 - Target Detection using Sparse Recovery Algorithm ...... 180  
  9.3.3 Steering Vector Error ..................................................... 182  
9.4 Simulation Results ............................................................ 184  
9.5 Conclusions ........................................................................ 196  

A .................................................................................. 198  
  A.1 Proof of the moment results in Section 2.2.1.3 ....................... 198  
  A.2 Proof of Lemma 8 ............................................................... 200  

Bibliography ......................................................................... 202
List of Figures

1.1 Uniform linear array. .................................................. 3

1.2 Sparse representation of image. (a) Original $256 \times 256$ image, (b) Wavelet transform coefficients (arranged in random order for enhanced visibility), (c) Reconstructed images from $x$, where $k = 6560$. ........................................ 7

1.3 (a) Cardiac MR image sequence. (b) Cardiac MR images are sparsified using 2-D discrete wavelet transform. Wavelet coefficient vectors $x$ of different frames. .................................................. 8

2.1 The minimum number of measurement $m$ necessary to recover a $k$-sparse signal in dimension $n = 1000$. For each $k$, 100 different $k$-sparse signals are constructed. Then, the smallest $m$ is selected that can recover at least 95 signals. .................................................. 36

2.2 The percentage of 100 input signals correctly recovered as a function of the sparsity level $k$ for fixed measurements $m = 400$ in signal dimension $n = 1000$. 37

2.3 Computation time in seconds (averaged over 100 runs) versus sparsity ($k$). The number of measurements $m = 400$ in dimension $n = 1000$. ........ 38

2.4 The percentage of 100 input signals correctly recovered as a function of the sparsity level $k$ for fixed measurements $m = 350$ in signal $x$ of dimension $n = 1000$. The coefficients of nonzero $x$’s follow power-law. (a) $p = 2$, (b) $p = 3$. .................................................. 39

2.5 Average SNR (from 100 runs) of the signals recovered by ISL0, ASA-SL0 and O-SL0 as a function of $\rho$, where $m = 400$, and $n = 1000$. Results for different values of $k$ and $\mu$ are shown. (a) Average SNR as function of $\rho$ for $\mu = 0.02$, $\sigma_{\min} = 0.04$, (b) Average SNR as function of $\rho$ for $\mu = 0.05$. $\sigma_{\min} = 0.1$. .. 40
2.6 Average SNR (from 100 runs) of the signals recovered by ISL0 as a function of $\eta$, where $m = 400$, and $n = 1000$. Results for different values of $k$ and $\mu$ are shown. (a) Average SNR as a function of $\eta$ for $\mu = 0.01$. $\sigma_{\text{min}} = 0.01$, (b) Average SNR as a function of $\eta$ for $k = 100$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

2.7 The experimental RMSE of the ISL0 iterate $x^{(i)}$ and the its theoretical upper bound using (2.22), with $n = 1000$, $k = 100$, and $m = 400$. (a) RMSE at successive iterations for $\mu = 0.01$, (b) RMSE at successive iterations for $\mu = 0.05$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

2.8 The utility of the stopping criterion developed in (Sections 2.2.1.3 and 2.2.1.4), where $k = 100$, $m = 400$, $n = 1000$. (a) 96% confidence bound $a_\sigma$ on $h_\sigma$ versus $\sigma_{\text{min}}$, (b) Average SNR (on 100 runs) versus $\sigma_{\text{min}}$. . . . . . . . . . . 45

2.9 Average SNR’s (over 100 runs) versus measurements ($m$) with $n = 1000$, $k = 100$. (a) Average SNR versus $m$ for fixed $\mu = 0.02$, (b) Average SNR (on 100 runs) versus $m$ for fixed $\mu = 0.05$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

2.10 Single-pixel photo album. For ISL0, parameters are fixed at $\rho = 0.5$, $\eta = 0.5$. (a) Target image of a ‘R’, (b) ISL0 reconstruction from 50% random measurements, (c) Computation time comparison. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

3.1 The recovery rates (averaged over 100 trials) for randomly generated $X_0 \in \mathbb{R}^{128 \times 10}$ with different sparsity $k$. The value of $m$ is fixed to 50 and $r = 10$. 65

3.2 Average recovery time of various MMV techniques. The value of $n = 128$ and $m = 50$. (a) Average recovery time (on 100 runs) versus $r$ for fixed $k = 10$, (b) Average recovery time (on 100 runs) versus $k$ for different MMV techniques. $r = 10$. . . . . . . . . . . . . . . . . . . . . . . 66

3.3 Average SNR achieved by various MMV techniques. The value of $n = 128$ and $m = 50$. (a) Average SNR (on 100 runs) versus $k$ for fixed $r = 10$ and $\mu = 0.03$, (b) Average SNR (on 100 runs) versus $\mu$ for fixed $k = 20$ and $r = 10$. 68

3.4 Average SNR (on 100 runs) versus $r$ for MMV problem. $k = 15$, $\mu = 0.03$, $n = 128$ and $m = 50$. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
3.5 Recovery performance of algorithms for MMV with multiple measurement matrix. The value of $m$ is fixed to 50. (a) Signal recovery from noiseless data. The recovery rates for randomly generated $X_j$ with different sparsity $k$. (b) Signal recovery from noisy data. Average SNR versus $k$ for $\mu = 0.03$. 70

3.6 The recovery rates (averaged over 100 trials) for randomly generated $x \in \mathbb{R}^{128}$ with different block sparsity $k$. The value of $m$ is fixed to 50. (a) Block size $b = 4$. (b) Block size $b = 8$. 72

3.7 The recovery rates (averaged over 100 trials) for randomly generated $x \in \mathbb{R}^{128}$ with different values of $m$. (a) Block size $b = 4$, sparsity $k = 8$. (b) Block size $b = 8$, sparsity $k = 4$. 73

3.8 Average SNR achieved by various techniques. The value of $n = 128$, $b = 4$ and $m = 50$. (a) Average SNR (on 100 runs) versus $k$ for $\mu = 0.03$, (b) Average SNR (on 100 runs) versus $k$ for $\mu = 0.05$. 74

4.1 (a) Larynx MRI video sequence. (b) Larynx MRI images are sparsified using 2-D discrete wavelet transform. Then we calculate the support change of the MRI sequence. 78

4.2 Dependency of PMAP on different tuning factors. (a) - (b) dependency on $s$, i.e., error in known part of support. We take different $\hat{s}$ around the actual $s$ and compare performance. We fixed $w = 20$ and $\hat{s} = k - g + s$. (c) - (d) Average NRMSE (over 100 runs) vs $w$, keeping $\hat{s} = \lceil 0.2g \rceil$. In all experiments $m = \lceil 0.25n \rceil$, $k = \lceil 0.1n \rceil$. 88

4.3 Noiseless signal recovery as a function of measurements $m$. Performance of different algorithms evaluated for different values of $g$ and $s$. Average NRMSE (over 100 runs) vs $m$ keeping (a) $g = \lceil 0.55k \rceil$, $s = \lceil 0.1k \rceil$, (b) $g = \lceil 0.75k \rceil$, $s = \lceil 0.1k \rceil$, (c) $g = \lceil 0.55k \rceil$, $s = \lceil 0.25k \rceil$, (d) $g = \lceil 0.75k \rceil$, $s = \lceil 0.25k \rceil$, (e) $g = \lceil 0.55k \rceil$, $s = \lceil 0.4k \rceil$, (f) $g = \lceil 0.75k \rceil$, $s = \lceil 0.4k \rceil$. In all experiments $k = \lceil 0.1n \rceil$. 89

4.4 Noiseless signal recovery as a function of sparsity $k$, where $m = \lceil 0.25n \rceil$, $s = \lceil 0.25k \rceil$. 90
4.5 Noisy signal recovery as a function of measurements $m$. Performance of different algorithms evaluated for different values of $g$ and $\mu$. Average NRMSE (over 100 runs) vs $m$ keeping (a) $g = [0.55k], \mu = 1$, (b) $g = [0.75k], \mu = 1$, (c) $g = [0.55k], \mu = 3$, (d) $g = [0.75k], \mu = 3$, (e) $g = [0.55k], \mu = 5$, (a) $g = [0.75k], \mu = 4$. In all experiments $k = 0.1n$, $s = 0.25k$.

4.6 MRI images reconstruction by different algorithms. $g = [0.75k], s = [0.25k], \hat{\mu} = 0.001$. (a) Actual $128 \times 128$ sparsified Larynx image and its reconstruction by different algorithms when $m = [0.22n]$. (b) Actual $128 \times 128$ sparsified Cardiac image and its reconstruction by different algorithms when $m = [0.24n]$.

4.7 Reconstructing $128 \times 128$ MRI images. $g = [0.75k], s = [0.25k]$. (a) NRMSE vs $\frac{m}{n}$ for larynx image, (b) NRMSE vs $\frac{m}{n}$ for Cardiac image.

4.8 Reconstruction of $128 \times 128$ larynx image sequences. (a) Actual sparse sequences, (b) CS reconstruction (c) Modified-CS reconstruction (d) PMAP reconstruction. $m = [0.22n], g = [0.5k], \hat{\mu} = 0.01$.

4.9 Reconstruction of $128 \times 128$ MRI sequences. $g = [0.5k], \hat{\mu} = 0.01$. (a) Larynx sequences reconstruction performances of different algorithms. (b) Cardiac sequences reconstruction performances of different algorithm.

5.1 Resolving sources. DOA grid resolution $1^\circ$. (a) Uncorrelated sources at $10^\circ$ and $15^\circ$, SNR 5 dB, $r = 50$. (b) Correlated sources at $10^\circ$ and $20^\circ$, SNR 10 dB, $r = 50$.

5.2 Source estimation when DOA does not fall on the grid. Uncorrelated sources at $10^\circ$ and $15.3^\circ$, SNR 10 dB, $r = 50$. (a) DOA grid resolution $1^\circ$, (b) DOA grid resolution $0.5^\circ$.

5.3 DOA estimation with erroneous information about sensor locations. Uncorrelated sources at $10^\circ$ and $15^\circ$, SNR 10 dB, $r = 50$, grid resolution = $1^\circ$. (a) $\hat{\mu} = 0.02$. (b) $\hat{\mu} = 0.05$.

5.4 Maximum number of uncorrelated DOA estimation capability by different algorithms, $r = 100$, SNR = 10 dB, grid resolution = $1^\circ$. (a) Resolving uncorrelated 7 sources by using 8 sensors, (b) Resolving uncorrelated 8 sources by using 8 sensors.
5.5 DOA estimation. Grid resolution = 1°. (a) Resolving correlated 6 sources by using 8 sensors. SNR 10 dB, $r = 100$. (b) An example where JLZA fails to resolve DOA. Uncorrelated sources at 10° and 11°, SNR 5 dB, $r = 50$.

5.6 Frequency of detection of correlated sources. (a) Against number of snapshots, SNR = 10 dB. (b) Against input SNR, $r = 100$.

5.7 Broadband DOA estimation results for three sources at $-20°, -10°, 20°$ using an 8-sensor ULA, $r = 100$ snapshots, SNR = 20 dB. Sensor spacing is the half-wavelength associated with 2.5 GHz.

5.8 Use of JLZA-MMM in the presence of spatial aliasing with three sources located at $-20°, -15°, 20°$. The sensor spacing equals 2.5 times the smallest wavelength in the broadband signal.

5.9 Resolution of JLZA-MMM in localizing two closely spaced sources as a function of sensor spacing. Half wavelength spacing is 6 cm corresponding to 2.5 GHz. SNR=20dB and $r = 100$.

5.10 Use of JLZA-MMM in the presence of spatial aliasing with three sources located at $-20°, -15°, 20°$. The sensor spacing equals 5 times the smallest wavelength in the broadband signal.

6.1 Frequency spectrums by different algorithms. (a) Fourier spectrum. (b) IRLS. (c) $\ell_1$ (d) EISL0. Actual frequencies $[0.0027, 0.0033, 0.0082, 0.01] c/d$. Number of time samples $m = 120$, SNR = 15dB.

6.2 (a) Probability of recovering actual frequencies as a function of number of time samples ($m$) for fixed SNR= 15 dB. (b) Minimum number of samples needed to recover actual frequencies as a function of SNR. Actual frequencies $[0.0027, 0.0033, 0.0082, 0.01] c/d$.

6.3 Frequency bias as a function of SNR. Actual frequencies $[0.0027, 0.0033, 0.0082, 0.01] c/d$, $m = 150$.

6.4 A typical frequency spectrum. $m = 50$, SNR=5 dB and $k = 14$.

6.5 Resolution performance of different algorithms for $m = 50$, SNR = 5 dB, $k = 10$. (a) Probability of detecting the frequencies as a function of $|\hat{\Omega}_1 - \hat{\Omega}_2|/(2\pi)$ with $R=199$. (b) $|\hat{\Omega}_1 - \hat{\Omega}_2|/(2\pi)$ as a function of $R$. 

6.6 A typical frequency spectrum of SIAA. (a): $R = 199$ (b): $R = 399$. $m = 50$, $k = 10$, SNR=5 dB. ................................. 140
6.7 Empirical probability of detecting frequencies as a function of $k$. SNR = 5 dB. 142
6.8 Maximum number of frequency components is detected by different algorithms as a function of SNR. $m = 50$, $R=199$. ................................. 143
6.9 Frequency recovery time by different algorithms. (a) SIAA-MSC: E, L=0.6M, (b) SIAA-MSC: E, L=0.8M, (c) JLZA-MMM: E, $r=2$, (d) $\ell_{2,1}, r = 2$. $R = 199$, SNR=5 dB. ................................. 144
7.1 Bistatic MIMO radar system. ................................. 146
7.2 Estimating 8 targets. Targets are located at $\{14, -22\}, \{8, 10\}, \{16, 14\}, \{18, 16\}, \{28, -18\}, \{-10, -16\}, \{-30, 26\}, \{-16, 10\}$ degrees. $r = 10$, SNR=[1, $-5, 5, -1, 3, 5, 10, 5$] dB. (a) True locations (b) JLZA (c) 2D-Capon (d) PARAFAC. ................................. 151
7.3 Estimating 8 targets. Targets are located at $\{14, -22\}, \{8, 10\}, \{16, 14\}, \{18, 16\}, \{28, -18\}, \{-10, -16\}, \{-30, 26\}, \{-16, 10\}$ degrees. (a) Empirical probability of detecting targets as a function of $r$, SNR of all targets 2 dB (b) Empirical probability of detecting targets as a function of SNR, $r = 50$. ................................. 152
7.4 Angle bias as a function of SNR. Target arrangements are similar to Figure 7.2, $r = 50$. ................................. 153
8.1 A pulse train. ................................. 156
8.2 Range-Doppler images for target setting as in Table 8.2. $N = 13$ and $Q = 16$. (a) MF. (b) IAA with IV filter as the initial estimator. (c) 2D-SL0. (d) $\ell_1$. ................................. 164
8.3 Power distribution over Doppler at range bin 15. There are two targets with Doppler shifts $-210$ Hz and $110$ Hz. (a) Doppler resolution is $25$ Hz. The Doppler shifts of two targets are detected at $-200$ Hz and $100$ Hz . (b) Doppler resolution is $12.5$ Hz. The Doppler shifts of two targets are detected at $-212.5$ Hz and $112.5$ Hz . (c)Doppler resolution is $6.25$ Hz. The Doppler shifts of two targets are detected at $-212.5$ Hz and $112.5$ Hz . ................................. 166
8.4 Performance comparison of different algorithms. (a) Peak-to-Ripple ratio (PRR) of different algorithms plotted as functions of SNR. (b) Time Comparison for the setting in Table 8.2. $N = 13$ and $Q = 16$. ................................. 167
8.5 Recovery performance of $\beta$ in (9.1) ................................. 168
8.6 Receiver operating characteristic (ROC) for different algorithms. (a) 2D-SL0, (b) $\ell_1$, (c) IAA. ................................................................. 169

8.7 Range-Doppler imaging with many targets. Target setting is given in Table 8.3. $N = 13$ and $Q = 16$. (a) MF. (b) IAA with IV filter as the initial estimator. (C) 2D-SL0. (d) $\ell_1$. ................................................................. 170

8.8 Range-Doppler imaging in presence of range-spread clutter. $N = 13$ and $Q = 16$. (a) MF. (b) IAA with IV filter as the initial estimator. (C) 2D-SL0. (d) $\ell_1$. ................................................................. 171

9.1 Clutter detection. (a) True clutter, (b) IRLS, (c) Normalize $h(\delta)$ vs $\delta$ of the angle-Doppler image by different algorithms. ................................. 180

9.2 Angle-Doppler images. SNRs of two targets are 2 dB and 2 dB. SCNR=$-45.12$ dB. (a)CS recovery, (b) Modified-CS, (c) IAA, (d) PMAP. ......................... 187

9.3 Angle-Doppler images of clutter-and-noise only. (a) true angle-Doppler image, (b)CS, (c) IAA, (d) PMAP. ...................................................... 188

9.4 Resolution performance of algorithms. Average SCNR = $-65$ dB. (a) Doppler resolution, (b) Angle resolution, (c) Clutter resolution. ...................... 190

9.5 Angle-Doppler images. SNRs of two targets are 2 dB and 2 dB. SCNR=$-66.12$ dB. Azimuth angles of the targets are $-42^\circ$ and $21^\circ$. Normalize Doppler shifts are $-0.015$ and $-0.035$. (a) true image, (b) CS, (c) IAA, (d) PMAP. ........ 191

9.6 Robustness of algorithms. (a) Robustness against noise power. SNRs of two targets are 2 dB and 2 dB. Azimuth angles of the targets are $\{-42^\circ, 24^\circ\}$ and Doppler shifts are $\{-0.015, -0.035\}$. (b) Robustness against clutter power. SNRs of two targets are 2 dB, 2 dB and $\eta = 0.5$. Azimuth angles of the targets are $\{-42^\circ, 24^\circ\}$ and Doppler shifts are $\{-0.015, -0.035\}$. (c) Multiple targets detection. ....................................................... 193

9.7 Angle-Doppler images. SNRs of two targets are 2 dB and 2 dB. SCNR=$-58.22$ dB. Azimuth angles of the targets are $\{-42^\circ, 21^\circ\}$ and Doppler shifts are $\{-0.015, -0.035\}$. (a) True image, (b)CS, (c) IAA, (d) PMAP. .............. 194
9.8 Angle-Doppler images in presence of steering vector error. SNRs of two targets are 2 dB and 2 dB. SCNR = −64.8 dB. Azimuth angles of the targets are \{-42^\circ, 24^\circ\} and Doppler shifts are \{-0.015, -0.035\}. (a) IAA with steering vector error. (b) PMAP with steering vector error. (c) IAA after steering vector error correction. (d) PMAP after error correction. .......................... 195

9.9 Angle-Doppler images. SNRs of two targets are 2 dB. SCNR=−56.8 dB. Azimuth angles of the targets are \{-42^\circ, 24^\circ\} and Doppler shifts are \{-0.015, -0.035\}. (a) IAA with steering vector error, (b) IAA after steering vector error correction. ......................................................... 195

9.10 Frequency of detecting targets as a function of \(\nu\). Two targets with SNR =2 dB are placed at azimuth angle \{-42^\circ, 24^\circ\} and their Doppler shifts are \{-0.015, -0.0354\}. Average SCNR =−65 dB. .............................................. 196

9.11 Receiver operating characteristic (ROC) for PMAP. (a) Without steering vector error, (b) With steering vector error. ................................. 197
List of Tables

2.1 The OMP algorithm ................................................. 18
2.2 Minimization of the SL0 cost function: the basic idea. ............ 24
2.3 ASA-SL0 Algorithm ............................................. 30
2.4 ISL0 Algorithm .................................................. 34

3.1 JLZA Algorithm ..................................................... 55
3.2 Extension of JLZA Algorithm for Noisy Environment ................ 57
3.3 JLZA with Multiple Measurement Matrices (JLZA-MMM) .............. 59
3.4 The Block-OMP (BOMP) algorithm .................................. 61

4.1 MAP with partially known support (PMAP) .......................... 84

5.1 Computation time comparison. ...................................... 116

6.1 SparSpec algorithm for minimizing (6.6) ............................ 127
6.2 SIAA-MSC Algorithm [107] ....................................... 132

7.1 2-D Capon Based Estimator for Target Detection [115] ............... 149

8.1 Two Dimensional Robust SL0 (2D-SL0) Algorithm [128] .......... 162
8.2 Ground Truth 1 ................................................. 163
8.3 Ground Truth 2 .................................................. 170

9.1 Steering vector error approximation .................................. 185
Abstract

A vector is called sparse when most of its components are zero. Many natural signals admit sparse representations with respect to some bases. This dissertation deals with the problem of recovering a sparse signal from a small number of measurements formed by computing the inner product of the signal with the rows of a measurement matrix.

The task of recovering a sparse signal from its small number of measurements is a problem of finding the sparsest solution of an underdetermined system of linear equations. The problem can be solved by using a $\ell_0$ norm based optimization problem which is NP-hard in general. Some recent researches have demonstrated that the NP-hard problem can be solved tractably by using linear programming approach under some reasonable conditions. However, real-world applications often demand more efficient algorithms which are robust to measurement noise. To this aim, an efficient sparse signal recovery algorithm is developed in the first part of the thesis. The proposed algorithm uses a convex-concave procedure to optimize its cost function. A range of theoretical results are presented. The theoretical analysis of the algorithm gives a bound on the run-time estimation error. In many real world problems the resulting sparse signals exhibit additional structures. The proposed algorithm is then extended to exploit the structures of sparse signal. Experimental results demonstrated that in most settings the extended algorithm outperforms other conventional algorithms with a large margin. Finally an interesting sparse signal recovery approach is considered when a part of the support of the sparse signal is known in advance. A maximum a posteriori (MAP) estimation framework is considered to deal with the issue.

Second part of the thesis explores the applicability of the proposed algorithms for solving some practical problems. In some cases, the proposed algorithm exhibits additional advantages. For example, in the broadband direction of arrival (DOA) estimation problem, the proposed algorithm allows relaxing the half-wavelength sensor spacing restriction, which leads to a significant performance improvement. In some applications, the proposed algorithm can exploit underlying structure of the linear system. For example, in frequency estimation problem, it is possible to exploit the structure of the Fourier basis to achieve a significant reduction of the computational complexity.
Nomenclature

0 subscript index

\langle a, b \rangle \quad \text{Inner product of } a \text{ and } b

\otimes \quad \text{Kronecker product}

\| . \|_F \quad \text{Frobenious norm of a matrix}

A' \quad \text{Transpose of a matrix } A

A[i,i] \quad i\text{-th column of a matrix } A

A[i,j] \quad \text{The element of matrix } A \text{ at its } i\text{-th row and } j\text{-th column}

A^* \quad \text{Conjugate transpose of } A

A^\dagger \quad \text{Pseudo inverse of } A

I_n \quad n \times n \text{ identity matrix}
Chapter 1

Introduction

1.1 Sparse Distribution

The early work of Whitaker, Nyquist, and Shannon on sampling and representation of continuous signals allows signal processing to move from the analog to the digital domain. The Nyquist-Shannon sampling theorem states that a signal can be exactly reconstructed from its samples if the signal is uniformly sampled at a rate at least two times faster than the highest frequency of the signal. Unfortunately, in many practical applications, the resulting Nyquist rate is so high that we end up with too many samples. Therefore, we must consider compressing the samples in order to store or transmit them. In other applications signal acquisition is expensive, either because of a high cost per sample, or because data acquisition process is difficult. Examples include radar imaging and medical resonance imaging.

Transform coding can reduce the effective dimensionality of a large signal. Consider a real-valued, discrete-time signal $z \in \mathbb{R}^n$. Often there exists a basis $\{\psi_t\}_{t=1}^n$, which provides a so-called $k$-sparse representation of $z$, i.e. $z$ admits a representation $z = \psi_1x_1 + \cdots + \psi_nx_n$ where apart from only $k \ll n$ elements, all other elements of the set $\{x_t\}_{t=1}^n$ are practically zero. This phenomenon is very common, e.g. when $z$ represents an image then the wavelets provide a sparse representation. State-of-the-art compression algorithms exploit this fact as a small number $k$ of adaptively chosen transform coefficients $\{x_t\}_{t=1}^n$ are transmitted or stored rather than $n \gg k$ signal samples. However, in the standard transform coding framework, the complete set $\{x_t\}_{t=1}^n$ of transform coefficients is computed from $z$ and only a small subset is retained.

This observation motivates compressed sensing (CS)[1, 2, 3], a new researching
area, where instead of the usual approach of acquiring a finely sampled version of a signal, computing a full set of transform coefficients, and then discarding most of them, we can directly measure a compressed representation of the signal. In CS, we compute $m < n$ inner products between $z$ and another collection of column vectors $\{\varphi_i\}_{i=1}^m$ to form the measurements $y_i = \varphi_i'z$. Then

$$y := [y_1 \cdots y_m]' = \Omega'z = \Omega'\Psi x = \Phi x,$$

where $\Omega = [\varphi_1 \cdots \varphi_m]$ and $\Psi = [\psi_1 \cdots \psi_n]$. Clearly, $y \in \mathbb{R}^m$ and $\Phi = \Omega'\Psi \in \mathbb{R}^{m \times n}$ is referred to as the measurement matrix. As long as $\Phi$ satisfies some reasonable conditions, such measurements $\{y_t\}_{t=1}^m$ preserve the information content of the signal and $x$ can be reconstructed effectively from $y$. The impact of this approach goes far beyond compression. When acquiring data is dangerous, or expensive, we can manage with much less data than conventionally thought possible. For example, in medical imaging, the high radiation dose of an X-ray CT scan can be replaced by relatively few planar radiographs [4]. We can also apply CS method in conventional problems, to achieve nice results with small numbers of samples.

1.2 Decoding via Sparsity

Our goal is to decode $x$ given $y$ and $\Phi$. Since $m < n$, recovery of $x$ from $y$ is ill-posed in general. However, when the vectors $\{\varphi_i\}_{i=1}^m$ are incoherent\(^1\) [5] with the columns of $\Psi$ and $k < m$, then it is possible to recover $x$ from $y$. For example, if we know in advance which elements of $x$ are nonzero, then the most efficient and direct measurement scheme would simply project the signal into the associated $k$-dimensional subspace. However, the information about nonzero locations in $x$ is not known in practice. Hence, it is no longer possible to express the signal $x$ as a linear function of the measurement vector $y$. It requires solving the optimization problem

$$x = \arg \min_v \|v\|_0 \quad \text{subject to} \quad y = \Phi v,$$

where

$$\|v\|_0 := \lim_{\epsilon \to 0} \{|v_1|^{\epsilon} + \cdots + |v_n|^{\epsilon}\},$$

\(^1\)None of elements of the set $\{\varphi_i\}_{i=1}^m$ admits a sparse representation with respect to the basis $\{\psi_i\}_{i=1}^n$.\n
which is simply the number of nonzero components in $v$, also known as $\ell_0$ norm of $v$. Unfortunately, solving (1.2) is NP-hard [6] and also extremely sensitive to noise (any small amount of noise can change the number of zero components significantly). Hence decoding often relies on alternative optimization, which searches for the sparsest coefficients $x$ that agree with the measurement vector $y$. We will describe some decoding algorithms in the following chapters.

1.3 Illustrative Examples

Many signal processing applications require the identification and estimation of a few significant components of vectors of large dimension. The reason is that the original signal can be well approximated by those significant components. In this section we shall give examples of some signals having sparse representation with certain basis. We shall not give the details of the estimation process, rather more emphasis will be put in the construction of the sparse representation model.

Example 1.1. Direction of arrival estimation: Consider a uniform linear array (ULA), consisting of $m$ omnidirectional sensors (see Figure 1.1). The distance between any two sensors be $d$. Consider $k$ narrow-band signals $\{s_j(t)\}_{j=1}^k$ incident on the sensor array. Let $\theta$ be the direction of arrival (DOA) vector containing the DOA of individual
signals, i.e. the $j$ th component $\theta_j$ of $\theta$ gives the DOA of the signal $s_j(t)$. Let

$$y(t) = [ y_1(t) \cdots y_m(t) ]',$$

where $y_j(t)$ is the signal recorded after demodulation by the $j$ th sensor. The problem is to find the DOA of the signals.

Defining $s(t) = [ s_1(t) \cdots s_k(t) ]'$, and using the narrowband observation model [7], we have

$$y(t) = A(\theta)s(t). \tag{1.3}$$

Here $A(\theta)$ is the so-called manifold matrix. The manifold matrix consists of the steering vectors $\{ a(\theta_j) \}_{j=1}^k$:

$$A(\theta) = [ a(\theta_1) \cdots a(\theta_k) ].$$

The mapping $a(\theta)$ depends on the array geometry and the wave velocity, which are assumed to be known for any given $\theta$.

To cast DOA estimation problem as a sparse recovery problem, divide the whole area of interest into some discrete set of “potential locations”. Let the set of all potential DOAs be $\mathbb{G} = \{ \bar{\theta}_1, \cdots \bar{\theta}_n \}$, where typically $n \gg k$. Collect the steering vectors for each element of $\mathbb{G}$ in

$$\Phi = [ a(\bar{\theta}_1) \cdots a(\bar{\theta}_n) ].$$

Since $\mathbb{G}$ is known, $\Phi$ is known and is independent of $\theta$. Now, represent the signal field at time $t$ by $x(t) \in \mathbb{C}^n$, where the $j$ th component $x_j(t)$ of $x(t)$ is non-zero only if $\bar{\theta}_j = \theta$, and in that case $x_j(t) = s(t)$. Then one has a model

$$y(t) = \Phi x(t) \tag{1.4}$$

Since $k \ll n$, $x(t)$ is sparse. In effect, (1.4) lets us pose the problem of estimating $\theta$ as that of estimating a sparse $x(t)$.

**Example 1.2. Frequency estimation:** Consider a complex-valued continuous-time signal

$$y(t) = \sum_{j=1}^k a_j e^{i\omega_j t}. \tag{1.5}$$
The problem is to estimate the real valued frequencies \( \{\omega_j\}_{j=1}^k \) and the complex valued amplitude \( \{a_{j,\ell}\}_{j=1}^k \) from \( m \) samples of the observed data \( \{y(t)\}_{t=0}^{m-1} \). Using the same trick as before consider a uniform frequency grid

\[
\mathcal{G} = \{\bar{\omega}_\ell = 2\pi \ell/n : \ell = 0, 1, \ldots, n-1\},
\]

and define the DFT matrix \( \Psi \in \mathbb{C}^{m \times n} \) such that

\[
[\Psi]_{j,\ell} = \left\{ e^{i(j-1)\bar{\omega}_\ell} \right\}, \quad j \in \{1, \cdots, m\}, \ell \in \{1, \cdots, n\}
\]

Making \( n \) large enough, the grid \( \mathcal{G} \) becomes sufficiently dense. Then for every \( i \in \{1, \ldots, k\} \), we have \( \omega_i \approx \bar{\omega}_\ell \) for some \( \ell \in \{0, 1, \ldots, n-1\} \). Then we can write

\[
z := [y(t_0) \ y(t_1) \ \cdots \ y(t_{m-1})]' = \Psi x + \bar{e},
\]

The \( \ell \)-th component \( x_\ell \) is non-zero only if \( \bar{\omega}_\ell = \omega_i \) for some \( i \in \{1, \ldots, k\} \). This makes \( x \) a sparse vector, and the locations of the dominant peaks in \( x \) indicates the frequencies present in the data, while the dominant values give the complex-valued amplitudes.

**Example 1.3. Coherent spectrum estimation:** In last two examples, we were interested in estimating a single sparse signal \( x \) from given data \( y \). In this example, we consider an application whose sparse representation model has multiple sparse signals. We found that the sparse signals exhibit additional structure in the form that sparse signals have nonzero entries at the same locations.

Coherent spectrum estimation is a data analysis tool for estimating the frequencies present in a set of signals, which share the same frequencies. Consider \( J \) coherent signals \( \{y_j(t)\}_{j=1}^J \) of the form

\[
y_j(t) = \sum_{\ell=1}^k a_{j,\ell} e^{i\omega_{\ell}t}.
\]

Let us assume that the sampling instants of all signals are same. Define the vector of sampling instants as

\[
T = [t_0 \ t_1 \ \cdots \ t_{m-1}], \quad t_i < t_{i+1}.
\]

Now using the concepts in (1.6) set up a frequency grid \( \mathcal{G} \). Define the DFT matrix \( \Psi \in \mathbb{C}^{m \times n} \) similar to (1.7). We can write

\[
y_j := [y_j(t_0) \ y_j(t_1) \ \cdots \ y_j(t_{m-1})]' = \Psi x_j,
\]
Note that, the $\ell$-th component $x_{j,\ell}$ is non-zero only if $\bar{\omega}_\ell \approx \omega_j$ for some $j \in \{1, \ldots, k\}$. Hence, $x_j$ a sparse vector. Furthermore, every $y_j(t)$ have the same set of frequencies. Hence if the $\ell$-th component of $x_1$ is non-zero, then the $\ell$-th component of $x_j$ is nonzero for all $j$. Hence the matrix

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_J \end{bmatrix}$$

is joint row-sparse, i.e. it has only a few non-zero rows. Let $Y = [y_1, y_2, \cdots, y_J]$. Then we have a joint sparse data model

$$Y = \Psi X$$

Therefore, one way to solve the coherent spectral analysis problem is to estimate a joint sparse $X$ from $Y$.

**Example 1.4. Image compression:** Many natural images have sparse representation when expressed in wavelet basis. Consider a $p \times p$ image $Z$. Let, $z = \text{vec}(Z)$, where $\text{vec}(\cdot)$ is matrix vectorization operator. Let $n = p^2$. Let us expand the image on a wavelet basis $\{\psi_t\}_{t=1}^n$ such that $z = \psi_1 \bar{x}_1 + \cdots + \psi_n \bar{x}_n$. Let $\Psi = [\psi_1 \psi_2 \cdots \psi_n]$, then $z$ can be represented as

$$z = \Psi \bar{x}$$

(1.11)

where $\bar{x} \in \mathbb{R}^n$ represents the wavelet coefficients. For many natural images like $z$, most components of the vector $\bar{x}$ have negligible amplitude. For example, consider the image in Figure 1.2(a) and its wavelet coefficients $\bar{x}$ in Figure 1.2(b). Note that most of the components of $|\bar{x}|$ are small. Let $x$ be a vector constructed from $\bar{x}$ by retaining the components of $\bar{x}$ from its 99%-energy support while setting others to zero. Note that the actual dimension of $\bar{x}$ is 65536 in Figure 1.2(b). After retaining 99% energy support, the support of $x$ becomes 6560. The Figure 1.2(c) is reconstructed from $x$. There is very little difference between the actual and reconstructed image.

**Example 1.5. Magnetic resonance (MR) image sequence:** In this example, we consider a sparse representation model where some a-priori information about support of $x$ is known in advance. Many MR images have sparse representation when expressed in wavelet basis. Let us consider the cardiac MR image sequence\(^2\) in Figure 1.3 (a). Each MR image size is $32 \times 32$. Let $\bar{x}(t)$ be the wavelet coefficients vector of a MR image at time $t$. As before, let $x(t)$ be a vector constructed from $\bar{x}(t)$ by retaining the components

\(^2\)http://home.engineering.iastate.edu/~namrata/research/research.html
1.4 Thesis Outline

The thesis is divided into two parts. Outlines of the chapters in each part are given below.

- **Part I**

  Let $x$ be a sparse vector in the signal model (1.1). We need to estimate $x$ from $y$ and $\Phi$. Mathematically, one can write the reconstruction problem as in (1.2). Due to NP-hardness of solving (1.2), different alternatives are considered in literature. In Part I of the thesis, we discuss some sparse signal recovery algorithms. We propose some improvements of an algorithm. In some sparse signals, the locations of nonzero coefficients exhibit particular structures. We then concentrate on the structure of the sparse signal $x$. We develop some algorithms that can exploit the underlying structure of sparse signal. This can improve the recovery performance. The descriptions of the
1. Introduction

Figure 1.3: (a) Cardiac MR image sequence. (b) Cardiac MR images are sparsified using 2-D discrete wavelet transform. Wavelet coefficient vectors $x$ of different frames.

three chapters in this part follows next.

– Chapter 2
We first discuss some alternatives of $\ell_0$ norm based sparse recovery algorithm. We then consider the smoothed $\ell_0$ (SL0) approximation algorithm for sparse representation and propose an improved version of SL0. We give an upper bound on the run-time estimation error. This upper bound is tighter than the previously known bound. Subsequently, we develop a reliable stopping criterion. This criterion is helpful in avoiding the problems due to the underlying discontinuities of the $\ell_0$ cost function. Furthermore, we propose an alternative optimization strategy, which results in a Newton like algorithm. The optimization strategy also reveals that that some sparse recovery algorithms like, FOCUSS also follows Newton’s direction to optimize their cost functions. The contents of this chapter are partly based on the article
1.4. Thesis Outline


A preliminary version of this article has appeared also in


Some parts of this chapter are taken from


– Chapter 3

In this chapter, we develop some algorithms that can exploit structure of sparse signals. At first we consider the problem of finding a set of sparse signals that have nonzero coefficients in the same locations from a set of their compressed measurements. Such set of sparse signals is called joint-sparse. A cost function appropriate to the joint-sparse problem is developed, and an algorithm is derived. Second we consider another sparse recovery problem where the nonzero coefficients of a sparse signal occurs in clusters. In many practical applications the resulting sparse signals are complex valued. Hence, we extend formulations of the proposed algorithms for complex valued case. The materials of this chapter are partially taken from


Some parts of this chapter are taken from


– Chapter 4
We consider a sparse signal reconstruction scheme based on a maximum a posteriori (MAP) estimation method for compressive sensing when a part of the support of the sparse signal is known. This problem naturally arises, e.g., in magnetic resonance image (MRI) sequence, natural video sequences etc, where it is required to recursively reconstruct a sequence of mutually correlated sparse signals or images. Here one can use the support of the last signal as an estimate of the current signal support. However, the support information is often inaccurate, making the reconstruction problem challenging. We adopt a maximum a posteriori (MAP) estimation framework to deal with this issue. Simulation studies are performed, and the algorithm is applied to real MRI image sequences. The results are based on the article


A preliminary version of this article has appeared also in


• Part II

In the second part, we consider some practical problems and pose them to sparse signal representation problems with respect to some overcomplete basis. We seek to explore the potential of the algorithms developed in Part I to solve the problems. The performance of the algorithms are validated using simulation studies and experimental data.

– Chapter 5

The chapter demonstrates how the DOA estimation problem can be cast as the problem of recovering a joint-sparse signal. We consider both narrowband and broadband scenarios. We apply the joint-sparse recovery algorithm proposed in Chapter 3 to deal with the DOA estimation problem. Our algorithm can resolve closely spaced and highly correlated sources using a small number of noisy
snapsots. In addition, our algorithm can handle more sources than other state-of-the-art algorithms. For the broadband DOA estimation problem, the proposed algorithm allows relaxing the half-wavelength spacing restriction, which leads to a significant improvement in the resolution limit. The results are partly taken from


– Chapter 6
Spectral analysis is a useful tool for data analysis in signal processing. We consider frequency estimation of irregularly sampled signals. We enforce sparsity by imposing penalties based on an approximation of the \( \ell_0 \)-norm. Explicitly enforcing the sparsity of the representation is motivated by a desire to obtain a sharp estimate of the frequency spectrum that exhibits super-resolution. We consider frequency estimation from both single signal and multiple coherent signals. We find that the resulting algorithms can be implemented using a few FFTs and IFFTs. Hence, the implementation is very fast. We demonstrate the effectiveness of the sparse recovery approach using simulation experiments. The materials of this chapter are partially taken from


Some parts of this chapter are taken from


– Chapter 7
The problem of detection of multiple targets in a bistatic MIMO radar system is posed as a joint sparse signal recovery problem. We explore the potential of MIMO systems to locate targets in noisy environment with limited number of time samples, while the transmitted signals are not necessarily mutually orthog-
nal. Numerical experiments demonstrate that the proposed strategy outperforms existing algorithms. The contents of this chapter are taken from


– Chapter 8

In this chapter, we consider range-Doppler imaging problem. We find that under certain reasonable assumptions the problem can be posed to a two-dimensional sparse signal recovery problem with an overcomplete basis. The resulting problem can be solved using both $\ell_0$ and $\ell_1$ norm minimization algorithms. Target detection performances of the algorithms are illustrated using artificial data set. The results are based on the article


– Chapter 9

In this chapter, we consider a moving target detection method in presence of clutter based on a sparse representation of the sensor measurements of an airborne radar. In presence of clutter, the representation of sensor data is not sparse in general. We propose an easy way to estimate the clutter region. We then enforce sparsity by modeling the clutter as a single extended cluster of nonzero components. In particular, we pose the target detection in airborne radar as a problem of sparse signal recovery with partially known support. We apply the PMAP algorithm developed in Chapter 4 for target detection. The content of this chapter is a modification of concept of the article


1.5 Contributions

The target of the research work reported in this thesis is to develop a sparse signal recovery framework and seek the potential of the framework to solve some real world prob-
To achieve this goal, the author first considered an existing algorithm. The role of the author was to improve the algorithm to a form, that ensures better performance. A theoretical upper bound on the run-time estimation error is developed. In addition, the proposed modification gives an alternative optimization strategy of the existing algorithm, which results in a Newton like algorithm. The improved algorithm can take advantage from various type of structures in sparse signals, such as block-sparsity, joint-sparsity. Furthermore, the author has proposed a maximum a posteriori estimation based algorithm (see Chapter 4), that can improve sparse signal recovery performances when a part of the support of the sparse signal is known.

The potential of the proposed algorithms are explored by applied them in some practical problems in Part II of the thesis. In this part, the role of the author has been to develop sparse signal representation framework of some problems, and evaluate performance of the algorithms developed in Part I to solve the problems.

1.6 Topics for Future Research

The results of this work are encouraging and suggest that the proposed sparse recovery framework can be successfully used in many practical applications. However, there are several research questions that can be addressed in future.

- In Chapter 2, the analysis is based on the assumption that each component of $x$ is a Gaussian distributed random variable. However, different types of distribution may arise in practice. It is interesting to investigate whether the analysis can perform in more general form. If such an analysis exists then the theories of Chapter 2 will be more efficient.

- A MAP estimation approach is described in Chapter 4. The theme of the work is that by exploiting a priori information on support set of $x$, we can make the recovery performance better. We assume that the probability of a component of $x$ having large magnitude in the given support set is fixed. However, in some applications, the given support set can be divided into subsets such that the probability of a component of $x$ having large magnitude in a subset is different than the probability in other subset. A possible future work will be designing a MAP estimator to adopt the problem.
• In many practical applications, the overcomplete basis exhibit some structure. It is possible to make the sparse recovery algorithm better, robust and faster by exploiting the structure. For example, in Chapter 6, we exploit structure of the Fourier basis and the resultant algorithm becomes faster. Therefore another important research topic will be designing a more general form of the sparse recovery algorithm so that it can take advantages from most of the structured basis.

• In Chapter 5, we consider DOA estimation. To make an overcomplete representation of DOA estimation problem we divide the whole area of interest into some discrete set of “potential locations”. If actual DOA does not fall on the discretize grid then we consider grid enhancement. The similar grid enhancement step may consider in the applications of Chapter 6 and Chapter 7. However, it is shown that if the grid is fine enough then the sparse recovery algorithm cannot perform equally. An interesting future work will be developing an algorithm that can avoid grid selection.
Part I

Sparse Signal Recovery Algorithms
Chapter 2

Single Measurement Vector Problem

2.1 Introduction

Estimation of sparse vector $x$ from the measurement vector $y$ in (1.1) will be the main topic in this chapter. The estimation process requires solving the optimization problem in (1.2). However, (1.2) is a NP-hard problem. While the problem seems intractable, it has significant importance in signal processing, for example, in atomic decomposition on over-complete dictionaries [8, 5], blind source separation (BSS) [9, 10, 11], compressed sensing (CS) [1, 12, 13], image deconvolution, image denoising, electromagnetic imaging [14], and direction of arrival (DOA) finding. Hence researchers put significant effort into finding alternative approaches to solve the sparse recovery problem. In this chapter, we describe several alternative approaches. In addition, we propose an algorithm called improved smoothed $\ell_0$ (ISL0) for sparse signal recovery. In this chapter, we consider the measurement $y$ in (1.1) is a $m \times 1$ vector. The problem of recovering $x$ given $y$ and $\Phi$ is called single measurement vector (SMV) problem. In order to derive the results of the chapter, we rely on the following definitions.

**Definition 1.** A signal $x \in \mathbb{R}^n$ is called $k$-sparse if only $k$ of its components $x_i$ are nonzero and the remaining $(n-k)$ components are zero.

