

AN ADAPTIVE ℓ_1 - ℓ_2 -TYPE MODEL WITH HIERARCHIES FOR SPARSE SIGNAL RECONSTRUCTION PROBLEM*

YANYUN DING, ZHIXIAO YUE AND HAIBIN ZHANG[†]

Abstract: This paper addresses solving an adaptive ℓ_1 - ℓ_2 regularized model in the framework of hierarchical convex optimization for sparse signal reconstruction. This is realized in the framework of bi-level convex optimization, we can also turn the challenging bi-level model into a single-level constrained optimization problem through some priori information. The ℓ_1 - ℓ_2 norm regularized least-square sparse optimization is also called the elastic net problem, and numerous simulation and real-world data show that the elastic net often outperforms the Lasso. However, the elastic net is suitable for handling Gaussian noise in most cases. In this paper, we propose an adaptive and robust model for reconstructing sparse signals, say ℓ_p - ℓ_1 - ℓ_2 , where the ℓ_p -norm with $p \geq 1$ measures the data fidelity and ℓ_1 - ℓ_2 -term measures the sparsity. This model is robust and flexible in the sense of having the ability to deal with different types of noises. To solve this model, we employ an alternating direction method of multipliers (ADMM) based on introducing one or a pair of auxiliary variables. From the point of view of numerical computation, we use numerical experiments to demonstrate that both of our proposed model and algorithms outperform the Lasso model solved by ADMM on sparse signal reconstruction problem.

Key words: convex optimization, sparse signal reconstruction, hierarchical optimization, ℓ_p - ℓ_1 - ℓ_2 minimization, alternating direction method of multipliers

Mathematics Subject Classification: 15A04, 90-08, 90C25

1 Introduction

In this paper, we consider the following ℓ_1 - ℓ_2 specific convex bi-level optimization model:

$$\min_x \|x\|_1 + \frac{\beta}{2} \|x\|_2^2, \quad \text{subject to } x \in \mathcal{X}, \quad (1.1)$$

where $x \in \mathbb{R}^n$, $\beta \geq 0$ and \mathcal{X} is the feasible set. The above optimization task can not only extract sparsity but also select variable [26]. It is well known that compressive sensing (CS) is to sense sparse signals by acquiring a set of incomplete or even noise polluted measurements. Let $\underline{x} \in \mathbb{R}^n$ be a compressible or (approximately) sparse signal. The main idea of CS is to project \underline{x} onto a certain subspace via a linear operator A , i.e, $A\underline{x} = \bar{y}$ with $\bar{y} \in \mathbb{R}^m$ and $m \ll n$, and then reconstruct \underline{x} from the data \bar{y} . The feasible set \mathcal{X} can be formulated as the minimizers of another optimization model:

$$\mathcal{X} = \arg \min_x \left\{ \frac{1}{2} \|Ax - \bar{y}\|_2^2 \right\}.$$

*This work was supported by the National Natural Science Foundation of China (No. 11771003).

[†]Corresponding author.

However, in the process of acquiring, storing, transmitting, or displaying, the observation may be inevitably degenerated by various noises, such as heavy-tailed noise, Gaussian noise or uniformly distributed noise, etc. Actually, \bar{y} is not a real observation and not unique even under the same experimental conditions. The challenges of problem (1.1) come not only from the non-smoothness of the upper-level problem [24, 25] but also from the uncertainty of observation. Fortunately, we can utilize some prior knowledge to generate warm-start to guide the optimization process. Suppose \hat{x} is an approximation lower-level solution of (1.1) obtained by some prior knowledge. Following [17, Theorem 1], we define $b = A\hat{x}$, then the original bi-level model (1.1) can be equivalently reformulated as the following single-level constrained optimization problem:

$$\min_x \|x\|_1 + \frac{\beta}{2} \|x\|_2^2 \quad \text{subject to} \quad Ax = b. \quad (1.2)$$

The penalized form of the (1.2), in which the linear constraints are penalized by using a square of its ℓ_2 -norm,

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \|Ax - b\|_2^2 + \lambda(\|x\|_1 + \frac{\beta}{2} \|x\|_2^2), \quad (1.3)$$

where $\lambda > 0$ is a weighting parameter to balance both terms for minimization. This model also called the elastic net problem is proposed for variable selection and regularization in statistical analysis [36], which reduces to the Lasso problem [26] if $\beta = 0$. Although the Lasso has shown success in many situations [1, 6], it has some limitations and shortcomings [36, 26], for instance, the Lasso tends to select only one variable from the group but does not care which one is selected when there is a group of variables among which the pairwise correlations are very high [26]. Furthermore, Problem (1.3) has a ℓ_2 norm in the regularizer term, so it is less sensitive to noise than the Lasso problem. Compared to the Lasso, the elastic net simultaneously does automatic variable selection and continuous shrinkage, and it can select groups of correlated variables. The exact regularization property of ℓ_1 - ℓ_2 -type penalty has been studied in many literatures [20, 33], which has been extensively used in many fields, such as uncovering the consistent networks of functional disconnection in Alzheimers disease [23], estimating global bank network connectedness [10], neuroimaging [5], genome analysis [8] and so on.

The goal of this paper is to design a more flexible and robust reconstruction model which inherits the excellent properties of the regularizer term of elastic net model, as well as unified efficient algorithms with convergence guarantee to be capable of dealing with the different types of noise. Therefore, we consider the following more generalized ℓ_p - ℓ_1 - ℓ_2 minimization problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_p + \lambda(\|x\|_1 + \frac{\beta}{2} \|x\|_2^2), \quad (1.4)$$

where $\beta \geq 0$ and $\|\cdot\|_p$ is a ℓ_p -norm function with $p \geq 1$, e.g., $p = 1, 2, \infty$. The data fidelity with form $\|Ax - b\|_2$, say square-root-loss stated by Belloni et al. [4] is proved to be achieving the near-oracle rates of convergence without knowing the standard deviation of the noise under suitable design conditions [2, 9]. In addition, the data fidelity with form $\|Ax - b\|_1$ has also been evidently shown to be more robust than the least-square form when encountering not normal but heavy-tailed or heterogeneous noises [3, 19, 27, 30]. Meanwhile, the data fidelity with form $\|Ax - b\|_\infty$ is also known to be very suitable for dealing with the uniformly distributed noise and quantization error [31, 35]. Therefore, the model (1.4) has many attractive features. Not only does it have the advantage of the elastic net for variable selection, but the $\|Ax - b\|_p$ data fidelity term makes it can deal with different types of noise if p is chosen adaptively. For example, $p = 1$ for log-normal noise as well as heavy-tailed

noise, $p = 2$ for Gaussian noise but the quality of the solutions derived does not relies on the knowing of the standard deviation of the noise, and $p = \infty$ for uniformly distributed noise [11].

Nevertheless, model (1.4) is more difficult and challenging to solve due to the non-differentiability of the ℓ_p -norm comparing with (1.3). To tackle the tricky trouble, we design two algorithms based on alternating direction method of multipliers (ADMM) that can fully exploit the structures of the problem (1.4). For many convex optimization problems with separable structures, the ADMM is a widely applicable, easy to understand and implement method [32, 15]. The first algorithm for solving (1.4) is the semi-proximal ADMM (semi-ADMM) [13] by introducing a variable. Recently, the semi-ADMM has solved a series of problems via their favorable structures and achieved good numerical performance, see e.g., [12, 29, 28]. Another algorithm uses the direct extension of ADMM for a 3-block convex minimization problem on the introduction of a pair of variables, and the convergence result can be easily followed from [7]. In fact, the second algorithm also falls into the semi-ADMM framework, so the convergence results can also be obtained from [13]. We show that each subproblem involved in both algorithms is easily performed. In addition, some numerical experiments in signal reconstruction demonstrate that both of our proposed model and algorithms outperform the Lasso and the corresponding algorithm solved by ADMM.

The remaining parts of this paper are organized as follows. In Section 2, we summarize some basic definitions or concepts for subsequent arithmetic design. In Section 3, we apply the semi-ADMM to solve the model (1.4), the convergence results for the proposed algorithm under certain conditions are also included. In Section 4, we turn our attention to the application of direct extension of ADMM for 3-block convex problem evolved from (1.4) under the case of introducing two auxiliary variables. Numerical experiments and performance comparisons are reported in Section 5. Finally, the paper is concluded with some remarks in Section 6.

