UNIVERSITY OF CALIFORNIA,
IRVINE

Transformed L_1 Function, Sparse Optimization Algorithms and Applications

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mathematics

by

Shuai Zhang

Dissertation Committee:
Professor Jack Xin, Chair
Professor Long Chen
Professor Patrick Q. Guidotti

2017

DEDICATION

To my family, teachers, friends and collaborators.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I'd like to express my deepest gratitude to my advisor, Prof. Jack Xin, for continuous support of my Ph.D. study. I appreciate all his contributions of time, ideas, and funding to make my Ph.D. experience productive and stimulating. His advice on both my research and career have been invaluable. He has set an excellent example to me as a successful mathematician and professor.

I am thankful to Prof. Long Chen and Prof. Patrick Guidotti for serving as my committee members. I would like to thank Dr. Penghang Yin for helpful research discussions. I'd also like to thank Department of Mathematics at University of California, Irvine for its generous financial support during the past five years.

Last but not least, I am greatly indebted to my family members, especially my wife Weiwei Li, who has always been supportive of my pursuit of Ph.D. degree.

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

CURRICULUM VITAE

Shuai Zhang

EDUCATION

Doctor of Philosophy in Mathematics **2017**
University of California, Irvine *Irvine, CA*

Master of Science in Computational Mathematics **2012**
Shandong University *Shandong, China*

Bachelor of Science in Applied Mathematics **2009**
Shandong University *Shandong, China*

ABSTRACT OF THE DISSERTATION

Transformed L_1 Function, Sparse Optimization Algorithms and Applications

By

Shuai Zhang

Doctor of Philosophy in Mathematics

University of California, Irvine, 2017

Professor Jack Xin, Chair

A non-convex sparsity promoting penalty function, the transformed l_1 (TL1), is studied in optimization problems with its applications in compressed sensing (CS) and matrix completion. The TL1 penalty interpolates l_0 and l_1 norms through a nonnegative parameter $a \in (0, +\infty)$, similar to l_p with $p \in (0,1]$. TL1 is known in the statistics literature to enjoy three desired properties: unbiasedness, sparsity and Lipschitz continuity.

For compressed sensing problems, a RIP condition for TL1 exact recovery is proposed and proved. Next, difference of convex algorithms for TL1 (DCATL1) are presented in computing TL1-regularized constrained and unconstrained problems. For the unconstrained problem, we prove convergence of DCALT1 to a stationary point satisfying the first order optimality condition. In numerical experiments, we identify the optimal value $a = 1$, and compare DCATL1 with other CS algorithms. An explicit fixed point representation is also developed for the TL1 regularized minimization problem. The TL1 thresholding functions are in closed form for all parameter values. 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.

The TS1 penalty, as a matrix quasi-norm defined on its singular values, interpolates the rank and the nuclear norm through a nonnegative parameter $a \in (0, +\infty)$. We consider the unconstrained TS1 regularized low-rank matrix recovery problem and develop a fixed point representation for its global minimizer. The TS1 thresholding functions are in closed analytical form for all parameter values. We propose TS1 iterative thresholding algorithms and compare them with some state-of-the-art algorithms on matrix completion test problems.

Chapter 1

Introduction

1.1 Compressed Sensing

Compressed sensing [7, 21] has generated enormous interest and research activities in mathematics, statistics, signal processing, imaging and information sciences, among numerous other areas. One of the basic problems is to reconstruct a sparse signal under a few linear measurements (linear constraints) far less than the dimension of the ambient space of the signal. Consider a sparse signal $x \in \Re^N$, an $M \times N$ sensing matrix A and an observation $y \in \Re^M$, $M \ll N$, such that: $y = Ax + \epsilon$, where ϵ is an N-dimensional observation error. If x is sparse enough, it can be reconstructed exactly in the noise-free case and in stable manner in the noisy case provided that the sensing matrix A satisfies certain incoherence or the restricted isometry property (RIP) [7, 21].

The direct approach is l_0 optimization, including constrained formulation:

$$\min_{x \in \Re^N} \|x\|_0, \ s.t. \ \ y = Ax, \tag{1.1}$$

and the unconstrained l_0 regularized optimization:

$$\min_{x\in\Re^N}\{\|y-Ax\|_2^2+\lambda\|x\|_0\} \tag{1.2}$$

with positive regularization parameter λ. Since minimizing l_0 norm is NP-hard [48], many viable alternatives are available. Greedy methods (matching pursuit [45], othogonal matching pursuits (OMP) [60], and regularized OMP (ROMP) [49]) work well if the dimension N is not too large. For the unconstrained problem (1.2), the penalty decomposition method [41] replaces the term $\lambda\|x\|_0$ by $\rho_k\|x-z\|_2^2+\lambda\|z\|_0$, and minimizes over (x,z) for a diverging sequence ρ_k. The variable z allows the iterative hard thresholding procedure.

The relaxation approach is to replace l_0 norm $\|x\|_0$ by a continuous sparsity promoting penalty functions $P(x)$. Convex relaxation uniquely selects $P(\cdot)$ as the l_1 norm $\|x\|_1$. The resulting problems are known as basis pursuit (LASSO in the over-determined regime [59]). The l_1 algorithms include l_1-magic [7], Bregman and split Bregman methods [31, 67] and yall1 [65]. Theoretically, Candès and Tao introduced RIP condition and used it to establish the equivalent and unique global solution to l_0 minimization via l_1 relaxation among other stable recovery results [10, 8, 7].

There are also many choices of $P(\cdot)$ for non-convex relaxation. One is the l_p norm ($p\in(0,1)$) with l_0 equivalence under RIP [13]. The $l_{1/2}$ norm is representative of this class of functions, with the reweighted least squares and half-thresholding algorithms for computation [36, 64, 63]. Near the RIP regime, $l_{1/2}$ penalty tends to have higher success rate of sparse reconstruction than l_1. However, it is not as good as l_1 if the sensing matrix is far away from RIP [40, 66] as we shall see later as well. In the highly non-RIP (coherent) regime, it is recently found that the difference of l_1 and l_2 norm minimization gives the best sparse recovery results [66, 40]. It is therefore of both theoretical and practical interest to find a non-convex penalty that is consistently better than l_1 and always ranks among the top in sparse recovery whether the sensing matrix satisfies RIP or not.

In the statistics literature of variable selection, Fan and Li [24] advocated for classes of penalty functions with three desired properties: **unbiasedness, sparsity** and **continuity**. To help identify such a penalty function denoted by $\rho(\cdot)$, Fan and Lv [43] proposed the following condition for characterizing unbiasedness and sparsity promoting properties.

Condition 1. *The penalty function $\rho(\cdot)$ satisfies:*

(i) $\rho(t)$ *is increasing and concave in* $t \in [0,\infty)$*;*

(ii) $\rho'(t)$ *is continuous with* $\rho'(0+) \in (0,\infty)$*;*

(iii) if $\rho(t)$ *depends on a positive parameter* λ*, then* $\rho'(t;\lambda)$ *is increasing in* $\lambda \in (0,\infty)$ *and* $\rho'(0+)$ *is independent of* λ*.*

It follows that $\rho'(t)$ is positive and decreasing, and $\rho'(0+)$ is the upper bound of $\rho'(t)$. It is shown in [24] that penalties satisfying CONDITION 1 and $\lim_{t\to\infty}\rho^{'}(t)=0$ enjoy both unbiasedness and sparsity. Though continuity does not generally hold for this class of penalty functions, a special one parameter family of functions, the so called **transformed** l_1 **functions (TL1)** $\rho_a(t)$, where

$$\rho_a(t)=\frac{(a+1)|t|}{a+|t|},$$

with $a\in(0,+\infty)$, satisfies all three desired properties [24].

We shall study the minimization of TL1 functions for CS problems, in terms of theory, algorithms and computation. We proposed two classes of algorithms of TL1, which are difference of convex functions algorithm and thresholding algorithm.

1.2 Matrix Completion

Matrix rank minimization problems arise in many applications such as collaborative filtering in recommender systems [5, 33], minimum order system and low-dimensional Euclidean embedding in control theory [27, 28], network localization [34], and others [56]. The mathematical problem is:

$$\min_{X \in \Re^{m\times n}} \operatorname{rank}(X) \quad \text{s.t.} \quad X \in \text{Ł}, \tag{2.3}$$

where Ł is a convex set. In this paper, we are interested in methods for solving the affine rank minimization problem (ARMP)

$$\min_{X \in \Re^{m\times n}} \operatorname{rank}(X) \quad \text{s.t.} \quad \mathscr{A}(X) = b, \tag{2.4}$$

where X is the decision variable, and the linear transformation $\mathscr{A} : \Re^{m\times n} \to \Re^p$ and vector $b \in \Re^p$ are given. The matrix completion problem

$$\min_{X \in \Re^{m\times n}} \operatorname{rank}(X) \quad \text{s.t.} \quad X_{i,j} = M_{i,j}, \quad (i,j) \in \Omega \tag{2.5}$$

is a special case of (2.4), where X and M are both $m \times n$ matrices and Ω is a subset of index pairs (i,j).

The optimization problems above are known to be NP-hard. Many alternative penalties have been utilized as proxies for finding low rank solutions in both the constrained and unconstrained settings:

$$\min_{X \in \Re^{m\times n}} F(X) \quad \text{s.t.} \quad \mathscr{A}(X) = b \tag{2.6}$$

and

$$\min_{X\in\Re^{m\times n}} \frac{1}{2}\|\mathscr{A}(X)-b\|_2^2+\lambda F(X). \tag{2.7}$$

The penalty function $F(\cdot)$ is in terms of singular values of matrix X, typically $F(X)=\sum_i f(\sigma_i)$, where σ_i is the i-th largest singular value of X arranged in descending order. The Schatten p-norm (nuclear norm at $p=1$) results when $f(x)=x^p$, $p\in[0,1]$. At $p=0$ ($p=2$), F is the rank (Frobenius norm). Recovering rank under suitable conditions for $p\in(0,1]$ has been extensively studied in theories and algorithms [3, 9, 5, 35, 36, 42, 44, 47, 62]. Non-convex penalty based methods have shown better performance on hard problems [36, 47]. There is also a novel method to solve the constrained problem (2.6), from the perspective of gauge dual [29, 30].

Recently, a class of ℓ_1 based non-convex penalty, the transformed ℓ_1 (TL1), has been found effective and robust for compressed sensing problems [71, 72]. TL1 interpolates ℓ_0 and ℓ_1, similar to ℓ_p quasi-norm ($p\in(0,1)$). In the entire range of interpolation parameter, TL1 enjoys closed form iterative thresholding function, which is available for ℓ_p only at some specific values, like $p=0,1,1/2,2/3$, see [1, 12, 17, 64]. This feature allows TL1 to perform fast and robust sparse minimization in a much wider range than l_p quasi-norm. Moreover, the TL1 penalty boasts unbiasedness and Lipschitz continuity besides sparsity [24, 43].

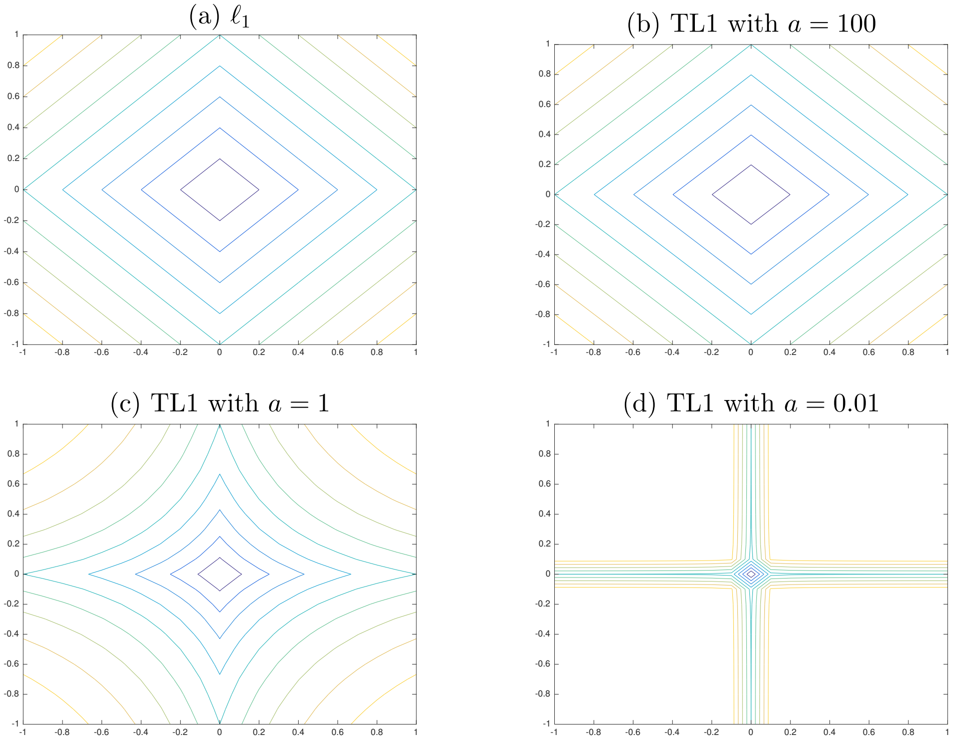
It is the goal of this paper to extend TL1 penalty to TS1 (transformed Schatten-1) for rank minimization and compare it with state of the art methods in the literature.

Chapter 2

Compressed Sensing

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 [7, 21] 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 [16, 20], 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 [67, 68] is similar for solving the constrained l_1 minimization (basis pursuit) problem. Recently, half and $\frac{2}{3}$-thesholding algorithms have been actively studied [12, 64] 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) [23, 66, 40], and the transformed l_1 (TL1) [72]. 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


(a) ℓ_1

(b) TL1 with $a = 100$

(c) TL1 with $a = 1$

(d) TL1 with $a = 0.01$

Figure 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.

matrices alike.

The TL1 penalty is a one parameter family of bilinear transformations composed with the absolute value 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 [24], MCP [70], log penalty

[46, 11], and capped l_1 [74]. We refer to Mazumder, Friedman and Hastie's paper [46] 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 [74]. 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 [43, 24].

In this chapter, we propose recovery theories and IT algorithms for TL1 regularized minimization with evaluation on CS test problems.

The TL1 penalty function $\rho_a(x)$ [43] is defined as

$$\rho_a(x) = \frac{(a+1)|x|}{a+|x|}, \tag{0.1}$$

where the parameter $a \in (0, +\infty)$. It interpolates the l_0 and l_1 norms as

$$\lim_{a \to 0^+} \rho_a(x) = \chi_{\{x \neq 0\}} \ and \ \lim_{a \to \infty} \rho_a(x) = |x|.$$

In Fig. (2.1), we compare level lines of l_1 and TL1 with different parameter $'a'$. With the adjustment of parameter $'a'$, the TL1 can approximate both l_1 and l_0 well. The TL1 function is Lipschitz continuous and satisfies Condition 1, thus enjoying the unbiasedness, sparsity and continuity properties [43].

Let us define TL1 regularization term $P_a(\cdot)$ as

$$P_a(x) = \sum_{i=1,...N} \rho_a(x_i), \tag{0.2}$$

In the following, we consider the constrained TL1 minimization model

$$\min_{x\in\Re^N} f(x) = \min_{x\in\Re^N} P_a(x) \ s.t. \ \ Ax = y, \tag{0.3}$$

and the unconstrained TL1-regularized model

$$\min_{x\in\Re^N} f(x) = \min_{x\in\Re^N} \frac{1}{2}\|Ax-y\|_2^2 + \lambda P_a(x). \tag{0.4}$$

2.1 TL1 RIP and Stable Recovery

Lemma 2.1.1. *For $a \geq 0$, any x_i and x_j in $\Re$, the following inequalities hold:*

$$\rho_a(|x_i+x_j|) \leq \rho_a(|x_i|+|x_j|) \leq \rho_a(|x_i|) + \rho_a(|x_j|) \leq 2\rho_a(\frac{|x_i|+|x_j|}{2}). \tag{1.5}$$

Proof. Let us prove these inequalities one by one, starting from the left.

1.) According to Condition 1, we know that $\rho_a(|t|)$ is increasing in the variable $|t|$. By triangle inequality $|x_i+x_j| \leq |x_i|+|x_j|$, we have:

$$\rho_a(|x_i+x_j|) \leq \rho_a(|x_i|+|x_j|).$$

2.)

$$\begin{aligned}
\rho_a(|x_i|)+\rho_a(|x_j|) &= \frac{(a+1)|x_i|}{a+|x_i|}+\frac{(a+1)|x_j|}{a+|x_j|} \\
&= \frac{a(a+1)(|x_i|+|x_j|+2|x_ix_j|/a)}{a(a+|x_i|+|x_j|+|x_ix_j|/a)} \\
&\geq \frac{(a+1)(|x_i|+|x_j|+|x_ix_j|/a)}{(a+|x_i|+|x_j|+|x_ix_j|/a)} \\
&= \rho_a(|x_i|+|x_j|+|x_ix_j|/a) \\
&\geq \rho_a(|x_i|+|x_j|).
\end{aligned}$$

3.) By concavity of the function $\rho_a(\cdot)$,

$$\frac{\rho_a(|x_i|)+\rho_a(|x_j|)}{2} \leq \rho_a\left(\frac{|x_i|+|x_j|}{2}\right).$$

□

Remark 2.1.1. *It follows from Lemma 2.1.1 that the triangular inequality holds for the function $\rho(x)\equiv\rho_a(|x|)$: $\rho(x_i+x_j)=\rho_a(|x_i+x_j|)\leq\rho_a(|x_i|)+\rho_a(|x_j|)=\rho(x_i)+\rho(x_j)$. Also we have: $\rho(x)\geq 0$, and $\rho(x)=0\Leftrightarrow x=0$. Our penalty function ρ acts almost like a norm. However, it lacks absolute scalability, or $\rho(cx)\neq|c|\rho(x)$ in general. The next lemma further analyses this inequality.*

Lemma 2.1.2.

$$\rho_a(|cx|)=\begin{cases}\leq|c|\rho_a(|x|) \text{ if } |c|>1; \\ \geq|c|\rho_a(|x|) \text{ if } |c|\leq 1.\end{cases} \tag{1.6}$$

Proof.

$$\begin{aligned}\rho_a(|cx|) &= \frac{(a+1)|c||x|}{a+|c||x|} \\ &= |c|\rho_a(|x|)\ \frac{a+|x|}{a+|cx|}.\end{aligned}$$

So if $|c|\leq 1$, the factor $\frac{a+|x|}{a+|cx|}\geq 1$. Then $\rho_a(|cx|)\geq |c|\rho_a(|x|)$. Similarly when $|c|>1$, we have $\rho_a(|cx|)\leq |c|\rho_a(|x|)$. □

2.1.1 RIP Condition for Constrained Model

For the constrained TL1 model (0.3), we present a theory on sparse recovery based on RIP [8]. Suppose β^0 is a sparsest solution for l_0 minimization s.t. $A\beta^0=y$, while another vector β is defined as

$$\beta = arg \min_{x\in\Re_N} \{P_a(x)|\quad Ax=y\}. \tag{1.7}$$

We addressed the question whether the two vectors β and β^0 are equal to each other. That is to say, under what condition we can recover the sparsest solution β^0 via solving the relaxation problem (0.3).

For an $M\times N$ matrix A and set $T\subset\{1,...,N\}$, let A_T be the matrix consisting of the column a_j of A for $j\in T$. Similarly for vector x, x_T is a sub-vector, consisting of components indexed from the set T.

Definition 2.1.1. *(Restricted Isometry Constant) For each number s, define the s-restricted isometry constant of matrix A as the smallest number δ_s such that for all subset T with $|T|\leq s$ and all $x\in\Re_{|T|}$, the inequality*

$$(1-\delta_s)\|x\|_2^2\leq\|A_Tx\|_2^2\leq(1+\delta_s)\|x\|_2^2$$

holds.

For a fixed y, the under-determined linear system has infinitely many solutions. Let x be one solution of $Ax=y$. It does not need to be the l_0 or ρ_a minimizer. If $P_a(x)>1$, we scale y by the positive scalar C as:

$$y_C=\frac{y}{C}; \quad x_C=\frac{x}{C}. \tag{1.8}$$

Now x_C is a solution to the modified problem: $Ax_C=y_C$. When C becomes larger, the number $P_a(x_C)$ is smaller and tends to 0 in the limit $C\to\infty$. Thus, we can find a constant $C\geq 1$, such that $P_a(x_C)\leq 1$. That is to say, for scaled vector x_C, we always have: $P_a(x_C)\leq 1$.

Since the penalty $\rho_a(t)$ is increasing in positive variable t, we have the inequality:

$$\begin{aligned} P_a(x_C) &\leq |T|\rho_a(|x_C|_\infty) \\ &= |T|\rho_a(\tfrac{|x|_\infty}{C}) \\ &= \frac{|T|(a+1)|x|_\infty}{aC+|x|_\infty}, \end{aligned}$$

where $|T|$ is the cardinality of the support set of vector x. For $P_a(x_C)\leq 1$, it suffices to impose:

$$\frac{|T|(a+1)|x|_\infty}{aC+|x|_\infty}\leq 1,$$

or:

$$C\geq \frac{|x|_\infty}{a}\left(a|T|+|T|-1\right). \tag{1.9}$$

Let β^0 be the l_0 minimizer for the constrained l_0 optimization problem (1.1) with support set

T. Due to the scale-invariance of l_0, β_C^0 (defined similarly as above) is a global l_0 minimizer for the modified problem:

$$\min_x \ \|x\|_0, \ s.t. \ y_C = Ax. \tag{1.10}$$

with the same support set T.

Then for the modified ρ_a optimization:

$$\min_x \ P_a(x), \ s.t. \ y_C = Ax, \tag{1.11}$$

we have the following RIP condition.

Theorem 2.1.1. *(TL1 Exact Sparse Recovery) For a given sensing matrix A, β_C^0 is the minimizer of (1.10), with C satisfying (1.9). T is the support set of β_C^0, with cardinality $|T|$. Suppose there is a number $R > |T|$, $b = (\frac{a}{a+1})^2 \frac{R}{|T|}$, such that*

$$\delta_R + b\delta_{R+|T|} < b - 1, \tag{1.12}$$

then the minimizer β_C for (1.11) is unique and equal to the minimizer β_C^0 in (1.10).

Proof. The proof generally follows the lines of arguments in [8] and [13], while using special properties of the penalty function ρ_a.
For simplicity, we denote β_C by β and β_C^0 by β^0.

