COMMUN. MATH. SCI.
Vol. 15, No. 2, pp. 511–537

MINIMIZATION OF TRANSFORMED L_1 PENALTY: CLOSED FORM REPRESENTATION AND ITERATIVE THRESHOLDING ALGORITHMS*

SHUAI ZHANG† AND JACK XIN‡

Abstract. The transformed l_1 penalty (TL1) functions are a one parameter family of bilinear transformations composed with the absolute value function. When acting on vectors, the TL1 penalty interpolates l_0 and l_1 similar to l_p norm, where p is in $(0,1)$. In our companion paper, we showed that TL1 is a robust sparsity promoting penalty in compressed sensing (CS) problems for a broad range of incoherent and coherent sensing matrices. Here we develop an explicit fixed point representation for the TL1 regularized minimization problem. The TL1 thresholding functions are in closed form for all parameter values. In contrast, the l_p thresholding functions (p is in $[0,1]$) are in closed form only for $p=0,1,1/2,2/3$, known as hard, soft, half, and 2/3 thresholding respectively. The TL1 threshold values differ in subcritical (supercritical) parameter regime where the TL1 threshold functions are continuous (discontinuous) similar to soft-thresholding (half-thresholding) functions. We propose TL1 iterative thresholding algorithms and compare them with hard and half thresholding algorithms in CS test problems. For both incoherent and coherent sensing matrices, a proposed TL1 iterative thresholding algorithm with adaptive subcritical and supercritical thresholds (TL1IT-s1 for short), consistently performs the best in sparse signal recovery with and without measurement noise.

Keywords. Transformed l_1 penalty, closed form thresholding functions, iterative thresholding algorithms, compressed sensing, robust recovery.

AMS subject classifications. 94A12, 94A15.

1. Introduction

Iterative thresholding (IT) algorithms merit our attention in high dimensional settings due to their simplicity, speed, and low computational costs. In compressed sensing (CS) problems [4, 10] under l_p sparsity penalty ($p\in[0,1]$), the corresponding thresholding functions are in closed form when $p=0,\frac{1}{2},\frac{2}{3},1$. The l_1 algorithm is known as soft-thresholding [8, 9], and the l_0 algorithm hard-thresholding [1, 2]. IT algorithms only involve scalar thresholding and matrix multiplication. We note that the linearized Bregman algorithm [21, 22] is similar for solving the constrained l_1 minimization (basis pursuit) problem. Recently, half and $\frac{2}{3}$-thesholding algorithms have been actively studied [7, 18] as non-convex alternatives to improve on l_1 (convex relaxation) and l_0 algorithms.

However, the non-convex l_p penalties ($p\in(0,1)$) are non-Lipschitz. There are also some Lipschitz continuous non-convex sparse penalties, including the difference of l_1 and l_2 norms (DL12) [11, 14, 20], and the transformed l_1 (TL1) [25]. When applied to CS problems, the difference of convex function algorithms (DCA) of DL12 are found to perform the best for highly coherent sensing matrices. In contrast, the DCAs of TL1 are the most robust (consistently ranked in the top among existing algorithms) for coherent and incoherent sensing matrices alike.

In this paper, as companion of [25], we develop robust and effective IT algorithms for TL1 regularized minimization with evaluation on CS test problems. The TL1 penalty is a one parameter family of bilinear transformations composed with the absolute value

*Received: September 25, 2015; accepted (in revised form): July 12, 2016. Communicated by Wotao Yin.

The work was partially supported by NSF grants DMS-0928427, DMS-1222507 and DMS-1522383.

†Department of Mathematics, University of California, Irvine, CA, 92697, USA (szhang3@uci.edu); Phone: (949)-824-5309. Fax: (949)-824-7993.

‡Department of Mathematics, University of California, Irvine, CA, 92697, USA (jxin@math.uci.edu).

function. The TL1 parameter, denoted by letter 'a', plays a similar role as p for l_p penalty. If 'a' is small (large), TL1 behaves like l_0 (l_1). If 'a' is near 1, TL1 is similar to $l_{1/2}$. However, a strikingly different phenomenon is that the TL1 thresholding function is in *closed form for all values of parameter 'a'*. Moreover, we found subcritical and supercritical parameter regimes of TL1 thresholding functions with thresholds expressed in different formulas. The subcritical TL1 thresholding functions are continuous, similar to the soft-thresholding (a.k.a. shrink) function of l_1 (Lasso). The supercritical TL1 thresholding functions have jump discontinuities, similar to $l_{1/2}$ or $l_{2/3}$.

Several common non-convex penalties in statistics are SCAD [12], MCP [24], log penalty [5, 16], and capped l_1 [27]. We refer to Mazumder, Friedman and Hastie's paper [16] for an overview. They appeared in the univariate regularization problem

$$\min_x\{\ \frac{1}{2}(x-y)^2+\lambda P(x)\ \},$$

and produced closed form thresholding formulas. TL1 is a smooth version of capped l_1 [27]. SCAD and MCP, corresponding to quadratic spline functions with one and two knots, have continuous thresholding functions. Log penalty and capped l_1 have discontinuous threshold functions. The TL1 thresholding function is unique in that it can be either continuous or discontinuous depending on parameters 'a' and λ. Also similar to SCAD, TL1 satisfies unbiasedness, sparsity and continuity conditions, which are desirable properties for variable selection [12, 15].

The solutions of TL1 regularized minimization problem satisfy a fixed point representation involving matrix multiplication and thresholding only. Direct fixed point iterative (DFA), semi-adaptive (TL1IT-s1) and adaptive iterative schemes (TL1IT-s2) are proposed. The semi-adaptive scheme (TL1IT-s1) updates the sparsity regularization parameter λ based on the sparsity estimate of the solution. The adaptive scheme (TL1IT-s2) also updates the TL1 parameter 'a', however only doing the subcritical thresholding.

We carried out extensive sparse signal recovery experiments in Section 5, with three algorithms: TL1IT-s1, Hard and Half-thresholding methods. For Gaussian sensing matrices with positive covariance, TL1IT-s1 leads the pack and half-thresholding is the second. For coherent over-sampled discrete cosine transform (DCT) matrices, TL1IT-s1 is again the leader and with considerable margin. The half thresholding algorithm drops to the distinct last. In the presence of measurement noise, the results are similar, with TL1IT-s1 maintaining its leader status in both classes of random sensing matrices. That TL1IT-s1 fairs much better than other methods may be attributed to the two built-in thresholding values. The early iterations are observed to go between the subcritical and supercritical regimes frequently. Also TL1IT-s1 is stable and robust when exact sparsity of solution is replaced by rough estimates as long as the number of linear measurements exceeds a certain level.

The rest of the paper is organized as follows. In Section 2, we overview TL1 minimization. In Section 3, we derive TL1 thresholding functions in closed form and show their continuity properties with details of the proof left in the appendix. The analysis is elementary yet delicate, and makes use of the Cardano formula on roots of cubic polynomials and algebraic identities. The fixed point representation for the TL1 regularized optimal solution follows. In Section 4, we propose three TL1IT schemes and derive the parameter update formulas for TL1IT-s1 and TL1IT-s2 based on the thresholding functions. We analyze convergence of the fixed parameter TL1IT algorithm. In Section 5, numerical experiments on CS test problems are carried out for TL1IT-s1, hard and half thresholding algorithms on Gaussian and over-sampled DCT matrices with a broad

range of coherence. The TL1IT-s1 leads in all cases, and inherits well the robustness and effective sparsity promoting capability of TL1 [25]. Concluding remarks are in Section 6.

2. Overview of TL1 minimization

The transformed l_1 (TL1) function $\rho_a(x)$ is defined as

$$\rho_a(x) = \frac{(a+1)|x|}{a+|x|}\ , \tag{2.1}$$

where parameter $a \in (0, +\infty)$; see [15] for its unbiasedness, sparsity and continuity properties. With the change of parameter 'a', TL1 interpolates l_0 and l_1 norms:

$$\lim_{a\to 0^+} \rho_a(x) = I_{\{x\neq 0\}}, \quad \lim_{a\to+\infty} \rho_a(x) = |x|.$$

In Figure 2.1, level lines of TL1 on the plane are shown at small and large values of parameter a, resembling those of l_1 (at $a=100$), $l_{1/2}$ (at $a=1$), and l_0 (at $a=0.01$).

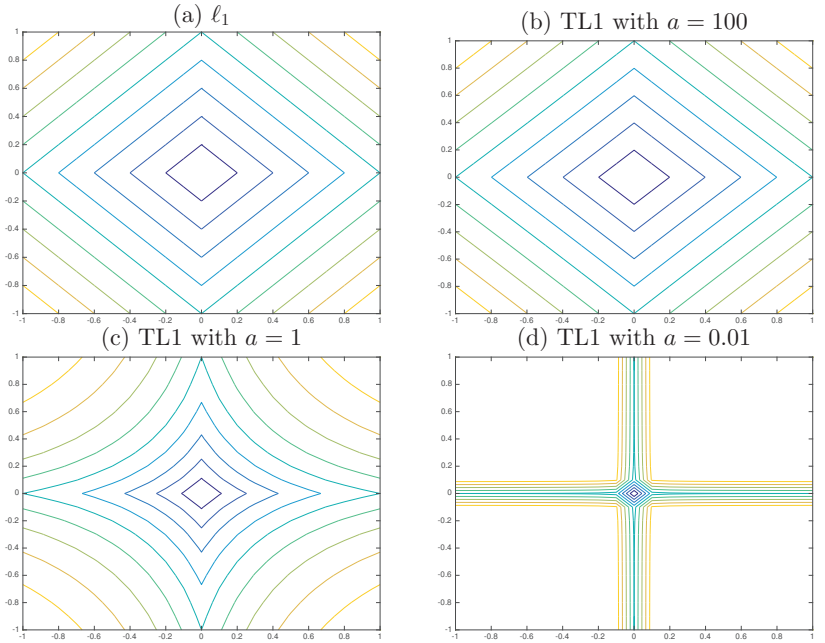

FIG. 2.1. *Level lines of TL1 with different parameters: $a=100$ (figure b), $a=1$ (figure c), $a=0.01$ (figure d). For large parameter a, the graph looks almost the same as l_1 (figure a). While for small value of a, it tends to the axis.*

Next, we want to expand the definition of TL1 to vector space. For vector $x = (x_1, x_2, \cdots, x_N)^T \in \Re^N$, we define

$$P_a(x) = \sum_{i=1}^{N} \rho_a(x_i). \tag{2.2}$$

In this paper, we will use TL1 instead of l_0 norm to solve application problems proposed from compressed sensing. The mathematical models can be generalized as

two categories: the constrained TL1 minimization:

$$\min_{x\in\Re^N} f(x) = \min_{x\in\Re^N} P_a(x) \ s.t. \ \ Ax = y, \tag{2.3}$$

and the unconstrained TL1-regularized minimization:

$$\min_{x\in\Re^N} f(x) = \min_{x\in\Re^N} \frac{1}{2}\|Ax-y\|_2^2 + \lambda P_a(x), \tag{2.4}$$

where λ is the trade-off Lagrange multiplier to control the amount of shrinkage.

The exact and stable recovery by TL1 for model (2.3) under the Restricted Isometry Property (RIP) [3, 4] conditions is established in the companion paper [25], where the difference of convex functions algorithms (DCA) for model (2.3) and (2.4) are also presented and compared with some state-of-the-art CS algorithms on sparse signal recovery problems. In paper [25], the authors find that TL1 is always among top performers in RIP and non-RIP categories alike. However, matrix multiplications and inverse operations are involved at each iteration step of TL1 DC algorithms, which increases run time and computation costs. Iterative thresholding (IT) algorithms usually are much faster, since only matrix-vector multiplications and elementwise scalar thresholding operations are needed. Also, due to precise threshold values, it needs fewer steps in IT to converge to sparse solutions. In order to reduce computation time, we shall explore thresholding property for TL1 penalty. In another paper [26], we expand TL1 thresholding and representation theories to low rank matrix completion problems via Schatten-1 quasi-norm.

