COMMUN. MATH. SCI.
Vol. 15, No. 3, pp. 839–862

TRANSFORMED SCHATTEN-1 ITERATIVE THRESHOLDING ALGORITHMS FOR LOW RANK MATRIX COMPLETION*

SHUAI ZHANG†, PENGHANG YIN‡, AND JACK XIN§

Abstract. We study a non-convex low-rank promoting penalty function, the transformed Schatten-1 (TS1), and its applications in matrix completion. 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. The TS1 threshold values differ in subcritical (supercritical) parameter regime where the TS1 threshold functions are continuous (discontinuous). We propose TS1 iterative thresholding algorithms and compare them with some state-of-the-art algorithms on matrix completion test problems. For problems with known rank, a fully adaptive TS1 iterative thresholding algorithm consistently performs the best under different conditions, where ground truth matrices are generated by multivariate Gaussian, (0,1) uniform and Chi-square distributions. For problems with unknown rank, TS1 algorithms with an additional rank estimation procedure approach the level of IRucL-q which is an iterative reweighted algorithm, non-convex in nature and best in performance.

Keywords. transformed Schatten-1 penalty; fixed point representation; closed form thresholding function; iterative thresholding algorithms; matrix completion

AMS subject classifications. 90C26; 90C46

1. Introduction

Low rank matrix completion problems arise in many applications such as collaborative filtering in recommender systems [4,17], minimum order system and low-dimensional Euclidean embedding in control theory [14,15], network localization [18], and others [26]. The mathematical problem is:

$$\min_{X\in\Re^{m\times n}} \operatorname{rank}(X) \quad \text{s.t.} \quad X\in Ł, \tag{1.1}$$

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

$$\min_{X\in\Re^{m\times n}} \operatorname{rank}(X) \quad \text{s.t.} \quad \mathscr{A}(X)=b \ in \ \Re^p, \tag{1.2}$$

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

$$\min_{X\in\Re^{m\times n}} \operatorname{rank}(X) \quad \text{s.t.} \quad X_{i,j}=M_{i,j}, \quad (i,j)\in\Omega \tag{1.3}$$

is a special case of (1.2), where X and M are both $m\times n$ matrices and Ω is a subset of index pairs $\{(i,j)\}$.

*Received: September 25, 2016; accepted (in revised form): October 28, 2016. Communicated by Wotao Yin.
The work was partially supported by NSF grant DMS-1222507 and DMS-1522383.
†Department of Mathematics, University of California, Irvine, CA, 92697, USA (szhang3@uci.edu). Phone: (949)-824-5309. Fax: (949)-824-7993.
‡Department of Mathematics, University of California, Irvine, CA, 92697, USA (penghany@uci.edu).
§Department of Mathematics, University of California, Irvine, CA, 92697, USA (jxin@math.uci.edu).

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:

$$\min_{X\in\Re^{m\times n}} F(X) \quad \text{s.t.} \quad \mathscr{A}(X)=b \tag{1.4}$$

and

$$\min_{X\in\Re^{m\times n}} \frac{1}{2}\|\mathscr{A}(X)-b\|_2^2+\lambda F(X). \tag{1.5}$$

The penalty function $F(\cdot)$ is defined on singular values of matrix X, typically $F(X)=\sum_i f(\sigma_i)$, where σ_i is the ith 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 [2–4,19–21,23,24,28]. Non-convex penalty based methods have shown better performance on hard problems [20, 24]. There is also a novel method to solve the constrained problem (1.4), from the perspective of gauge dual [32, 33].

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 [30, 31]. 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, 5, 7, 29]. 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 possesses the unbiasedness and Lipschitz continuity besides sparsity [12, 22].

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

The rest of the paper is organized as follows. In Section 2, 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, we propose two thresholding algorithms (TS1-s1 and TS1-s2) based on a fixed point equation of the global minimum. In Section 4, we compare TS1 algorithms with some state-of-the-art algorithms through numerical experiments in low rank matrix recovery and image inpainting. Concluding remarks are in Section 5.

