SPARSITY IN IMAGE PROCESSING AND MACHINE LEARNING: MODELING, COMPUTATION AND THEORY

by

YUE ZHANG

Submitted in partial fulfillment of the requirements
For the degree of Doctor of Philosophy

Dissertation Advisor: Dr. Weihong Guo

Department of Mathematics, Applied Mathematics and Statistics
CASE WESTERN RESERVE UNIVERSITY

May 2018
CASE WESTERN RESERVE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

We hereby approve the dissertation of

Yue Zhang

candidate for the Doctor of Philosophy degree *

Committee Chair: Dr. Weihong Guo
Dissertation Advisor
Associate Professor,
Department of Mathematics, Applied Mathematics and Statistics

Committee: Dr. Steven Izen
Professor,
Department of Mathematics, Applied Mathematics and Statistics

Committee: Dr. Erkki Somersalo
Professor,
Department of Mathematics, Applied Mathematics and Statistics

Committee: Dr. Soumya Ray
Associate Professor,
Department of Electrical Engineering and Computer Science

April 2, 2018

*We also certify that written approval has been obtained for any proprietary material contained therein.
Table of Contents

Table of Contents .................................................. iii
List of Tables ......................................................... v
List of Figures ......................................................... vi
Acknowledgement ..................................................... ix
Abstract ............................................................... x

I Introduction and Background ...................................... 1

1 Introduction ......................................................... 2
   1.1 The World of Sparsity ......................................... 2
   1.2 Mathematics of Sparse Recovery ............................. 4
   1.3 Contributions and Organization ............................. 11

2 Literature Review .................................................. 14
   2.1 Image Reconstruction ......................................... 15
   2.2 Algorithms for Dictionary Learning ......................... 18
   2.3 Feature Selection and the Lasso ............................. 20

II Sparse Modeling in Image Reconstruction ......................... 24

3 Projection Correction Modeling .................................. 25
   3.1 Model Overview ............................................... 26
   3.2 Problem Set-up ............................................... 28
   3.3 Details the Proposed PCM Framework ....................... 29

4 Domain Adaptive Regularization ................................ 33
   4.1 A Short Primer on Fractional Derivatives .................. 33
   4.2 Proposed TV-TFV Regularity ................................. 37
List of Tables

4.1 Numerical results for denoising with different regularity/penalty in Figure 4.3. .................................................. 45

5.1 Numerical comparison of the different models on different noise levels. . . 58
5.2 Numerical details for the 2D reconstruction in Figure 5.4 with bias error, $\sigma = 0.6$. .................................................. 59
5.3 Numerical results for denoising with different regularity/penalty in Figure 5.5. .................................................. 61

7.1 Numerical comparison of the relative reconstruction errors between DIDL and consensus algorithm with different sparsity ratios ($T/n$) and noise levels (standard deviation). The reconstruction error for the centralized K-SVD is also shown as a reference, denoted by $R_{cent}$. ................. 89
List of Figures

1.1 Denoising results for models with different regularities. The Tikhonov regularized model smooths the whole image and causes blurring effect. The TV model on the other hand preserves the edges and produces an image with better quality but oil-painting artifacts. .................................................. 7
1.2 Solutions from the Lasso model and the ridge regression model. The red dots are entries of the ground truth $w$. The blue dots are entries of the solutions from the Lasso model and the ridge regression model in (a) and (b) respectively. The ground truth $w$ is sparse. The Lasso model is able to recover a sparse solution while the ridge regression model leads to a solution that is nonzero almost everywhere. ................................. 12

3.1 Illustration of the general PCM framework. The given noisy measurements are first projected onto a space spanned by pre-selected basis $\{\phi_i\}_{i=1}^n$. The errors are then corrected by solving a minimization problem with regularity/penalty. ................................................................. 29
3.2 Geometric interpretation of optimization problem (3.3). The optimal $c_g$ is found by projecting the $c_f$ onto the closed convex set $Q$. ................................. 32

4.1 Illustration of image dilation. ................................................................. 38
4.2 Illustration of the TV-TFV regularization scheme in the image denoising problem. .............................................................................................................. 40
4.3 Denoising results for models with different regularity/penalty. Here $\alpha$ denotes the fractional order of derivative in the TFV model. ................................. 43
4.4 Detailed comparison between different models. ................................. 44

5.1 Illustration of jittered and uniform sampling schemes. .......................... 53
5.2 Bias-free reconstruction for piecewise a constant signal without noise: (a) Ground truth signal and its projection on $\mathcal{H}_n$. (b) Reconstruction from truncated Fourier series which suffers from Gibbs oscillation. (c) Reconstruction from projection stage. Here $m = 128$, $n = 32$, $\text{resol} = 1/256$, $\theta = 0.25$. (d) The difference between the reconstruction and the ground truth. 55

vii
5.3 (a) Piecewise linear function (red) and its projection onto \( H_n, n = 128 \) (blue). (b) The difference between the two. .......................... 56

5.4 Reconstruction comparison with bias error for the 2D image. Here \( m = 128 \times 128, n = 64 \times 64, imsize = 256 \times 256, \sigma = 0.6 \). ............... 59

5.5 Reconstruction comparison between PCM-TV, SS-TV-TFV and PCM-TV-TFV. Noisy image (b) is obtained from inverse Fourier transform. Here \( m = 128 \times 128, n = 96 \times 96, imsize = 256 \times 256, \sigma = 0.4 \). .............. 62

5.6 Zoomed-in comparison of the red square in Figure 5.5. Figure (a) is the true surface. Figure (b) shows the difference between the ground truth and the one reconstructed by PCM-TV. Relative error \( \approx 62\% \). Figure (c) shows the difference between the ground truth and the reconstruction by SS-TV-TFV. Relative error \( \approx 38\% \). Figure (d) shows the difference between the ground truth and the one reconstructed by PCM-TV-TFV. Relative error \( \approx 32\% \). .......................... 63

7.1 Average reconstruction error vs. communication times \( c \) for DIDL (red triangle) and Consensus algorithm update(blue star) at different settings. .......................... 88

7.2 Illustration of the patch extraction for a \( 256 \times 256 \) image. The blue window shows the first extracted \( 8 \times 8 \) patch of the image. Then the window slides to right with step size 1 (pixel) and extract another patch. After extracting all available patches in the horizontal direction, the window moves downwards for 1 (pixel) and repeat the extraction. The total number of patches will be \((256 - 7)^2 = 62,001\). .......................... 91

7.3 Visualization of image denoising results using centralized K-SVD, consensus dictionary learning algorithm (CDL) and the proposed DIDL. .......................... 93

7.4 Visualization of image denoising results using centralized K-SVD, consensus dictionary learning algorithm (CDL) and the proposed DIDL. .......................... 94

7.5 Visualization of image denoising results using centralized K-SVD, consensus dictionary learning algorithm (CDL) and the proposed DIDL. .......................... 95

9.1 Illustration of scaling in the recovered model. Left: \( g_1(u) \) with \( m = 3 \), Right: \( g_3(u) \). .......................... 122

9.3 Numerical tests on the relation between power constant \( k \) and the sample size \( n \) with fixed amount of noise \( \epsilon \). Left: \( g_1(u) \) with \( m = 3, \epsilon \sim \mathcal{N}(0, 2I) \). Right: \( g_1(u) \) with \( m = 5, \epsilon \sim \mathcal{N}(0, I) \). .......................... 123

9.4 Numerical simulations on the probability of successful group selection with different sample size \( n \) and \( \Sigma \) for polynomial families. The first row are simulations on \( g_1(w), m = 1 \) corresponds to the linear mapping function. The first column has \( \Sigma = I \), second column has \( \Sigma = \Sigma(0.2) \). .......................... 124
9.2 Numerical simulations on the probability of successful selection with different sample size $n$ and $\Sigma$. The first row are simulations on $g_1(w)$, $m = 1$ corresponds to the linear mapping function, second row corresponds to $g_2(w)$. The first column has $\Sigma = I$, second column has $\Sigma = \Sigma(0.2)$.

9.5 Numerical tests on the necessity of assumptions. $U[-1,1]$ is uniform distribution on $[-1,1]$, $t(n\nu = 1)$ is t-distribution with freedom $\nu = 1$, $Beta(2, 2)$ corresponds to Beta distribution with $\alpha = \beta = 2$, $Weibull(1, 1.5)$ is Weibull distribution with scale 1, shape 1.5, $N(0, 1)$ is standard normal for comparison. The “False Eg” line corresponds to tests on $g_4$ with data from Gaussian distribution $\Sigma = I$.

A.1 .......................... .......................... 136
Acknowledgement

There are so many people lighting the path of my life during the five years of my PhD career, that I want to thank. First of all, I would like to express my sincere gratitude to my advisor Dr. Weihong Guo for sharing her knowledge, passion, and philosophy of life. She has given me consistent support and encouragement along the way. She has taught me so much, more than just knowledge found in papers and textbooks. Three years ago, when I passed my qualifying exams, I knew nothing about doing research. She started to share her vision, guided me to read papers and gave me many helpful suggestions. I am really grateful for her supervision.

I would also like to thank every professor that has shared their vision and showed the beauty of knowledge to me: Dr. Alethea Barbaro, Dr. Jenný Brynjarsdóttir, Dr. Daniela Calvetti, Dr. Harold Connamacher, Dr. Julia Dobrosotskaya, Dr. Steven Izen, Dr. Michael Lewicki, Dr. Vincenzo Liberatore, Dr. Soumya Ray, Dr. Erkki Somersalo, Dr. Stanislaw Szarek, Dr. Wojbor Woyczynski, and Dr. Longhua Zhao.

I want to give special thanks to my committee members Dr. Steven Izen, Dr. Soumya Ray and Dr. Erkki Somersalo. I enjoy all discussions with them and appreciate their insightful suggestions.

During the last year, I had a wonderful internship at Siemens Healthcare Research. I want to thank my mentors Rui Liao and Shun Miao for their supervision and guidance. They have given me enormous help to build my confidence for both research and future life.
I would also thank my fellow graduate students for their friendship, support, and encouragement. Richard Lartey and Thomas Atta-Fosu made this journey a lot of fun. We sat through a lot of group discussions along with Dr. Guo.

I want to thank the department and National Science Foundation for their support and sponsorship. The devoted faculty and staff in the department have been really helpful during my graduate study. Finally, I would like to dedicate this thesis to my wife Qian and my parents. For Qian, I feel fortunate to have such a wonderful lady in my life. She has become my motivation for a better future. For my parents, I could not have accomplished this without their love and support.
Sparsity in Image Processing and Machine Learning: Modeling, Computation and Theory

Abstract

by

YUE ZHANG

Natural images are usually either sparse by themselves or sparse after a transform. For instance, images of stars from telescope are sparse by themselves and images of humans are sparse under the wavelet transform. Sparsity plays an important role in image processing and machine learning. How to build appropriate sparse models, how to numerically find solutions from the sparse models and how to derive theoretical guarantees of the correctness of the solutions are essential for the success of sparsity in applications. In this thesis, we study sparse modeling in image processing and machine learning. We propose a novel two-stage (projection followed by correction) modeling framework for image reconstruction from nonuniform Fourier measurements. We design domain adaptive sparsity regularization schemes that are able to maintain image smoothness as well as edge details. Various numerical results in both one-dimensional signals and two-dimensional images are also presented to demonstrate the superior performance of the proposed model compared with other methods.

We study the problem of learning data adaptive overcomplete bases so that each data example can be linearly represented by few basis vectors. In the specific case when the
input data size is large, we design a novel distributed dictionary learning algorithm which builds on the classical consensus approach. Most importantly, the proposed algorithm imposes a structured constraint to ensure that the derived dictionary has low coherence, that is, the columns of the dictionaries are nearly orthogonal.

We analyze the solution obtained from the well known Least Absolute Shrinkage and Selection Operator (Lasso) model in the feature selection scenario. A substantial challenge in feature selection to specify the relation between the factors/features and the observations (e.g., linear or nonlinear). The relation is however often unknown. We prove that under certain conditions, Lasso is able to find the correct features (asymptotically) even though the model is misspecified.

The contributions of this thesis are summarized below:

1. We propose a new two-stage model with domain adaptive sparsity constraints for image reconstruction from nonuniform Fourier measurements.

2. We design a new distributed dictionary learning algorithm.

3. We prove a new theorem on the correctness of Lasso model in sparse feature selection when the relationship between observations and features is misspecified.
Part I

Introduction and Background
Chapter 1

Introduction

“The phenomenon of ubiquitous compressibility raises very natural questions: why go to so much effort to acquire all the data when most of what we get will be thrown away? Can we not just directly measure the part that will not end up being thrown away?”

– David L. Donoho

1.1 The World of Sparsity

The popularity of sparse modeling dates back to the 1990s, when researchers from different fields independently developed models with sparsity constraint and reported state-of-the-art performance. In image processing, Rudin, Osher and Fatemi [1] proposed an image noise removal model with total variation (TV) regularization. The underlying assumption is that the edges only constitute a small portion (and thus are sparse) in the domain of natural images. The TV model can effectively clean images and has achieved a great success.
It has since been applied to several image processing problems such as deblurring (e.g. [2, 3, 4]), reconstruction (e.g. [5, 6, 7]), segmentation (e.g. [8, 9]), inpainting (e.g. [10, 11]), and super-resolution (e.g. [12, 13]).

In statistics, the Lasso model (least absolute shrinkage and selection operator) for variable selection in linear models was proposed by Tibshirani in [14]. Compared with the classical ridge regression (a version of Tikhonov regularization), the Lasso model utilizes $l_1$-norm regularization over the weights of the regressors and can successfully shrink the weights of the irrelevant variables to zero. It thus gives a sparse solution. This idea has since been made popular with generalizations and applications to elastic net [15], adaptive Lasso [16], and group Lasso [17].

In signal processing, a model based on a similar idea named basis pursuit was pioneered by Mallat and Zhang in [18] and Chen and Donoho in [19, 20]. Instead of the traditional orthonormal basis, the basis pursuit model aims to decompose given signals by an overcomplete basis called a dictionary under which these signals can be sparsely represented.

In a different context, sparsity promoting methods have been widely studied in inverse problems and Bayesian statistics for a long time, see [21] for a recent review of regularization from a Bayesian perspective.

Inspired by the success of these sparsity promoting methods, Candès, Romberg and Tao [5] generalized the concept into undercomplete linear systems and showed exact signal reconstruction from highly incomplete Fourier data with a suitable sampling scheme. This can significantly reduce the sampling rate required by the well known Shannon-Nyquist theorem, and consequently gave rise to the subject of compressed sensing. The theoretical study of the basis pursuit model as well as the development of efficient algorithms have attracted substantial research interest ever since. The success of compressed sensing has been demonstrated in medical imaging [22, 23], remote sensing [24, 25], networked data
management [26, 27], bioinformatics [28, 29] and other fields.

In neuroscience, Olshausen and Field [30, 31] studied a basis learning model to understand the receptive fields of cells in the mammalian primary visual cortex. Compared with previous fixed off-the-shelf dictionaries, they proposed to learn data dependent bases from natural images. This work laid down the foundation of the dictionary learning problem. It has inspired the development of several models and algorithms in computer vision and natural language processing, including the extension of independent component analysis (ICA) to the overcomplete basis in a Bayesian perspective [32], development of the K-SVD algorithm for efficient dictionary updates [33, 34], robust speech recognition using learned exemplar-based sparse representation and nonnegative matrix factorization (NMF) [35], discriminative basis design for image recognition and classification [36, 37, 38, 39].

1.2 Mathematics of Sparse Recovery

Most of the previously mentioned problems start from a simple linear form,

\[ b = Ax + \epsilon \]  (1.1)

where \( b \in \mathbb{R}^m \) is the given data/observation vector, \( \epsilon \in \mathbb{R}^m \) is an unknown vector representing noise/perturbation, \( x \in \mathbb{R}^n \) is the signal/image to be estimated. The matrix \( A \in \mathbb{R}^{m \times n} \) depends on the application. For example, it is usually a known matrix for image reconstruction and feature selection. It needs to be estimated in the case of dictionary learning.

A classical approach to solve problem (1.1) is to find the best estimation of \( x \) by
minimizing an $l_2$ loss function together with a quadratic penalty called Tikhonov regularity,

$$\hat{x}_{TIK} = \arg\min_x \left\{ \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|Lx\|_2^2 \right\},$$  \hspace{1cm} (1.2)

where $L$ is typically an identity matrix or a matrix approximation of a differential operator, $\lambda$ is a regularization parameter providing tradeoffs between data fitting and regularity/penalty [40, 41]. However, the Tikhonov regularization suffers from several drawbacks and is usually suboptimal for many problems. One of the main drawbacks commonly known in image processing is that it can cause a blurry effect, that is, the estimated image $\hat{x}_{TIK}$ is usually smoothed and blurred.

To address these issues, sparse modeling replaces the $l_2$-norm penalty with $l_1$-norm regularization and seeks to find the optimal solution $\hat{x}_S$, defined as

$$\hat{x}_S = \arg\min_x \left\{ \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|Lx\|_1 \right\},$$  \hspace{1cm} (1.3)

where $\|\cdot\|_1$ denotes the standard $l_1$-norm which sums up the absolute values of all the components of a given vector. Under suitable conditions, model (1.3) returns a solution $\hat{x}_S$ with $L\hat{x}_S$ mostly zero. To better understand the effectiveness of $l_1$ regularity in applications, we briefly review $l_1$ based image denoising, dictionary learning and feature selection. These models will be further discussed in this thesis.

**Total variation model [1].** Let $\Omega$ be a bounded open domain in $\mathbb{R}^2$. We consider the problem of recovering a clean gray scale image from noisy data. In a continuous setting, the clean image can be described using a function $u : \Omega \to \mathbb{R}^+$. Similarly, we denote the noisy image intensity function as $\tilde{u}(x)$. Assume

$$\tilde{u}(x) = u(x) + \epsilon,$$
where $\epsilon$ has zero mean and constant variance, and assume $u$ is smooth with bounded variation. The Rudin-Osher-Fatemi (ROF) \cite{1} model can be formulated as

$$u_{\text{ROF}} = \arg\min_u \left\{ \frac{1}{2} \int_{\Omega} (u(x) - \bar{u}(x))^2 dx + \lambda \int_{\Omega} |\nabla u(x)| dx \right\}. \quad (1.4)$$

The term $\int_{\Omega} |\nabla u(x)| dx$ is the total variation of $u$. In discrete forms where we focus on pixel values $u(x_{ij}), i, j = 1, 2, \cdots, n$ over the natural Cartesian grid on $[0, 1]^2$, the ROF model becomes

$$U_{\text{ROF}} = \arg\min_U \left\{ \frac{1}{2} \| U - \bar{U} \|_2^2 + \lambda (\| D_1 U \|_1 + \| D_2 U \|_1) \right\}, \quad (1.5)$$

where $U$ denotes the vectorized image by arranging pixel values $u(x_{ij})$ in a lexicographical order, and $D_1 U$ and $D_2 U$ correspond to the vectorized discrete derivatives of $u(x)$ along the $x$ and $y$ directions, respectively.

Figure 1.1 shows the superior performance of the ROF model over the classical Tikhonov regularization models for the denoising problem. It can effectively remove the noise and preserve the edges. However, it can be seen that the denoised image has oil-painting artifacts. This is due to the fact that the TV regularization leads to piecewise constant solutions. A natural solution is to use spatially adaptive regularization to maintain both the smooth part and the sharp edges, as is proposed in Part II of this thesis.

**Dictionary Learning \cite{31}**. Given training signals $Y = [y_1, y_2, \ldots, y_N], y_i \in \mathbb{R}^n$, we seek to construct a dictionary $D = [d_1, d_2, \ldots, d_K] \in \mathbb{R}^{n \times K}$ such that each member $y_i$ has a sparse representation. In other words, each of the given signals can be expressed as a linear combination of only a few vectors in the dictionary. Each vector $d_i$ is called an atom.
Figure 1.1: Denoising results for models with different regularities. The Tikhonov regularized model smooths the whole image and causes blurring effect. The TV model on the other hand preserves the edges and produces an image with better quality but oil-painting artifacts.
Mathematically, the data generation model can be written as,

\[ y_i = D x_i, \quad \text{with } x_i \in \mathbb{R}^K \text{ being sparse, } \forall i = 1, 2, ..., N. \]

The coefficient vectors \( x_i \)'s are called sparse codes. The dictionary learning problem aims to compute these basis vectors as well as the sparse codes from the data \( y_i \)'s. Denote \( \| \cdot \|_0 \) as the number of nonzeros in a vector. In the dictionary learning literature, finding the dictionary \( D \) as well as the associated sparse codes usually leads to the following optimization problem,

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{N} \| y_i - D x_i \|_2^2 + \lambda \mathcal{R}(D) \\
\text{subject to} & \quad \| x_i \|_0 \leq T_0, \quad \forall i = 1, 2, ..., N, \\
& \quad d_j \in \Gamma, \quad \forall j = 1, 2, ..., K.
\end{align*}
\] (1.6)

Here, \( T_0 \) is a constant determined by the prior knowledge of the sparsity of \( x \), \( \Gamma \) is a set of vectors satisfying certain properties, e.g. \( \Gamma = \{ d \mid \| d \|_2 = 1 \} \), \( \mathcal{R}(D) \) denotes some structural constraint on \( D \), e.g. \( \| D^T D - I \|_F^2 \) which constrains the coherence of the dictionary vectors.

\( K \), the number of atoms, is usually larger than \( n \), the dimension of each basis vector. The dictionary \( D \) is therefore overcomplete (more columns than rows). It has been shown that overcompleteness of the dictionary can lead to sparser representations of the signal and is able to promote greater robustness in the presence of noise. However, since both the dictionary \( D \) and the coefficients \( x_i \)'s are unknown, problem (1.6) is ill-posed and non-convex. We summarize a popular alternating minimization algorithm that alternates between the optimization of \( D \) and \( x_i \)'s in Algorithm 1. Unfortunately, the convergence of
such scheme is not guaranteed in general.

In practice, the given dataset $Y = [y_1, y_2, ..., y_N]$ can be very large so that dictionary learning in a single machine is difficult. For instance in image denoising problems, $y_i$’s are usually overlapping patches extracted from images. For a single $256 \times 256$, the standard set-up extracts all $8 \times 8$ patches with 1-pixel overlap. The total number of patches can easily exceed 60K. Processing the multi-image denoising task is then very time consuming. To tackle such challenge, we propose a distributed algorithm that learns a dictionary from data stored in different machines. This will be discussed in Part III.

Algorithm 1 Generic dictionary learning algorithm

1: **Input:** dataset $\{y_i\}_{i=1}^N$, sparsity level $T_0$, number of atoms $K$, regularization parameter $\lambda$.

2: **Initialize** $D$ randomly or from the data, with $d_i \in \Gamma$, for $\forall i = 1, 2, ..., K$.

3: **Sparse coding stage.** Fix $D$, update $x_i$’s for $i = 1, 2, ..., N$. This can be done by solving sparsity recovery problems for each $x_i$, that is

$$\begin{align*}
\text{minimize}_{x_i} & \quad \sum_{i=1}^N \|y_i - Dx_i\|_2^2 \\
\text{subject to} & \quad \|x_i\|_0 \leq T_0, \quad \forall i = 1, 2, ..., N.
\end{align*}$$

The can be solved using orthogonal matching pursuit OMP [42] algorithm.

4: **Dictionary update stage.** Fix $x_i$’s, update $D$.

$$\begin{align*}
\text{minimize}_{D} & \quad \sum_{i=1}^N \|y_i - Dx_i\|_2^2 + \lambda \mathcal{R}(D) \\
\text{subject to} & \quad d_j \in \Gamma, \quad \forall j = 1, 2, ..., K.
\end{align*}$$

This can be done using Method of Optimal Directions (MOD) [43, 44], or the K-SVD algorithm [33] etc.

5: **Repeat** step 2 and 3 until converge.

6: **Output:** $D$ and $x_i$’s, $i = 1, 2, ..., N$.

The Lasso Model [14]. Assume we are given $n$ independent observations $(x_i, y_i)$,
\[ y_i = x_i^T w + \epsilon_i, \quad i = 1, 2, \ldots, n, \]

where \( \epsilon_i \)'s are independent and identically distributed Gaussian random variables with constant variance, \( \epsilon_i \sim \mathcal{N}(0, \sigma^2) \), \( x_i' \in \mathbb{R}^p \) are independent and identically distributed feature vectors, \( w \in \mathbb{R}^p \) is the weight vector we want to recover. The problem of finding the support (location of nonzero entries) of \( w \) is known as feature selection, since the nonzero parts of \( w \) correspond to features making contribution in generating \( y \).

Let \( y = (y_1, y_2, \ldots, y_n)^T \), \( \epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^T \) and \( X \) be the \( n \times p \) data matrix whose \( i \)th row is \( x_i^T \). Finding the optimal \( w \) corresponds to the following feature selection model:

\[
\hat{w} = \arg\min_w \left\{ \|y - Xw\|^2_2 + \lambda \mathcal{R}(w) \right\},
\]

\( \mathcal{R}: \mathbb{R}^p \rightarrow \mathbb{R} \) is a convex regularization function, which is \( \|\cdot\|_1 \) for classical Lasso and \( \|\cdot\|_{1,2} \) for group Lasso [17]. Here, suppose the index set \( \{1, 2, \ldots, p\} \) is parsed into \( m \) non-overlapping subsets \( I_1, I_2, \ldots, I_m \) such that \( \bigcup_{j=1}^m I_j = \{1, 2, \ldots, p\} \) and \( I_j \cap I_k = \emptyset \). For a given vector \( w = (w_1, w_2, \ldots, w_p) \in \mathbb{R}^p \), the \( l_{1,2} \)-norm corresponding to such group division is defined as \( \|w\|_{1,2} := \sum_{j=1}^m \|w_{I_j}\|_2 \).

