Two-level ℓ_1 Minimization for Compressed Sensing

Xiaolin Huang, *Member, IEEE*, Yipeng Liu, *Member, IEEE*, Lei Shi, Sabine Van Huffel, *Fellow, IEEE*, and Johan A.K. Suykens, *Senior Member, IEEE*

***Abstract*—Compressed sensing using ℓ_1 minimization has been widely and successfully applied. To further enhance the sparsity, some non-convex penalties have been established, such as the ℓ_q-norm with $0 < q < 1$, the smoothly clipped absolute deviation penalty, and the truncated ℓ_1-norm. In this paper, a new non-convex penalty, which gives two different weights according to the order of the absolute value and is hence called two-level ℓ_1-norm, is proposed. On the one hand, compared with the classical ℓ_1 minimization, two-level ℓ_1 minimization leads to a more sparse solution. On the other hand, compared with the other non-convex penalty, two-level ℓ_1 minimization can be solved efficiently by local linearization. The corresponding iteratively reweighted method and soft thresholding algorithm are established. The soft thresholding operator for the two-level ℓ_1-norm is expansive and its convergence is proved by investigating a surrogate function. In numerical experiments, the proposed algorithms provide good sparse signal estimation performance, which implies two-level ℓ_1 minimization a promising technique for compressed sensing.**

***Index Terms*—compressed sensing, non-convex penalty, ℓ_1 minimization, thresholding algorithm**

I. Introduction

COMPRESSED sensing can acquire and reconstruct a signal from relatively fewer measurements than the classical Nyquist sampling. Random linear projections are used to subsample the analogue signal and get the digital measurements. The basics of compressed sensing can be found in [1] [2] [3]. The sampling can be formulated in discrete form as $b = \Phi x$, where $\Phi \in \mathbb{R}^{m\times n}$ is the sampling matrix (also called measurement matrix), $x \in \mathbb{R}^n$ is the signal, and $b \in \mathbb{R}^m$ represents the compressive measurements. Generally, we have $m \ll n$. The signal is then reconstructed from these projections using signal recovery methods which exploit the fact that many natural signals are sparse in certain representations, i.e., $x = \Psi\theta$ where Ψ is the representation matrix and θ is a sparse vector.

To recover a sparse signal θ from b, one can consider the following optimization problem

$$\begin{aligned} \min_{\theta} \quad & \|\theta\|_{\ell_0} \\ \text{s.t.} \quad & \Phi\Psi\theta = b, \end{aligned} \tag{1}$$

where $\|\theta\|_{\ell_0}$ counts the number of nonzero components of θ. We first consider the case $\Psi = \mathbf{I}_{n\times n}$, which stands for the identity matrix in $\mathbb{R}^{n\times n}$. This setting assumes that the signal x is sparse in the time domain and then (1) becomes

$$\begin{aligned} \min_{x} \quad & \|x\|_{\ell_0} \\ \text{s.t.} \quad & \Phi x = b. \end{aligned} \tag{2}$$

In this paper, the theoretical part focuses on (2) and one application for $\Psi \neq \mathbf{I}_{n\times n}$ is considered in numerical experiments.

Problem (2) is NP-hard and hard to solve. To effectively recover sparse signals, some fast algorithms have been established, including orthogonal matching pursuit [4], CoSaMP [5], subspace pursuit [6], and iterative hard thresholding [7]. Another possibility is to use a penalty function $L(x) : \mathbb{R}^n \mapsto \mathbb{R}$ to approach $\|x\|_{\ell_0}$ and then to minimize the penalty function with the constraint that $\Phi x = b$, i.e.,

$$\begin{aligned} \min_{x} \quad & L(x) \\ \text{s.t.} \quad & \Phi x = b. \end{aligned}$$

The most popular choice of the penalty function is to use ℓ_1-norm, i.e., $L(x) = \|x\|_{\ell_1} = \sum_{i=1}^n |x(i)|$, where $x(i)$ is the i-th component of x. Then we obtain the following ℓ_1-minimization problem

$$\begin{aligned} \min_{x} \quad & \|x\|_{\ell_1} \\ \text{s.t.} \quad & \Phi x = b, \end{aligned} \tag{3}$$

which is also called basis pursuit [8]. Based on the ℓ_1-norm, some interesting techniques including Dantzig selector [9] and decoding by linear programming [10], and practical solving algorithms, such as an iterative soft thresholding algorithm [11], have been established and widely applied in many fields.

Minimizing the ℓ_1-norm is an efficient relaxation for (2). If Φ satisfies some conditions, such as the restricted isometry property, [10], the null space property [12], and the incoherence condition [13], then the sparse signal can be recovered by solving (3). But due to the relaxation, the accuracy is degenerated [14]. Moreover, for some applications, the result of ℓ_1 minimization is not sparse enough and cannot recover the original signals. The following is an interesting and simple example, which is given by [15] to intuitively explain the

Manuscript received 2013;

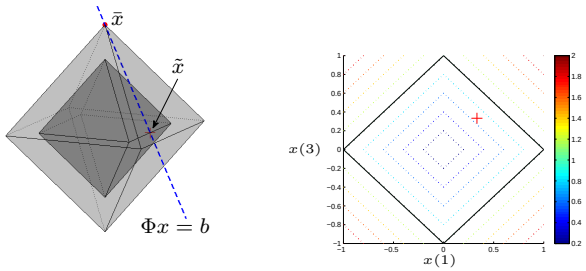
This work was supported by Research Council KUL: GOA/11/05 Ambiorics, GOA/10/09 MaNet, CoE EF/05/006, PFV/10/002 Optimization in Engineering (OPTEC), IOF-SCORES4CHEM, IDO 08/013 Autism, several PhD/postdoc & fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems and optimization), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (WOG: ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC), G.0377.12 (Structured models), G.0427.10N (Integrated EEG-fMRI), G.0108.11 (Compressed Sensing) G.0869.12N (Tumor imaging) G.0A5513N (Deep brain stimulation), IWT: TBM070713-Accelero, TBM080658-MRI (EEGfMRI), TBM110697-NeoGuard, PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare; Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2012-2017); iMinds; EU: ERNSI; ERC AdG A-DATADRIVE-B, FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940), FP7-HEALTH/ 2007-2013 (n260777); Contract Research: AMINAL; Other: Helmholtz: viCERP, ACCM, Bauknecht, Hoerbiger. L. Shi is also supported by the National Natural Science Foundation of China (11201079). Johan Suykens and Sabine Van Huffel are professor at KU Leuven, Belgium.

The authors are with Department of Electrical Engineering, ESAT-SCD-SISTA, KU Leuven, and with iMinds Future Health Department, KU Leuven, B-3001 Leuven, Belgium. (e-mails: huangxl06@mails.tsinghua.edu.cn, yipeng.liu@esat.kuleuven.be, leishi@fudan.edu.cn, sabine.vanhuffel@esat.kuleuven.be, johan.suykens@esat.kuleuven.be).

failure of ℓ_1 minimization for some sparse signals. In this example, the real signal is $\bar{x} = [0,\ 1,\ 0]^T$ and

$$\Phi = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix}.$$

Our task is to recover the signal from $b = \Phi\bar{x} = [1,\ 1]^T$. Unfortunately, ℓ_1 minimization does not lead to the real signal: instead of $\bar{x}$, $\tilde{x} = [1/3,\ 0,\ 1/3]^T$ is optimal to (3). Fig.1(a) (i.e., Fig.1(b) in [15]) shows the reason: the feasible set intersects the interior of the ℓ_1 ball $\{x : \|x\|_{\ell_1} \le \|\bar{x}\|_{\ell_1}\}$. Fig.1(b) plots the contour map of the ℓ_1-norm in the plane $x(2) = 0$, where $x(i)$ denotes the i-th component of x. The solid line represents the level set $\{x : \|x\|_{\ell_1} = \|\bar{x}\|_{\ell_1}\}$ and $\tilde{x}$ is marked by red cross. We observe that $\tilde{x}$ is in the interior of $\{x : \|x\|_{\ell_1} \le \|\bar{x}\|_{\ell_1}\}$, which makes that (3) cannot recover $\bar{x}$.