2 Preliminaries

In this section, we summarize some basic concepts in convex analysis for subsequent developments. Let \mathcal{X} be a finite dimensional real Euclidean space equipped with an inner product $\langle \cdot, \cdot \rangle$ and its induced norm $\|\cdot\|_2$. A subset \mathcal{C} of \mathcal{X} is said to be convex if $(1 - \lambda)x + \lambda y \in \mathcal{C}$ whenever $x \in \mathcal{C}$, $y \in \mathcal{C}$, and $0 \leq \lambda \leq 1$. Let $\|\cdot\|$ be a norm function defined on \mathcal{X} . Then its dual norm $\|\cdot\|_*$ is defined as:

$$\|x\|_* = \sup_y \{x^\top y \mid \|y\| \leq 1\}.$$

It is easy to see that the dual norm of ℓ_1 -norm is ℓ_∞ -norm, and the dual norm of ℓ_2 -norm is ℓ_2 -norm itself. Give $f : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ be a proper closed convex function. We use $\text{dom}(f)$ to denote the domain of f , that is, $\text{dom}(f) = \{x \in \mathbb{R}^n \mid f(x) < \infty\}$. The Fenchel conjugate of f is defined by $f^*(x^*) := \sup_{x \in \mathbb{R}^n} \{\langle x, x^* \rangle - f(x)\}$. Denote by $\Phi_{tf}(x)$ the Moreau envelope function [21, 34] of f with parameter $t > 0$,

$$\Phi_{tf}(x) := \min_{y \in \mathbb{R}^n} \{f(y) + \frac{1}{2t} \|y - x\|_2^2\}, \quad \forall x \in \mathbb{R}^n.$$

The proximal mapping of f with $t > 0$ is defined by

$$\text{Prox}_f(x) := \underset{y \in \mathbb{R}^n}{\text{argmin}} \{f(y) + \frac{1}{2t} \|y - x\|_2^2\}, \quad \forall x \in \mathbb{R}^n.$$

Then the following Moreau’s identity theorem [22, Theorem 35.1] will be used in the subsequent analysis:

$$\text{Prox}_{tf}(x) + t\text{Prox}_{f^*/t}(x/t) = x. \tag{2.1}$$

Let $\Pi_{\mathcal{C}}(z)$ denote the metric projection of z onto \mathcal{C} , which is the optimal solution of the minimization problem $\min_y \{\|y - z\|^2 \mid y \in \mathcal{C}\}$. For a nonempty closed convex set \mathcal{C} , the symbol $\delta_{\mathcal{C}}(x)$ represents the indicator function over \mathcal{C} such that $\delta_{\mathcal{C}}(x) = 0$ if $x \in \mathcal{C}$ and $+\infty$ otherwise. The Fenchel conjugate of an indicator function $\delta_{\mathcal{C}}(x)$ is named as support function defined by $\delta_{\mathcal{C}}^*(x) = \sup\{\langle x, y \rangle \mid y \in \mathcal{C}\}$. It is not hard to deduce that the Fenchel conjugate of $\|x\|_p$ is $\|x\|_p^* = \delta_{B_q^{(1)}}(x)$ where $B_q^{(1)} := \{x \mid \|x\|_q \leq 1\}$ and $1/p + 1/q = 1$.

Next, we summarize some well-known results in optimization literature, which play key roles in the algorithmic construction. The following Lemma 2.1 reports some applications of the Moreau’s identity (2.1) for some typical norm functions. We omit its proof here owing to that the following results are known in many optimization literatures [22].

Lemma 2.1. *For any $x^* \in \mathbb{R}^n$, it holds that*

(a) *Let $f(x) := \mu\|x\|_1$ with $\mu > 0$, then $f^*(x^*) = \delta_{B_{\infty}^{(\mu)}}(x^*)$ with $B_{\infty}^{(\mu)} := \{x^* \mid \|x^*\|_{\infty} \leq \mu\}$ and*

$$\text{Prox}_f(x^*) = x^* - \Pi_{B_{\infty}^{(\mu)}}(x^*) \quad \text{with} \quad (\Pi_{B_{\infty}^{(\mu)}}(x^*))_i = \begin{cases} x_i^*, & \text{if } |x_i^*| \leq \mu, \\ \text{sign}(x_i^*)\mu, & \text{if } |x_i^*| > \mu, \end{cases}$$

where $i = 1, \dots, n$ and $\text{sign}(\cdot)$ is a sign function of a vector.

(b) *Let $f(x) := \mu\|x\|_2$ with $\mu > 0$, then $f^*(x^*) = \delta_{B_2^{(\mu)}}(x^*)$ with $B_2^{(\mu)} := \{x^* \mid \|x^*\|_2 \leq \mu\}$ and*

$$\text{Prox}_f(x^*) = x^* - \Pi_{B_2^{(\mu)}}(x^*) \quad \text{with} \quad \Pi_{B_2^{(\mu)}}(x^*) = \begin{cases} x^*, & \text{if } \|x^*\|_2 \leq \mu, \\ \mu \frac{x^*}{\|x^*\|_2}, & \text{if } \|x^*\|_2 > \mu. \end{cases}$$

(c) [16] *Let $f(x) := \mu\|x\|_{\infty}$ with $\mu > 0$, then $f^*(x^*) = \delta_{B_1^{(\mu)}}(x^*)$ with $B_1^{(\mu)} := \{x^* \mid \|x^*\|_1 \leq \mu\}$ and*

$$\text{Prox}_f(x^*) = x^* - \Pi_{B_1^{(\mu)}}(x^*) \quad \text{with} \quad \Pi_{B_1^{(\mu)}}(x^*) = \begin{cases} x^*, & \text{if } \|x^*\|_1 \leq \mu, \\ \mu P_{x^*} \Pi_{\Delta_n}(P_{x^*} x^* / \mu), & \text{if } \|x^*\|_1 > \mu. \end{cases}$$

where $P_{x^*} := \text{Diag}(\text{sign}(x^*))$ and $\Pi_{\Delta_n}(\cdot)$ denotes the projection onto the simplex $\Delta_n := \{x \in \mathbb{R}^n \mid e_n^T x = 1, x \geq 0\}$, in which $\text{Diag}(\cdot)$ denotes a diagonal matrix with elements of a given vector on its diagonal positions.

We are now ready to illustrate how to turn model (1.1) to model (1.2) based on some prior knowledge inspired by [17]. Focus on the following composite optimization model:

$$\min_{\mathbf{x}} F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}), \tag{2.2}$$

where both $f, g: \mathbb{R}^n \rightarrow (-\infty, \infty]$ are extended-valued convex functions and g is possibly nonsmooth. The latent feasible set of (2.2) can be formulated as the minimizers of another optimization model:

$$\min_{\mathbf{x}} \Psi(\mathbf{x}) = h(\mathcal{A}\mathbf{x}), \tag{2.3}$$

where \mathcal{A} is some given linear operator and function h is closed, proper, convex and admits the properties that (i) h is continuously differentiable on $\text{dom } h$, assumed to be open, and

(ii) h is local strongly convex on $\text{dom } h$. Therefore, the problem (2.2) amount to solving the following convex bi-level optimization model:

$$\min_x F(x) \text{ subject to } x \in \arg \min_x \Psi(x). \tag{2.4}$$

We are now ready to state the following theorem in [17] to investigate the feasibility of our problem.