There are different popular techniques used in feature selection, such as subset selection and ridge regression (a version of Tikhonov regularization). Classical subset selection aims to select regressors that minimizes the Akaike information criterion (AIC) or the Bayesian information criterion (BIC) in a forward-backward greedy manner. It is called greedy because such method selects one regressor at a time that best minimizes the AIC or BIC at that step. In other words, it finds the \textit{locally} optimal solution at each selection step. The final obtained regressors are not necessarily globally optimal. The solution obtained by the
subset selection method can be extremely variable because of the greedy selection process. The Tikhonov regularization model has been well established in signal processing. However, for feature selection it suffers a major drawback in that most of the entries of \( \hat{w} \) are nonzero, especially in the presence of noise. It therefore cannot lead to sparse solutions.

Figure 1.2 shows an example of the solutions from the Lasso model and ridge regression model using synthetic data. The feature matrix \( X \in \mathbb{R}^{50 \times 200} \) is generated from a Gaussian distribution with zero mean and identity covariance. The standard deviation of the noise is set to 0.1. The regularization parameter \( \lambda \) is set to 1 for both models. As we can see, the Lasso model is able to recover a sparse solution. The ridge regression model leads to a solution that is nonzero almost everywhere. Indeed the Lasso model recovers the exact solution in this experiment. In Part IV, we further study the theoretical guarantees of the correctness of the Lasso model. We show that under suitable assumptions, the Lasso is able to recover the correct solution even when the model is misspecified.

### 1.3 Contributions and Organization

#### 1.3.1 Summary of contributions

The novelties and contributions of the work presented in this thesis are summarized as follows:

1. In Part II, we propose a novel two-stage projection correction model for image reconstruction from noisy linear Fourier measurements. In the correction stage, we introduce a new regularity that uses total variation (TV) and total fractional variation (TFV) adaptively on different image domains.
Figure 1.2: Solutions from the Lasso model and the ridge regression model. The red dots are entries of the ground truth $w$. The blue dots are entries of the solutions from the Lasso model and the ridge regression model in (a) and (b) respectively. The ground truth $w$ is sparse. The Lasso model is able to recover a sparse solution while the ridge regression model leads to a solution that is nonzero almost everywhere.

2. In Part III, we design a new distributed dictionary learning algorithm with generalized consensus updates.

3. In Part IV, we prove a new theorem on the correctness of the Lasso model when the data generation model is misspecified.

1.3.2 Organization of the thesis

In Part I, we present an overview of the related works on the sparse modeling in image reconstruction, dictionary learning algorithms and theoretical analysis of the Lasso. In Part II, we present the two-stage projection correction model for image reconstruction. In Part III, we show the proposed distributed dictionary learning algorithm. In Part IV, we analyze the Lasso model in the context of feature selection and model misspecification. We conclude and discuss the future directions in the final chapter.
Related works. Parts of the results in this thesis are based on previously published works with several collaborators. The projection correction modeling work for image reconstruction in Part II is based on the joint work with Weihong Guo and Guohui Song [45]. The theoretical analysis of the Lasso in Part IV is based on the joint work with Soumya Ray and Weihong Guo [46].
Chapter 2

Literature Review

“The $l_1$ methods assume that the truth is sparse, in some basis. If the assumption holds true, then the parameters can be efficiently estimated using $l_1$ penalties. If the assumption does not hold - so that the truth is dense - then no method will be able to recover the underlying model without a large amount of data per parameter.”

– Rob Tibshirani

In this chapter, we provide a concise review of the work related to the proposed model, algorithm and theoretical results in this thesis. The goal of this chapter is to give a brief introduction to the related classical works and further elaborate the novelty and contributions of the work in this thesis. In Section 2.1, we review the image reconstruction from Fourier measurements which is a fundamental problem in several applications. We summarize some popular discrete and continuous methods in the literature and provide the motivation of the proposed projection correction model. Section 2.2 provides an overview of the dictionary learning models, algorithms and their distributed variants. Finally, in Section 2.3
we review the theoretical development of the $l_1$-norm constraint models, especially the Lasso.

### 2.1 Image Reconstruction

Image reconstruction aims to recover the underlying 2D or 3D images from their measurements generated by a given imaging modality. For instance, standard computed tomography (CT) scan gathers the attenuation measurements when X-ray beam traverses a body and CT reconstruction uses the attenuation measurements to reconstruct cross-sectional (tomographic) images [47]. In this thesis, we study the case of image reconstruction from Fourier measurements, which is a fundamental problem in various applications such as magnetic resonance imaging (MRI) [22, 23, 48], and synthetic-aperture radar imaging [49, 50]. Given the sampled Fourier measurements $\hat{f}$ of an image, image reconstruction has usually been formulated as an optimization problem that minimizes an energy functional expressed in the following form,

$$\min_{f} \mathcal{L}(\hat{f} - SF f) + R(f),$$

where $F$ is the Fourier transform, $S$ denotes the sampling operator, $\mathcal{L}(\cdot)$ is a data fidelity term depending on an empirical estimation of the distribution of noise, and $R(\cdot)$ is a regularity term. For instance, $\mathcal{L}(\cdot)$ is often an $l_2$ loss when the noise is assumed to be Gaussian, and other formulations could also be found in [51]. A widely used regularity term $R(\cdot)$ is the $l_1$ type constraint incorporating certain sparsity prior knowledge. Such sparsity may come from edge estimation [52, 53, 54], wavelet transformation [55, 56, 57, 58], different orders of total variation (TV) [59, 60, 61, 62]. Most of the existing methods solve
the above optimization problem with various fidelity terms and regularity terms in either continuous or discrete setting. In this thesis, we will refer to them as one-stage methods.

Existing reconstruction methods in the literature can be classified into two categories: the discrete models and the continuous ones. The discrete models view the underlying image as a discrete vector with certain fixed resolution and usually obtain its approximation through solving a discrete optimization problem consisting of a fidelity term and a regularity term. There are various regularity terms used in the literature, such as TV [1, 63], total generalized variation [59, 64], and total fractional variation (TFV) [65, 66, 60]. Moreover, some efficient algorithms such as alternating direction method of multipliers (ADMM), primal-dual methods have been proposed [67, 68, 69] to solve corresponding optimization problems. On the other hand, the continuous models consider the underlying image as a piecewise smooth function and recover the image from a function approximation point of view. One of its advantages is the flexibility in setting resolution and it has been successfully employed in image super resolution [70, 71]. It has also been shown to have superior performance in generalized/infinite-dimensional compressive sensing reconstruction [72, 73].

In this thesis, we will leverage both discrete and continuous modeling to develop a two-stage projection correction model (PCM). The first stage (P-stage) employs a continuous model and the second stage (C-stage) imposes a discrete regularization/penalty term on the model. Specifically, at the P-stage, we consider the underlying image as a function in a processing domain that is usually a Hilbert space spanned by some basis functions such as polynomials or wavelets. We will find an “optimal” approximation (projection) in the processing domain by minimizing a certain data fidelity term. In other words, the P-stage projects the Fourier measurements into another processing domain that has an accurate representation of the underlying function. We estimate the coefficients of $f$ under the basis
by solving a least squares problem (P step in (2.1)). Due to imperfect selection of basis and errors in data, $c_f$ might contain some errors. In the C-stage, we update the coefficients by minimizing the sum of a fidelity term and a regularity term. We will employ a spatial regularity that takes image smoothness and sharpness into consideration. Concisely, our proposed PCM framework is as follows,

\begin{align}
\text{P : } c_f &= \underset{c}{\text{argmin}} \|S\mathcal{F}\Phi c - \hat{f}\|_2^2, \\
\text{C : } c_g &= \underset{c}{\text{argmin}} \left\{ \frac{1}{2}\|c - c_f\|_2^2 + \mu R(c) \right\}
\end{align}

where

- $\hat{f}$ is the given Fourier samples,
- $\Phi$ is the pre-selected basis that spans the processing domain,
- $\mathcal{F}$ denotes the continuous Fourier transform,
- $S$ is the sampling operator,
- $\mu$ is a constant regularization parameter.

We use $c_g$ obtained from the C-stage to obtain the reconstruction: $g = \Phi c_g$. Details of the PCM framework are presented in Chapter 3. Furthermore, at the C-stage, we propose a domain adaptive regularization term named TV-TFV which combines the advantages of total variation (TV) and total fractional variation (TFV). This helps maintain the edge details as well as the smoothness of the non-edge parts of the image while the standard TV- or TFV-based model can only achieve one of these. We will discuss the details in Chapter 4. Numerical experiments in Chapter 5 show the superior performance of the proposed approach compared to other popular ones.
2.2 Algorithms for Dictionary Learning

In recent years, there have been a substantial interest in alternatives to traditional signal representations. Instead of the fixed off-the-shelf bases or dictionaries, such as sinusoids, wavelets [74], and Gabor dictionaries [75], data dependent bases have been extensively studied and shown superior performance in many vision tasks, including image denoising [34], super-resolution [76], object recognition [77] and classification [78]. Indeed, many widely used model strategies can be viewed from the basis learning perspective. For instance, principal component analysis (PCA) learns bases that lie along the directions of maximum data variation. Independent component analysis (ICA) [79] decomposes a multivariate signal into additive subcomponents which are assumed non-Gaussian.

In this thesis, we study dictionary learning which forms an important component of sparse modeling. Starting from a set of learning data, it searches for an overcomplete basis in which each element of the training data can be sparsely represented. It was first introduced in Olshausen and Field’s work [30] which studied the receptive fields of cells in the mammalian primary visual cortex. The work showed that human visual cortex encodes the retinal images with an overcomplete basis set defined by the spatial receptive fields of simple cells. Thereafter, many algorithms are proposed for finding such basis from the given data.

Given a set of training signals, dictionary learning models usually solve two subproblems: 1) finding the ideal basis (dictionary); 2) solving the sparse coefficients (code) for each signal on this basis. Most existing algorithms solve the problem in an alternating and iterative manner: update either the dictionary or the sparse code while fixing the other. Strategies for updating the dictionary can be roughly classified into two categories: spectral methods and the gradient descent methods. The gradient based methods [43, 44, 80, 81]...
update the dictionary by maximizing \textit{a posteriori} with the fixed code. However, the sparsity of the code is ignored in this step. When the algorithm moves towards the next stage of finding the optimal code, the resulting solution might not be sparse. On the other hand, the spectral methods \cite{33, 82, 83} update the dictionary using restricted SVD over the support (nonzero parts) of the code. The zeros parts will then not be affected. Unfortunately, due to the complicated spectral structure of the code as well as the given signals, the convergence of such methods is not guaranteed.

One of the main challenges for modern dictionary learning algorithms comes from the big data regime. In machine learning literature, millions of images are used for training object recognition models, for example, the ImageNet challenge \cite{84}. Furthermore, dictionaries are usually learned from extracted patches rather than the whole images. Since a single image can easily produce hundreds of patches, solving the dictionary learning from a single machine is almost impossible. To tackle this big data challenge, several distributed algorithms have been proposed. The work in \cite{85, 86} adapted a widely used strategy named diffusion to distributed dictionary learning where the final learned dictionary is a weighted average of local updates. Recent works in \cite{87, 88} use alternating direction method of multipliers (ADMM) where each local dictionary is updated under a consensus constraint. The cloud K-SVD \cite{83} employs the distributed principal component model to refine the local dictionaries similar to the centralized K-SVD update. However, none of the existing methods considers the structures of the dictionary during the updates which focus on correct code recovery. In this thesis, we propose a novel distributed dictionary learning algorithm to learn incoherent dictionaries. In our method, local dictionaries are first obtained using classical spectral methods. We then adopt a clustering strategy to find a candidate for the unified dictionary. A generalized consensus update is then employed. Furthermore, We demonstrate its performance in part III with numerical experiments.
2.3 Feature Selection and the Lasso

Feature selection is an extremely important part of machine learning algorithms. Finding a good set of features reduces overfitting, improves robustness to noise and enables faster convergence to the target. Various strategies have been proposed in the literature for feature selection, including filter-based, wrapper-based and embedded methods. In this thesis, we are interested in embedded feature selection methods. Here, the learning objective is modified by introducing an extra term which typically encourages “sparsity” in the solution. To be precise, we consider the following data generation model,

\[
y_i = g(x_i^T w) + \epsilon_i, \quad i = 1, 2, ..., n,
\]

where \(\epsilon_i's\) are additive noise, function \(g : \mathbb{R} \rightarrow \mathbb{R}\) is a mapping function, \(x_i's \in \mathbb{R}^p\) are independent and identically distributed feature vectors. We assume that the \(x_i's\) are generated from an underlying Gaussian distribution, \(x_i \sim \mathcal{N}(0, \Sigma)\). Given data pairs \(\{y_i, x_i\}\), feature selection aims to find the support of \(w\), i.e., the indices of nonzero entries.

One of the best known examples of the embedded approach is the Least Absolute Shrinkage and Selection Operator, or Lasso \([14]\). The optimization problem of the Lasso model has the following form,

\[
\min_w \| Xw - y \|_2^2 + \lambda \| w \|_1,
\]

where \(\lambda\) is a regularization parameter. In the Lasso, feature sparsity is encouraged through the addition of a \(\lambda \| w \|_1\) term, where \(\lambda\) is a coefficient that trades off the importance of the loss and Lasso term. This \(l_1\)-norm term is a good convex approximation to the \(l_0\)-seminorm, which directly counts nonzero feature coefficients. It is good since one can show
that under suitable conditions, solving optimization problem with the $l_1$-norm regularity and $l_0$-seminorm regularity can both recover the exact sparse solution. Furthermore, the $l_1$-norm regularity is convex. As a consequence, in Lasso, the resulting optimization problem remains convex if the loss term is convex, so that there is a well-defined global optimum. Finally, recent advances in convex optimization have made great strides in designing efficient and effective methods to deal with objective functions involving Lasso term \[89\]. As a result, the Lasso is one of the most widely used feature selection models in machine learning, and has consistently shown good empirical results.

In this thesis, we are interested in theoretically characterizing the behavior of the Lasso. For any feature selection strategy, a key question is whether, given enough data, it can recover the “true” underlying set of features for different mappings. This question can be expressed through selection consistency: given enough data, can we be sure that the set of nonzero coefficients in the learned solution will be the same as that in the ground truth?

Typically, we are interested in the selection consistency of a feature selection technique under model misspecification, i.e. the ground truth mapping $g$ is different from our assumption. After all, in general we do not know what the target concept looks like. Thus in general, the function space of $g$ explored during learning may not contain the ground truth mapping. A key question then is whether a feature selection strategy can be selection consistent even in this case.

In Part IV, we focus on the Lasso feature selection when $g$ is misspecified. Building on prior work \[90, 91\], we prove that, for certain data distributions and nonlinear regularity over the mapping $g$, the answer is affirmative. This seems a surprising result and may further explain the empirical success of the Lasso. This result extends to the group Lasso as well. We then perform numerical studies to support the theoretical results.

We now briefly review the current state of the art. Mathematical details will be dis-
cussed in Part IV. The Lasso selection model [14] has been extensively studied in the past two decades. In the context of signal processing, this approach corresponds to basis pursuit, pioneered by Chen et al. in [19]. Benefiting from the $l_1$-norm regularization, the solution of the Lasso and basis pursuit can successfully achieve desired sparsity. When the data has no noise, a solid theoretical foundation has been built by many researchers, e.g. [92, 93, 5, 20].

The closest line of work to ours starts with [94], where the authors analyze the estimation consistency conditions for various Lasso type models: the regularization term $f(w)$ is not only $l_1$-norm, but any $l_q$-norm with $q \in (0, 2]$. In particular, they show that by choosing $\lambda$ that grows asymptotically as fast as $n^{1/2}$, Lasso tends to select the true model with non-vanishing probability as $n$ grows. A sufficient condition to guarantee the selection consistency of the Lasso, namely irrepresentability condition, was independently proposed by [95] and [96]. The necessity of this condition was proved in later work [16, 97]. A precise characterization of the relation between the regularization parameter $\lambda$, $n$ and $p$ (the dimension of $w$) to guarantee consistent selection is shown in [98]. In order to select the true features with high probability, one should expect $\lambda$ grows at least in the order of $\log(p^{-1/2})n^{1/2}$. While these results are substantially based on sparse linear regression, consistency results on other generalized linear models such as logistic regression have also been developed [99, 100]. In the context of nonlinear regression, [101] shows the advantage of the Lasso through numerical experiments when the mapping function is linearized by finite Gaussian basis functions.

Prior work [102, 103] provides a systematic approach to analyzing the estimation consistency of general sparse models. [103] introduces the notion of restricted strong convexity, a property that guarantees nice curvature structure of the loss function near the true features, and establishes a series of estimation error bounds on sparse regression models. This framework is extended to selection consistency analysis by [104].
Much of the work mentioned above relies on the target regression function being known to be a linear or logistic regression relation, even though obviously the Lasso model itself does not impose any such prior knowledge. Recent work by Thrampoulidis et al. [105] investigated the case of nonlinear link functions and presented consistency results for the Lasso. However, their results are only precise in the context of the estimation consistency which is not directly applicable in feature selection perspective. The work we present does not impose any prior knowledge on the form of the regression function beyond the qualifications outlined above, and is applicable to the standard, fixed-dimensional feature selection case with a fixed and unknown subset of relevant features.
Part II

Sparse Modeling in Image Reconstruction
Chapter 3

Projection Correction Modeling

In this part we present a novel two-stage projection correction modeling (PCM) framework for image reconstruction from (nonuniform) Fourier measurements. The idea can be extended to other types of measurements. PCM consists of a projection stage (P-stage) motivated by the multi-scale Galerkin method, and a correction stage (C-stage) with an edge guided regularity combining the advantages of total variation (TV) and total fractional variation (TFV). The P-stage allows for continuous modeling of the underlying image of interest. The given measurements are projected onto a space in which the image is well represented. We then enhance the reconstruction result at the C-stage that minimizes an energy functional consisting of a fidelity in the transformed domain and a novel edge guided regularity. In the following chapters, we further develop efficient proximal algorithms to solve the corresponding optimization problem. In particular in this chapter, we study the image reconstruction problem and demonstrate the effectiveness of the general Projection-Correction Modeling (PCM) scheme. In Chapter 4, we will discuss the domain adaptive TV-TFV regularity. We further combine PCM with TV-TFV to show a complete frame-
work and demonstrate the effectiveness with numerical results in Chapter 5. To our best knowledge, PCM-TV-TFV is the first two-stage model with domain adaptive regularity for image reconstruction from Fourier measurements.

3.1 Model Overview

In the image reconstruction problem, the given data are some finite uniformly or nonuniformly sampled Fourier measurements. The goal is to reconstruct the underlying image from these measurements. As discussed in Chapter 2, we will leverage both discrete and continuous models to develop a two-stage framework in which the first stage (P-stage) employs a continuous model and the second stage (C-stage) imposes a discrete regularization/penalty term on the model. At the P-stage, we consider the underlying image as a function \( f \) in a processing domain that is usually a Hilbert space spanned by some basis such as polynomials or wavelets. We will find an “optimal” approximation (projection) in the processing domain by minimizing a least squares term. In other words, the P-stage projects the Fourier measurements into another processing domain that has an accurate representation of the underlying function. We also point out that the P-stage could also be viewed as a dimension reduction step, since we project onto a subspace with much lower dimension. It will also help to reduce the computational time at the second stage. Due to the noise in the measurements and/or imperfect selection of the basis, the approximation in the P-stage will also contain errors. To further improve the reconstruction, we will impose a discrete regularization at the C-stage.

At the C-stage, we will find a “corrected” approximation in the same processing domain by minimizing the sum of a data fidelity term and a regularity term. The data fidelity is the difference between the corrected approximation and the approximated function obtained
at the P-stage. We will also employ a regularity term on the discrete vector that is the evaluation of the corrected approximation function on discrete grids. In particular, we will consider a hybrid regularity combining total variation and total fractional variation. The total variation (TV) regularization has very good performance in keeping edges in the reconstruction but suffers from the staircase artifact which causes oil-painted blocks. It is mainly due to the fact that TV is a local operator. On the other hand, the total fractional variation (TFV) is a recent proposed regularization term in image processing and has achieved promising results [65, 66, 60, 106, 107, 108]. The TFV is based on fractional order derivatives which are generalizations of the integer derivatives. It can be derived from Abel’s integral equation and more details can be found in [109, 110]. Imposing TFV reduces the staircase artifact due to its nonlocal nature. However, the edges are damped in the reconstructed image based on TFV regularization. Therefore, we will adopt a hybrid regularity with TV on the “edges” and TFV on the “smooth” part.

We remark that edge detection becomes an important task in our method, since we will not know where the true edges are before we proceed. We would like to mention that there is another research pipeline in multimodality image reconstruction where computed tomography (CT) and MRI scanning are run at the same time and CT images are used to enhance the performance of MRI as well as reduce the processing time [52, 111, 112, 113]. The set of edges are given since the CT is much faster than MRI. However, we consider a more challenging task in this thesis where the edges are not known but will be reconstructed recursively in our algorithm.
3.2 Problem Set-up

In this section we set up the problem and present the general idea of the proposed PCM framework for image reconstruction from Fourier measurements. Suppose the underlying image is a real-valued function \( f : \Omega \to \mathbb{R} \), where \( \Omega \subseteq \mathbb{R}^2 \) is a bounded domain of the image. We are given its Fourier data \( \hat{f} \) in the following form:

\[
\hat{f} = S \mathcal{F} f + \epsilon, \tag{3.1}
\]

- \( S \) is the sampling operator (might be uniform or nonuniform),
- \( \mathcal{F} \) is the continuous Fourier transform as

\[
\mathcal{F} f(w) = \int_{\Omega} f(x) \exp(-2\pi i w \cdot x) dx,
\]
- \( \epsilon \in \mathbb{C}^m \) is random noise.

Our goal is to recover the underlying image \( f \) from the given Fourier data. The challenges may come from the nonuniformness of \( S \) as well as the appearance of the noise \( \epsilon \). Since most images are assumed to be piecewise smooth with jumps at edges, the inverse Fourier transform will not work properly. The nonuniformness of the samples in the frequency domain will make the inverse process unstable and might bring extra approximation errors when the sampled data contain noise. Moreover, the Fourier basis is prone to Gibbs oscillations artifacts in representing a piecewise smooth function. To overcome the problems, we propose a two-stage PCM framework in next section. The general idea of such a PCM framework is presented in Figure 3.1.
We point out that the PCM framework can be viewed as a special case of the classical multiscale Galerkin method in finite element analysis. In contrast to the classical triangle or polyhedron mesh segmentation in a single scale, the region has elements in different scales. The advantage of the multiscale approach is to increase the numerical stability of the solver. More details can be found in Chapter 13 of [114].

Figure 3.1: Illustration of the general PCM framework. The given noisy measurements are first projected onto a space spanned by pre-selected basis \( \{ \phi_i \}_{i=1}^{n} \). The errors are then corrected by solving a minimization problem with regularity/penalty.

### 3.3 Details the Proposed PCM Framework

We now discuss the mathematical details of the proposed two-stage PCM framework.
3.3.1 The P-stage

Suppose \( \{ \phi_1, \phi_2, ..., \phi_n \} \) is a basis that could represent the underlying intensity function of interest and we look for an “optimal” approximation

\[
\tilde{f} = \sum_{j=1}^{n} (c_f)_j \phi_j,
\]

where the coefficients \( c_f \) are obtained through solving the following least squares problem:

\[
c_f = \arg\min_{c} \| \mathcal{S} \mathcal{F} c - \tilde{f} \|^2_2,
\tag{3.2}
\]

where \( \Phi c = \sum_{j=1}^{n} c_j \phi_j \). The above least squares problem could be solved efficiently by conjugate gradient solvers.

The performance of the P-stage depends on the selection of the basis \( \{ \phi_i \}_{i=1}^{n} \). We point out that the above framework has also been used in \([56, 115]\) for approximating the inverse frame operator with admissible frames, with subsequent analysis of the approximation error presented as well. In particular, the Fourier basis was used to obtain a stable and efficient numerical approximation of a smooth function from its nonuniform Fourier measurements. However, the Fourier basis might not be appropriate to represent a piecewise smooth function. Instead, we will consider wavelets, including classical ones such as the Haar or Daubechies wavelets or more recent ones such as curvelet \([116, 117]\), shearlet \([118, 119]\), which have a more accurate representation for piecewise smooth functions. In this regard, it could also be viewed as a generalization of the admissible frame method in \([56, 115]\). In this thesis, one-dimensional (1D) and 2D Haar wavelets are used in the numerical experiments, but the general PCM framework could also work with other wavelets.

We remark that even with a reasonable selection of basis \( \{ \phi_i \}_{i=1}^{n} \), \( c_f \) from eq. (3.2)
might not be accurate due to the noise in the data. We proceed to improve it in the correction stage to alleviate those effects.

### 3.3.2 The C-stage

We will find a “corrected” approximation

\[ g = \sum_{j=1}^{n} (c_g)_j \phi_j \]

in the same processing domain through solving the following regularization optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| c_g - c_f \|_2^2 + \lambda \mathcal{R}(c_g), \\
\text{subject to} & \quad \mathcal{R}(c_g) \leq T.
\end{align*}
\]

(3.3)

where \( \mathcal{R}(c_g) \) imposes prior knowledge. The idea is to find a new set of coefficients that are close to \( c_f \) (found in P-stage) such that the new approximation satisfies some regularities.