(a) (b)

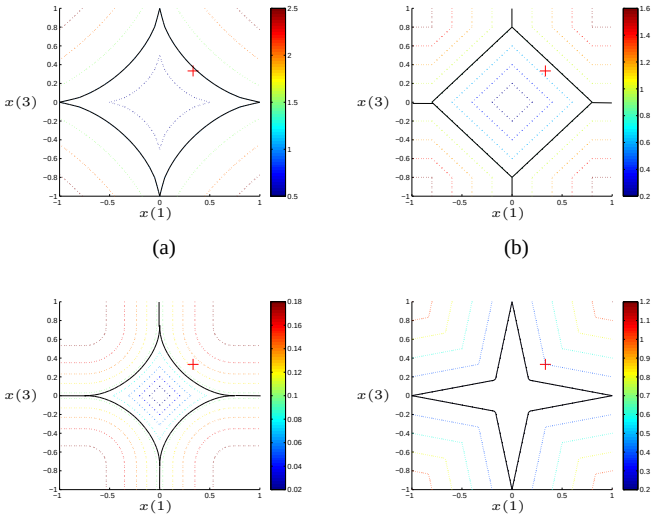
Fig. 1. An example where ℓ_1 minimization cannot recover $\bar{x}$: (a) (Fig.1 (b) in [15]) feasible set $\Phi x = b$ intersects the interior of the ℓ_1-norm ball $\{x : \|x\|_{\ell_1} \le \|\bar{x}\|_{\ell_1}\}$; (b) the contour map of the ℓ_1-norm in the plane $x(2) = 0$; the solid line is the level set $\{x : \|x\|_{\ell_1} = \|\bar{x}\|_{\ell_1}\}$.

Motivated by this observation, we know that one possible way of recovering $\bar{x}$ is to replace the ℓ_1-norm by another penalty $L(x)$ to shrink the penalty ball $\{x : L(x) \le L(\bar{x})\}$. Using the ℓ_q-norm with $0 < q < 1$ instead of the ℓ_1-norm is one potential choice, which has been investigated in [16] – [20]. The sparsity of ℓ_q minimization is related to the value of q. For example, when $q = 2/3$, $\tilde{x}$ cannot be excluded from the ℓ_q-norm ball, as shown in Fig.2(a). While, when $q = 1/2$, $\bar{x}$ can be recovered. Generally, the performance of minimizing the ℓ_q-norm is improved but its computational complexity increases when decreasing q.

Compared with ℓ_1 minimization, minimizing the ℓ_q-norm enhances sparsity but looses computational efficiency. To maintain the computational advantage of ℓ_1 minimization, researchers try to find a suitable penalty from modifying the ℓ_1-norm. In [21], the truncated ℓ_1-norm was proposed:

$$L^a_{\text{trunc}}(x) = \sum\nolimits_{i=1}^{n} \min\{a, |x(i)|\}.$$

The contour map of the truncated ℓ_1-norm with $a = 4/5$ is illustrated in Fig.2(b), from which one can see that if value a is properly selected, such penalty can lead to a good result. In the region $|x(i)| > a$, L^a_{trunc} performs like the ℓ_0-norm, i.e., the result is sparse but the optimization task is hard. In the region $|x(i)| \le a$, L^a_{trunc} is equal to the ℓ_1-norm and the sparsity is not enhanced. To smooth the transition from the ℓ_0-norm to the ℓ_1-norm, one can consider the smoothly clipped absolute deviation penalty (SCAD) established in [22]. SCAD, of which the contour map is illustrated in Fig.2(c), has two parameters a_1 and a_2. SCAD equals the ℓ_1-norm when $|x(i)| < a_1$, equals the $\ell_{1/2}$-norm when $a_1 \le |x(i)| < a_2$, and equals the ℓ_0-norm when $|x(i)| \ge a_2$. Both the truncated ℓ_1-norm and SCAD enhance the sparsity. One efficient local algorithm was given in [23] by applying difference of convex functions (DC) programming, which was developed by [24] and [25]. The basic idea of DC programming is to do linearization using the gradient and minimize the linearized system. Obviously, if the linearization is accurate in a large range, the minimization technique based on the linearized system is effective.



(a) (b) (c) (d)

Fig. 2. The contour map of several non-convex penalties: (a) the ℓ_q-norm with $q = 2/3$; (b) the truncated ℓ_1-norm with $a = 4/5$; (c) the smoothly clipped absolute deviation penalty with $a_1 = 1/5, a_2 = 3/4$; (d) the expected penalty, which can exclude $\tilde{x}$ and is piecewise linear.

Our goal in this paper is to establish a new penalty, of which the contour map is shown in Fig.2(d). In this figure, $\tilde{x}$ is excluded and the level set is piecewise linear. Thus, we can on the one hand enhance the sparsity and on the other hand establish effective algorithms based on linearization. In Section II, we will give the corresponding analytic formulation. Generally, the proposed penalty considers two different penalty levels for different components according to the order of the component absolute values. The proposed penalty is hence called two-level ℓ_1-norm. The two-level ℓ_1-norm can be conducted in the framework of reweighted ℓ_1-norm. But differently to some existing reweighted methods, such as [15] and [26], which give weights based on the component values, we focus on the order of the absolute values. This difference makes it possible to establish a soft thresholding algorithm for the two-level ℓ_1-norm.

The remainder of the paper is organized as follows: the formulation of the two-level ℓ_1-norm is given in Section II, where we also discuss some properties and establish an iterative algorithm. In Section III, a soft thresholding algorithm for two-level ℓ_1 minimization is established and its convergence is proved. The proposed algorithms then are evaluated by numerical experiments in Section IV. Section V ends the paper with concluding remarks.

II. Two-level ℓ_1 Minimization

A. Two-level ℓ_1-norm

Define a penalty function

$$L^{\rho}(x) = \min\{\rho|x(1)|+|x(2)|+|x(3)|, |x(1)|+\rho|x(2)| +|x(3)|, |x(1)|+|x(2)|+\rho|x(3)|\}. \quad (4)$$

When $\rho = 1$, $L^{\rho}(x)$ reduces to the ℓ_1-norm. With a decreasing value of ρ, the sparsity is enhanced. One can verify that $L^{\frac{1}{3}}(x)$ gives the contour map shown in Fig.2(d). For some other ρ values, we plot the corresponding $L^{\rho}(x)$ in Fig.3(a)–3(c).

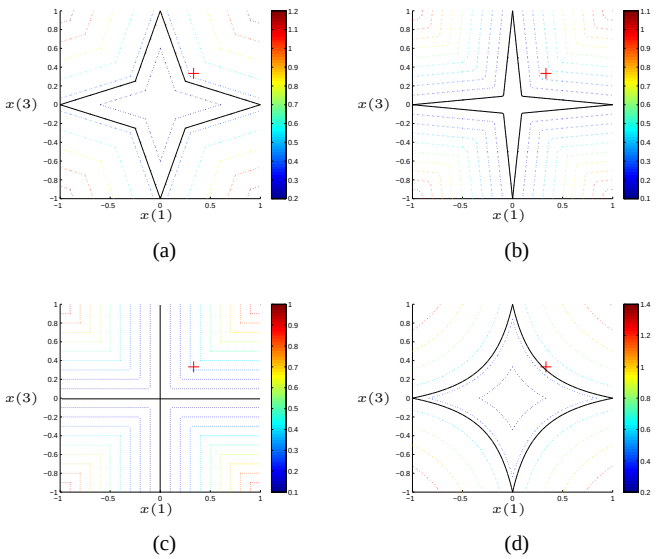


Fig. 3. The contour map of the two-level ℓ_1-norm and the iteratively reweighted ℓ_1-norm: (a) the two-level ℓ_1-norm with $\rho = 1/3$; (b) the two-level ℓ_1-norm with $\rho = 1/10$; (c) the two-level ℓ_1-norm with $\rho = 0$; (d) the iteratively reweighted ℓ_1-norm with $\epsilon = 1$.

Using (4), we aim at realizing the expected penalty in Fig.2(d). In a higher dimensional space, (4) is generalized into

$$L^{\rho,K}_{2\text{level}}(x) = \min_{I:|I|=K}\left\{\rho\sum_{i\in I}|x(i)| + \sum_{i\in I^c}|x(i)|\right\}, \quad (5)$$

where I is a subset of $\{1,2,\ldots,n\}$, $|I|$ stands for the cardinality of I, and I^c is the complement of S with respect to $\{1,2,\ldots,n\}$. In (5), there are two parameters ρ and K, which means that we give weight ρ for K components and weight 1 for the remainder. Thus, we call (5) *two-level ℓ_1-norm*.