3. Thresholding representation and closed-form solutions

The thresholding theories and algorithms for l_0 quasi-norm (hard-thresholding) [1,2] and l_1 norm (soft-thresholding) [8, 9] are well-known and widely tested. Recently, the closed form thresholding representation theories and algorithms for l_p $(p = 1/2, 2/3)$ regularized problems are proposed [7, 18] based on Cardano's root formula of cubic polynomials. However, these algorithms are limited to few specific values of parameter p. Here for TL1 regularization problem, we derive the closed form representation of optimal solution, under *any positive value of parameter a*.

Let us consider the unconstrained TL1 regularization model (2.4):

$$\min_x \frac{1}{2}\|Ax-y\|_2^2 + \lambda P_a(x),$$

for which the first order optimality condition is:

$$0 = A^T(Ax-y) + \lambda \cdot \nabla P_a(x). \tag{3.1}$$

Here $\nabla P_a(x) = (\partial\rho_a(x_1), ..., \partial\rho_a(x_N))$, and $\partial\rho_a(x_i) = \dfrac{a(a+1)SGN(x_i)}{(a+|x_i|)^2}$. $SGN(\cdot)$ is the set-valued signum function with $SGN(0) \in [-1, 1]$, instead of a single fixed value. In this paper, we will use $sgn(\cdot)$ to represent the standard signum function with $sgn(0) = 0$. From Equation (3.1), it is easy to get

$$x + \mu A^T(y - Ax) = x + \lambda\mu\nabla P_a(x). \tag{3.2}$$

We can rewrite the above equation, via introducing two operators

$$\begin{aligned} R_{\lambda\mu,a}(x) &= [I + \lambda\mu\nabla P_a(\cdot)]^{-1}(x), \\ B_\mu(x) &= x + \mu A^T(y - Ax). \end{aligned} \tag{3.3}$$

From Equation (3.2), we will get a representation equation for optimal solution x:

$$x = R_{\lambda\mu,a}(B_\mu(x)). \tag{3.4}$$

We will prove that the operator $R_{\lambda\mu,a}$ is diagonal under some requirements for parameters λ, μ and a. Before that, a closed form expression of proximal operator at scalar TL1 $\rho_a(\cdot)$ will be given and proved at following subsection. This optimal solution expression will be used to prove the threshold representation theorem for model (2.4).

3.1. Proximal point operator for TL1. Like [17], we introduce proximal operator $prox_{\lambda\rho_a}:\Re\to\Re$ for univariate TL1 (ρ_a) regularization problem,

$$prox_{\lambda\rho_a}(y) = arg\min_{x\in\Re}\left(\frac{1}{2}(y-x)^2+\lambda\rho_a(y)\right).$$

Proximal operator of a convex function usually intends to solve a small convex regularization problem, which often admits closed-form formula or an efficient specialized numerical methods. However, for non-convex functions, like l_p with $p\in(0,1)$, their related proximal operators do not have closed form solutions in general. There are many iterative algorithms to approximate optimal solution. But they need more computing time and sometimes only converge to local optimal or stationary point. In this subsection, we prove that for TL1 function, there indeed exists a closed-formed formula for its optimal solution.

For the convenience of our following theorems, we want to introduce three parameters:

$$\begin{cases} t_1^* = \dfrac{3}{2^{2/3}}(\lambda a(a+1))^{1/3} - a \\ t_2^* = \lambda\frac{a+1}{a} \\ t_3^* = \sqrt{2\lambda(a+1)} - \frac{a}{2}. \end{cases} \tag{3.5}$$

It can be checked that inequality $t_1^* \le t_3^* \le t_2^*$ holds. The equality is realized if $\lambda = \frac{a^2}{2(a+1)}$ (Appendix A).

LEMMA 3.1. *For different values of scalar variable x, the roots of the following two cubic polynomials in y satisfy properties:*

(1) If $x > t_1^*$, there are 3 distinct real roots of the cubic polynomial:

$$y(a+y)^2 - x(a+y)^2 + \lambda a(a+1) = 0.$$

Furthermore, the largest root y_0 is given by $y_0 = g_\lambda(x)$, where

$$g_\lambda(x) = sgn(x)\left\{\frac{2}{3}(a+|x|)cos(\frac{\varphi(x)}{3}) - \frac{2a}{3} + \frac{|x|}{3}\right\} \tag{3.6}$$

with $\varphi(x) = \arccos(1 - \frac{27\lambda a(a+1)}{2(a+|x|)^3})$, and $|g_\lambda(x)| \le |x|$.

(2) If $x < -t_1^*$, there are also 3 distinct real roots of cubic polynomial:

$$y(a-y)^2 - x(a-y)^2 - \lambda a(a+1) = 0.$$

Furthermore, the smallest root denoted by y_0, is given by $y_0 = g_\lambda(x)$.

Proof.

1.) First, we consider the roots of cubic equation:

$$y(a+y)^2 - x(a+y)^2 + \lambda a(a+1) = 0, \text{ when } x > t_1^*.$$

We apply variable substitution $\eta = y + a$ in the above equation, then it becomes

$$\eta^3 - (a+x)\eta^2 + \lambda a(a+1) = 0,$$

whose discriminant is:

$$\triangle = \lambda(a+1)a[4(a+x)^3 - 27\lambda(a+1)a].$$

Since $x \geq t^*$ and $\triangle > 0$, there are three distinct real roots for this cubic equation. Next, we change variables as $\eta = t + \frac{a}{3} + \frac{x}{3} = y + a$. The relation between y and t is: $y = t - \frac{2a}{3} + \frac{x}{3}$. In terms of t, the cubic polynomial is turned into a depressed cubic as:

$$t^3 + pt + q = 0,$$

where $p = -(a+x)^2/3$, and $q = \lambda a(a+1) - 2(a+x)^3/27$. The three roots in trigonometric form are:

$$\begin{aligned} t_0 &= \tfrac{2(a+x)}{3}\cos(\varphi/3) \\ t_1 &= \tfrac{2}{3}(a+x)\cos(\varphi/3+\pi/3) \\ t_2 &= -\tfrac{2}{3}(a+x)\cos(\pi/3-\varphi/3) \end{aligned} \tag{3.7}$$

where $\varphi = \arccos(1 - \frac{27\lambda a(a+1)}{2(a+x)^3})$.

Then $t_2 < 0$, and $t_0 > t_1 > t_2$. By the relation $y = t - \frac{2a}{3} + \frac{x}{3}$, the three roots in variable y are: $y_i = t_i - \frac{2a}{3} + \frac{x}{3}$, for $i = 1,2,3$. From these formula, we know that:

$$y_0 > y_1 > y_2.$$

Also it is easy to check that $y_0 \leq x$ and $y_2 < 0$, and the largest root $y_0 = g_\lambda(x)$, when $x > t_1^*$.

2.) Next, we discuss the roots of the cubic equation:

$$(a-y)^2 y - x(a-y)^2 - \lambda a(a+1) = 0, \text{ when } x < -t_1^*.$$

Here we set: $\eta = a - y$, and $t = \eta + \frac{x}{3} - \frac{a}{3}$. So $y = -t + \frac{x}{3} + \frac{2a}{3}$. By a similar analysis as in part (1), there are 3 distinct roots for polynomial equation: $y_0 < y_1 < y_2$ with the smallest solution

$$y_0 = -\frac{2}{3}(a-x)\cos(\varphi/3) + \frac{x}{3} + \frac{2a}{3},$$

where $\varphi = \arccos(1 - \frac{27\lambda a(a+1)}{2(a-x)^3})$. So we proved that the smallest solution is $y_0 = g_\lambda(x)$, when $x < -t_1^*$. ∎

Next let us define the function $f_{\lambda,x}(\cdot) : \Re \to \Re$,

$$f_{\lambda,x}(y) = \frac{1}{2}(y-x)^2 + \lambda\rho_a(y). \tag{3.8}$$

So $\partial f_{\lambda,x}(y) = y - x + \lambda \frac{a(a+1)SGN(y)}{(a+|y|)^2}$.

THEOREM 3.1. *The optimal solution* $y^*_\lambda(x) = arg\min_y f_{\lambda,x}(y)$ *is a threshold function with threshold value* t*:*

$$y^*_\lambda(x) = \begin{cases} 0, & |x| \le t \\ g_\lambda(x), & |x| > t \end{cases} \tag{3.9}$$

where $g_\lambda(\cdot)$ *is defined in Equation (3.6). The threshold parameter* t *depends on regularization parameter* λ*,*

(1) if $\lambda \le \frac{a^2}{2(a+1)}$ *(sub-critical),*

$$t = t^*_2 = \lambda \frac{a+1}{a};$$

(2) $\lambda > \frac{a^2}{2(a+1)}$ *(super-critical),*

$$t = t^*_3 = \sqrt{2\lambda(a+1)} - \frac{a}{2},$$

where parameters t^*_2 *and* t^*_3 *are defined in formula (3.5).*

Proof. In the following proof, we represent $y^*_\lambda(x)$ as y^* for simplicity. We split the value of x into 3 cases: $x = 0$, $x > 0$ and $x < 0$, then prove our conclusion case by case.

1.) $x = 0$.
In this case, optimization objective function is $f_{\lambda,x}(y) = \frac{1}{2}y^2 + \lambda\rho_a(y)$. Here the two factors $\frac{1}{2}y^2$ and $\lambda\rho_a(|y|)$ are both increasing for $y > 0$, and decreasing for $y < 0$. Thus $f(0)$ is the unique minimizer for function $f_{\lambda,x}(y)$. So

$$y^* = 0, \text{ when } x = 0.$$

2.) $x > 0$.
Since $\frac{1}{2}(y-x)^2$ and $\lambda\rho_a(y)$ are both decreasing for $y < 0$, our optimal solution will only be obtained at nonnegative values. Thus it just needs to consider all positive stationary points for function $f_\lambda(y)$ and also point 0.
When $y > 0$, we have:

$$f'_{\lambda,x}(y) = y - x + \lambda \frac{a(a+1)}{(a+y)^2},$$

and

$$f''_{\lambda,x}(y) = 1 - 2\lambda \frac{a(a+1)}{(a+y)^3}.$$

Since $f''_{\lambda,x}(y)$ is increasing, $f''_{\lambda,x}(0) = 2\lambda\frac{(a+1)}{a^2}$ determines the convexity for the function $f(y)$. In the following proof, we further discuss the value of y^* by two conditions: $\lambda \le \frac{a^2}{2(a+1)}$ and $\lambda > \frac{a^2}{2(a+1)}$.

2.1) $\lambda \le \frac{a^2}{2(a+1)}$.
So we have $\inf_{y>0} f''_\lambda(y) = f''_\lambda(0+) = 1 - 2\lambda\frac{(a+1)}{a^2} \ge 0$, which means function $f'_\lambda(y)$ is increasing for $y \ge 0$, with minimum value $f'_\lambda(0) = \lambda\frac{(a+1)}{a} - x = t^*_2 - x$.

i) When $0 \le x \le t_2^*$, $f'_{\lambda,x}(y)$ is always positive, thus the optimal value $y^* = 0$.

ii) When $x > t_2^*$, $f'_{\lambda,x}(y)$ is first negative then positive. Also $x \ge t_2^* \ge t_1^*$. The unique positive stationary point y^* of $f_{\lambda,x}(y)$ satisfies equation: $f'_\lambda(y^*) = 0$, which implies

$$y(a+y)^2 - x(a+y)^2 + \lambda a(a+1) = 0. \tag{3.10}$$

According to Lemma 3.1, the optimal value $y^* = y_0 = g_\lambda(x)$.
Above all, the value for y^* is:

$$y^* = \begin{cases} 0, & 0 \le x \le t_2^*; \\ g_\lambda(x), & x > t_2^* \end{cases} \tag{3.11}$$

under the condition $\lambda \le \frac{a^2}{2(a+1)}$.

