New Improved Penalty Methods for Sparse Reconstruction Based on Difference of Two Norms

2013 March, First Revision: 2013 August, Second Revision: 2014 August

Yingnan Wang

***Abstract*—Two new penalty methods for sparse reconstruction are proposed based on two types of difference of convex functions (DC for short) programming in which the DC objective functions are the difference of l_1 and l_{σ_q} norms and the difference of l_1 and l_r norms with $r > 1$. By introducing a generalized q-term shrinkage operator upon the special structure of l_{σ_q} norm, we design a proximal gradient algorithm for handling the DC l_1-l_{σ_q} model. And by employing the majorization scheme, we develop a majorized penalty algorithm for the DC l_1-l_r model. The convergence results of our new algorithms are presented as well. Extensive simulation results show that these two new algorithms offer improved signal recovery performance and require reduced computational effort relative to state-of-the-art l_1 and l_p ($p \in (0, 1)$)models.**

***Index Terms*—compressed sensing, difference of convex norms, penalty method, majorization algorithm, q-term shrinkage operator**

I. Introduction

THE sparse optimization is an essential problem for compressive sensing, signal processing, and image processing and reconstruction. The purpose of sparse optimization is to find a sparse solution with least nonzero components constrained by a linear system, i.e., the so-called l_0 minimization problem formulated as

$$(l_0) \quad \min_{x\in\mathbb{R}^n} \big\{ \|x\|_0 : Ax = b \big\}. \tag{1}$$

Here the l_0 norm $\|x\|_0$ counts the number of nonzeros. Due to the nonconvexity and noncontinuity of the l_0 norm, problem (l_0) is generally NP-hard [28]. How to approximately solve it with a "good" enough sparse solution in decent time is becoming more and more important. A remarkable strategy is the basis pursuit approach which replaces l_0 norm with l_1 norm (see, e.g. [11]–[14])

$$(l_1) \quad \min_{x\in\mathbb{R}^n} \big\{ \|x\|_1 : Ax = b \big\}. \tag{2}$$

Benefiting from the restricted isometry properties proposed by Candès and Tao [5], the l_1 relaxation model allows us to exactly recovery l_0 problem via a convex linear program with high probability for Gaussian measurements [5]–[7]. Another famous relaxation strategy that lies between the minimizations (l_0) and (l_1) is to employ the l_p $(0 < p < 1)$ norm as the surrogate for l_0 norm [17]:

$$(l_p) \quad \min_{x\in\mathbb{R}^n} \big\{ \|x\|_p : Ax = b \big\}. \tag{3}$$

The special properties of l_p norm itself ensure that minimizing l_p outperforms minimizing l_1 for sparse solution under weaker restricted isometry property and fewer measurements for smaller p [8]–[10]. A great deal of research has been conducted into l_1 and l_p problems including all kinds of variants and related algorithms, as you can see in [3] and references therein. Comparing to the convex l_1 relaxation, the non-convex problem (l_p) is generally more difficult to handle. However, it was shown in [25] that the potential reduction method can solve this special non-convex problem in polynomial time with arbitrarily given accuracy.

Besides the l_p norm surrogate for the l_0 norm, other forms of non-convex sparsity inducing penalty functions are emerging in recent literature and have received considerable attentions. Most recently, the majority of such sparsity inducing functions are unified as the notion of DC programming in [19], including log-sum [4], smoothly clipped absolute deviation (SCAD) [15], atan [26], minimax concave penalty [31], and capped-l_1 penalty [20] [32] [33]. Generally, DC programming problem can be solved through a primal-dual convex relaxations algorithm which is famous in the literature of DC Programming [24]. Such general iterative framework is usually not practical for large-scale problems. Other algorithms appeared as for solving application problems of DC programming in the area of finance and insurance, data analysis, machine learning as well as signal processing. For details, we refer to [1], [18], [23], [27] and [29]. However, among the above mentioned DC programming approaches for sparse reconstruction, most of them are mainly preserving the *separability properties* of both l_0 and l_1 norms and thus considering the approximated objective functions that can be expressed as $\sum(g(|x_i|) - h(|x_i|))$.

Unlike the separable sparsity inducing functions involved in the aforementioned DC programming for problem (l_0), we establish two specific types of DC programming with *un-separable* objective functions, which are in the form of difference functions between l_1 norm and some "$l_\sharp$" norm. By introducing a new notion "l_{σ_q}" denoting the sum of q largest elements of a vector in magnitude (i.e., the l_1 norm of q-term best approximation of a vector), the above "$l_\sharp$" norm is hereby in this paper taken as l_{σ_q} norm or the classical l_r norm with $r > 1$. Obviously l_{σ_q} and l_r $(r > 1)$ are regular convex norms.

Beijing Jiaotong University (Beijing 100044, P.R. China) & Department of Combinatorics and Optimization-University of Waterloo (Waterloo, N2L 3G1, Canada), e-mail: wynchinada@gmail.com.

The corresponding DC programs are as follows:

$$(P_{\sigma_q}) \quad \min_{x\in\mathbb{R}^n}\big\{\|x\|_1-\epsilon\|x\|_{\sigma_q} : Ax=b\big\}, \tag{4}$$

and

$$(P_r) \quad \min_{x\in\mathbb{R}^n}\big\{\|x\|_1-\epsilon\|x\|_{r} : Ax=b\big\}, \tag{5}$$

where $\epsilon\in(0,1]$, $\|x\|_{\sigma_q}$ is defined as the sum of the q largest elements of x in magnitude (default $\|x\|_{\sigma_q}=0$ if $q=0$), $q\in\{1,\cdots,n\}$ and $r>1$. Apparently, for problem (P_{σ_q}), if $q=0$, this problem is exactly problem (l_1); if $q=1$, this corresponding problem is to minimize the DC function of l_1 norm and l_∞ norm by the factor ϵ. And for problem (P_r), if $r=2$ and $\epsilon=1$, it comes to the case studied in the recent work [30].

In this paper, we will study the aforementioned two types of DC programs both from graph comparisons and theoretical analysis. Additionally, two iterative algorithms for solving problems (P_{σ_q}) and (P_r) as well as the convergence results will be established. Especially, for choices of q in model (P_{σ_q}), we will take a scheme of "double" continuation and warm start skills along with the reduction process of penalty factors. Such strategy is proved to be with high efficiency by simulated tests. Moreover, the computational results demonstrate that both of the DC approaches of l_1-ϵl_{σ_q} model and l_1-ϵl_r model are efficient and competitive in the aspects of sparsity and accuracy compared to l_1 model and l_p model. Especially for noise free case, model (P_{σ_q}) (even with $q=1$) can always receive perfect recovery with almost 100% success rate as well as model (P_r). And for noisy case, our models can return sparse solutions with much smaller noise error $\|Ax-b\|_2$ and exactness error $\|x-x_{\rm orig}\|_\infty$ than l_1 model and l_p model. Owing to the higher accuracy and less computing cost, our models can be successfully applied to imaging reconstruction and signal recovery.

The organization of this paper is as follows. In Section II we introduce the model of (P_{σ_q}) by giving the recovery analysis and presenting a proximal gradient algorithm framework with generalized q-term shrinkage operator in each subproblem (PGGS) including its convergence result. In Section III, we give a straightforward idea of how (P_r) model works by setting up a new bound theory about l_1 norm and l_r norm with $r>1$, and then we propose a majorization algorithm framework and establish the convergence analysis. Computational results are reported in Section IV in three parts: (P_{σ_q}) models with different integers q; (P_r) models with different values of $r>1$; Overall comparison with existing methods of l_1 model and l_p model ($0<p<1$). Conclusions will be drawn in Section V and selected proofs will be stated in the last section.

II. Difference of l_1 and l_{σ_q} norms

The DC programming (P_{σ_q}) will be considered in this section. Similar to l_1 norm, l_2 norm, etc., we adopt the notation l_{σ_q} to denote the norm $\|x\|_{\sigma_q}$ which is defined as in (4). After studying on exact recovery, we design an iterative shrinkage algorithm involved a generalized q-term shrinkage operator based on proximal gradient algorithm framework, where the q-term shrinkage operator utilizes the special structure of l_{σ_q} norm.

A. Exact recovery analysis

The effectiveness of the relaxation model (P_{σ_q}) for the original problem (l_0) will be analyzed in this subsection. Firstly, we consider the case of $q=1$. In this case, problem (P_{σ_q}) is reduced to

$$(P_{\sigma_1}) \quad \min_{x\in\mathbb{R}^n}\big\{\|x\|_1-\epsilon\|x\|_{\infty} : Ax=b\big\}, \quad \epsilon\in(0,1]. \tag{6}$$

To illustrate how possibly model (P_{σ_q}) works better than model (l_1), we first draw the unit balls of $\|x\|_1$ and $\|x\|_1-\epsilon\|x\|_\infty$ in two-dimensional case as shown in Figure 1.

