Composite Minimization: Proximity Algorithms and Their Applications

Feishe Chen
Syracuse University

Follow this and additional works at: https://surface.syr.edu/etd
Part of the Physical Sciences and Mathematics Commons

Recommended Citation
https://surface.syr.edu/etd/383

This Dissertation is brought to you for free and open access by the SURFACE at SURFACE. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE. For more information, please contact surface@syr.edu.
ABSTRACT

Image and signal processing problems of practical importance, such as incomplete data recovery and compressed sensing, are often modeled as nonsmooth optimization problems whose objective functions are the sum of two terms, each of which is the composition of a prox-friendly function with a matrix. Therefore, there is a practical need to solve such optimization problems. Besides the nondifferentiability of the objective functions of the associated optimization problems and the larger dimension of the underlying images and signals, the sum of the objective functions is not, in general, prox-friendly, which makes solving the problems challenging. Many algorithms have been proposed in literature to attack these problems by making use of the prox-friendly functions in the problems. However, the efficiency of these algorithms relies heavily on the underlying structures of the matrices, particularly for large scale optimization problems. In this dissertation, we propose a novel algorithmic framework that exploits the availability of the prox-friendly functions, without requiring any structural information of the matrices. This makes our algorithms suitable for large scale optimization problems of interest. We also prove the convergence of the developed algorithms.

This dissertation has three main parts. In part 1, we consider the minimization of functions that are the sum of the compositions of prox-friendly functions with matrices. We characterize the solutions to the associated optimization problems as the solutions of fixed point equations that are formulated in terms of the proximity
operators of the dual of the prox-friendly functions. By making use of the flexibility provided by this characterization, we develop a block Gauss-Seidel iterative scheme for finding a solution to the optimization problem and prove its convergence. We discuss the connection of our developed algorithms with some existing ones and point out the advantages of our proposed scheme.

In part 2, we give a comprehensive study on the computation of the proximity operator of the \( \ell_p \)-norm with \( 0 \leq p < 1 \). Nonconvexity and non-smoothness have been recognized as important features of many optimization problems in image and signal processing. The nonconvex, nonsmooth \( \ell_p \)-regularization has been recognized as an efficient tool to identify the sparsity of wavelet coefficients of an image or signal under investigation. To solve an \( \ell_p \)-regularized optimization problem, the proximity operator of the \( \ell_p \)-norm needs to be computed in an accurate and computationally efficient way. We first study the general properties of the proximity operator of the \( \ell_p \)-norm. Then, we derive the explicit form of the proximity operators of the \( \ell_p \)-norm for \( p \in \{0, 1/2, 2/3, 1\} \). Using these explicit forms and the properties of the proximity operator of the \( \ell_p \)-norm, we develop an efficient algorithm to compute the proximity operator of the \( \ell_p \)-norm for any \( p \) between 0 and 1.

In part 3, the usefulness of the research results developed in the previous two parts is demonstrated in two types of applications, namely, image restoration and compressed sensing. A comparison with the results from some existing algorithms is also presented. For image restoration, the results developed in part 1 are applied
to solve the $\ell_2$-TV and $\ell_1$-TV models. The resulting restored images have higher peak signal-to-noise ratios and the developed algorithms require less CPU time than state-of-the-art algorithms. In addition, for compressed sensing applications, our algorithm has smaller $\ell_2$- and $\ell_\infty$-errors and shorter computation times than state-of-the-art algorithms. For compressed sensing with the $\ell_p$-regularization, our numerical simulations show smaller $\ell_2$- and $\ell_\infty$-errors than that from the $\ell_0$-regularization and $\ell_1$-regularization. In summary, our numerical simulations indicate that not only can our developed algorithms be applied to a wide variety of important optimization problems, but also they are more accurate and computationally efficient than state-of-the-art algorithms.
Composite Minimization: Proximity Algorithms and Their Applications

By

Feishe Chen

B.S., Sun Yat-sen University, 2007
M.S., Sun Yat-sen University, 2009

Submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mathematics

Syracuse University

December 2015
Acknowledgments

First and foremost, I would like to express my sincerest and deepest appreciation to Professor Lixin Shen. He is the most important reason for why I chose Syracuse University. As an advisor, he has led me to the field of numerical optimization and its applications to image processing and compressed sensing. Along the journey of my Ph.D study, he has provided me with guidance, patience, encouragement and other support. I really appreciate that he has spent a lot of time discussing research projects with me, correcting mistakes from my research manuscripts and teaching me how to write a research paper. As a friend, he shares with me his experience in academia or in life and enlightens me when I come across confusion. I would also like to thank Professor Shen and his wife Mrs. Hongling He for the healthy and organic vegetables every summer.

In addition, I am thankful for my other collaborators, Professor Yuesheng Xu and Dr. Bruce Suter. I really appreciate their effort and insightful comments on every paper we have coauthored. Their interest and passion in research are always inspiring. It is my great honor to have had the chance to discuss research with them.
and to admire their broad vision in research. In the meantime, I am grateful for their support during my graduate study.

My thanks also go to the faculty members from Syracuse University. In particular, I would like to thank all of my dissertation committee members, Professors Uday Banerjee, Dan Coman, Peng Gao, Eugene Poletsky, and Grace Wang, for contributing their time, reading my dissertation and offering suggestions to improve this dissertation.

I am grateful for the help from the Department of Mathematics. The teaching assistantship has not only provided me financial support but also equipped me with communication and presentation skills. I am thankful for the help from the staff members, Ms. Madaline Argiro, Mr. Henry Barwotoe, Ms. Kim Canino, Mr. Benjamin Cooper, Ms. Christine Gilmore, Ms. Beckie Moon, Ms. Julie O’Connor, Ms. Patricia O’Malley, Ms. Carolyn Sabloski, and Ms. Sandra Ware, during my graduate study in Syracuse in the past six years.

I am indebted to my student colleagues and friends for their help in making my life joyful in Syracuse. I especially would like to thank Qiang Guo, Xiaofei Hu, Si Li, Xiaoxia Liu, Xuefei Ma, Jinxia Xie, Xueying Zeng, and Liang Zhao. I also want to thank Professor Wu-Teh Hsiang for organizing the Chinese New Year parties and making my family and me feel welcome when we are thousands of miles away from home.

Finally, special thanks have to go to my parents and other family members for
their unconditional and endless support and assistance. I am especially thankful to have my wife Shiqi Chen and my son Ryan Chen, to whom this dissertation is dedicated. They make my life in Syracuse more meaningful.
# Contents

Acknowledgments vi

1 Introduction 1

1.1 Problem Statement 1

1.2 Previous Work 3

1.3 Motivation 5

1.4 Contributions 6

2 Composite Minimization: Proximity Algorithms 8

2.1 Introduction 8

2.2 Fixed Point Characterization 9

2.3 Fixed Point Algorithm 17

2.4 Convergence Analysis 19

2.5 Connections with Existing Algorithms 27

2.5.1 Connection with Chambolle and Pock’s Algorithm 30

2.5.2 Connection with Augmented Lagrangian Methods 31
Chapter 1

Introduction

1.1 Problem Statement

In this dissertation, we consider minimization problems of the form

$$\min \{f_1(A_1x) + f_2(A_2x) : x \in \mathbb{R}^n\},$$

(1.1)

where $A_i$ are $m_i \times n$ matrices for $i = 1, 2$. On the other hand, the functions $f_i : \mathbb{R}^{m_i} \to (-\infty, +\infty]$ may be nonsmooth, but prox-friendly. A function $\Phi$ is prox-friendly ([6, 28]) if it allows us to solve, relatively easily, a subproblem of the form

$$\min_w \Phi(w) + \lambda \|w\|^2$$

for $\lambda > 0$.

Model (1.1) admits a wide variety of applications of interest. For instance, the total variation (TV) based ROF denoising model [66], the $\ell_2$-TV image deblurring [3,
16, 59, 73], the $\ell_1$-TV image restoration [25, 41], the framelet based image deblurring [9, 10], image inpainting [10], the basis pursuit problem in compressed sensing [23], medical imaging [51, 52] and the SVM models [27, 71] in machine learning can be identified as special cases of model (1.1). In particular, we briefly mention three applications that are closed related to our research. For ease of exposition, we view the terms $f_1 \circ A_1$ and $f_2 \circ A_2$ in model (1.1) as the fidelity and regularization terms, respectively.

- **Image deblurring with $\ell_2$-fidelity term.** The aim of image deblurring is to recover the underlying image from a noisy blurred image. If the observed image is corrupted by noise of Gaussian type, an $\ell_2$-type function is favored for forming fidelity term. As a consequence, $f_1$ can be chosen as the $\ell_2$-norm or the indicator function over an $\ell_2$-ball whose radius indicates the noise power. The matrix $A_1$ is determined by the underlying imaging acquisition system. Various choices are available for the regularization term. For instance, if the tight frame regularizer [31, 65] is chosen, the matrix $A_2$ corresponds to the frame system and $f_2$ is simply the $\ell_1$-norm. If the total variation [66] is adopted, $A_2$ is the first order difference operator and $f_2$ is a variant of the $\ell_1$-norm.

- **Image deblurring with $\ell_1$-fidelity term.** When a blurred image is contaminated by noise of non-Gaussian type, the $\ell_2$-type function is not appropriate for forming fidelity term anymore. It is well accepted that the $\ell_1$-norm fidelity term can effectively suppress the effect of outliers that may contaminate a given image,
and is therefore particularly suitable for handling impulsive noise [19, 58]. In this case, the $\ell_1$-norm is preferred for function $f_1$ in the fidelity term. The matrix $A_1$ and the regularization term $f_2 \circ A_2$ can be chosen as those in the image deblurring model with $\ell_2$-fidelity term.

- **Compressed sensing.** The goal in compressed sensing is to recover the underlying sparse signal from incomplete measurements that are, possibly, contaminated by Gaussian white noise. As a consequence, $f_1$ should be an $\ell_2$-type function and $A_1$ is the associated measurement matrix. Further, in compressed sensing the signal of interest is sparsely represented in a suitably chosen transform domain. Hence, $A_2$ should be chosen as the transformation matrix associated with the transform. The function $f_2$ can be chosen to be the $\ell_p$-norm with $0 \leq p \leq 1$. A discussion on the $\ell_p$-norm as a sparse-promoting function will be given in Chapter 3.

### 1.2 Previous Work

A number of algorithms have been developed for solving the optimization problem (1.1). Depending whether the proximity operators of $f_1 \circ A_1$ and $f_2 \circ A_2$ are prox-friendly or not, the existing algorithms can be roughly categorized in three groups.

- **Both $f_1 \circ A_1$ and $f_2 \circ A_2$ are prox-friendly.** In this case, splitting algorithms such
as Douglas-Rachford algorithm [35, 50] can be adopted for solving problem (1.1).

- **Either** $f_1 \circ A_1$ or $f_2 \circ A_2$ **is prox-friendly.** Under this circumstance, the first order prima-dual algorithms recently developed in [15, 22, 36, 42] are suitable for solving problem (1.1).

- **Both** $f_1$ and $f_2$ **are prox-friendly while both** $f_1 \circ A_1$ and $f_2 \circ A_2$ **are not.** In this context, existing algorithms for the optimization problem (1.1) can be roughly classified into two classes. Class 1 collects the algorithms that produce exact solutions to problem (1.1) while Class 2 collects the algorithms that give approximate solutions to problem (1.1). As we know, the coupling of a prox-friendly function with a matrix causes the difficulty in solving the optimization problem (1.1). This difficulty is tackled in different ways in the development of algorithms in Class 1 and Class 2. In the development of algorithms in Class 1, two auxiliary variables are introduced to substitute the multiplications $A_1x$ and $A_2x$ in (1.1). As a result, the unconstrained optimization problem (1.1) is converted to a constrained one. The resulting constrained optimization problem can be solved by the split Bregman method [40], the Augmented Lagrangian method (ALM) [39, 43, 60, 63], or the alternating direction method of multipliers (ADMM) [8, 38]. Methods of the split Bregman, ALM and ADMM have been extensively applied in image restoration [1, 10, 40, 59, 70, 74]. In the development of algorithms in Class 2, some auxiliary variables are introduced, but used in a different way. For example, for the term $f_1(A_1x)$ in (1.1), we use a
variable $u$ to replace $A_1 x$ in the expression $f_1(A_1 x)$ and then enforce $u$ and $A_1 x$

close measured by the $\ell_2$-norm of their difference. As a result, the solutions to
the resulting optimization problem are no longer the solutions, but approximate
ones, to the optimization problem (1.1). Algorithms designed in this line can
be found in [21, 25, 32, 41, 55, 72, 76], and the references therein. A poten-
tial shortcoming of the algorithms in Class 2 is that the solutions produced by
these algorithms may not possess desirable features as expected from the origi-
nal problem. Therefore, algorithms in Class 1 are preferred for problem (1.1).

1.3 Motivation

Based upon the review presented above, our research will focus on enriching and
complementing the existing algorithms in Class 1. To motivate our work, let us state
assumptions on problem (1.1) in the following discussion and point out shortcomings
of the existing algorithms in Class 1. We assume that

A1. Both $f_1$ and $f_2$ are prox-friendly.

A2. Both $f_1 \circ A_1$ and $f_2 \circ A_2$ are not prox-friendly.

Under these assumptions, we briefly review a general procedure in the development
of the existing algorithms in Class 1. By introducing two auxiliary variables $u$ and $v$, problem (1.1) is converted to the following one

$$\min \{f_1(u) + f_2(v) : A_1 x - u = 0, A_2 x - v = 0, x \in \mathbb{R}^n, u \in \mathbb{R}^{m_1}, v \in \mathbb{R}^{m_2}\},$$  (1.2)
which is an optimization problem with linear constraints. The split Bregman method, ALM or ADMM can be adopted for solving the above constrained optimization problem. With any one of these algorithms, three sequences \( \{u^k\}, \{v^k\}, \) and \( \{x^k\} \) are generated. We can observe that the updating \( u \) and \( v \) are independent in the sense that the updated \( u^{k+1} \) is not used in updating \( v^{k+1} \), and vise versa. Therefore, the block Gauss-Seidel acceleration technique will not take effect. In addition, updating \( u \) and \( v \) may require solving large scale systems that could be expensive if the matrices \( A_1 \) and \( A_2 \) do not have special structures to exploit.

\section{1.4 Contributions}

In this dissertation, we propose a novel algorithmic framework that exploits the availability of the prox-friendly functions, without requiring any structural information of the matrices. This makes our proposed algorithms suitable for large scale optimization problems of interest. We also prove the convergence of the developed algorithms.

Our contributions are as follows:

- We characterize the solutions to the optimization problem (1.1) as the solutions of fixed point equations that are formulated in terms of the proximity operators of the dual of the prox-friendly functions \( f_1 \) and \( f_2 \). By making use of the flexibility provided by this characterization, we develop a block Gauss-Seidel iterative scheme for finding a solution to the optimization problem and prove its convergence. We discuss the connection of our developed algorithms with
some existing ones and point out the advantages of our proposed scheme.

- We give a comprehensive study on the computation of the proximity operator of the $\ell_p$-norm with $0 \leq p < 1$. We first study the general properties of the proximity operator of the $\ell_p$ norm. Then, we derive the explicit form of the proximity operators of the $\ell_p$ norm for $p \in \{0, 1/2, 2/3, 1\}$. Using these explicit forms and the properties of the proximity operator of the $\ell_p$-norm, we develop an efficient algorithm to compute the proximity operator of the $\ell_p$-norm for any $p$ between 0 and 1.

- We demonstrate the usefulness of our research results developed in two types of applications, namely, image restoration and compressed sensing. A comparison with the results from some existing algorithms is also presented. Our numerical simulations indicate that not only can our developed algorithms be applied to a wide variety of important optimization problems, but also they are more accurate and computationally efficient than state-of-the-art algorithms.
Chapter 2

Composite Minimization:

Proximity Algorithms

2.1 Introduction

In this chapter, we focus on convex composite minimization problem with form (1.1), that is,

\[
\min \{ f_1(A_1x) + f_2(A_2x) : x \in \mathbb{R}^n \},
\]

where \( f_1, f_2 \) are proper, lower semi-continuous, convex functions. We assume that both of \( f_1 \) and \( f_2 \) are prox-friendly functions but neither of \( f_1 \circ A_1 \) and \( f_2 \circ A_2 \) are prox-friendly. We characterize the solutions to composite minimization problem (1.1) as the solutions of fixed point equations that are formulated in terms of the proximity operators of the dual of \( f_1 \) and \( f_2 \). By making use of the flexibility provided
by this characterization, we develop a block Gauss-Seidel iterative scheme for finding a solution to the optimization problem. We show the proposed algorithm can be implemented efficiently when the functions $f_1$ and $f_2$ are prox-friendly. Further, convergence analysis on the proposed algorithm is fulfilled using firm non-expansiveness of the proximity operator. Lastly, connection of the proposed algorithm with the Chambolle and Pock’s primal-dual method (CP), the augmented lagrangian method (ALM) and the alternating direction method of multipliers (ADMM) will be discussed.

This chapter is organized in the following manner. In section 2.2, we provide characterization of solutions to general problem (1.1) via sub-differentials and fixed point equations based on proximity operators. In section 2.3, we propose a fixed point algorithm in term of proximity operators. The proposed algorithm employs block Gauss-Seidel acceleration. In section 2.4, convergence analysis on the proposed algorithm is provided in this section. In section 2.5, we discuss the connection of the proposed algorithms with CP[15], ALM and ADMM.

2.2 Fixed Point Characterization

In this section, we shall see that a solution of (1.1) can be characterized by fixed point equations in terms of proximity operators. An iterative algorithm based on the fixed-point equations will be proposed to solve model (1.1).

We begin with introducing our notation and reviewing some concepts from convex analysis. For a vector $x$ in the $d$-dimensional Euclidean space $\mathbb{R}^d$, we use $x_i$ to denote
the \(i\)th component of a vector \(x \in \mathbb{R}^d\) for \(i = 1, 2, \ldots, d\). We define \(\langle x, y \rangle := \sum_{i=1}^{d} x_i y_i\), for \(x, y \in \mathbb{R}^d\) the standard inner product in \(\mathbb{R}^d\). The \(\ell_2\)-norm induced by the inner product in \(\mathbb{R}^d\) is defined as \(\|\cdot\| := \sqrt{\langle \cdot, \cdot \rangle}\). For a \(k \times d\) matrix \(A\), its \(\ell_2\)-norm, denoted by \(\|A\|\) is defined by \(\|A\| = \max\{\|Ax\| : \|x\| = 1, x \in \mathbb{R}^d\}\). By \(\mathbb{S}_+^d\), we denote the set of all \(d \times d\) symmetric, positive definite matrix. Given a matrix \(H \in \mathbb{S}_+^d\), the weighted inner product associated with \(H\) in \(\mathbb{R}^d\) is defined by \(\langle x, y \rangle_H := \langle x, Hy \rangle\) and its induced norm is defined by \(\|x\|_H := \sqrt{\langle x, Hx \rangle}\). When \(H\) is the identity matrix, its associated weighted inner product and induced norm reduce to the standard inner product and \(\ell_2\)-norm in \(\mathbb{R}^d\) respectively. For the Hilbert space \(\mathbb{R}^d\), the class of all lower semicontinuous convex functions \(\psi : \mathbb{R}^d \to (-\infty, +\infty]\) such that \(\text{dom } \psi := \{x \in \mathbb{R}^d : \psi(x) < +\infty\} \neq \emptyset\) is denoted by \(\Gamma_0(\mathbb{R}^d)\).

We shall provide necessary and sufficient conditions for a solution to model (1.1). To this end, we first recall the definitions of sub-differential and Fenchel conjugate. The subdifferential of \(\psi \in \Gamma_0(\mathbb{R}^d)\), denoted by \(\partial \psi\), is a set-valued operator and is defined at \(x \in \mathbb{R}^d\) as follows:

\[
\partial \psi(x) := \{y \in \mathbb{R}^d : \psi(z) \geq \psi(x) + \langle y, z - x \rangle \text{ for all } z \in \mathbb{R}^d\}.
\]

For a function \(\psi \in \Gamma_0(\mathbb{R}^d)\), the sub-differential \(\partial \psi(x)\) is a non-empty compact set for any \(x \in \text{dom } \psi\) (see. e.g., [64]). For a function \(\psi : \mathbb{R}^d \to [-\infty, +\infty]\), the Fenchel conjugate of \(\psi\) at \(x \in \mathbb{R}^d\) is

\[
\psi^*(x) := \sup\{\langle y, x \rangle - \psi(y) : y \in \mathbb{R}^d\}.
\]

For a function \(\psi \in \Gamma_0(\mathbb{R}^d)\), its sub-differential and Fenchel conjugate are closely
related. Indeed, for a function $\psi \in \Gamma_0(\mathbb{R}^d)$, one has (see, e.g., [64, Proposition 11.3])

$$y \in \partial \psi(x) \iff x \in \partial \psi^*(y).$$

(2.1)

The following result provides a characterization to a solution to problem (1.1).

**Proposition 2.1.** Assume that the set of solutions to the optimization problem (1.1) is nonempty. A vector $x \in \mathbb{R}^n$ is a solution to problem (1.1) if and only if there exist vectors $u \in \mathbb{R}^{m_1}$ and $v \in \mathbb{R}^{m_2}$ such that the following relations hold

$$A_1x \in \partial f_1^*(u),$$

(2.2)

$$A_2x \in \partial f_2^*(v),$$

(2.3)

$$A_1^T u + A_2^T v = 0.$$  

(2.4)

**Proof.** Suppose $x$ is a solution to problem (1.1). By Fermat’s rule, $0 \in A_1^T \partial f_1(A_1x) + A_2^T \partial f_2(A_2x)$. Therefore, there exist $u \in \partial f_1(A_1x)$ and $v \in \partial f_2(A_2x)$ such that $0 = A_1^T u + A_2^T v$, that is, (2.4) holds. Further, by (2.1), $u \in \partial f_1(A_1x)$ and $v \in \partial f_2(A_2x)$ yield relations (2.2) and (2.3), respectively.