Minimizing the two-level ℓ_1-norm, i.e., $\min\{L^{\rho,K}_{2\text{level}}(x), \text{s.t. } \Phi x = b\}$, is formulated as

$$\begin{aligned} \min_{x} \quad & \min_{I:|I|=K}\left\{\rho\sum_{i\in I}|x(i)| + \sum_{i\in I^c}|x(i)|\right\} \\ \text{s.t.} \quad & \Phi x = b. \end{aligned} \quad (6)$$

Obviously, (6) can be posed as

$$\begin{aligned} \min_{x,I} \quad & \rho\sum_{i\in I}|x(i)| + \sum_{i\in I^c}|x(i)| \\ \text{s.t.} \quad & \Phi x = b \\ & |I| = K. \end{aligned} \quad (7)$$

When I is determined, (7) becomes a linear programming problem and two-level ℓ_1 minimization (6) can be formulated as one mixed integer linear programming problem (MILP). However, solving MILP is not practical when the problem scale is too large. Though (6) can be written as an MILP, it is essentially a continuous problem and we will in the following establish a descent algorithm.

For any point x, we can find a subset, which is related to x and denoted as $I(x)$, such that

$$L^{\rho,K}_{2\text{level}}(x) = \rho\sum_{i\in I(x)}|x(i)| + \sum_{i\in I^c(x)}|x(i)|. \quad (8)$$

Simple calculation shows that selecting the largest K components of $|x(i)|$ to form $I(x)$ makes (8) valid. Therefore,

$$|I(x)| = K \text{ and } |x(i)| \geq |x(j)|, \forall i \in I(x), \forall j \in I^c(x). \quad (9)$$

In the rest of this paper, when we write an index set as a function of one vector, like $I(x)$, it implies that such index set satisfies condition (9).

Any I satisfying (9) defines a polyhedron

$$D_I = \{x : |x(i)| \geq |x(j)|, \forall i \in I, \forall j \in I^c\},$$

in which the linearization of $L^{\rho,K}_{2\text{level}}(x)$ remains. That means, for any x_0, there is

$$L^{\rho,K}_{2\text{level}}(x) = \rho\sum_{i\in I(x_0)}|x(i)| + \sum_{i\in I^c(x_0)}|x(i)|, \quad \forall x \in D_{I(x_0)}. \quad (10)$$

Hence, we can minimize a linear function in one polyhedron to get a descent for $L^{\rho,K}_{2\text{level}}(x)$ from x_0. Specifically, we consider the following linear programming problem related to I,

$$\begin{aligned} \min_{x,z,t} \quad & \rho\sum_{i\in I} z(i) + \sum_{i\in I^c} z(i) \\ \text{s.t.} \quad & \Phi x = b \\ & z(i) \geq x(i), z(i) \geq -x(i), \forall i \in \{1,2,\ldots,n\} \\ & z(i) \geq t, \forall i \in I \\ & z(i) \leq t, \forall i \in I^c. \end{aligned} \quad (11)$$

One can verify that when we choose $I = I(x_0)$, x_0 (with $z(i) = |x_0(i)|, t = \min_{i\in I(x_0)} z(i)$) is feasible to (11) and solving (11) can decrease the objective value of (6) from $L^{\rho,K}_{2\text{level}}(x_0)$. Repeating the above process leads to a local optimum of two-level ℓ_1 minimization. Since (6) is non-convex, a good initial point is needed. In this paper, we use the result of ℓ_1 minimization as the initial solution and give the following descent algorithm.

Algorithm 1: Descent method for two-level ℓ_1-norm

- Set l, i.e., the iteration counter, to be zero;
- Solve (3), denote the result as x_0 and select $I := I(x_0)$;

repeat

- Set $l := l + 1$;
- Solve (11) and denote the result as x_l;
- Select $I := I(x_l)$ (if there are multiple choices, select the one not equaling to $I(x_{l-1})$);

until $L^{\rho,K}_{2\text{level}}(x_l) = L^{\rho,K}_{2\text{level}}(x_{l-1})$;

- Algorithm ends and return x_l.

B. Iteratively reweighted method for two-level ℓ_1-norm

Minimizing the two-level ℓ_1-norm by the above algorithm can converge to a local optimum and enhance the sparsity from ℓ_1 minimization. To speed up the local search for (6), we in this subsection develop another algorithm based on iteratively reweighted ℓ_1-minimization. In Algorithm 1, the locally linearized function (8) is minimized with constraint $x \in D_{I(x)}$. If we ignore this constraint, then minimizing (8) becomes a weighted ℓ_1 minimization problem,

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{n} w(i)|x(i)| \\ \text{s.t.} \quad & \Phi x = b, \end{aligned} \tag{12}$$

where $w(i) = \rho, i \in I(x)$ and $w(i) = 1, i \in I^c(x)$.

Weighted ℓ_1 minimization has been implemented in [15] to establish an iterative algorithm for enhancing sparsity. The iteratively reweighted procedure is setting

$$w_l(i) = \frac{1}{|x_{l-1}(i)| + \epsilon}, \tag{13}$$

where ϵ is a pre-given parameter, and solving (12) to update x_l. We can normalize the weight vector w such that $\|w\|_{\ell_2} = 1$ and plot the equivalent penalty in Fig.3(d), from which one can see its effect for enhancing sparsity. However, the convergence of the iteratively reweighted ℓ_1 minimization cannot be guaranteed. That algorithm essentially utilizes linearization at the current solution and minimizes the linearized system. It is well-known that without line search, minimization based on the locally linearized system can not guarantee the decrease of the objective value. In [15], the authors pre-defined the maximal number of iterations.

Drawing lessons from the success of iteratively reweighted ℓ_1 minimization, we do linearization for $L_{2\text{level}}^{\rho,K}(x)$ at the current solution and solve (12), which essentially minimizes the local objective function without constraints. Then we update the solution and the corresponding weight, and solve (12) again. The iteratively reweighted method for the two-level ℓ_1-norm is summarized in the following algorithm.

Algorithm 2: Reweighted method for two-level ℓ_1-norm

- Set $l := 0$ and $l_{\max}$ (the maximal number of iterations);
- Solve (3) and denote the result as x_0;

repeat

- Set $l := l + 1$;
- Set $w(i) := \rho, i \in I(x_{l-1})$, $w(i) := 1, i \in I^c(x_{l-1})$;
- Solve (12) and denote the result as x_l;

until $x_l = x_{l-1}$;

- Algorithm ends and return x_l.

In Algorithm 2, the stop criterion can be checked by $\|x_l - x_{l-1}\|_{\ell_2}$. In the experiments of this paper, the loop ends when $\|x_l - x_{l-1}\|_{\ell_2} \leq 10^{-6}$. The basic idea of the iteratively reweighted ℓ_1 minimization is that people care less about the components which have large absolute values, since they are not easy to become zero. The reweighted method for minimizing the two-level ℓ_1 has the similar idea: in Algorithm 2, we give a small penalty on the components which have large absolute values. The difference is that we do not care about the value of $|x(i)|$ but focus on the order of $|x(i)|$. This modification can further enhance the sparsity, which will be seen in numerical experiments, and reduce the computational time, which will be supported by numerical study and can be explained that the effective region of linearization of the two-level ℓ_1-norm is larger than that of the penalty used in the iteratively reweighted ℓ_1 minimization.

Similarly to the discussion in [15] on reweighted ℓ_1 minimization, we cannot prove convergence for Algorithm 2 neither. By comparison, Algorithm 1 can converge to a local optimum of (6). However, the convergence needs many steps and solving (11) takes much more time than (12). Though the precision of Algorithm 2 is less than Algorithm 1, numerical experiments will show that Algorithm 2 already gives good sparse estimation performance and hence we prefer to use Algorithm 2 in practice.