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

$$\begin{aligned}&(x,y)=\sum_i x_i y_i \quad \text{for vectors } x,y;\\ &\langle X,Y\rangle=\mathrm{tr}(Y^TX)=\sum_{i,j} X_{i,j}Y_{i,j} \quad \text{for matrices } X,Y.\end{aligned}$$

Assume matrix $X\in\Re^{m\times n}$ has r positive singular values $\sigma_1\geq\sigma_2\geq\ldots\geq\sigma_r>0$. 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 $\|\mathrm{vec}(X)\|_2 = \|X\|_F$, 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:

$$\begin{cases} I_k^s(i,i) = 1, \ \textit{the first } k \textit{ diagonal elements}; \\ I_k^s(i,j) = 0, \ \textit{others}. \end{cases} \tag{1.6}$$

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

$$\mathrm{tr_k}(X) = \sum_{i=1}^{k} X_{i,i}. \tag{1.7}$$

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

$$\begin{aligned} &C_\lambda(X) = \tfrac{1}{2}\|\mathscr{A}(X) - b\|_2^2 + \lambda T(X); \\ &C_{\lambda,\mu}(X,Z) = \mu\left\{ C_\lambda(X) - \tfrac{1}{2}\|\mathscr{A}(X) - \mathscr{A}(Z)\|_2^2 \right\} + \tfrac{1}{2}\|X - Z\|_F^2 \\ &\qquad\qquad = \mu\lambda T(X) + \tfrac{\mu}{2}\|b\|_2^2 - \tfrac{\mu}{2}\|\mathscr{A}(Z)\|_2^2 - \mu(\mathscr{A}(X), b - \mathscr{A}(Z)) + \tfrac{1}{2}\|X - Z\|_F^2; \\ &B_\mu(Z) = Z + \mu\mathscr{A}^*(b - \mathscr{A}(Z)). \end{aligned} \tag{1.8}$$

2. TS1 minimization and thresholding representation

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

$$T(X) = \sum_{i=1}^{\mathrm{rank}(X)} \rho_a(\sigma_i), \tag{2.1}$$

where $\rho_a(\cdot)$ is a linear-to-linear rational function with parameter $a \in (0,\infty)$ [30,31],

$$\rho_a(|x|) = \frac{(a+1)|x|}{a+|x|}. \tag{2.2}$$

With the change of parameter a, TL1 interpolates l_0 and l_1 norms:

$$\lim_{a\to 0^+} \rho_a(x) = 1_{\{x\neq 0\}}, \quad \lim_{a\to+\infty} \rho_a(x) = |x|.$$

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

We shall focus on TS1 regularized problem

$$\min_{X\in\Re^{m\times n}} \frac{1}{2}\|\mathscr{A}(X) - b\|_2^2 + \lambda T(X), \tag{2.3}$$

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, $\mathscr{A}(X) = (\langle A_1, X\rangle, ..., \langle A_p, X\rangle)^T$.