The above reasoning is reversible. That is, if there exist $u \in \mathbb{R}^{m_1}$ and $v \in \mathbb{R}^{m_2}$ such that (2.2)-(2.4) hold, then $x$ is a solution to problem (1.1).\hfill \Box

Based on Proposition 2.1, we shall provide fixed point equations characterization of a solution to model (1.1) in terms of proximity operator. For a function $\psi \in \Gamma_0(\mathbb{R}^d)$, the proximity operator of $\psi$ with respect to $H \in \mathbb{S}_+^d$, denoted by $\text{prox}_{\psi,H}$, is a mapping from $\mathbb{R}^d$ to itself, defined at $x \in \mathbb{R}^d$ by

$$\text{prox}_{\psi,H}(x) := \text{argmin} \left\{ \frac{1}{2} \| u - x \|^2_H + \psi(u) : u \in \mathbb{R}^d \right\}.$$  

(2.5)
In particular, we use $\text{prox}_{\lambda \psi}^{}$ for $\text{prox}_{\psi, \frac{1}{\lambda} I}$, where $\lambda > 0$ is a scalar.

The proximity operator is firmly non-expansive [2]. An operator $\mathcal{J} : \mathbb{R}^d \to \mathbb{R}^d$ is called firmly non-expansive with respect to a given matrix $H \in \mathbb{S}^d_+$ if for all $x, y \in \mathbb{R}^d$

$$\|\mathcal{J} y - \mathcal{J} x\|_H^2 \leq \langle \mathcal{J} y - \mathcal{J} x, y - x \rangle_H.$$  

It can be observed that a firm non-expansive operator is also Lipschitz continuous with Lipschitz constant 1. For the sake of completeness, the firm non-expansiveness of proximity operator will be shown in the following lemma.

**Lemma 2.2.** Given $\psi \in \Gamma_0(\mathbb{R}^d)$ and $H \in \mathbb{S}_+^d$, the proximity operator $\text{prox}_{\psi, H}$ is firmly non-expansive with respect to $H$.

**Proof.** Let $x, y \in \mathbb{R}^d$. By the definition of proximity operator, we have

$$0 \in \partial \psi(\text{prox}_{\psi, H}(x)) + H(\text{prox}_{\psi, H}(x) - x),$$

and

$$0 \in \partial \psi(\text{prox}_{\psi, H}(y)) + H(\text{prox}_{\psi, H}(y) - y),$$

i.e.,

$$H(x - \text{prox}_{\psi, H}(x)) \in \partial \psi(\text{prox}_{\psi, H}(x)),$$

and

$$H(y - \text{prox}_{\psi, H}(y)) \in \partial \psi(\text{prox}_{\psi, H}(y)).$$
The definition of sub-differential yields
\[
\begin{align*}
\langle \text{prox}_{\psi,H}(y) - \text{prox}_{\psi,H}(x), H(x - \text{prox}_{\psi,H}(x)) \rangle &+ \psi(\text{prox}_{\psi,H}(x)) \\
\langle \text{prox}_{\psi,H}(x) - \text{prox}_{\psi,H}(y), H(y - \text{prox}_{\psi,H}(y)) \rangle &+ \psi(\text{prox}_{\psi,H}(y)) \leq \psi(\text{prox}_{\psi,H}(x))
\end{align*}
\]
Adding the above two inequalities and rearranging terms yield
\[
\langle \text{prox}_{\psi,H}(y) - \text{prox}_{\psi,H}(x), H(\text{prox}_{\psi,H}(y) - \text{prox}_{\psi,H}(x)) \rangle \leq \langle \text{prox}_{\psi,H}(y) - \text{prox}_{\psi,H}(x), H(y - x) \rangle.
\]
This completes the proof. □

The sub-differential and the proximity operator are closely related. This relation is given in the next proposition.

**Proposition 2.3.** Let \( \psi \in \Gamma_0(\mathbb{R}^d) \), \( H \in \mathbb{S}^{d+} \) and \( x, y \in \mathbb{R}^d \). Then \( Hy \in \partial \psi(x) \) if and only if \( x = \text{prox}_{\psi,H}(x+y) \).

**Proof.** Assume \( x = \text{prox}_{\psi,H}(x+y) \). By the definition of proximity operator, we have
\[
x = \arg\min \left\{ \frac{1}{2} \| z - (x+y) \|^2_H + \psi(z) : z \in \mathbb{R}^d \right\}.
\]
Being the minimizer of the objective function above, \( x \) satisfies the inclusion \( 0 \in H(x - (x+y)) + \partial \psi(x) \), i.e., \( Hy \in \partial \psi(x) \). This shows that \( x = \text{prox}_{\psi,H}(x+y) \) implies \( Hy \in \partial \psi(x) \).

The above reasoning is reversible. That is, if \( Hy \in \partial \psi(x) \), then \( x = \text{prox}_{\psi,H}(x+y) \).

This completes the proof. □

In particular, if \( \psi \in \Gamma_0(\mathbb{R}^d) \) and \( \lambda > 0 \), by choosing \( H = \frac{1}{\chi} I \) we have from Proposition 2.3 that
\[
y \in \partial \psi(x) \iff x = \text{prox}_{\lambda \psi}(x + \lambda y).
\] (2.6)
With the relationship between the proximity operator and sub-differential given in Proposition 2.3, an inclusion involving sub-differential can be rephrased as an equation in terms of proximity operator. As a consequence, the characterization of a solution to model (1.1) described in Proposition 2.1 can be rewritten as fixed point equations.

**Proposition 2.4.** Assume that the set of solutions to the optimization problem (1.1) is nonempty. A vector \( x \in \mathbb{R}^n \) is a solution to model (1.1) if and only if for any positive numbers \( \alpha_1 > 0, \alpha_2 > 0, \gamma > 0 \), there exist \( u \in \mathbb{R}^{m_1} \) and \( v \in \mathbb{R}^{m_2} \) such that the following equations hold

\[
\begin{align*}
  u &= \text{prox}_{\alpha_1 f_1^*} (u + \alpha_1 A_1 x) \\
  v &= \text{prox}_{\alpha_2 f_2^*} (v + \alpha_2 A_2 x) \\
  x &= x - \gamma (A_1^T u + A_2^T v)
\end{align*}
\]  

(2.7)

**Proof.** It follows immediately from proposition 2.1 and equation (2.6). \( \square \)

We show equations in (2.7) can be rewritten in a compact form. To this end, we denote \( \mathbb{H} := \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n \) and define an operator \( \mathcal{T} : \mathbb{H} \to \mathbb{H} \) at \( \rho = (u, v, x) \in \mathbb{H} \) by

\[
\mathcal{T}(\rho) := (\text{prox}_{\alpha_1 f_1^*} (u), \text{prox}_{\alpha_2 f_2^*} (v), x) .
\]  

(2.8)

We next show \( \mathcal{T} \) defined in the above is the proximity operator of a new function with respect to a matrix in \( \mathbb{S}_+^d \). Actually, define \( F : \mathbb{H} \to \mathbb{R} \) at \( \rho \in \mathbb{H} \) as

\[
F(\rho) := f_1^* (u) + f_2^* (v)
\]  

(2.9)
and a diagonal matrix
\[ R := \text{diag} \left( \frac{1}{\alpha_1} I, \frac{1}{\alpha_2} I, \frac{1}{\gamma} I \right), \]  
(2.10)

where \( \frac{1}{\alpha_1} I, \frac{1}{\alpha_2} I, \frac{1}{\gamma} I \) are \( m_1 \times m_1, m_2 \times m_2 \) and \( n \times n \) scaled identity matrices respectively. With this notational preparation, we are ready to show that \( T \) is the proximity operator of \( F \) with respect to \( R \).

**Lemma 2.5.** For \( T, F \) and \( R \) defined by (2.8), (2.9) and (2.10), respectively, one has \( T = \text{prox}_{F,R} \).

**Proof.** For \( \rho = (u, v, x) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n \), by the definition of the proximity operator, we have that

\[
\text{prox}_{F,R}(\rho) = \arg\min \left\{ \frac{1}{2} \| \rho - \tilde{\rho} \|_R^2 + F(\tilde{\rho}) : \tilde{\rho} = (\tilde{u}, \tilde{v}, \tilde{x}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n \right\}
\]

\[
= \arg\min \left\{ \frac{1}{2} \| u - \tilde{u} \|_{\frac{1}{\alpha_1} I}^2 + f_1^*(\tilde{u}) + \frac{1}{2} \| v - \tilde{v} \|_{\frac{1}{\alpha_2} I}^2 + f_2^*(\tilde{v}) + \frac{1}{2} \| x - \tilde{x} \|_{\frac{1}{\gamma} I}^2 : (\tilde{u}, \tilde{v}, \tilde{x}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n \right\}
\]

\[
= \left( \text{prox}_{\alpha_1 f_1}(u), \text{prox}_{\alpha_2 f_2}(v), x \right)
\]

\[
= T(\rho).
\]

Lemma 2.2 and Lemma 2.5 ensure that the operator \( T \) is firmly non-expansive with respect to matrix \( R \).

Define
\[
S_1 := \begin{bmatrix}
0 & 0 & A_1 \\
0 & 0 & A_2 \\
-A_1 & -A_2 & 0
\end{bmatrix}
\]  
(2.12)
and
\[ E_1 := I + R^{-1}S_1. \] (2.13)

Then for \( \rho = (u, v, x) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n \), the characterization in (2.7) can be rewritten in a compact form as
\[ \rho = T(E_1 \rho). \] (2.14)

By Proposition 2.4 and equation (2.14), a solution to problem (1.1) is essentially a fixed point of the operator \( T \circ E_1 \). Although the operator \( T \) is firmly non-expansive, the composition \( T \circ E_1 \) might not be due to the expansivity of \( E_1 \). We shall show this in the following lemma.

For a \( d \times d \) matrix \( A \), the norm \( \| A \|_H \) with respect to an \( H \in S_+^d \) is defined as
\[ \| A \|_H := \max \{ \| Ax \| H : x \in \mathbb{R}^d, \| x \|_H = 1 \}. \]

**Lemma 2.6.** Let \( R \) and \( E_1 \) be defined in (2.10) and (2.12), respectively. Then \( \| E_1 \|_R > 1 \).

**Proof.** Given any \( \rho \in \mathbb{H} \) with \( \| \rho \|_R = 1 \). By the definition of \( E_1 \), one have
\[ \| E_1 \rho \|_R^2 = \| (I + R^{-1}S_1) \rho \|_R^2 \]
\[ = \| \rho \|_R^2 + 2 \langle \rho, RR^{-1}S_1 \rho \rangle + \| R^{-1}S_1 \rho \|_R^2 \]
\[ = 1 + 2 \langle \rho, S_1 \rho \rangle + \| R^{-1}S_1 \rho \|_R^2. \]

Noting that \( S_1 \) is a nonzero skew matrix, we have \( \langle \rho, S_1 \rho \rangle = 0 \) and there exists some \( \rho \in \mathbb{H} \) with \( \| \rho \|_R = 1 \) such that \( S_1 \rho \neq 0 \). Hence \( \| E_1 \|_R > 1 \).
CHAPTER 2. COMPOSITE MINIMIZATION: PROXIMITY ALGORITHMS  

As a consequence of Lemma 2.6, the sequence \( \{\rho_k\} \) generated by \( \rho^{k+1} = T(E_1\rho^k) \) with a given initial guess \( \rho^0 \), may not converge. Actually, it was already observed numerically in the application of the L1/TV model for impulsive noise removal [55].

2.3 Fixed Point Algorithm

Our goal is to develop an algorithm that can be used for finding a solution to equation (2.7) (i.e., (2.14)). It was pointed out in the previous section that a simple iterative scheme would not be enough to yield a solution to equation (2.14). Since any solution to problem (1.1) is also a solution to equation (2.7), this motivates us to derive from (2.7) a mathematically equivalent characterization with which an iterative scheme derived from the new characterization will lead to a solution to problem (1.1).

Proposition 2.7. Assume that the set of solutions to the optimization problem (1.1) is nonempty. A vector \( x \in \mathbb{R}^n \) is a solution to (1.1) if and only if for any positive numbers \( \alpha_1 > 0, \alpha_2 > 0, \beta > 0, \gamma > 0 \), there exist \( u \in \mathbb{R}^{m_1}, v \in \mathbb{R}^{m_2} \) such that the following hold

\[
\begin{align*}
    u &= \text{prox}_{\alpha_1 f_1} \left( u + \alpha_1 A_1(x - \beta(A_1^\top u + A_2^\top v)) \right), \\
    v &= \text{prox}_{\alpha_2 f_2} \left( v + \alpha_2 A_2(x - \beta(A_1^\top u + A_2^\top v)) \right), \\
    x &= x - \gamma(A_1^\top u + A_2^\top v). 
\end{align*}
\]

(2.15)

Proof. This follows from Proposition 2.4 and the fact that \( A_1^\top u + A_2^\top v = 0 \).  

Next, we present an iterative scheme for finding solutions to (2.15). For purposes of comparison, we first include an iterative scheme arising from the characterization
CHAPTER 2. COMPOSITE MINIMIZATION: PROXIMITY ALGORITHMS

given in (2.7). Beginning with an initial estimate \((u^0, v^0, x^0) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n\), this scheme updates its variables as follows:

\[
\begin{align*}
    u^{k+1} &= \text{prox}_{\alpha_1 f_1^*} \left( u^k + \alpha_1 A_1 x^k \right) \\
    v^{k+1} &= \text{prox}_{\alpha_2 f_2^*} \left( v^k + \alpha_2 A_2 x^k \right) \\
    x^{k+1} &= x^k - \gamma (A_1^\top u^{k+1} + A_2^\top v^{k+1})
\end{align*}
\]

(2.16)

From the above scheme, we can see that updating \(u^{k+1}\) and \(v^{k+1}\) can be parallelized in the sense that the computation of \(v^{k+1}\) is independent of the update of \(u^{k+1}\). We then turn to the characterization given in Proposition 2.15. Beginning with an initial estimate \((u^0, v^0, x^0) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n\), we propose an iterative scheme arising from (2.15) that iterates as

\[
\begin{align*}
    u^{k+1} &= \text{prox}_{\alpha_1 f_1^*} \left( u^k + \alpha_1 A_1 \left( x^k - \beta (A_1^\top u^k + A_2^\top v^k) \right) \right) \\
    v^{k+1} &= \text{prox}_{\alpha_2 f_2^*} \left( v^k + \alpha_2 A_2 \left( x^k - \beta (A_1^\top u^{k+1} + A_2^\top v^{k+1}) \right) \right) \\
    x^{k+1} &= x^k - \gamma (A_1^\top u^{k+1} + A_2^\top v^{k+1})
\end{align*}
\]

(2.17)

The above scheme (2.17) shows that the update \(u^{k+1}\) can be immediately used in computing \(v^{k+1}\). Algorithm 1 describes an entire procedure for finding a solution to problem (1.1) based on the characterization in Proposition 2.7.

<table>
<thead>
<tr>
<th>Algorithm 1: Gauss-Seidel Method for Model (1.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Initialization: (u^0 \in \mathbb{R}^{m_1}, \quad v^0 \in \mathbb{R}^{m_2}, \quad x^0 \in \mathbb{R}^n;) (\alpha_1, \alpha_2, \beta, \gamma &gt; 0).</td>
</tr>
<tr>
<td><strong>Result:</strong> (x^\infty)</td>
</tr>
<tr>
<td><strong>while</strong> it is not convergent <strong>do</strong></td>
</tr>
<tr>
<td>Computing ((u^{k+1}, v^{k+1}, x^{k+1})) via the iterative scheme (2.17).</td>
</tr>
</tbody>
</table>

In the following section, convergence analysis of Algorithm 1 will be given.
2.4 Convergence Analysis

In this section, our effort will be devoted to the convergence analysis of Algorithm 1. For easy of exposition, let us introduce the following notation:

\begin{align*}
S_1 &= \begin{bmatrix}
0 & 0 & A_1 \\
0 & 0 & A_2 \\
-A_1^T & -A_2^T & 0
\end{bmatrix}, \\
S_2 &= \begin{bmatrix}
0 & -\beta A_1 A_2^T & 0 \\
-\beta A_2 A_1^T & 0 & 0 \\
0 & 0 & 0 \\
\frac{1}{\alpha_1} I - \beta A_1 A_1^T & 0 & 0 \\
0 & \frac{1}{\alpha_2} I - \beta A_2 A_2^T & 0 \\
0 & 0 & \frac{1}{\gamma} I
\end{bmatrix}, \\
P &= \begin{bmatrix}
\frac{1}{\alpha_1} I - \beta A_1 A_1^T & 0 & 0 \\
0 & \frac{1}{\alpha_2} I - \beta A_2 A_2^T & 0 \\
0 & 0 & \frac{1}{\gamma} I
\end{bmatrix}, \\
E &= R^{-1}(P + S_1 + S_2).
\end{align*}

Then the fixed point equations (2.15) can be rewritten in the compact form

\[ \rho = T(E\rho), \quad (2.19) \]

where \( \rho = (u, v, x) \) and \( T \) is defined by (2.8).

One useful property of the operator \( T \circ E \) is as follows.

**Lemma 2.8.** Let \( T, R, \) and \( E \) be defined in (2.8), (2.10), and (2.18), respectively. If pairs \( (\rho_i, a_i) \in \mathbb{H} \times \mathbb{H} \) with \( \rho_i = (u_i, v_i, x_i) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_1} \times \mathbb{R}^n, i = 1, 2, \) satisfy

\[ \rho_i = T(E\rho_i + R^{-1}a_i), \quad (2.20) \]
then
\begin{equation}
\langle \rho_2 - \rho_1, a_2 - a_1 \rangle \geq \beta \| A_1^\top (u_2 - u_1) + A_2^\top (v_2 - v_1) \|^2. \tag{2.21}
\end{equation}

Proof. By Lemma 2.5 and the firm non-expansiveness of $T$,
\begin{align*}
\| \rho_2 - \rho_1 \|_R^2 &= \| T(E\rho_2 + R^{-1}a_2) - T(E\rho_1 + R^{-1}a_1) \|_R^2 \\
&\leq \langle \rho_2 - \rho_1, RE(\rho_2 - \rho_1) + a_2 - a_1 \rangle \\
&= \langle \rho_2 - \rho_1, (P + S_1 + S_2)(\rho_2 - \rho_1) \rangle + \langle \rho_2 - \rho_1, a_2 - a_1 \rangle \\
&= \langle \rho_2 - \rho_1, P(\rho_2 - \rho_1) \rangle + \langle \rho_2 - \rho_1, S_1(\rho_2 - \rho_1) \rangle \\
&\quad + \langle \rho_2 - \rho_1, S_2(\rho_2 - \rho_1) \rangle + \langle \rho_2 - \rho_1, a_2 - a_1 \rangle.
\end{align*}
Noting that $S_1$ is a skewed matrix, we have that $\langle \rho_2 - \rho_1, S_1(\rho_2 - \rho_1) \rangle = 0$. Thus,
\begin{equation}
\langle \rho_2 - \rho_1, a_2 - a_1 \rangle \geq \langle \rho_2 - \rho_1, (R - P)(\rho_2 - \rho_1) \rangle - \langle \rho_2 - \rho_1, S_2(\rho_2 - \rho_1) \rangle. \tag{2.22}
\end{equation}

By the definitions of $P$ and $S_2$ given in (2.18), we have that
\begin{align*}
\langle \rho_2 - \rho_1, (R - P)(\rho_2 - \rho_1) \rangle &= \beta \| A_1^\top (u_2 - u_1) \|^2 + \beta \| A_2^\top (v_2 - v_1) \|^2 \\
\langle \rho_2 - \rho_1, S_2(\rho_2 - \rho_1) \rangle &= 2\beta \langle A_1^\top (u_2 - u_1), A_2^\top (v_2 - v_1) \rangle.
\end{align*}

Hence,
\begin{align*}
\langle \rho_2 - \rho_1, a_2 - a_1 \rangle &\geq \beta \| A_1^\top (u_2 - u_1) \|^2 + \beta \| A_2^\top (v_2 - v_1) \|^2 \\
&\quad + 2\beta \langle A_1^\top (u_2 - u_1), A_2^\top (v_2 - v_1) \rangle \\
&= \beta \| A_1^\top (u_2 - u_1) + A_2^\top (v_2 - v_1) \|^2.
\end{align*}

This completes the proof. \qed
CHAPTER 2. COMPOSITE MINIMIZATION: PROXIMITY ALGORITHMS

For a given $a \in \mathbb{H}$, we define $\mathcal{M} : \mathbb{H} \to \mathbb{H}$ by $\rho = \mathcal{M}(a)$ if $(\rho, a)$ satisfies (2.20).

Lemma 2.8 actually implies the monotonicity of the operator $\mathcal{M}$. We next will discuss the relation between two consecutive iterations from the iterative scheme (2.17). For ease of exposition, we introduce the following notation:

\[
L = \begin{bmatrix}
0 & 0 & 0 \\
-\beta A_2 A_1^T & 0 & 0 \\
-A_1^T & -A_2^T & 0 \\
0 & -\beta A_1 A_2^T & A_1 \\
0 & 0 & A_2 \\
0 & 0 & 0
\end{bmatrix},
\]

\[
U = \begin{bmatrix}
0 & -\beta A_1 A_2^T & A_1 \\
0 & 0 & A_2 \\
0 & 0 & 0
\end{bmatrix}.
\]

Then the matrix $E$ defined in (2.18) can be also written as

\[
E = R^{-1}(L + P + U).
\]

As a result, the iterative scheme in (2.17) can be rephrased as

\[
\rho^{k+1} = \mathcal{T}(R^{-1}L\rho^{k+1} + R^{-1}(P + U)\tilde{\rho}^k),
\]

where $\rho^k = (u^k, v^k, x^k)$. One can notice from equation (2.24) that $\rho^{k+1}$ is expressed in an implicit way, but can be computed explicitly as shown in (2.17).