**Definition 2.** A matrix $\Phi \in \mathbb{R}^{m \times n}$, where $m < n$, is called a URP matrix if every $m$ columns of $\Phi$ are linearly independent.
Table 2.1: The OMP algorithm

<table>
<thead>
<tr>
<th>Initialization</th>
<th>1. Set $r^{(0)} = y, s = 1$.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Repeat</strong></td>
<td>2. Select the index of a column of $\Phi$ such that $i_s = \text{argmax}_{1 \leq i \leq n}</td>
</tr>
<tr>
<td></td>
<td>3. Update the approximation $y^{(s)} = \text{argmin}<em>{\tilde{y}} |y - \tilde{y}|<em>2^2$, such that $\tilde{y} \in \text{span}{\phi</em>{i_1}, \cdots, \phi</em>{i_s}}$.</td>
</tr>
<tr>
<td></td>
<td>4. $r^{(s)} = y - y^{(s)}, s = s + 1$.</td>
</tr>
<tr>
<td><strong>Until</strong></td>
<td>a stopping criterion is satisfied.</td>
</tr>
</tbody>
</table>

**Definition 3.** Mutual coherence of a matrix $\Phi$ is the maximum absolute inner product between any two columns of $\Phi$

$$M(\Phi) = \max_{1 \leq i,j \leq n, i \neq j} |G(i,j)|$$

where $G = \Phi^{\prime}\Phi$.

**Definition 4.** The coherence between two matrices $\Phi \in \mathbb{R}^{n \times n}$ and $\Psi \in \mathbb{R}^{n \times n}$ is

$$Q(\Phi, \Psi) = \max_{1 \leq i,j \leq n} |\langle \phi_i, \psi_j \rangle|$$

**Definition 5.** [15] For each integer $k = 1, 2, \cdots$, define the isometry constant $\varpi_k$ of a matrix $\Phi$ as the smallest number such that

$$(1 - \varpi_k)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \varpi_k)\|x\|_2^2 \quad (2.1)$$

holds for all $k$-sparse vectors $x$.

### 2.1.1 Orthogonal Matching Pursuit

Orthogonal matching pursuit (OMP) works in greedy fashion [16, 17, 18, 19]. The outline of the algorithm is given in Table 2.1. At each step, OMP selects a column $\phi_i$ from $\Phi$ such that $\phi_i$ has the maximum projection on a residual signal $r$. After each step the representation coefficients with respect to the atoms\(^1\) chosen so far are found via least-squares.

In general, the basic iteration of OMP is repeated until the residual norm is below some predetermined threshold. OMP is an appealing algorithm and is very simple to

\(^1\)Each column of $\Phi$ is called an atom. In OMP, it is assumed that the atoms have been normalized.
implement. It is easy to program. There are several variants of the OMP are available in literature [19]. However, the greedy algorithm can not perform equally in noisy environment [20, 21].

2.1.2 Basis Pursuit

The most popular alternative approach for solving (1.2) is called Basis Pursuit (BP) [8, 2, 22]. In BP the $\ell_0$ norm in (1.2) is replaced by $\ell_1$ norm. Hence the optimization problem becomes

$$x = \arg \min_v \|v\|_1 \quad \text{subject to} \quad y = \Phi v,$$

(2.2)

BP can be posed as a linear program [8] and can be solved efficiently [23], thereby the large scale problems remain tractable. The following lemma demonstrate that if $x$ is sufficiently sparse, the recovery of $x$ via $\ell_1$ is often exact.

**Theorem 1.** [24] Let $z \in \mathbb{R}^n$ has a $k$-sparse representation $x_0$ in the basis $\Psi$. Select $m$ measurements in the $\Omega$ domain uniformly at random such that $y = \Omega \Psi x_0$. Then if

$$m \geq C.(Q(\Omega, \Psi))^2.k.\log(n/\epsilon)$$

(2.3)

for some positive constant $C$, then

$$\text{Prob}\{x = x_0\} > 1 - \epsilon$$

in (2.2).

However, in order to be really powerful, $\ell_1$ needs to deal with noisy signal. To this aim, researchers consider the problem of recovering a vector $x \in \mathbb{R}^n$ from data

$$y = \Phi x + e$$

(2.4)

where $e$ is a stochastic or deterministic unknown error term. In noisy environment the optimization approach in (2.2) is modified as [3]

$$x^* = \arg \min_v \|v\|_1 \quad \text{subject to} \quad \|\Phi v - y\|_2 \leq \vartheta$$

(2.5)

where $\vartheta$ bounds the amount of noise in the data. The optimization problem in (2.5) is a convex problem (a second-order cone program) and can be solved efficiently.
Theorem 2. [25, 12] Suppose that \( \hat{x} \) is an arbitrary vector in \( \mathbb{R}^n \) and let \( \hat{x}^{[k]} \) be the truncated vector corresponding to the \( k \) largest values of \( |\hat{x}| \). Assume that \( \pi_{2k} < \sqrt{2} - 1 \) (see Definition 5). Then the solution \( x_* \) to (2.5) obeys

\[
\|x_* - \hat{x}\|_2 \leq C_0 \|x - \hat{x}^{[k]}\|_1 / \sqrt{k} + C_1 \varpi
\]

for some constants \( C_0 \) and \( C_1 \).

The value of \( C_0 \) and \( C_1 \) depend mainly on \( \pi_{2k} \).

2.1.3 FOCUSS

The advantage of \( \ell_1 \) based optimization in (2.2) is that it is convex, and can be solved efficiently. However, the cost is that more measurements are required, depending logarithmically on \( n \). On the contrary, it is shown in [26] that a nonconvex variant of basis pursuit will produce exact reconstruction with fewer measurements. Specifically, the \( \ell_1 \) norm is replaced with the \( \ell_p \) norm, where \( 0 < p < 1 \).

\[
\min_{x} \|x\|_p^p \text{ subject to } y = \Phi x
\]

(2.7)

The condition of Theorem 1 has been generalized [27] to \( (0 < p \leq 1) \)

\[
m \geq C_1(p)k + pC_2(p)k \log(n/k)
\]

(2.8)

for some constants \( C_1, C_2 \), and are bounded in \( p \).

FOCUSS stands for FOCal Underdetermined System Solver (FOCUSS) [28, 29]. FOCUSS relaxes \( \ell_0 \)-norm in (1.2) with an \( \ell_p \) for \( 0 < p < 1 \). In FOCUSS, a low-resolution initial estimate of the real signal is refined in successive iterations to obtain the final localized energy solution. The iterations are based on weighted norm minimization of the variable of optimization, where the weighting matrix depends on the solution to the last iteration.

In order to deal with the \( \ell_p \) norm, \( (0 < p < 1) \) a popular diversity measure is used in [29],

\[
\mathcal{F}_p(x) = \|x\|_p^p
\]

(2.9)

Consider the optimization problem:

\[
x_* (p) := \arg \min_{x \in \mathcal{X}} \mathcal{F}_p(x), \quad \mathcal{X} := \{x \in \mathbb{R}^n : \Phi x = y\}.
\]

(2.10)
The Lagrangian associated with (2.10) is

\[ L_p(x,\nu) = F_p(x) + \nu'(\Phi x - y), \]  

(2.11)

where \( \nu \in \mathbb{R}^{m \times 1} \) is the vector of Lagrange multipliers. The minimizer of \( L_p(x,\nu) \) with respect to \( x \) is a function of \( \nu \), and is denoted by \( \chi(\nu) \):

\[ \chi(\nu) = \arg\min_x L_p(x,\nu). \]

Thus \( \chi(\nu) \) is a stationary point of \( L_p(x,\nu) \) for a given value of \( \nu \), i.e.

\[ \frac{\partial L_p(x,\nu)}{\partial x} \bigg|_{x=\chi(\nu)} = \frac{\partial F_p(x)}{\partial x} \bigg|_{x=\chi(\nu)} + \Phi'\nu = 0. \]  

(2.12)

Note that \( F_p \) is non differentiable at any point with a zero component. A common approach for dealing with this issue is to regularize the optimization problem, by adding a small constant with \( x \) [30]. Denoting the \( j \)th component of \( x \) by \( x_j \), it is readily verified that

\[ \frac{\partial F_p(x)}{\partial x_j} = p|x_j|^{p-2}x_j \Rightarrow \frac{\partial F_p(x)}{\partial x} = pW(x)x, \]  

(2.13)

where

\[ W(x) := \text{diag}\{|x_1|^{p-2}, \ldots, |x_n|^{p-2}\}. \]

Then (2.12)-(2.13) gives

\[ \chi(\nu) = -\frac{1}{p}W^{-1}[\chi(\nu)]\Phi'\nu. \]  

(2.14)

Now for all \( \nu \) we have \( L_p(x,\nu) = F_p(x) \) for on the linear manifold \( \mathbb{X} \). Thus, if there exists \( \nu_* \) such that \( \Phi\chi(\nu_*) = y \), then

\[ \chi(\nu_*) = \arg\min_x L_p(x,\nu_*) = \arg\min_{x \in \mathbb{X}} L_p(x,\nu_*) \]

\[ = \arg\min_{x \in \mathbb{X}} F_p(x) \]

\[ = x_*(p). \]  

(2.15)

Now combining (2.14) and (2.15) we get

\[ \nu_* = -p[\Phi W^{-1}(x_*)\Phi']^{-1}\Phi x_*, \]

which upon substitution in (2.14) gives

\[ x_* = g(x_*), \]

(2.16)
where
\[ g(x) := W^{-1}(x)\Phi' \left[ \Phi W^{-1}(x)\Phi' \right]^{-1} y. \]

Equation (2.16) is nonlinear, and cannot be solved analytically in general. Nevertheless it is used in a fixed point iteration \([29]\), where the iterate \(x^{(i+1)}\) in the \((i + 1)\)-th iteration is given in terms of \(x^{(i)}\) as
\[
x^{(i+1)} = g(x^{(i)}) = W^{-1}(x^{(i)})\Phi' \left[ \Phi W^{-1}(x^{(i)})\Phi' \right]^{-1} y,
\]
and the corresponding algorithm is called FOCUSS. Several results exist in literature \([28, 29]\) justifying the above iterative approach, which has been tested in many practical problems. However, one of the main drawbacks of FOCUSS is the large computation cost involved in every iteration. In general, FOCUSS takes large number of iterations to obtain a sparse solution.

### 2.1.4 Contributions and Motivation

Despite the noise sensitivity of the \(\ell_0\) norm, it gives the highest possibility of sparse recovery with relatively small number of measurements \([1]\). This motivates the use of approximate \(\ell_0\) functions to solve (1.2) \([30, 31, 32, 33, 34]\). A hyperbolic tangent based \(\ell_0\) approximation approach is developed in \([32]\), however its performance deteriorates when signal sparsity increases and its computation time is very high. The “smoothed \(\ell_0\)” (SL0) cost function \([33, 34]\) approximates the negated \(\ell_0\) norm using a class of Gaussian functions. A steepest accent algorithm is derived in \([34]\) to maximize the SL0 cost function. The algorithm is fast, and gives significantly improved performance in noisy environments. However, there is no suitable stopping criterion of SL0. Also the error bound of SL0 is somewhat loose.

We propose some improvements over the SL0 methodology \([34]\). First, we give a theoretical bound on the run-time estimation error, which is significantly tighter than the previously known result \([34]\). This bound can be computed for every iterate of the numerical optimizer of SL0, and does not require the knowledge of the signal sparsity. Subsequently, we give a systematic method to find a stopping criterion for the SL0 algorithm.

We also investigate alternative strategies that could be adopted to optimize the SL0 cost function. The steepest accent algorithm in \([34]\) is designed to make a trade-off
between the computation time and estimation performance. We refer to this approach as the “original SL0 (O-SL0) algorithm”. O-SL0 being a steepest ascent approach, often requires a large number of iterations. To address this issue we propose an alternative Newton like ascent direction. This ascent direction is motivated by the Lagrangian formulation of the SL0 cost function, and can be shown to be the Newton direction associated with a convex-concave procedure to optimize the SL0 function. The resulting optimization strategy is called the “improved SL0” (ISL0) algorithm. Based on the optimization strategy it can be shown that FOCUSS is also a convex-concave procedure which follows Newton’s direction to optimize its diversity measure. ISL0 has also been tested on different applications using real-world data.

2.2 Smoothed \( \ell_0 \) (SL0) cost function and its minimization

In order to deal with the discontinuities in \( \|x\|_0 \) on the hyperplanes passing through one or more axes of co-ordinates, a Gaussian family of functions is used in [34]. Define

\[
f_\sigma(\alpha) = e^{-\frac{\alpha^2}{2\sigma^2}}.
\]

Then it is readily verified that, as \( \sigma \to 0 \), the function

\[
F_\sigma(x) = \sum_{t=1}^{n} f_\sigma(x_t).
\]

behaves like \( n - \|x\|_0 \), motivating the following approximate reformulation of (1.2):

\[
x_*(\sigma) := \arg \max_{x \in \mathbb{X}} F_\sigma(x), \quad \mathbb{X} := \{x \in \mathbb{R}^n : \Phi x = y\},
\]

while we take \( \sigma \to 0 \). Like \( \|x\|_0 \), the function \( F_\sigma(x) \) has many local maxima for a small \( \sigma \). However, as \( \sigma \) increases, \( F_\sigma(x) \) becomes smoother, and for a sufficiently large \( \sigma \), (2.20) admits a unique solution:

\[
\lim_{\sigma \to \infty} x_*(\sigma) = \Phi'(\Phi\Phi')^{-1}y
\]

[34]. This motivates solving a sequence optimization problems. At the first step we solve (2.20) for \( \sigma = \sigma_1 \) (which is a sufficiently large number) by initializing the numerical optimizer at \( x = \Phi'(\Phi\Phi')^{-1}y \). Subsequently we reduce \( \sigma \) by a small factor \( \rho \), and solve (2.20) again for \( \sigma = \rho \sigma_1 \). This time we initialize the numerical optimization at \( x_*(\sigma_1) \) obtained in the previous iteration. If \( \rho \) is small enough, one can expect that \( \|x_*(\sigma_1) - x_*(\rho \sigma_1)\|_2 \) is small.
Table 2.2: Minimization of the SL0 cost function: the basic idea.

<table>
<thead>
<tr>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set $x^{(0)} = \Phi' (\Phi \Phi')^{-1} y$.</td>
</tr>
<tr>
<td>2. Set a $\sigma$ decreasing factor $0 &lt; \rho &lt; 1$.</td>
</tr>
<tr>
<td>3. Set a large $\sigma$ and a small $\sigma_{\min}$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. $x(\sigma) := \arg \max_{x \in \mathbb{X}} F_{\sigma}(x)$.</td>
</tr>
<tr>
<td>5. $\sigma = \rho \sigma$.</td>
</tr>
</tbody>
</table>

while $\sigma > \sigma_{\min}$

Hence it is sufficient if the numerical optimizer is capable of finding a local maximum point. Once we have $x_*(\rho \sigma_1)$ we can solve (2.20) again to find $x_*(\rho^2 \sigma_1)$ as before. This iterative procedure is repeated until the $i$-th iteration when either the value $\rho^{i-1} \sigma_1$ is sufficiently small or $\|x_*(\rho^{i-1} \sigma_1) - x_*(\rho^{i-2} \sigma_1)\|_2$ is sufficiently small, and $x_*(\rho^{i-1} \sigma_1)$ is taken as the final solution. The central idea to optimize SL0, proposed in [34], is shown in Table 2.2.

2.2.1 Analysis

In the sequel, $x_0$ denotes the true value of $x$, while $x^{(i)}$ denotes the $i$-th iterate of the numerical algorithm used to optimize the SL0 function. There are two goals in this section. First we aim to establish a mechanism which lets us compute an upper bound on $\|x^{(i)} - x_0\|_2$ in every iteration of any algorithm used to optimize the SL0 cost function. Secondly we want to derive a reliable stopping criterion. Taking $\sigma$ very close to 0 makes $F_{\sigma}$ extremely non-smooth. As a result, the estimation error tends to increase if we decrease $\sigma$ beyond a threshold. We give a systematic procedures to pre-compute a good value of $\sigma_{\min}$ (see Table 2.2).

2.2.1.1 An upper bound on the estimation error

**Assumption 1.** The measured data $y$ satisfies $y = \Phi x_0 + e$ for some $k$-sparse $x_0 \in \mathbb{R}^n$ where $k < m/2$.

**Assumption 2.** Every $m$ columns of $\Phi$ are linearly independent [28]. Furthermore, $\Phi \Phi' = I$ [35]

If the rows of $\Phi$ are not orthonormal, we can always choose an orthonormal basis of the row-space of $\Phi$ using, for instance, the QR factorization of $\Phi$. Let $\Phi = R \Phi_1$, where
2.2. Smoothed ℓ₀ (SL₀) cost function and its minimization

$R$ is lower triangular, and $\Phi_1 \Phi'_1 = I$. Then the problem can be posed as $R^{-1} y = \Phi_1 x$. It is shown in [1] that under Assumptions 1-2, $y$ preserves $x_0$ uniquely provided that $e = 0$. For any partition of $\Phi = [\Phi_1 \Phi_2]$, we have

$$I - \Phi_1 \Phi'_1 = \Phi \Phi' - \Phi_1 \Phi'_1 = \Phi_2 \Phi'_2, \quad (2.21)$$

which is a non-negative definite matrix. Hence the maximum eigenvalue of $\Phi_1$ is no larger than 1.

**Notation:** Let $T \subset \{1, \cdots, n\}$ be a set of indices. Then we denote (using Matlab notation)

$$\Phi_T = \Phi(:, T),$$

i.e. $\Phi_T$ is the submatrix obtained by extracting the columns of $\Phi$ corresponding to the indices of $T$. For a row or a column vector $a$, similarly, we define $a_T = a(T)$. In addition, $\text{card}(T)$ denotes cardinality of $T$.

**Lemma 1.** Let the Assumptions 1-2 hold. Consider any $\overline{x} \in \mathbb{X}$ (see (2.20) for the definition of $\mathbb{X}$) such that at least $n - m/2$ components of $\overline{x}$ have absolute values below $h$. Denote $\tilde{x} = \overline{x} - x_0$ and define $T = \{t \in \{1, \cdots, n\} : |\tilde{x}_t| \geq h\}$. Let $\alpha$ be the minimum eigenvalue of $\Phi'_T \Phi_T$. Then

$$||\overline{x} - x_0||^2 \leq \frac{1}{\alpha}(\sqrt{n}h + \|e\|_2)^2 + nh^2. \quad (2.22)$$

**Proof.** Let $T_c$ be the complement of $T$ in $\{1, \cdots, n\}$. Since $n - m/2$ components of $\overline{x}$ have absolute values below $h$, and only $k$ components of $x_0$ are nonzero, then

$$\text{card}(T_c) \geq n - m/2 - k \geq n - m,$$

implying $\text{card}(T) \leq m$. Now as $\overline{x} \in \mathbb{X}$ it follows that

$$e = \Phi \tilde{x} = \Phi_T \tilde{x}_T + \Phi_{T_c} \tilde{x}_{T_c},$$

$$\Rightarrow ||\Phi_T \tilde{x}_T||_2 \leq ||e||_2 + ||\Phi_{T_c} \tilde{x}_{T_c}||_2. \quad (2.23)$$

Recall that $||\Phi_{T_c}||_2 \leq 1$, see the discussion around (2.21). Since $\alpha$ is the minimum eigenvalue of $\Phi'_T \Phi_T$, (2.23) gives

$$\sqrt{\alpha}||\tilde{x}_T||_2 \leq ||\tilde{x}_{T_c}||_2 + ||e||_2. \quad (2.24)$$
As \( \text{card}(\mathcal{T}) \leq m \), Assumption 2 gives \( \alpha > 0 \). Using (2.24),

\[
||\hat{x}||_2^2 = ||\hat{x}_\mathcal{T}||_2^2 + ||\hat{x}_{\overline{\mathcal{T}}}||_2^2 \leq \frac{1}{\alpha}(||\hat{x}_\mathcal{T}||_2 + ||e||_2)^2 + ||\hat{x}_{\overline{\mathcal{T}}}||_2^2.
\]  

(2.25)

Now using \( ||\hat{x}_\mathcal{T}||_2^2 \leq h_{\mathcal{T}}^2 \) in (2.25) we get (2.22).

Let \( h^{(i)} \) be the smallest number such that at least \( n - m/2 \) components of \( x^{(i)} \) have absolute values below \( h^{(i)} \). Then during numerical optimization it is straightforward to find \( h^{(i)} \) by inspecting \( m/2 \) largest components of \( |x^{(i)}| \). Similarly we find \( \mathcal{T}^{(i)} = \{ t \in \{ 1, \cdots , n \} : |\hat{x}_t^{(i)}| \geq h^{(i)} \} \), and calculate \( \alpha^{(i)} \) as the minimum eigenvalue of \( \Phi'_{\mathcal{T}^{(i)}} \Phi_{\mathcal{T}^{(i)}} \).

Then we can obtain an upper bound on \( ||x^{(i)} - x_0||_2 \) using Lemma 1.

If the components of \( e \) are independent and identically distributed with a finite variance \( \mu^2 \), then \( ||e||_2 \approx \sqrt{m\mu} \). Consequently, the error bound (2.22) is \( O(\sqrt{n}) \). This is significantly tighter than the \( O(n) \) bound obtained in [34]. Moreover, to compute the bound (2.22) in a practical scenario we recommend replacing \( ||e||_2 \) by \( \sqrt{m\mu} \) as \( ||e||_2 \) is unknown in practice. This is done in generating the plots in Figure 2.7, where we compare \( ||x^{(i)} - x_0||_2 \) against the run-time upper bound.

2.2.1.2 A stopping criterion

Let \( h_\sigma \) be the smallest number such that \( n - m/2 \) components of \( x_\sigma(\sigma) \) have absolute values below \( h_\sigma \). Then \( h_\sigma \) is a measure of how sparse \( x_\sigma(\sigma) \) is. In addition, having a priori knowledge of \( h_\sigma \) is also useful in detecting if the optimizer is stuck at a local maximum point. Our goal is to give a probabilistic upper bound on \( h_\sigma \) for any \( \sigma \). Lemma 2 is the first step in that direction.

**Lemma 2.** Let \( x \in \mathbb{R}^n \) and \( h \) be such that the set

\[
\mathcal{T}(x) := \{ t \in \{ 1, \cdots , n \} : |x_t| > h \}
\]

has a cardinality \( m/2 \), and

\[
n - m/2 \leq F_\sigma(x).
\]

Then

\[
h \leq \sigma \sqrt{-2\log \{ 1 - 2[n - F_\sigma(x)]/m \}}.
\]  

(2.26)
2.2. Smoothed $\ell_0$ (SL0) cost function and its minimization

Proof. Let $T_c$ denote the complement of $T$ in $\{1, 2, \ldots, n\}$. If $t \in T$ then $f_\sigma(x_t) \leq f_\sigma(h)$. In addition, $f_\sigma(x_t) \leq 1$ for any $t$. Hence

$$F_\sigma(x) = \sum_{t \in T} f_\sigma(x_t) + \sum_{t \in T_c} f_\sigma(x_t) \leq \frac{m}{2} f_\sigma(h) + \left(n - \frac{m}{2}\right) \leq n - \frac{m}{2} \{1 - f_\sigma(h)\},$$

which implies

$$f_\sigma(h) \geq 1 - 2[n - F_\sigma(x)]/m. \quad (2.27)$$

While, $F_\sigma(x) \leq n$ is always true, $n - m/2 \leq F_\sigma(x)$ may not hold. If $n - m/2 > F_\sigma(x)$, then the right hand side of (2.27) is negative. Since $f_\sigma(h) \geq 0$ anyway, (2.27) fails to give any new information if $n - m/2 > F_\sigma(x)$. Otherwise, by simplifying (2.27) we get (2.26), where the right-hand slide is less than $\infty$.

Now if we have a way to predict $F_\sigma(x_*(\sigma))$, then we can use Lemma 2 to predict an upper bound on $h_\sigma$. This analysis will be used to find how $h_\sigma$ should vary with $\sigma$, which will let us give a reliable stopping criterion.

We introduce a randomized framework, usually known as the Bernouilli-Gaussian model, which has been used by many authors, e.g. [34].

Assumption 3. For $t = 1, \ldots, n$, the $t$-th component $x_{0t}$ of $x_0$ is given by the product of two random variables

$$x_{0t} = u_t v_t$$

where $v_t$ is a Gaussian distributed random variable with mean zero and variance $\sigma^2$, while $u_t$ is a discrete random variable which can assume values in the set $\{0, 1\}$ such that

$$\text{Prob}\{u_t = 1\} = p, \quad \text{Prob}\{u_t = 0\} = 1 - p.$$

In addition, $v_t$ is independent of $v_{t1}$ if $t \neq t_1$, and similarly $u_t$ is independent of $u_{t1}$ if $t \neq t_1$. Furthermore, $v_t$ is independent of $u_{t1}$ for all $t$ and $t_1$.

Assumption 4. The noise $e$ is Gaussian with mean 0 and covariance matrix $\mu^2 I_m$. In addition $e$ is independent of

$$u = [u_1 \ldots u_n], \quad v = [v_1 \ldots v_n].$$
Ideally $p = k/n$. In practice $k$ is unknown. Nevertheless, $p < m/(2n)$ can be used. The variances $\sigma^2$ and $\mu^2$ depend on the nature of signals and measurement noise that is expected, and varies from one application to other. However, it is quite reasonable to assume that the user has an estimate of these values.

By Assumption 1, we know that $x_0 \notin X$ unless $e = 0$. However, the point on $X$ nearest to $x_0$ is

$$x_1 = x_0 + (\Phi\Phi')^{-1}\Phi' e = x_0 + \Phi' e,$$

recall that $\Phi\Phi' = I_m$ by Assumption 2. Hence

$$F_\sigma(x_1(\sigma)) \geq F_\sigma(x_1). \tag{2.28}$$

If $x_1(\sigma)$ should be close to $x_0$ for small $\sigma$, it is natural to expect that the lower bound (2.28) on $F(x_1)$ is tight. In the following we explore two ways to obtain the properties of the probability distribution function of $F_\sigma(x_1)$ using Assumptions 3 and 4.

### 2.2.1.3 The analytical approach

Define

$$r_{ij}(\theta) := \begin{cases} 
\sigma^{-2}(\sigma^2\theta + \mu^2\phi_i'\phi_i), & i = j \\
\sigma^{-2}\mu^2\phi_i'\phi_j, & i \neq j.
\end{cases} \tag{2.29}$$

and

$$q_{ij}(\theta_1, \theta_2) = [(1 + r_{ii}(\theta_1))\{1 + r_{jj}(\theta_2)\} - r_{ij}^2]^{-1/2} \tag{2.30}$$

Then it can shown that (see Appendix A.1 for a proof)

$$E\{F_\sigma(x_1)\} = \sum_{i=1}^n \bigg[p\{1 + r_{ii}(1)\}^{-1/2} + (1 - p)\{1 + r_{ii}(0)\}^{-1/2} \bigg], \tag{2.31}$$

and

$$E\{F^2_\sigma(x_1)\} = \sum_{i=1}^n \bigg[p\{1 + 2r_{ii}(1)\}^{-1/2} + (1 - p)\{1 + 2r_{ii}(0)\}^{-1/2} \bigg] + \sum_{i=2}^{n-1} \sum_{j=1}^{i-1} \bigg[p^2 q_{ij}(1,1) + (1 - p)^2 q_{ij}(0,0) + p(1 - p)\{q_{ij}(0,1) + q_{ij}(1,0)\} \bigg]. \tag{2.32}$$
Denote
\[ b(\epsilon) = E\{F_\sigma(x_1)\} - \epsilon^{-1}\sqrt{E\{F_\sigma^2(x_1)\} - [E\{F_\sigma(x_1)\}]^2} \]
Note that the second term in the right hand side of the equation is the standard deviation of \( F_\sigma(x_1) \). Then using Tchebychev’s inequality it follows that
\[ F_\sigma(x_*) \geq F_\sigma(x_1) \geq b(\epsilon) \] \hspace{1cm} (2.33)
with probability \( 1 - \epsilon^2 \). Therefore, if \( b(\epsilon) \geq n - m/2 \), then
\[ h_\sigma \leq \sigma \sqrt{-2\log\{1 - 2[n - b(\epsilon)]/m\}} =: a(\sigma) \] \hspace{1cm} (2.34)
with probability \( 1 - \epsilon^2 \) by Lemma 2. The right hand side of (2.34) is independent of the problem data, and can be pre-computed as a function of \( \sigma \) even if \( y \) is not available.

### 2.2.1.4 Monte-Carlo simulation approach

The Tchebychev’s bound in (2.33) can be somewhat loose in some cases. An alternative approach to obtain a more tight bound would be to use Monte-carlo simulations. In this approach one simulates a large number of realizations of the random variable \( x_1 \) according to the rules postulated in Assumptions 3 and 4, and computes \( F_\sigma(x_1) \) for each of these realizations. These realizations of \( F_\sigma(x_1) \) can then be used to compute an estimate of the probability distribution function of \( F_\sigma(x_1) \). The accuracy of the distribution function estimation improves with the increase in the number of Monte-Carlo simulations performed, and for any \( \epsilon \) satisfying \( 0 < \epsilon < 1 \), one can estimate an empirical \( b_{mc}(\epsilon) \) such that
\[ F_\sigma(x_*) \geq F_\sigma(x_1) \geq b_{mc}(\epsilon), \] \hspace{1cm} (2.35)
with probability \( 1 - \epsilon^2 \). Subsequently, similar to (2.34) one can derive a bound where
\[ h_\sigma \leq \sigma \sqrt{-2\log\{1 - 2[n - b_{mc}(\epsilon)]/m\}} =: a(\sigma) \] \hspace{1cm} (2.36)
with probability \( 1 - \epsilon^2 \). Although obtaining an accurate bound this way involves more computation than the analytical approach, all the computations can done in advance without the knowledge of \( y \). Define
\[ \sigma_* := \arg\min_\sigma a(\sigma). \]
The upper bound on \( ||x_*(\sigma) - x_0||_2 \) in Lemma 1 decreases when \( h_\sigma \) decreases. Hence it is natural to conjecture that \( ||x_*(\sigma) - x_0||_2 \) will be minimum near \( \sigma = \sigma_* \). This motivates the
choice \( \sigma_{\text{min}} = \sigma^* \). Numerical simulation studies support the conjecture, which is illustrated in Section 2.3.3 using an example in Figure 6. We note that the Tchebychev bound (2.34) can also be used to find \( \sigma_{\text{min}} \). However, in our experience the Monte-Carlo simulation approach gives better results.

2.2.2 Optimization strategies

2.2.2.1 Steepest ascent algorithms

Table 2.3: ASA-SL0 Algorithm

<table>
<thead>
<tr>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set ( x^{(0)} = \Phi'(\Phi\Phi')^{-1}y ).</td>
</tr>
<tr>
<td>2. Set a ( \sigma ) decreasing factor ( 0 &lt; \rho &lt; 1 ).</td>
</tr>
<tr>
<td>3. Set ( 0 &lt; \eta &lt; 1 ), and a large ( \sigma ).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repeat</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. ( \tau = x^{(i)} + (\sigma^2)\nabla F_\sigma(x^{(i)}) )</td>
</tr>
<tr>
<td>5. Project ( \tau ) onto ( X = {x</td>
</tr>
<tr>
<td>( x^{(i+1)} = \tau - \Phi'(\Phi\Phi')^{-1}(\Phi\tau - y) )</td>
</tr>
<tr>
<td>6. If ( \tau^{(i)} =</td>
</tr>
<tr>
<td>( \sigma = \rho \sigma ).</td>
</tr>
</tbody>
</table>

**while** \( \sigma > \sigma_{\text{min}} \).

While the analysis of the previous section gives a reliable stopping criterion, there are many ways to implement the basic algorithm in Table 2.2. For example, in [34] a steepest ascent algorithm is used in step 4, Table 2.2 to maximize \( F_\sigma(x) \). A natural way to implement the steepest ascent approach is given in Table 2.3. Numerical simulation experiments reveal that the resulting steepest ascent algorithm in Table 2.3 gives high estimation accuracy, and we refer to this as “accurate steepest ascent SL0 (ASA-SL0)” algorithm. However, ASA-SL0 requires a large number of iterations, and hence large computation time. This issue is dealt with in [34] by using an alternative approach, which we call the “original SL0 (O-SL0) algorithm”. While maximizing \( F_\sigma(x) \) for a fixed \( \sigma \) (see step 4 in Table 2.2), O-SL0 does not iterate until convergence to a maximum point. Instead it takes a fixed number (three, according to the recommendation in [34]) of steepest ascent steps, and moves to step 5. In effect O-SL0 trades off estimation accuracy with computation time. Compared to ASA-SL0, the O-SL0 approach requires significantly smaller computation time, but sacrifices some estimation performance.
2.2. Smoothed $\ell_0$ (SL0) cost function and its minimization

2.2.2.2 Using an alternative ascent direction

Slow convergence is typical of steepest ascent direction [37]. In convex optimization the Gauss-Newton direction is often used [23]. Motivated by these facts, we resort to an alternative direction to optimize the cost function of SL0.

The Lagrangian associated with (2.20) is

$$L_\sigma(x, \nu) = F_\sigma(x) + \nu'(\Phi x - y),$$

(2.37)

where $\nu \in \mathbb{R}^{m \times 1}$ is the vector of Lagrange multipliers. If $(x^*, \nu^*)$ is a stationary point of $L_\sigma(x, \nu)$, then

$$\frac{\partial L(x^*, \nu^*)}{\partial x} = \frac{\partial F_\sigma(x^*)}{\partial x} + \Phi' \nu^* = 0,$$

(2.38)

$$\frac{\partial L(x^*, \nu^*)}{\partial \nu} = \Phi x^* - y = 0.$$

It is readily verified that

$$\frac{\partial F_\sigma(x)}{\partial x_t} = -\frac{x_t}{\sigma^2} f_\sigma(x_t).$$

Hence,

$$\frac{\partial F_\sigma(x)}{\partial x} = -\frac{1}{\sigma^2} W(x)x,$$

(2.39)

where

$$W(x) := \text{diag}\{f_\sigma(x_1), \cdots, f_\sigma(x_n)\}.$$ 

Then (3.9)-(2.39) gives

$$x^* = \sigma^2 W^{-1}(x^*) \Phi' \nu^*$$

(2.40)

Substituting for $x^*$ in the second equation of (3.9), solving for $\nu^*$ and substituting this expression for $\nu^*$ in (3.11) we get

$$x^* = W^{-1}(x^*) \Phi' [\Phi W^{-1}(x^*) \Phi']^{-1} y.$$ 

(2.41)

Equation (2.41) is nonlinear, and cannot be solved analytically. Nevertheless, as the following lemma suggests, it does lead to an alternative ascent direction.

**Lemma 3.** Let us define the map $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that

$$g(x) = W^{-1}(x) \Phi' [\Phi W^{-1}(x) \Phi']^{-1} y.$$ 

(2.42)
Then $\Phi g(x) = y$. Let $x \in \mathbb{R}^n$ such that $\Phi x = y$, $g(x) \neq x$, and

$$
\left[ \frac{\partial F_{\sigma}(x)}{\partial x} \right] \neq 0.
$$

Then $F_{\sigma}(x)$ is an increasing function along the direction $g(x) - x$.

**Proof.** Pre-multiplying (2.42) by $\Phi$ we verify that $\Phi g(x) = y$ for all $x \in \mathbb{R}^n$. Now suppose $x$ satisfies $\Phi x = y$, and (2.43) holds. Then using (2.39) we get

$$
\left[ \frac{\partial F_{\sigma}(x)}{\partial x} \right] \{g(x) - x\}
\quad = \quad \frac{1}{\sigma^2} x' \{ W(x) - \Phi' \left[ \Phi W^{-1}(x) \Phi' \right]^{-1} \Phi \} x
\quad = \quad \frac{1}{\sigma^2} \{ W^{0.5}(x)x \} \Pi \{ W^{0.5}(x)x \}.
$$

(2.44)

where

$$
\Pi = I - W^{-0.5}(x) \Phi' \left[ \Phi W^{-1}(x) \Phi' \right]^{-1} \Phi W^{-0.5}(x)
$$

is the orthogonal projection operator on the nullspace of $\Phi W^{-0.5}(x)$. From (2.44) it is clear that

$$
\left[ \frac{\partial F_{\sigma}(x)}{\partial x} \right] \{g(x) - x\} \geq 0,
$$

(2.45)

with equality is satisfied only if $W^{0.5}(x)x$ resides in the columnspace of $W^{-0.5}(x)\Phi'$, which means

$$
W^{0.5}(x)x = W^{-0.5}(x)\Phi' \beta \quad \Rightarrow \quad \frac{\partial F_{\sigma}(x)}{\partial x} = \Phi' \frac{-\beta}{\sigma^2}
$$

for some $\beta \neq 0$. This implies that we have equality in (2.45) only if

$$
\frac{-\beta}{\sigma^2} = \left[ \Phi W^{-1}(x) \Phi' \right]^{-1} y \quad \Rightarrow \quad x = g(x).
$$

However $x \neq g(x)$ by assumption. Hence we have a strict inequality in (2.45). This means the inner-product between $g(x) - x$ and the gradient of $F_{\sigma}(x)$ is always positive. So, the value of $F_{\sigma}(x)$ increases along the line joining $x$ and $g(x)$.

Lemma 3 ensures that $F_{\sigma}$ is increasing at $x$ along $g(x) - x$, making it an ascent-direction. In fact, this direction is the same as the Newton direction of a convex-concave procedure used to optimize the SL0 cost function.
2.2. Smoothed $\ell_0 (SL0)$ cost function and its minimization

2.2.2.3 Connection to the Newton direction

First consider a related optimization problem

$$x_\ast(\sigma, \lambda) = \arg \max_x L_\sigma(x),$$
$$L_\sigma(x) := F_\sigma(x) - \lambda \|y - \Phi x\|^2_2.$$ (2.46)

where we have included an additional term $\lambda \|y - \Phi x\|^2_2$ which is often used to penalize the measurement noise in $y$. Here $\lambda$ determines how much weight is given to the residual $y - \Phi x$. If we want to nullify the residual we must make $\lambda$ very large, so that the solution is forced to lie on $X$. Thus we have $\lim_{\lambda \to \infty} x_\ast(\sigma, \lambda) = x_\ast(\sigma)$. It is readily verified that

$$\frac{\partial L_\sigma(x)}{\partial x} = -W(x)x/\sigma^2 + \lambda \Phi'(y - \Phi x),$$
$$\frac{\partial^2 L_\sigma(x)}{\partial x \partial x'} = -W(x)/\sigma^2 + 2\Gamma W(x)\Gamma/\sigma^4 - \lambda \Phi'\Phi,$$

where $\Gamma = \text{diag}(x)$. The Hessian has a negative definite part: $H_1(x) = -W(x)/\sigma^2 - \lambda \Phi'\Phi$, which is associated to the concave component of the cost; and a positive semidefinite part $H_2(x) = 2\Gamma W(x)\Gamma/\sigma^4$ associated to the convex part of the cost function. Hence the convex-concave procedure to maximize $L_\sigma$ will drop the positive definite part of the Hessian [38, 39] and evaluate the Newton ascent direction as

$$d(x) = -H_1^{-1}(x) \frac{\partial L_\sigma(x)}{\partial x} = - \left[ \frac{W(x)}{\sigma^2} + \lambda \Phi' \Phi \right]^{-1} \left[ \frac{W(x) x}{\sigma^2} - \lambda \Phi'(y - \Phi x) \right].$$ (2.47)

Now if $x \in X$, then $y = \Phi x$. Then for all $x \in X$ using the matrix inversion lemma in (2.47) we get

$$d(x) = -[W(x)/\sigma^2 + \lambda \Phi'\Phi]^{-1}[W(x)x/\sigma^2]$$
$$= -x + \sigma^2 W^{-1}(x)\Phi'(I_m/\lambda + \sigma^2 \Phi W^{-1}(x)\Phi')^{-1}y, \quad x \in X.$$ (2.48)

Now taking $\lambda \to \infty$ it is readily verified that $d(x) = g(x) - x$.

Based on the findings in that section, it can be shown that some sparse signal recovery approaches, like, FOCUSS is a convex-concave procedure. It follows a Newton-like decent direction by retaining the positive definite component of the Hessian matrix. This idea can use to improve performance of FOCUSS algorithm.
2. Single Measurement Vector Problem

Table 2.4: ISL0 Algorithm

<table>
<thead>
<tr>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set $x^{(0)} = \Phi' (\Phi \Phi')^{-1} y$.</td>
</tr>
<tr>
<td>2. Set $\sigma = 2 \max_i</td>
</tr>
<tr>
<td>3. Set $\rho, \eta, \gamma \in (0,1)$.</td>
</tr>
</tbody>
</table>

repeat

| 4. Set $\kappa = 1$. |
| 5. while $F_\sigma(\kappa g(x^{(i)}) + (1 - \kappa)x^{(i)}) < F_\sigma(x^{(i)})$ |
| $\kappa = \gamma \kappa$. |
| end |

| 6. $x^{(i+1)} = \kappa g(x^{(i)}) + (1 - \kappa)x^{(i)}$ |
| 7. If $\tau^{(i)} = ||x^{(i+1)} - x^{(i)}||_2 < \eta \sigma$ |
| $\sigma = \rho \sigma$. |

while $\sigma > \sigma_{\text{min}}$ .

2.2.2.4 ISL0 Algorithm

By compiling the results presented so far we propose the “improved SL0” (ISL0) optimization strategy in Table 2.4. In short, this is a modified version of ASA-SL0 with the steepest ascent direction replaced by $g(x^{(i)}) - x^{(i)}$. The initialization steps (steps 1-2) are identical to the O-SL0 algorithm [34]. In each iteration, along the ascent-direction $g(x) - x$ we find the step-length $\kappa$ using the standard backtracking strategy (step 4-5) [23]. The inner-iteration for maximizing $F_\sigma$ for a given $\sigma$ terminates when $\tau^{(i)} < \eta \sigma$, see step 7. Hence for any $\eta$ more accurate solutions are sought for smaller $\sigma$ values, and this saves the computation time. On the other hand, a smaller $\eta$ results an increase in the number of inner iterations, and hence an increase in the computation time. The experimental results show that the performance of ISL0 remains almost unaffected for a wide range of $\eta$. Upon convergence of each inner iteration we lower $\sigma$ by a factor $\rho$. Our experimental results, see Section 2.3.3, suggest that in ISL0 we can choose $\rho$ between 0.9 and 0.4. Note that choosing a smaller $\eta$ and a larger $\rho$ increases the reliability in the cost of computation time. Our experimental study suggests that choosing $\rho = 0.5$, and $\eta = 0.5$ makes a good tradeoff. ISL0 stops when $\sigma$ becomes smaller than a predefined threshold $\sigma_{\text{min}} = \sigma_*$ derived in Section 2.2.1.

Evaluation of the function $g(x)$ in step 5 requires inverting the matrix $W(x^{(i)})$, see (2.42), which might be close to singular. To cope with this issue we use a regularization and use $W(x^{(i)}) + \hat{\epsilon} I_n$ instead of $W(x^{(i)})$. Where, $\hat{\epsilon}$ is a small number, whose value depends
on digital computer precision. In our experiments, we set $\hat{\epsilon} = 10^{-10}$.

Note that the main computation cost of ISL0 is around equation (2.42). It is readily verified that

$$
\tilde{y}^{(i)} := [\Phi W^{-1}(x^{(i)})\Phi']^{-1} y
= \arg\min_v \|\Phi W^{-1}(x^{(i)})\Phi' v - y\|^2_2. \tag{2.49}
$$

For the large scale problems it is more efficient to compute $\tilde{y}$ by solving the optimization problem (2.49) using an iterative algorithm like a conjugate gradient method, see e.g. [40]. One can use $\tilde{y}^{(i-1)}$ to initialize the iterative algorithm. This approach has been used in the simulation discussed in Section 2.3.4.

2.3 Experiments

In this section, we validate the results presented so far using numerical simulations. Three types of experiments are presented: (i) exact signal recovery when $e = 0$, (ii) recovery of the power law decaying signals, (iii) approximate signal recovery using noisy measurements, and (iv) application of ISL0 on real world data. The performance of ISL0, O-SL0 and ASA-SL0 algorithms are compared with LP (basis pursuit) [22], hyperbolic tangent based approximation (Tangent) [32], Gradient Projection for Sparse Reconstruction (GPSR-BB) [41] and orthogonal matching pursuit (OMP). The simulations are performed in MATLAB7 environment using an Intel Core 2 Duo, 2.66 GHz processor with 2GB of memory, under Mac OS X Version 10.5.5 operating system.

2.3.1 Signal recovery from noise-free measurements

In this experimental setup, we consider $e = 0$. We randomly choose the support of a $k$-sparse signal $x_0$, and set the non-zero components randomly to either $+1$ or $-1$. The matrix $\Phi$ is generated as follows. First we construct a random matrix $A \in \mathbb{R}^{m \times n}$ by drawing its entries from a standard normal density. The entries in $A$ are mutually independent. Next, we construct $\Phi$ such that the rows of $\Phi$ constitute an orthonormal basis of the row-space of $A$.

We say the signal $x \in \mathbb{R}^n$ is recovered exactly if the recovered signal $x^r$ satisfies

$$
\|x - x^r\|_\infty < \varepsilon. \tag{2.50}
$$
Figure 2.1: The minimum number of measurement $m$ necessary to recover a $k$ sparse signal in dimension $n = 1000$. For each $k$, 100 different $k$-sparse signals are constructed. Then, the smallest $m$ is selected that can recover at least 95 signals.

Except for GPSR-BB, the value of $\varepsilon = 10^{-5}$.