Fig. 1. Level sets: l_1 v.s. l_{σ_1}

Actually, it is not difficult to verify that for general $n-$dimensional vector space, $\|x\|_1-\epsilon\|x\|_\infty$ is the union of n numbers of bounded level sets of weighted l_1 norm: $(1-\epsilon)|x_1|+|x_2|+\cdots+|x_n|,\cdots,|x_1|+|x_2|+\cdots+(1-\epsilon)|x_n|$. This also implies that problem (P_{σ_1}) can be decomposed into n numbers of l_1 problems. The following example shows that with some special data sets (A,b), (P_{σ_1}) can exactly and uniquely recover problem (l_0) while (l_1) can not.

Example 2.1: $A=\begin{bmatrix}3&0&1\\0&3&1\end{bmatrix}$, $b=\begin{bmatrix}1\\1\end{bmatrix}$, $r>2$.

$(l_0):$ optimal solution $x^{(0)}=(0,0,1)^T, \|x^{(0)}\|_0=1$

$(P_1):$ optimal solution $x^{(1)}=(\frac{1}{3},\frac{1}{3},0)^T, \|x^{(1)}\|_0=2$

$(P_{\sigma_1}):$ optimal solution $x^{(\sim)}=(0,0,1)^T$ if $\epsilon\in(\frac{1}{2},1)$.

Motivated by the above example, we can find a class of data sets that allows for the exact recovery of problem (P_{σ_1}) but not of (l_1).

Theorem 2.2: Let $(A,b)\in\mathbb{R}^{m\times(m+1)}\times\mathbb{R}^m, rank(A)=m$. If $A=(B,N)$ with invertible $B\in\mathbb{R}^{m\times m}$ and $N\in\mathbb{R}^m$ satisfying

$$\frac{(B^{-1}N)_i}{(B^{-1}b)_i}=\frac{1}{k}, \quad \|B^{-1}N\|_1<1, \quad k\neq 0,$$

and

$$(B^{-1}b)_i\neq 0, \quad (B^{-1}N)_i\neq 0, \quad i=1\cdots m,$$

then for any

$$1\geq\epsilon>\frac{1-\|B^{-1}N\|_1}{1-\|B^{-1}N\|_\infty}, \tag{7}$$

problem (P_{σ_1}) exactly and uniquely solves (l_0) but problem P_1 can not.

Secondly, we will further consider the general case of integer q. we will show a clear fact with exact recovery equivalence between problem (l_0) and (P_{σ_q}) for general integer q. Motivated by the idea of difference of l_1 and l_∞ approach, it is natural to further consider minimizing difference of l_1 and l_{σ_q} in order to get more sparser solution for l_0 problem. To demonstrate this, let us take a look at the following graphs of $\|x\|_1-\|x\|_{\sigma_1}$ and $\|x\|_1-\|x\|_{\sigma_2}$ in three dimensions as shown in Figures 2 and 3.

Fig. 2. $n = 3,\ \|x\|_1 - \|x\|_{\sigma_1} \le 0.6$

Fig. 3. $n = 3,\ \|x\|_1 - \|x\|_{\sigma_2} \le 0.2$

The exact recovery result is presented in the following theorem.

Theorem 2.3: Let $\epsilon = 1$. Assume that x^0 is the unique solution of (l_0), and $|supp(x^0)| = q$. Then problem (P_{σ_q}) uniquely and exactly recovers x^0.

Proof: Since $|supp(x^0)| = q$, $\|x^0\|_1$=$\|x^0\|_{\sigma_q}$. For any x satisfying $Ax = b$, we have

$$\|x\|_1 - \|x\|_{\sigma_q} \ge 0 = \|x^0\|_1 - \|x^0\|_{\sigma_q}.$$

This means x^0 is a solution of (P_{σ_q}). Moreover, if there exists some distinct $x^1 \ne x^0$ that also solves (P_{σ_q}), then we have

$$\|x^1\|_1 - \|x^1\|_{\sigma_q} = \|x^0\|_1 - \|x^0\|_{\sigma_q} = 0,$$

which implies $\|x^1\|_0 \le q$. Therefore, x^1 is also an optimal solution to (l_0). This contradicts the assumption. Thus, x^0 is the unique solution of problem (P_{σ_q}). ■

Actually, in practical computation, such assumptions of $\epsilon = 1$ or $q = |supp(x^0)|$ can be relaxed to a smaller value due to the computational results in Section IV.

B. Algorithm framework and convergence result

Let us consider the least squares regularization for problem (4) as follows:

$$\min_x \quad f_1(x) := \frac{1}{2}\|Ax - b\|^2 + \mu(\|x\|_1 - \epsilon\|x\|_{\sigma_q}), \qquad (8)$$

where $\mu > 0$, $\epsilon \in (0, 1)$. Notice that the restriction on ϵ guarantees that $f_1(x) \ge 0$ for all x. To solve (8), we consider the following standard proximal gradient algorithm:

1. **Start:** Let x^0 be given. Set $L > \lambda_{\max}(A^TA)$ with $\lambda_{\max}$ the maximal eigenvalue.
2. **For** $t = 0, 1, \ldots$, find

$$x^{t+1} \in \operatorname*{Arg\,min}_x \Big\{ \langle A^T(Ax^t - b), x - x^t\rangle + \frac{L}{2}\|x - x^t\|^2 + \mu(\|x\|_1 - \epsilon\|x\|_{\sigma_q}) \Big\}. \qquad (9)$$

Note that subproblem (9) can equivalently formulated as

$$\min_x \left\{ \frac{L}{2} \left\| x - \left(x^t - \frac{1}{L}A^T(Ax^t - b) \right) \right\|^2 + \mu(\|x\|_1 - \epsilon\|x\|_{\sigma_q}) \right\}.$$

Thus, it suffices to consider the solutions to the following optimization problem

$$\min_x \frac{1}{2}\|x - y\|^2 + \lambda_1\|x\|_1 - \lambda_2\|x\|_{\sigma_q}, \qquad (10)$$

with a given vector y and positive numbers $\lambda_1 > \lambda_2 > 0$. An explicit solution of this problem is given in the following proposition.

Proposition 2.4: Let $\{i_1, \ldots, i_n\}$ be indices such that

$$|y_{i_1}| \ge |y_{i_2}| \ge |y_{i_3}| \ge \cdots \ge |y_{i_n}|.$$

Then $x^* := \phi(y)$ with

$$x_i^* = \begin{cases} \operatorname{sign}(y_i)\max\{|y_i| - (\lambda_1 - \lambda_2), 0\} & \text{if } i = i_1, i_2, \ldots, i_q, \\ \operatorname{sign}(y_i)\max\{|y_i| - \lambda_1, 0\} & \text{otherwise.} \end{cases} \qquad (11)$$

is a solution to (10).

The operator ϕ as defined in Proposition 2.4 is actually a generalized version of shrinkage function, and hence we call it the ***generalized q-term shrinkage operator***. Subsequently, we briefly call the following summarized Proximal Gradient algorithm with this generalized q-term Shrinkage operator as PGGS.

PGGS algorithm:

1. **Start:** Let x^0 be given. Set $L > \lambda_{\max}(A^TA)$.
2. **For** $t = 0, 1, \ldots$,

$$y^{t+1} = \left(x^t - \frac{1}{L}A^T(Ax^t - b) \right),$$

Sort y^{t+1} as $|y_{i_1}| \ge |y_{i_2}| \ge |y_{i_3}| \ge \cdots \ge |y_{i_n}|$,

$$x^{t+1} = \begin{cases} \operatorname{sign}(y_i)\max\{|y_i| - \mu(1-\epsilon), 0\} & \text{if } i = i_l, \\ \operatorname{sign}(y_i)\max\{|y_i| - \mu, 0\} & \text{otherwise.} \end{cases}$$

(where $l = 1, \cdots, q$)

End (for)

We establish the convergence result for PGGS as follows.

Theorem 2.5: The sequence $\{x^t\}$ generated by the above PGGS algorithm converges to a stationary point of problem (8).