Theorem 2.2. ([17, Theorem 1]) *Let \mathcal{S} be the solution set of (2.3) (i.e., $\mathcal{S} := \arg \min_x \Psi(x)$), then \mathcal{A} is invariant on \mathcal{S} . That is, given any $\bar{x} \in \mathcal{S}$, \mathcal{S} can be explicitly characterized as $\mathcal{S} = \{x | \mathcal{A}(x) = \mathcal{A}(\bar{x})\}$.*

When choose $\bar{y} = \mathcal{A}(\bar{x})$, the original bilevel model in (2.4) can be equivalently reformulated as the following single-level constrained optimization problem:

$$\min_x f(x) + g(x), \text{ subject to } \mathcal{A}(x) = \bar{y}.$$

Following Theorem 2.2, we define $b = A\hat{x}$ for any $\hat{x} \in \mathcal{X}$ in the model (1.1), then it can be equivalently reformulated as a single-level constrained optimization problem

$$\min_x \|x\|_1 + \frac{\beta}{2} \|x\|_2^2 \quad \text{subject to} \quad Ax = b.$$

Therefore, we only need to solve the above single-level convex optimization problem by some priori knowledge or information.

At the end of this part, we turn to briefly review the content of ADMM for the design of subsequent algorithms. Let $\mathbb{X}, \mathbb{Y}, \mathbb{Z}$ be finite dimensional real Euclidian spaces. Consider the convex optimization problem

$$\begin{aligned} \min_{y,z} \quad & f(y) + g(z) \\ \text{s.t.} \quad & \mathcal{A}^*y + \mathcal{B}^*z = c, \end{aligned} \tag{2.5}$$

where $f : \mathbb{Y} \rightarrow (-\infty, +\infty]$ and $g : \mathbb{Z} \rightarrow (-\infty, +\infty]$ are closed proper convex functions, $\mathcal{A} : \mathbb{X} \rightarrow \mathbb{Y}$ and $\mathcal{B} : \mathbb{X} \rightarrow \mathbb{Z}$ are given linear maps, and $c \in \mathbb{X}$ is given data. The augmented Lagrangian function associated with (2.5) is

$$\mathcal{L}_\sigma(y, z; x) = f(y) + g(z) + \langle x, \mathcal{A}^*y + \mathcal{B}^*z - c \rangle + \frac{\sigma}{2} \|\mathcal{A}^*y + \mathcal{B}^*z - c\|^2,$$

where $x \in \mathbb{X}$ is a multiplier and $\sigma > 0$ is penalty parameter. Starting from $(x^0, y^0, z^0) \in \mathbb{X} \times (\text{dom}(f)) \times (\text{dom}(g))$, the semi-proximal ADMM of Fazel et al. [13] for solving (2.5) takes the following form

$$\begin{cases} y^{k+1} = \operatorname{argmin}_y \mathcal{L}_\sigma(y, z^k; x^k) + \frac{1}{2} \|y - y^k\|_{\mathcal{S}}^2, \\ z^{k+1} = \operatorname{argmin}_z \mathcal{L}_\sigma(y^{k+1}, z; x^k) + \frac{1}{2} \|z - z^k\|_{\mathcal{T}}^2, \\ x^{k+1} = x^k + \xi \tau (\mathcal{A}^*y^{k+1} + \mathcal{B}^*z^{k+1} - c), \end{cases} \tag{2.6}$$

where $\xi \in (0, (1 + \sqrt{5})/2)$ is a step length, and \mathcal{S} and \mathcal{T} are self-adjoint positive semi-definite linear operators. The convergence result of the iterative scheme (2.6) under some constraint qualifications can be found in Theorem B.1 of [13]. However, the direct extension of ADMM is not necessarily convergent to multi-block minimization problem where its objective function is the sum of more than two separable convex functions [7, 14].

3 Semi-proximal ADMM for Solving (1.4)

The popular first-order ADMM can be applied to solve (1.4). In this section, we introduce the implementation details of the semi-proximal ADMM for the problem (1.4). We first introduce an auxiliary variable $y := Ax - b$ and reformulate (1.4) as follows

$$\begin{aligned} \min_{x,y} \quad & \|y\|_p + \lambda(\|x\|_1 + \frac{\beta}{2}\|x\|_2^2) \\ \text{s.t.} \quad & Ax - y = b. \end{aligned} \quad (3.1)$$

Given $\sigma > 0$, the augmented Lagrangian function associated with problem (3.1) is given by

$$\mathcal{L}_\sigma(x, y; u) = \|y\|_p + \lambda(\|x\|_1 + \frac{\beta}{2}\|x\|_2^2) + \langle u, Ax - y - b \rangle + \frac{\sigma}{2}\|Ax - b - y\|_2^2,$$

where $u \in \mathbb{R}^m$ is a multiplier associated with the constraint. Based on the above augmented Lagrangian function, with the given $(x^k, y^k; u^k)$, the new iteration $(x^{k+1}, y^{k+1}; u^{k+1})$ is generated for solving (1.4) via the iterative scheme:

$$\begin{cases} y^{k+1} = \arg \min_{y \in \mathbb{R}^m} \mathcal{L}_\sigma(x^k, y; u^k), \\ x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ \mathcal{L}_\sigma(x, y^{k+1}; u^k) + \frac{\sigma}{2}\|x - x^k\|_{\mathcal{T}}^2 \right\}, \\ u^{k+1} = u^k + \tau\sigma(Ax^{k+1} - y^{k+1} - b), \end{cases} \quad (3.2)$$

where $\mathcal{T} := (\zeta I_n - A^\top A)$ with $\zeta > 0$ be a positive scalar such that \mathcal{T} be positive semidefinite.

Observing that each step of the iterative scheme (3.2) involves solving a convex minimization problem, we now illustrate that a simple closed-form solution is permitted for each subproblem, which leads to the framework being easy to implement. Firstly, we can get for every $k = 0, 1, \dots$ that

$$\begin{aligned} y^{k+1} &= \arg \min_{y \in \mathbb{R}^m} \mathcal{L}_\sigma(x^k, y; u^k) \\ &= \arg \min_{y \in \mathbb{R}^m} \left\{ \|y\|_p + \frac{\sigma}{2}\|Ax^k - b - y + \sigma^{-1}u^k\|_2^2 \right\} \\ &= \text{Prox}_{\sigma^{-1}\|\cdot\|_p}(Ax^k - b + \sigma^{-1}u^k). \end{aligned}$$

Secondly, let $\eta := \lambda\beta + \sigma\zeta$, for every $k = 0, 1, \dots$, we have

$$\begin{aligned} x^{k+1} &= \arg \min_{x \in \mathbb{R}^n} \left\{ \mathcal{L}_\sigma(x, y^{k+1}; u^k) + \frac{\sigma}{2}\|x - x^k\|_{\mathcal{T}}^2 \right\} \\ &= \arg \min_{x \in \mathbb{R}^n} \left\{ \lambda\|x\|_1 + \frac{\lambda\beta}{2}\|x\|_2^2 + \frac{\sigma}{2}\|Ax - b - y^{k+1} + \sigma^{-1}u^k\|_2^2 + \frac{\sigma}{2}\|x - x^k\|_{\mathcal{T}}^2 \right\} \\ &= \text{Prox}_{\eta^{-1}\lambda\|\cdot\|_1} \left(x^k - \frac{\sigma A^\top (Ax^k - b - y^{k+1} + \sigma^{-1}u^k) + \lambda\beta x^k}{\eta} \right). \end{aligned}$$

In summary, we are ready to state the full steps of the semi-ADMM while it is used to solve the problem (1.4) as follows:

Algorithm 1

Step 0. Choose starting point $(x^0, y^0; u^0)$. Choose positive constants ζ such that \mathcal{T} is positive semi-definite. Input data b , choose model parameters $\lambda > 0, \beta > 0$ and positive

constants $\sigma > 0, \tau \in (0, (1 + \sqrt{5})/2)$. For $k = 0, 1, \dots$, do the following operations iteratively.

Step 1. Given x^k and u^k , compute

$$y^{k+1} = \text{Prox}_{\sigma^{-1}\|\cdot\|_p}(Ax^k - b + \sigma^{-1}u^k).$$

Step 2. Given y^{k+1} and u^k , compute

$$x^{k+1} = \text{Prox}_{\eta^{-1}\lambda\|\cdot\|_1}\left(x^k - \frac{\sigma A^\top(Ax^k - b - y^{k+1} + \sigma^{-1}u^k) + \lambda\beta x^k}{\eta}\right).$$

Step 3. Given x^{k+1} and y^{k+1} , compute

$$u^{k+1} = u^k + \tau\sigma(Ax^{k+1} - y^{k+1} - b).$$

Step 4. Set $k := k + 1$.