**Lemma 2.9.** Let $\mathcal{T}$, $R$ be defined in (2.8), (2.10) and let $L$, $U$ be defined in (2.23), respectively. For $\rho_i = (u_i, v_i, x_i)$ and $\tilde{\rho}_i = (\tilde{u}_i, \tilde{v}_i, \tilde{x}_i)$, $i = 1, 2$, if the pairs $(\rho_i, \tilde{\rho}_i) \in \mathbb{H} \times \mathbb{H}$ satisfy

\[
\rho_i = \mathcal{T}(R^{-1}L\rho_i + R^{-1}(P + U)\tilde{\rho}_i),
\]

(2.25)
CHAPTER 2. COMPOSITE MINIMIZATION: PROXIMITY ALGORITHMS

then

\[
\langle \rho_2 - \rho_1, P[(\rho_2 - \tilde{\rho}) - (\rho_1 - \tilde{\rho}_1)] \rangle \leq (\gamma - \beta)\|A_1^T(u_2 - u_1) + A_2^T(v_2 - v_1)\|^2
\]

\[
+ \beta \langle A_1^T(u_2 - u_1), A_2^T[(v_2 - \tilde{v}_2) - (v_1 - \tilde{v}_1)] \rangle.
\]

Proof. Notice that \(\rho_i = T(E\rho_i + R^{-1}(P+U)(\tilde{\rho}_i - \rho_i))\). By identifying \((P+U)(\tilde{\rho}_i - \rho_i)\) as \(a_i\) in Lemma 2.8, we obtain the following

\[
\langle \rho_2 - \rho_1, P[(\rho_2 - \tilde{\rho}) - (\rho_1 - \tilde{\rho}_1)] \rangle \leq \langle \rho_2 - \rho_1, U[(\tilde{\rho}_2 - \rho_2) - (\tilde{\rho}_1 - \rho_1)] \rangle
\]

\[
- \beta \|A_1^T(u_2 - u_1) + A_2^T(v_2 - v_1)\|^2.
\]

Substituting \(U\) defined in (2.23) back to (2.26) and rearranging the terms yield

\[
\langle \rho_2 - \rho_1, P[(\rho_2 - \tilde{\rho}) - (\rho_1 - \tilde{\rho}_1)] \rangle \leq \langle A_1^T(u_2 - u_1), (\tilde{x}_2 - x_2) - (\tilde{x}_1 - x_1) \rangle
\]

\[
+ \langle A_2^T(v_2 - v_1), (\tilde{x} - x) - (\tilde{x}_1 - x_1) \rangle
\]

\[
- \beta \|A_1^T(u_2 - u_1) + A_2^T(v_2 - v_1)\|^2
\]

\[
+ \beta \langle A_1^T(u_2 - u_1), A_2^T[(v_2 - \tilde{v}_2) - (v_1 - \tilde{v}_1)] \rangle \tag{2.27}
\]

Further, equation (2.25) implies \(\tilde{x}_i - x_i = \gamma(A_1^T u_i + A_2^T v_i)\). Substituting this back in (2.27), we have

\[
\langle \rho_2 - \rho_1, P[(\rho_2 - \tilde{\rho}) - (\rho_1 - \tilde{\rho}_1)] \rangle \leq \gamma \|A_1^T(u_2 - u_1) + A_2^T(v_2 - v_1)\|^2
\]

\[
- \beta \|A_1^T(u_2 - u_1) + A_2^T(v_2 - v_1)\|^2 \tag{2.28}
\]

\[
+ \beta \langle A_1^T(u_2 - u_1), A_2^T[(v_2 - \tilde{v}_2) - (v_1 - \tilde{v}_1)] \rangle
\]

which completes the proof.

Based on the result in Lemma 2.9, we shall establish a relationship between the sequence \(\{(u_k, v^k, x^k) : k \in \mathbb{N}\}\) generated by the iterative scheme (2.17) and \(\hat{\rho} = \)
(\hat{u}, \hat{v}, \hat{x}) that satisfies fixed point equations (2.15). To this end, we introduce
\[
P_1 = \frac{1}{\alpha_1} I - \beta A_1 A_1^T \\
P_2 = \frac{1}{\alpha_2} I - \beta A_2 A_2^T
\] (2.29)

**Lemma 2.10.** Let $\alpha_1, \alpha_2, \beta, \gamma$ be positive, let $\hat{\rho} = (\hat{u}, \hat{v}, \hat{x}) \in \mathbb{H}$ be a solution to the fixed point equation (2.15), and let $\{\rho^k = (u^k, v^k, x^k) : k \in \mathbb{N}\}$ be the sequence generated by (2.17). If $\|A_1\|^2 < \frac{1}{\alpha_1 \beta}, \|A_2\|^2 < \frac{1}{\alpha_2 \beta}$, then the following equation holds:
\[
(\|\rho^{k+1} - \hat{\rho}\|^2 + \beta \|A_2^T (v^{k+1} - \hat{v})\|^2) - (\|\rho^k - \hat{\rho}\|^2 + \beta \|A_2^T (v^k - \hat{v})\|^2) \leq y^k, \quad (2.30)
\]
where
\[
y^k = -\|u^{k+1} - u^k\|^2_{P_1} - \|v^{k+1} - v^k\|^2_{P_2} - (\beta - \gamma) \|A_1^T u^{k+1} + A_2^T v^{k+1}\|^2 - \beta \|A_1^T u^{k+1} + A_2^T v^k\|^2.
\]

**Proof.** Since the positive parameters $\alpha_1, \alpha_2, \beta$ and $\gamma$ satisfy $\|A_1\|^2 < \frac{1}{\alpha_1 \beta}, \|A_2\|^2 < \frac{1}{\alpha_2 \beta}$, the matrices $P_1, P_2$ in (2.29) and $P$ in (2.18) are symmetric and positive definite. Notice that $\rho^{k+1}, \rho^k$ and $\hat{\rho}$ satisfy
\[
\rho^{k+1} = T(R^{-1} L \rho^{k+1} + R^{-1} (P + U) \rho^k),
\]
and
\[
\hat{\rho} = T(R^{-1} L \hat{\rho} + R^{-1} (P + U) \hat{\rho}).
\]

Identifying $\rho^{k+1}, \rho^k$, and $\hat{\rho}$, respectively, as $\rho_2, \hat{\rho}_2, \rho_1$ in Lemma 2.9 together with $A_1^T \hat{u} + A_2^T \hat{v} = 0$ leads to
\[
\langle \rho^{k+1} - \hat{\rho}, P(\rho^{k+1} - \rho^k) \rangle \leq (\gamma - \beta) \|A_1^T u^{k+1} + A_2^T v^{k+1}\|^2 + \beta \langle A_1^T (u^{k+1} - \hat{u}), A_2^T (v^{k+1} - v^k) \rangle.
\]
CHAPTER 2. COMPOSITE MINIMIZATION: PROXIMITY ALGORITHMS

Using the identity $2(\rho^{k+1} - \hat{\rho}, P(\rho^{k+1} - \rho)) = \|\rho^{k+1} - \hat{\rho}\|_P^2 - \|\rho - \hat{\rho}\|_P^2 + \|\rho^{k+1} - \rho\|_P^2$ and noticing that $\|\rho^{k+1} - \rho\|_P^2 = \|u^{k+1} - u\|_P^2 + \|v^{k+1} - v\|_P^2 + \frac{1}{\gamma}\|x^{k+1} - x\|^2$ and $x^{k+1} - x = -\gamma(A_1^T u^{k+1} + A_2^T v^{k+1})$, we obtain

$$\|\rho^{k+1} - \hat{\rho}\|_P^2 - \|\rho - \hat{\rho}\|_P^2 \leq -\|u^{k+1} - u\|_P^2 - \|v^{k+1} - v\|_P^2$$

$$+2(\gamma - \beta)\|A_1^T u^{k+1} + A_2^T v^{k+1}\|^2$$

$$+2\beta\langle A_1^T (u^{k+1} - \hat{u}), A_2^T (v^{k+1} - \hat{v}) \rangle. \quad (2.31)$$

It can be verified by using $A_1^T \hat{u} + A_2^T \hat{v} = 0$ that the sum of the last two terms in (2.31) equals to $-\beta\|A_1^T u^{k+1} + A_2^T v^{k+1}\|^2 - \beta\|A_2^T (v^{k+1} - \hat{v})\|^2 + \beta\|A_2^T (v^{k+1} - \hat{v})\|^2$. Therefore,

$$\left(\|\rho^{k+1} - \hat{\rho}\|_P^2 + \beta\|A_2^T (v^{k+1} - \hat{v})\|^2\right) - \left(\|\rho^k - \beta\hat{\rho}\|_P^2 + \|A_2^T (v^k - \hat{v})\|^2\right) \leq y^k. \quad (2.32)$$

This completes the proof of the result.

We are ready to prove the convergence of the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ generated by the iterative scheme (2.17).

**Theorem 2.11.** Let $\alpha_1, \alpha_2, \beta, \gamma$ be positive numbers and let $\{\rho^k = (u^k, v^k, x^k) : k \in \mathbb{N}\}$ be the sequence generated by scheme (2.17). If $\|A_1\|^2 < \frac{1}{\alpha_1\beta}, \|A_2\|^2 < \frac{1}{\alpha_2\beta}$, and $0 < \gamma \leq \beta$, then the sequence $\{(u^k, v^k, x^k) : k \in \mathbb{N}\}$ converges to a triple $\hat{\rho} = (\hat{u}, \hat{v}, \hat{x})$, a solution of (2.15).
CHAPTER 2. COMPOSITE MINIMIZATION: PROXIMITY ALGORITHMS

Proof. We will show \( \{(u^k, v^k, x^k) : k \in \mathbb{N}\} \) converges to a triple \((\hat{u}, \hat{v}, \hat{x})\) satisfying (2.15) by three steps. Firstly, by Lemma 2.10 we show \( \{\rho^k = (u^k, v^k, x^k) : k \in \mathbb{N}\} \) is bounded and therefore the sequence has a convergent subsequence. Next, we show that this convergent subsequence converges to a triple \( \hat{\rho} = (\hat{u}, \hat{v}, \hat{x}) \) satisfying (2.15). Finally, we show the entire sequence \( \{(u^k, v^k, x^k) : k \in \mathbb{N}\} \) converges to this triple.

If \( \|A_1\|_2^2 < \frac{1}{\alpha_1 \beta}, \|A_2\|_2^2 < \frac{1}{\alpha_2 \beta} \), then \( P_1, P_2, P \) are symmetric and positive definite. If \( 0 < \gamma \leq \beta \), the values of \( y^k \) in Lemma 2.10 is non-positive. Thus, from (2.30) we know that the sequence \( \{\|\rho^k - \hat{\rho}\|_P^2 + \|A_2^\top (v^k - \hat{v})\|^2 : k \in \mathbb{N}\} \) is nonincreasing and convergent. This implies the boundedness of the sequence \( \{\|\rho^k - \rho\|_P : k \in \mathbb{N}\} \).

Therefore, the sequence \( \{(u^k, v^k, x^k) : k \in \mathbb{N}\} \) is bounded. Hence, there exists a convergent subsequence \( \{(u^{k_i}, v^{k_i}, x^{k_i}) : i \in \mathbb{N}\} \) such that for some vector \((\tilde{u}, \tilde{v}, \tilde{x}) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \times \mathbb{R}^n\)

\[
\lim_{i \to \infty} (u^{k_i}, v^{k_i}, x^{k_i}) = (\tilde{u}, \tilde{v}, \tilde{x}) \tag{2.33}
\]

We shall show that \((\tilde{u}, \tilde{v}, \tilde{x})\) satisfies the fixed point equations (2.15). Summing (2.30) for \( k \) from 1 to infinity, we conclude that

\[
\|\rho^1 - \hat{\rho}\|_P^2 + \beta \|A_2^\top v^1 - \hat{v}\|^2 \geq \sum_{k=1}^{\infty} \|u^{k+1} - u^k\|_{P_1}^2 + \sum_{k=1}^{\infty} \|v^{k+1} - v^k\|_{P_2}^2 + \sum_{k=1}^{\infty} (\beta - \gamma) \|A_1^\top u^{k+1} + A_2^\top v^{k+1}\|_2^2
\]
The convergence of three series in the above inequality yield that
\[
\begin{align*}
\lim_{k \to \infty} u^{k+1} - u^k &= 0 \\
\lim_{k \to \infty} v^{k+1} - v^k &= 0 \\
\lim_{k \to \infty} A_1^T u^{k+1} + A_2^T v^{k+1} &= 0 \\
\lim_{k \to \infty} x^{k+1} - x^k &= \lim_{k \to \infty} -\gamma (A_1^T u^{k+1} + A_2^T v^{k+1}) = 0,
\end{align*}
\]
which particularly indicates
\[
\begin{align*}
\lim_{i \to \infty} u^{ki+1} - u^{ki} &= 0 \\
\lim_{i \to \infty} v^{ki+1} - v^{ki} &= 0 \\
\lim_{i \to \infty} A_1^T u^{ki+1} + A_2^T v^{ki+1} &= 0 \\
\lim_{i \to \infty} x^{ki+1} - x^{ki} &= 0
\end{align*}
\] (2.34)

By (2.33) and (2.34), we have that
\[
\begin{align*}
\lim_{i \to \infty} u^{ki+1} &= \tilde{u} \\
\lim_{i \to \infty} v^{ki+1} &= \tilde{v} \\
\lim_{i \to \infty} x^{ki+1} &= \tilde{x}
\end{align*}
\] (2.35)

In (2.17), the involved proximity operators and matrices are continuous operators.

Equations (2.33) and (2.35) imply that \((\tilde{u}, \tilde{v}, \tilde{x})\) satisfies (2.15).

Now, let us take \((\hat{u}, \hat{v}, \hat{x}) = (\tilde{u}, \tilde{v}, \tilde{x})\). Then from (2.33) we have that
\[
\lim_{i \to \infty} \left( \|\rho^{ki} - \hat{\rho}^i\|_P^2 + \beta \|A_2^T (v^{ki} - \hat{v})\|^2 \right) = 0.
\]

The monotonicity and convergence of the sequence \(\{\|\rho^k - \hat{\rho}\|_P^2 + \beta \|A_2^T (v^k - \hat{v})\|^2 : k \in \mathbb{N}\} \) imply that
\[
\lim_{k \to \infty} \left( \|\rho^k - \hat{\rho}\|_P^2 + \beta \|A_2^T (v^k - \hat{v})\|^2 \right) = 0.
\]
Thus, the sequence \( \{\rho^k = (u^k, v^k, x^k) : k \in \mathbb{N}\} \) converges to a triple \( \hat{\rho} = (\hat{u}, \hat{v}, \hat{x}) \) satisfying (2.15). This completes the proof of this theorem.

\[\square\]

## 2.5 Connections with Existing Algorithms

In this section, we point out the connections of our proposed algorithm with several well-known methods. Specifically, we would explore the connection of the proposed algorithm with Chambolle and Pock’s (CP) Primal-Dual method, Augmented Lagrangian Method (ALM) and Alternating Direction Method of Multipliers (ADMM).

To this end, we first consider a degenerated form of Algorithm 1 without Gauss-Seidel acceleration between \( u \) and \( v \) and with equal parameters \( \alpha_1 = \alpha_2 = \alpha \). This degenerated form is presented in Algorithm 2.

**Algorithm 2: Degenerated form of Algorithm 1**

**Input:** Initialization: \( u^0 \in \mathbb{R}^{m_1}, \ v^0 \in \mathbb{R}^{m_2}, \ x^0 \in \mathbb{R}^n; \) parameters \( \alpha, \beta, \gamma \).

**Result:** \( x^\infty \)

**while it is not convergent do**

\[
\begin{align*}
    u^{k+1} &= \text{prox}_{\alpha f_1^*}(u^k + \alpha A_1(x^k - \beta(A_1^T u^k + A_2^T v^k))) \\
    v^{k+1} &= \text{prox}_{\alpha f_2^*}(v^k + \alpha A_2(x^k - \beta(A_1^T u^k + A_2^T v^k))) \\
    x^{k+1} &= x^k - \gamma(A_1^T u^{k+1} + A_2^T v^{k+1})
\end{align*}
\]

(2.36)
By letting 

\[ w^k := (u^k; v^k), \quad A := \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \quad \text{and} \quad f^*(w) := f^*_1(u) + f^*_2(v), \]

we can rewrite (2.36) in a more compact form

\[
\begin{align*}
    w^{k+1} &= \text{prox}_{\alpha f^*}(w^k + \alpha A(x^k - \beta A^T w^k)) \\
    x^{k+1} &= x^k - \gamma A^T w^{k+1}
\end{align*}
\]

(2.37)

In the meantime, the fixed point equations corresponding to scheme (2.37) have the following form

\[
\begin{align*}
    w &= \text{prox}_{\alpha f^*}(w + \alpha A(x - \beta A^T w)) \\
    x &= x - \gamma A^T w
\end{align*}
\]

(2.38)

The fixed point equations (2.38) characterize a solution \( x \) to the following minimization problem

\[
\min \{ f(Ax) : x \in \mathbb{R}^n \},
\]

(2.39)

where \( f(w) \) is defined by \( f(w) := f_1(u) + f_2(v) \).

Next, we will show that we can specify scheme (2.37) as a special case of scheme (2.17) and therefore the convergence of scheme (2.37) follows automatically. To cast scheme (2.37) into scheme (2.17), we let

\[
    u = w, \quad f^*_1 = f^*, \quad f^*_2 = 0, \quad A_1 = A, \quad A_2 = 0, \quad \alpha_1 = \alpha
\]

(2.40)

in scheme (2.17). For such the choice of those quantities, we are able to rewrite
scheme (2.17) as

\[
\begin{aligned}
  w^{k+1} &= \text{prox}_{\alpha f^*}(w^k + \alpha A(x^k - \beta(A^Tw^k))) \\
  v^{k+1} &= v^k \\
  x^{k+1} &= x^k - \gamma(A^Tw^{k+1})
\end{aligned}
\]

(2.41)

from which one can notice that sequence \( \{v^k : k \in \mathbb{N}\} \) is a constant vector sequence.

By ignoring the trivial step involving \( v^{k+1} \), scheme (2.41) becomes scheme (2.37).

**Lemma 2.12.** Let \( \alpha, \beta, \gamma \) be positive, let \( \hat{\rho} = (\hat{w}, \hat{x}) \in H \) satisfy the fixed point equations (2.38), and let \( \{\rho^k = (w^k, x^k) : k \in \mathbb{N}\} \) be the sequence generated by (2.37).

Set

\[
Q := \frac{1}{\alpha}I - \beta AA^T, \quad P := \begin{bmatrix} Q & \frac{1}{\gamma}I \end{bmatrix}.
\]

If \( \|A\|^2 < \frac{1}{\alpha \beta} \), then

\[
\|\rho^{k+1} - \hat{\rho}\|_P^2 - \|\rho^k - \hat{\rho}\|_P^2 \leq -\|w^{k+1} - w^k\|_Q - (2\beta - \gamma)\|A^T(w^{k+1} - \hat{w})\|^2.
\]

(2.42)

**Proof.** This is an immediate result of Lemma 2.10 by specifying corresponding quantities as in (2.40) and noticing that \( v^{k+1} = v^k \) for such the choice of those quantities.

With Lemma 2.12, we can prove our result on the convergence of the sequence \( \{(w^k, x^k) : k \in \mathbb{N}\} \) generated by scheme (2.37).

**Theorem 2.13.** Let \( \alpha, \beta, \gamma \) be positive, let \( \hat{\rho} = (\hat{w}, \hat{x}) \in H \) satisfy the fixed point equations (2.38), and let \( \{\rho^k = (w^k, x^k) : k \in \mathbb{N}\} \) be the sequence generated by (2.37).

If \( \|A\|^2 < \frac{1}{\alpha \beta} \) and \( 0 < \gamma \leq 2\beta \) then the sequence \( \{(w^k, x^k) : k \in \mathbb{N}\} \) converges to a pair \( (\hat{w}, \hat{x}) \) satisfying (2.38).
CHAPTER 2. COMPOSITE MINIMIZATION: PROXIMITY ALGORITHMS

Proof. It follows the proof of Theorem 2.11 by specifying corresponding quantities in scheme (2.17) as in (2.40) and using Lemma 2.12.

To guarantee convergence, it is necessary for Algorithm 1 that \(0 < \alpha_1 \beta < \frac{1}{\|A_1\|^2}\) and \(0 < \alpha_2 \beta < \frac{1}{\|A_2\|^2}\), for Algorithm 2 that \(0 < \alpha \beta < \frac{1}{\|A_1; A_2\|^2}\). It can be noticed that \(\max\{\|A_1\|^2, \|A_2\|^2\} \leq \|A_1; A_2\|^2\), which implies \(\min\{\frac{1}{\|A_1\|^2}, \frac{1}{\|A_2\|^2}\} \geq \frac{1}{\|A_1; A_2\|^2}\). Hence, more flexibility exhibits for the choice of \(\alpha_1, \alpha_2, \beta\) in Algorithm 1 than for the choice of \(\alpha, \beta\) in Algorithm 2.