Let $e = \beta - \beta^0$, and we want to prove that the vector $e = 0$. It is clear that, $e_{T^c} = \beta_{T^c}$, since T is the support set of β^0. By the triangular inequality of ρ_a, we have:

$$P_a(\beta^0) - P_a(e_T) = P_a(\beta^0) - P_a(-e_T) \leq P_a(\beta_T).$$

Then

$$\begin{aligned}
P_a(\beta^0) - P_a(e_T) + P_a(e_{T^c}) &\leq P_a(\beta_T) + P_a(\beta_{T^c}) \\
&= P_a(\beta) \\
&\leq P_a(\beta^0)
\end{aligned}$$

It follows that:

$$P_a(\beta_{T^c}) = P_a(e_{T^c}) \leq P_a(e_T). \tag{1.13}$$

Now let us arrange the components at T^c in the order of decreasing magnitude of $|e|$ and partition into L parts: $T^c = T_1 \cup T_2 \cup ... \cup T_L$, where each T_j has R elements (except possibly T_L with less). Also denote $T = T_0$ and $T_{01} = T \cup T_1$. Since $Ae = A(\beta - \beta^0) = 0$, it follows that

$$\begin{aligned}
0 &= \|Ae\|_2 \\
&= \|A_{T_{01}} e_{T_{01}} + \sum_{j=2}^{L} A_{T_j} e_{T_j}\|_2 \\
&\geq \|A_{T_{01}} e_{T_{01}}\|_2 - \sum_{j=2}^{L} \|A_{T_j} e_{T_j}\|_2 \\
&\geq \sqrt{1 - \delta_{|T|+R}} \|e_{T_{01}}\|_2 - \sqrt{1 + \delta_R} \sum_{j=2}^{L} \|e_{T_j}\|_2
\end{aligned} \tag{1.14}$$

At the next step, we derive two inequalities between the l_2 norm and function P_a, in order to use the inequality (1.13). Since

$$\begin{aligned}
\rho_a(|t|) = \frac{(a+1)|t|}{a+|t|} &\leq (\frac{a+1}{a})|t| \\
&= (1 + \frac{1}{a})|t|
\end{aligned}$$

we have:

$$\begin{aligned}P_a(e_{T_0}) = \sum_{i\in T_0} \rho_a(|e_i|)\\ &\leq (1+\tfrac{1}{a})\|e_{T_0}\|_1\\ &\leq (1+\tfrac{1}{a})\sqrt{|T|}\ \ \|e_{T_0}\|_2\\ &\leq (1+\tfrac{1}{a})\sqrt{|T|}\ \ \|e_{T_{01}}\|_2.\end{aligned} \tag{1.15}$$

Now we estimate the l_2 norm of e_{T_j} from above in terms of P_a. It follows from β being the minimizer of the problem (1.11) and the definition of x_C (1.8) that

$$P_a(\beta_{T^c}) \leq P_a(\beta) \leq P_a(x_C) \leq 1.$$

For each $i\in T^c$, $\rho_a(\beta_i) \leq P_a(\beta_{T^c}) \leq 1$. Also since

$$\begin{aligned}&\frac{(a+1)|\beta_i|}{a+|\beta_i|} \leq 1\\ &\Leftrightarrow (a+1)|\beta_i| \leq a+|\beta_i|\\ &\Leftrightarrow |\beta_i| \leq 1\end{aligned} \tag{1.16}$$

we have

$$|e_i| = |\beta_i| \leq \frac{(a+1)|\beta_i|}{a+|\beta_i|} = \rho_a(|\beta_i|) \quad \text{for every } i\in T^c.$$

It is known that function $\rho_a(t)$ is increasing for non-negative variable $t\geq 0$, and

$$|e_i| \leq |e_k| \ \ for\ \ \forall\ i\in T_j \ \ and\ \ \forall\ k\in T_{j-1}$$

,where $j=2,3,...,L$. Thus we will have

$$\begin{aligned} &|e_i| \leq \rho_a(|e_i|) \leq P_a(e_{T_{j-1}})/R \\ \Rightarrow\ &\|e_{T_j}\|_2^2 \leq \frac{P_a(e_{T_{j-1}})^2}{R} \\ \Rightarrow\ &\|e_{T_j}\|_2 \leq \frac{P_a(e_{T_{j-1}})}{R^{1/2}} \\ \Rightarrow\ &\sum_{j=2}^{L} \|e_{T_j}\|_2 \leq \sum_{j=1}^{L} \frac{P_a(e_{T_j})}{R^{1/2}} \end{aligned} \tag{1.17}$$

Finally, plug (1.15) and (1.17) into inequality (1.14) to get:

$$\begin{aligned} 0 &\geq \sqrt{1-\delta_{|T|+R}}\frac{a}{(a+1)|T|^{1/2}}P_a(e_T) - \sqrt{1+\delta_R}\frac{1}{R^{1/2}}P_a(e_T) \\ &\geq \frac{P_a(e_T)}{R^{1/2}}\left(\sqrt{1-\delta_{R+|T|}}\frac{a}{a+1}\sqrt{\frac{R}{|T|}} - \sqrt{1+\delta_R}\right) \end{aligned} \tag{1.18}$$

Derived from the given RIP condition (1.12), factor $\sqrt{1-\delta_{R+|T|}}\frac{a}{a+1}\sqrt{\frac{R}{|T|}} - \sqrt{1+\delta_R}$ is strictly positive, hence $P_a(e_T)=0$, and $e_T=0$. Also by inequality (1.13), $e_{T^c}=0$. We have proved that $\beta_C=\beta_C^0$. The equivalence of (1.11) and (1.10) holds. If there is another vector β is the optimal solution of (1.11), we can prove that it is also equal to β_C^0, using the same procedure. Hence β_C is unique.

□

Remark 2.1.2. *In Theorem 2.1.1, if we choose $R=3|T|$, RIP condition (1.12) is*

$$\delta_{3|T|} + 3\frac{a^2}{(a+1)^2}\delta_{4|T|} < 3\frac{a^2}{(a+1)^2} - 1.$$

This inequality will approach $\delta_{3|T|}+3\delta_{4|T|}<2$ as parameter a goes to $+\infty$, which is the RIP condition proposed in [8]. Similarly, it can be proved as [8], that if matrix $A \in \Re^{m\times n}$ is *sampled from i.i.d univariate Gaussian distribution, RIP condition (1.12) will be satisfied with overwhelming probability for large enough a and small ratio* $\frac{|T|}{m}$.

Next, we prove that TL1 recovery is stable under noisy measurements, i.e.,

$$\min \quad P_a(\beta), \ s.t. \ \|y_C - A\beta\|_2 \leq \tau. \tag{1.19}$$

Theorem 2.1.2. *(Stable Recovery Theory)*

Under the same RIP condition in theorem 2.1.1, the solution β_C^n *for optimization (1.19) satisfies*

$$\|\beta_C^n - \beta_C^0\|_2 \leq D\tau,$$

for some constant D depending only on the RIP condition.

Proof. Set $n = A\beta - y_C$. In the proof, we use three related notations listed below for clarity:

(i) $\beta_C^n \Rightarrow$ optimal solution for the noisy constrained problem (1.19);

(ii) $\beta_C \Rightarrow$ optimal solution for the noiseless constrained problem (1.11);

(iii) $\beta_C^0 \Rightarrow$ optimal solution for the l_0 problem (1.10).

Let T be the support set of β_C^0, i.e., $T = supp(\beta_C^0)$, and vector $e = \beta_C^n - \beta_C^0$. Following the proof of Theorem 2.1.1, we obtain:

$$\sum_{j=2}^{L} \|e_{T_j}\|_2 \leq \sum_{j=1}^{L} \frac{P_a(e_{T_j})}{R^{1/2}} = \frac{P_a(e_{T^c})}{R^{1/2}}$$

and

$$\|e_{T_{01}}\|_2 \geq \frac{a}{(a+1)\sqrt{|T|}} P_a(e_T).$$

Further, due to the inequality $P_a(\beta^n_{T^c}) = P_a(e_{T^c}) \leq P_a(e_T)$ and inequality (1.14), we get

$$\|Ae\|_2 \geq \frac{P_a(e_T)}{R^{1/2}} C_\delta,$$

where $C_\delta = \sqrt{1-\delta_{R+|T|}}\frac{a}{a+1}\sqrt{\frac{R}{|T|}} - \sqrt{1+\delta_R}$.

By the initial assumption on the size of observation noise, we have

$$\|Ae\|_2 = \|A\beta^n_C - A\beta^0_C\|_2 = \|n\|_2 \leq \tau, \tag{1.20}$$

so we have: $P_a(e_T) \leq \frac{\tau R_{1/2}}{C_\delta}$.

On the other hand, we know that $P_a(\beta_C) \leq 1$ and β_C is in the feasible set of the noisy problem (1.19). Thus we have the inequality: $P_a(\beta^n_C) \leq P_a(\beta_C) \leq 1$. By (1.16), $\beta^n_{C,i} \leq 1$ for each i. So, we have

$$|\beta^n_{C,i}| \leq \rho_a(|\beta^n_{C,i}|). \tag{1.21}$$

It follows that

$$\begin{aligned}
\|e\|_2 &\leq \|e_T\|_2 + \|e_{T^c}\|_2 = \|e_T\|_2 + \|\beta^n_{C,T^c}\|_2 \\
&\leq \frac{\|A_T e_T\|_2}{\sqrt{1-\delta_T}} + \|\beta^n_{C,T^c}\|_1 \\
&\leq \frac{\|A_T e_T\|_2}{\sqrt{1-\delta_T}} + P_a(\beta^n_{C,T^c}) = \frac{\|A_T e_T\|_2}{\sqrt{1-\delta_T}} + P_a(e_{T^c}) \\
&\leq \frac{\tau}{\sqrt{1-\delta_R}} + P_a(e_T) \leq D\tau.
\end{aligned}$$

where constant number D depends on δ_R and $\delta_{R+|T|}$. The second inequality uses the definition of RIP, while the first inequality in the last row comes from (1.20). □

2.1.2 Sparsity of Local Minimizer

We study properties of local minimizers of both the constrained problem (0.3) and the unconstrained model (0.4). As in l_p and l_{1-2} minimization [66, 40], a local minimizer of TL1 minimization extracts linearly independent columns from the sensing matrix A, with no requirement for A to satisfy RIP. Reversely, we state additional conditions on A for a stationary point to be a local minimizer besides the linear independence of the corresponding column vectors.

Theorem 2.1.3. *(Local minimizer of constrained model)*

Suppose x^* is a local minimizer of the constrained problem (0.3) and $T^* = supp(x^*)$, then A_{T^*} is of full column rank, i.e. columns of A_{T^*} are linearly independent.

Proof. Here we argue by contradiction. Suppose that the column vectors of A_{T^*} are not linearly independent, then there exists non-zero vector $v \in ker(A)$, such that $supp(v) \subseteq T^*$. For any neighbourhood of x^*, $N(x^*, r)$, we can scale v so that:

$$\|v\|_2 \leq \min\{r; \quad |x_i^*|, i \in T^*.\} \tag{1.22}$$

Next we define:

$$\xi_1 = x^* + v;$$
$$\xi_2 = x^* - v,$$

so both ξ_1 and ξ_2, $\in \mathscr{B}(x^*, r)$, and $x^* = \frac{1}{2}(\xi_1 + \xi_2)$. On the other hand, from $supp(v) \subseteq T^*$, we have that $supp(\xi_1), supp(\xi_2) \subseteq T^*$. Moreover, due to the inequality (1.22), vectors x^*, x_1, and x_2 are located in the same orthant, i.e. $sign(x^*_i) = sign(\xi_{1,i}) = sign(\xi_{2,i})$, for any index i. It means that $\frac{1}{2}|\xi_1| + \frac{1}{2}|\xi_2| = \frac{1}{2}|\xi_1 + \xi_2|$. Since the penalty function $P_a(t)$ is strictly concave for non-negative variable t,

$$\begin{aligned}\tfrac{1}{2}P_a(\xi_1) + \tfrac{1}{2}P_a(\xi_2) &= \tfrac{1}{2}P_a(|\xi_1|) + \tfrac{1}{2}P_a(|\xi_2|) \\ &< P_a(\tfrac{1}{2}|\xi_1| + \tfrac{1}{2}|\xi_2|) = P_a(\tfrac{1}{2}|\xi_1 + \xi_2|) = P_a(x^*).\end{aligned}$$

So for any fixed r, we can find two vectors ξ_1 and ξ_2 in the neighbourhood $\mathscr{B}(x^*, r)$, such that $\min\{P_a(\xi_1), P_a(\xi_2)\} \leq \frac{1}{2}P_a(\xi_1) + \frac{1}{2}P_a(\xi_2) < P_a(x^*)$. Both vectors are in the feasible set of the constrained problem (0.3), in contradiction with the assumption that x^* is a local minimizer. □

The same property also holds for the local minimizers of unconstrained model (0.4), because a local minimizer of the unconstrained problem is also a local minimizer for a constrained optimization model [8, 66]. We skip the details and state the result below.

Theorem 2.1.4. *(Local minimizer of unconstrained model)*

Suppose x^* is a local minimizer of the unconstrained problem (0.4) and $T^* = supp(x^*)$, then columns of A_{T^*} are linearly independent.

Remark 2.1.3. *From the two theorems above, we conclude the following facts:*

(i) For any local minimizer of (0.3) or (0.4), e.g. x^*, the sparsity of x^* is at most rank(A);

(ii) The number of local minimizers is finite, for both problem (0.3) and (0.4).

In [43], the authors studied sufficient conditions of a strict local minimizer for minimizing any penalty functions satisfying Condition 1. Here we specialize and simplify it for our concave TL1 function ρ_a.

For a convex function $h(\cdot)$, the subdifferential $\partial h(x)$ at $x \in dom h$ is the closed convex set:

$$\partial h(x) := \{y \in \Re^N : h(z) \geq h(x) + \langle z - x, y \rangle, \ \ \forall z \in \Re^N\}, \tag{1.23}$$

which generalizes the derivative in the sense that h is differentiable at x if and only if $\partial h(x)$ is a singleton or $\{\nabla h(x)\}$.

The TL1 penalty function $p_a(\cdot)$ can be written as a difference of two convex functions:

$$\begin{aligned} \rho_a(t) &= \frac{(a+1)|t|}{a+|t|} \\ &= \frac{(a+1)|t|}{a} - \left(\frac{(a+1)|t|}{a} - \frac{(a+1)|t|}{a+|t|} \right) \\ &= \frac{(a+1)|t|}{a} - \frac{(a+1)t^2}{a(a+|t|)}. \end{aligned} \tag{1.24}$$

Thus we can define the general derivative of function $P_a(\cdot)$, as the difference of two convex derivatives,

$$\partial P_a(x) = \frac{a+1}{a} \partial \|x\|_1 - \partial \varphi_a(x), \tag{1.25}$$

where $\partial \|x\|_1$ is the subdifferential of $\|x\|_1$ and

$$\varphi_a(x) = \frac{a+1}{a} \|x\|_1 - P_a(x) = \sum_{i=1}^{N} \frac{(a+1)|x_i|^2}{a(a+|x_i|)}, \tag{1.26}$$

which is differentiable. As we know, $\partial \|x\|_1 = \{sgn(x_i)\}_{i=1,\ldots,N}$, where

$$sgn(t) = \begin{cases} sign(t), & \text{if} \ \ t \neq 0, \\ [-1,1], & \text{otherwise}. \end{cases} \tag{1.27}$$

Definition 2.1.2. *(Maximum concavity and local concavity of the penalty function)*

For a penalty function ρ, we define its maximum concavity as:

$$\kappa(\rho) = \sup_{t_1,t_2\in(0,\infty),t_1<t_2} -\frac{\rho^{'}(t_2)-\rho^{'}(t_1)}{t_2-t_1} \tag{1.28}$$

and its local concavity of ρ at a point $b=(b_1,b_2,...,b_R)^t \in \Re^R$ with $\|b\|_0 = R$ as:

$$\kappa(\rho;b) = \lim_{\epsilon\to 0+} \max_{1\le j\le R} \sup_{t_1,t_2\in(|b_j|-\epsilon,|b_j|+\epsilon),t_1<t_2} -\frac{\rho^{'}(t_2)-\rho^{'}(t_1)}{t_2-t_1}. \tag{1.29}$$

In [43], Lv and Fan proposed a set of sufficient conditions for the (strict) local minimizer of (0.4).

Condition 2. *For vector $\beta \in \Re^N$, $\lambda > 0$ with the support set $T = supp(\beta)$:*

(i) Matrix $Q = A_T^t A_T$ is non-singular, i.e. matrix A_T is column independent;

(ii) For vector $z = \frac{1}{\lambda}A^t(y - A\beta)$, $\|z_{T^c}\|_\infty < \rho_a^{'}(0+) = \frac{a+1}{a}$;

(iii) Vector β_T satisfies the stationary point equation: $\beta_T = Q^{-1}A_T^t y - \lambda Q^{-1}\partial P_a(\beta_T)$. Here β_T are all non-zero, so $\partial P_a(\beta)$ is well-defined and determined.

(iv) $\lambda_{min}(Q) \ge \lambda\kappa(\rho_a;\beta_T)$, where $\lambda_{min}(\cdot)$ denotes the smallest eigenvalues of a given symmetric matrix.

Since T is the support set of vector β, function $\rho_a(t)$ is twice differentiable at each element β_j, $\forall j \in T$. Thus we can explicitly write the formula of $\kappa(\rho_a;\beta_T)$, which is equal to

$$\begin{aligned}\kappa(\rho_a;\beta_T) &= \max_{j\in T}\{-\rho_a^{''}(\beta_j)\}\\ &= \max_{j\in T}\frac{2a(a+1)}{(a+|\beta_j|)^3}.\end{aligned}$$

It is convenient to define

$$\beta_{min} = \min_{j \in T} |\beta_j|, \tag{1.30}$$

so $\kappa(\rho_a;\beta_T) = \dfrac{2a(a+1)}{(a+|\beta_{min}|)^3}$.

Here we present the theory and give a simplified proof to illustrate (i)-(iv) in Condition 2.

Theorem 2.1.5. *If a vector $\beta \in \Re^N$ satisfies all four requirements in Condition 2, then β is a local minimizer of problem (0.4).*
Furthermore, if the inequality of (iv) in Condition 2 is strict, then the vector β is a strict local minimizer.

Proof. Let us define a subspace of $\Re^N$ as: $S = \{x \in \Re^N | x_{T^c} = 0\}$ and denote the optimal objective function as

$$\ell(x) = 2^{-1}\|Ax-y\|_2^2 + \lambda P_a(x) = 2^{-1}\|Ax-y\|_2^2 + \lambda \sum_{j=1}^{N} \rho_a(x_j).$$

First, the objective function $\ell(\cdot)$ is convex in ball area $\mathscr{B}(\beta, r_0) \cap S$, where r_0 is a positive number (the radius) and $r_0 < \beta_{min}$, defined as (1.30). This is because in convex area $\mathscr{B}(\beta, r_0) \cap S$, function $\ell(x)$ is twice differentiable and also its Hessian matrix of second partial derivatives is positive semidefinite due to Condition 1, (iv) of Condition 2, and the definition of $\kappa(\rho_a;\beta_T)$.

By equation (iii) in Condition 2, β_T is a local minimizer of $\ell(\cdot)$ in S. Next, we show that the sparse vector β is indeed a local minimizer of $\ell(x)$ in $\Re^N$. Because of the inequality (ii) in Condition 2, there exists a $\delta \in (0,\infty)$ and a positive number $r_1 < \delta$, such that

$$\|w(x)_{T^c}\|_\infty < \rho_a^{'}(\delta) \le \rho_a^{'}(0+),$$

for any vector $x \in \mathscr{B}(\beta, r_1)$ and $w(x) = \frac{1}{\lambda} A^t(y - Ax)$. We can further shrink r_1 if necessary so that $r_1 < r_0$, and then $\mathscr{B}(\beta, r_1) \subseteq \mathscr{B}(\beta, r_0)$.

$\forall \beta_1 \in \mathscr{B}(\beta, r_1)$, and define β_2 as the projection of β_1 onto set S. Then each related element pairs from β_1 and β_2 sits at the same side of 1-dimensional x-axis, where $\partial P_a(x)$ is well defined for x lying between β_1 and β_2. Thus we have

$$\ell(\beta_1) = \ell(\beta_2) + \bigtriangledown^t \ell(\beta_0)(\beta_1 - \beta_2),$$

where β_0 lies on the line segment joining β_1 and β_2.

Furthermore, it is easy to derive these facts

$$(\beta_1 - \beta_2)_T = 0, \quad \beta_0 \in \mathscr{B}(\beta, r_1) \quad and \quad sign(\beta_{0,T^c}) = sign(\beta_{1,T^c}).$$

Thus if vector $\beta_1 \notin S$,

$$\begin{aligned}
\ell(\beta_1) - \ell(\beta_2) &= \partial \ell(\beta_0)_{T^c} * \beta_{1,T^c} \\
&= -\lambda [\lambda^{-1} A^t_{T^c}(y - A\beta_0)]^t \beta_{1,T^c} + \lambda \sum_{j \in T^c} \rho_a^{'}(\beta_{0,j}) \beta_{1,j} \\
&> -\lambda \rho_a^{'}(\delta) \|\beta_{1,T^c}\|_1 + \lambda \sum_{j \in T^c} \rho_a^{'}(|\beta_{0,j}|) |\beta_{1,j}| \\
&\geq -\lambda \rho_a^{'}(\delta) \|\beta_{1,T^c}\|_1 + \lambda \rho_a^{'}(\delta) \|\beta_{1,T^c}\|_1 = 0,
\end{aligned}$$

where $*$ stands for vector cross product and we also used the fact for $j \in T^c$, $|\beta_{0,j}| \leq \delta$.

Since β_2 is a projection on S and it belongs to the ball $\mathscr{B}(\beta, r_1) \subseteq \mathscr{B}(\beta, r_0)$, we will have

$$\begin{cases} \ell(\beta_1) > \ell(\beta_2) \geq \ell(\beta), & \text{if } \beta_1 \notin S; \\ \ell(\beta_1) \geq \ell(\beta), & \text{if } \beta_1 \in S. \end{cases} \tag{1.31}$$

The (iv) in Condition 2 is only used in the first part of the proof. If we has the strict inequality $\lambda_{min}(Q) > \lambda\kappa(\rho_a;\beta_T)$, then β_T is a strict local minimizer in S, as the function $\ell(\cdot)$ is strictly convex in the intersection $\mathscr{B}(\beta, r_0) \cap S$ and the first inequality of (1.31). Further, the same proof shows that β is a strict local minimizer in $\Re^N$.

□

2.2 DCATL1

DC (Difference of Convex functions) Programming and DCA (DC Algorithms) was introduced in 1985 by Pham Dinh Tao, and extensively developed by Le Thi Hoai An and Pham Dinh Tao to become a useful tool for non-convex optimization ([52, 37] and references therein).

A standard DC program is of the form

$$\alpha = \inf\{f(x) = g(x) - h(x) : x \in \Re^n\} \qquad (P_{dc}),$$

where g, h are lower semicontinuous proper convex functions on $\Re^n$. Here f is called a DC function, while $g - h$ is a DC decomposition of f.

The DCA is an iterative method and generates a sequence $\{x^k\}$. For example, at the current point x^l of iteration, function $h(x)$ is approximated by its affine minorization $h_l(x)$, defined by

$$h_l(x) = h(x^l) + \langle x - x^l, y^l \rangle, \quad y^l \in \partial h(x^l).$$

Then the original model is converted to solve a convex program in the form:

$$\inf\{g(x)-h_l(x):x\in\Re^d\}\Leftrightarrow\inf\{g(x)-\langle x,y^l\rangle:x\in\Re^d\},$$

where the optimal solution is denoted as x^{l+1}.

2.2.1 Algorithm for Unconstrained Model — DCATL1

For the following unconstrained optimization problem (0.4):

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

we propose a DC decomposition scheme $f(x)=g(x)-h(x)$, where

$$\begin{cases} g(x) = \frac{1}{2}\|Ax-y\|_2^2+c\|x\|_2^2+\lambda\frac{(a+1)}{a}\|x\|_1; \\ h(x) = \lambda\varphi_a(x)+c\|x\|_2^2. \end{cases} \tag{2.32}$$

Here function $\varphi_a(x)$ is defined in equation (1.26). Thus function $h(x)$ is differentiable. Additional factor $c\|x\|_2^2$ with hyperparameter c is used to improve the convexity of these two functions, and will be used in the convergence theorem.