According to (2.36), the value of $\sigma_{\text{min}}$ should be zero in noiseless cases. Hence, for ISL0, O-SL0 and ASA-SL0 we set a small $\sigma_{\text{min}} = 10^{-5}$. The other parameters of ASA-SL0 and ISL0 algorithms are fixed to $\eta = 0.5$, and $\gamma = 0.5$. For O-SL0, we follow the recommendations in [34], and for each $\sigma$ we take 3 steepest ascent steps. As suggested in [32], we use the Trust-Region\(^2\) approach to solve the optimization problem arising in hyperbolic tangent based approximation.

Figure 2.1 shows the plot of the minimum number of measurements $m$ needed by different algorithms to recover a $k$ sparse signal in dimension $n = 1000$ as a function of $k$. For each $k$, we simulate 100 different $k$-sparse signals. Subsequently, the smallest $m$ is selected that enables recovery at least 95 times. As expected, when $k$ increases, a larger $m$ is necessary. Note that ISL0 performs better than other algorithms. Moreover, when LP performs almost similarly to ISL0, the performance of GPSR-BB is relatively poor. Hyperbolic tangent based approximation performs similarly to LP and ISL0 for small $k$. However as $k$ increases the performance degrades. ASA-SL0 shows performance improvement compared to O-SL0. Figure 2.2 shows the plot of the recovery rates achieved

\(^2\)The standard procedures in Matlab Optimization Toolbox is used.
2.3. Experiments

Figure 2.2: The percentage of 100 input signals correctly recovered as a function of the sparsity level $k$ for fixed measurements $m = 400$ in signal dimension $n = 1000$. By different methods as functions of $k$. Here $n = 1000$, $m = 400$. Results are averaged over 100 independent experiments. We set $\rho = 0.95, 0.05$ and 0.125 for O-SL0, ASA-SL0 and ISL0, respectively. With this choice the computation time of O-SL0, ASA-SL0 and ISL0 are the same on average. Note that ISL0 performs better than other algorithms. LP performs almost similarly to ISL0. Figure 6.4 shows a significant performance improvement of ISL0 over ASA-SL0 and O-SL0. According to our experience, the performance of ASA-SL0 can be improved significantly by increasing $\rho$ and decreasing $\eta$. However this improvement comes at the cost of a very large number of iterations, and hence the computation time. Figure 2.3 shows a comparison of required computation time. We set $\rho = 0.3$ for ISL0 (see discussion around Figure 2.3.3). This is done to ensure that all SL0 variants have the same capability for a given sparsity. We define the "capability" as the number of non-zero elements which gives 100% success in recovery. Except GPSR-BB all algorithms give 100% recovery in the entire range. GPSR-BB gives 80% recovery for $k = 100$. The computation time of the hyperbolic tangent based $\ell_0$ approximation method is much higher than other methods, and is not shown.
2. Single Measurement Vector Problem

2.3.2 Recovery of power-law decaying signals

Estimation results for power-law decaying sparse signals are shown Figure 2.4. We consider a signal $s$ such that

$$|s_i| = c i^{-p}, \ i = 0, 1, \ldots, k - 1.$$  \hfill (2.51)

Here we take $c = 10^4$. The sign of $s_i$ is chosen randomly. To generate a $k$ sparse $x$ using $s$, we choose a random support of cardinality $k$, and put $\{s_i\}_{i=0}^{k-1}$ there. The value of $\epsilon = 10^{-3}$ in (2.50). The matrix $\Phi$ is generated in a manner similar to that for the simulations described in Section 2.3.1. In order to equalize the computation time for ASA-SL0, ISL0, O-SL0, we choose $\rho = 0.98, 0.8$ and 0.25 for O-SL0, ISL0 and ASA-SL0, respectively. Moreover, we set $\eta = 0.1$ for ASA-SL0 and ISL0. Figure 2.4(a) shows the results for $p = 2$, and Figure 2.4(b) gives the results for $p = 3$. In both cases ISL0 maintains its superiority over other methods.
2.3. Experiments

Figure 2.4: The percentage of 100 input signals correctly recovered as a function of the sparsity level \( k \) for fixed measurements \( m = 350 \) in signal \( x \) of dimension \( n = 1000 \). The coefficients of nonzero \( x \)'s follow power-law. (a) \( p = 2 \), (b) \( p = 3 \).
Figure 2.5: Average SNR (from 100 runs) of the signals recovered by ISL0, ASA-SL0 and O-SL0 as a function of $\rho$, where $m = 400$, and $n = 1000$. Results for different values of $k$ and $\mu$ are shown. (a) Average SNR as function of $\rho$ for $\mu = 0.02$, $\sigma_{\min} = 0.04$, (b) Average SNR as function of $\rho$ for $\mu = 0.05$, $\sigma_{\min} = 0.1$. 
2.3.3 Approximate Signal Recovery in Noisy Environment

In these experiments the data are generated according to the model given by Assumptions 3 and 4. The matrix $\Phi$ is generated in a manner similar to that for the simulations described in Section 2.3.1. We maintain $\sigma^2 = 1$, allow $\mu > 0$, and denote $k = np$. We use $\eta = 0.5$ for both ASA-SL0 and ISL0. For SL0 we take 3 steepest ascent steps for every $\sigma$. In addition the same $\sigma_{\text{min}}$ is used for O-SL0, ASA-SL0 and ISL0. The value of $\sigma_{\text{min}}$ is determined using the concepts proposed in Section 2.2.1.3 and 2.2.1.4. In this section we use the signal-to-noise ratio (SNR) of the recovered signal as the performance measure. If the recovered signal is $x^r$ then the SNR in dB is defined as

$$\text{SNR} = 20 \log_{10}\left(\frac{\|x_0\|_2}{\|x_0 - x^r\|_2}\right).$$

First, we study the dependence of ISL0 on $\rho$. Recall that $\rho$ determines at what rate $\sigma$ is lowered in ISL0, ASA-SL0 and O-SL0. It is a crucial tuning factor. In Figure 2.5(a)-2.5(b) we plot the average SNR of the signal recovered by ISL0, ASA-SL0 and O-SL0 as functions of $\rho$. Figure 2.5(a) shows the results when $\mu = 0.02$, while Figure 2.5(b) shows the results for $\mu = 0.05$. In both Figures we consider two different values of $k$. As can be seen in Figure 2.5(a)-2.5(b), ISL0 and ASA-SL0 are somewhat less sensitive to $\rho$ compared to O-SL0. This is because both ISL0 and ASA-SL0 repeat the internal loop until convergence. O-SL0, on the other hand performs better if $\rho$ is large. A large $\rho$ means $\sigma$ is decreased gradually. Hence 3 inner iterations are sufficient. In fact, when $\mu = 0.05$, O-SL0 outperforms ASA-SL0 if we take $\rho > 0.5$. Since ISL0 is practically insensitive to any increase in $\rho$ beyond 0.3 the choice $\rho = 0.5$ gives the minimal computation time without compromising the performance.

Next we study the sensitivity of the performance of ISL0 with respect to change in $\eta$, which determines how frequently ISL0 lowers the value of $\sigma$. Figure 2.6(a)-2.6(b) show the dependency of ISL0 on $\eta$, where we notice that for a broad range of $\eta$ the performance of ISL0 is nearly unaffected by $\eta$. Thus, we choose $\eta = 0.5$ to save computation time while keeping the likelihood of getting trapped in a local maxima to a low level.

Next we demonstrate the validity and utility of the results presented in Section

---

3Power-law decay is very common, e.g. wavelets coefficients of signals follow such pattern. Unlike, (2.51), a typical power-law decaying signal is not strictly sparse, i.e. $|s_k| = c i^{-p}$, $\forall i$ for a typical power-law decaying $s$. However, our experimental results show that none of the methods perform well in this case (particularly if $p > 2.5$).
Figure 2.6: Average SNR (from 100 runs) of the signals recovered by ISL0 as a function of $\eta$, where $m = 400$, and $n = 1000$. Results for different values of $k$ and $\mu$ are shown. (a) Average SNR as a function of $\eta$ for $\mu = 0.01$. $\sigma_{\min} = 0.01$, (b) Average SNR as a function of $\eta$ for $k = 100$. 
2.3. Experiments

Figure 2.7: The experimental RMSE of the ISLO iterate $x^{(i)}$ and the its theoretical upper bound using (2.22), with $n = 1000$, $k = 100$, and $m = 400$. (a) RMSE at successive iterations for $\mu = 0.01$, (b) RMSE at successive iterations for $\mu = 0.05$. 
2. Single Measurement Vector Problem

2.2.1. In Figure 2.7 we compare the root mean square error \( \| x_0 - x^{(i)} \|_2 / \sqrt{n} \) associated with the \( i \)-th iterate \( x^{(i)} \) against the theoretical bound given by Lemma 1. While computing the upper bound given by the right hand side of (2.22), we replace \( \| e \|_2 \) by its unbiased estimate \( \sqrt{m \mu} \), as \( \| e \|_2 \) is unknown in practice. It can be shown that the difference between the right hand side and the left hand side of (2.23) has a larger mean and a larger second order moment when \( \mu \) increases. Consequently, the difference between the left hand side and the right hand side of (2.22) is expected to increase. Nevertheless, the findings in Figure 2.7 shows that the bound given by (2.22) is reasonably tight for \( \mu = 0.05 \).

Figure 2.8(a)-2.8(b) justifies the effectiveness of proposed stoping criterion derived in Section 2.2.1.3 and 2.2.1.4. In Figure 2.8(a) we plot the upper bound \( a_\sigma \) on \( h_\sigma \) using the Monte-Carlo simulation approach given by (2.36). Here we set \( \epsilon = 0.2 \). Hence the bounds in Figure 2.8(a) are actually 96% confidence bounds. Note in Figure 2.8(a) that, the value of \( \sigma_{\min} \) for which the theoretical bound on \( h_\sigma \) attains its minimum value depends on \( \mu \). When we set \( \mu = 0.02 \) then \( \sigma_{\min} = 0.04 \). Now in Figure 2.8(b) we see that when \( \mu = 0.02 \), the experimental SNR of the recovered signal indeed attains its peak for \( \sigma_{\min} = 0.04 \). This phenomenon repeats for \( \mu = 0.05 \) as well, when \( \sigma_{\min} = 0.1 \). This is interesting to note that optimal value of the terminating \( \sigma \) depends significantly on \( \mu \). Additionally, the figures also demonstrate that our proposed stopping criteria is equally applicable for ASA-SL0.

In Figure 2.9 we analyze the overall performance of ISL0 and compare it with some other algorithms. The value of \( \sigma_{\min} \) chosen using the stopping criterion derived in Section 2.2.1. The same value of \( \sigma_{\min} \) is used for O-SL0, ASA-SL0 and ISL0. We take \( \rho = 0.3 \) for ISL0 and ASA-SL0. For O-SL0 we set \( \rho = 0.9 \). With this choice, ISL0 and O-SL0 have similar computation time.

For LP we use \( \ell_1 \)-Magic package and solve
\[
\min \| x \|_1, \quad \text{subject to } \| \Phi x - y \|_2 \leq \mu \sqrt{m},
\]
as recommended in [22]. For GPSR-BB we adopt the recommendation in [41], and solve
\[
\min_{x} \ \zeta \| x \|_1 + \frac{1}{2} \| \Phi x - y \|_2^2.
\]
A wide range of numerical experiments for \( \mu = 0.01 \) to 0.05 suggests that \( \zeta = 0.08 \) is a good choice. For orthogonal matching pursuit we use the SparseLab package [42].

Note in Figure 2.9 that ISL0 outperforms the other algorithms. We note in passing that the performance of ASA-SL0 can be made similar to ISL0 by taking a sufficiently large
Figure 2.8: The utility of the stopping criterion developed in (Sections 2.2.1.3 and 2.2.1.4), where $k = 100, m = 400, n = 1000$. (a) 96% confidence bound $a_\sigma$ on $h_\sigma$ versus $\sigma_{\min}$, (b) Average SNR (on 100 runs) versus $\sigma_{\min}$. 
2. Single Measurement Vector Problem

ρ. For instance, if we run ASA-SL0 with ρ = 0.95 in the examples of Figure 2.9, then ASA-SL0 performs almost as good as ISL0. However, this performance improvement comes with a significantly increased computation time.

2.3.4 Application of ISL0 in single pixel imaging

To apply ISL0 in single pixel imaging we resort to the total variation (TV) minimization framework. When x represents an image, it is conventionally expressed as a 2-D array. Given an injective linear transformation $y = \Phi \cdot \text{vec}(x)$ of the image $x$, the TV approach minimizes the $\ell_1$ norm of the complex representation of the gradient image $\nabla x$ defined as

$$[\nabla x]_{i,j} = (x_{i+1,j} - x_{i,j}) + i (x_{i,j+1} - x_{i,j}).$$

Therefore, the TV approach solves

$$\min_x \sum_{i,j} |(\nabla x)_{i,j}| \text{ subject to } \Phi \cdot \text{vec}(x) = y.$$  

ISL0 can be extended readily to handle the TV framework where we can use the techniques proposed in this paper to minimize the Gaussian approximation of the $\ell_0$ norm $\nabla x$, and solve

$$\max_x \sum_{i,j} \exp\{-|\nabla x|_{i,j}^2/(2\sigma^2)\}, \text{ subject to } \Phi \cdot \text{vec}(x) = y.$$  

Figure 2.10(b) illustrates a target image and reconstruction using ISL0. The images were taken$^4$ by a single pixel camera prototype using $n = 64 \times 64$. Figures 2.10(c) comparison of the recovery time using standard LP based TV minimization and ISL0. Note that ISL0 requires much lower time compared to LP.

2.4 Conclusion

In this chapter, we visit some popular algorithms that deals with recovering sparse signals from a small number of measurements formed by computing the inner products of the signal with the rows of a measurement matrix. To this end we found that smoothed $\ell_0$ (SL0) algorithm outperforms other algorithms. We then propose some improvements over the SL0 approach. The SL0 algorithm belongs to the class of algorithms which minimize the $\ell_0$ norm.

$^4$The dataset is available at http://dsp.rice.edu/cscamera. The dataset contains both $y$ and $\Phi$. 
2.4. Conclusion

Figure 2.9: Average SNR’s (over 100 runs) versus measurements \((m)\) with \(n = 1000, k = 100\). (a) Average SNR versus \(m\) for fixed \(\mu = 0.02\), (b) Average SNR (on 100 runs) versus \(m\) for fixed \(\mu = 0.05\).
2. Single Measurement Vector Problem

Figure 2.10: Single-pixel photo album. For ISL0, parameters are fixed at $\rho = 0.5$, $\eta = 0.5$.
(a) Target image of a ‘R’, (b) ISL0 reconstruction from 50% random measurements, (c) Computation time comparison.
of the recovered signal subject to the linear constraints given by the measurements. SL0 approach approximates the $\ell_0$ norm by the Gaussian family of functions with a variable variance $\sigma^2$ and solve a sequence of optimization problems. In Section 2.2.1.1 we have derived a $O(\sqrt{n})$ bound on the estimation error, which can be evaluated at the run-time. Subsequently, we propose a method to automatically calculate the parameter $\sigma_{\text{min}}$, the smallest $\sigma$ which has to be used by the SL0 proposed in Section 2.2.1.2. This can be used to avoid the problems caused by the non-smooth nature of the underlying cost function for smaller values of $\sigma$. Finally, we have proposed a convex-concave procedure to optimize the SL0 cost function. This algorithm, called ISL0, is similar to the Newton algorithm. ISL0 is tested on numerically simulated data, as well as on real-world experimental data.
Chapter 3

Structured Sparse Signal Recovery

3.1 Introduction

The previous chapter described some sparse signal recovery algorithms where we assumed that the locations of the nonzero components of the sparse signal $x$ are independent from each other. In some applications, the sparse signal models exhibit additional structure in the form that there exists some interdependency between nonzero signal component locations. For example, let us consider the class of multiband signals, whose frequency support resides within several continuous intervals, spread over a wide spectrum [43, 44]. Then the resulting sparse signal has the structure that the nonzero coefficients are occurring in clusters. In some other applications, the resulting data model generates multiple sparse signals where all sparse signals have nonzero entries at the same locations. Recently some new CS theory and algorithms show that the number of measurements $m$ necessary to recover the sparse signal can reduce significantly by exploiting the dependencies among the nonzero locations [45, 46, 47, 20]. In this chapter we propose some modifications of ISL0 algorithm so that it can take advantage from underlying structures of sparse signal. Subsequently, we modify ISL0 further so that it can work with complex valued signals. Complex valued sparse signals arise in many practical applications, e.g., direction of arrival estimation [48], frequency estimation [49], radar signal processing [50]. As expected intuitively, a simple modification of ISL0 allows us to work with complex valued signals.
3.1.1 Contributions and Motivation

Most of the methods that exploit the structure of the sparse signal achieve a good recovery rate for a relatively small value of \( k \). However, to the best of our knowledge, it has not been possible to achieve good recovery rate when \( k \) is close to the upper theoretical bound. This in some applications can lead to limitations. For example, in source separation application \( k \) denotes the number of sources that a fixed sensor array can handle, and being able to recover signals with larger \( k \) leads to a significant improvement. Another important issue is the performance of the algorithms in presence of measurement noise. In the numerical simulations it was found that most of the algorithms suffer from performance degradation in presence of noise. We propose some modifications of ISL0 to overcome the limitations. We have found that in most cases (see Section 3.5) ISL0 outperforms other existing algorithms. In some structured sparse signals, like multiple measurement vectors problem, the proposed algorithm can achieve the theoretical limit of sparsity \( k \) with a very high likelihood.

3.2 Multiple Measurement Vectors Problem

3.2.1 Background

Let us consider recovering a jointly row sparse matrix \( X_0 \in \mathbb{R}^{n \times r} \) from \( Y \) formed by pre-multiplying \( X_0 \) by \( \Phi \):

\[
Y = \Phi X_0. \tag{3.1}
\]

By the term “jointly row sparse” we mean that only \( k \) rows of \( X_0 \) are non-zero. The problem of recovering jointly sparse matrices is often called the “multiple measurement vectors” (MMV) problem, which arises naturally in source localization, neuromagnetic imaging [20], and equalization of sparse communication channels [51], and several other applications [52]. The following lemma [20] gives the sufficient conditions for the existence of the unique solution to the MMV problem.

**Lemma 4.** Let rank \((Y) = r \leq m\), and every \( m \) columns of \( \Phi \) forms a basis of \( \mathbb{R}^m \). Then a solution to (3.1) with \( k \) nonzero rows is unique provided that \( k \leq \left\lceil \frac{(m + r)}{2} \right\rceil - 1 \) (where \( \lceil a \rceil \) is the smallest integer not less than \( a \)).
Researchers spanning a diverse range of viewpoints have advocated mixed-norm minimization approach to solve the MMV problem. Here one solves

$$\min_{X \in \mathcal{X}} \|X\|_{p,q}, \quad \mathcal{X} := \{X \in \mathbb{R}^{n \times r} : Y = \Phi X\},$$

(3.2)

for various combinations of $p$ and $q$, where the mixed norm $\|X\|_{p,q}$ is defined as

$$\|X\|_{p,q} = \left( \sum_{j=1}^{n} \left( \sum_{i=1}^{r} |X[j,i]|^p \right)^{q/p} \right)^{1/q}.$$  

(3.3)

Cotter et al. [20] use $p = 2, q \leq 1$, Tropp [53] analyzes for $p = 1, q = \infty$ and Elder et al. [46] use $p = 2, q = 1$. In [46] the sufficient conditions for MMV recovery by using $p = 2$ and $q = 1$ are presented and it is shown that the problem with $p = 2$ and $q = 1$ can be posed as a convex optimization problem. Berg et al. [52] analyze some properties when $p = q = 1$ and $p = 2, q = 1$. In a broad sense, all these algorithms offer some form of convex relaxation to the case $p = 2, q = 0$, which guarantees unique recovery under the assumptions of Lemma 4. Unfortunately, this case of $p = 2, q = 0$ leads to a nonconvex problem. Compressive MUSIC (CS-MUSIC) [54, 55] is a joint sparse algorithm that creates a link between CS and array signal processing. Model-based CS (Model-CS), proposed in [56], provides a guideline on how to create structured signal recovery algorithms with provable performance guarantees. However, both CS-MUSIC and Model-CS assume that signal sparsity is known in advance, which is unknown in practice. Simulation results demonstrate that the algorithms can not perform equally when signal sparsity is unknown.

### 3.2.1.1 $\ell_{1,1}$-norm Solution

Let $p = 1, q = 1$ in (3.2). Hence the optimization problem becomes

$$\min_{X \in \mathcal{X}} \|X\|_{1,1}.$$  

(3.4)

The advantage of the formulation in (3.4) is that it is a convex optimization problem and can be solved using linear programming [57]. It has been shown in [57] that if some conditions hold then the solution of (3.4) is unique.

### 3.2.1.2 Reduce MMV and Boost

Reduce MMV and Boost (ReMBo) [58], is an algorithm that reduces an MMV problem to a series of SMV problems. To describe ReMBO, we need the following definition
Definition 6. For a given vector $x \in \mathbb{R}^n$ the support function is defined as
\[ S(x) = \{ i \in \{1, \cdots, n\} : |x_i| \neq 0 \}. \]

Similarly support function of a matrix $X \in \mathbb{R}^{n \times r}$ is
\[ S(X) = \{ i \in \{1, \cdots, n\} : \|X[i,:]\|_2 \neq 0 \} \]
which describes the location of nonzero rows in $X$.

Then ReMBo algorithm works based on the following lemma

Lemma 5. [58] Let $\Phi \in \mathbb{R}^{m \times n}$ be a URP matrix and $Y = \Phi X_0$, where $X_0 \in \mathbb{R}^{n \times r}$ has only $k$ nonzero rows. Also $m \geq 2k$. Let $w$ be a random vector of length $r$ with an absolute continuous distribution. Define the vectors $y = Yw$ and $\bar{x} = X_0w$. Consider the SMV system $y = \Phi x$. Then
1. For every realization of $w$, the vector $\bar{x}$ is the unique $k$-sparse solution of the SMV system.
2. $S(x) = S(X)$ with probability one.

ReMBo proceeds by taking a random vector $w \in \mathbb{R}^r$ and combining the individual observations in $Y$ into a single weighted observation $y := Yw$. Subsequently, it solves a single measurement vector problem $\Phi x = y$ by using a suitable algorithm and checks if the computed solution $\bar{x}$ is sufficiently sparse. If not, the above steps are repeated with a different $w$. The algorithm stops when a maximum number of trials is reached. Once the ReMBo algorithm has found a sufficiently sparse solution it solves a least-squares problem using only those columns in $\Phi$ corresponding to the support $x$.

3.2.2 Joint $\ell_{2,0}$ Approximation Algorithm (JLZA)

In the following we develop an algorithm to solve (3.2) with an approximation to $\|X\|_{2,0}$. One can write $\|X\|_{2,0}$ as
\[ \|X\|_{2,0} = \sum_{j=1}^{n} I(\|X[j,:]\|_2) \]
where $I$ is the indicator function
\[ I(\alpha) = \begin{cases} 1, & |\alpha| > 0 \\ 0, & \text{otherwise.} \end{cases} \]
(3.5)
Consequently, the function
\[ \hat{F}_\sigma(X) = \sum_{j=1}^{n} f_\sigma(\|X[j,:]\|_2). \] (3.6)
behaves like \( n - \|X\|_{2,0} \) when \( \sigma \to 0 \), where \( f_\sigma(\alpha) \) is defined in (2.18). Then following the strategic of (2.20) one can consider the approximate way of reformulating (3.2) for \( p = 2 \) and \( q = 0 \):
\[ X_* = \arg \max_{X \in \tilde{X}} \hat{F}_\sigma(X). \] (3.7)
where we take \( \sigma \to 0 \). \( \hat{F}_\sigma(X) \) has many local maxima for small values of \( \sigma \). As discuss around (2.20), we take a large \( \sigma \) initially and solve (3.7). Subsequently, \( \sigma \) is reduced by some small factor and (3.7) is solved again.

3.2.2.1 Algorithm Derivation

The Lagrangian \( L_\sigma(X, \Upsilon) \) for (3.7) is
\[ L_\sigma(X, \Upsilon) = \hat{F}_\sigma(X) + \text{Trace}\{\Upsilon'(\Phi X - Y)\}. \] (3.8)
Let us define the matrix \( \frac{\partial L_\sigma}{\partial X} \), such that \( \left[ \frac{\partial L_\sigma}{\partial X} \right]_{ij} = \frac{\partial L_\sigma}{\partial X[i,j]} \). Now (3.7) implies that the stationary point \((X_*, \Upsilon_*)\) of \( L_\sigma(X, \Upsilon) \) satisfies,
\[ \frac{\partial L_\sigma(X_*, \Upsilon_*)}{\partial X} = \frac{\partial \hat{F}_\sigma(X_*)}{\partial X} + \Phi' \Upsilon_* = 0, \] (3.9)
\[ \frac{\partial L_\sigma(X_*, \Upsilon_*)}{\partial \Upsilon} = \Phi X_* - Y = 0. \]
Also it is readily verified that
\[ \left[ \frac{\partial \hat{F}_\sigma(X)}{\partial X} \right]_{ij} = -\frac{f_\sigma(\|X[i,:]\|_2)}{\sigma^2} X[i,j] \]
Hence,
\[ \frac{\partial \hat{F}_\sigma(X)}{\partial X} = -\frac{1}{\sigma^2} W(X)X. \] (3.10)
where \( W(X) = \text{diag}\{f_\sigma(\|X[1,:]\|_2), \cdots, f_\sigma(\|X[n,:]\|_2)\} \). Then (3.9)-(3.10) gives
\[ X_* = \sigma^2 W^{-1}(X_*) \Phi' \Upsilon_. \] (3.11)
Substitute this for \( X_* \) in the second equation of (3.9), then solve for \( \Upsilon_* \), and substitute the solution to \( \Upsilon_* \) in (3.11) to get
\[ X_* = W^{-1}(X_*) \Phi' \left[ \Phi W^{-1}(X_*) \Phi' \right]^{-1} Y. \] (3.12)
3.2. Multiple Measurement Vectors Problem

Table 3.1: JLZA Algorithm

<table>
<thead>
<tr>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set ( X^{(0)} = \Phi' (\Phi \Phi')^{-1} Y ).</td>
</tr>
<tr>
<td>2. Set ( \sigma = \max_i | X^{(0)} |_1 ) and ( \rho, \eta, \gamma \in [0, 1) ).</td>
</tr>
</tbody>
</table>

\[ \text{repeat} \]

| 3. Set \( \kappa = 1 \). |
| 4. while \( \hat{F}_\sigma (\kappa \bar{\zeta}(X^{(i)}) + (1 - \kappa)X^{(i)}) < \hat{F}_\sigma (X^{(i)}) \) \( \kappa = \gamma \kappa \). |
| 5. \( X^{(i+1)} = \kappa \bar{\zeta}(X^{(i)}) + (1 - \kappa)X^{(i)} \). |
| 6. \( \tau^{(i)} = \| X^{(i+1)} - X^{(i)} \|_2 < \eta \sigma \) then \( \sigma = \rho \sigma \). |

\[ \text{until} \quad \sigma \geq \sigma_{\min} \]

Equation (3.12) is nonlinear, and it can be used in a fixed point iteration. Let us define the map \( \bar{\zeta} : \mathbb{R}^{n \times r} \rightarrow \mathbb{R}^{n \times r} \) such that

\[ \bar{\zeta}(X) = W^{-1} (X) \Phi' [\Phi W^{-1} (X) \Phi']^{-1} Y. \]  

Then using the concept in Lemma 3, it can be shown that \( \hat{F}_\sigma \) is an increasing function along the direction \( \bar{\zeta}(X) - X \). Hence the algorithm will converge at least to a local maxima.

Based on this idea, the JLZA algorithm is given in Table 3.1. The algorithm is similar to the ISL0 algorithm in Table 2.4. In the algorithm, \( X^{(i)} \) denotes the value of \( X \) updated at \( i \)-th iteration. For maximizing \( \hat{F}_\sigma \) we use \( \bar{\zeta}(X) - X \) as ascent-direction. JLZA has same tuning parameters as ISL0 in Table 2.4. A short discussion about tuning parameters selection can be found in Section 2.2.2.4. The final value of \( \sigma_{\min} \) depends on the noise level. For noiseless sparse recovery, \( \sigma_{\min} \) is fixed near to zero (\( \sigma_{\min} = 10^{-5} \)). However, in noisy case, \( \sigma_{\min} \) should be left at some smaller value as the system can not approximate the optimal \( X \) exactly and the solution fluctuates randomly (we shall reformulate JLZA for noisy environment in the following section.

3.2.2.2 Extension to Noisy Case

The algorithm proposed in Section 3.2.2.1 performs well when signal is noiseless or signal is corrupted by small amount of noise. However, in presence of large noise we need a more robust algorithm. To this aim we extend JLZA for noisy case. In noisy environment
one can modify the data model in (3.1)

\[ Y = \Phi X + E. \]  

(3.14)

where \( \Phi \in \mathbb{R}^{m \times n}, X \in \mathbb{R}^{n \times r} \). Also \( E \in \mathbb{R}^{m \times r} \) is the additive noise. We consider the following way of solving \( X \) given \( Y \) in (3.14):

\[ X_\star(\sigma) = \arg \max_X L_\sigma(X), \]

\[ L_\sigma(X) := \tilde{F}_\sigma(X) - \lambda \| Y - \Phi X \|_F^2. \]  

(3.15)

for small values of \( \sigma \), where \( \tilde{F}_\sigma(X) \) is defined in (3.6). The parameter \( \lambda \) controls the tradeoff between the sparsity of the signal and the residual energy. This strategy has been used in a number of sparse signal representation problems [8, 59].

Similar to JLZ A (Section 3.2.2), \( X_\star \) admits a closed-form solution as \( \sigma \to \infty \). To see this, first note that \( \lim_{\sigma \to \infty} \tilde{F}_\sigma(\alpha) = 1 \). Using this in (3.15), we have

\[ \lim_{\sigma \to \infty} X_\star(\sigma) = \Phi' (\Phi \Phi')^{-1} Y. \]

Hence, we take a large \( \sigma \) initially and solve (3.15). Subsequently, \( \sigma \) is reduced by some small amount and (3.15) is solved again. The procedure is repeated until a convergence criterion is satisfied. The following Lemma shows that the extended JLZA will converge at least to a local minima.

**Lemma 6.** Define the mapping \( \zeta : \mathbb{R}^{n \times r} \to \mathbb{R}^{n \times r} \) such that

\[ \zeta(X) = 2\lambda \left[ W(X)/\sigma^2 + \lambda \Phi' \Phi \right]^{-1} \Phi' Y, \]  

(3.16)

where \( W(X) \) is a diagonal matrix:

\[ W(X) = \begin{bmatrix} f_\sigma(\|X[1,:]\|_2) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f_\sigma(\|X[n,:]\|_2) \end{bmatrix}. \]  

(3.17)

Then \( X_\star(\sigma) = \zeta\{X_\star(\sigma)\} \). In addition, for any \( X \) there exits a real-valued scalar \( \kappa \geq 0 \) such that

\[ L_\sigma\{\kappa \zeta(X) + (1 - \kappa)X\} \geq L_\sigma(X). \]  

(3.18)

The proof can be derived easily using the results in Section 2.2.2.3. Also a more general form of the Lemma 6 has been stated in Lemma 8. The proof of Lemma 8 is provided in Appendix A.2.
3.2. Multiple Measurement Vectors Problem

Table 3.2: Extension of JLZA Algorithm for Noisy Environment

<table>
<thead>
<tr>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set $X^{(0)} = \Phi'(\Phi\Phi')^{-1}Y$.</td>
</tr>
<tr>
<td>2. Set $\sigma = \max_i |X^{(0)[i,:]}|<em>1$, $\lambda \in [1, 10^{10}]$ and $\rho, \eta, \gamma \in [0, 1)$, $\sigma</em>{\text{min}} \in [0.1, 10^{-4}]$.</td>
</tr>
</tbody>
</table>

repeat

3. Set $\kappa = 1$.

4. while $L_\sigma \{ \kappa \zeta(X^{(i)}) + (1 - \kappa)X^{(i)} \} < L_\sigma(X^{(i)})$

   $\kappa = \gamma \kappa$.

end

5. $X^{(i+1)} = \kappa \zeta(X^{(i)}) + (1 - \kappa)X^{(i)}$

6. If $\tau^{(i)} = \|X^{(i+1)} - X^{(i)}\|_2 < \eta \sigma$ then $\sigma = \rho \sigma$.

while $\sigma \geq \sigma_{\text{min}}$.

Based on Lemma 6 the proposed algorithm is given in Table 3.2. We start with $X^{(0)} = X_*(\infty)$, and set $\sigma = \max_i \|X^{(0)[i,:]}\|_1$ [34]. The value of $\lambda$ depends on the noise level. See [48, 21] for a discussion on methods for choosing $\lambda$. Extension of JLZA has similar tuning parameters as ISL0 in Table 2.4. The inner-iteration for minimizing $L_\sigma$ for a given $\sigma$ terminates when $\tau^{(i)} < \eta \sigma$, (see Step 6). However, the internal loop for a given $\sigma$ can be repeated a fixed and small number of times (for example, 3 times) which for increasing the speed [34]. Upon convergence of each inner iteration, we lower $\sigma$ by a factor $\rho$ (Step 6). The final value of $\sigma_{\text{min}}$ depends on the noise level. A direction of choosing $\sigma_{\text{min}}$ can be found in Section 2.2.1.2.

The computational load associated with the proposed algorithm is mainly due to the computation of $\zeta(X)$. Evaluating $\zeta(X)$ as given in (3.16) requires $O(n^3 + mn^2 + rmn)$ flops. Most of it is incurred while inverting an $n \times n$ matrix. A significant reduction in computation is achieved by using

$$
\zeta(X) = W^{-1}\Phi'[I_m/(\lambda\sigma^2) + \Phi W^{-1}\Phi']^{-1}Y, \quad (3.19)
$$

which is obtained by applying the matrix inversion lemma to (3.16). Using (3.19), one requires $O(nmr^2 + m^3 + nmr)$ flops to compute $\zeta(X)$ (typically $m \ll n$). However, this represents the computational load for only one iteration. Although the number of iterations needed by JLZA cannot be predicted in advance, it takes less than 15 iterations in practice.
3.2.2.3 MMV with Multiple Measurement Matrices

Consider a set of joint sparse problems of the form

\[ Y_j = \Phi_j X_j + E_j, \; j \in \{1, 2, \ldots, J\} \]  

(3.20)

where each \( X_j \in \mathbb{R}^{n \times r} \) is joint row-sparse, and \( \forall i \neq j \)

\[ \Phi_i \neq \Phi_j. \]

An interesting case arises when \( X = [X_1 \; X_2 \; \cdots \; X_J] \)

(3.21)
is also joint row-sparse. We will call this problem MMV with multiple measurement matrices. The problem arises in some applications, e.g. direction of arrival estimation using wideband signals, coherent frequency estimation, etc. More details about these applications will follow in the later chapters.

One can solve \( X \) given \( \{Y_j\}_{j=1}^{J} \) by the minimization of the mixed \( \ell_{2,1} \) norm of \( X \). We solve

\[
\min_X \sum_{j=1}^{n} \|X[j, :]\|_2 \quad \text{subject to} \quad \sum_{j=1}^{J} \|Y_j - \Phi_j X_j\|_2^2 \leq \nu \]

(3.22)
The choice of \( \nu \) depends on the noise level [5]. The problem in (3.22) can be posed as a convex optimization problem and can be solved efficiently. However, the main drawback of \( \ell_{2,1} \) is that it cannot perform equally well in noisy environment. Moreover, the computation time of the optimization increases with increasing \( r \). Hence we consider an extension of JLZA for MMV with multiple measurement matrices problem.

We estimate \( X \) by successively decreasing \( \sigma \) in

\[
X_\ast(\sigma) = \arg \max_X \mathbf{L}_\sigma(X) \\
\mathbf{L}_\sigma(X) = \hat{F}_\sigma(X) - \lambda \sum_{j=1}^{J} \|Y_j - \Phi_j X_j\|_F^2.
\]

(3.23)

We can also extend Lemma 6 in this case, which leads to an iterative algorithm similar to the one described in Table 3.2.

**Lemma 7.** Let us partition

\[ X_\ast = [X_{1\ast} \; X_{2\ast} \; \cdots \; X_{J\ast}] \]
3.2. Multiple Measurement Vectors Problem

Table 3.3: JLZA with Multiple Measurement Matrices (JLZA-M MM)

<table>
<thead>
<tr>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set $X_j^{(0)} = \Phi_j' (\Phi_j \Phi_j')^{-1} Y_j$, $j = 1, \ldots, J$. Form $X^{(0)} = [X_1^{(0)} X_2^{(0)} \cdots X_J^{(0)}]$.</td>
</tr>
<tr>
<td>2. Set $\sigma = \max_i |X^{(i)}_j[i,:]|<em>1$, $\lambda \in [1, 10^{10}]$ and $\rho, \eta, \gamma \in [0, 1)$, $\sigma</em>{\min} \in [0.5, 10^{-4}]$.</td>
</tr>
</tbody>
</table>

repeat
| 3. Set $\kappa = 1$. |
| 4. while $L_\sigma \{ \kappa \zeta(X^{(i)}) + (1 - \kappa)X^{(i)} \} < L_\sigma(X^{(i)})$ then $\kappa = \gamma \kappa$. |
| 5. $X^{(i+1)} = \kappa \zeta(X^{(i)}) + (1 - \kappa)X^{(i)}$ |
| 6. If $\tau^{(i)} = \|X^{(i+1)} - X^{(i)}\|_F < \eta \sigma$ then $\sigma = \rho \sigma$. |

until $\sigma \geq \sigma_{\min}$

Define the mappings $\zeta_j : \mathbb{R}^{n \times r} \to \mathbb{R}^{n \times r}$, $j \in \{1, 2, \ldots, J\}$ such that

$$\zeta_j(X) = \lambda \left[ W(X)/\sigma^2 + \lambda \Phi_j' \Phi_j \right]^{-1} \Phi_j' Y_j,$$

(3.24)

where $W(X)$ is a diagonal matrix:

$$W(X) = \begin{bmatrix} f_\sigma(\|X[1,:]\|_2) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f_\sigma(\|X[n,:]\|_2) \end{bmatrix}.$$

(3.25)

Then $X_j^*(\sigma) = \zeta_j\{X_\sigma(\sigma)\}$, $j \in \{1, 2, \ldots, J\}$. In addition, if we define the map $\zeta : \mathbb{R}^{n \times J_r} \to \mathbb{R}^{n \times J_r}$ such that

$$\zeta(X) := [\zeta_1(X) \ \zeta_2(X) \ \cdots \ \zeta_J(X)],$$

then for any $X$ there exists a real-valued scalar $\kappa \geq 0$ such that

$$L_\sigma[\kappa \zeta(X) + (1 - \kappa)X] \geq L_\sigma(X).$$

(3.26)

Proof. The proof is similar to that of Lemma 6.

The algorithm given in Table 3.3 is a straightforward extension of its single measurement matrix counterpart in Table 3.2 using Lemma 7. Note that the complexity of the JLZA-MMM is $J$ times that of the JLZA in Table 3.2.
3. Structured Sparse Signal Recovery

3.3 Block Sparse Signals

3.3.1 Background

In a block sparse signal $x \in \mathbb{R}^n$ the nonzero entries occur in clusters. Let $x \in \mathbb{R}^n$ be divided into $M$ blocks and the $j$-th block of $x$ is denoted by $x[j]$. Then $x$ can be viewed as a concatenation of blocks i.e.,

$$x = \left[ \begin{array}{cccc} x_1 & \cdots & x_{b_1} \\ x_{b_1+1} & \cdots & x_{b_1+b_2} \\ \vdots & \ddots & \vdots \\ x_{b_{M-1}+1} & \cdots & x_n \end{array} \right]$$

(3.27)

The vector $x$ is called block $k$-sparse over $\mathcal{B} = \{b_1, \ldots, b_M\}$, if

$$\sum_{j=1}^{M} I(\|x[j]\|_2) = k,$$

where $I$ is an indicator function defined in (3.5) and $n = \sum_{j=1}^{M} b_j$ [46]. In the following, we assume that the block lengths $b_1, b_2, \ldots, b_M$ are known.

Block sparse signals arise in multi-band signals [43], in measurements of gene expression levels [60] etc. Let us consider the data model

$$y = \Phi x,$$

(3.28)

where $\Phi \in \mathbb{R}^{m \times n}$ with $m < n$ and $x \in \mathbb{R}^n$ is a block sparse signal. To recover $x$ from $y$ and $\Phi$ one can exploit the block sparse structure of $x$. The reconstruction process requires solving

$$\min_x \sum_{j=1}^{M} I(\|x[j]\|_2) \text{ subject to } y = \Phi x$$

(3.29)

where $I$ is the indicator function defined in (3.5). However, the problem in NP-hard. A relaxation is proposed in [46, 45]

$$\min_x \sum_{j=1}^{M} \|x[j]\|_2 \text{ subject to } y = \Phi x$$

(3.30)

An extension of OMP is proposed in [47]. The resulting algorithm is called block-OMP (BOMP). In BOMP the matrix $\Phi$ is divided into blocks

$$\Phi = \left[ \begin{array}{cccc} \phi_1 & \cdots & \phi_{b_1} \\ \phi_{b_1+1} & \cdots & \phi_{b_1+b_2} \\ \vdots & \ddots & \vdots \\ \phi_{b_{M-1}+1} & \cdots & \phi_n \end{array} \right]$$

(3.31)
3.3. Block Sparse Signals

Table 3.4: The Block-OMP (BOMP) algorithm

<table>
<thead>
<tr>
<th>Initialization</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set $r^{(0)} = y, s = 1$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Repeat</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Select the index of a block of $\Phi$ such that $i_s = \text{argmax}_{1 \leq i \leq M} | \Phi[i]'r^{(s-1)} |_2$</td>
<td></td>
</tr>
<tr>
<td>3. Update the approximation $y^{(s)} = \text{argmin}_y | y - \tilde{y} |_2^2$, such that $\tilde{y} \in \text{span}{\Phi[i_1], \ldots, \Phi[i_s]}$</td>
<td></td>
</tr>
<tr>
<td>4. $r^{(s)} = y - y^{(s)}$, $s = s + 1$</td>
<td></td>
</tr>
</tbody>
</table>

Until a stopping criterion is satisfied

where $\phi_i$ indicates $i$-th column of $\Phi$. The outline of BOMP algorithm is given in Table 3.4. The algorithm is similar to the OMP algorithm given in Table 2.1. The difference is that, BOMP selects a block of $\Phi$ at every iteration, whereas OMP considers single column of $\Phi$ at every iteration. The recovery conditions of BOMP are also analyzed [47]. Regularized Group Orthogonal Matching Pursuit (ReGOMP) is another variant, combining ideas from [47] and [61]. In the following we show that ISL0 can extend for block sparse recovery. The resulting algorithm is called block-ISL0 (BISL0).

3.3.2 Block ISL0

In block-ISL0 we approximate (3.29) using Gaussian functions. Defining

$$\|x\|_{2/0} = \sum_{j=1}^{M} I(\|x[j]\|_2)$$

the function

$$\bar{F}_\sigma(x) = \sum_{j=1}^{M} f_\sigma(\|x[j]\|_2).$$

(3.32)

behaves like $M - \|x\|_{2/0}$ when $\sigma \to 0$. Then using the strategy in (2.20) one can consider the following way to solve $x$ given $y$:

$$x_* = \text{arg max}_{x \in \mathcal{X}} \bar{F}_\sigma(x).$$

(3.33)

Then taking Lagrangian in (3.33) and following the procedure of Section 2.2.2.2 one can verify that

$$x_* = \bar{W}^{-1}(x_*)\Phi' \left[ \Phi\bar{W}^{-1}(x_*)\Phi' \right]^{-1} y.$$
where

\[
\bar{W}(x) := \text{diag}\left\{ f_\sigma(\|x[1]\|_2) \cdots f_\sigma(\|x[1]\|_2) \right\} \text{ for } b_1 \text{ terms}
\]

\[
\cdots f_\sigma(\|x[M]\|_2) \cdots f_\sigma(\|x[M]\|_2) \right\} \text{ for } b_M \text{ terms}.
\]

(3.35)

Let us define the map \( \bar{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n \) such that

\[
\bar{g}(x) = \bar{W}^{-1}(x)\Phi' \left[ \Phi \bar{W}^{-1}(x)\Phi' \right]^{-1} y.
\]

(3.36)

Then using the concept in Lemma 3, it can be shown that \( \bar{F}_\sigma(x) \) is an increasing function along the direction \( \bar{g}(x) - x \).

Using this idea one can implement BISL0 algorithm by the iterative procedure proposed in Table 3.1.