2.5.1 Connection with Chambolle and Pock’s Algorithm

First of all, let us review Chambolle and Pock’s (CP) algorithm [15] for solving the following optimization problem

\[
\min\{f(Ax) + g(x) : x \in \mathbb{R}^n\},
\]

where \(f \in \Gamma_0(\mathbb{R}^m), g \in \Gamma_0(\mathbb{R}^n),\) and \(A\) is a matrix of size \(m \times n\). We assume that model (2.43) has a minimizer. The CP algorithm proposed in [15] for model (2.43) can be written as

\[
\begin{align*}
(w^{k+1}, x^{k+1}, \bar{x}^{k+1}) &= \left(\text{prox}_{\sigma f^*}(w^k + \sigma A\bar{x}^k), \right. \\
&\left. \quad \text{prox}_{\tau g}(x^k - \tau A^T w^{k+1}), \right. \\
&\left. \quad 2x^{k+1} - x^k. \right)
\end{align*}
\]

For any initial guess \((x^0, \bar{x}^0, w^0) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m\), the sequence \(\{(x^k, w^k) : k \in \mathbb{N}\}\) converges as long as \(0 < \sigma \tau < \|A\|^{-2}\).
CHAPTER 2. COMPOSITE MINIMIZATION: PROXIMITY ALGORITHMS

In particular, when we set \( g = 0 \), a direct computation shows that \( \text{prox}_{\tau g} \) is the identity operator for any \( \tau > 0 \). Set \( \alpha = \sigma \) and \( \beta = 2\tau \). Accordingly, the general CP method in (2.44) becomes

\[
\begin{align*}
    w^{k+1} &= \text{prox}_{\alpha f^*} \left( w^k + \alpha A \left( x^{k-1} - \beta A^T w^k \right) \right), \\
x^{k+1} &= x^k - \frac{\beta}{2} A^T w^{k+1}.
\end{align*}
\]  
(2.45)

On the other hand, when we set \( g = 0 \), model (2.43) reduces to model (2.39). Our algorithm for model (2.39) is presented in scheme (2.37).

Therefore, by comparing the CP algorithm and the scheme (2.37) for model (2.39), we can see that the CP algorithm uses \( x^{k-1} \) while the scheme (2.37) uses \( x^k \) in the computation of \( w^{k+1} \). Further, the step size of the CP algorithm for updating \( x^{k+1} \) is fixed as \( \frac{\beta}{2} \) while it can be any number in \((0, 2\beta]\) for the scheme (2.37). Although, the relation \( 0 < \alpha \beta < 2\|A\|^{-2} \) is required for the CP algorithm while the relation \( 0 < \alpha \beta < \|A\|^{-2} \) is needed for the scheme (2.37), for a fixed \( \alpha \), we can choose the step size for the scheme (2.37) twice bigger than that for the CP algorithm.

2.5.2 Connection with Augmented Lagrangian Methods

As discussed earlier, a reduced iterative scheme (2.36) from Algorithm 1 can be written in a compact form of (2.37). Notice that the first step involving the proximity operator \( \text{prox}_{\alpha f^*} \) is equivalent to find the minimizer of a minimization problem.
can verify that (2.37) is equivalent to the following iterative scheme

\[
\begin{align*}
    w^{k+1} &= \arg\min \left\{ f^*(w) - \langle x^k, A^T w \rangle + \frac{\beta}{2} \| A^T w \|^2 + \frac{1}{2} \| w - w^k \|^2_Q : w \in \mathbb{R}^m \right\} \\
    x^{k+1} &= x^k - \gamma A^T w^{k+1}
\end{align*}
\]  

(2.46)

where \( Q = \frac{1}{\alpha} I - \beta AA^T \) is a positive definite matrix. The condition \( \alpha \beta < \frac{1}{\| A \|^2} \) ensures the positive definiteness of \( Q \). In the literature of nonlinear programming [5], augmented Lagrangian methods (ALMs) are often used to convert a constrained optimization problem to an unconstrained one by adding the objective function a penalty term associated with the constraints. If we choose \( Q = 0 \) and \( \gamma = \beta \) in (2.46), it reduces to the augmented Lagrangian method:

\[
\begin{align*}
    w^{k+1} &= \arg\min \left\{ f^*(w) - \langle x^k, A^T w \rangle + \frac{\beta}{2} \| A^T w \|^2 : w \in \mathbb{R}^m \right\} \\
    x^{k+1} &= x^k - \beta A^T w^{k+1}
\end{align*}
\]  

(2.47)

Even though we can assume that the proximity operator of \( f \) has a closed form, there is lack of an effective way to update \( w^{k+1} \) in (2.47) when \( A \) is not the identity matrix. However, the vector \( w^{k+1} \) in the scheme (2.46) can be effectively updated once a proper \( Q \) is chosen. This essentially illustrates that Algorithm 2 is superior to the ALM from the numerical implementation point of view.
2.5.3 Connection with Alternating Direction Method of Multipliers

Similarly, the iterative scheme (2.17) in Algorithm 1 can be cast as a special case of the following scheme

\[
\begin{align*}
 u^{k+1} &= \arg\min_u \left\{ f_1^*(u) + f_2^*(v^k) - \langle x^k, A_1^T u + A_2^T v^k \rangle \\
 &\quad \quad + \frac{\beta}{2} \| A_1^T u + A_2^T v^k \|^2 + \frac{1}{2} \| u - u^k \|^2_{Q_1} : u \in \mathbb{R}^{m_1} \right\} \\
 v^{k+1} &= \arg\min_v \left\{ f_1^*(u^{k+1}) + f_2^*(v) - \langle x^k, A_1^T u^{k+1} + A_2^T v \rangle \\
 &\quad \quad + \frac{\beta}{2} \| A_1^T u^{k+1} + A_2^T v \|^2 + \frac{1}{2} \| v - v^k \|^2_{Q_2} : v \in \mathbb{R}^{m_2} \right\} \\
 x^{k+1} &= x^k - \gamma (A_1^T u^{k+1} + A_2^T v^{k+1})
\end{align*}
\] (2.48)

where \( Q_1 = \frac{1}{\alpha_1} I - \beta A_1 A_1^T \), \( Q_2 = \frac{1}{\alpha_2} I - \beta A_2 A_2^T \) are positive definite matrices. The positive definiteness of \( Q_1 \) and \( Q_2 \) will be guaranteed under the conditions \( 0 < \alpha_1 \beta < \frac{1}{\| A_1 \|^2} \) and \( 0 < \alpha_2 \beta < \frac{1}{\| A_2 \|^2} \). If \( Q_1 \) and \( Q_2 \) are taken as zero matrices and \( \gamma = \beta \), the scheme (2.48) reduces to the alternating direction method of multipliers (ADMM):

\[
\begin{align*}
 u^{k+1} &= \arg\min_u \left\{ f_1^*(u) + f_2^*(v^k) - \langle x^k, A_1^T u + A_2^T v^k \rangle \\
 &\quad \quad + \frac{\beta}{2} \| A_1^T u + A_2^T v^k \|^2 : u \in \mathbb{R}^{m_1} \right\} \\
 v^{k+1} &= \arg\min_v \left\{ f_1^*(u^{k+1}) + f_2^*(v) - \langle x^k, A_1^T u^{k+1} + A_2^T v \rangle \\
 &\quad \quad + \frac{\beta}{2} \| A_1^T u^{k+1} + A_2^T v \|^2 : v \in \mathbb{R}^{m_2} \right\} \\
 x^{k+1} &= x^k - (A_1^T u^{k+1} + A_2^T v^{k+1})
\end{align*}
\] (2.49)

Similar to what we have observed for the ALM, solving the two optimization problems in (2.49) is still challenging in general when both \( A_1 \) and \( A_2 \) are not the identity matrix. However, the vectors \( u^{k+1} \) and \( v^{k+1} \) in the scheme (2.48) can be effectively
updated once $Q_1$ and $Q_2$ are properly chosen. Hence our Algorithm 1 is superior to the ADMM from the numerical implementation point of view.
Chapter 3

Computing the Proximity Operator of the $\ell_p$-Norm

3.1 Introduction

The notion of sparsity has been widely explored recently in compressed sensing, matrix completion, machine learning, and image recovery. Typically, the sparsity of a signal is characterized by the $\ell_0$-norm of the signal that is essentially the number of non-zero components in the signal. Due to the non-convexity, it is often relaxed to the $\ell_1$-norm which is convex and can promote sparsity as well. Seeking a solution to a problem via the $\ell_1$-regularization has become the focus of attention of a massive volume of research in the context of compressed sensing. The usefulness of the $\ell_0$- and $\ell_1$-norm in sparsity-aware applications comes from the fact that their proximity
operators have explicit forms and can be implemented easily. The concrete forms of
the proximity operators of the $\ell_0$- and $\ell_1$-norm will be given in the next section. The
proximity operator, introduced early in [56], is a useful and convenient tool in character-
ing the solutions of optimization problems and developing iterative algorithms
for finding them. Some recent applications of the proximity operator in signal and
image processing can be found in [21, 29, 47, 45, 48, 49, 54, 61, 62, 67, 68] and the
references therein.

Our main interest is to study the proximity operator of the $\ell_p$-norm with $0 < p < 1$. The $\ell_p$-
regularization has been introduced in existing work. In [7], the $\ell_p$-
regularization was introduced for image reconstruction. In [53, 57, 69], the $\ell_p$-norm
is naturally involved in statistically modeling the wavelet coefficients of an image.
Particularly, a popular generalized Gaussian distribution (GGD) of the form $P(x) \sim
\exp(-|x/s|^p)$ is often adopted with the value for $p$ typically being in the range of
$[1/2, 1]$. Therefore, it is highly needed to compute the proximity operator of the
$\ell_p$-norm with $0 < p < 1$.

Unfortunately, the proximity operator of the $\ell_p$-norm with $0 < p < 1$ does not
have an explicit form except a few of $p$ values. In [44], finding the proximity operator
of the $\ell_p$-norm for $p = 1/2$ (or $2/3$) is formulated as finding a root of a corresponding
cubic (or quartic) polynomial. In the approach proposed in [44], all the roots of the
polynomial should be computed and then compared to select a proper root by some
discriminate conditions. Recently, the closed-form of the proximity operator of the $\ell_p$-
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

The $\ell_p$-norm was reported in [75] for $p = 1/2$ and in [14] for $p = 2/3$. Applications in image deconvolution with $\ell_p$-regularization ($p = 1/2, 2/3$) were also reported in [14, 44, 75].

For our contribution, we have a systematic study on the computation of the proximity operator of the $\ell_p$ norm with $0 < p < 1$. The properties of the proximity operator of the $\ell_p$ norm are presented by analyzing the objective function of optimization problem associated with the proximity operator. By using these properties, the closed-form of the proximity operator of the $\ell_p$ ($p = 1/2, 2/3$) norm is given accompanying with an alternative, but simple, proof in comparison with that given in [14, 75]. For computing the proximity operator of the $\ell_p$ norm with $p$ not being $0, 1/2, 2/3, 1$, we need to solve a nonlinear equation associated with the proximity operator. We propose to use Newton’s method to solve this equation. To make Newton’s method efficiently, the initial estimate for Newton’s method requires to be very close to the true solution of the equation. We suggest a way to locate this initial estimate by exploiting the availability of the proximity operators of the $\ell_{0^{-}}, \ell_{1/2^{-}}, \ell_{2/3^{-}},$ and $\ell_1$-norm.

The outline of this part is as follows. In section 3.2 we present the properties of the proximity operator of the $\ell_p$-norm with $0 < p < 1$. In section 3.3, we give the explicit forms of the proximity operator of the $\ell_p$-norm for $p = 1/2$ and $p = 2/3$. In section 3.4, we apply Newton’s method to develop a numerical algorithm to compute the proximity operator of the $\ell_p$-norm for $0 < p < 1$. 
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

3.2 Properties of the Proximity Operator of the $\ell_p$-Norm

For a vector $x = (x_1, \ldots, x_d)^\top$ in $\mathbb{R}^d$, its $\ell_0$-norm $\|x\|_0$ is simply the number of nonzero entries in $x$ and its $\ell_p$-norm for $p > 0$ is defined by $\|x\|_p := \left(\sum_{i=1}^{d} |x_i|^p\right)^{1/p}$. Note that $\|\cdot\|_p$ is only a quasi-norm for $0 < p < 1$. The proximity operator of the $\ell_p$-norm with index $\mu > 0$ at $x \in \mathbb{R}^d$ is a set-valued operator from $\mathbb{R}^d \to 2^{\mathbb{R}^d}$, with $2^{\mathbb{R}^d}$ denoting the collection of all sets of vectors in $\mathbb{R}^d$, and is defined as

$$\text{prox}_{\mu\|\cdot\|_p}(x) := \arg\min_{u \in \mathbb{R}^d} \left\{ \mu\|u\|_p^p + \frac{1}{2}\|u - x\|_2^2 \right\}.$$  (3.1)

Here, $\|\cdot\|_0^0$ should be understood as $\|\cdot\|_0$ when $p = 0$. By the definition of the proximity operator, we have that

$$\text{prox}_{\mu\|\cdot\|_p}(x) = \text{prox}_{\mu\|\cdot\|_p}(x_1) \times \cdots \times \text{prox}_{\mu\|\cdot\|_p}(x_d),$$  (3.2)

for $x \in \mathbb{R}^d$. Therefore, in order to compute the proximity operator of the $\ell_p$-norm we only need to compute $\text{prox}_{\mu\|\cdot\|_p}$ the proximity operator of the function $|\cdot|^p$ in $\mathbb{R}$. To simplify our notation in the rest of discussion, we set

$$\mathcal{T}_{\mu,p} := \text{prox}_{\mu\|\cdot\|_p}.$$

The proximity operators $\mathcal{T}_{\mu,0}$ and $\mathcal{T}_{\mu,1}$ are the well-known hard- and soft-thresholding
operators, respectively. Both have closed forms at $x \in \mathbb{R}$ as follows:

$$T_{\mu,0}(x) = \begin{cases} 
\{0\}, & \text{if } |x| < \sqrt{2\mu}; \\
\{0, x\}, & \text{if } |x| = \sqrt{2\mu}; \\
\{x\}, & \text{otherwise.}
\end{cases} \quad (3.3)$$

$$T_{\mu,1}(x) = \operatorname{sign}(x) \cdot \max\{|x| - \mu, 0\}. \quad (3.4)$$

Explicit forms of $T_{\mu,p}$ for $p = 1/2$ and $2/3$ were discussed in [44]. The proximity operator $T_{\mu,p}$ was also studied in [75].

For arbitrary $p \geq 0$ and $\mu > 0$, the operator $T_{\mu,p}$ at $x \in \mathbb{R}$ is the collection of the minimizers of the function

$$J_{\mu,p,x}(u) := \mu|u|^p + \frac{1}{2}(u - x)^2 \quad (3.5)$$

over $\mathbb{R}$. That is,

$$T_{\mu,p}(x) := \operatorname{argmin}\{J_{\mu,p,x}(u) : u \in \mathbb{R}\}. \quad (3.6)$$

Since the function $J_{\mu,p,x}$ is continuous for $p > 0$ and lower semi-continuous for $p = 0$ on $\mathbb{R}$ and $\lim_{|u| \to +\infty} J_{\mu,p,x}(u) = +\infty$, then there exists $u_*$ such that $J_{\mu,p,x}(u_*) \leq J_{\mu,p,x}(u)$ for all $u \in \mathbb{R}$. That is, $u_* \in T_{\mu,p}(x)$, equivalently, the set $T_{\mu,p}(x)$ is non-empty for any $x$.

In the rest of this section, we will present the properties of the proximity operator $T_{\mu,p}$ for $p \geq 0$.

**Lemma 3.1.** For any $p \geq 0$ and $\mu > 0$, it holds that $T_{\mu,p}(-x) = -T_{\mu,p}(x)$ for all $x \in \mathbb{R}$. 
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

Proof. We first prove that $\mathcal{T}_{\mu,p}(-x) \subseteq -\mathcal{T}_{\mu,p}(x)$. By definition (3.5), the relation $J_{\mu,p,-x}(u) = J_{\mu,p,x}(-u)$ holds for $u \in \mathbb{R}$. Now, suppose that $u_* \in \mathcal{T}_{\mu,p}(-x)$. Thus $J_{\mu,p,-x}(u_*) \leq J_{\mu,p,x}(-u)$ for all $u \in \mathbb{R}$. With the help of the above relation, this inequality is equivalent to $J_{\mu,p,x}(-u) \leq J_{\mu,p,x}(-u_*)$ which implies $-u_* \in \mathcal{T}_{\mu,p}(x)$. That is, $u_* \in -\mathcal{T}_{\mu,p}(x)$. All above arguments are reversible, therefore we can show that $-\mathcal{T}_{\mu,p}(x) \subseteq \mathcal{T}_{\mu,p}(-x)$. This completes the proof. \qed

With Lemma 3.1, it will be sufficient to study the set $\mathcal{T}_{\mu,p}(x)$ for all non-negative $x$. Specifically, for $x = 0$, we can straightforwardly derive that for all $p \geq 0$ and $\mu > 0$

$$\mathcal{T}_{\mu,p}(0) = \{0\}. \quad (3.7)$$

For $x$ being positive, the following lemma characterizes the elements in the set $\mathcal{T}_{\mu,p}(x)$.

Lemma 3.2. For $p \geq 0$, $\mu > 0$ and $x > 0$, then every non-zero element in $\mathcal{T}_{\mu,p}(x)$ is positive and its value is less or equal to $x$. Moreover, assume that the set $\mathcal{T}_{\mu,p}(x)$ has a non-zero element, say, $u_*$, then $u_* = x$ if $p = 0$; and $u_* < x$ if $p > 0$.

Proof. Suppose that $u_*$ is in $\mathcal{T}_{\mu,p}(x)$ and is non-zero. If $u_*$ is negative, then $|u_* - x| < |u_* - x|$ which yields $J_{\mu,p,x}(-u_* \neq J_{\mu,p,x}(u_*)$. Thus, $u_*$ is not in $\mathcal{T}_{\mu,p}(x)$, which contradicts our assumption. Hence, $u_*$ must be positive. We further show that $u_* \leq x$. If it is not true, that is $u_* > x$. We then define $\bar{u} := 2x - u_*$. It can be verified directly that $|\bar{u}| < u_*$ and $|\bar{u} - x| = |u_* - x|$. Therefore, we have $J_{\mu,p,x}(\bar{u}) < J_{\mu,p,x}(u_*)$. This implies that $u_*$ is not in $\mathcal{T}_{\mu,p}(x)$, thus, contradicts our assumption again. Hence, $u_* \leq x$. 
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

As we know that $T_{\mu,0}$ is the hard-thresholding operator, by equation (3.3), every non-zero element $u_*$ in the set $T_{\mu,0}(x)$ is identical to $x$.

Finally, we prove that $u_* < x$ for $p > 0$. Since $u_* \leq x$, we only need to show that $x$ is not in the set $T_{\mu,p}(x)$. Actually it follows from the fact that $J'_{\mu,p,x}(x) = \mu px^{p-1} > 0$ for any positive number $x$.

We remark that for $p > 1$, the set $T_{\mu,p}(x)$ contains only one element and this element is non-zero if and only if $x$ is non-zero. By Lemma 3.1 and equation (3.7), we consider the case of $x$ being positive. Since the function $J_{\mu,p,x}$ is strictly convex and coercive, the set $T_{\mu,p}(x)$ has a unique element. We further notice that $J'_{\mu,p,x}(0) = -x < 0$. Hence, the element cannot be zero. Hence, we can view $T_{\mu,p}$ as an operator from $\mathbb{R}$ to $\mathbb{R}$. This is a shrinkage, but not sparse-promoting, operator in the sense that $0 \neq |T_{\mu,p}(x)| < |x|$ for any non-zero number $x$.

The situation, however, is completely different for $p = 0$ and $p = 1$ as indicated by the hard- and soft-thresholding operators, respectively. Therefore, we turn our attention for the operator $T_{\mu,p}$ with $0 < p < 1$.

For convenience, given any $\mu > 0$ and $0 < p < 1$, we define

$$\varpi_{\mu,p} := (\mu p(1-p))^{\frac{1}{2-p}}, \quad \tilde{\tau}_{\mu,p} := (2-p)(\mu p)^{\frac{1}{2-p}}(1-p)^{\frac{2}{2-p}}.$$  \hspace{1cm} (3.8)

and

$$\tau_{\mu,p} := \frac{2-p}{2(1-p)}(2\mu(1-p))^{\frac{1}{2-p}}, \quad \varrho_{\mu,p} := (2\mu(1-p))^{\frac{1}{2-p}}.$$  \hspace{1cm} (3.9)

We first study the convexity of the function $J_{\mu,p,x}$ on the interval $[0, +\infty)$ for positive $x$. For our convenience, the first and the second derivatives of $J_{\mu,p,x}$ at $u > 0$
are given as follows:

\[ J'_{\mu,p,x}(u) = u + \mu pu^{p-1} - x \quad \text{and} \quad J''_{\mu,p,x}(u) = 1 + \mu p(p - 1)u^{p-2}. \]

**Lemma 3.3.** For any fixed \(0 < p < 1\), \(\mu > 0\), and \(x > 0\), the following statements hold for the function \(J_{\mu,p,x}\).

(i) \(J_{\mu,p,x}(u)\) is concave on \([0, \tilde{\tau}_{\mu,p}]\) and convex on \([\tilde{\tau}_{\mu,p}, +\infty)\);

(ii) If \(x \leq \tilde{\tau}_{\mu,p}\), then \(J_{\mu,p,x}\) is increasing on \([0, +\infty)\);

(iii) If \(x > \tilde{\tau}_{\mu,p}\), then \(J'_{\mu,p,x}(\cdot)\) has exactly two roots, namely, \(u_-\) and \(u_+\), on the interval \((0, +\infty)\). Moreover, the roots satisfy the inequality \(u_- < \overline{\omega}_{\mu,p} < u_+\), the function \(J_{\mu,p,x}(u)\) has a local maximum at \(u = u_-\) and a local minimum at \(u = u_+\).

**Proof.** Item (i): The function \(J_{\mu,p,x}\) is continuous on \([0, \infty)\). One can directly verify that \(J''_{\mu,p,x}(\overline{\omega}_{\mu,p}) = 0\), \(J''_{\mu,p,x}(u) < 0\) for \(u \in (0, \overline{\omega}_{\mu,p})\), and \(J''_{\mu,p,x}(u) > 0\) for \(u \in (\overline{\omega}_{\mu,p}, +\infty)\). Hence, the function \(J_{\mu,p,x}(u)\) having the inflation point at \(u = \overline{\omega}_{\mu,p}\), is concave on \([0, \overline{\omega}_{\mu,p}]\) and convex on \([\overline{\omega}_{\mu,p}, +\infty)\).