C. Analysis based on null space property

As analyzed previously, with the decrease of ρ, minimizing the two-level ℓ_1-norm takes more computational time and leads to a more sparse result. If we take $\rho = 0$, minimizing the two-level ℓ_1-norm has some similarity to the iterative support detection (ISD, [26]), which sets the weights for components larger than one threshold to be zero. Though ISD focuses on the value not the order of the absolute values of components, the analysis in [26] is applicable for the two-level ℓ_1-norm as well. The analysis for ISD is mainly based on the *null space property* (NSP). We use x_I to denote a vector satisfying

$$x_I(i) = \begin{cases} x(i), & i \in I, \\ 0, & i \notin I. \end{cases}$$

Then NSP can be expressed as below,

Definition 1: For a matrix $\Phi \in \mathbb{R}^{m\times n}$, if for order s and $\gamma > 0$, there is

$$\|\eta_S\|_{\ell_1} \leq \gamma \|\eta_{S^c}\|_{\ell_1}, \forall \eta \in \mathcal{N}(\Phi), \forall S : |S| \leq s,$$

where $\mathcal{N}(\Phi)$ stands for the null space of Φ, then we say that Φ has the null space property of order s for γ.

NSP was defined in [12], where the relationship between NSP and the *restricted isometry property* (RIP, [10]) was also given: if matrix Φ has RIP of order k with constant δ, then Φ satisfies NSP of order s_1 for $\gamma = \sqrt{s_2/s_1}(1+\delta)/(1-\delta)$, where $s_1 + s_2 = k$. In [27] and [28], NSP has been applied to investigate the properties for the iteratively reweighted least squares minimization. To analyze the performance of ISD, researchers extended NSP to the truncated-NSP, which was defined in [26] and is reviewed below,

Definition 2: For a matrix $\Phi \in \mathbb{R}^{m\times n}$, if for order s, integer h, and $\gamma > 0$, there is

$$\begin{aligned} \|\eta_S\|_{\ell_1} \leq \gamma \|\eta_{H\cap S^c}\|_{\ell_1}, \\ \forall \eta \in \mathcal{N}(\Phi), \forall S : |S| \leq s, \forall H : |H| = h, \end{aligned}$$

then Φ has truncated-NSP of order s for h and γ.

Using truncated-NSP, we give a sufficient condition for successful recovery using two-level ℓ_1 minimization, as below.

Theorem 1: Suppose $y = \Phi\bar{x}$ and Φ has truncated-NSP of order $\|\bar{x}\|_{\ell_0}$ for K and γ. If $\gamma < 1$, $\bar{x}$ is the minimizer of (6).

Proof: $\bar{x}$ being a minimizer of (6) means that

$$L_{2\text{level}}^{\rho,K}(\bar{x}+v) \geq L_{2\text{level}}^{\rho,K}(\bar{x}), \forall v \in \mathcal{N}(\Phi).$$

Denote $I_0 = I(\bar{x})$, $I_1 = I(\bar{x}+v)$, and $\bar{S} = \text{supp}(\bar{x})$, where $\text{supp}(\bar{x})$ is the support set of $\bar{x}$. For any index set $I \subseteq \{1, 2, \ldots, n\}$, there is

$$\|\bar{x}_I\|_{\ell_1} = \|\bar{x}_{I\cap\bar{S}}\|_{\ell_1}.$$

Therefore,

$$\begin{aligned} L_{2\text{level}}^{\rho,K}(\bar{x}+v) &= \rho\|\bar{x}_{I_1} + v_{I_1}\|_{\ell_1} + \|\bar{x}_{I_1^c} + v_{I_1^c}\|_{\ell_1} \\ &= \rho\left(\|\bar{x}_{I_1\cap\bar{S}} + v_{I_1\cap\bar{S}}\|_{\ell_1} + \|v_{I_1\cap\bar{S}^c}\|_{\ell_1}\right) \\ &\quad + \|\bar{x}_{I_1^c\cap\bar{S}} + v_{I_1\cap\bar{S}}\|_{\ell_1} + \|v_{I_1^c\cap\bar{S}^c}\|_{\ell_1}. \end{aligned}$$

Considering the first item, we have

$$\begin{aligned} & \|\bar{x}_{I_1\cap\bar{S}} + v_{I_1\cap\bar{S}}\|_{\ell_1} + \|v_{I_1\cap\bar{S}^c}\| \\ = \; & \left(\|\bar{x}_{I_1\cap\bar{S}} + v_{I_1\cap\bar{S}}\|_{\ell_1} - \|\bar{x}_{I_1\cap\bar{S}}\|_{\ell_1} + \|v_{I_1\cap\bar{S}}\|_{\ell_1}\right) \\ & + \|\bar{x}_{I_1\cap\bar{S}}\|_{\ell_1} + \|v_{I_1\cap\bar{S}^c}\|_{\ell_1} - \|v_{I_1\cap\bar{S}}\|_{\ell_1} \\ \geq \; & \|\bar{x}_{I_1}\|_{\ell_1} + \left(\|v_{I_1\cap\bar{S}^c}\|_{\ell_1} - \|v_{\bar{S}}\|_{\ell_1}\right) \\ \geq \; & \|\bar{x}_{I_1}\|_{\ell_1}. \end{aligned}$$

The last inequality comes from the condition that Φ has truncated-NSP of order $\|\bar{x}\|_{\ell_0}$ for K and γ.

Similarly, we know that

$$\|\bar{x}_{I_1^c\cap\bar{S}} + v_{I_1\cap\bar{S}}\|_{\ell_1} + \|v_{I_1^c\cap\bar{S}^c}\| \geq \|\bar{x}_{I_1^c}\|_{\ell_1},$$

and hence

$$L_{2\text{level}}^{\rho,K}(\bar{x}+v) \geq \rho\|\bar{x}_{I_1}\|_{\ell_1} + \|\bar{x}_{I_1^c}\|_{\ell_1}.$$

According to the definition of $I(\bar{x})$, we know that I_0 includes k components with the largest absolute values $|\bar{x}(i)|$, thus

$$\rho\|\bar{x}_{I_1}\|_{\ell_1} + \|\bar{x}_{I_1^c}\|_{\ell_1} \geq \rho\|\bar{x}_{I_0}\|_{\ell_1} + \|\bar{x}_{I_0^c}\|_{\ell_1} = L_{2\text{level}}^{\rho,K}(\bar{x}),$$

which follows that

$$L_{2\text{level}}^{\rho,K}(\bar{x}+v) \geq L_{2\text{level}}^{\rho,K}(\bar{x}), \forall v \in \mathcal{N}(\Phi),$$

and this theorem is proved. ■

III. Thresholding Algorithm for Minimizing Two-level ℓ_1-norm

To suppress additive noise in measurements, researchers give penalty on the violation of $\Phi x = b$ and modify ℓ_1 minimization (3) into the following problem,

$$\min_x \quad \|\Phi x - b\|_{\ell_2}^2 + 2\tau\|x\|_{\ell_1}, \tag{14}$$

where $\tau > 0$ is a trade-off parameter related to the noise level. For solving (14), an iterative soft thresholding algorithm has been established in [11] and applied widely. Similarly, we in this section consider

$$\min_x \quad \|\Phi x - b\|_{\ell_2}^2 + 2\tau L_{2\text{level}}^{\rho,K}(x), \tag{15}$$

and establish a thresholding algorithm.

Define a soft thresholding operator $\mathbb{S}^{\tau,K}$ as

$$\begin{aligned} & \left(\mathbb{S}^{\tau,K,\rho}(x)\right)(i) \\ = \; & \begin{cases} x(i) - \rho\tau\text{sign}(x(i)), & \text{if } i \in I(x), |x(i)| \geq \rho\tau, \\ 0, & \text{if } i \in I(x), |x(i)| < \rho\tau, \\ x(i) - \tau\text{sign}(x(i)), & \text{if } i \in I^c(x), |x(i)| \geq \tau, \\ 0, & \text{if } i \in I^c(x), |x(i)| < \tau. \end{cases} \end{aligned}$$

We can write the local optimality condition of (15) as

$$x = \mathbb{S}^{\tau,K,\rho}(x + \Phi^T(b - \Phi x)). \tag{16}$$

Then motivated by the optimality condition (16), we establish the following iterative soft thresholding algorithm for (15).

Algorithm 3: Two-level soft thresholding algorithm

- Set $l := 0$ and $x_0 := \Phi^T b$;

repeat

- $x_{l+1} := \mathbb{S}^{\tau,K,\rho}(x_l + \Phi^T(b - \Phi x_l))$;
- Set $l := l + 1$;

until $x_l := x_{l-1}$;

- Algorithm ends and return x_l.