In terms of optimization, it provides the proximal guidance during the algorithm implementation. Defining \( Q = \{ c \mid \mathcal{R}(c) \leq T \} \) for some constant \( T \). If \( \mathcal{R}(\cdot) \) is a convex function, it can be verified using the method of Lagrange multipliers that problem (3.3) is equivalent to

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| c_g - c_f \|_2^2 \\
\text{subject to} & \quad \mathcal{R}(c_g) \leq T.
\end{align*}
\]

(3.4)

(3.5)

Problem (3.4) has a nice geometric interpretation: *for a given \( c_f \), find the closest point in \( Q \).* When \( Q \) is a closed convex set, the targeted \( c_g \) becomes the projection of \( c_f \) onto \( Q \), as is illustrated in Figure 3.2. For some special set \( Q \) (such as \( l_2 \)-norm convex ball), we have a closed-form solution of \( c_g \). We show this in Appendix A.
Figure 3.2: Geometric interpretation of optimization problem (3.3). The optimal $c_g$ is found by projecting the $c_f$ onto the closed convex set $Q$. 

\[ c_g \text{ found by projecting } c_f \text{ onto } Q \]
Chapter 4

Domain Adaptive Regularization

In this chapter, we discuss the choice of the regularity term $R(c_g)$ in the C-stage of PCM. In Section 4.1, we briefly review the fractional derivatives and their matrix approximations. In Section 4.2, we propose a new domain adaptive regularity which combines the advantages of total variation and total fractional variation. Furthermore, we show the application of the proposed regularity in image denoising. The numerical experiments are summarized in Chapter 5.

4.1 A Short Primer on Fractional Derivatives

One popular choice of the regularity term $R(c_g)$ is $\|c_g\|_1$, which assumes that the underlying function has a sparse representation in the processing domain. However, this might not be the best choice for the problem of image reconstruction from Fourier measurements. The underlying intensity function is not always sparse under the basis. Sparsity itself does not necessarily incorporate spatial smoothness or sharpness. Instead, we impose spatial regu-
larization that takes smoothness and sharpness into consideration. In particular, we shall employ a regularity that combines total variation (TV) and total fractional variation (TFV). To this end, we first review some definitions and notation of fractional order derivatives.

We point out that there are several definitions of fractional order derivatives, such as Riemann-Liouville (RL), Grunwald-Letnikov [120], and Caputo [121]. We adopt the RL definition [109, 110] in this thesis. The left, right, and central RL derivatives of order \( \alpha \in (n - 1, n) \), \( n \in \mathbb{N} \), for a function \( f(x) \) supported on an interval \([a, b]\), are defined by

\[
aD_x^\alpha f(x) = \frac{1}{\Gamma(n - \alpha)} \frac{d^n}{dx^n} \int_a^x (x - \tau)^{n-\alpha-1} f(\tau) d\tau,
\]

\[
xD_b^\alpha f(x) = \frac{(-1)^n}{\Gamma(n - \alpha)} \frac{d^n}{dx^n} \int_x^b (\tau - x)^{n-\alpha-1} f(\tau) d\tau,
\]

and

\[
aD_b^\alpha f(x) = \frac{1}{2} (aD_x^\alpha f(x) + xD_b^\alpha f(x) \cdot (-1)^n), \text{ respectively},
\]

where \( \Gamma(\alpha) \) is the Euler's Gamma function

\[
\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt.
\]

We next introduce the discretization of the RL fractional derivative. For simplicity of presentation, we will consider the 1D discretization over an interval \([a, b]\). The discretization over a 2D regular domain will be a direct extension of this procedure along horizontal and vertical directions. We consider the following \( n \) equidistant nodes on \([a, b]\):

\[
x_i = \frac{(i - 1)(b - a)}{n} + a, \quad i = 1, 2, \ldots, n.
\]

Let \( L^{(\alpha)} \) and \( R^{(\alpha)} \) be the matrix approximations of the left- and right-sided RL order \( \alpha \)
derivative operator \( aD^\alpha_x \) and \( xD^\alpha_b \) respectively. In other words, \( L^{(\alpha)} f(x_i) \) and \( R^{(\alpha)} f(x_i) \) approximate the order \( \alpha \) derivatives of \( f \) at \( x_i \). With Dirichlet boundary conditions \( f(a) = f(b) = 0 \), it follows [122] that \( L^{(\alpha)} \) and \( R^{(\alpha)} \) are two triangular Toeplitz matrices with the following structure:

\[
L^{(\alpha)}_n = n \begin{pmatrix}
w_0^\alpha & w_1^\alpha & \cdots & \cdots & w_{n-1}^\alpha & w_n^\alpha \\
0 & w_0^\alpha & w_1^\alpha & \cdots & \cdots & w_{n-1}^\alpha \\
0 & 0 & \ddots & \cdots & \cdots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & w_0^\alpha & w_1^\alpha & \vdots \\
0 & \cdots & \cdots & 0 & w_0^\alpha & w_1^\alpha \\
0 & \cdots & \cdots & \cdots & \cdots & 0 & w_0^\alpha
\end{pmatrix}
\]

and

\[
R^{(\alpha)}_n = n \begin{pmatrix}
w_0^\alpha & 0 & 0 & \cdots & \cdots & 0 & 0 \\
w_1^\alpha & w_0^\alpha & 0 & \cdots & \cdots & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
w_{n-1}^\alpha & \cdots & \cdots & w_0^\alpha & \cdots & 0 & 0 \\
w_n^\alpha & w_{n-1}^\alpha & \cdots & \cdots & w_1^\alpha & w_0^\alpha & 0 \\
w_n^\alpha & w_{n-1}^\alpha & \cdots & \cdots & \cdots & \cdots & \vdots
\end{pmatrix},
\]

where \( w_0^\alpha = 1, w_j^\alpha = (-1)^j \binom{\alpha}{j} \). These coefficients then can be constructed iteratively:

\[
w_j^\alpha = (1 - \frac{\alpha}{j})w_{j-1}^\alpha, \quad j = 1, 2, ..., n.
\]

Furthermore, when \( \alpha \in (1, 2) \), the matrix approximation of the central RL derivative \( C^{(\alpha)} \) becomes

\[
C^{(\alpha)}_n = \frac{1}{2}(L^{(\alpha)} + R^{(\alpha)}).
\]
Let $u$ be the discretized image under an ordinary $xy$-coordinate. For simplicity, we assume that the original image $u$ is square with $n = \sqrt{N}$ rows and columns. Let $U \in \mathbb{R}^N$ be the vectorized image with pixel values lexicographically ordered in a column vector. Let $\otimes$ denote the Kronecker product. It follows (from applying the central RL derivative $C^{(\alpha)}$ to the images along the $x$-direction) that

$$u^{(\alpha)}_x = (I_n \otimes C^{(\alpha)}_n)U,$$

where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix. Similarly, along the $y$-direction we will have

$$u^{(\alpha)}_y = (C^{(\alpha)}_n \otimes I_n)U.$$

The $l_1$ norm regularization over $u^{(\alpha)}_x$ and $u^{(\alpha)}_y$ leads to the TFV models [60].

We remark that even though $C^{(\alpha)}_n$ is a dense matrix, the matrix $I_n \otimes C^{(\alpha)}_n$ will be sparse. Furthermore, we observe that $w^\alpha_j$ decays very fast. For example, take $\alpha = 1.3$, the first six $w_j$’s,

$$w_0 = 1, \quad w_1 = -1.3, \quad w_2 = 0.195, \quad w_3 = 0.0455,$$

$$w_4 = 0.0493, \quad w_5 = 0.01, \quad w_6 = 0.006...$$

To further enhance the sparsity of the operation matrix, one can truncate $w_j$’s at a certain level to improve the efficiency of the program.
4.2 Proposed TV-TFV Regularity

Our proposed C-stage is in the following form:

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| c_g - c_f \|_2^2 + \mu_t \| \nabla g \|_1 + \mu_f \| \nabla^{\alpha} g \|_{\Gamma^c} \\
\text{subject to} & \quad g = \Phi c_g.
\end{align*}$$

(4.1)

where $\Gamma$ is an open domain centered around the edges of objects in the image, $\Gamma^c$ is its complement, $\nabla^{\alpha}$ is the discretized $\alpha$-order fractional differential operator.

We point out that an important question in the above model eq. (4.1) is how to select $\Gamma$. It denotes an open region centered around the edges rather than the edges themselves. The reason is that the set of edges has measure zero in $\mathbb{R}^2$ and it will not be useful to apply the 2D total (fractional) variation.

Construction of the $\Gamma$ set. Next we discuss the construction of the $\Gamma$ set. We would first detect the edges of the image reconstructed from the P-stage. In order to obtain accurate edges, we find a rough estimation at the initial step and then update it iteratively with the reconstructed image during the implementation of the reconstruction algorithm. Specifically, we use the first few iterations of the TV model [1] as a warm start to obtain an initial estimate of the edges. Meanwhile, we use an early termination in the iterative method to avoid staircase artifacts. We then apply a filter edge detector such as Sobel, Canny, and Prewitt filters [123, 124, 125] on the initial result to improve the accuracy of the edges.

Once the edges are obtained, an initial $\Gamma$ is readily constructed via morphologic dilation [126]; see Figure 4.1 for a visual illustration. The original image is a binary valued image. This operation converts the neighborhood of 0’s of the question mark to all 1’s. For all the numerical experiments, the converted neighborhood width is set to be 5 pixels.
Since the reconstruction algorithm we use runs iteratively, we obtain an intermediate image at each iteration. For each of the intermediate images, we construct the corresponding $\Gamma$ set. We then update the $\Gamma$ set through a few iterations of the previous results. Here, we abuse notation that we use $\Gamma$ to denote a binary matrix that is in the same size of the image. An entry of $\Gamma$ is 1 if the corresponding pixel is selected into the set.

We next discuss the update of $\Gamma$ in more detail. We denote $\Gamma$ obtained at the $i$th iteration by $\Gamma^{(i)}$. When estimating $\Gamma^{(k)}$, where $k > i$, we use the previous $\Gamma^{(i)}$'s,

$$\Gamma^{(k)} = \text{round} \left( \frac{1}{k - 1} \sum_{i=1}^{k-1} \Gamma^{(i)} \right),$$

where $\text{round}(\cdot)$ is the standard round function that returns the closest integer. This is to ensure that $\Gamma^{(k)}$ will be a binary matrix. A more general thresholding scheme with a given distribution $\mathcal{D}$ at a certain confidence level $t \in [0, 1)$ will be

$$\Gamma^{(k)} = \left(\sum_{i=q}^{k-1} w_i \Gamma^{(i)} > t \right)$$
for some $1 \leq q \leq k - 1$, $q \in \mathbb{N}_+$, $w_i \sim \mathcal{D}$, $w_q \leq w_{q+1} \leq \cdots \leq w_{k-1}$, $\sum_{i=q}^{k-1} w_i = 1$. Here $\mathbb{N}_+$ stands for the set of natural numbers.

We could further reduce the computational cost by ending the iterations early. We point out that the reconstructed image will become more accurate during the iterations of the algorithm and thus the variance in the detected edges is negligible after certain iterations. Therefore it is reasonable to use a small number of iterations in edge detection.

We summarize the procedure for reconstructing the initial $\Gamma$ set as follows:

1. Warm up and edge detection.
   
   (a) Run ROF [1] model for a few iterations (in our experiment $3 \sim 5$ will be enough). This depends on a rough estimate of the noise level. A quick noise variance estimation can be found at [127].

   (b) Use an filter based edge detector such as Canny or Sobel filters to detect the edge set.

2. Obtain $\Gamma$ through image dilation.

$\Gamma$ is updated as the underlying image is iteratively updated.

### 4.2.1 TV-TFV Regularity Solver

In this section we illustrate the advantages of the proposed domain adaptive regularity by applying it to the denoising problem. Let $\tilde{f}$ be a noisy image, we retrieve the underlying clean image through the following optimization problem,

$$
\minimize_f \quad \frac{1}{2}\|f - \tilde{f}\|^2 + \mu_\varepsilon \|\nabla f|_{\Gamma}\|_1 + \mu_f \|\nabla^\alpha f|_{\Gamma^c}\|_1,
$$

(4.2)
where \( \Gamma \) again is an open domain centered around the “edge” set and \( \Gamma^c \) is its complement. If \( \mu_f \) is set to 0 and the width of \( \Gamma \) is large enough, the above model will reduce to the classical TV denoising model.

\[
\min_f \quad \frac{1}{2} \| f - \tilde{f} \|^2 + \mu_t \| \nabla f \|_1.
\]

Similarly, if \( \Gamma \) is set to be \( \emptyset \), it will become the TFV model,

\[
\min_f \quad \frac{1}{2} \| f - \tilde{f} \|^2 + \mu_f \| \nabla f \|_1.
\]

The workflow of the image denoising problem with TV-TFV regularization is summarized in Figure 4.2.

Figure 4.2: Illustration of the TV-TFV regularization scheme in the image denoising problem.

Now we introduce the algorithm for solving the above TV-TFV regularization problem.
We start with a warm-up procedure to get an estimate of the edge regions. We then employ the alternating direction method of multipliers (ADMM) [68, 67] to derive the algorithm, as is presented in Algorithm 2. For more details of the ADMM algorithm, see Appendix B. Here the function \( \text{shrink}_{\mu/\lambda}(\cdot) \) is the componentwise soft thresholding function, defined as follows,

\[
(\text{shrink}_{\mu/\lambda}(v))_i = \begin{cases} 
  v_i - \lambda & \text{if } v_i > \mu/\lambda, \\
  v_i + \lambda & \text{if } v_i < -\mu/\lambda, \\
  0 & |v_i| \leq \lambda.
\end{cases}
\]

We demonstrate that the proposed TV-TFV regularization at the C-stage has a superior performance by numerical comparison with the TV denoising model and the TFV denoising model in the following subsection.

### 4.2.2 Performance of TV-TFV regularity

We consider the following general image denoising problem:

\[
\min_f \frac{1}{2} \|f - \tilde{f}\|_2^2 + \lambda R(f),
\]

where \( \tilde{f} \) is the given noisy image. We will compare three different regularities in the above model: TV, TFV, and TV-TFV.

We display the noisy image and the reconstructed images from these three denoising methods in Figure 4.3\(^1\). The regularization parameter \( \lambda \) is selected independently for each model to achieve best performance empirically. To better understand the performance, we

---

\(^1\)Image retrieved from \url{http://radiopaedia.org/} by Frank Gaillard.
Algorithm 2 TV-TFV denoising algorithm

Input: Noisy image $\tilde{f}$, confidence level $t$, length of stored sequence $n$.

1. Initialize: $\tilde{\Gamma}_0 = \Omega$, $f^1 = \tilde{f}$.

2. Warm up with $3 \sim 5$ TV iterations and update $\tilde{\Gamma}_0$ to $\tilde{\Gamma}_1$.

4. for $k = 1, 2, \ldots$ do
   if $k > n$ then
      6. $\tilde{\Gamma}^{(k)} = (\sum_{i=k-n+1}^{k} \tilde{\Gamma}^{(i)}) > n \ast t$;
   end if

8. Update $\Gamma^{(k)} = \text{dilate}(\tilde{\Gamma}^{(k)})$;
   Update $d^k = \nabla f|_{\Gamma^{(k)}}$, $e^k = \nabla^\alpha f|_{(\Gamma^{(k)})^c}$.

10. Solve $f$-subproblem

$$f^{k+1} = \argmin_f \left\{ \frac{1}{2} \| f - \tilde{f} \|^2_2 + \mu_t \| \nabla f|_{\Gamma^{(k)}} - d^k + dd^k \|_2^2 ight\}$$

$$+ \mu_f \| \nabla^\alpha f|_{\Gamma^{(k)}} - e^k + ee^k \|_2^2 \}$$

$d$-subproblem

$$d^{k+1} = \text{shrink}_{\mu_t/\lambda_t}(\nabla f^{k+1}|_{\Gamma^{(k)}} + dd^k)$$

12. $dd$-subproblem

$$dd^{k+1} = dd^k + \gamma_1(\nabla f^{k+1}|_{\Gamma^{(k)}} - d^{k+1})$$

$e$-subproblem

$$e^{k+1} = \text{shrink}_{\mu_f/\lambda_f}(\nabla^\alpha f|_{\Gamma^{(k)}} + ee^k)$$

14. $ee$-subproblem

$$ee^{k+1} = ee^k + \gamma_2(\nabla^\alpha f|_{\Gamma^{(k)}} - e^{k+1})$$

if $\| f^{k+1} - f^k \|_2 / \| f^k \|_2 < \epsilon$ then

16. break
end if
18. end for

Output: Denoised image $f^{k+1}$. 

42
zoom in on a region of interest (ROI) of the images and display them in Figure 4.4.

![Denoising results for models with different regularity/penalty. Here $\alpha$ denotes the fractional order of derivative in the TFV model.](image)

In Figure 4.4, we observe that the anisotropic TV suffers from the staircase artifact due to the fact that the TV regularization leads to piecewise constant solutions. On the other hand, the reconstruction with TFV regularity has a blurry effect on the edges. This is not surprising because the TFV is a nonlocal method and it is less edge sensitive than TV. Instead, the TV-TFV regularity has fewer artifacts and has a better reconstruction of both the edges and the overall image. Compared to the images obtained from TV or TFV model, the TV-TFV model recovers more details of the image.

We also present the numerical results on two measures, peak signal-to-noise ratio (psnr) and signal-to-noise ratio (snr). Let $u$ be the true image of size $d_1 \times d_2$ and $\hat{u}$ be the denoised
Figure 4.4: Detailed comparison between different models.
The measures psnr and snr are defined as follows:

\[
\text{psnr} = 10 \log_{10} \frac{d_1 d_2 (\max_{i,j} u_{ij})^2}{\|u - \hat{u}\|_F^2},
\]
\[
\text{snr} = 10 \log_{10} \frac{\|\text{mean}(\hat{u}) - u\|_F^2}{\|\hat{u} - u\|_F^2},
\]
\[
\text{rela\_err} = \frac{\|u - \hat{u}\|_F}{\|u\|_F}.
\]

The numerical results of different performance measurements are summarized in Table 5.3. The numerical results are consistent with our visual observation. The TV-TFV regularity shows better results in all the measurements.

Table 4.1: Numerical results for denoising with different regularity/penalty in Figure 4.3.

<table>
<thead>
<tr>
<th>Model</th>
<th>psnr</th>
<th>snr</th>
<th>rela_err</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td>20.9199</td>
<td>6.0853</td>
<td>0.3914</td>
</tr>
<tr>
<td>TV</td>
<td>31.0007</td>
<td>16.1661</td>
<td>0.1226</td>
</tr>
<tr>
<td>TFV</td>
<td>31.6910</td>
<td>16.8565</td>
<td>0.1133</td>
</tr>
<tr>
<td>TV-TFV</td>
<td><strong>32.1948</strong></td>
<td><strong>17.3602</strong></td>
<td>0.1069</td>
</tr>
</tbody>
</table>

We avoid using the notations of \(f, \hat{f}\), since \(\hat{f}\) is referred as the Fourier measurements of \(f\) through this part of the thesis.
Chapter 5

PCM-TV-TFV Modeling Framework

In this chapter, we present our novel PCM-TV-TFV model by incorporating the TV and TFV joint regularities into the PCM framework. To our best knowledge, this is the first two-stage model with domain adaptive regularities that is applied in image reconstruction from Fourier measurements. We first show the application of the PCM-TV-TFV in the image reconstruction and the corresponding algorithm. In Section 5.3, we demonstrate the effectiveness of the proposed model and compare them with widely used models.
5.1 Proposed PCM-TV-TFV Overview

Given the sampled Fourier measurements, our proposed two-stage reconstruction model consists of the following P-stage and the C-stage:

\[ P: \quad c_f = \min_{c} \| S \mathcal{F} c - \hat{f} \|_2^2, \]

\[ C: \quad \minimize_{c_g} \frac{1}{2} \| c_g - c_f \|_2^2 + \mu_t \| \nabla g |_{\Gamma} \|_1 + \mu_f \| \nabla^* g |_{\Gamma^c} \|_1 \]

subject to \( g = \mathcal{F} c_g \). \hfill (5.1)

where

- \( \hat{f} \) is the sampled Fourier measurements,
- \( \Phi \) is the pre-selected basis, \( \mathcal{F} \) denotes the continuous Fourier transform,
- \( S \) is the sampling operator,
- \( \Gamma \) is an open domain centered around the edges and \( \Gamma^c \) is its complement,
- \( \mu_t \) and \( \mu_f \) are constant regularization parameters,
- \( g \) is the image to be reconstructed.

The P-stage solves a least squares problem in which the given measurements are projected onto the space spanned by the columns of \( \Phi \). The bias and errors introduced in this step is then corrected by the adaptive TV-TFV regularization in the C-stage. We will demonstrate the advantages of the above proposed two-stage model through numerical comparison with other popular models later.
5.2 General PCM Model Solver

In this section we develop a proximal algorithm scheme for solving the general PCM optimization problem. Moreover, we also introduce an algorithm based on the split Bregman method [68] for solving the TV-TFV regularity problem. We then combine them to derive a specific algorithm for the PCM-TV-TFV model.

First we consider the following general PCM model with a general regularity term $\mathcal{R}$:

$$
P : \quad c_f = \arg\min_c \| \mathcal{S} \mathcal{F} c - \hat{f} \|^2_2,
$$

$$
C : \quad \text{minimize} \quad \frac{1}{2} \| c_g - c_f \|^2_2 + \lambda \mathcal{R}(c_g).
$$

We remark that the proximal algorithms refer to a class of algorithms that are widely used in modern convex optimization literature; for a comprehensive survey, see [69]. Now we review the definition of the proximal operator: for a convex function $\mathcal{R}(\cdot)$, the proximal operator $\text{prox}[69]$ is defined as

$$
\text{prox}_{\lambda \mathcal{R}}(v) = \arg\min_c \left\{ \mathcal{R}(c) + \frac{1}{2\lambda} \| c - v \|^2_2 \right\}. \quad (5.2)
$$

We point out that computing the proximal operator is equivalent to solving a trust region problem [128, 69]. Moreover, one can derive the closed forms of such proximal operators for many popular functions. For example, if $\mathcal{R}(c) = c^T A c$ for some symmetric positive semidefinite matrix $A$, then

$$
\text{prox}_{\lambda \mathcal{R}}(v) = (A + \frac{1}{\lambda} I)^{-1} v / \lambda.
$$
If $\mathcal{R}(c) = \|c\|_1$, we have

$$
(\text{prox}_{\lambda \mathcal{R}}(v))_i = \begin{cases} 
v_i - \lambda & \text{if } v_i > \lambda, \\
v_i + \lambda & \text{if } v_i < -\lambda, \\
0 & \text{if } |v_i| \leq \lambda,
\end{cases}
$$

(5.3)

which is the componentwise soft thresholding operator. In this case, this is the same as the \textbf{shrink} function in Section 4.2.1. Throughout the rest of the thesis, we will use $\text{prox}_{\lambda}(v)$ to denote this soft thresholding operation.

In the following, we present an efficient proximal algorithm in Algorithm 3 for solving the general PCM optimization problem, based on Nesterov’s accelerated gradient method as well as the fast iterative shrinkage-thresholding algorithm (FISTA) algorithm [129, 130, 67]. We use $A^\dagger$ to represent the pseudoinverse of the matrix $A$, given by $A^\dagger = (A^*A)^{-1}A^*$. 

\textbf{Algorithm 3} Accelerated proximal algorithm

1: \textbf{Input:} $\hat{f}$, $\Phi$, $c_0^g$, $d^0$, $\lambda^0$.
2: \textbf{P-stage:} Construct $\mathcal{S}\mathcal{F}\Phi$ from 3.2, solve $c_f = (\mathcal{S}\mathcal{F}\Phi)^\dagger \hat{f}$.
3: \textbf{C-stage:}
4: Initialize $c_0^g = c_f$, $k = 0$.
5: \textbf{while} rela\_err $>$ $\epsilon$ \textbf{do}
6: \hspace{1em} $k = k + 1$;
7: \hspace{1em} $\lambda^k = \frac{k}{k+3}$;
8: \hspace{1em} $c_g^k = \text{prox}_{\lambda^k \mathcal{R}}(d^k - \lambda^k(d^k - c_f))$;
9: \hspace{1em} $d^{k+1} = c_g^k + \lambda_k(c_g^k - c_g^{k-1})$;
10: \hspace{1em} rela\_err $= \|c_g^k - c_g^{k-1}\|_2/\|c_g^{k-1}\|_2$;
11: \textbf{end while}
12: \textbf{Construct} $g = \Phi c_g^k$.
13: \textbf{Output:} Reconstructed image $g$. 

49
In Algorithm 3, the P-stage step is designed to solve the least squares problem in problem (5.1) which has a closed-form solution. The C-stage involves an iterative proximal operator evaluation problem. As we have seen from (5.3) and (5.2), this step may also have a closed can be solved efficiently, for the evaluation problem of general proximal operators, see [69].

5.2.1 PCM-TV-TFV Solver

We now present an algorithm for solving the specific PCM-TV-TFV model (5.1) for image reconstruction with domain adaptive TV-TFV regularity, using a direct application of the alternating direction method of multipliers (ADMM) [67]. Since a direct solution of model (5.1) is not feasible, we decompose it into small sub-problems where each of them can be easily solved. Furthermore, for reconstructing a $256 \times 256$ image, the entire algorithm only takes a few seconds which is at the same level of time complexity of classical algorithms. However, we show in Section 5.3 that the proposed model outperforms the widely used models.
Algorithm 4 PCM-TV-TFV algorithm

**Input:** \( \hat{f}, \Phi, c^0, \lambda^0 \).

2: **P-stage:** Construct \( \mathcal{S}_f \Phi \) from eq. (3.2), solve \( c_f = (\mathcal{S}_f \Phi)^\dagger \hat{f} \).