Item (ii): It suffices to show \(J'_{\mu,p,x}(u) \geq 0\) on \((0, \infty)\). Define \(h(u) := J'_{\mu,p,x}(u)\). One can check that \(h\) is convex on \((0, \infty)\) and has unique global minimizer at \(u = \overline{\omega}_{\mu,p}\) with the minimal value \(h(\overline{\omega}_{\mu,p}) = \tilde{\tau}_{\mu,p} - x\). Hence, \(h(u) \geq h(\overline{\omega}_{\mu,p}) \geq 0\).

Item (iii): With the function \(h\) defined in the above, one can verify that \(\lim_{u \to 0^+} h(u) = +\infty\), \(\lim_{u \to +\infty} h(u) = +\infty\), and \(h(\overline{\omega}_{\mu,p}) = \tilde{\tau}_{\mu,p} - x > 0\). Further, it can be verified that \(h'(u) < 0\) for \(u \in (0, \overline{\omega}_{\mu,p})\) and \(h'(u) > 0\) for \(u \in (\overline{\omega}_{\mu,p}, +\infty)\), that is, \(h\) is decreasing.
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

on $(0, \varpi_{\mu, p})$ and increasing on $(\varpi_{\mu, p}, +\infty)$. Putting all above together and using the mean value theorem, there exist a unique number $u_- \in (0, \varpi_{\mu, p})$ and a unique number $u_+ \in (\varpi_{\mu, p}, +\infty)$ such that $h(u_-) = h(u_+) = 0$. From item (i), one gets that $J_{\mu, p, x}(u)$ has a local maximum at $u = u_-$ and a local minimum at $u = u_+$. \hfill \Box

In what follows, we will focus on the relationship between $x$ and $T_{\mu, p}(x)$.

Proposition 3.4. Let $x$ be a positive number. For any fixed $0 < p < 1$ and $\mu > 0$, $J_{\mu, p, x}(u)$ defined on $[0, +\infty)$ attains its global minimum at $u = 0$ if and only if $x \leq \tau_{\mu, p}$. In particular, $u = 0$ is the unique minimizer of $J_{\mu, p, x}(u)$ on $[0, \infty)$ if and only if $x < \tau_{\mu, p}$.

Proof. Assume $J_{\mu, p, x}(0) \leq J_{\mu, p, x}(u)$ for all $u > 0$, that is, $0 \leq \mu u^p + \frac{1}{2} u^2 - u x$ which is the same as $0 \leq \mu u^{p-1} + \frac{1}{2} u - x$. Define $g(u) := \mu u^{p-1} + \frac{1}{2} u$. Clearly the function $g$ is strictly convex on $(0, +\infty)$ and achieves its minimum value at $u = \varrho_{\mu, p}$ over this interval. The minimum value $g(\varrho_{\mu, p})$ is equal to $\tau_{\mu, p}$ which should be greater than or equal to $x$.

Conversely, $x \leq \tau_{\mu, p}$ implies that $u = 0$ is a minimizer of $J_{\mu, p, x}(u)$ since the above arguments are reversible.

The equivalence between the uniqueness and the strict inequality follows in the same manner by replacing “$\leq$” with “$<$” in the proof. \hfill \Box

Note that $\tilde{\tau}_{\mu, p} < \tau_{\mu, p}$ for any $0 < p < 1$ and $\mu > 0$. One knows from item (ii) of
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

Lemma 3.3 that $J_{\mu,p,x}(u)$ achieves its global minimum only at $u = 0$ when $x \leq \tilde{\tau}_{\mu,p}$.

This observation is further confirmed by Proposition 3.4.

For $x > \tilde{\tau}_{\mu,p}$, from item (iii) of Lemma 3.3 we know that $J_{\mu,p,x}(u)$ has its local minimums at $u = 0$ and $u = u_+$. We will then determine which one provides the global minimum of $J_{\mu,p,x}$. It turns out from Proposition 3.4 that if $\tilde{\tau}_{\mu,p} < x < \tau_{\mu,p}$, then the global minimum of $J_{\mu,p,x}$ achieves only at $u = 0$, i.e., $T_{\mu,p}(x) = \{0\}$. The cases of $x = \tau_{\mu,p}$ and $x > \tau_{\mu,p}$ will be studied in the following results.

**Proposition 3.5.** For any $0 < p < 1$, $\mu > 0$, if $x = \tau_{\mu,p}$ then $u_+ = \varrho_{\mu,p}$ and the function $J_{\mu,p,x}$ attains its global minimizers at both $u = 0$ and $u = u_+$. That is, $T_{\mu,p}(x) = \{0, u_+\}$.

**Proof.** It can be directly seen that $\varpi_{\mu,p} < \varrho_{\mu,p}$ and $J'_{\mu,p,x}(\varrho_{\mu,p}) = 0$. By Lemma 3.3, one has $u_+ = \varrho_{\mu,p}$. Further, one computes $J_{\mu,p,x}(u_+) = J_{\mu,p,x}(0) = \frac{1}{2}\tau_{\mu,p}^2$. Hence, $J_{\mu,p,x}$ achieves its minima at both $u = 0$ and $u = u_+$. □

**Proposition 3.6.** For any $0 < p < 1$, $\mu > 0$, if $x > \tau_{\mu,p}$ then the function $J_{\mu,p,x}$ attains its global minimum at $u = u_+$. That is, $T_{\mu,p}(x) = \{u_+\}$.

**Proof.** By Lemma 3.3, the function $J_{\mu,p,x}$ has local minima at both $u = 0$ and $u = u_+$. By Proposition 3.4, one can conclude that $J_{\mu,p,x}$ has its global minimum at $u = u_+$. This completes the proof. □

Table 3.1 presents the properties of the function $J_{\mu,p,x}$ for $\mu > 0$, $0 < p < 1$, and $x \geq 0$. Figure 3.1 provides plots of the objective functions $J_{\mu,p,x}$ for various values
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

of $x$. More precisely, Figure 3.1(a) presents an instance of item (ii) in Lemma 3.3; Figure 3.1(b) presents an instance of item (iii) in Lemma 3.3 and Proposition 3.4; Figure 3.1(c) reflects the situation of Proposition 3.5; and Figure 3.1(d) gives an example of Proposition 3.6.

Table 3.1: The properties of the function $J_{\mu,p,x}$ for $\mu > 0$, $0 < p < 1$, and $x \geq 0$. “l-min” and “g-min” stand for the local minimum and global minimum, respectively.

<table>
<thead>
<tr>
<th>$u \in [0, \bar{\omega}_{\mu,p}]$</th>
<th>$u \in [\bar{\omega}_{\mu,p}, +\infty)$</th>
<th>$u = 0$</th>
<th>$u = u_-$</th>
<th>$u = u_+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \in [0, \hat{\tau}_{\mu,p}]$</td>
<td>concave</td>
<td>convex</td>
<td>g-min</td>
<td>–</td>
</tr>
<tr>
<td>$x \in (\hat{\tau}<em>{\mu,p}, \tau</em>{\mu,p})$</td>
<td>concave</td>
<td>convex</td>
<td>g-min</td>
<td>l-max</td>
</tr>
<tr>
<td>$x = \tau_{\mu,p}$</td>
<td>concave</td>
<td>convex</td>
<td>g-min</td>
<td>l-max</td>
</tr>
<tr>
<td>$x \in (\tau_{\mu,p}, +\infty)$</td>
<td>concave</td>
<td>convex</td>
<td>l-min</td>
<td>l-max</td>
</tr>
</tbody>
</table>

Figure 3.1: The curves of $J_{\mu,p,x}$ for different choices of $x$: (a) $x \leq \hat{\tau}_{\mu,p}$ ($x = 0.9\hat{\tau}_{\mu,p}$); (b) $\hat{\tau}_{\mu,p} < x < \tau_{\mu,p}$ ($x = \hat{\tau}_{\mu,p} + 0.8(\tau_{\mu,p} - \hat{\tau}_{\mu,p})$); (c) $x = \tau_{\mu,p}$; and (d) $x > \tau_{\mu,p}$ ($x = 1.2\tau_{\mu,p}$).
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE \( \ell_p \)-NORM

From Table 3.1, Lemma 3.1, Propositions 3.4, 3.5, and 3.6, we conclude that

\[
T_{\mu,p}(x) = \begin{cases} 
0, & \text{if } |x| < \tau_{\mu,p}; \\
\{\text{sign}(x) \cdot \varrho_{\mu,p}\}, & \text{if } |x| = \tau_{\mu,p}; \\
\{\text{sign}(x) \cdot u_+\}, & \text{otherwise},
\end{cases}
\]

(3.10)

where \( u_+ \) is the largest zero of the first derivative function \( J'_{\mu,p,|x|} \).

From equation (3.10) one can see that \( T_{\mu,p} \) is a sparse-promoting thresholding operator with threshold \( \tau_{\mu,p} \). The monotonicity of the threshold \( \tau_{\mu,p} \) with respect to \( p \) is described in the following.

**Proposition 3.7.** If \( 0 < \mu \leq \frac{1}{2} \), \( \tau_{\mu,p} \) as a function of \( p \) for a fixed \( \mu \) is decreasing on the interval \((0, 1)\); if \( \mu > \frac{1}{2} \), it is increasing on the interval \((0, 1 - \frac{1}{2\mu})\) and decreasing on the interval \((1 - \frac{1}{2\mu}, 1)\).

**Proof.** Let us define \( f(p) := \ln(\tau_{\mu,p}) \). Then the monotonicity of \( f(p) \) is consistent with that of \( \tau_{\mu,p} \) as a function of \( p \). By the definition of \( \tau_{\mu,p} \) in (3.9), we have that \( f'(p) = \frac{1}{(2-p)^2} \ln[2\mu(1-p)] \). If \( \mu \leq \frac{1}{2} \), then \( f'(p) < 0 \) on \((0, 1)\). Therefore \( \tau_{\mu,p} \) is decreasing on \((0, 1)\) with respect to \( p \). If \( \mu > \frac{1}{2} \), we have that \( f'(p) > 0 \) on \((0, 1 - \frac{1}{2\mu})\) and \( f'(p) < 0 \) on \((1 - \frac{1}{2\mu}, 1)\). Therefore, \( \tau_{\mu,p} \) is increasing on \((0, 1 - \frac{1}{2\mu})\) and decreasing on \((1 - \frac{1}{2\mu}, 1)\) with respect to \( p \).

Figure 3.2 and Figure 3.3 display the proximity operators \( T_{\mu,p} \) for various values of \( p \) and \( \mu \). Figure 3.2 depicts the proximity operators \( T_{1/3,p} \) for \( p = 0, \frac{1}{2}, \frac{2}{3}, \frac{4}{5}, 1 \) while Figure 3.3 depicts the proximity operators \( T_{3,p} \) for \( p = 0, \frac{1}{2}, \frac{2}{3}, \frac{4}{5}, \frac{9}{10}, 1 \). The blue curves
connecting point \((\tau_{\mu,p}, \varrho_{\mu,p})\) (or \((-\tau_{\mu,p}, -\varrho_{\mu,p})\)) on black curve and point \((\tau_{\mu,p}, 0)\) (or \((-\tau_{\mu,p}, 0)\)) on red curve in both Figure 3.2 and Figure 3.3 capture the evolution of the \((\tau_{\mu,p}, \varrho_{\mu,p})\) as \(p\) changes from 0 to 1. The evolution curve of \((\tau_{\mu,p}, \varrho_{\mu,p})\) also validates the statements in Proposition 3.7. Therefore, the main issue is to compute \(T_{\mu,p}(x)\) for \(x > \tau_{\mu,p}\). That is, we need to find \(u_+\) at which the function \(J_{\mu,p,x}\) attains its global minimum. Actually, by Proposition 3.6, \(u_+\) is the largest root of the equation

\[u + \mu pu^{p-1} - x = 0.\]  

(3.11)

In section 3.3, we shall show that equation (3.11) can be converted to a polynomial when \(p\) is an rational number. In section 3.4, we shall numerically compute \(u_+\) as a
Figure 3.3: The proximity operators $T_{\mu,p}$ for $\mu = 3$ and $p = 0, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{5}{6}, 1$. 
solution of equation (3.11) via Newton’s method when \( p \) is not 0, 1/2, 2/3, or 1.

To close this section, we will show that the proximity operator \( T_{\mu,p} \) for \( 0 < p < 1 \) is not non-expansive. Recall that an operator \( Q : \mathbb{R} \to \mathbb{R} \) is non-expansive if 
\[
|Q(x) - Q(y)| \leq |x - y|
\]
for all \( x, y \in \mathbb{R} \).

**Lemma 3.8.** Let \( 0 < p < 1 \) and \( \mu > 0 \), and let \( x_1 \) and \( x_2 \) be two real numbers satisfying \( x_1 < x_2 \). Then for any \( u_i \in T_{\mu,p}(x_i) \), for \( i = 1, 2 \), it holds that \( u_1 \leq u_2 \). Furthermore, if \( x_1 < -\tau_{\mu,p} \) or \( x_2 > \tau_{\mu,p} \), then \( u_1 \) is strictly less than \( u_2 \).

**Proof.** By Lemma 3.1 and equation (3.10), it is sufficient to consider the case of both \( x_1 \) and \( x_2 \) positive. By the definition of the proximity operator, one has that 
\[
J_{\mu,p,x_1}(u_1) \leq J_{\mu,p,x_1}(u_2)
\]
\[
J_{\mu,p,x_2}(u_2) \leq J_{\mu,p,x_2}(u_1).
\]
Adding the above two inequations leads to \((u_2 - u_1)(x_2 - x_1) \geq 0\). Since \( x_1 < x_2 \) one obtains \( u_1 \leq u_2 \).

Next, we show that if \( x_2 > \tau_{\mu,p} \) then \( u_1 < u_2 \). In fact, if \( x_1 < \tau_{\mu,p} \), then \( u_1 = 0 \) from equation (3.10) and \( u_2 > \omega_{\mu,p} > 0 \) by Lemma 3.3. Hence, \( u_1 < u_2 \). If \( x_1 = \tau_{\mu,p} \) and \( u_1 = 0, u_1 < u_2 \) holds. Therefore, we assume that \( x_1 \geq \tau_{\mu,p} \) and \( u_1 \neq 0 \). Then by item (iii) of Lemma 3.3 or equation (3.11) the pairs \((x_i, u_i), i = 1, 2\), satisfy the following equations \( \mu p u_i^{p-1} + u_i - x_i = 0 \). It indicates that \( u_1 \neq u_2 \). Hence, \( u_1 < u_2 \). This completes the proof. \( \Box \)
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

Proposition 3.9. Let $0 < p < 1$ and $\mu > 0$, and let $x_1$ and $x_2$ be two real numbers such that $x_1 < x_2$. Then for any $u_i \in T_{\mu,p}(x_i)$, $i = 1, 2$, the following statements hold:

(i) if $x_1$ and $x_2$ have different signs or both them lie in the interval $(-\tau_{\mu,p}, \tau_{\mu,p})$, then $u_2 - u_1 < x_2 - x_1$.

(ii) if both $x_1$ and $x_2$ lie in the interval $(\tau_{\mu,p}, +\infty)$ or $(-\infty, -\tau_{\mu,p})$, then $u_2 - u_1 > x_2 - x_1$.

Proof. If $x_1$ and $x_2$ have different signs, then Item (i) follows immediately from Lemma 3.1 and Lemma 3.2. If both them lie in the interval $(-\tau_{\mu,p}, \tau_{\mu,p})$, then item (i) follows from equation (3.10).

Now, we turn to prove item (ii). We first assume that both $x_1$ and $x_2$ lie in the interval $(\tau_{\mu,p}, +\infty)$. By Lemma 3.3, $u_1$ is the critical point of $J_{\mu,p,x_1}$ while $u_2$ is the critical point of $J_{\mu,p,x_2}$. Therefore, one has $x_i = u_i + \mu p u_i^{p-1}$, for $i = 1, 2$. It follows that

$$x_2 - x_1 = u_2 - u_1 + \mu p (u_2^{p-1} - u_1^{p-1}).$$

Since $x_1 < x_2$, one has $u_1 < u_2$ by Lemma 3.8. Hence, the difference $u_2^{p-1} - u_1^{p-1}$ in the above equation is strictly less than zero. It yields that $u_2 - u_1 > x_2 - x_1$.

By Lemma 3.1, the result of item (ii) is also true for $x_1$ and $x_2$ lie in the interval $(-\infty, -\tau_{\mu,p})$. The proof is complete. \hfill $\square$

We conclude from item (ii) in Proposition 3.9 that $T_{\mu,p}$ is not non-expansive.
3.3 The Proximity Operators of the $\ell_{1/2}$- and $\ell_{2/3}$-Norm

As we mentioned in section 3.2 that the proximity operators $T_{\mu,1/2}$ and $T_{\mu,2/3}$ have been discussed in [14], based on the properties presented in section 3.2 we will provide the explicit forms of $T_{\mu,1/2}$ and $T_{\mu,2/3}$ with alternative, but much simple, proofs for them.

We begin to show that for a rational number $0 < p < 1$ and $x > \tau_{\mu,p}$ the computation of $T_{\mu,p}(x)$ can be reduced to finding the largest zero of a polynomial. More precisely, let us write $p = \frac{l}{k}$, where $k$, $l$ are relatively prime integers. Actually, set $t = u^{\frac{1}{k}}$. Substituting $u$ in (3.11) by $t^k$ and simplifying the resulting equation lead to the following polynomial equation

$$t^{2k-l} - xt^{k-l} + \frac{l}{k} \mu = 0, \quad t > 0. \tag{3.12}$$

Since $k$ and $l$ are relatively prime integers and $k > l$, the least degree of the polynomial in (3.12) is 3 only when $k = 2$ and $l = 1$, that is, $p = 1/2$. The second least degree of the polynomial is 4 only when $k = 3$ and $l = 2$, that is, $p = 2/3$. In the following, we shall present the closed-form formulas for the proximity operators $T_{\mu,p}$ with $p = 1/2$ and $p = 2/3$. For other choices of the rational number $p$, we need to find the solutions of equation (3.12) where the degree of the polynomial is higher than 4, therefore, it is hardly to have a closed-form formula for representing the roots of the polynomial.

The closed-form formula for the proximity operators $T_{\mu,1/2}$ is given as follows.
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

Proposition 3.10. Let $\mu > 0$. Then for $x \in \mathbb{R}$

\[
\mathcal{T}_{\mu, 1/2}(x) = \begin{cases} 
\{0\}, & \text{if } |x| < \frac{2}{3} \mu^{2}; \\
\{0, \text{sign}(x) \cdot \mu^{2}\}, & \text{if } |x| = \frac{2}{3} \mu^{2}; \\
\left\{ \frac{2}{3} x \left( 1 + \cos \left( \frac{2}{3} \cos^{-1} \left( -\frac{3^{3/2}}{4} \mu |x|^{-3/2} \right) \right) \right) \right\}, & \text{otherwise.}
\end{cases}
\] (3.13)

Proof. One can check that $\tau_{\mu, 1/2} = \frac{3}{2} \mu^{2}$ and $\rho_{\mu, 1/2} = \mu^{2}$. By equation (3.10) and the fact of sign$(x)|x| = x$, one just needs to show that for $x > \frac{3}{2} \mu^{2}$,

\[
u_{+} = \frac{2}{3} x \left( 1 + \cos \left( \frac{2}{3} \cos^{-1} \left( -\frac{3^{3/2}}{4} \mu |x|^{-3/2} \right) \right) \right).
\] (3.14)

When $p = 1/2$, equation (3.12) becomes

\[t^3 - xt + \frac{\mu}{2} = 0.
\]

By item (iii) of Lemma 3.3, the above cubic equation has three real roots with two positive roots and one negative root. Furthermore, $u_{+}$ is the square of the largest root of the cubic equation. Fortunately, using a formula in [?] this largest root is

\[t = 2 \sqrt{\frac{x}{3}} \cos \left( \frac{1}{3} \cos^{-1} \left( -\frac{3\mu}{4x} \sqrt{\frac{3}{x}} \right) \right).
\]

Hence $u_{+} = t^2$. Using the formula $\cos^2 \theta = \frac{1}{2}(1 + \cos(2\theta))$, we know that $t^2$ exactly equals to the right-hand side of equation (3.14). This completes the proof of the result.

Next, we present the closed-form formula for the proximity operators $\mathcal{T}_{\mu, 2/3}$. \hfill \Box
Proposition 3.11. Let $\mu > 0$. Then for $x \in \mathbb{R}$

$$
\mathcal{T}_{\mu, 2/3}(x) = \begin{cases} 
\{0\}, & \text{if } |x| < 2(\frac{2}{3}\mu)^{\frac{3}{4}}; \\
\{0, \text{sign}(x) \cdot (\frac{2}{3}\mu)^{\frac{3}{4}}\}, & \text{if } |x| = 2(\frac{2}{3}\mu)^{\frac{3}{4}}; \\
\{\text{sign}(x) \cdot \frac{1}{8} \left(\sqrt{2z} + \sqrt{\frac{2|x|}{\sqrt{2z}} - 2z}\right)^3\}, & \text{otherwise},
\end{cases}
$$

(3.15)

where

$$
z = \left(\frac{1}{16} x^2 + \sqrt{\frac{x^4}{256} - \frac{8\mu^3}{729}}\right)^{\frac{1}{3}} + \left(\frac{1}{16} x^2 - \sqrt{\frac{x^4}{256} - \frac{8\mu^3}{729}}\right)^{\frac{1}{3}}.
$$

(3.16)

Proof. One can check that $\tau_{\mu, 2/3} = 2(\frac{2}{3}\mu)^{\frac{3}{4}}$ and $\varrho_{\mu, 2/3} = (\frac{2}{3}\mu)^{\frac{3}{4}}$. By equation (3.10) and Lemma 3.1 we only need to compute the proximity operator $\mathcal{T}_{\mu, 2/3}(x)$ for $x > 2(\frac{2}{3}\mu)^{\frac{3}{4}}$.