C. Choice of q: continuation skill

Similar to the frequently used continuation scheme for parameter μ which is run from a cold start, we start with a large value of μ and then decrease μ until the desired value is reached. At each value, the solution obtained with the previous values is used as current initialization. Here, for choice of q, we also take a continuation scheme as μ's changing:

1. Initialize $q_0 = 1$, $\mu = \mu_0$.
2. For $i = 0, \cdots, l$, solving (P_{σ_q}) with $\mu = \mu_i$, let $x^{(i)}$ be the optimal solution. Set q_{i+1} to be the estimated value from the sparsity of previous solution $x^{(i)}$:
   $q_{i+1} = size(find(abs(x^{(i)}) > Tol_q))$, $Tol_q > 0$ is given.

This is kind of "double" continuation scheme both for q and μ. The computational results in Section IV-C will show the efficiency of such a strategy.

III. Difference of l_1 and l_r norms ($r > 1$)

A natural question follows the above idea that using difference of two norms to approximate l_0 norm, which is, how would it be if we extend to a general form: difference of l_1 norm and l_r norm with $r > 1$? It is known that

$$1 \le \frac{\|x\|_1}{\|x\|_2} \le \sqrt{\|x\|_0}, \qquad \forall x \in \mathbb{R}^n,\ x \ne 0.$$

Roughly speaking, the ratio of $\|\cdot\|_1$ and $\|\cdot\|_2$ is small for sparse vectors that have many zero (or near-zero) elements. In

general, by Hölder inequality,

$$1 \le \frac{\|x\|_1}{\|x\|_r} \le \|x\|_0^{1-\frac{1}{r}}, \quad \forall x \in \mathbb{R}^n,\ x \neq 0.$$

If r decreases to 1, then the right hand side $\|x\|_0^{1-\frac{1}{r}}$ decreases to 1 too, which means the fraction $\frac{\|x\|_1}{\|x\|_r}$ is bounded tighter by $\|x\|_0$ while r close to one. Thus, minimizing the difference between $\|\cdot\|_1$ and $\|\cdot\|_r$ is a natural way to obtain a sparse solution as possible as it could.

This also can be observed throughout the following comparison between graphs for $\|x\|_1 - \|x\|_r$ with different values of r ($r > 1$) and $\|x\|_p$ ($0 < p < 1$), from which we can get some straightforward idea that $\|x\|_1 - \|x\|_r$ relatively better approximates to the cardinality function $\|x\|_0$.

Fig. 4. Comparison between l_p and DC l_1-l_r

A. Bound theory for difference of l_1 and l_r norms ($r > 1$)

Motivated by the most recent bound result on l_1 norm and l_p pseudo-norm ($0 < p < 1$) of Lemma II.2 in [34], it is not difficult to establish the following bound result on l_1 norm and l_r norm ($r > 1$).

Theorem 3.1: For $r > 1$ and $0 \neq x \in \mathbb{R}^n$ with $k = \|x\|_0$. Let $\|x\|_{-\infty} := \min\{|x_i| > 0\}$. We have

$$k^{1-\frac{1}{r}} \ge \frac{\|x\|_1}{\|x\|_r} \ge k^{1-\frac{1}{r}} - \frac{\tau_r k(\|x\|_\infty - \|x\|_{-\infty})}{\|x\|_r}, \tag{12}$$

where

$$\tau_r := r^{\frac{r}{1-r}} - r^{\frac{1}{1-r}}. \tag{13}$$

Moreover, τ_r is a non-increasing and convex function of $r \in (1, +\infty)$ with

$$\tau_1 := \lim_{r \to 1^+} \tau_r = 0,\ \tau_\infty := \lim_{r \to +\infty} \tau_r = 1.$$

We present the graph of function τ_r in the following figure.

Fig. 5. Graph of function τ_r respects to $r \in (1, 20]$

Remark 3.2: Theorem 3.1 gives us a good explanation about the relationship between $\frac{\|x\|_1}{\|x\|_r}$ and $\|x\|_0$. It is consistent with the fact as demonstrated in the above figure that the closer of r to 1 the better approximation of $\|x\|_1 - \|x\|_r$ to $\|x\|_0$.

B. Algorithm framework

We will propose a majorized penalty algorithm for solving the following least-squares variant of (5):

$$\min_{x \in \mathbb{R}^n} f_2(x) := \big\{\frac{1}{2}\|Ax - b\|_2^2 + \mu(\|x\|_1 - \epsilon\|x\|_r)\big\}, \quad \mu > 0, \tag{14}$$

where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $\mu > 0$, $1 > \epsilon > 0$. Note that the restriction on ϵ guarantees that $f_2(x) \ge 0$ for all x. We describe an algorithm for solving (14), based on the majorized penalty approach introduced in [18]. We start by constructing a majorization of f_2. Let $L := \lambda_{\max}(A^T A)$. Then for any x, $y \in \mathbb{R}^n$, we have

$$\frac{1}{2}\|Ax - b\|_2^2 \le \frac{1}{2}\|Ay - b\|_2^2 + \langle A^T(Ay - b), x - y\rangle + \frac{L}{2}\|x - y\|_2^2.$$

Moreover, using convexity of norm $\|x\|_r$, we see that for any x, $y \in \mathbb{R}^n$,

$$\|x\|_r \ge \|y\|_r + \langle g(y), x - y\rangle, \quad g(y) \in \partial\|y\|_r,$$

where

$$[g(y)]_i = \begin{cases} \frac{\operatorname{sign}(y_i)|y_i|^{r-1}}{\|y\|_r^{r-1}} & \text{if } y \neq 0, \\ 0 & \text{otherwise,} \end{cases}$$

Hence, if we define

$$\begin{aligned} F(x, y) :=& \frac{1}{2}\|Ay - b\|_2^2 + \langle A^T(Ay - b), x - y\rangle + \frac{L}{2}\|x - y\|_2^2 \\ &+ \mu(\|x\|_1 - \epsilon\|y\|_r - \epsilon\langle g(y), x - y\rangle), \end{aligned}$$

then for any $x, y \in \mathbb{R}^n$, we have

$$F(x, y) \ge f_2(x) \ \text{ and } \ F(y, y) = f_2(y),$$

which means that F is a majorization of f_2. Using this majorization function, we next describe the majorization approach for solving (14).

Starting with an initial iterate x^0, the majorized penalty approach updates x^t by solving

$$x^{t+1} = \arg\min_x F(x, x^t). \tag{15}$$

Using the special form of F, we can obtain an explicit formula for x^{t+1} in terms of the shrinkage operator as follows:

$$\begin{aligned} x^{t+1} =& \arg\min_x \Big\{\langle A^T(Ax^t - b), x - x^t\rangle + \frac{L}{2}\|x - x^t\|_2^2 \\ &+ \mu(\|x\|_1 - \epsilon\langle g(x^t), x - x^t\rangle)\Big\} \\ =& \arg\min_x \left\{\frac{L}{2}\left\|x - x^t + \frac{1}{L}[A^T(Ax^t - b) - \mu\epsilon g(x^t)]\right\|_2^2 \right. \\ &\left. + \mu\|x\|_1\right\} \\ =& \operatorname{sgn}(v^t) \circ \max\left\{|v^t| - \frac{\mu}{L}, 0\right\}, \end{aligned} \tag{16}$$

where

$$v^t := x^t - \frac{1}{L}[A^T(Ax^t - b) - \mu\epsilon g(x^t)].$$

We summarize the algorithm framework in the following table.

Majorized penalty algorithm (MajorP for short):

1. **Start:** Let x^0 be given. Set $L > \lambda_{\max}(A^T A)$.
2. **For** $t = 0, 1, \ldots$, find

$$\begin{aligned} x^{t+1} =& \arg\min_x \Big\{\langle A^T(Ax^t - b), x - x^t\rangle + \frac{L}{2}\|x - x^t\|_2^2 \\ &+ \mu(\|x\|_1 - \epsilon\langle g(x^t), x - x^t\rangle)\Big\} \\ =& \operatorname{sgn}\left(x^t - \frac{1}{L}[A^T(Ax^t - b) - \mu\epsilon g(x^t)]\right) \\ &\circ \max\left\{|v^t| - \frac{\mu}{L}, 0\right\}, \end{aligned} \tag{17}$$

End (for)

It is interesting to note that, in view of (16), the above algorithm reduces to the proximal gradient algorithm as applied to (14) when $\epsilon = 0$.

C. Convergence result

Actually, it has been mentioned in the recent paper of Sun and Gao [18] that, their majorized penalty method and the related results can be suitably adapted to establish the above algorithm. Notice that the convergence result in [18, Theorem 3.4] concerning the majorization approach does not directly apply to our algorithm because our majorization function is nonsmooth in x. We present the convergence result as follows.

Theorem 3.3: Let $\{x^t\}$ be generated according to (17). Then

$$\frac{L}{2}\|x^t - x^{t+1}\|_2^2 \leq f_2(x^t) - f_2(x^{t+1}). \tag{18}$$

Furthermore, the sequence $\{x^t\}$ is bounded, and any cluster point is a stationary point of (14).