3: **C-stage:** Initialize \( \bar{\Gamma}^{(0)} = \Omega \), confidence level \( t \), length of stored sequence \( n \).

4: **Warm up** with 3 ~ 5 TV iterations and update \( \bar{\Gamma}^{(1)} \) to \( \bar{\Gamma}^{(0)} \).

for \( k = 1, 2, \ldots \) do

6: if \( k > n \) then

7: \( \bar{\Gamma}^{(k)} = (\sum_{i=k-n+1}^{k} \bar{\Gamma}^{(i)}) > n \ast t \);

8: end if

9: Update \( \Gamma^{(k)} = \text{dilate}(\bar{\Gamma}^{(k)}) \);

10: **Update** \( d^k = \nabla(\Phi c_g^k)|_{\Gamma^{(k)}}, e^k = \nabla^\alpha(\Phi e_g^k)|_{\Gamma^{(k)}} \).

11: Solve \( c_g \)-subproblem

\[
\begin{align*}
c_g^{k+1} &= \arg\min_{c_g} \left\{ \frac{1}{2} \| c_g - c_f \|^2 + \mu_t \| \nabla(\Phi c_g)|_{\Gamma^{(k)}} - d^k + dd^k \|^2 \right. \\
&\quad \left. + \mu_f \| \nabla^\alpha(\Phi c_g)|_{\Gamma^{(k)}} - e^k + ee^k \|^2 \right\}
\end{align*}
\]

12: **d-subproblem**

\[
d^{k+1} = \text{prox}_{\mu_t/\lambda_t}(\nabla(\Phi c_g^{k+1})|_{\Gamma^{(k)}} + dd^k)
\]

13: **dd-subproblem**

\[
\begin{align*}
dd^{k+1} &= dd^k + \gamma_1(\nabla(\Phi c_g^{k+1})|_{\Gamma^{(k)}} - d^{k+1})
\end{align*}
\]

14: **e-subproblem**

\[
e^{k+1} = \text{prox}_{\mu_f/\lambda_f}(\nabla^\alpha(\Phi e_g^{k+1})|_{\Gamma^{(k)}} + ee^k)
\]

15: **ee-subproblem**

\[
ee^{k+1} = ee^k + \gamma_2(\nabla^\alpha(\Phi e_g^{k+1})|_{\Gamma^{(k)}} - e^{k+1})
\]

16: if \( \| c_g^{k+1} - c_g^{k} \|_2 / \| c_g^{k} \|_2 < \epsilon \) then

17: break

18: end if

end for

20: **Output:** Reconstructed image \( g \).
5.3 Numerical Results

In this section we demonstrate the superior performance of the proposed two-stage PCM framework for image reconstruction from Fourier measurements. In particular, we will use numerical experiments to show that

1. the projection step itself can achieve accurate recovery in the case without noise and bias error;

2. the PCM with TV regularity (PCM-TV) has a better performance than many of the state-of-the-art continuous models;

3. the TV-TFV regularity leads to better results in image denoising;

4. the PCM with TV-TFV regularity (PCM-TV-TFV) model further improves the results of the PCM-TV.

We will focus on the reconstructing a function from nonuniform measurements. In particular, we consider the jittered sampling in the frequency domain, that is, we assume the sampling is taken at the following frequencies:

\[ w_k = k + \eta_k, \quad \eta_k \sim U[-\theta, \theta], \quad k = \frac{-m}{2}, \ldots, \frac{m}{2} - 1. \]

Here \( U[-\theta, \theta] \) is the uniform distribution over the closed interval \([−θ, θ]\). We display an example of the jittered sampling in Figure 5.1. For the sampling scheme in 2D, we simply conduct jittered sampling on each of \( x \) and \( y \) directions. Most of the samples are off-the-grid, meaning that they are not on the 2D Cartesian grids. All the experiments were run with MATLAB R2015b on a PC with an Intel i7-3820 3.60 GHz CPU and 16.0 GB RAM.
5.3.1 Accurate Recovery of the P-Stage

We will use a simple example to demonstrate the accurate signal recovery generated by the P-stage when there is no noise and no bias error in the model. For a selected processing domain $\mathcal{H}_n = \{\phi_i\}_{i=1}^n$, we say it has no bias error if the underlying function $f \in \mathcal{H}_n$.

We will use the classical Haar wavelets in the processing domain. In the 1D case, the mother wavelet $\psi(x)$ is

$$
\psi(x) = \begin{cases} 
1 & \text{if } 0 \leq x < 1/2, \\
-1 & \text{if } 1/2 \leq x < 1, \\
0 & \text{otherwise.}
\end{cases}
$$

and its descendants are $\psi_{p,k} = 2^{p/2}\psi(2^px - k)$, $x \in [0, 1)$, $k = 0, 1, 2, \ldots, 2^p - 1$. The selected processing domain can be expressed as $\mathcal{H}_n = \text{span}\{\psi_{p,k}\}_{p,k}$. Here $n$ is the total number of basis vectors in this processing domain. Furthermore, we have $n = 2^p$. The 2D Haar wavelet is formulated by tensor product.
We consider the following piecewise constant test function:

\[
f(x) = \begin{cases} 
-1/2 & \text{if } -1/8 \leq x < 1/4, \\
1 & \text{if } 1/4 \leq x < 1/2, \\
-1 & \text{if } 1/2 \leq x < 5/8, \\
1/2 & \text{if } 5/8 \leq x < 3/4, \\
0 & \text{otherwise.}
\end{cases}
\]

We choose the first 32 \((n = 32, p = 5, k = 31)\) Haar wavelets. One can verify that the support of this test function is a subset of that of the selected Haar wavelets. Figure 5.2 shows that our proposed projection stage achieves a very accurate recovery for noiseless case.

### 5.3.2 PCM-TV Model

We consider in this subsection the case where the reconstruction contains bias error. In other words, the underlying function does not lie in the finite-dimensional subspace spanned by the chosen basis. We will consider the following piecewise linear test function \(f(x)\):

\[
f(x) = \begin{cases} 
1 & \text{if } 1/16 \leq x < 1/8, \\
-1/2 & \text{if } 1/8 \leq x < 1/4, \\
1 & \text{if } 1/4 \leq x < 1/2, \\
-\frac{8}{3}x + \frac{7}{3} & \text{if } 1/2 \leq x < 7/8, \\
0 & \text{otherwise.}
\end{cases}
\]
Figure 5.2: Bias-free reconstruction for piecewise a constant signal without noise: (a) Ground truth signal and its projection on $H_n$. (b) Reconstruction from truncated Fourier series which suffers from Gibbs oscillation. (c) Reconstruction from projection stage. Here $m = 128$, $n = 32$, $resol = 1/256$, $\theta = 0.25$. (d) The difference between the reconstruction and the ground truth.
We will use the same Haar wavelets to construct the processing domain. Figure 5.3 displays the bias error. In Figure 5.3 (b), we see that there are oscillating errors in the difference of the reconstructed function and the ground truth. This is due to the fact that the reconstructed function by Haar wavelets is a piecewise constant function, while the function \( f(x) \) is piecewise linear. Approximating the linear part of \( f(x) (1/2 \leq x \leq 7/8) \) with piecewise constant functions thus produces oscillating errors.

![Figure 5.3: (a) Piecewise linear function (red) and its projection onto \( \mathcal{H}_N, n = 128 \) (blue). (b) The difference between the two.](image)

Suppose we are given \( m = 256 \) nonuniform Fourier measurements with some added Gaussian noise \( \epsilon \) at various noise levels \( \sigma = \|\epsilon\|_2 / \|\hat{f}\|_\infty \).

We will use TV as the regularity at the C-stage of our two-stage PCM method and call it PCM-TV. Moreover, we will compare the proposed two-stage PCM-TV with the following popular one-stage method with different penalty:

- \( l_1 \) regularization model,

\[
\min_{\epsilon} \frac{1}{2} \|S\mathcal{F}\Phi c - \hat{f}\|_2^2 + \lambda \|c\|_1.
\]
• Tikhonov regularization model,

\[
\min_c \frac{1}{2} \| \mathcal{SF} \Phi c - \hat{f} \|^2 + \lambda \|c\|^2.
\]

• single-stage TV (SS-TV) model,

\[
\min_c \frac{1}{2} \| \mathcal{SF} \Phi c - \hat{f} \|^2 + \lambda \|\nabla \Phi c\|_1.
\]

In particular, we will compare them in terms of four different performance measures. To make this part self-contained, we list their definitions here. Let \(u\) be the true image of size \(d_1 \times d_2\) and \(\hat{u}\) be the reconstruction (of the same size). The measures peak signal-to-noise ratio (psnr), signal-to-noise ratio (snr), and relative error (rela_err) defined as follows:

\[
\text{psnr} = 10 \log_{10} \frac{d_1 d_2 (\max_{i,j} u_{ij})^2}{\| u - \hat{u} \|^2_F},
\]

\[
\text{snr} = 10 \log_{10} \frac{\| \text{mean}(\hat{u}) - u \|^2_F}{\| \hat{u} - u \|^2_F},
\]

\[
\text{rela_err} = \frac{\| u - \hat{u} \|^2_F}{\| u \|^2_F}.
\]

We also measures the structural similarity (ssim) [131] between \(u\) and \(\hat{u}\), defined as follows:

\[
\text{ssim} = \frac{(2\mu_u \mu_{\hat{u}} + c_1)(2\sigma_{u\hat{u}} + c_2)}{(\mu_u^2 + \mu_{\hat{u}}^2 + c_1)(\sigma_u^2 + \sigma_{\hat{u}}^2 + c_2)},
\]

where

• \(\mu_u\) is the average value of \(u\),

• \(\mu_{\hat{u}}\) is the average value of \(\hat{u}\),
• $\sigma^2_u$ is the variance of $u$,

• $\sigma^2_{\hat{u}}$ is the variance of $\hat{u}$,

• $\sigma_{u\hat{u}}$ is the covariance of $u$ and $\hat{u}$,

• $c_1$ and $c_2$ are constants that stabilize the division of the weak denominator. In all of our experiments, we follow the standard setup [131] to set $c_1 = (0.01 \ast d_1 \ast d_2)^2$ and $c_2 = (0.03 \ast d_1 \ast d_2)^2$.

The ssim index is usually calculated on the filtered images rather than the original ones. In all of our experiments, we use the 2D Gaussian filter of size $11 \times 11$ and standard deviation as 1.5. We present the numerical results in Table 5.1.

Table 5.1: Numerical comparison of the different models on different noise levels.

<table>
<thead>
<tr>
<th>$\sigma$</th>
<th>Model</th>
<th>ssim</th>
<th>psnr</th>
<th>snr</th>
<th>rela_err</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Proposed</td>
<td>0.9283</td>
<td>39.14</td>
<td>33.58</td>
<td>0.0161</td>
</tr>
<tr>
<td></td>
<td>$l_1$ regularization</td>
<td>0.8478</td>
<td>36.45</td>
<td>30.88</td>
<td>0.0220</td>
</tr>
<tr>
<td></td>
<td>SS-TV</td>
<td>0.8466</td>
<td>35.72</td>
<td>30.16</td>
<td>0.0239</td>
</tr>
<tr>
<td></td>
<td>Tikhonov</td>
<td>0.7640</td>
<td>32.17</td>
<td>26.60</td>
<td>0.0360</td>
</tr>
<tr>
<td>0.4</td>
<td>Proposed</td>
<td>0.8505</td>
<td>29.41</td>
<td>23.84</td>
<td>0.0495</td>
</tr>
<tr>
<td></td>
<td>$l_1$ regularization</td>
<td>0.7092</td>
<td>26.21</td>
<td>20.64</td>
<td>0.0715</td>
</tr>
<tr>
<td></td>
<td>SS-TV</td>
<td>0.6830</td>
<td>23.81</td>
<td>18.24</td>
<td>0.0942</td>
</tr>
<tr>
<td></td>
<td>Tikhonov</td>
<td>0.3041</td>
<td>21.09</td>
<td>15.52</td>
<td>0.1289</td>
</tr>
<tr>
<td>0.7</td>
<td>Proposed</td>
<td>0.7096</td>
<td>24.87</td>
<td>19.30</td>
<td>0.0834</td>
</tr>
<tr>
<td></td>
<td>$l_1$ regularization</td>
<td>0.6221</td>
<td>21.93</td>
<td>16.36</td>
<td>0.1170</td>
</tr>
<tr>
<td></td>
<td>SS-TV</td>
<td>0.6816</td>
<td>22.60</td>
<td>17.03</td>
<td>0.1083</td>
</tr>
<tr>
<td></td>
<td>Tikhonov</td>
<td>0.1414</td>
<td>15.51</td>
<td>9.951</td>
<td>0.2445</td>
</tr>
</tbody>
</table>

We can observe from Table 5.1 that our proposed PCM-TV has a better performance than all three other one-stage methods.
We next consider the 2D case. We will consider the 2D function $f$ with a randomly chosen square support of $[0.25, 0.5] \times [0.61, 0.83]$ where the entire image region is defined as $[0, 1]^2$. We display it in Figure 5.4. The 2D Haar wavelet used here is a direct product of the 1D Haar wavelet leading to $\mathcal{H}_n^2 := \mathcal{H}_n \otimes \mathcal{H}_n$.

![Figure 5.4: Reconstruction comparison with bias error for the 2D image. Here $m = 128 \times 128$, $n = 64 \times 64$, imsize = 256 \times 256, $\sigma = 0.6$.](image)

In particular, we will use a piecewise constant test function whose support does not lie in $\mathcal{H}_n^2$. It consists of four squares and the white ones do not lie in $\mathcal{H}_n$, which will cause the bias error in reconstruction.

We will compare the proposed PCM-TV model with the SS-TV method. We point out that TV regularization usually yields better results than $l_1$ regularization and $l_2$ regularization in 2D imaging problem. We present the corresponding numerical results and computational time of the split Bregman iterations in Table 5.2.

Table 5.2: Numerical details for the 2D reconstruction in Figure 5.4 with bias error, $\sigma = 0.6$.

<table>
<thead>
<tr>
<th>Model</th>
<th>psnr</th>
<th>snr</th>
<th>rela_err</th>
<th>time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS-TV</td>
<td>33.8196</td>
<td>20.87</td>
<td>0.086</td>
<td>10.8</td>
</tr>
<tr>
<td>PCM-TV</td>
<td>35.6589</td>
<td>22.7092</td>
<td>0.069</td>
<td>11.0</td>
</tr>
</tbody>
</table>
We observe from the above numerical results that the two-stage PCM-TV has a better performance than the SS-TV method. We point out that the two-stage PCM-TV model is equivalent to the following optimization problem:

\[
\min_{c_g} \frac{1}{2} \| c_g - A^\dagger \hat{f} \|^2 + \lambda \mathcal{R}(c_g),
\]

where \( A = S \Phi \) and \( \mathcal{R} \) is the TV operator. The corresponding first order optimality condition (from Fermat’s rule [132]) implies that

\[
0 \in c_g - A^\dagger \hat{f} + \lambda \partial \mathcal{R}(c_g), \tag{5.4}
\]

where \( \partial \mathcal{R} \) denotes the subdifferential of \( \mathcal{R} \) which is a set. On the other hand, for the SS-TV model

\[
\min_{c_g} \frac{1}{2} \| Ac_g - \hat{f} \|^2 + \lambda \mathcal{R}(c_g),
\]

the first order condition implies that

\[
0 \in c_g - A^\dagger \hat{f} + \lambda (A^*A)^{-1} \partial \mathcal{R}(c_g). \tag{5.5}
\]

Compared with eq. (5.4), the descent direction in eq. (5.5) is distorted by the factor \( (A^*A)^{-1} \). This might not only cause extra computational cost but also result numerical instability and extra errors in computing the inverse of \( A^*A \).
5.3.3 Numerical Comparison between PCM-TV-TFV Model and PCM-TV Model

Finally, we combine the projection stage and the correction stage with TV-TFV regularity instead of TV regularity to further improve the performance.

We display the ground truth and the reconstructed images from various methods in Figure 5.5, including the inverse Fourier method, the PCM-TV method, the PCM-TV-TFV method, and the SS-TV-TFV method. We point out that the optimization parameters $\lambda$, $\mu_t$, and $\mu_f$ of the above models are all selected to achieve the best performance empirically.

Table 5.3: Numerical results for denoising with different regularity/penalty in Figure 5.5.

<table>
<thead>
<tr>
<th>Model</th>
<th>psnr</th>
<th>snr</th>
<th>rela_err</th>
<th>time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM-TV</td>
<td>23.1515</td>
<td>20.4278</td>
<td>0.0952</td>
<td><strong>12.3</strong></td>
</tr>
<tr>
<td>SS-TV-TFV</td>
<td>24.1506</td>
<td>21.1566</td>
<td>0.0875</td>
<td>37.1</td>
</tr>
<tr>
<td>PCM-TV-TFV</td>
<td><strong>25.3128</strong></td>
<td><strong>21.9563</strong></td>
<td><strong>0.0798</strong></td>
<td>24.4</td>
</tr>
</tbody>
</table>

We point out that the noisy image in Figure 5.5 is obtained directly by the inverse Fourier transform and we can see that the noise level is quite high in this case. All the PCM-TV, SS-TV-TFV, and PCM-TV-TFV models are able to produce more reasonable visual results. To see a deep comparison, we zoom in on the red square part of Figure 5.5 and present the approximation errors in Figure 5.6.

From Table 5.3 and Figure 5.6, we can observe that the proposed PCM-TV-TFV has a better performance than SS-TV-TFV in both accuracy and efficiency. Due to the extra effort of computing the TFV term, it takes more time than the PCM-TV method, but it obtains a much better accuracy.
Figure 5.5: Reconstruction comparison between PCM-TV, SS-TV-TFV and PCM-TV-TFV. Noisy image (b) is obtained from inverse Fourier transform. Here $m = 128 \times 128$, $n = 96 \times 96$, $imsize = 256 \times 256$, $\sigma = 0.4$. 
Figure 5.6: Zoomed-in comparison of the red square in Figure 5.5. Figure (a) is the true surface. Figure (b) shows the difference between the ground truth and the one reconstructed by PCM-TV. Relative error $\approx 62\%$. Figure (c) shows the difference between the ground truth and the reconstruction by SS-TV-TFV. Relative error $\approx 38\%$. Figure (d) shows the difference between the ground truth and the one reconstructed by PCM-TV-TFV. Relative error $\approx 32\%$. 
Part III

Sparse Modeling in Dictionary Learning
Chapter 6

Dictionary Learning and Classical Algorithms

In this part we study dictionary learning in the context of sparse modeling. Dictionary learning aims to find an overcomplete basis through which the given data can be sparsely represented. In this chapter, we first review two classical dictionary learning algorithms, namely the Method of Optimal Directions (MOD) and the K-SVD algorithm. Both of them are non-distributed algorithms that can only run on a single machine/processor. We then review the widely used consensus strategy in the distributed dictionary learning problem and discuss its limitations.

6.1 Overview

We first review the mathematical formulation of the dictionary learning problem. Given a dataset \( Y = [y_1, y_2, ..., y_N] \in \mathbb{R}^{n \times N} \), with each data point \( y_i \in \mathbb{R}^n \), we aim to find an
overcomplete dictionary \( D = [d_1, d_2, ..., d_K] \in \mathbb{R}^{n \times K}, d_i \in \mathbb{R}^n, K > n \), that leads to the best representation for each member in the set, under sparsity constraints. In other words, each of the given signals can be expressed as a linear combination of only a few vectors in this dictionary, or mathematically,

\[
y_i = D x_i, \quad x_i \in \mathbb{R}^K \text{ is sparse, } \forall i = 1, 2, ..., N.
\]

The coefficients vectors \( x_i \)'s are called sparse codes. Finding the desired dictionary \( D \) and the codes \( x_i \)'s leads to the following optimization problem,

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{N} \| y_i - Dx_i \|_2^2 \\
\text{subject to} & \quad \| x_i \|_0 \leq T_0, \quad \forall i = 1, 2, ..., N, \\
& \quad \| d_i \|_2 = 1, \quad \forall i = 1, 2, ..., K.
\end{align*}
\]

(6.1)

where \( T_0 \) is a constant determined by some prior knowledge of the sparsity of the \( x_i \)'s. To remove the magnitude ambiguity between \( D \) and \( x_i \), we usually assume that \( d_i \)'s are all unit vectors. To simplify notation, problem (6.1) has a matrix interpretation as

\[
\begin{align*}
\text{minimize} & \quad \| Y - DX \|_F^2 \\
\text{subject to} & \quad \| x_i \|_0 \leq T_0, \quad \forall i = 1, 2, ..., N, \\
& \quad \| d_i \|_2 = 1, \quad \forall i = 1, 2, ..., K.
\end{align*}
\]

(6.2)

Here, \( X \in \mathbb{R}^{K \times N} \) is the code matrix with the \( i \)th column as \( x_i \). Unfortunately, this problem is ill-posed, \( i.e. \) the solution is not unique. For example, let \( P \) be any permutation matrix, \( D^* \) and \( X^* \) be a pair of solution, then \( D^* P \) and \( P^{-1} X^* \) is also a pair of solution. Many algorithms have been developed in the literature and shown empirically to have great
performance. In next section, we review some classical algorithms for dictionary learning under a non-distributed setting, i.e. algorithms that work in a single machine.

### 6.2 Non-Distributed Algorithms

Existing algorithms for problem (6.2) leverage a two-stage optimization scheme that alternates between the dictionary \( D \) and the code \( X \) updates. More precisely, these algorithms iteratively update \( X \) and \( D \) by alternating between the following two stages,

1. **sparse coding stage**: given a randomly initialized dictionary \( D \in \mathbb{R}^{n \times K} \) or that is obtained from the previous iterate, this stage finds the sparse code matrix \( X \) by solving the following optimization problem,

\[
\begin{align*}
\text{minimize} & \quad \|Y - DX\|_F^2, \\
\text{subject to} & \quad \|x_i\|_0 \leq T_0, \quad \forall i = 1, 2, ..., N;
\end{align*}
\]  

(6.3)

2. **dictionary update stage**: given a previously obtained code matrix \( X \), this stage finds the optimal dictionary \( D \) by solving the following optimization problem,

\[
\begin{align*}
\text{minimize} & \quad \|Y - DX\|_F^2, \\
\text{subject to} & \quad \|d_i\|_0 = 1, \quad \forall i = 1, 2, ..., K.
\end{align*}
\]  

(6.4)

There have been many algorithms proposed for solving the optimization problems in each stage. In this section, we first review a popular algorithm called the orthogonal matching pursuit (OMP) algorithm that is used in the sparse coding stage. We then discuss two algorithms that are widely used in the dictionary update stage, namely the Method of Optimal Directions (MOD) and the K-SVD algorithm. A recent review on dictionary
The Orthogonal Matching Pursuit algorithm [18, 42]. Notice that in the sparse coding stage, the Frobenius loss is separable with respect to the $x_i$’s, we can then solve the optimization problem for each of them. To be precise, solving problem (6.3) is equivalent to solving $N$ individual problems,

$$\begin{align}
\minimize_{x_i} \quad & \|y_i - Dx_i\|_F^2,
\text{subject to} \quad & \|x_i\|_0 \leq T_0,
\end{align}$$

for $i = 1, 2, ..., N$. Unfortunately this is an NP-hard problem [134, 75]. Finding each optimal $x_i$ requires an exhaustive search of all the subsets of $D$ with order up to $T_0$. The OMP algorithm is a coordinate descent algorithm that iteratively selects one entry of $x_i$ at a time and updates it. It works in a similar way as the forward selection technique in statistics [133]. Several variants of OMP have been proposed that essentially differ in the way the new entry is updated at each iteration. In this thesis, we use the variant that was introduced by Gharavi-Alkhansari and Huang [135]. The algorithm is summarized in Algorithm 5. Even though the algorithm only provides an approximate solution of problem (6.5), OMP was shown by Tropp and Gilbert in [42] to be able to reliably recover a good solution if the dictionary $D$ is incoherent. We discuss this in Chapter 7. We next review two widely used algorithms in the dictionary update stage.

The Method of Optimal Directions [43, 44]. MOD is probably the earliest approach to tackle the optimization problem in the dictionary update stage. Suppose we have obtained the code matrix $X$ returned in the sparse coding stage, the MOD algorithm updates
**Algorithm 5** Orthogonal Matching Pursuit (OMP) algorithm.

1: **Target Problem:**

\[
\begin{align*}
\text{minimize} & \quad \|y - Dx\|_F^2 \\
\text{subject to} & \quad \|x\|_0 \leq T_0. 
\end{align*}
\]  

2: **Input:** Signals \( y \in \mathbb{R}^n \), sparsity parameter \( T_0 \), dictionary \( D_0 \in \mathbb{R}^{n \times K} \).

3: **Initialize:** active set \( \Gamma = \emptyset \) and \( x \leftarrow 0 \).

4: **while** \( |\Gamma| < T_0 \) **do**

5: select a new coordinate \( \hat{i} \) that leads to the smallest residual:

\[
(\hat{i}, \hat{x}) \in \text{argmin}_{i \in \Gamma^c, x \in \mathbb{R}^{|\Gamma|+1}} \|y - D_{\Gamma \cup \{i\}}x\|_2^2;
\]

6: update the active set and the solution \( x \):

\[
\Gamma \leftarrow \gamma \cup \{\hat{i}\}; \\
x_\Gamma \leftarrow \hat{x} \quad \text{and} \quad x_{\Gamma^c} \leftarrow 0.
\]

7: **end while**

8: **Output:** the sparse code \( x \).

dictionary \( D \) by solving the following optimization problem,

\[
\begin{align*}
\text{minimize} & \quad \|Y - DX\|_F^2
\end{align*}
\]  

(6.7)

Notice that problem (6.7) is not equivalent to problem (6.4), since we drop the constraints on \( d_i \)'s. When \( X \) has full column-rank, problem (6.7) has a closed form solution,

\[
D = YX(X(X^T)^{-1}).
\]  

(6.8)

We then normalize each column of \( D \) and return to the sparse coding stage to update \( X \). We summarize the entire algorithm for dictionary learning using the OMP and MOD in Algorithm 6.
Algorithm 6 Dictionary learning with OMP and MOD.