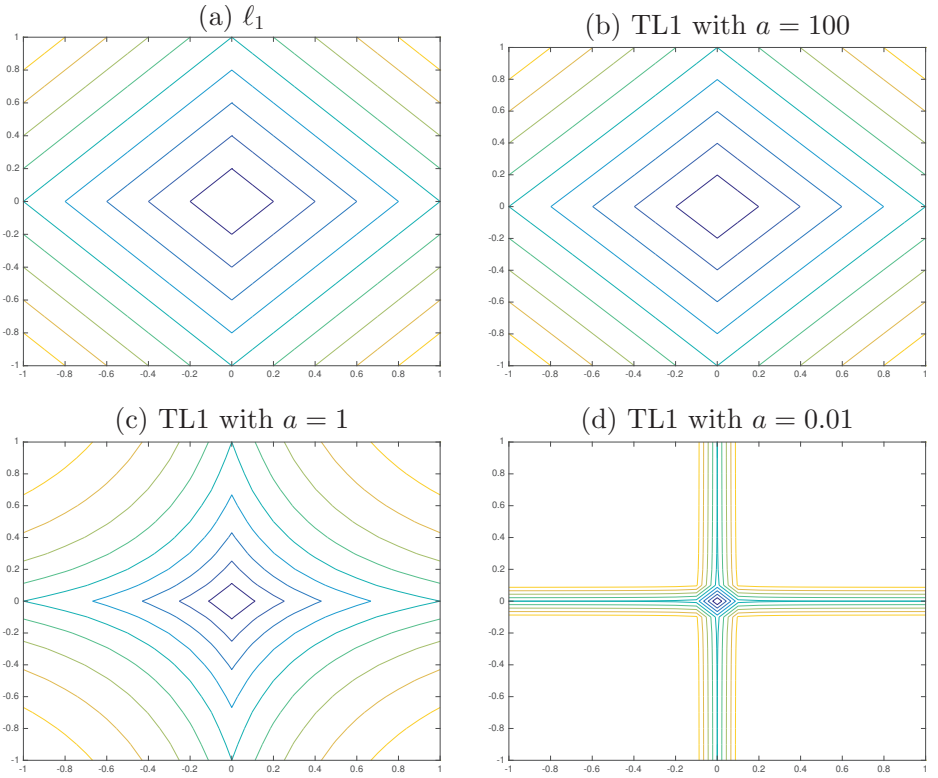


Fig. 2.1: *Level lines of TL1 with different parameters:* $a=100$ *(figure b),* $a=1$ *(figure c),* $a=0.01$ *(figure d). For large parameter a, the graph looks almost the same as* l_1 *(figure a). While for small value of a, it tends to the axis.*

2.1. Overview of TL1 minimization. To set the stage for the discussion of the TS1 regularized problem (2.3), we review the following results on one-dimensional TL1 optimization [30].

Let us consider the unconstrained TL1 regularized problem:

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

where matrix $A\in\Re^{m\times n}$, vector $y\in\Re^m$ are given, $P_a(x)=\sum_i \rho_a(|x_i|)$ and function $\rho_a(\cdot)$ is as in Equation (2.2).

In this subsection of TL1 minimization, we want to overwrite operator $B_\mu(\cdot)$ over vector x, instead of matrix field as before in (1.8),

$$B_\mu(x)=x+\mu A^T(y-Ax). \tag{2.5}$$

In the following Theorem (2.1), we prove that there exists a closed form expression for proximal operator $prox_{\lambda\rho_a}$ on univariate TL1 regularization problem, where $prox_{\lambda\rho_a}(x)=\arg\min_{y\in\Re} \frac{1}{2}(y-x)^2+\lambda\rho_a(y)$.

Proximal operator of a convex function usually intends to solve a small convex regularization problem, which often admits a closed-form formula or an efficient specialized

numerical method. 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 some local optimal or stationary points. In this subsection, we will show that for TL1 function, there indeed exists a closed-formed formula for its optimal solution.

Different with other thresholding operators, TL1 has 2 threshold value formulas depending on regularization parameter λ and TL1 parameter 'a'. We present them here with same notation as [30].

$$\begin{cases} t_2^* = \lambda\frac{a+1}{a} & \textit{(sub-critical parameter)} \\ t_3^* = \sqrt{2\lambda(a+1)} - \frac{a}{2} & \textit{(super-critical parameter).} \end{cases} \tag{2.6}$$

The inequality $t_3^* \le t_2^*$ holds and the equality is realized if and only if $\lambda = \frac{a^2}{2(a+1)}$, see [30].

Let $\mathrm{sgn}(\cdot)$ be the standard signum function with $\mathrm{sgn}(0)=0$, and

$$h_\lambda(x) = \mathrm{sgn}(x)\left\{\frac{2}{3}(a+|x|)\cos\left(\frac{\varphi(x)}{3}\right) - \frac{2a}{3} + \frac{|x|}{3}\right\} \tag{2.7}$$

with $\varphi(x) = \arccos(1 - \frac{27\lambda a(a+1)}{2(a+|x|)^3})$. In general, $|h_\lambda(x)| \le |x|$, see [30].

THEOREM 2.1 ([30]). *The optimal solution of* $y^* = \arg\min_{y\in\Re}\{\frac{1}{2}(y-x)^2 + \lambda\rho_a(|y|)\}$ *is a thresholding function of the form:*

$$y^* = \begin{cases} 0, & |x| \le t \\ h_\lambda(x), & |x| > t \end{cases} \tag{2.8}$$

where $h_\lambda(\cdot)$ *is defined in Equation (2.7), and the threshold parameter* t *depends on* λ *as follows:*

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

$$t = t_2^* = \lambda\frac{a+1}{a};$$

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

$$t = t_3^* = \sqrt{2\lambda(a+1)} - \frac{a}{2}.$$

According to the above theorem, we introduce thresholding operator $g_{\lambda,a}(\cdot)$ in $\Re$,

$$g_{\lambda,a}(w) = \begin{cases} 0, & \text{if } |w| \le t; \\ h_\lambda(w), & \text{if } |w| > t, \end{cases} \tag{2.9}$$

where t is the thresholding value in Theorem 2.1 and $h_\lambda(\cdot)$ in Equation (2.7).