When $p = 2/3$, equation (3.12) becomes

$$
t^4 - xt + \frac{2\mu}{3} = 0,
$$

(3.17)

which also has two and only two positive roots and whose largest root can lead to $u_+$. We now find the largest positive root of equation (3.17). For any real number $w$, equation (3.17) is identical to the following one

$$
(t^2 + w)^2 = 2wt^2 + xt + (w^2 - \frac{2}{3}\mu).
$$

(3.18)

In particular, we can choose a specific $w$ so that the expression of the right-hand side of the above equation can be completed in square with respect to the variable $t$. This requires that $w$ satisfies the following equation

$$
w^3 - \frac{2}{3} \mu w - \frac{1}{8} x^2 = 0,
$$
which has at least one real solution. Actually, \( w = z \) with \( z \) given by (3.16) is the solution of this cubic equation. With this choice, equation (3.18) is equivalent to the following two equations

\[
t^2 + \sqrt{2}zt + \left( z + \frac{\sqrt{2}x}{4}z^{-1/2} \right) = 0 \quad \text{and} \quad t^2 - \sqrt{2}zt + \left( z - \frac{\sqrt{2}x}{4}z^{-1/2} \right) = 0
\]

The first quadratic equation has two complex roots while the second one has two real roots. We therefore only need to find the largest root of the second quadratic equation. Actually, this root is

\[
t = \frac{1}{2} \left( \sqrt{2}z + \sqrt{\frac{2x}{\sqrt{2}z} - 2z} \right).
\]

This completes the proof. \( \square \)

3.4 Computing the Proximity Operator of the \( \ell_p \)-Norm (0 < \( p < 1 \))

In the previous sections, we have already presented the closed-form formulas for the proximity operator \( T_{\mu,p} \) with \( p \) being 0, 1/2, 2/3 and 1. For other choice of \( p \), we shall develop in this section a numerical algorithm for computing the proximity operator \( T_{\mu,p} \).

According to (3.10), it is essential to develop a numerical scheme that can compute \( T_{\mu,p}(x) \) for all \( x > \tau_{\mu,p} \). Note that in this case the set \( T_{\mu,p}(x) \) only contains one element and we simply use \( T_{\mu,p}(x) \) to denote this element. We further know that \( T_{\mu,p}(x) \) with
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

$x > \tau_{\mu,p}$ is the largest solution of equation (3.11). To locate this solution, we propose to approximate it by Newton’s method. For fixed $\mu > 0$, $0 < p < 1$, and $x > \tau_{\mu,p}$, define

$$H(u) := u - \frac{h(u)}{h'(u)}$$

(3.19)

where $h(u) := u + \mu p u^{p-1} - x$. Newton’s method begins with an estimate $u_0$ of $T_{\mu,p}(x)$ and then defines inductively

$$u_{n+1} = H(u_n),$$

(3.20)

where $n \geq 0$.

**Proposition 3.12.** For $p \in (0,1)$, $\mu > 0$, and $x > \tau_{\mu,p}$, if an initial estimate $u_0 > T_{\mu,p}(x)$ then the sequence generated by (3.20) is decreasing and bounded below by $T_{\mu,p}(x)$. Moreover, the sequence converges to $T_{\mu,p}(x)$.

**Proof.** Let us prove the sequence $\{u_n : n \in \mathbb{N}\}$ is decreasing and bounded below by $T_{\mu,p}(x)$ first. We proceed inductively. For $n = 0$, it is true by the assumption that $u_0 > T_{\mu,p}(x)$. Now suppose that $T_{\mu,p}(x) \leq u_{k+1} \leq u_k$ for all $0 \leq k \leq n - 1$. We show that $T_{\mu,p}(x) \leq u_{n+1} \leq u_n$. By Lemma 3.3, the function $h$ is increasing and convex on $[T_{\mu,p}(x), +\infty)$. Hence, $h(u_n) > 0$ and $h'(u_n) > 0$. We obtain that $u_{n+1} < u_n$ from the identity $0 = h(u_n) + h'(u_n)(u_{n+1} - u_n)$. By the convexity of $h$, we have that $h(u) > h(u_n) + h'(u_n)(u - u_n)$ for all $u \in [T_{\mu,p}(x), +\infty)$ and $u \neq u_n$. In particular, taking $u = T_{\mu,p}(x)$ in this inequality, we have that $0 > h(u_n) + h'(u_n)(T_{\mu,p}(x) - u_n)$.

This yields $T_{\mu,p}(x) \leq u_{n+1}$. Therefore, the sequence converges, says, $\lim_{n \to +\infty} u_n =$
CHAPTER 3. COMPUTING THE PROXIMITY OPERATOR OF THE $\ell_p$-NORM

$u_* \geq T_{\mu,p}(x)$. Taking the limit on both sides of equation (3.20) leads to $h(u_*) = 0$. Using Lemma 3.3 again, we know that $u_* = T_{\mu,p}(x)$.

By Lemma 3.2, we can choose $u_0 = x$ as our initial estimator in Newton’s method (3.20). A better initial estimator can be chosen as well. To this end, we need the following technical lemma.

**Lemma 3.13.** Let $\mu > 0$ and $x > 0$. Let $p_1$ and $p_2$ be two numbers in $[0, 1]$ with $p_1 < p_2$. Define $p_{12} := (p_1/p_2)^{1/(p_2-p_1)}$. Then for any $u_i \in T_{\mu,p_i}(x)$, $i = 1, 2$, we have that $u_1 \leq u_2$ if $\max\{u_1, u_2\} \leq p_{12}$; $u_1 \geq u_2$ if $\min\{u_1, u_2\} \geq p_{12}$.

**Proof.** Since $x > 0$ and $u_i \in T_{\mu,p_i}(x)$, then both $u_1$ and $u_2$ are non-negative from Lemma 3.2. By the definition of the proximity operator, from $u_i \in T_{\mu,p_i}(x)$, we have that $J_{\mu,p_i,x}(u_1) \leq J_{\mu,p_1,x}(u_2)$ and $J_{\mu,p_2,x}(u_2) \leq J_{\mu,p_2,x}(u_1)$. Adding these two inequalities together yields

$$u_1^{p_1} - u_1^{p_2} \leq u_2^{p_1} - u_2^{p_2}. \quad (3.21)$$

Define $g(u) := u^{p_1} - u^{p_2}$ on the interval $[0, +\infty)$. We can check directly that $g$ is continuous, is increasing on $[0, p_{12}]$, and decreasing on $[p_{12}, +\infty)$. Then, the results of this theorem follows from inequality (3.21).

For any $p \in (0, 1)$, but not $1/2$ and $2/3$, we define $p_-$ to be the largest element in
the set \( \{0, 1/2, 2/3, 1\} \) that is smaller than \( p \). That is,

\[
p_\pm = \begin{cases} 
0, & \text{if } p \in (0, 1/2); \\
1/2, & \text{if } p \in (1/2, 2/3); \\
2/3, & \text{if } p \in (2/3, 1). 
\end{cases}
\]

**Proposition 3.14.** For \( p \in (0, 1) \setminus \{1/2, 2/3\}, \mu > 0, \) and \( x > \max\{\tau_{\mu,p}, \tau_{\mu,p_-}\} \), if

\[
x > \max\{\tau_{\mu,p} - \varrho_{\mu,p}, \tau_{\mu,p_-} - \varrho_{\mu,p_-}\} + \left(\frac{p_-}{p}\right)^{1/(p-p_-)},
\]

(3.22)

then

\[
T_{\mu,p}(x) \leq T_{\mu,p_-}(x).
\]

**Proof.** By Item (ii) in Proposition 3.9, we have that \( T_{\mu,p}(x) > x - (\tau_{\mu,p} - \varrho_{\mu,p}) \) and \( T_{\mu,p_-}(x) > x - (\tau_{\mu,p_-} - \varrho_{\mu,p_-}) \). By the assumption (3.22) together the proceeding two inequalities, both \( T_{\mu,p}(x) \) and \( T_{\mu,p_-}(x) \) are bigger than \( (p_-/p)^{1/(p-p_-)} \). Our result follows from Lemma 3.13.

To summarize the above discussions, a detailed pseudocode for computing the proximity operator \( T_{\mu,p}(x) \) including stopping criteria is given in Algorithm 3. We point it out that the proximity operator \( T_{3,4/5} \) showing in Figure ??(a) and the proximity operators \( T_{3,4/5} \) and \( T_{3,2/3} \) showing in Figure ??(b) are computed numerically through Algorithm 3.
Algorithm 3: (Computing $T_{\mu,p}(x)$ for $0 < p < 1$)

**Input:** $p \in [0, 1]$, $\mu > 0$, $\epsilon > 0$, and $x \in \mathbb{R}$

**Result:** $T_{\mu,p}(x)$

begin

\[ T_{\mu,p}(x) = \begin{cases} 
0, & \text{if } |x| < \tau_{\mu,p}; \\
0, \text{sign}(x) \cdot 2^{\mu,p}, & \text{if } |x| = \tau_{\mu,p}; \\
\end{cases} \]

else

\begin{align*}
&\text{Determine an initial estimator } u_0 \text{ for the Newton iteration;} \\
&\text{if } |x| > \max\{\tau_{\mu,p} - 2^{\mu,p}, \tau_{\mu,p} - 2^{\mu,p-1}\} + \left(\frac{p}{p-1}\right)^{1/(p-p-1)} \text{ and } |x| > \tau_{\mu,p-1}
\end{align*}

then

\[ u_0 = T_{\mu,p-1}(|x|) \]

else \[ u_0 = |x| \]

Newton’s iteration;

\begin{algorithmic}
\State $u_n = u_{n-1}$
\State $u_{n+1} = H(u_n)$
\State the final iterate is denoted by $u_\infty$;
\end{algorithmic}

end
In this chapter, we formulate application problems in image processing and compressed sensing as composite minimization problems. For convex composite minimization problems arising from image deblurring and compressed sensing, proposed algorithms from chapter 2 will be applied and comparisons of proposed algorithms with other algorithms will be performed. Also, an algorithm using the proximity operator of the $\ell_p$-norm will be developed to solve the $\ell_p$-regularized compressed sensing problem.
CHAPTER 4. APPLICATIONS AND NUMERICAL EXPERIMENTS

4.1 Applications

In the field of engineering, many application problems including image processing, compressed sensing are aiming to recover underlying image or signal from a degraded version. A degraded image or signal $y$ can be modeled as

$$y = Mx + \eta,$$

where $x \in \mathbb{R}^n$ is the image or signal to be reconstructed, $M$ is a $d \times n$ matrix that models the measurement process, and $\eta \in \mathbb{R}^d$ is an additive noise. In linear inverse problem (4.1), the goal is to recover image $x$ when $y$ and $M$ are given. For different choices of $M$, recovering $x$ becomes different application problems. For instance, it becomes the deblurring problem if $M$ represents a blurring matrix; it becomes the inpainting problem if $M$ represents a projection of an image onto some known pixels domain. If $M$ is the identity matrix, it reduces to the denoising problem. If matrix $M$ models incomplete measurement and $x$ has (approximately) sparse representation, recovering $x$ becomes a problem in compressed sensing. But linear inverse problem (4.1) is usually ill-posed in image processing or compressed sensing. For instance, in image deblurring, linear inverse problem (4.1) is ill-posed in the sense that the blurring matrix $A$ is ill-conditioned and solution could be sensitive to the additive noise. With incomplete measurement in compressed sensing, linear inverse problem (4.1) has infinite number of solutions.

To recover the underlying image or signal $x$ in (4.1), one powerful method is regularization method. A regularized model for model (4.1) can be derived from
Bayesian rule depending on the prior information of image or signal \( x \) to be recovered and the type of the additive noise. In Bayesian approach, we assume that the degraded image or signal \( y \) is a realization from a random vector \( Y \) and the underlying image or signal \( x \) is a realization of another random vector \( X \). By Bayesian formula, the conditional \textit{a posteriori} probability \( p(x|y) \), the probability that \( x \) occurs when \( y \) is observed, is given by

\[
p(x|y) = \frac{p(y|x)p_X(x)}{p_Y(y)}.
\]  

(4.2)

To find an estimate of \( x \), a maximum \textit{a posteriori} expectation maximization could be used by maximizing the conditional \textit{a posteriori} probability \( p(x|y) \). By taking the negative logarithm of equation (4.2) and ignoring the constant term \( \log p_Y(y) \), an estimate of \( x \) is equivalent to a solution to the following minimization problem

\[
\min_x \{- \log p(y|x) - \log p_X(x)\}.
\]  

(4.3)

The term \( \log p_X(x) \) is used to regularize a solution from the assumption on prior information of \( x \). Gibbs prior is usually assigned to the random vector \( X \) in practice. Hence, the prior \( p_X(x) \) has form

\[
p_X(x) = \frac{1}{T}e^{-\gamma E(x)},
\]  

(4.4)

where \( T \) is a normalization factor, \( \gamma \) is a positive number and \( E(x) \) is a given energy function of \( x \). The choice of the energy function varies from application to application. In image processing, as one choice of energy function \( E(x) \), the total variation \( \|x\|_{TV}[66] \) has been extensively used due to the fact that the total variation is sensitive
to geometric features of images, such as edges. Another alternative is \( E(x) = \| W x \|_1 \), where \( W \) is a matrix representation of wavelet or framelet since natural images tend to be sparse in the wavelet or framlet domain\([26, 30, 31, 65]\). In compressed sensing, if the underlying signal itself is sparse, the \( \ell_p \)-function \( \| x \|_p^p \) with \( 0 \leq p \leq 1 \) is appropriate for the energy function \( E(x) \) due to the fact that \( \ell_p \)-norm is sparsity-promoting. While if the signal is not sparse itself but is sparse in the transformation domain associated with a linear transform \( T \), then \( E(x) = \| T x \|_p \) is suitable.

The expression \( \log p(y|x) \) in (4.3) is viewed as a fidelity term measuring the discrepancy between the noisy observation \( y \) and an ideal one. The fidelity term \( \log p(y|x) \) depends on the property of the additive noise \( \eta \). For convenience, we assume the noise \( \eta \) is \( d \)-dimensional. When the noise \( \eta \) is of Gaussian type, it is assumed that the components \( \eta_i \) of \( \eta \) are independently and identically distributed (i.i.d.) from a Gaussian distribution \( \mathcal{N}(0, \sigma^2) \). It follows that the likelihood \( p(y|x) = (2\pi)^{-d/2} \sigma^{-d} e^{-\frac{\|y - M x\|^2}{2\sigma^2}} \).

Putting the expression \( p_X(x) \) and \( p(y|x) \) into (4.3) and ignoring the constant, we obtain an equivalent model of (4.3) when Gaussian noise is involved

\[
\min_x \left\{ \frac{1}{2} \| y - M x \|^2 + \mu E(x) \right\},
\]

where \( \mu \) is a positive parameter related to the noise standard deviation \( \sigma \) and parameter \( \gamma \) in the Gibbs prior. As shown by model (4.5), an \( \ell_2 \)-fidelity term is appropriate for Gaussian noise corrupted data from statistics point of view. However, if the observation involves impulse noise rather than Gaussian noise, an \( \ell_2 \)-fidelity term is not suitable anymore. If the observation \( y \) is corrupted by salt-and-pepper noise (a
special type of impulse noise) with a noise level $0 < r < 1$, $y$ can be modeled as

$$y_i = \begin{cases} 0, & \text{with probability } \frac{r}{2}, \\ 255, & \text{with probability } \frac{r}{2}, \\ (Mx)_i, & \text{with probability } 1 - r, \end{cases}$$  \tag{4.6}

where $y_i$ is the $i$-th component of $y$. For observation corrupted by salt-pepper noise given in (4.6), we have that

$$p(y|x) = \left( \frac{r}{2} \right)^{|\{i : y_i \neq (Mx)_i\}|} \cdot (1 - r)^{|\{i : y_i = (Mx)_i\}|},$$  \tag{4.7}

where $|S|$ denotes the number of elements in the set $S$. Note that $|\{i : y_i \neq (Mx)_i\}| = \|Mx - y\|_0$, where $\| \cdot \|_0$ denotes the number of non-zero elements in a vector. Then the equation (4.7) becomes

$$p(y|x) = (1 - r)^d \left( \frac{2}{r} - 2 \right)^{-\|Mx - y\|_0}. \tag{4.8}$$

Putting the expression $p_X(x)$ and $p(y|x)$ into (4.3) and ignoring the constant, we obtain an equivalent model of (4.3) when salt-and-pepper noise is involved

$$\min_x \{ \|y - Mx\|_0 + \mu E(x) \},$$  \tag{4.9}

where $\mu$ is a positive parameter related to the corruption percentage $r$ and parameter $\gamma$ in the Gibbs prior. The non-convexity of the fidelity term $\|y - Mx\|_0$ introduces numerical difficulties in solving the minimization problem (4.9). To overcome the numerical difficulty resulted from the non-convexity of the term $\|y - Mx\|_0$, one way is to relax the non-convex term $\|y - Mx\|_0$ to a convex function $\|y - Mx\|_1$. With
such a relaxation, model (4.9) becomes

$$\min_x \{ \|y - Mx\|_1 + \mu \mathcal{E}(x) \}. \quad (4.10)$$

In fact, the $\ell_1$-norm fidelity term was first proposed by Nikolova for the total variation regularization of images corrupted by impulse noise[58]. Its effectiveness in handling impulse noise can be also found in [19]. The suitability of replacing the $\ell_0$-norm by the $\ell_1$-norm was also addressed in compressed sensing[12].

The parameter $\mu$ in both (4.5) and (4.10) is called the regularization parameter and need be predetermined. This regularization parameter balances the fitness of observed data and preservation of prior information of underlying solution. If noise power is less, more weight should be placed on the fidelity(fitting) term and therefore smaller value of $\mu$ should be chosen; while bigger value of $\mu$ is desired if noise power is more. But it is still challenging to choose an appropriate regularization parameter in practice. If an estimated upper bound of the noise power is available, an unconstrained model can be substituted by a constrained model without introducing regularization parameter. In particular, if the involved noise is Gaussian type, a variant model of (4.5) has the form

$$\min_x \{ \mathcal{E}(x) : \|y - Mx\| \leq \epsilon \}, \quad (4.11)$$

where $\epsilon^2$ is an estimated upper bound on the noise power of Gaussian noise. Between models (4.5) and (4.11), another difference is on the differentiability of the fidelity term. As a quadratic function, the fidelity term in (4.5) is differentiable, while the fidelity term in (4.11) that can be written as an indicator function is non-smooth.
The exact form of models (4.5), (4.10) and (4.11) is highly related to the type of noise in (4.1), the choice of regularization term and the matrix $M$. In the following, the $\ell_2$-TV and $\ell_1$-TV models and models in compressed sensing will be reviewed accordingly.

### 4.1.1 Applications to Image Deblurring

In this section, we first identify two well-known image deblurring models, namely the $\ell_2$-TV and $\ell_1$-TV models, as special cases of the general model (1.1). We then give details on how Algorithms 1 and 2 are applied. In particular, we present the explicit expressions of the proximity operators of $f_1^*$ and $f_2^*$. Since the total variation (TV) is involved in both image deblurring models, we begin with presenting the discrete setting for total variation.

For convenience of exposition, we assume that an image considered has a size of $\sqrt{n} \times \sqrt{n}$. The image is treated as a vector in $\mathbb{R}^n$ in such a way that the $ij$-th pixel of the image corresponds to the $(i + (j-1)\sqrt{n})$-th component of the vector in $\mathbb{R}^n$. The total variation of the image $x$ can be expressed as the composite function of a convex function $\psi : \mathbb{R}^{2n} \to \mathbb{R}$ and a $2n \times n$ matrix $B$. To define the matrix $B$, we
need a $\sqrt{n} \times \sqrt{n}$ difference matrix $D$ as follows:

$$D := \begin{bmatrix} 0 & -1 & 1 \\ -1 & \ddots & \ddots \\ & \ddots & \ddots \\ -1 & 1 \end{bmatrix}.$$  

The matrix $D$ will be used to “differentiate” pixel values along rows or along columns of an image matrix. Through the matrix Kronecker product $\otimes$, we define the $2n \times n$ matrix $B$ by

$$B := \begin{bmatrix} I \otimes D \\ D \otimes I \end{bmatrix},$$  

(4.12)

where $I$ is the $\sqrt{n} \times \sqrt{n}$ identity matrix. The matrix $B$ will be used to “differentiate” the entire image matrix. The norm of $B$ is $\|B\|^2 = 8 \sin^2 \left( \frac{\sqrt{n}-1}{2\sqrt{n}} \pi \right)$ (see [54]).

We define $\psi : \mathbb{R}^{2n} \to \mathbb{R}$ at $v \in \mathbb{R}^{2n}$ as

$$\psi(v) := \sum_{i=1}^{n} \| [v_i, v_{n+i}]^T \|. $$  

(4.13)

Based on the definition of the $2n \times n$ matrix $B$ and the convex function $\psi$, the (isotropic) total variation of an image $x$ can be denoted by $\psi(Bx)$, i.e.

$$\|x\|_{TV} := \psi(Bx).$$  

(4.14)

The $\ell_2$-TV Image Deblurring Model

If the additive noise is Gaussian type, model (4.5) can be adopted. Using the total variation as the energy function yields the $\ell_2$-TV image deblurring model. The $\ell_2$-TV
image deblurring model has the form of
\[
\min \left\{ \frac{1}{2} \|Mx - y\|^2 + \mu \|x\|_{TV} : x \in \mathbb{R}^n \right\},
\] (4.15)

where \(\mu\) is a regularization parameter.