According to the local optimality condition (16), we can expect that the stationary point of the two-level soft thresholding algorithm is a local optimum of (15). In the following, we will prove the convergence of Algorithm 3. The proof is different to that of the soft thresholding algorithm proposed by [11] for the ℓ_1-norm, since $\mathbb{S}^{\tau,K,\rho}$ itself is expansive. Instead of analysis based on non-expansive operator, the convergence of Algorithm 3 is guaranteed by showing that a surrogate function can be decreased by the proposed two-level soft thresholding.

Theorem 2: When $\|\Phi\|_2 < 1$, Algorithm 3 converges to a local optimum of (15).

Proof: Define the following function $F : \mathbb{R}^{2n} \mapsto \mathbb{R}$:

$$F(x,y) = \|\Phi x - b\|_{\ell_2}^2 + \|x - y\|_{\ell_2}^2 + 2\tau L_{2\text{level}}^{\rho,K}(y). \tag{17}$$

Algorithm 3 defines the following update formulation,

$$\begin{aligned} y_l &= \mathbb{S}^{\tau,K,\rho}(x_l), \\ x_{l+1} &= y_l + \Phi^T(b - \Phi y_l). \end{aligned}$$

We will show that the process above can iteratively decrease the value of $F(x,y)$, which is guaranteed by proving that for any $y \in \mathbb{R}^n$,

$$F(x_l, y) \geq F(x_l, y_l) \geq F(y_l, y_l) \geq F(x_{l+1}, y_l). \tag{18}$$

When x is fixed to be x_l, $F(x,y)$ is a function with respect to y, denoted by $g_{x_l}(y)$. Suppose y^* is the minimum of $g_{x_l}(y)$, then $g_{x_l}(y)$ can be written as a separable function as $g_{x_l}(y) = c_{x_l} + \sum_{i=1}^n g_{x_l, I(y^*)}^i(y(i))$, where $c_{x_l} = \|\Phi x_l - b\|_{\ell_2}^2$ is a constant and

$$\begin{aligned} & g_{x_l, I(y^*)}^i(y(i)) \\ = \; & \begin{cases} (x_l(i) - y(i))^2 + \tau|x_l(i) - y(i)|, & i \in I(y^*), \\ (x_l(i) - y(i))^2 + \rho\tau|x_l(i) - y(i)|, & i \notin I(y^*). \end{cases} \end{aligned}$$

Therefore, simple calculation for the minimum of $g_{x_l, I(y^*)}^i$ shows that for any $i \in I(y^*)$,

$$y^*(i) = \begin{cases} x_l(i) - \rho\tau\text{sign}(x_l(i)), & |x_l(i)| \geq \rho\tau, \\ 0, & |x_l(i)| < \rho\tau. \end{cases}$$

Similarly, for any $i \in I^c(y^*)$,

$$y^*(i) = \begin{cases} x_l(i) - \tau \text{sign}(x_l(i)), & |x_l(i)| \geq \tau, \\ 0, & |x_l(i)| < \tau. \end{cases}$$

In the following, we show that $g_{x_l}(y_l) \leq g_{x_l}(y^*)$. Recalling the formulation of $y_l = \mathbb{S}^{\tau,K,\rho}(x_l)$, one can verify that

$$y^*(i) = y_l(i), \forall i \in (I(x_l) \cap I(y^*)) \cup (I^c(x_l) \cap I^c(y^*)).$$

Then we focus only on $I(x_l) \bigcap I^c(y^*)$. If there exists an i such that $i \in I(x_l) \bigcap I^c(y^*)$, one can find another index such that $j \in I^c(x_l) \bigcap I(y^*)$, since $|I(x_l)| = |I^c(y^*)|$. Then $|x_l(i)| > |x_l(j)|$ and $|y_l^*(i)| < |y_l^*(j)|$.

There are the following four cases:

1) $|x_l(i)| \leq \tau, |x_l(j)| \leq \rho\tau$, then $y_l^*(i) = y_l^*(j) = 0$, which conflicts with the assumption $|y_l^*(i)| < |y_l^*(j)|$;
2) $|x_l(i)| \leq \tau, |x_l(j)| > \rho\tau$, then

$$\begin{aligned} & g^i_{x_l, I(y^*)}(y^*(i)) + g^j_{x_l, I(y^*)}(y^*(j)) \\ = \; & x_l^2(i) + \rho|x_l(i)| + 3\rho^2\tau^2 \\ \geq \; & 3\rho^2\tau^2 + x_l^2(j) + \rho|x_l(j)| \\ \geq \; & g^i_{x_l, I(x_l)}(x_l(i) - \rho\tau \text{sign}(x_l(i))) + g^j_{x_l, I(x_l)}(0) \\ = \; & g^i_{x_l, I(y_l)}(y_l(i)) + g^j_{x_l, I(y_l)}(y_l(j)); \end{aligned}$$

3) $|x_l(i)| > \tau, |x_l(j)| \leq \rho\tau$, then $y_l^*(i) = x_l(i) - \tau \text{sign}(x_l(i))$ and $y_l^*(j) = 0$, which conflicts with the assumption $|y_l^*(i)| < |y_l^*(j)|$;
4) $|x_l(i)| > \tau, |x_l(j)| > \rho\tau$, then

$$\begin{aligned} & g^i_{x_l, I(y^*)}(y^*(i)) + g^j_{x_l, I(y^*)}(y^*(j)) \\ = \; & 3\tau^2 + 3\rho^2\tau^2 \\ = \; & g^i_{x_l, I(y_l)}(y_l(i)) + g^j_{x_l, I(y_l)}(y_l(j)). \end{aligned}$$

The discussion above means $g_{x_l}(y_l) \leq g_{x_l}(y^*)$, i.e., $y_l = \mathbb{S}^{\tau,K,\rho}(x_l)$ is a minimum of $\min_y F(x_l, y)$, which proves the validity of the first inequality of (18).

The second inequality of (18) equally means

$$\|\Phi x_l - b\|_{\ell_2}^2 + \|x_l - y_l\|_{\ell_2}^2 - \|\Phi y_l - b\|_{\ell_2}^2 \geq 0. \tag{19}$$

According to the fact

$$\|\Phi x_l - b\|_{\ell_2}^2 - \|\Phi y_l - b\|_{\ell_2}^2 \geq -\|\Phi(x_l - y_l)\|_{\ell_2}^2,$$

we know that the left hand of (19) is larger than or equal to

$$\begin{aligned} & \|x_l - y_l\|_{\ell_2}^2 - \|\Phi(x_l - y_l)\|_{\ell_2}^2 \\ = \; & (x_l - y_l)^T \left(\mathbf{I}_{n\times n} - \Phi^T\Phi\right)(x_l - y_l) \\ \geq \; & 0, \end{aligned}$$

where the last inequality comes from $\|\Phi\|_2 < 1$.

Next, we verify the third inequality of (18) by considering the following optimization problem,

$$\min_x \|\Phi x - b\|_{\ell_2}^2 + \|x - y\|_{\ell_2}^2. \tag{20}$$

Its optimality condition can be written as

$$-\Phi^T(b - \Phi x) + (x - y_l) = 0,$$

from which, we formulate the following update scheme

$$x_{h+1} = y_l + \Phi^T(b - \Phi x_h), \tag{21}$$

where h indicates the iteration step. Similarly to the analysis for Landweber iterative method, one can find that (21) produces a non-increasing sequence of the objective value of (20), if $\|\Phi\|_2 < 1$. Moreover, by setting $x_h = y_l$, (21) indeed gives x_{l+1}. Hence, we know that $F(y_l, y_l) \geq F(x_{l+1}, y_l)$.