2.2) $\lambda > \frac{a^2}{2(a+1)}$.
In this case, due to the sign of $f''_\lambda(y)$, we know that function $f'_{\lambda,x}(y)$ is decreasing at first then switches to be increasing at the domain $[0,\infty)$. Its minimum obtained at point $\overline{y} = (2\lambda a(a+1))^{1/3} - a$ and

$$f'_\lambda(\overline{y}) = \frac{3}{2^{2/3}}(\lambda(a+1)a)^{1/3} - a - x = t_1 - x.$$

Thus $f'_\lambda(y) \ge t_1^* - x$, for $y \ge 0$.
i) When $0 \le x \le t_1^*$, function $f_\lambda(y)$ is always increasing. Thus optimal value $y^* = 0$.
ii) When $t_2^* \le x$, $f'_\lambda(0+) \le 0$. So function $f_\lambda(y)$ is decreasing first, then increasing. There is only one positive stationary point, which is also the optimal solution. Using Lemma 3.1, we know that $y^* = g_\lambda(x)$.
iii) When $t_1^* < x < t_2^*$, $f'_\lambda(0+) > 0$. Thus function $f_\lambda(y)$ is first increasing, then decreasing and finally increasing, which implies that there are two positive stationary points and the larger one is a local minima. Using Lemma 3.1 again, the local minimize point will be $y_0 = g_\lambda(x)$, the largest root of Equation (3.10). But we still need to compare $f_\lambda(0)$ and $f_\lambda(y_0)$ to distinguish the global optimal y^*. Since $y_0 - x + \lambda\frac{a(a+1)}{(a+y_0)^2} = 0$, which implies $\lambda\frac{(a+1)}{a+y_0} = \frac{(x-y_0)(a+y_0)}{a}$, we have

$$\begin{aligned} f_\lambda(y_0) - f_\lambda(0) &= \frac{1}{2}y_0^2 - y_0 x + \lambda\frac{(a+1)y_0}{a+y_0} \\ &= y_0(\frac{1}{2}y_0 - x + \lambda\frac{(a+1)}{a+y_0}) \\ &= y_0(\frac{1}{2}y_0 - x + \frac{(x-y_0)(a+y_0)}{a}) \\ &= y_0^2(\frac{x-y_0}{a} - \frac{1}{2}) = y_0^2((x - g_\lambda(x))/a - 1/2). \end{aligned} \tag{3.12}$$

It can be proved that parameter t_3^* is the unique root of $t - g_\lambda(t) - \frac{a}{2} = 0$ in $[t_1^*, t_2^*]$ (see Appendix B). For $t_1^* \le t \le t_3^*$, $t - g_\lambda(t) - \frac{a}{2} \ge 0$; for $t_3^* \le t \le t_2^*$,

$t - g_\lambda(t) - \frac{a}{2} \le 0$. So in the third case: $t_1^* < x < t_2^*$: if $t_1^* < x \le t_3^*$, $y^* = 0$; if $x > t_3^*$, $y^* = y_0 = g_\lambda(x)$.

Finally we know that under the condition $\lambda > \frac{a^2}{2(a+1)}$:

$$y^* = \begin{cases} 0, & 0 \le x \le t_3^*; \\ g_\lambda(x), & x > t_3^*, \end{cases} \tag{3.13}$$

3.) $x < 0$.

Notice that

$$\inf_y f_{\lambda,x}(y) = \inf_y f_{\lambda,x}(-y) = \inf_y \ \frac{1}{2}(y - |x|))^2 + \rho_a(y),$$

so $y^*(x) = -y^*(-x)$, which implies that the formula obtained when $x > 0$ above, can extend to the case: $x < 0$ by odd symmetry. Formula (3.9) holds.

Summarizing results from all cases, the proof is complete. ∎

3.2. Optimal point representation for regularized TL1 (2.4). Next, we will show that the optimal solution of the TL1 regularized problem (2.4) can be expressed by a thresholding function. Let us introduce two auxiliary objective functions. For any given positive parameters λ, μ and vector $z \in \Re^N$, define:

$$\begin{aligned} C_\lambda(x) &= \tfrac{1}{2}\|y - Ax\|_2^2 + \lambda P_a(x) \\ C_\mu(x,z) &= \mu\left\{C_\lambda(x) - \tfrac{1}{2}\|Ax - Az\|_2^2\right\} + \tfrac{1}{2}\|x - z\|_2^2. \end{aligned} \tag{3.14}$$

The first function $C_\lambda(x)$ comes from the objective of TL1 regularization problem (2.4).

Starting from this subsection till the end of this paper, we substitute parameter λ in threshold value t_i^* with the product of λ and μ, which are

$$\begin{cases} t_1^* = \dfrac{3}{2^{2/3}}(\lambda\mu a(a+1))^{1/3} - a \\ t_2^* = \lambda\mu\frac{a+1}{a} \\ t_3^* = \sqrt{2\lambda\mu(a+1)} - \frac{a}{2}. \end{cases} \tag{3.15}$$

LEMMA 3.2. *If $x^s = (x_1^s, \cdots, x_N^s)^T$ is a minimizer of $C_\mu(x,z)$ with fixed parameters $\{\mu, a, \lambda, z\}$, then there exists a positive number $t = t_2^* I_{\left\{\lambda\mu \le \frac{a^2}{2(a+1)}\right\}} + t_3^* I_{\left\{\lambda\mu > \frac{a^2}{2(a+1)}\right\}}$, such that: for $i = 1, \cdots, N$,*

$$\begin{aligned} x_i^s &= 0, && \text{when } abs([B_\mu(z)]_i) \le t; \\ x_i^s &= g_{\lambda\mu}([B_\mu(z)]_i), && \text{when } abs([B_\mu(z)]_i) > t. \end{aligned} \tag{3.16}$$

Here the function $g_{\lambda\mu}(\cdot)$ is same as Equation (3.6) with parameter $\lambda\mu$ in place of λ there. $B_\mu(z) = z + \mu A^T(y - Az) \in \Re^N$, as in Equation (3.3).

Proof. The second auxiliary objective function can be rewritten as

$$\begin{aligned} C_\mu(x,z) =& \frac{1}{2}\|x - [(I - \mu A^T A)z + \mu A^T y]\|_2^2 + \lambda\mu P_a(x) \\ &+ \frac{1}{2}\mu\|y\|_2^2 + \frac{1}{2}\|z\|_2^2 - \frac{1}{2}\mu\|Az\|_2^2 - \frac{1}{2}\|(I - \mu A^T A)z + \mu A^T y\|_2^2 \\ =& \frac{1}{2}\sum_{i=1}^N (x_i - [B_\mu(z)]_i)^2 + \lambda\mu\sum_{i=1}^N \rho_a(x_i) \end{aligned}$$

$$+\frac{1}{2}\mu\|y\|_2^2+\frac{1}{2}\|z\|_2^2-\frac{1}{2}\mu\|Az\|_2^2-\frac{1}{2}\|(I-\mu A^TA)z+\mu A^Ty\|_2^2, \tag{3.17}$$

which implies that

$$\begin{aligned} x^s =& arg\min_{x\in\Re^N} C_\mu(x,z) \\ =& arg\min_{x\in\Re^N}\left\{\frac{1}{2}\sum_{i=1}^N(x_i-[B_\mu(z)]_i)^2+\lambda\mu\sum_{i=1}^N\rho_a(x_i)\right\}. \end{aligned} \tag{3.18}$$

Since each component x_i is decoupled, the above minimum can be calculated by minimizing with respect to each x_i individually. For the component-wise minimization, the objective function is:

$$f(x_i,z)=\frac{1}{2}(x_i-[B_\mu(z)]_i)^2+\lambda\mu\rho_a(|x_i|). \tag{3.19}$$

Then by Theorem (3.1), the proof of our Lemma is complete. □

Based on Lemma 3.2, we have the following representation theorem.

THEOREM 3.2. *If $x^*=(x_1^*,x_2^*,...,x_N^*)^T$ is a TL1 regularized solution of model (2.4) with a and λ being positive constants, and $0<\mu<\|A\|^{-2}$, then letting $t=t_2^*1_{\left\{\lambda\mu\le\frac{a^2}{2(a+1)}\right\}}+t_3^*1_{\left\{\lambda\mu>\frac{a^2}{2(a+1)}\right\}}$, the optimal solution satisfies*

$$x_i^*=\begin{cases} g_{\lambda\mu}([B_\mu(x^*)]_i), & if\ |[B_\mu(x^*)]_i|>t \\ 0, & others. \end{cases} \tag{3.20}$$

Proof. The condition $0<\mu<\|A\|^{-2}$ implies

$$\begin{aligned} C_\mu(x,x^*)=&\mu\{\frac{1}{2}\|y-Ax\|_2^2+\lambda P_a(x)\}+\frac{1}{2}\{-\mu\|Ax-Ax^*\|_2^2+\|x-x^*\|_2^2\} \\ \ge&\mu\{\frac{1}{2}\|y-Ax\|_2^2+\lambda P_a(x)\} \\ \ge&C_\mu(x^*,x^*), \end{aligned} \tag{3.21}$$

for any $x\in\Re^N$. So it shows that x^* is a minimizer of $C_\mu(x,x^*)$ as long as x^* is a TL1 solution of model (2.4). In view of Lemma (3.2), we finish the proof. □

4. TL1 thresholding algorithms

In this section, we propose three iterative thresholding algorithms for regularized TL1 optimization problem (2.4), based on Theorem 3.2.

We want to introduce a thresholding operator $G_{\lambda\mu,a}(\cdot):\Re\to\Re$ as

$$G_{\lambda\mu,a}(w)=\begin{cases} 0, & \text{if } |w|\le t; \\ g_{\lambda\mu}(w), & \text{if } |w|>t. \end{cases} \tag{4.1}$$

and expand it to vector space $\Re^N$,

$$G_{\lambda\mu,a}(x)=(G_{\lambda\mu,a}(x_1),...,G_{\lambda\mu,a}(x_N)).$$

According to Theorem 3.2, optimal solution of model (2.4) satisfies representation equation

$$x=G_{\lambda\mu,a}(B_\mu(x)). \tag{4.2}$$

4.1. Direct fixed point iterative algorithm — DFA. A natural idea is to develop an iterative algorithm based on the above fixed point representation directly, with fixed values for parameters: λ, μ and a. We call it direct fixed point iterative algorithm (DFA), for which the iterative scheme is

$$x^{n+1} = G_{\lambda\mu,a}(x^n + \mu A^T(y - Ax^n)) = G_{\lambda\mu,a}(B_\mu(x^n)), \tag{4.3}$$

at $(n+1)$-th step. Recall that the thresholding parameter t is:

$$t = \begin{cases} t_2^* = \lambda\mu\frac{a+1}{a}, & \text{if } \lambda \le \frac{a^2}{2(a+1)\mu}, \\ t_3^* = \sqrt{2\lambda\mu(a+1)} - \frac{a}{2}, & \text{if } \lambda > \frac{a^2}{2(a+1)\mu}. \end{cases} \tag{4.4}$$

In DFA, we have 2 tuning parameters: product term $\lambda\mu$ and TL1 parameter a, which are fixed and can be determined by cross-validation based on different categories of matrix A. Two adaptive iterative thresholding (IT) algorithms will be introduced later.

REMARK 4.1. In TL1 proximal thresholding operator $G_{\lambda\mu,a}$, the threshold value t varies with other parameters:

$$t = t_2^* I_{\left\{\lambda\mu \le \frac{a^2}{2(a+1)}\right\}} + t_3^* I_{\left\{\lambda\mu > \frac{a^2}{2(a+1)}\right\}}.$$

Since $t \ge t_3^* = \sqrt{2\lambda\mu(a+1)} - \frac{a}{2}$, the larger the λ, the larger the threshold value t, and therefore the sparser the solution from the thresholding algorithm.

