OPTIMIZING ℓ_1 LOSS REGULARIZER AND ITS APPLICATION TO EEG INVERSE PROBLEM

by

KIRAN KUMAR MAINALI

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2020

OPTIMIZING ℓ_1 LOSS REGULARIZER AND ITS APPLICATION TO EEG INVERSE PROBLEM

The members of the Committee approve the doctoral dissertation of KIRAN KUMAR MAINALI

Dr. Ren-Cang Li ______________________

Dr. Li Wang ______________________

(Supervising Professors)

Dr. Jianzhong Su ______________________

Dr. Hristo V. Kojouharov ______________________

Dean of the Graduate School ______________________

To my parents, Kedar and Sabitri Mainali.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisors Dr. Ren-Cang Li and Dr. Li Wang. This journey of my research career at the University of Texas at Arlington (UTA) would not have been possible without their continuous support, patience, and encouragement. I can't imagine coming to this point in my research without the benefit of their immense knowledge and intellectual curiosity. They have helped me grow personally, academically, research-wise, and professionally. I greatly appreciate their recommendations and guidance in applying for fellowships, internships, workshops, summer schools, and conferences which helped me grow in my academic and professional career.

I am very thankful to my dissertation committee members Dr. Hristo V. Kojouharov, and Dr. Jianzhong Su for their encouragement and positive feedback. My professors and advisors were always there to support me with their insightful comments, recommendations, and constructive feedback. Their guidance and mentorship have benefitted me in every step. I am forever grateful.

I am very thankful to Dr. Li for his guidance to apply for the prestigious National Science Foundation Mathematical Sciences Graduate Internship Program (NSF-MSGI). The summer internship at Lawrence Berkeley National Laboratory (LBNL) in 2018 through NSF-MSGI was an incredible learning experience. I found wonderful friends and mentors at LBNL. This internship and the relationships I

made contributed greatly to learning the skills of high-performance computing and numerical linear algebra.

I am very happy to be a part of our data science research group at the UTA math department where I learned many techniques in data science research. Our weekly meetings and discussions helped me grasp the ideas of complex topics easily. I am very thankful to Saul Covarrubias for his many technical bits of help in my research. I learned a lot about the computational and research skills from you. I am very thankful to Faezeh Soleimani for her academic help and wonderful friendship.

I am very thankful to all of my friends at the UTA math department for their company, suggestions, and help. The academic meetings, involvement in graduate student chapter activities, days of preparing for prelim examinations, and the informal gatherings are memorable. I will always remember the group dinners after the departmental exams. I am thankful to my wonderful office mates Hrishabh, Saul, and Saber at PKH 430. You all have made this experience very enjoyable.

The outreach activities through the Society for Industrial and Applied Mathematics (SIAM) UTA chapter as a treasurer helped the growth of my academic career. I am very thankful to all the members of the chapter especially Mayowa and Imelda for their wonderful company. I was very pleased to give economic strength to SIAM through several midterm reviews. I learned to lead and organize meetings and share ideas in formal and informal talks which made me confident.

I would like to express my deep gratitude to all administrative staff of the department for their continuous support in my academic career at UTA. My sincere appreciation goes to the office of graduate studies for providing a dissertation fellowship for the summer of 2020, my last semester at UTA. The dissertation fellowship helped me finish my work on time. I am very thankful to all of my incredible students at

UTA, whose feedback and curiosity have made me a strong and committed professor of mathematics for the days to come.

Last but not least, I'd like to express my most profound gratitude to my family for always supporting my studies. This journey would have been impossible without the unconditional love, continuous support, and encouragement of my family. My mom and dad have sacrificed many sleepless nights to bring me here today. I remember all the struggles that you had to bring me here. You could not go to school for formal education but always made the education of your children a top priority. You are the one who dreamed for my Ph.D. I chased your dream with passion and finally accomplished it. I can't express my feelings for you in words, thank you so much for all of your hard work. I love you!

I am thankful to my brother Badri and his wife Indira who always guided me on the right path. Thank you so much for being such an incredible brother. I am humbled to my elder sister Sushila for her love, support, and guidance. I am here because of your determination. You always choose the right path and the rest of us follow your steps. My sincere credit goes to you for motivating me to come to the USA for my Ph.D. studies. Without you, it was not possible. Thank you so much for being such a nice sister and my guardian. I am thankful to my younger sister Kalpana for her love and care. You always choose what I like the most. My sincere respect goes to my brother-in-law Dinesh for his support and being like a friend to me. My nephew Sudin and niece Sampada are the apple of my eyes. Thank you for bringing a smile to my face even if I was tired when returning from school.

I am very much thankful to my uncle Pramod Kharel and his family for unconditionally supporting me and my family. You were there for all the difficult times that we had. Without your support, I wouldn't have been able to accomplish

all these. You are my role model. I appreciate what you have done for encouraging me to strive towards the goal.

I would like to thank my wife Aradhana, an amazing wife and true friend for believing in my potential. You are a true blessing in my life. I am indebted to you for all your care, your understanding, encouragement, and love. Your true belief in my potential always pushed me to move on. You were there when I needed support and someone to lean on. Your incredible presence in my life always motivates me to be better. I am truly thankful for your presence and care. I love you!

I am thankful to the family of my wife, Dilip, Sushma, and Sudish Gyanwali for always keeping me in your heart and prayer. Your blessings, support, and care always encouraged me to move on to the right path. Moreover, my sincere gratitude goes to Kaka Mrugendra Mehta and Kaki Jennifer Timms Mehta for keeping me in their prayers all the time. I am thankful to God for having such wonderful guardian in my life. Thank you, Kaki, for working hard on editing and polishing my dissertation. I can't express my feelings for both of you in words. I love you both!

Lastly, I would like to thank all of my teachers, well-wishers, families, and friends whom I neglected to mention. Thank you, everyone, who made my journey more beautiful and easier.

July 15, 2020

ABSTRACT

OPTIMIZING ℓ_1 LOSS REGULARIZER AND ITS APPLICATION TO EEG INVERSE PROBLEM

KIRAN KUMAR MAINALI, Ph.D.

The University of Texas at Arlington, 2020

Supervising Professors: Ren-Cang Li and Li Wang

Sparse reconstruction occurs frequently in science and engineering and real-world applications, including statistics, finance, imaging, biological system, compressed sensing, and, today more than ever, machine learning and data science in general. Mathematically, they are often modeled as ℓ_1-minimization problems. There are a number of existing numerical methods that can efficiently solve such ℓ_1-minimization problems, such as Alternating Direction Methods of Multipliers (ADMM), Fast Iterative Shrinkage Thresholding Algorithm (FISTA), and Homotopy algorithm.

In this dissertation, we will introduce a special type of ℓ_1-minimization problem called the Sylvester Least Absolute Shrinkage and Selection Operator (SLASSO) problem. In theory, an SLASSO problem can be converted to a standard LASSO problem and then solved by any existing numerical method, but the converted LASSO problem is too large scale to be practical even if the SLASSO problem is modest. The first contribution of this dissertation is a novel method to solve an SLASSO problem without conversion, making it practical to solve a fairly large sized SLASSO problem.

Our second contribution is a new structured Electroencephalogram (EEG)/Magnetoencephalogram (MEG) Source Imaging (ESI) model that groups the time-varying signals of a similar structure and uses the mixed norm estimation for accurate results. The model is then solved alternatingly. Numerical simulations compare favorably with the state-of-the-art ESI methods, demonstrating the effectiveness of the model and efficient numerical treatment.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS v

ABSTRACT ix

LIST OF ILLUSTRATIONS xiv

LIST OF ALGORITHMS xvii

LIST OF TABLES xviii

LIST OF ABBREVIATIONS xx

LIST OF SYMBOLS xxii

Chapter Page

1. INTRODUCTION 1
   - 1.1 Motivations 1
   - 1.2 Inverse Problems 2
   - 1.3 Major Contributions and Organization of the Dissertation 8
2. REVIEW OF NUMERICAL OPTIMIZATION TECHNIQUES 12
   - 2.1 Iterative Shrinkage Thresholding Algorithm (ISTA) 12
   - 2.2 Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) 19
   - 2.3 Alternating Direction Method of Multipliers (ADMM) 21
     - 2.3.1 Dual ascent 21
     - 2.3.2 ADMM Algorithm 23
     - 2.3.3 Solving LASSO problem using ADMM 24
   - 2.4 ℓ_1-Homotopy Algorithm 26
   - 2.5 Iterative Reweighting via Homotopy 31
   - 2.6 Reweighted ℓ_1-minimization for sparsity enhancement 35

3. APPLICATION OF ℓ_1-MINIMIZATION TO EEG BRAIN SOURCE LOCALIZATION PROBLEM. 39
3.1 Introduction 39
3.2 The EEG Inverse Problem 41
3.3 Numerical Results 44
3.3.1 Experiment setup 44
3.3.2 Numerical Experiments 45
4. SYLVESTER LASSO AND ITS APPLICATION TO EEG INVERSE PROBLEM 53
4.1 Introduction 53
4.2 Application of Sylvester LASSO to EEG Inverse Problem 54
4.3 Pre-processing 55
4.4 Algorithms for Sylvester type LASSO Problem 57
4.4.1 ADMM 57
4.4.2 FISTA 60
4.5 Iterative Reweighting for Sylvester FISTA 61
4.6 Numerical Experiments 63
4.6.1 Experimental setting 63
4.6.2 Numerical Results 64
5. ESI MODEL CAPTURING THE SOURCE ACTIVATION PATTERN 78
5.1 Introduction 78
5.2 The Principle of Maximum Entropy 79
5.3 Model Formulation 81
5.4 Numerical Algorithm 87
5.5 Numerical Experiments 95
5.5.1 Experiment setting 96

5.5.2 Description of Data and Parameters 98

5.5.3 Model Validation Metrics 100

5.5.4 Convergence of the Proposed Model 102

5.5.5 Simulation Results 104

6. CONCLUSIONS AND FUTURE WORK 114

6.1 Summary 114

6.2 Future Work 116

REFERENCES 119

BIOGRAPHICAL STATEMENT 130

LIST OF ILLUSTRATIONS

Figure | Page

1.1 Sparse signal recovery is the problem of estimating unknown signal of dimension n based on m noisy observations, when $m \ll n$. 2

1.2 For the randomly generated data with 80 responses ($m = 80$) and 100 predictors ($n = 100$), the solution generated by the ridge model provides many coefficients estimations very close to zero whereas coefficient estimation by LASSO are either non-zero or explicitly zero. 5

1.3 Geometric illustration of the two-dimensional case of estimation for the LASSO (left) and the ridge regression (right). This illustration for ℓ_1- and ℓ_2-penalty is inspired by Figure 6.7 of [1] and Figure 10 of [2] and modified with text in the context of this dissertation. 7

3.1 Brain model with cortical surface represented by triangular mesh where each triangle represents the brain voxel. We consider that each current dipole is located at the center of the triangular mesh and orientation of the dipole is perpendicular to the cortical surface. 42

3.2 In EEG inverse problem, we set up the EEG electrodes on the scalp as shown in figure 1 from which we record the signal data X. With the help of recorded signal information X and lead field matrix L, our task is to determine the activated source location S as shown in 2. 43

3.3 EEG channel layout of ICBM 152 - Neuroscan Cap 128 edited in (a) and (b) from two different sides of the head model. 45

3.4 Behavior in convergence of the objective value of the problem (3.3) by different benchmark algorithms for $C = 0.1$. 47

3.5 EEG signals from 22 out of 108 channels in the first 0.9 milliseconds. 49

3.6 Source recoveries by benchmark ℓ_1-optimization algorithms against ground truth. 51

4.1 Decay of the objective function values in successive iterations when $\lambda = 1$. In ADMM, the objective function value at the k-th iteration is referred to $f(X^{(k)}) + g(Z^{(k)})$ whereas $f(X^{(k)}) + g(X^{(k)})$ in ISTA and FISTA. 65

4.2 Convergence of the three algorithms for solving the Sylvester type LASSO problem (4.13). 66

4.3 Reconstruction error by different weighting schemes used in Algorithm 4.3. 69

4.4 Sparsity plot of the solutions by Algorithm 4.3 in 4 reweighting steps compared with the initial solution of FISTA and ground truth. 73

4.5 Reconstruction error by the proposed algorithm at different noise levels. 75

4.6 Sparsity recovery by the proposed algorithm at different noise levels. . 76

5.1 Entropy of two-class set as a function of $p(+)$. Figure and example are taken from [3]. 80

5.2 The relation of movements of certain body parts and corresponding activation regions in the brain cortex. Both figures and their descriptions are taken from Brain Connection blog available at `https://brainconnection.brainhq.com/2013/03/05/the-anatomy-of-movement/`. 82

5.3 (a), (b), and (c) show the estimations of the source amplitudes of EEG inverse problem (5.3) by ℓ_2-, ℓ_1-, and $\ell_{2,1}-$norm penalty, respectively. The non-zero coefficients are shown in white. While ℓ_2-norm penalty yields only non-zero coefficients, $\ell_{2,1}$-norm penalty promotes non-zero coefficients with a row structure (only a few sources have non-zero amplitude over the entire time interval of interest). This illustration is inspired by Figure 1 of [4]. 86
5.4 Sparsity comparison of different ESI methods. 98
5.5 EEG channel layout of BioSemi Neuroscan cap with 64 channels (in front and back views). 99
5.6 Convergence of the proposed ESI method on clean and noisy data. . . . 103
5.7 Evolution of the objective function values evaluated at each variable updates in successive iterations. 104
5.8 Clean data vs. noisy data at different noise levels. 106
5.9 ROC curves by different ESI models. 108
5.10 Figure (a) displays the cortical region and source activation based on synthetic data having 350 brain voxels. Figure (b) shows the corresponding projected source in the high resolution cortical region having 15002 triangular vertices. 110
5.11 Source recovery plots by different ESI methods with $\mathrm{SNR}_C = 10$ dB and $\mathrm{SNR}_S = \infty$. 110
5.12 Source recovery plots by different ESI methods with $\mathrm{SNR}_C = 30$ dB and $\mathrm{SNR}_S = 30$ dB. 111
5.13 Source recovery plots by different ESI methods with $\mathrm{SNR}_C = 20$ dB and $\mathrm{SNR}_S = 30$ dB. 112

LIST OF ALGORITHMS

Algorithm Page

2.1 ISTA with constant stepsize 17

2.2 ISTA with backtracking 18

2.3 FISTA with constant stepsize 19

2.4 FISTA with backtracking 20

2.5 ADMM Algorithm for solving LASSO Problem. 26

2.6 ℓ_1-Homotopy Algorithm 30

2.7 Iterative Reweighting via Homotopy 34

2.8 Iterative Reweighting 36

4.1 ADMM Algorithm for Sylvester type LASSO Problem (4.1) 60

4.2 FISTA with backtracking for Sylvester type LASSO. 61

4.3 Iterative Reweighting for Sylvester type LASSO 63

5.1 Projected Gradient Descent for R-subproblem. 92

5.2 The ADMM framework for the proposed ESI model 95

LIST OF TABLES

Table Page

3.1 Results of source reconstruction by benchmark algorithms with scaling factor $C = 0.005$ for λ. 48

3.2 Results of source reconstruction by benchmark algorithms in different noise levels with scaling factor $C = 0.0001$ for λ. 49

4.1 Recovery results for problem (4.3) with $\lambda = 1$. 67

4.2 Recovery results for (4.3) with $\lambda = 1$ and two different levels of noise in measurements. 67

4.3 Recovery results for the problem (4.3) with different noise levels. We use $\lambda = 1$ for ADMM, and $\lambda = 0.1 \times \lambda_{\max}$ for ISTA and FISTA. 68

4.4 Results of iterative reweighting via six different weight schemes with two reweighting steps for each weight scheme. 70

4.5 Recovery results by Algorithm 4.3 for noiseless data with initial weight matrix having all diagonal entries one. 71

4.6 Results of recovery by Algorithm 4.3 for the Sylvester type LASSO problem with synthetic data at three different noise levels. 74

5.1 Quality of source reconstructions in different error metrics for clean data. The parameter values used in the proposed model are $\alpha = 0.01, \gamma_1 = 0.001, \gamma_2 = \gamma_3 = 0.001, \lambda = 0.0001$, and $\sigma = 0.02$. 105

5.2 Quality of source reconstructions in different error metrics for the data with different noise levels. The parameter values used in the proposed model are as follows: for SNR = 10 dB, $\alpha = 0.01, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.01, \lambda = 0.01$, and $\sigma = 0.1$; for SNR = 20 dB, $\alpha = 0.01, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.6, \lambda = 0.01$, and $\sigma = 0.2$; for SNR = 30 dB, $\alpha = 0.01, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.3, \lambda = 0.001$, and $\sigma = 0.8$. 107

5.3 Quality of source reconstructions by different ESI algorithms in different error metrics for synthetic data with noise in channels 30 dB, noise in sources 30 dB, and 20 dB. The parameter values in the proposed model: for $\mathrm{SNR}_S = 20$ dB, $\alpha = 0.001, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.01, \lambda = 0.1$, and $\sigma = 0.01$, and for $\mathrm{SNR}_S = 30$ dB, $\alpha = 0.01, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.01, \lambda = 0.1$, and $\sigma = 0.01$. 108

5.4 Quality of source reconstructions by different ESI algorithms under different error metrics for synthetic data with noise in channels 20 dB, noise in source 30 dB, and 20 dB. The parameter values in the proposed model: for $\mathrm{SNR}_S = 20$ dB and 30 dB, $\alpha = 0.5, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.01, \lambda = 0.1$, and $\sigma = 0.01$. 109

LIST OF ABBREVIATIONS

ADMM	Alternating Direction Method of Multipliers
AUC	Area Under ROC Curve
DALY	Disability Adjusted Life Year
dB	Decibels
DF	Data Fitting
EEG	Electroencephalogram
ESI	Electroencephalography Source Imaging
FISTA	Fast Iterative Shrinkage - Thresholding Algorithm
FN	False Negative
FP	False Positive
FPR	False Positive Rate
ISTA	Iterative Shrinkage - Thresholding Algorithm
KKT	Karush-Kuhn-Tucker
LASSO	Least Absolute Shrinkage and Selection Operator

MCE	Minimum Current Estimates
MNE	Minimum Norm Estimates
MSE	Mean Squared Error
MxNE	Mixed Norm Estimates
OLS	Ordinary Least Square
PGD	Projected Gradient Descent
RE	Reconstruction Error
ROC	Receiver Operating Characteristic
RT	Total CPU time
SNR	Signal to Noise Ratio
SU	Spectral Unmixing
TN	True Negative
TP	True Positive
TPR	True Positive Rate
TV	Total Variation

LIST OF SYMBOLS

$\mathcal{C}^{1,1}$	Class of continuously differentiable function f with Lipschitz continuous gradient
∇f	Gradient of the function f
$\mathscr{D}(f)$	Domain of the function f
$\|x\|_1$	1-norm of vector x
$\|x\|_2$	Euclidean norm of vector x
$\|x\|_\infty$	Infinity norm of vector x
$\|X\|_F$	Frobenius norm of matrix X
$\|X\|_{1,1}$	$L_{p,q}$ norm of matrix X with $(p=1,\ q=1)$
$\|X\|_{2,1}$	$L_{p,q}$ norm of matrix X with $(p=2,\ q=1)$
$\mathbf{prox}_{\lambda,f}(x)$	Proximal operator of function f corresponding to parameter λ at x
$\mathcal{S}_\alpha(x)$	Soft-thresholding operator corrsponding to parameter α at x
$(x)_+$	Positive part of x, i.e., $(x)_+ = \max(x,0)$
$\text{sign}(x)$	sign function acting on x
$L(f)$	Lipschitz constant for functional f
$\lambda_{\max}(A)$	Largest eigenvalue of a square matrix A
ϵ^{abs}	Absolute tolerance

ϵ^{rel}	Relative tolerance
Γ	Support set of the solution
Γ^c	Complement of the support set Γ
$\|\Gamma\|$	The cardinality of the set Γ
$\partial f(x)$	Subdifferential of function f at x
A^T	Transpose of the matrix A
$\Pi_{\mathcal{R}}(x)$	Projection of vector x onto set $\mathcal{R}$
$x^{(k)}$	Value of x in k-th iteration step

CHAPTER 1

INTRODUCTION

1.1 Motivations

High dimensional data are ubiquitous in the modern era of science and technology. This data is generated in a very large quantities from multiple sources. Estimating the high dimensional data based on incomplete linear observations has been discussed broadly in the compressed sensing community [5–8]. Acquisition of compressible high dimensional data from minimal measurements has a wide variety of applications in applied mathematics, computer science, and electrical engineering such as magnetic resonance imaging (MRI) [9], image processing [10], signal processing [11], imaging technique [12], and so forth. Mathematically speaking, when the equations are linear, one would like to determine the object $x \in \mathbb{R}^n$ from the noisy observations $b = Ax + N_\epsilon$, where A is an $m \times n$ measurement matrix with fewer rows than columns; i.e., $m \ll n$ and N_ϵ is noise. If A has rank m, such a problem has $(n - m)$ number of free variables, thus by the fundamental principle of linear algebra, problems having such attribute have infinitely many solutions and it is impossible to identify which of these available solutions is correct without having some additional information about the data. Figure 1.1 helps to describe this situation in a broad overview. However, in many practical applications, the data we are interested in recovering are compressible (sparse). For example, an image of millions of pixels is very sparse over the wavelet basis, namely, a small fraction of wavelet coefficients that are enough to recover images [13].

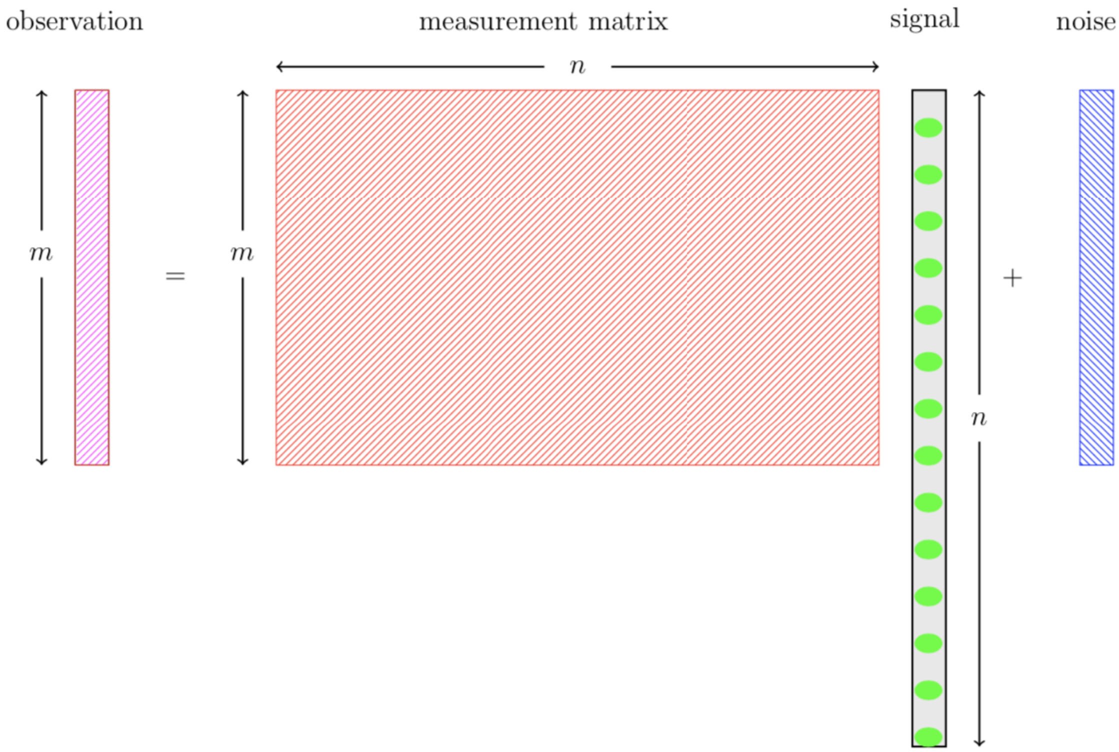


Figure 1.1: Sparse signal recovery is the problem of estimating unknown signal of dimension n based on m noisy observations, when $m \ll n$.

1.2 Inverse Problems

The data of practical interests are compressible over certain basis. This feature of data opens the room for recovering the signals of interest accurately from incomplete linear measurements in compressive sensing. However, even if the signal of interest is sparse, it is a non trivial task to recover the signal as we do not know the locations of the non-zero elements in the recovered signal a priori. Several algorithms are proposed to solve this problem, for example, ℓ_1-magic [14], basis pursuit denoising [15, 16], ℓ_1-homotopy [17], log-barrier method [18], LASSO (Least Absolute Shrinkage and Selection Operator) [1, 19–21] , and so forth, namely the ℓ_1-minimization algorithm. The term inverse problems refer to the general framework used to convert observed measurements into information about the object of interest. For example, given tomographic measurements of an object, we might wish to know about the internal composition and structure [22, 23]. Having the ability to solve the inverse problems is useful as it provides information about the physical quantities that we are unable to observe directly. There are several applications of the inverse problem in physical sciences. The work of Bal [24], Hansen [25], and the references therein illustrate several areas where the inverse problem arises in applications and their mathematical formulations. In this dissertation, we focus on an inverse problem that has to deal with a system of linear algebraic equations. To proceed further, we define two key terminologies that will be used repetitively throughout this dissertation.

Definition 1.2.1 (convex set)**.** A set C is *convex* [18] if the line segment between any two points in C lies in C, i.e., if for any x_1, $x_2 \in C$ and any θ with $0 \leq \theta \leq 1$, we have

$$\theta x_1 + (1-\theta)x_2 \in C.$$

Definition 1.2.2 (convex function)**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is *convex* if the domain of f, $\mathscr{D}(f)$ is a convex set and if for all x, $y \in \mathscr{D}(f)$, and θ with $0 \leq \theta \leq 1$, we have

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y).$$

Definition 1.2.3 (sparsity of a vector)**.** A vector $x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$ is said to be *sparse* if most of its entries are zero. In particular, x is said to be *k-sparse*, if k out of n entries of x are zero. In the case where only a few entries of x are large (significant) and the rest of the entries are zero or very small, then the vector x is called weakly sparse or compressible.

Also, we consider the definition of well-posed problem by Hadamard [26];

Definition 1.2.4 (well-posed problem)**.** A mathematical problem is *well-posed* if

1. a solution exists;

2. the solution is unique;

3. the solution depends continuously on data.

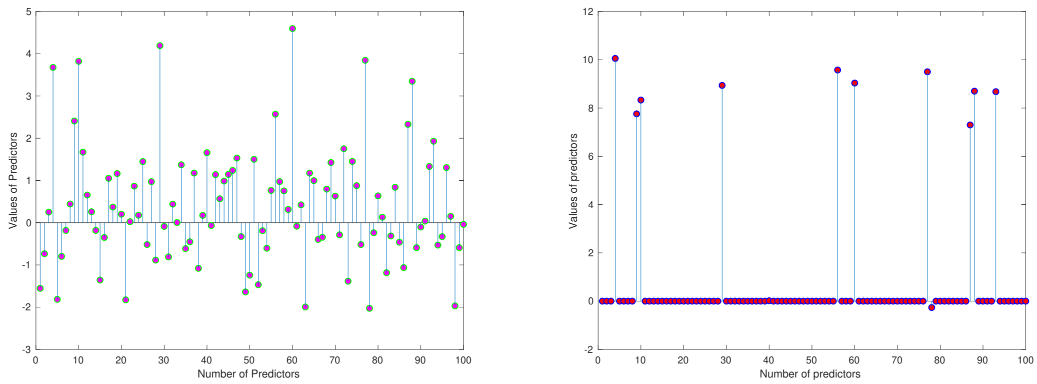
If at least one of the conditions above does not hold, the problem is called *ill-posed*.

Suppose $b \in \mathbb{R}^m$ is a measured data and $A \in \mathbb{R}^{m\times n}$ $(m \ll n)$ a matrix of measurement or matrix associated with measurement operator and we are interested in recovering the data $x \in \mathbb{R}^n$ which is related by the following linear relation:

$$b = Ax + N_\epsilon, \tag{1.1}$$

where N_ϵ is a vector associated with measurement error. The problem associated with obtaining the solution of the above underdetermined linear system (1.1) as pictured in Figure 1.1 is a highly ill-posed inverse problem. As the underdetermined system has infinitely many solutions, finding the right solution from the group of infinitely many solutions is very challenging without any prior assumption to the solution. But, in many practical applications, solutions that we are looking for are sparse. For example, in mathematical biology where sparsity techniques are needed to map DNA breakpoints in cancer genomes or select the most important genes from high dimensional gene sequence [27, 28]. Similarly, in the literature of the simple regression model, thousands of predictors are involved to get limited response variables of particular interests. Out of the huge number of predictors, only a few of them have important roles in building an efficient regression model. So appropriate variable selection procedures are used to obtain the sparse solution [1, 20]. A similar situation occurs in predicting the price of the stocks, the profit of the company, risk factors for investment, etc. from significant predictors from the thousands of interactions in market trends in finance and economics's data [29–31].

To find feasible and meaningful solutions out of infinitely many possible solutions, regularization plays an important role. In variable selection problems in linear



(a) Coefficients estimated by ridge model (1.2).

(b) Coefficients estimated by LASSO model (1.3).

Figure 1.2: For the randomly generated data with 80 responses ($m = 80$) and 100 predictors ($n = 100$), the solution generated by the ridge model provides many coefficients estimations very close to zero whereas coefficient estimation by LASSO are either non-zero or explicitly zero.

regression, the ℓ_2-penalty was introduced [32] and it is called ridge regression. The ridge regression model can be expressed as

$$\hat{x}_{\text{ridge}} = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_2^2, \tag{1.2}$$

where $\|x\|_2^2 = \sum_{i=1}^n x_i^2$ is the ℓ_2-norm of a vector $x \in \mathbb{R}^n$. The objective function of the ridge regression is strictly convex and (1.2) has a unique minimizer. The purpose of the ridge regression is to estimate the predictors by making the residual $r = Ax - b$ small. The second term $\lambda\|x\|_2^2$ is called a shrinkage penalty which is small when $x_1, x_2, \cdots, x_n$ are close to zero and which has the effect of shrinking the estimates of x_i towards zero. Moreover, $\lambda > 0$ is a tuning parameter which controls how fast the coefficients are shrunk towards zero, i.e., the larger the value of λ the greater the effect of shrinkage.

The solution of the ridge regression model turns to select all the predictors into the model and that will cause difficulties in interpreting and analyzing the model. The major issues in high dimensional data are to choose the smallest subset of predictors which can simplify the original model while fulfilling the certain statistical criteria, e.g., the smallest possible Mean Squared Error (MSE), the largest adjusted R^2, etc. Also, the model with fewer predictors makes the interpretation of the model easier than the model with a full set of active variables. To find the best subset selection requires solving 2^n sub-models which is not feasible when n (the number of predictors) is very large. This is the reason why statisticians like to create the best variable selection model. There are several stepwise subset selection methods such as stepwise forward selection, stepwise backward elimination, and so forth [1, 20]. To overcome the computational burden in multistep variable selection methods, R. Tibshirani developed a one-step variable selection method in 1996 called the Least Absolute Shrinkage and Selection Operator (LASSO) [19]. The LASSO model has several applications in physical and biological sciences (see e.g. [27–31]). The LASSO model is based on ℓ_1-penalty which can be described as

$$\hat{x}_{\text{lasso}} = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1, \tag{1.3}$$

where $\|x\|_1 = \sum_{i=1}^n |x_i|$ is the ℓ_1-norm of a vector $x \in \mathbb{R}^n$. The LASSO model is strictly convex and shrinks the coefficient estimates towards zero as ridge regression. However, in the case of LASSO, the ℓ_1-penalty has the effect of forcing some of the coefficients estimates explicitly equal to zero when the tuning parameter $\lambda > 0$ is sufficiently large. In this regard, LASSO promotes the sparsity in coefficient estimation and performs the best variable selection compared to the ridge regression. As a consequence, models generated by LASSO are much easier to interpret. Figure 1.2 describes the nature of the solution between the ridge regression and the LASSO model.

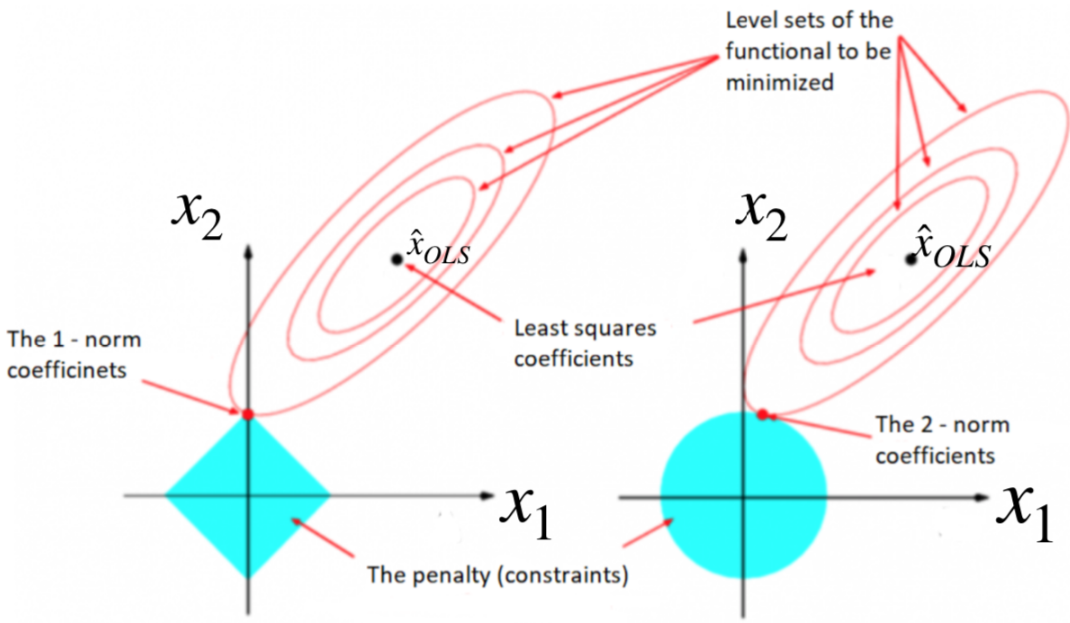


Figure 1.3: Geometric illustration of the two-dimensional case of estimation for the LASSO (left) and the ridge regression (right). This illustration for ℓ_1- and ℓ_2-penalty is inspired by Figure 6.7 of [1] and Figure 10 of [2] and modified with text in the context of this dissertation.