The convergence of semi-ADMM can be easily followed by using the known convergence result of Fazel et al. [13] since $\|x\|_2^2$ is strongly convex and \mathcal{T} is positive semi-definite. We can see that the semi-ADMM algorithm is easy to implement. However, the improper value of the parameter ζ will lead to an inaccurate solution of Step 2 in Algorithm 1. In the next section, we will introduce two auxiliary variables and use direct extension of ADMM to solve a 3-block problem evolved from model (1.4).

4 Direct Extension of ADMM for Solving (1.4)

The popular semi-ADMM can be applied to solve (1.4), but this algorithm needs to rely on semi-proximal terms, which may cause an inaccurate solution. In this section, we introduce the implementation details of the direct extended of ADMM for the problem (1.4). We first introduce a pair of auxiliary variables $y := Ax - b$ and $z := x$, then (1.4) can be reformulated as follows

$$\begin{aligned} \min_{x,y,z} \quad & \|y\|_p + \lambda(\|z\|_1 + \frac{\beta}{2}\|x\|_2^2) \\ \text{s.t.} \quad & Ax - y = b, \\ & x - z = 0. \end{aligned} \tag{4.1}$$

Given $\sigma > 0$, the augmented Lagrangian function associated with problem (4.1) is given by

$$\mathcal{L}_\sigma(x, y, z; u, v) = \|y\|_p + \lambda(\|z\|_1 + \frac{\beta}{2}\|x\|_2^2) + \langle u, Ax - y - b \rangle + \frac{\sigma}{2}\|Ax - b - y\|_2^2 + \langle v, x - z \rangle + \frac{\sigma}{2}\|x - z\|_2^2,$$

where $u \in \mathbb{R}^m, v \in \mathbb{R}^n$ are multipliers associated with the constraints. Based on the above augmented Lagrangian function, with the given $(x^k, y^k, z^k; u^k, v^k)$, the new iteration $(x^{k+1}, y^{k+1}, z^{k+1}; u^{k+1}, v^{k+1})$ is generated for solving (1.4) via the iterative scheme:

$$\begin{cases} x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \mathcal{L}_\sigma(x, y^k, z^k; u^k, v^k), \\ y^{k+1} = \arg \min_{y \in \mathbb{R}^m} \mathcal{L}_\sigma(x^{k+1}, y, z^k; u^k, v^k), \\ z^{k+1} = \arg \min_{z \in \mathbb{R}^n} \mathcal{L}_\sigma(x^{k+1}, y^{k+1}, z; u^k, v^k), \\ u^{k+1} = u^k + \tau\sigma(Ax^{k+1} - y^{k+1} - b), \\ v^{k+1} = v^k + \tau\sigma(x^{k+1} - z^{k+1}). \end{cases} \tag{4.2}$$

Actually, it is not difficult to see from the favorable structures of problem (4.1) that (4.2) is equivalent to the following format:

$$\begin{cases} x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \mathcal{L}_\sigma(x, y^k, z^k; u^k, v^k), \\ \begin{pmatrix} y^{k+1} \\ z^{k+1} \end{pmatrix} = \arg \min_{y \in \mathbb{R}^m, z \in \mathbb{R}^n} \mathcal{L}_\sigma(x^{k+1}, y, z; u^k, v^k), \\ u^{k+1} = u^k + \tau \sigma (Ax^{k+1} - y^{k+1} - b), \\ v^{k+1} = v^k + \tau \sigma (x^{k+1} - z^{k+1}), \end{cases}$$

which also falls into the framework of semi-ADMM with the semi-proximal operators being $\mathbf{0}$, and thus the convergence result is guaranteed in the literature. For the sake of simplicity, we omit the statement of the convergence theorem here. In fact, the convergence result of the direct extension of ADMM in (4.2) for solving (1.4) can also be obtained by [7, 14].

We now illustrate that a simple closed-form solution of (4.2) is permitted for each sub-problem, which is also easy to implement. Firstly, let $\gamma := \lambda\beta + \sigma$, we can get for every $k = 0, 1, \dots$ that

$$\begin{aligned} x^{k+1} &= \arg \min_{x \in \mathbb{R}^n} \mathcal{L}_\sigma(x, y^k, z^k; u^k, v^k) \\ &= \arg \min_{x \in \mathbb{R}^n} \left\{ \frac{\lambda\beta}{2} \|x\|_2^2 + \langle A^\top u^k + v^k, x \rangle + \frac{\sigma}{2} \|Ax - b - y^k\|_2^2 + \frac{\sigma}{2} \|x - z^k\|_2^2 \right\} \\ &= (\gamma I_n + \sigma A^\top A)^{-1} (\sigma z^k - v^k + A^\top (\sigma(b + y^k) - u^k)). \end{aligned}$$

Secondly, for every $k = 0, 1, \dots$, we have

$$\begin{aligned} y^{k+1} &= \arg \min_{y \in \mathbb{R}^m} \mathcal{L}_\sigma(x^{k+1}, y, z^k; u^k, v^k) \\ &= \arg \min_{y \in \mathbb{R}^m} \left\{ \|y\|_p + \langle u^k, -y \rangle + \frac{\sigma}{2} \|Ax^{k+1} - b - y\|_2^2 \right\} \\ &= \text{Prox}_{\sigma^{-1}\|\cdot\|_p}(Ax^k - b + \sigma^{-1}u^k). \end{aligned}$$

Thirdly, for every $k = 0, 1, \dots$, we have

$$\begin{aligned} z^{k+1} &= \arg \min_{z \in \mathbb{R}^n} \mathcal{L}_\sigma(x^{k+1}, y^{k+1}, z; u^k, v^k) \\ &= \arg \min_{z \in \mathbb{R}^n} \left\{ \lambda \|z\|_1 + \langle v^k, -z \rangle + \frac{\sigma}{2} \|x^{k+1} - z\|_2^2 \right\} \\ &= \text{Prox}_{\sigma^{-1}\lambda\|\cdot\|_1}(x^{k+1} + \sigma^{-1}v^k). \end{aligned}$$

In summary, we are ready to state the full steps of the direct extension of ADMM while it is used to solve the problem (1.4) as follows:

Algorithm 2

Step 0. Choose starting point $(x^0, y^0, z^0; u^0, v^0)$. Input data b , choose model parameters $\lambda > 0, \beta > 0$ and positive constants $\sigma > 0, \tau \in (0, (1 + \sqrt{5})/2)$. For $k = 0, 1, \dots$, do the following operations iteratively.

Step 1. Given y^k, z^k , and u^k , compute

$$x^{k+1} = (\gamma I_n + \sigma A^\top A)^{-1} (\sigma z^k - v^k + A^\top (\sigma(b + y^k) - u^k)).$$

Step 2. Given x^{k+1} and u^k , compute

$$(y^{k+1}, z^{k+1}) = (\text{Prox}_{\sigma^{-1}\|\cdot\|_p}(Ax^k - b + \sigma^{-1}u^k), \text{Prox}_{\sigma^{-1}\lambda\|\cdot\|_1}(x^{k+1} + \sigma^{-1}v^k)).$$

Step 3. Given x^{k+1} , y^{k+1} , and z^{k+1} , compute

$$\begin{aligned} u^{k+1} &= u^k + \tau\sigma(Ax^{k+1} - y^{k+1} - b), \\ v^{k+1} &= v^k + \tau\sigma(x^{k+1} - z^{k+1}). \end{aligned}$$

Step 4. Set $k := k + 1$.

We can see that the direct extension of ADMM is also easy to implement. However, this algorithm includes the process of matrix inversion, which may consume a lot of time and memory for large-scale problems. In the implementation of Step 1 of Algorithm 2, we first compute the Cholesky decomposition of $I_n + \sigma\gamma^{-1}A^\top A$ and then solve the linear system of equations in each iteration by using the pre-computed Cholesky factor or by an iterative solver such as the preconditioned conjugate gradient method. The above approach can greatly reduce the calculation time of the algorithm when dealing with large-scale problems.