### 3.4 ISL0 for Complex Valued Signals

#### 3.4.1 Complex Valued Multiple Measurement Vectors Problem

The algorithm proposed in Section 3.2.2.2 is formulated for real valued signal. However, complex valued sparse signals are common in practice. To this aim we extend JLZA for complex case. Let us consider a data model similar to (3.14)

\[
Y = \Phi X + E
\]

(3.37)

where \( \Phi \in \mathbb{C}^{m \times n}, X \in \mathbb{C}^{n \times r} \). Also \( E \in \mathbb{C}^{m \times r} \) is the additive noise. We assume that the signals are proper i.e., the real and imaginary components of each complex signal are independent. Now consider a optimization problem similar to (3.15),

\[
X_*(\sigma) = \arg \max_X \quad L_\sigma(X),
\]

\[
L_\sigma(X) := \bar{F}_\sigma(X) - \lambda \|Y - \Phi X\|_F^2.
\]

(3.38)

**Lemma 8.** Define the mapping \( \zeta : \mathbb{C}^{n \times r} \rightarrow \mathbb{C}^{n \times r} \) such that

\[
\zeta(X) = 2\lambda \left[ W(X)/\sigma^2 + 2\lambda \Phi^*\Phi \right]^{-1} \Phi^*Y,
\]

(3.39)

where \( W(X) \) is a diagonal matrix:

\[
W(X) = \begin{bmatrix}
    f_\sigma(\|X[1, :]\|_2) & \cdots & 0 \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & f_\sigma(\|X[n, :]\|_2)
\end{bmatrix}.
\]

(3.40)
3.5. Experiments

Then \( X_*(\sigma) = \zeta \{ X_*(\sigma) \} \). In addition, for any \( X \) there exits a real-valued scalar \( \kappa \geq 0 \) such that

\[
L_\sigma \{ \kappa \zeta(X) + (1 - \kappa)X \} \geq L_\sigma(X). \tag{3.41}
\]

**Proof.** See Appendix A.2.

By expressing \( X \) in real and imaginary parts, i.e. \( X = X_r + iX_i \), we can view (3.38) as an optimization problem in \( [X_r \ X_i] \in \mathbb{R}^{n \times 2r} \), and (3.41) ensures that \( L_\sigma(X) \) is increasing in the direction \( [\text{Re}\{\zeta(X)\} - X_r \ \text{Im}\{\zeta(X)\} - X_i] \). Hence by using a backtracking algorithm, one can choose a suitable step length to find the next iterate. As in the real valued case (see Section 3.2.2.2), the ascent direction \( \zeta(X) - X \) turns out to be very similar to the Newton direction. While proving Lemma 8 in Appendix A.2, we derive that

\[
\zeta(X) - X = \left[ W(X)/\sigma^2 + 2\lambda \Phi^*\Phi \right]^{-1} G(X), \tag{3.42}
\]

where,

\[
G(X) := \frac{\partial L_\sigma(X_r, X_i)}{\partial X_r} + i \frac{\partial L_\sigma(X_r, X_i)}{\partial X_i}, \tag{3.43}
\]

is the complex-valued representation of the gradient of \( L_\sigma \) at \( X \). Note that \( \lambda \Phi^*\Phi \) is the complex-valued representation of the Hessian of \( \|Y - \Phi X\|_F^2 \). This is singular, as the number of rows in \( \Phi \) is much smaller than the number of columns. On the other hand, \( W(X) \) is also a positive definite matrix, and (3.42) can be seen as regularization approach. When we take \( \sigma \to 0 \), this regularization works provided that \( X \) has only a few non-zero rows (see definition of \( W \) in (3.40)), i.e., if \( X \) is jointly row sparse. In fact, the regularization can be viewed as a convex-concave procedure since it turns out that \( -W(X)/\sigma^2 - 2\lambda \Phi^*\Phi \) is the Hessian of the concave part of \( L_\sigma(X) \). The proposed JLZA algorithm is similar to Table 3.2.

The complex valued extension of ISL0 for MMV with multiple measurement matrices and block-sparse signal are similar and have the same form.

### 3.5 Experiments

In this section, we validate the results presented so far using numerical simulations. The performance evaluation of the extensions of JLZA (Section 3.2.2.2) is not carried here. We will demonstrate its performance in the following chapters. Two types of experiments are presented: exact signal recovery in noiseless case and approximate signal recovery in
noisy environment. We choose $n = 128$ and $m = 50$. For JLZA, JLZA-MMM and BISL0 algorithms, the value of $\sigma_{\text{min}}$ is same. For noiseless sparse recovery, $\sigma_{\text{min}}$ is fixed near to zero ($\sigma_{\text{min}} = 10^{-5}$). However, in noisy case, $\sigma_{\text{min}} = 0.05$. Experimental results domesticate that the dependency of the proposed algorithms on tuning parameters $\gamma, \rho$ and $\eta$ are similar to the dependency of ISL0 on those parameters. Hence we fixed $\gamma, \rho$ and $\eta$ to 0.5, 0.3 and 0.5 which is recommended in Section 2.3.3.

3.5.1 Simulations with Multiple Measurement Vectors

For experimental setup, we consider the procedure of [58]. The following steps are repeated 100 times for each experiment. i) We construct $\Phi \in \mathbb{R}^{50 \times 128}$ consisting of iid Gaussian random variables with zero mean and variance one, ii) for every sparsity ($k$), we generate a $k$ row sparse matrix $X_0 \in \mathbb{R}^{n \times r}$. The nonzero location set is drawn uniformly at random among $\binom{n}{k}$ choices. The nonzero elements are iid Gaussian random variables, as in (i). We compare the performance of JLZA to $\ell_1$, $\ell_2$, ReMBo, CS-MUSIC [54] and Model-CS [56]. We also decompose the MMV problem into a series of SMV problems. That is we consider every column of $Y$ independently, which generates $r$ SMV problems. We then solve them independently by using $\ell_1$ (SMV-$\ell_1$) and orthogonal matching pursuit (SMV-OMP) algorithms. We use SDPT3 [62] through the CVX interface [63] for ReMBo, $\ell_{1,1}$ and $\ell_{2,1}$\footnote{Source code at http://www.cs.ubc.ca/~mpf/?n=JointSparse}. In the ReMBo algorithm, the components of $w \in \mathbb{R}^r$ (see Section 3.2) are iid random variables with uniform distribution on $[-1, 1]$, while the associated SMV problem is solved by using both basis pursuit (ReMBo-BP) and orthogonal matching pursuit (ReMBo-OMP). In orthogonal matching pursuit, we stop computation when the second norm of residual error becomes smaller than $10^{-16}$. We allow a maximum of 10 iterations to recover $X_0$ using different $w$.

In noiseless cases, we say $X_0$ to be recovered when the reconstructed $X_*$ satisfies $\|X_0 - X_*\|_\infty < 10^{-5}$. In the noisy environment, the objective function of the convex relaxation based approaches (eq. (3.2)) is modified as,

$$\min_X \|X\|_{p,q} \text{ subject to } \|Y - \Phi X\|_F \leq \mu,$$

where $\mu^2$ is the variance of the noise signal. The similar model is used for the SMV solver.
3.5. Experiments

Figure 3.1: The recovery rates (averaged over 100 trials) for randomly generated $X_0 \in \mathbb{R}^{128 \times 10}$ with different sparsity $k$. The value of $m$ is fixed to 50 and $r = 10$.

used for ReMBo. The SNR is used to measure performance. SNR is defined as

$$SNR(X_s) = \frac{1}{r} \sum_{i=1}^{r} 20 \log_{10} \left( \frac{\|X_s[:, i]\|_2}{\|X_s[:, i] - X_0[:, i]\|_2} \right).$$

Figure 3.1 shows recovery rate as a function of $k$ for various techniques in noiseless case. Note that JLZA achieves a very high recovery rate for $k \leq 29$, and 29 happens to be the upper bound given by Lemma 4. ReMBo and CS-MUSIC perform better when $k \leq 20$ and $k \leq 24$ respectively. Note that the recovery performance of $\ell_1$ algorithm degrades significantly when we do not exploit joint-sparsity structure. For example, SMV-$\ell_1$ performance degrades when $k \geq 10$. Note in Figure 3.2 that $\ell_{1,1}$ and $\ell_{2,1}$ require large recovery time compared to ReMBo. Note that while the recovery time of ReMBo is not affected by $r$, it increases quickly with $k$ (i.e. $k \geq 15$). The average recovery time of JLZA is not affected by $r$ and $k$, which, however, will increase with increasing $m$. This is because the evaluation of $\zeta(X)$ in (3.13) depends on $m$. Also, the recovery time of Model-CS increases with increasing sparsity ($k$).

The performance of different algorithms in noisy environment is investigated in Figure 3.3-3.4. A model including additive noise can be written as $Y = \Phi X + E$, where $E$ consists of iid Gaussian random variables with mean zero and variance $\mu^2$. RemBo-OMP, SMV-$\ell_1$ and SMV-OMP perform poorly with noisy data, and results are not shown. As
Figure 3.2: Average recovery time of various MMV techniques. The value of $n = 128$ and $m = 50$. (a) Average recovery time (on 100 runs) versus $r$ for fixed $k = 10$, (b) Average recovery time (on 100 runs) versus $k$ for different MMV techniques. $r = 10$. 
opposed to convex optimization methods, JLZA demonstrates an improved recovery rate in noisy environment. Figure 7.3(a) shows the level of sparsity needed to estimate a noisy signal from a fixed measurement. Observe that JLZA enjoys a clear superiority over other algorithms. Also, for each algorithm there is a critical level of sparsity. When sparsity increases beyond this level, the performance drops rapidly. The level is 15, 18, 25 and 32 for ReMBo-BP, $\ell_1$, $\ell_2$, $\ell_1$ and JLZA respectively. Figure 7.3(b) demonstrates the effect of $\mu$ on joint-sparse signal for different algorithms.

Finally, Figure 3.4 provides a comparison of SNR archived by different algorithms as a function of $r$, showing the improvement achieved by JLZA.

We generate $J$ joint sparse problems of the form (3.20),

$$Y_j = \Phi_j X_j + E_j, \quad j \in \{1, 2, \ldots, J\}$$

We assume that the size of all measurement matrices are same, i.e., $\{\Phi_j \in \mathbb{R}^{m \times n}\}_{j=1}^J$ also $\{X_j \in \mathbb{R}^{n \times r}\}_{j=1}^J$. Each $\Phi_j \in \mathbb{R}^{m \times n}$ is generated independently using the procedure described in Section 3.5.1. For every sparsity ($k$), we generate $J$ matrices $\{X_j\}_{j=1}^J$, such that each matrix is $k$ row sparse. Furthermore the indices of nonzero rows of all $\{X_j\}_{j=1}^J$ are same. The nonzero location set is drawn uniformly at random among $\binom{n}{k}$ choices. The nonzero elements are iid Gaussian random variables. We then say that $X = [X_1 \ X_2 \ldots X_J]$ is also joint row-sparse.

For reconstruction purpose we consider two strategies. In the first approach we consider the JLZA-MMM algorithm in Table 3.3. We also decompose the problem into a series of MMV problems. That is we consider every $Y_j$ and $X_j$ independently, which generates $J$ number of MMV problems. We then solve them independently by using the JLZA algorithm proposed in Table 3.1. We do not change $\lambda$ in Table 3.3. In all cases $\lambda$ was fixed to $10^{10}$. The signal recovery conditions and the definition of SNR are same to Section 3.5.1. For all simulations $r = 3$ and $J = 3$.

Figure 3.5(a) shows recovery rate as a function of $k$ for various techniques in noiseless case. We set $E_j = 0$. It is clear from the figure that JLZA-MMM leads to a significant performance improvement. Note that JLZA can give 100% signal recovery when $k \leq 25$. However JLZA-MMM shows 100% recovery for $k \leq 42$. Another interesting outcome of this simulation is that the performance of signal recovery from MMV with multiple measurement matrices is better than MMV with single measurement matrix. Let us compare the Figure 3.1 and Figure 3.5(a). In Figure 3.1 we consider one measurement matrix $\Phi \in \mathbb{R}^{50 \times 128}$ and
Figure 3.3: Average SNR achieved by various MMV techniques. The value of $n = 128$ and $m = 50$. (a) Average SNR (on 100 runs) versus $k$ for fixed $r = 10$ and $\mu = 0.03$, (b) Average SNR (on 100 runs) versus $\mu$ for fixed $k = 20$ and $r = 10$. 
3.5. Experiments

Figure 3.4: Average SNR (on 100 runs) versus $r$ for MMV problem. $k = 15$, $\mu = 0.03$, $n = 128$ and $m = 50$.

single $k$-row sparse matrix $X \in \mathbb{R}^{128 \times 10}$. Again in Figure 3.5(a), we consider 3 measurement matrices $\{\Phi_j \in \mathbb{R}^{50 \times 128}\}_{j=1}^3$ and three row sparse matrices $\{X_j \in \mathbb{R}^{128 \times 3}\}_{j=1}^3$. Note that the size of $X \in \mathbb{R}^{128 \times 10}$ and $X \in \mathbb{R}^{128 \times 9}$ are almost the same. However, recovery performance of JLZA-MMM is better. For example, JLZA in Figure 3.1 gives 100% recovery when $k \leq 32$ whereas it is $k \leq 42$ for JLZA-MMM in Figure 3.5(a).

We then evaluate the performance of algorithms in noisy environment in Figure 3.5(b). The noisy term $E_j$ in (3.20) consists of iid Gaussian random variables with mean zero and variance $\mu^2$. As expected, JLZA-MMV performs better than JLZA.

3.5.2 Block sparse problem

For experimental setup, we consider the procedure of [47]. We consider all blocks have same length in (3.27) i.e., $b_1 = b_2 = \cdots = b_M = b$. We set $n = 128$. The following steps are repeated 100 times for each experiment. i) We construct $\Phi \in \mathbb{R}^{m \times n}$ consisting of iid Gaussian random variables with zero mean and variance one. ii) generate a signal $x \in \mathbb{R}^n$ with all elements are zero. The signal $x$ is then divided into consecutive blocks of length $b$, then $M = n/b$. iii) for every block sparsity ($k$), the locations of the nonzero blocks of $x$ are chosen uniformly at random among all possible $\binom{M}{k}$ locations. The nonzero elements are iid Gaussian random variables, as in (i). We compare the performance of BISL0 to BOMP
Figure 3.5: Recovery performance of algorithms for MMV with multiple measurement matrix. The value of $m$ is fixed to 50. (a) Signal recovery from noiseless data. The recovery rates for randomly generated $X_j$ with different sparsity $k$. (b) Signal recovery from noisy data. Average SNR versus $k$ for $\mu = 0.03$. 
and ReGOMP. We also treat the block spare signal as a simple sparse signal. That is we ignore the block structure of the signal and try to recover using $\ell_1$ i.e. by solving

$$\min_{x \in \mathbb{X}} \|x\|_1$$

In orthogonal matching pursuit, we stop computation when the second norm of residual error becomes smaller than $10^{-16}$. The signal recovery conditions and definition of SNR are same to Section 3.5.1.

Figures 3.6(a)-3.6(b) demonstrate the recovery performance of different algorithms in noiseless cases. We consider two different block sizes. In all cases the recover performance of BISL0 is better than other algorithms. BOMP and ReGOMP performs almost similarly. However performance of $\ell_1$ is worse.

Figures 3.7(a)-3.7(b) show the signal recovery rates as a function of measurements ($m$). Note that larger block size $b$ increases performance of the algorithms. In both figures, the actual number of nonzero components in $x$ is same i.e 32. However, the experiment with larger block (Figure 3.7(b)) shows that it takes smaller measurements $m$ to recover the signal. For example in Figure 3.7(a), BISL0 achieves 90\% recovery when $m = 55$. However, when block length increases in Figure 3.7(b), BISL0 needs 50 measurements to achieve the same rate of recovery.

Finally, Figures 3.8(a)-3.8(b) illustrate the performance of different algorithms in presence of noise. For noisy case we consider the model in (2.4), where $e$ consists of iid Gaussian random variables with mean zero and variance $\mu^2$. As expected BISL0 outperforms other algorithms. For example, in Figure 3.8(a), BISL0 maintains an average SNR of 42dB when block sparsity $k \leq 7$. On the other hand BOMP has an average SNR 31dB when $k \leq 6$. Note that $\ell_1$ outperforms BOMP at lower sparsity. However as sparsity increasing the relative performance of $\ell_1$ degrades quickly compared to BOMP.

3.6 Conclusion

In this chapter we propose some modifications of ISL0 algorithm. At first we have demonstrated how the structure of sparse signals can be used to improve the signal recovery performance. We found that one can reduce the number of compressive measurements necessary to recover $x$ by exploiting the structure. We exploit two types of structure of sparse signal. In one case, we consider a set of sparse signals that are joint sparse. We reformulate
Figure 3.6: The recovery rates (averaged over 100 trials) for randomly generated $x \in \mathbb{R}^{128}$ with different block sparsity $k$. The value of $m$ is fixed to 50. (a) Block size $b = 4$. (b) Block size $b = 8$. 
Figure 3.7: The recovery rates (averaged over 100 trials) for randomly generated $x \in \mathbb{R}^{128}$ with different values of $m$. (a) Block size $b = 4$, sparsity $k = 8$. (b) Block size $b = 8$, sparsity $k = 4$. 
Figure 3.8: Average SNR achieved by various techniques. The value of $n = 128$, $b = 4$ and $m = 50$. (a) Average SNR (on 100 runs) versus $k$ for $\mu = 0.03$, (b) Average SNR (on 100 runs) versus $k$ for $\mu = 0.05$. 
ISL0 for joint sparse signal and the resulting algorithm is called joint $\ell_{2,0}$ approximation (JLZA). We find that one can get significant recovery performance improvement by exploiting the structure. In other case, the sparse signal have block sparsity structure. In Section 3.3, we show that a simple modification of ISL0 can easily exploit the block sparsity. Finally we consider complex valued sparse signal in Section 3.4. We modify ISL0 to work with complex valued signal. The modifications make ISL0 more applicable for practical applications.
Chapter 4

Sparse Signal Recovery with Prior Information

4.1 Introduction

In Chapter 1 we assumed that the support of sparse signal $x$ is completely unknown. In many problems, some a priori information about support of $x$ i.e., $S(x)$ (see Definition 6) and variance of $x$ are available, which can be used to facilitate the recovery process motivating the research in CS with partially known support [64, 65]. Typical applications include magnetic resonance imaging (MRI) [66] and video processing [67]. Here it is required to recursively reconstruct a sequence of highly correlated images, which admit sparse representations under certain transformations, e.g., wavelets, discrete cosine transform (DCT), total variation (TV), etc. In these problems it is natural to use the support of the signal at time epoch $t$ to estimate that at epoch $t+1$. For example, consider the larynx MRI sequence in Figure 1(a). Let $\tilde{x}(t)$ be the vector of 2-D wavelet coefficients of the image at time $t$, and let $x(t)$ be a vector constructed from $\tilde{x}(t)$ by retaining the components of $\tilde{x}(t)$ from its 99.95%-energy support while setting others to zero so that supp$\{x(t)\}$ is the largest among the sets $K \subset I$ such that

$$\sum_{i \in K} |\tilde{x}_i(t)|^2 < 0.9999\|\tilde{x}(t)\|^2_2$$

It is wellknown that there is very little difference between the images reconstructed using $\tilde{x}(t)$ and $x(t)$ . Let $\setminus$ be the set difference operator, i.e. $X \setminus Y = \{x \in X : x \not\in Y\}$. 

76
4.2 Problem definition and related work

Define
\[ \vartheta(t) = \frac{\text{card}[S\{x(t)\} \setminus S\{x(t-1)\}]}{\text{card}[S\{x(t)\}]} \]
which is a measure of the difference in \( S\{x(t)\} \) and \( S\{x(t-1)\} \). Figure 1(b) shows the plot of \( \vartheta(t) \) as a function of \( t \) for the the MRI sequences in Figure 1(a). Note that on average the change of support is around 3% only. Thus if we have \( x(t-1) \), we can use \( S\{x(t-1)\} \) as an estimate of \( S\{x(t)\} \).

4.1.1 Contributions and Motivation

Recent papers on CS with partially known support [64, 65, 68] demonstrate that the prior knowledge about the support can be used to significantly lower the number of data samples needed for reconstruction. In clinical MRI, this means reduced scan time, high temporal resolution, and improved in patient comfort. A review of the algorithms for CS with partial support are given in Section 4.2. Most of these algorithms assume that the sparsity pattern is changing slowly with time. Hence the support information is fairly accurate. This assumption may not hold for low-frame-rate videos. In this chapter, we propose a MAP estimation approach called PMAP (partially known support using MAP). We set up a \textit{priori} density function for \( x \), such that a component of \( x \) within the known support is of large magnitude with a high probability. On the other hand, according to the prior, a component of \( x \) outside the known support is of large magnitude with a very small probability. We propose numerical strategies to solve the resulting optimization problem. The idea works also when the observed measurements are noisy. The proposed algorithm has been tested in simulations with real-world data.

4.2 Problem definition and related work

\textit{Notation:} Define \( \mathbb{I} = \{1, 2, \ldots, n\} \). Let \( T \subset \mathbb{I} \). We define \( T^c := \{i \in \mathbb{I} : i \notin T\} \). Finally for \( T_1 \subset \mathbb{I} \) we define \( T_1 \setminus T := T_1 \cap T^c \). For a real number \( z \) we denote \( \lceil z \rceil \) is the smallest integer not less than \( z \).

Let \( T = S(x) \), and \( k = \text{card}(T) \). Note that \( T \) is unknown. However, we have a prior estimate \( T_g \) of \( T \), and \( T_g \neq T \). The set corresponding to “misinformation” is \( T_s := T_g \setminus T \), i.e., according to the prior information we think \( T_s \subset T \), but in reality \( T_s \cap T = \emptyset \). Similarly, the unknown part of the support is \( T_u := T \setminus T_g \), i.e. according to the prior information we
4. Sparse Signal Recovery with Prior Information

Figure 4.1: (a) Larynx MRI video sequence. (b) Larynx MRI images are sparsified using 2-D discrete wavelet transform. Then we calculate the support change of the MRI sequence.

think \( T_u \cap T = \emptyset \), but in reality \( T_u \subset T \). Let \( g = \text{card}(T_g) \), \( u = \text{card}(T_u) \), \( s = \text{card}(T_s) \).

Note that \( T_g \) is known, and thus \( g \) is known \(^1\). The error-sets \( T_u \) and \( T_s \) are unknown. The problem is to estimate sparse \( x \) that satisfies (1.1).

One way to address the problem is to solve [64]

\[
    x = \arg \min_{v \in \mathbb{X}} \|v_{T_g}\|_0. \tag{4.1}
\]

The following lemma, gives the sufficient conditions for the existence of the unique solution to the problem.

**Lemma 9.** (Proposition 1,[64]) Under Assumption 2, (4.1) has a unique solution if \( g + 2u < m \).

Since (4.1) is an NP-hard problem [5, 15], several alternative methods have been proposed. Least squares CS-residual (LS-CS) [68] is a two step procedure. First, a least squares (LS) estimate of \( x \) is computed using \( T_g \), along with the associated residual

\[
    \tilde{y} = y - \Phi_{T_g} [\Phi_{T_g} \Phi_{T_g}' ]^{-1} \Phi_{T_g}' y,
\]

\(^1\)A direction for estimating values of different parameters can be found in Section 4.5.1.
In second step, LS-CS uses a CS algorithm to find a sparse solution $\hat{x}$ such that $\tilde{y} = \Phi \hat{x}$. The final estimate of $x$ is $\bar{x} + \hat{x}$. In our experience LS-CS works well when $T_g$ is an accurate estimate of $T$. However, it cannot perform equally to $\ell_1$ based approaches in noiseless environment. Recently proposed modified-CS [64] uses a modified $\ell_1$ minimization approach. Modified-CS estimates $x$ as

$$x = \arg \min_{v \in \mathbb{X}} \|v_{T_g}\|_1.$$  

(4.2)

In [64] it is shown that modified-CS requires less data samples for sparse recovery. Weighted $\ell_1$ minimization [65] approach computes $x$ as

$$x = \arg \min_{v \in \mathbb{X}} \left( \|v_{T_g}\|_1 + \bar{w} \|v_{T_g}\|_1 \right)$$

(4.3)

where $\bar{w}$ is a positive constant whose value needs to determine from the problem in hand.

In the following we propose a maximum a posteriori based approach to estimate $x$.

Kalman filtered compressed sensing [69], uses a reduced order Kalman filter that can utilize the given support set $T_g$ to estimate new additions to the support set. Another kalman filtered based technique has also been proposed in [70]. However, kalman filter based techniques can perform well only when sparsity pattern changes slowly with time. Regularized modified basis pursuit denoising (BPDN) (reg-mod-BPDN) algorithm [71] assumes that partial information of signal support and signal value are known in advance. In [72], a suite of dynamic algorithms has been presented for solving $\ell_1$ minimization programs for streaming sets of measurements. In [73], an iterative algorithm is presented for recovering time-varying sparse signals based on the belief propagation (BP) methodology [74].

4.3 Maximum a posteriori estimation (MAP)

4.3.1 MAP framework

Suppose the probability density function of $x$ is $p(x)$. Then according to MAP approach [75] we minimize negative log-likelihood function

$$\min_{x \in \mathbb{X}} - \ln[p(x)],$$

(4.4)

to estimate $x$. In the following we propose a candidate for $p(x)$, which is suitable to represent the prior knowledge in form of $T_g$. We denote the Gaussian density function as

$$N(x_i, \nu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left\{ -\frac{(x_i - \nu)^2}{2\sigma^2} \right\}.$$  

(4.5)
Note that \( x \) is a sparse signal. Hence it has two types of components. The majority of its components are of very small magnitude, which constitute \( x_{\mathcal{T}^c} \). Thus, for \( i \in \mathcal{T}^c \) we assume that the probability density function of \( x_i \) is \( N(x_i, 0, \hat{\mu}) \), where \( \hat{\mu} \) is a very small number.

On the other hand, for \( i \in \mathcal{T} \) it is reasonable to assume that the density function of \( x_i \) is \( N(x_i, 0, \sigma) \), where \( \sigma \gg \hat{\mu} \). If we had no information available about \( \mathcal{T} \), then for every \( i \in \mathcal{I} \) there are two possibilities:

i) \( i \in \mathcal{T} \) with probability \( k/n \); and

ii) \( i \in \mathcal{T}^c \) with probability \( 1 - k/n \).

Hence it is appropriate to postulate that \( x_i \) has a Gaussian mixture density \( (k/n)N(x_i, 0, \sigma) + (1 - k/n)N(x_i, 0, \hat{\mu}) \) when no prior knowledge about \( \mathcal{T} \) is available. This Gaussian mixture model are popular in simulations [34] and performance analysis in CS literature (see Chapter 2, Section 2.2.1.2).

When a priori estimate \( \mathcal{T}_g \) of \( \mathcal{T} \) is given, we adapt accordingly. If \( i \in \mathcal{T}_g \), then \( i \in \mathcal{T} \) with probability \( 1 - s/g \), because only \( s \) elements among \( g \) elements of \( \mathcal{T}_g \) are not in \( \mathcal{T} \). On other hand if \( i \in \mathcal{T}_g^c \) then \( i \in \mathcal{T} \) with probability \( u/(n-g) \), because only \( u \) elements among \( n-g \) elements of \( \mathcal{T}_g^c \) are in \( \mathcal{T} \). Define

\[
p_i = \begin{cases} 
1 - s/g, & i \in \mathcal{T}_g \\
u/(n-g), & i \in \mathcal{T}_g^c
\end{cases}
\]  

(4.6)

Then we can postulate that \( x_i \) has a a priori probability density function

\[
h(x_i, p_i) = p_i N(x_i, 0, \sigma) + (1 - p_i) N(x_i, 0, \hat{\mu}).
\]  

(4.7)

Assuming that the components of \( x \) are mutually independent, we have \( p(x) = \prod_{i=1}^{n} h(x_i, p_i) \).

Consequently, the MAP estimation approach requires us to solve

\[
\min_{x \in \mathcal{X}} - \sum_{i=1}^{n} \ln[h(x_i, p_i)].
\]  

(4.8)

Now using (4.5) and (4.7) we have

\[
- \ln[h(x_i, p_i)] = \frac{x_i^2}{2 \sigma^2} - \ln \left[ 1 + r_i \exp \left\{ -\frac{x_i^2}{2 \sigma^2} \right\} \right] + \text{constant}
\]  

(4.9)

where

\[
r_i = \frac{1 - p_i}{p_i \hat{\mu}}, \quad \frac{1}{\sigma^2} = \frac{1}{\hat{\mu}^2} - \frac{1}{\sigma^2}, \quad \text{and} \quad \sigma \gg \hat{\mu}
\]  

(4.10)
4.3. Maximum a posteriori estimation (MAP)

The expression (4.9) plays a central role in the sequel. We use (4.9) for analyzing the MAP cost function and deriving a numerical algorithm to minimize it. First we establish that solving (4.8) indeed yields a sparse solution provided that \( \hat{\mu} \) is sufficiently small. Note that if we assume \( s = 0 \) in (4.6), then the optimization problem in (4.8) will be similar to (4.1). However, we always consider the \( s > 0 \).

4.3.2 Solution strategy

Like many optimization problems encountered in CS literature \([76, 26]\), (4.8) is a nonconvex problem, and no known numerical algorithms with guaranteed convergence properties exist for (4.8). To deal with this we use a sequential optimization approach, which has been proved very useful in similar problems (see Section 2.2). Since \( \rho \gg \hat{\mu} \), it follows that \( |\hat{\mu} - \hat{\sigma}| \) is a small number, and \( \rho \gg \hat{\sigma} \). When \( i \in T_g \), then \( r_i = \rho s / [\hat{\mu}(g - s)] \) is neither a large nor a small number. Thus for \( i \in T_g \), the first term in the right hand side of (4.9) dominates, and hence \(-\ln[h(x_i, p_i)]\) is slightly nonconvex. On the other hand, if \( i \in T^c_g \), then \( r_i = (\frac{n - 2}{u} - 1) \hat{\mu} \gg 1 \), inducing a high degree of nonconvexity in \(-\ln[h(x_i, p_i)]\). To deal with the nonconvex optimization problem we adopt a sequential optimization approach where we sequentially minimize the cost functions

\[
\varphi_j(x) = \sum_{i=1}^{n} \left\{ \frac{x_i^2}{2\hat{\sigma}^2} - \ln \left[ 1 + \rho_{ij} \exp \left( -\frac{x_i^2}{2\hat{\sigma}_j^2} \right) \right] \right\}, \tag{4.11}
\]

where

\[
\hat{\sigma}_j = (\hat{\sigma}/\hat{\sigma}_0)^{j/w} \hat{\sigma}_0, \quad \rho_{ij} = (r_i/r_{i_0})^{j/w} r_{i_0}, \quad j = 0, 1, 2, \ldots, w,
\]

where \( \hat{\sigma}_0 \) is a suitably chosen large positive number, \( r_{i_0} \) is a small positive number (see below), and \( w \) is a user chosen integer. At \( j = 0 \) we take \( \hat{\mu} = \rho \) and hence \( \hat{\sigma} \to \infty \). As a result the logarithmic term of (4.11) becomes almost independent of \( x_i \) and can be treated as a constant. Hence \( \varphi_0(x) = \sum_{i=1}^{n} x_i^2 / (2\hat{\sigma}^2) + \text{Constant} \), is a convex function. As we increase the value of \( j \) form 0 to \( w \), we gradually transform \( \varphi_j \) from a convex function \( \varphi_0 \) to our desired likelihood function \( \varphi_w \). If \( w \) is sufficiently large, then the change from \( \varphi_{j-1} \) to \( \varphi_j \) is small, and so is the change from \( x^{(j-1)} \) to \( x^{(j)} \) where

\[
x^{(j)} := \arg\min_{x \in \mathbb{R}} \varphi_j(x). \tag{4.12}
\]

Now it is wellknown that \( x^{(0)} = \Phi'(\Phi')^{-1} y \) \([34]\). Since \( ||x^{(1)} - x^{(0)}|| \) is expected to be a small number, it is natural to initialize an iterative numerical optimization algorithm to
compute $x^{(1)}$ at $x^{(0)}$. In this way it is highly likely that the numerical algorithm will reach
at $x^{(1)}$. Next we can use $x^{(1)}$ to initialize the algorithm to compute $x^{(2)}$, and the procedure
is repeated with the value of $j$ is incremented at each step until $j = w$.

Recall that for $j = 0$, we take $\hat{\mu} = \varrho$. Now as $\hat{\mu} \to \varrho^+$, we have $r_i \to (1 - p_i)/p_i$
and $\exp\{-x_i^2/(2\hat{\sigma}^2)\} \to 1$ for all $i \in I$, see (4.10). This motivates the choice

$$r_{i_0} = (1 - p_i)/p_i.$$ 

Also by letting

$$\hat{\sigma}_0 \geq 4 \max_i |x_i^{(0)}|;$$

we ensure that $\exp\{-[x_i^{(0)}]^2/(2\hat{\sigma}_0^2)\} \geq 0.966$ for all $i$. As $||x^{(1)} - x^{(0)}||$ is small, this choice
of $\hat{\sigma}_0$ ensures that $\exp\{-x_i^2/(2\hat{\sigma}_0^2)\} \approx 1$ for all $x$ satisfying $||x - x^{(0)}|| < ||x^{(1)} - x^{(0)}||$ [34].

### 4.3.3 Minimizing $\varphi_j$

In this section we explore some properties of $x^{(j)}$, and develop a numerical algo-
rithm to compute it. The Lagrangian associated with the optimization problem in (4.12)
is

$$L_j(x, \nu) = \varphi_j(x) + \nu^T(\Phi x - y),$$

(4.13)

where $\nu \in \mathbb{R}^{m \times 1}$ is the vector of Lagrange multipliers. The minimizer of $L_j(x, \nu)$ with
respect to $x$ is a function of $\nu$, and is denoted by $\chi_j(\nu)$:

$$\chi_j(\nu) = \arg \min_x L_j(x, \nu).$$

Thus $\chi_j(\nu)$ is a stationary point of $L_j(x, \nu)$ for a given value of $\nu$, i.e.

$$\left. \frac{\partial L_j(x, \nu)}{\partial x} \right|_{x=\chi_j(\nu)} = \left. \frac{\partial \varphi_j(x)}{\partial x} \right|_{x=\chi_j(\nu)} + \Phi^T \nu = 0. \quad (4.14)$$

It is readily verified that

$$\left. \frac{\partial \varphi_j(x)}{\partial x_i} \right|_{x=\chi_j(\nu)} = \xi_j(x_i)x_i \quad (4.15)$$

where

$$\xi_j(x_i) = \frac{1}{\varrho^2} + \frac{\rho_{ij} \exp \left( \frac{-x_i^2}{2\hat{\sigma}_j^2} \right)}{\hat{\sigma}_j^2 \left[ 1 + \rho_{ij} \exp \left( \frac{-x_i^2}{2\hat{\sigma}_j^2} \right) \right]}.$$

(4.16)
Consequently, we have
\[
\frac{\partial L_j(x, \nu)}{\partial x} = W_j(x)x,
\]
where \( W_j(x) \) is a \( n \times n \) diagonal matrix defined as
\[
W_j(x) = \text{diag}\{\xi_j(x_1), \xi_j(x_2), \ldots, \xi_j(x_n)\}
\]

From (4.14)-(4.17) we get
\[
\chi_j(\nu) = -W_j^{-1}[\chi_j(\nu)]\Phi'\nu.
\]
However for \( x \in X \) we have \( y = \Phi x \), and \( L_j(x, \nu) = \varphi_j(x) \) for all \( \nu \). Thus, if we can find a \( \nu^* \) such that \( \Phi \chi(\nu^*) = y \), then
\[
\chi_j(\nu^*) = \arg\min_x L_j(x, \nu^*) = \arg\min_{x \in X} \varphi_j(x) = x^{(j)}.
\]
Then by (4.18) and (4.19) we see that \( \nu^* \) must satisfy
\[
\nu^* = -[\Phi W_j^{-1}(x^{(j)})\Phi']^{-1}\Phi x^{(j)},
\]
which upon substitution in (4.18) gives
\[
x^{(j)} = \hat{g}_j(x^{(j)}),
\]
where
\[
\hat{g}_j(x) := W_j^{-1}(x)\Phi' \left[ \Phi W_j^{-1}(x)\Phi' \right]^{-1} y.
\]
Equation (4.20) is nonlinear, and cannot be solved analytically. However, using the concept in Lemma 3, it can be shown that \( \varphi_j(x) \) is a decreasing function along the direction \( \hat{g}_j(x) - x \). Hence we can use a simple numerical algorithm for computing \( x^{(j)} \).

4.3.4 Numerical algorithm for solving MAP optimization

By compiling the results presented so far we propose the “MAP with partially known support” (PMAP) optimization strategy in Table 4.1. The algorithm starts with the assumption that the value of \( \rho, \hat{\mu}, s \) and \( k \) are known. In Section 4.5.1.1 we consider the sensitivity of PMAP to the choice of these parameters. We can calculate \( \hat{\sigma} \) and \( \{r_i\}_{i=1}^2 \) using (4.6) and (4.10). In each iteration, along the decent-direction \( \hat{g}_j(x) - x \) we find the step-length \( \kappa \) using the standard backtracking strategy (step 4-5) [23]. We set \( \beta = 0.5 \). The inner-iteration for updating \( x \) for a given \( j \) terminates when the relative change in the
**Table 4.1: MAP with partially known support (PMAP)**

<table>
<thead>
<tr>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set $x = \Phi'(\Phi\Phi')^{-1}y$.</td>
</tr>
<tr>
<td>2. Set $\sigma_0 = 4 \max_i</td>
</tr>
<tr>
<td>3. Set $j = 1$, choose $\eta \in (0, 1]$, $\beta = 0.5$ and $w$</td>
</tr>
</tbody>
</table>

repeat

4. Set $\kappa = 1$.

5. while $\varphi_j[\kappa \hat{g}_j(x) + (1 - \kappa)x] > \varphi_j(x)$

   $\kappa = \beta \kappa$.

end

6. $x_o = x$, and $x = \kappa \hat{g}_j(x) + (1 - \kappa)x$

7. If $\frac{\|x - x_o\|_2^2}{\|x_o\|_2^2} < \eta$

   $j = j + 1$.

end

while $j \leq w$.

magnitude of $x$ is below $\eta$, see step 7. Upon convergence of each inner iteration we increment $j$ (step 7). Our experimental results (see Section 4.5.1.1), suggest that in PMAP we can choose $w$ between 15 and 40. Note that choosing a smaller $\eta$ and a larger $m$ increases the reliability in the cost of computation time. Our experimental study suggests that choosing $w = 20$, and $\eta = 0.02$ makes a good tradeoff. Upon convergence for $j = w$, PMAP stops its iteration.

### 4.4 PMAP in noisy environment

So far we assumed noise-free data. However practical data may be noisy. Hence we consider the data model in (2.4), i.e.

$$y = \Phi x + e.$$  

We assume $e$ is a zero-mean Gaussian random vector with a covariance matrix $I/\lambda$. Then the probability density function of $e$ be

$$p_{err}(e) = \text{constant} \times \exp \left\{-\frac{\lambda}{2}||e||^2_2\right\}.$$  

Then the conditional density of $y$ given $x$ is $p_1(y|x) = p_{err}(y - \Phi x)$. Consequently the joint density of $x$ and $y$ is given by

$$\tilde{p}(y, x) = p_1(y|x)p(x) = p_{err}(y - \Phi x)p(x).$$
Then the joint negative log-likelihood function of $x$ and $y$ is given by

$$
\bar{\mathcal{P}}(x) = -\sum_{i=1}^{n} \ln[h(x_i, p_i)] + \frac{\lambda}{2} \|y - \Phi x\|_2^2.
$$

which we propose to minimize with respect to $x$. Our strategy to solve the problem is similar to that outlined in the previous section. One may consider a sequence of functions

$$
\bar{\mathcal{P}}_j(x) = \mathcal{P}_j(x) + \frac{\lambda}{2} \|y - \Phi x\|_2^2; \quad j = 0, 1, \cdots, w,
$$

so that $\bar{\mathcal{P}}_0(x) = \sum_{i=1}^{n} x_i^2/(2\sigma^2) + \frac{\lambda}{2} \|y - \Phi x\|_2^2$ with a global minimum point as

$$
\bar{x}^{(0)} = \lambda \left[ I_n / \sigma^2 + \Phi^T \Phi \right]^{-1} \Phi^T y.
$$

Define $\bar{x}^{(j)} = \arg\min_x \bar{\mathcal{P}}_j(x)$. At $\lambda \to \infty$, we have a closed form solution $\bar{x}^{(0)} = \Phi^T (\Phi \Phi^T)^{-1} y$. Hence we start at $\lambda \to \infty$. If $||\bar{x}^{(1)} - \bar{x}^{(0)}||$ is a small number, it is natural to initialize an iterative numerical optimization algorithm to compute $x^{(1)}$ at $\bar{x}^{(0)}$. However, if for $j = 1$, we set the value of $\lambda$ to the reciprocal of noise variance, then $||x^{(1)} - \bar{x}^{(0)}||$ may not a small number. Hence we adopt the sequential optimization strategy like (4.11). Modify (4.23) to

$$
\bar{\mathcal{P}}_j(x) = \mathcal{P}_j(x) + \frac{\lambda_j}{2} \|y - \Phi x\|_2^2.
$$

where $\lambda_j = (\lambda / \lambda_0)^{j/w} \lambda_0$ and $\lambda_0$ is a suitably chosen large positive number. Define

$$
d_0 = \min_i D_i
$$

where $\{D_i\}_{i=1}^{m}$ are the diagonal components of $\Phi^T \Phi$. Note that for $j = 0$, we choose $\lambda \to \infty$ in (4.24) and hence we have $(\frac{I_m}{\lambda \sigma^2} + \Phi^T \Phi) \approx \Phi^T \Phi$. This motivates the choice $\lambda_0 = \frac{1}{\epsilon d \sigma^2}$ such that $1 + \epsilon \approx 1$. Next we use $\bar{x}^{(0)}$ to initialize the algorithm to compute $\bar{x}^{(1)}$, and the procedure is repeated with the value of $j$ is incremented at each step until $j = w$. In our experiments we choose $\epsilon = 10^{-10}$.

As we cannot compute $\bar{x}^{(j)}$ using a Gauss-Newton type algorithm, we can modify our algorithm in the following way. Differentiating $\bar{\mathcal{P}}_j(x)$ and equating to zero it follows that

$$
\bar{x}^{(j)} = \bar{g}_j\{\bar{x}^{(j)}\}
$$

where the mapping $\bar{g}_j : \mathbb{R}^n \to \mathbb{R}^n$ is defined as

$$
\bar{g}_j(x) = \lambda_j \left[ 2W_j(x) + \lambda_j \Phi^T \Phi \right]^{-1} \Phi^T y
$$

$$
= W_j^{-1}(x) \Phi \left[ 2I_m / \lambda_j + \Phi W_j^{-1}(x) \Phi^* \right]^{-1} y,
$$
where in the last equality we have used the matrix inversion lemma. It can be shown that 
\( \tilde{\varphi}_j(x) \) is decreasing along \( \tilde{g}_j(x) - x \). Thus the algorithm in Table 4.1 can be used in noisy case with \( x = \hat{x}^{(0)} \) in Step 1, \( \varphi \) replaced by \( \tilde{\varphi} \) in Step 5, and \( \hat{g}(x) \) replaced by \( \tilde{g}(x) \) in Steps 5 and 7.

Now consider complex valued data model 
\[ y = \Phi x + e, \]
where \( \Phi \in \mathbb{C}^{m \times n} \) and \( x \in \mathbb{C}^n \) is a sparse signal. Also \( e \in \mathbb{C}^m \) is the additive noise. If we assume that the underlying distributions are circular symmetry complex Gaussian then the modification of PMAP is straightforward as before.

4.5 Experimental results

In this section, we empirically evaluate the performance of the proposed algorithm. We investigate dependency of PAMP on different parameters by using artificial data set. We then compare the performance of PMAP to other algorithms using both artificial and real experimental data. We compare the performance of PMAP algorithm with \( \ell_1 \) (conventional CS) [22, 63], modified-CS [64], weighted \( \ell_1 \) [65] and Dynamic-BPDN [72]. For weighted \( \ell_1 \) we set \( \bar{w} = 3 \), see (4.3), as recommended in [65]. We use CVX framework [63] for solving the optimization problem in CS, weighted-\( \ell_1 \), and modified-CS. In all experiments, we consider the PMAP formulation described in Section 4.3.3. The performance of PMAP modifications (Section 4.4) will evaluate in Chapter 8.

4.5.1 Simulation using artificial data set

The artificial data set is generated by following the procedure described in [64]. We choose \( n = 512 \) and value of \( m \) is varied between \( [0.15n] \sim [0.33n] \). The elements of \( \Phi \) are taken as mutually independent Gaussian random variables with zero mean and variance unity. Recall that \( k = \text{card}(T) \), where \( T = S(x) \). The elements of \( T \) are chosen uniformly at random among \( \binom{n}{k} \) choices. We set \( q = 10 \), and \( \hat{\mu} = 10^{-5} \). The value of \( g \) is varied from \( [0.75k] \) to \( [0.55k] \). Also, the value of \( s \) is varied from \( [0.1k] \) to \( [0.4k] \). The set \( T_s \) is generated by drawing \( s \) elements uniformly at random from \( T^c \). Similarly, \( T_u \) is generated by drawing \( u = k - g + s \) elements uniformly at random from \( T \). Finally we form \( T_g = (T \setminus T_u) \cup T_s \), which is used in estimation. In each simulation a new realization of all above quantities are used. We consider the measured data \( y \) satisfies \( y = \Phi x + e \), where \( e \) is zero mean, Gaussian random vector with a covariance matrix \( \mu^2 I \). Note that,
4.5. Experimental results

the actual values of $k$, $u$ and $s$ are not known in practice. However, estimation of those parameters must be used in the PMAP implementation. In the following, the estimation of $k$, $u$, $s$ are denoted by $\hat{k}$, $\hat{u}$ and $\hat{s}$ respectively. We assume that $g$ is known, which is a reasonable assumption in practice. The normalized root mean squared error [64] defined as $\text{NRMSE} := \frac{\|x_0 - \hat{x}\|_2}{\|x_0\|_2}$ is used as performance-measure, where $\hat{x}$ is the estimate of the true sparse signal $x_0$.