It is interesting to compare the TL1 thresholding function with the hard/soft thresholding function of l_0/l_1 regularization, and the half thresholding function of $l_{1/2}$ regularization. These three functions ([1,8,18]) are:

$$H_{\lambda,0}(x) = \begin{cases} x, & |x| > (2\lambda)^{1/2} \\ 0, & \text{otherwise} \end{cases} \tag{4.5}$$

$$H_{\lambda,1}(x) = \begin{cases} x - sgn(x)\,\lambda, & |x| > \lambda \\ 0, & \text{otherwise} \end{cases} \tag{4.6}$$

and

$$H_{\lambda,1/2}(x) = \begin{cases} f_{2\lambda,1/2}(x), & |x| > \frac{(54)^{1/3}}{4}(2\lambda)^{2/3} \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

where $f_{\lambda,1/2}(x) = \frac{2}{3}x\left(1 + \cos(\frac{2\pi}{3} - \frac{2}{3}\Phi_\lambda(x))\right)$ and $\Phi_\lambda(x) = \arccos(\frac{\lambda}{8}(\frac{|x|}{3})^{-\frac{3}{2}})$.

In Figure 4.1, we plot the closed-form thresholding formulas (3.9) for $\lambda \le$ and $\lambda > \frac{a^2}{2(a+1)}$ respectively. We observe and prove that when $\lambda < \frac{a^2}{2(a+1)}$, the TL1 threshold function is continuous (Appendix C), same as soft-thresholding function. While if $\lambda > \frac{a^2}{2(a+1)}$, the TL1 thresholding function has a jump discontinuity at threshold, similar to half-thresholding function. For different threshold scheme, it is believed that continuous formula is more stable, while discontinuous formula separates nonzero and trivial coefficients more efficiently and sometimes converges faster [16].

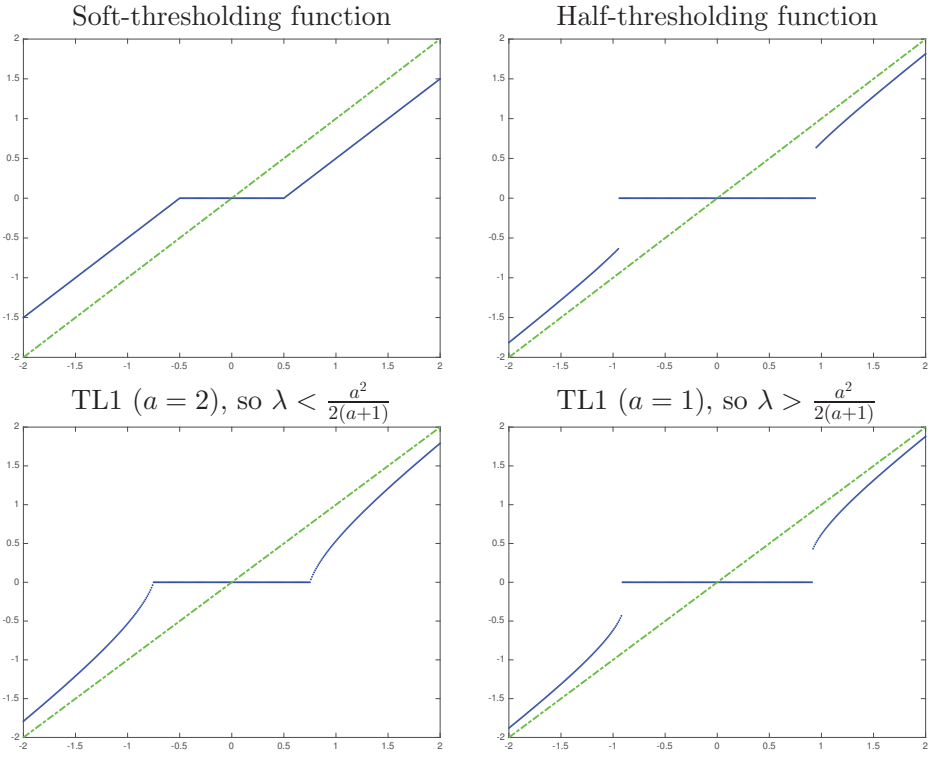

FIG. 4.1. *Soft/half (top left/right), TL1 (sub/super critical, lower left/right) thresholding functions at* $\lambda = 1/2$.

4.2. Convergence Theory for DFA. We establish the convergence theory for direct fixed point iterative algorithm, similar to [18, 23, 25]. Recall in Equation (3.14), we introduced two functions $C_\lambda(x)$ (the objective function in TL1 regularization), and $C_\mu(x,z)$. They will appear in the proof of:

THEOREM 4.1. *Let* $\{x^n\}$ *be the sequence generated by the iteration scheme (4.3) under the condition* $\|A\|^2 < 1/\mu$. *Then:*

1) $\{x^n\}$ *is a minimizing sequence of the function* $C_\lambda(x)$. *If the initial vector* $x^0 = 0$ *and* $\lambda > \frac{\|y\|^2}{2(a+1)}$, *the sequence* $\{x^n\}$ *is bounded.*
2) $\{x^n\}$ *is asymptotically regular, i.e.* $\lim_{n\to\infty} \|x^{n+1} - x^n\| = 0$.
3) *Any limit point* x^* *of* $\{x^n\}$ *is a stationary point satisfying Equation (4.2), that is* $x^* = G_{\lambda\mu,a}(B_\mu(x^*))$.

Proof.

1) From the proof of Lemma (3.2), we can see that

$$C_\mu(x^{n+1}, x^n) = \min_x C_\mu(x, x^n).$$

By the definition of function $C_\lambda(x)$ and $C_\mu(x,z)$ (3.14), we have the following equation:

$$C_\lambda(x^{n+1}) = \frac{1}{\mu}\left[C_\mu(x^{n+1}, x^n) - \frac{1}{2}\|x^{n+1} - x^n\|_2^2\right] + \frac{1}{2}\|Ax^{n+1} - Ax^n\|_2^2$$

Further since $\|A\|^2 < 1/\mu$,

$$\begin{aligned} C_\lambda(x^{n+1}) &\le \frac{1}{\mu}\left\{ C_\mu(x^n, x^n) - \frac{1}{2}\|x^{n+1}-x^n\|_2^2 \right\} + \frac{1}{2}\|Ax^{n+1}-Ax^n\|_2^2 \\ &= C_\lambda(x^n) + \frac{1}{2}(\|A(x^{n+1}-x^n)\|_2^2 - \frac{1}{\mu}\|x^{n+1}-x^n\|_2^2) \\ &\le C_\lambda(x^n). \end{aligned} \tag{4.8}$$

So we know that sequence $\{C_\lambda(x^n)\}$ is decreasing monotonically.
In DFA, if we set trivial initial vector $x^0 = 0$ and parameter λ satisfying $\lambda > \frac{\|y\|^2}{2(a+1)}$, we show that $\{x^n\}$ is bounded. Since $\{C_\lambda(x^n)\}$ is decreasing,

$$C_\lambda(x^n) \le C_\lambda(x^0), \quad \text{for any } n.$$

So we have $\lambda P_a(x^n) \le C_\lambda(x^0)$. As $\|x^n\|_\infty$ be the largest entry in absolute value of vector x^n, $\lambda\rho_a(\|x^n\|_\infty) \le C_\lambda(x^0)$. Due to the definition of ρ_a, it is easy to check that the above inequality is equivalent to

$$\left(\lambda(a+1) - C_\lambda(x^0) \right) \|x^n\|_\infty \le a C_\lambda(x^0).$$

In order to bound $\{x^n\}$, we need the condition $\lambda > C_\lambda(x^0)/(a+1)$. Especially when x^0 is zero, one sufficient condition for $\{x^n\}$ to be bounded is

$$\lambda > \frac{\|y\|^2}{2(a+1)}.$$

2) Since $\|A\|^2 < 1/\mu$, we denote $\epsilon = 1 - \mu\|A\|^2 > 0$. Then we have the inequality $\mu\|A(x^{n+1}-x^n)\|_2^2 \le (1-\epsilon)\|x^{n+1}-x^n\|^2$, which can be rewritten as

$$\|x^{n+1}-x^n\|^2 \le \frac{1}{\epsilon}\|x^{n+1}-x^n\|^2 - \frac{\mu}{\epsilon}\|A(x^{n+1}-x^n)\|_2^2.$$

In the above inequality, we sum the index n from 1 to N and find:

$$\begin{aligned} \sum_{n=1}^{N} \|x^{n+1}-x^n\|^2 &\le \tfrac{1}{\epsilon}\sum_{n=1}^{N}\|x^{n+1}-x^n\|^2 - \tfrac{\mu}{\epsilon}\sum_{n=1}^{N}\|A(x^{n+1}-x^n)\|_2^2 \\ &\le \tfrac{\mu}{\epsilon}\sum_{n=1}^{N} 2\left(C_\lambda(x^n) - C_\lambda(x^{n+1})\right) \\ &\le \tfrac{2\mu}{\epsilon} C_\lambda(x^0), \end{aligned}$$

where the last second inequality comes from Equation (4.8) above . Thus the infinite sum of sequence $\|x^{n+1}-x^n\|^2$ is convergent, which implies that

$$\lim_{n\to\infty} \|x^{n+1}-x^n\| = 0.$$

3) Denote $L_{\lambda,\mu}(z,x) = \frac{1}{2}\|z - B_\mu(x)\|^2 + \lambda\mu P_a(z)$ and

$$D_{\lambda,\mu}(x) = L_{\lambda,\mu}(x,x) - \min_z L_{\lambda,\mu}(z,x).$$

By its definition and the proof of Lemma 3.2 (especially Equation (3.18)), we have $D_{\lambda,\mu}(x) \ge 0$ and

$$D_{\lambda,\mu}(x) = 0 \text{ if and only if } x \text{ satisfies Equation (4.2).}$$

Assume that x^* is a limit point of $\{x^n\}$ and a subsequence of x^n (still denoted the same) converges to it. Because of DFA iterative scheme (4.3), we have $x^{n+1} = arg\min_z L_{\lambda,\mu}(z, x^n)$, which implies that

$$\begin{aligned} &D_{\lambda,\mu}(x^n) = L_{\lambda,\mu}(x^n, x^n) - L_{\lambda,\mu}(x^{n+1}, x^n) \\ &= \lambda\mu\left(P_a(x^n) - P_a(x^{n+1})\right) - \tfrac{1}{2}\|x^{n+1} - x^n\|^2 + \left\langle \mu A^t(Ax^n - y), x^n - x^{n+1}\right\rangle. \end{aligned}$$

Thus we know

$$\begin{aligned} &\lambda P_a(x^n) - \lambda P_a(x^{n+1}) \\ &= \tfrac{1}{2\mu}\|x^{n+1} - x^n\|^2 + \tfrac{1}{\mu} D_{\lambda,\mu}(x^n) + \left\langle A^t(Ax^n - y), x^n - x^{n+1}\right\rangle, \end{aligned}$$

from which we get

$$\begin{aligned} C_\lambda(x^n) - C_\lambda(x^{n+1}) &= \lambda P_a(x^n) - \lambda P_a(x^{n+1}) + \tfrac{1}{2}\|Ax^n - y\|^2 - \tfrac{1}{2}\|Ax^{n+1} - y\|^2 \\ &= \tfrac{1}{2\mu}\|x^{n+1} - x^n\|^2 + \tfrac{1}{\mu} D_{\lambda,\mu}(x^n) - \tfrac{1}{2}\|A(x^n - x^{n+1})\|_2^2 \\ &\geq \tfrac{1}{\mu} D_{\lambda,\mu}(x^n) + \tfrac{1}{2}(\tfrac{1}{\mu} - \|A\|^2)\|x^n - x^{n+1}\|^2. \end{aligned}$$

So $0 \leq D_{\lambda,\mu}(x^n) \leq \mu\left(C_\lambda(x^n) - C_\lambda(x^{n+1})\right)$. Also we know from part (1) of this theorem that $\{C_\lambda(x^n)\}$ converges, so $\lim_{n\to\infty} D_{\lambda,\mu}(x^n) = 0$. Thus as the limit point of the sequence x^n, the point x^* satisfies Equation (4.2).

□

4.3. Semi-adaptive thresholding algorithm — TL1IT-s1. In the following 2 subsections, we present two adaptive parameter TL1 algorithms. We begin with formulating an optimality condition on the regularization parameter λ, which serves as the basis for parameter selection and updating in the semi-adaptive algorithm.