5 Numerical Experiments

In this section, we conduct some numerical experiments to demonstrate the superiority of model (1.4) and the practical performance of the two algorithms. All the experiments are performed with Microsoft Windows 10 and MATLAB R2018a, and run on a PC with an Intel Core i7 CPU at 1.80 GHz and 8 GB of memory.

We conduct experiments on two types of sensing matrices. Define the test sensing matrix as $A = [a_1, \dots, a_n] \in \mathbb{R}^{m \times n}$. One type matrix is the random Gaussian (GAU) matrix defined as

$$a_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_m/m), \quad i = 1, \dots, n,$$

and another is the random partial DCT (PDCT) matrix with the following expression

$$a_i = \frac{1}{\sqrt{m}} \cos(2i\pi\xi), \quad i = 1, \dots, n,$$

where $\xi \in \mathbb{R}^m \sim \mathcal{U}([0, 1]^m)$, i.e., the components of ξ are uniformly and independently sampled from $[0, 1]$.

For comparison in a relatively fair way, we measure the quality of the reconstruction solutions using the relative error defined as

$$\text{RLNE} := \frac{\|\bar{x} - \underline{x}\|_2}{\|\underline{x}\|_2},$$

where \bar{x} and \underline{x} are the reconstructed and ground truth signals, respectively. Besides, we also define

$$\text{RelErr} := \frac{\|x^{k+1} - x^k\|_2}{\max\{\|x^k\|_2, 1\}}, \quad \text{and} \quad \text{ResErr} := \frac{\|Ax - b\|_2}{\|b\|_2}.$$

In each tested algorithm, we stop the iterative process if $\text{RelErr} \leq 1e - 4$ or the iteration number achieves 2000. For two algorithms, we set $\tau = 1.618$ and $\zeta = 1.2$ for Algorithm 1. Other parameters values will be determined adaptively at each experiment. We tried different starting points for each algorithm and found that all of them are insensitive towards

starting points. Therefore, we initialize the starting points as zero in all experiments of the following.

In order to highlight the robustness and practicality of the model (1.4), we mainly test three different types of noises in this part. The observation b is obtained by

$$b = A * \underline{x} + \alpha * \text{ns},$$

where α is the noise level, and “ns” is one of the following types of noise: log-normal noise, Gaussian noise and uniform noise. Here, we recover a sparse signal of being polluted by different kinds of noises via model (1.4) with different p values. In order to deal with log-normal noise as well as heavy-tailed noise, we set $p = 1$; for handling Gaussian noise, p is chosen as 2; and we set $p = \infty$ for uniformly distributed noise.

5.1 Comparison of the LASSO model under different noise

In this part, we consider the noise level $\alpha = 0.01$. We compare our two algorithms on model (1.4) and model with the Lasso model solved by ADMM. The compared model and algorithms in [18] are as follows,

$$\begin{aligned} \min_{x,y} \quad & \frac{1}{2} \|Ay - b\|_2^2 + \lambda \|x\|_1 \\ \text{s.t.} \quad & x - y = 0. \end{aligned} \tag{5.1}$$

Given $\delta > 0$, the augmented Lagrangian function associated with problem (5.1) is given by

$$\mathcal{L}_\delta(x, y; u) = \frac{1}{2} \|Ay - b\|_2^2 + \lambda \|x\|_1 + \delta \langle u, x - y \rangle + \frac{\delta}{2} \|x - y\|_2^2,$$

and the corresponding ADMM iteration is:

$$\begin{cases} x^{k+1} = \arg \min_x \mathcal{L}_\delta(x, y^k; u^k) = \arg \min_x \{ \lambda \|x\|_1 + \frac{\delta}{2} \|x - y^k + u^k\|_2^2 \}, \\ y^{k+1} = \arg \min_y \mathcal{L}_\delta(x^{k+1}, y; u^k) = \arg \min_y \{ \frac{1}{2} \|Ay - b\|_2^2 + \frac{\delta}{2} \|x^{k+1} - y + u^k\|_2^2 \}, \\ u^{k+1} = u^k + x^{k+1} - y^{k+1}. \end{cases}$$

The specific setting of the algorithm and details of parameters are shown in the literature [18].

We set PDCT sensing matrix with 64×128 size, sparsity k , i.e., the number of nonzero elements of the original solution, is fixed 20. For Algorithm 1 and Algorithm 2, the model parameter β is fixed as 0.01. Other parameter settings are the same for the three algorithms.

At the case of log-normal noise, we set $p = 1$, i.e., the data fidelity term is $\|Ax - b\|_1$. Besides, the weighting parameter λ in (1.4) is chosen as $\lambda = 0.01$, and $\sigma = \sqrt{2} \|AA^\top\|_2$. The original signal and the reconstructed signals recovered by Algorithm 1, Algorithm 2, and Lasso model solved by ADMM are listed respectively in Figure 1. In this figure, the original signal is denoted by black stars “*” and the recovered signals are denoted by “○” marked in different colors. Comparing each plot in Figure 1 from left to right, we clearly see that all the stars in (c) are circled exactly by the blue circles with a symbol “⊗” which indicates that the use of $\|Ax - b\|_1$ is better. Moreover, we also see that the final relative error of the solution derived by Algorithm 1 and Algorithm 2 are significantly smaller than the one produced by Lasso model, which once again indicates that the advantage of $\|Ax - b\|_1$.

For the Gaussian noise, we set the parameter $\lambda = 0.06$, $\sigma = 1$ and $p = 2$ for the data fidelity term, i.e., $\|Ax - b\|_2$. The results of each algorithm are listed in Figure 2. Intuitively

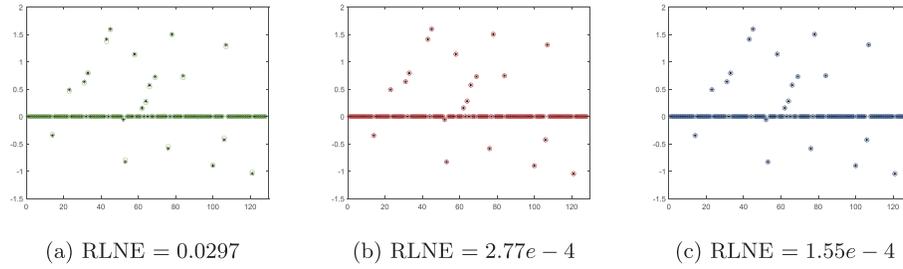


Figure 1: *Log-normal noise: the original signal (black stars) versus the recovery signals by Lasso model solved by ADMM (green circles), Algorithm 1 (red circles), and by Algorithm 2 (blue circles).*

at first glance, Algorithm 1 and 2 have better recovery effects, while Lasso model solved by ADMM cannot recover signals well. Furthermore, we observe the RLNE values of three algorithms and find that the values of Algorithm 1 and Algorithm 2 are less than the value of Lasso solved by ADMM. The above shows that $\|Ax - b\|_2$ is better than the least squares data fidelity term. Comparing the results on handling Gaussian noise of Algorithm 1 and 2, we find that the direct extension ADMM is slightly better than semi-ADMM.

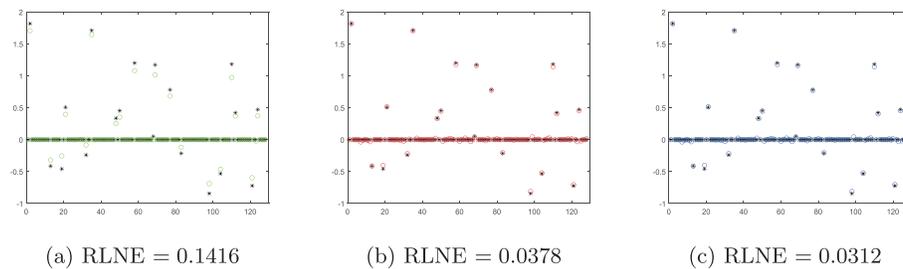


Figure 2: *Gaussian noise: the original signal (black stars) versus the recovery signals by Lasso model solved by ADMM (green circles), Algorithm 1 (red circles), and by Algorithm 2 (blue circles).*

Finally, we turn to the problem of uniform noise, we set $\lambda = 0.008$, $\sigma = 2$ and choose the the data fidelity term as $\|Ax - b\|_\infty$, i.e., $p = \infty$. The numerical results of three algorithms are shown in Figure 3. We found that the three algorithms seem to be able to deal with uniform noise better, but we found that the RLNE values obtained by Algorithm 1 and 2 are much smaller than those obtained by Lasso model solved by ADMM. This shows that $\|Ax - b\|_\infty$ is better than the lasso model when dealing with uniform noise. In addition, we still find that the Algorithm 2 is still better than the Algorithm 1 indicating that the direct extension ADMM is better than semi-ADMM.