Now, let us set
\[m_1 = n, \quad m_2 = 2n, \quad f_1 := \frac{1}{2} \| -y \|^2, \quad f_2 := \mu \psi, \quad A_1 := M, \quad \text{and} \quad A_2 := B,
\]
where \(\psi\) is given by (4.13) and \(B\) is defined by (4.12). Then the \(\ell_2\)-TV image deblurring mode (4.15) can be viewed as a special case of model (1.1). In addition, \(f_1 \in \Gamma_0(\mathbb{R}^n)\) and \(f_2 \in \Gamma_0(\mathbb{R}^{2n})\). Therefore, both Algorithms 1 and 2 can be applied for the \(\ell_2\)-TV model. Furthermore, we give the explicit forms of the proximity operators \(\text{prox}_{\alpha f_1^*}\) and \(\text{prox}_{\alpha f_2^*}\) for any positive number \(\alpha\). Actually, by the definition of Fenchel conjugate, we have
\[
f_1^*(u) = \frac{1}{2} \|u\|^2 + \langle u, y \rangle.
\]

By the definition of proximity operator, we have that for \(u \in \mathbb{R}^n\)
\[
\text{prox}_{\alpha f_1^*}(u) = \frac{1}{1 + \alpha} u - \frac{\alpha}{1 + \alpha} y.
\]

By introducing the \(\ell_2\)-ball \(B = \{p \in \mathbb{R}^2 : \|p\| \leq \mu\}\), for \(v \in \mathbb{R}^{2n}\) we have
\[
f_2^*(v) = \sum_{i=1}^{n} \iota_B([v_i, v_{n+i}]^T),
\]
where the indicator function \(\iota_B\) over the non-empty set \(B\) is defined by
\[
\iota_B(p) = \begin{cases} 
0 & \text{if } p \in B, \\
\infty & \text{otherwise}
\end{cases}.
\]
For $v \in \mathbb{R}^{2n}$, we write $z = \text{prox}_{\alpha f_2^*}(v)$. Then for $i = 1, 2, \ldots, n$, we have that

$$
[z_i, z_{n+i}]^\top = \min\{\| [v_i, v_{n+i}]^\top \|, \mu \} \frac{[v_i, v_{n+i}]^\top}{\| [v_i, v_{n+i}]^\top \|}.
$$

(4.16)

The $\ell_1$-TV Image Deblurring Model

The $\ell_1$-TV image deblurring model is usually used for the recovery of an unknown image $x \in \mathbb{R}^n$ from an impulse noise corrupted observable data $y \in \mathbb{R}^n$ modeled by (4.6), where $M$ represents a blurring matrix and $\eta$ is an impulse noise. To recover the underlying image $x$ from an observed data with impulse noise corruption, we adopt model (4.10) with the $\ell_1$-norm fidelity term. Replacing the energy function by total variation yields the $\ell_1$-TV image deblurring model. The $\ell_1$-TV image deblurring model has the form of

$$
\min\{\| Mx - y \|_1 + \mu \| x \|_{TV} : x \in \mathbb{R}^n \},
$$

(4.17)

where $\mu$ is again the regularization parameter.

Now, let us set

$$
m_1 = n, \quad m_2 = 2n, \quad f_1 := \| \cdot - y \|_1, \quad f_2 := \mu \psi, \quad A_1 := M, \quad \text{and} \quad A_2 := B,
$$

where $\psi$ is given by (4.13) and $B$ is defined by (4.12). Then the $\ell_1$-TV image deblurring model (4.17) can be viewed as a special case of model (1.1). In addition, $f_1 \in \Gamma_0(\mathbb{R}^n)$ and $f_2 \in \Gamma_0(\mathbb{R}^{2n})$. Therefore, both Algorithms 1 and 2 can be applied for the $\ell_1$-TV model. Further, the proximity operator $\text{prox}_{\alpha f_2^*}$ has been given via (4.16). We just need to present the proximity operator $\text{prox}_{\alpha f_1^*}$. Actually, we have
that for \( u \in \mathbb{R}^n \)

\[
(prox_{\lambda f_1}(u))_i = \begin{cases}
 y_i + \text{sign}(u_i - y_i)(|u_i - y_i| - \lambda), & \text{if } |u_i - y_i| \geq \lambda; \\
 y_i, & \text{otherwise},
\end{cases}
\]

where \( i = 1, 2, \ldots, n \). Using the Moreau’s identity \( prox_{\alpha f_1^*(u)} = u - \alpha \text{prox}_{\frac{1}{\alpha f_2}}(\frac{u}{\alpha}) \), we have that for \( u \in \mathbb{R}^n \)

\[
(prox_{\alpha f_1^*(u)})_i = \begin{cases}
 \text{sign}(u_i - \alpha y_i), & \text{if } |u_i - \alpha y_i| \geq 1; \\
 u_i - \alpha y_i, & \text{otherwise},
\end{cases}
\]

where \( i = 1, 2, \ldots, n \).

In summary, for both the \( \ell_2\)-TV and \( \ell_1\)-TV image deblurring models, the associated proximity operators \( prox_{\alpha f_1^*} \) and \( prox_{\alpha f_2^*} \) have closed forms. As a consequence, the sequence \( \{(u^k, v^k, x^k) : k \in \mathbb{N}\} \) generated by Algorithms 1 and 2 can be efficiently computed.

### 4.1.2 Application to Compressed Sensing

In this section, we consider the problems from compressed sensing. The breakthrough of the compressed sensing theory is that one can represent a signal at a rate significantly below the Nyquist sampling frequency [13]. The basic principle in compressed sensing is that a sparse or compressible signal can be reconstructed from a small number of measurements, measured through appropriate linear combinations of signal values, via an optimization approach. An essential goal in compressed sensing is to reconstruct the ideal signal from a small number of measurements. A key to this goal
is the notion of sparsity. It was shown mathematically in [13] that under the sparsity assumption, the signal can be exactly reconstructed from the given measurements and the chance of its being wrong is infinitesimally small. The sparsity of the signal can be captured by using regularization with the $\ell_1$-norm or $\ell_p$-norm with $0 \leq p < 1$.

In the seminal work [11, 33], the compressed sensing problem was described as solving the $\ell_1$-minimization problem subject to linear constraints that involve measurements and a measurement matrix. In the work [17, 18], an exact recovery of a sparse signal was described by solving the $\ell_p$-minimization problem.

We identify the $\ell_1$-minimization or $\ell_p$-minimization problems in compressed sensing as special cases of the general composite minimization model (1.1). If the $\ell_1$-norm is adopted for regularization, the composite minimization problem has convex objective function and proposed algorithms from chapter 2 can be applied. If the $\ell_p$-regularization ($0 \leq q < 1$) is chosen, existing algorithms arising from convex minimization may be extended to solve the non-convex $\ell_p$-minimization problem.

**The $\ell_1$-Regularization for Compressed Sensing**

In this section, we consider the $\ell_1$-regularized minimization problem from compressed sensing and identify it as a special case of the convex composite minimization model (1.1). The proposed algorithms from chapter 2 can be applied by providing explicit form of the proximity operators $\text{prox}_{\alpha f_1^*}$ and $\text{prox}_{\alpha f_2^*}$ for the specific functions $f_1^*$ and $f_2^*$. 
In compressed sensing, the signal of interest $x \in \mathbb{R}^n$ is assumed to have (approximately) sparse representation in some linear transform domain. A collected signal $y \in \mathbb{R}^{m_1}$ is modeled by (4.1), where $M$ is an $m_1 \times n$ ($m_1 < n$) matrix and models the incomplete measurement. By convention, $\eta \in \mathbb{R}^{m_1}$ in (4.1) represents an additive Gaussian noise. The underlying signal $x$ can be restored by solving the unconstrained model (4.5) or the constrained model (4.11) with an appropriate energy function $\mathcal{E}(x)$.

If an upper bound of noise power is available and $x$ has a sparse representation under a linear transform $T$ (an $m_2 \times n$ matrix), restoring the underlying signal $x$ can be formulated as solving the following $\ell_1$-minimization problem

$$\min\{\|Tx\|_1 : x \in \mathbb{R}^n\}, \text{ subject to } \|Mx - y\| \leq \epsilon,$$

where $\epsilon^2$ indicates the upper bound of noise power. Let $C := \{ u \in \mathbb{R}^{m_1} : \|u - y\| \leq \epsilon \}$ and $f_1 := \iota_C$, $f_2 := \| \cdot \|_1$, $A_1 := M$, $A_2 := T$, then the minimization problem (4.18) can be viewed as a special case of the general problem (1.1). The conjugate function $f_2^*$ of $f_2 = \| \cdot \|_1$ is the indicator function $\iota_B$, where $B = \{ v \in \mathbb{R}^{m_2} : \|v\|_\infty \leq 1 \}$ represents the unit $l_\infty$-ball. As a result, the proximity operator $\text{prox}_{\alpha f_2^*}(v)$ is the projection of $v$ onto the set $B$. Indeed, we have for $v \in \mathbb{R}^{m_2}$

$$(\text{prox}_{\alpha f_2^*}(v))_i = \begin{cases} v_i, & \text{if } |u_i| \leq 1, \\ \text{sign}(u_i), & \text{otherwise} \end{cases},$$

for $i = 1, \cdots, m_2$. Since the function $f_1$ is an indicator function over the $\ell_2$-ball $C$ with center $y$ and radius $\epsilon$, the proximity operator $\text{prox}_{\lambda f_1}(u)$ is the projection of $v$
onto $C$, i.e.,

$$
\text{prox}_{\lambda f_1}(u) = \begin{cases} 
  u, & \text{if } \|u - y\| \leq \epsilon, \\
  y + \frac{\epsilon}{\|u - y\|}(u - y), & \text{otherwise}
\end{cases}
$$

Using the Moreau’s identity $\text{prox}_{\alpha f_1}^*(u) = u - \alpha \text{prox}_{\frac{1}{\alpha} f_1}(u)$, we can get

$$
\text{prox}_{\alpha f_1}(u) = \begin{cases} 
  0, & \text{if } \|u - \alpha y\| \leq \alpha \epsilon, \\
  (1 - \frac{\alpha \epsilon}{\|u - \alpha y\|})(u - \alpha y), & \text{otherwise}
\end{cases}
$$

As displayed above, the proximity operators $\text{prox}_{\alpha f_1}$ and $\text{prox}_{\alpha f_2}$ associated to model (4.18) have closed form. As a consequence, the sequence $\{(u_k, v_k, x_k) : k \in \mathbb{N}\}$ generated by Algorithms 1 and 2 can be efficiently computed as well for problem (4.18).

**The $\ell_p$-Regularization for Compressed Sensing**

In this section, we consider the non-convex composite minimization problem with the $\ell_p$-regularization ($0 < p < 1$) in compressed sensing. For the $\ell_p$-regularization, researchers have shown their interest and considerable effort has been devoted to its study[7, 17, 18, 24, 37, 46, 53, 57, 69, 75]. It has been shown from numerical experiment that using the $\ell_p$-norm promotes sparser solutions and lower prediction errors for model selection when compared to the use of the $\ell_1$-norm[75]. Moreover, it has also been proven that fewer measurements as well as weaker conditions are required for sparse signal recovery[18, 46, 75]. For simplicity, it is assumed that the signal itself is sparse, i.e., the matrix $T$ is the identity matrix. Replacing the $\ell_1$-norm
in model (4.18) by the $\ell_p$-norm yields the following variant model

$$\min_x \{ \|x\|_p^p : \|Mx - y\| \leq \epsilon \}. \quad (4.19)$$

When $p = 1$, model (4.19) reduces to the constrained basis pursuit denoising model in [23], which is convex and has been solved by many algorithms, see, for example, [4, 34] and references therein. We also developed an accurate and efficient algorithm for solving the optimization problem with $p = 1$ in [20].

However, replacing the $\ell_1$-norm by the $\ell_p$-norm with $0 < p < 1$ results in a non-convex model in (4.19). Desirable properties involving Fenchel conjugate that are seen in proper semi-continuous convex function, would not be seen in the $\ell_p$-norm. The proposed algorithms in chapter 2 would not work appropriately for the $\ell_p$-minimization problem (4.19). In the numerical experiment for the non-convex $\ell_p$-regularized compressed sensing, we will extend the algorithm for basis pursuit denoising model in our recent work[20] to solve the $\ell_p$-minimization problem (4.19).

### 4.2 Numerical Experiments

In this section, numerical experiments are carried out to demonstrate the performance of our proposed Algorithms 1 and 2 for the image deblurring and the $\ell_1$-regularized compressed sensing. Numerical performance of the $\ell_p$-regularization for compressed sensing is also presented. For convex composite minimization problem, the Chambolle-Pock (CP) algorithm and ZBO algorithm[77] for (1.2) are compared
to the proposed algorithms from chapter 2 for the $\ell_2$-TV, $\ell_1$-TV image deblurring and
the $\ell_1$-regularized compressed sensing model (4.18). The ZBO algorithm proposed in
[77] solves model (1.2) via the following scheme

$$
\begin{align*}
  w^{k+1} &= \text{argmin} \left\{ f(w) - \langle \lambda^k, Ax^k - w \rangle + \frac{\beta}{2} \| Ax^k - w \|^2 \right\} \\
  &\quad + \frac{1}{2} \| w - w^k \|^2_{Q_1} : w \in \mathbb{R}^m \\
  x^{k+1} &= \text{argmin} \left\{ -\langle \lambda^k, Ax - w^{k+1} \rangle + \frac{\beta}{2} \| Ax - w^{k+1} \|^2 \right\} \\
  &\quad + \frac{1}{2} \| x - x^k \|^2_{Q_2} : x \in \mathbb{R}^n \\
  \lambda^{k+1} &= \lambda^k - \gamma (Ax^{k+1} - w^{k+1})
\end{align*}
$$

where $Q_1$, $Q_2$ are positive definite matrices, and $\beta$, $\gamma > 0$. When $Q_1$ and $Q_2$ are
chosen as $Q_1 = \left( \frac{1}{\alpha_1} - \beta \right) I$ and $Q_2 = \frac{\alpha_2}{\alpha_2} I - \beta A^\top A$ respectively, scheme (4.20) has closed
form. The positive definiteness of $Q_1$ and $Q_2$ ensures that $\alpha_1 \beta < 1$ and $\alpha_2 \beta < \frac{1}{\|A\|^2}$.
Each algorithm is carried out until the stopping criterion $\| x^{k+1} - x^k \|^2 / \| x^k \|^2 \leq Tol$ is
satisfied, where $Tol$ representing the tolerance, will be specified differently in different
applications.

### 4.2.1 Parameter Settings

Prior to applying Algorithms 1 and 2, the CP algorithm and the ZBO algorithm to the
$\ell_2$-TV model and the $\ell_1$-TV model and the $\ell_1$-regularized compressed sensing model,
the parameters arising from those algorithms need to be determined. Convergence
analysis of the algorithms specifies the relation between these parameters. Under the
conditions on parameters that guarantee convergence, we notice that larger product

of the parameters results in faster convergence [20]. Therefore, once one parameter is fixed, the others can be described by this fixed one. To this end, we fix the value of the parameter $\beta$ in each above algorithm and then figure out the values of the others.

The setting of parameters is described as follow.

For Algorithm 1, the positive parameters $\alpha_1$, $\alpha_2$, and $\gamma$ satisfy

$$\alpha_1 < \frac{1}{\beta \|A_1\|^2}, \quad \alpha_2 < \frac{1}{\beta \|A_2\|^2}, \quad \text{and} \quad \gamma \leq \beta. \quad (4.21)$$

For Algorithm 2, the parameters $\alpha$ and $\gamma$ satisfy

$$\alpha < \frac{1}{\beta \|[A_1; A_2]\|^2} \quad \text{and} \quad \gamma \leq 2\beta. \quad (4.22)$$

For the CP algorithm (see (2.45)), we set

$$\alpha < \frac{2}{\beta \|[A_1; A_2]\|^2}. \quad (4.23)$$

For the ZBO algorithm, the parameters $\alpha_1$, $\alpha_2$, and $\gamma$ satisfy

$$\alpha_1 < \frac{1}{\beta}, \quad \alpha_2 < \frac{1}{\beta \|[A_1; A_2]\|^2}, \quad \text{and} \quad \gamma \leq \beta. \quad (4.24)$$

With such settings on the parameters for the algorithms, the convergence of Algorithm 1, Algorithm 2, and the CP method are guaranteed by Theorem 2.13, Theorem 2.11, and a result from [15], respectively. With the given stopping criterion, the parameter $\beta$ in each algorithm is chosen in a way that it would produce better recovered images in terms of PSNR value for image deblurring and $\beta$ is chosen to produce better recovered signal in terms of $\ell_2$-error for compressed sensing.
4.2.2 Numerical Results for Image Deblurring

In this section, numerical experiments for image deblurring are carried out to demonstrate the performance of our proposed Algorithms 1 and 2 for the 256 × 256 test images “Cameraman”, “Peppers”, “Goldhill” and 512 × 512 test image “Lena”. The tolerance $Tol$ in the stopping criterion is chosen to be $10^{-6}$. The quality of the recovered images from each algorithm is evaluated by the peak-signal-to-noise ratio (PSNR), which is defined as $\text{PSNR} := 20 \log_{10} \left( \frac{255}{\| \tilde{x} - x \|} \right)$, where $x \in \mathbb{R}^n$ is the original image and $\tilde{x}$ represents the recovered image. The evolution curve of the function values with respect to iteration will be also adopted to evaluate the performance of algorithms.

In our simulations, blurring matrices $M$ in model (4.1) for image deblurring are generated by a rotationally symmetric Gaussian lowpass filter of size “hsize” with standard deviation “sigma” from the MATLAB script `fspecial('gaussian',hsize,sigma)`. Such matrix $M$ is referred to as the $(hsize,\text{sigma})$-GBM. We remark that the norm of $M$ is always 1, i.e.,

$$\| M \| = 1. \quad (4.25)$$

The (15,10)-GBM and (21,10)-GBM are used to generate blurred images in our experiments. To compute the pixel values under the operation of $M$ and $B$ near the boundary of images, we choose to use “symmetric” type for the boundary extension. Let $B$ be the difference matrix defined by (4.12). We know $\| B \| < \sqrt{8}$. As a result, the parameters are set in the following way.
CHAPTER 4. APPLICATIONS AND NUMERICAL EXPERIMENTS

For Algorithm 1, we set the parameters $\alpha_1$, $\alpha_2$, and $\gamma$ as follows:

$$\alpha_1 := \frac{0.999}{\beta}, \quad \alpha_2 := \frac{1}{8\beta}, \quad \text{and} \quad \gamma := \beta. \quad (4.26)$$

For Algorithm 2, we set the parameters $\alpha$ and $\gamma$ as follows:

$$\alpha := \frac{1}{8\beta} \quad \text{and} \quad \gamma := 2\beta. \quad (4.27)$$

For the CP algorithm (see (2.45)), we set

$$\alpha := \frac{1}{4\beta}. \quad (4.28)$$

For the ZBO algorithm, we set the parameters $\alpha_1$, $\alpha_2$, and $\gamma$ as follows:

$$\alpha_1 := \frac{0.999}{\beta}, \quad \alpha_2 := \frac{1}{8\beta}, \quad \text{and} \quad \gamma := \beta. \quad (4.29)$$

Numerical Results for the $\ell_2$-TV Image Deblurring

In problems of image deblurring with the $\ell_2$-TV model, a noisy image is obtained by blurring an ideal image with a $(hsize, \text{sigma})$-GBM followed by adding white Gaussian noise. Two blurring matrices, namely $(21, 10)$-GBM and $(15, 10)$-GBM, are used in our experiments.

For blurring matrix $(21, 10)$-GBM, the white noise with mean zero and standard deviation 1 is added to blurred images while for blurring matrix $(15, 10)$-GBM, the additive white noise has mean zero and standard deviation 5. The regularization parameter in the $\ell_2$-TV model (4.15) is set as $\mu = 0.02$ for blurring matrices $(21, 10)$-GBM and as $\mu = 0.2$ for blurring matrices $(15, 10)$-GBM. With these settings, numerical results for four test images “Cameraman”, “Lena”, “Peppers”, and “Goldhill” are
CHAPTER 4. APPLICATIONS AND NUMERICAL EXPERIMENTS

reported in Table 4.1 for (21, 10)-GBM and in Table 4.2 for (15, 10)-GBM in terms of numbers of iterations, the CPU times, and the PSNR values. As shown in the tables, Algorithm 1 performs best in terms of computational cost (total iterations and CPU time). The quality of the recovered images from Algorithm 1 is better than or comparable to the quality of recovered images from other algorithms in terms of PSNR values. The evolution curves of function values for each images are shown in Figures 4.1, 4.2, 4.3, and 4.4 for (21, 10)-GBM, and in Figures 4.5, 4.6, 4.7, and 4.8 for (15, 10)-GBM. Also, as shown in the Figures, the sequence of function values at \( \{x^k : k \in \mathbb{N}\} \) generated by Algorithm 1 approaches the minimum value fastest, followed by sequences from Algorithm 2 and then by that from CP and ZBO algorithms. The performance of CP and ZBO algorithms is quite similar in terms of iterations, CPU time, PSNR and evolution of function values.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cameraman</th>
<th>Lena</th>
<th>Peppers</th>
<th>Goldhill</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Itrs</td>
<td>CPU(s)</td>
<td>PSNR</td>
<td>Itrs</td>
</tr>
<tr>
<td>CP</td>
<td>177</td>
<td>18.82</td>
<td>23.09</td>
<td>157</td>
</tr>
<tr>
<td>ZBO</td>
<td>177</td>
<td>19.90</td>
<td>23.09</td>
<td>157</td>
</tr>
<tr>
<td>Alg. 2</td>
<td>151</td>
<td>15.98</td>
<td>23.35</td>
<td>140</td>
</tr>
<tr>
<td>Alg. 1</td>
<td>93</td>
<td>11.34</td>
<td>23.45</td>
<td>84</td>
</tr>
</tbody>
</table>

Table 4.1: Numerical results for the \( \ell_2\)-TV model for images blurred by the (21, 10)-GBM.
CHAPTER 4. APPLICATIONS AND NUMERICAL EXPERIMENTS

Table 4.2: Numerical results for the $\ell_2$-TV model for images blurred by the (15, 10)-GBM.

### Numerical Results for the $\ell_1$-TV Image Deblurring

In problems of image deblurring with the $\ell_1$-TV model, a noisy image is obtained by blurring an ideal image with a ($hsize, sigma$)-GBM followed by adding impulse noise. Two blurring matrices, namely (21, 10)-GBM and (15, 10)-GBM, are used again in our experiments.