4.5.1.1 Dependence on the Parameters

If $k$, $g$ and $\hat{\mu}$ are assumed to be known, then there are two tuning parameters in PMAP algorithm, namely $s$ and $w$. The actual value of $s$ is not known in practice. Nevertheless, the experimental results in Figure 4.2(a)-4.2(b) suggest that knowing a rough estimate of $s$ is sufficient. In these experiments we set $k = \lceil 0.1n \rceil$, and different values of $s$ and $g$ to generate data. However, when we recover $x$ using PMAP we use $\hat{s} \neq s$ and set $\hat{u} = k - g + \hat{s}$. In all experiments we set $w = 20$ (this value is motivated by the results shown in Figure 4.2(c)-4.2(d)). The results demonstrate that the performance of PMAP remain almost same if $\hat{s} \in [0.1g, 0.4g]$. Hence in all the following experiments we take $\hat{s} = \lceil 0.2g \rceil$, regardless of the true value of $s$.

Next we study the dependence of PMAP on $w$. Recall from Table 4.1 that $w$ determines how fast $\hat{\sigma}_j, \rho_{i,j}$ change with $j$. Clearly, a slow change is desirable, which means a large $w$ is preferred. But a large $w$ means large computation time. In Figure 4.2(c)-4.2(d), we plot the average NRMSE of the recovered signal as functions of $w$. The results are shown for different combinations of $\mu$ and $g$. As can be seen in Figure 4.2(c)-4.2(d), PMAP is practically insensitive to any increase of $w$ beyond 15. Thus, we use $w = 20$ for all the experiments described below.

4.5.1.2 Performance comparison - noiseless case

In these experiments we set $\mu = 0$. We vary $m$ between 75 and 170 (i.e. between $[0.15n]$ and $[0.33n]$). We consider different values of $g$ and $s$. We set $k = [0.1n]$. The results are shown in Figures 4.3(a)-4.3(f). There are few interesting outcomes of the experiments. First, for a fixed $g$ the number of measurements necessary to achieve a target NRMSE increases with increasing $s$. In Figure 4.3(b), when $g = [0.75k]$ and $s = [0.1k]$, PMAP needs $m \geq [0.21n]$ to ensure an NRMSE below $3 \times 10^{-4}$. However, when $g = [0.75k]$
Figure 4.2: Dependency of PMAP on different tuning factors. (a) - (b) dependency on $s$, i.e., error in known part of support. We take different $\hat{s}$ around the actual $s$ and compare performance. We fixed $w = 20$ and $\hat{u} = k - g + \hat{s}$. (c) - (d) Average NRMSE (over 100 runs) vs $w$, keeping $\hat{s} = [0.2g]$. In all experiments $m = [0.25n]$, $k = [0.1n]$. 
4.5. Experimental results

Figure 4.3: Noiseless signal recovery as a function of measurements $m$. Performance of different algorithms evaluated for different values of $g$ and $s$. Average NRMSE (over 100 runs) vs $m$ keeping (a) $g = \lceil 0.55k \rceil$, $s = \lceil 0.1k \rceil$, (b) $g = \lceil 0.75k \rceil$, $s = \lceil 0.1k \rceil$, (c) $g = \lceil 0.55k \rceil$, $s = \lceil 0.25k \rceil$, (d) $g = \lceil 0.75k \rceil$, $s = \lceil 0.25k \rceil$, (e) $g = \lceil 0.55k \rceil$, $s = \lceil 0.4k \rceil$, (f) $g = \lceil 0.75k \rceil$, $s = \lceil 0.4k \rceil$. In all experiments $k = \lceil 0.1n \rceil$. 
and $s = [0.4k]$ (Figure 4.3(f)) PMAP needs $m \geq 0.23n$ for the same performance. Second, for a fixed $s$ the number of measurements necessary to achieve a target NRMSE increases with decreasing $g$. In Figure 4.3(b), when $g = [0.75k]$ and $s = [0.1k]$ PMAP needs $m \geq 0.21n$ to achieve a NRMSE below $3 \times 10^{-4}$. However, in Figure 4.3(a), when $g = [0.55k]$, PMAP needs $m \geq 0.225n$ for the same performance. Note that PMAP outperforms all other algorithms. Also performances of modified-CS and weighted-$\ell_1$ are better than CS when $s$ is small. However, modified-CS and weighted-$\ell_1$ perform similarly to CS when $s$ increases (Figure 4.3(e)). Figure 4.4 shows performance of different algorithms as a function of $k$, sparsity of the signal. Note that PMAP outperforms all other algorithms. For $m = [0.25n]$, PMAP works fine if $k \leq [0.11n]$. However, modified-CS and weighted-$\ell_1$ performs well for $k \leq [0.085n]$.

![Figure 4.4: Noiseless signal recovery as a function of sparsity $k$, where $m = [0.25n]$, $s = [0.25k]$.](image)

For reference, to solve a problem with $m = [0.21n], k = [0.1n], g = [0.75k], s = [0.25k]$, the computation time require for CS, modified-CS, weighted-$\ell_1$ and PMAP are 1.54, 2.11, 1.88 and 0.51 sec respectively.

### 4.5.1.3 Performance comparison - noisy data

We set different $\mu$ for different experiments. In all experiments we set $k = [0.1n], s = [0.25k]$. The results are shown in Figure 4.5(a)-4.5(f). As expected performance of
4.5. Experimental results

algorithms decrease with increasing $\mu$. In Figures 4.5(b) and 4.5(d), let us consider when $m = \lceil 0.235n \rceil$, $g = \lceil 0.75k \rceil$. The NRMSE of PMAP are 0.02 and 0.08 when $\mu = 1$ and 3 respectively. Similarly, the NRMSE of modified-CS are 0.2 and 0.25 when $\mu = 1$ and 3 respectively. As before, performance of algorithm decreases with decreasing $g$. For example, PMAP requires $\lceil 0.254n \rceil$ measurement to achieve NRMSE of 0.021 when $g = 0.55k$ (Figure 4.5(a)). PMAP outperforms all algorithms.

4.5.2 Simulations using real data

4.5.2.1 Sparsified MRI image

We use Larynx\(^2\) and Cardiac\(^3\) image sequences in this experiment [65, 66]. All images are $128 \times 128$. The MRI images have sparse representation when we compute its 2D wavelet coefficients using a two-level Daubechies-4 wavelet. Hence, for each image we compute the vector $x$ of wavelet coefficients, and then pre-multiply by a random matrix $\Phi$ consisting of mutually independent, zero mean, unity variance Gaussian entries to obtain the corresponding measurement vector $y$. The value of $m$ is different for different experiments. We use CVX framework [63] for solving the optimization problem in CS, weighted-$\ell_1$ and modified-CS. Since CVX cannot work with large sized problems, we divide the entire images into 4 parts. Each sub-image is $64 \times 64$.

First we consider a single image reconstruction problem. Since a vector of wavelet coefficients $x$ are not strictly sparse, here $T = S(x)$ denotes the 99.95%-energy support of $x$, and as before $k = \text{card}(T)$. In MRI image sequences, we take large number of measurements for the first frame in the sequence (see the following paragraph), and the frame is recovered using conventional CS. Hence we can use the value of $k$ of the first frame as an approximation of $k$ for the following frames. We set $g = \lceil 0.75k \rceil$ and $s = \lceil 0.25k \rceil$. The ‘true’ part of the set $T_g$ i.e. $T_g \backslash T_s$ is constructed by selecting support of the first $g - s$ largest components of $|x|$. The set $T_s$ is generated by drawing $s$ elements uniformly at random from $T^c$. Next we form $T_g = (T_g \backslash T_s) \cup T_s$, which we use in the reconstruction process. In the PMAP reconstruction we use $\hat{s} = \lceil 0.2g \rceil$, $\hat{u} = k - g + \hat{s}$, $\hat{\mu} = 0.001$ and $w = 20$. Figure 4.6(a) shows an example of reconstruction of larynx image when $m = \lceil 0.22n \rceil$. The performance of CS is poor due to

\(^2\)http://home.engineering.iastate.edu/~namrata/research/research.html

\(^3\)http://bisp.kaist.ac.kr/research\_02.htm
Figure 4.5: Noisy signal recovery as a function of measurements $m$. Performance of different algorithms evaluated for different values of $g$ and $\mu$. Average NRMSE (over 100 runs) vs $m$ keeping (a) $g = \lceil0.55k\rceil$, $\mu = 1$, (b) $g = \lceil0.75k\rceil$, $\mu = 1$, (c) $g = \lceil0.55k\rceil$, $\mu = 3$, (d) $g = \lceil0.75k\rceil$, $\mu = 3$, (e) $g = \lceil0.55k\rceil$, $\mu = 5$, (a) $g = \lceil0.75k\rceil$, $\mu = 4$. In all experiments $k = 0.1n$, $s = 0.25k$. 
4.5. Experimental results

Figure 4.6: MRI images reconstruction by different algorithms. $g = [0.75k], s = [0.25k], \hat{\mu} = 0.001$. (a) Actual $128 \times 128$ sparsified Larynx image and its reconstruction by different algorithms when $m = [0.22n]$. (b) Actual $128 \times 128$ sparsified Cardiac image and its reconstruction by different algorithms when $m = [0.24n]$. 
Figure 4.7: Reconstructing $128 \times 128$ MRI images. $g = \lceil 0.75k \rceil$, $s = \lceil 0.25k \rceil$. (a) NRMSE vs $\frac{m}{n}$ for larynx image, (b) NRMSE vs $\frac{m}{n}$ for Cardiac image.
4.6 Conclusion

small number of measurements $m$. Also the image recovered by Modified-CS is quite noisy. Nevertheless, PMAP reconstruction is quite clear. Figure 4.6(b) shows the reconstruction of the cardiac image. In Figure 4.7, we show how NRMSE for different methods vary with $\frac{m}{n}$. In case of larynx image reconstruction (Figure 4.7(a)) PMAP performs much better compared to other algorithms. Weighted-$\ell_1$ and modified-CS performs similarly.

Next we consider reconstruction of MRI image sequences. The measurement vector for the $k$th frame in the sequence is denoted by $y_k$, and the corresponding vector of wavelet coefficients is denoted as $x_k$. We take $m = \lceil 0.5n \rceil$ for the first frame in the sequence, i.e., $y_1 \in \mathbb{R}^{\lceil 0.5n \rceil}$ for the subsequent frames we take a smaller $m$. We recover an estimate $\hat{x}_1$ of $x_1$ from $y_1$ by using conventional CS. For $i > 1$, we estimate $k$ as the cardinality of 99.95%-energy support of $\hat{x}_{i-1}$, and take $T_g$ as the support of the $\lceil 0.5k \rceil$ largest components of $|\hat{x}_{i-1}|$, and use different algorithms to recover the successive frames. Figure 4.8 shows 3rd to 6th frames of true larynx MRI image sequences and their recovered version with $m = \lceil 0.22n \rceil$. Note that PMAP offers much better performance. Figure 4.9 compares the MRI sequences recovery performance of different algorithms. We vary $m$ between $\lceil 0.16n \rceil$ to $\lceil 0.33n \rceil$. For a particular $m$ we reconstruct all frames in the MRI sequence by an algorithm. The average NRMSE over all frames is plotted as a function of $\frac{m}{n}$. We do not include the results of weighted-$\ell_1$ as it performs similarly to modified-CS. Note that PMAP outperforms other algorithms for both MRI sequences.

4.6 Conclusion

We consider a MAP estimation based approach to recover sparse signal when a part of the support of sparse signal is known. A wide range of simulation is performed using both artificial and real data. In contrast to convex relaxation algorithms, the proposed algorithm shows a clear performance improvement.
Figure 4.8: Reconstruction of 128×128 larynx image sequences. (a) Actual sparse sequences, (b) CS reconstruction (c) Modified-CS reconstruction (d) PMAP reconstruction. $m = \lceil 0.22n \rceil, g = \lceil 0.5k \rceil, \hat{\mu} = 0.01$. 
Figure 4.9: Reconstruction of $128 \times 128$ MRI sequences. $g = \lceil 0.5k \rceil, \hat{\mu} = 0.01$. (a) Larynx sequences reconstruction performances of different algorithms. (b) Cardiac sequences reconstruction performances of different algorithm.
Part II

Applications
Chapter 5

Direction-of-Arrival Estimation

5.1 Introduction

Direction-of-arrival (DOA) estimation is a major functional requirement in smart antennas, mobile communication systems, various types of imaging systems, and target tracking applications [7, 77, 78]. Many high-resolution algorithms are available, see [78] and references therein. The popular algorithms can be classified into three broad categories. The algorithm like Capon [79], obtains a non-parametric spatial spectrum by designing a data-adaptive spatial filter bank. The subspace algorithms like MUSIC [80], ESPRIT [81] or weighted subspace fitting [82, 83], exploit the low-rank structure of the noise-free signal. The deterministic maximum likelihood (DML) and stochastic maximum likelihood (SML) [7] methods enjoy excellent statistical properties, but require accurate initialization to ensure convergence to a global minimum. All these methods rely on the statistical properties of the data, and thus, need a sufficiently large number of samples for accurate estimation. In addition, if the sources are strongly correlated, then the covariance matrix of the noise-free data tends to lose rank. This leads to performance degradation, particularly when the measurements are noisy.

Sparse signal representation has been applied for spectral analysis [84, 85, 28, 86, 87, 88, 48]. In [84], a Cauchy-prior is used to enforce sparsity in a temporal spectrum. Jeffs [85] uses an $\ell_p$-norm penalty with $p \leq 1$ in a sparse antenna array design. Gorodnitsky et al. [28] use a recursive weighted least-squares algorithm called FOCUSS for source localization. Fuchs [86, 87] formulates the source localization problem as a sparse recovery problem in the beam-space domain. Cotter [89] combines MMV and matching pursuit (MP)
5. Direction-of-Arrival Estimation

to solve the joint-sparse recovery problem in DOA estimation. Another successful sparse recovery algorithm for DOA estimation is $\ell_1$-SVD [48]. It combines the SVD step of the subspace algorithms with a sparse recovery method based on $\ell_1$-norm minimization. The $\ell_1$-SVD algorithm can handle closely spaced correlated sources if the number of sources is known. However, this method suffers from some performance degradation when the number of sources is unknown.

5.1.1 Contributions

In this chapter we apply the extended JLZA algorithm (Section 3.4.1) for DOA estimation. We consider both narrowband and broadband DOA estimation. First, we pose the narrowband DOA estimation problem as a MMV problem. Subsequently, we consider JLZA-MMM algorithm (Section 3.2.2.3) to deal with the MMV problem with multiple measurement matrices arising in broadband DOA estimation, where the manifold matrices for different frequency bands are different. JLZA-MMM allows us to enforce joint row sparsity in the concatenated signal matrix of all frequencies. This allows a sensor spacing larger than the smallest half-wavelength of the signal, which in turn results in a significant improvement in the DOA resolution performance. This is possible because the aliasing effect is suppressed by enforcing the joint sparsity of the recovered spatial spectrum across the whole frequency range under consideration. The advantages of the proposed algorithm is demonstrated via numerical simulations, achieving high-resolution without the need of a good initialization, and with only a few snapshots. In addition, the actual number of sources need not be known a priori. This method can be applied reliably with both correlated and uncorrelated sources.

5.2 Problem formulation

Consider $k$ narrow-band signals $\{s_j(t)\}_{j=1}^k$ incident on a sensor array, consisting of $m$ omnidirectional sensors. Let

$$y(t) = [y_1(t) \cdots y_m(t)]',$$

where $y_j(t)$ is the signal recorded after demodulation by the $j$ th sensor. Defining

$$s(t) = [s_1(t) \cdots s_k(t)]',$$
and using the narrowband observation model [7, 77], we have

\[ y(t) = A(\theta)s(t) + e(t). \]  

(5.1)

Here \( A(\theta) \) is the manifold matrix, \( \theta \) is the DOA vector containing the directions of arrival of individual signals, i.e. the \( j \) th component \( \theta_j \) of \( \theta \) gives the DOA of the signal \( s_j(t) \), and \( e(t) \) denotes the measurement noise. The manifold matrix consists of the steering vectors \( \{ a(\theta_j) \}_{j=1}^k \):

\[ A(\theta) = [ a(\theta_1) \cdots a(\theta_k) ]. \]

The mapping \( a(\theta) \) depends on the array geometry and the wave velocity, which are assumed to be known for any given \( \theta \). The problem is to find \( \theta \) and \( k \) from \( \{ y_j(t) \}_{j=1}^m \).

### 5.2.1 DOA Estimation as a Joint-Sparse Recovery Problem

We divide the whole area of interest into some discrete set of “potential locations”. It could be a set of pixels/voxels in the near field case, or a grid of directions-of-arrival angles in the far-field case. Consider the far-field scenario, and let the set of all potential DOAs be \( G = \{ \bar{\theta}_1, \ldots, \bar{\theta}_n \} \), where typically \( n \gg k \). The choice of \( G \) is similar to that used in the Capon or MUSIC algorithms. Collect the steering vectors for each element of \( G \) in

\[ \Phi = [ a(\bar{\theta}_1) \cdots a(\bar{\theta}_n) ]. \]

Since \( G \) is known, \( \Phi \) is known and is independent of \( \theta \). Now, represent the signal field at time \( t \) by \( x(t) \in \mathbb{C}^n \), where the \( j \) th component \( x_j(t) \) of \( x(t) \) is non-zero only if \( \bar{\theta}_j = \theta_\ell \) for some \( \ell \), and in that case \( x_j(t) = s_\ell(t) \). Then one has a model

\[ y(t) = \Phi x(t) + \bar{e}(t), \]  

(5.2)

where \( \bar{e}(t) \) is the residual due to measurement noise and model-errors. Since \( k \ll n \), \( x(t) \) is sparse. Note that the equality \( \bar{\theta}_j = \theta_\ell \) may not hold exactly for any \( \ell \in \{1, 2, \ldots, k\} \) in practice. Nevertheless, by making \( G \) dense enough, one can ensure \( \bar{\theta}_j \approx \theta_\ell \) closely, and the remaining modeling error is absorbed in the residual term \( \bar{e}(t) \). In effect, (5.2) lets us pose the problem of estimating \( k \) and \( \theta \) as that of estimating a sparse \( x(t) \). One major advantage of this framework is that it lets us work with an unknown \( k \) using the methodologies in compressive sensing [1]. If there is a reliable algorithm to recover the sparse \( x(t) \) from \( y(t) \) using (5.2), then all but a few components of the final solution \( x(t) \) will have very small
magnitudes, while a few dominant “spikes” in $x(t)$ represent the actual sources. Thus, if the $j$th component $x_j(t)$ is a dominant component in the recovered $x(t)$, then we infer that there is a source with DOA $\hat{\theta}_j$, with an associated signal $x_j(t)$. Finally, the number of these dominant spikes gives $k$.

It is well-known that (5.2) admits a unique $k$-sparse solution $x(t)$ if $k \leq m/2$, and every $m$ columns of $\Phi$ form a basis of $\mathbb{C}^m$. The latter is called the unique representation property, and is closely connected to the concept of an un-ambiguous array. Apart from the limit on $k$, the single snapshot setting in (5.2) has another problem. Until now, there is no good algorithm that can ensure sparse signal recovery in the presence of noise. Since noise is ubiquitous in practical problems, we turn to the so-called joint sparse formulation (see Section 3.2). In practice, we have several snapshots $\{y(t)\}_{t=1}^r$. Algorithms like MUSIC, ESPRIT, and Capon rely on the second-order statistics of the data, and hence require a sufficiently large $r$ so that the statistics can be estimated accurately. Using (5.2), we can write

$$Y := [ y(1) \cdots y(r) ] = \Phi X + E,$$

(5.3)

where $X = [ x(1) \cdots x(r) ]$ is a sparse matrix, and $E = [ \bar{e}_1 \cdots \bar{e}_r ]$. If the DOA vector $\theta$ is time-invariant over the period of measurement, then for all $t$ the non-zero dominant peaks in $x(t)$ occur at the same locations corresponding to the actual DOAs. In other words, only $k$ rows of $X$ are non-zero. Such a matrix is called jointly $k$-sparse. Hence the DOA estimation problem can be posed as the MMV problem [20] of finding a jointly sparse $X$ from $Y$.

5.2.2 Results on the Joint-Sparse Recovery Problem

For now, assume that $E = 0$, and $Y, X$ are real-valued. The Lemma 4 gives sufficient conditions for the existence of a unique solution to the MMV problem. Putting Lemma 4 in the context of the DOA estimation problem, assume that $r > m$, and the matrix

$$S = [ s(1) \cdots s(r) ]$$

has rank $k$. Then the DOAs can be estimated uniquely using the joint sparse framework if $k \leq m - 1$. It is interesting that all the subspace algorithms, e.g., MUSIC or ESPRIT,
have the same limitation. Moreover, even if Capon and other filter-bank approaches do not need \( k < m \), these methods rarely perform well in practice if \( k \geq m \).

As described in Section 3.2 a \( \ell_{2,0} \)-norm based minimization approach can be used to solve the MMV problem. However, since zero norm leads NP-hard problem, different relaxations are used in literature.

### 5.3 \( \ell_1 \)-SVD

\( \ell_1 \)-SVD [48] uses \( \ell_{2,1} \) to solve the MMV problem. In presence of noise, \( \ell_1 \)-SVD algorithm considers the following way to solve \( X \) given \( Y \) in (5.3)

\[
\min_X \| Y - \Phi X \|_F^2 + \varsigma \| X \|_{2,1}
\]  

(5.4)

However, the main problem of \( \ell_{2,1} \) is that the computation time of the optimization in (5.4) increases with increasing \( r \) (see discussion around Figure 3.2(a)). To reduce both the computational complexity and sensitivity to noise, \( \ell_1 \)-SVD uses singular value decomposition (SVD) of the data matrix \( Y \in \mathbb{C}^{m \times r} \). Similar to other subspace algorithms (i.e., MUSIC, Pisarenko) the \( \ell_1 \)-SVD keeps the signal subspace and mold the problem with reduced dimensions into the multiple-sample sparse spectrum estimation problem. Mathematically, this translates into the following representation. Let the SVD of \( Y \) be given as

\[
Y = ULV'.
\]  

(5.5)

Keep a reduced \( m \times k \) dimensional matrix \( Y_S \), which contains most of the signal power

\[
Y_S = ULB_k = YVB_k
\]

\[
= (\Phi X + E)V B_k
\]  

(5.6)

where \( B_k = [I_k \ 0]' \), \( 0 \) is a \( k \times (r - k) \) matrix of zero. Let \( X_S = XV B_k \) and \( E_S = EV B_k \), then (5.3) becomes

\[
Y_S = \Phi X_S + E_S
\]  

(5.7)

Then the optimization problem in (5.4) can be recast as

\[
\min_{X_S} \| Y_S - \Phi X_S \|_F^2 + \varsigma \| X_S \|_{2,1}
\]  

(5.8)
The optimization problem can be solved using an efficient second-order-cone (SOC) programming algorithm [90].

However, one of the drawbacks of this approach is its computational cost. For optimizing the objective function (5.8) in SOC framework using an interior point implementation the cost is $O((k \times n)^3)$. The computation cost increases with increasing number of source. Another problem of this approach is that the formulation uses information about the number of sources $k$. However $k$ is unknown in practice. Experimental results demonstrate that $\ell_1$-SVD can not perform equally when $k$ is unknown. Due the limitations of $\ell_1$-SVD, we turn our attention to $\ell_{2,0}$ approximation based algorithm for DOA estimation.

5.4 DOA estimation using JLZA

5.4.1 Narrowband DOA estimation

The joint-sparse DOA representation in (5.3) is same to the representation in (3.37). Hence we consider the formulation in (3.38) to solve $X$ given $Y$, i.e.,

$$X_*(\sigma) = \arg \max_X \ L_\sigma(X),$$

$$L_\sigma(X) := F_\sigma(X) - \lambda \| Y - \Phi X \|_F^2,$$

for small values of $\sigma$. We can use the extension of JLZA proposed in Section 3.4 to resolve DOA. We need to tune the values of two parameters for implementing the algorithm. They are $\sigma_{\text{min}}$ and $\lambda$. Their values depend on noise. Experimental results suggest that $\lambda = 5$ and $\sigma_{\text{min}} = 0.001$ are the good choice for 5 dB to 20 dB SNR.

5.4.1.1 Acceleration via QR Factorization

Typically, the matrix $X \in \mathbb{C}^{n \times r}$ is large, as $n$ is a large number (we need $n = 180$ to achieve $1^\circ$ spatial resolution). If the number of data samples $r$ is large, the algorithm may become slow. To accelerate the algorithm, we use the QR factorization $Y/\sqrt{r} = RQ$, where $R \in \mathbb{C}^{m \times m}$ is a nonsingular upper triangular matrix, and $Q \in \mathbb{C}^{m \times r}$ is such that $QQ^* = I$. When $E = 0$, then

$$\text{row span}\{X\} \subset \text{row span}\{Q\}. \quad (5.9)$$

Consequently, $\| Y - \Phi X \|_F^2 = \| R - \Phi \tilde{X} \|_F^2$, where $\tilde{X} = XQ^* \in \mathbb{C}^{n \times m}$ must be jointly row-sparse, and is of significantly smaller size than $X$. Hence, it is more efficient to estimate $\tilde{X}$
via the successive solutions

$$\hat{X}_*(\sigma) = \arg \max_{\hat{X}} F_\sigma(\hat{X}) - \lambda \|R - \Phi \hat{X}\|_F^2$$

(5.10)

for decreasing values of $\sigma$. If $E \neq 0$, then (5.9) does not hold. Nevertheless, for a realistic noise intensity, the rows of $X$ have very small components in the nullspace of $Q$. Hence, we can use (5.10) without noticeably affecting the performance.

### 5.4.1.2 Grid Enhancement

We assume that all DOAs fall on the discretize grid. In practice, this is not always true. However, simulation results in Figures 5.2(a) and 5.2(b) demonstrate that JLZA can approximately locate the sources even if the DOAs do not fall on the grid. Hence, a dense grid is necessary to achieve fine resolution, but making the grid too dense results in large computation time. This motivates the adaptive grid-refinement strategy proposed in [48]. We start with a fairly coarse grid and obtain an approximate spatial spectrum. Subsequently, we make the grid finer around the approximate source locations and refine the estimates. Let $\Phi_\varepsilon$ be the manifold matrix for a grid density $\varepsilon$ with an associated estimate $\hat{X}_\varepsilon$. Then, we compute the residual magnitude $\xi_\varepsilon = \|Y - \Phi_\varepsilon \hat{X}_\varepsilon\|_F$. We continue the iterative grid refinement process as long as a larger $\varepsilon$ results in a smaller $\xi_\varepsilon$. In practice, if we make the grid too dense, the residual magnitude actually increases. This is supported by the observations made by Donoho in [91], and Chen et al. [8, 92]. These articles report cases where sparsity allows one to resolve beyond the nominal Rayleigh spacing limit, but eventually the sensitivity to errors grows large, depending on the number of sources within one Rayleigh spacing. Because it is not possible to make the grid infinitely fine, JLZA cannot obtain the exact DOAs even if the data are noise-free. On the other hand, popular methods like MUSIC, ESPRIT, etc, can find the exact DOAs when the data are noise-free. However, in practice the data are noisy, and it is shown in Section 5.5 that JLZA is much more robust to noise compared to algorithms like MUSIC or ESPRIT.

### 5.4.2 Broadband DOA estimation

A standard way to deal with broadband DOA estimation problem is to split the broadband signal into several narrowband signals by passing the observed signal through a bank of narrowband filters [93, 94, 95]. Subsequently, the narrowband model (5.3) is applied
to each narrowband filter output. Suppose that we have narrowband data at frequencies \(\{\omega_j\}_{j=1}^{J}\), and let \(\Phi_j\) be the “over-complete” manifold matrix at frequency \(\omega_j\). Then, the narrowband model at frequency \(\omega_j\) is of the form

\[
Y_j = \Phi_j X_j + E_j, \quad j \in \{1, 2, \ldots, J\}.
\]

Here, \(E_j\) is the additive noise at frequency \(\omega_j\) and \(X_j\) is the jointly row-sparse signal matrix at frequency \(\omega_j\). Now,

\[
X := \begin{bmatrix} X_1 & X_2 & \cdots & X_J \end{bmatrix}
\]

is jointly row-sparse. This is because if \(X_j[\ell,:]\) is non-zero for some \(\ell\), then there is source signal at frequency \(\omega_j\) at direction \(\theta_{\ell}\). Therefore, we would expect signals at other frequencies from the direction \(\tilde{\theta}_\ell\) as well, making \(X_j[\ell,:]\) non-zero for all \(j\). Hence, the broadband model is same to the MMV with multiple measurement matrices model in Section 3.2.2.3 (see Section 3.4 for extension in complex case). We can consider the optimization problem in (3.23). Consequently, we use the JLZA-MMM algorithm proposed in Table 3.3. In numerical experiments of the wideband case, we vary the SNR from 10 dB to 30 dB and fix \(\lambda = 10\) and \(\sigma_{\min} = 0.001\).

By enforcing \(X\) to be jointly row sparse, we can allow sensor spacing larger than the smallest half-wavelength in the signal. This results in a significant improvement in the DOA resolution performance, (see numerical examples in Figure 5.8). Having larger than half-wavelength sensor spacing with narrowband signals causes spatial aliasing. As a result, one detects nonexistent sources from spurious DOAs, which are simply the aliased images of the actual sources. By making \(X\) jointly row sparse, JLZA looks for directions in which it finds dominant peaks over the whole frequency range. As a result, it can suppress the aliasing effect.

### 5.5 Simulation Results

We compare the performance of JLZA with \(\ell_1\)-SVD [48], MUSIC [80], Capon’s method [79], and MMV FOCUSS (M-FOCUSS) [20].

#### 5.5.1 Narrowband signal

We consider a uniform linear array (ULA) with \(m = 8\), where the sensors are separated by a half wavelength of the narrowband signal. Each narrowband signal is generated
from a zero mean Gaussian distribution. JLZA starts with a uniform grid with 1° resolution, i.e. \( n = 180 \), and \( \Phi \in \mathbb{C}^{8 \times 180} \). The measurements are corrupted by temporally and spatially uncorrelated zero-mean noise sequence. We assume \( k \) is unknown. Since MUSIC, \( \ell_1 \)-SVD needs information about the value of \( k \), we use the value of \( m - 1 \) instead of \( k \). We point out that the performance of \( \ell_1 \)-SVD might improve if the number of sources is assumed to be known.

Figure 5.1(a) shows the DOA estimation results when two uncorrelated sources are placed at 10° and 15°. We take \( r = 50 \) and SNR = 5 dB. Only JLZA can resolve sources. Note that \( \ell_1 \)-SVD can produces peaks around the actual locations. However, \( \ell_1 \)-SVD provides additional peak around −30°. Figure 5.1(b) shows the results when we simulate two strongly correlated sources at 10° and 20° with a correlation coefficient 0.99. Note that JLZA can locate the sources clearly, while among the other methods, only \( \ell_1 \)-SVD is able to locate the sources. However, as before, \( \ell_1 \)-SVD detects several spurious sources as well. In our experience, \( \ell_1 \)-SVD detects such spurious sources if the number of sources is unknown. Also algorithms like Capon or MUSIC can resolve the sources only if the number of snapshots \( r \geq 140 \).

In Figures 5.2(a) and 5.2(b), we show the spatial spectrum plots obtained by various methods when we simulate two uncorrelated sources at 10° and 15.3°. We take \( r = 50 \) and SNR = 10 dB. In Figure 5.2(a), we show the results when the grid resolution used in JLZA is 1°. Although the DOA of the second source is not in its dictionary, JLZA can approximately locate the second source at 16°. The result of subsequent grid enhancement is shown in Figure 5.2(b), where JLZA uses 0.5° grid resolution. JLZA locates the second source at 15.5°. In this example, further grid enhancement does not improve the results. Also, note in Figures 5.2(a) and 5.2(b) that among the other algorithms only \( \ell_1 \)-SVD is able to resolve the sources, but \( \ell_1 \)-SVD also detects two other spurious sources.

Figures 5.3(a)- 5.3(b) illustrates the robustness of JLZA when the sensor locations are not known exactly. Let the vector of ideal sensor locations be \( z \in \mathbb{R}^m \). Due to manufacturing defects, the vector \( \hat{z} \) of actual sensor locations is given by

\[
\hat{z} = z + \hat{e} \tag{5.11}
\]

where \( \hat{e} \) is an additive noise with a variance \( \mu^2 I_m \). Hence we generate the data for \( \hat{z} \), but carry out the DOA estimation using the vector \( z \) of ideal locations. Figure 5.3(a) and 5.3(a) shows typical spatial spectra obtained by various methods when \( \mu = 0.02 \) and \( \hat{\mu} = 0.05 \).
Figure 5.1: Resolving sources. DOA grid resolution 1°. (a) Uncorrelated sources at 10° and 15°, SNR 5 dB, $r = 50$. (b) Correlated sources at 10° and 20°, SNR 10 dB, $r = 50$. 
Figure 5.2: Source estimation when DOA does not fall on the grid. Uncorrelated sources at 10° and 15.3°, SNR 10 dB, $r = 50$. (a) DOA grid resolution 1°, (b) DOA grid resolution 0.5°.
Figure 5.3: DOA estimation with erroneous information about sensor locations. Uncorrelated sources at $10^\circ$ and $15^\circ$, SNR 10 dB, $r = 50$, grid resolution = $1^\circ$. (a) $\hat{\mu} = 0.02$. (b) $\hat{\mu} = 0.05$. 
respectively. There are two uncorrelated sources at 10° and 15°. Note that JLZA obtains biased estimates with sensor error. The bias depends on $\hat{\mu}$, as expected. If $\hat{\mu}$ increases, the bias increases. For example, JLZA produces peak at 10° and 15° when $\hat{\mu} = 0.02$. However, the sources are located at 10° and 16° when $\hat{\mu} = 0.05$. Nevertheless, JLZA is still able to resolve sources with reasonable accuracy. As far as the other methods are concerned, none of these are able to resolve the sources.

Some DOA estimation algorithms work well when the number of sources is small, but break down as the number increases. Lemma 4 suggests that for an $m$-sensor array, JLZA can detect at least $m - 1$ sources, provided that $X$ has rank $m - 1$. In our simulations, we have found that at 10 dB SNR, JLZA can resolve 7 uncorrelated sources using an 8-sensor ULA with half-wavelength sensor spacing, provided that minimum spatial separation between the sources is at least 10° (Figure 5.4(a)). However, it is interesting to note what happens when signals from 8 sources impinge to 8 sensors. In some settings (see Figure 5.4(b)) JLZA is still able to locate the sources. It is interesting to investigate how JLZA performs when the sources are highly correlated, whereby $X$ tends to lose rank. Such a situation is depicted in Figure 5.5(a), where we simulate 6 strongly correlated sources at $-60°, -30°, -15°, 15°, 30°$, and $60°$, where the correlation coefficient between each couple is 0.99. The SNR is 10 dB, and $r = 100$. As can be seen in Figure 5.5(a), JLZA can detect and resolve all the sources. It is interesting to notice that $\ell_1$-SVD detects the sources with some bias, but as in other cases, it also detects another spurious source at 0°.

For each method, there is a limit beyond which it fails to resolve DOAs. In order to detect when JLZA fails, we carry out an experiment where we simulate sources at 10° and 11°, and decrease the SNR. JLZA fails when the SNR is 5 dB. In Figure 5.5(b), we note that JLZA produces a blunt peak around the location, and detects a distributed source.

Next, we investigate the impact of the number of snapshots on the performance of JLZA. Here, we simulate two strongly correlated sources with a correlation coefficient of 0.99 at 10° and 20°. First, we keep the SNR fixed at 10 dB and vary the number of snapshots $r$. For each value of $r$, we carry out 100 independent simulations, and the results are shown in Figure 5.6(a), where we plot the frequency at which the different algorithms detect and locate the sources correctly against $r$. Note that JLZA outperforms the other methods by a large margin. While $\ell_1$-SVD requires about 85 snapshots to resolve the sources consistently, JLZA requires only about 35 snapshots. The plots for MUSIC and CAPON are not shown, as they are unable to resolve the sources when $r < 140$. A similar plot
Figure 5.4: Maximum number of uncorrelated DOA estimation capability by different algorithms, $r = 100$, SNR = 10 dB, grid resolution = 1°. (a) Resolving uncorrelated 7 sources by using 8 sensors, (b) Resolving uncorrelated 8 sources by using 8 sensors.
Figure 5.5: DOA estimation. Grid resolution = 1°. (a) Resolving correlated 6 sources by using 8 sensors. SNR 10 dB, $r = 100$, (b) An example where JLZA fails to resolve DOA. Uncorrelated sources at $10^\circ$ and $11^\circ$, SNR 5 dB, $r = 50$. 
5. Direction-of-Arrival Estimation

![Graph](image)

Figure 5.6: Frequency of detection of correlated sources. (a) Against number of snapshots, SNR = 10 dB. (b) Against input SNR, $r = 100$.

<table>
<thead>
<tr>
<th>Table 5.1: Computation time comparison.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samples</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>50</td>
</tr>
<tr>
<td>75</td>
</tr>
<tr>
<td>100</td>
</tr>
<tr>
<td>125</td>
</tr>
<tr>
<td>150</td>
</tr>
</tbody>
</table>

depicting the impact of SNR on various algorithms is shown in Figure 5.6(b), where we fix the number of snapshots at $r = 100$ and vary the SNR. In this case, we notice that JLZA outperforms $\ell_1$-SVD by a narrow margin.

Finally, in Table 5.1 we compare the computation time for different algorithms. Here, we consider the case of resolving two uncorrelated sources located at $10^\circ$ and $15^\circ$, while the number of snapshots is varied from 50 to 150. The SNR is fixed at 10 dB (computation time is similar when sources are correlated, or when the number of sources increases). For JLZA computation time includes the time taken to compute the QR factorization of $Y$. As can be seen in Table 5.1, $r$ has little effect on the computation time of JLZA and $\ell_1$-SVD. While MUSIC is fastest among all, $\ell_1$-SVD turns out to be the slowest. However, it is important to note that the performance of MUSIC is much worse than that of JLZA or $\ell_1$-SVD.
5.5. Simulation Results

5.5.2 Broadband signal

We simulate the signal received by an 8-sensor ULA when $k$ radio-frequency broadband signals are incident on the array. Each source consists of 10 sinusoids with frequencies randomly chosen from the interval $[1, 2.5]$ GHz. The received signal is sampled at 7.5 GHz. The sampled data is filtered through a bank of first-order bandpass filters of the form

$$H_\omega(z) = \frac{(1 - q)e^{i\omega}}{z - qe^{i\omega}}$$

(5.12)

It can be shown that $H_\omega$ is a narrow-band filter centered at digital frequency $\omega$ (which is related to the analog frequency via the standard relationship). The bandwidth of the filter is controlled by $q$, where $0 < q < 1$. Taking $q \to 1$ makes the bandwidth smaller, but makes the filter less stable. We take $q = 0.99$. The filterbank consists of 50 filters with center frequencies uniformly distributed over the interval $[1, 2.5]$ GHz. In Figure 5.7, we show the results when the sensor separation equals the half-wavelength associated with 2.5 GHz. We simulate 3 sources at $-20^\circ$, $-10^\circ$, $20^\circ$, set SNR = 20 dB with $r = 100$ snapshots. Capon’s method almost fails to resolve the sources at $-20^\circ$, $-10^\circ$, and $\ell_1$-SVD also gives noisy results. However, JLZA-MMM resolves all the sources clearly.

Next we increase the sensor spacing to 2.5 times the wavelength associated with 2.5 GHz, and consider three sources located at $-20^\circ$, $-15^\circ$, $20^\circ$. In this setting, both $\ell_1$-SVD and Capon suffer from severe aliasing, and are unable to give any meaningful results. However, as can be seen in Figure 5.8(a), JLZA-MMM can suppress the aliasing effects by enforcing joint sparsity over the whole frequency range.

Next, we examine impact of sensor separation on the resolution performance of JLZA-MMM. We keep two sources fixed at $-20^\circ$, $20^\circ$, and move the third source from $0^\circ$ toward the source at $-20^\circ$. The minimum spacing $\delta$ between the first and the third sources at which JLZA-MMM can still separate all three sources turns out to be related to the sensor spacing $\Delta$ via an approximate empirical relation

$$\delta = 8 - 1.33(\Delta - 0.5), \quad 0.5 \leq \Delta \leq 3.5.$$  

where $\delta$ is expressed in degrees and $\Delta$ is expressed in the number of wavelengths associated with the 2.5 GHz frequency. Fig 5.9 shows the result. When the sensor spacing is above 4 times the 2.5-GHz wavelength JLZA-MMM starts suffering from some aliasing effects. Such a scenario is depicted in Figure 5.10(a) where the sensor spacing is 5 times the 2.5-GHz
wavelength. Note that JLZA-MMM is still able to resolve three sources at $20^\circ$, $-15^\circ$ and $-20^\circ$. However, it also detects another spurious source at $0^\circ$.

5.6 Conclusion

In this chapter, we pose the DOA estimation problem as a multiple measurement vectors (MMV) problem and exploit joint sparsity to effectively resolve source locations. In particular, we apply the JLZA algorithm to resolve the DOA estimation problem. We extend our idea from a narrowband to wideband scenario by developing a novel approach to MMV problem with multiple measurement matrices. The proposed algorithm is tested on numerically simulated data. In comparison to other well known methods, JLZA shows clear improvements in estimation performance. The algorithm can perform well with a limited number of snapshots, and does not require any knowledge about the number of sources. The algorithm also exhibits better estimation performance for highly correlated sources. It is found that JLZA can resolve more sources than other state-of-the-art approaches. Furthermore, JLZA is very efficient in suppressing aliasing effects when the sensor spacing is larger than the half-wavelength associated with the highest frequency. This turns out to be a great advantage, as JLZA can allow as large as 3 wavelength sensor spacing to achieve very high resolution performance.

It is noted that to accelerate the algorithms, one can beneficially use a reformulation using the QR factorization (see Section 5.4.1.1), and although this only holds for $E=0$, it is stated that for realistic noise intensities, this will be an accurate approximation. Clearly, this would depend on the application, but it is a reasonable approximation, and it is noted that even if the matrix is thus perturb somewhat, the nullspace of $Q$ will change relatively slowly with this perturbation.
5.6. Conclusion

Figure 5.7: Broadband DOA estimation results for three sources at $-20^\circ$, $-10^\circ$, $20^\circ$ using an 8-sensor ULA, $r = 100$ snapshots, SNR = 20 dB. Sensor spacing is the half-wavelength associated with 2.5 GHz.
Figure 5.8: Use of JLZA-MMM in the presence of spatial aliasing with three sources located at $-20^\circ$, $-15^\circ$, $20^\circ$. The sensor spacing equals 2.5 times the smallest wavelength in the broadband signal.
Figure 5.9: Resolution of JLZA-MMM in localizing two closely spaced sources as a function of sensor spacing. Half wavelength spacing is 6 cm corresponding to 2.5 GHz. SNR=20dB and $r = 100$. 
Figure 5.10: Use of JLZA-MMM in the presence of spatial aliasing with three sources located at $-20^\circ$, $-15^\circ$, $20^\circ$. The sensor spacing equals 5 times the smallest wavelength in the broadband signal.
Chapter 6

Spectrum Analysis from Irregularly Sampled Data

6.1 Introduction

Consider a complex-valued continuous-time signal

\[ y(t) = \sum_{i=1}^{k} a_i e^{i\Omega_i t}, \]  

(6.1)

which is being sampled at time instants \( \{t_i\}_{i=0}^{m-1} \). The problem is to estimate the complex-valued amplitudes \( \{a_i\}_{i=1}^{k} \) and the frequencies \( \{\Omega_i\}_{i=1}^{k} \) from the samples \( \{y(t_i)\}_{i=1}^{m} \). This setting is motivated by some applications, experimental conditions cause irregular sampling. For example, earth’s daily rotation and annual revolution often causes periodic gaps in the observation. Prewhitenning techniques estimate frequencies by iterative deconvolution of the observed spectrum. Ad-hoc refinements such as CLEAN [96] or CLEANEST [97] has been used to improve the efficiency of prewhitening techniques. Recently, spectral analysis has been modeled as a linear inverse problem, and it is used as an efficient alternative to parametric methods to achieve high resolution [98, 84, 28, 99]. The spectrum is modeled according to (6.1), and spectral lines are estimated by locating a few nonzero values in the corresponding amplitude vector, i.e., by reconstructing a sparse vector. This makes (6.1) is well suited for compressed sensing (CS) [1, 2].
6. Spectrum Analysis from Irregularly Sampled Data

6.1.1 Contributions

In this chapter we pose the frequency estimation problem as a sparse signal recovery problem. We found that ISL0 based algorithms (Section 3.2.2) can use effectively for frequency estimation. In fact, ISL0 requires a few FFTs and IFFTs for frequency estimation. This makes the algorithm very fast. Consequently we evaluate the performance of different frequency estimation algorithms.