Algorithm 1: DCA for unconstrained transformed $l1$ penalty minimization

```
Define:   $\epsilon_{outer}>0$
Initialize:   $x^0=0,n=0$
while $|x^{n+1}-x^n|>\epsilon_{outer}$ do
    $v^n=\partial h(x^n)=\lambda\partial\varphi_a(x^n)+2cx^n$
    $x^{n+1}=arg\min_{x\in\Re^N}\{\frac{1}{2}\|Ax-y\|_2^2+c\|x\|^2+\lambda\frac{(a+1)}{a}\|x\|_1-\langle x,v^n\rangle\}$
    then $n+1\rightarrow n$
end while
```

At each step, we need to solve a strongly convex l_1-regularized sub-problem, which is:

$$\begin{aligned} x^{n+1} &= arg\min_{x\in\Re^N}\{\tfrac{1}{2}\|Ax-y\|_2^2+c\|x\|^2+\lambda\frac{(a+1)}{a}\|x\|_1-\langle x,v^n\rangle\} \\ &= arg\min_{x\in\Re^N}\{\tfrac{1}{2}x^t(A^tA+2cI)x-\langle x,v^n+A^ty\rangle\ +\lambda\frac{(a+1)}{a}\|x\|_1\}. \end{aligned} \tag{2.33}$$

We now employ the Alternating Direction Method of Multipliers (ADMM). After introduction a new variable z, the sub-problem is recast as:

$$\begin{aligned} \min_{x,z\in\Re^N}\{\ \ \tfrac{1}{2}x^t(A^tA+2cI)x-\langle x,v^n+A^ty\rangle\ +\lambda\frac{(a+1)}{a}\|z\|_1\} \\ s.t.\ x-z=0. \end{aligned} \tag{2.34}$$

Define the augmented Lagrangian function as:

$$L(x,z,u)=\frac{1}{2}x^t(A^tA+2cI)x-\langle x,v^n+A^ty\rangle\ +\lambda\frac{(a+1)}{a}\|z\|_1+\frac{\delta}{2}\|x-z\|_2^2+u^t(x-z),$$

where u is the Lagrange multiplier, and $\delta>0$ is a penalty parameter. The ADMM consists of three iterations:

$$\begin{cases} x^{k+1}=arg\min_x\ \ L(x,z^k,u^k); \\ z^{k+1}=arg\min_z\ \ L(x^{k+1},z,u^k); \\ u^{k+1}=u^k+\delta(x^{k+1}-z^{k+1}). \end{cases}$$

The first two steps have closed-form solutions and are described in Algorithm 2, where $shink(.,.)$ is a soft-thresholding operator given by:

$$shrink(x,r)_i=sgn(x_i)\max\{|x_i|-r,0\}.$$

Algorithm 2: ADMM for subproblem (2.33)

```
Initial guess: $x^0$, $z^0$, $u^0$ and iterative index $k=0$
while not converged do
    $x^{k+1} := (A^tA + 2cI + \delta I)^{-1}(A^ty - v^n + \delta z^k - u^k)$
    $z^{k+1} := shrink(\ x^{k+1} + u^k, \frac{a+1}{a\delta}\lambda\ )$
    $u^{k+1} := u^k + \delta(x^{k+1} - z^{k+1})$
    then $k+1 \rightarrow k$
end while
```

2.2.2 Convergence Theory for Unconstrained DCATL1

We present a convergence theory for the Algorithm 1 (DCATL1). We prove that the sequence $\{f(x^n)\}$ is decreasing and convergent, while the sequence $\{x^n\}$ is bounded under some requirement on λ. Its sub-limit vector x^* is a stationary point satisfying the first order optimality condition. Our proof is based on the convergent theory of DCA for $l_1 - l_2$ penalty function [66] besides the general results [54, 55].

Definition 2.2.1. *(Modulus of strong convexity) For a convex function $f(x)$, the modulus of strong convexity of f on $\Re^N$, denoted as $m(f)$, is defined by*

$$m(f) := sup\{\rho > 0 : f - \frac{\rho}{2}\|.\|_2^2 \text{ is convex on } \Re^N\}.$$

Let us recall a useful inequality from Proposition A.1 in [55] concerning the sequence $f(x^n)$.

Lemma 2.2.1. *Suppose that $f(x) = g(x) - h(x)$ is a D.C. decomposition, and the sequence $\{x^n\}$ is generated by (2.33), then*

$$f(x^n) - f(x^{n+1}) \geq \frac{m(g) + m(h)}{2}\|x^{n+1} - x^n\|_2^2.$$

Here is the convergence theory for our unconstrained Algorithm 1 — DCATL1. The objective function is : $f(x)=\frac{1}{2}\|Ax-y\|_2^2+\lambda P_a(x)$.

Theorem 2.2.1. *The sequences $\{x^n\}$ and $\{f(x^n)\}$ in Algorithm 1 satisfy:*

1. *Sequence $\{f(x^n)\}$ is decreasing and convergent.*
2. *$\|x^{n+1}-x^n\|_2\to 0$ as $n\to\infty$. If $\lambda>\dfrac{\|y\|_2^2}{2(a+1)}$, $\{x^n\}_{n=1}^\infty$ is bounded.*
3. *Any subsequential limit vector x^* of $\{x^n\}$ satisfies the first order optimality condition:*

$$0\in A^T(Ax^*-y)+\lambda\partial P_a(x^*), \tag{2.35}$$

implying that x^* is a stationary point of (0.4).

Proof. 1. By the definition of $g(x)$ and $h(x)$ in equation (2.32), it is easy to see that:

$$m(g)\geq 2c;$$
$$m(h)\geq 2c.$$

By Lemma 2.2.1, we have:

$$\begin{aligned} f(x^n)-f(x^{n+1}) &\geq \frac{m(g)+m(h)}{2}\|x^{n+1}-x^n\|_2^2 \\ &\geq 2c\|x^{n+1}-x^n\|_2^2. \end{aligned}$$

So the sequence $\{f(x^n)\}$ is decreasing and non-negative, thus convergent.

2. It follows from the convergence of $\{f(x^n)\}$ that:

$$\|x^{n+1}-x^n\|_2^2\leq\frac{f(x^n)-f(x^{n+1})}{2c}\to 0, \quad as\ n\to\infty.$$

If $y=0$, since the initial vector $x^0=0$, and the sequence $\{f(x^n)\}$ is decreasing, we have $f(x^n)=0$, $\forall n\geq 1$. So $x^n=0$, and the boundedness holds.

Consider non-zero vector y. Then

$$f(x^n)=\frac{1}{2}\|Ax^n-y\|_2^2+\lambda P_a(x^n)\leq f(x^0)=\frac{1}{2}\|y\|_2^2,$$

So $\lambda P_a(x^n)\leq \frac{1}{2}\|y\|_2^2$, implying $2\lambda\rho_a(\|x^n\|_\infty)\leq\|y\|_2^2$, or:

$$\frac{2\lambda(a+1)||x^n\|_\infty}{a+\|x^n\|_\infty}\leq\|y\|_2^2.$$

So if $\lambda>\dfrac{\|y\|_2^2}{2(a+1)}$, then

$$|x^n|_\infty\leq\frac{a\|y\|_2^2}{2\lambda(a+1)-\|y\|_2^2}.$$

Thus the sequence $\{x^n\}_{n=1}^\infty$ is bounded.

3. Let $\{x^{n_k}\}$ be a subsequence of $\{x^n\}$ which converges to x^*. So the optimality condition at the n_k-th step of Algorithm 1 is expressed as:

$$\begin{aligned}0\in\ & A^T(Ax^{n_k}-y)+2c(x^{n_k}-x^{n_k-1})\\ & +\lambda(\tfrac{a+1}{a})\partial\|x^{n_k}\|_1-\lambda\partial\varphi_a(x^{n_k-1}).\end{aligned}\tag{2.36}$$

Since $\|x^{n+1}-x^n\|_2\to 0$ as $n\to\infty$ and x^{n_k} converges to x^*, as shown in Proposition 3.1 of [66], we have that for sufficiently large index n_k,

$$\partial\|x^{n_k}\|_1\subseteq\partial\|x^*\|_1.$$

Letting $n_k \to \infty$ in (2.36), we have

$$0 \in A^T(Ax^* - y) + \lambda(\frac{a+1}{a})\partial\|x^*\|_1 - \lambda\partial\varphi_a(x^*).$$

By the definition of $\partial P_a(x)$ at (1.25), we have $0 \in A^T(Ax^* - y) + \lambda\partial P_a(x^*)$.

□

Remark 2.2.1. *The above theorem says that the sub-sequence limit x^* is a stationary point for (0.4). Let $T^* = supp(x^*)$, there exists vector $w \in \partial P_a(x^*)$, s.t.*

$$\begin{aligned}
& 0 = A^t(Ax^* - y) + \lambda w \\
\Rightarrow \quad & 0 = A^t_{T^*}(A_{T^*}x^*_{T^*} - y) + \lambda w_{T^*} \\
\Rightarrow \quad & 0 = Qx^*_{T^*} - A^t_{T^*}y + \lambda w_{T^*} \\
\Rightarrow & \; x^*_{T^*} = Q^{-1}A^t_{T^*}y - \lambda Q^{-1}w_{T^*}.
\end{aligned} \tag{2.37}$$

So (iii) of Condition 2 is automatically satisfied by x^*. If (i), (ii) and (iv) are also satisfied, the limit point x^* is a local minimizer of (0.4).

2.2.3 Algorithm for Constrained Model

Here we also give a DCA scheme to solve the constrained problem (0.3)

$$\begin{aligned}
& \min_{x \in \Re^N} P_a(x) \ s.t. \ \ Ax = y. \\
& \Leftrightarrow \\
& \min_{x \in \Re^N} \frac{a+1}{a}\|x\|_1 - \varphi_a(x) \ s.t. \ \ Ax = y.
\end{aligned}$$

We can rewrite the above optimization as

$$\min_{x\in\Re^N} \frac{a+1}{a}\|x\|_1 + \chi(x)_{\{Ax=y\}} - \varphi_a(x) = g(x) - h(x), \tag{2.38}$$

where $g(x) = \frac{a+1}{a}\|x\|_1 + \chi(x)_{\{Ax=y\}}$ is a polyhedral convex function [54].

Choose vector $z = \partial\varphi_a(x)$, then the convex sub-problem is:

$$\min_{x\in\Re^N} \frac{a+1}{a}\|x\|_1 - \langle z,x\rangle \ s.t. \ \ Ax = y. \tag{2.39}$$

To solve (2.39), we introduce two Lagrange multipliers u,v and define an augmented Lagrangian:

$$L_\delta(x,w,u,v) = \frac{a+1}{a}\|w\|_1 - z^t x + u^t(x-w) + v^t(Ax-y) + \frac{\delta}{2}\|x-w\|^2 + \frac{\delta}{2}\|Ax-y\|^2,$$

where $\delta > 0$. ADMM finds a saddle point (x^*,w^*,u^*,v^*), such that:

$$L_\delta(x^*,w^*,u,v) \le L_\delta(x^*,w^*,u^*,v^*) \le L_\delta(x,w,u^*,v^*) \quad \forall x,w,u,v$$

by alternately minimizing L_δ with respect to x, minimizing with respect to y and updating the dual variables u and v. The saddle point x^* will be a solution to (2.39). The overall algorithm for solving the constrained TL1 is described in Algorithm (3).

Algorithm 3: DCA method for constrained TL1 minimization

```
Define $\epsilon_{outer} > 0$, $\epsilon_{inner} > 0$. Initialize $x^0 = 0$ and outer loop index $n = 0$
while $\|x^n - x^{n+1}\| \geq \epsilon_{outer}$ do
  $z = \partial \varphi_a(x^n)$
  Initialization of inner loop: $x_{in}^0 = w^0 = x^n$, $v^0 = 0$ and $u^0 = 0$.
  Set inner index $j = 0$.
  while $\|x_{in}^j - x^{j+1}\| \geq \epsilon_{inner}$ do
    $x_{in}^{j+1} := (A^tA + I)^{-1}(w^j + A^ty + \frac{z - u^j - A^tv^j}{\delta})$
    $w^j = shrink(\ x_{in}^{j+1} + \frac{u^j}{\delta}, \frac{a+1}{a\delta}\ )$
    $u^{j+1} := u^j + \delta(x^{j+1} - w^j)$
    $v^{j+1} := v^j + \delta(Ax^{j+1} - y)$
  end while
  $x^n = x_{in}^j$ and $n = n + 1$.
end while
```

According to DC decomposition scheme (2.38), Algorithm 3 is a polyhedral DC program. Similar convergence theorem as the unconstrained model in last section can be proved. Furthermore, due to property of polyhedral DC programs, this constrained DCA also has a finite convergence. It means that if the inner subproblem (2.39) is exactly solved, $\{x^n\}$, the sequence generated by this iterative DC algorithm, has finite subsequential limit points [54].

2.2.4 Numerical Experiments

In this section, we use two classes of randomly generated matrices to illustrate the effectiveness of our Algorithms: DCATL1 (difference convex algorithm for transformed l_1 penalty) and its constrained version. We compare them separately with several state-of-the-art solvers on recovering sparse vectors:

- unconstrained algorithms:

  (i) Reweighted $l_{1/2}$ [36];

  (ii) DCA l_{1-2} algorithm [66, 40];

  (iii) CEL0 [57]

- constrained algorithms:

  (i) Bregman algorithm [67];

  (ii) Yall1;

  (iii) $Lp-RLS$ [14].

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

The two classes of random matrices are:

1) Gaussian matrix.

2) Over-sampled DCT with factor F.

We did not use prior information of the true sparsity of the original signal x^*. Also, for all the tests, the computation is initialized with zero vectors. In fact, the DCATL1 does not guarantee a global minimum in general, due to nonconvexity of the problem. Indeed we observe that DCATL1 with random starts often gets stuck at local minima especially when the matrix A is ill-conditioned (e.g. A has a large condition number or is highly coherent). In the numerical experiments, by setting $x_0 = 0$, we find that DCATL1 usually produces a global minimizer. The intuition behind our choice is that by using zero vector as initial guess, the first step of our algorithm reduces to solving an unconstrained weighted l_1 problem. So

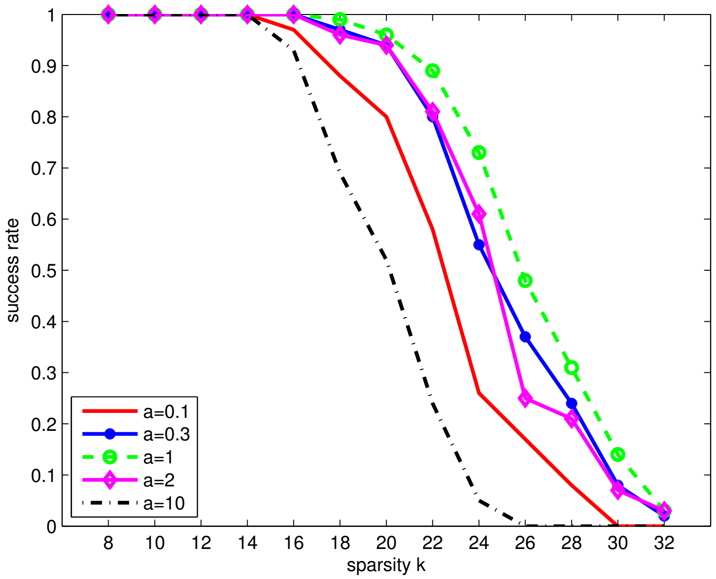


Figure 2.2: Numerical tests on parameter a with $M=64$, $N=256$ by the unconstrained DCATL1 method.

basically we are minimizing TL1 on the basis of l_1, which possibly explains why minimization of TL1 initialized by $x0=0$ always outperforms l_1.

Choice of Parameter: 'a'

In DCATL1, parameter a is also very important. When a tends to zero, the penalty function approaches the l_0 norm. If a goes to $+\infty$, objective function will be more convex and act like the l_1 optimization. So choosing a better a will improve the effectiveness and success rate for our algorithm.

We tested DCATL1 on recovering sparse vectors with different parameter a, varying among $\{0.1 \;\; 0.3 \;\; 1 \;\; 2 \;\; 10\}$. In this test, A is a 64×256 random matrix generated by normal Gaussian distribution. The true vector x^* is also a randomly generated sparse vector with sparsity k in the set $\{8 \;\; 10 \;\; 12 \; ... \;\; 32\}$. Here the regularization parameter λ was set to be 10^{-5} for all tests. Although the best λ should be dependent in general, we considered the noiseless case and $\lambda = 10^{-5}$ is small enough to approximately enforce $Ax = Ax^*$. For each a, we sampled 100 times with different A and x^*. The recovered vector x_r is accepted and recorded as one success if the relative error: $\frac{\|x_r - x^*\|_2}{\|x^*\|_2} \leq 10^{-3}$.

Fig. 2.8 shows the success rate using DCATL1 over 100 independent trials for various parameter a and sparsity k. From the figure, we see that DCATL1 with $a = 1$ is the best among all tested values. Also numerical results for $a = 0.3$ and $a = 2$ (near 1), are better than those with 0.1 and 10. This is because the objective function is more non-convex at a smaller a and thus more difficult to solve. On the other hand, iterations are more likely to stop at a local ℓ_1 minima far from ℓ_0 solution if a is too large. Thus in all the following tests, we set the parameter $a = 1$.

2.2.5 Numerical Experiment for Unconstrained Algorithm

Gaussian matrix

We use $\mathcal{N}(0, \Sigma)$, the multi-variable normal distribution to generate Gaussian matrix A. Here covariance matrix is $\Sigma = \{(1-r) * \chi_{(i=j)} + r\}_{i,j}$, where the value of '$r$' varies from 0 to 0.8. In theory, the larger the r is, the more difficult it is to recover true sparse vector. For matrix A, the row number and column number are set to be $M = 64$ and $N = 1024$. The sparsity k varies among $\{5 \;\; 7 \;\; 9 \; ... \;\; 25\}$.

We compare four algorithms in terms of success rate. Denote x_r as a reconstructed solution by a certain algorithm. We consider one algorithm to be successful, if the relative error of x_r to the truth solution x is less that 0.001, *i.e.*, $\frac{\|x_r - x\|}{\|x\|} < 1.e-3$. In order to improve success rates for all compared algorithms, we set tolerance parameter to be smaller or maximum cycle number to be higher inside each algorithm. As a result, it takes a long time to run one realization using all algorithms separately.

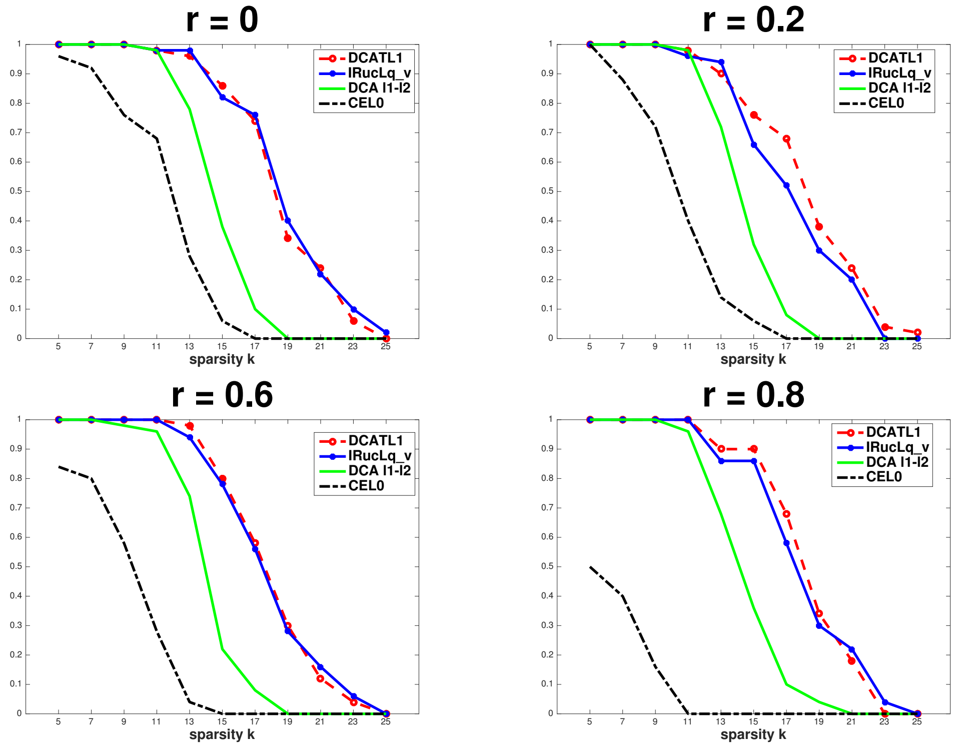


Figure 2.3: Numerical tests for unconstrained algorithms under Gaussian generated matrices: M = 64 , N = 1024 with different coherence r.

The success rate of each algorithm is plotted in Figure 2.9 with parameter r from the set: $\{0 \quad 0.2 \quad 0.6 \quad 0.8\}$. For all cases, DCATL1 and reweighted $l_{1/2}$ algorithms (IRucLq-v) performed almost the same and both were much better than the other two, while the CEL0 has the lowest success rate.

Over-sampled DCT

The over-sampled DCT matrices A [26] [40] are:

$$\begin{aligned} &A = [a_1, ..., a_N] \in \Re^{M \times N}, \\ &where \;\; a_j = \frac{1}{\sqrt{M}} cos(\frac{2\pi\omega(j-1)}{F}), \;\; j = 1, ..., N, \end{aligned} \tag{2.40}$$

and ω is a random vector, drawn uniformly from $(0,1)^M$.

Such matrices appear as the real part of the complex discrete Fourier matrices in spectral estimation [26] An important property is their high coherence: for a 100×1000 matrix with $F = 10$, the coherence is 0.9981, while the coherence of the same size matrix with $F = 20$, 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 [4], and this minimum length is 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 those four algorithms in terms of success rate. Denote x_r as a reconstructed solution by a certain algorithm. We consider one algorithm successful, if the relative error of x_r to the truth solution x is less that 0.001, *i.e.*, $\frac{\|x_r - x\|}{\|x\|} < 0.001$. The success rate is averaged over 50 random realizations.

Fig. 2.4 shows success rates for those algorithms with increasing factor F from 2 to 20. The sensing matrix is of size 100×1500. It is interesting to see that along with the increasing of value F, DCA of $l_1 - l_2$ algorithm performs better and better, especially after $F \geq 10$, and it has the highest success rate among all. Meanwhile, reweighted $l_{1/2}$ is better for low coherent matrices. When $F \geq 10$, it is almost impossible for it to recover sparse solution for the high coherent matrix. Our DCATL1, however, is more robust and consistently performed near the top, sometimes even the best. So it is a valuable choice for solving sparse optimization problems where coherence of sensing matrix is unknown.

We further look at the success rates of DCATL1 with different combinations of sparsity and

Table 2.1: The success rates (%) of DCATL1 for different combination of sparsity and minimum separation lengths.

sparsity	5	8	11	14	17	20
1RL	100	100	95	70	22	0
2RL	100	100	98	74	19	5
3RL	100	100	97	71	19	3
4RL	100	100	100	71	20	1
5RL	100	100	96	70	28	1

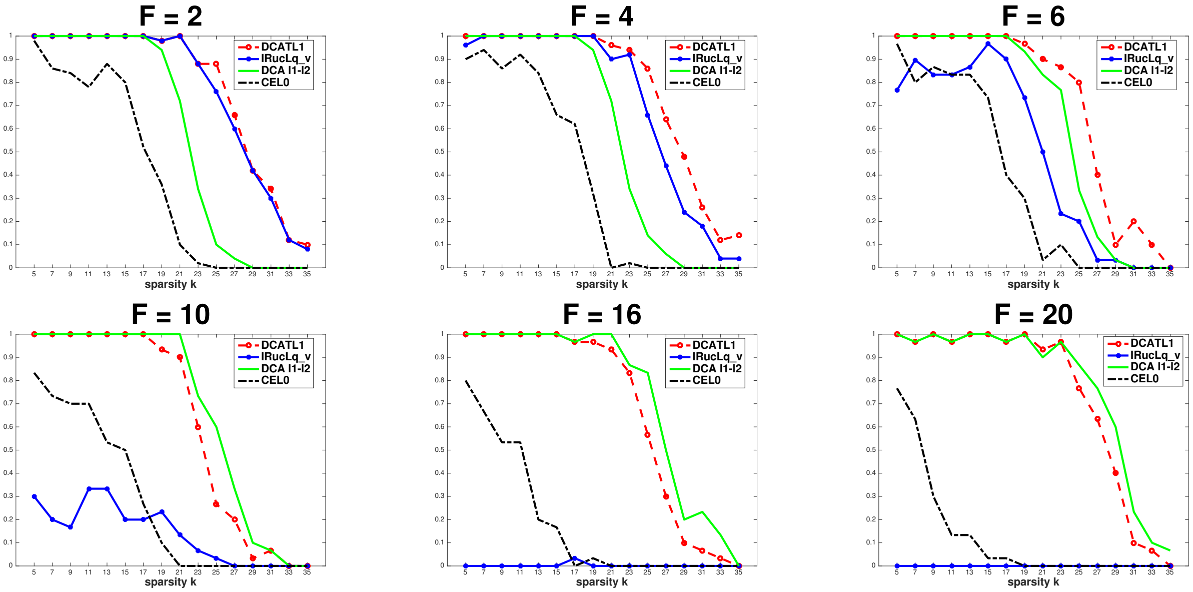


Figure 2.4: Numerical test for unconstrained algorithms under over-sampled DCT matrices: $M = 100$, $N = 1500$ with different F, and peaks of solutions separated by $2RL = 2F$.

separation lengths for the over-sampled DCT matrix A. The rates are recorded in Table 2.1, which shows that when the separation is above with the minimum length, the sparsity relative to M plays more important role in determining the success rates of recovery.