The geometric intuition behind the nature in solution between LASSO and the ridge regression is presented in Figure 1.3, where we consider the estimation of two dimensional vector $x = (x_1, x_2) \in \mathbb{R}^2$ for visualization purposes. The shaded areas are the feasible sets for LASSO and the ridge regression. The ℓ_1-constraint creates the convex diamond, whereas ℓ_2-constraint creates the shape of a disk. The $\hat{x}_{OLS}$ in the center of the contours represents the solution of an ordinary least square (OLS) described by

$$\hat{x}_{OLS} = \underset{x \in \mathbb{R}^n}{\arg\min} \frac{1}{2}\|Ax - b\|_2^2.$$

The elliptic contours are contours of squares error with the OLS estimator in the center. Note that the ridge regression has a circular constraint and the intersection of level curves and the feasible set do not intersect on an axis. So, the ridge estimate can be shrunk close to zero, but not exactly equal to zero. On the other hand, the LASSO constraint has a corner at each axis, and so the level curves often intersect the feasible region at an axis. In Figure 1.3, the intersection occurs at $x_1 = 0$, thus x_2 is chosen as the only relevant parameter in the model. The situation for a higher dimension is more complicated to visualize but it follows the same principle. A 3-dimensional case is depicted in [19, 21]. In this regard, LASSO is one of the better options for a sparse solution or a method of better subset selection that provides the results with better interpretability. In this dissertation, we will use ℓ_1-penalty heavily to build the new models to solve brain source imaging inverse problems.

1.3 Major Contributions and Organization of the Dissertation

In this dissertation, we focus on the ℓ_1-minimization problems that relate Electroencephalogram (EEG) Source Imaging (ESI) problems. We design a novel mathematical model that incorporates plausible neurophysiological assumptions to answer the challenges in ESI problems. We develop an efficient algorithm that solves the proposed ESI model efficiently that outperforms the popular methods designed to solve the problems in brain source imaging.

In Chapter 2, we discuss current state-of-the-art algorithms designed to solve the problem (1.3). We discuss the mathematical backgrounds for solving convex optimization problems in general and their solution procedures. We briefly discuss the popular ℓ_1-minimization algorithms such as Alternating Direction Methods of Multipliers (ADMM), Iterative Shrinkage and Thresholding Algorithm (ISTA), Fast Iterative Shrinkage and Thresholding Algorithm (FISTA), and ℓ_1-homotopy. We discuss their convergence properties and their performances in ℓ_1-minimization problems. Furthermore, we explore iterative reweighting techniques for sparsity enhancement and better solution reconstruction for the inverse problems.

Chapter 3, focuses on the application of ℓ_1-minimization in the EEG inverse problem for brain source reconstruction. We explore the classical ESI models and the use of ℓ_1-minimization techniques in brain source imaging. We briefly discuss the procedure for solving the basic ESI model which can be described as a matrix recovery problem formulated as

$$\arg\min_{S\in\mathbb{R}^{n\times k}} \|X - LS\|_F^2 + \lambda\|S\|_{1,1},$$

where $X \in \mathbb{R}^{m\times k}$, $L \in \mathbb{R}^{m\times n}$, and $S \in \mathbb{R}^{n\times k}$ are matrices whose context will be described in chapter 3 in detail. Here, $\|\cdot\|_F$ and $\|\cdot\|_{1,1}$ denote the Frobenius and $L_{p,q}$ with $(p = 1, q = 1)$ norm of a matrix:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2}$$

and

$$\|A\|_{1,1} = \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|$$

respectively, where $A \in \mathbb{R}^{m\times n}$. We analyze the results and visualize them in the context of source reconstruction as well as the performances of the different algorithms in the recovery process under different noise level.

In Chapter 4, we present our novel approach for solving a special type of ℓ_1-minimization problem called the Sylvester LASSO model and show its relevance to the ESI problem. The Sylvester LASSO is also a matrix recovery problem that can be described by the following convex optimization model

$$\arg\min_{X\in\mathbb{R}^{m\times n}} \|AX - B\|_F^2 + \|XC - D\|_F^2 + \|X\|_{1,1}$$

where $A \in \mathbb{R}^{q\times m}, B \in \mathbb{R}^{q\times n}, C \in \mathbb{R}^{n\times p}, D \in \mathbb{R}^{m\times p}$. Conversion of the problem from the Sylvester form to a regular LASSO form will increase the size of the problem drastically for which the benchmark ℓ_1-solvers cannot be employed because of huge memory requirement and computational complexity. We present our novel ideas to handle the large size data matrices by extracting their structures which allow solving the large scale problem on a personal computer. We present our numerical results and the performance of the algorithm in sparse source reconstruction in detail.

Chapter 5 focuses on solving a new ESI model by incorporating the group structure of the similar EEG signals together. Classical ESI methods often assume that the brain source activities at different time points are unrelated, which makes ESI analysis sensitive to noise. To effectively deal with noise while maintaining flexibility and continuity among brain activation patterns, we propose a new mathematical model that groups the time-varying signals of a similar structure and apply the mixed norm estimation for accurate results. We develop and discuss the solution procedure for solving the following convex optimization model

$$\begin{aligned}\arg\min_{S,C,R} h(S,C,R) = \|X - LS\|_F^2 + \lambda \sum_{i=1}^{N_t}\sum_{k=1}^{K}[r_{i,k}\|s_i - c_k\|^2 + \alpha r_{i,k}\log\, r_{i,k}] \\ + \gamma_1 \sum_{k=1}^{K} \|S\ \mathrm{diag}(r_k)\|_{2,1} + \gamma_2 \sum_{k=1}^{K} \|c_k\|_1 + \gamma_3 \sum_{i=1}^{N_t} \|s_i\|_1,\end{aligned}$$

where $X \in \mathbb{R}^{N_c\times N_t}, L \in \mathbb{R}^{N_c\times N_s}, S \in \mathbb{R}^{N_s\times N_t}, R \in \mathbb{R}^{N_t\times K}$, $C \in \mathbb{R}^{N_s\times K}$, and

$$\mathrm{diag}(r_k) = \begin{pmatrix} r_{1,k} & 0 & \cdots & 0 \\ 0 & r_{2,k} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_{N_t,k} \end{pmatrix} \in \mathbb{R}^{N_t\times N_t}$$

is a matrix that populates k-th column of matrix R in diagonal. Context of the model, algorithm, and numerical simulation results will be discussed in detal.

In chapter 6, we discuss the relevance and importance of our work on solving ℓ_1-optimization and ESI problems. We summarize the conclusions of the dissertation and the extension of our work in other possible areas of data science.

CHAPTER 2

REVIEW OF NUMERICAL OPTIMIZATION TECHNIQUES

In this chapter, we discuss numerical optimization techniques to solve the standard LASSO problem

$$\hat{x}_{\text{lasso}} = \arg\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 , \tag{2.1}$$

where $A \in \mathbb{R}^{m \times n}$ $(m \ll n)$, $b \in \mathbb{R}^{m \times 1}$, and $\lambda > 0$ is a regularization parameter. There are several numerical optimization techniques to solve problem (2.1). Among them, we will review current state-of-the-art ℓ_1-minimizing algorithms in this chapter. The ideas of solving ℓ_1-minimization problems discussed in this chapter will be frequently used in later chapters.

2.1 Iterative Shrinkage Thresholding Algorithm (ISTA)

We start this section by two important definitions.

Definition 2.1.1 (Proximal Operator [33, 34])**.** The *proximal operator* $\mathbf{prox}_{\lambda,f} : \mathbb{R}^n \to \mathbb{R}^n$ of a function f with parameter $\lambda > 0$ is defined as

$$\mathbf{prox}_{\lambda,f}(v) = \arg\min_x \left(f(x) + \frac{1}{2\lambda} \|x - v\|_2^2 \right).$$

Definition 2.1.2 (Shrinkage/Soft-thresholding Operator [35])**.** The *shrinkage operator* $\mathcal{S}_\alpha : \mathbb{R}^n \to \mathbb{R}^n$ with parameter $\alpha > 0$ is defined as

$$\mathcal{S}_\alpha(x)_i = \operatorname{sign}(x_i)(|x_i| - \alpha)_+ =: \operatorname{shrink}(x_i, \alpha) \text{ for } 1 \le i \le n,$$

where $(p)_+ = \max(p, 0)$ and

$$\text{sign}(x_i) = \begin{cases} 1 & \text{if } x_i > 0, \\ -1 & \text{if } x_i < 0, \\ 0 & \text{if } x_i = 0. \end{cases}$$

Definition 2.1.3 (subdifferential [34,36])**.** Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function. The *subdifferential* of f at x is a set, denoted by $\partial f(x)$, and defined as

$$\partial f(x) = \left\{ y \mid f(z) \geq f(x) + y^T(z - x) \text{ for all } z \in \mathscr{D}(f) \right\}.$$

Before moving further, we present a result in Theorem 2.1.1 that will be used in this dissertation multiple times.

Theorem 2.1.1. *Let $f(x) = \lambda\|x\|_1$ and $p > 0$. Then*

$$x^* = \mathbf{prox}_{p,f}(v) = \arg\min_{x \in \mathbb{R}^n} \left(\lambda\|x\|_1 + \frac{1}{2p} \|x - v\|_2^2 \right) \tag{2.2}$$

is given by

$$\begin{aligned} x_i^* &= \begin{cases} v_i - p\lambda & \textit{if } v_i > p\lambda, \\ 0 & \textit{if } |v_i| \leq p\lambda, \\ v_i + p\lambda & \textit{if } v_i < -p\lambda \end{cases} \\ &= \text{sign}(v_i)(|v_i| - p\lambda)_+ \\ &= \mathcal{S}_{p\lambda}(v)_i. \end{aligned} \tag{2.3}$$

Proof. Since the optimization problem (2.2) is convex and decoupled, it suffices to prove that x^* given by (2.3) satisfies the following KKT optimality condition [18, page 241-245]

$$0 \in \lambda\partial|x_i^*| + \frac{1}{p}(x_i^* - v_i) \text{ for } 1 \leq i \leq n, \tag{2.4}$$

where, by calculation,

$$\partial|x^*| = \left\{ u \,:\, u_i \begin{cases} = 1 & \text{if } x_i^* > 0, \\ \in [-1,1] & \text{if } x_i^* = 0, \\ = -1 & \text{if } x_i^* < 0. \end{cases} \right\}. \tag{2.5}$$

Case I : Suppose $v_i > p\lambda$. We have $x_i^* = v_i - p\lambda > 0$ from (2.3). Now,

$$\begin{aligned} \lambda\partial|x_i^*| + \frac{1}{p}(x_i^* - v_i) &= \lambda + \frac{1}{p}(v_i - p\lambda - v_i) \quad \text{(using (2.3) and (2.5))} \\ &= \lambda + (-\lambda) \\ &= 0. \end{aligned}$$

Case II : Suppose $v_i < -p\lambda$. We have $x_i^* = v_i + p\lambda < 0$ from (2.3). Now,

$$\begin{aligned} \lambda\partial|x_i^*| + \frac{1}{p}(x_i^* - v_i) &= -\lambda + \frac{1}{p}(v_i + p\lambda - v_i) \\ &= -\lambda + \lambda \\ &= 0. \end{aligned}$$

Case III : Suppose $|v_i| \leq p\lambda$. We have $x_i^* = 0$ from (2.3). Now,

$$\begin{aligned} & |v_i| \leq p\lambda \\ \Longrightarrow & -p\lambda \leq v_i \leq p\lambda \\ \Longrightarrow & -p\lambda - v_i \leq 0 \leq p\lambda - v_i \\ \Longrightarrow & -\lambda - \frac{v_i}{p} \leq 0 \leq \lambda - \frac{v_i}{p} \\ \Longrightarrow & \, 0 \in \left[-\lambda - \frac{v_i}{p}, \lambda - \frac{v_i}{p}\right] = \lambda\partial|x_i^*| + \frac{1}{p}(x_i^* - v_i). \end{aligned}$$

Therefore, (2.4) holds. □

Consider an unconstrained minimization problem of a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$

$$\min_{x \in \mathbb{R}^n} f(x). \tag{2.6}$$

One of the classical methods for solving (2.6) is the gradient descent algorithm which generates a sequence $\{x^{(k)}\}$ via the following iteration:

$$x^{(k)} = x^{(k-1)} - t^{(k)}\nabla f(x^{(k-1)}), \text{ given } x^{(0)} \in \mathbb{R}^n, \tag{2.7}$$

where $t^{(k)} > 0$ is a stepsize.

If f is differentiable, the first order linear approximation of f near $x^{(k-1)}$ can be written as

$$\tilde{f}(x^{(k)}) = f(x^{(k-1)}) + \nabla f(x^{(k-1)})^T(x^{(k)} - x^{(k-1)}). \tag{2.8}$$

The proximal operator of the function $\tilde{f}$ corresponding to $t^{(k)} > 0$ is given by

$$\begin{aligned}\mathbf{prox}_{t^{(k)},\tilde{f}}\left(x^{(k-1)}\right) &= \arg\min_{x^{(k)}}\left\{\tilde{f}\left(x^{(k)}\right) + \frac{1}{2t^{(k)}}\left\|x^{(k)} - x^{(k-1)}\right\|_2^2\right\} \\ &= \arg\min_{x^{(k)}}\Big\{ f\left(x^{(k-1)}\right) + \nabla f\left(x^{(k-1)}\right)^T\left(x^{(k)} - x^{(k-1)}\right) \\ &\quad + \frac{1}{2t^{(k)}}\left\|x^{(k)} - x^{(k-1)}\right\|_2^2\Big\}.\end{aligned}$$

Setting $\nabla\left(\mathbf{prox}_{t^{(k)},\tilde{f}}\left(x^{(k-1)}\right)\right) = \nabla f\left(x^{(k-1)}\right) + \frac{1}{t^{(k)}}\left(x^{(k)} - x^{(k-1)}\right) = 0$ gives

$$x^{(k)} = x^{(k-1)} - t^{(k)}\nabla f\left(x^{(k-1)}\right),$$

which is the gradient descent iteration (2.7). Therefore, (2.7) can be viewed as obtained by a proximal regularization of the linearized function f at $x^{(k-1)}$, i.e.,

$$x^{(k)} = \arg\min_{x}\left\{f\left(x^{(k-1)}\right) + \left\langle x - x^{(k-1)}, \nabla f\left(x^{(k-1)}\right)\right\rangle + \frac{1}{2t^{(k)}}\left\|x - x^{(k-1)}\right\|_2^2\right\}. \tag{2.9}$$

Linearizing function f in the nonsmooth ℓ_1 regularized problem

$$\min_{x\in\mathbb{R}^n}\{f(x) + \lambda\|x\|_1\}$$

gives the following iterative scheme

$$x^{(k)} = \arg\min_{x}\left\{f\left(x^{(k-1)}\right) + \left\langle x - x^{(k-1)}, \nabla f\left(x^{(k-1)}\right)\right\rangle + \frac{1}{2t^{(k)}}\left\|x - x^{(k-1)}\right\|_2^2 + \lambda\|x\|_1\right\}. \tag{2.10}$$

Let $r = x - x^{(k-1)}$. We have

$$\nabla f\left(x^{(k-1)}\right)^T r + \frac{1}{2t^{(k)}}\|r\|_2^2 \;=\; \frac{1}{2t^{(k)}}\left\|r + t^{(k)}\nabla f\left(x^{(k-1)}\right)\right\|_2^2 - \frac{t^{(k)}}{2}\left\|\nabla f\left(x^{(k-1)}\right)\right\|_2^2.$$

So (2.10) can be expressed as

$$\begin{aligned}
x^{(k)} &= \arg\min_x \left\{ \frac{1}{2t^{(k)}}\left\|x - \left(x^{(k-1)} - t^{(k)}\nabla f\left(x^{(k-1)}\right)\right)\right\|_2^2 - \frac{t^{(k)}}{2}\left\|\nabla f\left(x^{(k-1)}\right)\right\|_2^2 + \lambda\|x\|_1 \right\} \\
&= \arg\min_x \left\{ \frac{1}{2t^{(k)}}\left\|x - \left(x^{(k-1)} - t^{(k)}\nabla f\left(x^{(k-1)}\right)\right)\right\|_2^2 + \lambda\|x\|_1 \right\} \\
&= \mathcal{S}_{\lambda t^{(k)}}\left(x^{(k-1)} - t^{(k)}\nabla f\left(x^{(k-1)}\right)\right) \qquad (2.11) \\
&= \operatorname{shrink}\left(x^{(k-1)} - t^{(k)}\nabla f\left(x^{(k-1)}\right), \lambda t^{(k)}\right),
\end{aligned}$$

where (2.11) is justified by Theorem 2.1.1. In particular, when $f(x) = (1/2)\|Ax - b\|_2^2$, (2.11) can be written as

$$x^{(k)} = \mathcal{S}_{\lambda t^{(k)}}\left(x^{(k)} - 2t^{(k)}A^T(Ax^{(k-1)} - b)\right), \qquad (2.12)$$

which is called the basic ISTA iteration with step size $t > 0$. Now we extend the previous ideas to the following general problem

$$\min_{x\in\mathbb{R}^n} F(x), \qquad (2.13)$$

where $F(x) = f(x) + g(x)$, and

- $g : \mathbb{R}^n \to \mathbb{R}$ is a continuous convex function, possibly nonsmooth;
- $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth convex function of the type $C^{1,1}$, i.e., continuously differentiable with Lipschitz continuous gradient

  $$\|\nabla f(x) - \nabla f(y)\|_2 \le L(f)\|x - y\|_2 \text{ for every } x, y \in \mathbb{R}^n,$$

  where $L(f) > 0$ is the Lipschitz constant of ∇f. In the standard LASSO problem, $f(x) = (1/2)\|Ax - b\|_2^2$ and $g(x) = \lambda\|x\|_1$ for which the smallest Lipschitz constant of the gradient ∇f is $L(f) = \lambda_{\max}(A^TA)$.

For any given $L > 0$, consider the following approximation of $F(x)$ at a given point y,

$$Q_L(x, y) = f(y) + \langle x - y, \nabla f(y) \rangle + \frac{L}{2}\|x - y\|_2^2 + g(x). \tag{2.14}$$

Let $P_L(y)$ be the unique minimizer of (2.14) over x, i.e.,

$$P_L(y) = \arg\min_{x \in \mathbb{R}^n} \left\{ f(y) + \langle x - y, \nabla f(y) \rangle + \frac{L}{2}\|x - y\|_2^2 + g(x) \right\}, \tag{2.15}$$

which can be expressed as

$$P_L(y) = \arg\min_{x \in \mathbb{R}^n} \left\{ g(x) + \frac{L}{2} \left\| x - \left(y - \frac{1}{L}\nabla f(y) \right) \right\|_2^2 \right\}. \tag{2.16}$$

Therefore, the basic step of ISTA for solving the problem (2.13) is

$$x^{(k)} = P_L(x^{(k-1)}), \tag{2.17}$$

where the reciprocal of L plays the role of a stepsize. Now, we summarize the ISTA algorithm with a fixed stepsize for solving (2.13) in Algorithm 2.1.

Algorithm 2.1 ISTA with constant stepsize

Input: Lipschitz constant $L := L(f)$ of ∇f, and error tolerance ϵ;

Output: a solution of problem (2.13).

1: **Initialize:** $x^{(0)} \in \mathbb{R}^n$, $k = 1$;
2: **while** $|F(x^{(k)}) - F(x^{(k-1)})| > \epsilon$ **do**
3: $\quad x^{(k)} = P_L(x^{(k-1)})$;
4: $\quad k = k + 1$;
5: **end while**
6: **return** last $x^{(k)} \in \mathbb{R}^n$.

For the standard LASSO problem (2.1), Algorithm 2.1 reduces to the iterative shrinkage method (2.12) with step size $t = \frac{1}{L(f)}$. The major drawback of Algorithm 2.1

is that the Lipschitz constant $L(f)$ is computationally expensive to obtain for large scale problems. For instance, in the LASSO problem $L(f)$ depends on the largest eigenvalue of the matrix A^TA which is not readily known for large-scale problems. Therefore, we now introduce an ISTA algorithm with backtracking to determine a stepsize in each step.

Algorithm 2.2 ISTA with backtracking

Input: error tolerance ϵ and parameter $\eta > 1$;

Output: a solution of problem (2.13).

1: **Initialize:** $x^{(0)} \in \mathbb{R}^n$, $L^{(0)} > 0$, $k = 1$;

2: **while** $|F(x^{(k)}) - F(x^{(k-1)})| > \epsilon$ **do**

3: Find the smallest nonnegative integers i_k such that with $\bar{L} = \eta^{i_k} L^{(k-1)}$

$F\left(P_{\bar{L}}(x^{(k-1)})\right) \leq Q_{\bar{L}}\left(P_{\bar{L}}(x^{(k-1)}), x^{(k-1)}\right)$; ▷ Backtracking line search step.

4: Set $L^{(k)} = \eta^{i_k} L^{(k-1)}$ and compute

$$x^{(k)} = P_{L^{(k)}}(x^{(k-1)});$$ ▷ Solution update step.

5: $k = k + 1$;

6: **end while**

7: **return** last $x^{(k)} \in \mathbb{R}^n$.

It is worth mentioning the performance of ISTA that we described in Algorithm 2.1 and Algorithm 2.2. Here, we mention the convergence rate of the algorithm in Theorem 2.1.2. For a detailed proof, the reader is referred to [35].

Theorem 2.1.2. *Let* $\{x^{(k)}\}$ *be the sequence generated by the relation* (2.17)*. Then for any* $k \geq 1$

$$F(x^{(k)}) - F(x^*) \leq \frac{\alpha L(f) \|x^{(0)} - x^*\|^2}{2k} \quad \textit{for some } x^* \in X_*,$$

where $\alpha = 1$ in the constant stepsize setting and $\alpha = \eta$ in the backtracking stepsize setting, and $X_* = \arg\min F$.

Theorem 2.1.2 tells us that the objective function value $F(x^{(k)})$ converges to the minimum $F(x^*)$ at a rate of convergence no worse than $\mathcal{O}\left(\frac{1}{k}\right)$. In the next subsection we discuss another algorithm with a much faster rate of convergence to ISTA.

2.2 Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

As we know, ISTA converges at the rate of $\mathcal{O}\left(\frac{1}{k}\right)$. When $g(x) \equiv 0$, the general model (2.13) consists of minimizing a smooth convex function and ISTA reduces to the gradient descent method. In this case, the existence of a gradient method with an $\mathcal{O}\left(\frac{1}{k^2}\right)$ convergence rate is proven in [37]. In [35], the same convergence result is extended to model (2.13). We now present the fast version of ISTA with constant stepsize in Algorithm 2.3.

Algorithm 2.3 FISTA with constant stepsize

Input: Lipschitz constant $L := L(f)$ of ∇f, and error tolerance ϵ;

Output: a solution of problem (2.13).

1: **Initialize:** $y^{(1)}$, $x^{(0)} \in \mathbb{R}^n$, $t^{(1)} = 1$, $k = 1$;
2: **while** $|F(x^{(k)}) - F(x^{(k-1)})| > \epsilon$ **do**
3: $\quad x^{(k)} = P_L(y^{(k)})$;
4: $\quad t^{(k+1)} = \frac{1+\sqrt{1+4(t^{(k)})^2}}{2}$;
5: $\quad y^{(k+1)} = x^{(k)} + \left(\frac{t^{(k)}-1}{t^{(k+1)}}\right)(x^{(k)} - x^{(k-1)})$;
6: $\quad k = k + 1$;
7: **end while**
8: **return** last $x^{(k)} \in \mathbb{R}^n$.

We have the same issue of computational burden for finding the value of Lipschitz constant L for large scale problems in Algorithm 2.3 as we had in Algorithm 2.1. As a remedy, we now present FISTA with backtracking in Algorithm 2.4.

Algorithm 2.4 FISTA with backtracking

Input: error tolerance ϵ and parameter $\eta > 1$;

Output: a solution of problem (2.13).

1: **Initialize:** $x^{(0)}$, $y^{(1)} \in \mathbb{R}^n$, $L^{(0)} > 0$, $t^{(1)} = 1$, $k = 1$;

2: **while** $|F(x^{(k)}) - F(x^{(k-1)})| > \epsilon$ **do**

3: Find the smallest nonnegative integers i_k such that with $\bar{L} = \eta^{i_k} L^{(k-1)}$ $F\left(P_{\bar{L}}\left(y^{(k)}\right)\right) \leq Q_{\bar{L}}\left(P_{\bar{L}}\left(y^{(k)}\right), y^{(k)}\right)$; ▷ Backtracking line search step.

4: Set $L^{(k)} = \eta^{i_k} L^{(k-1)}$ and compute ▷ Updating solution.

$$x^{(k)} = P_{L^{(k)}}\left(y^{(k)}\right);$$
$$t^{(k+1)} = \frac{1+\sqrt{1+4(t^{(k)})^2}}{2};$$
$$y^{(k+1)} = x^{(k)} + \left(\frac{t^{(k)}-1}{t^{(k+1)}}\right)(x^{(k)} - x^{(k-1)});$$

5: $k = k + 1$;

6: **end while**

7: **return** last $x^{(k)} \in \mathbb{R}^n$.

We now state the convergence result of Algorithm 2.3 and Algorithm 2.4. For a detailed proof and further discussions, the reader is referred to [37] and references therein.

Theorem 2.2.1. *Let $\{x^{(k)}\}$ be generated by FISTA. Then for any $k \geq 1$*

$$F\left(x^{(k)}\right) - F\left(x^*\right) \leq \frac{2\alpha L(f)\|x^{(0)} - x^*\|^2}{(k+1)^2} \quad \text{for some } x^* \in X_*,$$

where $\alpha = 1$ in the constant stepsize setting (Algorithm 2.3) and $\alpha = \eta$ in the backtracking stepsize setting (Algorithm 2.4).

Theorem 2.2.1 says that the objective function value of FISTA converges at a rate no worse than $\mathcal{O}\left(\frac{1}{k^2}\right)$.

2.3 Alternating Direction Method of Multipliers (ADMM)

In this section, we will discuss ADMM [38], popularly used algorithm for solving optimization problems arising in a wide variety of applications such as deep learning [39], constrained sparse regression [40], sparse signal recovery [41], image restoration and denoising [42], the Dantzig selector [43], support vector machines [44], signal processing and ℓ_1 optimization [45], and so forth. As ADMM consists of minimizing an augmented Lagrangian function jointly with respect to primal and dual variables update, we first discuss the key idea of dual ascent as a motivation to this algorithm.

2.3.1 Dual ascent

We start with the concept of dual ascent from the notion of conjugate function.

Definition 2.3.1. ([18]) Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. The function $f^* : \mathbb{R}^n \to \mathbb{R}$ defined as $f^*(y) = \sup_{x \in \mathscr{D}(f)} (y^T x - f(x))$ is called the *conjugate function* of f.

Let us consider the equality constrained convex optimization problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & Ax = b, \end{aligned} \tag{2.18}$$

where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, and $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function.

The Lagrangian function for the problem (2.18) is

$$L(x, y) = f(x) + y^T (Ax - b)$$

and the dual function is

$$\begin{aligned} g(y) = \inf_x L(x,y) &= \inf_x \left\{ f(x) + y^T(Ax - b) \right\} \\ &= \inf_x \left\{ y^T Ax + f(x) \right\} - y^T b \\ &= -f^*(-A^T y) - b^T y, \end{aligned}$$

where $y \in \mathbb{R}^m$ is the dual variable, and f^* is the conjugate of f. The dual problem of (2.18) is to find the maximizer of dual function $g(y)$

$$y^* = \arg\max_{y \in \mathbb{R}^m} g(y) = \arg\max_{y \in \mathbb{R}^m} \left\{ \inf_{x \in \mathbb{R}^n} L(x,y) \right\}.$$

Let x^* be the optimal solution of primal problem (2.18), $p^* = f(x^*)$, and $d^* = g(y^*)$. The weak duality condition gives the lower bound for the primal optimal value, i.e. $d^* \leq p^*$. If f is strictly convex, then the strong duality holds, i.e., $d^* = p^*$. Moreover, we can recover a primal optimal solution x^* from a dual optimal solution y^* as

$$x^* = \arg\min_x L(x, y^*).$$

In the dual ascent method, we solve the dual problem using gradient ascent. It is given by

$$x^{(k+1)} := \arg\min_x L\left(x, y^{(k)}\right), \tag{2.19}$$

$$y^{(k+1)} := y^{(k)} + \alpha^{(k)} \left(Ax^{(k+1)} - b\right), \tag{2.20}$$

where $\alpha^{(k)} > 0$ is a step size. The primal and dual variable updates are given by (2.19) and (2.20), respectively. Intuitively, the algorithm is called dual ascent, since with appropriate choice of stepsize $\alpha^{(k)}$, the dual function increases in each step, i.e., $g\left(y^{(k+1)}\right) > g\left(y^{(k)}\right)$.

2.3.2 ADMM Algorithm

The ADMM algorithm solves the problem of the following form

$$\begin{aligned} \underset{x\in\mathbb{R}^n,\ z\in\mathbb{R}^m}{\text{minimize}} \quad & f(x)+g(z) \\ \text{subject to} \quad & Ax+Bz=c, \end{aligned} \tag{2.21}$$

where $A\in\mathbb{R}^{p\times n}$, $B\in\mathbb{R}^{p\times m}$, and $c\in\mathbb{R}^p$. We assume that both functions f and g are convex. Here, the only difference between the general problem (2.18) and (2.21) is that the variable x in (2.18) is split into two parts, namely x and z in (2.21). Also, the objective function is separable according to the split. The augmented Lagrangian [38] of the problem (2.21) is formulated as

$$L_\rho(x,z,y)=f(x)+g(z)+y^T(Ax+Bz-c)+(\rho/2)\left\|Ax+Bz-c\right\|_2^2,$$

where $\rho>0$ is the augmented Lagrangian parameter. ADMM consists of the following iterations

$$x^{(k+1)} := \underset{x}{\operatorname{argmin}}\ L_\rho\left(x,z^{(k)},y^{(k)}\right), \tag{2.22}$$

$$z^{(k+1)} := \underset{z}{\operatorname{argmin}}\ L_\rho\left(x^{(k+1)},z,y^{(k)}\right), \tag{2.23}$$

$$y^{(k+1)} := y^{(k)}+\rho\left(Ax^{(k+1)}+Bz^{(k+1)}-c\right). \tag{2.24}$$

The iterations consist of an x-minimization step (2.22), a z-minimization step (2.23) and a dual variable update (2.24) which takes ρ as the stepsize. Let $r=Ax+Bz-c$. We have

$$\begin{aligned} y^Tr+(\rho/2)\left\|r\right\|_2^2 &= (\rho/2)\left\|r+(1/\rho)y\right\|_2^2-(1/2\rho)\left\|y\right\|_2^2 \\ &= (\rho/2)\left\|r+u\right\|_2^2-(\rho/2)\left\|u\right\|_2^2, \end{aligned}$$

where $u=(1/\rho)y$ is the scaled dual variable. We can reformulate the ADMM iterations as

$$x^{(k+1)} := \arg\min_x\left\{f(x)+(\rho/2)\left\|Ax+Bz^{(k)}-c+u^{(k)}\right\|_2^2\right\},$$

$$
\begin{aligned}
z^{(k+1)} &:= \arg\min_z \left\{ g(z) + (\rho/2) \left\| Ax^{(k+1)} + Bz - c + u^{(k)} \right\|_2^2 \right\}, \\
u^{(k+1)} &:= u^{(k)} + Ax^{(k+1)} + Bz^{(k+1)} - c.
\end{aligned}
$$

The stopping criteria for the ADMM algorithm are determined by the primal and dual residuals. The primal and dual residuals at the k-th iteration are defined as

$$
\begin{aligned}
r^{(k)} &= Ax^{(k)} + Bz^{(k)} - c, \\
s^{(k)} &= \rho A^T B \left(z^{(k)} - z^{(k-1)} \right),
\end{aligned}
$$

respectively. The ADMM algorithm is stopped when the primal and dual residuals are smaller than preset tolerances, i.e.,

$$
\|r^{(k)}\|_2 \le \epsilon^{\text{pri}} \quad \text{and} \quad \|s^{(k)}\|_2 \le \epsilon^{\text{dual}}, \tag{2.25}
$$

where $\epsilon^{\text{pri}} > 0$ and $\epsilon^{\text{dual}} > 0$ are feasibility tolerances for the primal and dual feasibility conditions, respectively. These tolerances can be chosen using an absolute and relative criterion, such as

$$
\begin{aligned}
\epsilon^{\text{pri}} &= \sqrt{p}\,\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \max\left\{ \|Ax^{(k)}\|_2, \|Bz^{(k)}\|_2, \|c\|_2 \right\}, \\
\epsilon^{\text{dual}} &= \sqrt{n}\,\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \|A^T y^{(k)}\|_2,
\end{aligned}
$$

where $\epsilon^{\text{abs}} > 0$ and $\epsilon^{\text{rel}} > 0$ are absolute and relative tolerances, respectively. Theoretical justification of the above stopping criteria are explained in [38, §3.3.1].

2.3.3 Solving LASSO problem using ADMM

The LASSO problem

$$
\min_{x \in \mathbb{R}^n} (1/2)\|Ax - b\|_2^2 + \lambda \|x\|_1, \tag{2.26}
$$

with $A \in \mathbb{R}^{m\times n}$ $(m \ll n)$, $b \in \mathbb{R}^{m\times 1}$, and $\lambda > 0$ can be expressed into the standard ADMM formulation as

$$\begin{array}{ll} \text{minimize} & f(x) + g(z) \\ \text{subject to} & x - z = 0, \end{array}$$

where $f(x) = (1/2)\|Ax - b\|_2^2$ and $g(z) = \lambda\|z\|_1$. Its augmented Lagrangian function is given by

$$L_\rho(x, y, z) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z\|_1 + y^T(x - z) + \frac{\rho}{2}\|x - z\|_2^2.$$

Let $r = x - z$. We have

$$\begin{aligned} y^T r + \frac{\rho}{2}\|r\|_2^2 &= \frac{\rho}{2}\|r + (1/\rho)y\|_2^2 - \frac{1}{2\rho}\|y\|_2^2 \\ &= \frac{\rho}{2}\|r + u\|_2^2 - \frac{\rho}{2}\|u\|_2^2, \end{aligned}$$

where $u = (1/\rho)y$. The x-update rule then is

$$\begin{aligned} x^{(k+1)} &= \arg\min_x \left\{ \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|z^{(k)}\|_1 + y^T\left(x - z^{(k)}\right) + \frac{\rho}{2}\left\|x - z^{(k)}\right\|_2^2 \right\} \\ &= \arg\min_x \left\{ \frac{1}{2}\|Ax - b\|_2^2 + \frac{\rho}{2}\left\|x - z^{(k)} + u^{(k)}\right\|_2^2 \right\} \\ &= \left(A^T A + \rho I\right)^{-1}\left(A^T b + \rho\left(z^{(k)} - u^{(k)}\right)\right). \end{aligned}$$

Similarly, the z-update is

$$\begin{aligned} z^{(k+1)} &= \arg\min_z \left\{ \frac{1}{2}\left\|Ax^{(k+1)} - b\right\|_2^2 + \lambda\|z\|_1 + \rho\left\langle y^{(k)}, \left(x^{(k+1)} - z\right)\right\rangle + \frac{\rho}{2}\left\|x^{(k+1)} - z\right\|_2^2 \right\} \\ &= \arg\min_z \left\{ \lambda\|z\|_1 + \frac{\rho}{2}\left\|z - \left(x^{(k+1)} + u^{(k)}\right)\right\|_2^2 \right\} \\ &= \mathcal{S}_{\lambda/\rho}\left(x^{(k+1)} + u^{(k)}\right) = \text{shrink}\left(x^{(k+1)} + u^{(k)}, \frac{\lambda}{\rho}\right). \end{aligned}$$

Finally, the dual variable u-update is

$$u^{(k+1)} = u^{(k)} + x^{(k+1)} - z^{(k+1)}.$$

We now summarize the process for solving LASSO (2.26) by ADMM in Algorithm 2.5.