From these figures, we can visibly see that the quality of the solutions derived by Algorithm 1 and Algorithm 2 are better. From these limited numerical experiments, it can be concluded that, as far as the three types of noise are concerned, our proposed model (1.4) has the ability to get produce higher quality reconstruction results if the data fidelity term is chosen adaptively. Moreover, we found that Algorithm 2 is the best one of the three algorithms, indicating that the direct extension ADMM is more effective than semi-ADMM of gaining lower RLNE values. Next, we will continue to only study the performance of

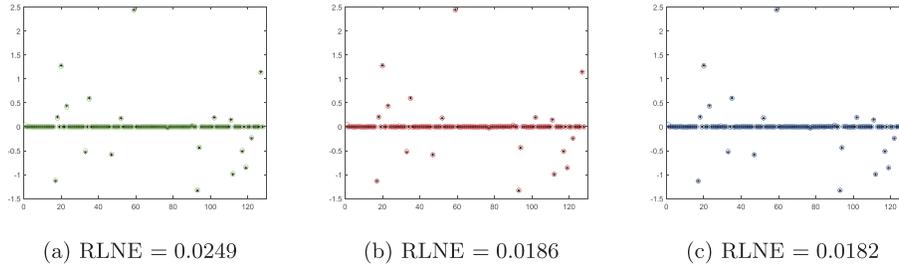


Figure 3: *Uniform noise: the original signal (black stars) versus the recovery signals by Lasso model solved by ADMM (green circles), Algorithm 1 (red circles), and by Algorithm 1 (blue circles).*

Algorithms 1 and Algorithms 2 on different parameters and sensing matrices owing to the poor performance of Lasso.

5.2 Test the performance of the algorithms with various β

In this part, we mainly verify the performance of Algorithm 1 and Algorithm 2 with various β under different noise types. In this test, we also set the noise level as $\alpha = 0.01$ and also choose PDCT sensing matrices with 64×128 size. The sparsity k is fixed 20. For the three types of noise, all the parameter settings are the same as Section 5.1 except for β and λ . The table contains the values of β and λ , the CPU time required in seconds (Time), the objective function value of (1.4) (Obj), the smallest values RLNE and ResErr, and the number of outer iterations (Iter). The results are listed in Table 1. From Table 1, we see that the RLNE and ResErr values at the last two columns are always smaller, which once again shows that our proposed model (1.4) and algorithms indeed benefit the quality of the reconstruction solutions. Observing the results row-by-row, we find that, Algorithm 2 takes more time than Algorithm 1, but the RLNE value is always less than Algorithm 1 in most cases, which explains that both algorithms have their own advantages. In addition, we also test different values of λ and observe that the two algorithms are not sensitive to changes in the values of λ .

The following experiment is a more intuitive and detailed test of the influence of different parameter values on the algorithms. From the above experiments, we find that the performance of Algorithm 1 and Algorithm 2 is very similar. Therefore, we mainly verify the performance of Algorithm 1 with various β and λ under Gaussian noise in the following test. In this test, we set the noise level as $\alpha = 0.001$ and choose GUA sensing matrices with 400×800 size. The sparsity k is fixed starts from 30 and ends at 50 with an increment of 5. The experiment results are shown in Figure 4. Firstly, we pay attention to the picture on the left, which shows the performance of Algorithm 1 under different β values when $\lambda = 0.01$ is set. We find that with the increase of sparsity, in the case of $\beta \leq 1.5$, Algorithm 1 is effective, and the performance of Algorithm 1 is not much different when $\beta \leq 1$. Therefore, when β takes an appropriate value, the model and algorithm are feasible and effective. Secondly, we turn to the figure on the right-hand side of Figure 4, which shows the performance of Algorithm 1 with different parameters λ when the β is set to 0.05. We can see from the RLNE value of the vertical axis that the performance of Algorithm 1 is always excellent when $\lambda \leq 1.1$. The algorithm is not very sensitive to λ . In addition, we found from this test that the algorithm performs best when $\lambda = 0.05$.

Table 1: The performance of the Algorithm 1 and Algorithm 2 with various β under different noise.