In [30], the authors proved that when $\lambda < \frac{a^2}{2(a+1)}$, the TL1 thresholding function is continuous, same as the soft-thresholding function [8, 9]. While if $\lambda > \frac{a^2}{2(a+1)}$, the TL1 thresholding function has a jump discontinuity at threshold value, similar to the half-thresholding function [29]. For different thresholding schemes, it is believed that continuous formula is more stable, while discontinuous formula separates nonzero and trivial coefficients more efficiently and sometimes converges faster.

We have the following representation theorem for TL1 regularized problem (2.4).

THEOREM 2.2 ([30]). *If $x^*=(x_1^*,x_2^*,...,x_n^*)^T$ is a TL1 regularized solution (2.4) with a and λ being positive constants, and $0<\mu<\|A\|^{-2}$, then letting $t=t_2^*I_{\left\{\lambda\mu\le\frac{a^2}{2(a+1)}\right\}}+t_3^*I_{\left\{\lambda\mu>\frac{a^2}{2(a+1)}\right\}}$, the optimal solution satisfies the fixed point equation:*

$$x_i^*=g_{\lambda\mu,a}([B_\mu(x^*)]_i)\quad\forall\ i=1,...,n. \tag{2.10}$$

In the following, we will extend this result to TS1 low rank matrix completion and propose 2 thresholding algorithms based on it.

2.2. TS1 thresholding representation theory. Here we assume $m\le n$. 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

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

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

LEMMA 2.1 (Ky Fan k-norm inequality). *For a matrix $X\in\Re^{m\times n}$ with SVD: $X=UDV^T$, where diagonal elements of D are arranged in decreasing order, we have:*

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

that is, $\mathrm{tr_k}(X)\le\mathrm{tr_k}(D)=\|X\|_{Fk}$, $\forall k=1,2,...,m$. 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 1.1.

Another proof of this inequality without using dominance theorem is available. We leave it in the appendix for readers' convenience, making the paper self-contained.

THEOREM 2.3. *Consider matrix $Y\in\Re^{m\times n}$ that admits a singular value decomposition of the form: $Y=U\,\mathrm{Diag}(\sigma)V^T$, where $\sigma=(\sigma_1,...,\sigma_m)$. Then the unique global minimizer of $\min_{X\in\Re^{m\times n}}\frac{1}{2}\|X-Y\|_F^2+\lambda T(X)$ is:*

$$X^s=G_{\lambda,a}(Y)=U\mathrm{Diag}(g_{\lambda,a}(\sigma))V^T, \tag{2.11}$$

where $g_{\lambda,a}(\cdot)$ is defined in Equation (2.9) and applied entrywise to σ.

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

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

So

$$\begin{aligned}X^s&=\arg\min_{X\in\Re^{m\times n}}\frac{1}{2}\|X-Y\|_F^2+\lambda T(X)\\&=U\left\{\arg\min_{X\in\Re^{m\times n}}\frac{1}{2}\|X-Diag(\sigma)\|_F^2+\lambda T(X)\right\}V^T.\end{aligned} \tag{2.12}$$

Next we want to show:

$$\begin{aligned}&\underset{X\in\Re^{m\times n}}{\arg\min}\ \frac{1}{2}\|X-\mathrm{Diag}(\sigma)\|_F^2+\lambda T(X)\\ =&\underset{\{D\in\Re^{m\times n}\text{ is diagonal}\}}{\arg\min}\ \frac{1}{2}\|D-\mathrm{Diag}(\sigma)\|_F^2+\lambda T(D).\end{aligned}\tag{2.13}$$

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

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

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 1.1. 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 2.1 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 (2.13).