Algorithm 2.5 ADMM Algorithm for solving LASSO Problem.

Input: $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m, \epsilon,\ \rho > 0,\ \lambda > 0$, absolute tolerance $\epsilon^{\text{abs}} > 0$, and relative tolerance $\epsilon^{\text{rel}} > 0$;

Output: a solution of the problem (2.26).

1: **Initialize:** $k = 0,\ u^{(0)}, z^{(0)} \in \mathbb{R}^n$;

2: **while** the convergence criteria (2.25) are not satisfied **do**

3: $\quad x^{(k+1)} = \left(A^T A + \rho I\right)^{-1} \left(A^T b + \rho \left(z^{(k)} - u^{(k)}\right)\right)$;

4: $\quad z^{(k+1)} = \mathcal{S}_{\lambda/\rho}\left(x^{(k+1)} + u^{(k)}\right)$;

5: $\quad u^{(k+1)} = u^{(k)} + x^{(k+1)} - z^{(k+1)}$;

6: $\quad k = k + 1$;

7: **end while**

8: **return** last $z^{(k)} \in \mathbb{R}^n$.

The convergence of ADMM has been widely discussed in many scientific research [38, 46]. Also, the objective function values in ADMM convergence at a rate no worse than $\mathcal{O}\left(\frac{1}{k}\right)$ [47].

2.4 ℓ_1-Homotopy Algorithm

In this section, we discuss another ℓ_1 optimization technique called the ℓ_1-homotopy. There are many algorithms based on the homotopy approach to solve ℓ_1-minimization, e.g., [48, 49]. Our presentation of ℓ_1-homotopy algorithm is based on Asif and Romberg [17].

Suppose y is a vector of observations that satisfies the linear system of equations $y = A\bar{x} + N_\epsilon$, where $\bar{x}$ is a sparse unknown vector of interest, $A \in \mathbb{R}^{m\times n}$ is a measurement matrix, and N_ϵ is a noise vector. We solve the following ℓ_1-minimization problem to recover $\bar{x}$:

$$\min_{x\in\mathbb{R}^n} \|Wx\|_1 + \frac{1}{2}\|Ax - y\|_2^2, \tag{2.27}$$

where $W = \mathrm{diag}(w) \in \mathbb{R}^{n\times n}$ is a diagonal matrix containing the positive weights $w \in \mathbb{R}^n$ on the diagonal. The choice of weights gives the flexibility for imposing dynamic penalties instead of a single ℓ_1 penalty parameter in LASSO which is used to enhance the performance of sparsity recovery in the signal [50–54]. The homotopy methods give a general framework for solving the problem starting from the available solution, and a series of simple problems are solved along the homotopy path toward the final solution of the original problem. This process is controlled by a homotopy parameter lying between 0 and 1, linking the two endpoints of the homotopy path.

We now describe the homotopy method for solving (2.27) by taking the homotopy parameter $\epsilon \in [0, 1]$, warm-start vector $\hat{x}$, and solving the following optimization problem as ϵ moves from 0 to 1.

$$f_\epsilon = \min_x \|Wx\|_1 + \frac{1}{2}\|Ax - y\|_2^2 + (1-\epsilon)u^T x, \tag{2.28}$$

where u is defined as

$$u \;\equiv\; -W\hat{z} - A^T(A\hat{x} - y) \;:\; \hat{z} \begin{cases} = \mathrm{sign}(\hat{x}) & \text{on } \hat{\Gamma}, \\ < 1 & \text{on } \hat{\Gamma}^c \end{cases} \tag{2.29}$$

with $\hat{\Gamma} = \{i \mid \hat{x}(i) \neq 0\}$ is the support set of the vector $\hat{x}$ and $\hat{\Gamma}^c$ is the complement of the support set $\hat{\Gamma}$. When $\epsilon = 0$,

$$\partial f_\epsilon \;=\; W\hat{z} + A^T(A\hat{x} - y) - W\hat{z} - A^T(A\hat{x} - y) = 0,$$

which shows that $\hat{x}$ is an optimal solution of (2.28) with $\epsilon = 0$. The main idea of the homotopy algorithm is, as ϵ changes from 0 to 1, the problem (2.28) gradually transforms into the problem (2.27) and the solution of (2.28) follows a piecewise linear path from $\hat{x}$ towards the solution of problem (2.27). Suppose x^* is a solution of the problem (2.28). Then it satisfies the following KKT optimality conditions [18, page 241-245]:

$$Wg + A^T(Ax^* - y) + (1-\epsilon)u = 0, \tag{2.30}$$
$$\|g\|_\infty \leq 1, g^T x^* = \|x^*\|_1,$$

where $g \in \partial\|x^*\|_1$ denotes the subdifferential of $\|x^*\|_1$. For any values of $\epsilon \in [0,1]$, the solution x^* must satisfy the following (splitting the KKT optimality condition (2.30) into the support set and its complement),

$$a_i^T(Ax^* - y) + (1-\epsilon)u_i = -w_i z_i \text{ for all } i \in \Gamma, \tag{2.31}$$
$$|a_i^T(Ax^* - y) + (1-\epsilon)u_i| \leq w_i \text{ for all } i \in \Gamma^c,$$

where Γ is the support set of the solution x^*, z denotes its sign on the support set, and a_i denotes the i-th column of the matrix A. In every homotopy step, we go from one critical value (value of ϵ where support set changes) of ϵ to next by updating the support set of the solution until it hits 1. As ϵ increases by a small value δ, the objective function value f_ϵ changes to $f_{\epsilon+\delta}$, i.e.,

$$f_{\epsilon+\delta} = \min_x \ \|Wx\|_1 + \frac{1}{2}\|Ax - y\|_2^2 + (1-\epsilon-\delta)u^T x. \tag{2.32}$$

Thus, as ϵ is increased by a small value δ, the solution moving along the update direction v_x should satisfy the following optimality conditions :

$$a_i^T(Ax^* - y) + (1-\epsilon)u_i + \delta(a_i^T A v_x - u_i) = -w_i z_i \text{ for all } i \in \Gamma, \tag{2.33}$$

$$|\underbrace{a_i^T(Ax^* - y) + (1-\epsilon)u_i}_{p_i} + \underbrace{\delta(a_i^T A v_x - u_i)}_{d_i}| \leq w_i \text{ for all } i \in \Gamma^c. \tag{2.34}$$

Therefore, the update direction that keeps the solution optimal as we change δ can be expressed as

$$v_x = \begin{cases} (A_\Gamma^T A_\Gamma)^{-1} u_\Gamma & \text{on } \Gamma, \\ 0 & \text{otherwise.} \end{cases} \tag{2.35}$$

Let δ^- denote the smallest step size such that a active constraint in (2.33) becomes inactive which shrinks an existing element at an index γ^- on the support set Γ to 0. We have

$$\delta^- = \min_{i \in \Gamma} \left(\frac{-x_i^*}{(v_x)_i} \right)_+ .$$

Let δ^+ denote the smallest step size such that an inactive constraint in (2.34) becomes active at an index γ^+ indicating that it should enter into the support Γ. We have from inequality constraint (2.34)

$$\begin{aligned} & |p_i + \delta d_i| \leq w_i \\ \implies & \frac{-w_i - p_i}{d_i} \leq \delta \leq \frac{w_i - p_i}{d_i} \\ \implies & \delta^+ = \min_{i \in \Gamma^c} \left(\frac{w_i - p_i}{d_i}, \frac{-w_i - p_i}{d_i} \right)_+ , \end{aligned}$$

where $\min\limits_{x \in \mathbb{R}^n} (x)_+$ means we just take the positive argument of the vector x and take the minimum over those positive arguments. Thus, if one of the optimality conditions in (2.33) and (2.34) is violated, we add or remove the elements from the support set Γ. The smallest step size that causes one of these changes in the support is computed as

$$\delta^* = \min\,(\delta^+, \delta^-). \tag{2.36}$$

If γ^+ is added to the support, then in the next iteration we check whether the sign constraint in (2.33) is violated or not. If the sign of z_{γ^+} and $(v_x)_{\gamma^+}$ are different, then we remove the entry of Γ at the position of γ^+ and recompute the update direction v_x. In summary, the ℓ_1-homotopy algorithm is presented in Algorithm 2.6.

Algorithm 2.6 ℓ_1-Homotopy Algorithm

Input: $A \in \mathbb{R}^{m\times n}$, $y \in \mathbb{R}^m$, diagonal weight matrix $W \in \mathbb{R}^{n\times n}$, warm-start vector $\hat{x}$, and a vector $u \in \mathbb{R}^n$ as defined in (2.29);

Output: a solution of problem (2.27).

Initialize: $\epsilon = 0, x^* = \hat{x}$;
while $\epsilon \neq 1$ **do**
    Compute v_x as in (2.35); ▷ Update direction;
    Compute p and d as in (2.34);
    Compute $\delta^* = \min(\delta^+, \delta^-)$ as in (2.36); ▷ Step size
    if $\epsilon + \delta^* > 1$ **then**
        $\delta^* = 1 - \epsilon$ ▷ Last iteration
        $x^* = x^* + \delta^* v_x$ ▷ Final solution
        break
    end if
    $x^* = x^* + \delta^* v_x$ ▷ Update solution
    $\epsilon = \epsilon + \delta^*$ ▷ Update the homotopy parameter
    if $\delta^* = \delta^-$ **then**
        $\Gamma = \Gamma \backslash \gamma^-$ ▷ Remove an element from the support
    else
        $\Gamma = \Gamma \cup \gamma^+$ ▷ Add a new element to the support
    end if
end while
return $x^* \in \mathbb{R}^n$.

2.5 Iterative Reweighting via Homotopy

In this subsection, we present another variant of the homotopy algorithm, called iterative reweighting via homotopy, to solve the ℓ_1-minimization problem. We present a detailed discussion of iterative reweighting via homotopy algorithm based on [55]. The idea behind the iterative reweighting via homotopy is similar to the ideas discussed in ℓ_1-homotopy in Section 2.4. In ℓ_1-homotopy, we start with the initial guess $\hat{x}$ and a vector u defined by $\hat{x}$ and initial weight W so that $\hat{x}$ is optimal when the homotopy parameter $\epsilon = 0$. Then we trace the homotopy path along with this initial solution $\hat{x}$ and u to the exact solution as homotopy parameter ϵ changes from 0 to 1. But in iterative reweighting via homotopy, we want to solve the following weighted ℓ_1-minimization problem:

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{n} w_i|x_i| + \frac{1}{2}\|Ax-y\|_2^2, \tag{2.37}$$

where $A \in \mathbb{R}^{m\times n}$, $y \in \mathbb{R}^n$, and w_i is a positive weight serving as a ℓ_1 penalty for each i. In the iterative reweighting algorithm, we solve the problem (2.37) for given weights $w_i > 0$, then we update w_i to new weights $\tilde{w}_i > 0$ and solve the following new problem for better signal recovery

$$\min_x \sum_{i=1}^{n} \tilde{w}_i|x_i| + \frac{1}{2}\|Ax-y\|_2^2. \tag{2.38}$$

To make the bridge between the problem (2.37) with starting weights and the problem (2.38) with new weights, we solve the following homotopy path problem,

$$\min_x \sum_{i=1}^{n} \left((1-\epsilon)w_i + \epsilon\tilde{w}_i\right)|x_i| + \frac{1}{2}\|Ax-y\|_2^2, \tag{2.39}$$

where $\epsilon \in [0,1]$ is a homotopy parameter. Once ϵ changes from 0 to 1 in (2.39), the solutions follow the linear homotopy path from the solution of (2.37) to the solution of (2.38). From the similar discussion and analysis made in Section 2.4, for any value of homotopy paramater ϵ, the solution x^* of optimization problem (2.39) satisfies the following KKT optimality conditions:

$$
\begin{aligned}
a_i^T(Ax^* - y) &= -((1-\epsilon)w_i + \epsilon\tilde{w}_i)z_i, \quad \text{for all } i \in \Gamma, \\
|a_i^T(Ax^* - y)| &< (1-\epsilon)w_i + \epsilon\tilde{w}_i, \quad \text{for all } i \in \Gamma^c,
\end{aligned}
$$

where a_i, Γ, and z_i denote the i-th column of the matrix A, support set of the solution vector x^*, and the sign of x^* on the support set Γ, respectively. As the homotopy parameter ϵ is increased by a small step δ, the solution moves in the update direction v_x. The optimality condition is changed according to the new step δ and update direction as follows:

$$
\begin{aligned}
A_\Gamma^T(Ax^* - y) + \delta A_\Gamma^T A v_x &= -\left((1-\epsilon)W + \epsilon\tilde{W}\right)z + \delta(W - \tilde{W})z \text{ on } \Gamma, && (2.40) \\
|\underbrace{a_i^T(Ax^* - y)}_{p_i} + \delta\underbrace{a_i^T A v_x}_{d_i}| &\le \underbrace{((1-\epsilon)w_i + \epsilon\tilde{w}_i)}_{q_i} + \delta\underbrace{(\tilde{w}_i - w_i)}_{s_i}, \ \forall\, i \in \Gamma^c, && (2.41)
\end{aligned}
$$

where W and $\tilde{W}$ denotes $|\Gamma| \times |\Gamma|$ diagonal matrices with entries on diagonal being w and $\tilde{w}$ on Γ, respectively. Here, $|\Gamma|$ denotes the cardinality of the support set Γ.

The update direction v_x as we move δ step further by keeping the solution along the homotopy path can be obtained by analyzing the optimality condition (2.40) by,

$$
v_x = \begin{cases} (A_\Gamma^T A_\Gamma)^{-1}(W - \tilde{W})z & \text{on } \Gamma, \\ 0 & \text{on } \Gamma^c. \end{cases} \tag{2.42}
$$

The solution progresses in the homotopy path with small step size δ in update direction v_x until it violates one of the optimality constraints (2.40) or (2.41) which causes either a new element to enter the support (inactive constraint (2.41) becomes active) or the existing element shrinks to zero (when active constraint (2.40) is violated) on the support. The value of the stepsize δ that takes the solution to such a critical value of ϵ can be computed as $\delta^* = \min(\delta^+, \delta^-)$, where

$$\begin{aligned} \delta^+ &= \min_{i \in \Gamma^c} \left(\frac{q_i - p_i}{-s_i + d_i}, \frac{-q_i - p_i}{s_i + d_i} \right)_+, \\ \delta^- &= \min_{i \in \Gamma} \left(\frac{-x_i^*}{(v_x)_i} \right)_+ . \end{aligned} \tag{2.43}$$

Here, δ^+ is the smallest step size that causes an inactive constraint in (2.41) to become active which means the index for that instance, γ^+ enters into the support. Also, δ^- is the smallest step size that causes a violation of a constraint in (2.40) which causes an entry for that index γ^- shrinks to zero . We perform the homotopic continuation from one critical value to another critical value unless the homotopic parameter ϵ hits 1. In summary, the entire process of iterative reweighting via homotopy is described in Algorithm 2.7.

Algorithm 2.7 Iterative Reweighting via Homotopy

Input: $A \in \mathbb{R}^{m\times n}$, $y \in \mathbb{R}^m$, warm-start vector $\hat{x}$, weight vector $w \in \mathbb{R}^n$, and modified weight vector $\tilde{w} \in \mathbb{R}^n$;

Output: a solution of problem (2.37).

```
Initialize : ϵ = 0, x* = x̂;
while ϵ ≠ 1 do
    Compute v_x as in (2.42);                        ▷ Update direction;
    Compute p, d, q, and s as in (2.41);
    Compute δ* = min(δ+, δ−) as in (2.43);           ▷ Step size
    if ϵ + δ* > 1 then
        δ* = 1 − ϵ                                   ▷ Last iteration
        x* = x* + δ* v_x                             ▷ Final solution
        break
    end if
    x* = x* + δ* v_x                                 ▷ Update solution
    ϵ = ϵ + δ*                                       ▷ Update the homotopy parameter
    if δ* = δ− then
        Γ ← Γ\γ−                                     ▷ Remove an element from the support
    else
        Γ ← Γ ∪ γ+                                   ▷ Add a new element to the support
    end if
end while
return x* ∈ ℝ^n.
```

2.6 Reweighted ℓ_1-minimization for sparsity enhancement

Consider the weighted ℓ_1-norm minimization problem:

$$\min_{x\in\mathbb{R}^n} \|Wx\|_1 + \frac{1}{2}\|Ax-y\|_2^2, \tag{2.44}$$

where $A \in \mathbb{R}^{m\times n}$, $y \in \mathbb{R}^m$, and $W = \mathrm{diag}(w) \in \mathbb{R}^{n\times n}$ is a diagonal matrix containing the positive weights $w \in \mathbb{R}^n$ on the diagonal. In the ℓ_1-minimization problem (2.44), one of the major concerns is to choose appropriate weights to improve reconstruction and sparsity in the solution. In this subsection, we provide some of the well-known weight improvement techniques (see e.g. [50, 54]), which show that the reweighted ℓ_1-minimization outperforms unweighted ℓ_1-minimization in many situations. The key feature of reweighted ℓ_1-minimization is to solve the series of weighted ℓ_1 problems

$$\min_{x\in\mathbb{R}^n} \|W^{(t)}x\|_1 + \frac{1}{2}\|Ax-y\|_2^2, \tag{2.45}$$

where $W^{(t)} = \mathrm{diag}(w^{(t)})$, and $w^{(t)} \in \mathbb{R}^n$ is a vector of positive weights determined by the previous solution $x^{(t-1)}$ of (2.45). We determine the next improved solution $x^{(t+1)}$ of (2.45) by a the new weight $w^{(t+1)}$ determined by the current solution $x^{(t)}$. The whole process of iterative reweighting is presented in Algorithm 2.8.

Using Algorithm 2.8, we recalculate weights in each iteration for sparse and improved signal reconstruction. In step 5 of Algorithm 2.8, we used the weight modifying scheme introduced by Candès, Wakins, and Boyd (CWB) [50]. Besides CWB, there are many other popularly used reweighted schemes which can be used in step 5 of Algorithm 2.8. Here, we briefly review some [54, 56].

- (NW1 method) For $p \in (0,1)$ and $\delta > 0$,

$$w_i^{(t+1)} = \frac{p + (|x_i^{(t)}| + \delta)^{1-p}}{(|x_i^{(t)}| + \delta)^{1-p}\left[|x_i^{(t)}| + \delta + (|x_i^{(t)}| + \delta)^p\right]},\ i = 1,\ 2, \cdots, n.$$

Algorithm 2.8 Iterative Reweighting

Input: $A \in \mathbb{R}^{m\times n}$, $y \in \mathbb{R}^m$, parameter $\delta > 0$, and maximum iteration $N_{\max}$;

Output: a modified solution of problem (2.38).

1: **Initialize** : $t = 0,\ w_i^{(0)} = 1\ \forall\ i$;

2: **while** $t \le N_{\max}$ **do**

3: Solve the weighted ℓ_1-minimization problem by any appropriate ℓ_1 solving methods such as Algorithm 2.2, Algorithm 2.4, Algorithm 2.5, Algorithm 2.6, or Algorithm 2.7.

$x^{(t)} = \underset{x\in\mathbb{R}^n}{\arg\min}\ \|W^{(t)}x\|_1 + \frac{1}{2}\|Ax - y\|_2^2$;

4: **for** $i = 1, 2, \cdots, n$ **do**

5: $w_i^{(t+1)} = \frac{1}{|x_i^{(t)}|+\delta}$; ▷ Update weights.

6: **end for**

7: $t = t + 1$;

8: **end while**

9: **return** last $x^{(t)} \in \mathbb{R}^n$.

- (Wlp method) For $p \in (0, 1)$ and $\delta > 0$,

$$w_i^{(t+1)} = \frac{1}{(|x_i^{(t)}| + \delta)^{(1-p)}},\ i = 1,\ 2, \cdots, n.$$

- (NW2 method) For $\delta > 0$ and p, $q \in (0, 1)$,

$$w_i^{(t+1)} = \frac{q + (|x_i^{(t)}| + \delta)^{1-q}}{(|x_i^{(t)}| + \delta)^{1-q}\left[|x_i^{(t)}| + \delta + (|x_i^{(t)}| + \delta)^q\right]^{1-p}},\ i = 1,\ 2, \cdots, n.$$

- (NW3 method) For $\delta > 0$ and p, $q \in (0, 1)$,

$$w_i^{(t+1)} = \frac{1 + 2(|x_i^{(t)}| + \delta)}{\left[|x_i^{(t)}| + \delta + (|x_i^{(t)}| + \delta)^2\right]^{1-p}},\ i = 1,\ 2, \cdots, n.$$

- (NW4 Method) For $p \in (0, \infty)$ and $\delta > 0$,

$$w_i^{(t+1)} = \frac{1 + (|x_i^{(t)}| + \delta)^p}{(|x_i^{(t)}| + \delta)^{p+1}},\ i = 1,\ 2, \cdots, n.$$

For a detailed theoretical discussion and convergence results of the above reweighting schemes, the reader can explore [54, 56], and the references therein. We will extend these reweighting schemes to solve the Sylvester type LASSO problem in Chapter 4.

While performing the numerical simulation for the LASSO problem, parameter tuning is one of the biggest issue. To find the right parameter value for λ, cross-validation techniques are suggested. But, if we have a bound for λ for which the output of the standard LASSO problem is a zero vector, then with the help of that bound we can scale the value of λ down to get the idea of sparsity level in the solution. In [38, §11.1.1-page 88] and [21, page 24 (exercise 2.1)], the largest possible value of parameter λ for which LASSO will return a zero vector as the solution is provided. We present the proof in Theorem 2.6.1 for the sake of clarity.

Theorem 2.6.1. *Let $A \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. If $\lambda > \|A^T b\|_\infty$, then the minimizer of the LASSO problem*

$$\min_{x\in\mathbb{R}^n} f(x) \quad \textit{with} \quad f(x) = \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1 \tag{2.46}$$

is $x_* = 0$.

Proof. The problem (2.46) is a convex problem. Its minimizer x_* satisfies

$$0 \ni \partial f(x_*), \quad \text{the subgradient of } f \text{ at } x_*. \tag{2.47}$$

We have

$$\partial f(x_*) = \left\{ A^T(Ax_* - b) + \lambda s \,:\, s_i \begin{cases} = 1 & \text{if } (x_*)_i > 0, \\ \in [-1, 1] & \text{if } (x_*)_i = 0, \\ = -1 & \text{if } (x_*)_i < 0. \end{cases} \right\}. \tag{2.48}$$

The condition (2.47) implies that there is an s_* whose entries $(s_*)_i$ are as described in (2.48) such that

$$A^T(Ax_* - b) + \lambda s_* = 0 \quad \Rightarrow \quad A^T A x_* + \lambda s_* - A^T b = 0,$$

pre-multiplying which by x_*^T gives

$$x_*^T A^T A x_* + \lambda x_*^T s_* - x_*^T A^T b = 0. \tag{2.49}$$

It can be seen that $x_*^T s_* = \|x_*\|_1$ and thus

$$\begin{aligned} \lambda x_*^T s_* - x_*^T A^T b &\geq \lambda \|x_*\|_1 - \|x_*^T\|_\infty \|A^T b\|_\infty \\ &= \|x_*\|_1 (\lambda - \|A^T b\|_\infty) \\ &\geq 0 \end{aligned}$$

because $\|x_*\|_1 \geq 0$ and $\lambda > \|A^T b\|_\infty$. On the other hand, $x_*^T A^T A x_* \geq 0$ because $A^T A$ is positive semi-definite. Therefore, in light of (2.49), we must have

$$\|x_*\|_1 (\lambda - \|A^T b\|_\infty) = 0 \quad \Rightarrow \quad \|x_*\|_1 = 0,$$

as expected. □

We use this threshold for parameter λ to perform numerical simulations in the following chapters.

CHAPTER 3

APPLICATION OF ℓ_1-MINIMIZATION TO EEG BRAIN SOURCE LOCALIZATION PROBLEM.

3.1 Introduction

The Electroencephalography(EEG) procedure measures the real-time electric potentials of brain cells caused by activation of the neurons. The procedure is used to diagnostically detect a potential disorder in the brain. Examples of EEG applications in clinical use include real-time monitoring of patients' sleep apnea [57, 58], detection and prediction of epilepsy seizures [59,60], depth of anesthesia, coma, encephalopathies, and brain death [61], etc. EEG electrodes measure the electric activities of the brain from the scalp surface instead of directly measuring the active neurons in the brain. However, it does not provide information about conclusive locations and distributions of the related activated sources, which are of interest to the neuroscience community. The problem of localizing neural activities in the cortical surface with the help of recorded EEG signals is referred to as the source imaging (ESI) which is inherently an "ill-posed" linear inverse problem because the number of the potential brain sources is larger than the number of EEG recording sensors placed on the scalp, which implies that the different neural activity patterns on source space could result in the same electromagnetic field measurements.

The EEG source localization problem is highly "ill-posed" and has infinitely many solutions. Additional regularization terms are needed to determine an appropriate solution. Several studies have been done regarding the possible regularizations in this context. The minimum norm estimate (MNE) [62] imposes ℓ_2-norm penalty and achieves a unique solution. There are variants of ℓ_2-norm penalties methods such as the dynamic statistical parametric mapping (dSPM) [63] and the standardized low-resolution brain electromagnetic tomography (sLORETA) [64]. Some of these methods also use the combination of other methods such as the weighted minimum norm-LORETA (WMN-LORETA) [65] which combines the LORETA method and weighted minimum norm. The ℓ_2-norm based methods estimate the over-diffuse source reconstruction. The minimum current estimate (MCE) [66] is introduced with ℓ_1 penalty for sparse source reconstruction which overcomes the overestimation of active area sizes obtained in ℓ_2-norm based methods.

The previously mentioned source localization methods estimate the source locations at each time point independently. A number of other regularization techniques are studied to promote temporal smoothness. One of the popularly used methods is the mixed-norm estimate (MxNE) [4]. This method uses two-level ℓ_1/ℓ_2 to achieve smooth temporal estimates, and three-level mixed-norm to promote spatially non-overlapping sources between different experimental conditions. The time-frequency mixed-norm estimate (TF-MxNE) [67] uses structured sparse priors in the time-frequency domain for a better estimation of the non-stationary and transient source signal. The graph regularized EEG source imaging with in-class consistency and out-class discrimination [68] is proposed to utilize the label information of the different brain states and understand the source localization problem in a supervised way. Liu, Wang, Rosenberger, Lou, and Quin proposed a task-related EEG source localization via the graph regularized low-rank representation model [69] to characterize the low-rank structure of the source activation. Recently, Wang, Liu, Lou, Li, and Purdon [70] proposed a probabilistic structure learning for EEG/MEG source imaging with a hierarchical graph prior to characterize the denoised micro-states and the manifold structure of the source activation pattern of the brain. This study effectively deals with the bilevel noises existing in sources and channels.

We further discuss the source localization process based on MCE method in the following section. We apply the ℓ_1 optimization methods discussed in Chapter 2 to find the solution of the EEG inverse problem.

3.2 The EEG Inverse Problem

In this section, we describe the EEG inverse problem mathematically and present the results of source localization using the current state-of-the-art algorithms. The EEG recording model can be represented in the following linear equation:

$$X = LS + N, \tag{3.1}$$

where $X \in \mathbb{R}^{m\times k}$ is the EEG measured signal for the set of m electrodes/EEG channels at k time points, $L \in \mathbb{R}^{m\times n}$ is the lead field matrix that maps the brain source signal to sensors on the scalp, $S \in \mathbb{R}^{n\times k}$ is a source matrix that represents the electrical activity at n locations and the k time points in the brain, and $N \in \mathbb{R}^{m\times k}$ represents the noise on signal acquisition from each of the m EEG electrodes at the k time points. In Figure 3.1, we present the brain model, where each triangle represents the source space (brain voxel) and the number of the triangles is equal to n.

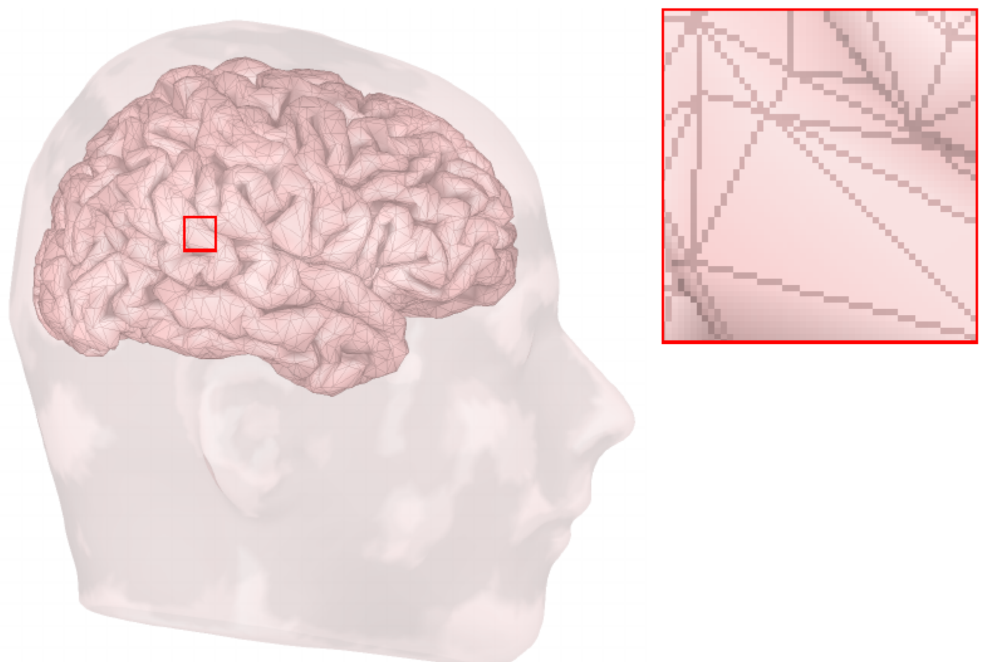
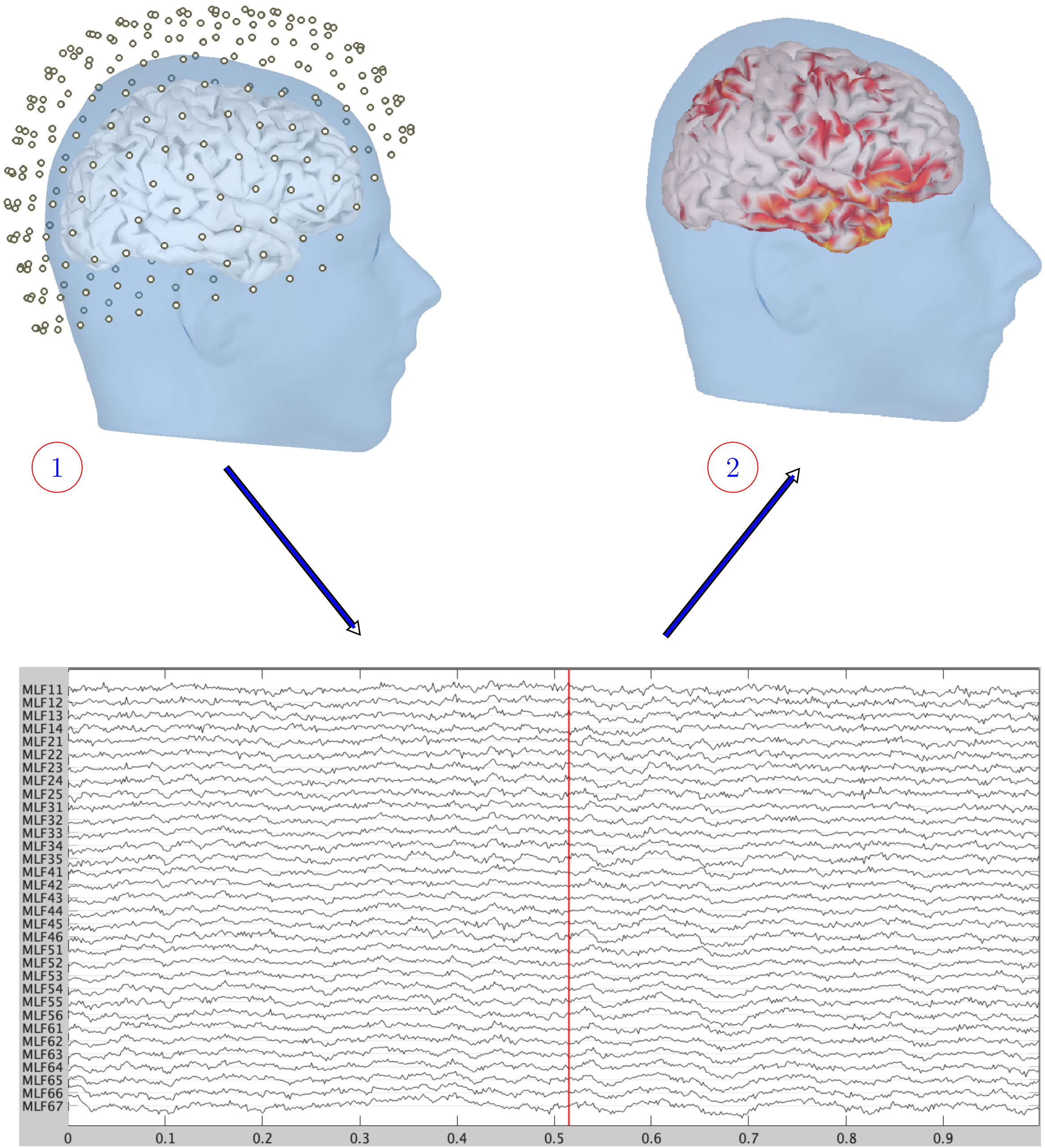


Figure 3.1: Brain model with cortical surface represented by triangular mesh where each triangle represents the brain voxel. We consider that each current dipole is located at the center of the triangular mesh and orientation of the dipole is perpendicular to the cortical surface.

The lead field matrix (or gain matrix) plays an important role. Each column of the lead field matrix represents the brain activation pattern of the particular source to the corresponding electrode. We obtain the lead field matrix by solving the Maxwell's equation. More technical details can be found in [71, 72]. In Figure 3.2, we illustrate the process of EEG source localization visually.



EEG signals for a given time unit.

Figure 3.2: In EEG inverse problem, we set up the EEG electrodes on the scalp as shown in figure ① from which we record the signal data X. With the help of recorded signal information X and lead field matrix L, our task is to determine the activated source location S as shown in ②.