6.2 Spectrum Analysis of Single Signal

6.2.1 Sparse Representation

Let $\delta$ be the largest real number such that there exits integers $d_1, \ldots, d_{m-1}$ as

$$t_i = t_0 + d_i \delta. \tag{6.2}$$

Note that $\delta$ can even be very small depending on the set $\{t_i\}_{i=0}^{m-1}$. Then

$$y(t_i) = \sum_{\ell=1}^{k} a_\ell e^{i\omega_\ell(t_0 + d_i \delta)} = \sum_{\ell=1}^{k} b_\ell e^{i d_i \omega_\ell}$$

where we define $d_0 = 0$, $\omega_\ell = \delta \omega_\ell$, $\ell = 1, \ldots, k$, and

$$b_\ell = a_\ell e^{i\overline{\omega_\ell}t_0}.$$

Now choose $n \geq d_{m-1}$, and consider a uniform frequency grid

$$G = \{\bar{\omega}_\ell = 2\pi \ell/n : \ell = 0, 1, \ldots, n-1\},$$

and define the DFT matrix $\Psi \in \mathbb{C}^{n \times n}$ such that

$$[\Psi]_{j,\ell} = \{e^{i(j-1)\omega_\ell} \}, \quad j, \ell \in \{1, \ldots, n\} \tag{6.3}$$

There may not exist any $\ell \in \{0, 1, \ldots, n-1\}$ such that a given $\omega_j = \bar{\omega}_\ell$, or in other words, it is likely that $\omega_j$ may not lie on the grid $G$. However, making $n$ large enough, the grid $G$ becomes sufficiently dense. Then for every $i \in \{1, \ldots, k\}$, we have $\omega_i \approx \bar{\omega}_\ell$ for some $\ell \in \{0, 1, \ldots, n-1\}$. Then we can write

$$z := [y(t_0) \ y(t_0 + \delta) \ \cdots \ y(t_0 + (n-1)\delta)]' = \Psi x + \bar{e},$$

If any $(t_i - t_0)$ is strictly irrational then $\delta$ does not exists. However, for the practical purposes one can work with an arbitrary close rational approximation.
where $\bar{e}$ accounts for measurement noise and unmodelled residuals; the components of $x$ are mostly zero. The $\ell$-th component $x_\ell$ is non-zero only if $\bar{\omega}_\ell = \omega_i$ for some $i \in \{1, \ldots, k\}$. This makes $x$ as a sparse vector, and the locations of the dominant peaks in $x$ indicates the frequencies present in the data, while the dominant values give the complex-valued amplitudes.

If we know $z$ we could estimate $x$ by inverting $\Psi$, as the DFT matrix is nonsingular. However, we have

$$y := [y(t_0) \ y(t_1) \ \cdots \ y(t_{m-1})]' = \Theta x + \Theta \bar{e},$$

(6.4)

where $\Theta \in \mathbb{R}^{m \times n}$ is formed such that the $i$-th row of $\Theta$ is the same as the $n_i + 1$-th row of $n \times n$ identity matrix. Note that $\Phi$ can be seen as a random partial Fourier matrix and in that case $\Phi$ has a small restricted isometry constant [100]. In order to guarantee unique identifiability of $x$ from $y$ we must have

$$m \leq n/2,$$

The ability to recover $x$ uniquely from $y$ is not sufficient to guarantee identifiability of the frequencies $\{\bar{\omega}_\ell\}_{\ell=1}^k$. Any sampling process leads to aliasing. Suppose that it is possible to recover the sparse $x$ from $y$, then we have estimates of $\{\omega_\ell\}_{\ell=1}^k$. These can be used to uniquely identify the continuous-time frequencies $\{\bar{\omega}_\ell\}_{\ell=1}^k$ only if we know a feasible set $\mathcal{O}$ a priori, such that $\bar{\Omega}_\ell \subset \mathcal{O}, \ell = 1, \ldots, k$. In addition, the intersection between $\mathcal{O}$ and any of its aliased version should be empty, i.e. $\mathcal{O} \cap (\mathcal{O} + 2\pi s/\delta) = \emptyset$ for every nonzero $s \in \mathbb{Z}$. We can allow $\mathcal{O}$ to be made up of several smaller mutually disjoint intervals. The simplest (and most popular) examples of $\mathcal{O}$ are $\mathcal{O} = [-\pi/\delta, \pi/\delta]$ and $\mathcal{O} = [0, 2\pi/\delta]$. However, more complicated cases like $\mathcal{O} = [0, \pi/\delta] \cup (3\pi/\delta, 4\pi/\delta)$ can also be allowed. On the other hand, $\mathcal{O} = [-2\pi/\delta, 2\pi/\delta]$ is an example where the identifiability is lost due to aliasing.

In noise-free case, (i.e., $e = 0$) the optimization problem to solve $x$ given $y$ in (6.4) can be written as

$$x_* = \arg \min_x \|x\|_0 \text{ subject to } y = \Phi x$$

(6.5)

which is NP-hard. To deal with the issue different relaxation has been proposed.
6.2.2 SparSpec

SparSpec [49] proposes an alternative approach of solving (6.5) to estimate frequencies in the time series. For noisy data the problem in (6.5) is relaxed as [5]

\[ x_\star = \arg \min_x \varsigma \|x\|_1 + \frac{1}{2} \| \Phi x - y \|_2^2 \]  

(6.6)

where \( \varsigma \) balances between noise and sparsity.

In the case of real-valued variable \( x_i \), the minimization of criterion of (6.6) can be written as a constrained quadratic optimization problem [101], which can be solved using standard algorithms from many programming libraries. However, in this case, variable \( x_i \) is complex-valued. The function in (6.6) is not differentiable for every vector containing a zero value (the complex modulus function is not differentiable at zero), and hence the authors in [49] avoided classic gradient-based descent techniques.

To compute the solution of (6.6), SparSpec considers an iterative coordinate descent (ICD) procedure [102], which consists in performing successive one-dimensional minimization steps with respect to each complex-valued parameter \( x_i \). As problem (6.6) is convex, the convergence of such algorithm toward the minimum of the criterion is ensured. The advantage of the particular ICD approach (used in SparSpec) is that every scalar minimization has an explicit solution that can be computed instantaneously. Let us define

\[ \mathcal{E}_i = y - \sum_{j \neq i} \Phi[:,j] x_j; \quad 1 \leq i, j \leq n \]

and consider the optimization problem (6.6) only as a function of \( x_i \). Then SparSpec can be viewed as a particular application of the Block Coordinate Relaxation (BCR) algorithm proposed in [103]. It is shown in [104] that

\[ x_{i \min} = \arg \min_{x_i} \varsigma \|x\|_1 + \frac{1}{2} \| \Phi x - y \|_2^2 \]

\[ = T_\varsigma^s(\Phi[:,i]^* \mathcal{E}_i) \]  

(6.7)

where \( T_\varsigma^s \) is the complex soft shrinkage function [3]. For \( \forall z = ae^{i\theta} \) the function \( T_\varsigma^s \) is defined as

\[ T_\varsigma^s(z) = \begin{cases} 
(a - \varsigma)e^{i\theta} : & \text{if } a > \varsigma, \\
0 : & \text{otherwise.} 
\end{cases} \]  

(6.8)

Table 6.1 summarizes the SparSpec algorithm.
6.2. Spectrum Analysis of Single Signal

Table 6.1: SparSpec algorithm for minimizing (6.6)

<table>
<thead>
<tr>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set ( x = \Phi'(\Phi\Phi')^{-1}y ).</td>
</tr>
</tbody>
</table>

Repeat

for \( i = 1 : n \)

2. \( \mathcal{E}_i = y - \sum_{j \neq i} \Phi[:,j]x_j \);

3. \( x_i = T_{\mathcal{C}}(\Phi[:,i]^*\mathcal{E}_i) \).

end

Until: a stopping criterion is satisfied.

6.2.3 An approach Based on \( \ell_0 \) Approximation

Let us consider the following optimization problem to estimate \( x \) from noisy observation \( y \) in (6.4):

\[
x^*_p = \arg\min_x \|x\|_p + \frac{\lambda}{2}\|y - \Phi x\|_2^2.
\]  

(6.9)

The parameter \( \lambda \) controls the tradeoff between the sparsity of the signal and the residual energy [8]. For \( p = 2 \), differentiating (6.9) and equating to zero we have

\[
x^{(2)}_* - \lambda \Phi^*(y - \Phi \hat{x}) = 0 \Rightarrow x^{(2)}_* = \lambda (I_n + \lambda \Phi^* \Phi)^{-1} \Phi^* y,
\]  

(6.10)

After simplifying (6.10) using the matrix inversion lemma gives

\[
x^{(2)}_* = \lambda [I_n - \Phi^*(1/\lambda + \Phi \Phi^*)^{-1} \Phi] \Phi^* y = \frac{\lambda}{\lambda + 1} \Phi^* y.
\]

Note that in above we use \( \Phi \Phi^* = I_m \). Now \( \Phi^* y = \Psi^* (\Theta^* y) \), which is the DFT of \( \Theta^* y \). Also if there is no noise in the data \( y \) then one takes \( \lambda \to \infty \), and thus \( x^{(2)}_* \) is same as the DFT of \( \Theta^* y \). In any case one can use the FFT algorithm to compute \( x^{(2)}_* \).

The minimum two-norm solution \( x^{(2)}_* \) is not sparse in general. Sparsity is best enforced by making \( p = 0 \). Since the choice \( p = 0 \) leads to a NP-hard problem, we consider the following way to solve \( x \) given \( y \) in (6.4):

\[
x_*(\sigma) = \arg\max_x \quad L_\sigma(x),
\]

\[
L_\sigma(x) := F_\sigma(x) - \frac{\lambda}{2}\|y - \Phi x\|_2^2.
\]  

(6.11)

for small values of \( \sigma \), where \( F_\sigma \) is defined in (2.19). In fact, \( x_* \) admits a closed form solution as \( \sigma \to \infty \). To see this first note that \( \lim_{\sigma \to \infty} f_\sigma(\alpha) = 1 \). Using this in (2.20) we have

\[
\lim_{\sigma \to \infty} x_*(\sigma) = \Phi^* (\Phi \Phi^*)^{-1} y = \Psi^*(\Theta^* y)/n,
\]
the DFT of $\Theta^*y$.

Note that equation (6.11) is the SMV version of the MMV problem in (3.38). Hence we use similar procedure of Section 3.4.1 to evaluate JLZA. A sequential optimization method is used to maximize $L_{\sigma_{\min}}$. According to Lemma 8, it can be shown that $L_{\sigma}(X)$ is increasing along $\zeta(x) - x$, where the mapping $\zeta: \mathbb{C}^n \to \mathbb{C}^n$ is defined as

$$\zeta(x) = \lambda \left[ W(x)/\sigma^2 + \lambda \Phi^*\Phi \right]^{-1} \Phi^*y, \tag{6.12}$$

where

$$W(x) = \text{diag}\{f_\sigma(|x_1|), \ldots, f_\sigma(|x_n|)\}.$$ 

Again, denoting $\mathcal{K} = \lambda \sigma^2$, and using the matrix inversion lemma in (6.12) one can verify that

$$\zeta(x) = W^{-1}\Psi^*\Theta^*[I_m/\mathcal{K} + \Theta\Psi W^{-1}\Psi^*\Theta^*]^{-1}y. \tag{6.13}$$

It is tempting to use (6.12) to set up a fixed point iteration to solve $x^*$. The algorithm in Table 3.2 is used here. Since we are using SMV version of the extended JLZA algorithm in Table 3.2, we call it extended ISL0 (EISL0).

### 6.2.3.1 A Direction to Fast Computation

Since $W$ is a diagonal matrix, it follows using (6.3) that

$$[\Psi W^{-1}\Psi^*]_{q,\ell} = \sum_{j=1}^n w_j e^{i(q-\ell)\omega_j-1} = [\Psi w]_{q-\ell},$$

where $w = [1/W_{1,1} \ 1/W_{2,2} \ \cdots \ 1/W_{n,n}]^T$. Consequently, $\Psi W^{-1}\Psi^*$ is a Toeplitz matrix. Also, $\Psi w$ can be computed efficiently using IFFT. The time complexity of FFT is $O(n \log n)$. As $\Theta$ is a row-selector, $R := \Theta\Psi W^{-1}\Psi^*\Theta^*$ is a just small block of $\Psi W^{-1}\Psi^*$. Next, we solve the $m \times m$ positive definite system of equations

$$[R + I_m/\mathcal{K}]z = y. \tag{6.14}$$

The complexity of solving the system be $O(m^2)$. Subsequently, to compute $\zeta(x)$ in (6.13), we calculate $\hat{z} = \Psi^* (\Theta^*z)$ by computing the FFT of $\Theta^*z$. Note that forming $\Theta^*z$ does not require any multiplication. Finally, as $W$ is diagonal, we need $n$ multiplications to compute $W^{-1}\hat{z}$. Hence, in computing (6.13), one requires $O(n + 2n \log n + m^3)$ flops to compute $\zeta(x)$. Note that for uniform sampling $R + I_m/\mathcal{K}$ is Toeplitz, and we can solve (6.14) using a fast Toeplitz system solver.
6.3 Coherent Spectrum Analysis

6.3.1 Background

In the last section we described different methods for estimating frequencies from an irregularly sampled signal. In a related problem one is interested in estimating the frequencies present in a set of signals, which share the same frequencies [105, 106]. The magnitude squared coherence spectrum (MSC), for example, provides a frequency-dependent measure of the linear relationship between two data sequences [78]. We assume the observed signals are irregularly sampled in an asynchronous manner. Spectral analysis methods from unevenly sampled signals often use some least squares Fourier transform estimator [105, 106], which suffer from poor resolution performance. The segmented iterative adaptive approach (SIAA-MSC) [107] can handle two signals, and offers better resolution if the number of samples is sufficiently large. A generalized version of coherent spectral estimation from multiple signals is proposed in [108], but it requires synchronous uniform sampling. In this section we first show that the coherent frequency estimation problem can be posed as the problem of estimating a jointly row sparse matrix \( X \). We then describe the SIAA-MSC algorithm. Finally, we shown how JLZA can use effectively for coherent spectrum analysis.

6.3.2 Joint Sparse Model

Consider \( J \) coherent signals \( \{ y_j(t) \}_{j=1}^J \) of the form

\[
y_j(t) = \sum_{\ell=1}^{k} a_{j,\ell} e^{i\Omega \ell t}.
\]  

(6.15)

Define the vector of sampling instants for \( y_j(t) \) as

\[
T_j = [ t_{j,1} \ t_{j,2} \ \cdots \ t_{j,m} ], \quad t_{j,i} < t_{j,i+1}.
\]  

(6.16)

We assume that \( T_j \)'s have the same length for all \( J \) to keep the description simple. However, the proposed algorithm in Section 6.3.3.2 can be extended trivially to a more general case. Let \( t_0 = \min_j t_{j,1} = \min_{j,\ell} t_{j,\ell} \) and \( \delta \) be the largest positive real number such that there are nonnegative integers \( d_{j,i} \) satisfying

\[
t_{j,i} = t_0 + d_{j,i}\delta, \quad j = 1, \ldots, J, \quad i = 1, \ldots, m,
\]  

(6.17)
Note that if \( j_* = \arg \min_j t_{j,1} \) then \( d_{j,1} = 0 \). Then by defining \( \omega_\ell = \delta \bar{\omega}_\ell \), and \( b_{j,\ell} = a_{j,\ell} e^{i\omega_\ell t_0} \), and using (6.15) we can write for any nonnegative integer \( q \) that
\[
y_j(t_0 + q\delta) = \sum_{\ell=1}^{k} b_{j,\ell} e^{i\omega_\ell q}, \quad j = 1, \ldots, J; \quad q = 0, 1, 2, \ldots
\]

Choose \( n \geq \max_j d_{j,m} \), set up a frequency grid
\[
\mathcal{G} = \{ \bar{\omega}_\ell = 2\pi \ell/n : \ell = 0, 1, \ldots, n - 1 \}, \tag{6.18}
\]
and define the \( n \)-point DFT matrix \( \Psi \in \mathbb{C}^{n \times n} \) similar to (6.3). If \( n \) is large enough, then the grid \( \mathcal{G} \) becomes sufficiently dense. Then for every \( j \in \{1, \ldots, k\} \), we have \( \omega_j \approx \bar{\omega}_\ell \) for some \( \ell \in \{0, 1, \ldots, n - 1\} \), and we can write
\[
z_j := [y_j(t_0) \ y_j(t_0 + \delta) \ \cdots \ y_j(t_0 + (n - 1)\delta)]' = \Psi x_j + \bar{e}_j, \tag{6.19}
\]
\( \bar{e}_j \) accounts for measurement noise and unmodelled residuals. Note that, the \( \ell \)-th component \( x_{j,\ell} \) is non-zero only if \( \bar{\omega}_\ell \approx \omega_j \) for some \( j \in \{1, \ldots, k\} \). Hence, \( x_j \) a sparse vector. One can estimate frequencies in the spectrum by locating the dominant peaks in \( x_j \). Furthermore, every \( y_j(t) \) have the same frequencies. Hence if the \( \ell \)-th component of \( x_1 \) is non-zero, then the \( \ell \)-th component of \( x_j \) is nonzero for all \( j \). Hence the matrix
\[
X = [ x_1 \ x_2 \ \cdots \ x_J ]
\]
is joint row-sparse, i.e. it has only a few non-zero rows.

By (6.17) the sampled measurements of the signals are given by
\[
y_j := [y_j(t_{j,1}) \ \cdots \ y_j(t_{j,m})]' = \Theta_j x_j + e_j, \tag{6.20}
\]
where \( \Theta_j \) is a matrix such that its \( i \)-th row is the same as the \( d_{j,i} + 1 \)-th row of the \( n \times n \) identity matrix, and \( e_j = \Theta_j \bar{e}_j \). Therefore the problem of coherent spectral analysis is the problem of estimating a joint sparse \( X \) from \( \{y_j\}_{j=1}^J \), given the “measurement matrices” \( \{\Theta_j\}_{j=1}^J \). In the following we denote \( \Phi_j = \Theta_j \Psi \) for short. The column of \( \Phi_j \) corresponding to frequency \( \bar{\omega} \) is denoted by \( \phi_j(\bar{\omega}) \). The component of \( x_j \) corresponding to frequency \( \bar{\omega} \) is denoted as \( x_j(\bar{\omega}) \).
6.3. Coherent Spectrum Analysis

6.3.3 Estimation of $X$

6.3.3.1 Segmented Iterative Adaptive Approach-Magnitude Squared Coherence (SIAA-MSC) Estimator

SIAA-MSC [107] is a nonparametric Capon-based MSC estimator that utilizes a segmented reformulation of the iterative adaptive approach (IAA) [109] for spectrum estimation. The outline of SIAA-MSC is given in Table 6.2. SIAA-MSC handles two signals at a time. Let the irregularly sampled data of two signals be $y_1 \in \mathbb{C}^m$ and $y_2 \in \mathbb{C}^m$ respectively. The MSC spectrum between the two signals is defined as [107]

$$\mathcal{D}_\omega = \frac{|\Upsilon_{1,2}(\bar{\omega})|^2}{\Upsilon_{1,1}(\bar{\omega})\Upsilon_{2,2}(\bar{\omega})}$$  \hspace{1cm} (6.21)

where $\Upsilon_{p,q}$ denotes the cross-spectral density between the two signals. For example, $\Upsilon_{1,2}$ and $\Upsilon_{1,1}$ are estimated from the given data $y_1 \cdot y_2$ and $y_1 \cdot y_1$ respectively. Let us consider the Capon based MSC estimates [110, 111]

$$\Upsilon_{p,q} = h_p^* R_{p,q} h_q$$  \hspace{1cm} (6.22)

where $R_{1,2}$ represents the cross-covariance matrix between given data vectors. $h_p$ is the filter formed such that

$$h_p = \arg\min_{h_p} h_p^* R_{p,p} h_p \text{ subject to } h_p^* \phi_p(\bar{\omega}) = 1$$  \hspace{1cm} (6.23)

where $R_{p,p}$ represents the covariance matrix of $y_p$. The solution of (6.23) is the Capon filter [112]

$$h_p = \frac{R_{p,p}^{-1} \phi_p(\bar{\omega})}{\phi_p^*(\bar{\omega}) R_{p,p}^{-1} \phi_p(\bar{\omega})}$$  \hspace{1cm} (6.24)

Note that the length of the Capon filter is same to the length of given data i.e., $m$. However, longer filter lengths generate spurious peaks in the spectrum [112]. On the other hand, a shorter filter increases the estimator bias. SIAA-MSC makes a tradeoff by dividing the data into $M$ segments of length $L$ each and then form the average spectral estimates as in (6.22). In Step 5 of the Table 6.2, the algorithm estimates $\mathbf{x}_1$ and $\mathbf{x}_2$ by using weighted least squares based nonparametric iterative adaptive approach (IAA). The detail of IAA can be found in [109]. However, this approach is slow in general and cannot perform as good as in presence of noise in the data. These motivate the alternative $\ell_{2,0}$ norm minimization approach described below.
Table 6.2: SIAA-MSC Algorithm [107]

**Initialization**

1. Set filter length $L$, then $M = m - L + 1$.
2. Set $\hat{\Upsilon}_{p,q}(\bar{\omega}_i) = 0; 0 \leq i \leq n - 1; 1 \leq p, q \leq 2$
3. Divide $y_p$ into $M$ vectors of length $L$ each, such that $j$-th vector 
    $y_p^j = [y_p(t_{p,j}) \cdots y_p(t_{p,j} + L - 1)]'$.
4. Divide $\phi_p(\bar{\omega}_i)$ into $M$ vectors of length $L$ each, such that $j$-th vector 
    $\phi_{p,j}(\bar{\omega}_i) = [e^{(i\bar{\omega}_it_{p,j})} \cdots e^{(i\bar{\omega}_it_{p,j} + L - 1)}]'$.
   for $j = 1, \cdots , M$
5. Estimate $x_1$ and $x_2$ from $y_1^j$ and $y_2^j$ by using IAA algorithm.
6. $\hat{R}_{p,q} = \sum_{i=1}^{n} x_p^s(\bar{\omega}_i)x_q(\bar{\omega}_i)\phi_{p,j}(\bar{\omega}_i)\phi_{q,j}^*(\bar{\omega}_i); 1 \leq p, q \leq 2$
7. $h_{p,j}(\bar{\omega}_i) = \frac{\phi_{p,j}^*(\bar{\omega}_i)(\hat{R}_{p,p})^{-1}\phi_{p,j}(\bar{\omega}_i)}{\phi_{p,j}^*(\bar{\omega}_i)(\hat{R}_{p,p})^{-1}\phi_{p,j}(\bar{\omega}_i)}; 0 \leq i \leq n - 1; 1 \leq p, q \leq 2$
8. $\hat{\Upsilon}_{p,q}(\bar{\omega}_i) = \hat{\Upsilon}_{p,q}(\bar{\omega}_i) + h_{p,j}^*(\bar{\omega}_i)\hat{R}_{p,q}h_{p,j}(\bar{\omega}_i); 0 \leq i \leq n - 1; 1 \leq p, q \leq 2$
9. $\hat{\Upsilon}_{p,q}(\bar{\omega}_i) = \hat{\Upsilon}_{p,q}(\bar{\omega}_i)M 0 \leq i \leq n - 1; 1 \leq p, q \leq 2$
10. Estimate MSC spectrum $\hat{D}_{\bar{\omega}} = \frac{|\hat{\Upsilon}_{1,2}(\bar{\omega})|^2}{\hat{\Upsilon}_{1,1}(\bar{\omega})\hat{\Upsilon}_{2,2}(\bar{\omega})}$

### 6.3.3.2 Estimation using JLZA

Compressive MUSIC (CS-MUSIC) [54] and Model based CS (Model-CS) [56] are two joint sparse recovery algorithms. However, they need the information about signal sparsity. In spectral analysis, the number of frequencies in the signal is not known in advance. Moreover, the algorithms is not applicable for coherent spectrum analysis when $T_j$‘s are different in (6.16).

The joint $\ell_{2,0}$ approximation (JLZA) algorithm (Section 3.2.2) is one of the most successful methods for joint sparse recovery, which has been extended in Section 3.4 to account for the case in (6.20). Here we approximate the zero norm by sum of Gaussian functions. The idea is to take a small real number $\sigma$, and solve

$$X_*(\sigma) := \arg \max_X L_\sigma(X),$$

$$L_\sigma(X) := \sum_{j=1}^{n} e^{-\frac{\|x_{(j,:)}\|^2}{2\sigma^2}} - \frac{1}{2} \sum_{j=1}^{J} \|y_j - \Phi_j x_j\|^2_2$$

(6.25)

Since $\sigma \to 0$ leads an NP hard problem. A sequential optimization method is used to
maximize $L_{\sigma_{\min}}$. It can be shown that as $\sigma \to \infty$,
\[
X_*(\infty) := \begin{bmatrix} \Phi_1^*(\Phi_1 \Phi_1^*)^{-1} y_1 & \cdots & \Phi_J^*(\Phi_J \Phi_J^*)^{-1} y_J \end{bmatrix}
= \Psi^*[\Theta_1^* y_1 \cdots \Theta_J^* y_J]/n
\]
is the maximizer of $L_{\sigma}$. Note that the optimization problem of (6.25) is same to the problem in (3.22) with $r = 1$. We use the algorithm in Table 3.3 to estimate the coherent spectrum. Note that by using the tricks in Section 6.2.3.1 one can decrease computation cost of the algorithm significantly.

6.4 Simulation Results

6.4.1 Spectrum Estimation of Single Signal

The artificial data set of this simulation is similar to the astronomical dataset described in [97] (Data set A), which consist in three sinusoids with periods 370, 300 and 100 days and amplitudes 3, 2.828 and 3, respectively. An initial data set was generated with 200 points sampled every 10 days. The initial dataset is resampled so that a gap of 100 days is introduced after every 365 days. To make the problem a bit trickier, we follow a strategy used in [49]. A fourth sinusoid was added with period 122.5 days and amplitude 3, such that the sidelobes caused by the annual gaps (for periods 370 and 122.5 days) superimpose at a period of 184 days $(1/122.5 - 1/365 = 1/370 + 1/365 = 1/184)$, generating a high false peak in the Fourier spectrum. Hence there are four frequencies in the actual signal i.e. $[1/370, 1/300, 1/122.5, 1/100] = [0.0027, 0.0033, 0.0082, 0.01]$ cycles/day. When generating signal we add random phase $[0, 2\pi)$ to every sinusoid. It is further assumed that the time samples $(y_i)$ are corrupted by uncorrelated zero mean noise sequence. Since the smallest sampling instants is $\delta = 10$ days, hence maximum frequency $f_{\max}$ should be set to less than $0.5 \times 1/10 = 5.10^{-2}$ c/d. We set $f_{\max} = 5^{-2}$ and $n = 2000$. We compare the performance of EISL0 with SparSpec $^2$ [49], Iterative Reweighted Least-Squares (IRLS) for complex valued variable [99] and $\ell_1$. For $\ell_1$ we use the following convex optimization problem,
\[
x_* = \min_x \|x\|_1 \quad \text{subject to} \quad \|\Phi^*(y - \Phi x)\|_\infty < \mu
\]
where $\mu = 3$. We vary the signal SNR between 10 dB to 30 dB and found that $\lambda = 50$ and $\sigma_{\min} = 0.01$ are good choice for EISL0.

$^2$http://www.ast.obs-mip.fr/Softwares
In Figure 6.1 we compare the frequency spectrum obtained using different methods with 120 time samples. Note that $\ell_1$ and IRLS can produce picks around the original locations. However, they generate many sidelobes. Only EISL0 has sharp estimation of actual frequencies. Figure 6.2(a) shows the probability of detecting frequencies as a function of $m$ when SNR = 15dB. Performance-wise, EISL0 exhibits clear superiority over the others. Where SparSpec needs 150 samples to estimate frequencies EISL0 takes only 120 samples.

Figure 6.2(b) demonstrate the performance of different algorithms in noisy environment. The interesting observation is that at higher SNR all algorithms perform similarly, however as SNR decreases most of them fails to locate frequencies. Note that above 20dB SNR, number of samples changes linearly for all algorithms. However, IRLS, $\ell_1$ and SparSpec cannot perform equally when SNR decreases below 20 dB. On the other hand, EISL0 perform equally until SNR = 8 dB. Finally Figure 6.3 shows the bias of different algorithms as functions of SNR. We take 150 samples and compute frequencies recovered by different
6.4. Simulation Results

Figure 6.2: (a) Probability of recovering actual frequencies as a function of number of time samples ($m$) for fixed SNR= 15 dB. (b) Minimum number of samples needed to recover actual frequencies as a function of SNR. Actual frequencies $[0.0027, 0.0033, 0.0082, 0.01]c/d$.

algorithms. Let the $i$-th frequency in the signal be $f_i$ and its estimation by an algorithm be $\hat{f}_i$, then bias is calculated as

$$f_b = \frac{1}{k} \sum_{i=1}^{k} |f_i - \hat{f}_i|.$$
6. Spectrum Analysis from Irregularly Sampled Data

Figure 6.3: Frequency bias as a function of SNR. Actual frequencies \([0.0027, 0.0033, 0.0082, 0.01]c/d, m = 150\).

The bias property of SparSpec and EISL0 are better compared to other two algorithms.

### 6.4.2 Spectrum Estimation of Coherent Signals

We compare the performance of JLZA-MMM for coherence spectrum estimation with a Welch-type LS-MSC estimator [106] and SIAA-MSC [107]. It should be noted that the comparison of JLZA with SIAA-MSC is not fully appropriate as SIAA-MSC estimates the coherence between the signals, whereas JLZA method estimates the joint frequencies in the signals. The settings recommended in [107] are used for MSC estimators. Two different values for the filter length \(L\) are used for SIAA-MSC. According to Table 6.2, SIAA uses IAA algorithm for spectrum estimation. We use 12 iterations of IAA for better results. The CVX interface [63] is used for \(\ell_{2,1}\). The measurement noise is taken as a zero-mean white complex Gaussian process with a variance \(\mu^2\). The value of \(\mu\) is different in different simulations. In all simulations we set \(|a_{j,\ell}| = 1\), for \(\forall j, \ell\), and for this setting the frequency-wise signal to noise ratio (SNR) in dB is given by \(\text{SNR} = -10 \log_{10}(\mu^2)\). The phases of \(a_{j,\ell}\)'s are iid random variables, uniform on \([0, 2\pi)\). The sampling instants \(t_{j,i}\) are drawn uniformly at random from the set \(\{0, 1, \ldots, R\}\), where \(R\) can vary. Similarly the frequencies \(\{\omega_j\}_{j=1}^k\) are drawn uniformly at random from the set \(\{2\pi q/1000, q = 1, \ldots, 999\}\). In all experiments we set \(n = 1024 = 2^{10}\) i.e., \(\Phi \in \mathbb{C}^{m \times 1024}\). This allows using radix-2 FFT/IFFT.
for computing $\zeta_j(X)$. Note that none of the elements of $\{\mathcal{U}_j\}_{j=1}^k$ lie on the grid $\mathcal{G}$. The experiments where $T_1 = T_2 = \cdots = T_J$, are denoted by ‘E’, e.g. SIAA-MSC:E. We use ‘D’ to indicate no two $T_j$’s are the same, e.g. SIAA-MSC:D. The signal SNR is varied between 0 dB to 5 dB. We find that $\sigma_{\min} = 0.01$ and $\lambda = 10$ are good choice for the setup.

Fig. 6.4, shows typical frequency spectrums for $k = 14$, $J = 2$, $m = 50$, SNR= 5 dB, and $R = 199$ obtained from a realization. Although SIAA-MSCs can resolve some of the frequencies, they also generate many spurious peaks. Both $\ell_{2,1}$ and JLZA-MMM can detect all frequencies. However, $\ell_{2,1}$ produces two spurious peaks at $0.9102 \times 2\pi$ and $0.9502 \times 2\pi$ rad/sec (in about 40% cases $\ell_{2,1}$ produces spurious peaks in this setting).

In Fig. 6.5(a), we examine the resolution performance of different algorithms. Here $k = 10$ and $\mathcal{U}_1 = 2\pi \times 0.05$ rad/sec. We vary $\mathcal{U}_2$ and plot the empirical probability of correctly detecting all ten frequencies as a function of $|\mathcal{U}_1 - \mathcal{U}_2|/(2\pi)$. The plot is based on 100 Monte-Carlo simulations, where in each simulation the 8 frequency locations $[\mathcal{U}_3, \cdots , \mathcal{U}_{10}]$, the noise realization and the phases of $a_{j,\ell}$’s are varied. For a signal with $k$ frequency components, we say the frequencies are detected correctly when an algorithm produces $k$ spikes, and the maximum absolute difference between any actual frequency and recovered frequency is $0.001/\pi$ rad/sec. We say a location in the spectrum is a spike when the location is a local maximum on the spectral plot and its absolute magnitude is above a threshold $\epsilon$. The value of $\epsilon = 0.01$ for $\ell_{2,1}$ and JLZA-MMM. However, SIAA generates many spurious peaks and hence we set $\epsilon = 0.1$ for SIAA. Fig. 6.5(b), shows the frequency resolution achievable by different algorithms as a function of $R$. In this setting, $\mathcal{U}_1 = 0.05 \times 2\pi$ and $\mathcal{U}_2$ is varied to control $|\mathcal{U}_1 - \mathcal{U}_2|$. The minimum $|\mathcal{U}_1 - \mathcal{U}_2|$ for which an algorithm detects all 10 frequencies with an empirical probability of 80% is recorded as a function of $R$. The resolution performance increases with increasing $J$. Surprisingly, the performance of SIAA degrades as $R$ increases, particularly when $R > 250$. The behavior of SIAA for large $R$ is illustrated in Fig. 6.6 In this Figure, both two signals (top and bottom) have same $k$ and $m$. However, in Figure 6.6(a) $R = 199$ whereas in Figure 6.6(b) $R = 399$. Note that when $R = 199$, SIAA recovers some of the frequencies, however at $R = 399$ SIAA fails completely.

Figures 6.7(a)-6.7(b) shows the empirical probability of detecting the frequencies as a function of $k$, the number of frequencies in the signals. We set SNR = 5 dB. The plots are based on 100 Monte-Carlo simulations with the frequency locations, noise-realization and phases of $a_{j,\ell}$ varying in each simulation. Also the criterion used for deciding whether the
Figure 6.4: A typical frequency spectrum. $m = 50$, SNR=5 dB and $k = 14$. 
6.4. Simulation Results

Figure 6.5: Resolution performance of different algorithms for $m = 50$, SNR = 5 dB, $k = 10$. (a) Probability of detecting the frequencies as a function of $|\bar{u}_1 - \bar{u}_2|/(2\pi)$ with $R=199$. (b) $|\bar{u}_1 - \bar{u}_2|/(2\pi)$ as a function of $R$.

frequencies have been detected correctly remains as described above. We have two different settings of $R$. In Figure 6.7(a) $R = 199$ and $m = 50$. As expected, the detection improves with increasing $J$. Moreover, when $T_j$’s in (6.16) are different, all algorithms give improved results. Note that $\ell_2,1$ and JLZA-MMM outperform other algorithms, with JLZA-MMM
Figure 6.6: A typical frequency spectrum of SIAA. (a): $R = 199$ (b): $R = 399$. $m = 50$, $k = 10$, SNR=5 dB.
being somewhat better than $\ell_{2,1}$. To illustrate how the proposed algorithms work with signals having more frequency components, we introduce Figure 6.7(b). We set $R = 1023$ and $m = 200$. Figure 6.7(b) demonstrates that the proposed algorithms can perform equally when the number of frequency components increases. As expected, performance of JLZA-MMM increases rapidly with increasing $J$. Note that SIAA algorithm is not shown in the graph as it can not perform equally when we increase $R$.

Figure 6.8 shows the maximum number of frequencies detected with an empirical probability of 80% by different algorithms as a function of SNR. In this setup, we increase $k$ for a fixed SNR and apply different algorithms. As before, JLZA-MMM outperforms all algorithms.

Finally, we compare computation time of different algorithms in Figure 6.9. Here, we consider the case of detecting signals with SNR=5 dB, $r = 2$, while the number of samples ($m$) and frequency components ($k$) vary between 30 to 60 and 4 to 14, respectively. As can be seen in Figure 6.9, the computation time of SIAA increases quickly with $m$, while $m$ has a little effect on the computation time of JLZA-MMM, which is at least 200 times faster than the other methods. Also $\ell_{2,1}$ is faster than SIAA. Finally as can be seen from Figure 6.9, $k$ has very little effect on the recovery time.

### 6.5 Conclusion

Spectral estimation using sparse representation is proposed in this chapter, which lead to a significant performance improvement at a much lower computational cost. At first we consider, spectrum analysis of a irregularly sampled single signal in Section 6.2. Subsequently, we extend the idea for coherent spectrum analysis in Section 6.3. In particular, the JLZA algorithm outperforms other state-of-the-art approaches.
Figure 6.7: Empirical probability of detecting frequencies as a function of $k$. SNR = 5 dB.
Figure 6.8: Maximum number of frequency components is detected by different algorithms as a function of SNR. $m = 50$, $R=199$. 

### 6.5. Conclusion
Figure 6.9: Frequency recovery time by different algorithms. (a) SIAA-MSC: E, L=0.6M, (b) SIAA-MSC: E, L=0.8M, (c) JLZA-MMM: E, r=2, (d) $\ell_2,1; r=2$. $R = 199$, SNR=5 dB.
Chapter 7

Target Detection using MIMO Radar System

7.1 Introduction

Multiple-input multiple-output (MIMO) radar is a multistatic architecture consisting of multiple transmitters and receivers [113, 114]. It is designed to exploit the spatial diversity of radar backscatter. Unlike the conventional beamforming radar systems, a MIMO radar uses multiple antennas to simultaneously transmit several independent waveforms and use multiple antennas to receive the reflected signals. There are two different kinds of MIMO radar arrangements. In one case the transmitting antenna elements are far enough from each other compared to their distance from the target [113]. Therefore, the target radar cross sections (RCS) are independent random variables for different transmitting paths. As a result an increased spatial diversity can be obtained. In the other case, the targets are assumed to be far field. The distances between transmitting antennas are small enough compared to the distance between the target and the radar array. Hence, the target RCS is identical for all transmitting paths. As a result, the phase differences caused by different transmitting antennas along with the phase differences caused by different receiving antennas can form a new virtual array steering vector [114]. Using the advantage, one can create a very long array steering vector with a small number of antennas. Thus, a better spatial resolution for clutter can be obtained. In this work we consider the second scenario.
A two dimensional bistatic MIMO radar scheme is proposed in [115]. Two-dimensional spatial spectrum estimation is carried out by using Capon’s method at the receiver. However, the Capon estimator needs two-dimensional (2-D) angle search. In [116], a PARAFAC-based technique is used for 2-D spatial spectrum estimation. The algorithm can separate closely spaced targets provided that the number of targets is known. A MUSIC derived algorithm called multi-invariance multiple signal classification (MI-MUSIC) has been proposed in [117]. However, MI-MUSIC needs many time samples to estimate targets. A low complexity ESPRIT-based algorithm is used in [118, 119]. However, ESPRIT cannot be used when the manifold matrix associated with the array is not a Vandermonde matrix.

In this chapter we present an alternative target localization algorithm for bistatic MIMO radar system. We represent the received signal of a bistatic MIMO radar into a multiple measurement vectors (MMV) problem and solve it efficiently by using the JLZA algorithm. JLZA with MMV allows us to enforce joint sparsity on the targets’ reflection coefficients which in turn results in a significant improvement of resolution performance and robustness against noise. Unlike most other state-of-art algorithms, the proposed method handle both orthogonal and non-orthogonal waveforms. We are able to achieve high-resolution performance without the need of a good initialization and with a limited number of time samples.

### 7.2 Problem Formulation

Consider a bistatic MIMO radar system consists $m_t$ transmit-antennas and $m_r$ receive-antennas (Figure 7.1). The system transmits $r$ separate pulses. Let the signal
transmitted by $m_t$ transmitting antennas at every pulse period be $S \in \mathbb{C}^{m_t \times u}$, where $u$ presents the length of coding sequence within one pulse period. The signals are reflected from $k$ targets, and received by $m_r$ receiving antennas. The angular position of $j$-th target is denoted by $\{\theta_j, \omega_j\}$, where $\theta_j$ and $\omega_j$ are the angles of the target with respect to the transmit and receive antenna arrays, respectively. The reflection coefficient of the $j$-th target during the $q$-th pulse be $\beta_{jq}$. Let

$$A_t(\theta) = \begin{bmatrix} a_t(\theta_1) & \cdots & a_t(\theta_k) \end{bmatrix} \in \mathbb{C}^{m_t \times k},$$  \hspace{1cm} (7.1)$$

$$A_r(\omega) = \begin{bmatrix} a_r(\omega_1) & \cdots & a_r(\omega_k) \end{bmatrix} \in \mathbb{C}^{m_r \times k},$$ \hspace{1cm} (7.2)

be the transmit and receive manifold matrices, respectively; while $a_t(\theta_j)$ and $a_r(\omega_j)$ are the steering vectors corresponding to the target-angle pair $(\theta_j, \omega_j)$ with respect to transmit and receive antenna arrays, respectively. The matrices $A_t(\theta)$ and $A_r(\omega)$ are assumed constant over the duration of the $r$ pulses. The target reflection coefficients (RCS) are assumed to remain constant during a pulse period and vary independently from one pulse to the next pulse. This target model is known as the Swerling Case II model [120].

The vector-valued signal received at the receiving array during $q$-th pulse period is given by [115, 116]

$$Y_q = A_r(\omega)B(q)A_t^t(\theta)S + E_q$$ \hspace{1cm} (7.3)

where $Y_q \in \mathbb{C}^{m_r \times u}$. Also $B = \text{diag}(\beta_1(q), \ldots, \beta_k(q))$ is a diagonal matrix whose diagonal components are the target reflection coefficients for the $q$-th pulse period. Also $E_q \in \mathbb{C}^{m_r \times u}$ represents the residual term, which includes the unmodeled noise.

To arrive at an overcomplete representation we extend the idea of Section 5.2.1. We divide the whole area of interest in some discrete set of angular-positions. Let the two-dimensional grid $G$ consisting of all potential angular position pairs $(\bar{\theta}_j, \bar{\omega}_\ell)$ be:

$$G = \{(\bar{\theta}_j, \bar{\omega}_\ell) : (j, \ell) \in \{1, \ldots, G\} \times \{1, \ldots, G\}\}.$$ 

Subsequently, we construct the transmit and receive array manifold matrices composed of the steering vectors corresponding to angles in $G$:

$$\Phi_t = \begin{bmatrix} a_t(\bar{\theta}_1) & \cdots & a_t(\bar{\theta}_G) \end{bmatrix},$$ 

$$\Phi_r = \begin{bmatrix} a_r(\bar{\omega}_1) & \cdots & a_r(\bar{\omega}_G) \end{bmatrix}.$$ \hspace{1cm} (7.4)
Let \( X_q \in \mathbb{C}^{G \times G} \) be the matrix of reflection coefficients of fictitious targets at \( G^2 \) possible grid points of interest during \( q \)-th pulse period. Note that \( X_q \) is not a diagonal matrix. Instead, the entry \( X_q[i,j] \) indicates the reflection coefficient of the target located at \( (\bar{\theta}_i, \bar{\omega}_j) \) during \( q \)-th pulse. Now the overcomplete version of (7.3) becomes

\[
Y_q = \Phi_r X_q \Phi_t S + \bar{E}_q, \quad q = 1, 2, \ldots, r.
\]  
(7.5)

Since the user designs the grid \( G \), the matrices \( \Phi_t, \Phi_r \) are known and do not depend on the actual target angles \( \{(\theta_j, \omega_j)\}_{j=1}^k \). Note that the equality \( \bar{\theta}_i = \theta_j \) and \( \bar{\omega}_\ell = \omega_j \) may not hold exactly for any \( j \) in practice. Nevertheless, by making \( G \) dense enough one can ensure \( \bar{\theta}_i \approx \theta_j \) and \( \bar{\omega}_\ell \approx \omega_j \) closely, and the remaining model error is absorbed by the residual term \( \bar{E}_q \). In practice, there exists few targets in the area of interest and hence \( k \ll G^2 \), which makes \( X_q \) is a sparse matrix, and \( X_q[i,j] \) is nonzero only if \( \bar{\theta}_i \approx \theta_j \) and \( \bar{\omega}_\ell \approx \omega_j \) for some \( j \in \{1, \ldots, k\} \). In effect, the representation (7.5) lets us pose the problem of estimating \( k \) and \( \{\theta, \omega\} \) as that of estimating a sparse matrix \( X \).