For the blurring matrix (21, 10)-GBM, the impulse noise with noise level $p = 0.3$ is added to blurred images while the additive impulsive noise has noise level $p = 0.5$ for the blurring matrix (15, 10)-GBM. We set the regularization parameter $\mu = 0.01$ for (21, 10)-GBM and $\mu = 0.02$ for (15, 10)-GBM in the $\ell_1$-TV model (4.17). With these settings, numerical results for four test images “Cameraman”, “Lena”, “Peppers”, and “Goldhill” are reported in Table 4.3 for (21, 10)-GBM and in Table 4.4 for (15, 10)-GBM in terms of numbers of iterations, the CPU times, and the PSNR values. One can observed from the Tables that Algorithm 1 yields higher PSNR value and consumes less CPU time than Algorithm 2, CP and ZBO algorithms. The evolution curves of function values with respect to iteration are shown in Figure...
Figure 4.1: Evolution of function values for the $\ell_2$-TV model for image “Cameraman” blurred by the (21, 10)-GBM.

4.9, 4.10, 4.11 and 4.12 for (21, 10)-GBM, and in Figure 4.13, 4.14, 4.15 and 4.16 for (15, 10)-GBM. It can be noticed that sequence of function values generated from Algorithm 1 approaches the minimum value fastest. Further, visual quality of the deblurred images is shown in Figure 4.17 and Figure 4.18 for each algorithm. The visual improvement by Algorithm 1 over CP and the ZBO algorithm can be seen by the deblurred images.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cameraman</th>
<th>Lena</th>
<th>Peppers</th>
<th>Goldhill</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Itrs</td>
<td>CPU(s)</td>
<td>PSNR</td>
<td>Itrs</td>
</tr>
<tr>
<td>CP</td>
<td>367</td>
<td>44.39</td>
<td>23.47</td>
<td>354</td>
</tr>
<tr>
<td>ZBO</td>
<td>368</td>
<td>45.09</td>
<td>23.48</td>
<td>355</td>
</tr>
<tr>
<td>Alg. 2</td>
<td>275</td>
<td>29.92</td>
<td>23.57</td>
<td>272</td>
</tr>
<tr>
<td>Alg. 1</td>
<td>192</td>
<td>21.34</td>
<td>24.22</td>
<td>189</td>
</tr>
</tbody>
</table>

Table 4.3: Numerical results for the $\ell_1$-TV model for images blurred by the (21, 10)-GBM.
Figure 4.2: Evolution of function values for the $\ell_2$-TV model for image “Peppers” blurred by the $(21, 10)$-GBM.

### 4.2.3 Numerical Results for the $\ell_1$-Regularized Compressed Sensing

This section will be devoted to the comparison of Algorithm 1 to CP and ZBO for the $\ell_1$-regularized compressed sensing problem (4.18). We assume that the underlying signal is not necessarily sparse itself, but is sparse when mapped to another domain via linear transform.

First of all, we describe how the signals are generated. The underlying signal we
Figure 4.3: Evolution of function values for the $\ell_2$-TV model for image “Lena” blurred by the $(21,10)$-GBM. The considered function is obtained by sampling the piecewise linear function

$$s(t) = \begin{cases} 
1 + 2t, & 0 \leq t \leq 1 \\
3, & 0 < t \leq 2 \\
5 - t, & 2 < t \leq 3 \\
2, & 3 < t \leq 4 \\
6 - t, & 4 < t \leq 5 
\end{cases} \quad (4.30)$$

To obtain the original test signal, we take 512 sample points with equal width from the function (4.30). The original test signal is shown in Figure 4.19. The $m_1 \times n$ random matrix $M$ whose entries are i.i.d. from standard normal distribution $\mathcal{N}(0, 1)$ is given in advance. The norm of $M$ denoted by $\|M\|$ can be calculated by MATLAB script `norm(M)`. The observed signal is acquired after the underlying signal passes by the measurement matrix $M$ and is contaminated by Gaussian noise, which is modeled...
Figure 4.4: Evolution of function values for the $\ell_2$-TV model for image “Goldhill” blurred by the $(21,10)$-GBM.

by equation (4.1).

Further, to use model (4.18), an appropriate matrix $T$ by which the original signal is mapped to a sparse signal, and the parameter $\epsilon$ indicating the noise power need be determined. With the given original signal as in Figure 4.19, a good choice of $T$ will be a matrix that represents the high-pass filters of a framelet. Specifically, the high-pass filters associated with $T$ are chosen as $h_1 = \frac{\sqrt{2}}{4}[1,0,-1]$ and $h_2 = \frac{1}{4}[-1,2,-1]$. With symmetric extension on the boundary of the signal, the matrix $T$ of size $2n \times n$
Figure 4.5: Evolution of function values for the $\ell_2$-TV model for image “Cameraman” blurred by the (15,10)-GBM.

associated with those two high-pass filters has explicit form

$$T = \begin{bmatrix}
\sqrt{2}/4 & -\sqrt{2}/4 \\
\sqrt{2}/4 & 0 & -\sqrt{2}/4 \\
\sqrt{2}/4 & 0 & -\sqrt{2}/4 & \cdots & \cdots \\
1/4 & -1/4 \\
-1/4 & 2/4 & -1/4 & \cdots & \cdots \\
-1/4 & 2/4 & -1/4 \\
-1/4 & 1/4 \\
\end{bmatrix}$$

(4.31)

where the upper half corresponds to the high-pass filter $h_1$ and the lower half corre-
Figure 4.6: Evolution of function values for the $\ell_2$-TV model for image “Peppers” blurred by the (15,10)-GBM.

Corresponds to the other high-pass filter $h_2$. The norm $\|T\| \leq 1$ since $T$ only contains the part of high-pass filters of a tight frame system. The parameter of noise power is set as $\epsilon = \sqrt{m_1}\sigma$, where $\sigma$ represents the variance of the Gaussian noise. Regarding the variance of the noise, we will choose $\sigma = 0.05$ and $\sigma = 0$ in our experiment. In the case $\sigma = 0$, it implies that the observed signal is noise free. For different noise level, we set $Tol$ differently. When $\sigma = 0.05$ $Tol$ is set to be $10^{-4}$, while it is set to be $10^{-6}$ when $\sigma = 0$. The maximal number of iterations allowed for each algorithm is set to 5,000.

To solve model (4.18), Algorithm 1, CP and ZBO will be applied. The setup of parameters $\alpha$ introduced in those algorithms are chosen in the similar manner as in image deblurring discussed earlier. That is, if $\beta$ is assumed to be predetermined, $\alpha$’s and $\gamma$’s are set to be as large as possible under the condition that the convergence
Figure 4.7: Evolution of function values for the $\ell_2$-TV model for image “Lena” blurred by the (15,10)-GBM.

is guaranteed. The parameter $\beta$ for each algorithm is chosen such that it will yield smaller relative $\ell_2$-error. The setting of parameters is described as follow.

For Algorithm 1, the positive parameters $\alpha_1$, $\alpha_2$, and $\gamma$ satisfy

$$\alpha_1 := \frac{0.999}{\beta \|M\|^2}, \quad \alpha_2 := \frac{0.999}{\beta}, \quad \text{and} \quad \gamma := \beta. \quad (4.32)$$

For the CP algorithm (see (2.45)), we set

$$\alpha := 2 \frac{0.999}{\beta \|[M;T]\|^2}. \quad (4.33)$$

For the ZBO algorithm, the parameters $\alpha_1$, $\alpha_2$, and $\gamma$ satisfy

$$\alpha_1 := \frac{0.999}{\beta}, \quad \alpha_2 := \frac{0.999}{\beta \|[M;T]\|^2}, \quad \text{and} \quad \gamma := \beta. \quad (4.34)$$

Numerical experiment is conducted on different settings on the $m \times n$ measurement matrix $M$ and on the variance $\sigma$ of the Gaussian noise. With the given original
CHAPTER 4. APPLICATIONS AND NUMERICAL EXPERIMENTS

Figure 4.8: Evolution of function values for the $\ell_2$-TV model for image “Goldhill” blurred by the (15,10)-GBM.

signal shown in Figure 4.19, one should notice that $n = 512$. For each setting of $m_1$ and $\sigma$, Table 4.5 reports the performance of each algorithms in terms of iteration, CPU time consumed, relative $\ell_2$-error and absolute error under the same stopping criterions. Figures 4.20, 4.21, 4.22 and 4.23 show the evolution curves of function values with respect to iteration. Figures 4.24, 4.25, 4.26 and 4.27 show the relative $\ell_2$-error respectively with respect to iteration. One can easily observe that Algorithm 1 outperforms CP and ZBO algorithms dramatically for all of the metrics. Smaller relative $\ell_2$-error and function values can be achieved much faster from Algorithm 1.
CHAPTER 4. APPLICATIONS AND NUMERICAL EXPERIMENTS

88

Table 4.4: Numerical results for the $\ell_1$-TV model for images blurred by the (15, 10)-GBM.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cameroon</th>
<th>Lena</th>
<th>Peppers</th>
<th>Goldhill</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP</td>
<td>280</td>
<td>29.78</td>
<td>23.77</td>
<td>289</td>
</tr>
<tr>
<td>ZBO</td>
<td>281</td>
<td>31.51</td>
<td>23.77</td>
<td>290</td>
</tr>
<tr>
<td>Alg. 1</td>
<td>147</td>
<td>16.10</td>
<td>24.42</td>
<td>149</td>
</tr>
</tbody>
</table>

Table 4.5: Numerical results for the $\ell_1$-regularized compressed sensing.

4.2.4 Numerical Results for the $\ell_p$-Regularized Compressed Sensing

This section is devoted to showing the numerical performance of the $\ell_p$-regularization for compressed sensing. We demonstrate numerically that the $\ell_p$-regularization with $0 < p < 1$ often performs better than the $\ell_1$-regularization for compressed sensing in terms of the quality of recovered sparse signals.

We begin with a description on the sensing matrix and sparse signals. In our experiments, the sensing matrix $M$ of size $m \times n$ ($m < n$) is a random matrix whose elements are generated independently from standard normal distribution $\mathcal{N}(0, 1)$. According to [4], a length-$n$, $s$-sparse signal (a signal having exactly $s$ nonzero com-
CHAPTER 4. APPLICATIONS AND NUMERICAL EXPERIMENTS

Figure 4.9: Evolution of function values for the $\ell_1$-TV model for image “Cameraman” blurred by the (21, 10)-GBM.

ponents) $x$ in our experiments, is generated in such a way that non-zero components are given by

$$
\eta_1 10^\theta,
$$

(4.35)

where $\eta_1 = \pm 1$ with probability $1/2$ and $\theta$ is uniformly distributed in $[0, 1]$. The locations of the nonzero components are randomly permuted. Clearly, the range of the magnitude of nonzero components of an $s$-sparse signal is $[1, 10]$. The sparsity $s$ is chosen to be $0.01n$. The noisy observed signal $y \in \mathbb{R}^m$ is obtained by equation (4.1), where the entries in the noise vector $\eta$ are i.i.d. from a Gaussian distribution with mean zero and standard variance $\sigma$. The standard deviation $\sigma$ is chosen as 0.05.

The underlying sparse signal $x$ to be recovered is formulated as a solution to the $\ell_p$-regularized minimization problem (4.19), where $\epsilon = \sqrt{m} \sigma$ denotes the upper bound of noise power. The extension of an algorithm proposed in [20] to solve problem
Figure 4.10: Evolution of function values for the $\ell_1$-TV model for image “Peppers” blurred by the (21,10)-GBM.

(4.19) with $p < 1$ is described in Algorithm 4. The key step in the implementation of Algorithm 4 is to evaluate the proximity operator of the $\ell_p$-norm which can be efficiently computed by Algorithm 3. We remark that the convergence analysis of Algorithm 4 was already given in [20] for $p = 1$, but is unknown for $p < 1$ due to the difficulty caused by the non-convexity of the corresponding cost function of problem (4.19).

In our numerical experiments, we will investigate the performance of model (4.19) using Algorithm 4 for various values of $p \in \{0, 1/2, 2/3, 4/5, 1\}$. The maximum number of iterations of Algorithm 4 is set to be 1000. The accuracy of a solution obtained from Algorithm 4 with a specific value of $p$ is quantified by the relative $\ell_2$-error and the absolute $\ell_\infty$-error defined, respectively, as follows:

$$\|x - x_\diamond\|/\|x\| \text{ and } \|x - x_\diamond\|_\infty,$$

(4.36)
Algorithm 4: Algorithm for model (4.19)

**Input**: Initialization: \( v^0 \in \mathbb{R}^m, x^0 \in \mathbb{R}^n, \epsilon > 0, \alpha > 0, \text{ and } \beta > 0 \) with
\[
\frac{\beta}{\alpha} < \frac{1}{\|M\|_2} \Rightarrow \text{set } v^{-1} = v^0 - (Ax^0 - y);
\]

**Result**: \( x^\infty \)

while it is not convergent do

Step 1:
\[
x^{k+1} \leftarrow T_{\frac{1}{\alpha}} \left( x^k - \frac{\beta}{\alpha} A^T (2v^k - v^{k-1}) \right)
\]

Step 2: Denote \( p^k := Ax^{k+1} + v^k - y \).

\[
v^{k+1} \leftarrow \begin{cases} 
0, & \text{if } \|p^k\|_2 < \epsilon; \\
\left(1 - \frac{\epsilon}{\|p^k\|_2}\right) (p^k), & \text{otherwise.}
\end{cases}
\]
where \( x \) is the true data and \( x_\diamond \) is the restored data by Algorithm 4. All those errors reported in this section are the means and standard derivation of these relative errors from simulations that were performed 50 trials.

To use Algorithm 4, one needs to fix the parameters \( \alpha \) and \( \beta \) such that \( \beta/\alpha < \frac{1}{\|A\|^2} \). It has been demonstrated numerically in [20] that Algorithm 4 for \( p = 1 \) performs best in terms of the errors in (4.36) for a large ratio \( \beta/\alpha \). Therefore, we set \( \beta = \frac{0.999}{\|A\|^2} \alpha \) in our numerical experiments. In such the way, \( \alpha \) is essentially the only parameter that needs to be determined. The parameter \( \alpha \) is chosen such that it would produce relatively optimal average error over the 50 trials.

The parameters used in our experiments are \( n = 4096 \), \( m \in \{256, 512\} \) and \( p \in \{0, 1/2, 2/3, 4/5, 1\} \). The numerical results over 50 trials are reported in Table 4.6 and Figure 4.28, 4.29. For \( m = 512 \), one can see that the performance of Algorithm
Figure 4.12: Evolution of function values for the $\ell_1$-TV model for image “Goldhill” blurred by the (21,10)-GBM.

4 with $p \in \{0, 1/2, 2/3, 4/5\}$ is comparable, but better than that with $p = 1$. For $m = 256$, the performance of Algorithm 4 with $p \in \{1/2, 2/3, 4/5\}$ is comparable, but better than that with $p \in \{0, 1\}$. We can conclude that Algorithm 4 with $0 < p < 1$ performs superiorly to that with $p = 0$, 1 in terms of accuracy and robustness, particularly, in the scenario of a small number of measurements.
Figure 4.13: Evolution of function values for the $\ell_1$-TV model for image “Cameraman” blurred by the (15, 10)-GBM.

Figure 4.14: Evolution of function values for the $\ell_1$-TV model for image “Peppers” blurred by the (15, 10)-GBM.
Figure 4.15: Evolution of function values for the $\ell_1$-TV model for image “Lena” blurred by the (15, 10)-GBM.

Figure 4.16: Evolution of function values for the $\ell_1$-TV model for image “Goldhill” blurred by the (15, 10)-GBM.
Figure 4.17: Recovered images of “Cameraman”, “Lena”, “Peppers”, and “Goldhill” (from top row to bottom row) with the $\ell_1$-TV model for images blurred by the (21, 10)-GBM and corrupted by impulsive noise of level $p = 0.3$. Row 1: the CP; Row 2: ZBO; Row 3: Algorithm 2; Row 4: Algorithm 1.
Figure 4.18: Recovered images of “Cameraman”, “Lena”, “Peppers”, and “Goldhill” (from top row to bottom row) with the $\ell_1$-TV model for images blurred by the (15, 10)-GBM and corrupted by impulsive noise of level $p = 0.5$. Row 1: the CP; Row 2: the ZBO; Row 3: Algorithm 2; Row 4: Algorithm 1.
Figure 4.19: Original test signal for the $\ell_1$-regularized compressed sensing.

Figure 4.20: Evolution of function values for model (4.18) of compressed sensing with respect to iteration. $m_2 = n/2$, $\sigma = 0.05$. 
Figure 4.21: Evolution of function values for model (4.18) of compressed sensing with respect to iteration. $m_2 = n/4$, $\sigma = 0.05$.

Figure 4.22: Evolution of function values for model (4.18) of compressed sensing with respect to iteration. $m_2 = n/2$, $\sigma = 0$. 
Figure 4.23: Evolution of function values for model (4.18) of compressed sensing with respect to iteration. $m_2 = n/4$, $\sigma = 0$.

Figure 4.24: Evolution of relative $\ell_2$-error of recovered signal with respect to iteration. $m_2 = n/2$, $\sigma = 0.05$
Figure 4.25: Evolution of relative $\ell_2$-error of recovered signal with respect to iteration. $m_2 = n/4, \sigma = 0.05$.

Figure 4.26: Evolution of relative $\ell_2$-error of recovered signal with respect to iteration. $m_2 = n/2, \sigma = 0$. 
Figure 4.27: Evolution of relative $\ell_2$-error of recovered signal with respect to iteration.

$m_2 = n/4$, $\sigma = 0$.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$\ell_2$-error</th>
<th>$\ell_\infty$-error</th>
<th>$\ell_2$-error</th>
<th>$\ell_\infty$-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(2.2978e-1, 1.5371e-1)</td>
<td>(2.5759e0, 9.5095e-1)</td>
<td>(5.0219e-4, 7.0712e-5)</td>
<td>(5.6873e-3, 1.0695e-3)</td>
</tr>
<tr>
<td>1/2</td>
<td>(3.2857e-3, 1.1173e-2)</td>
<td>(4.1458e-2, 1.6342e-1)</td>
<td>(5.5597e-4, 1.2796e-4)</td>
<td>(6.2438e-3, 1.3898e-3)</td>
</tr>
<tr>
<td>2/3</td>
<td>(1.1604e-3, 1.650e-4)</td>
<td>(1.2177e-2, 2.3400e-3)</td>
<td>(6.6470e-4, 1.5286e-4)</td>
<td>(7.2339e-3, 1.4871e-3)</td>
</tr>
<tr>
<td>4/5</td>
<td>(1.3583e-3, 3.4781e-4)</td>
<td>(1.3723e-2, 3.8123e-3)</td>
<td>(7.4240e-4, 1.3873e-4)</td>
<td>(7.7716e-3, 1.4080e-3)</td>
</tr>
<tr>
<td>1</td>
<td>(1.0070e-1, 1.3429e-1)</td>
<td>(7.2380e-1, 9.5382e-1)</td>
<td>(1.7355e-3, 1.9504e-4)</td>
<td>(1.3659e-2, 1.5971e-3)</td>
</tr>
</tbody>
</table>

Table 4.6: The pairs of the means and standard derivation of the relative $\ell_2$-error and the absolute $\ell_\infty$-error over 50 simulations are given for the recovery for each value of $p$. 
Figure 4.28: Errors of recovered data for each of 50 trials. Row 1-5 represents the performance of Algorithm 4 for $p = 0, 1/2, 2/3, 4/5, 1$ respectively. The first and second column represents the relatively $\ell_2$-error, absolute $\ell_\infty$-error respectively. The setting for $m, n$ is $n = 4096, m = 256$. 
Figure 4.29: Errors of recovered data for each of 50 trials. Row 1-5 represents the performance of Algorithm 4 for $p = 0, 1/2, 2/3, 4/5, 1$ respectively. The first and second column represents the relatively $\ell_2$-error, absolute $\ell_\infty$-error respectively. The setting for $m, n$ is $n = 4094, m = 512$. 
Chapter 5

Future Research

The following lines of research are proposed as ways of further advancing understanding of composite optimization problems and the $\ell_p$-regularization:

1. *The rate of convergence of Algorithm 1.* In this thesis, the convergence analysis of Algorithm 1 has been studied. It is worthy to investigate in what rate the proposed algorithm will converge to a desirable solution. It is also interesting to know if the Nesterov acceleration technique can be adapted to Algorithm 1 with an improved rate of convergence.

2. *Composite convex optimization problems with three or more terms.* In our current study, we developed algorithms for composite convex optimization problems with two terms. This might limit its applications. For example, in compressed sensing MRI, the function to be minimized in [51, 52] is the sum of three terms, namely, a sparsity promoting term, a total variation regularization
CHAPTER 5. FUTURE RESEARCH

106

term, and a fidelity term that describes the data consistency. Hence, there is a practical need to extend our current research to composite convex optimization problems with three or more terms.

3. Convergence analysis for the $\ell_p$-regularization. In our current research, the convergence analysis of Algorithm 4 is missing due to the difficulty caused by the nonconvexity of the $\ell_p$-norm with $0 \leq p < 1$, even though Algorithm 4 performs well in our numerical experiments. Therefore, convergence analysis of Algorithm 4 will provide theoretical guarantee of its numerical performance. Its impact may go beyond the current research context.
Chapter 6

Published and Completed Research Work

Research Work Has Been Published


Research Work Has Been Completed

Bibliography


BIBLIOGRAPHY


BIOGRAPHICAL DATA

NAME OF AUTHOR: Feishe Chen

PLACE OF BIRTH: Zhanjiang, China.


GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

Syracuse University, NY, USA

Sun Yat-sen University, Guangzhou, China

DEGREES AWARDED:

M.S., Sun Yat-sen University, 2009

B.S., Sun Yat-sen University, 2007