The lead field matrix L has fewer rows than the columns, which makes the source localization problem highly "ill-posed". We use the ℓ_1-norm penalty as a regularization to estimate an appropriate solution. The source estimation problem under the ℓ_1 penalty is a convex optimization problem, which is expressed as

$$\underset{S\in\mathbb{R}^{n\times k}}{\arg\min}\ \|X - LS\|_F^2 + \lambda\|S\|_{1,1}, \tag{3.2}$$

where $\lambda > 0$ is a regularization parameter. Since the problem (3.2) is decoupled, we can recover each column of S by solving the following k independent optimization problems:

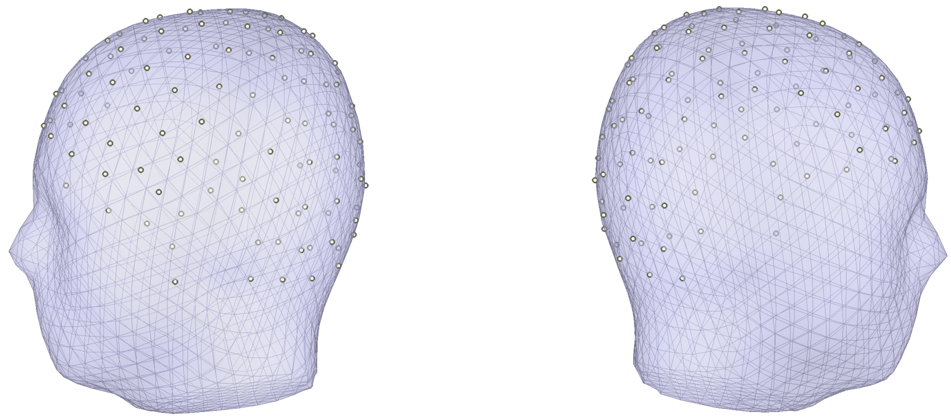
$$s_i = \underset{s_i\in\mathbb{R}^{n}}{\arg\min}\ \|x_i - Ls_i\|_2^2 + \lambda\|s_i\|_1, \tag{3.3}$$

where x_i and s_i denote the i-th column of the matrix X and S respectively. For each i, the problem in (3.3) is the standard LASSO problem.

3.3 Numerical Results

3.3.1 Experiment setup

In any brain imaging problem, we do not know the underlying ground truth. We form synthetic data to characterize the performance of the ℓ_1-minimization algorithms to solve the ℓ_1-minimization problem (3.3). We use Brainstorm [73] GUI available in MATLAB to plot the source activation results in a real head model template. Brainstorm will be used for brain structure segmentation and cortical surface reconstruction. For real head model design, we use standard ICBM 152 template for Neuroscan Quick Cap for 128 channels and edit the channel location to create the head model for 108 channels. The electrode layout on the head surface is shown in Figure 3.3.



(a) Electrode layout on the left hemisphere. (b) Electrode layout on the right hemisphere.

Figure 3.3: EEG channel layout of ICBM 152 - Neuroscan Cap 128 edited in (a) and (b) from two different sides of the head model.

In our experiment, the lead field matrix L is 108×2004, and it represents the mapping between 108 EEG electrodes (channels) and 2004 brain voxels. In reality, EEG measurements are contaminated with noise. We consider the noise in the signal acquisition process at different noise levels. The amount of noise is determined by the signal-to-noise ratio (SNR) values. The SNR [17, 68] value is defined as

$$\text{Signal to noise ratio (SNR)} = 20 \, \log_{10} \frac{\|S\|_F}{\|S - \hat{S}\|_F},$$

where S and $\hat{S}$ denote the ground truth and reconstructed source respectively. The SNR values are measured in decibels (dB). By definition, the lower SNR values represent the larger amount of noise in data. The numerical results of source reconstruction will be presented in the next section using both noisy and clean data.

3.3.2 Numerical Experiments

In this subsection, we present the numerical results of source reconstruction for EEG problem (3.3) using the benchmark algorithms described in Chapter 2.

We measure their performance in terms of run time (**cputime** in MATLAB) and reconstruction error. The reconstruction error (RE) is defined as

$$\text{Reconstruction Error(RE)} = \frac{\|\hat{S} - S\|_F}{\|S\|_F}.$$

The reconstruction estimate is considered good if the RE value is close to zero. Before describing the numerical results, we observe how quickly different ℓ_1 optimization algorithms help decay the objective function value of problem (3.3) in Figure 3.4. In this result, we use zero vector (or cold start) as an initial guess for each column recovery of source matrix S in all five algorithms, namely ADMM, ISTA, FISTA, and ℓ_1-homotopy. The weighted ℓ_1-homotopy considers the solution of ℓ_1-homotopy as an initial guess which we call a warm start. The choice of an optimal value of the parameter λ is always a demanding task and problem dependent. In this work, we first consider

$$\lambda_{\max} = \max\{\|L^T x_1\|_\infty, \cdots, \|L^T x_k\|_\infty\},$$

and then we scale down $\lambda_{\max}$ by a scaling factor $C \in (0, 1)$, i.e., $\lambda = C\lambda_{\max}$. We use a fixed value of λ for each column recovery of matrix S in all the algorithms.

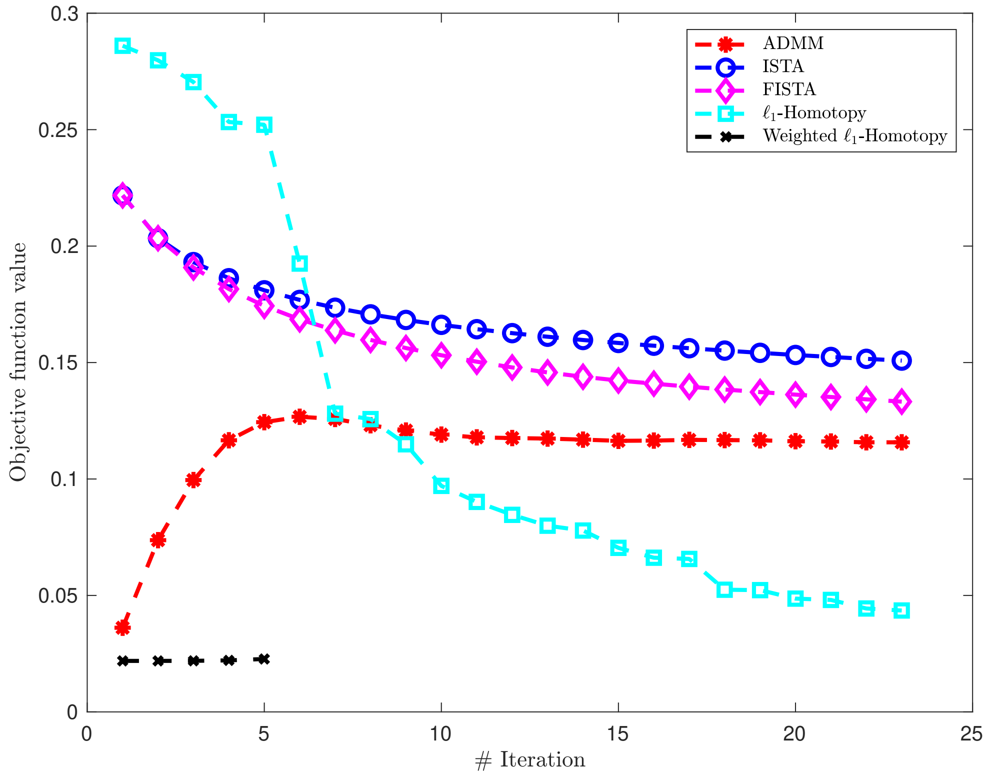


Figure 3.4: Behavior in convergence of the objective value of the problem (3.3) by different benchmark algorithms for $C = 0.1$.

From Figure 3.4, we see that the weighted ℓ_1-homotopy algorithm converges faster to the optimum in less number of iterations than the other four algorithms. The weighted ℓ_1-homotopy achieves the solution faster since it uses warm start initialization. With the cold start, ℓ_1-homotopy converges faster among all.

We present the numerical results of ℓ_1 optimization algorithms on noise-free synthetic data (SNR $= \infty$ dB) in Table 4.1. In this test, we choose regularization parameter $\lambda = C\lambda_{\max}$. The table shows the performance in source localization of the benchmark algorithm discussed in Chapter 2 in terms of reconstruction error and runtime. We use the **cputime** function in MATLAB to record the runtime of the algorithm in seconds. We use MacBook Pro PC having 16 gigabytes (GB) random access memory (RAM) with 2.7 GHz Intel Core i7 processor to run all the numerical simulations. In Table 4.1, the best performance results are highlighted.

SNR = ∞ dB (noiseless)		
Algorithms	Runtime (RT)	Reconstruction Error (RE)
ADMM	45.04	0.770588
ISTA	59.18	0.958876
FISTA	54.18	0.849367
ℓ_1-homotopy	**31.73**	**0.571420**
Weighted ℓ_1-homotopy	**18.54**	**0.553751**

Table 3.1: Results of source reconstruction by benchmark algorithms with scaling factor $C = 0.005$ for λ.

From Table 4.1, we see that the weighted ℓ_1-homotopy has superior performance on source localization in terms of RE and RT values. We use cold start for ADMM, ISTA, FISTA, and ℓ_1-homotopy but a warm start for weighted ℓ_1-homotopy which is the solution of ℓ_1-homotopy. Since the warm start for weighted ℓ_1-homotopy is already improved, it takes less time to converge to the solution and gives a better result. With the cold start, ℓ_1-homotopy has faster convergence and a better RE value. Also, FISTA converges faster than ISTA and has better reconstruction.

In the next step, we consider noise with different noise levels (30 dB, 20 dB, and 10 dB) in the channels while keeping the source noise-free. The results in Table 4.2 present the performance of the benchmark ℓ_1-minimization algorithms for noisy data. The results in Table 4.2 indicate the effect of noise in data. All of the benchmark ℓ_1 solvers have poor performance in source reconstruction compared to noise-free data. The ADMM and homotopy based algorithms have better performance in all noise

Algorithms	SNR = 30 dB		SNR = 20 dB		SNR = 10 dB	
	RT	RE	RT	RE	RT	RE
ADMM	**33.28**	**0.9767**	**33.57**	**0.9774**	**32.06**	**0.9804**
ISTA	**60.17**	0.9971	**59.12**	0.9971	**58.80**	0.9971
FISTA	61.77	0.9944	58.84	0.9944	59.99	0.9944
ℓ_1-homotopy	137.56	**0.9842**	134.17	**0.9849**	148.14	**0.9894**
Weighted ℓ_1-homotopy	75.82	0.9879	80.55	0.9886	98.60	0.9962

Table 3.2: Results of source reconstruction by benchmark algorithms in different noise levels with scaling factor $C = 0.0001$ for λ.

levels. We used the same cold start initialization and a fixed value of λ for all the five algorithms.

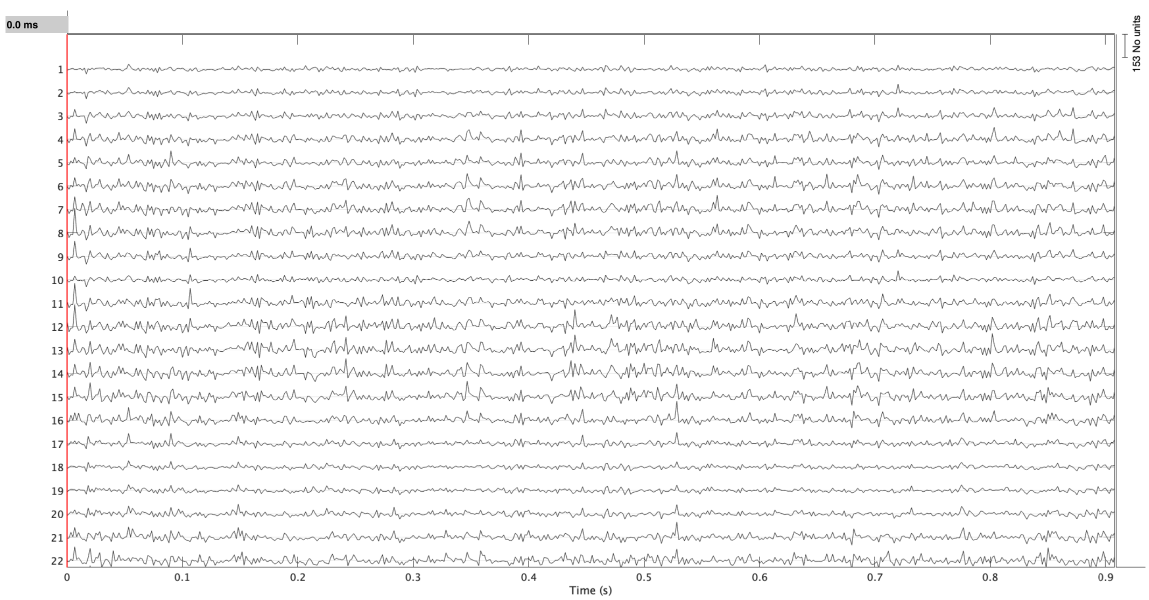
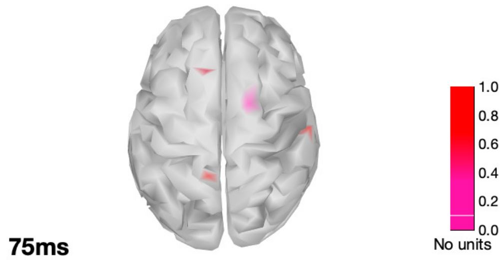
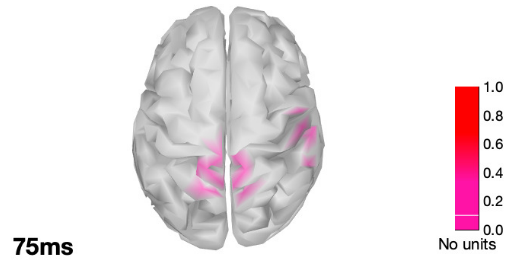


Figure 3.5: EEG signals from 22 out of 108 channels in the first 0.9 milliseconds.

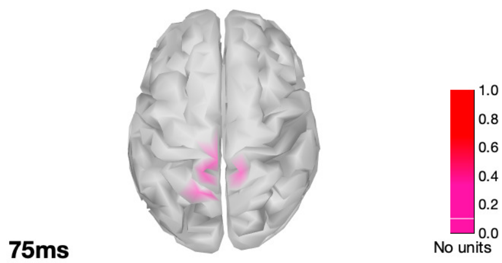
We now visualize the numerical results of the benchmark ℓ_1-minimization algorithms presented in Table 4.1 using Brainstorm GUI in MATLAB for source localization on a real head model. In Figure 3.5, we present the sample of the first 22 out of 108 channels of EEG measurements out of 108 for the first 0.9 milliseconds (ms). In Figure 3.6, we visualize the plots of source recoveries from the benchmark algorithms. In this experiment, we neglect the 10% of the smallest values from both ground truth and reconstructed solutions of all the benchmark algorithms for clear visualization. We compute solutions by widely used source localization methods such as sLORETA and dSPM using Brainstorm. All the plots in Figure 3.6 are captured for time instance at 75 ms. The first plot in Figure 3.6 (a) represents the true source or ground truth activation. The color in the cortical surface represents the strength of the activated signal around the region. We can see four different activated source locations in ground truth. We can see the highly diffused and less accurate source localization results from ℓ_2-norm based methods sLORETA and dSPM. The recovery of FISTA is more accurate compared to ISTA as evidenced by brain plots. The ℓ_1-homotopy and weighted ℓ_1-homotopy reconstruct the source location accurately.



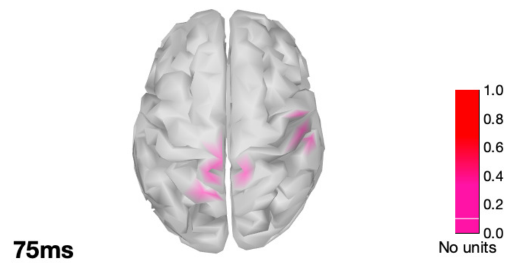
(a) Ground truth.



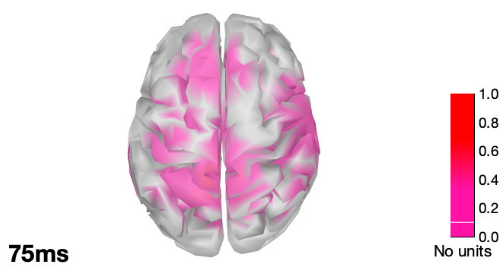
(b) Source recovery by ADMM.



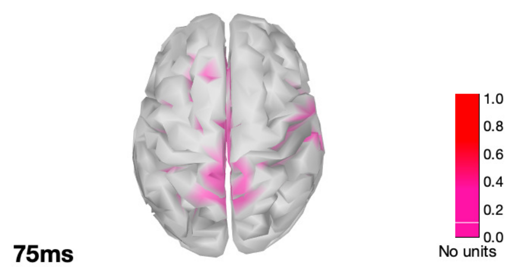
(c) Source recovery by ISTA.



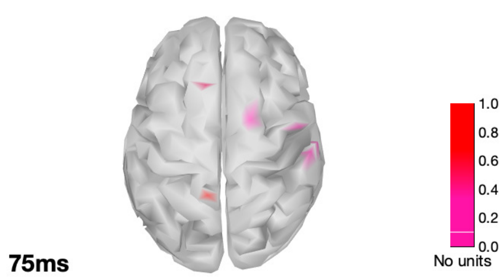
(d) Source recovery by FISTA.



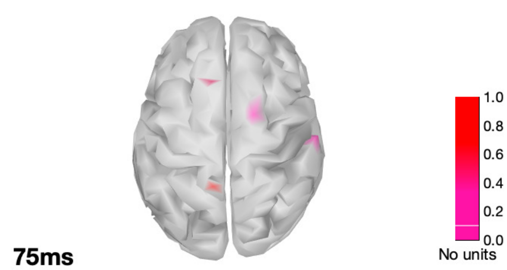
(e) Source recovery by sLORETA.



(f) Source recovery by dSPM.



(g) Source recovery by ℓ_1-homotopy.



(h) Source recovery by weighted ℓ_1-homotopy.

Figure 3.6: Source recoveries by benchmark ℓ_1-optimization algorithms against ground truth.

In summary, the ℓ_1-optimization techniques can help solving the EEG source localization problem. The homotopy based methods can solve the highly "ill-posed" inverse problem more accurately. We will use similar techniques to those explored in this chapter in the following chapters.

CHAPTER 4

SYLVESTER LASSO AND ITS APPLICATION TO EEG INVERSE PROBLEM

4.1 Introduction

In this section, we introduce the Sylvester type LASSO problem which is formulated as follows:

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|AX-B\|_F^2+\frac{1}{2}\|XC-D\|_F^2+\lambda\|X\|_{1,1}, \tag{4.1}$$

where $A\in\mathbb{R}^{q\times m}, B\in\mathbb{R}^{q\times n}, C\in\mathbb{R}^{n\times p},\ D\in\mathbb{R}^{m\times p}$, and $\lambda>0$ is a regularization parameter.

Definition 4.1.1. For any matrix $A=(a_{ij})\in\mathbb{R}^{m\times n}$, the *vec-form* of the matrix A is defined as

$$\mathrm{vec}(A)=A(:)=(a_{11},\cdots,a_{m1},a_{12},\cdots a_{m2},\cdots,a_{1n},\cdots,a_{mn})^T,$$

i.e., the entries of A are stacked columnwise to form a vector in $\mathbb{R}^{mn}$.

Definition 4.1.2. (Kronecker product) The *Kronecker product* of $A\in\mathbb{R}^{m\times n}$ and $B\in\mathbb{R}^{p\times q}$ is defined as

$$A\otimes B=\begin{pmatrix} a_{11}B & \cdots & a_{1m}B\\ \vdots & \ddots & \vdots\\ a_{m1}B & \cdots & a_{mn}B\end{pmatrix}\in\mathbb{R}^{mp\times nq}.$$

The problem (4.1) can be rewritten in the vector form as

$$\min_{X(:)\in\mathbb{R}^{mn}} \frac{1}{2}\|(I_n\otimes A)X(:)-B(:)\|_2^2+\frac{1}{2}\|(C^T\otimes I_m)X(:)-D(:)\|_2^2+\lambda\|X(:)\|_1. \tag{4.2}$$

Define,

$$\mathcal{M}=\begin{bmatrix}(I_n\otimes A)\\ (C^T\otimes I_m)\end{bmatrix}\in\mathbb{R}^{(nq+pm)\times mn},\mathcal{V}=\begin{bmatrix}B(:)\\ D(:)\end{bmatrix}\in\mathbb{R}^{(nq+pm)}.$$

Problem (4.2) can be reformulated as

$$\min_{X(:)\in\mathbb{R}^{mn}} \frac{1}{2}\|\mathcal{M}X(:)-\mathcal{V}\|_2^2+\lambda\|X(:)\|_1, \tag{4.3}$$

which is now a standard LASSO problem. In the next section, we will describe a Sylvester LASSO to solve the EEG inverse problem under a specific neurophysiological assumption.

4.2 Application of Sylvester LASSO to EEG Inverse Problem

The EEG recording signals as described in Chapter 3 can be represented in the following linear equation:

$$B = AX + N, \tag{4.4}$$

where $B \in \mathbb{R}^{q\times n}$ is the EEG measured signal for the set of q electrodes/EEG channels in n time points, $A \in \mathbb{R}^{q\times m}$ is the lead field matrix that maps the m sources of the brain to q sensors on the scalp, $X \in \mathbb{R}^{m\times n}$ is a source matrix that represents the electrical activity at m sources of the brain in n time points, and $N \in \mathbb{R}^{q\times n}$ represents the noise on signal acquisition from each of the q EEG electrodes in n time points. In Chapter 3, we discussed the solution of the “ill-posed” inverse problem (4.4), under the assumption that the brain source activation is sparse. Total variation minimization of the source signals can be a feasible neurophysiological assumption of the EEG inverse problem. The total variation (TV) of the brain sources in n time points is expressed as,

$$\mathrm{TV(X)} \;=\; \sum_{i=1}^{n-1}\|x_{i+1}-x_i\|_2^2 = \|XC\|_F^2,$$

where x_i denotes the i-th column of matrix X, and $C = \begin{pmatrix} -1 & 0 & 0 & \dots & 0 \\ 1 & -1 & 0 & \dots & 0 \\ 0 & 1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -1 \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$.

Problem (4.4) for estimating the source X with both sparsity and TV minimization assumptions is expressed as the following optimization problem

$$\min_{X \in \mathbb{R}^{m \times n}} \|AX - B\|_F^2 + \|XC\|_F^2 + \lambda \|X\|_{1,1}. \tag{4.5}$$

Problem (4.5) is a particular case (when $D = 0$) of the Sylvester type LASSO problem (4.1). In the next section, we will discuss computational challenges for solving problem (4.1) and numerical techniques for handling those challenges.

4.3 Pre-processing

In this section, we discuss the process of solving problem (4.3) without explicitly forming it. The matrix $\mathcal{M}$ has a blockwise non-zero structure in real applications, it is too big to store on personal computers with limited memory. For example, for $A \in \mathbb{R}^{108 \times 2004}, B \in \mathbb{R}^{108 \times 600}, C \in \mathbb{R}^{600 \times 300}$, and $D \in \mathbb{R}^{2004 \times 300}$, $\mathcal{M} \in \mathbb{R}^{666{,}000 \times 1{,}202{,}400}$ consumes 6,406 gigabytes of memory in double precision. Applying any ℓ_1 solvers straightforwardly would be infeasible. We must exploit the structure of $\mathcal{M}$. In this regard, we keep it in the background without forming it explicitly and perform all computations based on input matrices A, B, C, and D, which are easy to handle. Many operations need to be performed with the matrix $\mathcal{M}$ while performing any ℓ_1 solvers like ADMM, ISTA, FISTA, or ℓ_1-homotopy algorithm as discussed in Chapter

2. We now explain how we formulate and implement the algorithms without forming $\mathcal{M}$ explicitly.

- Matrix-vector product with $\mathcal{M}$. Let $Y(:) \in \mathbb{R}^{mn}$ be any vector for which we want to compute the product with $\mathcal{M}$, i.e., $\mathcal{M} \times Y(:)$. We have

$$\begin{aligned}\mathcal{M} \times Y(:) &= \begin{bmatrix} (I_n \otimes A) \\ \\ (C^T \otimes I_m) \end{bmatrix} \times Y(:) \\ &= \begin{bmatrix} (I_n \otimes A)Y(:) \\ \\ (C^T \otimes I_m)Y(:) \end{bmatrix} \\ &= \begin{bmatrix} (AY)(:) \\ \\ (YC)(:) \end{bmatrix}.\end{aligned}$$

- Matrix-vector product with $\mathcal{M}^T$. Suppose $p = \left[\begin{smallmatrix} p_1 \\ p_2 \end{smallmatrix}\right] \in \mathbb{R}^{(nq+pm)}$ is any vector for which we need to compute the product $\mathcal{M}^T \times p$, where $p_1 \in \mathbb{R}^{nq}$ and $p_2 \in \mathbb{R}^{pm}$. We have

$$\begin{aligned}\mathcal{M}^T \times p &= \begin{bmatrix} (I_n \otimes A^T) & (C \otimes I_m) \end{bmatrix} \times p \\ &= \begin{bmatrix} (I_n \otimes A^T) & (C \otimes I_m) \end{bmatrix} \begin{bmatrix} p_1 \\ \\ p_2 \end{bmatrix} \\ &= (I_n \otimes A^T)p_1 + (C \otimes I_m)p_2 \\ &= (A^T P_1 + P_2 C^T)(:),\end{aligned}$$

where $P_1 = \textbf{reshape}(p_1, q, n) \in \mathbb{R}^{q \times n}$, $P_2 = \textbf{reshape}(p_2, m, p) \in \mathbb{R}^{m \times p}$ are matrices when p_1 and p_2 are folded back respectively.

4.4 Algorithms for Sylvester type LASSO Problem

In this section, we discuss how to adapt the benchmark ℓ_1 optimization algorithms to efficiently solve (4.3). We will present ADMM and FISTA in detail.

4.4.1 ADMM

As discussed earlier, solving (4.1) is equivalent to solving (4.3). To apply ADMM, we reformulate problem (4.1) as

$$\begin{aligned} \text{minimize} \quad & f(X) + g(Z) \\ \text{subject to} \quad & X - Z = 0, \end{aligned} \tag{4.6}$$

where $f(X) = \frac{1}{2}\|AX - B\|_F^2 + \frac{1}{2}\|XC - D\|_F^2$ and $g(Z) = \lambda\|Z\|_{1,1}$. The augmented Lagrangian function of problem (4.6) is:

$$L_\rho(X, Y, Z) = \frac{1}{2}\|AX - B\|_F^2 + \frac{1}{2}\|XC - D\|_F^2 + \lambda\|Z\|_{1,1} + \langle Y, (X - Z)\rangle_F + \frac{\rho}{2}\|X - Z\|_F^2,$$

where $\langle \cdot, \cdot \rangle_F$ denotes the inner product of matrices as defined in Definition 4.4.1 below.

Definition 4.4.1 (Matrix inner product)**.** For $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{m\times n}$, the *matrix inner product* between them is defined as

$$\langle A, B\rangle_F = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}^* B_{ij} = \text{trace}(A^* B),$$

where A^* is the complex conjugate transpose of A.

The ADMM algorithm consists of two primal updates (X- and Z-updates) and dual variable update (Y-update). We start from the X-update which is to find the minimizer of the augmented Lagrangian over X while keeping Y and Z constants:

$$\min_{X\in\mathbb{R}^{m\times n}} L_\rho(X, Y, Z).$$

To obtain the closed form solution of the X-update rule, we set $\nabla_X L_\rho(X, Y, Z) = 0$ to get

$$A^T(AX - B) + (XC - D)C^T + U + \rho(X - Z) = 0$$

$$\implies \qquad PX + XQ = K + \rho(Z - U), \tag{4.7}$$

where $U = \frac{1}{\rho}Y$, $P = (A^TA + \rho I_m)$, $Q = CC^T$, and $K = A^TB + DC^T$ with $I_m \in \mathbb{R}^{m\times m}$ is the identity matrix. The X-update rule for given $Y^{(k)}$ and $Z^{(k)}$ is thus expressed as

$$PX^{(k+1)} + X^{(k+1)}Q = K + \rho(Z^{(k)} - U^{(k)}). \tag{4.8}$$

We need to solve the Sylvester equation (4.8) to find the solution of X-update. Bartels and Stewart [74] proposed a numerically stable method known as the Bartels-Stewart algorithm to solve the Sylvester equation. Following this work, Golub, Nash, and Loan [75] introduced an improved version of the Bartels-Stewart algorithm known as the Hessenberg-Schur algorithm to solve the Sylvester equation. We use the MATLAB built-in function **sylvester** to solve (4.8) which uses the Hessenberg-Schur variant of the Bartels-Stewart algorithm. The reader is referred to [74–76], and references therein for a detailed discussion.

The Z-update rule is expressed as

$$\begin{aligned} Z^{(k+1)} &= \underset{Z\in\mathbb{R}^{m\times n}}{\arg\min}\; L_\rho(X^{(k+1)}, Y^{(k)}, Z) \\ &= \underset{Z\in\mathbb{R}^{m\times n}}{\arg\min}\; \left\{\lambda\|Z\|_{1,1} + \frac{\rho}{2}\|Z - (X^{(k+1)} + U^{(k)})\|_F^2\right\} \\ &= \text{shrink}\left(X^{(k+1)} + U^{(k)}, \frac{\lambda}{\rho}\right), \end{aligned} \tag{4.9}$$

where $\text{shrink}(A, \alpha) = \text{sign}(a_{ij})(|a_{ij}| - \alpha)_+$ for $A = (a_{ij}) \in \mathbb{R}^{m\times n}$.

Finally, the dual update (U-update) rule is described as

$$U^{(k+1)} = U^{(k)} + X^{(k+1)} - Z^{(k+1)}. \tag{4.10}$$

We determine the stopping criteria for ADMM by the primal and dual residuals. In Sylvester type LASSO problem, the primal and dual residuals at the k-th step are defined as

$$R^{(k)} = X^{(k)} - Z^{(k)},$$

$$S^{(k)} \;=\; -\rho\left(Z^{(k)} - Z^{(k-1)}\right),$$

respectively. A reasonable stopping criterion is that the primal and dual residuals are small, i.e.,

$$\|R^{(k)}\|_F \le \epsilon^{\text{pri}} \quad \text{and} \quad \|S^{(k)}\|_F \le \epsilon^{\text{dual}}, \tag{4.11}$$

where $\epsilon^{\text{pri}} > 0$ and $\epsilon^{\text{dual}} > 0$ are feasibility tolerances for the primal and dual feasibility conditions, respectively. These tolerances are chosen using an absolute and relative criterion, such as

$$\begin{aligned} \epsilon^{\text{pri}} &= \sqrt{mn}\,\epsilon^{\text{abs}} + \epsilon^{\text{rel}} \max\left\{\|X^{(k)}\|_F, \|Z^{(k)}\|_F\right\}, \\ \epsilon^{\text{dual}} &= \sqrt{mn}\,\epsilon^{\text{abs}} + \epsilon^{\text{rel}}\|\rho\, U^{(k)}\|_F, \end{aligned}$$

where $\epsilon^{\text{abs}} > 0$ and $\epsilon^{\text{rel}} > 0$ are absolute and relative tolerances, respectively. The reader is referred to [38, §3.3.1] for theoretical justification of the above stopping criteria.

We summarize the ADMM algorithm for solving Sylvester type LASSO (4.1) in Algorithm 4.1.

Algorithm 4.1 ADMM Algorithm for Sylvester type LASSO Problem (4.1)

Input: $A \in \mathbb{R}^{q\times m}$, $B \in \mathbb{R}^{q\times n}$, $C \in \mathbb{R}^{n\times p}$, $D \in \mathbb{R}^{m\times p}$, $\lambda > 0$, $\rho > 0$, absolute tolerance $\epsilon^{\text{abs}} > 0$, and relative tolerance $\epsilon^{\text{rel}} > 0$;

Output: a solution of problem (4.1).

1: **Initialize:** $U^{(0)}, Z^{(0)} \in \mathbb{R}^{m\times n}$ as zero matrices, and $k = 0$;

2: **while** the stopping criteria (4.11) are not satisfied **do**

3: $\quad$ X-update as described in (4.8);

4: $\quad$ Z-update as described in (4.9);

5: $\quad$ U-update as described in (4.10);

6: $\quad$ $k = k + 1$;

7: **end while**

8: **return** last $Z^{(k)} \in \mathbb{R}^{m\times n}$.

4.4.2 FISTA

Let $H(X) = f(X) + g(X)$, where $f(X) = \frac{1}{2}\|AX - B\|_F^2 + \frac{1}{2}\|XC - D\|_F^2$, and $g(X) = \lambda\|X\|_{1,1}$. For $L > 0$, the quadratic approximation of $H(X)$ at Y can be written as

$$Q_L(X, Y) = f(Y) + \langle X - Y, \nabla f(Y)\rangle + \frac{L}{2}\|X - Y\|_F^2 + g(X). \tag{4.12}$$

The minimizer of (4.12) over X can be expressed as

$$\begin{aligned} P_L(Y) &= \arg\min_X \left\{ g(X) + \frac{L}{2}\left\| X - \left(Y - \frac{1}{L}\nabla f(Y)\right)\right\|_F^2 \right\} \\ &= \text{shrink}\left(Y - \frac{1}{L}\nabla f(Y), \frac{\lambda}{L}\right). \end{aligned}$$

Algorithm 4.2 below describes the process for solving the Sylvester type LASSO (4.3) using FISTA with backtracking.

Algorithm 4.2 FISTA with backtracking for Sylvester type LASSO.

Input: $A \in \mathbb{R}^{q\times m}$, $B \in \mathbb{R}^{q\times n}$, $C \in \mathbb{R}^{n\times p}$, $D \in \mathbb{R}^{m\times p}$, $\lambda > 0$, $\eta > 1$, and tolerance ϵ;

Output: a solution of problem (4.1).

Initialize: $Y^{(1)}, X^{(0)} \in \mathbb{R}^{m\times n}$ as zero matrices, $L^{(0)} > 1$, $t^{(1)} = 1$, and $k = 1$;

while $|H(X^{(k)}) - H(X^{(k-1)})| > \epsilon$ **do**

  Find the smallest nonnegative integers i_k such that with $\bar{L} = \eta^{i_k} L^{(k-1)}$

$$H(P_{\bar{L}}(Y^{(k)})) \leq Q_{\bar{L}}(P_{\bar{L}}(Y^{(k)}), Y^{(k)});$$ ▷ Backtracking line search step.

  Set $L^{(k)} = \eta^{i_k} L^{(k-1)}$ and compute ▷ Updating solution.

$$X^{(k)} = P_{L^{(k)}}(Y^{(k)}),$$
$$t^{(k+1)} = \frac{1+\sqrt{1+4(t^{(k)})^2}}{2},$$
$$Y^{(k+1)} = X^{(k)} + \left(\frac{t^{(k)}-1}{t^{(k+1)}}\right)(X^{(k)} - X^{(k-1)});$$

  $k = k + 1$;

end while

return last $X^{(k)} \in \mathbb{R}^{m\times n}$.

4.5 Iterative Reweighting for Sylvester FISTA

The major difficulty for any parameter dependent optimization model is to determine the best parameter for the model. We are solving ℓ_1-minimization problem (4.1) which is dependent on a single parameter value λ. Different values of regularization parameter λ changes the sparsity level of recovery. Instead of working on a single regularization parameter λ, we assign different weights to the elements to control the sparsity. This helps us to keep the recovery in a meaningful direction. We solve the weighted ℓ_1 problem with a dynamic ℓ_1 penalty as

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|AX - B\|_F^2 + \frac{1}{2}\|XC - D\|_F^2 + \|W \odot X\|_{1,1}, \tag{4.13}$$

where $W \in \mathbb{R}^{m \times n}$ is a matrix of positive weights, and $A \odot B$ denotes the Hadamard product of the matrices A and B as defined in Definition 4.5.1.