2.2.6 Numerical Experiment for Constrained Algorithm

For constrained algorithms, we performed similar numerical experiments. An algorithm is considered successful if the relative error of the numerical result x_r from the ground truth x is less than 0.001, or $\frac{\|x_r - x\|}{\|x\|} < 0.001$. We did 50 trials to compute average success rates for

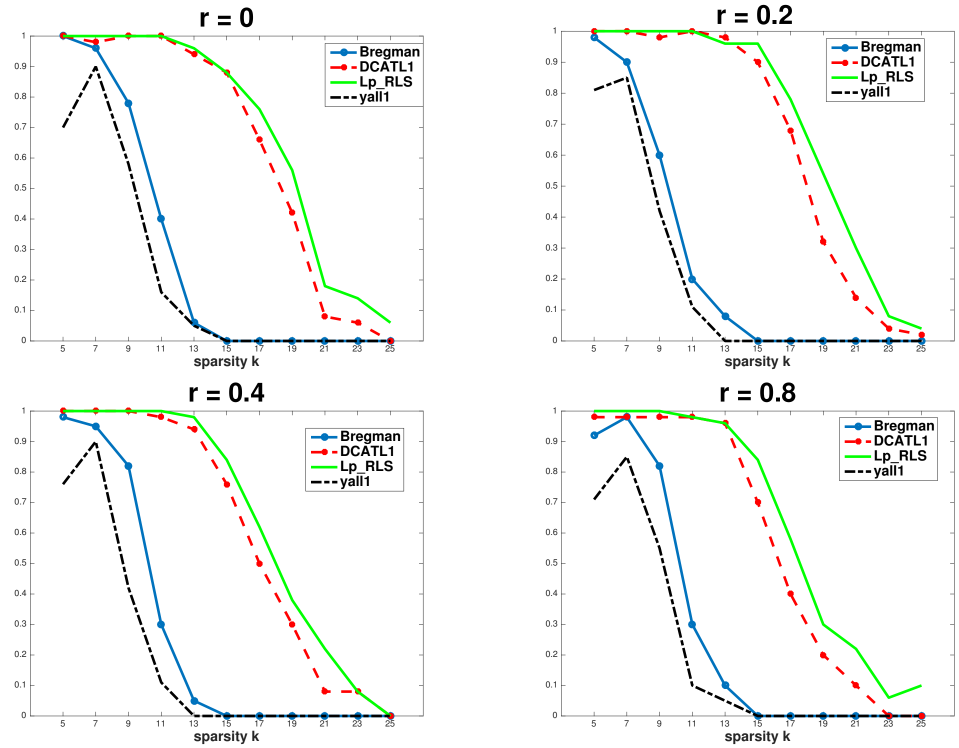


Figure 2.5: Comparison of constrained algorithms for 64×1024 Gaussian random matrices with different coherence parameter r. The data points are averaged over 50 trials.

all the numerical experiments as for the unconstrained algorithms.

Gaussian Random Matrices

We fix parameters $(M,N) = (64, 1024)$, while covariance parameter r is varied from 0 to 0.8. Comparison is with the reweighted $l_{1/2}$ and two l_1 algorithms (Bregman and yall1). In Fig. (2.5), we see that $Lp-RLS$ is the best among the four algorthms with DCATL1 trailing not much behind.

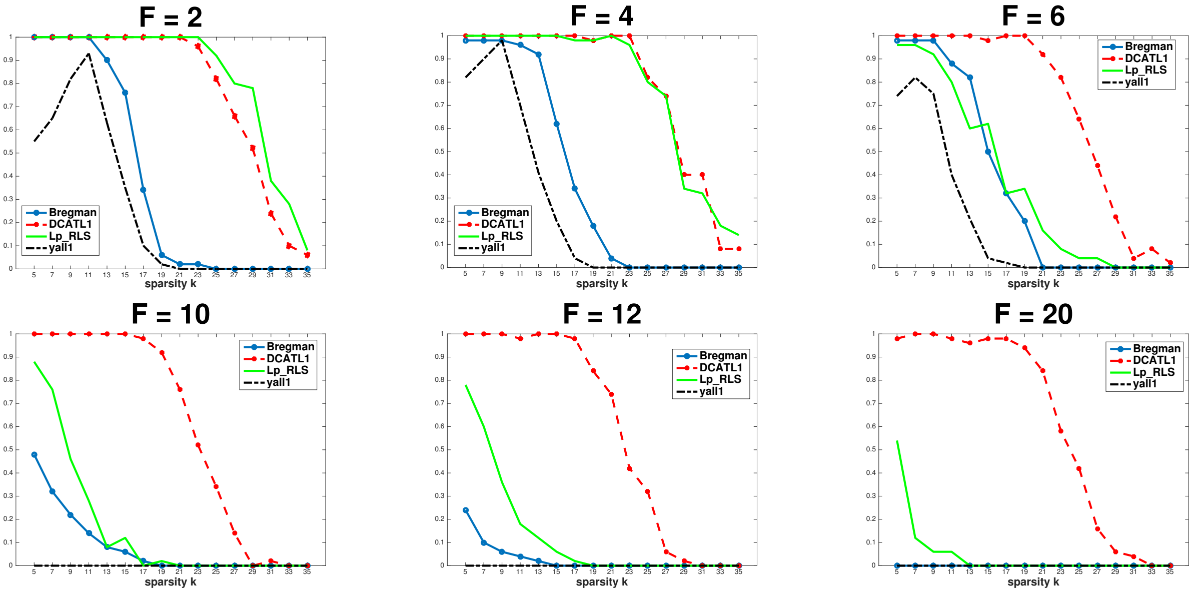


Figure 2.6: Comparison of success rates of constrained algorithms for the over-sampled DCT random matrices: $(M,N)=(100,1500)$ with different F values, peak separation by $2RL=2F$.

Over-sampled DCT

We fix $(M,N)=(100,1500)$, and vary parameter F from 2 to 20, so the coherence of there matrices has a wider range and almost reaches 1 at the high end. In Fig. (2.6), when F is small, say $F=2,4$, $Lp-RLS$ still performs the best, similar to the case of Gaussian matrices. However, with increasing F, the success rates for $Lp-RLS$ declines quickly, worse than the Bregman l_1 algorithm at $F=6,10$. The performance for DCATL1 is very stable and maintains a high level consistently even at the very high end of coherence $(F=20)$.

2.3 Thresholding TL1

The thresholding theories and algorithms for l_0 quasi-norm (hard-thresholding) [1, 2] and l_1 norm (soft-thresholding) [16, 20] 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 [12, 64] 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.*

2.3.1 Thresholding Representation and Closed-Form Solutions

Let us consider the unconstrained TL1 regularization model (0.4):

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

for which the first order optimality condition is:

$$0=A^T(Ax-y)+\lambda\cdot\nabla P_a(x). \tag{3.41}$$

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 (3.41), it is easy to get

$$x+\mu A^T(y-Ax)=x+\lambda\mu\nabla P_a(x). \tag{3.42}$$

We can rewrite the above equation, via introducing two operators

$$\begin{aligned} &R_{\lambda\mu,a}(x)=[I+\lambda\mu\nabla P_a(\cdot)]^{-1}(x),\\ &B_\mu(x)=x+\mu A^T(y-Ax). \end{aligned} \tag{3.43}$$

From equation (3.42), we will get a representation equation for optimal solution x:

$$x = R_{\lambda\mu,a}(B_\mu(x)). \tag{3.44}$$

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 (0.4).

Proximal Point Operator for TL1

Like [53], 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:

$$\begin{cases} t_1^* = \frac{3}{2^{2/3}}(\lambda a(a+1))^{1/3} - a \\ t_2^* = \lambda \frac{a+1}{a} \\ t_3^* = \sqrt{2\lambda(a+1)} - \frac{a}{2}. \end{cases} \tag{3.45}$$

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

Lemma 2.3.1. *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

   $$g_\lambda(x) = sgn(x)\left\{\frac{2}{3}(a+|x|)cos(\frac{\varphi(x)}{3}) - \frac{2a}{3} + \frac{|x|}{3}\right\} \tag{3.46}$$

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

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:

$$\begin{aligned} t_0 &= \tfrac{2(a+x)}{3}\cos(\varphi/3) \\ t_1 &= \tfrac{2}{3}(a+x)\cos(\varphi/3 + \pi/3) \\ t_2 &= -\tfrac{2}{3}(a+x)\cos(\pi/3 - \varphi/3) \end{aligned} \tag{3.47}$$

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 $f_{\lambda,x}(\cdot) : \Re \to \Re$,

$$f_{\lambda,x}(y) = \frac{1}{2}(y-x)^2 + \lambda \rho_a(y). \tag{3.48}$$

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

Theorem 2.3.1. *The optimal solution* $y_\lambda^*(x) = arg\min_y f_{\lambda,x}(y)$ *is a threshold function with* *threshold value t :*

$$y_\lambda^*(x) = \begin{cases} 0, & |x| \le t \\ g_\lambda(x), & |x| > t \end{cases} \tag{3.49}$$

where $g_\lambda(\cdot)$ is defined in (3.46). 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 (3.45).

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: $\lambda\leq\frac{a^2}{2(a+1)}$ and $\lambda>\frac{a^2}{2(a+1)}$.

2.1) $\lambda\leq\frac{a^2}{2(a+1)}$.

So we have $\inf_{y>0}f^{''}_{\lambda}(y)=f^{''}_{\lambda}(0+)=1-2\lambda\frac{(a+1)}{a^2}\geq 0$, which means function $f^{'}_{\lambda}(y)$ is increasing for $y\geq 0$, with minimum value $f^{'}_{\lambda}(0)=\lambda\frac{(a+1)}{a}-x=t_2^*-x$.

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

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

$$y(a+y)^2-x(a+y)^2+\lambda a(a+1)=0. \tag{3.50}$$

According to Lemma 2.3.1, the optimal value $y^*=y_0=g_\lambda(x)$.

Above all, the value for y^* is :

$$y^*=\begin{cases}0, & 0\leq x\leq t_2^*;\\ g_\lambda(x), \; x>t_2^* & \end{cases} \tag{3.51}$$

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

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 $f_\lambda^{'}(y) \geq t_1^* - x$, for $y \geq 0$.

i) When $0 \leq x \leq t_1^*$, function $f_\lambda(y)$ is always increasing. Thus optimal value $y^* = 0$.

ii) When $t_2^* \leq x$, $f_\lambda^{'}(0+) \leq 0$. 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 2.3.1, 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 2.3.1 again, the local minimize point will be $y_0 = g_\lambda(x)$, the largest root of equation (3.50). 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

$$\begin{aligned} f_\lambda(y_0) - f_\lambda(0) &= \tfrac{1}{2}y_0^2 - y_0 x + \lambda\tfrac{(a+1)y_0}{a+y_0} \\ &= y_0(\tfrac{1}{2}y_0 - x + \lambda\tfrac{(a+1)}{a+y_0}) \\ &= y_0(\tfrac{1}{2}y_0 - x + \tfrac{(x-y_0)(a+y_0)}{a}) \\ &= y_0^2(\tfrac{x-y_0}{a} - \tfrac{1}{2}) = y_0^2((x - g_\lambda(x))/a - 1/2) \end{aligned} \tag{3.52}$$

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

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

$$y^* = \begin{cases} 0, \qquad 0 \leq x \leq t_3^*; \\ g_\lambda(x),\, x > t_3^*, \end{cases} \tag{3.53}$$

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 (3.49) holds.

Summarizing results from all cases, the proof is complete.

□

2.3.2 Optimal Point Representation for Regularized TL1

Next, we will show that the optimal solution of the TL1 regularized problem (0.4) 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:

$$\begin{aligned} &C_\lambda(x) = \tfrac{1}{2}\|y - Ax\|_2^2 + \lambda P_a(x) \\ &C_\mu(x,z) = \mu\left\{C_\lambda(x) - \tfrac{1}{2}\|Ax - Az\|_2^2\right\} + \tfrac{1}{2}\|x - z\|_2^2. \end{aligned} \tag{3.54}$$

The first function $C_\lambda(x)$ comes from the objective of TL1 regularization problem (0.4).

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

$$\begin{cases} t_1^* = \frac{3}{2^{2/3}}(\lambda\mu a(a+1))^{1/3} - a \\ t_2^* = \lambda\mu\frac{a+1}{a} \\ t_3^* = \sqrt{2\lambda\mu(a+1)} - \frac{a}{2}. \end{cases} \tag{3.55}$$

Lemma 2.3.2. *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 $i = 1, \cdots, N$,*

$$\begin{aligned} &x_i^s = 0, && \textit{when } abs([B_\mu(z)]_i) \le t; \\ &x_i^s = g_{\lambda\mu}([B_\mu(z)]_i), && \textit{when } abs([B_\mu(z)]_i) > t. \end{aligned} \tag{3.56}$$

Here the function $g_{\lambda\mu}(\cdot)$ is same as (3.46) with parameter $\lambda\mu$ in place of λ there. $B_\mu(z) = z + \mu A^T(y - Az) \in \Re^N$, as in (3.43).

Proof. The second auxiliary objective function can be rewritten as

$$\begin{aligned} C_\mu(x,z) &= \tfrac{1}{2}\|x - [(I - \mu A^T A)z + \mu A^T y]\|_2^2 + \lambda\mu P_a(x) \\ &\quad + \tfrac{1}{2}\mu\|y\|_2^2 + \tfrac{1}{2}\|z\|_2^2 - \tfrac{1}{2}\mu\|Az\|_2^2 - \tfrac{1}{2}\|(I - \mu A^T A)z + \mu A^T y\|_2^2 \\ &= \tfrac{1}{2}\sum_{i=1}^N (x_i - [B_\mu(z)]_i)^2 + \lambda\mu\sum_{i=1}^N \rho_a(x_i) \\ &\quad + \tfrac{1}{2}\mu\|y\|_2^2 + \tfrac{1}{2}\|z\|_2^2 - \tfrac{1}{2}\mu\|Az\|_2^2 - \tfrac{1}{2}\|(I - \mu A^T A)z + \mu A^T y\|_2^2, \end{aligned} \tag{3.57}$$

which implies that

$$\begin{aligned} x^s &= arg\min_{x\in\Re^N} C_\mu(x,z) \\ &= arg\min_{x\in\Re^N} \left\{ \tfrac{1}{2}\sum_{i=1}^N (x_i - [B_\mu(z)]_i)^2 + \lambda\mu\sum_{i=1}^N \rho_a(x_i) \right\} \end{aligned} \tag{3.58}$$

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 :

$$f(x_i,z) = \frac{1}{2}(x_i - [B_\mu(z)]_i)^2 + \lambda\mu\rho_a(|x_i|). \tag{3.59}$$

Then by Theorem (2.3.1), the proof of our Lemma is complete.

□

Based on Lemma 2.3.2, we have the following representation theorem.

Theorem 2.3.2. *If* $x^* = (x_1^*, x_2^*, ..., x_N^*)^T$ *is a TL1 regularized solution of (0.4) with* a *and* λ *being positive constants, and* $0 < \mu < \|A\|^{-2}$*, then letting* $t = t_2^* 1_{\{\lambda\mu \leq \frac{a^2}{2(a+1)}\}} + t_3^* 1_{\{\lambda\mu > \frac{a^2}{2(a+1)}\}}$*, the optimal solution satisfies*

$$x_i^* = \begin{cases} g_{\lambda\mu}([B_\mu(x^*)]_i), & \textit{if } |[B_\mu(x^*)]_i| > t \\ 0, & \textit{others}. \end{cases} \tag{3.60}$$

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

$$\begin{aligned} C_\mu(x,x^*) &= \mu\{\tfrac{1}{2}\|y - Ax\|_2^2 + \lambda P_a(x)\} \\ &\quad + \tfrac{1}{2}\{-\mu\|Ax - Ax^*\|_2^2 + \|x - x^*\|_2^2\} \\ &\geq \mu\{\tfrac{1}{2}\|y - Ax\|_2^2 + \lambda P_a(x)\} \\ &\geq C_\mu(x^*,x^*), \end{aligned} \tag{3.61}$$

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 (0.4). In view of Lemma (2.3.2), we finish the proof.

□

2.3.3 TL1 Thresholding Algorithms

In this section, we propose 3 iterative thresholding algorithms for regularized TL1 optimization problem (0.4), based on Theorem 2.3.2.

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

$$G_{\lambda\mu,a}(w)=\begin{cases} 0, & \text{if } \ |w|\le t; \\ g_{\lambda\mu}(w), & \text{if } \ |w|>t. \end{cases} \tag{3.1}$$

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

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

According to Theorem 2.3.2, optimal solution of model (0.4) satisfies representation equation

$$x=G_{\lambda\mu,a}(B_\mu(x)). \tag{3.2}$$

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

$$x^{n+1}=G_{\lambda\mu,a}(x^n+\mu A^T(y-Ax^n))=G_{\lambda\mu,a}(B_\mu(x^n)), \tag{3.3}$$

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

$$t = \begin{cases} t_2^* = \lambda\mu\frac{a+1}{a}, & \text{if } \lambda \le \frac{a^2}{2(a+1)\mu}, \\ t_3^* = \sqrt{2\lambda\mu(a+1)} - \frac{a}{2}, & \text{if } \lambda > \frac{a^2}{2(a+1)\mu}. \end{cases} \tag{3.4}$$

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 2.3.1. *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 ([2, 16, 64]) are:

$$H_{\lambda,0}(x) = \begin{cases} x, & |x| > (2\lambda)^{1/2} \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

$$H_{\lambda,1}(x) = \begin{cases} x - sgn(x)\lambda, & |x| > \lambda \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

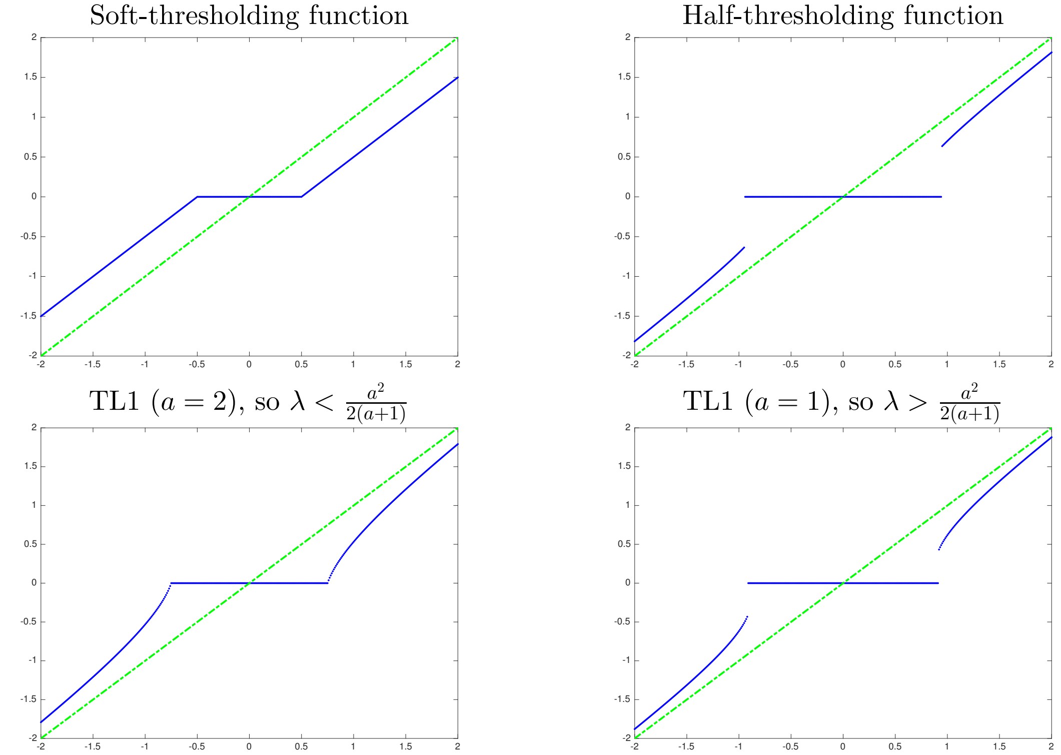


Figure 2.7: Soft/half (top left/right), TL1 (sub/super critical, lower left/right) thresholding functions at $\lambda = 1/2$.

and

$$H_{\lambda,1/2}(x) = \begin{cases} f_{2\lambda,1/2}(x), & |x| > \frac{(54)^{1/3}}{4}(2\lambda)^{2/3} \\ 0, & \text{otherwise} \end{cases} \tag{3.7}$$

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 Fig. 2.7, we plot the closed-form thresholding formulas (3.49) for $\lambda \leq$ 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 [46].

Convergence Theory for DFA

We establish the convergence theory for direct fixed point iterative algorithm, similar to [69, 64, 72]. Recall in (3.54), 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 2.3.3. *Let $\{x^n\}$ be the sequence generated by the iteration scheme (3.3) 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. $\lim\limits_{n\to\infty} \|x^{n+1} - x^n\| = 0$.

3) Any limit point x^* of $\{x^n\}$ is a stationary point satisfying equation (3.2), that is $x^* = G_{\lambda\mu,a}(B_\mu(x^*))$.

Proof. 1) From the proof of Lemma (2.3.2), 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 $C_\mu(x,z)$ (3.54), 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 $\|A\|^2 < 1/\mu$,

$$\begin{aligned} C_\lambda(x^{n+1}) &\leq \tfrac{1}{\mu}\left\{C_\mu(x^n,x^n) - \tfrac{1}{2}\|x^{n+1} - x^n\|_2^2\right\} + \tfrac{1}{2}\|Ax^{n+1} - Ax^n\|_2^2 \\ &= C_\lambda(x^n) + \tfrac{1}{2}(\|A(x^{n+1} - x^n)\|_2^2 - \tfrac{1}{\mu}\|x^{n+1} - x^n\|_2^2) \\ &\leq C_\lambda(x^n) \end{aligned} \tag{3.8}$$

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,

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

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

$$\big(\ \lambda(a+1) - C_\lambda(x^0)\ \big)\|x^n\|_\infty \leq a C_\lambda(x^0).$$

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 $\mu\|A(x^{n+1} - x^n)\|_2^2 \leq (1-\epsilon)\|x^{n+1} - x^n\|^2$, which can be rewritten as

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

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

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

where the last second inequality comes from (3.8) 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 2.3.2 (especially (3.58)), we have $D_{\lambda,\mu}(x)\geq 0$ and

$$D_{\lambda,\mu}(x)=0 \text{ if and only if } x \text{ satisfies (3.2).}$$

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 (3.3), we have $x^{n+1}=arg\min_z L_{\lambda,\mu}(z,x^n)$, which implies that

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

Thus we know

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

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 $0\leq D_{\lambda,\mu}(x^n)\leq \mu(C_\lambda(x^n)-C_\lambda(x^{n+1}))$. Also we know from part (1) of this theorem that $\{C_\lambda(x^n)\}$ converges, so $\lim\limits_{n\to\infty} D_{\lambda,\mu}(x^n)=0$. Thus as the limit point of the sequence x^n, the point x^* satisfies equation (3.2).

□

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 (0.4). 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, $|B_\mu(x^*)|_1\geq|B_\mu(x^*)|_2\geq\ldots\geq|B_\mu(x^*)|_N$. Then, the following inequalities hold:

$$\begin{aligned}&|B_\mu(x^*)|_i>t \Leftrightarrow i\in\{1,2,\ldots,k\},\\ &|B_\mu(x^*)|_j\leq t \Leftrightarrow j\in\{k+1,k+2,\ldots,N\},\end{aligned}\tag{3.9}$$

where t is our threshold value.

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

$$\begin{aligned} &|B_\mu(x^*)|_k \geq t \geq t_3^* = \sqrt{2\lambda\mu(a+1)} - \tfrac{a}{2}; \\ &|B_\mu(x^*)|_{k+1} \leq t \leq t_2^* = \lambda\mu\tfrac{a+1}{a}. \end{aligned} \tag{3.10}$$

It follows that

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

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

```
Algorithm 4: 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 $\lambda_1^n \leq \frac{a^2}{2(a+1)\mu}$ 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 $|z^n(i)| \leq t$, 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 $|z^n(i)| \leq t$, then $x^{n+1}(i) = 0$.
    end
    $n \rightarrow n+1$;
end
```

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

$$\lambda^* = \begin{cases} \lambda_1, & \text{if } \lambda_1 \leq \frac{a^2}{2(a+1)\mu}, \text{ then } \lambda^* \leq \frac{a^2}{2(a+1)\mu} \Rightarrow t = t_2^*; \\ \lambda_2, & \text{if } \lambda_1 > \frac{a^2}{2(a+1)\mu}, \text{ then } \lambda^* > \frac{a^2}{2(a+1)\mu} \Rightarrow t = t_3^*. \end{cases} \tag{3.11}$$

In practice, we approximate x^* by x^n in (3.11), 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.