From (18), we can expect that iterations defined by Algorithm 3 provide a non-increasing sequence of $F(x, y)$. Suppose that after l steps, $F(x, y)$ converges, i.e., $F(x_l, y_l) = F(x_{l+1}, y_{l+1})$. According to (18), we have

$$F(x_l, y_l) = F(y_l, y_l) = F(x_{l+1}, y_l) = F(x_{l+1}, y_{l+1}). \tag{22}$$

The last equality of (22) means that y_l satisfies the optimality condition for $g_{x_{l+1}}(y)$, i.e., $y_l = y_{l+1} = \mathbb{S}^{\tau,K,\rho}(x_{l+1})$, where x_{l+1} is obtained from (21) by setting $x_h = y_l$. Thus, we have

$$y_{l+1} = \mathbb{S}^{\tau,K,\rho}(y_l + \Phi^T(b - \Phi y_l)),$$

which means that y_{l+1} meets (16), the optimality condition of (15). Therefore, we proved that Algorithm 3 converges to a local optimum of (15). ■

IV. Numerical Experiments

In this paper, we introduce the two-level ℓ_1-norm for sparse signal recovery. From Fig.3, one can expect the sparsity of the two-level ℓ_1-norm and its computational effectiveness, compared with other non-convex penalties. There are two parameters in (5): the number of components having a small penalty and the weight for such components. Parameter ρ trades off between sparsity and computational complexity, which can be observed from the contour map of $L_{2\text{level}}^{\rho,K}(x)$. When ρ is small, the result is more sparse but is harder to solve. In the following experiments, we will test the performance of the two-level ℓ_1 minimization with $\rho = 0, \frac{1}{10}, \frac{1}{3}$.

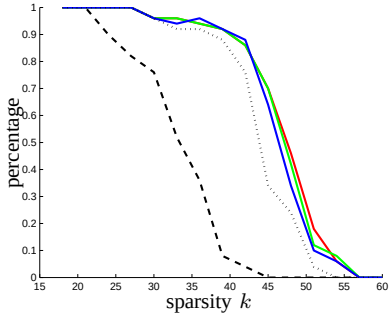
For any $\rho < 1$, $L_{2\text{level}}^{\rho,K}(x)$ is non-convex. The established algorithms do not have global search capability. To achieve a good solution, we use the result of ℓ_1-minimization, denoted by x_0, as the starting point for Algorithm 1 and Algorithm 2. There are $\|x_0\|_{\ell_0}$ non-zero components in x_0 and we minimize the two-level ℓ_1-norm to enhance the sparsity from x_0. $L_{2\text{level}}^{\rho,K}(x)$ gives less penalty for K components and tries to make the rest $\|x_0\|_{\ell_0} - K$ components to be zero. Hence, when we have no prior knowledge about the sparsity of real signals, we set K according to x_0. In the experiments, we test several K values and suggest to use $K = \lfloor \frac{1}{3}\|x_0\|_{\ell_0} \rfloor$, where $\lfloor a \rfloor$ stands for the smallest integer not less than a. The proposed two-level soft thresholding algorithm (Algorithm 3) does not calculate x_0, but we can expect that $\|x_0\|_{\ell_0}$ is less than m, the number of equality constraints. Hence, for Algorithm 3, K can be heuristically chosen to be $K = \lfloor \frac{M}{5} \rfloor$. All the experiments are done in Matlab R2011a in Core 2-2.83 GHz, 2.96G RAM.

A. Simulated sparse signal recovery

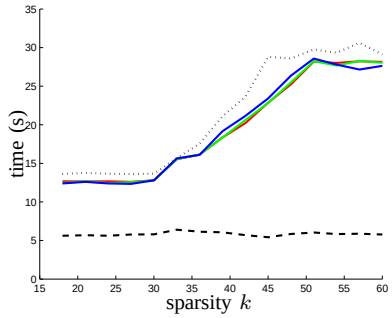
We first evaluate the performance of two-level ℓ_1 minimization on sparse signal recovery problem and compare the result with ℓ_1 minimization and iteratively reweighted ℓ_1 minimization. We randomly generate a k-sparse signal $\bar{x} \in \mathbb{R}^n$, i.e., $\|\bar{x}\|_{\ell_0} = k$, and a sampling matrix $\Phi \in \mathbb{R}^{m\times n}$. Then we

solve (6) to recover the signal from the observed $b = \Phi\bar{x}$. In this paper, the experiment used in [15] is repeated: we set $n = 256, m = 100$, and the components of $\bar{x}$ and the elements of Φ come from standard Gaussian distribution. Different k-sparse signals are considered. We first solve ℓ_1 minimization (3) and if the maximum absolute error between the result and $\bar{x}$ is smaller than 10^{-3}, we say that the signal is recovered. For each k, we repeat the above process 100 times, calculate the recovery percentage and the average computational time.

In this experiment, we recover the signal by Algorithm 2 in comparison with the results of ℓ_1 minimization and iteratively reweighted ℓ_1 minimization. These methods involve only linear programming and we simply use the linear programming solver embedded in Matlab. As introduced previously, there is one parameter in the update formulation (13) of iteratively reweighted ℓ_1 minimization. Besides, one should give the maximum number of iterations $l_{\max}$. According to the result given in [15], we set $\epsilon = 0.1$ and $l_{\max} = 4$. Then we apply Algorithm 2 to minimize the two-level ℓ_1-norm for signal recovery. In Algorithm 2, we also set $l_{\max} = 4$ and report the obtained results for different ρ in Fig.4.



(a)

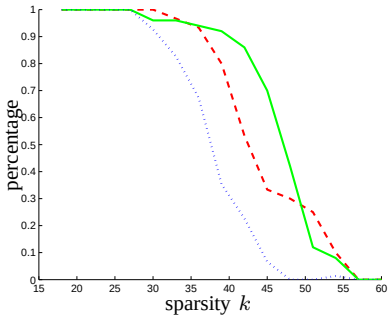


(b)

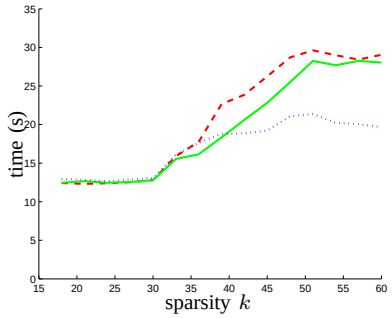
Fig. 4. Performance on signal recovery of ℓ_1 minimization (dashed line, [1]), iteratively reweighted ℓ_1 minimization (dotted line, [15]), and two-level ℓ_1 minimization ($\rho = 0$ red line; $\rho = 1/10$ green line; $\rho = 1/3$ blue line): (a) recovery percentage; (b) average computational time.

From Fig.4(a), one can see that two-level ℓ_1 minimization indeed can enhance the sparsity from the result of ℓ_1 minimization. We are also concerned about the average computational time, which is plotted in Fig4(b). One can see that the computational time of Algorithm 2 is not more than that of the iteratively reweighted ℓ_1 minimization, which coincides with our expectation in Section II-B. Generally, a small ρ corresponds to a more computational time and a more sparse result. But the difference between different ρ is not significant and we suggest $\rho = 1/10$ in practice.

As mentioned previously, the results plotted in Fig.4 are obtained by using $K = \lfloor \|x_0\|_{\ell_0}/3 \rfloor$, where x_0 is the result of ℓ_1 minimization (3). In the following, we test different K values for $\rho = 1/10$. The corresponding recovery percentage and average computational time are illustrated in Fig.5, from which we suggest $K = \lfloor \|x_0\|_{\ell_0}/3 \rfloor$.



(a)



(b)

Fig. 5. Performance on signal recovery for K: $K = \lfloor \|x_0\|_{\ell_0}/2 \rfloor$ (red dashed line), $K = \lfloor \|x_0\|_{\ell_0}/3 \rfloor$ (green solid line), and $K = \lfloor \|x_0\|_{\ell_0}/10 \rfloor$ (blue dotted line): (a) recovery percentage; (b) average computational time.