Definition 4.5.1 (Hadamard product [77]). The *Hadamard product* of two matrices $A = [a_{ij}]$ and $B = [b_{ij}]$ of the same size is defined as entrywise product of A and B, i.e.,

$$A \odot B = [a_{ij} b_{ij}].$$

In section 2.6 of Chapter 2, we discussed the iterative reweighting technique to solve weighted ℓ_1-minimization problem under different weight modification schemes. We adopt a similar idea to solve the weighted Sylvester type LASSO problem (4.13). In iterative reweighting, we have to solve a series of weighted ℓ_1-minimization problems using FISTA because of its convergence speed. We summarize the process of the iterative reweighting to solve (4.13) in Algorithm 4.3, where we choose NW4 weight scheme to update the weights in step 5.

Algorithm 4.3 Iterative Reweighting for Sylvester type LASSO

Input: $A \in \mathbb{R}^{q\times m}, B \in \mathbb{R}^{q\times n}, C \in \mathbb{R}^{n\times p},$ and $D \in \mathbb{R}^{m\times p}$, $\delta > 0$, $p \in (0,1)$, maximum iteration $N_{\max}$;

Output: improved solution of problem (4.13).

Initialize : $k = 0,\ W^{(0)} = \left(w_{ij}^{(0)}\right) = 1$ for all i, j ;
while $k \le N_{\max}$ **do**
  Solve the weighted ℓ_1-minimization problem (4.13) using Algorithm 4.2:
  $$X^{(k)} = \underset{X\in\mathbb{R}^{m\times n}}{\arg\min}\ \tfrac{1}{2}\|AX - B\|_F^2 + \tfrac{1}{2}\|XC - D\|_F^2 + \|W^{(k)} \odot X\|_{1,1};$$
  for $i = 1, 2, \cdots, m$ **do**
    for $j = 1, 2, \cdots, n$ **do**
      $w_{ij}^{(k+1)} = \frac{1+(|x_{ij}^{(k)}|+\delta)^p}{(|x_{ij}^{(k)}|+\delta)^{p+1}}$; ▷ Update weights (NW4 weighted scheme) .
    end for
  end for
  $k = k + 1$;
end while
return last $X^{(k)} \in \mathbb{R}^{m\times n}$.

4.6 Numerical Experiments

4.6.1 Experimental setting

In this subsection, we solve the problem (4.1) with synthetic data under different noise levels determined by signal-to-noise ratio (SNR). We generate the random input matrices $A \in \mathbb{R}^{108\times 2004}, B \in \mathbb{R}^{108\times 600}, C \in \mathbb{R}^{600\times 300}$ and $D \in \mathbb{R}^{2004\times 300}$ to observe the performance of the proposed algorithm. These matrices consist of the values drawn from the standard uniform distribution on (0,1) using the **rand** function in MATLAB. We discussed in section 4.3, $\mathcal{M} \in \mathbb{R}^{666{,}000\times 1{,}202{,}400}$ is too big to be stored on a regular laptop or personal computer. We perform all numerical computations without forming $\mathcal{M}$. We calculate the following quantities as error measurement metrics to evaluate the performance of Algorithm 4.3.

- Sparsity Ratio: This is one of the important measures to compare the sparsity of recovery defined as

$$\text{Sparsity Ratio (SR)} = \frac{\text{Nonzero entries of the recovered solution}}{\text{Total entries of the solution}}.$$

- Runtime (RT): We check the total CPU time taken by an algorithm to solve the problem. We use the **cputime** function of MATLAB to record the execution time.

- Reconstruction Error (RE): For S and $\hat{S}$, the exact and reconstructed solutions, respectively, we define

$$\text{RE} = \frac{\|S - \hat{S}\|_F}{\|S\|_F}.$$

4.6.2 Numerical Results

In this subsection, we provide the numerical solutions of problem (4.3) with and without the reweighting technique. We first discuss the convergence behavior of the solvers for solving (4.13).

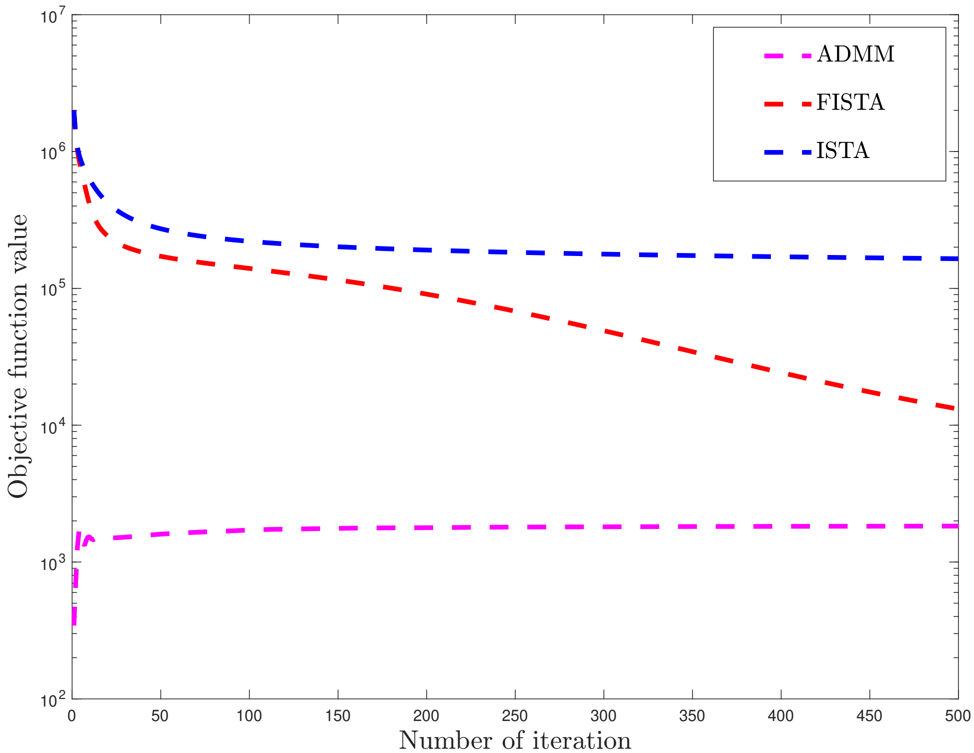


Figure 4.1: Decay of the objective function values in successive iterations when $\lambda = 1$. In ADMM, the objective function value at the k-th iteration is referred to $f(X^{(k)}) + g(Z^{(k)})$ whereas $f(X^{(k)}) + g(X^{(k)})$ in ISTA and FISTA.

In Figure 4.1, we plot objective function values evaluated at updated solution in each iteration. As we can see in Figure 4.1, FISTA has a faster convergence than ISTA, as expected. However, both ISTA and FISTA are slower than ADMM which quickly converges to the global minimum solution in fewer than twenty steps.

In Figure 4.2, we plot the quantity $|H(X^{(k)}) - H(X^{*})|$ against successive iteration with regularization parameter $\lambda = 1$ in all three algorithms, where $H(X^{(k)}) = f(X^{(k)}) + g(X^{(k)})$ in ISTA and FISTA, and $H(X^{(k)}) = f(X^{(k)}) + g(Z^{(k)})$ in ADMM. It is clear from Figure 4.2 that ISTA has the poorest performance, FISTA progresses slowly while ADMM converges to the solution, the fastest.

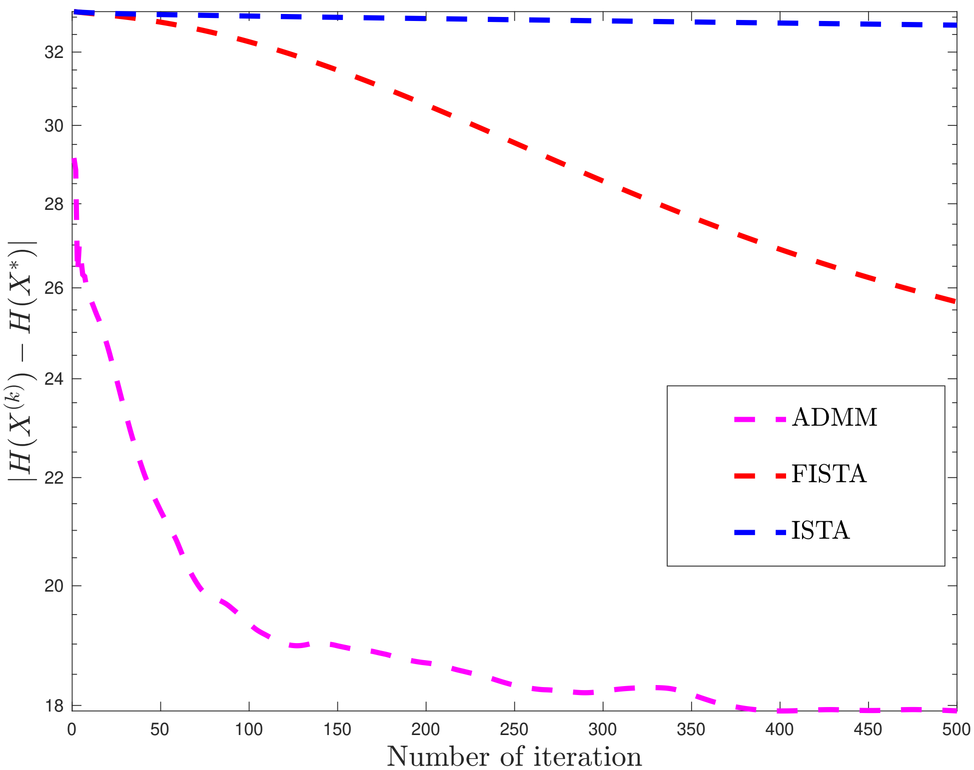


Figure 4.2: Convergence of the three algorithms for solving the Sylvester type LASSO problem (4.13).

We present numerical simulation results for problem (4.3) with a fixed value of λ on simulated data without noise. For the synthetic data, the ground truth or reference solution has SR value 0.0040 which means the matrix that we are interested in recovering has only 4% non-zero entries. While comparing the performance of the algorithms regarding the sparsity recovery, SR close to 0.0040 is considered good. In Table 4.1, we present the performance of ADMM, ISTA, and FISTA in terms of RE, SR, and RT values. ADMM has the best performance in solution recovery in terms of RE and SR among the three algorithms.

SNR = ∞ dB (noiseless)			
Algorithms	RE	SR	RT (sec.)
ADMM	**0.5572**	**0.0209**	514.06
ISTA	0.9792	0.9931	**420.48**
FISTA	0.6855	0.8423	446.95

Table 4.1: Recovery results for problem (4.3) with $\lambda = 1$.

We further explore the recovery results of the algorithms by adding noise to the measurements with two different levels of noise at SNR values 30 and 20 dB. We choose $\lambda = 1$ for all three algorithms. ADMM performs the best in solution recovery even in noisy data compared to the ISTA and FISTA. In terms of running time, ADMM is slowest.

Algorithms	SNR = 30 dB			SNR = 20 dB		
	RE	SR	RT (sec.)	RE	SR	RT (sec.)
ADMM	**0.5559**	**0.0491**	584.92	**0.6144**	**0.24211**	531.14
ISTA	0.9792	0.9936	**422.22**	0.9792	0.9952	**410.40**
FISTA	0.6860	0.8662	436.79	0.6914	0.9313	424.44

Table 4.2: Recovery results for (4.3) with $\lambda = 1$ and two different levels of noise in measurements.

The SR values in both situations presented in Table 4.1 and Table 4.2 suggest that the sparsity recoveries by ISTA and FISTA are almost dense with $\lambda = 1$. It is difficult to choose which model validation measurements are the best to judge the quality of the recovery. If we wish to recover a more sparse solution by increasing λ, then reconstruction error may get worse. To investigate this situation, we use the same penalty for ADMM as we used for previous results in Table 4.2 and increase it for ISTA and FISTA. We recall the result that we established in Theorem 2.6.1 which provides the least possible value of the penalty parameter $\lambda_{\max} = \|\mathcal{M}^T\mathcal{V}\|_\infty$

for which the entries of the recovery are all zero. For different choices of penalty parameter, we choose $\lambda = C\lambda_{\max}$, where $C \in (0,1)$.

Algorithms	SNR = 30 dB			SNR = 20 dB		
	RE	SR	RT (sec.)	RE	SR	RT (sec.)
ADMM	**0.5559**	0.0491	584.92	**0.6144**	0.24211	531.14
ISTA	0.9999	0.0207	**341.11**	0.9999	0.0207	343.50
FISTA	1.0121	**0.0048**	346.48	1.0121	**0.0048**	**334.37**

Table 4.3: Recovery results for the problem (4.3) with different noise levels. We use $\lambda = 1$ for ADMM, and $\lambda = 0.1 \times \lambda_{\max}$ for ISTA and FISTA.

It is seen in Table 4.3 that the large value of λ used in ISTA and FISTA helped to recover sparse solutions. At the same time, the reconstruction error increases compared to the results shown in Table 4.2.

The results in Table 4.1 and Table 4.2 suggests that the ℓ_1-minimization problem (4.3) based on a fixed penalty parameter λ is not the best approach. Instead of solving (4.3) with a fixed parameter λ, we should solve corresponding weighted ℓ_1-minimization problem (4.13) that dynamically choose penalty parameter for better solution recovery.

In Section 2.6 of Chapter 2, we discussed the advantage of iterative reweighting techniques for solving the weighted problem (4.13) and different reweighting schemes in the literature. We first present how iterative reweighting techniques help improve recovery solutions in Figure 4.3. We use FISTA to solve the weighted ℓ_1-minimization problem (4.13) in all six reweighting schemes with four reweighting steps (or four iteration in Algorithm 4.3).

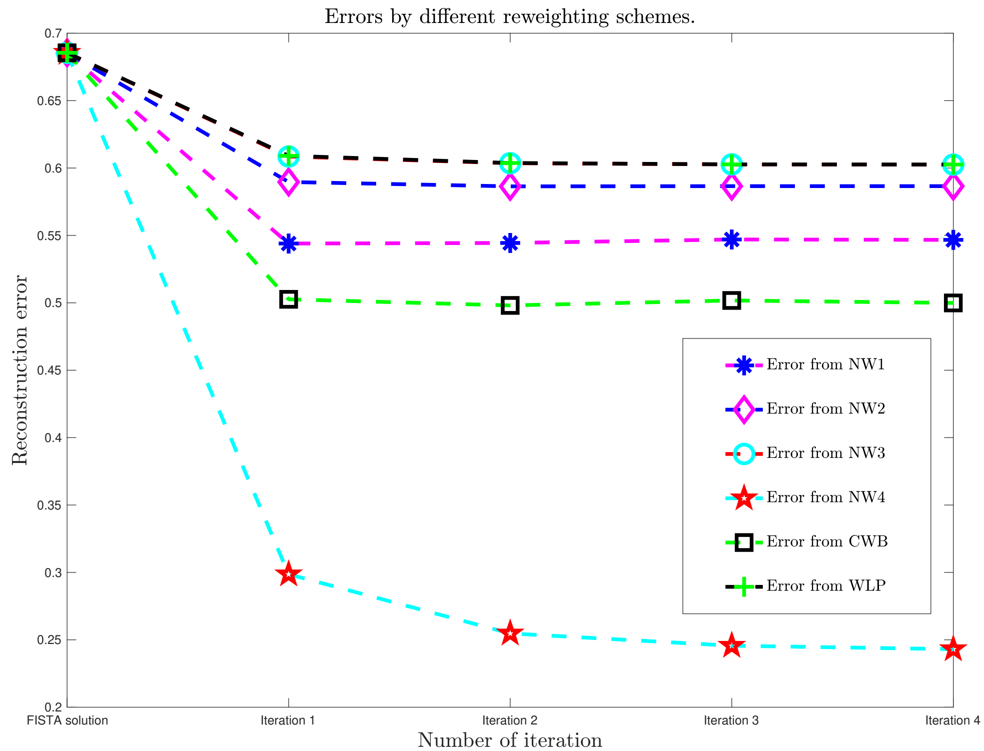


Figure 4.3: Reconstruction error by different weighting schemes used in Algorithm 4.3.

The effectiveness of the iterative reweighting techniques is seen in Figure 4.3. Without reweighting, FISTA recovers the solution with 68% reconstruction error. All of the reweighting schemes contribute to improve the solution in the first reweighting step. We do not see significant improvements from NW1, NW2, NW3, CWB, and WLP reweighting schemes after the first reweighting step. NW4 helps to improve the solution recovery in each reweighting step. Reducing RE from 68% to 30% in one step of reweighting by NW4 is a great achievement. In four reweighting steps, NW4 reduces RE form 68% to 24% which is a significant improvement in solution. The result in Figure 4.3 motivates us to see the impact of iterative reweighting in all three algorithms solving the weighted Sylvester type LASSO problem (4.13).

Weight Schemes	ISTA			FISTA			ADMM		
	RE	SR	RT	RE	SR	RT	RE	SR	RT
None ⋆	0.9792	0.9931	420.48	0.6855	0.8423	446.95	0.5572	0.0209	514.06
NW1	0.9784	0.4152	339.26	0.5450	0.0722	325.96	0.5358	0.0040	1028.84
	0.9782	0.3772	322.10	0.5422	0.0431	317.73	0.5419	0.0039	1029.58
NW2	0.9789	0.6565	334.29	0.5909	0.1554	318.80	0.5287	0.0046	1102.67
	0.9788	0.6442	339.64	0.5856	0.1237	317.56	0.5310	0.0042	1110.28
NW3	0.9790	0.7849	342.39	0.6096	0.2348	321.07	0.5281	0.0054	1101.29
	0.9790	0.7808	330.90	0.6033	0.2085	319.02	0.5254	0.0048	1106.40
NW4	0.9759	0.0675	330.29	**0.2986**	**0.0090**	**329.52**	0.5995	0.0033	1020.96
	0.9778	0.0407	331.51	**0.2547**	**0.0047**	**321.99**	0.6163	0.0031	1050.06
CWB	0.9782	0.3050	317.98	0.5035	0.0462	325.17	0.5462	0.0038	1101.39
	0.9779	0.2575	322.09	0.4958	0.0217	325.98	0.5513	0.0037	1109.83
WLP	0.9790	0.7880	316.80	0.6102	0.2380	333.78	0.5211	0.0054	1105.08
	0.9790	0.7838	318.14	0.6035	0.2107	331.76	0.5204	0.0048	1102.03

⋆ This is the result without reweighting.

Table 4.4: Results of iterative reweighting via six different weight schemes with two reweighting steps for each weight scheme.

The numerical results in Table 4.4 describe the performances of the algorithms for the weighted Sylvester LASSO problem (4.13) using the iterative reweighting technique as outlined in Algorithm 4.3 for all three methods ISTA, FISTA, and ADMM. We use the cold start initialization in all three algorithms and initialize the weight matrix having diagonal entries all one. The solutions of the weighted problem (4.13) by the methods without reweighting is listed on the row called None* in Table 4.4. Using these as initial solutions, we update weights according to the six different weight schemes respectively. We run all three methods twice with updated weights as outlined in Algorithm 4.3. The numerical results corresponding to six different weight schemes are presented in Table 4.4.

In terms of RT, both ISTA and FISTA are similar but ADMM takes more time for all reweighting schemes. ISTA has poor performance with all six different reweighting schemes in each of the three measures, and could not reduce RE to less than 90%. ADMM recovers SR to very close to the ground truth for all of the reweighting schemes. None of the weighting schemes could reduce RE by 52%. On the other hand, FISTA has the best performance in terms of both RE and SR in the first run without reweighting. All of the weighting schemes help to reduce the error from an initial 68% to below 60% in one step of reweighting. Among the six weight schemes, CWB and NW4 have the best performance in terms of reducing RE and maintaining the sparsity in solution. Among the six weight schemes, the NW4 weight scheme has superior performance. After 2 reweighting steps, it almost recovers the sparsity level similar to the ground truth. Also, only after two reweighting steps, RE is reduced from 68% to 25%. The best solution recovery from FISTA algorithms using NW4 is highlighted in Table 4.4.

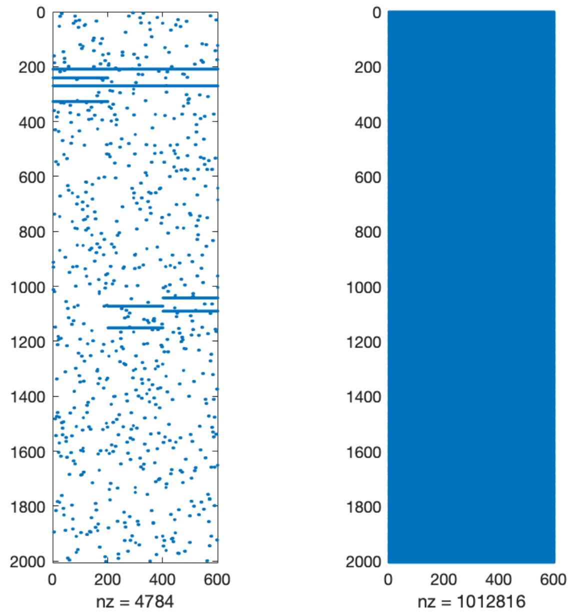
The numerical results in Table 4.4 suggest that one of the best approaches for solving the weighted Sylvester LASSO problem (4.13) is using the iterative reweighting technique with NW4. Algorithm 4.3 summarizes the procedure for solving the problem (4.13) with NW4.

SNR = ∞ dB (noiseless)			
Number of iteration	RE	SR	RT (sec.)
FISTA solution	**0.6855**	**0.8423**	**319.95**
Iteration 1	0.2986	0.0090	317.97
Iteration 2	0.2547	0.0047	317.69
Iteration 3	0.2456	0.0043	327.31
Iteration 4	**0.2431**	**0.0042**	**318.76**

Table 4.5: Recovery results by Algorithm 4.3 for noiseless data with initial weight matrix having all diagonal entries one.

In Table 4.5, we summarize the performance of Algorithm 4.3 for noiseless synthetic data. Measurements for the solution of the initial FISTA run are listed on the first row called FISTA initial. The proposed algorithm recovers solutions with smaller reconstruction error and better sparsity recovery.

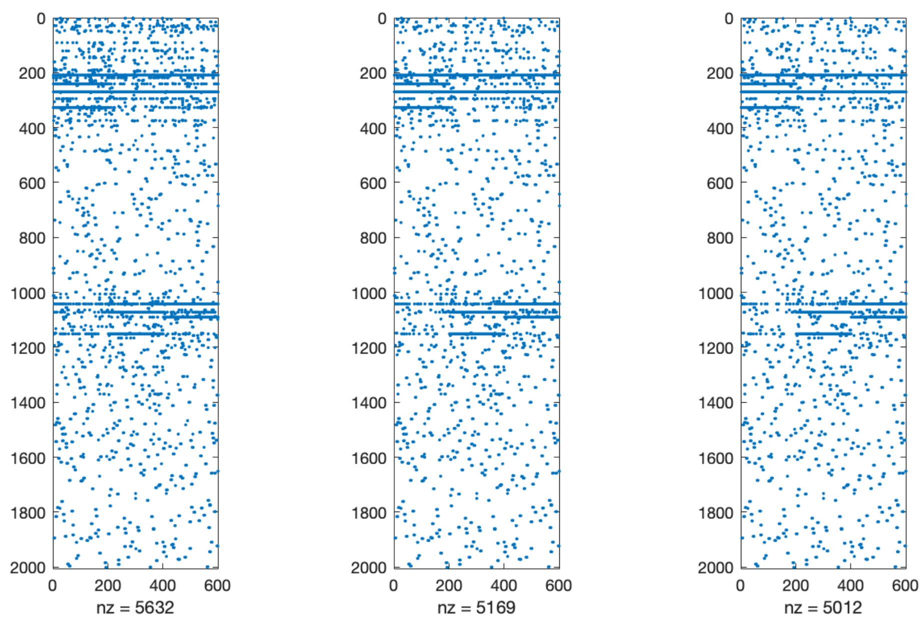
We present matrix sparsity plots (or spy plots) in Figure 4.4 using the **spy** function in MATLAB for the results of recovery in Table 4.5. The figurative representation of matrix recovery in Figure 4.4 shows the effectiveness of our proposed method for solving Sylvester LASSO problem (4.13). The first spy plot in Figure 4.4 is for the ground truth or exact solution, the second plot is the recovery result of the initial run of FISTA without reweighting, the third plot is the recovery from the first reweighting, and the last plot is the recovery from the fourth reweighting of Algorithm 4.3.



(a) Ground truth:
SR = 0.0040

(b) FISTA solution:
RE=0.685, SR=0.842

(c) First iteration:
RE=0.298, SR=0.009



(d) Second iteration:
RE=0.254, SR = 0.0046

(e) Third iteration:
RE = 0.245, SR = 0.0042

(f) Fourth iteration:
RE = 0.243, SR = 0.0041

Figure 4.4: Sparsity plot of the solutions by Algorithm 4.3 in 4 reweighting steps compared with the initial solution of FISTA and ground truth.

Our proposed method works well for noise-free data as we witnessed Figure 4.4 and Table 4.5. But in practice, we have to deal with noise. We consider the noise in synthetic data with three different noise levels determined by SNR being 30 dB, 20 dB, and 10 dB, respectively. The results of the recovery of the proposed method with noisy data are shown in Table 4.6.

Number of iteration	SNR = 30 dB			SNR = 20 dB			SNR = 10 dB		
	RE	SR	RT	RE	SR	RT	RE	SR	RT
FISTA solution	**0.6859**	**0.8661**	**330.24**	**0.6914**	**0.9312**	**319.97**	**0.7549**	**0.9896**	**335.59**
Iteration 1	0.2986	0.0090	340.76	0.3018	0.0093	321.48	0.3431	0.0121	336.32
Iteration 2	0.2535	0.0046	330.02	0.2585	0.0046	317.79	0.2900	0.0050	324.29
Iteration 3	0.2416	0.0043	317.74	0.2480	0.0043	318.20	0.2730	0.0044	328.84
Iteration 4	**0.2389**	**0.0041**	**311.48**	**0.2452**	**0.0041**	**315.90**	**0.2667**	**0.0042**	**327.89**

Table 4.6: Results of recovery by Algorithm 4.3 for the Sylvester type LASSO problem with synthetic data at three different noise levels.

We used the cold start initialization and the identity matrix as an initial weight to run Algorithm 4.3 for all three different noise levels. The results in Table 4.6 shows how the proposed model performs at different levels of noise. The first row in Table 4.6 called FISTA initial shows the solution results of the initial FISTA run. The results in Table 4.6 show that the proposed method is robust in solution reconstruction for smaller to larger noises. When the data has 30 dB noise, the initial FISTA run recovers the solution with RE 68% and SR 0.86. As we apply our proposed algorithm with four reweighting with NW4, RE drops to 23% and SR is almost the same as the ground truth. The larger noise on data affects recovery. Even with larger noise in data, the recovery results of the proposed model is not highly affected. We plot reconstruction error in three different noise levels in Figure 4.5 to see the impact of noise on our proposed model for recovering the solution.

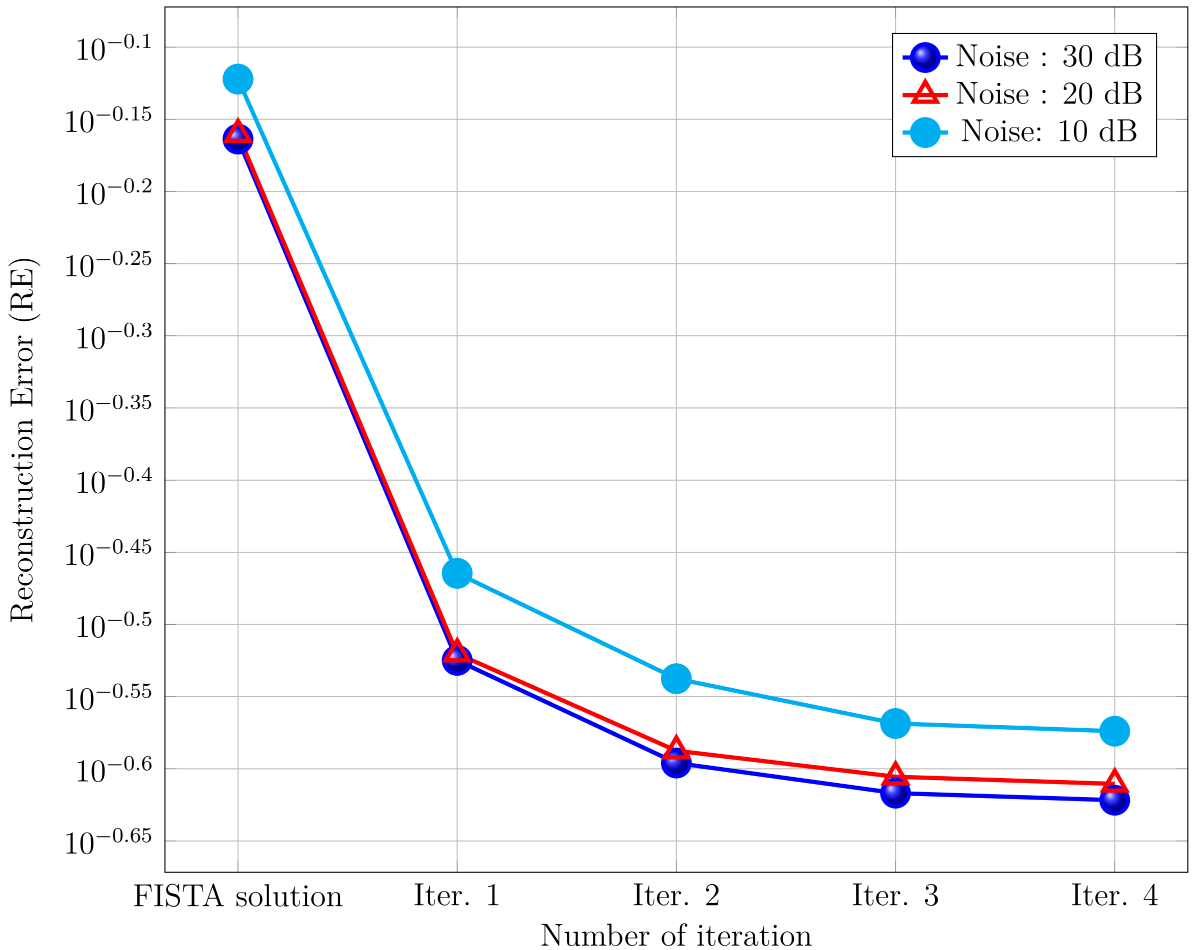


Figure 4.5: Reconstruction error by the proposed algorithm at different noise levels.

The error plots in Figure 4.5 show the stability of our proposed model for solution recovery in smaller to larger noise levels in data. The recovery of our proposed algorithm is robust even though data has large noises. Similarly, we plot SR by our proposed model at different noise levels in Figure 4.6.

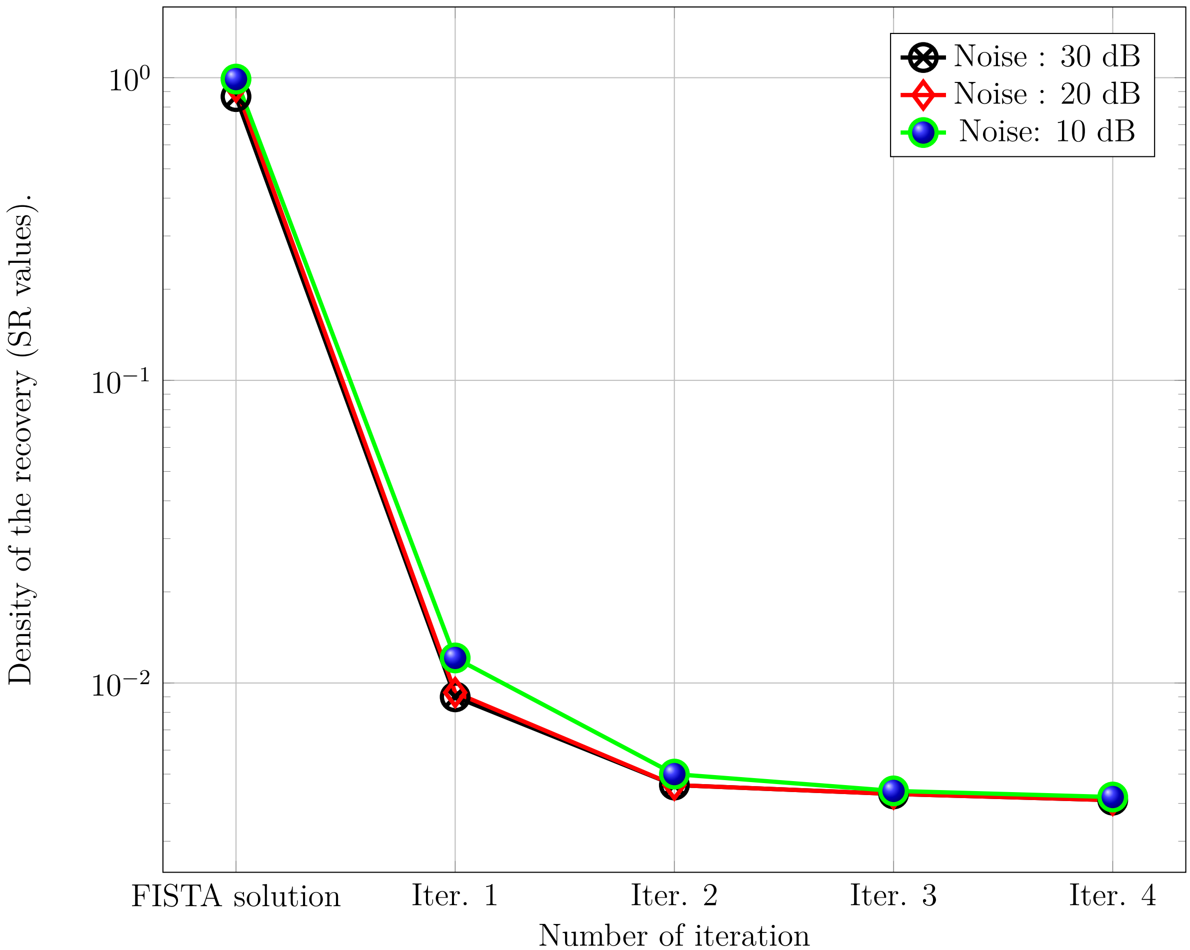


Figure 4.6: Sparsity recovery by the proposed algorithm at different noise levels.

The results in Figure 4.6 show the stability in the sparse reconstruction of our proposed algorithm for three different noise levels. In terms of both sparsity and reconstruction error, our proposed method works for both clean and noisy data.

In summary, our proposed algorithm to solve the Sylvester type LASSO problem has superior performance in solution reconstruction. It can solve the computationally expensive problem by extracting the features of the large data. Our comprehensive numerical simulation results show that the iterative reweighting techniques are effective.