		Log-normal noise							
β	λ	Algorithm 1				Algorithm 2			
		Time	Obj	RLNE	ResErr	Time	Obj	RLNE	ResErr
0.01	0.02	0.0060	0.2655	2.59e-04	1.68e-04	0.0232	0.2646	3.12e-04	1.09e-04
	0.03	0.0080	0.4662	6.94e-02	1.06e-04	0.0344	0.4664	6.83e-02	1.10e-04
	0.05	0.0049	0.7326	3.81e-04	2.43e-04	0.0262	0.7296	1.46e-04	4.99e-05
0.05	0.01	0.0158	0.1614	8.09e-05	5.60e-05	0.0655	0.1614	1.55e-04	5.64e-05
	0.03	0.0039	0.4715	2.40e-04	1.50e-04	0.0227	0.4699	2.00e-04	6.99e-05
	0.05	0.0043	0.7003	1.47e-04	9.96e-05	0.0195	0.7005	2.54e-04	9.83e-05
0.1	0.01	0.0100	0.1098	1.46e-04	5.23e-05	0.0452	0.1104	1.80e-04	9.50e-05
	0.03	0.0279	0.5530	2.11e-03	1.60e-06	0.1257	0.5530	7.03e-05	1.15e-06
	0.06	0.0031	0.9671	1.18e-04	7.80e-05	0.0165	0.9669	1.79e-04	6.25e-05
0.5	0.01	0.0189	0.2054	8.74e-02	1.56e-05	0.1061	0.2055	8.56e-02	2.50e-05
	0.03	0.0074	0.3630	2.99e-01	9.02e-05	0.0309	0.3629	2.99e-01	8.18e-05
	0.06	0.0037	1.1955	1.06e-04	7.84e-05	0.0198	1.1963	3.07e-04	1.08e-04
1	0.01	0.0155	0.2543	4.00e-01	5.83e-06	0.0689	0.2544	3.98e-01	8.07e-06
	0.03	0.0080	0.5370	8.59e-02	2.07e-05	0.0396	0.5374	8.45e-02	4.51e-05
	0.06	0.0058	0.7865	2.16e-02	2.13e-04	0.0372	0.7851	1.68e-02	9.88e-05
1.5	0.01	0.0141	0.2360	3.43e-01	1.04e-06	0.0555	0.2361	3.42e-01	6.82e-06
	0.03	0.0053	0.4529	2.28e-01	4.12e-05	0.0314	0.4535	2.28e-01	8.27e-05
	0.06	0.0036	1.3684	4.97e-01	1.57e-05	0.0157	1.3684	4.97e-01	1.97e-05
		Gaussian noise							
0.01	0.01	0.0120	0.1265	4.35e-02	3.48e-05	0.0734	0.1267	4.19e-02	4.64e-05
	0.03	0.0118	0.4412	8.55e-02	7.26e-05	0.0451	0.4413	8.54e-02	7.29e-05
	0.08	0.0080	1.1809	2.43e-02	1.11Ee-04	0.0382	1.1803	2.22e-02	7.54e-05
0.05	0.03	0.0114	0.4516	5.66e-02	8.06e-05	0.0716	0.4514	5.56e-02	6.81e-05
	0.1	0.0073	1.4585	5.63e-02	1.53e-04	0.0459	1.4585	3.99e-02	1.54e-04
	0.02	0.0102	0.2497	8.96e-02	9.35e-05	0.0472	0.2495	8.81e-02	7.82e-05
0.1	0.05	0.0081	0.6856	4.36e-02	1.84e-04	0.0400	0.6838	4.25e-02	7.61e-05
	0.1	0.0088	1.5762	1.25e-01	1.55e-04	0.0376	1.5756	1.19e-01	1.11e-04
	0.03	0.0121	0.6187	2.40e-01	5.13e-06	0.0540	0.6189	2.39e-01	1.69e-05
0.5	0.06	0.0064	1.2998	6.74e-02	1.36e-04	0.0301	1.2993	7.03e-02	1.07e-04
	0.1	0.0070	1.9544	4.06e-01	3.86e-05	0.0305	1.9548	4.05e-01	6.23e-05
	0.02	0.0088	0.4128	3.53e-01	2.03e-05	0.0484	0.4131	3.53e-01	3.20e-05
1	0.07	0.0053	1.4178	3.77e-01	2.94e-05	0.0220	1.4183	3.76e-01	5.91e-05
	0.1	0.0044	1.5681	2.53e-01	9.82e-05	0.0224	1.5673	2.51e-01	3.76e-05
	0.02	0.0094	0.4494	4.20e-01	7.24e-06	0.0545	0.4494	4.19e-01	5.87e-06
1.5	0.06	0.0048	1.4385	3.80e-01	6.69e-05	0.0244	1.4382	3.80e-01	4.73e-05
	0.1	0.0030	1.9498	3.00e-01	6.56e-05	0.0171	1.9497	3.00e-01	5.51e-05
			Uniform noise						
0.01	0.008	0.0775	0.1283	1.86e-02	1.14e-04	0.1093	0.1282	1.82e-02	6.61e-05
	0.01	0.1290	0.1407	2.17e-02	1.10e-04	0.1553	0.1407	2.13e-02	6.15e-05
	0.03	0.5199	0.4978	2.74e-02	1.76e-02	0.6322	0.4979	2.71e-02	1.71e-02
0.05	0.005	0.0724	0.0754	2.42e-02	9.70e-05	0.0840	0.0754	2.44e-02	7.78e-05
	0.01	0.1146	0.1167	2.19e-02	1.19e-04	0.1864	0.1166	2.13e-02	7.14e-05
	0.03	1.8493	0.4276	1.24e-01	7.45e-02	2.0027	0.4276	1.26e-01	6.97e-02
0.1	0.005	0.0576	0.1003	2.02e-02	1.13e-04	0.0644	0.1003	2.07e-02	1.08e-04
	0.01	0.1385	0.1363	3.70e-02	1.05e-04	0.2094	0.1363	3.45e-02	3.96e-05
	0.03	1.4328	0.6187	1.61e-01	6.25e-02	1.5150	0.6188	1.62e-01	6.06e-02
0.5	0.005	0.2033	0.1294	1.27e-02	2.64e-05	0.3482	0.1294	1.34e-02	1.19e-05
	0.01	0.8395	0.1827	5.76e-02	9.42e-06	0.6924	0.1827	5.95e-02	1.40e-05
	0.03	1.8507	0.5630	1.13e-01	7.01e-02	1.9221	0.5630	1.16e-01	6.79e-02
1	0.005	0.5779	0.1188	3.54e-02	1.18e-05	0.5520	0.1187	3.61e-02	1.09e-05
	0.01	0.6213	0.2583	2.56e-02	8.77e-06	0.5086	0.2583	2.80e-02	8.41e-06
	0.001	1.3861	0.0255	1.82e-02	5.08e-08	1.4076	0.0255	1.81e-02	5.25e-08
1.5	0.003	0.9506	0.0744	1.66e-02	7.80e-06	0.8195	0.0744	1.67e-02	9.00e-06

In summary, we have found through a series of experiments that our model and algorithms are effective and insensitive to the choice of parameters. Therefore, it shows that the algorithms are universal and simple, and easy to operate.

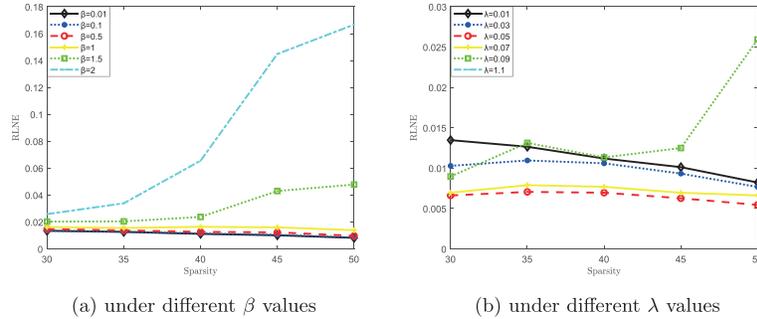


Figure 4: The performance of Algorithm 1 under different parameter values

5.3 Test on two dimensional MRI reconstruction

In this section, we apply the proposed model (1.4) and algorithms to the two-dimensional phantom image reconstruction problem. The phantom image with size 64×64 is compressible under a Haar wavelet basis. Therefore, the observation data can be chosen as the wavelet coefficients sampled by the product of a partial FFT matrix and inverse Haar wavelet transform. In view of the fact that the experimental results of Algorithm 1 and Algorithm 2 are very similar, we mainly test the performance of Algorithm 1 in this experiment. we choose model parameters $\lambda = 0.05, \beta = 0.01$ and $\sigma = \sqrt{2}, \tau = 1.618$. In addition, we consider the noise level $\alpha = 1e - 3$. At the cases of log-normal noise and Gaussian noise, we set $p = 1$ and $p = 2$, respectively.

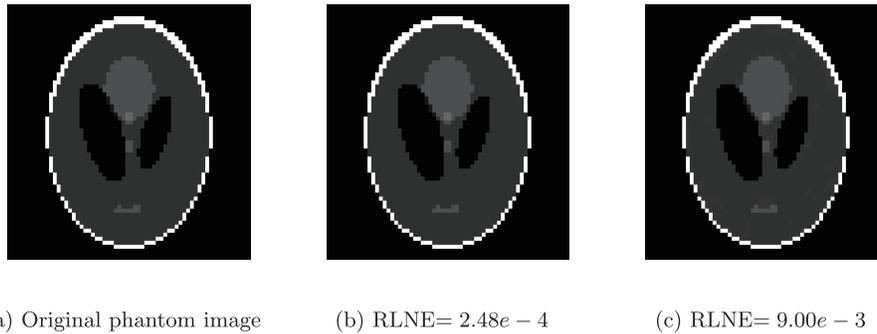


Figure 5: The performance of Algorithm 1 on rebuilding phantom image corrupted by log-normal noise (b) and Gaussian noise (c)

The sampling matrix for two dimensional MRI is the compound of a partial FFT and an inverse wavelet transform with size 2133×4096 . The image under wavelet transformation has 792 nonzero entries. The numerical results derived by Algorithm 1 are listed in Figure 5. From left to right in Figure 5 is the original phantom image, the reconstructed image under log-normal noise, and the reconstructed image under Gaussian noise respectively. At first glance, we see that Algorithm 1 works on different types of noise successfully to produce

acceptable reconstructions. More accurately, the RLNE values obtained on log-normal noise and Gaussian noise are $2.48e - 4$ and $9.00e - 3$, respectively. From this simple test, we conclude that our proposed model and Algorithm 1 are really effective and practical in rebuilding phantom images corrupted by log-normal noise or Gaussian noise.