IV. Numerical experiments

In this section, the numerical experiments are presented to demonstrate the performance of utilizing difference of norms penalty approach to sparse signal reconstruction, including comparisons with existing models (l_0) and (l_p). Before proceeding to the computational results, we need to define some notations and data sets. For convenience, we use the notations: DC l_1-l_∞, DC l_1-l_{σ_q}, and DC l_1-l_r, to represent the DC Programs (P_{σ_1}), (P_{σ_q}), and (P_r).

For each data set, the matrix A and the vector b are randomly generated by the following matlab codes:

$$\begin{aligned} &x_{\rm orig} = \text{zeros}(n,1), \qquad y = \text{randperm}(n),\\ &A = \text{randn}(m,n), \qquad A = \text{orth}(A')',\\ &x_{\rm orig}(y(1:k)) = 2 * \text{randn}(k,1),\\ &b = A * x_{\rm orig} + \sigma * \cdot\text{randn}(m,1). \end{aligned}$$

The parameter σ will be zero for noiseless case study, otherwise it is always taken as $\sigma = 0.01$. The same A and b would then be used in each algorithm when comparing the corresponding given set of models. Both for noise case and noiseless case, we set the tolerance of estimated sparsity from previous iteration used in PGGS algorithm for model (P_{σ_q}) as $Tol_q =$ 1e-4, and all the ϵ in corresponding models are set as $\epsilon = 0.9$. The stopping criteria for PGGS and MajorP are given by

$$\frac{\|x^{(k+1)} - x^{(k)}\|_2}{\max\{1, \|x\|_2\}} \leq \text{1e-4}\mu.$$

In the overall comparison subsection, we will use P(success) to denote the successfully recovered probability where we call $x_{\rm orig}$ is successfully recovered by solution x if

$$\|x - x_{\rm orig}\|_\infty \leq Tol_{suc}.$$

For noiseless case, $Tol_{suc} =$ 1e-3; for noise case, $Tol_{suc} =$ 5e-1.

A. Results for DC l_1-l_{σ_q} with different $1 \leq q \leq k$

Let $n = 256$, $m = 64$, $k = 10$. For each fixed q from 1 to 10, randomly generating 20 samples and applying PGGS algorithm proposed in Section II-B to problem (P_{σ_q}), we can see from the following computational results that the sparsity of solutions of problem (P_{σ_q}) decreases to k as q increases to k. In Table I, each row corresponds to each sample and when $q = 0$, (P_{σ_q}) is exactly (l_1) problem. Table II shows that the error (first row) and cputime (the second row) are also decreasing when q is getting closer to k.

TABLE I
DC l_1-l_{σ_q} WITH DIFFERENT q: SPARSITY WITHOUT NOISE

q	0	1	2	3	4	5	6	7	8	9	10
1	5	12	13	10	10	10	10	10	10	10	10
2	10	10	10	10	10	10	10	10	10	10	10
3	10	10	10	10	10	10	10	10	10	10	10
4	12	10	10	10	10	10	10	10	10	10	10
5	4	11	10	10	10	10	10	10	10	10	10
6	12	11	10	10	10	10	10	10	10	10	10
7	14	10	10	10	10	10	10	10	10	10	10
8	25	13	11	11	11	10	10	10	10	10	10
9	11	10	10	10	10	10	10	10	10	10	10
10	13	11	10	11	10	10	10	10	10	10	10
11	14	11	11	11	11	10	10	10	10	10	10
12	14	12	10	10	10	10	10	10	10	10	10
13	11	10	10	10	10	10	10	10	10	10	10
14	10	10	10	10	10	10	10	10	10	10	10
15	10	10	10	10	10	11	10	10	10	10	10
16	15	11	11	12	10	10	10	10	10	10	10
17	10	10	10	10	10	10	10	10	10	10	10
18	11	11	10	10	10	10	10	10	10	10	10
19	14	10	10	10	10	10	10	10	10	10	10
20	10	10	10	10	10	10	10	10	10	10	10

TABLE II
DC l_1-l_r: AVERAGE CPU TIME AND ERROR WITHOUT NOISE

$q=0$	1	2	3	4	5
0.0063	0.0002	0.0002	0.0002	0.0002	0.0002
0.8812	0.0695	0.0633	0.0555	0.0484	0.0414
	6	7	8	9	10
	0.0002	0.0001	0.0001	0.0001	0.0001
	0.0430	0.0383	0.0359	0.0297	0.0250

B. Results for DC l_1-l_r ($r > 1$) with different r

Firstly, we compare the tests results for $n = 256 : 256 : 2048$, $m = n/4$, $k = floor(m/6)$, Sample: 20. Corresponding results are presented in Figures 6 and 7. All the tests get perfect recovery.

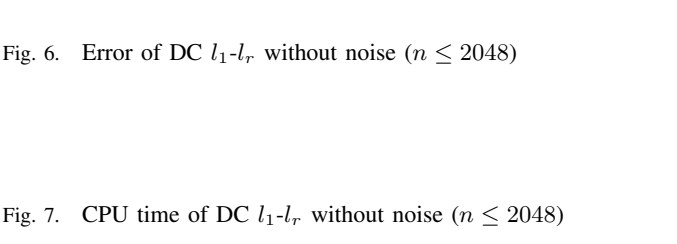

Fig. 6. Error of DC l_1-l_r without noise ($n \leq 2048$)

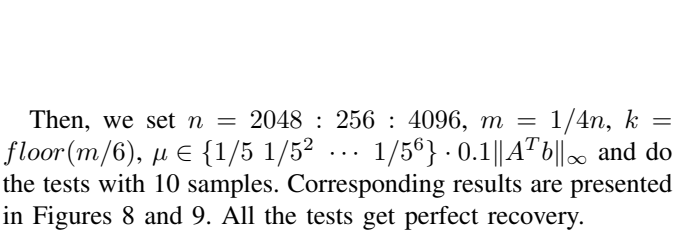

Fig. 7. CPU time of DC l_1-l_r without noise ($n \leq 2048$)

Then, we set $n = 2048 : 256 : 4096$, $m = 1/4n$, $k = floor(m/6)$, $\mu \in \{1/5\ 1/5^2\ \cdots\ 1/5^6\} \cdot 0.1\|A^T b\|_\infty$ and do the tests with 10 samples. Corresponding results are presented in Figures 8 and 9. All the tests get perfect recovery.

TABLE III
DC l_1-l_r: AVERAGE CPU TIME WITHOUT NOISE ($n \leq 2048$)

r	1.2	1.5	1.8	2	3
n	CPU Time				
256	0.064	0.060	0.056	0.055	0.060
512	0.120	0.115	0.136	0.115	0.141
768	0.187	0.192	0.231	0.182	0.238
1024	0.289	0.314	0.377	0.313	0.395
1280	0.563	0.627	0.723	0.674	0.796
1536	0.809	0.915	1.061	1.009	1.185
1792	0.952	1.089	1.250	1.186	1.361
2048	1.239	1.414	1.595	1.529	1.777

Observed from Figures 6 to 9 and Table III, Table IV, we found an interesting fact that unlike the case of $n < 1024$ where DC l_1-l_2 always spent the least CPU time among five models, when $n \geq 1024$, the CPU time of DC l_1-$l_{1.2}$ model is at least 25% less than DC l_1-l_2 model and is always the fastest one among the five models. Model of DC l_1-$l_{1.8}$ is almost the slowest one all over the tests. And DC l_1-$l_{1.8}$ model is between model DC l_1-$l_{1.8}$ and model DC l_1-l_2 in CPU time. For errors $\|Ax-b\|_2^2$ and $\|x-x_{\text{orig}}\|_\infty$, they are always decreasing when r is getting closer to 1.

Fig. 8. Error of DC l_1-l_r without noise ($n \geq 2048$)

Fig. 9. CPU time of DC l_1-l_r without noise ($n \geq 2048$)

TABLE IV
DC l_1-l_r: AVERAGE CPU TIME WITHOUT NOISE ($n \geq 2048$)

r	1.2	1.5	1.8	2	3
n	CPU Time				
2304	1.529	1.770	2.046	1.987	2.314
2560	1.740	2.010	2.317	2.240	2.517
2816	2.067	2.353	2.753	2.670	3.032
3072	2.496	2.831	3.225	3.153	3.578
3328	3.054	3.259	3.698	3.629	4.078
3584	3.314	3.923	4.481	4.371	4.929
3840	3.553	4.279	4.832	4.729	5.343
4096	3.940	4.631	5.254	5.090	5.718

C. Overall comparison

In this subsection, we compare the performance of five approximation models for sparse optimization l_0 problem and aim at confirming the evidence that the difference of two norms approaches are performing better than l_1 and l_p penalties ($0 < p < 1$). The five models include l_1 model, l_p model ($0 < p < 1$), DC l_1-l_∞, (P_{σ_q}) (with continuation scheme for choice of q), and DC l_1-$l_{1.2}$.