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:

$$a = \lambda\mu + \sqrt{(\lambda\mu)^2 + 2\lambda\mu}. \tag{3.12}$$

The threshold value is:

$$t = t_2^* = \lambda\mu\frac{a+1}{a} = \frac{\lambda\mu}{2} + \frac{\sqrt{(\lambda\mu)^2 + 2\lambda\mu}}{2}. \tag{3.13}$$

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

$$\begin{aligned} &|B_\mu(x^*)|_i > t \Leftrightarrow i \in \{1,2,...,k\}, \\ &|B_\mu(x^*)|_j \le t \Leftrightarrow j \in \{k+1,k+2,...,N\}. \end{aligned} \tag{3.14}$$

So, for parameter λ, we have:

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

Once the value of λ is determined, the parameter a is given by (2.27).

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

$$\begin{aligned} &\lambda_n = \frac{1}{\mu_n}\frac{2|B_{\mu_n}(x^*)|_{k+1}^2}{1+2|B_{\mu_n}(x^*)|_{k+1}}; \\ &a_n = \lambda_n\mu_n + \sqrt{(\lambda_n\mu_n)^2 + 2\lambda_n\mu_n}. \end{aligned} \tag{3.15}$$

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 5: Adaptive TL1 Thresholding Algorithm — TL1IT-s2
Initialize: $x^0$, $\mu_0 = \frac{(1-\varepsilon)}{\|A\|^2}$;
while not converged do
    $\mu = \mu_0$;    $z^n := x^n + \mu A^T(y - Ax^n)$;
    $\lambda_n = \frac{1}{\mu}\frac{2|z_{k+1}^n|^2}{1+2|z_{k+1}^n|}$;
    $a_n = \lambda_n\mu + \sqrt{(\lambda_n\mu)^2 + 2\lambda_n\mu}$;
    $t = \frac{\lambda_n\mu}{2} + \frac{\sqrt{(\lambda_n\mu)^2+2\lambda_n\mu}}{2}$;
    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
```

2.3.4 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** [1], **Half-thresholding** [64]. Here all these thresholding algorithms need a sparsity estimation to accelerate convergence. Also the Hard Thresholding algorithm (AIHT) in [1] 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, [65]) a number of times, e.g. 20 times, to produce a better initial guess x^0. This procedure is similar to algorithm DCATL1 [72] 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 equal to 3000.

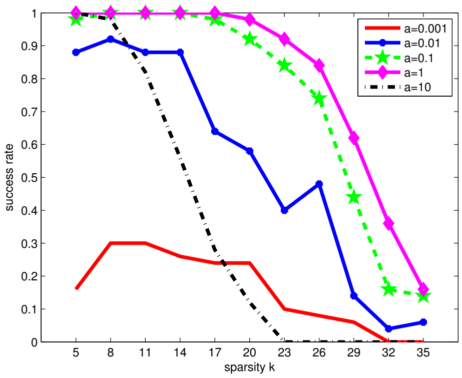


Figure 2.8: Sparse recovery success rates for selection of parameter a with 128×512 Gaussian random matrices and TL1IT-s1 method.

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 (2.8) 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 $\mu = \frac{0.99}{\|A\|^2}$.

2.3.5 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 $\{5,\ 8,\ 11, \cdots,\ 35\}$.

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 2.9 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 $k \geq 26$. 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.

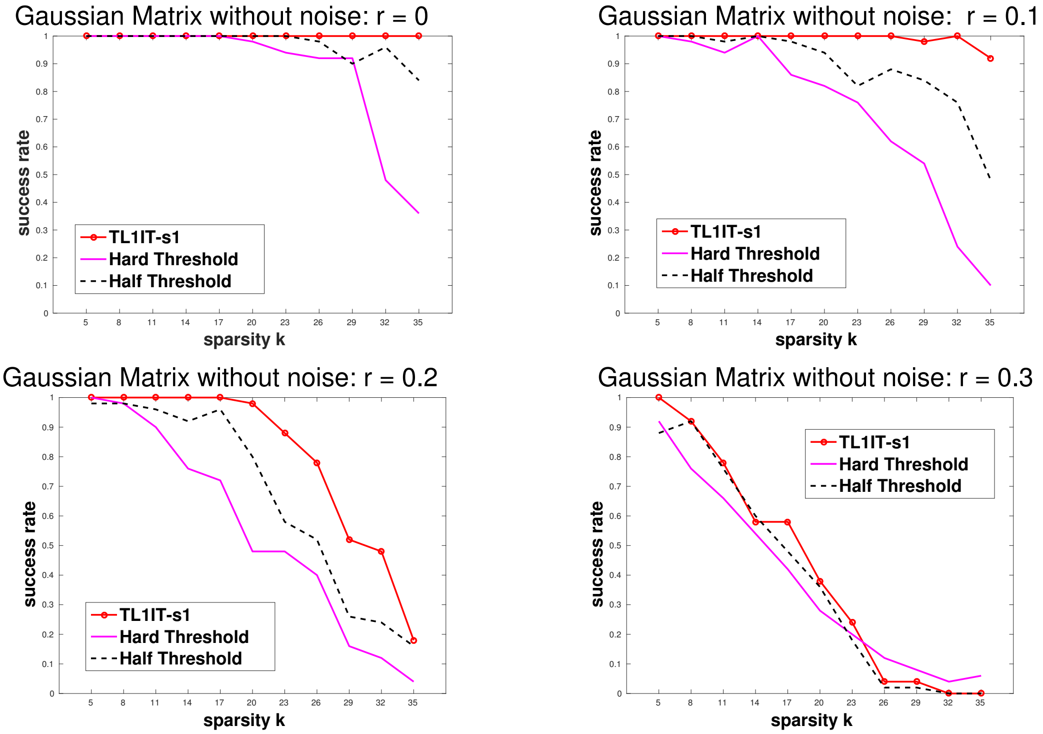


Figure 2.9: Sparse recovery algorithm comparison for 128×512 Gaussian sensing matrices without measurement noise at covariance parameter $r = 0, 0.1, 0.2, 0.3$.

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 2.2: 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 2.9, 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 [26, 40] are:

$$
\begin{aligned}
&A = [a_1, ..., a_N] \in \Re^{M \times N} \\
&\text{where } \; a_j = \frac{1}{\sqrt{M}} cos(\frac{2\pi\omega(j-1)}{F}), \;\; j = 1, ..., N, \\
&\text{ and } \omega \text{ is a random vector, drawn uniformly from } (0,1)^M.
\end{aligned}
\tag{3.1}
$$

Such matrices appear as the real part of the complex discrete Fourier matrices in spectral estimation and super-resolution problems [4, 26]. 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 [4], 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-

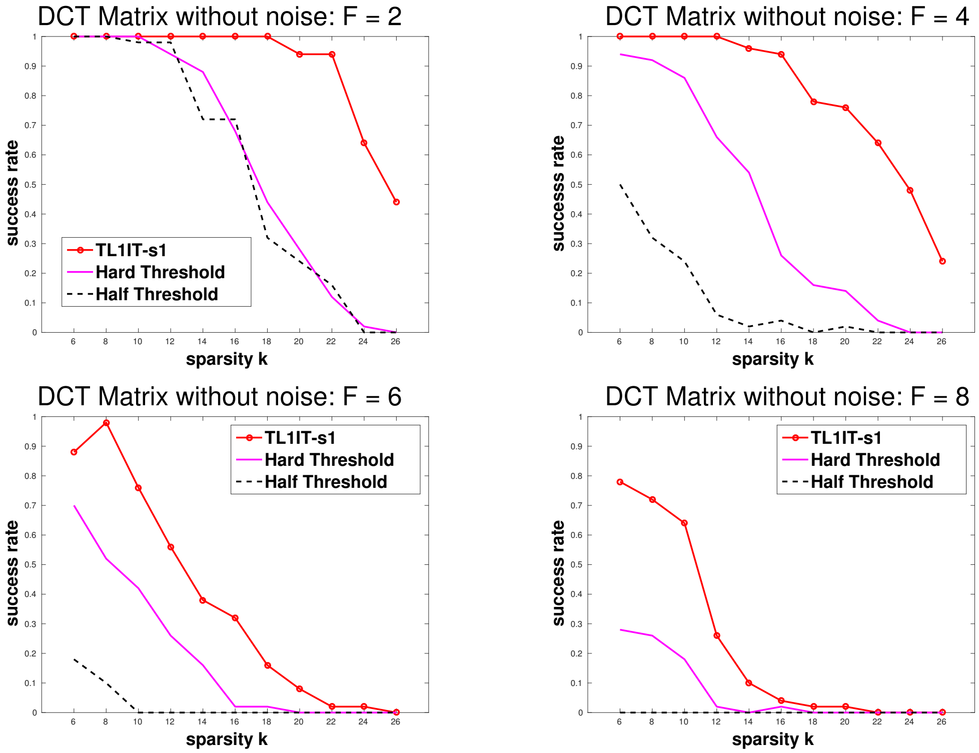


Figure 2.10: Algorithm comparison for 100×1500 over-sampled DCT random matrices without noise at different factor F.

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 2.10 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 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.

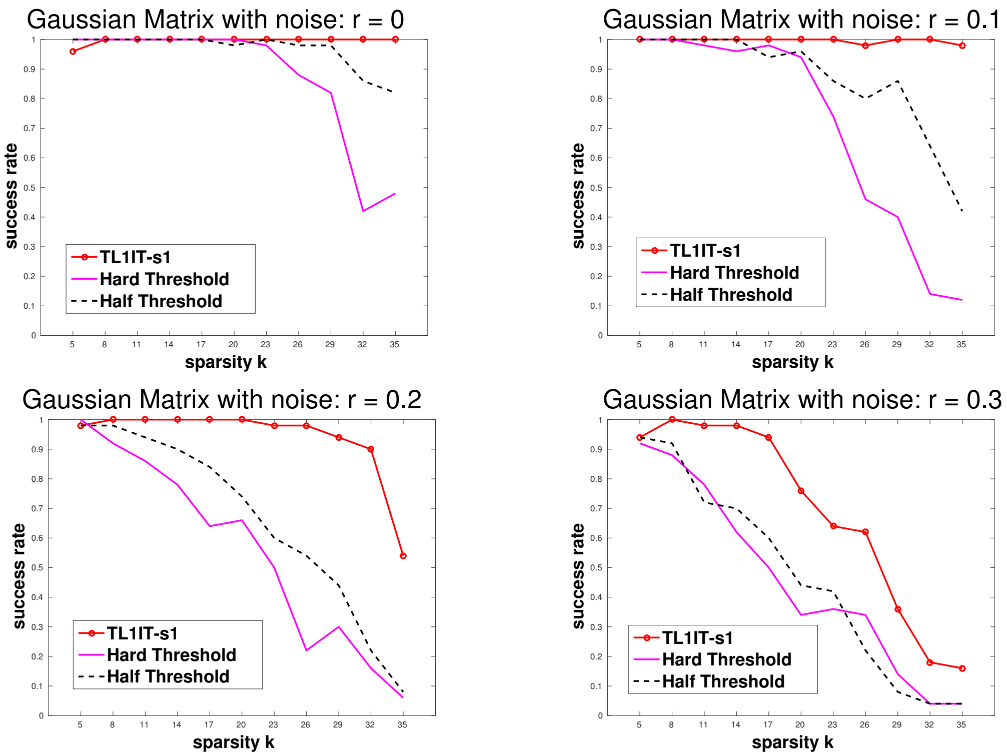


Figure 2.11: Algorithm comparison in success rates for 128×512 Gaussian sensing matrices with additive noise at different coherence r.

2.3.6 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 $|\varepsilon|_\infty \leq 0.01$.

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 $\frac{\|x^r - x\|}{\|x\|} \leq 10^{-2}$.

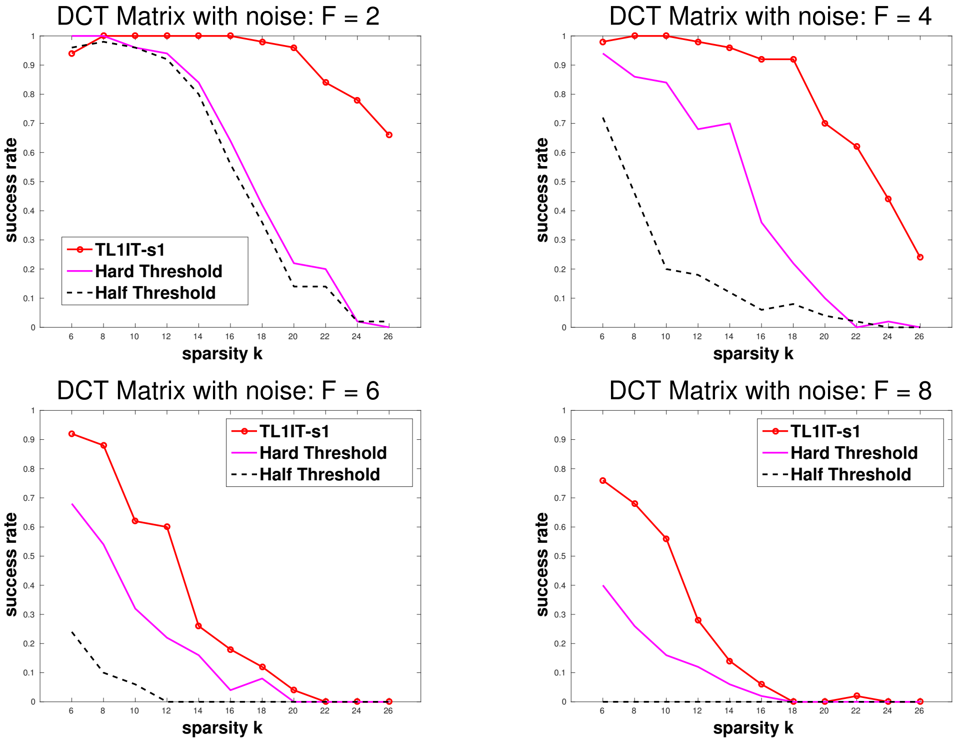


Figure 2.12: Algorithm comparison for over-sampled DCT matrices with additive noise: $M=100$, $N=1500$ at $F=2,4,6,8$.

The numerical results are shown in Figure 2.11. 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

Fig. 2.12 shows results of three algorithms under the over-sampled DCT sensing matrices. Relative error of 0.01 or under 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 ($F=6,8$).

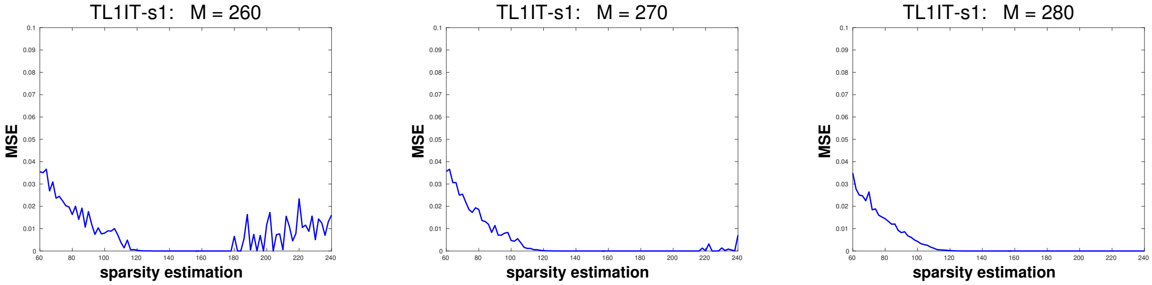


Figure 2.13: Robustness tests (mean square error vs. sparsity) for TL1IT-s1 thresholding algorithm under Gaussian sensing matrices: $r=0, N=512$ and number of measurements $M=260, 270, 280$. The real sparsity is fixed as $k=130$.

2.3.7 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 2.13 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 2.13, 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.

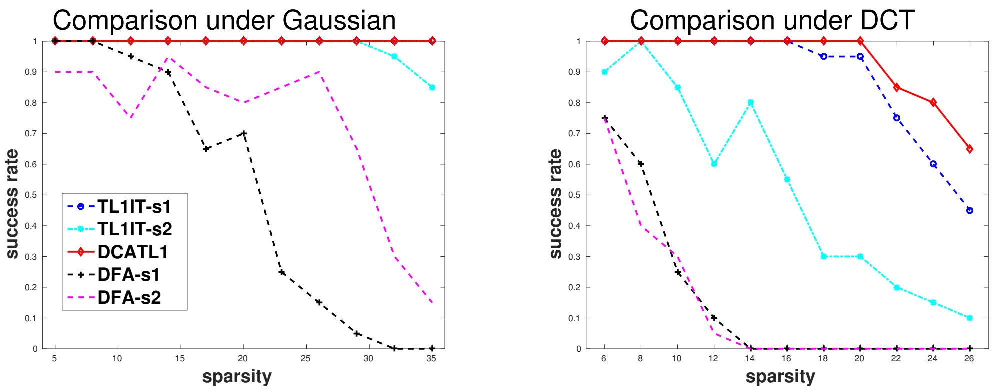


Figure 2.14: TL1 algorithms comparison. Y-axis is success rate from 20 random tests with accepted relative error 10^{-3}. X-axis is sparsity value k. Left: 128×512 Gaussian sensing matrices with sparsity $k=5,\cdots,35$. Right: 100×1500 Gaussian sensing matrices with sparsity $k=6,\cdots,26$.

2.3.8 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 [72], 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 $\lambda\mu>a^2/2(a+1)$.

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 2.14. 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.

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.

Chapter 3

Matrix Completion

This chapter is organized as follows. In section 3.1, we present the transformed Schatten-1 function (TS1), the TS1 regularized minimization problems, and a derivation of thresholding representation of the global minimum. In section 3.2, we propose two thresholding algorithms (TS1-s1 and TS1-s2) based on a fixed point equation of the global minimum. In section 3.4, we compare TS1 algorithms with some state-of-the-art algorithms through numerical experiments in low rank matrix recovery and image inpainting.

Notation

Here we set the notations for this chapter. Two kinds of inner products are used in the following sections, one is between matrices and the other is a bilinear operation for vectors:

$$(x,y) = \sum_i x_i y_i \quad \text{for vectors } x,y;$$

$$\langle X,Y \rangle = \mathrm{tr}(Y^T X) = \sum_{i,j} X_{i,j} Y_{i,j} \quad \text{for matrices } X,Y.$$

Assume matrix $X \in \Re^{m\times n}$ has r positive singular values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0$. Let us introduce some common matrix norms or quasi-norms as,

- Nuclear norm: $\|X\|_* = \sum_{i=1}^{r} \sigma_i$;
- Schatten p quasi-norm: $\|X\|_p = (\sum_{i=1}^{r} \sigma_i^p)^{1/p}$, for $p \in (0,1)$;
- Frobenius norm: $\|X\|_F = (\sum_{i=1}^{r} \sigma_i^2)^{\frac{1}{2}}$. Also $\|X\|_F^2 = \langle X, X \rangle = \sum_{i,j} X_{i,j}^2$.
- Ky Fan k-norm: $\|X\|_{Fk} = \sum_{i=1}^{k} \sigma_i$, for $1 \le k \le r$;
- Induced L^2 norm: $\|X\|_{L^2} = \max_{\|v\|_2 = 1} \|Xv\|_2 = \sigma_1$.

Define function vec$(\cdot)$ to unfold one matrix columnwise into a vector. So it is clear that $\|\text{vec}(X)\|_2 = \|X\|_F$, where the left hand side norm is vector's ℓ_2 norm.

Define the shrinkage identity k matrix $I_k^s \in \Re^{m \times n}$ as following,

$$\begin{cases} I_k^s(i,i) = 1, \ \textit{the first } k \textit{ diagonal elements}; \\ I_k^s(i,j) = 0, \ \textit{others}. \end{cases} \tag{0.1}$$

Operator $\mathrm{tr_k}(\cdot)$ is defined as the first k partial trace of a matrix,

$$\mathrm{tr_k}(X) = \sum_{i=1}^{k} X_{i,i}. \tag{0.2}$$

The following matrix functions will be used in the proof of next section, and we want to write them out first here for reference:

$$\begin{aligned} C_\lambda(X) &= \tfrac{1}{2}\|\mathscr{A}(X) - b\|_2^2 + \lambda T(X); \\ C_{\lambda,\mu}(X,Z) &= \mu\left\{ \ C_\lambda(X) - \tfrac{1}{2}\|\mathscr{A}(X) - \mathscr{A}(Z)\|_2^2 \ \right\} + \tfrac{1}{2}\|X - Z\|_F^2 \\ &= \mu\lambda T(X) + \tfrac{\mu}{2}\|b\|_2^2 - \tfrac{\mu}{2}\|\mathscr{A}(Z)\|_2^2 - \mu\langle \mathscr{A}(X), b - \mathscr{A}(Z)\rangle + \tfrac{1}{2}\|X - Z\|_F^2; \\ B_\mu(Z) &= Z + \mu\mathscr{A}^*(b - \mathscr{A}(Z)). \end{aligned} \tag{0.3}$$

3.1 TS1 minimization and thresholding representation

First, let us introduce Transformed Schatten-1 penalty function(TS1) based on the singular values of a matrix:

$$T(X) = \sum_{i=1}^{\mathrm{rank}(X)} \rho_a(\sigma_i), \tag{1.4}$$

where $\rho_a(\cdot)$ is a linear-to-linear rational function with parameter $a \in (0, \infty)$ [71, 72],

$$\rho_a(|x|) = \frac{(a+1)|x|}{a+|x|}. \tag{1.5}$$

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 Fig. 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 $a = 0.01$).

We shall focus on TS1 regularized problem

$$\min_{X \in \Re^{m \times n}} \frac{1}{2} \|\mathscr{A}(X) - b\|_2^2 + \lambda T(X), \tag{1.6}$$

where the linear transform $\mathscr{A} : \Re^{m \times n} \to \Re^p$ can be determined by p given matrices $A_1, ..., A_p \in \Re^{m \times n}$, that is, $\mathscr{A}(X) = (\langle A_1, X \rangle, ..., \langle A_p, X \rangle)^T$.

3.1.1 TS1 thresholding representation theory

Here we assume $m \leq n$. For a matrix $X \in \Re^{m\times n}$ with rank equal to r, its singular values vector $\sigma = (\sigma_1, ..., \sigma_m)$ is arranged as

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r > 0 = \sigma_{r+1} = ... = \sigma_m.$$

The singular value decomposition (SVD) is $X = UDV^T$, where $U = (U_{i,j})_{m\times m}$ and $V = (V_{i,j})_{n\times n}$ are unitary matrices, with $D = Diag(\sigma) \in \Re^{m\times n}$ diagonal.

In [25], Ky Fan proved the dominance theorem and derive the following Ky Fan k-norm inequality.

Lemma 3.1.1. *(Ky Fan k-norm inequality) For a matrix $X \in \Re^{m\times n}$ with SVD: $X = U\,D\,V^T$, where diagonal elements of D are arranged in decreasing order, we have:*

$$\langle X, I_k^s \rangle \leq \langle D, I_k^s \rangle,$$

that is, $\mathrm{tr_k}(X) \leq \mathrm{tr_k}(D) = \|X\|_{Fk}$, *$\forall k = 1, 2, ..., m$. The inequalities become equalities if and only if $X = D$. Here matrix I_k^s and operator* $\mathrm{tr_k}(\cdot)$ *are defined in section 3.*

Here we give another proof of this inequality without using dominance theorem is available, making the paper self-contained.

Proof. Since $X = U\mathrm{Diag}(\sigma)V^T$, the (j,k)-th entry of matrix X is $X_{j,k} = \sum\limits_{i=1}^{m} \sigma_i U_{j,i} V_{k,i}$.

Thus, we have

$$\begin{aligned} \mathrm{tr_k}(X) &= \sum_{j=1}^{k} X_{j,j} = \sum_{j=1}^{k}\sum_{i=1}^{m} \sigma_i U_{j,i} V_{j,i} \\ &= \sum_{i=1}^{m}\sum_{j=1}^{k} \sigma_i U_{j,i} V_{j,i} = \sum_{i=1}^{m} \sigma_i w_i^{(k)}, \end{aligned} \tag{1.7}$$

where the weight $w_i^{(k)}$ for the singular value σ_i is defined as:

$$w_i^{(k)} = \sum_{j=1}^{k} U_{j,i} V_{j,i}, \quad i = 1,2,...,m. \tag{1.8}$$

Notice that,

$$|w_i^{(k)}| \leq \sum_{j=1}^{k} |U_{j,i}||V_{j,i}| \leq \|U(:,i)\|_2 \|V(:,i)\|_2 \leq 1, \tag{1.9}$$

where $U(:,i)$ and $V(:,i)$ are the i-th column vectors for U and V. Also for weights $\{w_i^{(k)}\}$,

$$\begin{aligned} \sum_{i=1}^{m} |w_i^{(k)}| &\leq \sum_{i=1}^{m}\sum_{j=1}^{k} |U_{j,i}||V_{j,i}| = \sum_{j=1}^{k}\sum_{i=1}^{m} |U_{j,i}||V_{j,i}| \\ &\leq \sum_{j=1}^{k} \ \|U(j,:)\|_2 \ \|V(j,:)\|_2 \leq k, \end{aligned} \tag{1.10}$$

where $U(j,:)$ and $V(j,:)$ are the j-th row vectors for U and V, respectively.

All the m weights are bounded by 1, with absolute sum at most $k \leq m$. Note that σ_i's are in decreasing order. By equation (1.7), we have, for all $k = 1,2,...,m$,

$$\mathrm{tr_k}(X) \leq \sum_{i=1}^{m} \sigma_i |w_i^{(k)}| \leq \sum_{i=1}^{k} \sigma_i = \mathrm{tr_k}(D) = \|X\|_{Fk}.$$

Next, we prove the second part of the lemma — equality condition, by mathematical induction. Suppose that for a given matrix X, $\mathrm{tr_k}(X) = \mathrm{tr_k}(D)$, $\forall\ k = 1,...,m$. Here, it is convenient to define $X_i = \sigma_i U_i V_i^T$, where V_i (U_i) is the i-th column vector of V (U). Then matrix X can be decomposed as the sum of r rank-1 matrices, $X = \sum_{i=1}^{r} X_i$.

When $k = 1$, according to $tr_1(X) = tr_1(D)$ and the proof above, we know that

$$w_1^{(1)} = 1 \ \text{ and } w_i^{(1)} = 0 \ \text{ for } i = 2,...,m.$$

By the definition of weights $w_i^{(k)}$ in (1.8), we have $w_1^{(1)} = U_{1,1}V_{1,1} = 1$. Since U and V are both unitary matrices, we have:

$$U_{1,1} = V_{1,1} = \pm 1; \quad U_{1,j} = U_{j,1} = V_{1,j} = V_{j,1} = 0 \text{ for } j \neq 1.$$

Then vectors U_1 (V_1) is the first standard basis vector in space $\Re^m$ ($\Re^n$). The matrix $X_1 = \sigma_1 U_1 V_1^T$ is diagonal

$$X_1 = \begin{bmatrix} \sigma_1 & & & \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}_{m \times n}$$

For any index i, $1 \leq i \leq k-1$, suppose that

$$U_{i,i} = V_{i,i} = \pm 1; \quad U_{i,j} = U_{j,i} = V_{i,j} = V_{j,i} = 0 \text{ for any index } j \neq i. \tag{1.11}$$

Then matrix $X_i = \sigma_i U_i V_i^T$, with $1 \leq i \leq k-1$, is diagonal and can be expressed as

$$X_i = \begin{bmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & & & \\ & & & \sigma_i & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix}_{m \times n} \qquad \longleftarrow (i\text{-th row})$$

Under those conditions, let us consider the case with index $i = k$. Clearly, we have $tr_k(X) = tr_k(D)$. Similarly as before, thanks to the formula (1.7) and inequalities (1.9) and (1.10), it is true that

$$w_i^{(k)} = 1 \text{ for } i = 1, \ldots, k; \quad \text{and } w_i^{(k)} = 0 \text{ for } i > k.$$

Furthermore, by definition (1.8), $w_k^{(k)} = \sum_{j=1}^{k} U_{j,k} V_{j,k} = U_{k,k} V_{k,k} = 1$. This is because $U_{j,k} = V_{j,k} = 0$ for index $j < k$, by the assumption (1.11) . Thus vectors U_k and V_k are also standard basis vectors with the k-th entry to be ±1. Then

$$X_k = \sigma_k U_k V_k^T = \begin{bmatrix} 0 & & & & & & \\ & \ddots & & & & & \\ & & 0 & & & & \\ & & & \sigma_k & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{bmatrix}_{m \times n} \longleftarrow (k\text{-th row})$$

Finally, we prove that all matrices $\{X_i\}_{i=1,\cdots,r}$ are diagonal. So the original matrix $X = \sum_{i=1}^{r} X_i$ is equal to the diagonal matrix D. The other direction is obvious. We finish the proof. □

Theorem 3.1.1. *Consider matrix* $Y \in \Re^{m \times n}$ *that admits a singular value decomposition of the form:* $Y = U \operatorname{Diag}(\sigma) V^T$, *where* $\sigma = (\sigma_1, \ldots, \sigma_m)$. *Then the unique global minimizer of* $\min_{X \in \Re^{m \times n}} \frac{1}{2} \|X - Y\|_F^2 + \lambda T(X)$ *is:*

$$X^s = G_{\lambda,a}(Y) = U \operatorname{Diag}(g_{\lambda,a}(\sigma)) V^T, \tag{1.12}$$

where $g_{\lambda,a}(\cdot)$ *is defined in (3.46) and applied entrywise to* σ.

Proof. First due to the unitary invariance property of Frobenius norm and $Y = U \operatorname{Diag}(\sigma) V^T$, we have

$$\frac{1}{2}\|X-Y\|_F^2+\lambda T(X)=\frac{1}{2}\|U^TXV-\text{Diag}(\sigma)\|_F^2+\lambda T(U^TXV).$$

So

$$\begin{aligned} X^s &= \operatorname*{arg\,min}_{X\in\Re^{m\times n}} \tfrac{1}{2}\|X-Y\|_F^2+\lambda T(X) \\ &= U\left\{\operatorname*{arg\,min}_{X\in\Re^{m\times n}} \tfrac{1}{2}\|X-Diag(\sigma)\|_F^2+\lambda T(X)\right\}V^T. \end{aligned} \tag{1.13}$$

Next we want to show:

$$\begin{aligned} &\operatorname*{arg\,min}_{X\in\Re^{m\times n}} \tfrac{1}{2}\|X-\text{Diag}(\sigma)\|_F^2+\lambda T(X) \\ &= \text{arg}\min_{\{D\in\Re^{m\times n}\text{ is diagonal}\}} \tfrac{1}{2}\|D-\text{Diag}(\sigma)\|_F^2+\lambda T(D) \end{aligned} \tag{1.14}$$

For any $X\in\Re^{m\times n}$, suppose it admits SVD: $X=U_x Diag(\sigma_x)V_x^T$. Denote

$$D_x=\text{Diag}(\sigma_x) \text{ and } D_y=\text{Diag}(\sigma).$$

We can rewrite diagonal matrix D_y as $D_y=\sum_i^m \nabla\sigma_i I_i^s$, where $\nabla\sigma_i=\sigma_i-\sigma_{i+1}\geq 0$ for $i=1,2,...,m-1$, and $\nabla\sigma_m=\sigma_m$. So simply, $\sum_{i=k}^m \nabla\sigma_i=\sigma_k$. Note that the shrinkage identity i matrix I_i^s is defined in section 3. So:

$$\begin{aligned} \langle X, D_y\rangle &= \langle X, \sum_i^m \nabla\sigma_i I_i^s\rangle = \sum_i^m \langle X, \nabla\sigma_i I_i^s\rangle \\ &\leq \sum_i^m \langle D_x, \nabla\sigma_i I_i^s\rangle = \langle D_x, D_y\rangle, \end{aligned}$$

where we used Lemma 3.1.1 for the inequality. The equality holds if and only if $X=D_x$.

Thus we have

$$\begin{aligned}\|X-D_y\|_F^2 &= \|X\|_F^2+\|D_y\|_F^2-2\langle X,D_y\rangle \\ &\geq \|D_x\|_F^2+\|D_y\|_F^2-2\langle D_x,D_y\rangle = \|D_x-D_y\|_F^2.\end{aligned}$$

Also due to $T(X)=T(D_x)$,

$$\frac{1}{2}\|X-D_y\|_F^2+\lambda T(X)\geq \frac{1}{2}\|D_x-D_y\|_F^2+\lambda T(D_x).$$

Only when $X=D_x$ is a diagonal matrix, the above will become equality. So we proved equation (1.14).

Denote a diagonal matrix $D\in\Re^{m\times n}$ as $D=\mathrm{Diag}(d)$. Then:

$$\frac{1}{2}\|D-\mathrm{Diag}(\sigma)\|_F^2+\lambda T(D)=\sum_{i=1}^{m}\left\{\frac{1}{2}\|d_i-\sigma_i\|_2^2+\lambda\rho_a(|d_i|)\right\}$$

By Theorem 2.3.1, we have $g_{\lambda,a}(\sigma_i)=arg\min_{d}\{\ \frac{1}{2}\|d-\sigma_i\|_2^2+\lambda\rho_a(|d|)\ \}\geq 0$. It follows that

$$\begin{aligned}&\arg\min_{\{D\in\Re^{m\times n}\text{and D is diagonal}\}}\tfrac{1}{2}\|D-\mathrm{Diag}(\sigma)\|_F^2+\lambda\ T(D)\\ &=\arg\min_{X\in\Re^{m\times n}}\tfrac{1}{2}\|X-\mathrm{Diag}(\sigma)\|_F^2+\lambda\ T(X) \qquad (1.15)\\ &=\mathrm{Diag}(g_{\lambda,a}(\sigma)).\end{aligned}$$

In view of (1.13), the matrix $X^s=UDiag(g_{\lambda,a}(\sigma))V^T$ is a global minimizer, which will be denoted as $G_{\lambda,a}(Y)$.

Let X_1 be another optimal solution for the problem $\min_{X\in\Re^{m\times n}}\frac{1}{2}\|X-Y\|_F^2+\lambda T(X)$ and denote $\widehat{X_1}=U^TX_1V$. Then $\widehat{X_1}$ will be an optimal solution of $\arg\min_{X\in\Re^{m\times n}}\frac{1}{2}\|X-\mathrm{Diag}(\sigma)\|_F^2+\lambda T(X)$. Based on the above proof and Theorem 2.3.1, we have $\widehat{X_1}=Diag(g_{\lambda,a}(\sigma))$. Since U and V are unitary, $X_1=U\widehat{X_1}V^T=X^s$. We proved that the matrix $G_{\lambda,a}(Y)$ is the unique optimal solution for the optimization problem. The proof is complete. □

Lemma 3.1.2. *For any fixed $\lambda > 0$, $\mu > 0$ and matrix $Z \in \Re^{m\times n}$, let $X^s = G_{\lambda\mu,a}(B_\mu(Z))$, then X^s is the unique global minimizer of the problem $\min\limits_{X\in\Re^{m\times n}} C_{\lambda,\mu}(X,Z)$, where the matrix function $C_{\lambda,\mu}(X,Z)$ is defined in (0.3) of section 3.*

Proof. First, we will rewrite the formula of $C_{\lambda,\mu}(X,Z)$. Note that $\mathscr{A}(X)$ and $\mathscr{A}(Z)$ are vectors in space $\Re^p$. Thus in the formula of $C_{\lambda,\mu}(X,Z)$, there exist norms and inner products for both matrices and vectors. By definition,

$$\begin{aligned} C_{\lambda,\mu}(X,Z) &= \tfrac{1}{2}\|X\|_F^2 - \langle X,Z\rangle + \tfrac{1}{2}\|Z\|_F^2 + \lambda\mu T(X) + \tfrac{\mu}{2}\|b\|_2^2 \\ &\quad -\mu(\mathscr{A}(X), b-\mathscr{A}(Z)) - \tfrac{\mu}{2}\|\mathscr{A}(Z)\|_2^2 \\ &= \tfrac{1}{2}\|X\|_F^2 + \tfrac{1}{2}\|Z\|_F^2 + \tfrac{\mu}{2}\|b\|_2^2 - \tfrac{\mu}{2}\|\mathscr{A}(Z)\|_2^2 \\ &\quad +\lambda\mu T(X) - \langle\ X, Z+\mu\mathscr{A}^*(b-\mathscr{A}(Z))\ \rangle \\ &= \tfrac{1}{2}\|X - B_\mu(Z)\|_F^2 + \lambda\mu T(X) \\ &\quad -\tfrac{1}{2}\|B_\mu(Z)\|_F^2 + \tfrac{1}{2}\|Z\|_F^2 + \tfrac{\mu}{2}\|b\|_2^2 - \tfrac{\mu}{2}\|\mathscr{A}(Z)\|_2^2 \end{aligned} \tag{1.16}$$

Thus if we fix matrix Z,

$$\underset{X\in\Re^{m\times n}}{\arg\min}\, C_{\lambda,\mu}(X,Z) = \underset{X\in\Re^{m\times n}}{\arg\min}\, \tfrac{1}{2}\|X - B_\mu(Z)\|_F^2 + \lambda\mu T(X) \tag{1.17}$$

Then by Theorem 3.1.1, X^s is the unique global minimizer. □

Theorem 3.1.2. *For fixed parameters, $\lambda > 0$ and $0 < \mu < \|\mathscr{A}\|_2^{-2}$. If X^* is a global minimizer for problem $C_\lambda(X)$, then X^* is the unique global minimizer for problem $\min\limits_{X\in\Re^{m\times n}} C_{\lambda,\mu}(X,X^*)$.*

Proof.

$$\begin{aligned}
C_{\lambda,\mu}(X,X^*) &= \mu\{\tfrac{1}{2}\|\mathscr{A}(X)-b\|_2^2+\lambda T(X)\} \\
&\quad +\tfrac{1}{2}\{\|X-X^*\|_F^2-\mu\|\mathscr{A}(X)-\mathscr{A}(X^*)\|_2^2\} \\
&\geq \mu\{\ \tfrac{1}{2}\|\mathscr{A}(X)-b\|_2^2+\lambda T(X)\ \}=\mu C_\lambda(X) \\
&\geq \mu C_\lambda(X^*)=C_{\lambda,\mu}(X^*,X^*)
\end{aligned} \tag{1.18}$$

The first inequality is due to the fact:

$$\begin{aligned}
\|\mathscr{A}(X)-\mathscr{A}(X^*)\|_2^2 &= \|A\text{vec}(X)-A\text{vec}(X^*)\|_2^2 \\
&\leq \|A\|_2^2\ \|\text{vec}(X-X^*)\|_2^2 \\
&\leq \|\mathscr{A}\|_2^2\ \|X-X^*\|_F^2
\end{aligned} \tag{1.19}$$

From the above inequalities, we know that X^* is an optimal solution for $\min\limits_{X\in\Re^{m\times n}} C_{\lambda,\mu}(X,X^*)$. The uniqueness of X^* follows from Lemma 3.1.2.

□

By the above Theorems and Lemmas, if X^* is a global minimizer of $C_\lambda(X)$, it is also the unique global minimizer of $C_{\lambda,\mu}(X,Z)$ with $Z=X^*$, which has the closed form solution formula. Thus we arrive at the following fixed point equation for this global minimizer X^*:

$$X^*=G_{\lambda\mu,a}(B_\mu(X^*)). \tag{1.20}$$

Suppose the SVD for matrix $B_\mu(X^*)$ is $U\text{Diag}(\sigma_b^*)V^T$, then

$$X^*=U\text{Diag}(g_{\lambda\mu,a}(\sigma_b^*))V^T,$$

which means that the singular values of X^* satisfy $\sigma_i^*=g_{\lambda\mu,a}(\sigma_{b,i}^*)$, for $i=1,...,m$.

3.2 TS1 thresholding algorithms

Next we will utilize the fixed point equation (1.20) to derive two thresholding algorithms for TS1 regularized problem (1.6). As in [71, 72], from the equation $X^* = G_{\lambda\mu,a}(B_\mu(X^*)) = U\mathrm{Diag}(g_{\lambda\mu,a}(\sigma))V^T$, we will replace the optimal matrix X^* with X^k on the left and X^{k-1} on the right at the k-th step of iteration as:

$$\begin{aligned} X^k &= G_{\lambda\mu,a}(B_\mu(X^{k-1})) \\ &= U^{k-1}\,\mathrm{Diag}\left(g_{\lambda\mu,a}(\sigma^{k-1})\right)V^{k-1,T}, \end{aligned} \tag{2.21}$$

where unitary matrices U^{k-1}, V^{k-1} and singular values $\{\sigma^{k-1}\}$ come from the SVD decomposition of matrix $B_\mu(X^{k-1})$. Operator $g_{\lambda\mu,a}(\cdot)$ is defined in (3.46), and

$$g_{\lambda\mu,a}(w) = \begin{cases} 0, & \text{if } |w| < t; \\ h_{\lambda\mu}(w), & \text{if } |w| \geq t. \end{cases} \tag{2.22}$$

Recall that the thresholding parameter t is:

$$t = \begin{cases} t_2^* = \lambda\mu\frac{a+1}{a}, & \text{if } \lambda \leq \frac{a^2}{2(a+1)\mu}; \\ t_3^* = \sqrt{2\lambda\mu(a+1)} - \frac{a}{2}, & \text{if } \lambda > \frac{a^2}{2(a+1)\mu}. \end{cases} \tag{2.23}$$

With an initial matrix X^0, we obtain an iterative algorithm, called TS1 iterative thresholding (IT) algorithm. It is the basic TS1 iterative scheme. Later, two adaptive and more efficient IT algorithms (TS1-s1 and TS1-s2) will be introduced.

3.2.1 Semi-Adaptive Thresholding Algorithm – TS1-s1

We begin with formulating an optimal condition for regularization parameter λ, which serves as the basis for the parameter selection and updating in this semi-adaptive algorithm.

Suppose the optimal solution matrix X has rank r, by prior knowledge or estimation. Here, we still assume $m \leq n$. For any μ, denote $B_\mu(X) = X + \mu A^T(b - \mathscr{A}(X))$ and $\{\sigma_i\}_{i=1}^m$ are the m non-negative singular values for $B_\mu(X)$.

Suppose that X^* is the optimal solution matrix of (1.6), and the singular values of matrix $B_\mu(X^*)$ are denoted as $\sigma_1^* \geq \sigma_2^* \geq ... \geq \sigma_m^*$. Then by the fixed equation (1.20), the following inequalities hold:

$$\begin{aligned} &\sigma_i^* > t \Leftrightarrow i \in \{1,2,...,r\}, \\ &\sigma_j^* \leq t \Leftrightarrow j \in \{r+1, r+2, ..., m\}, \end{aligned} \tag{2.24}$$

where t is our threshold value. Recall that $t_3^* \leq t \leq t_2^*$. So

$$\begin{aligned} &\sigma_r^* \geq t \geq t_3^* = \sqrt{2\lambda\mu(a+1)} - \tfrac{a}{2}; \\ &\sigma_{r+1}^* \leq t \leq t_2^* = \lambda\mu\tfrac{a+1}{a}. \end{aligned} \tag{2.25}$$

It follows that

$$\lambda_1 \equiv \frac{a\sigma_{r+1}^*}{\mu(a+1)} \leq \lambda \leq \lambda_2 \equiv \frac{(a+2\sigma_r^*)^2}{8(a+1)\mu}$$

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

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

(I) When $\lambda_1 \leq \frac{a^2}{2(a+1)\mu}$, set $\lambda^* = \lambda_1$.

Then we will have $\lambda^* \leq \frac{a^2}{2(a+1)\mu}$ and thus thresholding value $t = t_2^*$;

(II) When $\lambda_1 > \frac{a^2}{2(a+1)\mu}$, set $\lambda^* = \lambda_2$.

Then $\lambda^* > \frac{a^2}{2(a+1)\mu}$. Thus we choose thresholding value $t = t_3^*$.

In practice, we approximate $B_\mu(X^*)$ by $B_\mu(X^n)$ in the above formula, that is, the i-th singular value σ_i^* is approximated by σ_i^n at the n-th iteration step. Thus, we have $\lambda_{1,n} = \dfrac{a\sigma_{r+1}^n}{\mu(a+1)}$, and $\lambda_{2,n} = \dfrac{(a+2\sigma_r^n)^2}{8(a+1)\mu}$. We choose optimal parameter λ at the n-th step as

$$\lambda_n = \begin{cases} \lambda_{1,n}, & \text{if} \quad \lambda_{1,n} \le \frac{a^2}{2(a+1)\mu}, \\ \lambda_{2,n}, & \text{if} \quad \lambda_{1,n} > \frac{a^2}{2(a+1)\mu}. \end{cases} \tag{2.26}$$

This way, we obtain an adaptive iterative algorithm without pre-setting the regularization parameter λ. The TL1 parameter a is still free and needs to be selected beforehand. Thus the algorithm is overall semi-adaptive, called TS1-s1 for short and summarized in Algorithm 6.