6 Conclusions

In this paper, we focus on solving a ℓ_1 - ℓ_2 special convex bi-level programming problem. Motivated by [17], the challenging bi-level optimization problem can be transformed into a single-level constrained optimization task based on the prior knowledge. The Lasso model is a special penalized form of the single-level ℓ_1 - ℓ_2 constrained problem, which can handle sparse optimization problems well under certain conditions. However, it also has certain limitations in variable selection. Compared to the Lasso, the elastic net simultaneously makes automatic variable selection and continuous shrinkage, and it can select groups of correlated variables. But the elastic net model is only applicable to Gaussian noise in most cases, and the least square loss term relies on the knowing of the standard deviation of the noise. In this paper, we propose a more general model, namely, ℓ_p - ℓ_1 - ℓ_2 model. The $\|Ax - b\|_p$ data fidelity term makes it can deal with different types of noise if p is chosen adaptively, e.g., $p = 1$ for log-normal noise and heavy-tailed noise, $p = 2$ for Gaussian noise, and $p = \infty$ for uniformly distributed noise. In addition, ℓ_1 - ℓ_2 -term same as the penalty of elastic net measures the sparsity, which also can select groups of correlated variables. We use the ADMMs to solve this model, which are simple and easy to operate and have good convergence properties. The first algorithm is the semi-ADMM by introducing a variable. Another algorithm is to use the direct extension of ADMM for 3-block convex minimization problem on the introduction of a pair of variables, which is also equivalent to semi-ADMM with the semi-proximal operators being zero. We show that each subproblem involved in both algorithms is easily performed. In addition, some numerical experiments in one dimensional signal reconstruction and two dimensional MRI reconstruction demonstrate that both of our proposed model and algorithms perform well. From this point of view, the proposed model and algorithms have the potential for the signal reconstruction problem.

Acknowledgements

We would like to thank the anonymous referees for their useful comments and suggestions which improved this paper greatly.

References

- [1] R.G. Baraniuk, Compressive sensing, *IEEE Signal Process Mag.* 24 (2007), 118–121.
- [2] P.C. Bellec, G. Lecué and A.B. Tsybakov, Slope meets lasso: improved oracle bounds and optimality, *Ann. Statist.* 46 (2018) 3603–3642.
- [3] A. Belloni and V. Chernozhukov, L1-penalized quantile regression in high-dimensional sparse models, *Ann. Statist.* 39 (2011) 82–130.
- [4] A. Belloni, V. Chernozhukov and L. Wang, Square-root lasso: pivotal recovery of sparse signals via conic programming, *Biometrika* 98 (2011) 791–806.

- [5] F. Bunea, Y. She, H. Ombao, A. Gongvatana, K. Devlin and R. Cohen, Penalized least squares regression methods and applications to neuroimaging, *NeuroImage* 55 (2011) 1519–1527.
- [6] E.J. Candés and M. B. Wakin, An introduction to compressive sampling, *IEEE Signal Process Mag.* 25 (2008), 21–30.
- [7] C.H. Chen, B.S. He, Y.Y. Ye and X.M. Yuan, The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent, *Math. Program.* 155 (2014), 57–59.
- [8] S. Cho, H. Kim, S. Oh, K. Kim and T. Park, Elastic-net regularization approaches for genome-wide association studies of rheumatoid arthritis, *BMC Proc.* 3 (2009), 1–6.
- [9] Y. Cui, J.S. Pang and B. Sen, Composite difference-max programs for modern statistical estimation problems, *SIAM J. Optim.* 28 (2018), 3344–3374.
- [10] M. Demirer, F.X. Diebold, L. Liu and K. Yilmaz, Estimating global bank network connectedness, *J. Appl. Econ.* 33 (2018), 1–15.
- [11] Y.Y. Ding, H.B. Zhang, P.L. Li and Y.H. Xiao, An efficient semismooth Newton method for adaptive sparse signal reconstruction problems, preprint, (2022), arXiv: 2111.12252.
- [12] Y.Y. Ding and Y.H. Xiao, Symmetric gauss-seidel technique-based alternating direction methods of multipliers for transform invariant low-rank textures problem, *J. Math. Imaging Vis.* 69 (2018) 1220–1230.
- [13] M. Fazel, T.K. Pong, D.F. Sun and P. Tseng, Hankel matrix rank minimization with applications to system identification and realization, *SIAM J. Matrix Anal. Appl.* 34 (2013), 946–977.
- [14] B.S. He, Modified alternating directions method of multipliers for convex optimization with three separable functions, *Oper. Res. Trans.* 19 (2015), 57–70.
- [15] B.S. He, X.M. Yuan, Alternating direction method of multipliers for linear programming, *J. Oper. Res. Soc. China* 4 (2016) 425–436.
- [16] M.X. Lin, D.F. Sun and K.C. Toh, Efficient algorithms for multivariate shape-constrained convex regression problems, preprint, (2020), arXiv: 2002.11410.
- [17] R.S. Liu, L. Ma, X.M. Yuan, S.Z. Zeng and J. Zhang, Task-oriented convex bilevel optimization with latent feasibility, *IEEE Trans. Image Process* 31 (2022) 1190–1203.
- [18] Y.F. Lou and M. Yan, Fast L1-L2 minimization via a proximal operator, *J. Sci. Comput.* 74 (2018), 767–785.
- [19] Z.S. Lu, Iterative reweighted minimization methods for l_p regularized unconstrained nonlinear programming, *Math. Program.* 147(2014) 277–307.
- [20] F.P. Michael and T. Pual, Exact regularization of convex programs, *SIAM J. Optim.* 18 (2007) 1326–1350.
- [21] J.J. Moreau, Proximité et dualité dans un espace hilbertien, *Bulletin de la Société Mathématique de France.* 93 (1965) 273–299.

- [22] R.T. Rockafellar, *Convex Analysis*, Princeton University Press, 2015.
- [23] S.J. Teipel, M.J. Grothe, C.D. Metzger, G. Timo, S. Christian, E. Michael, et al, Robust detection of impaired resting state functional connectivity networks in Alzheimers disease using elastic net regularized regression, *Front. Aging Neurosci.* 8 (2017): 318.
- [24] S. Sabach, and S. Shtern, A first order method for solving convex bilevel optimization problems, *SIAM J. Optim.* 27 (2017), 640–660.
- [25] M. Solodov, An explicit descent method for bilevel convex optimization, *J. Convex Anal.* 14 (2007) 227–237.
- [26] R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. Series B Stat. Methodol.* 58 (1996) 267–288.
- [27] L. Wang, L_1 penalized LAD estimator for high dimensional linear regression, *J. Multivar. Anal.* 120 (2013) 135–151.
- [28] S.Y. Wang, Y.H. Xiao, Z.F. Jin, An efficient algorithm for batch images alignment with adaptive rank correction term, *J. Comput.* 346 (2019) 171–183.
- [29] Y.H. Xiao, P.L. Li and S. Lu, Sparse estimation of high-dimensional inverse covariance matrices with explicit eigenvalue constraints, *J. Oper. Res. Soc. China* 9 (2021) 543–568.
- [30] X.C. Xiu, L.C. Kong, Y. Li and H.D. Qi, Iterative reweighted methods for $\ell_1 - \ell_p$ minimization, *Comput. Optim. Appl.* 70 (2018) 201–219.
- [31] Y.H. Xue, Y.F. Feng, and C.L. Wu, An efficient and globally convergent algorithm for $\ell_{p,q} - \ell_r$ model in group sparse optimization, *Commun. Math. Sci.* 18 (2020) 227–258.
- [32] J.F. Yang and Y. Zhang, Local linear convergence of an ADMM-Type splitting framework for equality constrained optimization, *J. Oper. Res. Soc. China* 9 (2021) 307–319.
- [33] W.T. Yin, Analysis and generalizations of the linearized Bregman method, *SIAM J. Imag. Sci.* 3 (2010) 856–877.
- [34] K. Yosida, *Functional Analysis*, Springer Berlin, 1964.
- [35] Z. Zhang and W. Y. Wei, Primal-Dual approach for uniform noise removal, in: *ISST 2015*. Atlantis Press, pp. 103–106.
- [36] H. Zuo and T. Hastie, Regularization and variable selection via the elastic net, *J. R. Stat. Soc. Series B Stat. Methodol.* 67 (2005) 301–320.

Manuscript received 30 September 2021
revised 19 January 2022, 21 April 2022
accepted for publication 21 April 2022

YANYUN DING

Department of Operations Research and Information Engineering
Beijing University of Technology, Beijing 100124, P.R. China
E-mail address: dingyanyunhenu@163.com

ZHIXIAO YUE

Department of mathematics
Southern University of Science and Technology
Shenzhen 518055, P.R. China
E-mail address: 12032011@mail.sustech.edu.cn

HAIBIN ZHANG

Department of Operations Research and Information Engineering
Beijing University of Technology, Beijing 100124, P.R. China
E-mail address: zhanghaibin@bjut.edu.cn