1) Existing algorithms for l_1 and l_p: We will briefly introduce one of existing popular algorithms for l_1 problem, and two algorithms for l_p problem.

Gradient projection sparse reconstruction algorithm (GPSR) for l_1 problem proposed by Figueiredo, Nowak and Wright [16] to l_1 problem: $\min_x \ \frac{1}{2}\|Ax-b\|_2^2 + \mu\|x\|_1$. We use the matlab file *GPSR_BB* downloaded from *http://www.lx.it.pt/ mtf/GPSR/* with options: 'STOPCRITERION',2, 'verbose',0, 'Continuation',1, 'ContinuationSteps',-1.

Iteratively reweighted least square algorithm (IRLS): noise free case presented in [10] by Chartrand and Yin for l_p problem with linear equalities: $\min_x \left\{\|x\|_p^p : Ax = b\right\}$, where they replace the l_p objective function by a weighted l_2 norm: $\min_x \left\{\sum_i w_i x_i^2 : Ax = b\right\}$. The weights are computed from the previous $(k-1)$th iterate $w_i = ((x_i^{(k-1)})^2 + \epsilon)^{p/2-1}$, and the solution is explicitly given by $x^{(k)} = Diag(w)^{-1}A^T(ADiag(w)^{-1}A^T)^{-1}b$. In our computations, we take $p = 0.5$ and use the same strategy suggested in [10] for how to repeatedly reduce ϵ from 1 to 1e-8.

Iteratively reweighted l_1 minimization algorithm (IRL1): with noisy case proposed in [4] by Candès, Wakin and Boyd for solving l_p problem: $\min_x \left\{\frac{1}{2}\|Ax-b\|^2 + \mu\|x\|_p^p\right\}$, where they replace the above l_p objective function by a weighted l_1 norm: $\min_x \left\{\frac{1}{2}\|Ax-b\|^2 + \mu\|Diag(w)x\|_1\right\}$. The weights are computed from the previous $(k-1)$th iterate by $w_i = (x_i^{(k-1)} + \epsilon)^{p-1}$. We will take $p = 0.5$, fix $\epsilon = 0.1$, and run a number of 4 reweighting iterations. In each weighted l_1 problem, we implemented the GPSR matlab codes as mentioned before with $\mu = 1/5^2 \cdot \|(ADiag(w)^{-1})^T b\|_\infty$.

2) Comparison of performance: For convenience, we number the algorithms used in this section as in Table V and list the choices of μ for continuation process by algorithms GPSR, PGGS, MajorP in Table VI.

TABLE V
NOTATION NUMBERS FOR ALGORITHMS

(1) GPSR solving problem (l_1)
(2) PGGS solving DC l_1-l_∞
(3) PGGS solving DC l_1-l_{σ_q}
(4) MajorP solving DC l_1-l_2
(5) IRLS solving l_p for noiseless case and IRL1 for noise case

TABLE VI
CHOICES FOR THE LIST OF VALUES USED FOR μ

	Without Noise	With Noise
GPSR	$1/5^5 \cdot 0.1\|A^Tb\|_\infty$	$0.1\|A^Tb\|_\infty$
PGGS	$[1,\ 1/5,\ \cdots,\ 1/5^5] \cdot 0.1\|A^Tb\|_\infty$	$[1,\ 0.3] \cdot 0.1\|A^Tb\|_\infty$
MajorP	$[1,\ 1/5,\ \cdots,\ 1/5^5] \cdot 0.1\|A^Tb\|_\infty$	$[1,\ 0.3] \cdot 0.1\|A^Tb\|_\infty$

Results of sparsity for noiseless case and noise case are presented in Figure 10 and Figure 11 respectively, where red horizontal line is the true value of sparsity, blue '+' is the estimation value in each instance. The results of error, CPU time and P(success) are presented in Table VII (without noise) and Table VIII (with noise). In order to give more precise comparison of sparsity for noise case, we compute the l_∞ norm error and l_2 norm error between the computed sparsity and the true sparsity k (based on 100 samples) and demonstrate them in Figure 3 and Table IX respectively.

TABLE VII
AVERAGE ERROR-P(SUCCESS)-CPU TIME WITHOUT NOISE

	(1)	(2)	(3)	(4)	(5)
n			Error $\|Ax-b\|_2$		
256	3.5e-4	2.1e-4	2e-5	1.7e-4	0.0
512	5.6e-4	3.5e-4	4e-5	3.0e-4	0.0
1024	9.4e-4	5.8e-4	6e-5	5.2e-4	0.0
2048	1.4e-3	8.6e-4	9e-5	7.9e-4	0.0
n			Error $\|x-x_{\text{orig}}\|_\infty$		
256	4.1e-3	2.4e-4	2e-5	2.0e-4	4.9e-4
512	2.2e-3	3.1e-4	3e-5	2.6e-4	7.8e-4
1024	4.4e-3	3.9e-4	4e-5	3.3e-4	3.5e-4
2048	2.5e-3	4.3e-4	4e-5	3.9e-4	3.7e-4
n			P(success)		
256	0.45	1.00	1.00	1.00	0.80
512	0.35	1.00	1.00	1.00	0.70
1024	0.10	1.00	1.00	1.00	0.95
2048	0.00	1.00	1.00	1.00	0.95
n			CPU Time		
256	0.030	0.096	0.043	0.038	0.261
512	0.034	0.152	0.075	0.071	1.237
1024	0.079	0.478	0.214	0.245	8.909
2048	0.236	1.687	0.796	1.156	81.668

Fig. 10. Sparsity without noise

TABLE VIII
AVERAGE ERROR-P(SUCCESS)-CPU TIME WITH NOISE

	(1)	(2)	(3)	(4)	(5)
n			Error $\|Ax-b\|_2$		
256	0.60	0.19	0.09	0.16	1.94
512	1.04	0.34	0.14	0.29	2.91
1024	1.60	0.53	0.24	0.48	4.32
2048	2.45	0.82	0.34	0.76	6.03
n			Error $\|x-x_{\text{orig}}\|_\infty$		
256	0.32	0.16	0.102	0.121	1.24
512	0.52	0.19	0.047	0.142	2.17
1024	0.31	0.10	0.045	0.075	1.69
2048	0.69	0.23	0.193	0.192	2.09
n			P(success)		
256	0.20	0.98	0.99	0.99	0.00
512	0.02	0.98	1.00	1.00	0.00
1024	0.00	0.98	1.00	1.00	0.00
2048	0.00	0.90	1.00	0.97	0.00
n			CPU Time		
256	0.010	0.029	0.020	0.015	0.148
512	0.014	0.045	0.035	0.026	0.195
1024	0.027	0.094	0.079	0.064	0.446
2048	0.058	0.348	0.288	0.267	1.445

Fig. 11. Sparsity with noise

Fig. 12. Error $\|x-x_{\text{orig}}\|_\infty$ with noise

From all the above experimental results, we have the following observations.

TABLE IX
AVERAGE SPARSITY l_2 ERROR WITH NOISE

	(1)	(2)	(3)	(4)	(5)
n		Sparsity	l_2	Error	
256	25.6	44.8	14.2	18.1	17.6
512	41.3	80.2	24.0	31.6	36.9
1024	73.5	106.9	34.5	48.9	81.7
2048	147.3	200.7	69.7	93.8	174.7

1) For noise-free case, among the five different approaches, we clearly know that:

- **a)** The GPSR algorithm for solving l_1 problem is the fastest but returns the worst sparsity (Figure 10), larger stable error $\|Ax-b\|_2$ and exactness error $\|x-x_{\text{orig}}\|_\infty$ than the three models proposed by this paper, and the lowest P(success).
- **b)** The IRLS algorithm for solving l_p problem always has "0" error because of its design for keeping the equality of the constraints. Meanwhile, to pay for the price, it runs a lot cputime. Moreover, the solution have a long "tail" with small elements as shown in the last line of Figure 10.
- **c)** The three new approaches of DC l_1-l_∞, DC l_1-l_{σ_q}, DC l_1-l_2, always returned with 100% success-recovery and smallest errors. **What the outstanding fact is that, DC l_1-l_{σ_q} approach always can obtain a sparsest solution with small errors (1e-5) both for stable error $\|Ax-b\|_2$ and exactness error $\|x-x_{\text{orig}}\|_\infty$.**

2) When with noise, the summary is similar to the noise-free case in some aspects:

- **a')** The GPSR algorithm for solving l_1 problem is still the fastest but returns relatively larger errors $\|Ax-b\|_2$ and $\|x-x_{\text{orig}}\|_\infty$ (Figure 11, Figure 12), and low probability of success-recovery.
- **b')** The IRL1 algorithm for solving l_p problem with noise has biggest errors, and poorest P(success) rate.
- **c')** The performance between the three new approaches of DC l_1-l_∞, DC l_1-l_{σ_q}, DC l_1-l_2 are enlarged with each other compared to noiseless case. Although they all returned smallest error $\|x-x_{\text{orig}}\|_\infty$. Model DC l_1-l_∞ shows weak robustness compared to DC l_1-l_{σ_q} and DC l_1-l_2.