```
Algorithm 6: TS1-s1 thresholding algorithm
    Initialize:  Given $X^0$ and parameter $\mu$ and $a$.
    while NOT converged do
        1. $Y^n = B_\mu(X^n) = X^n - \mu\mathscr{A}^*(\mathscr{A}(X^n) - b)$,
           and compute SVD of $Y^n$ as $Y^n = U\,\mathrm{Diag}(\sigma)\,V^T$ ;
        2. Determine the value for $\lambda_n$ by (2.26),
           then obtain the thresholding value $t_n$ by (2.23);
        3. $X^{n+1} = G_{\lambda_n\mu,a}(Y^n) = U\mathrm{Diag}(g_{\lambda_n\mu,a}(\sigma))V^T$;
        Then, $n \to n+1$.
    end while
```

3.2.2 Adaptive Thresholding Algorithm – TS1-s2

Different from TS1-s1 where the parameter 'a' needs to be determined manually, here at each iterative step, we choose $a=a_n$ such that equality $\lambda_n=\frac{a_n^2}{2(a_n+1)\mu_n}$ holds. The threshold value t is given by a single formula with $t=t_3^*=t_2^*$.

Putting $\lambda=\frac{a^2}{2(a+1)\mu}$ at critical value, the parameter a is expressed as:

$$a=\lambda\mu+\sqrt{(\lambda\mu)^2+2\lambda\mu}. \tag{2.27}$$

The threshold value is:

$$t=\lambda\mu\frac{a+1}{a}=\frac{\lambda\mu}{2}+\frac{\sqrt{(\lambda\mu)^2+2\lambda\mu}}{2}. \tag{2.28}$$

Let X^* be the TL1 optimal solution and σ^* be the singular values for matrix $B_\mu(X^*)$. Then we have the following inequalities:

$$\begin{aligned} &\sigma_i^*>t\Leftrightarrow i\in\{1,2,...,r\},\\ &\sigma_j^*\leq t\Leftrightarrow j\in\{r+1,r+2,...,m\}. \end{aligned} \tag{2.29}$$

So, for parameter λ, we have:

$$\frac{2(\sigma_{r+1}^*)^2}{1+2\sigma_{r+1}^*}\leq\lambda\leq\frac{2(\sigma_r^*)^2}{1+2\sigma_r^*}.$$

Once the value of λ is determined, the parameter a is given by (2.27).

In the iterative method, we approximate the optimal solution X^* by X^n and further use $B_\mu(X^n)$'s singular values $\{\sigma_i^n\}_i$ to replace those of $B_\mu(X^*)$. The resulting parameter selection is:

$$\begin{aligned}&\lambda_n = \frac{2(\sigma^n_{r+1})^2}{1+2\sigma^n_{r+1}};\\&a_n = \lambda_n\mu_n + \sqrt{(\lambda_n\mu_n)^2 + 2\lambda_n\mu_n}.\end{aligned} \tag{2.30}$$

In this algorithm (TS1-s2 for short), only parameter μ is fixed, satisfying inequality $\mu \in (0, \|A\|^{-2})$. Its algorithm is summarized in Algorithm 7.

Algorithm 7: TS1-s2 thresholding algorithm