Let us consider the so called k-sparsity problem for model (2.4). The solution is k-sparse by prior knowledge or estimation. For any μ, denote $B_\mu(x) = x + \mu A^T(b - Ax)$ and $|B_\mu(x)|$ is the vector from taking absolute value of each entry of $B_\mu(x)$. Suppose that x^* is the TL1 solution, and without loss of generality, $|B_\mu(x^*)|_1 \geq |B_\mu(x^*)|_2 \geq ... \geq |B_\mu(x^*)|_N$. Then, the following inequalities hold:

$$\begin{aligned} &|B_\mu(x^*)|_i > t \Leftrightarrow i \in \{1, 2, ..., k\}, \\ &|B_\mu(x^*)|_j \leq t \Leftrightarrow j \in \{k+1, k+2, ..., N\}, \end{aligned} \tag{4.9}$$

where t is our threshold value.

Recall that $t_3^* \leq t \leq t_2^*$. So

$$\begin{aligned} &|B_\mu(x^*)|_k \geq t \geq t_3^* = \sqrt{2\lambda\mu(a+1)} - \tfrac{a}{2}; \\ &|B_\mu(x^*)|_{k+1} \leq t \leq t_2^* = \lambda\mu\tfrac{a+1}{a}. \end{aligned} \tag{4.10}$$

It follows that

$$\lambda_1 \equiv \frac{a|B_\mu(x^*)|_{k+1}}{\mu(a+1)} \leq \lambda \leq \lambda_2 \equiv \frac{(a+2|B_\mu(x^*)|_k)^2}{8(a+1)\mu} \tag{4.11}$$

or $\lambda^* \in [\lambda_1, \lambda_2]$.

The above estimate helps to set optimal regularization parameter. A choice of λ^* is

$$\lambda^* = \begin{cases} \lambda_1, & \text{if } \lambda_1 \leq \frac{a^2}{2(a+1)\mu}, \text{ then } \lambda^* \leq \frac{a^2}{2(a+1)\mu} \Rightarrow t = t_2^*; \\ \lambda_2, & \text{if } \lambda_1 > \frac{a^2}{2(a+1)\mu}, \text{ then } \lambda^* > \frac{a^2}{2(a+1)\mu} \Rightarrow t = t_3^*. \end{cases} \tag{4.12}$$

Algorithm 1: TL1 Thresholding Algorithm — TL1IT-s1

Initialize: x^0; $\mu_0 = \frac{(1-\varepsilon)}{\|A\|^2}$ and a;
while *not converged* **do**
  $\mu = \mu_0$; $z^n := B_\mu(x^n) = x^n + \mu A^T(y - Ax^n)$;
  $\lambda_1^n = \frac{a|z^n|_{k+1}}{\mu(a+1)}$; $\lambda_2^n = \frac{(a+2|z^n|_k)^2}{8(a+1)\mu}$;
  if $\lambda_1^n \le \frac{a^2}{2(a+1)\mu}$ **then**
    $\lambda = \lambda_1^n$; $t = \lambda\mu\frac{a+1}{a}$;
    for i = 1:length(x)
      if $|z^n(i)| > t$, then $x^{n+1}(i) = g_{\lambda\mu}(z^n(i))$;
      if $|z^n(i)| \le t$, then $x^{n+1}(i) = 0$.
  else
    $\lambda = \lambda_2^n$; $t = \sqrt{2\lambda\mu(a+1)} - \frac{a}{2}$;
    for i = 1:length(x)
      if $|z^n(i)| > t$, then $x^{n+1}(i) = g_{\lambda\mu}(z^n(i))$;
      if $|z^n(i)| \le t$, then $x^{n+1}(i) = 0$.
  end
  $n \to n+1$;
end

In practice, we approximate x^* by x^n in (4.11), so

$$\lambda_1 = \frac{a|B_\mu(x^n)|_{k+1}}{\mu(a+1)}, \qquad \lambda_2 = \frac{(a+2|B_\mu(x^n)|_k)^2}{8(a+1)\mu},$$

at each iteration step. So we have an adaptive iterative algorithm without pre-setting the regularization parameter λ. Also the TL1 parameter a is still free (to be selected), thus this algorithm is overall semi-adaptive, which is named TL1IT-s1 for short and summarized in Algorithm 1.

4.4. Adaptive thresholding Algorithm — TL1IT-s2. For TL1IT-s1 algorithm, at each iteration step, it is required to compare λ_n and $\frac{a^2}{2(a+1)\mu}$. Here instead, we vary TL1 parameter 'a' and choose $a = a_n$ in each iteration, such that the inequality $\lambda_n \le \frac{a_n^2}{2(a_n+1)\mu_n}$ holds.

The thresholding scheme is now simplified to just one threshold parameter $t = t_2^*$. Putting $\lambda = \frac{a^2}{2(a+1)\mu}$ at critical value, the parameter a is expressed as:

$$a = \lambda\mu + \sqrt{(\lambda\mu)^2 + 2\lambda\mu}. \tag{4.13}$$

The threshold value is:

$$t = t_2^* = \lambda\mu\frac{a+1}{a} = \frac{\lambda\mu}{2} + \frac{\sqrt{(\lambda\mu)^2 + 2\lambda\mu}}{2}. \tag{4.14}$$

Let x^* be the TL1 optimal solution. Then we have the following inequalities:

$$\begin{aligned} &|B_\mu(x^*)|_i > t \Leftrightarrow i \in \{1,2,...,k\}, \\ &|B_\mu(x^*)|_j \le t \Leftrightarrow j \in \{k+1,k+2,...,N\}. \end{aligned} \tag{4.15}$$

So, for parameter λ, we have:

$$\frac{1}{\mu}\frac{2|B_\mu(x^*)|^2_{k+1}}{1+2|B_\mu(x^*)|_{k+1}} \le \lambda \le \frac{1}{\mu}\frac{2|B_\mu(x^*)|^2_{k}}{1+2|B_\mu(x^*)|_{k}}.$$

Once the value of λ is determined, the parameter a is given by (4.13).

In the iterative method, we approximate the optimal solution x^* by x^n. The resulting parameter selection is:

$$\begin{aligned} \lambda_n &= \frac{1}{\mu_n}\frac{2|B_{\mu_n}(x^*)|^2_{k+1}}{1+2|B_{\mu_n}(x^*)|_{k+1}}; \\ a_n &= \lambda_n\mu_n + \sqrt{(\lambda_n\mu_n)^2+2\lambda_n\mu_n}. \end{aligned} \tag{4.16}$$

In this algorithm (TL1IT-s2 for short), only parameter μ is fixed and $\mu \in (0, \|A\|^{-2})$. The summary is below (Algorithm 2).

Algorithm 2: Adaptive TL1 Thresholding Algorithm — TL1IT-s2

Initialize: x^0, $\mu_0 = \frac{(1-\varepsilon)}{\|A\|^2}$;
while *not converged* **do**
  $\mu = \mu_0$; $z^n := x^n + \mu A^T(y - Ax^n)$;
  $\lambda_n = \frac{1}{\mu}\frac{2|z^n_{k+1}|^2}{1+2|z^n_{k+1}|}$;
  $a_n = \lambda_n\mu + \sqrt{(\lambda_n\mu)^2+2\lambda_n\mu}$;
  $t = \frac{\lambda_n\mu}{2} + \frac{\sqrt{(\lambda_n\mu)^2+2\lambda_n\mu}}{2}$;
  for i = 1:length(x)
    if $|z^n(i)| > t$, then $x^{n+1}(i) = g_{\lambda_n\mu}(z^n(i))$;
    if $|z^n(i)| \le t$, then $x^{n+1}(i) = 0$.
  $n \to n+1$;
end

5. Numerical experiments

In this section, we carried out a series of numerical experiments to demonstrate the performance of the TL1 thresholding algorithm: semi-adaptive TL1IT-s1. All the experiments here are conducted by applying our algorithm to sparse signal recovery in compressed sensing. Two classes of randomly generated sensing matrices are used to compare our algorithms with the state-of-the-art iterative non-convex thresholding solvers: **Hard-thresholding** [2], **Half-thresholding** [18]. Here all these thresholding algorithms need a sparsity estimation to accelerate convergence. Also the Hard Thresholding algorithm (AIHT) in [2] has an additional double over-relaxation step for significant speedup in convergence. In the following run time comparison of the three algorithms, AIHT is clearly the most efficient under the uncorrelated Gaussian sensing matrix.

We also tested on the adaptive scheme: TL1IT-s2. However, its performance is always no better than TL1IT-s1, and so its results are not shown here. We suggest to use TL1IT-s1 first in CS applications. That TL1IT-s2 is not as competitive as TL1IT-s1 may be attributed to its limited thresholding scheme. Utilizing double thresholding

schemes is helpful for TL1IT. We noticed in our computations that at the beginning of iterations, the λ_n's cross the critical value $\frac{a^2}{2(a+1)\mu}$ frequently. Later on, they tend to stay on one side, depending on the sensing matrix A. However, the sub-critical threshold is used for all A's in TL1IT-s2.

Here we compare only the non-convex iterative thresholding methods, and did not include the soft-thresholding algorithm. The two classes of random matrices are:

1) Gaussian matrices.
2) Over-sampled discrete cosine transform (DCT) matrices with factor F.

All our tests were performed on a *Lenovo* desktop: 16 GB of RAM and Intel Core processor $i7-4770$ with CPU at $3.40GHz \times 8$ under 64-bit Ubuntu system.

The TL1 thresholding algorithms do not guarantee a global minimum in general, due to nonconvexity. Indeed we observed that TL1 thresholding with random starts may get stuck at local minima especially when the matrix A is ill-conditioned (e.g. A has a large condition number or is highly coherent). A good initial vector x^0 is important for thresholding algorithms. In our numerical experiments, instead of having $x^0 = 0$ or random, we apply YALL1 (an alternating direction l_1 method, [19]) a number of times, e.g. 20 times, to produce a better initial guess x^0. This procedure is similar to algorithm DCATL1 [25] initiated at zero vector so that the first step of DCATL1 reduces to solving an unconstrained l_1 regularized problem. For all these iterative algorithms, we implement a unified stopping criterion as $\frac{\|x^{n+1}-x^n\|}{\|x^n\|} \leq 10^{-8}$ or maximum iteration step equals 3000.

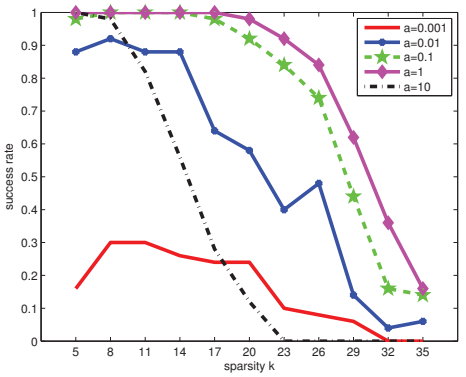

FIG. 5.1. *Sparse recovery success rates for selection of parameter a with* 128×512 *Gaussian random matrices and TL1IT-s1 method.*

5.1. Optimal parameter testing for TL1IT-s1. In TL1IT-s1, the parameter 'a' is still free. When 'a' tends to zero, the penalty function approaches the l_0 norm. We tested TL1IT-s1 on sparse vector recovery with different 'a' values, varying among $\{0.001, 0.01, 0.1, 1, 100\}$. In this test, matrix A is a 128×512 random matrix, generated by multivariate normal distribution $\sim \mathcal{N}(0, \Sigma)$. Here the covariance matrix $\Sigma = \{1_{(i=j)} + 0.2 \times 1_{(i \neq j)}\}_{i,j}$. The true sparse vector x^* is also randomly generated under Gaussian distribution, with sparsity k from the set $\{8, 10, 12, \cdots, 32\}$.

For each value of 'a', we conducted 100 test runs with different samples of A and ground truth vector x^*. The recovery is successful if the relative error: $\frac{\|x_r - x^*\|_2}{\|x^*\|_2} \leq 10^{-2}$.