B. Simulated error correction

One interesting variation of compressed sensing is proposed in [10] for decoding, where a fraction of codeword are corrupted by noise. Again, we repeat the experiment in [15] for this application. In this experiment, a signal $\bar{x} \in \mathbb{R}^n$ and a coding matrix $\Phi \in \mathbb{R}^{m\times n}$ with $m = 512, n = 128$ are randomly generated by choosing entries from a standard Gaussian distribution. We wish to decode $\bar{x}$ from the received codeword $b = \Phi\bar{x} + e$, where e is an unknown sparse corruption pattern. e being sparse means that k components of the codeword are corrupted by completely arbitrary value. In this experiment, we randomly select k components from b and let them be corrupted by a sign flip. To decode $\bar{x}$ from b, [10] solves the following ℓ_1 minimization problem,

$$\min_x \quad \|\Phi x - b\|_{\ell_1}. \tag{23}$$

Denote the result of decoding as $\tilde{x}$. Again, if $\|\bar{x} - \tilde{x}\|_{\ell_\infty} < 10^{-3}$, we claim that the error is corrected and the signal is

recovered. The percentage of recovery using (23) is shown in Fig.6(a) by dashed line. To improve the recovery performance, one can follow the idea of (12) and consider the following reweighted problem,

$$\min_{x} \quad \|W(\Phi x - b)\|_{\ell_1}, \tag{24}$$

where W is the diagonal matrix with diagonal entries w_1, $w_2, \ldots, w_m$ and zero otherwise. In [15], W is iteratively updated according to the components of the residual $\Phi x_{l-1} - b$ and the formulation is similar to (13). To apply the two-level ℓ_1-norm in error correction, Algorithm 2 can be modified into:

Algorithm 4: Two-level soft thresholding algorithm

- Set $l := 0$ and $l_{\max}$ (the maximal number of iterations);
- Solve (23) and denote the result as x_0;

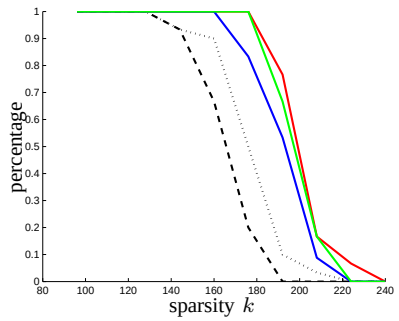
repeat

- Set $l := l + 1$;
- Let $r := \Phi x_{l-1} - b$ and then accordingly set $w_i := \rho, i \in I(r)$, $w_i := 1, i \in I^c(r)$;
- Solve (24) and denote the result as x_l;

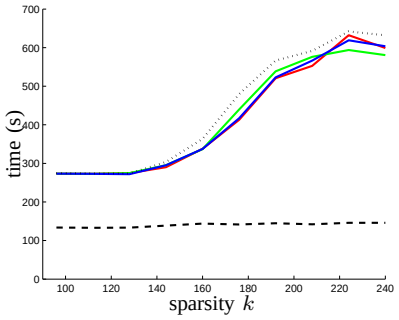
until $x_l := x_{l-1}$;

- Algorithm ends and return x_l.

For each k, we repeat the process above 100 times and calculate the recovery percentage for ℓ_1 minimization, iteratively reweighted ℓ_1 minimization, and the proposed two-level ℓ_1 minimization. In Fig.6(a) and Fig.6(b), we report the recovery percentage and the average computational time for different k, respectively, showing the performance of Algorithm 4.



(a)



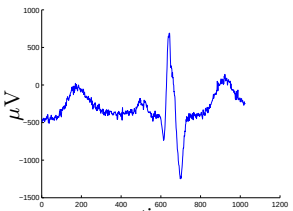
(b)

Fig. 6. Performance on error correction of ℓ_1 minimization (dashed line, [1]), iteratively reweighted ℓ_1 minimization (dotted line, [15]), and two-level ℓ_1 minimization ($\rho = 0$ red line; $\rho = \frac{1}{10}$ green line; $\rho = \frac{1}{3}$ blue line): (a) recovery percentage; (b) average computational time.

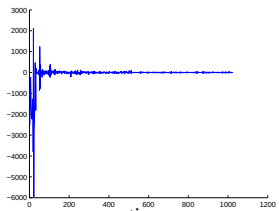
C. ECG signal recovery

In the above synthetic experiments, we have shown the performance of two-level ℓ_1 minimization. Generally, though two-level ℓ_1 minimization is non-convex and global optimality cannot be guaranteed, the algorithms for two-level ℓ_1 minimization give more sparse results in practice. In the following, we consider (15) and evaluate Algorithm 1, which are applicable to many biomedical signals. Here, we consider real-life electrocardiography (ECG) data as an example, and compare the performance of Algorithm 1 with the result of the soft thresholding algorithm, which solves ℓ_1 minimization (14). In this experiment, we set $\tau = 1$ for both (14) and (15), according to the noise level.

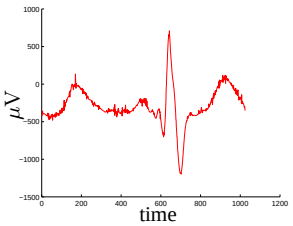
The used ECG data come from the National Metrology Institute of Germany, which is online available in the PhysioNet [29], [30]. The record includes 15 simultaneously measured signals sampled from one subject simultaneously. Each signal is digitized at 1000 samples per second, with 16 bit resolution over a range of ± 16.384 mV. On special request to the contributors of the database, recordings may be available at sampling rates up to 10 kHz. For each signal channel, there are 38400 data points. We start from the first 1024 data and randomly generate one Gaussian matrix $\Phi \in \mathbb{R}^{m \times n}$, where $n = 1024$ and m varies from 64 to 1024. Then we move to the next 1024 data and so on. The ECG data is not sparse in the time domain as shown in Fig.8(a), which plots $\bar{x}$ for time $1 \sim 1024$ of signal channel no.1. Following the suggestion in [31], we apply the orthogonal Daubechies wavelets (db 10), which is reported to be the most popular wavelet family for ECG compression, to design Ψ and get $\bar{\theta}$ such that $\bar{x} = \Psi\bar{\theta}$. Then, θ has sparsity as shown in Fig.8(b). We generate one Φ with $m = 300$ and compute $b = \Phi\Psi\bar{\theta}$. Then we can apply Algorithm 3 to recover θ. The resulted $\tilde{\theta}$ and the corresponding reconstructed signal $\tilde{x} = \Psi\tilde{\theta}$ are illustrated by Fig.7(c) and Fig.7(d), respectively.



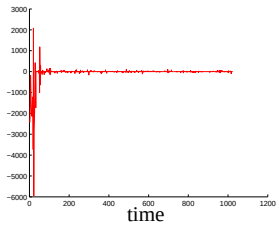
(a)



(b)



(c)



(d)

Fig. 7. The ECG signal in signal channel No.1: (a) $\bar{x}$; (b) $\bar{\theta}$; (c) $\tilde{x}$ recovered by two-level soft thresholding; (d) $\tilde{\theta}$ recovered by two-level soft thresholding.

From Fig.7, one can see that two-level soft thresholding can recover sparse patten θ and also has de-noising effect. To measure the accuracy of recovery, we calculate the mean of squared error (MSE) between the recovered signal and the original signal. The MSEs for $m = 64, 128, 256, 512$ are reported in Table I, where the computational times and the number of nonzero components in $\tilde{\theta}$ are given as well. We also illustrate the performance of soft thresholding and two-level soft thresholding for four signal channels in Fig.8, where the average MSE and average computing time for each m are plotted. By comparison, we find the good performance of Algorithm 3, which uses a bit more time than soft thresholding and achieves a better accuracy, especially when m is between 100 and 500.

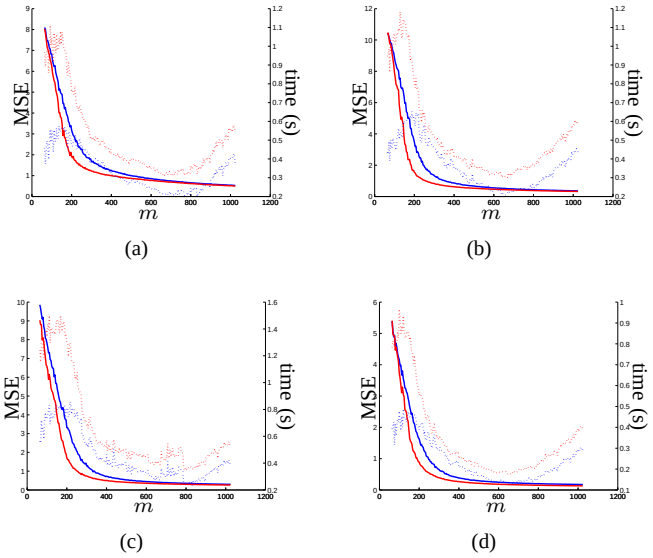


Fig. 8. MSE (solid lines) and computational time (dotted lines) of soft thresholding (blue lines) and two-level soft thresholding (red lines, [11]): (a) channel no.1; (b) channel no.6; (c) channel no.10; (d) channel no.14.