```
Initialize:   Given $X^0$ and parameter $\mu$.
while NOT converged do
    1. $Y^n = X^n - \mu\mathscr{A}^*(\mathscr{A}(X^n) - b)$, and compute SVD of $Y^n$ as $Y^n = U\,\mathrm{Diag}(\sigma)\,V^T$ ;
    2. Determine the values for $\lambda^n$ and $a^n$ by (2.30),
       then update threshold value $t^n = \lambda^n\mu\frac{a^n+1}{a^n}$;
    3. $X^{n+1} = G_{\lambda^n\mu,a^n}(Y^n) = U\,\mathrm{Diag}(g_{\lambda^n\mu,a}(\sigma))\,V^T$;
    Then $n \to n+1$.
end while
```

3.3 Numerical experiments

In this section, we present numerical experiments to illustrate the effectiveness of our Algorithms: semi-adaptive TS1-s1 and adaptive TS1-s2, compared with several state-of-art solvers on matrix completion problems [1]. The comparison solvers include:

- LMaFit [62],
- FPCA [44],
- sIRLs-q [47],

[1] TS1 matlab codes can be downloaded from https://github.com/zsivine/TS1-algorithms

- IRucLq-M [36],
- LRGeomCG [61]

The code LMAFit solves a low-rank factorization model, instead of computing SVD which usually takes a big chunk of computation time. Also part of its codes is written in C, same as LRGeomCG. So once this method converges, it is the fastest method among all comparisons. All others codes are implemented under Matlab environment and involve SVD approximated by fast Monte Carlo algorithms [18, 19]. FPCA is a nuclear norm minimization code, while sIRLs-q and IRucLq-M are iterative reweighted least square algorithms for Schatten-q quasi-norm optimizations. LRGeomCG algorithm explores matrix completion based on Riemannian optimization. It tries to minimize the least-square distance on the sampling set over the Riemannian manifold of fixed-rank matrices. When the rank information is known or well approximated, this method is efficient and accurate, as shown in these experiments below, especially for standard Gaussian matrices. But a drawback of LRGeomCG is that the rank of the manifold is fixed. Basically, it is hard for it to handle unknown rank cases.

In our TS1 algorithms, MC SVD algorithm [19] is implemented at each iteration step, same as FPCA. We also tried another fast SVD approximation algorithms, but MC SVD is the most suitable one, satisfying both speed and accuracy requirements in one iterative algorithm. All our tests were performed on a *Lenovo* desktop: 16 GB of RAM and Intel@ Core Quad processor i7-4770 with CPU at 3.40GHz under 64-bit Ubuntu system.

We tested and compared these solvers on low rank matrix completion problems under various conditions, including multivariate Gaussian, uniform and χ^2 distributions. We also tested the algorithms on grayscale image recovery from partial observations (image inpainting).

3.3.1 Implementation details

In the following series of tests, we generated random matrices

$$M = M_L M_R^T \in \mathcal{R}_{m\times n},$$

where matrices M_L and M_R are in spaces $\mathcal{R}^{m\times r}$ and $\mathcal{R}^{n\times r}$ respectively.

By setting parameter r to be small, we obtain a low rank matrix M with rank at most r. After this step, we uniformly random-sampled a subset ω with p entries from M. The following quantities help to quantify the difficulty of a recovery problem.

- SR (Sampling ratio): SR $= p/mn$.
- FR (Freedom ratio): FR $= r(m+n-r)/p$, which is the freedom of rank r matrix divided by the number of measurement. According to [44] , if FR >1, there are infinite number of matrices with rank r and the given entries.
- r_m (Maximum rank with which the matrix can be recovered):

  $$r_m = \lfloor \frac{m+n-\sqrt{(m+n)^2-4p}}{2} \rfloor \quad \textit{(floor function)},$$

  which is defined as the largest rank such that FR ≤ 1.

The TS1 thresholding algorithms do not guarantee a global minimum in general, similar to non-convex schemes in 1-dimensional compressed sensing problems. Indeed we observe that TS1 thresholding with random starts may get stuck at local minima especially when parameter FR (freedom ratio) is high or the matrix completion is difficult. A good initial matrix X^0 is important for thresholding algorithms. In our numerical experiments, instead of choosing $X^0=0$ or random, we set X^0 equal to matrix M whose elements are as observed on Ω and zero elsewhere.

The stopping criterion is

$$\frac{\|X^{n+1}-X^n\|_F}{\max\{\|X^n\|_F,1\}}\leq tol$$

where X^{n+1} and X^n are numerical results from two contiguous iterative steps, and tol is a moderately small number. In all these following experiments, we fix $tol=10^{-6}$ with maximum iteration steps 1000.

We also use the relative error

$$\text{rel.err}=\frac{\|X_{opt}-M\|_F}{\|M\|_F} \tag{3.31}$$

to estimate the closeness of X_{opt} to M, where X_{opt} is the "optimal" solution produced by all numerical algorithms.

Rank estimation

For thresholding algorithms, rank r is the most important parameter, especially for our TS1 methods, where thresholding value t is determined based on r. If the true rank r is unknown, we adopt the rank decreasing estimation method (also called maximum eigengap method) as in [36, 62], thereby extending both TS1-s1 and TS1-s2 schemes to work with an overestimated initial rank parameter K. In the following tests, unless otherwise specified, we set $K=\lfloor 1.5r\rfloor$. The idea behind this estimation method is as follows. Suppose that at step n, our current matrix is X. The eigenvalues of X^TX are arranged with descending order and $\lambda_{r_{min}}\geq\lambda_{r_{min}+1}\geq...\geq\lambda_{K+1}>0$ is the r_{min}-th through $K+1$-th eigenvalues of X^TX, where r_{min} is manually specified minimum rank estimate. Then we compute the quotient sequence

$\widehat{\lambda}_i = \lambda_i / \lambda_{i+1}$, $i = r_{min}, ..., K$. Let

$$\widetilde{K} = \underset{r_{min} \le i \le K}{\arg\min}\, \widehat{\lambda}_i,$$

the corresponding index for maximal element of $\{\widehat{\lambda}_i\}$. If the eigenvalue gap indicator

$$\tau = \widehat{\lambda}_{\widetilde{K}}(K - r_{min} + 1) / \sum_{i \neq \widetilde{K}} \widehat{\lambda}_i \quad > 10,$$

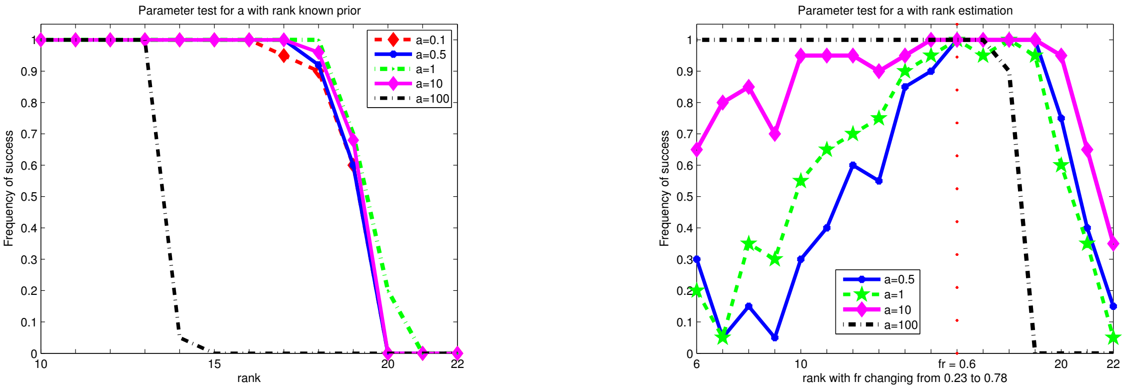
we adjust our rank estimator from K to $\widetilde{K}$. During numerical simulations, we did this adjustment only once for each problem. In most cases, this estimation adjustment is quite satisfactory and the adjusted estimate is very close to the true rank r.

Choice of a: optimal parameter testing for TS1-s1.

A major difference between TS1-s1 and TS1-s2 is the choice of parameter a, which influences the behaviour of penalty function $\rho_a(\cdot)$ of TS1. When 'a' tends to zero, the function $T(X)$ approaches the rank.

We tested TS1-s1 on small size low rank matrix completion with different 'a' values, varying among $\{0.1, 0.5, 1, 10, 100\}$, for both known rank scheme and the scheme with rank estimation. In these tests, $M = M_L M_R^T$ is a 100×100 random matrix, where M_L and M_R are generated under i.i.d standard normal distribution. The rank r of M varies from 10 to 22.

For each value of 'a', we conducted 50 independent tests with different M and sample index set ω. We declared M to be recovered successfully if the relative error (3.31) was less than 5×10^{-3}. The test results for known rank scheme and rank estimation scheme are both shown in Figure 3.1. The success rate curves of rank estimation scheme are not as clustered as those of known rank scheme. In order to clearly identify the optimal parameter 'a', we ignored the curve of $a = 0.1$ in the right figure as it is always below all others. The vertical red dotted line there indicates the position where FR $= 0.6$.



Rank is known Rank is estimated

Figure 3.1: Optimal parameter test for semi-adaptive method: TS1-s1

It is interesting to see that for known rank scheme, parameter $a = 1$ is the optimal strategy, which coincides with the optimal parameter setting in [71]. It is observed that when we use thresholding algorithm under transformed L1 (TL1) or transformed Schatten-1 (TS1) quasi norm, it is usually optimal to set $a = 1$ with given information of sparsity or rank. However, for the scheme with rank estimation, it is more complicated. Based on our tests, if FR < 0.6, it is better to set $a \geq 100$ to reach good performance. On the other hand, if FR > 0.6, $a = 10$ is nearly the optimal choice. So for all the following tests, when we apply TS1-s1 with rank estimation, the parameter a is set to be

$$a = \begin{cases} 1000, & \text{if } \ \mathrm{FR} < 0.6; \\ 10, & \text{if } \ \mathrm{FR} \geq 0.6. \end{cases}$$

In applications where FR is not available, we suggest to use $a = 10$, since its performance is also acceptable if FR < 0.6.

3.3.2 Completion of Random Matrices

The ground truth matrix M is generated as the matrix product of two low rank matrices M_L and M_R. Their dimensions are $m \times r$ and $n \times r$ respectively, with $r \ll \min(m,n)$. In these following experiments, except clearly stated, M_L and M_R are generated with multivariate normal distribution $\mathcal{N}(\mu,\Sigma)$, with $\mu = 1$ and

$$\Sigma = \{(1 - cov) * I_{(i=j)} + cov\}_{r \times r}$$

determined by parameter cov. Thus matrix $M = M_L M_R^T$ has rank at most r.

It is known that success recovery is related to FR. The higher FR is, the harder it is to recover the original low rank matrix. In the first batch of tests, we varied rank r and fixed all other parameters, i.e. matrix size (m,n), sampling rate (sr). Thus FR was changing along with rank.

It is observed that the performance of TS1-s1 and TS1-s2 are very different, due to adopting single or double thresholds. TS1-s2 uses only one (smooth) thresholding scheme with changing parameter a. It converges faster than TS1-s1 when the rank is known, see subsection 3.3.2. On the other hand, TS1-s1 utilizes two (smooth and discontinuous) thresholding schemes, and is more robust in case of overestimated rank. TS1-s1 outperforms TS1-s2 when rank estimation is used in lieu of the true rank value, see subsection 3.3.2. IRucL-q method is found to be very robust for varied covariance and rank estimation, yet it underperforms TS1 methods at high FR, even with more computing time. Though TS1 methods rely on the same rank estimation method as IRucL-q, IRucL-q achieves the best results in the absence of true rank value. A possible reason is that in IRucL-q iterations, the singular values of matrix X are computed more accurately. In TS1, singular values are computed by fast Monte Carlo method at every iteration. Due to random sampling of Monte Carlo method, there are more errors especially at the beginning stage of iteration. The resulting matrices

X^n may cause less accurate rank estimation.

Matrix completion with known rank

In this subsection, we implemented all six algorithms under the condition that true rank value is given. They are TS1-s1, TS1-s2, sIRLS-q, IRucL-q, LMaFit and LRGeomCG. We skipped FPCA since rank is always adaptively estimated there.

Gaussian matrices with different ranks

In these tests, matrix $M = M_L M_R^T$ was generated under uncorrelated normal distribution with $\mu = 1$. We conducted tests both on low dimensional matrices with $m = n = 100$ (Table 3.1) and high dimensional matrices with $m = n = 1000$ (Table 3.2). Tests on non-square matrices with $m \neq n$ show similar results.

In Table 3.1, rank r varies from 5 to 18, while FR increases from 0.2437 up to 0.8190. For lower rank (less than 15), LMaFit is the best algorithm with low relative errors and fast convergence speed. Part of the reason is that this method does not involve SVD (singular value decomposition) operations during iteration.

LRGeomCG approaches the performance of LMaFit when $r \leq 10$. However, as FR values are above 0.7, it became hard for LMaFit to find truth low rank matrix M. Its performance is not as good as stated in paper [61] with possible reason that we generate M with mean μ equal to 1, instead of 0 in [61]. We also tested LRGeomCG with $\mu = 0$ where it has very small relative error and also fast convergence rate.

It is also noticed that in Table 3.1, the two TS1 algorithms performed very well and remained stable for different FR values. At similar order of accuracy, the TL1s are faster than IRucL-q.

Table 3.1: Comparison of TS1-s1, TS1-s2, sIRLS-q, IRucL-q, LMaFit and LRGeomCG on recovery of uncorrelated multivariate Gaussian matrices at known rank, $m=n=100$, SR $=0.4$, with stopping criterion $tol=10^{-6}$.

Problem		TS1-s1		TS1-s2		sIRLS-q*	
rank	FR	rel.err	time	rel.err	time	rel.err	time
5	0.2437	1.89e-05	0.11	7.58e-07	0.13	7.09e-06	0.80
6	0.2910	7.13e-06	0.14	7.37e-07	0.15	8.59e-06	1.01
7	0.3377	1.39e-05	0.15	6.34e-07	0.17	8.14e-06	1.09
8	0.3840	2.04e-05	0.16	7.70e-07	0.20	1.31e-05	1.43
9	0.4298	2.08e-05	0.23	9.97e-07	0.25	2.02e-05	1.88
10	0.4750	3.26e-05	0.33	1.11e-06	0.34	1.93e-02	4.49
14	0.6510	1.10e-05	0.53	1.03e-05	0.52	—	—
15	0.6937	1.05e-05	0.66	9.88e-06	0.64	—	—
16	0.7360	3.86e-05	0.91	1.79e-05	0.87	—	—
17	0.7778	1.50e-04	1.03	7.10e-05	1.00	—	—
18	0.8190	5.63e-04	1.00	4.15e-04	1.00	—	—

Problem		IRucL-q		LMaFit		LRGeomCG	
rank	FR	rel.err	time	rel.err	time	rel.err	time
5	0.2437	7.86e-06	1.82	1.96e-06	0.02	1.03e-06	0.03
6	0.2910	1.14e-05	2.15	2.18e-06	0.02	1.22e-06	0.04
7	0.3377	1.28e-05	2.24	2.27e-06	0.03	1.37e-06	0.05
8	0.3840	3.03e-05	2.33	2.67e-06	0.03	1.66e-06	0.06
9	0.4298	1.68e-04	2.38	3.21e-06	0.05	1.88e-06	0.07
10	0.4750	3.21e-04	2.49	3.54e-06	0.08	1.87e-06	0.08
14	0.6510	3.80e-05	7.25	5.74e-06	0.21	3.20e-02	0.34
15	0.6937	5.28e-05	9.29	5.87e-02	0.33	3.49e-02	0.47
16	0.7360	7.57e-05	12.34	1.44e-01	0.34	1.91e-01	0.99
17	0.7778	9.40e-05	15.31	3.80e-01	0.39	5.73e-01	0.71
18	0.8190	1.49e-04	22.27	4.43e-01	0.40	9.17e-01	0.94

* Notes: 1. The sIRLS-q iterations did not converge when rank >14 and FR ≥ 0.65. Comparison is skipped over this range. *Results for rank (11,12,13) are skipped as they differ little from rank 14. Similar rank samplings occur in Tables 4.(2-3), 4.(5-6).*

2. Matrix M is generated from multivariate normal distribution with mean $\mu=1$, instead of 0.

Table 3.2: Numerical experiments on recovery of uncorrelated multivariate Gaussian matrices at known rank, $m = n = 1000$, SR $= 0.3$.

Problem		TS1-s1		TS1-s2		sIRLS-q	
rank	FR	rel.err	time	rel.err	time	rel.err	time
50	0.3250	5.95e-06	8.06	5.88e-06	6.95	4.85e-06	45.20
70	0.4503	6.94e-06	13.37	6.78e-06	11.95	2.46e-02	128.65
90	0.5730	7.83e-06	22.13	7.77e-06	18.81	9.86e-02	206.32
110	0.6930	1.23e-04	29.91	3.47e-05	29.50	2.27e-01	282.84

Problem		IRucL-q		LMaFit		LRGeomCG	
rank	FR	rel.err	time	rel.err	time	rel.err	time
50	0.3250	9.55e-06	485.30	1.74e-06	6.04	1.11e-06	8.31
70	0.4503	3.77e-05	606.95	3.54e-02	23.20	1.50e-06	20.87
90	0.5730	4.16e-04	623.37	1.60e-01	24.94	2.13e-06	52.77
110	0.6930	2.41e-03	640.66	2.45e-01	29.19	3.22e-06	112.30

For large size matrices ($m = n = 1000$), rank r is varied from 50 to 110, see table 3.2. The sIRLS-q and LMaFit only worked for lower FR. IRucL-q can still produce satisfactory results with relative error around 10^{-3}, but its iterations took longer time. In [36], it was carried out by high speed-performance CPU with many cores. Here we used an ordinary processor with only 4 cores and 8 threads. It is believed that with a better machine, IRucL-q will be much faster, since parallel computing is embedded in its codes. As seen in the table, LRGeomCG is always convergent and achieves almost same accuracy with TS1-s1 and TS1-s2. However, its computation time grows fast with increasing rank.

A little difference between the two TS1 algorithms begins to emerge when the matrix size is large. Although when rank is given, they all performed better than other schemes, adaptive TS1-s2 is a little faster than semi-adaptive TS1-s1. It is believed by choosing optimal parameter a, TS1-s1 will be improved. The parameter a is related to matrix M, i.e. how it is generated, its inner structure, and dimension. In TS1-s2, the value of parameter a does not need to be manually determined.

Table 3.3: Numerical experiments on multivariate Gaussian matrices with varying covariance at known rank, $m = n = 100$, SR $= 0.4$.

Problem		TS1-s1		TS1-s2		sIRLS-q	
rank	cor	rel.err	time	rel.err	time	rel.err	time
5	0.5	6.44e-06	0.17	5.74e-07	0.12	3.35e-02	3.75
5	0.6	7.28e-06	0.28	7.15e-07	0.13	1.34e-01	5.58
5	0.7	3.32e-02	0.58	7.65e-07	0.17	2.15e-01	6.16
8	0.4	7.55e-06	0.34	7.96e-07	0.21	1.43e-01	6.47
8	0.5	9.84e-03	0.51	6.14e-06	0.19	2.68e-01	6.19
8	0.6	3.01e-02	0.81	7.71e-06	0.23	2.95e-01	6.26
8	0.7	6.86e-02	0.86	7.16e-06	0.50	3.33e-01	6.80
Problem		IRucL-q		LMaFit		LRGeomCG	
rank	cor	rel.err	time	rel.err	time	rel.err	time
5	0.5	8.21e-06	1.86	2.48e-02	0.07	1.12e-06	0.06
5	0.6	8.76e-06	1.85	4.48e-02	0.15	6.98e-02	0.09
5	0.7	1.37e-05	1.71	1.10e-01	0.27	1.22e-01	0.11
8	0.4	1.92e-05	2.50	1.98e-02	0.18	5.42e-02	0.17
8	0.5	1.38e-05	2.54	1.21e-01	0.25	1.17e-01	0.17
8	0.6	1.40e-05	2.51	1.85e-01	0.27	1.83e-01	0.23
8	0.7	1.10e-05	2.35	2.44e-01	0.25	2.21e-01	0.29

Gaussian Matrices with Different Covariance

In this subsection, the rank r, the sampling rate, and the freedom ratio FR are fixed. We varied parameter *cov* to generate covariance matrices of multivariate normal distribution.

In Table 3.3, we chose two rank values, $r = 5$ and $r = 8$. It is harder to recover the original matrix M when it is more coherent. IRucL-q does better in this regime. Its mean computing time and relative errors are less influenced by the changing *cov*. Results on large size matrices are shown in Table 3.4. TS1-s2 scheme is much better than TS1-s1, both in relative error and computing time. In small size matrix experiments, TS1-s2 is the best among comparisons.

In Table 3.4, we fixed rank $= 30$ with *cov* among $\{0.1, ..., 0.7\}$. TS1-s2 is still satisfactory both in accuracy and speed for low covariance (i.e $cov \leq 0.6$). However, for $cov \geq 0.7$, relative errors increased from 10^{-6} to around 10^{-4}. It is also observed that IRucL-q algorithm is very stable and robust under covariance change.

Table 3.4: Numerical experiments on multivariate Gaussian matrices with varying covariance at known rank, $m = n = 1000$, SR $= 0.4$.

Problem		TS1-s1		TS1-s2		sIRLS-q	
rank	cor	rel.err	time	rel.err	time	rel.err	time
30	0.1	3.07e-06	9.71	3.07e-06	3.98	4.36e-07	13.80
30	0.2	2.90e-06	11.07	2.94e-06	3.92	1.28e-05	33.89
30	0.3	5.54e-03	26.64	3.02e-06	4.13	6.65e-02	46.02
30	0.4	1.19e-02	28.58	3.08e-06	4.31	1.08e-01	50.95
30	0.5	4.76e-02	34.25	2.89e-06	5.89	1.50e-01	52.64
30	0.6	6.89e-02	35.69	2.89e-06	10.28	1.89e-01	55.70
30	0.7	8.01e-02	33.92	6.99e-04	20.09	2.03e-01	51.03
Problem		IRucL-q		LMaFit		LRGeomCG	
rank	cor	rel.err	time	rel.err	time	rel.err	time
30	0.1	3.13e-06	222.90	1.19e-06	1.83	6.77e-07	4.88
30	0.2	3.16e-06	221.34	1.14e-06	3.16	5.68e-07	8.84
30	0.3	3.05e-06	218.57	1.21e-06	6.93	5.45e-03	15.45
30	0.4	3.29e-06	214.52	2.06e-02	14.72	4.82e-02	19.15
30	0.5	3.12e-06	209.05	6.45e-02	17.34	8.41e-02	20.99
30	0.6	3.30e-06	207.94	9.09e-02	18.38	1.42e-01	21.81
30	0.7	3.15e-06	210.06	1.15e-01	16.37	1.67e-01	21.63

Matrices from other distributions

We also compare algorithms with other distributions, including (0,1) uniform distribution and Chi-square distribution with k = 1 (degree of freedom). All other parameters are same as Table 3.1. The results are displayed at Table 3.5 (uniform distribution) and Table 3.6 (Chi-square distribution). Only partial numerical results are showed here with rank $r = 7, 8, 9, 10, 14, 15$. From these two tables, two TS1 algorithms have satisfying relative errors and stable performance, same as IRuccL-q. For these two non-Gaussian distributions, it becomes harder to successfully recover low rank matrix for LMaFit and LRGeomCG, especially when rank $r > 10$.