1: **Input:** Signals \(Y\), sparsity parameter \(T_0\), number of dictionary atoms \(K\), maximum iteration number \(t\).

2: **Initialize:** dictionary \(D^0\) from data samples or a known basis.

3: for \(i = 0, 1, 2, ..., t\) do

4: Given dictionary \(D^i\), update sparse code matrix \(X^i\) by the OMP algorithm.

5: Given \(X^i\), solve dictionary \(D^{i+1}\) from

\[
D^{i+1} \leftarrow Y X^i (X^i (X^i)^T)^{-1}
\]

6: Normalize each column of \(D^{i+1}\),

\[
d^{i+1}_j \leftarrow \frac{d^{i+1}_j}{\|d^{i+1}_j\|_2}, \quad j = 1, 2, ..., K.
\]

7: end for

8: **Output:** the dictionary \(D^t\).

The MOD algorithm is easy to implement and shows good empirical performance in several applications of dictionary learning [133]. The main drawback is that it does not scale well with the size of the data. The code matrix \(X\) has the same number of columns as \(Y\) and it can be computationally expensive to obtain the inverse of \(XX^T\) in eq. (6.8). Next, we introduce the widely used spectral method for dictionary learning, the K-SVD.

**K-SVD.** The K-SVD algorithm, proposed by Aharon et al. [33], is currently one of the most popular methods for dictionary learning. The algorithm leverages the same sparse coding strategy as the OMP, but each dictionary column is updated separately using spectral decomposition over the support of the code. When updating the \(j\)th column \(d_j\), the algorithm first finds the subset \(\Omega\) of signals which uses \(d_j\) in the current sparse decomposition. Then, the optimal \(d_j\) is obtained from SVD decomposition of \(\Omega\). In the mean time, the support of \(j\)th row of \(X\) is extracted and updated accordingly. The steps are summarized in Algorithm 7. Although the algorithm in general does not have convergence
guarantees, it usually converges very fast and is widely used in the applications of the dictionary learning nowadays [34, 36, 77].

Algorithm 7 Dictionary learning with OMP and K-SVD algorithm.

1: **Input:** Signals $Y$, sparsity parameter $T_0$, number of dictionary atoms $K$, maximum iteration number $t$.
2: **Initialize:** $D \leftarrow D_0$.
3: **for** $i = 1, 2, 3, ..., t$ **do**
4: Given dictionary $D_i$, update sparse code matrix $X_i$ by OMP algorithm in Algorithm 5.
5: Fix $x_i$'s, update $D$:
6: **for** $j=1,2,...,K$ **do**
7: Define index set $\Gamma := \{t = \{1, ..., N\} : x^t[j] \neq 0\}$ where $x^t$ denotes the $t^{th}$ row of $X$.
8: Compute the overall representation error matrix $E_j$, by
   $$E_j = Y - \sum_{t \neq j} d_t x^t$$
9: Find the sub-matrix of $E_j$ corresponding to $\Gamma$ and obtain $E_j^\Gamma$.
10: Apply SVD decomposition $E_j^\Gamma = U\Sigma V^T$. Update $d_j$ to be the first column of $U$ and the nonzero part of $x$ to be the first column of $V$ multiplied by $\sigma_1$, i.e. the largest singular value.
11: **end for**
12: **end for**
13: **Output:** the dictionary $D$.

All of the algorithms in this section work only on a single machine. In the next section, we briefly review the consensus strategy which is widely used in the distributed dictionary learning models.
6.3 Consensus Optimization Method

In this thesis, we consider a network consisting of $m$ local workers (computers or processors) and a master node distributed over a spatial domain. The local workers can only communicate with the master node. That is, the information in each local worker can only be sent to the master node. Such network is commonly studied in the literature [67]. In this thesis, our goal is to design a distributed algorithm for dictionary learning that works in such a network. In this work, we forbid any dataset transfer between the local workers and the master node while the only information that is allowed to be exchanged between them is the learned dictionary in each worker. Compared to the network settings in the previous distributed dictionary learning work [83, 87], such a requirement preserves the privacy of each local dataset.

We now set up the target mathematical problem. Given local datasets $Y_1, Y_2, ..., Y_m$ (not necessarily in equal size) distributed in $m$ local workers, our goal is to solve for a common dictionary $D$ for these datasets. In other words, all the data points in these datasets can be sparsely represented by this dictionary $D$. Finding such a dictionary corresponds to the following optimization problem,

$$\minimize_{D, X_i} \sum_{i=1}^{m} \| Y_i - D_i X_i \|_F^2$$

subject to

$$\| x_{i,j} \|_0 \leq T_0, \quad \forall j = 1, ..., N_i,$$

$$D_i = D, \quad i = 1, 2, ..., m,$$

(6.9)

where $x_{i,j}$ is the $j$th column of matrix $X_i$. This is known as the global consensus problem. The term consensus comes from the fact that all local dictionaries $D_i$’s should all agree, i.e. be equal. A widely used strategy in practice is called the (average) consensus strategy [136, 67]. The common dictionary $D$ is obtained by averaging the local dictionaries $D_i$’s.
A typical distributed algorithm with this strategy is shown in Algorithm 8.

It can be seen that the consensus update step is an simple averaging step. We remark that the consensus strategy was originally proposed in statistics for finding common vector variables [137]. Such a strategy can be rather problematic for the dictionary learning problem. We give a simple counter example. Suppose the master node gathers $D_1$ and $D_2$ from the local workers, where

$$D_1 = \begin{bmatrix} -1 & 1 & 2 \\ 1 & -2 & 0 \end{bmatrix}, D_2 = \begin{bmatrix} 1 & 2 & 1 \\ -1 & 0 & -2 \end{bmatrix}. $$

The consensus update will provide a common dictionary

$$D = \frac{1}{2}(D_1 + D_2) = \begin{bmatrix} 0 & 3/2 & 3/2 \\ 0 & -1 & -1 \end{bmatrix}. $$

However, this update is undesirable. If we compare $D_1$ and $D_2$ closely, they have the same columns except that the first column of $D_1$ is the negative of the first column of $D_2$. The correct common dictionary should be either $D_1$ or $D_2$. The simple averaging step returns a wrong result. As a consequence, the Algorithm 8 could converge very slowly (or even diverge). Given such an observation, we propose a novel distributed algorithm, specifically designed for dictionary learning, with generalized consensus update. We delve into the formulation of the proposed model in the next chapter.
Algorithm 8 Distributed dictionary learning with consensus update.

1: **Input:** Local datasets $Y_i, i = 1, 2, \ldots, m$, sparsity parameter $T_0$, number of dictionary atoms $K$, maximum communication number $C$.
2: **Initialize:** dictionary $D^0$ from data samples or a known basis.
3: **for** $cm = 1, 2, \ldots, C$ **do**
4:   **for** each **local worker** do
5:       Obtain $D_i^{cm}$ from the master node.
6:       Update $D_i^{cm}$, $X_i^{cm}$ by the MOD or the K-SVD algorithm on each local machine.
7:   **end for**
8: **while** in **master node** do
9:   Gather $D_i^{cm}$ from local machines.
10:  Update $D^{cm+1}$ from $D^{cm+1} = \frac{1}{m} \sum_{i=1}^{m} D_i$. (consensus update)
11: **Broadcast** $D^{cm+1}$ to local workers.
12: **end while**
13: **end for**
14: **Output:** the dictionary $D^C$. 

74
Chapter 7

Distributed Dictionary Learning

In this chapter, we address the limitations of the widely used averaging technique for dictionary learning and propose a novel clustering based strategy together with generalized consensus selection to learn a mutually incoherent dictionary from distributed data. We first review the classical consensus optimization in a new perspective from sparse modeling. The standard averaging scheme can be naturally generalized to a support recovery problem. In addition, we enforce the learned common dictionary to be incoherent, which implicitly places a further constraint on the selection matrix.

7.1 Generalized Consensus: Perspective From Sparse Modeling

As discussed in the previous chapter, the widely used consensus updates for estimating the common dictionary $D^k$ at the $(k+1)^{th}$ iteration is simply averaging all the locally obtained
Before we show the proposed generalized consensus update, let us consider an extreme case, where each local dictionary is identical and is equal to the ground truth. For simplicity, we drop the index $k$ in the following discussions. Since all the $D_i$’s are exactly the same, it is obvious that expression (7.1) returns the ground truth $D$. On the other hand, eq. (7.1) has the following matrix representation:

$$D_A S = D$$

where $D_A \in \mathbb{R}^{n \times Km}$ simply pads all $D_i$’s together (‘A’ refers to ‘all’),

$$D_A = [D_1, D_2, \cdots, D_m] = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,K} & d_{2,1} & d_{2,2} & \cdots & d_{m,1} & \cdots & d_{m,K} \end{bmatrix},$$

where $d_{i,j}$ represents the $j$th column of local dictionary $D_i$. Matrix $S \in \mathbb{R}^{Km \times K}$ is a selection matrix, written as,

$$S = \left[ \frac{1}{m} I_K, \frac{1}{m} I_K, \cdots, \frac{1}{m} I_K \right]^T,$$
where $I_K \in \mathbb{R}^{K \times K}$ denotes the $K \times K$ identity matrix. We can also express $S$ explicitly as

$$
S = \begin{bmatrix}
  s_{1,1} & 0 & \cdots & 0 \\
  0 & s_{2,2} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & s_{K,K} \\
  s_{K+1,1} & 0 & \cdots & 0 \\
  0 & s_{K+2,2} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & s_{2K,K} \\
  s_{K(m-1)+1,1} & 0 & \cdots & 0 \\
  0 & s_{K(m-1)+2,2} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  0 & 0 & \cdots & s_{Km,K}
\end{bmatrix}.
$$

Here all labeled $s_{ij}$’s are $1/m$.

We can observe that this matrix $S$ has two properties,

1. **Column-wise sparsity.** Most of the entries in each column are zeros. This is due to the fact that there is only one column selected from each $D_i$.

2. **Column-wise orthogonality.** This is, $S^T S = I_K/m$.

However, in the standard consensus problem, such $S$ is fixed. This leads to the drawbacks we discussed earlier. First, this results in an incorrect update step if columns of $D_i$’s are permuted, even though all of them are correct. Second, this way of updating $D$ is not structure dependent. That is, the obtained common dictionary does not necessarily maintain any special structures after this update. After broadcasting the obtained dictionary to the
other machines, the local sparse coding stage (fix $D_i$, update $X_i$) could fail completely due to the unstructured dictionary. In next section, we show that it is important to use an incoherent dictionary for successfully solving for the correct $X_i$’s with the OMP algorithm.

The question is, can we generalize $S$ to be adaptive such that the selection of the columns of $D_A$ is automatic? The answer is affirmative. Given matrix $D$ and $D_A$, we propose the following generalized consensus strategy by solving a sparse modeling problem,

$$
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{K} \| s_i \|_1 \\
\text{subject to} & \quad D_A S = D, \\
& \quad \| S^T S - I/m \|_F^2 \leq \epsilon, 
\end{align*}
$$

(7.2)

where $s_i$ is the $i$th column of $S$. The summation term in the objective function limits the $l_1$-norm of the columns of $S$, which as we have seen before, can effectively produce a sparse solution. This matches the column-wise sparsity observation. The constraint on the Gram matrix $S^T S$ would enforce $S$ to be column-wise uncorrelated. This is a weaker condition than strict orthogonality. By doing so, we allow different columns of $S$ to share supports, i.e. each row of $S$ can have more than one nonzero entries. This offers more flexibility and robustness in finding the optimal selection. Finally, it is also easier to solve than the strict orthogonality constraint which may result in optimization over the Stiefel manifold that could be computationally expensive.

Now, there is a major challenge in problem (7.2): we do not know the ground truth $D$. Otherwise there is nothing to solve. Indeed, the solution to problem (7.2) becomes trivial if there is no constraint on the targeted dictionary $D$. What kind of constraint on the desired dictionary is suitable for an optimal learning problem? We discuss this in the next section.
7.2 Finding the Optimal Common Dictionary

Since the ground truth dictionary \( D \) is not known, we then aim to find the best candidate to approximate it in the optimization problem (7.2). Given the gathered local dictionaries, our best estimation (the maximum likelihood estimation) \( D_0 \) for an initial is the permuted average:

\[
D_0 = \frac{1}{m} \sum_{i=1}^{m} \iota(D_i).
\]  

(7.3)

where \( \iota(\cdot) : \mathbb{R}^{n \times K} \rightarrow \mathbb{R}^{n \times K} \) is a permutation function to match the columns of \( D_i \)'s with those of \( D \). One of the approaches to approximate this function is the unsupervised clustering strategy. In this thesis, we use the simple \( K \)-means algorithm to cluster and average the columns of \( D_i \)'s, where \( K \) in our case is the number of atoms in \( D \).

We are now ready to solve problem (7.2) after replacing \( D \) with \( D_0 \). There is only one question left: by simply advocating sparsity and (almost) orthogonality we do not have guarantee of any structure on the final learned common dictionary \( D \). It is possible that we could obtain the exact same solution as the simple permuted average selection in eq. (7.3). To address this issue, we require the final common dictionary to have an incoherent structure, i.e. \( \| D^T D - I \|_F^2 \leq \epsilon_D \), where \( \epsilon_D \) is a given constant from prior knowledge. Specifically, we have the following definition for the coherence of a matrix,

**Definition 1** (Coherence). The coherence of a given matrix \( A \in \mathbb{R}^{n \times K} \) is

\[
\mu(A) = \max_{j \neq k} \frac{|\langle a_j, a_k \rangle|}{\|a_j\|_2 \cdot \|a_k\|_2}
\]

where \( a_j \) and \( a_k \) denote the \( j \)th and \( k \)th columns of \( A \), respectively.

A dictionary \( D \in \mathbb{R}^{n \times K} \) is said to be incoherent if \( \mu(D) \) is small\(^1\). It can be seen that

\(^1\)In the literature, a matrix \( A \) is incoherent if \( \mu(A) \) is \( O(\sqrt{n \log n/n}) \).
the coherence is a factor that determines how close are the columns of a matrix to being orthogonal. Every orthogonal matrix has an coherence of 0. For overcomplete matrix \( A \in \mathbb{R}^{n \times K} \), we always have \( \mu(A) > 0 \). A general overcomplete matrix \( A \) always has a lower bound on the coherence, which is,

\[
\mu(A) \geq \sqrt{\frac{K - n}{n(K - 1)}}.
\]

If every pair of different columns of matrix \( A \) has an inner product (normalized) equal to the bound, then \( \{a_i\}_{i=1}^K \) is called an optimal Grassmannian frame, see [138, 139] for details. Furthermore, an incoherent dictionary design guarantees the successful recovery of the support of the sparse code \( X \) in the sparse coding stage on each local machine. To be precise, recall that in the sparse coding stage, we are solving the following optimization problem for each code \( x \) (the column of matrix \( X \)),

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

Denote the solution of this optimization problem by \( x_S \). In [140], Gilbert, Muthukrishnan and Strauss showed that a reasonably good solution can be generated from Orthogonal Matching Pursuit algorithm (OMP) if \( D \) is incoherent. To be precise, they established the following theorem:

**Theorem 1** (Gilbert-Muthukrishnan-Strauss [140]). Let \( D \) have coherence \( \mu \) and assume that

\[
T < \frac{1}{8\sqrt{2}}\mu^{-1} - 1.
\]

For the code \( x \), Orthogonal Matching Pursuit generates an approximation \( x_S \) which satis-
\[ \| x^* - x_S \|_2 \leq 8\sqrt{T} \| x^* - x_{opt} \|_2, \]

where \( x^* \) is the ground truth, \( x_{opt} \) is the optimal \( T \)-term approximation.

A direct consequence of Theorem 1 is that if the underlying ground truth \( x^* \) is \( T \)-sparse, \textit{i.e.} the size of the support is correctly specified in the above problem, OMP returns the exact solution. A weaker condition named strong irrepresentable condition guarantees a sparse support recovery by solving the relaxed optimization problem where \( \| \cdot \|_0 \) is replaced by \( \| \cdot \|_1 \). We focus on the development of the algorithm in the remainder of the chapter and detail the theoretical analysis of the sparse modeling in Part IV.

Finally, putting everything together, given an initialized \( D_0 \), we propose the following generalized consensus optimization problem for the common dictionary update,

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| D - D_0 \|_F^2 + \lambda \sum_{i=1}^{K} \| s_i \|_1 \\
\text{subject to} & \quad D_A S = D, \\
& \quad \| S^T S - I/m \|_F^2 \leq \epsilon_S, \\
& \quad \| D^T D - I \|_F^2 \leq \epsilon_D,
\end{align*}
\]

(7.4)

where \( \epsilon_S \) and \( \epsilon_D \) are constants. In the next section, we discuss an efficient algorithm based on the alternating direction method of multipliers (ADMM) \cite{Boyd2011} to solve problem (7.4).
7.3 Distributed Incoherent Dictionary Learning (DIDL)

Problem (7.4) can be reformulated as a single variable optimization problem by substituting the constraint $D_A S = D$ into the objective function and the incoherent constraint over $D$,

$$
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| D_A S - D_0 \|_F^2 + \lambda \sum_{i=1}^{K} \| s_i \|_1 \\
\text{subject to} & \quad \| S^T S - I/m \|_F^2 \leq \epsilon_S, \quad \| S^T D_A^T D_A S - I \|_F^2 \leq \epsilon_D.
\end{align*}
$$

(7.5)

This problem has a smooth convex fidelity $\frac{1}{2} \| D_A S - D_0 \|_F^2$ and a non-smooth convex regularization term $\sum_{i=1}^{K} \| s_i \|_1$. Similar to the strategy used in Part II, we leverage the ADMM algorithm to decompose this function into easily solvable sub-problems by introducing auxiliary variable $W$ and $Z$. At the $(k+1)^{th}$ iteration, this corresponds to,

$$
S^{k+1} = \arg\min_S \left\{ \frac{1}{2} \| D_A S - D_0 \|_F^2 + \frac{\rho}{2} \sum_{i=1}^{K} \| s_i - w^{k+1}_i + z^{k+1}_i \|_F^2 \\
+ \mu_S \| S^T S - I/m \|_F^2 + \mu_D \| S^T D_A^T D_A S - I \|_F^2 \right\}, \quad (S\ sub-problem)
$$

$$
w^{k+1}_i = \arg\min_{w_i} \left\{ \lambda \| w_i \|_1 + \frac{\rho}{2} \| s_i - w_i + z^{k+1}_i \|_F^2 \right\}, \quad i = 1, 2, ..., K, \quad (W\ sub-problem)
$$

$$
Z^{k+1} = Z^k + \rho (S^{k+1} - W^{k+1}). \quad (Z\ sub-problem)
$$

Here $w_i$’s and $z_i$’s are columns of $W$ and $Z$ accordingly. Note that the updates on $W$ and $Z$ are direct consequences of standard ADMM, similar to the previous results in Part II. There are two quadratic constraints in the $S$ sub-problem (implicitly) controlling the incoherence of $S$ and $D$, balanced by $\mu_S$ and $\mu_D$. Taking derivatives directly with respect to $S$ and
setting it to 0 does not give a closed form solution to $S$, since this leads to

$$D_A^T(D_A S - D_0) + \rho(S - W^k + Z^k) + 4\mu_S S(S^T S - I/m) + 4\mu_D D_A^T D_A S(S^T D_A D_A S - I) = 0.$$ 

The $S$ sub-problem can also be solved using gradient descent method. However, it is computationally expensive. At the $k$th iteration, the gradient descent step for updating $S^{k+1}$ will be

$$S^{k+1} = S^k - \alpha D_A^T(D_A S^k - D_0) + \rho(S^k - W^k + Z^k) + 4\mu_S S^k((S^k)^T S^k - I/m) + 4\mu_D D_A^T D_A S^k((S^k)^T D_A D_A S^k - I),$$

which includes many matrix multiplications between $D_A$ and $S^k$. For simplicity, denote the Gram matrix $D_A^T D_A$ as $G$. At the $(k+1)$th iteration, we follow the idea of linearized ADMM\(^2\) and propose to linearize these quadratic terms as follows:

$$G S - D_A^T D_0 + \rho(S - W^k + Z^k) + 4\mu_S S^k((S^k)^T S - I/m) + 4\mu_D G S^k((S^k)^T G S - I) = 0.$$ 

That is, we change the quadratic terms of $S$ to linear ones by replacing one of $S$ variables with the previous iterate $S^k$. Furthermore, denoting $S^k(S^k)^T$ by $H^k$, we can obtain a

\(^2\)This is indeed different from what linearized ADMM does. However, we adopt the same logic/strategy that reduces the difficulty of the problem by employing an asynchronous step in updating $S$. 

83
closed-form solution by solving the following linear equation,

\[
(G + \rho I_{Km} + 4\mu S H^k + 4\mu D G H^k G)S = D^T_A D_0 + \rho(W^k - Z^k) + 4\mu S/m S^k + 4\mu D G S^k,
\]

where \( I_{Km} \in \mathbb{R}^{Km \times Km} \) is the identity matrix.

Once \( S^{k+1} \) is obtained, we can update \( W \) by solving for its columns from the \( W \) subproblem,

\[
w_i^{k+1} = \text{prox}_{\lambda/\delta}(s_i^{k+1} + z_i^k), \quad i = 1, 2, ..., K, \tag{7.7}
\]

where \( \text{prox}(\cdot) \) is the same proximal function defined in Part II as

\[
(\text{prox}_\lambda(v))_i = \begin{cases} 
  v_i - \lambda & \text{if } v_i > \lambda, \\
  v_i + \lambda & \text{if } v_i < -\lambda, \\
  0 & |v_i| \leq \lambda,
\end{cases}
\]

for any vector \( v = (v_1, v_2, ..., v_n) \in \mathbb{R}^n \).

Finally, we obtain \( S^* \) after the ADMM stops and construct the common dictionary from \( D = D_A S^* \). The overall proposed distributed incoherent dictionary learning algorithm is summarized in Algorithm 9.

### 7.4 Numerical Experiments

In this section we demonstrate the effectiveness of the proposed algorithm for dictionary learning with synthetic data and natural images. In the thesis, we consider a network topology where each local worker only exchanges information with the master node while
Algorithm 9 DIDL algorithm.

1: **Input:** signals $Y_i \in \mathbb{R}^{n \times N_i}$, $i = 1, 2, ..., m$ distributed in $m$ local machines, sparsity parameter $T_0$, number of dictionary atoms $K$, maximum communication number $C$.

2: **Initialize:** $D^0$ randomly.

3: **for** $cm = 1, 2, ..., C$ **do**

4:   **for** each local worker **do**

5:     Obtain $D^{cm}$ from the master node.

6:     Update $D^{cm}_i, X^{cm}_i$ by the MOD or the K-SVD algorithm on each local machine.

7:   **end for**

8: **while** in master node **do**

9:     Gather $D^{cm}_i$ from local machines, run K-means clustering to get $D^{cm}_0$.

10:    Initialized $S^0$ from the active support by K-means.

11: **for** $k = 0, 1, 2, ..., $ **do**

12:     update $S^{k+1}$ by solving the $S$ sub-problem (7.6);

13:     update $W^{k+1}$ by solving the $W$ sub-problem (7.7);

14:     update $Z^{k+1}$ from the $Z$ sub-problem.

15: **end for**

16: Stop when criterion is met. Obtain the final solution $S^*$.

17: **Update** common dictionary $D^{cm}$ by $D^{cm} = D^{cm}_A S^*$.

18: **Broadcast** $D^{cm+1}$ to local workers.

19: **end while**

20: **end for**

21: **Output:** the common dictionary $D^C$. 

---

85
the local workers don’t exchange data. Such networks are efficient since they reduce the communication cost between the local workers. We compare the performance of the DIDL algorithm with the consensus strategy, where the common dictionary is obtained as an average of local estimators. Furthermore, for certain experiments in which it is to process all the data on a single machine, we also compare the performance of these distributed algorithms with the non-distributed K-SVD algorithm. All of the experiments are run with MATLAB 2015b on a single PC (under simulated distributed settings) with Intel i7-3820 processor and 16 GB RAM.

7.4.1 Dictionary Learning From Synthetic Data

In this subsection, we evaluate the proposed DIDL algorithm to find the best sparse representations from synthetic data, which are distributed in different local nodes.

Following the setup in previous works [85, 87] which leverages consensus updates, we first randomly generate a dictionary \( D \) that consists of 60 atoms \( (K = 60) \). Each of the atoms has a dimension of 40 \( (n = 40) \) and follows an independent and identically distributed Gaussian distribution with zero mean and an identity covariance. Furthermore, each atom is normalized with respect to its \( l_2 \)-norm. A dataset \( Y \) that contains 4000 \( (N = 4000) \) examples of dimension 40 are created by randomly combining any 6 atoms \( (T = 6) \) in the dictionary. The weights in these combinations (\( i.e. \) nonzero entries of the code \( X \)) are also randomly (independent and identically distributed) generated from a standard Gaussian. In this section, the examples are distributed across 6 local nodes \( (m = 6) \) with each storing 600 \( \sim \) 700 (non-repeating) examples. Note that it is usually the case that local machines store different amounts (but in the same level) of data. We consider the following settings,
1. Noise-free data. The examples are generated without any noise.