CHAPTER 5

ESI MODEL CAPTURING THE SOURCE ACTIVATION PATTERN

5.1 Introduction

Over the past few decades, various techniques have been developed for non-invasive measurements of brain source activities. EEG is one among those due to its portability, low cost, and effectiveness for brain source localizations. Given recorded EEG signals, reconstructing the source activities inside the brain is referred to as the EEG source imaging (ESI). As we discussed in Chapter 3, ESI is a highly "ill-posed" inverse problem and requires prior assumptions or regularizations to estimate a solution. We also discussed some of the popular neurophysiological assumptions (regularizations) that researchers considered and current state-of-the-art methods for solving the ESI problem. The presences of noises in both EEG signals and brain sources make the ESI problem challenging. Very few studies have considered noises in both channel and source spaces in the ESI problem. Recognizing certain pattern or structure in measured EEG signals is essential to denoise properly. To effectively deal with noises in both sources and channels, Wang, Liu, Lou, Li, and Purdon [70] proposed a probabilistic model with specially designed hierarchical prior that models both the micro-states and manifold structure in ESI. Motivated by [70], we propose a new ESI model that gathers EEG signals with a similar activation pattern and use a mixed norm penalty to enhance group sparsity. Before moving to the model formulation, we provide a brief description of the principle of maximum entropy in the next section which will be used in our ESI model.

5.2 The Principle of Maximum Entropy

The entropy measures the amount of uncertainty contained in a probability distribution. It is applied to measure the disorder of a set and widely used in machine learning [78], statistics [3], and information theory [79].

Definition 5.2.1 (Entropy [3, 78, 79])**.** Let S be a partition of the instances into n target attributes $\{s_1, s_2, \cdots, s_n\}$. The *entropy* of S relative to n targets is defined as

$$\text{Entropy}(S) = \sum_{i=1}^{n} p_i \log\left(\frac{1}{p_i}\right),$$

where p_i is the probability of an instance in target attribute s_i and

$$\sum_{i=1}^{n} p_i = 1, \quad p_i \geq 0 \quad \text{for } i = 1, \cdots, n. \tag{5.1}$$

The lowest value of the entropy is zero, which represents no disorder in a set, i.e., all instances in a set have a single target attribute. The entropy is maximum when a set has instances with equally mixed target attributes or the probability distribution (5.1) is uniform. In general, the large value of the entropy represents the higher disorder in a set.

As an example, consider a set having 10 instances of two classes, + and -. The probabilities corresponding to positive and negative classes are denoted by $p(+)$ and $p(-)$ such that $p(+) = 1 - p(-)$. In Figure 5.1, the set has all negative instances (or $p(+) = 0$) at the lower left such that it has a minimal disorder and the entropy zero. When we start to switch the class labels of elements of the set from - to +, the entropy increases. Entropy is maximum when a set has instances mixed with equal number of positive and negative classes. As positive classes are increased, the entropy lowers again and hits zero when the set contains only the positive classes.

We observed that a high entropy translates to high unpredictability of the probability distribution. Maximizing the entropy is consistent with maximizing the

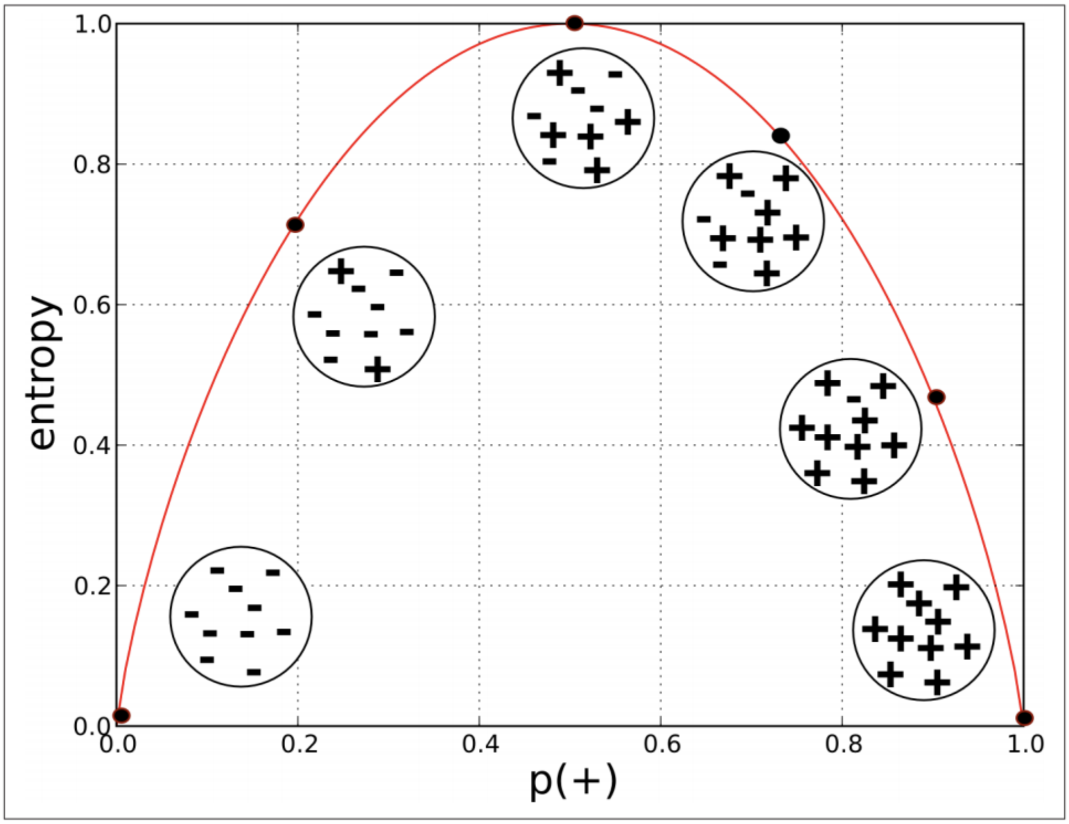


Figure 5.1: Entropy of two-class set as a function of $p(+)$. Figure and example are taken from [3].

unpredictability, given little information we may know about a distribution. If nothing is known about the distribution except that it belongs to certain class, we choose the distribution that maximizes the entropy. In another word, maximizing the entropy minimizes the amount of prior information built into the distribution. For example, the most informative distribution we can imagine is where we know that an event will occur 100% of the time, giving an entropy zero. The least informative distribution we can imagine is a uniform distribution, where each event in the sample space has an equal chance of occurring, and has the maximum entropy. The principle of maximum entropy is based on the premise that when estimating a probability distribution, one should select the distribution which leaves the largest remaining uncertainty, i.e., the maximum entropy, consistent with the constraints of the probability distribution [80]. The general mathematical formulation of the entropy maximization can be expressed as

$$\begin{aligned} \text{maximize} \quad & H(p) = \sum_{i=1}^{n} p_i \log\left(\frac{1}{p_i}\right) \\ \text{subject to} \quad & p_i \geq 0 \quad \text{for } i = 1, \cdots, n, \\ & \sum_{i=1}^{n} p_i = 1, \\ & \sum_{i=1}^{n} p_i r_{ij} = \beta_j \quad \text{for } 1 \leq j \leq m. \end{aligned} \tag{5.2}$$

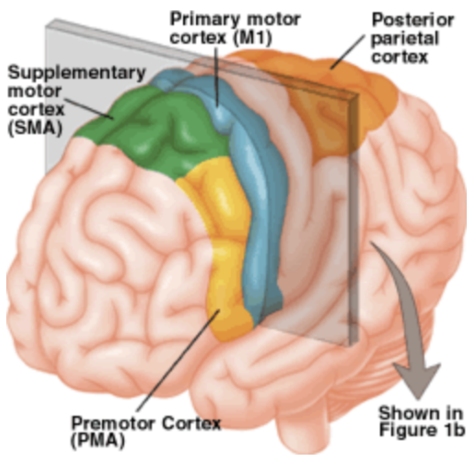
The optimal solution of the problem (5.2) is obtained by solving the Lagrangian of (5.2) for p_i [80, 81]. For detailed mathematical description of maximum entropy distribution, the reader is referred to [82].

5.3 Model Formulation

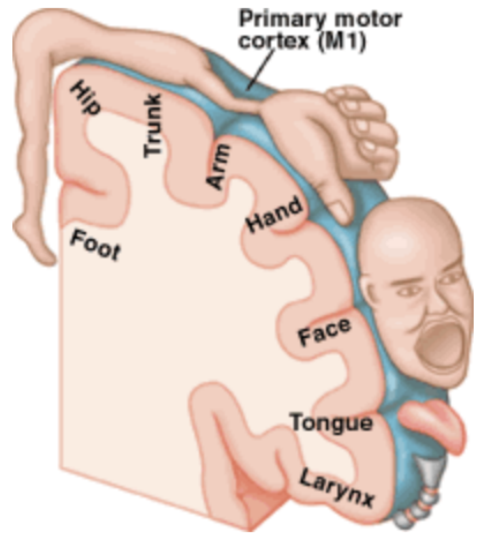
Let $X \in \mathbb{R}^{N_c \times N_t}$ be a matrix of EEG signals from N_c sensors (or channels) in N_t time points. The mapping from N_s brain sources to N_c channels is through the lead field matrix L obtained by descretizing the Maxwell's equation [71, 72]. Given X and L, the objective of ESI is to recover the source activations (or source signals), denoted by $S = [s_1, s_2, \cdots, s_{N_t}] \in \mathbb{R}^{N_s \times N_t}$, where each column s_i of S represents the electric potentials in N_s source locations for one of the N_t time points. The ESI model with sparsity prior as described in Chapter 3 is expressed as

$$\underset{S \in \mathbb{R}^{N_s \times N_t}}{\arg\min} \ \|X - LS\|_F^2 + \gamma_3 \|S\|_{1,1}, \tag{5.3}$$

where $\gamma_3 > 0$ is a penalty parameter for controlling the sparsity to the solution estimate.



(a) Principal cortical domains of the motor system. The primary motor cortex (M1) generates the signals that control the execution of movement.



(b) A figurative representation of the body map encoded in primary motor cortex.

Figure 5.2: The relation of movements of certain body parts and corresponding activation regions in the brain cortex. Both figures and their descriptions are taken from Brain Connection blog available at `https://brainconnection.brainhq.com/2013/03/05/the-anatomy-of-movement/`.

The human brain is divided into many cortical regions or brain sources. Each region is activated with certain human action [83]. For example, voluntary movements require activation of the motor and cerebral cortex. The cerebral cortex is also activated by coordinated sequences of movements, decision making about appropriate behavioral startegies and choices. These cortical regions generate signals to execute the desired actions [84]. Figure 5.2 explains how certain physical movements activate the motor cortex in the brain. In our proposed model, we would like to investigate the intrinsic features of region-wise source activations determined by certain human actions.

In the proposed model, we keep the classical sparsity prior and region-wise source activations determined by different human actions. The region-wise source activation is referred to as a block pattern of the source activation and would like to exploit these block structures in source localizations. We form the clusters of similar source signals to study block structure by introducing the latent variables in $C = [c_1, \cdots, c_K] \in \mathbb{R}^{N_s \times K}$, which we call the landmarks. We apply the clustering technique to N_t time point signals of N_s brain sources and form K clusters or landmarks out of them. Since the landmarks are the mean of the denoised source signals in S, it is reasonable for landmarks c_i to inherit the same sparsity property of S [70]. The objective function (5.3) with landmarks c_i is expressed as

$$\underset{S,C}{\arg\min}\ \|X - LS\|_F^2 + \gamma_2 \sum_{k=1}^{K} \|c_k\|_1 + \gamma_3 \sum_{i=1}^{N_t} \|s_i\|_1, \tag{5.4}$$

where $\gamma_2 > 0$ is a sparsity penalty for landmarks.

The relation between landmarks and sources are described by introducing a probability matrix $R \in \mathbb{R}^{N_t \times K}$, whose entries $r_{i,k}$ represent the probability of associating s_i with landmark c_k. Since we do not have much prior information about the probability distribution of each row of the assignment matrix R, we apply the principle of maximum entropy for the distribution in order to maximize the entropy. To that end, for fixed i, we minimize the negative of the entropy. In other words, we solve the following problem for estimating the probability distribution, the i-th row of R

$$\begin{aligned} \min H(r_{i,k}) &= -\sum_{k=1}^{K} r_{i,k} \log\left(\frac{1}{r_{i,k}}\right) \\ \text{subject to} \quad r_{i,k} &\geq 0 \quad \text{for } k = 1, \cdots, K, \\ \sum_{k=1}^{K} r_{i,k} &= 1. \end{aligned} \tag{5.5}$$

In addition to estimate the probability distribution for the rows of assignment matrix R, we would like to minimize the square distance between the electric potentials s_i in different time points associated with the same landmark. If s_i is associated with c_k, we minimize the square distance between them. If not, we do not want to consider the distance between them. To effectively deal with these situations, we scale the square distance between s_i and c_k by the probability of their assignment $r_{i,k}$. Thus, for each s_i, we would like to minimize the following sum of scaled squared distances

$$\sum_{k=1}^{K} r_{i,k}\|s_i - c_k\|^2.$$

Therefore, the scaled squared distances between all s_i and the landmarks c_k are obtained by solving the following problem

$$\min \sum_{i=1}^{N_t} \sum_{k=1}^{K} r_{i,k}\|s_i - c_k\|^2. \tag{5.6}$$

The combination of (5.5) and (5.6) are explained in [70] to estimate the true probability distribution of the sources using the kernel density estimation on C with the help of K landmarks. The reader is referred to [70] for a detailed probabilistic approach to explain these two constraints. We incorporate the estimate of the joint probability distribution of each row of assignment matrix R given by (5.5) along with the interaction of all s_i to the landmarks c_k given by (5.6) into the objective function (5.4) to give

$$\begin{aligned} \underset{S,C,R}{\arg\min}\ \|X - LS\|_F^2 + \lambda \sum_{i=1}^{N_t} \sum_{k=1}^{K} [r_{i,k}\|s_i - c_k\|^2 + \alpha r_{i,k} \log\, r_{i,k}] + \gamma_2 \sum_{k=1}^{K} \|c_k\|_1 \\ + \gamma_3 \sum_{i=1}^{N_t} \|s_i\|_1, \end{aligned} \tag{5.7}$$

where $\lambda > 0$ and $\alpha > 0$ are the regularization and smoothening parameters, respectively.

In order to capture the structure of the region-wise source activations we bring all the electric potentials s_i associated with the same landmark in one place. We form a matrix ($S\mathrm{diag}(r_k)$), r_k being the k-th column of R, which gathers all s_i associated with the landmark c_k, where

$$\mathrm{diag}(r_k) = \begin{pmatrix} r_{1,k} & 0 & \cdots & 0 \\ 0 & r_{2,k} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_{N_t,k} \end{pmatrix} \in \mathbb{R}^{N_t \times N_t}.$$

Let us consider a simple example where electric potentials for the first three time points s_1, s_2, and s_3 are associated to the landmark c_1. In this case, the first three probabilities $r_{1,1}, r_{2,1}$, and $r_{3,1}$ are close to one. The rest of the probabilities $r_{4,1}, \cdots, r_{N_t,1}$ are close to zero such that the entries in **violet** color in $S\mathrm{diag}(r_1)$ in (5.8) are close to zero, where

$$\begin{aligned} & S\mathrm{diag}(r_1) \\ &= \begin{pmatrix} s_{1,1} & s_{1,2} & s_{1,3} & s_{1,4} & \cdots & s_{1,N_t} \\ s_{2,1} & s_{2,2} & s_{2,3} & s_{2,4} & \cdots & s_{2,N_t} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{N_s,1} & s_{N_s,2} & s_{N_s,3} & s_{N_s,4} & \cdots & s_{N_s,N_t} \end{pmatrix} \begin{pmatrix} r_{1,1} & 0 & 0 & 0 & \cdots & 0 \\ 0 & r_{2,1} & 0 & 0 & \cdots & 0 \\ 0 & 0 & r_{3,1} & 0 & \cdots & 0 \\ 0 & 0 & 0 & r_{4,1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & r_{N_t,1} \end{pmatrix} \\ &= \begin{pmatrix} r_{1,1}s_{1,1} & r_{2,1}s_{1,2} & r_{3,1}s_{1,3} & r_{4,1}s_{1,4} & \cdots & r_{N_t,1}s_{1,N_t} \\ r_{1,1}s_{2,1} & r_{2,1}s_{2,2} & r_{3,1}s_{2,3} & r_{4,1}s_{2,4} & \cdots & r_{N_t,1}s_{2,N_t} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ r_{1,1}s_{N_s,1} & r_{2,1}s_{N_s,2} & r_{3,1}s_{N_s,3} & r_{4,1}s_{N_s,4} & \cdots & r_{N_t,1}s_{N_s,N_t} \end{pmatrix}. \end{aligned}$$

Thus, all activated sources (rows in $S\mathrm{diag}(r_1)$) in the first three time points (the first three columns in $S\mathrm{diag}(r_1)$) having a similar source activation pattern determined by the landmark c_1 are non-zero. In other words, the rows corresponding to the activated sources in the first three-time points of $S\text{diag}(r_1)$ have a non-zero row structure. These non-zero row structures represent the block pattern on source activation caused by certain human actions.

Our goal is to capture the non-zero row structures of $S\text{diag}(r_k)$ for each k. The mixed-norm $\ell_{2,1}$ penalty is suggested [4] to capture such structures. Figure 5.3 shows the source estimations in terms of sparsity recovery for problem (5.3) under three different norm penalties.

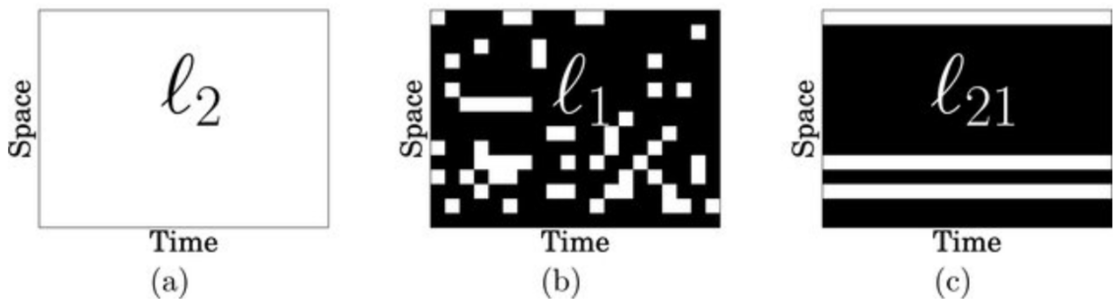


Figure 5.3: (a), (b), and (c) show the estimations of the source amplitudes of EEG inverse problem (5.3) by ℓ_2-, ℓ_1-, and $\ell_{2,1}-$norm penalty, respectively. The non-zero coefficients are shown in white. While ℓ_2-norm penalty yields only non-zero coefficients, $\ell_{2,1}$-norm penalty promotes non-zero coefficients with a row structure (only a few sources have non-zero amplitude over the entire time interval of interest). This illustration is inspired by Figure 1 of [4].

Motivated by Figure 5.3, we incorporate the mixed-norm penalty to capture the non-zero row structure of $S\text{diag}(r_k)$ into the objective function (5.7) as

$$\begin{aligned}\arg\min_{S,C,R} h(S,C,R) = \|X - LS\|_F^2 + \lambda \sum_{i=1}^{N_t}\sum_{k=1}^{K}[r_{i,k}\|s_i - c_k\|^2 + \alpha r_{i,k}\log r_{i,k}] \\ + \gamma_1 \sum_{k=1}^{K} \|S\ \text{diag}(r_k)\|_{2,1} + \gamma_2 \sum_{k=1}^{K}\|c_k\|_1 + \gamma_3 \sum_{i=1}^{N_t}\|s_i\|_1, \quad (5.8)\end{aligned}$$

where $\gamma_1 > 0$ is a mixed-norm penalty parameter promoting the row structured sparsity and

$$\|A\|_{2,1} = \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} a_{ij}^2}$$

is a $L_{p,q}$ mixed-norm with $p = 2$, $q = 1$ of $A = (a_{ij}) \in \mathbb{R}^{m\times n}$.

5.4 Numerical Algorithm

In this section, we develop a numerical optimization method to solve the proposed ESI model (5.8). Optimizing the model (5.8) over a mixed-norm directly brings difficulty in computing the gradients. We change the formulation of the proposed ESI model (5.8) to the following constrained optimization problem to simplify the mathematical derivation

$$\begin{aligned} &\arg\min_{S,C,R,M} \tilde{h}(S,C,R,M) \\ &\text{subject to} \quad S \text{ diag}(r_k) = M_k \quad \text{for } k = 1, \cdots, K, \end{aligned} \tag{5.9}$$

where

$$\begin{aligned} \tilde{h}(S,C,R,M) = \|X - LS\|_F^2 + \lambda \sum_{i=1}^{N_t}\sum_{k=1}^{K} [r_{i,k}\|s_i - c_k\|^2 + \alpha r_{i,k} \log r_{i,k}] \\ + \gamma_1 \sum_{k=1}^{K} \|M_k\|_{2,1} + \gamma_2 \sum_{k=1}^{K} \|c_k\|_1 + \gamma_3 \sum_{i=1}^{N_t} \|s_i\|_1, \end{aligned}$$

$M = \{M_k\}_{k=1}^K$ with $M_k \in \mathbb{R}^{N_s\times N_t}$ for each k.

We apply the ADMM framework to solve K constraints optimization problem (5.9). We minimize the following augmented Lagrangian function of (5.9)

$$\begin{aligned} \arg\min_{S,C,R,M,P} g(S,C,R,M,P) = \|X - LS\|_F^2 + \lambda \sum_{i=1}^{N_t}\sum_{k=1}^{K} [r_{i,k}\|s_i - c_k\|^2 + \alpha r_{i,k} \log r_{i,k}] \\ + \gamma_1 \sum_{k=1}^{K} \|M_k\|_{2,1} + \sum_{k=1}^{K} \langle P_k, S \text{ diag}(r_k) - M_k\rangle \end{aligned}$$

$$+\frac{\sigma}{2}\sum_{k=1}^{K}\|S\ \mathrm{diag}(r_k)-M_k\|_F^2$$
$$+\gamma_2\sum_{k=1}^{K}\|c_k\|_1+\gamma_3\sum_{i=1}^{N_t}\|s_i\|_1, \tag{5.10}$$

where $\sigma > 0$ is an augmented Lagrangian parameter, and $P_k \in \mathbb{R}^{N_s\times N_t}$ is dual variable for each k. We now find the minimizer of (5.10) with respect to the variables S, C, R, M, and P, alternatingly.

We minimize (5.10) over the variable S while C, R, M, P are being fixed. We denote the i-th column of the k-th dual variable P_k by $(p_k)_i$. Also, the i-th column of the k-th matrix M_k is denoted by $(m_k)_i$. The S-subproblem takes the following form

$$\begin{aligned}
& \arg\min_{S}\ g(S,C,R,M,P)\\
= \ & \arg\min_{S}\ \|X-LS\|_F^2+\lambda\sum_{i=1}^{N_t}\sum_{k=1}^{K}r_{i,k}\|s_i-c_k\|^2+\sum_{k=1}^{K}\langle P_k, S\ \mathrm{diag}(r_k)-M_k\rangle\\
& +\gamma_3\sum_{i=1}^{N_t}\|s_i\|_1+\frac{\sigma}{2}\sum_{k=1}^{K}\|S\ \mathrm{diag}(r_k)-M_k\|_F^2\\
= \ & \arg\min_{s_i}\ \sum_{i=1}^{N_t}\Bigg(\|x_i-Ls_i\|_2^2+\gamma_3\|s_i\|_1+\lambda\sum_{k=1}^{K}r_{i,k}\|s_i-c_k\|^2\\
& +\frac{\sigma}{2}\sum_{k=1}^{K}\|r_{i,k}s_i-(m_k)_i\|^2\Bigg)+\sum_{k=1}^{K}\mathrm{trace}(P_k^T S\ \mathrm{diag}(r_k))\\
= \ & \arg\min_{s_i}\ \sum_{i=1}^{N_t}\Bigg(\|x_i-Ls_i\|_2^2+\gamma_3\|s_i\|_1+\lambda\sum_{k=1}^{K}r_{i,k}\|s_i-c_k\|^2+\sum_{k=1}^{K}r_{i,k}(p_k)_i^T s_i\\
& +\frac{\sigma}{2}\sum_{k=1}^{K}\|r_{i,k}s_i-(m_k)_i\|^2\Bigg)\\
= \ & \arg\min_{s_i}\ \sum_{i=1}^{N_t}\Bigg(s_i^T L^T L s_i-2x_i^T L s_i+x_i^T x_i+\gamma_3\|s_i\|_1+\lambda s_i^T s_i\sum_{k=1}^{K}r_{i,k}\\
& -2\lambda\sum_{k=1}^{K}r_{i,k}c_k^T s_i+\lambda\sum_{k=1}^{K}r_{i,k}c_k^T c_k+\sum_{k=1}^{K}r_{i,k}(p_k)_i^T s_i+\frac{\sigma}{2}\sum_{k=1}^{K}r_{i,k}^2 s_i^T s_i
\end{aligned}$$

$$
\begin{aligned}
& -\sigma \sum_{k=1}^{K} r_{i,k} (m_k)_i^T s_i + \frac{\sigma}{2} \sum_{k=1}^{K} (m_k)_i^T (m_k)_i^T \Bigg) \\
= \;& \arg\min_{s_i} \; \sum_{i=1}^{N_t} \Bigg(s_i^T L^T L s_i + \lambda s_i^T s_i + \frac{\sigma}{2} \sum_{k=1}^{K} r_{i,k}^2 s_i^T s_i + \gamma_3 \|s_i\|_1 \\
& -2 \left\{ x_i^T L + \sum_{k=1}^{K} \left[\lambda r_{i,k} c_k^T - r_{i,k} (p_k)_i^T + \frac{\sigma}{2} r_{i,k} (m_k)_i^T \right] \right\} s_i \Bigg) \\
= \;& \arg\min_{s_i} \sum_{i=1}^{N_t} \left(s_i^T \left\{ L^T L + \left(\lambda + \frac{\sigma}{2} \sum_{k=1}^{K} r_{i,k}^2 \right) I \right\} s_i - 2 b_i^T s_i + \gamma_3 \|s_i\|_1 \right) \qquad (5.11) \\
= \;& \arg\min_{s_i} \sum_{i=1}^{N_t} \left(s_i^T U^T U s_i - 2 b_i^T s_i + \gamma_3 \|s_i\|_1 \right) \\
= \;& \arg\min_{s_i} \sum_{i=1}^{N_t} (\|U s_i - U^{-T} b_i\|_2^2 + \gamma_3 \|s_i\|_1), \qquad (5.12)
\end{aligned}
$$

where U is the Cholesky factor of $L^T L + \left(\lambda + \frac{\sigma}{2} \sum_{k=1}^{K} r_{i,k}^2 \right) I = U^T U$ and $b_i = \left(x_i^T L + \sum_{k=1}^{K} \left[\lambda r_{i,k} c_k^T - r_{i,k} (p_k)_i^T + \frac{\sigma}{2} r_{i,k} (m_k)_i^T \right] \right)^T$ in (5.11). Therefore, the S-subproblem (5.12) is equivalent to solving N_t independent strictly convex subproblem:

$$
s_t := \arg\min_{s_t} \|U s_t - U^{-T} b_t\|_2^2 + \gamma_3 \|s_t\|_1. \qquad (5.13)
$$

The objective function of R-subproblem is expressed as

$$
\begin{aligned}
& \arg\min_{R} \; g(S, C, R, M, P) \\
= \;& \arg\min_{R} \; \|X - LS\|_F^2 + \lambda \sum_{i=1}^{N_t} \sum_{k=1}^{K} [r_{i,k} \|s_i - c_k\|^2 + \alpha r_{i,k} \log\, r_{i,k}] \\
& + \gamma_1 \sum_{k=1}^{K} \|M_k\|_{2,1} + \frac{\sigma}{2} \sum_{k=1}^{K} \|S \;\mathrm{diag}(r_k) - M_k\|_F^2 + \sum_{k=1}^{K} \langle P_k, S \;\mathrm{diag}(r_k) - M_k \rangle \\
& + \gamma_2 \sum_{k=1}^{K} \|c_k\|_1 + \gamma_3 \sum_{i=1}^{N_t} \|s_i\|_1 \\
= \;& \arg\min_{R} \; \lambda \sum_{i=1}^{N_t} \sum_{k=1}^{K} \left[r_{i,k} \|s_i - c_k\|^2 + \alpha r_{i,k} \log\, r_{i,k} \right] + \frac{\sigma}{2} \sum_{k=1}^{K} \|S \;\mathrm{diag}(r_k) - M_k\|_F^2 \\
& + \sum_{k=1}^{K} \mathrm{trace} \left(P_k^T S \;\mathrm{diag}(r_k) \right)
\end{aligned}
$$

$$
\begin{aligned}
&= \arg\min_R \lambda \sum_{i=1}^{N_t}\sum_{k=1}^{K}\left[r_{i,k}\|s_i - c_k\|^2 + \alpha r_{i,k}\log\, r_{i,k}\right] + \frac{\sigma}{2}\sum_{i=1}^{Nt}\sum_{k=1}^{K}\|r_{i,k}s_i - (m_k)_i\|_2^2 \\
&\quad + \sum_{i=1}^{Nt}\sum_{k=1}^{K} r_{i,k}(p_k)_i^T s_i \\
&= \arg\min_R \sum_{i=1}^{N_t}\sum_{k=1}^{K}\Big[\lambda r_{i,k}\|s_i - c_k\|^2 + \lambda\alpha r_{i,k}\log\, r_{i,k} + \frac{\sigma}{2} r_{i,k}^2 s_i^T s_i - \sigma r_{i,k} s_i^T (m_k)_i \\
&\quad + \frac{\sigma}{2}(m_k)_i^T (m_k)_i + r_{i,k}(p_k)_i^T s_i\Big] \\
&= \arg\min_R \sum_{i=1}^{N_t}\sum_{k=1}^{K}\Big[\lambda\alpha r_{i,k}\log\, r_{i,k} + \frac{\sigma}{2} r_{i,k}^2 s_i^T s_i + r_{i,k}\big(\lambda\|s_i - c_k\|^2 + (p_k)_i^T s_i \\
&\qquad\qquad - \sigma s_i^T (m_k)_i\big)\Big].
\end{aligned}
$$

Each row of R represents the probabilities associated the i-th electric potential s_i with source activation to each landmark c_k and thus the i-th row of R lies in $\mathcal{R}_i = \{r_{i,k} | r_{i,k} \geq 0, \sum_{k=1}^{K} r_{i,k} = 1\}$. To recover $R(i,:)$, the i-th row of R, we use the projected gradient descent (PGD) [85] method which solves the following N_t subproblems:

$$
\arg\min_{r_{i,k}\in\mathcal{R}_i} \sum_{k=1}^{K}\Big[\lambda\alpha r_{i,k}\log\, r_{i,k} + \frac{\sigma}{2} r_{i,k}^2 s_i^T s_i + r_{i,k}\big(\lambda\|s_i - c_k\|^2 + (p_k)_i^T s_i - \sigma s_i^T (m_k)_i\big)\Big]. \tag{5.14}
$$

Also, the elementwise gradient of the objective function of (5.14) can be expressed as

$$
\lambda\alpha(1 + \log\, r_{i,k}) + \sigma r_{i,k} s_i^T s_i + \lambda\|s_i - c_k\|^2 + (p_k)_i^T s_i - \sigma s_i^T (m_k)_i. \tag{5.15}
$$

We transform the objective of (5.14) and elementwise gradient (5.15) in vector form which helps formulating the PGD to solve (5.14). We define the following

$$
\begin{aligned}
&t_{i,k} = \lambda\|s_i - c_k\|^2 + (p_k)_i^T s_i - \sigma s_i^T (m_k)_i, \\
&v = (t_{i,1}, \cdots, t_{i,K}) \in \mathbb{R}^{1\times K}, \\
&R(i,:) = (r_{i,1}, \cdots, r_{i,K}) \in \mathbb{R}^{1\times K}, \text{ and}
\end{aligned}
$$

$$d = (\log r_{i,1}, \cdots, \log r_{i,K}) \in \mathbb{R}^{1\times K}.$$

The optimization problem (5.14) can be reformulated in the vector form as

$$\underset{R(i,:)\in\mathcal{R}_i}{\arg\min}\ f(R(i,:)), \tag{5.16}$$

where $f(R(i,:)) = \lambda\alpha R(i,:)d^T + \frac{\sigma}{2}s_i^T s_i R(i,:)^T R(i,:) + R(i,:)v^T$. Conventionally, the gradient of the objective function that can be decomposed into the columns is defined as a column vector. In our case, we decompose the objective function of (5.14) into the rows and define the gradient as a row vector which is expressed as

$$\nabla f(R(i,:)) = \lambda\alpha(\mathbf{1}^T + d) + \sigma s_i^T s_i R(i,:) + v, \tag{5.17}$$

where $\mathbf{1} \in \mathbb{R}^K$ is a vector of all ones.

We define the projection of a vector to a set in Definition 5.4.1 before describing the PGD to solve (5.14).

Definition 5.4.1. The *projection* [34] of a vector y, onto a nonempty convex set X is defined as

$$\Pi_X(y) = \underset{x\in X}{\arg\min}\ \|x-y\|_2^2.$$

For R-subproblem (5.14), projection set is $\mathcal{R}_i$ in Algorithm 5.1,

$$\Pi_{\mathcal{R}_i}(y) = \underset{x\in\mathcal{R}_i}{\arg\min}\ \|x-y\|_2^2. \tag{5.18}$$

There are several well-studied optimization methods in the literature [86–88] to solve (5.18). We use the method proposed by Duchi, Shalev-Shwartz, Singer, and Chandra [88] and their MATLAB package[1] for the projection step. We use the formula suggested in [70] to initialize R as

$$r_{i,k} = \frac{\exp\left(-\frac{\|s_i - c_k\|^2}{\alpha}\right)}{\sum_{k=1}^{K}\exp\left(-\frac{\|s_i-c_k\|^2}{\alpha}\right)}, \forall\ i,k. \tag{5.19}$$

[1] Code available online: `https://web.stanford.edu/~jduchi/projects/DuchiShSiCh08/ProjectOntoSimplex.m`

Algorithm 5.1 Projected Gradient Descent for R-subproblem.

Input: small fixed step-size $\gamma > 0$, maximum iteration $N_{\max}$;

Output: a solution of R-subproblem (5.14).

Initialize: $R^{(0)} = R$ by (5.19);
for $i = 1, \cdots, N_t$ **do**
  set $t = 1$, $R^{(1)}(i,:) = R^{(0)}(i,:)$;
  while $t \leq N_{\max}$ **do**
    $R^{(t+1)}(i,:) = \Pi_{\mathcal{R}_i}\left(R^{(t)}(i,:) - \gamma \nabla f(R^{(t)}(i,:))\right)$;
    $t = t + 1$;
  end while
  end for
return last $R^{(t)}$.

We summarize the PGD algorithm for solving the N_t independent problems in (5.14) for the R-subproblem of the proposed algorithm in Algorithm 5.1.