Above all, our computational results suggest that either with noise or not, the approaches of DC l_1-l_{σ_q} solved by PGGS algorithm, and DC l_1-l_2 solved by MajorP, are very efficient and competitive methods for sparse reconstruction problem. By using them, we can get sparser solutions with good quality in terms of exactness and stability in very decent time. Moreover, DC l_1-l_{σ_q} approach performance relative better than DC l_1-l_2 in aspects of errors. However, DC l_1-l_2 approach runs less CPU time than DC l_1-l_{σ_q} when $n \leq 2048$.

D. Simulated signal test

In this subsection, we fix $n = 1024$, $k = 42$ and x_{orig}. Generate the data set A and b without noise by different values of m: $m = 1/4n$, $1/5n$ and $1/(5.5)n$. As Figures 13 to

15 demonstrate, our new models can get exact recovery with fewer measurements of observations.

Fig. 13. Simulated signal test without noise $m = 1/4n$

Fig. 14. Simulated signal test without noise $m = \lfloor 1/5n \rfloor$

Fig. 15. Simulated signal test without noise $m = \lfloor 1/(5.5)n \rfloor$

V. Conclusion

We have introduced two penalty methods for solving sparse signal reconstruction problems, of which the objective functions are in a common form of DC functions. The DC objective functions considered in this paper are specified to the difference of two norms, such as difference of l_1 and l_∞, difference of l_1 and l_{σ_q}, and difference of l_1 and l_r ($r > 1$). Computational results demonstrated that our DC approach of l_1-l_{σ_q} is efficient to obtain a sparse solution with less error than l_1 and l_p models. Moreover, the DC approach of l_1-l_r with $r > 1$ also can recover the original sparse signals even with fewer measurements of observations. From the overall experimental comparisons with l_1 model and l_p model, we conclude that our methods of minimizing differences of norms are competitive for sparse reconstruction problems, such as image reconstruction and signal recovery, no matter in sparsity or in accuracy.

VI. Proofs

Proof of Theorem 2.2
Let $\eta \in \mathbb{R}^m$, $\zeta \in \mathbb{R}^m$ with

$$\eta_i = (B^{-1}b)_i,\ \zeta_i = (B^{-1}N)_i,\ i = 1, \cdots, m,$$

and $x^1 := (0, \cdots, 0, k)$, $x^2 := (\eta_1, \cdots, \eta_m, 0)$. Then x^1, x^2 are solutions of $Ax = b$. By assumption, we have

$$\eta = k\zeta,\ \|\zeta\|_1 < 1, \quad \|\eta\|_1 < k.$$

Since $(B^{-1}b)_i \neq 0$ for each i and note that for any $\theta \neq 0$,

$$(I, B^{-1}N)(0, \cdots, \theta, \cdots, 0, 0)^T = (0, \cdots, \theta, \cdots, 0, 0)^T \neq B^{-1}b.$$

Then we conclude that x^1 is the unique solution of (l_0). On the other hand, problem (l_1) can be reduced to

$$\begin{aligned}
&\min_{z\in\mathbb{R}} \ |z| + \|B^{-1}b - B^{-1}Nz\|_1 \\
= &\min_{z\in\mathbb{R}} \ |z| + \sum_i |\eta_i - \zeta_i z| \\
= &\min_{z\in\mathbb{R}} \ |z| + \|\zeta\|_1 \cdot |k - z| \\
\geq &\min_{z\in\mathbb{R}} \ (1 - \|\zeta\|_1)|z| + k\|\zeta_i\|_1 \\
\geq &\ k\|\zeta\|_1 = \|\eta\|_1,
\end{aligned}$$

where the $\geq$ is by triangular inequality and can be attained at $z = 0$. Then, the optimal solution attained at $z^* = 0$, and hence $x^* = (B^{-1}b, 0) = (\eta, 0)$, i.e., x^2 solves problem (l_1). Therefore, we conclude that problem (l_1) doesn't solve (l_0). Next, we show that under condition (7), problem P_{σ_1} uniquely solves (l_0). We need to show x^1 is the unique solution of problem P_{σ_1}.

Firstly, problem problem P_{σ_1} is equivalent to

$$\begin{aligned}
&\min_x \big\{ \|x\|_1 - \epsilon\|x\|\infty : x = (\eta - z\cdot\zeta, z), z \in \mathbb{R} \big\} \\
= &\min_x \big\{ \|x\|_1 - \epsilon\|x\|\infty : x = ((k - z)\zeta, z), z \in \mathbb{R} \big\} \\
= &\min_z \big\{ |z| + \|\zeta\|_1 \cdot |k - z| - \epsilon \max\{|z|, \|\zeta\|_\infty \cdot |k - z|\} \big\}.
\end{aligned}$$

The above function has four break points

$$\left\{ \frac{k\|\zeta\|_\infty}{\|\zeta\|_\infty - 1}, \frac{k\|\zeta\|_\infty}{\|\zeta\|_\infty + 1}, 0, k \right\}.$$

In order to show $x^2 = (0, \cdots, 0, k)$ is the unique minimization solution, we only need to show

$$\begin{aligned}
(-\|\zeta\|_1 - 1 + \epsilon)\tfrac{k\|\zeta\|_\infty}{\|\zeta\|_\infty - 1} + k\|\zeta\|_1 &> (1 - \epsilon)k \\
-\epsilon\|\zeta\|_\infty + \|\zeta\|_1 &> (1 - \epsilon)k \\
(-\|\zeta\|_1 + 1 - \epsilon)\tfrac{k\|\zeta\|_\infty}{\|\zeta\|_\infty + 1} + k\|\zeta\|_1 &> (1 - \epsilon)k.
\end{aligned}$$

It is not difficult to get

$$(-\|\zeta\|_1 - 1 + \epsilon)\frac{k\|\zeta\|_\infty}{\|\zeta\|_\infty - 1} + k\|\zeta\|_1 > -\epsilon\|\zeta\|_\infty + \|\zeta\|_1$$

and

$$(-\|\zeta\|_1 + 1 - \epsilon)\frac{k\|\zeta\|_\infty}{\|\zeta\|_\infty + 1} + k\|\zeta\|_1 > -\epsilon\|\zeta\|_\infty + \|\zeta\|_1,$$

which are by $(-\|\zeta\|_1 - 1 + \epsilon)\frac{k\|\zeta\|_\infty}{\|\zeta\|_\infty - 1} > 0$, $(-\|\zeta\|_1 + 1)\frac{k\|\zeta\|_\infty}{\|\zeta\|_\infty + 1} > 0$ and $0 < \epsilon\frac{k\|\zeta\|_\infty}{\|\zeta\|_\infty + 1} < \epsilon\|\zeta\|_\infty$. Until now, only $-\epsilon\|\zeta\|_\infty + \|\zeta\|_1 > (1 - \epsilon)k$ left to be shown, which is precisely

$$\epsilon > \frac{k - k\|\zeta\|_1}{k - k\|\zeta\|_\infty} = \frac{1 - \|\zeta\|_1}{1 - \|\zeta\|_\infty}.$$

This completes the proof. □

Proof of Proposition 2.4
Notice that (10) can be equivalently formulated as

$$\begin{aligned}
&\min_{|\mathcal{S}|=k} \min_x \frac{1}{2}\|x - y\|^2 + \lambda_1\|x\|_1 - \lambda_2 \sum_{i\in\mathcal{S}} |x_i| \\
= &\min_{|\mathcal{S}|=k} \min_x \frac{1}{2}\|x - y\|^2 + \lambda_1 \sum_{i\notin\mathcal{S}} |x_i| + (\lambda_1 - \lambda_2) \sum_{i\in\mathcal{S}} |x_i|.
\end{aligned} \tag{19}$$