Figure (5.1) shows the success rate vs. sparsity using TL1IT-s1 over 100 independent trials for various parameter a and sparsity k. We see that the algorithm with $a=1$ is the best among all tested parameter values. Thus in the subsequent computation, we set the parameter $a=1$. The parameter $\mu=\frac{0.99}{\|A\|^2}$.

5.2. Signal recovery without noise.

Gaussian Sensing Matrix. The sensing matrix A is drawn from $\mathcal{N}(0,\Sigma)$, the multi-variable normal distribution with covariance matrix $\Sigma=\{(1-r)1_{(i=j)}+r\}_{i,j}$, where r ranges from 0 to 0.8. The larger parameter r is, the more difficult it is to recover the sparse ground truth vector. The matrix A is 128×512, and the sparsity k varies among $\{5,\ 8,\ 11,\cdots,\ 35\}$.

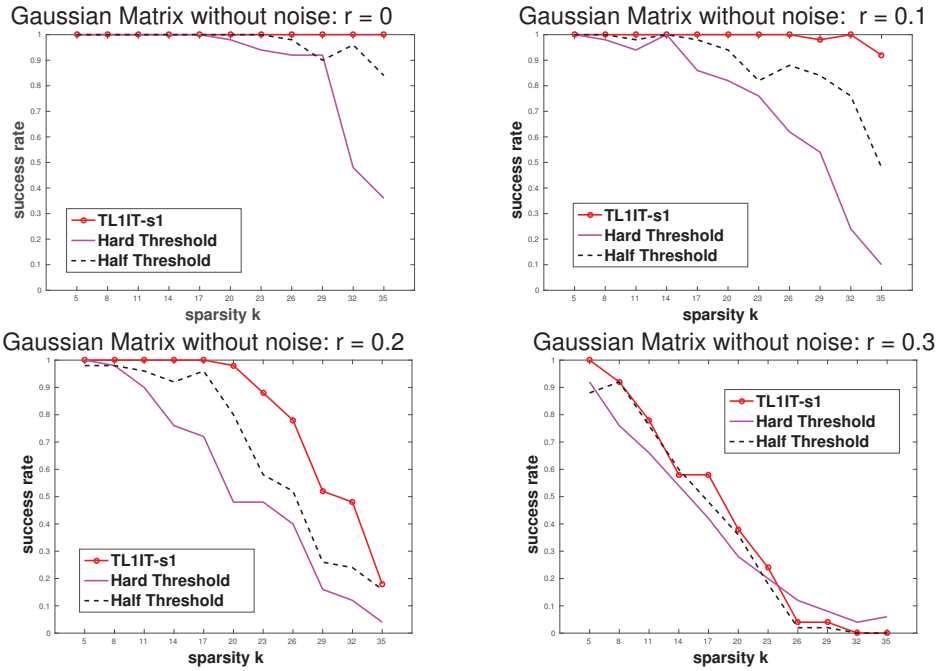

FIG. 5.2. *Sparse recovery algorithm comparison for* 128×512 *Gaussian sensing matrices without measurement noise at covariance parameter* $r=0,0.1,0.2,0.3$.

We compare the three IT algorithms in terms of success rate averaged over 50 random trials. A success is recorded if the relative error of recovery is less than 0.001. The success rate of each algorithm is plotted in Figure 5.2 with parameter r from the set: $\{0,\ 0.1,\ 0.2,\ 0.3\}$.

We see that all three algorithms can accurately recover the signal when r and sparsity k are both small. However, the success rates decline, along with the increase of r and sparsity k. At $r=0$, the TL1IT-s1 scheme recovers almost all testing signals from different sparsity. Half thresholding algorithm maintains nearly the same high success rates with a slight decrease when $k\geq26$. At $r=0.3$, TL1IT-s1 leads the half thresholding algorithm with a small margin. In all cases, TL1IT-s1 outperforms the other two, while the half thresholding algorithm is the second.

sparsity	5	8	11	14	17	20
TL1IT-s1	0.031	0.054	0.047	0.055	0.053	0.059
Hard	**0.003**	**0.003**	**0.005**	**0.006**	**0.007**	**0.007**
Half	0.019	0.017	0.017	0.023	0.020	0.025

TABLE 5.1. *Time efficiency (in sec) comparison for 3 algorithms under Gaussian matrices.*

Comparison of time efficiency under Gaussian measurements. One interesting question is about the time efficiency for different thresholding algorithms. As seen from Figure 5.2, almost all the 3 algorithms, under Gaussian matrices with covariance parameter $r=0$ and sparsity $k=5,\cdots,20$, achieve 100 % success recovery. So we measured the average convergent time over 20 random tests in the above situation (see Table 1), where all the parameters are tuned to obtain relative errors around 10^{-5}.

From the table, we know that Hard Thresholding algorithm costs the least time among all three. So under this uncorrelated normal distribution measurement, Hard Thresholding algorithm is the most efficient, with Half Thresholding algorithm the second. Though TL1IT-s1 has the lowest relative error in recovery, it takes more time. One reason is that TL1IT-s1 iterations go between two thresholding schemes, which makes it more adaptive to data for a higher computational cost.

Over-sampled DCT sensing matrix. The over-sampled DCT matrices [13, 14] are:

$$\begin{aligned} &A=[a_1,...,a_N]\in\Re^{M\times N}\\ &\text{where}\quad a_j=\frac{1}{\sqrt{M}}cos(\frac{2\pi\omega(j-1)}{F}),\quad j=1,...,N,\\ &\text{and } \omega \text{ is a random vector, drawn uniformly from } (0,1)^M. \end{aligned} \tag{5.1}$$

Such matrices appear as the real part of the complex discrete Fourier matrices in spectral estimation and super-resolution problems [6, 13]. An important property is their high coherence measured by the maximum of absolute value of cosine of the angles between each pair of column vectors of A. For a 100×1000 over-sampled DCT matrix at $F=10$, the coherence is about 0.9981, while at $F=20$ the coherence of the same size matrix is typically 0.9999.

The sparse recovery under such matrices is possible only if the non-zero elements of solution x are sufficiently separated. This phenomenon is characterized as *minimum separation* in [6], with minimum length referred as the Rayleigh length (RL). The value of RL for matrix A is equal to the factor F. It is closely related to the coherence in the sense that larger F corresponds to larger coherence of a matrix. We find empirically that at least 2RL is necessary to ensure optimal sparse recovery with spikes further apart for more coherent matrices.

Under the assumption of sparse signal with $2RL$ separated spikes, we compare the four non-convex IT algorithms in terms of success rate. The sensing matrix A is of size 100×1500. A success is recorded if the relative recovery error is less than 0.001. The success rate is averaged over 50 random realizations.

Figure 5.3 shows success rates for the four algorithms with increasing factor F from 2 to 8. Along with the increasing F, the success rates for the algorithms decrease, though at different rates of decline. In all plots, TL1IT-s1 is the best with the highest success rates. At $F=2$, both half thresholding and hard thresholding successfully recover signal

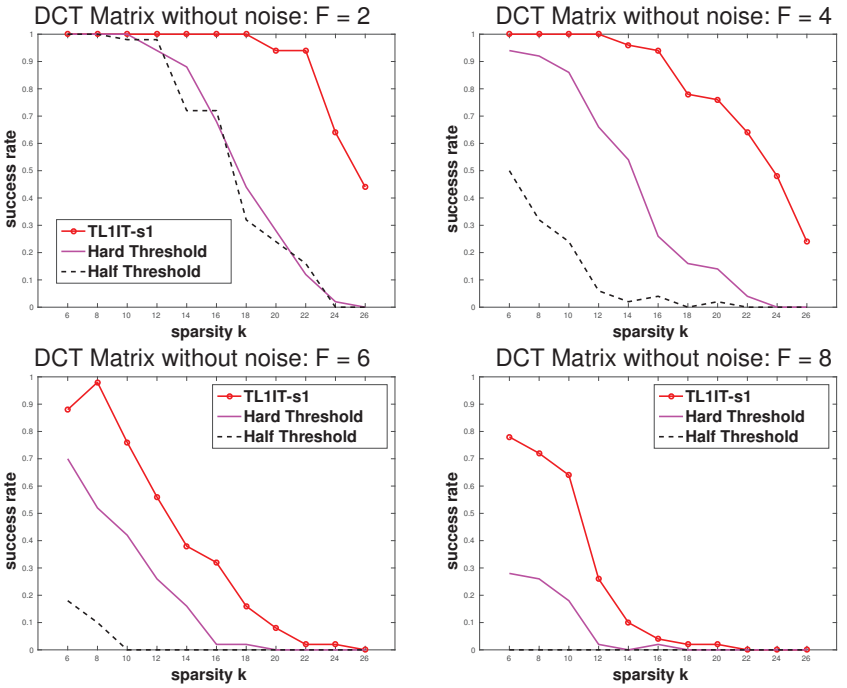

FIG. 5.3. *Algorithm comparison for* 100×1500 *over-sampled DCT random matrices without noise at different factor* F.

in the regime of small sparsity k. However when F becomes larger, the half thresholding algorithm deteriorates sharply. Especially at $F=8$, it lies almost flat.

5.3. Signal recovery in noise. Let us consider recovering signal in noise based on the model $y=Ax+\varepsilon$, where ε is drawn from independent Gaussian $\varepsilon\in\mathcal{N}(0,\sigma^2)$ with $\sigma=0.01$. The non-zero entries of sparse vector x are drawn from $\mathcal{N}(0,4)$. In order to recover signal with certain accuracy, the error ε can not be too large. So in our test runs, we also limit the noise amplitude as $|\varepsilon|_\infty\leq0.01$.

Gaussian sensing matrix. Here we use the same method in Part B to obtain Gaussian matrix A. Parameter r and sparsity k are in the same set $\{0,\ 0.2,\ 0.4,\ 0.5\}$ and $\{5,\ 8,\ 11,\ ...,\ 35\}$. Due to the presence of noise, it becomes harder to accurately recover the original signal x. So we tune down the requirement for a success to relative error $\frac{\|x^r-x\|}{\|x\|}\leq10^{-2}$.

The numerical results are shown in Figure 5.4. In this experiment, TL1IT-s1 again has the best performance, with half thresholding algorithm the second. At $r=0$, TL1IT-s1 scheme is robust and recovers signals successfully in almost all runs, which is the same case under both noisy and noiseless conditions.

Over-sampled DCT sensing matrix. Figure 5.5 shows results of three algorithms under the over-sampled DCT sensing matrices. Relative error $\frac{\|x^r-x\|}{\|x\|}\leq10^{-2}$ qualifies for a success. In this case, TL1IT-s1 is also the best numerical method, same as in the noise free tests. It degrades most slowly under high coherence sensing matrices $(F=6,8)$.

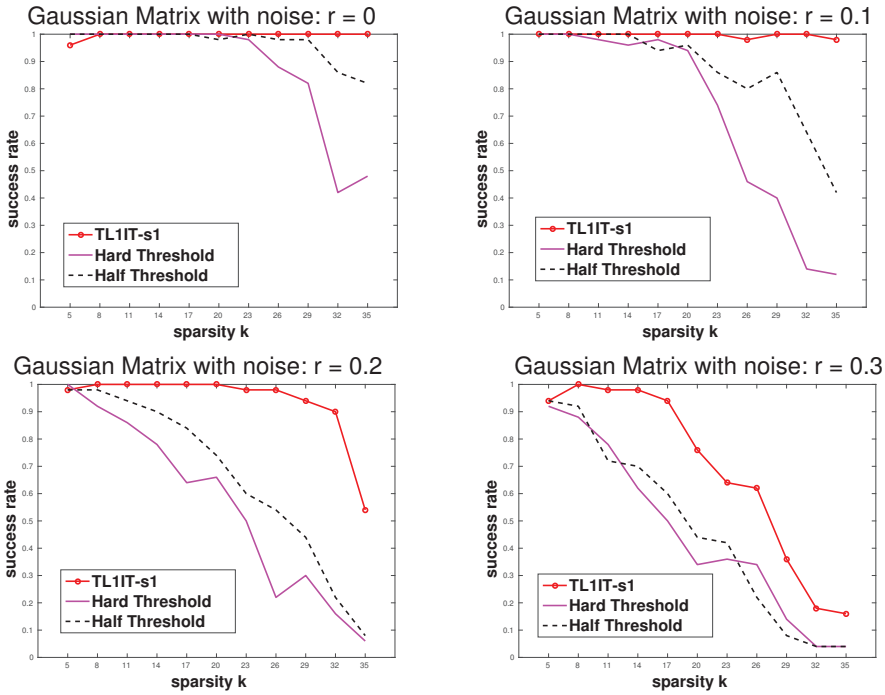

FIG. 5.4. *Algorithm comparison in success rates for* 128×512 *Gaussian sensing matrices with additive noise at different coherence* r.