2. Noisy data. Each of the examples is generated with additive Gaussian noise with zero mean and covariance of $\sigma^2 I$, where $I$ is the identity matrix.

We randomly generate data under these two settings and conduct dictionary learning using both DIDL and the consensus algorithm. We repeat this 100 times for each setup. For both algorithms, the communication number is fixed at 10, i.e. the local workers only exchange dictionary updates 10 times with the master node. Let $D$ and $X$ be the ground truth dictionary and sparse codes respectively in all these settings. Denote the dictionary and sparse codes learned from DIDL by $D_{DI}$, $X_{DI}$ and from consensus algorithm by $D_C$, $X_C$. Since the solution for dictionary learning in general is not unique, we compare the performance based on the relative reconstruction error, defined by

$$R_{DI} = \frac{\|DX - D_{DI}X_{DI}\|_F}{\|DX\|_F}, \quad R_C = \frac{\|DX - D_CX_C\|_F}{\|DX\|_F}.$$

Figure 7.1 shows the results of the experiments. For both noise-free and noisy settings, we can see that the reconstruction error overall decays during the communications. Compared with the consensus algorithm, our proposed DIDL has a sharper slope in the error decay and reaches smaller reconstruction errors at the same communication iteration.

To further demonstrate the performance of DIDL, we vary the sparsity ratio ($T/n$) as well as the noise standard deviation $\sigma$ and repeat the evaluations. Furthermore, to better understand the performance limit of the distributed algorithms, we run the K-SVD algorithm on the entire data on a single node without any splitting (known as centralized K-SVD). Obviously, the centralized K-SVD works the best. Our goal is for the proposed distributed dictionary learning results to be better than that of the consensus approach. The final results are summarized in Table 7.1.
Figure 7.1: Average reconstruction error vs. communication times \( c \) for DIDL (red triangle) and Consensus algorithm update (blue star) at different settings.

(a) Noisy Free Setting, \( T = 4, n = 40 \).

(b) Noisy Setting, \( T = 4, n = 40, \sigma = 0.1 \).
Table 7.1: Numerical comparison of the relative reconstruction errors between DIDL and consensus algorithm with different sparsity ratios \(T/n\) and noise levels (standard deviation). The reconstruction error for the centralized K-SVD is also shown as a reference, denoted by \(R_{\text{cent}}\).

<table>
<thead>
<tr>
<th>Sparsity ((T/n))</th>
<th>Noise ((\sigma))</th>
<th>(R_C)</th>
<th>(R_{DI})</th>
<th>(R_{\text{cent}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/40</td>
<td>0</td>
<td>12.3%</td>
<td>10.0%</td>
<td>7.47%</td>
</tr>
<tr>
<td>4/40</td>
<td>0</td>
<td>16.4%</td>
<td>7.05%</td>
<td>2.78%</td>
</tr>
<tr>
<td>8/60</td>
<td>0</td>
<td>24.6%</td>
<td>17.1%</td>
<td>15.8%</td>
</tr>
<tr>
<td>8/40</td>
<td>0.3</td>
<td>28.2%</td>
<td>25.4%</td>
<td>20.0%</td>
</tr>
<tr>
<td>8/40</td>
<td>0.1</td>
<td>21.1%</td>
<td>16.7%</td>
<td>9.75%</td>
</tr>
<tr>
<td>4/40</td>
<td>0.3</td>
<td>24.2%</td>
<td>17.7%</td>
<td>20.2%</td>
</tr>
<tr>
<td>4/40</td>
<td>0.1</td>
<td>15.7%</td>
<td>7.65%</td>
<td>3.61%</td>
</tr>
</tbody>
</table>

In Table 7.1, we see that our proposed algorithm has in general better performance than the consensus updates. For some cases, DIDL achieves the same level of performance as the centralized algorithm. Next, we apply the proposed algorithm to the denoising problem for natural images.

### 7.4.2 Dictionary Learning for Image De-noising

Dictionary learning has been successfully applied to image denoising problem. We use image denoising as an application example to further demonstrate the advantage of the proposed distributed dictionary learning approach. We adopt the model proposed by Elad and Aharon [34], where the desired image \(Y^*\) (vectorized) is recovered from the following optimization problem,

\[
\{Y^*, D^*, X^*\} = \arg\min_{Y,D,X} \left\{ \lambda\|Y - \hat{Y}\|_2^2 + \sum_{ij} \mu_{ij}\|x_{ij}\|_0 + \sum_{ij} \|Dx_{ij} - R_{ij}Y\|_2^2 \right\},
\]  

(7.8)
where \( \hat{Y} \in \mathbb{R}^{N_0^2 \times 1} \) is the vectorized corrupted image with additive noise, \( R_{ij} \in \mathbb{R}^{n \times N^2} \) is an extraction matrix such that \( R_{ij}Y \) is the vectorization of a \( \sqrt{n} \times \sqrt{n} \) patch centered at pixel \((i, j)\), \( D \) is a learned dictionary from the extracted patches and \( x_{ij} \) (a vector) is the sparse code associated with each patch. Given a learned dictionary \( D^\ast \) and the sparse code \( x_{ij}^\ast \) for each patch, problem (7.8) is a simple quadratic problem for which a closed form solution for \( Y^\ast \) can be obtained,

\[
Y^\ast = \left( \lambda I + \sum_{ij} R_{ij}^T R_{ij} \right)^{-1} \left( \lambda \hat{Y} + \sum_{ij} R_{ij}^T D^\ast x_{ij}^\ast \right).
\]  

(7.9)

We briefly summarize the steps of the entire procedure (readers can refer to [34] for more details)

1. Given a corrupted image \( \hat{Y} \), we first extract all the \( \sqrt{n} \times \sqrt{n} \) patches from the image. These patches are then vectorized and organized into a \( n \times N \) matrix \( \hat{Z} \), where \( N \) is the total number of patches extracted.

2. We run the K-SVD algorithm to learn a dictionary \( D^\ast \) and the sparse code \( x_{ij}^\ast \) from \( Z \). Notice that although \( Z \) is noisy, we can limit the sparsity level of \( x_{ij} \) such that sparse approximations to the columns \( \hat{Z} \) are obtained. These approximations will represent the clean image patches.

3. Finally, we fuse all the denoised patches back to reconstruct the final cleaned image \( Y \) by the closed form eq. (7.9).

In this subsection, we follow the same setup as in [34] but replace the centralized K-SVD algorithm with the distributed ones. Specifically, dictionaries appearing in all the experiments are set at a fixed size of \( 64 \times 256 \). Each extracted patch has a size of \( 8 \times 8 \). For a noisy \( 256 \times 256 \) image, all the available patches are extracted. In order to compare the
performance of the distributed algorithms with the centralized K-SVD, we extract every second patch in a $512 \times 512$ image. For all the sparse coding stage involved in these algorithms (i.e. fixed $D$, update $X$), the sparsity level $T$ is fixed as 16. The communication time is fixed as 3 and the number of local nodes are 6 for all the experiments in the subsection. Illustration of the patch extraction for a $256 \times 256$ image. We visualize the

**Figure 7.2:** Illustration of the patch extraction for a $256 \times 256$ image. The blue window shows the first extracted $8 \times 8$ patch of the image. Then the window slides to right with step size 1 (pixel) and extract another patch. After extracting all available patches in the horizontal direction, the window moves downwards for 1 (pixel) and repeat the extraction. The total number of patches will be $(256 - 7)^2 = 62,001$.

results for different figures with various levels of additive noise in Figures 7.3, 7.4 and 7.5. We use the peak signal-to-noise ratio (PSNR) to compare the performance, which is defined as follows,

$$PSNR = 10 \log_{10} \frac{N_0^2 (\max_i Y_{i*})^2}{\|Y^* - \hat{Y}\|_2^2},$$
where $Y_i^*$ is the $i$th pixel value of the vectorized image $Y^* \in \mathbb{R}^{N_0^2}$. A higher PSNR will in general indicate an image of better quality. As we can see from Figures 7.3, 7.4 and 7.5, the proposed DIDL shows better performance than the classical consensus algorithm and is able to produce very close denoising results as the centralized K-SVD. For Figure 7.3, the difference of the PSNR of the denoised image produced by the centralized K-SVD algorithm and DIDL is only 0.07. For Figure 7.5, the difference is only 0.06.
Figure 7.3: Visualization of image denoising results using centralized K-SVD, consensus dictionary learning algorithm (CDL) and the proposed DIDL.
Figure 7.4: Visualization of image denoising results using centralized K-SVD, consensus dictionary learning algorithm (CDL) and the proposed DIDL.
Figure 7.5: Visualization of image denoising results using centralized K-SVD, consensus dictionary learning algorithm (CDL) and the proposed DIDL.
Part IV

Sparse Modeling in Feature Selection
Chapter 8

Feature Selection with the Lasso

An important question in feature selection is whether a selection strategy recovers the “true” set of features, given enough data. In this part of the thesis, we study this question in the context of the popular Least Absolute Shrinkage and Selection Operator (Lasso) feature selection strategy. In particular, we consider the scenario when the model is misspecified so that the learned model is linear while the underlying relation between the features and the observations is (probably) nonlinear. In Section 8.1, we set up the problem and notation. In Section 8.2, we introduce some classical concepts and related theorems that are fundamental to our work. In Section 8.3, we review the primal dual witness method which is a widely used technique on the theoretical study of the optimization problem with $l_1$-regularity. In Chapter 9, we prove a new theorem on the selection consistency of the Lasso model. We also carry out numerical studies to empirically verify the theoretical results and explore the necessity of the conditions under which the proof holds.
8.1 Problem Set-up

Assume we are given \( n \) independent observations \((x_i, y_i), i = 1, 2, ..., n\) which are generated by some nonlinear model:

\[
y_i = g(x_i^T w) + \epsilon_i, \quad i = 1, 2, ..., n,
\]

where \( \epsilon_i \)'s are independent and identically distributed Gaussian random variables, \( \epsilon_i \sim N(0, \sigma^2) \), function \( g : \mathbb{R} \rightarrow \mathbb{R} \) is a nonlinear mapping function which is not known \textit{a priori}, \( x_i \)'s are independent and identically distributed feature vectors. In this work, following prior work in this area [90], we assume that the \( x_i \)'s are generated from an underlying Gaussian distribution, \( x_i \sim N(0, \Sigma) \). \( w \in \mathbb{R}^p \) is the weight vector we want to recover. We assume \( w \) has unit \( l_2 \)-norm, \( \| w \|_2 = 1 \). Without loss of generality, we assume \( w = (w_1, w_2, ..., w_q, w_{q+1}, ..., w_p)^T \), where \( w_j \neq 0 \) for \( j = 1, ..., q \), and \( w_j = 0 \) for \( j = q + 1, ..., p \). Let \( w_r = (w_1, w_2, ..., w_q)^T \) and \( w_z = (w_{q+1}, ..., w_p)^T \) denote the nonzero and zero parts of \( w \) respectively. Here the subscripts \( r \) and \( z \) can be read as “representable parts” and “zeros”.

Let \( y = (y_1, y_2, ..., y_n)^T \), \( \epsilon = (\epsilon_1, \epsilon_2, ..., \epsilon_n)^T \) and \( X \) be the \( n \times p \) data matrix whose \( i \)th row is \( x_i^T \). We consider the following feature selection model:

\[
\min_w \| Xw - y \|^2_2 + \lambda f(w).
\]

\( f : \mathbb{R}^p \rightarrow \mathbb{R} \) is a convex regularization function, which is \( \| \cdot \|_1 \) for the classical Lasso and \( \| \cdot \|_{1,2} \) for the group Lasso. Here, suppose the index set \( \{1, 2, ..., p\} \) is parsed into \( m \) non-overlapping subsets \( I_1, I_2, ..., I_m \) such that \( \bigcup_{j=1}^m I_j = \{1, 2, ..., p\} \) and \( I_j \cap I_k = \emptyset \). For a given vector \( w = (w_1, w_2, ..., w_p) \in \mathbb{R}^p \), the \( l_{1,2} \)-norm corresponding to such group division
is defined as $\|w\|_{1,2} := \sum_{j=1}^{m} \|w_{1j}\|_2$.

The solution $\hat{w}^{(n)}$ to this model is defined as:

$$
\hat{w}^{(n)} = \arg\min_w \left\{ \|Xw - y\|_2^2 + \lambda^{(n)} f(w) \right\}.
$$

(8.1)

We use the superscript $n$ to emphasize that the solution of the Lasso may depend on the number of the observations. Likewise, the regularization parameter $\lambda$ may depend on $n$ as well, and we use $\lambda^{(n)}$ when we wish to make this dependence explicit. We next formalize the notion of consistency, which is used to evaluate the performance of the technique.

**Definition 2** (Estimation Consistency). The solution $\hat{w}^{(n)}$ obtained from (8.1) is called estimation consistent if

$$
\|\hat{w}^{(n)} - w\|_2 \to_p 0, \quad n \to \infty.
$$

Here $\to_p$ means converges in probability.

**Definition 3** (Selection Consistency). The solution $\hat{w}^{(n)}$ obtained from (8.1) is called selection consistent if

$$
P(\text{supp}(\hat{w}^{n}) = \text{supp}(w)) \to 1, \quad n \to \infty,
$$

where $\text{supp}(w) = \{i|w_i \neq 0\}$ is the support of $w$.

Note that one consistency result does not necessarily imply the other. For example, consider $\hat{w}^{(n)} = (1, 2, \frac{1}{n}, \frac{1}{n^2}, ...)$ and $w = (1, 2, 0, 0, ...)$. Then $\|w - \hat{w}^{(n)}\|_2$ can be small enough such that $\hat{w}^{(n)}$ is estimation consistent with $w$, but the supports are different for any arbitrary large $n$. One can also easily construct an example that is selection consistent but not estimation consistent.

We now introduce a set of sufficient conditions that will allow us to guarantee selection
consistency for the Lasso even though the underlying model is generated by some unknown nonlinear link function.

**Assumptions.** Using the notation of representable parts and zeros above, we write the data covariance matrix \( \Sigma \in \mathbb{R}^{p \times p} \) as:

\[
\Sigma = \begin{pmatrix}
\Sigma_{rr} & \Sigma_{rz} \\
\Sigma_{zr} & \Sigma_{zz}
\end{pmatrix}
\]

We assume \( \Sigma \) is invertible and \( \Sigma_{rr} \) has bounded positive eigenvalues away from 0, that is, \( 0 < \Lambda_{min} \leq \Lambda(\Sigma_{rr}) \leq \Lambda_{max} < \infty \), for some constants \( \Lambda_{min} \) and \( \Lambda_{max} \). Here \( \Lambda(\Sigma_{rr}) \) denotes the eigenvalues of \( \Sigma_{rr} \). Furthermore, We assume the following:

- \( y \) has finite fourth moment \( E(y^4) < \infty \);
- The link function \( g \) is differentiable almost everywhere and \( E(|g(t)|) < \infty \) and \( E(|g'(t)|) < \infty \), for \( t \sim \mathcal{N}(0, 1) \);
- \( E(x_j^T x_j | g(x_i^T w)|^2) < \infty \), for \( j = 1, 2, ..., n \).

The last two assumptions are closely related to the existence of a practical solution as well as sufficient for accomplishing the desired result. We show in Section 9.2 that for some experimental functions that violate these assumptions, the Lasso fails to select the right features.

### 8.2 Theoretical Preparations

Before we proceed to our main results, we first introduce some concepts and present several useful lemmas. In prior work [90] [in Section 3, theorem 1], it is shown that even when the observed \( y_i \)'s are generated by some unknown link function \( g \), under certain assumptions,
the least squares estimator with linear regression fit is asymptotically centered around the true predictor times a scaling constant.

**Theorem 2. [90]** Let \( y_i = g(x_i^T w) + \epsilon_i, \ i = 1, ..., n \), \( x_i's \) are independent normals with mean 0 and non-singular covariance matrix \( \Sigma \), \( \epsilon_i's \) are independent of \( x_i's \) and have finite variance \( \sigma^2 \). Let \( \hat{w} \) be the ordinary least squares estimator, i.e.,

\[
\hat{w} = \text{argmin}_w \| Xw - y \|_2^2.
\]

Then \( \sqrt{n}(\hat{w} - \mu w) \) is asymptotically normal with mean 0 and covariance matrix

\[
\sigma^2 \Sigma^{-1} + \Sigma^{-1} E\{ h(x)^2 x x^T \} \Sigma^{-1},
\]

where

\[
h(x) = g(x^T w) - \mu x^T w - \gamma,
\]

\[
\gamma = E\{ g(x^T w) - \mu x^T w \},
\]

\[
\mu = \text{Cov}\{ g(x^T w), x^T w \} / \text{Var}\{ x^T w \}.
\]

Furthermore, if \( w \) is scaled properly such that \( \| \sqrt{\Sigma} w \| = 1 \), then \( \mu = E[tg(t)], \ t \sim \mathcal{N}(0, 1) \).

This implies the following result.

**Corollary 2.1.** For the ordinary least squares with identity mapping function \( g \), \( \mu = E(t^2) = 1 \) with \( t \sim \mathcal{N}(0, 1) \) and \( h(x) = 0 \), we have the classical asymptotic estimation on least squares solution

\[
\sqrt{n}(\hat{w} - w) \to_d \mathcal{N}(0, \sigma^2 \Sigma^{-1}),
\]

where ‘\( \to_d \)’ means convergence in distribution.
In order to be able to select the right features, the intuition is that the irrelevant features cannot be highly correlated to the relevant features. The following definition is the same as the strong irrepresentable condition proposed by [95] which describes this property quantitatively. To be consistent with the framework in this paper, we call this strong $\mu-$irrepresentable condition:

**Definition 4.** (Strong $\mu-$irrepresentable condition). We say that the strong $\mu-$irrepresentable condition holds, if there exists a constant $s \in (0, 1]$, such that

$$|\Sigma_{zr}(\Sigma_{rr})^{-1}\text{sign}(\mu w)| \leq 1 - s.$$ 

Here $\Sigma_{zr}$ and $\Sigma_{rr}$ are the sub covariance matrices of $x_i's$ defined in Section 8.1 and the “$\leq$" holds element-wise. Here the $\text{sign}(\cdot)$ denotes an element-wise application of the standard sign function.

The following proposition [91] [Section 4] indicates that after appropriate scaling, even though the discrepancy $y - \mu X w$ may generally depend on both $X$ and $w$, the projection of it onto $X$ is well behaved in the sense that the expectation of the projection is zero.

**Proposition 3.** [91] Let $\tilde{X} = XQ^{-1}$, $\tilde{z}_i = g(\tilde{x}_i^T w) - \bar{\mu} \tilde{x}_i^T w$, $i = 1, 2, ..., n$ where $Q^T Q = \Sigma$, $\bar{\mu} = \text{Cov}(g(\tilde{x}_i^T w), \tilde{x}_i^T w)/\text{Var}(\tilde{x}_i^T w)$, $\tilde{x}_i$ is the $i$th row of $\tilde{X}$, then

$$E(\tilde{X}^T \tilde{z}) = 0.$$

**Proof.** It’s sufficient to show for any $v \in \mathbb{R}^p$ and $v \neq 0$, $E(v^T \tilde{X}^T \tilde{z}) = 0$. Without loss of
generality, we assume \( \|v\|_2 = \|w\|_2 = 1 \),

\[
E(v^T \tilde{X}^T z) = E(v^T (\sum_{i=1}^{n} \tilde{z}_i \tilde{x}_i))
\]

\[
= \sum_{i=1}^{n} (E(g(\tilde{x}_i^T w) v^T \tilde{x}_i) - E(\tilde{\mu} v^T M^{-T} x_i x_i^T M^{-1} w))
\]

\[
= \sum_{i=1}^{n} E(g(\tilde{x}_i^T w) v^T \tilde{x}_i) - n\tilde{\mu} v^T w
\]

The second term in the last equation comes from the fact that \( x_i \sim \mathcal{N}(0, \Sigma) \) and \( M^T M = \Sigma \). Note that \( t_i := \tilde{x}_i^T w \) has the distribution of \( \mathcal{N}(0, 1) \), using decomposition \( v^T \tilde{x}_i = v^T w t_i + t_i^\perp \), with \( t_i^\perp \) independent of \( t_i \), the first term becomes:

\[
\sum_{i=1}^{n} E(g(\tilde{x}_i^T w) v^T \tilde{x}_i) = \sum_{i=1}^{n} E\{g(t_i)(v^T w t_i + t_i^\perp)\}
\]

\[
= \sum_{i=1}^{n} v^T w E(g(t_i)t_i)
\]

\[
= n\tilde{\mu} v^T w
\]

for any \( v \). The last equation comes from the fact that \( E(g(\tilde{x}_i^T w) \tilde{x}_i^T w) = \tilde{\mu} Var(\tilde{x}_i^T w) = \tilde{\mu} \).

Therefore \( E(\tilde{X}^T \tilde{z}) = 0 \). \( \square \)

Note that in the linear setting when \( y = Xw + \epsilon \), \( z \) becomes \( \epsilon \) which is independent of the columns of \( X \), and thus proposition 3 holds naturally. It can be verified that the proposition fails to hold without the transformation on \( X \).

Note that since we require \( E(y^4) < \infty \), we will be able to use a bootstrap strategy to estimate the covariance matrix by re-sampling the data \( x_i's \) and thus make confidence intervals on \( \hat{w} \) for large enough \( n \).
8.3 Primal Dual Witness Method

In this section\(^1\), we describe an important result due to Wainwright [98], who introduced a proof technique called the primal-dual witness for the case when mapping function \(g\) is linear and known. In Chapter 9, we use this technique to show our main results on the consistency study of the Lasso under unknown nonlinear targets.

Here we assume a standard linear model,

\[ y = Xw + \epsilon, \quad \epsilon \in \mathcal{N}(0, \sigma^2 I) \]

for fixed \(X\) and \(w\), with the scaling \(\|X_j\|^2 \leq n\). Continuing the previous setting, let \(r\) be the true support set. Let \(X_r\) be the matrix formed by first \(q\) columns of \(X\) corresponding to the nonzero entries in \(w\). We assume that \(X_r\) has full column rank, in other words, \(X_r^TX_r\) is invertible. Our goal is to show that, for some value of \(\lambda\), the Lasso solution \(\hat{w}\) has an signed active set that exactly equals the true signed support set, \(\text{sign}\{\hat{w}\} = \text{sign} w\) with high probability. The primal-dual witness method plugs the true support \(r\) into the Karush-Kuhn-Tucker (KKT) conditions [141] for the Lasso and checks whether they can be verified.

First of all, the KKT conditions for the Lasso imply that any solution \(\hat{w}\) must satisfy

\[ X^T(y - X\hat{w}) = \lambda s \quad (8.2) \]

\(^1\)This section is based on [98] and http://www.stat.cmu.edu/~larry/=sml/sparsity.pdf
where \( s \in \partial \|\hat{w}\|_1 \), a subgradient of the \( l_1 \)-norm evaluated at \( \hat{w} \), which is

\[
s_j = \begin{cases} 
+1 & \hat{w}_j > 0 \\
-1 & \hat{w}_j < 0, \ j = 1, ..., n \\
[-1, 1] & \hat{w}_j = 0
\end{cases}
\]

We start by breaking equation eq. (8.2) into two blocks, over \( r \) and \( z \). We then get

\[
X_r^T(y - X_rw_r) = \lambda s_r \tag{8.3}
\]

\[
X_z^T(y - X_rw_r) = \lambda s_z \tag{8.4}
\]

if we can satisfy the two conditions (8.3) and (8.4) with a proper subgradient \( s \), such that

\[
s_r = \text{sign}(w_r) \quad \|s_z\|_\infty = \max_{i \in r} |s_i| < 1,
\]

then we have met our goal: we have recovered a unique Lasso solution whose active set is \( r \), and whose active signs are \( \text{sign}(w_r) \) (since this exactly meets the KKT condition, the uniqueness comes from the full rank assumption on \( X_r \)).

Now, solving \( \hat{w}_r \) yields

\[
\hat{w}_r = (X_r^TX_r)^{-1}(X_r^Ty - \lambda \text{sign}(w_r)) \tag{8.5}
\]

where we have substituted \( s_r = \text{sign}(w_r) \). From eq. (8.4), this implies that \( s_z \) must satisfy

\[
s_z = \frac{1}{\lambda} X_z^T(I - X_r(X_r^TX_r)^{-1}X_r^T)y + X_z^T(X_r^TX_r)^{-1}\text{sign}(w_r). \tag{8.6}
\]
For concreteness, the primal-dual witness method consists of the following steps:

1. First, we obtain $\hat{w}_r$ by solving the restricted (over set $r$) Lasso problem to obtain eq. (8.3). We set $\hat{w}_z = 0$.

2. Second, we solve for the subgradient over the $z$ parts, $s_z$ as in eq. (8.4).

3. At last, we check whether the sign consistency condition $\text{sign}(\hat{w}_r) = \text{sign}(w_r)$ holds, and that $\|s_z\|_\infty < 1$. If these two conditions pass, then we have verified the existence of a unique Lasso solution that exactly recovers the true support and sign.

To be clear, this procedure is not a practical method for solving the Lasso problem, since solving equations (8.3) and (8.4) requires knowledge of the unknown support set $r$. Rather, the utility of this constructive procedure is as a proof technique: it succeeds if and only if the Lasso has a unique optimal solution with the correct signed support. In next chapter, we utilize the primal dual witness technique to show our main results.
Chapter 9

Selection Consistency of the Lasso

In this chapter, we show our main contribution to the theoretical study of the sparsity. That is, we prove a new theorem on the selection consistency of the Lasso for feature selection with misspecified models. Our theorem is shown in Theorem 5. At the end, we show comprehensive numerical experiments to demonstrate our results.