Moreover, note also that for each i and $\xi > 0$, we have

$$\begin{aligned} & \min_{x_i} \frac{1}{2}(x_i - y_i)^2 + \xi|x_i| \\ = \quad & P_\xi(|y_i|) := \begin{cases} \frac{1}{2}|y_i|^2 & \text{if } |y_i| \le \xi, \\ \xi|y_i| - \frac{1}{2}\xi^2 & \text{otherwise,} \end{cases} \end{aligned} \tag{20}$$

with the minimum attained at $x_i = \mathrm{sign}(y_i)\max\{|y_i| - \xi, 0\}$. Substituting (20) into (19), we see that the right hand side of (19) can be further rewritten as

$$\begin{aligned} & \min_{|\mathcal{S}|=k} \sum_{i \notin \mathcal{S}} P_{\lambda_1}(|y_i|) + \sum_{i \in \mathcal{S}} P_{\lambda_1 - \lambda_2}(|y_i|) \\ = \quad & \sum_{i=1}^{n} P_{\lambda_1}(|y_i|) - \max_{|\mathcal{S}|=k} \sum_{i \in \mathcal{S}} [P_{\lambda_1}(|y_i|) - P_{\lambda_1 - \lambda_2}(|y_i|)]. \end{aligned} \tag{21}$$

Since $\lambda_1 > \lambda_1 - \lambda_2 > 0$, it is routine to show from definition that $u \mapsto P_{\lambda_1}(u) - P_{\lambda_1 - \lambda_2}(u)$ is an increasing function for positive values of u. Thus, the maximum in (21) is attained by choosing $\mathcal{S} = \{i_1, \ldots, i_k\}$. The formula (11) now follows from this and the fact that the minimum in (20) is attained at $x_i = \mathrm{sign}(y_i)\max\{|y_i| - \xi, 0\}$. □

Proof of Theorem 2.5

We note first that the algorithm is monotone, i.e., $f_1(x^{t+1}) \le f_1(x^t)$ for $t \ge 0$. Indeed, from the definition of L and Taylor's inequality, we have

$$\begin{aligned} f_1(x^{t+1}) \le & \frac{1}{2}\|Ax^t - b\|^2 + \langle A^T(Ax^t - b), x^{t+1} - x^t \rangle \\ & + \frac{L}{2}\|x^{t+1} - x^t\|^2 + \mu(\|x^{t+1}\|_1 - \epsilon\|x^{t+1}\|_{\sigma_q}. \end{aligned}$$

Combining this with the definition of x^{t+1}, we see immediately that $f_1(x^{t+1}) \le f_1(x^t)$. As a further consequence, we note that

$$\mu(1-\epsilon)\|x^t\|_1 \le \mu(\|x^t\|_1 - \epsilon\|x^t\|_{\sigma_q}) \le f_1(x^t) \le f_1(x^0).$$

Since $\mu(1-\epsilon) > 0$, we see that $\{x^t\}$ is bounded. On the other hand, the objective function $f_1(x)$ of (8) is a square term plus a piecewise linear function which ensures that $f_1(x)$ is semi-algebraic and hence it satisfies KL inequality (for definitions of semi-algebraic and KL inequality please refer to [2]) and the references therein). Thus, from [2, Theorem 5.1], we conclude that the sequence $\{x^t\}$ is convergent to a stationary point of (8) (for more details please see the appendix of Explanation for proof of Theorem 2.5). □

Proof of Theorem 3.1

We only need to establish the right side of the inequalities since the left side is already know by Hölder inequality. Assume $x = (z, 0)$ with $z \in \mathbb{R}^k$ and $\|z\|_0 = k$. Without loss of generality, we only need to prove the case $z \in \Omega := \{(z_1, z_2, \cdots, z_k) \ne 0 \mid z_1 \ge z_2 \ge \cdots \ge z_k \ge 0\}$ due to the symmetry of components $|z_1|, |z_2|, \cdots, |z_k|$. Clearly, $z_1 \ne 0$. Notice that if the inequality (12) holds for any $(1, z_2, \cdots, z_k) \in \Omega$, then we can immediately generalize the conclusion to all $z \in \Omega$ through substituting $z/z_1, z \in \Omega$ into (12) and eliminating the common factor $1/z_1$. Henceforth, it remains to show

$$\|z\|_1 \ge \frac{\|z\|_r}{k^{1/r-1}} - \tau_r k(1 - z_k), \tag{22}$$

with $z \in \{(1, z_2, \cdots, z_k) \mid 1 \ge z_2 \ge \cdots \ge z_k \ge 0\}$, where τ_r is a function of r specified in (13).

First, for any given $r \in (0, 1]$ define that

$$h(z) := k^{1-1/r}\|z\|_r - \|z\|_1.$$

It is easy to verify that $h(z)$ is a convex function on $\mathbb{R}^k_+$. Since the maximum of a convex function always arrives on the boundary, we have

$$\begin{aligned} h(z_k): \quad &= \max_{1 \ge z_2 \ge z_3 \ge \cdots \ge z_k} h(1, z_2, z_3, \cdots, z_k) \\ &= h(1, \cdots, 1, z_k, \cdots, z_k), \quad z_k \in [0, 1] \end{aligned}$$

Letting the distribution of 1 appear for t times ($1 \le t \le k$) in the maximum solution of h, we have

$$h(z_k) = t(1 - z_k) + kx_k - \frac{(t(1 - z_k^r) + kx_k^r)^{1/r}}{k^{1/r-1}}.$$

By the convexity of h and $h(1) = 0$, it follows that

$$h(z_k) \le (1 - z_k)h(0) + z_n h(1) = (1 - z_k)h(0).$$

Then it holds that

$$\begin{aligned} h(z) \quad &\le \quad h(z_k) \le (1 - z_k)h(0) \\ &= \quad (1 - z_k)(k^{1-1/r}t^{1/r} - t) \\ &\le \quad (1 - z_k) \max_{r \in \{1, 2, \cdots, k\}} \{k^{1-1/r}t^{1/r} - t\} \\ &\le \quad (1 - z_k) \max_{0 < t \le k} \{k^{1-1/r}t^{1/r}\} \\ &= \quad (1 - z_k)\tau_r k, \end{aligned}$$

where τ_r is defined as (13) and the last equality holds when $r_1 = r\frac{r}{1-r}k \in (0, k]$ for any $r \in (0, 1]$.

By computing the first and second order partial derivatives of τ_r on r, it is easy to verify that τ_r is a non-increasing convex function of $r \in (0, 1]$ and

$$\lim_{r \to 1^+} \tau_r = 0 \text{ and } \lim_{r \to +\infty} \tau_r = 1.$$

Thus the proof is completed. □

Proof of Theorem 3.3

First, note that z^{t+1} minimizes $F(x, x^t)$. Then the first-order optimality condition gives

$$0 \in A^T(Ax^t - b) + L(x^{t+1} - x^t) + \mu\partial\|x^{t+1}\|_1 - \mu\epsilon g(x^t), \tag{23}$$

where g is defined in Section III-B. Using this and the definition of subdifferential, we obtain immediately that

$$\begin{aligned} & \mu\|x^t\|_1 - \mu\|x^{t+1}\|_1 \\ \ge & \langle -A^T(Ax^t - b) - L(x^{t+1} - x^t) + \mu\epsilon g(x^t), x^t - x^{t+1} \rangle \\ = & \langle -A^T(Ax^t - b) + \mu\epsilon g(x^t), x^t - x^{t+1} \rangle + L\|x^{t+1} - x^t\|_2^2 \\ = & \langle A^T(Ax^t - b) - \mu\epsilon g(x^t), x^{t+1} - x^t \rangle + L\|x^{t+1} - x^t\|_2^2. \end{aligned}$$

Hence,

$$\mu\|x^{t+1}\|_1 - \mu\|x^t\|_1 + \langle A^T(Ax^t - b) - \mu\epsilon g(x^t), x^{t+1} - x^t\rangle) \leq -L\|x^{t+1} - x^t\|_2^2.$$

Using this relation together with the definition of F, we obtain for any $t \geq 1$ that

$$\begin{aligned} & f_2(x^{t+1}) - f_2(x^t) \leq F(x^{t+1}, x^t) - f_2(x^t) \\ &= \langle A^T(Ax^t - b), x^{t+1} - x^t\rangle + \frac{L}{2}\|x^{t+1} - x^t\|_2^2 \\ &\quad + \mu(\|x^{t+1}\|_1 - \|x^t\|_1 - \epsilon\langle g(x^t), x^{t+1} - x^t\rangle) \\ &= \frac{L}{2}\|x^{t+1} - x^t\|_2^2 + \mu\|x^{t+1}\|_1 - \mu\|x^t\|_1 \\ &\quad + \langle A^T(Ax^t - b) - \mu\epsilon g(x^t), x^{t+1} - x^t\rangle) \\ &\leq \frac{L}{2}\|x^{t+1} - x^t\|_2^2 - L\|x^{t+1} - x^t\|_2^2 \leq -\frac{L}{2}\|x^{t+1} - x^t\|_2^2, \end{aligned}$$

which immediately gives (18).