V. Conclusion and Further Study

To enhance the sparsity of the ℓ_1-norm, we proposed the two-level ℓ_1-norm in this paper. From the plots of $L^{\rho,K}_{\text{2level}}(x)$, one can observe the sparsity of the result of two-level ℓ_1 minimization. Meanwhile, $L^{\rho,K}_{\text{2level}}(x)$ is piecewise linear and optimization technique based on locally linearized system can be effectively used for minimizing the two-level ℓ_1-norm. In this paper, we established a two-level reweighted ℓ_1 minimization, which takes less time than iteratively reweighted ℓ_1 minimization. Also, a two-level soft thresholding algorithm was given and its convergence was proved. Though the two-level ℓ_1-norm is not convex, the proposed algorithms have illustrated good performance on numerical experiments.

In the future, one can consider global search methods for the two-level ℓ_1-norm and enhance the sparsity further. Besides, accelerated techniques are also interesting, for which one can explore some techniques designed for ℓ_1 minimization, including fast interactive soft thresholding (FISTA) [32], fast weighted interactive soft thresholding (FWISTA) [33], approximate message passing (AMP) [34], etc.

TABLE I
Recovery Performance for ECG signal

No.	m	soft thresholding			two-level soft thresholding		
		MSE	Time (s)	#nonzero	MSE	Time (s)	#nonzero
1	64	8.097	0.457	#94	8.035	0.966	#92
	128	5.649	0.509	#159	4.794	0.987	#156
	256	2.099	0.456	#255	1.393	0.617	#248
	512	0.977	0.291	#406	0.859	0.399	#390
2	64	11.02	0.539	#104	11.22	1.090	#101
	128	8.311	0.598	#169	6.836	1.211	#166
	256	2.077	0.568	#251	1.122	0.619	#236
	512	0.737	0.297	#368	0.616	0.378	#351
3	64	11.41	0.464	#104	11.22	0.988	#102
	128	8.228	0.507	#173	6.832	1.026	#170
	256	2.638	0.602	#263	1.425	0.769	#253
	512	0.859	0.348	#390	0.690	0.449	#372
4	64	7.945	0.422	#92	8.229	0.882	#91
	128	5.910	0.477	#158	5.408	0.983	#155
	256	1.705	0.482	#240	1.065	0.556	#230
	512	0.754	0.257	#368	0.642	0.337	#350
5	64	8.411	0.459	#96	8.410	0.914	#93
	128	5.698	0.603	#159	4.385	1.216	#154
	256	1.996	0.651	#254	1.269	0.801	#243
	512	0.845	0.440	#387	0.716	0.559	#370
6	64	10.38	0.436	#103	10.39	0.947	#101
	128	7.562	0.538	#167	5.439	1.183	#162
	256	1.816	0.492	#243	0.995	0.574	#228
	512	0.641	0.272	#349	0.519	0.354	#327
7	64	10.79	0.617	#101	10.60	1.145	#100
	128	7.462	0.651	#171	7.080	1.187	#171
	256	2.192	0.576	#256	1.033	0.803	#227
	512	0.526	0.323	#314	0.385	0.423	#287
8	64	12.72	0.474	#109	12.66	0.960	#107
	128	9.206	0.550	#178	7.400	1.157	#171
	256	2.422	0.638	#254	1.024	0.752	#226
	512	0.524	0.313	#317	0.388	0.388	#282
9	64	16.17	0.560	#119	15.58	1.168	#115
	128	11.59	0.767	#184	9.091	1.463	#181
	256	3.623	0.849	#275	1.404	1.149	#248
	512	0.577	0.462	#331	0.406	0.582	#295
10	64	9.855	0.562	#100	9.032	1.243	#97
	128	6.490	0.732	#164	5.198	1.375	#164
	256	1.905	0.730	#245	0.982	0.956	#220
	512	0.530	0.389	#315	0.396	0.488	#285
11	64	6.015	0.475	#87	6.105	0.863	#88
	128	4.088	0.470	#150	3.474	0.935	#148
	256	1.232	0.440	#222	0.737	0.532	#203
	512	0.469	0.233	#296	0.378	0.301	#272
12	64	5.237	0.399	#85	5.535	0.879	#82
	128	3.484	0.516	#143	2.708	0.966	#138
	256	0.932	0.412	#206	0.589	0.431	#184
	512	0.411	0.286	#276	0.335	0.349	#249
13	64	4.301	0.438	#80	4.038	0.843	#80
	128	2.753	0.417	#139	2.306	0.744	#131
	256	0.738	0.307	#186	0.425	0.387	#147
	512	0.260	0.156	#187	0.187	0.206	#147
14	64	7.265	0.403	#92	7.577	0.878	#90
	128	5.046	0.516	#155	3.923	1.005	#150
	256	1.293	0.409	#224	0.741	0.460	#202
	512	0.473	0.197	#299	0.379	0.250	#274
15	64	5.390	0.393	#83	5.409	0.784	#82
	128	3.260	0.437	#143	2.562	0.829	#139
	256	0.902	0.314	#203	0.503	0.392	#167
	512	0.284	0.168	#205	0.209	0.213	#164

REFERENCES

[1] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[2] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[3] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge University Press Cambridge, 2012, vol. 95.

[4] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[5] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[6] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[7] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[8] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[9] E. J. Candes and T. Tao, "The dantzig selector: statistical estimation when p is much larger than n," *The Annals of Statistics*, pp. 2313–2351, 2007.

[10] ——, "Decoding by linear programming," *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[11] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[12] A. Cohen, W. Dahmen, and R. DeVore, "Compressed sensing and best k-term approximation," *Journal of the American Mathematical Society*, vol. 22, no. 1, pp. 211–231, 2009.

[13] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[14] D. L. Donoho, J. Iain, A. Maleki, and A. Montanari, "Compressed Sensing over ℓ_p-balls: Minimax mean square error," in *2011 IEEE International Symposium on Information Theory Proceedings*, 2011, pp. 129–133.

[15] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 877–905, 2008.

[16] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Processing Letters*, vol. 14, no. 10, pp. 707–710, 2007.

[17] X. Chen, F. Xu, and Y. Ye, "Lower bound theory of nonzero entries in solutions of ℓ_2–ℓ_p minimization," *SIAM Journal on Scientific Computing*, vol. 32, no. 5, pp. 2832–2852, 2010.

[18] M. J. Lai and J. Wang, "An unconstrained ℓ_q minimization with $0 < q \leq 1$ for sparse solution of underdetermined linear systems," *SIAM Journal on Optimization*, vol. 21, no. 1, pp. 82–101, 2011.

[19] Z. Xu, X. Chang, F. Xu, and H. Zhang, "$\ell_{1/2}$ regularization: A thresholding representation theory and a fast solver," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1013–1027, 2012.

[20] L. Shi, X. Huang, and J.A.K. Suykens, "Sparse kernel regression with coefficient-based ℓ_q-regularization", Internal Report 13-50, ESAT-SISTA, KU Leuven, submitted.

[21] T. Zhang, "Some sharp performance bounds for least squares regression with ℓ_1 regularization," *The Annals of Statistics*, vol. 37, no. 5A, pp. 2109–2144, 2009.

[22] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[23] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with a certain family of nonconvex penalties and DC programming," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4686–4698, 2009.

[24] R. Horst and N. Thoai, "DC programming: overview," *Journal of Optimization Theory and Applications*, vol. 103, no. 1, pp. 1–43, 1999.

[25] L. An and P. Tao, "The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems," *Annals of Operations Research*, vol. 133, no. 1, pp. 23–46, 2005.

[26] Y. Wang and W. Yin, "Sparse signal reconstruction via iterative support detection," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 462–491, 2010.

[27] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.

[28] ——, "Iteratively re-weighted least squares minimization: Proof of faster than linear rate for sparse recovery," in *the 42nd IEEE Annual Conference on Information Sciences and Systems*, 2008, pp. 26–29.

[29] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[30] G. B. Moody, R. G. Mark, and A. L. Goldberger. "PhysioNet: a web-based resource for the study of physiologic signals," *IEEE Engineering in Medicine and Biology Magazine* vol. 20, no. 3, 70–75, 2001.

[31] P. S. Addison, "Wavelet transforms and the ECG: a review," *Physiological Measurement*, vol. 26, no. 5, p. R155, 2005.

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

[33] M. Guerquin-Kern, M. Haberlin, K. Pruessmann, and M. Unser, "A fast wavelet-based reconstruction method for magnetic resonance imaging," *IEEE Transactions on Medical Imaging*, vol. 30, no. 9, pp. 1649–1660, 2011.

[34] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.