5.4. Robustness under sparsity estimation. In the previous numerical experiments, the sparsity of the problem is known and used in all thresholding algorithms. However, in many applications, the sparsity of problem may be hard to know exactly. Instead, one may only have a rough estimate of the sparsity. How is the performance of the TL1IT-s1 when the exact sparsity k is replaced by a rough estimate?

Here we perform simulations to verify the robustness of TL1IT-s1 algorithm with respect to sparsity estimation. Different from previous examples, Figure 5.6 shows mean square error (MSE), instead of relative l_2 error. The sensing matrix A is generated from Gaussian distribution with $r = 0$. Number of columns, M varies over several values, while the number of rows, N, is fixed at 512. In each experiment, we change the sparsity estimation for the algorithm from 60 to 240. The real sparsity is $k = 130$. This way, we test the robustness of the TL1IT algorithms under both underestimation and overestimation of sparsity.

In Figure 5.6, we see that TL1IT-s1 scheme is robust with respect to sparsity estimation, especially for sparsity over-estimation. In other words, TL1IT scheme can withstand the estimation error if given enough measurements.

5.5. Comparison among TL1 Algorithms. We have proposed three TL1 thresholding algorithms: DFA with fixed parameters, semi-adaptive algorithm – TL1IT-s1 and adaptive algorithm – TL1IT-s2. Also in [25], we presented a TL1 difference of convex function algorithm – DCATL1. Here we compare all four TL1 algorithms, under both Gaussian and Over-sampled DCT sensing matrices. For the fixed parameter DFA, we tested two thresholding schemes: DFA-s1 for continuous thresholding scheme under $\lambda\mu < a^2/2(a+1)$, and DFA-s2 for discontinuous thresholding scheme under $\lambda\mu > a^2/2(a+1)$.

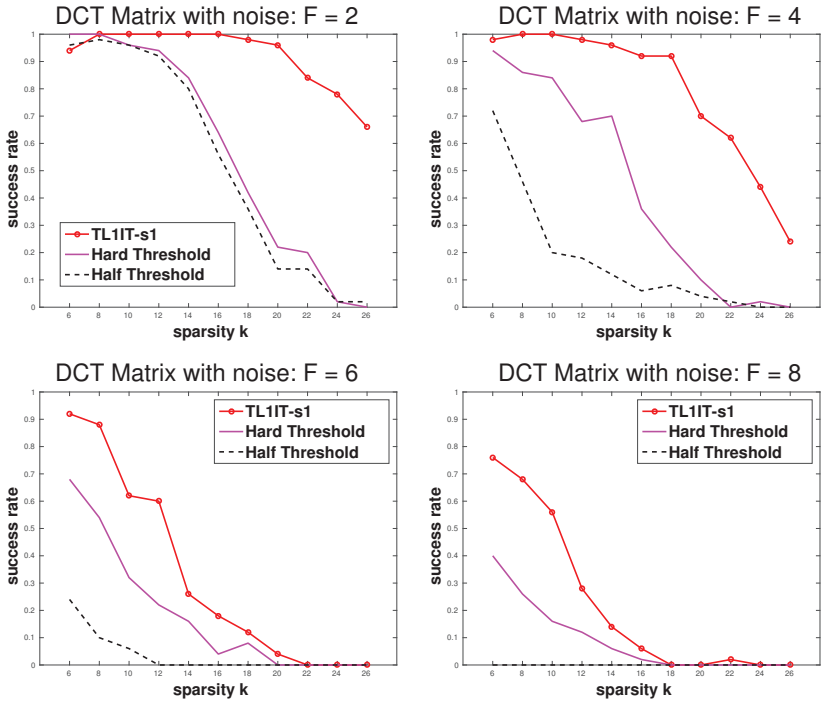

FIG. 5.5. *Algorithm comparison for over-sampled DCT matrices with additive noise:* $M=100$, $N=1500$ *at* $F=2,4,6,8$.

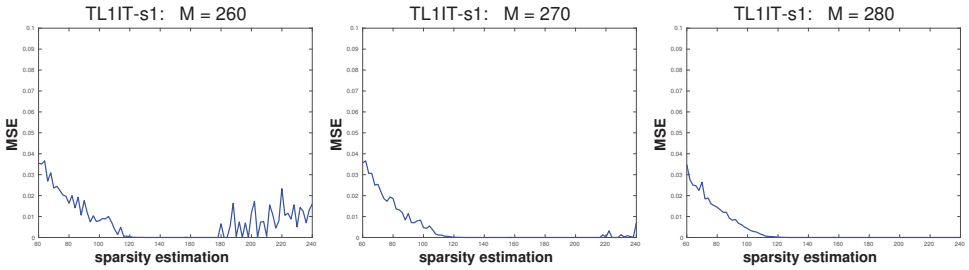

FIG. 5.6. *Robustness tests (mean square error vs. sparsity) for TL1IT-s1 thresholding algorithm under Gaussian sensing matrices:* $r=0, N=512$ *and number of measurements* $M=260,270,280$. *The real sparsity is fixed as* $k=130$.

In the comparison experiments, we chose Gaussian matrices with covariance parameter $r=0$ and Over-sampled DCT matrices with $F=2$. The results are showed in Figure 5.7. Under Gaussian sensing matrices, DCATL1 and TL1IT-s1 achieved 100% success rate to recover ground truth sparse vector, while TL1IT-s2 failed sometimes when sparsity is higher than 28. Also it is interesting to notice that DFA-s2 with discontinuous thresholding scheme behaved better than DFA-s1, the continuous thresholding scheme. For over-sampled DCT sensing tests, DCATL1 is clearly the best among all TL1 algorithms, with TL1IT-s1 the second. Also the performance of TL1IT-s2 declined sharply under this test, which is consistent with our previous numerical experiments for thresholding algorithms. Due to this fact, we only showed TL1IT-s1 in the plots for comparison with hard and half thresholding algorithms.

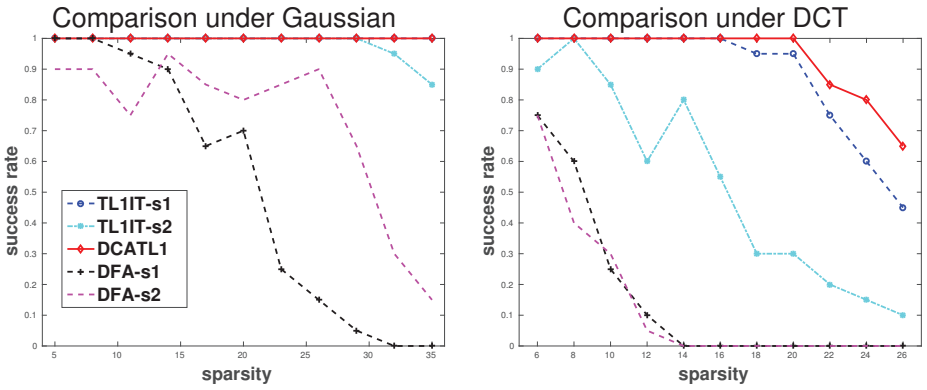

FIG. 5.7. *TL1 algorithms comparison. Y-axis is success rate from 20 random tests with accepted relative error* 10^{-3}. *X-axis is sparsity value* k. *Left:* 128×512 *Gaussian sensing matrices with sparsity* $k=5,\cdots,35$. *Right:* 100×1500 *Gaussian sensing matrices with sparsity* $k=6,\cdots,26$.

The two adaptive TL1 thresholding algorithms are far ahead of 2 DFA algorithms, which shows the advantages of adaptivity. Although DCATL1 out-performed all TL1 thresholding algorithms in the above tests, it requires two nested iterations, and an inverse matrix operation, which is costly for a large size sensing matrix. So for large scale CS applications, thresholding algorithms will have their advantages, including parallel implementations.

6. Conclusion

We have studied compressed sensing problems with the transformed l_1 penalty function for the unconstrained regularization model. We established a precise thresholding representation theory with closed form thresholding formula, and proposed three iterative thresholding schemes. The TL1 thresholding schemes can be either continuous (as in soft-thresholding of l_1) or discontinuous (as in half-thresholding of $l_{1/2}$), depending on whether the parameters belong to the subcritical or supercritical regime. Correspondingly, there are two parameter setting strategies for regularization parameter λ, when the k-sparsity problem is solved. A convergence theorem is proved for the fixed parameter TL1 algorithm (DFA).

Numerical experiments showed that the semi-adaptive TL1It-s1 algorithm is the best performer for sparse signal recovery under sensing matrices with a broad range of coherence and under controlled measurement noise. TL1IT-s1 is also robust under sparsity estimation error.

In a future work, we plan to explore TL1 thresholding algorithms for imaging science among other higher dimensional problems.

Appendix A. Relations of three parameters: t_1^*, t_2^* and t_3^*.

$$\begin{cases} t_1^* = \dfrac{3}{2^{2/3}}(\lambda a(a+1))^{1/3} - a; \\ t_2^* = \lambda\frac{a+1}{a} \\ t_3^* = \sqrt{2\lambda(a+1)} - \frac{a}{2}. \end{cases}$$

In this appendix, we prove that

$$t_1^* \le t_3^* \le t_2^*,$$

for all positive parameters λ and a. Also when $\lambda = \frac{a^2}{2(a+1)}$, they are equal to $\frac{a}{2}$.

(1) $t_1^* \le t_3^*$.
Consider the following equivalent relations:

$$\begin{aligned} t_1^* \le t_3^* \Leftrightarrow & \quad \frac{3}{2^{2/3}}(\lambda a(a+1))^{1/3} \le \tfrac{a}{2} + \sqrt{2\lambda(a+1)} \\ \Leftrightarrow & \quad 0 \le (\sqrt{2\lambda(a+1)})^3 + \tfrac{a^3}{8} - \tfrac{15}{4}a(a+1)\lambda + \tfrac{3a^2}{4}\sqrt{2\lambda(a+1)} \end{aligned}$$

Denote $\beta = \sqrt{\lambda}$, then function $P(\lambda) = (\sqrt{2\lambda(a+1)})^3 + \frac{a^3}{8} - \frac{15}{4}a(a+1)\lambda + \frac{3a^2}{4}\sqrt{2\lambda(a+1)}$ can be rewriten as a cubic polynomial of β:

$$\beta^3(2(a+1))^{3/2} - \beta^2\frac{15}{8}a(2(a+1)) + \beta\frac{3a^2}{4}\sqrt{2(a+1)} + \frac{a^3}{8}.$$

This polynomial can be factorized as

$$(2(a+1))^{3/2}\left(\beta - \frac{a}{\sqrt{2(a+1)}}\right)^2\left(\beta + \frac{a}{8\sqrt{2(a+1)}}\right).$$

Thus for nonnegative parameter $\lambda = \beta^2$, it is always true that $P(\lambda) \ge 0$. Therefore, we have $t_1^* \le t_3^*$. They are equal to $\frac{a}{2}$ if and only if $\lambda = \frac{a^2}{2(a+1)}$.