### 7.3 Joint Sparse Representation

Equation (7.5) can be re-written as

\[
\bar{Y}_q = \Phi_r X_q \Phi_t S + \bar{E}_q, \quad q = 1, 2, \ldots, r.
\]  
(7.6)

where \( R_s = (1/u)SS^* \), \( \bar{Y}_q = (1/u)Y_q S^* \) and \( \bar{E}_q = (1/u)E_q S^* \). Note that when the transmitted signals are orthonormal, then \( R_s \) will be an identity matrix. Now

\[
y_q := \text{vec}(\bar{Y}_q) = (\Psi_t \otimes \Phi_r) \text{vec}(X_q) + v_q = \Phi x_q + v_q
\]  
(7.7)

where \( \Psi_t = R'_t \Phi_t, \Phi = \Psi_t \otimes \Phi_r, x_q := \text{vec}(X_q) \in \mathbb{C}^{G^2} \) and \( v_q := \text{vec}(\bar{E}_q) \). One interesting point is that every element of the set \( \{X_q\}_{q=1}^r \) share the same sparsity pattern. This is because if there is a target at \( (\bar{\theta}_j, \bar{\omega}_\ell) \), and the target does not change its position over \( r \) pulse periods, then \( X_q[j, \ell] \neq 0, \forall q \). Otherwise, if no target presents at \( (\bar{\theta}_j, \bar{\omega}_\ell) \), then \( X_q[j, \ell] = 0, \forall q \).

Writing (7.7) for \( q = 1, \ldots, r \) and arranging in a compact form we get

\[
Y_A := [ y_1 \cdots y_r ] := \Phi X_A + V_A
\]  
(7.8)

where only \( k \) rows of the matrix \( X_A = [ x_1 \cdots x_r ] \) are non-zero.
7.4 Target Detection using Capon Based Estimator

A Capon based target estimator for bistatic MIMO radar system is proposed in [115]. Let us define the data covariance matrix as

\[ R_y = \frac{1}{r} Y_A Y_A^* \]

The Capon estimator of \((\bar{\theta}, \bar{\omega})\) can be written in the form

\[ D_{cap}(\bar{\theta}, \bar{\omega}) = \frac{1}{((R'_s a_t(\theta)) \otimes a_r(\bar{\omega}))^* R_y^{-1}((R'_s a_t(\theta)) \otimes a_r(\bar{\omega}))} \] (7.9)

The algorithm is summarized in Table 7.1. If there is a true target at \((\bar{\theta}, \bar{\omega})\) then the Capon estimator generates a peak at that location.

7.5 Target Detection using JLZA

The problem of recovering a joint sparse \(X_A\) from the measurement \(Y_A\) in (7.8) is a multiple measurement vectors (MMV) problem. Hence we adopt the idea of the JLZA approach proposed in Section 3.4.1. The idea is to take a small real number \(\sigma\), and solve

\[ X_\sigma(\sigma) := \arg \max_{X_A} \quad L_\sigma(X_A), \]

\[ L_\sigma(X_A) := \sum_{j=1}^{G^2} e^{-\frac{\|X_A[x_j]\|_2^2}{2\sigma^2}} - \frac{1}{2} \|Y_A - \Phi X_A\|_F^2. \]

The resulting algorithm is similar to the algorithm in Table 3.2, where further details can be found.

7.6 Simulation Results

For experimental setup, we consider a MIMO radar system with a nonuniform linear array with \(m_t = m_r = 9\) antennas. Let \(\nu = \frac{T}{2}\) and \(K = 100\). For transmitting
antennas, we draw locations \( \{d_j\}_{j=1}^{m_t} \) of the sensor arrays uniformly at random from a set \( \mathbb{L} = \{\ell \nu : \ell = 1, \ldots, K\} \). Where \( d_1 = 0 \) and \( d_1 < d_2 < \cdots < d_{m_t} \). It follows that
\[
a_t(\theta) = [e^{i(2\pi/\lambda)d_1 \sin \theta} \cdots e^{i(2\pi/\lambda)d_{m_t} \sin \theta}],
\]
where \( \lambda \) denotes the carrier wavelength. The locations of receiving antennas are selected in similar way. We compare the performance of JLZA with 2D-Capon [115] and PARAFAC-based technique [116]. The transmitted signal \( S \) is generated from a zero mean Gaussian distribution where \( u \) is fixed to 32. We assume that the noise \( E_q \in \mathbb{C}^{m_r \times u} \) are mutually independent, complex Gaussian distributed with zero-mean, and covariance matrix \( \mu^2 I \). With this noise covariance we define the SNR of \( p \)-th target as
\[
\text{SNR} = 10 \log_{10} \left( \frac{\delta_p^2}{\mu^2} \right),
\]
where \( \delta_p^2 \) is the power of \( p \)-th target and \( \mu = 0.5 \). We discretize the angle space by increments of 1° from \(-40^\circ\) to \(40^\circ\), i.e., \( G = 81 \). For an arrangement with \( k \) targets, we say the targets are detected correctly when an algorithm produces \( k \) spikes, and the maximum absolute difference between any actual angle (transmit/receive) of a target and recovered angle (transmit/receive) is \( 2^\circ \). We say a location in the spectrum is a spike when the location is a local maximum on the spectral plot and its absolute magnitude is above a threshold \( \epsilon \). The value of \( \epsilon = 0.2 \) for JLZA. For JLZA we fixed \( \sigma_{\min} = 0.001 \) and \( \lambda = 10 \). One can also perform a likelihood ratio test to detect the presence of the target.

Figure 7.2, shows the target recovery performance of different algorithms for \( k = 8 \) and \( r = 10 \). Note that with limited pulses \( (r) \), only JLZA can distinguish all 8 targets. 2D-Capon fails to locate targets properly. Three closely spaced targets located at \{8, 10\}, \{16, 14\} and \{18, 16\} degrees are recognized as a single target. PARAFAC-based method also fails to distinguish the three targets. Note that, PARAFAC is able to locate the target at \{18, 16\} degree. However, it fails to location other two targets. Also PARAFAC-based method locates two targets around \{-16, 10\} degree. Figure 7.3 (a) shows the empirical probability of detecting targets as a function of \( r \), the number of pulses. We consider the target setting similar to Figure 7.2. However, the SNR of all 8 targets are 2 dB. The plots are based on 100 Monte-Carlo simulations with the noise-realization and phases of \( \beta_p(q) \) are varying in each simulation. As expected, JLZA is able to locate targets using small number of pulses. In this particular setup, only 15 pulses are sufficient for JLZA to locate targets properly. 2D-Capon and PARAFAC-based methods require many pulses compared to JLZA. Also PARAFAC-based method performs somewhat better than 2D-Capon.

Figure 7.3 (b) demonstrate the performance of different algorithms in noisy envi-
Figure 7.2: Estimating 8 targets. Targets are located at \( \{14, -22\}, \{8, 10\}, \{16, 14\}, \{18, 16\}, \{28, -18\}, \{-10, -16\}, \{-30, 26\}, \{-16, 10\} \) degrees. \( r = 10, \) SNR=\([1, -5, 5, -1, 3, 5, 10, 5]\) dB. (a) True locations (b) JLZA (c) 2D-Capon (d) PARAFAC.
Figure 7.3: Estimating 8 targets. Targets are located at \( \{14, -22\}, \{8, 10\}, \{16, 14\}, \{18, 16\}, \{28, -18\}, \{-10, -16\}, \{-30, 26\}, \{-16, 10\} \) degrees. (a) Empirical probability of detecting targets as a function of \( r \), SNR of all targets 2 dB (b) Empirical probability of detecting targets as a function of SNR, \( r = 50 \).
7.6. Simulation Results

Figure 7.4: Angle bias as a function of SNR. Target arrangements are similar to Figure 7.2, $r = 50$.

As before we consider similar target setting as shown in Figure 7.2. The number of pulses $r = 50$. The SNR of all targets are same and are decreasing simultaneously. The empirical probability of detecting all targets is plotted as a function of SNR. As expected JLZA is robust against noise up to $-2$ dB. The interesting observation is that the performance of other two algorithms decrease rapidly with decreasing SNR. At $-2$ dB, their target detection probability drops to 30%. Finally, Figure 7.4 shows the bias by different algorithms as functions of SNR. Let the actual location of $p$-th target be $\{\theta_p, \omega_p\}$ and its estimation is $\{\bar{\theta}_p, \bar{\omega}_p\}$. Then the angle bias for a particular SNR is defined as

$$\tau = \frac{1}{2k} \sum_{p=1}^{k} \{ |\theta_p - \bar{\theta}_p| + |\omega_p - \bar{\omega}_p| \}. $$
Chapter 8

Range-Doppler Imaging

8.1 Introduction

Target localization and motion estimation using range-Doppler imaging [121, 50] has been an active research area in radar signal processing. Here one estimates the ranges and the Doppler frequency shifts of the targets. To achieve high range resolution in radar system, each transmitted pulse must have a small time-width. This requires very high instantaneous power. Pulse compression technique is widely used to overcome this problem (see [121, 122, 123] and references therein), where a relatively long train of modulated sub-pulses is transmitted towards the area of interest. For the same total transmitted energy, the sub-pulse train has a smaller peak power than a single pulse. We consider the radar system proposed in [50, 122], where range-Doppler imaging is performed by transmitting a train of probing pulses. Both negligible and non-negligible Doppler shifts are considered in [122]. Although the technique therein performs well for negligible Doppler shifts, it can not perform equally for non-negligible Doppler shifts. Two instrumental variables (IV) approaches have been proposed in [50]. The first one is called data dependent method where an IV filter (or sometimes called the mismatched filter, see [124, 125] ) is used. As the data dependent method can not achieve super Doppler resolution, an iterative adaptive approach (IAA) is used to overcome the limitation. IAA exhibits high Doppler resolution, but often it is not robust to noise.
8.1.1 Contributions and Motivation

Under certain reasonable assumptions the received data sequence of a radar system can represent as a sparse linear combination of certain vectors. These vectors form a so-called over-complete dictionary. In this framework the problem of estimating the complex amplitudes of the reflected signals becomes a sparse signal recovery problem. Enforcing the sparsity of the representation is motivated by the desire to obtain a super-resolution of the spatial spectrum. Recently, several source localization, and target detection problems have been cast as sparse signal recovery problems; see for example [126, 127]. A sparse signal reconstruction perspective for imaging of moving target using multi-static SAR has been addressed in [126]. The nonlinear, coupled problem of joint velocity and reflectivity estimation is efficiently linearized by defining an overcomplete velocity dictionary. The resulting optimization problem is solved through a convex relaxation formulation. In [127], a sparse representation algorithm for the structured overcomplete dictionaries is presented. This algorithm exploits the underlying structure of a dictionary, and relates the problem to a graph-search problem. A coordinate descent method is proposed to optimize an $\ell_p$ ($p < 1$) penalty for jointly optimizing the parameterized dictionary and obtaining a sparse solution using that dictionary.

For range-Doppler imaging the sparse signal is a two-dimensional signal, which is the image itself. This makes sparse recovery computationally challenging. To this end we have tested several $\ell_0$ and $\ell_1$ based methods. However, the conventional $\ell_1$ and $\ell_0$ optimization algorithms are computationally expensive. Hence, we use the 2-D sparse recovery algorithm SL0 (smoothed $\ell_0$) [128] for range-Doppler imaging. In our experiments, 2D-SL0 offers better resolution in presence of noise. In addition, we evaluate performance of the sparse recovery approaches in presence of land clutter. We found that without any modification, the proposed approach can locate moving targets in clutter environment having moderate signal to clutter ratio (SCR).

8.2 Problem Formulation

In this work we use data model similar to [50, 122]. Consider a radar system transmitting a probing pulse train shown in Figure 8.1. Each pulse train has $Q$ probing pulses and each probing pulse consists of $N$ sub-pulses (in some literature of radar tech-
nology, the sub-pulse is termed as chip). The sub-pulses may be rectangular or they may have different shape. We assume that all probing pulses in a pulse train are identical. The carrier frequency, pulse repetition frequency and sub-pulse bandwidth are $C$, $H$ and $D$ respectively, where $H \ll D \ll C$. The duration of each sub-pulse is $1/D$. Then the range resolution and Doppler resolution are given by $c/2D$ and $H/Q$, respectively. Here $c$ is the speed of propagation [50]. We assume there are no range ambiguities and Doppler frequency ambiguities. This is a common assumption [122]. The whole range bin of interest is divided into $R$ uniform sub-range bins as $\{1, \cdots, R\}$. Similar to [50], we assume that the Doppler frequency shift interval of interest is $[-H/2, H/2]$. We sub-divide this interval into $L$ smaller non-overlapping Doppler bins, so that the center frequency of the $\ell$-th Doppler bin is given by

$$G_\ell = -H/2 + H(\ell - 1/2)/L, \quad \ell = 1, 2, \ldots, L.$$ 

Consequently, the Doppler phase shift over one sub-pulse for the $\ell$-th Doppler frequency bin be

$$\bar{\omega}_\ell = 2\pi G_\ell / D, \quad \ell = 1, 2, \ldots, L.$$ 

Also the Doppler phase shift interval of interest be $[-\bar{\omega}_d, \bar{\omega}_d]$, where $\bar{\omega}_d = \pi H / D$. In other words, we fix a grid

$$\mathbb{G} = \{\bar{\omega}_\ell : \ell = 1, 2, \ldots, L\}. \quad (8.1)$$

The transmitter transmits $N$ sub-pulses modulated by a modulating code sequence $\{s_j\}_{j=1}^N$. The transmitted signals are reflected from targets and received at the receiver. The received signal is demodulated by a sub-pulse matched filter and converted from analog to digital. Since the range-bins are designed by the user, the return-trip delay associated with each range bin is known. The data vector $y_r(q)$ associated with the $r$-th range bin and $q$-th probing sub-pulse is constructed by windowing the received data for the $q$-th probing sub-pulse.
pulse with a rectangular window of length $N$. The center of the window is time-aligned with the return-trip delay for the $r$-th range bin. If all the targets are stationary then it can be shown that, (see [122])

$$y_r(q) = \sum_{j=-N+1}^{N-1} \beta_{r+j} \mathbf{J}_{-j} \mathbf{s} + e_r(q),$$

(8.2)

where $\beta_j$ is a complex-valued scalar proportional to the radar cross section of the target at $j$-th range bin, $e_r(q)$ is the additive noise, and

$$\mathbf{s} = [s_1 \cdots s_N]' .$$

In addition, the shift matrices $\mathbf{J}_j \in \mathbb{R}^{N \times N}$ are defined as follows:

$$\begin{bmatrix}
0 & 1 & 0 \\
0 & 1 & \\
\vdots & & 1 \\
0 & 0 & 1 \\
& & 0
\end{bmatrix},$$

(8.3)

and

$$\mathbf{J}_j = \begin{cases}
\mathbf{J}_1^j & j > 0, \\
\mathbf{I}_N & j = 0, \\
(\mathbf{J}_1^{-j})' & j < 0.
\end{cases}$$

(8.4)

Clearly, $y_r(q) \in \mathbb{C}^N$. As expected, the noise-free part of $y_r(q)$ is independent of $q$ when the positions of the targets are time-invariant. But when the targets are moving then we need to account the Doppler frequency shifts, and hence we have $q$-dependence in the noise-free part of $y_r(q)$. Assuming there is only one target in each range bin$^3$:

$$y_r(q) = \sum_{j=-N+1}^{N-1} \beta_{r+j} s_j e^{i\tau(q-1)\omega_{r+j}} + e_r(q),$$

(8.5)

where $\tau$ is the time gap between two consecutive probing pulses, $\omega_j$ denotes the Doppler frequency shift due to the motion of the target in the $j$-th range bin, and we write

$$s_j = \mathbf{J}_{-j} \mathbf{s},$$

$^3$We make this assumption for exposition purpose. We do not need this assumption, see discussion after (8.7).
Two mild assumptions are made while deriving (8.5). First, we are assuming that the time-gap $\tau$ between two consecutive pulses is large enough so that for all $q$, any return due to $q$-th pulse do not affect the return due to $q + 1$-th pulse. We are also assuming that the velocities of the targets are small enough so that their displacement during a pulse period can be neglected. This is equivalent to
\[
\frac{N((2v_{\text{max}}/c)C)}{D} \ll 1,
\]
where $v_{\text{max}}$ is the maximum possible speed of target. This assumption is well justified, and has been shown not to affect the estimation performance [50]. It is also worthwhile to note that in reality we don’t have a target in every range bin. In other words the vector
\[
[ \beta_1 \beta_2 \cdots \beta_R ]
\]
is in practice a sparse vector and most of its components are zero. This fundamental observation motivates the sparse signal model used below. Let us consider the Doppler grid $G$ in (8.1) and assume that for every $j$, there is a $\ell \in \{1,2,\ldots,L\}$ such that
\[
\omega_j = \bar{\omega}_\ell.
\]
In general, the above equality will not hold exactly. But if $G$ is sufficiently dense then this is a good approximation, and the small model errors are absorbed in the residual term $e_r(q)$ in the model. Now for each $j \in \{1,2,\ldots,R\}$, define \{\(x_{j,\ell}\)\}_{\ell=1}^{L} such that
\[
x_{j,\ell} = \begin{cases} 
\beta_j, & \text{if } \omega_j = \bar{\omega}_\ell \\
0, & \text{otherwise.} 
\end{cases}
\]
(8.6)
In this way we can visualize the range and Doppler shifts of the targets in form of an image, with each $x_{j,\ell}$ representing a pixel. Clearly this is sparse image, as all but just a few $x_{j,\ell}$ are zero. A non-zero $x_{j,\ell}$ represents a target at $j$-th range bin and $\ell$-th Doppler bin. In the range-Doppler image such a non-zero pixel should appear as a sharp spike standing on a zero background. Using (8.6) in (8.5) we can write
\[
y_r(q) = \sum_{j=-N+1}^{N-1} s_j \sum_{\ell=1}^{L} x_{r+j,\ell} e^{i\bar{\omega}_\ell(q-1)} + e_r(q).
\]
(8.7)
Recall the assumption that there is only one target in each range bin. This can be relaxed as (8.7) allows multiple targets in a range bin. However the targets in the same range bin
can be distinguished from each other only if they all have distinct Doppler frequency shifts. In the following we recast (8.7) into a more compact form which allows us to benefit from its special structure in designing $\ell_0$ approximation approaches. Defining

$$x_j = [x_{j,1} \ x_{j,2} \ \cdots \ x_{j,L}],$$

$$\theta_q = [e^{i(q-1)\tau \bar{\omega}_1} \ e^{i(q-1)\tau \bar{\omega}_2} \ \cdots \ e^{i(q-1)\tau \bar{\omega}_L}],$$

rewrite (8.7) as

$$y_r(q) = \sum_{j=-N+1}^{N-1} s_j x_{r+j} + e_r(q) = S_r X \theta_q + e_r(q). \quad (8.8)$$

In the last equality we use

$$S_r = \begin{cases} 
[ s_{1-r} \ s_{2-r} \ \cdots \ s_{N-1} \ 0_{1 \times (R-N+1-r)} ], & \text{if } r \leq N-1, \\
[ 0_{r-N} \ s_{-N+1} \ s_{-N+2} \ \cdots \ s_{N-1} \ 0_{R-N-r+1} ], & \text{if } N \leq r \leq R-(N-1) \\
[ 0_{r-N} \ s_{-N+1} \ s_{-N+2} \ \cdots \ s_{R-r} ], & \text{if } r > R-(N-1) 
\end{cases} \quad (8.9)$$

and the matrix of range-Doppler image

$$X := [x'_1 \ x'_2 \ \cdots \ x'_R].$$

Note that $[X]_{j,\ell} = [x_j]_{\ell} = x_{j,\ell}$. Now let

$$\Theta = [\theta_1 \ \theta_2 \ \cdots \ \theta_Q].$$

Then it follows that

$$[y_r(1) \ y_r(2) \ \cdots \ y_r(Q)] = S_r X \Theta + E_r, \quad (8.10)$$

where $E_r = [e_r(1) \ \cdots \ e_r(Q)]$. Now arranging the data collected from all range bins of interest i.e., by using (8.10) for $r = 1, \ldots, R$, we have

$$Y = \Phi X \Theta + E \quad (8.11)$$

where
8. Range-Doppler Imaging

\[ Y = \begin{bmatrix} y_1(1) & y_1(2) & \cdots & y_1(Q) \\ y_2(1) & y_2(2) & \cdots & y_2(Q) \\ \vdots & \vdots & \ddots & \vdots \\ y_R(1) & y_R(2) & \cdots & y_R(Q) \end{bmatrix}, \quad (8.12) \]

\[ S = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_R \end{bmatrix}, \quad (8.13) \]

and \( E = [ E_1' \ E_2' \ \cdots \ E_R' ]' \). Clearly, \( S_r \in \mathbb{C}^{N \times R} \) and hence \( S \in \mathbb{C}^{NR \times R} \). Also \( \Theta \in \mathbb{C}^{L \times Q} \).

8.3 Reconstruction Approaches

In this section we explore various ways to reconstruct \( X \) from \( Y \). In particular we want to enforce the sparsity constraint on \( X \). Ideally this can be done by minimizing \( ||\text{vec}(X)||_0 \) subject to the constraint \( Y = SX\Theta \). As this leads to an NP-hard problem, a common relaxation is to minimize \( ||\text{vec}(X)||_1 \) and solve a convex problem instead.

8.3.1 \( \ell_1 \) Optimization Approach

Using the \( \ell_1 \) paradigm we solve the range-Doppler imaging problem as

\[ \hat{X}_1 = \arg\min_X ||\text{vec}(X)||_1 \ \text{subject to} \ ||Y - SX\Theta||_F^2 \leq \vartheta. \quad (8.14) \]

The constant \( \vartheta \) is chosen by the user. It's value depends on the noise level. In our experiments we set \( \vartheta = 0.02 \). Now consider the economy size QR factorization of \( S \):

\[ S = QR^*, \quad (8.15) \]

where \( R \) is a nonsingular lower triangular matrix, and the columns of \( Q \) are mutually orthogonal. Now

\[
||Y - SX\Theta||_F^2 = \text{Tr}\{(Y - QR^*X\Theta)^*(Y - QR^*X\Theta)\}
= ||Y||_F^2 - ||Y_1||_F^2 + ||Y_1 - R^*X\Theta||_F^2,
\]
8.3. Reconstruction Approaches

where

\[ Y_1 = Q^* Y. \]

Note that the feasible set in (8.14) is non-empty only if

\[ \vartheta \geq ||Y||^2_F - ||Y_1||^2_F. \]  \hspace{1cm} (8.16)

When equality holds in (8.16) then (8.14) reduces to an optimization problem subject to an equality constraint:

\[ \hat{X}_1 = \arg \min_X ||\text{vec}(X)||_1 \text{ subject to } Y_1 = R^* X \Theta, \]  \hspace{1cm} (8.17)

which is the standard form used in reconstructing the sparse signal \( X \) in \( \ell_1 \) paradigm. Regardless of whether a strict inequality holds in (8.16) or not, this approach leads to a convex optimization problem.

8.3.2 \( \ell_0 \) Minimization Approach

Polynomial time algorithms exit for the \( \ell_1 \) minimization discussed above. This is its main advantage. However, in many cases the computation time of \( \ell_1 \) based methods turns out to be significantly more than the state-of-the-art \( \ell_0 \) optimization approaches [34, 129, 130, 131]. Hence we turn to the \( \ell_0 \) approximation approach. The role of the word approximation is to stress that actual \( \ell_0 \) minimization problem

\[ \hat{X}_0 = \arg \min_X ||\text{vec}(X)||_0 \text{ subject to } ||Y - SX \Theta||^2_F \leq \epsilon. \]  \hspace{1cm} (8.18)

Solving (8.18) is NP-hard. Instead, one minimizes an approximation of the zero norm. Let us define

\[ F_\sigma(X) = \sum_{k=1}^{R} \sum_{\ell=1}^{L} \exp \left( \frac{-|X_{k,\ell}|^2}{2\sigma^2} \right). \]  \hspace{1cm} (8.19)

and approximately reformulate (8.18) as

\[ \hat{X}_* = \arg \max_X F_\sigma(\text{vec}(X)) \text{ subject to } ||Y_1 - R^* X \Theta||^2_F \leq \vartheta_1, \]  \hspace{1cm} (8.20)

with a sufficiently small \( \sigma \). Note that in (8.20) we use (8.16) and set

\[ \vartheta_1 = \vartheta - ||Y||^2_F + ||Y_1||^2_F, \]  \hspace{1cm} (8.21)
Table 8.1: Two Dimensional Robust SL0 (2D-SL0) Algorithm [128]

- Initialization:
  1. Initialize $X_0$ to the minimum 2-norm solution of $Y_1 = R^* X \Theta$,
     i.e. $X_0 = (R^*)^\dagger Y_1 \Theta^\dagger$.
  2. Set $\sigma$ and set $i = 0$, $\mu = 1$, decreasing factor $\rho = 0.5$.

repeat
  3. for $k = 1, \cdots, K$
     a) $\bar{X} = X_i$.
     b) Compute $\nabla F_\sigma(\bar{X}) = [\delta_{i,j}]$, where $\delta_{i,j} := \bar{X}[i,j] e^{-|\bar{X}[i,j]|^2/2\sigma^2}$.
     c) $\bar{X} = \bar{X} - \mu \nabla F_\sigma(\bar{X})$.
     d) If $\|R^* \bar{X} \Theta - Y_1\|_F > \vartheta_1$
        $\bar{X} = \bar{X} - (R^*)^\dagger (R^* \bar{X} \Theta - Y_1) \Theta^\dagger$.
     e) $X_{i+1} = \bar{X}, i = i + 1$.
  4. $\sigma = \rho \sigma$.
until $\sigma < \sigma_{\text{min}}$

which should be a non-negative number in order to ensure that the feasible set is non-empty.

A fast reliable algorithm using a projective steepest ascent approach called two dimensional smoothed $\ell_0$ (2D-SL0) to solve (8.20) is developed in [128] and [99], and is given in Table 8.1.

In our experiments, we set $K = 10$ and initial value of $\sigma = 2 \max(|\text{vec}(X_0)|)$ [99]. There are several tuning parameters in 2D-SL0. However, in [34], it is shown that SL0 is not very sensitive to the parameters. The final value of $\sigma$ i.e. $\sigma_{\text{min}}$ and $\vartheta_1$ should be adjusted depending on the noise level. Algorithms for adjusting $\sigma_{\text{min}}$ have been presented in Section 2.2.1.2. Also, see [99] for a discussion on methods for choosing $\vartheta_1$. In our experiments these values are set to $\sigma_{\text{min}} = 0.05$ and $\vartheta_1 = 0.002$.

8.4 Simulation Results

In this section we present the numerical simulation results. We consider four techniques, namely, the standard matched filtering (MF) approach [122, 123], iterative adaptive approach (IAA) [50], $\ell_1$ based approach, and 2D-SL0. IAA is an iterative algorithm based on a weighted least squares criterion [132]. IAA usually converges in 10 to 15 iterations,
8.4. Simulation Results

Table 8.2: Ground Truth 1

<table>
<thead>
<tr>
<th>Target</th>
<th>Range bin</th>
<th>Velocity (m/s)</th>
<th>Doppler bin (Hz)</th>
<th>SNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>15</td>
<td>−37.5</td>
<td>−250</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>22.5</td>
<td>150</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>−5</td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>37.5</td>
<td>250</td>
<td>1</td>
</tr>
</tbody>
</table>

and thus, in our simulations we consider 15 iterations. Also a good initialization is required to obtain a good output from IAA. As recommended in [50], we use the output of a data independent instrumental variable (IV) filter to initialize the IAA. The detailed implementation of this data-dependent IV filter can be found in [50]. For $\ell_1$ we adopt the SPGL1 solver designed for large-scale sparse recovery problems [133]. We allow a maximum of 500 iterations in SPGL1. We assume that the noise vectors $\{e(q)\}_{q=1}^{Q}$ are mutually independent, complex Gaussian distributed with zero-mean, and covariance matrix $\delta^2 I$. With this noise covariance we define the SNR of a target located at the $(j, \ell)$-th range-Doppler bin as

$$\text{SNR} = 10 \log_{10}(|X_{j,\ell}|^2 / \delta^2)$$  \hspace{1cm} (8.22)

where $\delta = 0.5$. The code sequence $\{s_j\}_{j=1}^{N}$ of a pulse probe is taken as a 13-chip Barker sequence, hence $N = 13$. The same sequence is used for all $Q$ probing pulses in a pulse train. In all experiments we set system parameters $C = 1$ GHz, $D = 10$ MHz, $H = 2$ KHz. The range of interest is divided into $R = 40$ range bins. Doppler frequency shift interval of interest is $[-H/2, H/2]$. The Doppler resolution is 25 Hz i.e., $L = 80$, and hence

$$\mathcal{G}_{\ell} - \mathcal{G}_{\ell-1} = 25, \ \ell = 2, 3, \ldots, L.$$  

8.4.1 Effect of Noise

In this section, we test the ability of different algorithms to locate targets with low SNR in presence of targets with high SNR. We consider five targets. The targets arrangement is given in Table 8.2. The targets 1, 2 and 4 have high SNRs, while the targets 3 and 5 have low SNRs. The first target is moving at a speed $v = −37.5$ m/s which corresponds to a Doppler frequency $\mathcal{G}_1 = \frac{2\pi v C}{c} = −250$ Hz. Similarly the second and fifth targets have moving speeds 22.5 m/s and 37.5 m/s respectively. All other targets are still. The results for $Q = 16$ is shown in Figure 8.2. Note that MF fails to resolve
the targets. IAA algorithm can locate the targets with large SNRs, but fails to detect the targets with smaller SNR. In addition, IAA produces large ripples over the whole area of interest. Nevertheless, both 2D-SL0 and $\ell_1$ give sharp estimation of all five targets.

Figure 8.3 addressed an interesting issue when the targets do not lie on the discretized grid. One can obtain a close estimate of the target locations by discretizing the grid fine enough and then the gridding error will be small. However, when we make the grid very fine, the columns of the resulting system matrix will be highly correlated and then the proposed algorithms cannot perform equally [134]. We consider two targets. The targets are located at range bin 15 with SNRs 5 dB and 1 dB respectively. The Doppler shift of the targets are $-220$ Hz and $110$ Hz. We fixed $R = 40$. At first the Doppler resolution is 25 Hz i.e., $L = 80$. Hence, the Doppler shift of the both targets do not lie on the grid. Figure 8.3(a) shows the recovery result. Note that both 2D-SL0 and $\ell_1$ detect targets at $-200$ Hz and $100$ Hz. We then set Doppler resolution to 12.5 Hz. Unfortunately, none of the targets can fall on the discretized grid. The result in Figure 8.3(b) shows that the algorithms detect...
targets at $-212.5$ Hz and $112.5$ Hz. Hence produce better approximation. Finally, we set Doppler resolution to $6.25$ Hz in Figure 8.3(c). The result is not improved compared to Figure 8.3(b). We found that further enhancement of the Doppler grid cannot improve the results of both 2D-SL0 and $\ell_1$.

In the next experiment, we consider a target-arrangement similar to Table 8.2. However, the SNRs of all targets are equal. At every step, we sequentially decrease the SNRs of all targets equally, and measure the peak to ripple ratio (PRR) $P_1/P_2$ (in some literature, PRR is often termed as peak side-lobe ratio or PSLR). We use the following procedure to compute $P_1$ and $P_2$. Let $\hat{X}$ be estimate of $X$. Suppose there are $k$ targets located at the range-Doppler bins $\{(r_i, \ell_i), i = 1, 2, \ldots, k\}$. Then

$$P_1 = \sum_{i=1}^{k} \sum_{j_1=-1}^{1} \sum_{j_2=-1}^{1} |\hat{X}_{r_i+j_1, \ell_i+j_2}|^2, \quad P_2 = ||\hat{X}||^2_F - P_1.$$

The PRR plots are shown in Figure 8.4(a), where we can notice that both $\ell_1$ and 2D-SL0 gives about 10 times larger PRR compared to IAA and IV filter approach. Figure 8.4(b) compares the computation time for different algorithms for the setting in Table 8.2. Note that the computation time for IAA is largest among the algorithms considered here, and increases quickly with increasing the number of probing pulses in a pulse train. 2D-SL0 requires significant lower computation time while MF requires the least computation time.

Recall (9.1), $\beta_j$ is a complex-valued scalar proportional to the radar cross section of the target at $j$-th range bin. Since this parameter is related to the radar cross section, it is also of interest to investigate how accurate this is determined. We consider the target setup similar to Figure 8.4(a). To determine recovery accuracy we consider the NRMSE measure (see Section 4.5.1) of the recovered $\beta$. The result in Figure 8.5 demonstrates that both algorithms can recover $\beta$ with high accuracy when target SNR is high. However, performance degrades when target SNR is low. Hence, at lower SNR, one can utilize the nonlinear least-squares (NLS) approach [135] for better approximation of $\beta$.

Figure 8.6 shows the receiver operating characteristic (ROC) of different algorithms. We consider a target-arrangement similar to Table 8.2. However, the SNRs of all targets are equal. Let $\hat{X}$ be the estimate of $X$ for a particular SNR. We then select a threshold $\tau_i$. Any local maximum of $\hat{X}$ with an absolute value greater than $\tau_i$ is considered as a target. We vary $\tau_i$ within an interval $[\tau_L, \tau_H]$ and for each value of $\tau_i$ we record whether all actual targets are detected and whether any false targets are detected. After repeating
Figure 8.3: Power distribution over Doppler at range bin 15. There are two targets with Doppler shifts $-210$ Hz and 110 Hz. (a) Doppler resolution is 25 Hz. The Doppler shifts of two targets are detected at $-200$ Hz and 100 Hz. (b) Doppler resolution is 12.5 Hz. The Doppler shifts of two targets are detected at $-212.5$ Hz and 112.5 Hz. (c) Doppler resolution is 6.25 Hz. The Doppler shifts of two targets are detected at $-212.5$ Hz and 112.5 Hz.
the experiment 100 times, this procedure yields the empirical probability $P_a$ of detecting all the actual targets, and the empirical probability $P_f$ of detecting any false target for different values of $\tau_i$. In Figure 8.6, we plot $P_a$ as a function of $P_f$ for different algorithms and different SNR settings. The sensitivity of the algorithm is defined as the smallest SNR
for which \( P_f \leq 0.1 \) and \( P_a \geq 0.8 \). For 2D-SL0, \( \ell_1 \) and IAA, this particular SNRs are found to be \(-5\) dB, \(-1\) dB and 2 dB, respectively. These values correspond to the thresholds 0.033, 0.11 and 0.24, respectively.

### 8.4.2 Large Number of Targets

This experiment demonstrates two aspects of various algorithms. First, we study the effect of having a large number of targets in the range bin of interest. Second, we evaluate the resolution performance of different algorithms. We consider 15 targets. To illustrate Doppler resolution, we always keep two targets in the range bin 23. One target is still and the other target is moving slowly at \(-7.5\) m/s. For rest 13 targets, we select the range bins \([10 : 30]\) and Doppler bins \([-500 : 500]\). We then randomly pick 13 range bins and 13 Doppler bins from these intervals. We place 13 targets in the selected locations and set their moving velocities corresponding to the associated Doppler frequency. We fix their SNR randomly between 10 dB to 30 dB. A typical arrangement is shown in Table 8.3 and the corresponding output of different algorithms is shown in Figure 8.7. As before 2D-SL0 gives better results than the other methods. Note that \( \ell_1 \) also suffers a little in providing super Doppler resolution. Nevertheless, 2D-SL0 is still able to provide clear separation between all targets.
8.4. Simulation Results

8.4.3 Performance in Presence of Land Clutter

In this section, we compare performance of different algorithms in presence of land clutter. A clutter is largely stationary (internal clutter motion that induces Doppler spreading is small) and exists over all range. Hence, clutter can be presumed sparse in the Doppler domain but not in range bin. In this work, the clutter is modeled using a Rayleigh distribution (see [136] eq (6.43)), where the maximum absolute speed of any scatterer in the clutter is 3.75 m/s. The clutter to noise ratio (CNR) is 40 dB. We consider two targets in the range bin 10 and 15 with moving velocity $-30$ m/s and $37.5$ m/s (corresponding Doppler shifts are $-200$ Hz and $250$ Hz) respectively. The SNR of every target is 10 dB. The usual way to contend with this clutter is to apply clutter cancellation. However, the experimental result (Figure 8.8) demonstrates that the proposed algorithms need no clutter cancellation step. Note that $\ell_1$ outperforms all other algorithms in presence of clutter. 2D-SL0 is also able to detect the clutters, however, it has some noise at the boundary. IAA is also able to detect the targets fair accurately.
Figure 8.7: Range-Doppler imaging with many targets. Target setting is given in Table 8.3. $N = 13$ and $Q = 16$. (a) MF. (b) IAA with IV filter as the initial estimator. (C) 2D-SL0. (d) $\ell_1$

Table 8.3: Ground Truth 2

<table>
<thead>
<tr>
<th>Target</th>
<th>Range bin</th>
<th>Velocity (m/s)</th>
<th>Doppler bin (Hz)</th>
<th>SNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>$-3.75$</td>
<td>$-25$</td>
<td>30</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>$-18.75$</td>
<td>$-125$</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>30</td>
<td>200</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td>7.5</td>
<td>50</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>18</td>
<td>$-22.5$</td>
<td>$-150$</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>3.75</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>21</td>
<td>22.5</td>
<td>150</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>23</td>
<td>$-7.5$</td>
<td>$-50$</td>
<td>25</td>
</tr>
<tr>
<td>10</td>
<td>24</td>
<td>26.25</td>
<td>175</td>
<td>25</td>
</tr>
<tr>
<td>11</td>
<td>26</td>
<td>15</td>
<td>100</td>
<td>25</td>
</tr>
<tr>
<td>12</td>
<td>27</td>
<td>$-7.5$</td>
<td>$-50$</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>28</td>
<td>18.75</td>
<td>125</td>
<td>15</td>
</tr>
<tr>
<td>14</td>
<td>29</td>
<td>11.25</td>
<td>75</td>
<td>15</td>
</tr>
<tr>
<td>15</td>
<td>30</td>
<td>$-15$</td>
<td>$-100$</td>
<td>30</td>
</tr>
</tbody>
</table>
Figure 8.8: Range-Doppler imaging in presence of range-spread clutter. $N = 13$ and $Q = 16$. (a) MF. (b) IAA with IV filter as the initial estimator. (C) 2D-SL0. (d) $\ell_1$
8.5 Conclusion

In this chapter, we show that the range-Doppler imaging problem can be cast as a sparse signal reconstruction problem. The underlying model is suitable for both $\ell_1$ and $\ell_0$ approximation based sparse recovery algorithms. We have compared the performances of sparse signal recovery algorithms with the known imaging algorithms like MF and IAA in a number of situations. In our experiments the sparse recovery algorithms demonstrate clear advantage over the traditional algorithms. The sparse recovery algorithms are fast, give better resolution performance, more robust to noise, and can handle larger number of sources. The 2D-SL0 algorithm proposed in [128] makes a tradeoff between target estimation and computation time. The $\ell_1$ approach is found to be the best in presence of range-spread clutters. The proposed sparse recovery based algorithms depend on user parameters selection. However, the simulation results demonstrate that the algorithms are less sensitive to the user parameters for a wide range of SNR. On the other hand, other algorithms do not contain user parameters.
Chapter 9

Target Detection in MTI Radar

9.1 Introduction

9.1.1 Background

Airborne Ground-Moving-Target-Indicator (GMTI) radar systems illuminate both targets and the background clutter \[137, 138\]. Detection of targets in presence of non-stationary clutter and noise is a problem of interest in both radar and active sonar systems. Mitigation of clutter can be achieved, in principle, by using space-time adaptive processing (STAP) \[138\]. In STAP, the clutter-and-noise covariance matrix of the range bin of interest is estimated from training data. However, the estimation accuracy relies critically on the availability of sufficient target-free training data. Many approaches have been proposed for selecting high quality training data for accurate estimation of the covariance matrix \[137, 138\]. However, in presence of multiple targets, the angle-Doppler image formed by using the estimated covariance matrix is not optimal since the target statistics is not considered in the estimation.

Knowledge-aided STAP can overcome the limitation of requiring high quality training data \[139, 140\]. However, in complex propagation conditions and highly variable backscatter, the clutter return is often non-stationary. This results in inaccurate prior knowledge and performance degradation of STAP. Different methods have been proposed to avoid this limitation \[141, 142, 143, 144\]. The joint-domain localized method \[141\] considers the delay-and-sum (DAS) type of approaches to transform the data into the angle-Doppler domain. However, DAS is a data-independent approach which suffers from broad main
beam (smearing) and high side-lobe level (leakage) problems. Spatially adaptive detectors proposed in [142] can discriminate target echoes in presence of spatially structured interferences. However, this approach can not perform well to separate targets from Doppler spread clutter. Adaptive temporal processing in [143] models the clutter return as a complex Gaussian process, and employs the maximum likelihood (ML) approach to estimate the targets and the clutter parameters from multiple snapshots of the radar data. However, non-stationarity of clutter return across the range of interest limits the number of snapshots available for the temporal processing techniques. An abruptly time-varying autoregressive (ATVAR) model proposed in [144] can accommodate both slowly varying and rapidly varying non-stationaries in the clutter statistics during an intra-coherent processing interval (CPI). The clutter covariance estimates are derived using an approximate ML estimate of the time-varying autoregressive (TVAR) technique. These estimates are used to form a generalized likelihood ratio test (GLRT) for target detection.

Some algorithms achieve high resolution angle-Doppler imaging by processing the data in each range bin of interest (also known as primary data) independently [145, 146, 147]. These methods do not need training data for accurate knowledge of clutter statistics. In [145], a non-parametric adaptive algorithm called iterative adaptive approach (IAA) [132] is used for high resolution angle-Doppler imaging. IAA works better for both stationary and non-stationary clutter. However, its performance degrades when the target signal-to-noise ratio (SNR) is low. Moreover, in presence of steering vector error, IAA can not perform equally well. Different methods have been proposed to combat steering vector error in target localization [148, 149]. However, they are not directly applicable to angle-Doppler imaging. Sparse recovery approaches have been shown to achieve high resolution [150, 151, 146, 147]. Moreover, they are robust to noise. In sparse representation framework the problem of estimating the complex amplitudes of the reflected signals becomes a sparse signal recovery problem [5, 1]. In [150], a compressed sensing (CS) based high-resolution radar system has been proposed. It is shown that CS based radar has the potential to reduce the overall data rate and to simplify hardware design. A sparse angle-Doppler radar framework is proposed in [151]. However both of the approaches [150, 151] consider clutter free environment.
9.1.2 Contributions and Motivation

In last two chapters, we evaluated performance of different sparse recovery algorithms for target detection in radar system. The target detection performance of sparse recovery algorithms in presence of small clutter is demonstrated in Section 8.4.3. However, in presence of large clutter those algorithms cannot perform equally. The applications of sparse recovery approach in the STAP context has been considered in [146, 147]. However, those approaches fail to locate targets in presence of large non-stationary clutter.

In this chapter, we propose a sparse signal recovery algorithm that can resolve targets in presence of large clutter. The algorithm uses the primary data of each range bin of interest independently. The basic problem of finding the sparse signal representation of a signal in an overcomplete bases requires computing the sparest vector \( x \) satisfying the linear system of equation \( y = \Phi x \), where \( y \in \mathbb{C}^m \) and \( \Phi \in \mathbb{C}^{m \times n}, m < n \). Without the sparsity prior on \( x \), the system of equations \( y = \Phi x \) has infinitely many solutions. Additional information that \( x \) should be sufficiently sparse allows unique solvability [146, 1]. Hence, target detection problem in presence of small clutter can be posed as a sparse signal recovery problem. However, in presence of a large clutter, \( x \) is no longer sparse and hence the conventional sparse recovery approaches perform poorly. The method proposed in this chapter exploit the fact that under certain mild assumptions the clutter can represent as a large cluster in the angle-Doppler image. We demonstrate how to adopt the clutter information in the context of “sparse recovery with partially known support” (see Chapter 4). In MTI radar, one can use the approximate information of clutter location as the known part of support. We then apply the PMAP algorithm (see Section 4.4) to angle-Doppler imaging problem where we are able to handle a large extended scatterer (the clutter) and other point-scatterers (targets) within a unified framework. Sparse representation with such implicit clustering of the clutter allows us to restrict the clutter energy concentrated into a predefined area which in turn results in a significant improvement of the angle-Doppler resolution performance. This approach prevents energy “leakage” from the clutter, and in this way helps to detect the targets even when the target is moving slowly and the signal-to-noise ratio is low. In addition, we introduce a new method to combat the steering vector error that arises in practical MTI radar. We have found that under realistic steering vector error, the proposed correction approach is equally applicable to other conventional algorithms like, IAA [145]. In the experiments, the proposed approach exhibits a number of
advantages over other angle-Doppler imaging techniques, which include increased resolution, and improved robustness to noise. In addition, it does not require an accurate initialization.

9.2 Problem Formulation

The basic theory of STAP is a mature subject [138, 137]. Consider an airborne radar, which uses a \( M \) element sensor array. It transmits \( N \) pulses during a coherent processing interval (CPI) at a pulse repetition interval \( H \). The transmitted signals are reflected from the targets and the reflections are received. Let the whole range of interest be divided into \( R \) uniform range-bins. For a fixed elevation angle, the location of the \( t \)-th target can be specified by its range index \( i \), azimuth angle \( \bar{f}_t \), and normalized Doppler frequency \( \bar{\omega}_t \). In the following, we assume that the data due to the return for each range bin is processed independently. Hence we drop the range bin index. Then the \( t \)-th target is indicated by an angle-Doppler pair \((\bar{f}_t, \bar{\omega}_t)\). We assume there is no range and Doppler frequency ambiguities. This is a common assumption [145].