Table 3.5: Comparison with random matrices generated from (0,1) uniform distribution. Rank r is given and $m=n=100$, SR $=0.4$, with stopping criterion $tol=10^{-6}$.

Problem		TS1-s1		TS1-s2		sIRLS-q*	
rank	FR	rel.err	time	rel.err	time	rel.err	time
7	0.3377	5.67e-06	0.16	5.30e-06	0.14	7.30e-06	1.85
8	0.3840	6.73e-06	0.18	6.46e-06	0.15	1.96e-02	3.78
9	0.4298	9.13e-06	0.24	8.42e-06	0.20	—	—
10	0.4750	7.62e-06	0.27	7.12e-06	0.20	—	—
14	0.6510	2.23e-05	0.59	9.24e-06	0.44	—	—
15	0.6937	2.34e-05	0.81	1.12e-05	0.58	—	—
Problem		IRucL-q		LMaFit		LRGeomCG	
rank	FR	rel.err	time	rel.err	time	rel.err	time
7	0.3377	9.55e-06	5.00	1.98e-06	0.05	1.48e-06	0.08
8	0.3840	1.08e-05	4.86	2.41e-06	0.06	1.58e-06	0.10
9	0.4298	1.57e-05	6.48	2.26e-02	0.13	2.01e-06	0.14
10	0.4750	1.80e-05	7.09	7.28e-03	0.11	2.09e-06	0.13
14	0.6510	3.75e-05	13.15	1.66e-01	0.18	1.24e-01	0.44
15	0.6937	5.58e-05	17.14	2.18e-01	0.16	1.71e-01	0.76

Table 3.6: Comparison with random matrices generated from Chi-square distribution with $k=1$ (degree of freedom). Rank r is given and $m=n=100$, SR $=0.4$, with stopping criterion $tol=10^{-6}$.

Problem		TS1-s1		TS1-s2		sIRLS-q*	
rank	FR	rel.err	time	rel.err	time	rel.err	time
7	0.3377	9.09e-06	0.23	8.56e-06	0.20	1.82e-05	1.84
8	0.3840	1.06e-05	0.27	8.31e-06	0.22	1.69e-02	2.59
9	0.4298	9.90e-06	0.30	8.79e-06	0.25	—	—
10	0.4750	9.52e-06	0.33	8.64e-06	0.28	—	—
14	0.6510	1.48e-05	0.64	1.20e-05	0.58	—	—
15	0.6937	2.23e-05	0.83	1.32e-05	0.73	—	—
Problem		IRucL-q		LMaFit		LRGeomCG	
rank	FR	rel.err	time	rel.err	time	rel.err	time
7	0.3377	1.26e-05	5.65	3.08e-06	0.04	1.80e-06	0.05
8	0.3840	1.70e-05	7.15	3.29e-06	0.04	2.19e-06	0.06
9	0.4298	2.21e-05	8.33	3.75e-06	0.08	6.83e-03	0.11
10	0.4750	2.23e-05	8.56	4.25e-06	0.09	5.93e-02	0.14
14	0.6510	5.50e-05	14.69	1.44e-01	0.15	1.46e-01	0.34
15	0.6937	6.61e-05	17.75	2.54e-01	0.15	3.03e-01	0.57

Table 3.7: Numerical experiments for low rank matrix completion algorithms under rank estimation. True matrices are uncorrelated multivariate Gaussian, $m=n=100$, SR $=0.4$.

Problem		TS1-s1		TS1-s2		FPCA		IRucL-q		LMaFit	
rank	FR	rel.err	time	rel.err	time	rel.err	time	rel.err	time	rel.err	time
10	0.4750	7.46e-06	0.31	2.43e-03	0.38	2.26e-01	0.91	1.84e-05	3.41	2.64e-01	0.01
11	0.5198	1.04e-05	0.35	1.15e-02	0.52	2.23e-01	0.88	2.15e-05	4.09	2.48e-01	0.01
12	0.5640	9.94e-06	0.44	7.62e-03	0.54	2.28e-01	0.92	2.51e-05	4.46	2.44e-01	0.01
13	0.6078	3.71e-02	0.80	5.71e-03	0.68	2.25e-01	0.84	3.35e-05	5.61	2.24e-01	0.02
14	0.6510	7.02e-03	0.82	1.03e-03	0.65	2.23e-01	0.88	3.97e-05	6.41	2.19e-01	0.01
15	0.6937	4.96e-03	0.95	2.88e-03	0.92	2.18e-01	0.88	4.82e-05	7.86	2.12e-01	0.02

Matrix completion with rank estimation

We conducted numerical experiments on rank estimation schemes. The initial rank estimation is given as $1.5r$, which is a commonly used overestimate. FPCA [44] is included for comparison, while LRGeomCG and sIRLS-q are excluded. FPCA is a fast and robust iterative algorithm based on nuclear norm regularization.

We considered two classes of matrices: uncorrelated Gaussian matrices with changing rank; correlated Gaussian matrices with fixed rank ($r=5,10$). The results are shown in Table 3.7 and Table 3.8. It is interesting that under rank estimation, the semi-adaptive TS1-s1 fared much better than TS1-s2. In low rank and low covariance cases, TS1-s1 is the best in terms of accuracy and computing time among comparisons. However, in the regime of high covariance and rank, it became harder for TS1 methods to perform efficient recovery. IRucL-q did the best, being both stable and robust. In the most difficult case, at $rank=15$ and FR approximately equal to 0.7, IRucL-q can still obtain an accurate result with relative error around 10^{-5}.

Table 3.8: Numerical experiments on low rank matrix completion algorithms under rank estimation. True matrices are multivariate Gaussian with different covariance, $m = n = 100$, and SR $= 0.4$.

Problem		TS1-s1		TS1-s2		FPCA		IRucL-q		LMaFit	
rank	cor	rel.err	time	rel.err	time	rel.err	time	rel.err	time	rel.err	time
5	0.5	5.49e-06	0.20	6.77e-02	0.86	1.61e-05	0.12	7.50e-06	2.07	1.24e-01	0.01
5	0.6	5.45e-06	0.20	7.74e-02	0.91	1.69e-05	0.11	6.93e-06	1.76	9.12e-02	0.01
5	0.7	5.25e-06	0.25	1.04e-01	1.33	1.53e-05	0.12	4.71e-04	2.06	6.60e-02	0.01
10	0.5	1.10e-05	0.65	1.17e-01	1.14	1.21e-01	0.97	1.76e-05	3.35	9.66e-02	0.01
10	0.6	1.61e-02	0.76	1.32e-01	1.04	1.02e-01	0.86	2.72e-05	4.26	7.33e-02	0.01
10	0.7	9.14e-02	0.91	1.55e-01	0.93	9.11e-02	0.82	7.12e-04	4.59	5.06e-02	0.01

3.3.3 Image inpainting

As in [36, 62], we conducted grayscale image inpainting experiments to recover low rank images from partial observations, and compare with IRcuL-q and LMaFit algorithms. The 'boat' image (see Figure 3.2) is used to produce ground truth as in [36] with rank equal to 40 and at 512×512 resolution. Different levels of noisy disturbances are added to the original image M_o by the formula

$$M = M_o + \sigma \frac{\|M_o\|_F}{\|\varepsilon\|_F} \varepsilon,$$

where the matrix ε is a standard Gaussian.

Here we only applied scheme TS1-s2. For IRucL-q, we followed the setting in [36] by choosing $\alpha = 0.9$ and $\lambda = 10^{-2}\sigma$. Both fixed rank (LMaFit-fix) and increased rank (LMaFit-inc) schemes are implemented for LMaFit. We took fixed rank $r = 40$ for TS1-s2, LMaFit-fix and IRucL-q.

Computational results are in Table 3.9 with sampling ratios varying among $\{0.3, 0.4, 0.5\}$ and noise strength σ in $\{0.01, 0.05, 0.10, 0.15, 0.20, 0.25\}$. The performance for each algorithm is measured in CPU time, PSNR (peak-signal noise ratio), and MSE (mean squared error).

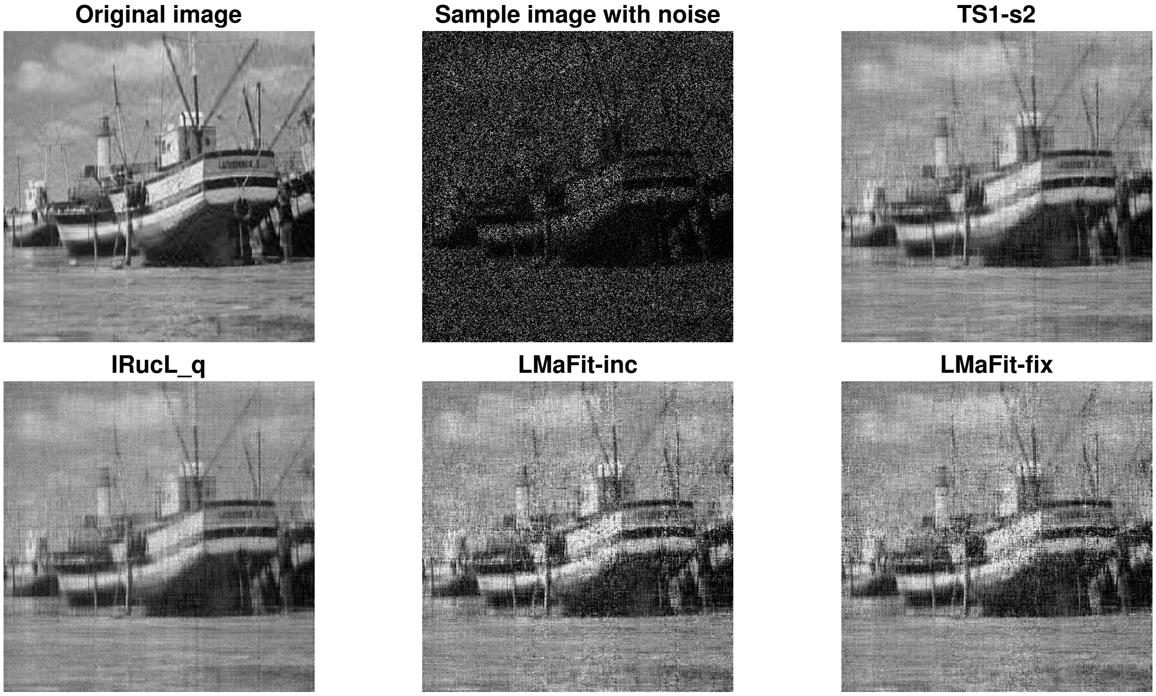


Figure 3.2: Image inpainting experiments with $\mathrm{SR}=0.3, \sigma=0.15$.

Here we focus more on PSNR values and placed the top 2 in bold for each experiment. We observed that IRucL-q and TS1-s2 fared about the same. Either one is better than LMaFit in most cases.

Table 3.9: Numerical experiments on boat image inpainting with algorithms TS1, IRcuL-q and LMaFit under different sampling ratio and noise levels.

Problem		TS1-s2			IRucL-q			LMaFit-inc			LMaFit-fix		
SR	σ	Time	PSNR	MSE	Time	PSNR	MSE	Time	PSNR	MSE	Time	PSNR	MSE
0.3	0.01	27.23	**44.21**	3.79e-5	85.97	43.28	4.70e-5	5.70	32.80	5.25e-4	2.17	**45.02**	3.15e-5
0.3	0.05	27.81	**30.55**	8.82e-4	58.25	**29.55**	1.11e-3	6.00	29.10	1.23e-3	2.81	29.28	1.18e-3
0.3	0.10	29.21	**24.89**	3.24e-3	24.26	**24.99**	3.17e-3	5.59	19.74	1.06e-2	5.74	18.52	1.41e-2
0.3	0.15	26.37	**22.57**	5.54e-3	27.61	**22.74**	5.33e-3	5.46	16.64	2.17e-2	4.84	15.98	2.52e-2
0.3	0.20	26.75	**20.89**	8.14e-3	24.45	**21.05**	7.85e-3	5.95	14.68	3.41e-2	3.52	14.03	3.95e-2
0.3	0.25	26.92	**19.60**	1.10e-2	23.75	**19.75**	1.06e-2	5.52	12.91	5.12e-2	1.85	12.73	5.33e-2
0.4	0.01	26.29	44.30	3.71e-5	80.19	43.25	4.74e-5	6.53	**44.84**	3.28e-5	2.93	**45.02**	3.15e-5
0.4	0.05	26.05	**30.58**	8.75e-4	63.20	**29.39**	1.15e-3	4.62	29.09	1.23e-3	3.12	27.91	1.62e-3
0.4	0.10	26.08	**24.74**	3.35e-3	32.58	**24.86**	3.27e-3	6.44	19.97	1.01e-2	8.00	19.19	1.21e-2
0.4	0.15	26.34	**22.57**	5.53e-3	26.30	**22.72**	5.35e-3	5.52	16.78	2.10e-2	2.86	16.21	2.40e-2
0.4	0.20	29.04	**20.89**	8.15e-3	20.73	**21.08**	7.81e-3	5.44	14.47	3.58e-2	2.25	14.43	3.61e-2
0.4	0.25	28.84	**19.56**	1.11e-2	20.48	**19.68**	1.08e-2	5.70	12.79	5.26e-2	2.35	12.57	5.54e-2
0.5	0.01	27.76	**44.26**	3.75e-5	82.42	43.30	4.67e-5	5.04	34.50	3.55e-4	2.79	**45.01**	3.15e-5
0.5	0.05	27.89	**30.54**	8.82e-4	64.19	29.47	1.13e-3	5.81	28.63	1.37e-3	2.79	**29.62**	1.09e-3
0.5	0.10	29.56	**24.80**	3.31e-3	30.50	**24.94**	3.21e-3	5.78	19.92	1.02e-2	3.54	19.09	1.23e-2
0.5	0.15	26.21	**22.59**	5.51e-3	24.24	**22.74**	5.32e-3	5.71	16.73	2.12e-2	2.67	16.32	2.33e-2
0.5	0.20	28.01	**20.89**	8.14e-3	22.51	**21.07**	7.82e-3	4.44	15.67	2.71e-2	2.42	14.38	3.65e-2
0.5	0.25	29.86	**19.52**	1.12e-2	18.32	**19.71**	1.07e-2	5.54	12.62	5.48e-2	3.24	12.74	5.32e-2

Chapter 4

Conclusion

A non-convex sparsity promoting penalty function, the transformed l_1 (TL1), is studied for optimization problems with its applications in compressed sensing (CS) and matrix completion. Exact recovery theory with RIP condition, as well as some local minima properties are proposed and proved. For compressed sensing problems, several TL1 algorithms are developed, including difference of convex functions and thresholding algorithms. They are tested with state-of-the-art methods, and show their advantages. TL1 is also expanded as a matrix quasi-norm, TS1, which is applied to solve matrix completion problems. A fixed point representation theory is proposed for the constrained matrix optimization. TS1 iterative thresholding algorithms are developed and compared with some state-of-the-art algorithms on matrix completion test problems. In the future, I will research and test acceleration methods to speed up DCATL1 algorithm.

BIBLIOGRAPHY

[1] T. Blumensath, *Accelerated iterative hard thresholding*, Signal Processing, 92(3), pp. 752–756, 2012.

[2] T. Blumensath, M. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629-654, 2008.

[3] J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[4] E. Candès, C. Fernandez-Granda, *Super-resolution from noisy data*, Journal of Fourier Analysis and Applications, 19(6):1229-1254, 2013.

[5] E. Candès, and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9 (2009), pp. 717-772.

[6] E. Candès, J. Romberg, T. Tao, *Robust uncertainty principles: Exact signal reconstruc-*

tion from highly incomplete Fourier information, IEEE Trans. Info. Theory, 52(2), 489-509, 2006.

[7] E. Candès, J. Romberg, T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure Applied Mathematics, 59(8):1207-1223, 2006.

[8] E. Candès, M. Rudelson, T. Tao, R. Vershynin, *Error correction via linear programming*, in 46th Annual IEEE Symposium on Foundations of Computer Science, pp. 668-681, 2005.

[9] E. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

[10] E. Candès, T. Tao, *Decoding by linear programming*, IEEE Trans. Info. Theory, 51(12):4203-4215, 2005.

[11] E. Candès, MB. Wakin and SP. Boyd, *Enhancing sparsity by reweighted ℓ_1 minimization*, Journal of Fourier analysis and applications 14.5-6 (2008): 877-905.

[12] W. Cao, J. Sun, and Z. Xu, Fast image deconvolution using closed-form thresholding formulas of regularization, *Journal of Visual Communication and Image Representation*, 24(1):31-41, 2013.

[13] R. Chartrand, *Nonconvex compressed sensing and error correction*, ICASSP 2007, vol. 3, p. III 889.

[14] R. Chartrand, W. Yin, *Iteratively reweighted algorithms for compressive sensing*, ICASSP 2008, pp. 3869-3872.

[15] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis. Low-rank matrix recovery from errors and erasures. *Information Theory, IEEE Transactions on*, 59(7):4324–4337, 2013.

[16] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413-1457, 2004.

[17] I. Daubechies, R. DeVore, M. Fornasier, C. Gunturk, *Iteratively reweighted least squares minimization for sparse recovery*, Comm. Pure Applied Math, 63(1), pp. 1–38, 2010.

[18] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[19] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[20] D. Donoho, *Denoising by soft-thresholding*, IEEE Trans. Info. Theory, 41(3), pp. 613–627, 1995.

[21] D. Donoho, *Compressed sensing*, IEEE Trans. Info. Theory, 52(4), 1289-1306, 2006.

[22] D. Donoho, M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization*, Proc. Nat. Acad. Scien. USA, vol. 100, pp. 2197-2202, Mar. 2003.

[23] E. Esser, Y. Lou and J. Xin, *A Method for Finding Structured Sparse Solutions to Nonnegative Least Squares Problems with Applications*, SIAM J. Imaging Sciences, 6(2013), pp. 2010-2046.

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

[25] K. Fan, *Maximum properties and inequalities for the eigenvalues of completely continuous operators,* Proc. Nat. Acad. Sci. U.S.A. 37 (1951), 760–766.

[26] A. Fannjiang, W. Liao, *Coherence Pattern-Guided Compressive Sensing with Unresolved Grids*, SIAM J. Imaging Sciences, Vol. 5, No. 1, pp. 179–202, 2012.

[27] M. Fazel, H. Hindi, and S. Boyd, *A rank minimization heuristic with application to minimum order system approximation*, In Proc. American Control Conference, Arlington, VA, 2001.

[28] M. Fazel, H. Hindi, and S. Boyd, *Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices*, in Proc. Amer. Control Confer., pp. 2156–2162, Denver, CO, 2003.

[29] M. Friedlander, I. Macedo, and T-K Pong. Gauge optimization and duality. *SIAM Journal on Optimization*, 24(4):1999–2022, 2014.

[30] M. Friedlander and I. Macedo. Low-Rank Spectral Optimization via Gauge Duality. SIAM Journal on Scientific Computing 38.3 (2016): A1616-A1638.

[31] T. Goldstein and S. Osher, *The Split Bregman Method for ℓ_1-regularized Problems*, SIAM Journal on Imaging Sciences, 2(1):323-343, 2009.

[32] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[33] D. Jannach, M. Zanker, A. Felfernig, G. Friedrich, "Recommender Systems: An Introduction", Cambridge Univ. Press, 2012.

[34] S. Ji, K-F Sze, Z. Zhou, A. So, Y. Ye, *Beyond Convex Relaxation: A Polynomial-Time Non-Convex Optimization Approach to Network Localization*, Proceedings of the 32nd IEEE International Conference on Computer Communications (INFOCOM 2013), 2013, pp. 2499-2507.

[35] R. Keshavan, A. Montanari, S. Oh, *Matrix completion from a few entries*, IEEE Trans. Info. Theory, 56 (6), 2980-2998, 2010.

[36] M-J Lai, Y. Xu, and W. Yin, *Improved Iteratively Reweighted Least Squares for Unconstrained Smoothed ℓ_q Minimization,* SIAM Journal on Numerical Analysis, 51(2):927-957, 2013.

[37] H.A. Le Thi, B.T.A. Thi, and H.M. Le, *Sparse signal recovery by difference of convex functions algorithms*, Intelligent Information and Database Systems, pp. 387-397. Springer, 2013.

[38] H.A. Le Thi, V. Ngai Huynh and T. Pham Dinh, *DC programming and DCA for general DC programs*, Advanced Computational Methods for Knowledge Engineering. Springer International Publishing, 2014. 15-35.

[39] H.A. Le Thi, T. Pham Dinh, H.M. Le, and X.T. Vo, *DC approximation approaches for sparse optimization*, European Journal of Operational Research, 244.1 (2015): 26-46.

[40] 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. Scientific Computing, 64, 178–196, 2015.

[41] Z. Lu and Y. Zhang, *Sparse approximation via penalty decomposition methods*, SIAM J. Optimization, 23(4):2448-2478, 2013.

[42] Z. Lu and Y. Zhang. Iterative reweighted singular value minimization methods for l_p regularized unconstrained matrix minimization. Technical report, Simon Fraser University, Burnaby, BC, Canada, 2014.

[43] J. Lv, and Y. Fan, *A unified approach to model selection and sparse recovery using regularized least squares,* Annals of Statistics, 37(6A), pp. 3498-3528, 2009.

[44] S. Ma, D. Goldfarb, and L. Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2):321–353, 2011.

[45] S. Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries*, IEEE Trans. Signal Processing, 41(12):3397-3415, 1993.

[46] R. Mazumder, J. Friedman, and T. Hastie, *SparseNet: Coordinate descent with nonconvex penalties*, Journal of the American Statistical Association, 106(495), pp. 1125-1138, 2011.

[47] K. Mohan and M. Fazel. Iterative reweighted algorithms for matrix rank minimization. *The Journal of Machine Learning Research*, 13(1):3441–3473, 2012.

[48] B. Natarajan, *Sparse approximate solutions to linear systems*, SIAM Journal on Computing, 24(2):227-234, 1995.

[49] D. Needell and R. Vershynin, *Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit,* IEEE Journal of Selected Topics in Signal Processing, 4(2):310-316, 2010.

[50] F. Nie, H. Huang, and C. Ding. Low-rank matrix recovery via efficient schatten p-norm minimization. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[51] M. Nikolova, *Local strong homogeneity of a regularized estimator*, SIAM Journal on Applied Mathematics 61.2 (2000): 633-658.

[52] C.S. Ong, H.A. Le Thi, *Learning sparse classifiers with difference of convex functions algorithms*, Optimization Methods and Software, 28(4):830-854, 2013.

[53] N. Parikh and SP. Boyd, *Proximal Algorithms*, Foundations and Trends in optimization 1.3 (2014): 127-239.

[54] T. Pham Dinh and H.A. Le Thi, *Convex analysis approach to d.c. programming: Theory, algorithms and applications*, Acta Mathematica Vietnamica, vol. 22, no. 1, pp. 289-355, 1997.

[55] T. Pham Dinh and H.A. Le Thi, *A DC optimization algorithm for solving the trust-region subproblem*, SIAM Journal on Optimization, 8(2), pp. 476–505, 1998.

[56] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[57] E. Soubies, L. Blanc-Féraud, and G. Aubert, *A Continuous Exact ℓ_0 Penalty (CEL0) for Least Squares Regularized Problem*, SIAM Journal on Imaging Sciences, 8.3 (2015): 1607-1639.

[58] T. Tao. *Topics in random matrix theory*, volume 132. American Mathematical Soc., 2012.

[59] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Royal. Statist. Soc, 58(1):267-288, 1996.

[60] J. Tropp and A. Gilbert, *Signal recovery from partial information via orthogonal matching pursuit*, IEEE Trans. Inform. Theory, 53(12):4655-4666, 2007

[61] B. Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.

[62] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4):333–361, 2012.

[63] F. Xu and S. Wang, *A hybrid simulated annealing thresholding algorithm for compressed sensing*, Signal Processing, 93:1577-1585, 2013.

[64] Z. Xu, X. Chang, F. Xu, and H. Zhang, $L_{1/2}$ regularization: A thresholding representation theory and a fast solver, *Neural Networks and Learning Systems, IEEE Transactions on*, 23(7):1013-1027, 2012.

[65] J. Yang and Y. Zhang, *Alternating direction algorithms for l_1 problems in compressive sensing*, SIAM Journal on Scientific Computing, 33(1):250-278, 2011.

[66] P. Yin, Y. Lou, Q. He, and J. Xin, *Minimization of L1 - L2 for compressed sensing*, SIAM Journal on Scientific Computing 37(1): A536 –A563, 2015.

[67] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for l_1-minimization with applications to compressed sensing*, SIAM Journal on Imaging Sciences, 1(1):143-168, 2008.

[68] W. Yin and S. Osher, *Error Forgetting of Bregman Iteration*, J. Sci. Computing, 54(2), pp. 684–695, 2013.

[69] J. Zeng, S. Lin, Y. Wang, and Z. Xu, $L_{1/2}$ regularization: Convergence of iterative half thresholding algorithm, Signal Processing, IEEE Transactions on, 62(9):2317-2329, 2014.

[70] C. Zhang, *Nearly unbiased variable selection under minimax concave penalty*, The Annals of statistics (2010): 894-942.

[71] S. Zhang and J. Xin, *Minimization of Transformed L_1 Penalty: Closed Form Representation and Iterative Thresholding Algorithms*, Communications in mathematical sciences, 15(2), pp. 511–537, 2017.

[72] 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.

[73] S. Zhang, P. Yin and J. Xin, *Transformed Schatten-1 iterative thresholding algorithms for matrix rank minimization and applications*, Communications in mathematical sciences, 15(3), pp. 839–862, 2017.

[74] T. Zhang, *Multi-stage convex relaxation for learning with sparse regularization*, Advances in Neural Information Processing Systems, pp. 1929-1936, 2009.