(2) $t_3^* \le t_2^*$.
This is because

$$\begin{aligned} t_3^* \le t_2^* \Leftrightarrow & \quad \sqrt{2\lambda(a+1)} \le \tfrac{a}{2} + \lambda\tfrac{a+1}{a} \\ \Leftrightarrow & \quad 2\lambda(a+1) \le \tfrac{a^2}{4} + \lambda(a+1) + \lambda^2\tfrac{(a+1)^2}{a^2} \\ \Leftrightarrow & \quad 0 \le \left(\tfrac{a}{2} - \lambda\tfrac{a+1}{a}\right)^2. \end{aligned}$$

So inequality $t_3^* \le t_2^*$ holds. Further, $t_3^* = t_2^* = a/2$ if and only if $\lambda = \frac{a^2}{2(a+1)}$.

Appendix B. Formula of optimal value y^* when $\lambda > \frac{a^2}{2(a+1)}$ and $t_1^* < x < t_2^*$. Define function $w(x) = x - g_\lambda(x) - \frac{a}{2}$, where

$$g_\lambda(x) = sgn(x)\left\{\frac{2}{3}(a+|x|)\cos(\frac{\varphi(x)}{3}) - \frac{2a}{3} + \frac{|x|}{3}\right\}$$

with $\varphi(x) = \arccos(1 - \frac{27\lambda a(a+1)}{2(a+|x|)^3})$.

(1) First, we need to check that $x = t_3^*$ indeed is a solution for equation $w(x) = 0$. Since $\lambda > \frac{a^2}{2(a+1)}$, $t_3^* = \sqrt{2\lambda(a+1)} - \frac{a}{2} > 0$. Thus:

$$\begin{aligned} \cos(\varphi(t_3^*)) &= 1 - \frac{27\lambda a(a+1)}{2(a+t_3^*)^3} \\ &= 1 - \frac{27\lambda a(a+1)}{2(\frac{a}{2} + \sqrt{2\lambda(a+1)})^3}. \end{aligned}$$

Further, by using the relation $\cos(\varphi) = 4\cos^3(\varphi/3) - 3\cos(\varphi/3)$ and $0 \le \varphi/3 \le \frac{\pi}{3}$, we have

$$\cos\left(\frac{\varphi(t_3^*)}{3}\right) = \frac{\sqrt{2\lambda(a+1)} - a/4}{a/2 + \sqrt{2\lambda(a+1)}}.$$

Plugging this formula into $g_\lambda(t_3^*)$ shows that $g_\lambda(t_3^*)=\sqrt{2\lambda(a+1)}-a=t_3^*-a/2$. So t_3^* is a root for function $w(t)$ and $t_3^*\in(t_1^*,t_2^*)$.

(2) Second we prove that the function $w(x)$ changes sign at $x=t_3^*$.
Notice that according to Lemma 3.1 , $g_\lambda(x)$ is the largest root for cubic polynomial $P(t)=t(a+t)^2-x(a+t)^2+\lambda a(a+1)$, if $x>t_1^*$.
Take $t=x$, we know $P(x)=\lambda a(a+1)>0$. Let us consider the value of $P(x-a/2)$. It is easy to check that: $P(x-a/2)<0\Leftrightarrow x>t_3^*$.

(a) $x\in(t_3^*,t_2^*)$.
We will have $P(x-a/2)<0$ and $P(x)>0$. While also the largest solution of $P(t)=0$ is $t=g_\lambda(x)<x$. Thus we are sure that $g_\lambda(x)\in(x-a/2,x)$, and then $x-g_\lambda(x)<a/2\Rightarrow w(x)<0$. So the optimal value is $y^*=y_0=g_\lambda(x)$.

(b) $x\in(t_1^*,t_3^*)$. We have $P(x-a/2)>0$ and $P(x)>0$. Due to the proof of Lemma 3.1, one possible situation is that there are two roots y_0 and y_1 within interval $(x-a/2,x)$. But we can exclude this case. This is because, by formula (3.7),

$$\begin{aligned} y_0-y_1 &= \tfrac{2(a+x)}{3}\{\cos(\varphi/3)-\cos(\varphi/3+\pi/3)\} \\ &= \tfrac{2(a+x)}{3}\{2\sin(\varphi/3+\pi/6)\sin(\pi/6)\} \\ &= \tfrac{2(a+x)}{3}\sin(\varphi/3+\pi/6). \end{aligned} \tag{B.1}$$

Here $\varphi/3\in[\pi/6,\pi/2]$. So $y_0-y_1\geq\frac{(a+x)}{3}$. Also we have $x>t_1^*>a/2$ when $\lambda>\frac{a^2}{2(a+1)}$. Thus

$$y_0-y_1>a/2,$$

which is in contradiction with the assumption that both y_0 and $y_1\in(x-a/2,x)$. So there are no roots for $P(t)=0$ in $(x-a/2,x)$. Then we know $y_0=g_\lambda(x)<x-a/2$. That is to say, $w(x)>0$, so the optimal value is $y^*=0$.

Appendix C. Continuity of TL1 threshold function at t_2^* when $\lambda\leq\frac{a^2}{2(a+1)}$.
Threshold operator $H_{\lambda,a}(\cdot)$ is defined as

$$H_{\lambda,a}(x)=\begin{cases}0, & \text{if } |x|\leq t;\\ g_\lambda(x), & \text{if } |x|>t.\end{cases}$$

When $\lambda\leq\frac{a^2}{2(a+1)}$, threshold value $t=t_2^*=\lambda\frac{a+1}{a}$.

To prove continuity as shown in Figure 4.1, the satisfaction of condition: $g_\lambda(t_2^*)=g_\lambda(-t_2^*)=0$ is sufficient.

According to formula (3.6), we substitute $x=\lambda\frac{a+1}{a}$ into function $\varphi(\cdot)$, then

$$\begin{aligned}\cos(\varphi) &= 1-\frac{27\lambda a(a+1)}{2(a+x)^3}\\ &= 1-\frac{27\lambda a(a+1)}{2(a+\lambda\frac{a+1}{a})^3}.\end{aligned}$$

(1) Firstly, consider $\lambda=\frac{a^2}{2(a+1)}$. Then $x=t_2^*=\frac{a}{2}$, so $\varphi=\arccos(-1)=\pi$. Thus $\cos(\varphi/3)=\frac{1}{2}$. By taking this into function g_λ, it is easy to check that $g_\lambda(t_2^*)=0$.

(2) Then, suppose $\lambda < \frac{a^2}{2(a+1)}$. In this case, $x = t_2^* > t_1^*$, so we have inequalities

$$-1 < d = \cos(\varphi) = 1 - \frac{27\lambda a(a+1)}{2(a+\lambda\frac{a+1}{a})^3} < 1.$$

From here, we know $\cos(\frac{\varphi}{3}) \in (\frac{1}{2}, 1)$.
Due to triple angle formula: $4\cos^3(\frac{\varphi}{3}) - 3\cos(\frac{\varphi}{3}) = \cos(\varphi) = d$, let us define a cubic polynomial $c(t) = 4t^3 - 3t - d$. Then we have: $c(-1) = -1 - d < 0$, $c(-1/2) = 1 - d > 0$, $c(1/2) = -1 - d < 0$ and $c(1) = 1 - d > 0$. So there exist three real roots for $c(t)$, and only one root is located in $(1/2, 1)$.
Further, we can check that $t^* = \dfrac{a - \frac{\lambda(a+1)}{2a}}{a + \frac{\lambda(a+1)}{a}}$ is a root of $c(t) = 0$ and also under the condition $\lambda < \frac{a^2}{2(a+1)}$, $\frac{1}{2} < t^* < 1$. From above discussion and triple angle formula, we can figure out that $\cos(\frac{\varphi}{3}) = \dfrac{a - \frac{\lambda(a+1)}{2a}}{a + \frac{\lambda(a+1)}{a}}$. Further, it is easy to check that $g_\lambda(t_2^*) = 0$.

REFERENCES

[1] T. Blumensath and M. Davies, *Iterative thresholding for sparse approximations*, J. Fourier Anal. and Appl., 14(5-6):629–654, 2008.
[2] T. Blumensath, *Accelerated iterative hard thresholding*, Signal Process., 92(3):752–756, 2012.
[3] E. Candès and T. Tao, *Decoding by linear programming*, IEEE Trans. Info. Theory, 51(12):4203–4215, 2005.
[4] E. Candès, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure Applied Math., 59(8):1207–1223, 2006.
[5] E. Candès, MB. Wakin, and SP. Boyd, *Enhancing sparsity by reweighted ℓ_1 minimization*, J. Fourier Anal. and Appl., 14(5-6):877–905, 2008.
[6] E. Candès and C. Fernandez-Granda, *Super-resolution from noisy data*, J. Fourier Anal. and Appl., 19(6):1229–1254, 2013.
[7] W. Cao, J. Sun, and Z. Xu, *Fast image deconvolution using closed-form thresholding formulas of regularization*, J. Vis. Commun. Image Represent., 24(1):31–41, 2013.
[8] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Applied Math., 57(11):1413–1457, 2004.
[9] D. Donoho, *Denoising by soft-thresholding*, IEEE Trans. Info. Theory, 41(3):613–627, 1995.
[10] D. Donoho, *Compressed sensing*, IEEE Trans. Info. Theory, 52(4):1289–1306, 2006.
[11] E. Esser, Y. Lou, and J. Xin, *A method for finding structured sparse solutions to non-negative least squares problems with applications*, SIAM J. Imaging Sci., 6:2010–2046, 2013.
[12] J. Fan and R. Li, *Variable selection via nonconcave penalized likelihood and its oracle properties*, J. Am. Stat. Assoc., 96(456):1348–1360, 2001.
[13] A. Fannjiang and W. Liao, *Coherence pattern-guided compressive sensing with unresolved grids*, SIAM J. Imaging Sci., 5(1):179–202, 2012.
[14] Y. Lou, P. Yin, Q. He, and J. Xin, *Computing sparse representation in a highly coherent dictionary based on difference of L1 and L2*, J. Sci. Computing, 1:178–196, 2015.
[15] J. Lv and Y. Fan, *A unified approach to model selection and sparse recovery using regularized least squares*, Ann. Stat., 37(6A):3498–3528, 2009.
[16] R. Mazumder, J. Friedman, and T. Hastie, *SparseNet: Coordinate descent with nonconvex penalties*, J. Am. Stat. Assoc., 106(495):1125–1138, 2011.
[17] N. Parikh and S.P. Boyd, *Proximal Algorithms*, Foundations and Trends in Optimization, Now Publishers Inc., 2014.
[18] Z. Xu, X. Chang, F. Xu, and H. Zhang, *$L_{1/2}$ regularization: A thresholding representation theory and a fast solver*, IEEE Trans. Neural Netw. Learn. Syst., 23(7):1013–1027, 2012.
[19] J. Yang and Y. Zhang, *Alternating direction algorithms for l_1 problems in compressive sensing*, SIAM J. Sci. Computing, 33(1):250–278, 2011.
[20] P. Yin, Y. Lou, Q. He, and J. Xin, *Minimization of L_{1-2} for compressed sensing*, SIAM J. Sci. Computing, 37(1):A536–A563, 2015.

[21] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for l_1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1(1):143–168, 2008.
[22] W. Yin and S. Osher, *Error forgetting of bregman iteration*, J. Sci. Computing, 54(2):684–695, 2013.
[23] J. Zeng, S. Lin, Y. Wang, and Z. Xu, *$L_{1/2}$ regularization: Convergence of iterative half thresholding algorithm*, IEEE Trans. Signal Process., 62(9):2317–2329, 2014.
[24] C. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, Ann. Stat., 38(2):894–942, 2010.
[25] S. Zhang and J. Xin, *Minimization of transformed L_1 penalty: theory, difference of convex function algorithm, and robust application in compressed sensing*, arXiv:1411.5735, 2014; CAM Report 14–68, UCLA.
[26] S. Zhang, P. Yin, and J. Xin, *Transformed Schatten-1 iterative thresholding algorithms for matrix rank minimization and applications*, arXiv:1506.04444, 2015.
[27] T. Zhang, *Multi-Stage Convex Relaxation for Learning with Sparse Regularization*, in Advances in Neural Information Processing Systems, MIT Press, 1929–1936, 2009.