In this work we make use of an overcomplete representation of received signal of the radar. We set up a uniform grid over the whole azimuth angle range consisting of \( T \) points. Similarly, we set up a \( L \) point uniform grid over the Doppler frequency range. Let \( X \in \mathbb{C}^{T \times L} \) be a matrix such that \( X_{t,\ell} \) (i.e. the element at \( t \)-th row and \( \ell \)-th column) denotes the complex-valued amplitude of the returned signal due to the reflection from a target located at \((f_t, \omega_\ell)\). Every \( X_{t,\ell} \) is assumed to remain constant during a CPI and is proportional to the radar-cross-section of the corresponding target. The data vector \( \mathbf{y}(j) \) obtained by matched filtering and digitizing the received signal due to the \( j \)-th pulse can be modeled as [146, 145]

\[
\mathbf{y}(j) = \sum_{t=1}^{T} \sum_{\ell=1}^{L} \phi(f_t) X_{t,\ell} e^{i2\pi(j-1)\omega_\ell} + \mathbf{e}(n),
\]

where \( \mathbf{y}(j) \in \mathbb{C}^{M \times 1} \), \( \mathbf{e}(j) \) accounts for additive noise, and \( \phi(f_t) \in \mathbb{C}^{M \times 1} \) denotes the array steering vector corresponding to an azimuth angle \( f_t \), which depends on the array geometry. Defining \( \Phi = [\phi(f_1) \cdots \phi(f_T)] \), we get

\[
\mathbf{Y} = [\mathbf{y}(1) \cdots \mathbf{y}(N)] = \Phi X \Theta + \mathbf{E}
\]

where \( \mathbf{E} = [\mathbf{e}(1) \cdots \mathbf{e}(N)] \), and the matrix \( \Theta \) is given element-wise as

\[
[\Theta]_{\ell,j} = e^{i2\pi\omega_\ell(j-1)} : \quad j = 1, \cdots, N
\]
and $\theta(\omega_t) = [1 \ e^{i2\pi \omega_t} \ldots e^{i2\pi \omega_t(N-1)}]$ indicates one row corresponding to Doppler $\omega_t$. We can use $X$ to visualize the azimuth angle and Doppler shifts of the targets in form of an image, with each $X_{t,\ell}$ representing a pixel. A non-zero $X_{t,\ell}$ represents a target at the $t$-th azimuth angle and the $\ell$-th Doppler bin. Since the user designs the grid, the matrices $\Phi, \Theta$ are known and do not depend on the actual target location $(\bar{f}_t, \bar{\omega}_t)$.

Note that for any given $t$, the true attribute $(\bar{f}_t, \bar{\omega}_t)$ may not lie exactly on a grid-point of our azimuth-Doppler grid. In other words, there may not exist $i$ and $j$ satisfying $\bar{f}_t = f_i$ and $\bar{\omega}_t = \omega_j$ exactly. Nevertheless, by making the grid dense enough one can ensure $\bar{f}_t \approx f_i$ and $\bar{\omega}_t \approx \omega_j$ for some $i$ and $j$, and the remaining model error is absorbed by the residual term $E$. Note that, in practice $X$ is a sparse matrix as only a few targets present in the area of interest. Suppose there are $P$ targets. Then ideally only $P$ pixels of $X$ are nonzero, i.e. $X_{i,j} \neq 0$ only if $f_i \approx \bar{f}_t$ and $\omega_j \approx \bar{\omega}_t$ for some $t \in \{1, 2, \ldots, P\}$. In effect, the representation (9.2) helps to pose the problem of estimating $(\bar{f}_k, \bar{\omega}_t)^P_{t=1}$ as that of estimating the sparse matrix $X$. Different algorithms has been proposed to exploit the sparsity [146, 147]. The central assumption of those algorithms is that the number of non-zero components in $X$ is very small. However, in presence of a big clutter, a large number of point scatterers are needed to represent the clutter ridge. As a result, the non-zero components of $X$ increases significantly, and $X$ can no longer be considered sparse. This is why many conventional algorithms fail in presence of clutter. In the following we adopt the PMAP algorithm to deal with this issue.

9.3 Proposed Reconstruction Approach

In the following we propose a two step algorithm. First we approximately identify the clutter location. We then apply a sparse recovery approach called PMAP to locate the moving targets.

9.3.1 Step 1 - Approximate Localization of the Clutter Spread

Let us consider a scatterer within the clutter located at an azimuth angle $f$ relative to the array. Then for a side-looking airborne radar and small carb angle, the normalized Doppler shift $\omega$ induced on the scatterer is given by [138]

$$\omega = \frac{2v_{air}T}{\gamma} \sin(f) =: \beta \bar{f},$$  \hfill (9.3)
where \( \hat{f} = \sin(f) \), \( v_{air} \) is the speed of airborne radar and \( \gamma \) is the operating wavelength. However, clutter has intrinsic motion resulting in random Doppler spreads causing the clutter energy to “leak” in both sides of the line \( \omega = \beta \hat{f} \) along the \( \omega \) direction.

Our goal here is to approximately estimate the spread of the clutter using a fast algorithm. This knowledge will be used later in the sparse signal recovery approach. Given the a priori information in form of (9.3), a maximum a posteriori estimation (MAP) approach appears to be most appropriate. We assume that there are no targets and the non-zero components in \( X \) are due to the clutter only. This assumption is reasonable in the sense that the clutter to signal ratio is generally above 30 dB in practice [136], and the target energy is significantly smaller than the clutter energy\(^1\). We assume that the a priori density of \( X_{t,\ell} \) is a Gaussian density with mean zero and variance \( \sigma_{t,\ell} \), and \( X_{t_1,\ell_1} \) is independent of \( X_{t_2,\ell_2} \) if \( t_1 \neq t_2 \) and/or \( \ell_1 \neq \ell_2 \). The fact that the clutter is concentrated around the line \( \omega = \beta \hat{f} \) is used to specify the value \( \sigma_{t,\ell} \). We set

\[
\sigma_{t,\ell} = e^{-\kappa(\omega_{\ell}-\beta \hat{f})^2},
\]

where \( \kappa \) is a constant whose value depends on the clutter spread. In practice, some knowledge of the variance of the Doppler spread due to intrinsic motion in the clutter is known from physical experiments [152, 153]. With this a priori density the maximum a posteriori estimate of \( X \) is given by [75]

\[
\hat{X}^{(0)} = \arg \min_{X} \sum_{t=1}^{T} \sum_{\ell=1}^{L} \frac{|X_{t,\ell}|^2}{2\sigma_{t,\ell}} \text{ subject to } Y = \Phi X \Theta. \tag{9.4}
\]

Note that we are assuming that the variance of the elements of \( E \) are negligible compared to the clutter energy, which is quite appropriate from a practical point of view. The solution to (9.4) can be given analytically, and is well-known. Let

\[
\Sigma = \text{diag} \left\{ \text{vec} \left( \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1L} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{T1} & \sigma_{T2} & \cdots & \sigma_{TL} \end{bmatrix} \right) \right\},
\]

\[
x^{(0)} = \text{vec}(\hat{X}^{(0)}), \quad y = \text{vec}(Y), \quad \Psi = \Theta' \otimes \Phi. \quad \text{Then}
\]

\[
x^{(0)} = \Sigma \Psi^* (\Psi \Sigma \Psi^*)^{-1} y. \tag{9.5}
\]

\(^1\)In fact, the typically large values of the clutter to signal ratio makes it challenging to detect targets in presence of clutter.
Since the values of $\sigma_{t,\ell}$ in (9.4) is just a guess, one would expect somewhat better estimation performance if we repeat the estimation with better estimates of $\sigma_{t,\ell}$. This suggests an iterative procedure where we compute the $i$-th iterate $\hat{x}^{(i)} = \text{vec}(\hat{X}^{(i)})$, $i > 0$ as

$$x^{(i)} = \Sigma_i \Psi^*(\Psi \Sigma_i \Psi^*)^{-1} y,$$

where for $i \geq 0$ we define

$$\Sigma_{i+1} = \text{diag} \left\{ \text{vec} \begin{pmatrix} |\hat{X}_{11}^{(i)}|^2 & \cdots & |\hat{X}_{1L}^{(i)}|^2 \\ \vdots & \ddots & \vdots \\ |\hat{X}_{T1}^{(i)}|^2 & \cdots & |\hat{X}_{TL}^{(i)}|^2 \end{pmatrix} \right\}.$$

The iterative approach described above has been used in several contexts in the signal processing literature, and is often called the iteratively re-weighted least squares (IRLS) approach [154]. The IRLS algorithm may take large number of iterations to converge. However we are not interested in convergence in the current context. Instead we want to get an approximate knowledge of clutter. We define the initial weights of IRLS using a MAP estimation. This approach can reduce the iterations of IRLS and helps to find an approximation of the clutter region in few iterations. We find that MAP/IRLS can approximate clutter after three iterations. By the word approximate we emphasize that we need not estimate the energy of each individual scatterers within the clutter accurately. Instead we seek an approximate knowledge of the region where most of the clutter-energy is concentrated. Figure 9.1 illustrates the idea. The angle-Doppler image of a typical clutter (considering $\beta = 0.5$ in (9.3)) is shown in Figure 9.1 (a). The recovered angle-Doppler image estimated after three IRLS iterations is shown in Figure 9.1(b).

We need a criterion to determine the clutter spread from the angle-Doppler image (Figure 9.1(b)) obtained using the MAP/IRLS approach described above. In that goal, let us define the cumulative distribution function

$$h(\delta) := \sum_{t=1}^{T} \left\{ \sum_{\ell \in L_\delta(t)} |\hat{X}_{t,\ell}^{(3)}|^2 \right\},$$

where $L_\delta(t) = \{\ell : \omega_\ell < \beta \hat{f}_t + \delta\}$. Figure 9.1(c) shows $h(\delta)$ as a function of $\delta$ for angle-Doppler images obtained by different algorithms. As expected, $h(\delta)$ increases rapidly around the clutter region i.e., around $\delta = 0$. Note that IRLS provides a closer estimation of clutter energy compared to DAS [141]. Now to approximate the clutter region we have to determine
a band around the line $\omega = \beta \hat{f}$ so that most of the energy lie within this band. Hence we set a small $\varepsilon > 0$, and then find the clutter region as

$$
\mathbb{I}_\varepsilon = \left\{ (t, \ell) : \varepsilon \leq \frac{h(\omega_t - \beta f_t)}{\sum_{t=1}^{T} \sum_{\ell=1}^{L} |X(t, \ell)|^2} \leq 1 - \varepsilon \right\}.
$$

(9.8)

In the experiments, we set $\varepsilon = 0.02$.

### 9.3.2 Step 2 - Target Detection using Sparse Recovery Algorithm

Once we have an approximate knowledge of the spread of the clutter in form of the set of indices $\mathbb{I}_\varepsilon$, we can formulate a sparse signal recovery approach for localizing the targets. The idea here is to treat the set of components $\{X(t, \ell)\}_{(t, \ell)\in \mathbb{I}_\varepsilon}$ as a single cluster or a single object in the angle-Doppler image, while the targets outside the clutter are treated as isolated point objects. This can be achieved by using the sparse recovery formulation proposed in Chapter 4. In the formulation of Chapter 4, we assumed that some a priori information about support of sparse $X$ is known and using the information we facilitate the recovery process. In MTI radar, we can consider $\mathbb{I}_\varepsilon$ as the known part of support. In sparse recovery paradigm one solves the over-determined system $Y = \Phi X \Theta$ such that the number of distinct objects in the image represented by the solution $X$ is minimized. Typically, the targets in the angle-Doppler image are point objects which span over one pixel. On the other hand the clutter is an extended object which span over an extended area approximated by the set $\mathbb{I}_\varepsilon$. In order to formalize the sparse recovery paradigm for detecting the target we need to minimize the number of objects in the image. Hence we
propose to solve the following optimization problem:

$$\min_X I \left\{ \sqrt{\sum_{(t, \ell) \in I_\varepsilon} |X_{t, \ell}|^2} \right\} + \sum_{(t, \ell) \in O} I(|X_{t, \ell}|),$$  \hspace{1cm} (9.9)

subject to the constraint $Y = \Phi X \Theta$, where $I$ is an indicator function defined in (3.5) and

$$O := \{(t, \ell) : t = 1, \ldots, T, \ \ell = 1, 2, \ldots, L\} \setminus I_\varepsilon.$$

Note that the constraint assumes noise-free data, which can be modified to accommodate the measurements noise using some standard techniques described later. Also note that the cost function in (9.9) essentially counts the number of objects in the image $X$ including the clutter supported over the set $I_\varepsilon$. If $I_\varepsilon = \emptyset$ then the cost function reduces to the $\ell_0$ norm of $\text{vec}(X)$. Hence under the assumption that $I_\varepsilon \neq \emptyset$, (9.9) reduces to

$$\min_X \sum_{(t, \ell) \in O} I(|X_{t, \ell}|), \hspace{1cm} \text{s.t. } Y = \Phi X \Theta \hspace{1cm} (9.10)$$

Note that the optimization problem in (9.10) is same to (4.1). However, solving (9.10) is NP-hard. To handle this issue, some relaxation schemes are used in the compressed sensing [64, 65]. A brief discussion of those methods is provided in Section 4.2. However, simulation results in Section 4.5 demonstrate that those methods can not perform well in noisy environment. We also found that the methods in [64, 65] cannot perform equally for target detection with MTI radar system. This is why we turn our attention to apply PMAP algorithm for target detection.

### 9.3.2.1 PMAP based Approach

A maximum a posteriori estimation based approach called PMAP (partially known support using MAP) is proposed in Section 4.3. We extend PMAP for noisy cases in Section 4.4. In the following we show that PMAP can extend readily for STAP context.

In PMAP, we set up a priori density function for $X$. According to the prior, a component of $X$ within support $I_\varepsilon$ is of large magnitude with a large probability, and, a component of $X$ outside the known support is of large magnitude with a very small probability. We then consider the PMAP solution strategy proposed in Section 4.4 to detect the targets. As before, we minimize a negative log-likelihood function like (4.22) and
to deal with the nonconvexity of the resulting problem, we solve a sequence of optimization problem, where

$$
\hat{X}^{(j)} = \arg \min_X \hat{\varphi}_j(X)
$$

(9.11)

where, 

$$
\hat{\varphi}_j(X) = \hat{\varphi}_j(X) + \frac{\lambda_j}{2} \| Y - \Phi X \Theta \|_F^2.
$$

and, 

$$
\hat{\varphi}_j(X) = \sum_{t=1}^{T} \sum_{\ell=1}^{L} \left\{ \frac{|X_{t,\ell}|^2}{2\sigma^2} - \ln \left[ 1 + \rho(t,\ell)_j \exp \left( -\frac{|X_{t,\ell}|^2}{2\hat{\sigma}_j^2} \right) \right] \right\},
$$

(9.12)

where

$$
\lambda_j = \frac{(\lambda/\lambda_0)^{j/w}}{\lambda_0}, \rho(t,\ell)_j = \frac{(r(t,\ell)/r(t,\ell)_0)^{j/w}r(t,\ell)_0}{r(t,\ell)_0^{j/w}}, \; r(t,\ell) = \left(1 - \rho(t,\ell) \right)^{\mu}
$$

$\sigma$ is the variance of clutter signal and $p(t,\ell)$ is the probability of $X_{t,\ell}$ taking a large magnitude.

In this work we select

$$
p(t,\ell) = \begin{cases} 0.99, & (t, \ell) \in \Omega \\ 0.01, & (t, \ell) \in \Omega \end{cases}
$$

(9.13)

The value of $\{p(t,\ell) : (t, \ell) \in \Omega\}$ is small due to the fact that the number of targets are significantly smaller than the number of clutter scatters. The value of $r(t,\ell)_0$ and $\hat{\sigma}$ can select using the procedure described in Section 4.3.2. Also Section 4.4 described the process of selecting the value of $\lambda$. At $j = 0$, we take $\mu = \sigma$ and solve the optimization problem in (9.11). We then increase $j$ from 0 to $w$ and at each step we solve the optimization problem in (9.11). Finally at $j = w$, we reach to our desire likelihood function $\hat{\varphi}_w(X)$.

### 9.3.3 Steering Vector Error

So far we have assumed that the matrix $\Phi$ is known exactly. In this section, we consider the error in $\Phi$ due to imperfections in sensor positioning. We consider a generic 3 dimensional arrangement of an arbitrary sensor array. The nominal location of the $i$-th sensor is indicated by $z^{(i)} = (z_1^{(i)}, z_2^{(i)}, z_3^{(i)})$. However, due to manufacturing errors, the actual sensor location may not be the same as the nominal locations. Let the coordinates of actual sensor locations be given by

$$
\bar{z}^{(i)} = z^{(i)} + \delta^{(i)}; \; \text{for} \; i = 1, \cdots, M - 1,
$$

(9.14)

where $\delta^{(i)} = (\delta_1^{(i)}, \delta_2^{(i)}, \delta_3^{(i)})$ denote the unknown error in positioning the $i$th sensor. Note that without any loss of generality we can set $\bar{z}^{(0)} = z^{(0)} = 0.$
In the following we distinguish between the true manifold matrix $\Phi$ associated with the true (but unknown) sensor positions $\{z^{(i)}\}_{i=0}^{M-1}$, and the nominal manifold matrix $\Phi$ corresponding to the nominal sensor positions $\{z^{(i)}\}_{i=0}^{M-1}$, and characterize $\Phi := \Phi - \Phi$ in terms of $\{\delta^{(i)}\}_{i=1}^{M-1}$.

The $i$-th row of the array manifold matrix in (9.2) is a function of the position of $i$-th sensor. In particular, we know the function $\gamma$ so that we can write

$$
\Phi = \begin{bmatrix}
\gamma(z^{(0)}) \\
\gamma(z^{(1)}) \\
\vdots \\
\gamma(z^{(M-1)}) \\
\end{bmatrix}, \quad \Phi = \begin{bmatrix}
\gamma(z^{(0)}) \\
\gamma(z^{(1)}) \\
\vdots \\
\gamma(z^{(M-1)}) \\
\end{bmatrix}.
$$

(9.15)

Now for small $||\delta^{(i)}||_2$ we can neglect the higher order terms of the Taylor’s series expansion of $\gamma(z^{(i)})$ in the neighborhood of $z^{(i)}$ to get

$$
\gamma(z^{(i)}) = \gamma(z^{(i)}) + \delta^{(i)}_1 \gamma_1(z^{(i)}) + \delta^{(i)}_2 \gamma_2(z^{(i)}) + \delta^{(i)}_3 \gamma_3(z^{(i)})
$$

(9.16)

where, $\gamma_t(z) = \frac{\partial \gamma(z)}{\partial z^t}$, $t = 1, 2, 3$. Now let

$$
\Phi_t = \begin{bmatrix}
0 \\
\gamma_t(z^{(1)}) \\
\vdots \\
\gamma_t(z^{(M-1)}) \\
\end{bmatrix}, \quad t = 1, 2, 3.
$$

(9.16)

Then by (9.15) and (9.16) we have

$$
\tilde{\Phi} = \Phi - \Phi = \Delta_1 \Phi_1 + \Delta_2 \Phi_2 + \Delta_3 \Phi_3
$$

where, $\Delta_t = \text{diag}(0 \cdot \delta^{(1)}_t \cdot \ldots \cdot \delta^{(M-1)}_t)$. Consequently, we have a modified model

$$
Y = \Phi X \Theta + E = \{\Phi + \tilde{\Phi}\} X \Theta + E
$$

$$
= \Phi X \Theta + \tilde{E}
$$

(9.17)

where, $\tilde{E} = \tilde{\Phi} X \Theta + E$. Thus if we knew $\tilde{E}$, then we could estimate $\{\Delta_t\}_{t=1}^3$ by solving a least squares problem

$$
\min_{\Delta_1, \Delta_2, \Delta_3} ||\tilde{E} - (\Delta_1 \Phi_1 + \Delta_2 \Phi_2 + \Delta_3 \Phi_3) X \Theta||_F^2.
$$

(9.18)
In the following we assume \( E \{ ||E||^2 \} = \mu^2 \), where \( E \) is the mathematical expectation operator. Furthermore, we assume the sensor errors \( \{ \delta^{(i)} \}_{i=1}^{M-1} \) are independent of \( E \), and are mutually independent and identically distributed zero mean random vectors with a known covariance matrix \( \nu^2 I \). Then it can be shown that

\[
\chi^2 := E \{ ||\tilde{E}||^2 \} = \mu^2 + \nu^2 \text{Tr}\{ \Phi_1 \Xi \Phi_1^* + \Phi_2 \Xi \Phi_2^* + \Phi_3 \Xi \Phi_3^* \} \tag{9.19}
\]

where \( \Xi = X \Theta \Theta^* X^* \). Using the observations made so far we now present an iterative approach which is able to correct the sensor positioning error while estimating \( X \). This approach relies on tuning the parameter \( \lambda \) in (9.11). If we neglect the sensor positioning error then according to the concept in Section 4.4 we should set

\[
\lambda = \frac{1}{\chi^2} \tag{9.20}
\]

which is the starting value of \( \lambda \) used in our algorithm. Note that we need \( X \) to compute \( \chi \). For this purpose we use the solution obtained via the IRLS approach discussed in Section 9.3.1. Using the value of \( \lambda \) in (9.20) we can obtain an initial estimate \( \tilde{X}^{(1)} \) which can be used to compute an estimate of \( \tilde{E} \). This estimate of \( \tilde{E} \) can be used to solve the optimization problem in (9.18) to estimate the sensor positioning errors. Next we can update the manifold matrix \( \Phi \) by incorporating the estimates of \( \{ \delta^{(i)} \}_{i=1}^{M-1} \), and the procedure can be repeated.

The resulting iterative approach is outlined in Table 9.1. The iterative algorithm in Table 9.1 is basically refines the estimates of \( X \) and \( \{ \delta^{(j)} \}_{j=0}^{M-1} \) in each iteration.

### 9.4 Simulation Results

We consider an airborne radar system that uses \( M = 10 \) elements uniform linear array. The sensors are separated by a half wavelength of the operating narrow-band frequency. In each CPI the radar transmits \( N = 32 \) pulses. The radar pulse repletion frequency (PRF) 1 KHz. We mainly consider the scenario where clutter has significant intrinsic clutter motion (ICM) [152, 153]. In all simulations, we consider \( \beta = 0.5 \) in (9.3). A typical angle-Doppler image is shown in Figure 9.1(a). We assume that the noise vectors \( \{ e(n) \}_{n=1}^{N} \) are mutually independent, complex Gaussian process with zero-mean, and covariance matrix \( \eta^2 I \). With this noise covariance we define the SNR of \( p \)-th target as

\[
\text{SNR} = 10 \log_{10}(\vartheta_p^2/\eta^2) \tag{9.21}
\]
9.4. Simulation Results

Table 9.1: Steering vector error approximation

<table>
<thead>
<tr>
<th>Initialization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Set $i = 0$, $\lambda^{(i)} = 1/\chi^2$, construct $\Phi$, ${\Phi_t}<em>{t=1}^3$ using nominal ${z^{(i)}}</em>{i=1}^{M-1}$.</td>
</tr>
</tbody>
</table>

repeat

2. Estimate $X$ using algorithm in Section 9.3.2.1 with $\lambda = \lambda^{(i)}$, denote the estimate by $\hat{X}^{(i)}$.

3. Calculate the residual $\tilde{E}^{(i)} = Y - \Phi \hat{X}^{(i)} \Theta$.

4. Solve $\min_{\Delta_1, \Delta_2, \Delta_3} ||\tilde{E}^{(i)} - \sum_{t=1}^{3} \Delta_t \Phi_t \hat{X}^{(i)} \Theta||^2_F$ to obtain estimates $\hat{\delta}_t^{(1)} \ldots \hat{\delta}_t^{(M-1)}$, and denote the associated estimate of $\Phi$ by $\hat{\Phi}_t$.

5. Set $z^{(j)} = z^{(j)} + \hat{\delta}_t^{(j)}$, for $j = 0$ $\ldots$ $M-1$, and update $\Phi$, $\{\Phi_t\}_{t=1}^3$ accordingly.

6. Calculate $\chi^2$ using (9.19) with $X = \hat{X}^{(i)}$.

7. Set $\lambda^{(i+1)} = 1/\chi^2$ and $i = i + 1$.

until $\sum_{j=1}^{3} ||\delta_t^{(j)}||^2 >$ a predefined threshold.

where $\vartheta_p^2$ is the power of $p$-th target and $\eta = 0.5$. For the simulation in Figure 9.2, we set $T = 90$ and $L = 64$. However, for all other simulations, we set $T = 90$ and $L = 128$. For an arrangement with $P$ targets, we say the targets are detected correctly when an algorithm produces $P$ spikes outside the clutter region, and the maximum absolute difference between any actual Doppler and recovered Doppler is $\pi/30$ radian. We say a location in the angle-Doppler image is a spike when the location is a local maximum on the image and its absolute magnitude is above a threshold $\epsilon$. The value of $\epsilon = 10^{-4}$ for PMAP. We consider four techniques, namely, the CS based approach, iterative adaptive approach (IAA) [145], and PMAP. The values of different tuning parameters of PMAP algorithm are same to those suggested in Section 4.5.1. IAA is an iterative algorithm based on a weighted least squares criterion [132]. IAA usually converges in 10 to 14 iterations [145, 132], and thus, in our simulations we consider 14 iterations. In CS based approach we consider the following optimization problem

$$\min_x ||x||_1 \text{ subject to } ||y - \Psi x||_2 \leq \epsilon_l$$  \hspace{1cm} (9.22)

where $\epsilon_l$ is set to $\eta$. We consider SPGL1 [133] framework to optimize (9.22).

The signal-to-clutter-and-noise ratio (SCNR) is defined as [145]
where $R_{CN}$ is the true clutter-and-noise covariance matrix which is defined as \cite{145}: $R_{CN} = \Psi Q \Psi^*$, where $Q \in \mathbb{R}^{S \times S}$ is the diagonal matrix with the clutter scatterer powers corresponding to the scanning points on the diagonal. Also $f_p$ and $\omega_p$ are the azimuth angle and normalize Doppler frequency of $p$-th target and $\Omega(f_p, \omega_p) = \theta(\omega_t)' \otimes \phi(f_t)$, for reference, to resolve two targets with SNR=2dB, SCNR=−65.22 dB, $T = 90$, and $L = 128$, the computation time require for CS, IAA, and PMAP are 35, 29 and 22 sec respectively.

Modified-CS and conventional CS based approach can not detect targets in clutter environment. Figure 9.2 shows target recovery results by different algorithms. We consider a clutter environment with two targets. The targets normalized Dopplers are \{-0.015, −0.04\} and angles are \{-42°, 32°\}. The target SNR is 2 dB and SCNR is −45.12 dB. Modified-CS has no suitable implementation to work in large scale dataset. Hence, we use CVX framework for implementing modified-CS. However, CVX cannot work with large sized problem. Hence to demonstrate the performance of modified-CS, we discretize the Doppler grid into $L = 64$ parts. Note that both CS and modified-CS fail to locate actual targets. In the following we do not include the results of modified-CS and we set $L = 128$.

We consider the angle-Doppler imaging of clutter-and-noise only in Figure 9.3. The intention of this experiment is to evaluate whether any algorithm detects false target in presence of large clutter. Figure 9.3(a) shows the true clutter. Note that both IAA and PMAP do not produce false target. However, IAA generates many ripples around the surface of clutter area. Sometimes, these ripples hinder actual target location when target is close to the clutter. As expected, PMAP (Figure 9.3 (d)) concentrate the clutter energy in a defined area and there is no spread of energy outside the area. Figure 9.4 demonstrates resolution performances of algorithms. We consider two targets. The SNR of each target is 2 dB. The average SCNR is −65 dB. At first we investigate the Doppler resolution achievable by the algorithms (Figure 9.4(a)). Both targets have azimuth angle −42° whereas first target has a fixed Doppler shift −0.015. At first the Doppler shift of second target is set to 0.05. We then change the Doppler shift of second target such that the difference of Doppler shifts between the two targets becomes smaller. The minimum difference of Doppler shifts for which an algorithm can separate two targets in 90% cases is defined Doppler resolution of the algorithm. The Doppler resolution of IAA and PMAP
9.4. Simulation Results

Figure 9.2: Angle-Doppler images. SNRs of two targets are 2 dB and 2 dB. SCNR = -45.12 dB. (a) CS recovery, (b) Modified-CS, (c) IAA, (d) PMAP.
Figure 9.3: Angle-Doppler images of clutter-and-noise only. (a) true angle-Doppler image, (b)CS, (c) IAA, (d) PMAP.
9.4. Simulation Results

are 0.035 and 0.025 respectively. The angle resolution of the algorithms is illustrated in Figure 9.4(b). The Doppler shift of two targets are 0.015 and 0.005. The azimuth angle of first target is fixed to 42°. The azimuth angle of second target is making closer to the angle of first one. The result in Figure 9.4(b) shows that the angle resolution of IAA is better than PMAP. The angle resolution of IAA and PMAP are 14° and 16° respectively.

Finally we investigate the target detection capability of algorithms when target is close to the clutter (Figure 9.4(c)). The first target has azimuth angle and Doppler shift are 42° and 0.015 respectively. The second target has a fixed Doppler shift 0.035 with variable azimuth angle. At first we set the azimuth angle to 30°. The azimuth angle of the second target is then making smaller so that it becomes closer to the clutter. Clearly PMAP performs better than IAA. PMAP resolves target closer to clutter. Figure 9.5 shows the angle-Doppler image when azimuth angle of second target is 21°. Note that CS fails to locate targets in presence of large clutter. IAA algorithm can give approximation of the target located at 42° azimuth angle. However, it cannot provide any estimation of the second target. PMAP detects the targets clearly. Note that by clustering the approximate clutter locations, we have several advantages. First, the clutter region is separate from the whole area which does not produce any ripples. Second, by enforcing the whole clutter as an independent component the number of active components in the whole angle-Doppler image are reduced significantly. This is very helpful in sparse recovery techniques. Finally, as observed in the simulations, the clustering concept helps PMAP to become robust in the noisy environment. Note that the actual targets in Fig. 9.5 (a) are not very bright due to low SNR. However, PMAP produces very clear image of the targets.

We investigate robustness of the algorithms in different environments in Figure 9.6. At first we show effect of noise variance η (see (9.21)) on different algorithms in Figure 9.6(a). We consider two targets. We fix target SNR to 2 dB. Clearly SCNR changes with changing η. We maintain the clutter power such that average SCNR remains −65 dB when η = 0.5. The detection of an algorithm is considered successful when it resolves all targets. The result is shown in Figure 9.6(a). PMAP outperforms IAA. One interesting outcome is that IAA can perform well for 0.3 ≤ η ≤ 0.6. The performance of IAA degrades slightly when we decreases η below 0.3. To illustrate the effect of clutter power on the detection performance we again consider two targets. The target SNR is 2 dB and η = 0.5. We then increase clutter power and try to resolve the targets using different algorithms (Figure 9.6(b)). We find that PMAP can resolve targets efficiently if clutter SCNR remains
Figure 9.4: Resolution performance of algorithms. Average SCNR = $-65$ dB. (a) Doppler resolution, (b) Angle resolution, (c) Clutter resolution.
Figure 9.5: Angle-Doppler images. SNRs of two targets are 2 dB and 2 dB. SCNR = −66.12 dB. Azimuth angles of the targets are −42° and 21°. Normalize Doppler shifts are −0.015 and −0.035. (a) true image, (b) CS, (c) IAA, (d) PMAP.
9. Target Detection in MTI Radar

above −75 dB. In Figure 9.6(c) we investigate the maximum number of targets that can be resolved by algorithms. When placing targets around the clutter we always maintain that the minimum separation between targets is sufficiently large such that the minimum separation satisfies the resolution criteria described around Figure 9.4. PMAP can separate up to 8 targets properly.

We claim in Section 9.3.1 that the algorithm needs an approximate knowledge of the true clutter. To justify the argument we consider a different clutter environment in Figure 9.7 (a). Here clutter is not distributed evenly on both sides of the diagonal of the angle-Doppler image. Also clutter is not distributed in whole diagonal area (similar to the simulations in [145]). However as demonstrate in Figure 9.7 (d), PMAP still locate the targets. IAA can gives an approximation of the left target.

Figure 9.8 illustrates the effectiveness of using steering vector error correction presented in Table 9.1. We set \( \nu = 0.01 \) to generate \( \delta \) in (9.14). We consider two targets with SNR 2 dB and SCNR = −64.8 dB. Figure 9.8 (a) and (b) shows that the range-Doppler imaging output by IAA and PMAP respectively, without error correction. Note that in presence of steering vector error none of them produces satisfactory output. We then use algorithm of Table 9.1 to reproduce \( \Phi \). The new output using modified \( \Phi \) is shown in Figure 9.8 (c) and (d). Note that PMAP separate targets clearly. IAA also improves its output. However in presence of large clutter, it can not locate targets. To clarify the applicability of the algorithm in IAA, we consider a different scenario in Figure 9.9. In this setup SCNR = −56.8 dB and target SNR = 2 dB. As can be seen in Figure 9.9 (b), IAA is able to detect the targets fairly. Figure 9.10 shows frequency of resolving targets as a function of steering vector error \( \nu \). The targets arrangement is similar to Figure 9.8 and average SCNR = −65 dB. We found that until \( \nu \leq 0.015 \), PMAP can resolve targets efficiently.

Figure 9.11 depicted the receiver operating characteristic (ROC) of PMAP. We consider two radar setup. In first setup, the radar is without steering vector error as considered in Figure 9.5. In second setup, we consider a radar system with steering vector error similar to Figure 9.8. The target-arrangements are similar to Figures'. However, the SNRs of all targets are equal and are decreasing sequentially. Let \( \hat{X} \) be estimate of \( X \) for a particular SNR. We then select a threshold \( \tau_i \). Any local maximum of \( \hat{X} \) with an absolute value is greater than \( \tau_i \) is considered as a target. We vary \( \tau_i \) within an interval \([\tau_s, \tau_e]\) and for each value of \( \tau_i \) we record whether all actual targets are detected and whether any false
9.4. Simulation Results

Figure 9.6: Robustness of algorithms. (a) Robustness against noise power. SNRs of two targets are 2 dB and 2 dB. Azimuth angles of the targets are \{-42°, 24°\} and Doppler shifts are \{-0.015, -0.035\}. (b) Robustness against clutter power. SNRs of two targets are 2 dB, 2 dB and \(\eta = 0.5\). Azimuth angles of the targets are \{-42°, 24°\} and Doppler shifts are \{-0.015, -0.035\}. (c) Multiple targets detection.
Figure 9.7: Angle-Doppler images. SNRs of two targets are 2 dB and 2 dB. SCNR = -58.22 dB. Azimuth angles of the targets are \{-42^\circ, 21^\circ\} and Doppler shifts are \{-0.015, -0.035\}. (a) True image, (b) CS, (c) IAA, (d) PMAP.
9.4. Simulation Results

Figure 9.8: Angle-Doppler images in presence of steering vector error. SNRs of two targets are 2 dB and 2 dB. SCNR = −64.8 dB. Azimuth angles of the targets are \{-42^\circ, 24^\circ\} and Doppler shifts are \{-0.015, -0.035\}. (a) IAA with steering vector error. (b) PMAP with steering vector error. (c) IAA after steering vector error correction. (d) PMAP after error correction.

Figure 9.9: Angle-Doppler images. SNRs of two targets are 2 dB. SCNR = −56.8 dB. Azimuth angles of the targets are \{-42^\circ, 24^\circ\} and Doppler shifts are \{-0.015, -0.035\}. (a) IAA with steering vector error; (b) IAA after steering vector error correction.
targets are detected. After repeating the experiment 100 times, this procedure yields the empirical probability $p_a$ of detecting all the actual targets, and the empirical probability $p_f$ of detecting any false target for different values of $\tau_i$. The sensitivity of the algorithm is defined as the smallest SNR for which $p_f \leq 0.1$ and $p_a \geq 0.85$. For both setups, the particular SNR is $-1$ dB. The value corresponding to the threshold is $10^{-4}$.

### 9.5 Conclusions

The angle-Doppler imaging problem in presence of clutter is posed as a sparse signal reconstruction problem. However, conventional sparse recovery approaches are not suitable for this framework. We find that under certain reasonable assumptions one can approximately estimate the clutter region. Using the clutter information, the problem of estimating targets can pose to the PMAP formulation proposed in Chapter 4. We compare the algorithm with other known imaging algorithms.
Figure 9.11: Receiver operating characteristic (ROC) for PMAP. (a) Without steering vector error, (b) With steering vector error.
A.1 Proof of the moment results in Section 2.2.1.3

Let $\phi_i$ be the $i$-th column of $\Phi$, i.e. $\Phi = [ \phi_1 \cdots \phi_n ]$, and define

$$\gamma_i = \left[ \begin{array}{cc} 0 & \cdots & 0 \\ u_i & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \end{array} \right]_{i-1 \text{ terms}} \ \ \ \ \left[ \begin{array}{c} 0 \\ \vdots \\ \phi_i' \end{array} \right]_{n-i \text{ terms}}$$

Then

$$x_1 = x_0 + \Phi' e = [ \gamma_1 w \cdots \gamma_n w ]'.$$

where

$$w = [ v_1 \cdots v_n e ]'.$$

Clearly $E\{ w \} = 0$ and

$$C := E\{ w w' \} = \begin{bmatrix} \phi^2 I_n & 0 \\ 0 & \mu^2 I_m \end{bmatrix}.$$ 

It is readily verified that

$$\gamma_i C \gamma_j' = \begin{cases} \phi^2 u_i^2 + \mu^2 \phi_i' \phi_i, & i = j \\ \mu^2 \phi_i' \phi_j, & i \neq j. \end{cases}$$

Consequently, observe in (2.29) that

$$r_{ij}(u_i^2) = \sigma^{-2} \gamma_i C \gamma_j'.$$

In the following we denote the mathematical expectation with respect to $w$ by $E_w$, while the mathematical expectation with respect to $u$ is denoted as $E_u$. Since $u$ and $w$ are independent random variables, it follows that

$$E\{ F_\sigma(x_1) \} = \sum_{i=1}^n E_u [ E_w \{ f_\sigma(\gamma_i w) \} ]. \quad (A.1)$$
Now

\[
E_w \{f_\sigma(\gamma_iw)\} = \frac{1}{\sqrt{2\pi}^{n+m}|C|} \times \int_{-\infty}^{+\infty} \exp \left\{ - \frac{w'(C^{-1} + \gamma_i'\sigma^{-2}\gamma_i)w}{2} \right\} dw.
\]

\[
= \{|C||C^{-1} + \gamma_i'\sigma^{-2}\gamma_i|\}^{-1/2}.
\]

Using the identity \(|I_{n+m} + C\gamma_i\sigma^{-2}\gamma_i| = 1 + \sigma^{-2}\gamma_iC\gamma_i'\) it follows that

\[
E_w \{f_\sigma(\gamma_iw)\} = \{1 + r_{ii}(u_i^2)\}^{-1/2}.
\]

Now taking expectation with respect to \(u\) we have

\[
E \{f_\sigma(\gamma_iw)\} = p\{1 + r_{ii}(1)\}^{-1/2} + (1 - p)\{1 + r_{ii}(0)\}^{-1/2}.
\] (A.2)

Combining (A.1) and (A.2) we get (2.31).

Next we compute \(E \{F_\sigma^2(x_1)\}\). First note that

\[
F_\sigma^2(x_1) = \left\{ \sum_{i=1}^{n} f_\sigma(\gamma_iw) \right\}^2
\]

\[
= \sum_{i=1}^{n} f_\sigma^2(\gamma_iw) + 2 \sum_{i=2}^{n} \sum_{j=1}^{i-1} f_\sigma(\gamma_iw)f_\sigma(\gamma_jw)
\]

A calculation similar as (A.2) gives

\[
E \{f_\sigma^2(\gamma_iw)\} = p\{1 + 2r_{ii}(1)\}^{-1/2} + (1 - p)\{1 + 2r_{ii}(0)\}^{-1/2}.
\] (A.3)

while for \(i \neq j\) we have

\[
E_w \{f_\sigma(\gamma_iw)f_\sigma(\gamma_jw)\} = \frac{1}{\sqrt{2\pi}^{n+m}|C|} \times \int_{-\infty}^{+\infty} \exp \left\{ - \frac{w'(C^{-1} + \gamma_i'\sigma^{-2}\gamma_i + \gamma_j'\sigma^{-2}\gamma_j)w}{2} \right\} dw
\]

\[
= \{|C||C^{-1} + \gamma_i'\sigma^{-2}\gamma_i + \gamma_j'\sigma^{-2}\gamma_j|\}^{-1/2}
\]

\[
= I_2 + \sigma^{-2} \begin{bmatrix} \gamma_i \\ \gamma_j \end{bmatrix} C \begin{bmatrix} \gamma_i' & \gamma_j' \end{bmatrix}^{-1/2}
\]

\[
= \left( \{1 + r_{ii}(u_i^2)\}\{1 + r_{jj}(u_j^2)\} - r_{ij}^2 \right)^{-1/2}.
\]
Consequently, using (2.30) and taking expectation with respect to \( u_i \) and \( u_j \) we get

\[
\mathbb{E}\{f_\sigma(\gamma_i w)f_\sigma(\gamma_j w)\} = p^2q_{ij}(1,1) + (1-p)^2q_{ij}(0,0) = p(1-p)\{q_{ij}(0,1) + g_{ij}(1,0)\}.
\]

(A.4)

Combining (A.3), (A.3) and (A.4) we get (2.32).

A.2 Proof of Lemma 8

In this appendix, we express \( X \) in terms of its real and imaginary parts: \( X = X_r + iX_i \). Let \( \ell \) denote the cost function in (3.38):

\[
\ell(X_r, X_i) = F_\sigma(X) - \lambda||Y - \Phi X||_F^2.
\]

In the following, we compute the derivatives of \( \ell \) with respect to \( X_r \) and \( X_i \). We define the matrix \( \frac{\partial P(X)}{\partial X_r} \) with elements

\[
\left[ \frac{\partial P(X)}{\partial X_r} \right]_{ij} = \frac{\partial P(X)}{\partial X_r[i, j]},
\]

with \( X_r[i, j] \) being the element of \( X_r \) at its \( i \)th row and \( j \)th column. Now denoting

\[
\ell_1(X_r, X_i) := ||Y - \Phi X||_F^2 = \text{Tr}[(Y - \Phi X)^*(Y - \Phi X)],
\]

it can be verified that

\[
\frac{\partial \ell_1(X_r, X_i)}{\partial X_r} = -2\text{Re}[\Phi^*(Y - \Phi X)],
\]

\[
\frac{\partial \ell_1(X_r, X_i)}{\partial X_i} = -2\text{Im}[\Phi^*(Y - \Phi X)].
\]

where \( \text{Re}(Z) \) and \( \text{Im}(Z) \) denote the real and imaginary parts of \( Z \), respectively. In addition, it follows that

\[
\frac{\partial F_\sigma(X)}{\partial X_r} = -\frac{1}{\sigma^2}W(X)X_r, \quad \frac{\partial F_\sigma(X)}{\partial X_i} = -\frac{1}{\sigma^2}W(X)X_i,
\]

where \( W(x) \) is defined in (3.40). Consequently,

\[
G(X) := \frac{\partial \ell(X_r, X_i)}{\partial X_r} + i\frac{\partial \ell(X_r, X_i)}{\partial X_i} = -\frac{1}{\sigma^2}W(X)X + 2\lambda \Phi^*(Y - \Phi X),
\]

which must vanish when evaluated at \( X = X_* \). Hence

\[
X_* = 2\lambda \left[ W(X_*)/\sigma^2 + 2\lambda \Phi^*\Phi \right]^{-1} \Phi^*Y = \zeta(X_*).
\]
Next, we show that $\zeta(X) - X$ gives a ascent direction for the cost function $\ell(X_r, X_i)$. For this, it is sufficient to show that

$$
\text{Tr} \left[ \text{Re}\{G(X)\}'\text{Re}\{\zeta(X) - X\} \\
+ \text{Im}\{G(X)\}'\text{Im}\{\zeta(X) - X\} \right] \\
= \text{Tr} \left( \text{Re}\{G^*(X)[\zeta(X) - X]\} \right) \geq 0.
$$

Note that the left hand side of the above inequality is the inner product between the gradient of $\ell$ at $X$ and

$$
\text{vec}[\text{Re}\{\zeta(X) - X\} \quad \text{Im}\{\zeta(X) - X\}].
$$

Now, note that

$$
\zeta(X) - X = \left[ W(X)/\sigma^2 + 2\lambda\Phi^*\Phi \right]^{-1} G(X).
$$

Since, $W(X)/\sigma^2 + 2\lambda\Phi^*\Phi$ is a positive definite matrix, so is

$$
[\zeta(X) - X]^*G(X) = G^*(X)[W(X)/\sigma^2 + 2\lambda\Phi^*\Phi]^{-1} G(X).
$$

Consequently, $\text{Tr}\{[\zeta(X) - X]^*G(X)\}$ is a real-valued positive quantity.
Bibliography