The M-subproblem of the optimization problem (5.10) is expressed as

$$\begin{aligned}\arg\min_{M} g(S,C,R,M,P) = \arg\min_{M_k} \gamma_1 \sum_{k=1}^{K} \|M_k\|_{2,1} + \frac{\sigma}{2}\sum_{k=1}^{K} \|S \text{ diag}(r_k) - M_k\|_F^2 \\ + \sum_{k=1}^{K} \langle P_k, S \text{ diag}(r_k) - M_k \rangle. \end{aligned} \tag{5.20}$$

The optimization problem (5.20) for a fixed k is expressed as

$$\begin{aligned} & \arg\min_{M_k} \gamma_1 \|M_k\|_{2,1} + \frac{\sigma}{2}\|S \text{ diag}(r_k) - M_k\|_F^2 + \langle P_k, S \text{ diag}(r_k) - M_k \rangle \\ = \; & \arg\min_{M_k} \gamma_1 \|M_k\|_{2,1} + \frac{\sigma}{2}\|S \text{ diag}(r_k) - M_k\|_F^2 - \text{ trace}(P_k^T M_k) \\ = \; & \arg\min_{M_k} \gamma_1 \|M_k\|_{2,1} + \frac{\sigma}{2}\|S \text{ diag}(r_k) + U_k - M_k\|_F^2 \\ = \; & \arg\min_{M_k^{(1)}, M_k^{(2)}, \cdots, M_k^{(N_s)}} \sum_{i=1}^{N_s} \Big\{ \gamma_1 \|M_k^{(i)}\|_2 \end{aligned}$$

$$+\frac{\sigma}{2}\|(S\ \mathrm{diag}(r_k)+U_k)^{(i)}-M_k^{(i)}\|_2^2\Big\}, \tag{5.21}$$

where $U_k = \frac{1}{\sigma}P_k$, and $M_k^{(i)}$ is the i-th row of M_k. For each i, (5.21) can be decomposed into the following N_s independent subproblems,

$$\arg\min_{M_k^{(i)}}\ \left\{\gamma_1\|M_k^{(i)}\|_2+\frac{\sigma}{2}\|(S\ \mathrm{diag}(r_k)+U_k)^{(i)}-M_k^{(i)}\|_2^2\right\}. \tag{5.22}$$

The problem (5.22) has a closed form solution (see e.g. [4, 89, 90]), which is expressed as

$$\hat{M}_k^{(i)}=\left(1-\frac{\gamma_1/\sigma}{\|(S\ \mathrm{diag}(r_k)+U_k)^{(i)}\|_2}\right)_+(S\ \mathrm{diag}(r_k)+U_k)^{(i)}.$$

Therefore, the minimizer of (5.20) for fixed k can be expressed as

$$\hat{M}_k=\begin{bmatrix}\left(1-\frac{\gamma_1/\sigma}{\|(S\ \mathrm{diag}(r_k)+U_k)^{(1)}\|_2}\right)_+(S\ \mathrm{diag}(r_k)+U_k)^{(1)}\\ \left(1-\frac{\gamma_1/\sigma}{\|(S\ \mathrm{diag}(r_k)+U_k)^{(2)}\|_2}\right)_+(S\ \mathrm{diag}(r_k)+U_k)^{(2)}\\ \vdots\\ \left(1-\frac{\gamma_1/\sigma}{\|(S\ \mathrm{diag}(r_k)+U_k)^{(N_s)}\|_2}\right)_+(S\ \mathrm{diag}(r_k)+U_k)^{(N_s)}\end{bmatrix}. \tag{5.23}$$

We now solve the C-subproblem of optimization problem (5.10), which is expressed as

$$\arg\min_C\ g(S,C,R,M,P)=\arg\min_C\ \lambda\sum_{i=1}^{N_t}\sum_{k=1}^{K}r_{i,k}\|s_i-c_k\|^2+\gamma_2\sum_{k=1}^{K}\|c_k\|_1. \tag{5.24}$$

Suppose $\Lambda=\mathrm{diag}(\mathbf{1}^TR)$. The optimization problem (5.24) can be reformulated as

$$\min_C\ \mathrm{trace}\left(C(\lambda\Lambda)C^T-2\lambda SRC^T\right)+\gamma_2\|C\|_{1,1}. \tag{5.25}$$

The diagonal matrix Λ is positive definite ($r_{i,k}>0$), invertible, and $\sqrt{\Lambda}=\mathrm{diag}(\sqrt{\Lambda_{i,i}})$. Let $P=\lambda\Lambda$. Problem (5.25) takes the following form

$$\begin{aligned}&\min_C\ \mathrm{trace}\ \left(CPC^T-2\lambda SRC^T\right)+\gamma_2\|C\|_{1,1}\\ =\ &\min_C\ \mathrm{trace}\ \Big[C\sqrt{P}\sqrt{P}C^T-\lambda C\sqrt{P}(\sqrt{P})^{-1}R^TS^T-\lambda SR(\sqrt{P})^{-1}\sqrt{P}C^T+\end{aligned}$$

$$
\begin{aligned}
& \qquad SR(\sqrt{P})^{-T}(\sqrt{P})^{-1}R^T S^T\Big] + \gamma_2\|C\|_{1,1} \\
= \;& \min_C \text{ trace } \Big[C\sqrt{P} - \lambda SR(\sqrt{P})^{-1}\Big]\Big[\sqrt{P}C^T - \lambda(\sqrt{P})^{-1}R^T S^T\Big] + \gamma_2\|C\|_{1,1} \\
= \;& \min_C \text{ trace } \Big[\sqrt{P}C^T - \lambda(\sqrt{P})^{-1}R^T S^T\Big]^T \Big[\sqrt{P}C^T - \lambda(\sqrt{P})^{-1}R^T S^T\Big] \\
& \qquad +\gamma_2\|C\|_{1,1}.
\end{aligned} \tag{5.26}
$$

Since $\|A\|_F^2 = \text{trace } (A^T A)$ for $A \in \mathbb{R}^{m\times n}$, the optimization problem (5.26) can be expressed as

$$
C := \arg\min_C \|\sqrt{P}C^T - \lambda(\sqrt{P})^{-1}R^T S^T\|_F^2 + \gamma_2\|C\|_{1,1}. \tag{5.27}
$$

The problem (5.27) is a ℓ_1 regularized quadratic programming, and strictly convex and hence there exists a unique solution. Furthermore, it can be efficiently solved by many well developed benchmark ℓ_1 solvers discussed in Chapter 2.

After updating all the variables S, C, R, and M, the Lagrange multipliers P_k are updated by the following rule

$$
P_k = P_k + \sigma(S \text{ diag}(r_k) - M_k). \tag{5.28}
$$

In summary, the overall ADMM framework for solving the optimization problem (5.10) is presented in Algorithm 5.2 with the following initializations: S is solved using FISTA for the inverse problem (5.3), the landmark C is obtained using K-means clustering algorithm [91] on initialized S, and R is obtained by (5.19) using the initialized S and C.

Algorithm 5.2 The ADMM framework for the proposed ESI model

Input: lead field matrix $L \in \mathbb{R}^{N_c \times N_s}$, preprocessed EEG signal matrix $X \in \mathbb{R}^{N_c \times N_t}$, tolerance ϵ, and positive parameters $\alpha, \lambda, \gamma_1, \gamma_2, \gamma_3$, σ;

Output: solution of the source localization problem (5.10).

1: **Initialize**: S, C, R, $P_k = 0$, $M_k = 0$ for all k, and $t = 1$.
2: **while** $\left| g(S^{(t+1)}, C^{(t+1)}, R^{(t+1)}, M^{(t+1)}, P^{(t+1)}) - g(S^{(t)}, C^{(t)}, R^{(t)}, M^{(t)}, P^{(t)}) \right| > \epsilon$ **do**
3: Update S by solving LASSO problem (5.13);
4: Update C by solving LASSO problem (5.27);
5: Update R using PGD as described in Algorithm 5.1;
6: Update M_k using (5.23) for each k;
7: Update the dual variables P_k using (5.28) for each k;
8: $t = t + 1$;
9: **end while**
10: **return** last $S^{(t)}$.

5.5 Numerical Experiments

In this section, we discuss numerical simulation results of optimization problem (5.10) for our proposed ESI model. We compare the performance of our method to other popular ESI methods on simulated data to illustrate its effectiveness. Since each variable updates of Algorithm 5.2 are independent, we parallelize our code using a parallel toolbox of MATLAB to speed up the numerical simulations. We consider different levels of noise in sensors and sources in terms of SNR values. We perform the visual studies of our model in the real head model template in Brainstorm [73] GUI available in MATLAB.

5.5.1 Experiment setting

In this subsection, we provide the description of the error metrics used to compare the results of the proposed method to other popular ESI methods. We consider three different benchmark methods for comparison, namely minimum current estimates (MCE) [66], minimum norm estimates (MNE) [62], and mixed norm estimates (MxNE) [4].

1. Minimum Norm Estimates (MNE): MNE uses the ℓ_2 penalty to estimate the solution of the underdetermined linear system

$$\begin{aligned} S_{\mathrm{MNE}} &= \underset{S\in\mathbb{R}^{N_s\times N_t}}{\arg\min} \|X - LS\|_F^2 + \alpha\|S\|_F^2 \\ &= \underset{s_i\in\mathbb{R}^{N_s}}{\arg\min} \sum_{i=1}^{N_t} \|x_i - Ls_i\|_2^2 + \alpha \sum_{i=1}^{N_t} \|s_i\|_2^2, \end{aligned} \tag{5.29}$$

where $\alpha > 0$ is a penalty parameter. The objective function (5.29) is differentiable, and S_{MNE} can be uniquely determined as

$$S_{\mathrm{MNE}} = (L^T L + \alpha I)^{-1} L^T X, \tag{5.30}$$

where $I \in \mathbb{R}^{N_s\times N_s}$ is an identity matrix. In Statistics, the solution of MNE objective (5.29) is referred to an estimator of the Ridge regression [32]. The ℓ_2-norm based method usually provides an over-diffused solution which we will observe in our numerical simulation.

2. Minimum Current Estimates (MCE): To capture the sparsely activated brain sources, Uutela, Hämäläinen, and Somersalo [66] used the ℓ_1-norm based penalty to estimate the solution of the underdetermined linear system

$$\begin{aligned} S_{\mathrm{MCE}} &= \underset{S\in\mathbb{R}^{N_s\times N_t}}{\arg\min} \|X - LS\|_F^2 + \lambda\|S\|_{1,1} \\ &= \underset{s_i\in\mathbb{R}^{N_s}}{\arg\min} \sum_{i=1}^{N_t} \|x_i - Ls_i\|_2^2 + \lambda \sum_{i=1}^{N_t} \|s_i\|_1, \end{aligned} \tag{5.31}$$

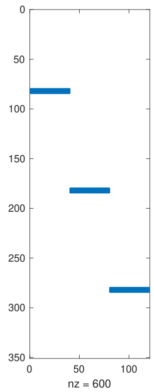
where $\lambda > 0$ is a penalty parameter controlling sparsity. The objective function of (5.31) is not differentiable. But, we can estimate each column of S_{MCE} using ℓ_1-minimization algorithms discussed in Chapter 2. In statistics, MCE (5.31) is called the LASSO problem.

3. Mixed Norm Estimates (MxNE): MxNE promotes spatially focal sources with smooth temporal estimates with a two-level ℓ_1/ℓ_2 mixed-norm. It also uses a three-level mixed-norm to promote spatially non-overlapping sources between different experimental conditions. To be specific, we are considering a bilevel mixed-norm, namely, the ℓ_1-norm on the source space and ℓ_2-norm across the time space. The reader can explore different bilevel and multilevel mixed-norm penalties in [4]. In our comparison studies, MxNE refers to the following optimization problem:

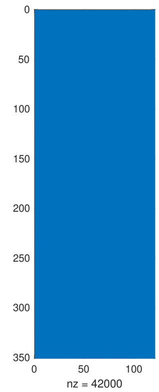
$$S_{\mathrm{MxNE}} = \underset{S \in \mathbb{R}^{N_s \times N_t}}{\arg\min} \|X - LS\|_F^2 + \beta \|S\|_{2,1}, \tag{5.32}$$

where $\beta > 0$ is a penalty parameter controlling the group sparsity. In order to solve (5.32), we need to solve the $\ell_{2,1}$ based proximal operator problem. MxNE promotes sparsity along the rows in recovery. The reader is referred to [4] for a theoretical derivation of the $\ell_{2,1}$ based proximal operator.

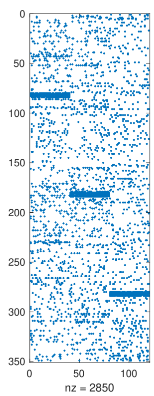
The results of source estimation in Figure 5.4 explain how different norms used in MCE, MNE, and MxNE change sparsity in solution.



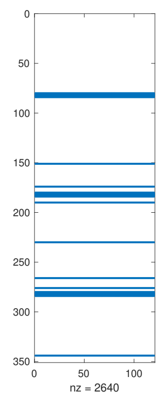
(a) Ground truth.



(b) Solution by MNE.



(c) Solution by MCE.

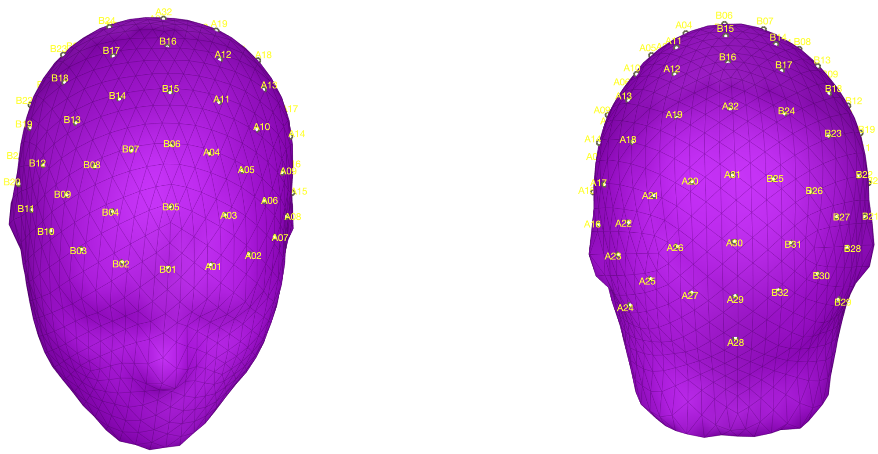


(d) Solution by MxNE.

Figure 5.4: Sparsity comparison of different ESI methods.

5.5.2 Description of Data and Parameters

In our numerical simulations, we use synthetic data to validate the performance of our model. We consider measurement matrix $X \in \mathbb{R}^{64\times 120}$ with 64 channels (N_c) and 120 time points (N_t) in milliseconds, and lead field matrix L that maps 350 brain sources (N_s) to 64 channels, i.e., $L \in \mathbb{R}^{64\times 350}$. The proposed model attempts to recover 350 brain sources at 120 time points, i.e., $S \in \mathbb{R}^{350\times 120}$. In our synthetic data, the ground truth matrix S has a row-wise block pattern as shown Figure 5.4-(a). We consider these patterns in S to represent region-wise source activations in the brain at different time instances, i.e., 0-40, 40-80, 80-120 milliseconds respectively. We expect our proposed model would successfully capture such activation patterns. We use standard BioSemi Neuroscan cap for 64 channels under the ICBM152 template in Brainstorm to present our visual study. The electrodes layout on the head surface is presented in Figure 5.5.



(a) Electrodes layout (in front view). (b) Electrodes layout (in back view).

Figure 5.5: EEG channel layout of BioSemi Neuroscan cap with 64 channels (in front and back views).

We perform simulation studies with different parameters involved in our proposed model. Our simulation studies showed that the proposed model is not highly sensitive to the parameters, where $\lambda \in \{0.001, 0.0001\}$, $\alpha \in \{0.01, 0.001\}$, $\gamma_1 \in \{0.1, 0.001\}$, $\sigma \in \{0.02, 0.2, 0.8\}$, and $\gamma_2 = \gamma_3$ with $\gamma_2 \in \{0.001, 0.01, 0.3\}$. The reconstruction is not sensitive to cluster number K in landmarks, i.e., the column number of matrix C. In our simulation, $K \in \{5, 10, 15\}$. We choose the penalty parameter for MNE $\alpha = 0.4$, for MCE $\lambda = 0.5$, and for MxNE $\beta = 0.01$, respectively.

We compute S_{MNE} as in (5.30) for the solution by MNE. The ℓ_1-homotopy algorithm is used to estimate the solution of MNE due to its superior performance. The solution estimates by MxNE is found using Scikit learn package for solving the multi-task LASSO problem in Python.

5.5.3 Model Validation Metrics

In this subsection, we discuss the metrics that we use to evaluate the performances of the proposed method and other competing methods.

1. Reconstruction Error (RE): This metric is used to evaluate the performances of the different ℓ_1-minimization algorithms in Chapter 3 and algorithms for solving the Sylvester LASSO problem in Chapter 4. A reconstruction is considered good if its RE value is close to zero.
2. Data Fitting (DF): In regression analysis, DF is a measurement of fitness for the regression model. To understand it in our context, consider the following linear relation in our EEG problem

$$X = LS + \mathcal{E}, \tag{5.33}$$

where $X \in \mathbb{R}^{N_c \times N_t}$, $L \in \mathbb{R}^{N_c \times N_s}$ $(N_c \ll N_s)$, $S \in \mathbb{R}^{N_s \times N_t}$, and $\mathcal{E} \in \mathbb{R}^{N_c \times N_t}$ is a measurement error. The linear model (5.33) is underdetermined, and we impose different regularizations to estimate its solution $\hat{S}$. We calculate the sum square total (SST) value to check the total deviation of the signals from each channel (x_i) to the mean $(\bar{x})$ of the signals along the time axis:

$$E_{\text{tot}} = \sum_{i=1}^{N_t} \|x_i - \bar{x}\|_2^2.$$

We also measure the total deviation of the signals from each channel (x_i) to the corresponding fitted value $\hat{x}_i = L\hat{s}_i$, i.e., the residual sum square (RSS):

$$E_{\text{res}} = \sum_{i=1}^{N_t} \|x_i - \hat{x}_i\|_2^2.$$

Finally, the value of the fitness or DF value (r^2), which describes the model fitting is defined as

$$r^2 = \left|1 - \frac{E_{\text{res}}}{E_{\text{tot}}}\right|.$$

In the best scenario, when the estimated EEG signals $\hat{x}_i$ exactly match the observed EEG signals x_i, E_{res} is zero and thus $r^2 = 1$. In general, the closer r^2 to 1, the better the model fitting.

3. Area under curve (AUC): A receiver operating characteristic (ROC) curve is one of the popularly used model validation techniques for evaluations of machine learning (ML) models [1, 20, 92]. The area under the ROC curve is called AUC which determines the accuracy of the ML models. We use this model validation technique in our context to assess the source detection accuracy of the proposed model. To construct the ROC curve, we need to understand the notion of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). These are statistical concepts and their detailed descriptions can be found in any of [1, 20, 92]. Suppose p_i denote the i-th brain source in source space and t_i denote the i-th time point in time space.

   - True Positive (TP): If p_i is activated at t_i and the proposed model predicts the same outcome, then it is called a true positive.
   - True Negative (TN): If the source p_i is not activated at t_i and the proposed model predicts the same outcome, then it is called a true negative.

- False Positive (FP): If the source p_i is not activated at t_i but the proposed model predicted that p_i is activated at t_i, then it is called a false positive. This type of prediction inaccuracy is also called a type I error.
- False negative (FN): If the source p_i is activated at t_i but the proposed model predicted that p_i is not activated at t_i, then it is called a false negative. This type of prediction inaccuracy is also called a type II error.

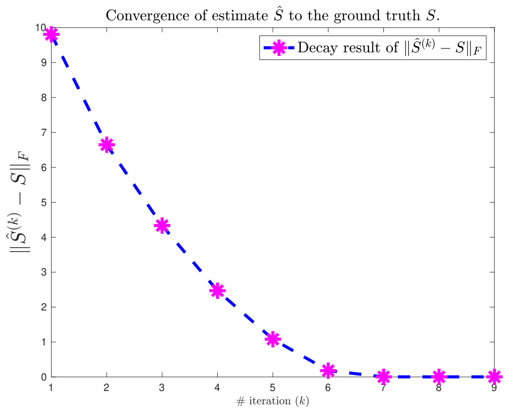
We define the true positive rate (TPR) and the false positive rate (FPR) in terms of TP, TN, FP, and FN as follows:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$
$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}.$$

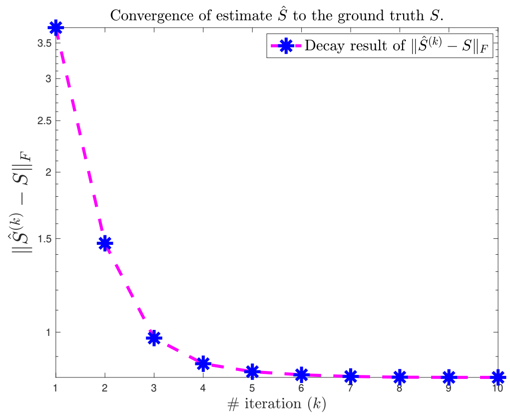
The ROC curve is then created by plotting TPR against FPR. In the case of 100% prediction accuracy, we get the value of AUC 1. In general, the closer AUC to 1, the better the prediction accuracy.

5.5.4 Convergence of the Proposed Model

In this subsection, we discuss the convergence behavior of our proposed ESI model. For this, we calculate the norm difference between the estimated solution $\hat{S}^{(k)}$ in each step and the ground truth S, i.e., $\|\hat{S}^{(k)} - S\|_F$. In Figure 5.6, we plot $\|\hat{S}^{(k)} - S\|_F$ against iteration k of the proposed model for clean and noisy synthetic data. In both plots (a) and (b) of Figure 5.6, algorithm stops when it meets the stopping criteria $\left|\|\hat{S}^{(k+1)} - S\|_F - \|\hat{S}^{(k)} - S\|_F\right| < 10^{-4}$. The results in Figure 5.6 (a) and (b) are obtained using clean and noisy data with noise level 30 dB in channels, respectively. The result shows that the estimation of the proposed ESI model in each successive iteration quickly converges to the ground truth. For clean data, the norm difference $\|\hat{S}^{(k)} - S\|_F$ goes to zero faster than for the case of noisy data, indicating the adverse effect of noise in solution reconstruction.



(a) Convergence result for clean data.



(b) Convergence result for noisy data.

Figure 5.6: Convergence of the proposed ESI method on clean and noisy data.

In our proposed model (5.10), we have variables to update, namely S, C, R, M, and P. In Figure 5.7, we record the objective function values after each variable update and plot them against the successive iterations. All of the five updates contribute to decrease the objective function value monotonically in each successive iterations.

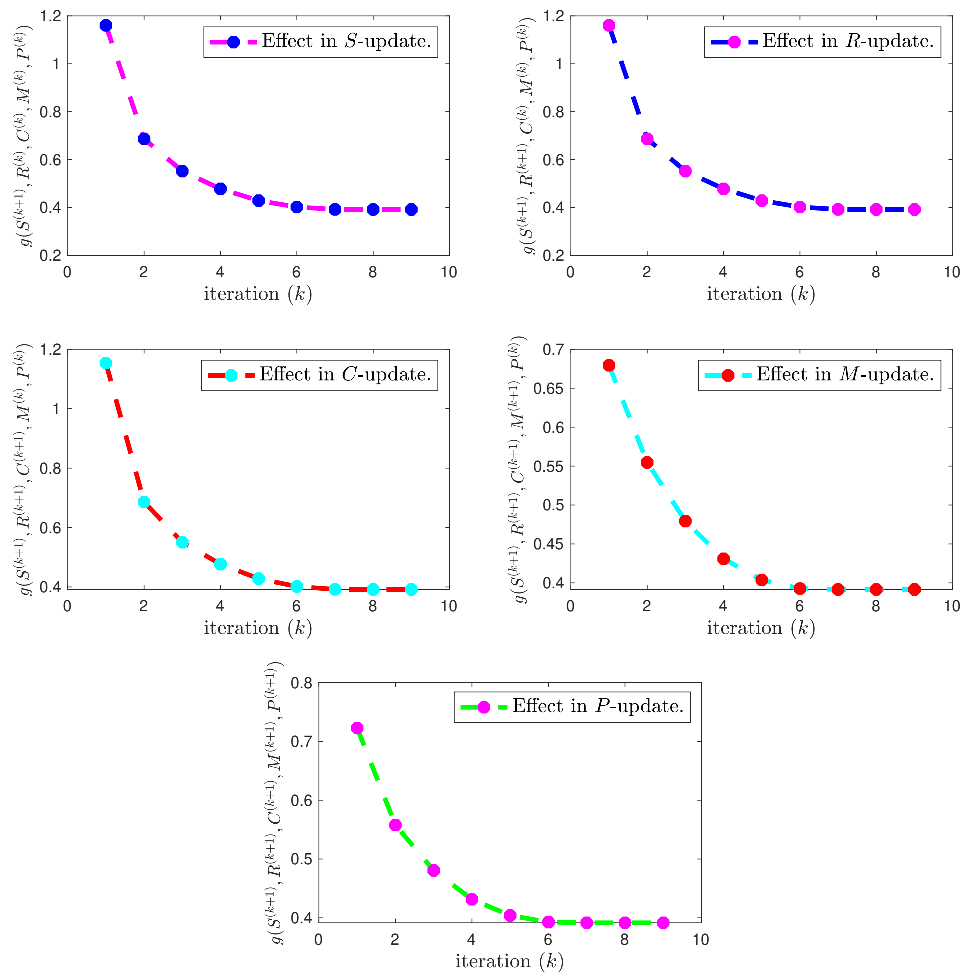


Figure 5.7: Evolution of the objective function values evaluated at each variable updates in successive iterations.

5.5.5 Simulation Results

In this subsection, we present numerical simulation results of our proposed method in terms of model validation metrics DF, RE, and AUC. To calculate the AUC value for each method, the average of the AUC values from each column recovery of the solution estimate $\hat{S}$ is used. We calculate DF and RE using the formulas explained in Subsection 5.5.3. We compare the performance of the proposed model and three different competing ESI models. First, we compare the performance of the proposed model with MNE, MCE, and MxNE on clean data. The summary of the performance results is presented in Table 5.1 and the best results are highlighted. We see that the proposed model outperforms other competing ESI methods. The proposed model captures the block structure of the synthetic data very well and recovers the solution with a reconstruction error only 0.02%. In terms of DF and AUC values, the proposed model performed very well. Moreover, MCE has comparable performance to our proposed ESI model.

SNR = ∞ dB (Noiseless)			
Algorithms	DF	AUC	RE
MCE	**1.0000**	0.9905	0.0043
MNE	0.9999	0.9018	0.9053
MxNE	0.9757	0.9997	0.0495
proposed	**1.0000**	**1.0000**	**0.0002**

Table 5.1: Quality of source reconstructions in different error metrics for clean data. The parameter values used in the proposed model are $\alpha = 0.01, \gamma_1 = 0.001, \gamma_2 = \gamma_3 = 0.001, \lambda = 0.0001$, and $\sigma = 0.02$.

The EEG signals are contaminated with noise during the signal acquisition process in the real application. We consider three different noise levels in channels, smaller to larger, namely SNR = 30 dB, 20 dB, and 10 dB. To understand the contamination of the noise to the clean signal in different noise levels, we plot the clean signal acquired from the 40-th channel in the first 60 time points and the noisy signal with different noise levels from the same channel in Figure 5.8. According to the definition of SNR large noise corresponds to small SNR value.

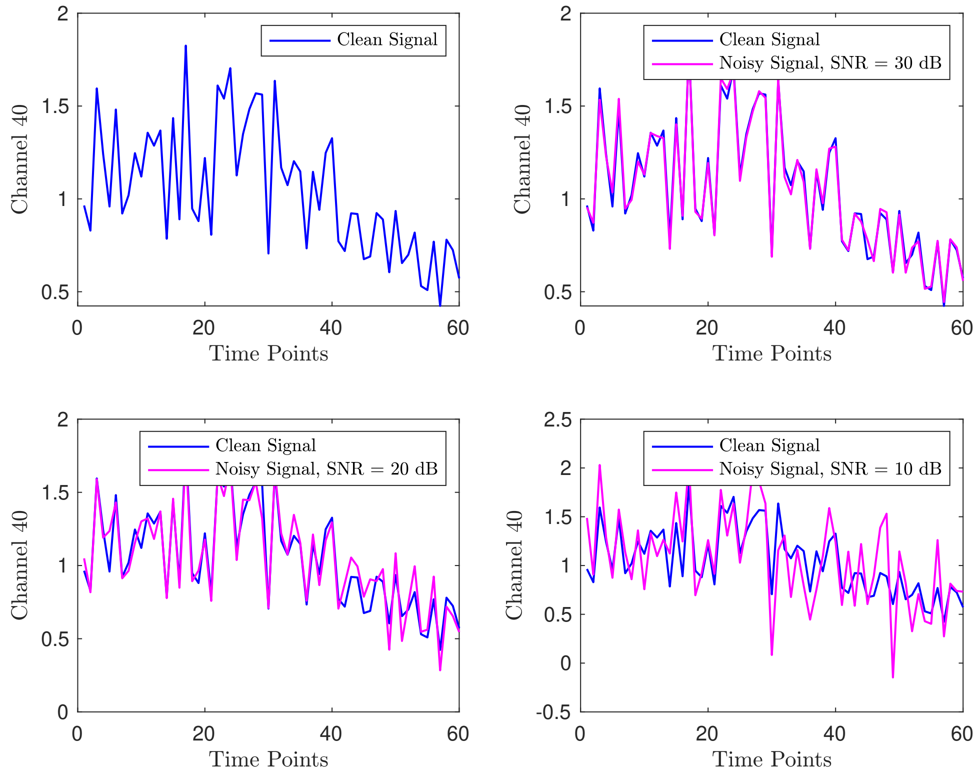


Figure 5.8: Clean data vs. noisy data at different noise levels.

We present a summary of the performance of competing methods and the proposed method in Table 5.2 for the data having noise in channels with three different noise levels. MNE has a highly diffusive solution that produces largest DF values but poorest AUC and RE in all three cases. In our synthetic data, the true source has non zero block structures which MxNE is capable to capture well, and thus results in the largest AUC in all three cases. In terms of model fitting and reducing the reconstruction error, MxNE is weaker compared to the proposed method. For all three noise levels, the proposed method has superior performance on capturing the block structures of the data, superior model fitting, and superior reconstruction error.

Methods	SNR = 30 dB			SNR = 20 dB			SNR = 10 dB		
	DF	AUC	RE	DF	AUC	RE	DF	AUC	RE
MCE	0.997	0.985	0.057	0.974	0.954	0.223	0.989	0.845	0.856
MNE	**1.000**	0.903	0.906	**1.000**	0.890	0.913	**1.000**	0.841	0.980
MxNE	0.921	**0.998**	0.103	0.563	**0.992**	0.300	0.721	**0.955**	0.717
proposed	0.996	0.988	**0.057**	0.970	0.956	**0.194**	**1.000**	0.936	**0.500**

Table 5.2: Quality of source reconstructions in different error metrics for the data with different noise levels. The parameter values used in the proposed model are as follows: for SNR = 10 dB, $\alpha = 0.01, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.01, \lambda = 0.01$, and $\sigma = 0.1$; for SNR = 20 dB, $\alpha = 0.01, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.6, \lambda = 0.01$, and $\sigma = 0.2$; for SNR = 30 dB, $\alpha = 0.01, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.3, \lambda = 0.001$, and $\sigma = 0.8$.

We plot the ROC curves in Figure 5.9 for one column recovery of the noisy data having noise level SNR = 20 dB. When the data has noise level SNR = 20 dB, MxNE has the largest AUC observable as having the largest area under the ROC curve in Figure 5.9.

Many ESI methods only consider noise in channels and discard the presence of noise in sources. We perform numerical experiments by considering additive white noise in both sources and sensors at different noise levels. We denote the level of noise in channels by $\mathrm{SNR_C}$ and in sources by $\mathrm{SNR_S}$ for a clear distinction. We perform the numerical experiments by keeping a fixed level of noise in channels and varying different levels of noise in sources. In the next two experiments, we fix two levels of noise $\mathrm{SNR_C}$ = 30 dB and $\mathrm{SNR_C}$ = 20 dB in channels and vary $\mathrm{SNR_S}$ = 30 dB, and 20 dB in sources. Experimented results are summarized in Table 5.3 and Table 5.4.

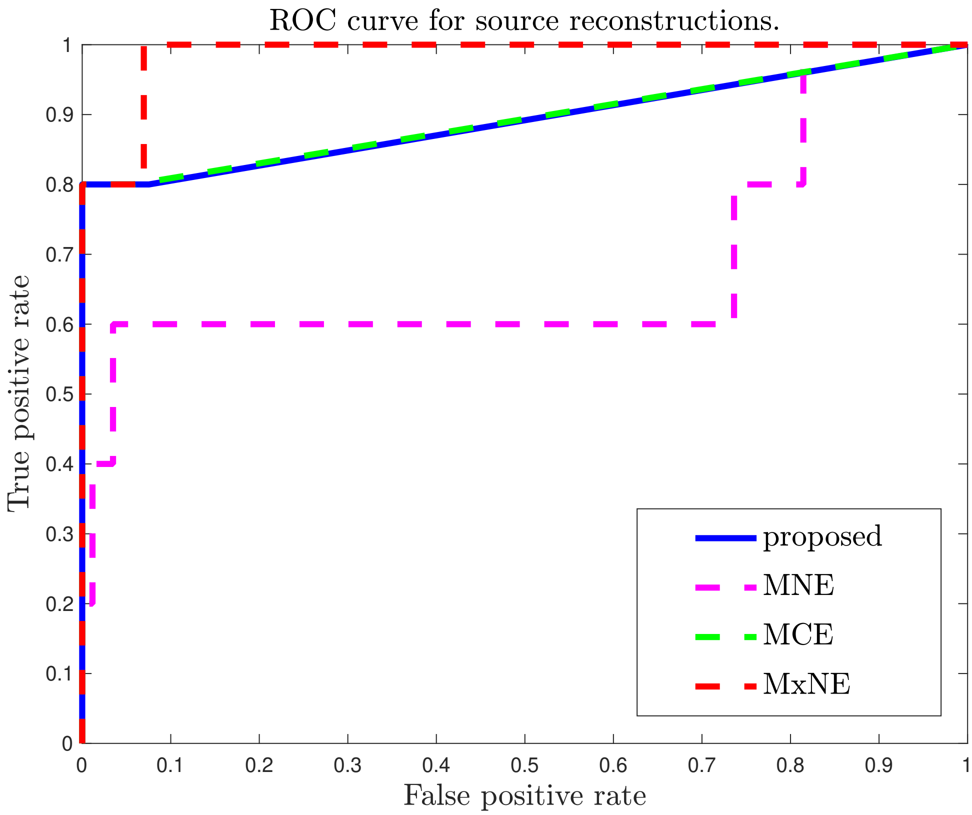


Figure 5.9: ROC curves by different ESI models.