We next show that $\{x^t\}$ is bounded. To this end, notice from (18) that $\{f_2(x^t)\}$ is a decreasing sequence. Hence, in particular, we have $f_2(x^t) \leq f_2(x^0)$, which implies

$$\begin{aligned} \mu(\|x^t\|_1 - \epsilon\|x^t\|) &\leq \frac{1}{2}\|Ax^t - b\|_2^2 + \mu(\|x^t\|_1 - \epsilon\|x^t\|_r) \\ &= f_2(x^t) \leq f_2(x^0). \end{aligned}$$

Since $\|x^t\|_1 \geq \|x^t\|_r$, it follows from the above relation that

$$\mu(1 - \epsilon)\|x^t\|_r \leq f_2(x^0).$$

This implies that $\{x^t\}$ is bounded since $\epsilon < 1$.

Finally, we show that any cluster point of $\{x^t\}$ is a stationary point of (14). Let x^* be a cluster point of $\{x^t\}$. Then there exists a subsequence $\{x^{t_j}\}$ such that $\lim_{j\to\infty} x^{t_j} = x^*$. On the other hand, by summing (18) from $t = 0$ to ∞, we obtain that

$$\frac{L}{2}\sum_{t=0}^{\infty}\|x^{t+1} - x^t\|_2^2 \leq f_2(x^0) - \lim_{t\to\infty} f_2(x^t) \leq f_2(x^0).$$

This implies that

$$\lim_{j\to\infty} x^{t_j} = \lim_{j\to\infty} x^{t_j+1} = x^*.$$

Using this last relation, the conclusion now follows immediately by taking limit in (23) along the subsequence $\{x^{t_j}\}$ and making use of the upper semicontinuity of (Clarke) subdifferentials. □

ACKNOWLEDGMENT

The work was supported in part by the National Basic Research Program of China (2010CB732501), and the National Natural Science Foundation of China (11301022). The authors would like to give their thanks to Ting Kei Pong for his helpful discussion about the majorized penalty algorithm and the generalized q-term shrinkage function. The authors also would like to thank Ziyan Luo, Xiang Lilan Zhang, Shenglong Zhou for his suggestion about the quality of this paper.

REFERENCES

[1] A. Alvarado, G. Scutari and J. S. Pang, A new decomposition method for multiuser DC-programming and its applications, *IEEE Transactions on Signal Processing*, vol.62, no.11, 2014.

[2] H. Attouch, J. Bolte, and B.F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods, *Math. Program., Ser. A*, vol. 137, pp. 91-129, 2013.

[3] A.M. Bruckstein, D.L. Donoho, M. Elad, From sparse solutions of systems of equations to sparse modeling of signals and images, *SIAM Rev.*, Vol. 51, pp. 34-81, 2009.

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

[5] E.J. Candès, T. Tao, Decoding by linear programming, *IEEE Trans. Inf. Theory*, vol. 51, pp. 4203-4215, 2005.

[6] E.J. Candès, J. Romberg, T. Tao, Exact signal reconstruction from highly incomplete frequency information, *IEEE Trans. Inf. Theory*, vol. 52, pp. 489-509, 2006.

[7] E.J. Candès, T. Tao, Near optimal signal recovery from random projections: universal encoding strategies?, *IEEE Trans. Inf. Theory*, vol. 52, pp. 5406-5425, 2006.

[8] R. Chartrand, Exact reconstructions of sparse signals via nonconvex minimization, *IEEE Signal Process. Lett.*, vol. 14, pp. 707-710, 2007.

[9] R. Chartrand, V. Staneva, Restricted isometry properties and nonconvex compressive sensing, *Inverse Problems*, vol. 24, pp. 1-14, 2008.

[10] R. Chartrand, W. Yin, Iteratively reweighted algorithms for compressive sensing, *in 33rd International conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3869-3872, 2008.

[11] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM J. Sci. Comput.*, vol. 20, pp. 33-61, 1998.

[12] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM Rev.*, vol. 43, pp. 129-159, 2001.

[13] D.L. Donoho, M. Elad, On the stability of the basis pursuit in the presence of noise, *Signal Process.*, vol. 86, pp. 511-532, 2006.

[14] D.L. Donoho, Compressed sensing, *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289-1306, 2006.

[15] J. Fan, R. Li, Variable selection via nonconcave pe- nalized likelihood and its oracle properties, *Journal of the American Statistical Association*, vol. 96(456), pp. 1348-1360, 2001.

[16] M.A.T. Figueiredo, R.D. Nowak, S.J. Wright, Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems, *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, pp. 586-597, 2007.

[17] S. Foucart, M.J. Lai, Sparsest solutions of underdeter- mined linear systems via l_q-minimization for $0 < q \leq 1$, *Appl. and Comput. Harmon. Anal.*, vol. 26(3), pp. 395-407, 2009.

[18] Y. Gao, D.F. Sun, A majorized penalty approach for calibrating rank constrained correlation matrix problems. Technical Report, Department of Mathematics, National University of Singapore, March 2010.

[19] G. Gasso, A. Rakotomamonjy, S. Canu, Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *IEEE Trans. Signal Process.*, vol. 57(12), pp. 4686-4698, 2009.

[20] P. Gong, J. Ye, C. Zhang, Multi-stage multi-task feature learning, In NIPS, pp. 1997-2005, 2012.

[21] P. Gong, J. Ye, C. Zhang, Robust multi-task feature learning, In SIGKDD, pp. 895-903, 2012.

[22] P. Gong, CH. Zhang, Z. Lu, J.Z. Huang, J. Ye, A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems, 2013.

[23] W. Guan, A. Gray, Sparse high-dimensional fractional-norm support vector machine via DC programming, *Computational Statistics and Data Analysis*, vol. 67, pp. 136-148, 2013.

[24] R. Horst, N.V. Thoai, Dc programming: overview, *Journal of Optimization Theory and Applications*, vol. 103, pp. 1-41, 1999.

[25] S. Ji, K.-F. Sze, Z. Zhou, A.M.-C. So, Y. Ye, Beyond convex relaxation: A polynomial-time non-convex optimization approach to network localization, *In: To Appear in the Proceedings of the 32nd IEEE International Conference on Computer Communications (INFOCOM 2013)*, Torino, 2013.

[26] N. Mourad, J.P. Reilly, Minimizing nonconvex functions for sparse vector reconstruction, *IEEE Trans. Signal Process.*, vol. 58, pp. 3485-3496, 2010.

[27] L.B. Montefusco, D. Lazzaro, S. Papi, A fast algorightm for nonconvex approaches to sparse recovery problems, *Signal Processing*, vol. 93, pp. 2636-2647, 2013.

[28] B.K. Natarajan, Sparse approximate solutions to linear systems, *SIAM J. Computing*, vol.24, pp. 227-234, 1995.
[29] Y. She, An iterative algorithm for fitting nonconvex penalized generalized linear models with grouped predictors, *Computational Statistics and Data Analysis*, vol. 56, pp. 2976-2990, 2012.
[30] P. Yin, Y. Lou, Q. He, J. Xin, Minimization of $l_{1?2}$ for compressed sensing, Technical report, 2014 ftp.math.ucla.edu/pub/camreport/cam14-01.pdf
[31] C.H. Zhang, Nearly unbiased variable selection under min- imax concave penalty, *The Annals of Statistics*, vol. 38, pp. 894-942, 2010.
[32] T. Zhang, Analysis of multi-stage convex relaxation for sparse regularization, *Journal of Machine Learning Research*, vol. 11, pp. 1081-1107, 2010.
[33] T. Zhang, Multi-stage convex relaxation for feature selection. Bernoulli, 2012.
[34] S. Zhou, L. Kong, Z. Luo, N. Xiu, New RIC Bounds via l_q-minimization with $0 < q \leq 1$ in Compressed Sensing, Beijing Jiaotong University, 2013.

Yingnan Wang is a research assistant in Department of Applied Mathematics, Beijing Jiaotong University. She received her PhD degree in Operations Research from Beijing Jiaotong University in 2011. From 2011 to 2013, she was a Post-Doctoral Fellow of Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Canada. Her research interests are in sparse optimization, non-smooth optimization and analysis and robust optimization.