9.1 Main Result

With the above preparation, given our prior assumptions in Section 8.1, we consider the solution \( \hat{w}^{(n)} \) to the generalized Lasso, where in this thesis we assume \( f(w) \) is either \( \| \cdot \|_1 \) for the classical Lasso or \( \| \cdot \|_{1,2} \) for the group Lasso:

\[
\hat{w}^{(n)} = \arg\min_w \left\{ \|Xw - y\|^2 + \lambda^{(n)} f(w) \right\},
\]

(9.1)

where \( y_i = g(x_i^T w) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2) \), \( X \) is the data matrix with independent and identically distributed rows and the \( i \)th row \( x_i^T \) satisfies \( x_i \sim \mathcal{N}(0, \Sigma), i = 1, 2, ..., n \).
First, consider the case \( f(w) = \|w\|_1 \). Let \( X_r \) be the first \( q \) columns of \( X \) corresponding to the nonzero entries in \( w \) and \( X_z \) be the remaining \( p - q \) columns. Then the following probability event hold:

**Proposition 4.** Assume that the strong \( \mu \)-irrepresentable condition holds for some constant \( s > 0 \). Then for large enough \( n \), we have

\[
P(\text{sign}(\hat{w}) = \text{sign}(\mu w)) \geq P(\Omega_1 \cap \Omega_2),
\]

with

\[
\Omega_1 = \{ |\Sigma^{-1} r^{-1} D_r| < \sqrt{n}(|\mu w_r| - \frac{\lambda(n)}{2n} \Sigma^{-1} r^{-1} \text{sign}(\mu w_r)) \},
\]
\[
\Omega_2 = \{ |\Sigma z^{-1} r^{-1} D_r - D_z| \leq \frac{\lambda(n)}{2\sqrt{n} s} \},
\]

where

\[
D_r = X_r^T (y - \mu X w) / \sqrt{n},
\]
\[
D_z = X_z^T (y - \mu X w) / \sqrt{n},
\]

and \( \mu \) is a constant defined in previous chapter.

Here \( \Omega_1 \) and \( \Omega_2 \) come directly from the first order optimality conditions of (9.1).

**Proof.** We will use the method proposed by [95] in random setting, which is shown as primal-dual witness (PDW) construction method in [98]. Consider the Lasso selection model:

\[
\hat{w}^{(n)} = \arg\min_w \left\{ \|X w - y\|^2 + \lambda_n \|w\|_1 \right\},
\]

(1)
where \( y = (y_1, y_2, \ldots, y_n)^T \), \( \epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^T \), \( y_i = g(x_i^T w) + \epsilon_i \), \( \epsilon_i \sim \mathcal{N}(0, \sigma^2) \), \( x_i \sim \mathcal{N}(0, \Sigma) \). Define the mapping function \( G : \mathbb{R}^n \to \mathbb{R} \) as a vector copy of \( g \), that is, \( G(w) := (g(w_1), g(w_2), \ldots, g(w_n))^T \).

Let \( \hat{w}^n = \hat{w}^{(n)} - \mu w \), to make the proof “clean” and without introducing more confusion, we drop the super script \( n \) of \( \hat{w}^{(n)} \).

Define \( V(u) = \|G(Xw) + \epsilon - \mu Xw - Xu\|^2 + \lambda_n \|u + \mu w\|_1 \). \( V \) is a convex function of \( u \).

It can be verified that \( \hat{u} \) is the minimizer of \( \min_u V(u) \). This is because

\[
V(\hat{u}) = \|G(Xw) + \epsilon - \mu Xw - X\hat{u}\|^2 + \lambda_n \|\hat{u} + \mu w\|_1 \\
= \|G(Xw) + \epsilon - \mu Xw - X(\hat{w} - \mu w)\|^2 + \lambda_n \|\hat{w} - \mu w + \mu w\|_1 \\
= \|G(Xw) + \epsilon - X\hat{w}\|^2 + \lambda_n \|\hat{w}\|_1 \\
= \|y - X\hat{w}\|^2 + \lambda_n \|\hat{w}\|_1. \quad (\text{Since } y_i = g(x_i^T w) + \epsilon.)
\]

By definition, the quantity at the last line of the equation has a minimum value with \( \hat{w} \).

Therefore \( \hat{u} \) is the minimizer of \( \min_u V(u) \).

Define \( z := y - \mu Xw = G(Xw) + \epsilon - \mu Xw \), then

\[
V(u) = \|G(Xw) + \epsilon - \mu Xw - Xu\|^2 + \lambda_n \|u + \mu w\|_1 \\
= \|z - Xu\|^2 + \lambda_n \|u + \mu w\|_1 \\
= u^TX^TXu - 2u^TX^Tz + z^Tz + \lambda_n \|u + \mu w\|_1.
\]

Since \( \hat{u} \) is the minimizer of \( V(u) \), to ensure the existence of \( \hat{u} \), we have

\[
0 \in \partial V(\hat{u}).
\]
Here $\partial V(\hat{u})$ is the set of sub-differentials of $V(u)$ at $\hat{u}$, where

$$\partial V(\hat{u}) = 2X^T X \hat{u} - 2X^T z + \lambda_n \partial \| \hat{u} + \mu w \|_1$$

with

$$\partial \| \hat{u} + \mu w \|_1 = \begin{cases} 
1 & \text{if } u + \mu w > 0, \\
-1 & \text{if } u + \mu w < 0, \\
[-1, 1] & \text{if } u + \mu w = 0.
\end{cases}$$

Our goal is to derive $\text{sign}(\hat{w}) = \text{sign}(\mu w)$, thus we require the deviation $|\hat{w} - \mu w|$ not to be too big, that is, $|\hat{w} - \mu w| < |\mu w|$. One can verify that if $|\hat{u}| = |\hat{w} - \mu w| < |\mu w|$, then $\text{sign}(\hat{w}) = \text{sign}(\mu w)$, that is $\text{sign}(\hat{u} + \mu w) = \text{sign}(\mu w)$.

Putting this and (2) together, we need

$$\begin{cases} 
0 \in \partial V(\hat{u}) = 2X^T X \hat{u} - 2X^T z + \lambda_n \partial \| \hat{u} + \mu w \|_1 \\
|\hat{u}| < |\mu w| & w \neq 0
\end{cases}$$

where

$$\partial \| \hat{u} + \mu w \|_1 = \begin{cases} 
\text{sign}(\mu w) & \text{if } w \neq 0, \\
[-1, 1] & \text{if } w = 0.
\end{cases}$$

Let $D = X^T z / \sqrt{n}$. By definition the covariance matrix of $x_i$s is $\Sigma = \text{cov}(X)$. According to the classical covariance matrix estimation theory, when $n$ is big enough, and interpreting the columns of the matrix $X$ as realizations of an underlying random variable, $\frac{1}{n} X^T X = \Sigma + O(\frac{1}{\sqrt{n}})$, that is, for fixed length of $w$, the covariance of $X$ converges in the order of $O(\frac{1}{\sqrt{n}})$. Assume $w_r$ contains the nonzeros entries of $w$, $w_z$ contains the zero entries, then
$D_r \in \mathbb{R}^q$, $D_z \in \mathbb{R}^{p-q}$, $\hat{u}_r \in \mathbb{R}^q$ and $\hat{u}_z \in \mathbb{R}^{p-q}$ are well defined, respectively. Therefore

$$V(u) = (\sqrt{n}u)^T(\Sigma + O(\frac{1}{\sqrt{n}}))(\sqrt{n}u) - 2(\sqrt{n}u)^TD$$

$$+ z^Tz + \lambda_n\|u + \mu w\|_1.$$  

We then have

$$\begin{cases} 
  n(\Sigma_{rr} + O(\frac{1}{\sqrt{n}}))\hat{u}_r - 2\sqrt{n}D_r = -\frac{\lambda_n}{2} \text{sign}(\mu w), \\
  |\hat{u}_r| < |\mu w|, \\
  n\Sigma_{zr}\hat{u}_r - 2\sqrt{n}D_z \in \lambda_n[-1, 1].
\end{cases}$$

These are sufficient conditions to guarantee the sign consistency. Under the assumption that $\Sigma$ is well conditioned, for large enough $n$, the subgradient condition is satisfied if the following events hold for large enough $n$:

$$|\Sigma^{-1}_{rr}D_r| < \sqrt{n}(|\mu w_r| - \frac{\lambda_n}{2n}|\Sigma^{-1}_{rr}\text{sign}(\mu w_r)|),$$

$$|\Sigma_{zr}\Sigma^{-1}_{rr}D_r - D_z| \leq \frac{\lambda_n}{2\sqrt{n}}(1 - |\Sigma_{zr}\Sigma^{-1}_{rr}\text{sign}(\mu w_r)|).$$

$(\Omega_1)$ and $(\Omega_2)$ are therefore sufficient conditions to get the $\mu-$sign consistency with probability 1. 

After combining this proposition with the Chebyshev’s inequality, we are able to establish the selection consistency of the Lasso. In other words, we will be able to select significant features with high probability. The following new theorem is our main contribution to the theoretical study of the sparsity:

**Theorem 5.** ($\mu-$sign consistency) Under the assumptions in Section 8.1 and the strong
\( \mu \)-irrepresentable condition, if \( \lambda^{(n)} \) is chosen such that \( \lambda^{(n)} \sim n^{k_3} \), with some constant \( k_3 \) satisfying
\[
\max \{ \frac{1+k_1}{2}, \frac{1+k_2}{2} \} < k_3 < 1, 0 \leq k_1 < 1, 0 \leq k_2 \leq 1, \text{ we have}
\]
\[
P(\text{sign}(\hat{w}) = \text{sign}(\mu w)) \geq 1 - O(e^{-n^{k_1}}) - O\left( \frac{1}{n^{2k_3 - 1 - k_2 s^2}} \right).
\]

Here \( k_2 \) depends on the choice of \( g \). In particular, \( k_2 = 0 \) when \( g \) is linear.

**Proof.** First, according to proposition 4, we have
\[
P(\text{sign}(\hat{w}) = \text{sign}(\mu w)) \geq P(\Omega_1 \cap \Omega_2).
\]

Since the first \( q \) entries are nonzeros, we have
\[
1 - P(\Omega_1 \cap \Omega_2) \leq P(\Omega_1^c) + P(\Omega_2^c)
\]
\[
\leq \sum_{i=1}^{q} P(|\alpha_i| \geq \sqrt{n}(|\mu w_r| - \frac{\lambda^{(n)}}{2n} |\Sigma_r^{-1} \text{sign}(\mu w_r)|) + \sum_{i=q+1}^{p} P(|\beta_i| \geq \frac{\lambda^{(n)}}{2\sqrt{n}} (1 - |\Sigma_{zr} \Sigma_{rr}^{-1} \text{sign}(\mu w_r)|),
\]

where \( \alpha = \Sigma_{rr}^{-1} D_r \) and \( \beta = \Sigma_{zr} \Sigma_{rr}^{-1} D_r - D_z \).

First we analyze the distribution of \( \alpha \). By definition,
\[
D_r = X_r^T(y - \mu X w)/\sqrt{n}, \quad D_z = X_z^T(y - \mu X w)/\sqrt{n}.
\]
Furthermore, as \( w = [w_r, w_z], w_z = 0 \), one can verify that \( Xw = X_rw_r \). Therefore,

\[
\alpha = \Sigma_{rr}^{-1}D_r = n(X_r^T X_r)^{-1}X_r^T (y - \mu X w)/\sqrt{n} \\
= \sqrt{n}(X_r^T X_r)^{-1}X_r^T (y - \mu X_r w_r) \\
= \sqrt{n}(X_r^T y - \mu w_r).
\]

Since \( X_r^T y \) is the least squares estimate coming from \( \min \| y - X_r w_r \|_2 = \| y - X w \|_2 \), by theorem 3.1, the least squares estimator has the following asymptotic behavior:

\[
\sqrt{n}(\hat{w} - \mu w) \sim \mathcal{N}(0, \sigma^2 \Sigma^{-1} + \Sigma^{-1} E\{h(x)^2 x x^T\} \Sigma^{-1}),
\]

where,

\[
h(x) = g(x^T w) - \mu x^T w - \gamma, \\
\gamma = E\{g(x^T w) - \mu x^T w\}, \\
\mu = \text{Cov}\{g(x^T w), x^T w\}/\text{Var}\{x^T w\}.
\]

Therefore \( \alpha \) behaves asymptotically as:

\[
\alpha \sim \mathcal{N}(0, \sigma^2 \Sigma_{rr}^{-1} + \Sigma_{rr}^{-1} E\{h(x_r)^2 x_r x_r^T\} \Sigma_{rr}^{-1}).
\]

By our assumption, the covariance matrix of \( \alpha \) is well defined, thus \( \alpha \) behaves as a Gaussian variable with mean 0 and bounded variance element-wise. The standard Gaussian tail estimation shows that if \( \lambda^{(n)}/n^{(1+k_1)/2} \to \infty \) and \( \lambda^{(n)}/n \to 0 \), for some constant \( 0 \leq k_1 < 1 \)
then

\[ \sum_{i=1}^{q} P(|\alpha_i| \geq \sqrt{n}(|\mu_{wr}| - \frac{\lambda(n)}{2n} |\Sigma_{rr}^{-1} \text{sign}(\mu_{wr})|)) = O(\exp(-n^{-k_1})). \]

Now we estimate \( \beta \):

\[
\beta = \Sigma_{zr} \Sigma_{rr}^{-1} D_r - D_z \\
= (\Sigma_{zr} \Sigma_{rr}^{-1} X_r^T - X_z^T)G(X_r,w_r) / \sqrt{n} \\
+ (\Sigma_{zr} \Sigma_{rr}^{-1} X_r^T - X_z^T)\epsilon / \sqrt{n} + O(1),
\]

where the mapping function \( G : \mathbb{R}^n \rightarrow \mathbb{R}^n \) is defined as a vector version of \( g \), that is, \( G(w) := (g(w_1), g(w_z), ..., g(w_n))^T \). The \( O(1) \) term comes from the empirical estimation rate of \( \Sigma \). As \( \epsilon \) is independent of \( X \), the second term of \( \beta \) has 0 mean and bounded variance element-wise. The first term of \( \beta \) characterizes the ‘linearity’ of function \( G(w) \): if \( G(w) \) is linear, then the first term vanishes to \( O(1) \). One should be aware that \( X_z \) is not independent of \( X_r \) and hence in general \( E(X_z^T G(X_r,w_r)) \neq 0 \). However, we will show that the expectation of \( \beta \) is still under control with our assumptions.

If \( \Sigma = I_{p \times p} \), then from proposition 3, \( E(X_r^T z) = 0 \). So we have:

\[
E(X_r^T z) = E \begin{pmatrix} X_r^T y \\ X_z^T y \end{pmatrix} = n\mu \begin{pmatrix} \Sigma_{rr} w_r \\ \Sigma_{zr} w_r \end{pmatrix} = 0.
\]

This leads to:

\[
E(\beta) = n\mu(\Sigma_{zr} \Sigma_{rr}^{-1} \Sigma_{rr} w_r - \Sigma_{zr} w_r) = 0.
\]
For more general $\Sigma$, we use the multivariate Stein’s lemma [142] [143] [Lemma 1]:

**Lemma 6.** Let $x = (x_1, ..., x_n)$ be multivariate normally distributed with mean vector $\mu$ and covariance matrix $\Sigma$. For any function $h(x_1, ..., x_n)$ such that $\partial h/\partial x_i$ exists almost everywhere and $E|\left(\frac{\partial}{\partial x_i} h(x)\right)| < \infty$, $i = 1, 2, ..., n$, the following fact holds:

$$\text{Cov}(x_1, h(x)) = \sum_{i=1}^{n} \text{Cov}(x_1, x_i) E\left(\frac{\partial}{\partial x_i} h(x)\right).$$

Using this lemma, the expectation of $\beta$:

$$E(\beta) = (\Sigma_{zr} \Sigma_{rr}^{-1} E(X_r^T G(X_r w_r)) -$$

$$E(X_r^T G(X_r w_r)))/\sqrt{n} + O(1)$$

$$= (\Sigma_{zr} w_r E(\sum_{j=1}^{n} g'(\sum_{i=1}^{q} w_{ij} x_{ji})) -$$

$$\Sigma_{zr} w_r E(\sum_{j=1}^{n} g'(\sum_{i=1}^{q} w_{ij} x_{ji}))/\sqrt{n} + O(1)$$

$$= O(1).$$

Finally, in order to apply Chebyshev’s inequality, we estimate the variance of $\beta$. First notice that by the strong $\mu$—irrepresentable condition, $X_r w_r$ follows $\mathcal{N}(0, q^2 w_r^T \Sigma_{rr} w_r)$ distribution element-wise. Thus each element of $X_r w_r$ is bounded between $[-4q^2 \Lambda_{\max}, 4q^2 \Lambda_{\max}]$ with probability 1. Based on our assumption, $g(w)$ is differentiable almost everywhere and $E(|g'(w)|) < \infty$. Expanding

115
the product of $X_r^T G(X_r w_r)$ leads to

$$\text{Var}(X_r^T G(X_r w_r) / \sqrt{n}) < c(g, \Lambda(\Sigma_{rr}), p, q) n,$$

where $c(g, \Lambda(\Sigma_{rr}), p, q)$ is a constant that depends on $g, \Lambda(\Sigma_{rr}), p$ and $q$. The same argument can be applied to $X_z^T G(X_r w_r)$. The second term of $\beta$ can be derived from a classical result [94]:

$$\text{Var}((\Sigma^{-1} r X_r - X_z^T \epsilon) / \sqrt{n}) = \Sigma_{zz} - \Sigma_{zr} \Sigma_{rr}^{-1} \Sigma_{rz}.$$

We get $\text{Var}(\beta) = O(n).$ Indeed, this is the worst case analysis. For general functions that satisfy our assumptions, $\text{Var}(\beta)$ can be smaller. More precisely, let $\text{Var}(\beta) = \Theta(n^{k_2})$, for some constant $0 \leq k_2 \leq 1$.

Finally, using Chebyshev’s inequality and by choosing $\lambda^{(n)}$ such that $\lambda^{(n)} \sim n^{k_3}$ for some constant $k_3$ such that $\max\{k_1 + 1, k_2 + 1\} < k_3 < 1$ and $0 \leq k_1 < 1, k_2$ is a constant depending on $\text{Var}(\beta)$, we get

$$\sum_{i=q+1}^{p} P(|\beta_i| \geq \frac{\lambda^{(n)}}{2 \sqrt{n}} (1 - |\Sigma_{zr} (\Sigma_{rr}^{-1}) \text{sign}(\mu w_r)|)) = O(\frac{4n \text{Var}(\beta)}{(\lambda^{(n)})^2 s^2}) = O(\frac{1}{n^{2k_3 - 1 - k_2 s^2}}).$$

Combining the results of $\alpha$ and $\beta$ completes the proof.

What this theorem shows is that, by maintaining regularization parameter $\lambda^{(n)}$ at a certain level, the sign of the solution of model (9.1) will be the same as the sign of a constant times the true solution with high probability when $n$ is large enough. The constant $k_2$ depends on the nonlinearity of the link function $g$. The theorem matches the intuition
that in order to select the features with high probability, the worse the \( g \) is (\( k_2 \) is closer to 1), the more regularization we need (a larger \( k_3 \) so that the last big-O term vanishes at a certain rate).

The practical use of this theorem is that, given enough data, by proper sampling, one can estimate the sign of \( \mu \) by its definition using an empirical procedure. This can verify whether \( \hat{w} \) and \( w \) have same or opposite signs.

By applying theorem 5, we can derive the classical convergence rate for the probability of sparse linear selection:

**Corollary 6.1.** When \( g(w) \) is linear, we have \( \mu = 1 \) and \( k_2 = 0 \). By choosing \( \lambda \sim n^{k_3} \), for some \( \frac{k_1 + 1}{2} \leq k_3 < 1, 0 \leq k_1 < 1 \), the probability of successful selection is:

\[
P(\text{sign}(\hat{w}) = \text{sign}(w)) \geq 1 - O(e^{-nk_1}) - O\left(\frac{1}{n^{2k_3-1}s^2}\right).
\]

The classical results [95, 96, 98] show that in the linear setting, we have \( P(\text{sign}(\hat{w}) = \text{sign}(w)) = 1 - O(e^{-nc}) \) for some constant \( c > 0 \). Thus our lower bound result is consistent with the classical version. By further looking into the details of the proof, one can verify that the difference comes from the different inequalities used: Gaussian tail inequality (in classical analysis) and Chebyshev’s inequality (in our setting). The former achieves an exponential decay and thus the two big-Os are combined.

**Extension to Group Lasso.** The following parts of this section shows that our study matches the previous results on the group Lasso [144]. That is, the group Lasso can also achieve consistent selection with probability 1 when the size of the data \( n \to \infty \). However, the results (both the previous work and ours) are limited in a way that the convergence rate is not known. To our best knowledge, the precise probability bound for the group Lasso is still an open problem.
A natural extension of the Lasso is the group Lasso, where the regularization \( f(x) \) becomes \( \| \cdot \|_{1,2} \). Given data \( X = (X_1, X_2, \ldots, X_m) \), each \( X_j \in \mathbb{R}^{n \times p_j} \) represents data in group \( j \). A formal definition of the group Lasso corresponds to:

\[
\min_{w_I} \left\| \sum_{j=1}^{m} X_j w_{I_j} - y \right\|^2 + \lambda(n) \sum_{j=1}^{m} d_j \|w_{I_j}\|
\]

Here \( d_j > 0 \) is a fixed weight for each group. In our setting, we consider \( y_i = g(\sum_{j=1}^{m} X_{ij} w_{I_j}) + \epsilon_i, \epsilon_i \sim \mathcal{N}(0, \sigma^2) \), \( X_j \sim \mathcal{N}(0, \Sigma_j), i = 1, \ldots, n, j = 1, \ldots, m \). For any minimizer of \( E(Xw - y)^2 \), we assume that \( E((Xw - y)^2|X) \) is almost surely greater than some constant \( c > 0 \). We will further assume that \( w \) is normalized with \( \Sigma_j \) such that \( \| \Sigma_j w_j \|_2 = 1 \), for \( j = 1, 2, \ldots, m \). Similar to the Lasso, we have the strong irrepresentable condition for the group Lasso [144]:

**Definition 5.** (Strong irrepresentable condition for the Group Lasso) The group Lasso is said to satisfy the strong irrepresentable condition if there exists \( s \in (0, 1) \), such that

\[
\max_{i \in I_J} \frac{1}{d_j} \left\| \Sigma_{X_iX_j} \Sigma_{X_j}^{-1} X_j \right\| \text{Diag}(d_j/\|w_{I_j}\|)w_{I_j}\| \leq 1 - s.
\]

Here \( \text{Diag}(d_j/\|w_{I_j}\|) \) denotes a block diagonal matrix with each block \( d_j/\|w_{I_j}\| I_p, I_j \) denotes the index set of all the groups with all \( w_j \) nonzeros, \( I_j^c \) denotes its complement. [144] shows that with properly chosen \( \lambda(n) \), one gets successful group selection with probability tending to 1 with large enough \( n \). The following corollary continues this in general:

**Corollary 6.2.** If the strong irrepresentability condition for the group Lasso holds, and if \( \lambda(n) \sim n^{k_3} \) such that \( \max\{\frac{1+k_1}{2}, \frac{1+k_2}{2}\} < k_3 < 1, 0 \leq k_1 < 1, 0 \leq k_2 \leq 1 \), with \( k_2 \) depending on the choice of \( g \), the probability of successful group selection: \( P(\text{sign}(\hat{w}) = \text{sign}(\mu w)) \to 1 \), with \( n \) large enough.
The proof is similar to that in [144] in which the author does not assume any regression relation between $X$ and $y$ which naturally suits our setting. Combining with theorem 2, we get the desired result. We first need the following lemma.

**Lemma 7.** [144] Let $\hat{w}_{I_j}$ be any minimizer of

$$\| \sum_{j=1}^{m} X_j w_{I_j} - y \|^2 + \lambda_n \sum_{j=1}^{m} d_j \| w_{I_j} \|.$$  

(9.2)

If $\lambda_n/n \to 0$, then $\hat{w}_{I_j}$ converges to the global minimizer in probability.

**Proof.** Since averaging the target function does not have impact on the optimal solution, it is sufficient to show that the lemma holds for

$$\frac{1}{n} \| \sum_{j=1}^{m} X_j w_{I_j} - y \|^2 + \lambda_n \sum_{j=1}^{m} d_j \| w_{I_j} \|.$$  

When $\lambda_n/n \to 0$, the solution $\hat{w}_{I_j}$ converges to the unique minimizer of the loss function $L(w_{I_j}) = \Sigma y^y - 2 \Sigma y X_j w_{I_j} + w_{I_j}^T \Sigma X_j X_j w_{I_j}$. As $\Sigma X_j X_j$ is positive definite, the least squares estimator $\hat{w}_{I_j}$ converges to the global minimizer with probability [94].

The following lemma implies that under our assumptions, the global minimizer of the group Lasso is indeed $\mu w_{I_j}$, [91, 105]:

**Lemma 8.** Let $\bar{w}$ be the ground truth weight vector. Define the loss function $L(w) = \frac{1}{n} (\| X w - y \|_2^2 - \| A \mu \bar{w} - y \|_2^2)$. Then we have

$$E(L(w)) = \| w - \mu \bar{w} \|_2^2$$

Notice that the second term in $L(w)$ does not affect the minimizer of the original least square loss $\frac{1}{n} \| X w - y \|_2^2$. Denote the truth weights in the group Lasso model as $\bar{w}_{I_j}$,
$j = 1, 2, \ldots, m$, combining lemma 7 and lemma 8, we obtain that the global minimizer of the group Lasso is $\mu \tilde{w}_{IJ}$. Hence $\hat{w}_{IJ}$ is estimation consistent. We show it’s selection consistent with probability tending to 1 with $n$ large enough.