$\mathrm{SNR_C}$ = 30 dB						
Algorithms	$\mathrm{SNR_S}$ = 30 dB			$\mathrm{SNR_S}$ = 20 dB		
	DF	AUC	RE	DF	AUC	RE
MCE	0.987	0.926	0.471	0.997	0.746	1.324
MNE	**1.000**	0.882	0.927	**1.000**	0.785	1.091
MxNE	0.106	**0.982**	0.468	0.147	**0.903**	0.987
proposed	**1.000**	0.974	**0.253**	**1.000**	0.858	**0.938**

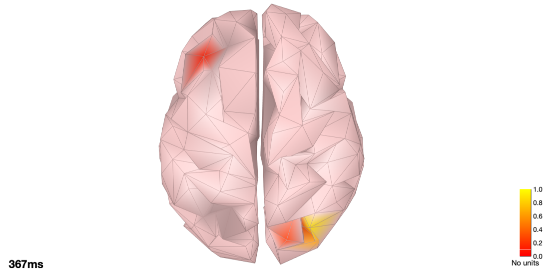
Table 5.3: Quality of source reconstructions by different ESI algorithms in different error metrics for synthetic data with noise in channels 30 dB, noise in sources 30 dB, and 20 dB. The parameter values in the proposed model: for $\mathrm{SNR}_S = 20$ dB, $\alpha = 0.001, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.01, \lambda = 0.1$, and $\sigma = 0.01$, and for $\mathrm{SNR}_S = 30$ dB, $\alpha = 0.01, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.01, \lambda = 0.1$, and $\sigma = 0.01$.

In Table 5.3, we present the numerical results of the ESI methods with SNR_C = 30 dB and varying noise in sources from SNR_C = 30 dB to SNR_C = 20 dB. For relatively large noise SNR_C = 20 dB in channels and vary noise in the sources from SNR_C = 30 dB to SNR_C = 20 dB, the results are presented in Table 5.4.

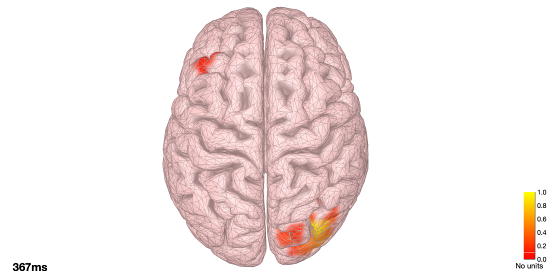
SNR_C = 20 dB						
Algorithms	SNR_S = 30 dB			SNR_S= 20 dB		
	DF	AUC	RE	DF	AUC	RE
MCE	0.986	0.902	0.534	0.998	0.727	1.330
MNE	**1.000**	0.876	0.933	**1.000**	0.773	1.095
MxNE	0.260	**0.980**	0.506	0.062	**0.890**	1.002
proposed	**1.000**	0.960	**0.287**	**1.000**	0.848	**0.931**

Table 5.4: Quality of source reconstructions by different ESI algorithms under different error metrics for synthetic data with noise in channels 20 dB, noise in source 30 dB, and 20 dB. The parameter values in the proposed model: for SNR_S = 20 dB and 30 dB, $\alpha = 0.5, \gamma_1 = 0.1, \gamma_2 = \gamma_3 = 0.01, \lambda = 0.1$, and $\sigma = 0.01$.

The results in Table 5.3, and Table 5.4 show the superior performance of the proposed model in source reconstruction at different noise levels. MxNE keeps row sparsity in reconstruction sources, so it is successful to capture the block pattern in data that is reflected in AUC in both tables. The model fitting (DF) values for MxNE are also lower compared to others. MCE has superior model fitting (DF) values due to its higher diffusive solution. However, smaller value of the prediction accuracy (AUC) and higher reconstruction error make MNE less favorable compared to other ESI methods. In nutshell, the proposed model has the best performance among all in terms of reconstruction error (RE) and data fitting (DF) values and comparable AUC to MxNE for data with bilevel noises.

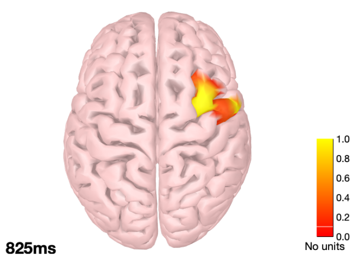


(a) Real source recovery plot having 350 triangular vertices.

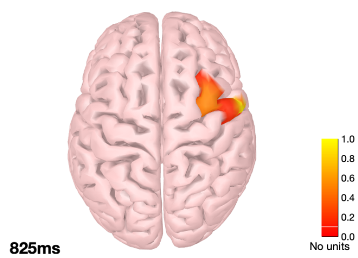


(b) Corresponding projected source space with 15002 triangular vertices available in default anatomy in Brainstorm.

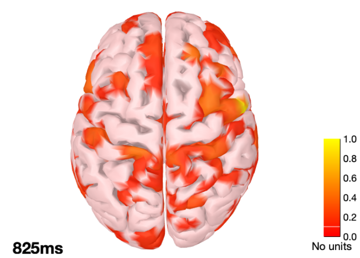
Figure 5.10: Figure (a) displays the cortical region and source activation based on synthetic data having 350 brain voxels. Figure (b) shows the corresponding projected source in the high resolution cortical region having 15002 triangular vertices.



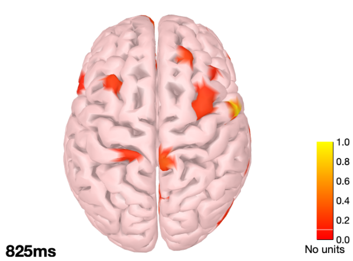
(a) Ground Truth



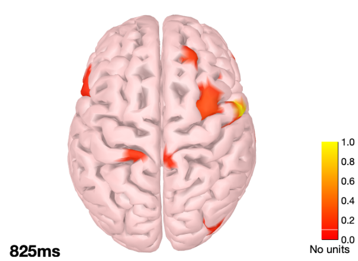
(b) By proposed



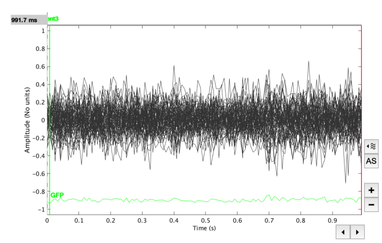
(c) By MNE



(d) By MCE



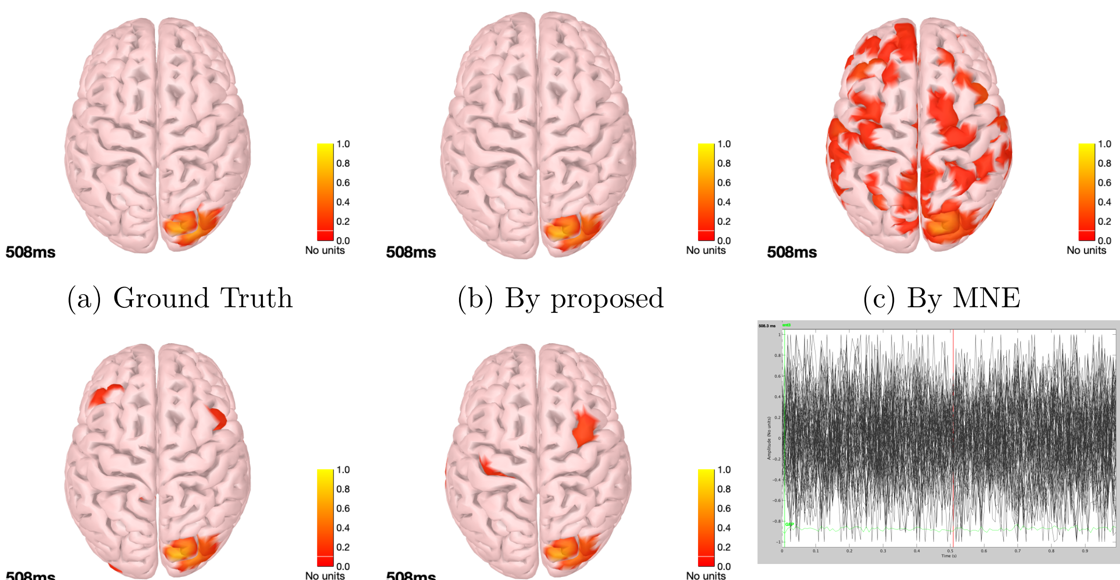
(e) By MxNE



(f) $\text{SNR}_C = 10$ dB

Figure 5.11: Source recovery plots by different ESI methods with $\text{SNR}_C = 10$ dB and $\text{SNR}_S = \infty$.

We visualize the numerical results shown in the above tables via the brain plot to examine the effectiveness of our proposed model. Our synthetic data considers only 350 number of sources which are represented by 350 triangular meshes in cortical regions. The brain plot results have low resolution image because of fewer number of triangular brain meshes in real head model. In order to achieve high quality brain plot, we project our source recovery data to the source template having 15002 triangular vertices available in Brainstorm default anatomy, as depicted in Figure 5.10.

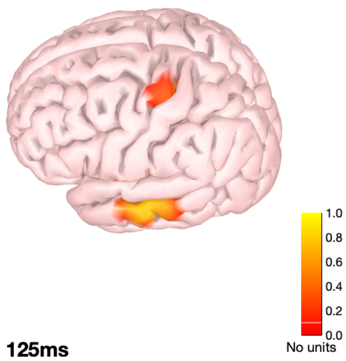


(a) Ground Truth (b) By proposed (c) By MNE

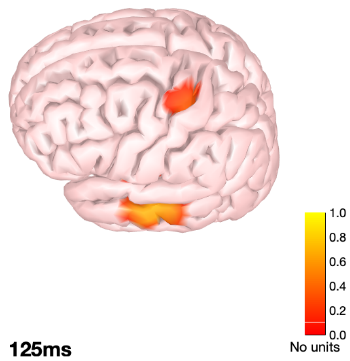
(d) By MCE (e) By MxNE (f) SNR_C = 30 dB, SNR_S = 30 dB

Figure 5.12: Source recovery plots by different ESI methods with $\text{SNR}_C = 30$ dB and $\text{SNR}_S = 30$ dB.

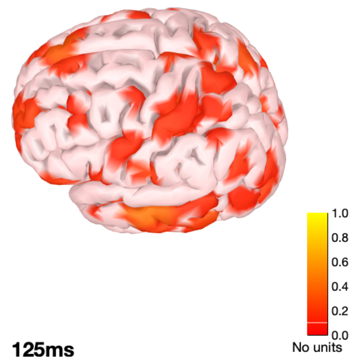
In Figure 5.11, we visualize the source reconstruction results of the simulation considering noise in channels only with $\text{SNR}_C = 10$ dB, and $\text{SNR}_S = \infty$ dB for which numerical results are shown in Table 5.2. In Figure 5.11 (a), activation of the ground truth in 825 mili seconds is shown. The rest of the brain plots in Figure 5.11 (b)-(e) are for the source recovery by the proposed method, MNE, MCE, and MxNE, respectively. The last plot, Figure 5.11 (f), presents the recorded EEG signals contaminated with noise having noise level $\text{SNR}_C = 10$ dB.



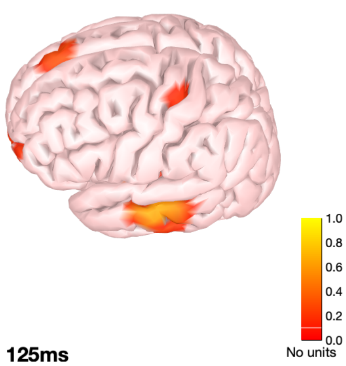
(a) Ground Truth



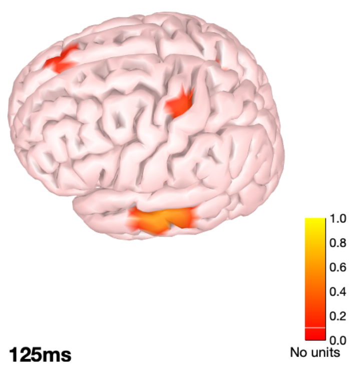
(b) By proposed



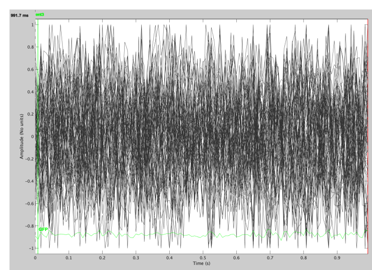
(c) By MNE



(d) By MCE



(e) By MxNE



(f) SNR_C = 20 dB, SNR_S = 30 dB

Figure 5.13: Source recovery plots by different ESI methods with $\text{SNR}_C = 20$ dB and $\text{SNR}_S = 30$ dB.

Similarly, in Figure 5.12 and Figure 5.13, we visualize the results for noise in both channels and sources at different noise levels. In any case, whether we consider noise in channels only or both channels and sources, source reconstruction by the proposed method is always more accurate than any other competing ESI methods.

The proposed method is significantly better when the data has high noise levels in sources and channels.

In conclusion, all simulation results show that the proposed model is successful to capture activation pattern in brain. When noises are presented in both sources and channels, the proposed model has superior performance in source reconstruction. The proposed model is successful in recognizing the precise source activation in all the cases. It has the best model fitting (DF), prediction accuracy (AUC), and less reconstruction error (RE) values for clean and noisy data compared to other popular state-of-the-art ESI models. The ESI problem with noise in both sources and channels was less studied, and this research fills that gap.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Summary

In this section, we summarize the mathematical optimization methods that we proposed in this dissertation to solve the inverse problem arising in electroencephalography source imaging (ESI). We studied the uses of different matrix norms and their abilities to fulfill different neurophysiological assumptions in the ESI problem.

We studied the Sylvester LASSO problem in Chapter 4 that uses the ℓ_1-norm minimization technique to recover the solution of the inverse problem under sparse prior. We find that this type of ℓ_1-minimization technique can be used to solve the EEG source imaging problem by minimizing the total variation of activated source signals in the brain. The conversion of the Sylvester LASSO to standard LASSO problem brings many computational challenges. We explore the technique to handle large coefficient matrix $\mathcal{M}$ and long vector $\mathcal{V}$ by extracting the block structure that makes problem solvable in personal computers with limited memory. We use iterative reweighting techniques to better solve the following weighted Sylvester LASSO problem

$$\underset{X(:)\in\mathbb{R}^{mn\times 1}}{\text{minimize}} \ \frac{1}{2}\|\mathcal{M}X(:)-\mathcal{V}\|_2^2+\|WX(:)\|_1. \tag{6.1}$$

We use FISTA to solve each reweighted problem (6.1) which converges quickly to the solution. Only 3 to 4 numbers of reweighting steps achieve the reconstruction error up to 24%. The sparsity recovery in the solution is very close to the ground truth solution for synthetic data. From all numerical simulations based on synthetic data, the proposed method for solving (6.1) is very efficient. The proposed method is robust on recovering the solution with smaller to high noise data. In this regard, our proposed model can be a good option to consider solving the Sylvester LASSO problem.

The precise source reconstruction helps cure several brain disorders. The activities inside the brain are always related to certain human behaviors. Detection of the source activation pattern is the primary task at any mathematical model for ESI. The proposed ESI model in Chapter 5 can effectively capture the block-wise activation pattern of the brain sources. The mathematical optimization problem developed in Chapter 5 can be solved efficiently by ADMM. Each matrix variable update can be decoupled and parallel computation is used to speed up the computations. The proposed method outperforms state-of-the-art ESI methods. The proposed method deals with data having noise in the source space and channels effectively. The plots of source localization results in a real head model by the proposed model show correct detections of the activated source locations very precisely compared to other competing ESI methods.

The research presented in this dissertation to model the ESI problem with new realistic neurophysiological assumptions to capture the block-wise source activation pattern of the brain fills the gap in the research in ESI. The proposed model captures the source activation pattern efficiently. In dealing with bilevel noises in ESI problems, the proposed model is a good choice compared to other popular ESI methods.

According to the Global Burden of Disease Study 2010 (GBD 2010)[1] report, a substantial proportion of the world's diseases came from mental and neurological disorders. In the global population, mental disorders accounted for the largest proportion of Disability-adjusted life years (DALYs) (56.7%), followed by 28.6% of

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4320057/

neurological disorders. In the U.S., WHO estimates [2] 18.7% of DALYs are in the category of neuropsychiatric disorders followed by 13.6% mental and 5.1% neurological disorders for which the U.S. alone spends $15.5 billion each year for neurological treatment. In this regard, research in brain source imaging has a high impact on medical practice to detect any brain disorder accurately and reduces the cost of diagnosis and treatment of patients. We believe that the model we developed to answer some practical neurophysiological conditions will contribute to the literature to develop efficient technology in this field.

6.2 Future Work

The research in this dissertation is motivated by the quest for better mathematical optimization models and tools for solving ESI inverse problems. We would like to contribute to a diverse spectrum of mathematical questions and challenges related to data science, image reconstruction, medical imaging, remote sensing, etc. The mathematical optimization plays a key role in wide areas of data science applications. Many machine learning tasks have a very similar structure to inverse problems, such as matrix completion, classification, clustering, and segmentation. We would like to enhance the ideas of convex and non-convex optimization techniques to broaden the applicability of our research in recent trends and applications in data science.

Hyperspectral imaging [93, 94] is a popular imaging technique in remote sensing. The purpose of hyperspectral imaging is to obtain the spectrum of reflectance or radian values for each pixel in the image of a scene, with the purpose of finding objects and identifying materials (endmembers) along with their proportions (abundances). In order to accurately identify the materials presented in the scene, a spectral unmixing

[2] https://www.ncbi.nlm.nih.gov/pubmed/23842577

(SU) problem has to be solved. SU is a source separation problem whose goal is to recover the signatures (properties of the materials) of the pure materials and to estimate their relative proportions (fractional abundances) in each pixel of the image.

Let $X \in \mathbb{R}^{L \times N}$ be a hyperspectral image formed by gathering the N many pixels x_k having L number of spectral bands (feature of the pixels such as color) in the columns of X. The signatures $s_p,\ p = 1, \cdots, P$ of the P endmembers considered for the unmixing are gathered in the column of a matrix $S \in \mathbb{R}^{L \times P}$. The abundance coefficients a_{pk} for each pixel $k = 1, \cdots, N$ and material $p = 1, \cdots, P$ are stored in a matrix $A \in \mathbb{R}^{P \times N}$. The linear mixing model for the whole hyperspectral image can be expressed as

$$X = SA + E, \tag{6.2}$$

where $E \in \mathbb{R}^{L \times N}$ is noise. Since the abundances are interpreted as proportions and they are required to be positive and the sum of the abundances in each pixel is required to be one. Therefore, estimating the matrix of abundances A of the material presented in each pixel of the hyperspectral image X using the information of extracted endmembers S is often carried out by solving the following constrained optimization problem:

$$\begin{aligned} &\underset{A \in \mathbb{R}^{P \times N}}{\arg\min}\ \|X - SA\|_F^2 \\ \text{subject to} \quad &\mathbf{1}_P^T A = \mathbf{1}_N^T \\ &a_{ij} \geq 0 \quad \forall i, j, \end{aligned} \tag{6.3}$$

where $\mathbf{1}_P \in \mathbb{R}^P$ denotes the vector of ones.

The ideas of grouping the similar source activation pattern using a mixed-norm strategy proposed in the ESI model in Chapter 5 can be extended to solve the spectral unmixing problem (6.3). We would like to extend our knowledge of solving the optimization problem with simplex constraint in our proposed ESI method to solve problem (6.3) by inducing group sparsity among the similar abundance coefficients.

Answering challenges in imaging problems of several domains requires extensive knowledge of mathematical optimization, mathematical modeling skills, and synthesis of powerful novel mathematical concepts. We would like to continue working on inverse problems in imaging and on related problems in data science. The effort on advancing the mathematical research on solving the challenges in the domain of data science will be continued.

REFERENCES

[1] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, 2nd ed., ser. Springer Text in Statistics. New York: Springer Science & Business Media, 2013.

[2] M. Pasha, "Krylov subspace type methods for the computation of non-negative or sparse solutions of ill-posed problems," Ph.D. dissertation, Kent, Ohio, 2020.

[3] F. Provost and T. Fawcett, *Data Science for Business.* O'Reilly Media, Inc., 2013.

[4] A. Gramfort, M. Kowalski, and M. Hämäläinen, "Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods," *Phys Med Biol.*, vol. 57, no. 7, pp. 1937–1961, 2012.

[5] E. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 2, pp. 589–592, 2008.

[6] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, 2006.

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

[8] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[9] M. Lustig, D. Donoho, and J. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, pp. 1182–1195, 2007.

[10] K. Egiazarian, A. Foi, and V. Katkovnik, "Compressed sensing image reconstruction via recursive spatially adaptive filtering," *IEEE International Conference on Image Processing*, vol. 1, pp. 549–552, 2007.

[11] M. Davenport, T. Boufounos, M. Wakin, and R. Baraniuk, "Signal processing with compressive measurements," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 445–460, 2010.

[12] D. Takhar, J. Laska, M. Wakin, M. Duarte, D. Sarvotham, K. Kelly, and R. Baraniuk, "A new compressive imaging camera architecture using optical-domain compressions," *Computational Imaging*, vol. 4, pp. 43–52, 2006.

[13] X. Yuan and R. Cohen, "Image compression based on compressive sensing: End-to-end comparison with JPEG," arXiv:1706.01000, 2020.

[14] E. Candès and J. Romberg, "ℓ_1-magic: Recovery of sparse signals via convex programming," Available at https://statweb.stanford.edu/~candes/software/l1magic/examples.html.

[15] S. Chen, "Basis pursuit," Ph.D. dissertation, CA, 1995.

[16] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[17] M. Asif and J. Romberg, "Sparse recovery of streaming signals using ℓ_1-homotopy," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4209–4223, 2014.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York: Cambridge University Press, 2007.

[19] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society*, vol. 58, no. 1, pp. 267–288, 1996.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer Science & Business Media, 2009.

[21] T. Hastie, R. Tibshirani, and M. Waiwright, *Statistical Learning with Sparsity: The LASSO and Generalizations*, ser. Monographs on Statistical & Applied Probability. New York: CRC Press, 2015, vol. 143.

[22] B. Chalmod, *Modeling and Inverse Problems in Imaging Analysis*, ser. Applied Mathematical Sciences. Springer, 2012, vol. 155.

[23] A. Mohamad-Djafari, *Inverse Problems in Vision and 3D Tomography.* John Wiley & Sons, 2013.

[24] G. Bal, "Introduction to inverse problems," Lecture Notes, Department of Applied Physics and Applied Mathematics, Columbia University, New York, 2012.

[25] P. Hansen, "Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems," *Numerical Algorithms*, vol. 6, no. 1, pp. 1–35, 1994.

[26] J. Hadamard, *Lectures on Cauchy's Problem in Linear Differential Equations*, ser. reprint. Dover, 1952.

[27] M. G. M. nos, A. Antoniadis, R. Cao, and W. Manteiga, "LASSO logistic regression, GSoft and the cyclic coordinate descent algorithm. application to gene expression data," *Statistical Applications in Genetics and Molecular Biology*, vol. 9, no. 1, pp. 1–28, 2010.

[28] L. Eberlin, R. Tibshirani, J. Zhang, T. Longacre, G. Berry, D. Bingham, J. Norton, R. Zare, and G. Poultsides, "Molecular assessment of surgical-resection margins of gastric cancer by mass-spectrometric imaging," *Proceedings of the National Academy of Sciences*, vol. 11, no. 7, pp. 2436–2441, 2014.

[29] J. Bai and S. Ng, "Forecasting economic time series using targeted predictors," *Journal of Econometrics*, vol. 146, no. 2, pp. 304–317, 2008.

[30] J. Fan, J. Lv, and L. Qi, "Sparse high-dimensional models in economics," *Annu. Rev. Econ.*, vol. 3, no. 1, pp. 291–317, 2011.

[31] S. Roy, D. Basu, and A. Abraham, "Stock market forecasting using LASSO linear regression model," *Afro-European Conference for Industrial Advancement*, pp. 371–381, 2015.

[32] A. Horel and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[33] A. Beck, *First - Order Methods in Optimization.* MOS-SIAM Series on Optimization, Mathematical Optimization Society, 2017, vol. 25.

[34] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

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

[36] L. Vandenberghe and S. Boyd, "Subgradients," EE364b, Stanford University, Lecture notes, Available at https://see.stanford.edu/materials/lsocoee364b/01-subgradients_notes.pdf.

[37] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $\mathcal{O}\left(\frac{1}{k^2}\right)$," *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[38] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[39] J. Wang, F. Yu, X. Chen, and L. Zhao, "ADMM for efficient deep learning with global convergence," arXiv:1905.13611, 2019.

[40] J. Bioucas-Dias and M. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," arXiv:1002.4572, 2010.

[41] M. Fadilli and J. Starck, "Monotone operator splitting for optimization problems in sparse recovery," *16th IEEE International Conference on Image Processing (ICIP)*, pp. 1461–1464, 2009.

[42] M. Figueiredo and J. Bioucas-Dias, "Restoration of poissonian images using alternating direction optimization," *IEEE Transactions on Image Processing*, vol. 19, pp. 3133–3145, 2010.

[43] Z. Lu, T. Pong, and Y. Zhang, "An alternating direction method for finding dantzig selectors," *Computational Statistics & Data Analysis*, vol. 56, no. 12, pp. 4037–4046, 2012.

[44] P. Forero, A. Cano, and G. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, pp. 1663–1707, 2010.

[45] P. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[46] L. Chen, D. Sun, and K.-C. Toh, "A note on the convergence of ADMM for linearly constrained convex optimization problems," *Computational Optimization and Applications*, vol. 66, no. 2, pp. 327–343, 2016.

[47] Y. Chen, "Alternating direction method of multipliers," Large-Scale Optimization for Data Science, Lecture notes, Available at http://www.princeton.edu/~yc5/ele522_optimization/lectures/ADMM.pdf.

[48] D. Donoho and Y. Tsaig, "Fast solution of ℓ_1-norm minimization problems when the solution may be sparse," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, 2008.

[49] S. P. Oliveira and D. E. Stewart, "Efficient basis pursuit DeNoising via active sets and homotopy," *SIAM J. Numer. Anal.*, vol. 12, no. 4, pp. 617–629, 1975.

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

[51] M. Khajehnejad, W. Xu, A. Avestimehr, and B. Hassibi, "Improved sparse recovery thresholds with two-step reweighted ℓ_1-minimization," *IEEE International Symposium on Information Theory Proceedings(ISIT)*, pp. 1603–1607, 2010.

[52] A. Charles and C. Rozell, "Dynamic filtering of time-varying sparse signals via ℓ_1-minimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 21, pp. 5644–5656, 2016.

[53] M. P. Friedlander, H. Mansour, R. Saab, and O. Yilmaz, "Recovering compressively sampled signals using partial support information," *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1122–1134, 2010.

[54] Y.-B. Zhao, *Sparse Optimization Theory and Methods*. CRC press, Taylor & Francis Group, Florida, 2018.

[55] J. Romberg and M. Asif, "Fast and accurate algorithms for re- weighted ℓ_1-norm minimization," *IEEE Transactions on Signal Processing*, vol. 61, no. 23, pp. 5905–5916, 2013.

[56] Y.-B. Zhao and D. Li, "Reweighted ℓ_1-minimization for sparse solutions to underdetermined linear system," *SIAM journal on Optimization*, vol. 22, no. 3, pp. 1065–1088, 2012.

[57] E. Svanborg and C. Guilleminault, "EEG frequency changes during sleep apneas," *American Sleep Disorders Association and Sleep Research Society*, vol. 19, no. 3, pp. 248–254, 1996.

[58] J. Perrier, P. Clochon, F. Bertran, J. Bullaand, P. Denise, M. Bocca, and C. Couque, “Specific EEG sleep pattern in the prefrontal cortex in primary insomnia,” *Plos One*, vol. 10, no. 1, p. e0116864, 2015.

[59] J. Xiang, Y. Wang, Y. Chen, Y. Liu, R. Kotecha, X. Huo, D. Rose, H. Fujiwara, N. Hemasilpin, K. Lee, F. Mangano, B. Jones, and DeGrauw, “Noninvasive localization of epileptogenic zones with ictal high-frequency neuromagnetic signals,” *Journal of Neurosurgery:Pediatrics*, vol. 5, no. 1, pp. 113–122, 2010.

[60] A. Sohrabpour, Y. Lu, P. Kankirawatana, J. Blount, H. Kim, and B. He, “Effect of EEG electrode number on epileptic source localization in pediatric patients,” *Clin Neurophysiol*, vol. 126, no. 3, pp. 472–480, 2015.

[61] M. Ullsperger and S. Debener, *Simultaneous EEG and FMRI*. Oxford University Press, 2010.

[62] M. Hämäläinen and R. Ilmoniemi, “Interpreting magnetic fields of the brain: Minimum norm estimates,” *Medical biological engineering and computing*, vol. 32, no. 1, pp. 35–42, 1994.

[63] A. Dale, A. Liu, B. Fischl, R. Buckner, J. Belliveau, J. Lewine, and E. Halgren, “Dynamical statistical parametric mapping: Combining fMRI and MEG for high-resolution imaging of cortical activity,” *Neuron*, vol. 26, no. 1, pp. 55–67, 2000.

[64] R. D. Pascual-Marqui, “Standaridzed low resolution brain electromagnetic tomography (sLORETA): technical details,” *Methods* & *findings in Experimental* & *clinical Pharmacology*, vol. 24, no. Suppl D, pp. 5–12, 2002.

[65] C. Song, Q. Wu, and T. Zhuang, “Hybrid weighted minimum norm method, a new method based on LORETA to solve EEG inverse problem,” *Proceedings of IEEE, Engineering in Medicine and Biology, 27th annual conference*, pp. 1079–1082, 2005.

[66] K. Uutela, M. Hämäläinen, and E. Somersalo, "Visualization of magnetoencephalographic data using minimum current estimates," *Neuro Image*, vol. 10, no. 2, pp. 173–180, 1999.

[67] M. Kowalski, A. Gramfort, D. Strohmeier, J. Haueisen, and M. Hämäläinen, "Time-frequency mixed-norm estimates: Sparse M/EEG imaging with non-stationary source activations," *NeuroImage*, vol. 70, pp. 410–422, 2013.

[68] F. Liu, J. Rosenberger, Y. Lou, R. Hosseini, J. Su, and S. Wang, "Graph regularized EEG source imaging with in-class consistency and out-class discrimination," *IEEE Transactions on Big Data*, vol. 3, no. 4, pp. 378–391, 2017.

[69] S. Wang, F. Liu, J. Rosenberger, Y. Lou, and J. Quin, "Task-related EEG source localization via graph regularized low-rank representation model," bioRxiv preprint:10.1101/246579, 2018.

[70] F. Liu, L. Wang, Y. Lou, R.-C. Li, and P. Purdon, "Probabilistic structure learning for EEG/MEG source imaging with hierarchical graph prior," arXiv preprint, arXiv:1906.02252, 2019.

[71] C. Wolters, L. Grasedyck, A. Amwander, and W. Hackbusch, "Efficient computation of lead field bases and influence matrix for the FEM-based EEG and MEG inverse problem," *Inverse Problems*, vol. 20, no. 4, pp. 1099–1116, 2004.

[72] M. Hämäläinen and J. Sarvas, "Feasibility of the homogeneous head model in the interpretation of neuromagnetic fields," *Phys. Med. Biol.*, vol. 32, no. 1, pp. 91–97, 1987.

[73] F. Tadel, S. Baillet, J. Mosher, D. Pantazis, and R. Leahy, "Brainstorm: A user-friendly application for MEG/EEG analysis," *Computational Intelligence and Neuroscience*, vol. 2011, no. 8, 2011.

[74] R. Bartels and G. Stewart, "Solution of the matrix equation $AX + XB = C$," *Communications of the ACM*, vol. 15, no. 9, pp. 820–826, 1972.

[75] G. Golub, S. Nash, and C. V. Loan, "A Hessenberg-Schur method for the problem $AX + XB = C$," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 909–913, 1979.

[76] Q. Wei, N. Dobigeon, and J. Tourneret, "Fast fusion of multi-band images based on solving a Sylvester equation," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 4109–4121, 2015.

[77] R. Horn and C. Johnson, *Matrix Analysis*, 2nd ed. New York: Cambridge University Press, 2013.

[78] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[79] C. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[80] S. Guiasu and A. Shenitzer, "The principle of maximum entropy," *The Mathematical Intelligence*, vol. 7, no. 1, pp. 42–48, 1985.

[81] Y. Xie, "ECE587: Information theory," Lecture Notes on Information Theory, Available online at https://www2.isye.gatech.edu/~yxie77/ece587.html, 2013.

[82] T. Cover and J. Thomas, *Elements of Information Theory*, 2nd ed. A John Wiley & Sons, Inc., Publication, 2006.

[83] M. Stoeckel, R. Seitz, C. Buetefisch, and M. Mishkin, "Congenitally altered motor experience alters somatotopic organization of human primary motor cortex," *Proceedings of the National Academy of Sciences of the USA*, vol. 106, no. 7, pp. 2395–2400, 2009.

[84] J. Knierim, "Neuroscience online, an electronic textbook for the neuroscience," The University of Texas Health Science Center at Houston. Available online at https://nba.uth.tmc.edu/neuroscience/m/s3/chapter03.html.

[85] M. Shun, "Projected gradient on L-Lipschitz convex function, convergence rate $\mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$," Available online at https://angms.science/doc/CVX/CVX_PGD.pdf.

[86] Y. Chen and X. Ye, "Projection onto a simplex," arXiv preprint arXiv:1101.6081v2, 2011.

[87] W. Wang and M. A. C.-P. nán, "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application," arXiv preprint, arXiv:1309.1541v1, 2013.

[88] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," *Proceedings of the 25th International Conference on Machine Learning-ICML*, pp. 272–279, 2008.

[89] Y. Hu, Z. Wei, and G. Yuan, "Accelerated proximal gradient algorithms for matrix $\ell_{2,1}$-norm minimization problem in multi-task feature learning," *Stat., Optim., Inf. Compute*, vol. 2, pp. 352–367, 2014.

[90] Y. Xiao, S.-Y. Wu, and B.-S. He, "A proximal alternating direction method for $\ell_{2,1}-$norm least squares problem in multi-task feature learning," *Journal of Industrial and Management Optimization*, vol. 8, no. 4, pp. 1057–1069, 2012.

[91] S. Lloyd, "Least squares quanization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[92] S. Raschka and V. Mirjalili, *Python Machine Learning*, 3rd ed. Packt Publishing Ltd., 2019.

[93] L. Drumetz, T. Meyer, J. Chanussot, A. Bertozzi, and C. Jutten, "Hyperspectral image unmixing with endmember bundles and group sparsity inducing mixed norms," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3435–3450, 2019.

[94] A. Bertozzi, L. Drumetz, T. Meyer, J. Chanussot, and C. Jutten, "Hyperspectral unmixing with material variablity using social sparsity," *IEEE International Conference on Image Processing (ICIP)*, pp. 2187–2191, 2016.

BIOGRAPHICAL STATEMENT

Kiran Kumar Mainali was born in Makawanpur, Nepal, in 1987. He finished his high school from Shree Jana Jagriti Secondary School, Bara. He moved to Kathmandu, the capital city of Nepal for his further studies. He finished his intermediate degree from Pashupati Multiple Campus (PMC), Chabahil in 2006. He continued his Bachelor of Arts degree in PMC with a major in mathematics and minor in sociology, and graduated in 2009. He graduated with his Master of Arts (MA) degree from Central Department of Mathematics, Tribhuvan University in 2012. He came to the University of Texas at Arlington in the fall of 2016 for his Ph.D. degree.

Kiran's research interest includes numerical linear algebra, numerical analysis, mathematical optimization, scientific computing, machine learning, and artificial intelligence. He wants to extend his research skills to solve the challenges in the field of data science.