It’s a standard result of [144] that, for the group Lasso, the irrepresentable condition implies the selection consistency between $\hat{w}_{IJ}$ and the global minimizer of eq. (9.2), which we have shown to be $\mu \tilde{w}_{IJ}$ in our setting. Thus we get the desired result.

9.2 Numerical Experiments

In this section, we present some numerical studies exploring the theoretical results described above. We describe experiments both to verify and illustrate our theoretical results as well as to test some of the assumptions we make.

Our setup for these studies is as follows. We generate data using several nonlinear targets and then solve a least squares problem with a linear hypothesis extended with a Lasso term:

Data: $y_i = g(x_i^T w) + \epsilon_i$,

Model: $\hat{w} = \arg\min_w \left\{ \|Xw - y\|_2^2 + \lambda \|w\|_1 \right\}$.

The data matrix $X$ is generated in a way that each row $x_i \sim \mathcal{N}(0, \Sigma)$. Here $\Sigma$ is either an identity matrix $I$ or has power-decay entries $\Sigma_{ij} = \rho^{|i-j|}$, $0 < \rho < 1$. We denote the latter as $\Sigma(\rho)$. Both of these choices satisfy the strong irrepresentability condition. Other types of covariance matrices that satisfy this condition are given in previous work [95]. The noise $\epsilon \sim \mathcal{N}(0, 0.04I)$ and $w \in \mathbb{R}^{100}$ with the first 10 entries nonzeros. The optimization problem is solved using the ADMM algorithm [89].
To make the experiments comprehensive, we select a variety of functions as well as one negative example for which the link function fails to satisfy our assumptions:

Polynomials: \( g_1(u) = u^m + u^{m-1} + \ldots + 1 \) \( (m > 2) \),

Sine: \( g_2(u) = \sin(mu) + \ldots + \sin(u) + 1 \),

Mixed: \( g_3(u) = \cos(u) + \frac{1}{1 + e^{-u}} - u^3 - 2 \),

False: \( g_4(u) = 1/u \),

Test: \( g_5(u) = (u + 1)^2 - 2u + \exp(u) + 1 \).

Figure 9.1 shows two examples illustrating the ‘scaling’ phenomenon: the solution of the selection model \( \hat{w} \) is close to the underlying truth \( w \) up to a scaling factor \( \mu \). In this figure, we plot the coefficients for \( g_1(u) \) with \( m = 3 \) (left) and \( g_3(u) \) (right). The training sample has 2000 examples generated with \( \Sigma = \Sigma(0.8) \). The regularization parameter \( \lambda^{(n)} \) is set to 0.8 during optimization. As the figure shows, the found solution \( \hat{w} \) is linearly related to \( w \), though the sign may be different. The left panel and the right panel corresponds to positive and negative \( \mu \) respectively \( (\mu_1 \approx 3.73 \) (left) , \( \mu_2 \approx -2.53 \) (right)).

We next verify our main result. In Figure 9.2 we plot the probability of a relevant feature being successfully selected versus the sample size \( n \) for different link functions \( g \) and covariance matrices \( \Sigma \). Each mark on the curves corresponds to an average over 200 trials using different random training samples.

To overcome the difficulty of not knowing the constants in theorem 5, we uniformly sample \( \lambda \) from \( n^{0.6} \) to \( n^{1.2} \). The reason this upper-bound exceeds \( n \) here is for experimental selection. As we only know \( \lambda = \Theta(n^k) \), \( k \in (\frac{1}{2}, 1) \), we use this setting to search for the unknown constant factor. The numerical results in Figure 9.2 are consistent with our theoretical results. The Lasso model is able to select the right features with a large enough
sample size even for highly nonlinear functions such as polynomials and sine functions. Even though \( w \in \mathbb{R}^{100} \), Figure 9.2 also indicates that 2400 to 4000 of samples are sufficient to achieve successful selection almost surely while in the linear case this reduces to only 200 to 500. Thus under the right conditions we may expect the Lasso to be very successful at selecting relevant features even with modest sample sizes.

Our theorem indicates that a consistent successful selection probability can be achieved by setting \( \lambda \) in the order of \( n^k \), where \( k \) is a constant depending on the noise. Figure 9.3 shows the numerical demonstration of this statement. The weight vector \( w \in \mathbb{R}^{100} \) has been normalized as \( \|w\|_2 = 1 \). The y-axis is obtained from \( \log(\lambda)/\log(n) \) which is equivalent to \( k \) with a shifting constant. The black pixels indicates successful selections with probability over 95%. This figure indicates that we can consistently choose \( \lambda \) as \( n^k \) with \( k \) as a constant to achieve satisfying selection performance.
Figure 9.3: Numerical tests on the relation between power constant $k$ and the sample size $n$ with fixed amount of noise $\epsilon$. Left: $g_1(u)$ with $m = 3$, $\epsilon \sim \mathcal{N}(0, 2I)$. Right: $g_1(u)$ with $m = 5$, $\epsilon \sim \mathcal{N}(0, I)$.

Figure 9.4 shows the results of consistency tests for the group Lasso. For simplicity, we only show this for the polynomial families. In this test, $w \in \mathbb{R}^{80}$ has eight groups of ten. The last four groups are set to be zeros. The $w$ in the first four groups are randomly chosen to be $\pm 1$. The results are consistent with our theory. Note that a relative large sample size may be required for successful group selection.
Figure 9.4: Numerical simulations on the probability of successful group selection with different sample size $n$ and $\Sigma$ for polynomial families. The first row are simulations on $g_1(w)$, $m = 1$ corresponds to the linear mapping function. The first column has $\Sigma = I$, second column has $\Sigma = \Sigma(0.2)$.

Next, we study the necessity of some of our assumptions. Throughout the paper we have used the assumption that the data follows a Gaussian distribution, following related work [91]. To get more insight into the Lasso as well as the necessity of this assumption, we run numerical tests on data $X$ generated from a variety of distributions, including the uniform distribution, Beta distribution, Weibull distribution, and student’s t-distribution. We use a test function $g_5$, a randomly chosen function that does not have any special patterns. Figure 9.5 shows the test results. We also test our assumptions on the link function from Section 8.1 on the target $g_4$, which fails to satisfy our expectation assumptions on the link function. In particular, $E(g_4'(t))$ does not exist when $t \sim \mathcal{N}(0, 1)$.

From Figure 9.5 we observe the following. First, feature selection from $g_4$, which violates our conditions, fails completely. Second, for the given function $g_5$, along with the Gaussian, the uniform distribution as well as the Weibull distribution with parameter
1 and 1.5 lead to successful feature selection. The former maintains thinner tails than
the Gaussian while the latter has heavier tails. However, the convergence rate of the
probability for Gaussian distribution outperforms the other two. Interestingly, two very
similar distributions show different behavior. The \( \text{Beta}(2, 2) \) distribution also has finite
support as the does the uniform distribution. The student’s t-distribution \( t(\nu = 1) \) maintains
heavier tails than the Gaussian. But both of them fail to achieve selection consistency.
These results indicate that there may be some room to relax the requirements we assume,
but not much. Understanding the gap between the necessary and sufficient conditions is a
direction for future work.
Figure 9.2: Numerical simulations on the probability of successful selection with different sample size $n$ and $\Sigma$. The first row are simulations on $g_1(w)$, $m = 1$ corresponds to the linear mapping function, second row corresponds to $g_2(w)$. The first column has $\Sigma = I$, second column has $\Sigma = \Sigma(0.2)$. 
Figure 9.5: Numerical tests on the necessity of assumptions. $U[-1,1]$ is uniform distribution on $[-1,1]$, $t(nv = 1)$ is t-distribution with freedom $\nu = 1$, $Beta(2,2)$ corresponds to Beta distribution with $\alpha = \beta = 2$, $Weibull(1,1.5)$ is Weibull distribution with scale 1, shape 1.5, $N(0,1)$ is standard normal for comparison. The “False Eg” line corresponds to tests on $g_4$ with data from Gaussian distribution $\Sigma = I$. 

127
Part V

Conclusions and Future Directions
Chapter 10

Conclusions and Future Work

In this thesis, we have studied three main problems in the context of sparse modeling, presenting novel models, algorithms, and theoretical analysis. In particular, we have established useful results for the following problems: image reconstruction, distributed dictionary learning and feature selection. While the approaches appear somewhat distinct, there is a unifying factor of sparsity (or a measure thereof) of the desired solutions, and we have developed key results and algorithms for these problems by leveraging some known and useful properties of the $l_1$-regularity. For each of these scenarios, we draw the conclusions and discuss the future work in the following sections.

10.1 Conclusions on PCM-TV-TFV

In Part II, we proposed a general two-stage PCM framework for image reconstruction from Fourier measurements. The projection step alleviates the challenges of reconstruction from the nonuniformness of the Fourier measurements, and the correction step further reduces
the noise and the bias effects in the previous step. We have shown that a precise edge
guided TV-TFV regularity possesses demonstrable advantages over the models with single
TV or TFV regularity. The numerical experiments demonstrate that such a combination
enhances the reconstruction of given data and obviates some drawbacks of the TV and
TFV regularizations. Furthermore, we have also shown that the proposed PCM-TV-TFV
has a superior performance even when the measurements have considerably high level of
noise.

A major key assumption of the PCM framework is that we have some prior information
of an appropriate basis for representing the underlying image. For example, wavelets might
be good choices for approximating natural images. Throughout the thesis we only use
Haar wavelets to demonstrate the advantages of the proposed two-stage PCM framework
in the numerical experiments. However, other bases such as shearlets, curvelets or adaptive
wavelets could also be incorporated to further improve the results. Moreover, the proposed
two-stage framework for Fourier measurements could also be extended to other linear
measurements such as the Radon transform.

10.2 Conclusions on Distributed Dictionary Learning Al-
gorithm

In Part III, we have reviewed the dictionary learning problem as well as the classical
centralized and distributive algorithms. To tackle the computational challenges with large
size of input data and address the limitations of the classical consensus approach, we
proposed a novel distributed dictionary learning algorithm for learning incoherent dictio-
naries with generalized consensus updates. Compared to the standard averaging scheme,
our algorithm automatically determines the desired selection matrix that can produce an
incoherent common dictionary. Since the master node only processes the dictionaries rather than the datasets, the proposed algorithm has negligible communication cost and is an efficient solution to practical dictionary learning problems. Furthermore, we conducted numerical experiments for both synthetic data and real world image processing tasks. The proposed algorithm achieves superior performance compared to the classical consensus algorithm.

While numerical experiments demonstrate the capacity of the proposed algorithm for dictionary learning, we do not have theoretical guarantees for convergence of the algorithm, and this sets up a future research work, where issues of convergence will be explored.

Throughout Part III we assume a fixed network topology where the all the local workers have access to the master node. There are certain circumstances where not all the local machines can access the master node. Furthermore, there might not even exist a master node. In these cases, the local workers can only exchange data with its neighbors. It is thus an interesting problem to adapt the proposed algorithm to these network topologies.

10.3 Conclusions on Feature Selection with the Lasso

In Part IV, we have shown the generalized selection consistency of the Lasso model when the observations are generated from some unknown link function that might be nonlinear. Under suitable assumptions, the Lasso model is still able to select the right features, but the weight is either dampened or amplified by an unknown constant. We have described the asymptotic probability behavior of the selection consistency of the Lasso solution and derived the classical consistency results as a special case.

There are some important open questions for future work, with most significant being the precise characterization of the necessary conditions to guarantee the selection con-
sistency in this nonlinear model. In [16, 97], the authors have shown the necessity of irrepre-
sentable condition when regression relation is known to be linear. Whether there ex-
ists other necessary conditions in this general setting remains unclear. It is also of great
interest to study the sufficient conditions under high dimensional settings, that is, when the
number of features is larger than that of samples. Furthermore, we have only done a few
numerical tests on the Gaussianity assumption. It remains interesting to check whether the
selection consistency still holds for other general distributions.
Appendix A

Projection on Sets

In this appendix, we show the mathematical formulations of the projection of a point \( x \) onto a convex set \( \mathcal{K} \) which is widely in general convex optimization. More details can be found in [145] and [146].

Sparse modeling leads to various models. For instance,

- **Lasso**:\^1

  \[
  \min_{x} \| x \|_1 \quad \text{subject to} \quad \| y - Ax \|_2 \leq \epsilon.
  \]

- **Dantzig selector:**

  \[
  \min_{x} \| x \|_1 \quad \text{subject to} \quad \| A^*(y - Ax) \|_{\infty} \leq \delta.
  \]

\^1Here we consider a variant of the Lasso model that is written as a constrained optimization problem.
• Wavelet and TV:

\[
\begin{align*}
\text{minimize} & \quad \|Wx\|_1 + \lambda \|x\|_{TV} \\
\text{subject to} & \quad \|y - Ax\|_2 \leq \delta.
\end{align*}
\]

• Low Rank:

\[
\begin{align*}
\text{minimize} & \quad \|X\|_* \\
\text{subject to} & \quad A(X) = y,
\end{align*}
\]

where \(\|\cdot\|_*\) is the nuclear norm defined by the summation of the singular values the matrix.

The so-called Conic form looks as follows:

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad A(x) + b \in \mathcal{K}
\end{align*}
\]

where \(f\) is a convex function but not necessarily smooth. \(A\) is a linear operator and \(\mathcal{K}\) is a closed convex cone: \(\{(y, t) \in \mathbb{R}^{n+1}: \|y\|_{\mathcal{K}} \leq t\}\).

The aforementioned models can all be mapped to this conic form. Consider the Lasso model, the objective function \(f(x)\) in the conic form is then \(\|x\|_1\). The linear mapping \(A(\cdot)\) maps \(x\) to \((Ax, 0)\), \(b\) can be defined as \((-y, \epsilon)\), \(K\) is then the epigraph of the \(l_2\)-norm, defined as \(L_2 = \{(y, t) \in \mathbb{R}^{n+1}: \|y\|_2 \leq t\}\). It can be seen that \(A(x) + b \in \mathcal{K}\) is equivalent to \((Ax - y, \epsilon) \in L_2\). By definition, this is the same as the constraint in the Lasso model.
The dual form of the conic formulation is as follows,

\[
\begin{align*}
\text{minimize} & \quad g(\lambda) \\
\text{subject to} & \quad \lambda \in \mathcal{K}^*,
\end{align*}
\]

where \( \mathcal{K}^* \) is the dual cone of \( \mathcal{K} \), \( g(\lambda) \) is the Lagrange dual function

\[
g(\lambda) = \inf_x L(x, \lambda) = \inf_x f(x) - \langle \lambda, A(x) + b \rangle.
\]

Sometimes the dual of the original problem is not differentiable, so we need refine the primal

\[
\begin{align*}
\text{minimize} & \quad f_\mu(x) := f(x) + \mu d(x) \\
\text{subject to} & \quad A(x) + b \in \mathcal{K},
\end{align*}
\]

where \( d(x) \) is a strongly convex function, e.g. entropy distance \( \ln n + \sum_{i=1}^n x_i \ln x_i \).

Then the dual form can be written as

\[
\begin{align*}
\text{minimize} & \quad g_\mu(\lambda) \\
\text{subject to} & \quad \lambda \in \mathcal{K}^*,
\end{align*}
\]

where \( g_\mu(\lambda) \) is a smoothed approximation of \( g \).

We can use projected gradient decedent method [145] to solve the dual problem,

\[
\lambda_{k+1} = \arg\min_{\lambda \in \mathcal{K}^*} \| \lambda_k + t_k \nabla g_\mu(\lambda_k) - \lambda \|_2.
\]

135
Now, all the problems are reduced to project a variable on a cone. Specifically, projecting a point \( x_0 \) on a cone \( \mathcal{K} \) corresponds to the following optimization problem,

\[
\begin{align*}
\text{minimize} & \quad \| x - x_0 \| \\
\text{subject to} & \quad x \in \mathcal{K}.
\end{align*}
\]

Different solutions will be derived if the target norms are different. Let \( P_{\mathcal{K}}(x_0) \) be the projection of \( x_0 \) onto \( \mathcal{K} \), we show some examples for different cone form of \( \mathcal{K} \).

1. Unit square in \( \mathbb{R}^2 \). Given \( \mathcal{K} = \{ x \in \mathbb{R}^2 : \| x \|_\infty \leq 1 \} \) and \( x_0 = (a, b) \), \( P_{\mathcal{K}}(x_0) \) w.r.t. the \( l_2 \)-norm will be

\[
P_{\mathcal{K}}(x_0) = (\text{sign}(a) \min(|a|, 1), \text{sign}(b) \min(|b|, 1)).
\]

An easy way to see this is from the simple 2-dimensional geometry, as is shown in Figure A.1.

![Figure A.1](image-url)
2. Rank-$k$ matrices. Given $\mathcal{K} = \{X \in \mathbb{R}^{m \times n} | \text{rank}(X) \leq k \}$ and $\text{rank}(X_0) = r$, $P_\mathcal{K}(X_0)$ w.r.t. the (spectral) $l_2$-norm will be,

$$P_\mathcal{K}(X_0) = \min\{k, r\} \sum_{i=1}^{\min\{k, r\}} \sigma_i u_i v_i^T,$$

where we assume $X_0$ has a singular value decomposition of $X = \sum_{i=1}^{r} \sigma_i u_i v_i^T$.

3. Scaled $l_2$-ball. Given $\mathcal{K} = \{x \in \mathbb{R}^n | \|x\|_2 \leq q \}$ and any $x_0 \in \mathbb{R}^n$, $P_\mathcal{K}(x_0)$ w.r.t. the $l_2$-norm will be

$$P_\mathcal{K}(x_0) = \begin{cases} qx_0 / \|x_0\|_2 & \text{if } \|x_0\|_2 \geq q, \\ x_0 & \text{otherwise.} \end{cases}$$

This can also be verified by simple geometry.

4. Hyperplane. Given $\mathcal{K} = \{x \in \mathbb{R}^n | a^T x = b \}$ and any $x_0 \in \mathbb{R}^n$, $P_\mathcal{K}(x_0)$ w.r.t. the $l_2$-norm will be

$$P_\mathcal{K}(x_0) = x_0 + \frac{(b - a^T x_0)a}{\|a\|_2^2}.$$
Appendix B

The Alternating Direction Method of Multipliers (ADMM)

The Alternating Direction Method of Multipliers (ADMM) is a widely used method in optimization problems. In this appendix, we consider how to solve the following optimization problem

\[
\begin{align*}
\text{minimize}_{x,z} \quad & f(x) + g(z) \\
\text{subject to} \quad & Ax + Bz = c
\end{align*}
\]

with variables \(x \in \mathbb{R}^n\) and \(z \in \mathbb{R}^m\), where \(A \in \mathbb{R}^{p \times n}, B \in \mathbb{R}^{p \times m}\). We assume that \(f\) and \(g\) are convex, but \(g\) is not necessarily smooth (differentiable). For example, \(g(x) = \|x\|_1\) is a convex nonsmooth function. The Lagrangian for problem (B.1) is\(^1\)

\[
L_0(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c)
\]

\(^1\)Here we follow the setup in [145]: we use \(y\) to denote the dual variable and also notice that the term \(y^T(Ax + Bz - c)\) is added to (instead of subtracted from) the objective function.
where $y$ is the dual variable or Lagrange multiplier. The constrained optimization problem (B.1) can be changed to an unconstrained optimization problem in the form of

$$\min_{x, y, z} L_0(x, z, y).$$

Augmented Lagrangian methods were developed with an added quadratic term on to $L_0(x, y, z)$ to increase the robustness of finding optimal solutions. The augmented Lagrangian for problem (B.1) is

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \rho \frac{1}{2} \|Ax + Bz - c\|_2^2,$$

where $\rho > 0$ is called the penalty parameter. The Alternating Direction Method of Multipliers (ADMM) is an iterative method which minimizes one of the variables $x$, $y$ and $z$ at a time. At the $(k + 1)$th iteration, it consists of

$$x^{k+1} := \arg\min_x L_\rho(x, z^k, y^k),$$

$$z^{k+1} := \arg\min_z L_\rho(x^{k+1}, z, y^k),$$

$$y^{k+1} := y^k + \rho( Ax^{k+1} + Bz^{k+1} - c),$$

where $\rho > 0$. Notice that $x^k$, $y^k$ and $z^k$ are already obtained from the last iteration and thus are constants in these problems. We can see that the problems (B.3), and (B.4) are simply single variable optimization problems. Instead of solving two variables jointly in the original problem (B.1), ADMM rewrites the problem into three sub-problems which are considerably easier to solve.

We demonstrate how to use ADMM to solve minimization problems involving sparsity
Basis pursuit [19]. The basis pursuit problem corresponds to the following optimization problem,

$$\begin{align*}
\text{minimize} \quad & \|x\|_1 \\
\text{subject to} \quad & Ax = b,
\end{align*}$$

where we have variable $x \in \mathbb{R}^n$, a known matrix $A \in \mathbb{R}^{m \times n}$, a known vector $b \in \mathbb{R}^m$, with $m < n$.

In ADMM form, problem (B.6) can be written as

$$\begin{align*}
\text{minimize} \quad & f(x) + \|z\|_1 \\
\text{subject to} \quad & x - z = 0,
\end{align*}$$

where $f$ is the indicator function of the affine set $Q := \{x \in \mathbb{R}^n \mid Ax = b\}$, defined as

$$f(x) = \begin{cases} 
0 & \text{if } x \in Q, \\
+\infty & \text{if } x \notin Q.
\end{cases}$$

The augmented Lagrange function will be

$$L_\rho(x, z, y) = f(x) + \|z\|_1 + y^T (x - z) + \frac{\rho}{2} \|x - z\|^2_2.$$
The minimization problem for \( x, z, y \) (as is in (B.3)-(B.5) will be

\[
x^{k+1} = \arg\min_x \left\{ f(x) + (y^k)^T (x - z^k) + \frac{\rho}{2} \|x - z^k\|_2^2 \right\}, \tag{B.7}
\]

\[
z^{k+1} = \arg\min_z \left\{ \|z\|_1 + (y^k)^T (x^{k+1} - z) + \frac{\rho}{2} \|x^{k+1} - z\|_2^2 \right\}, \tag{B.8}
\]

\[
y^{k+1} = y^k + \rho (x^{k+1} - z^{k+1}). \tag{B.9}
\]

Notice that by completing the square,

\[
(y^k)^T (x - z^k) + \frac{\rho}{2} \|x - z^k\|_2^2 = \frac{\rho}{2} \|x - z^k + \frac{1}{\rho} y^k\|_2^2 - \frac{2}{\rho} \|y^k\|_2^2
\]

\[
= \rho \|x - z^k + u^k\|_2^2 - \frac{\rho}{2} \|u^k\|_2^2,
\]

where \( u^k = \frac{1}{\rho} y^k \). Then the problem (B.3) becomes

\[
x^{k+1} = \arg\min_x \left\{ f(x) + \rho \|x - z^k + u^k\|_2^2 - \frac{\rho}{2} \|u^k\|_2^2 \right\}
\]

\[
= \arg\min_x \left\{ f(x) + \frac{\rho}{2} \|x - z^k + u^k\|_2^2 \right\}.
\]

As we have seen in Appendix A, this corresponds to the problem of finding the projection of the point \((z^k - u^k)\) onto the set indicated by \( f \), which is affine set \( Q \). It has a closed-form solution

\[
x^{k+1} = z^k - u^k - A^T (AA^T)^{-1} (A(z^k - u^k) - b). \tag{B.10}
\]

Similarly, it can be verified that

\[
z^{k+1} = \arg\min_z \left\{ \|z\|_1 + \frac{\rho}{2} \|x^{k+1} + z + u^k\|_2^2 \right\}.
\]

141
This problem has a closed-form solution

\[ z^{k+1} = \text{prox}_{1/\rho}(x^{k+1} + u^k), \quad (B.11) \]

where \( \text{prox}_{1/\rho} \) is the proximal operator and has the same definition as is in Chapter 5,

\[
(\text{prox}_\lambda(z))_i = \begin{cases} 
  z_i - \lambda & \text{if } z_i > \lambda, \\
  z_i + \lambda & \text{if } z_i < -\lambda, \\
  0 & |z_i| \leq \lambda.
\end{cases} \quad (B.12)
\]

Once we obtain \( x^{k+1} \) and \( z^{k+1} \), \( y^{k+1} \) can be updated by

\[ y^{k+1} = y^k + \rho(x^{k+1} - z^{k+1}). \quad (B.13) \]

The complete ADMM algorithm for solving the basis pursuit problem (B.6) is summarized in Algorithm 10.

**Algorithm 10** ADMM algorithm for solving the basis pursuit problem (B.6).

1. **Input:** matrix \( A \), vector \( b \), penalty parameter \( \rho \), maximum iteration number \( M \).
2. **Initialize** \( y^0, z^0 \).
3. **for** \( k = 1, 2, ..., M \) **do**
   4. Update \( x^k \) from Equation (B.10).
   5. Update \( z^k \) from Equation (B.11).
   6. Update \( y^k \) from (B.13).
   7. rela_err = \[ ||c^k_g - c^{k-1}_g||_2/||c^{k-1}_g||_2 \];
4. **end for**
5. **Output:** \( x^M, z^M \).

For more examples as well as detailed discussions on the ADMM algorithm, see [67].
Bibliography