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 2.1, 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

$$\begin{aligned}&\underset{\{D\in\Re^{m\times n}\text{and D is diagonal}\}}{\arg\min}\ \frac{1}{2}\|D-\mathrm{Diag}(\sigma)\|_F^2+\lambda\ T(D)\\ =&\underset{X\in\Re^{m\times n}}{\arg\min}\ \frac{1}{2}\|X-\mathrm{Diag}(\sigma)\|_F^2+\lambda\ T(X)\\ =&\mathrm{Diag}(g_{\lambda,a}(\sigma)).\end{aligned}\tag{2.14}$$

In view of Equation (2.12), 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 $\underset{X\in\Re^{m\times n}}{\arg\min}\frac{1}{2}\|X-$

$\mathrm{Diag}(\sigma)\|_F^2+\lambda T(X)$. Based on the above proof and Theorem 2.1, 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 2.2. *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 $\min_{X\in\Re^{m\times n}} C_{\lambda,\mu}(X,Z)$, where the matrix function $C_{\lambda,\mu}(X,Z)$ is defined in Equation (1.8) of Section 1.1.*

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,

$$\begin{aligned}
C_{\lambda,\mu}(X,Z)=&\frac{1}{2}\|X\|_F^2-\langle X,Z\rangle+\frac{1}{2}\|Z\|_F^2+\lambda\mu T(X)+\frac{\mu}{2}\|b\|_2^2\\
&-\mu(\mathscr{A}(X),b-\mathscr{A}(Z))-\frac{\mu}{2}\|\mathscr{A}(Z)\|_2^2\\
=&\frac{1}{2}\|X\|_F^2+\frac{1}{2}\|Z\|_F^2+\frac{\mu}{2}\|b\|_2^2-\frac{\mu}{2}\|\mathscr{A}(Z)\|_2^2\\
&+\lambda\mu T(X)-\langle\ X,Z+\mu\mathscr{A}^*(b-\mathscr{A}(Z))\ \rangle\\
=&\frac{1}{2}\|X-B_\mu(Z)\|_F^2+\lambda\mu T(X)\\
&-\frac{1}{2}\|B_\mu(Z)\|_F^2+\frac{1}{2}\|Z\|_F^2+\frac{\mu}{2}\|b\|_2^2-\frac{\mu}{2}\|\mathscr{A}(Z)\|_2^2. \qquad (2.15)
\end{aligned}$$

Thus if we fix matrix Z,

$$\arg\min_{X\in\Re^{m\times n}} C_{\lambda,\mu}(X,Z)=\arg\min_{X\in\Re^{m\times n}} \tfrac{1}{2}\|X-B_\mu(Z)\|_F^2+\lambda\mu T(X). \qquad (2.16)$$

Then by Theorem 2.3, X^s is the unique global minimizer. ∎

THEOREM 2.4. *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 $\min_{X\in\Re^{m\times n}} C_{\lambda,\mu}(X,X^*)$.*

Proof.

$$\begin{aligned}
C_{\lambda,\mu}(X,X^*)=&\mu\{\frac{1}{2}\|\mathscr{A}(X)-b\|_2^2+\lambda T(X)\}\\
&+\frac{1}{2}\{\|X-X^*\|_F^2-\mu\|\mathscr{A}(X)-\mathscr{A}(X^*)\|_2^2\}\\
\geq&\mu\{\ \frac{1}{2}\|\mathscr{A}(X)-b\|_2^2+\lambda T(X)\ \}=\mu C_\lambda(X)\\
\geq&\mu C_\lambda(X^*)=C_{\lambda,\mu}(X^*,X^*). \qquad (2.17)
\end{aligned}$$

The first inequality is due to the fact:

$$\begin{aligned}
\|\mathscr{A}(X)-\mathscr{A}(X^*)\|_2^2=&\|A\mathrm{vec}(X)-A\mathrm{vec}(X^*)\|_2^2\\
\leq&\|A\|_2^2\ \|\mathrm{vec}(X-X^*)\|_2^2\\
\leq&\|\mathscr{A}\|_2^2\ \|X-X^*\|_F^2. \qquad (2.18)
\end{aligned}$$

From the above inequalities, we know that X^* is an optimal solution for $\min_{X\in\Re^{m\times n}} C_{\lambda,\mu}(X,X^*)$. The uniqueness of X^* follows from Lemma 2.2. ∎

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^*:

$$X^* = G_{\lambda\mu,a}(B_\mu(X^*)). \tag{2.19}$$

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

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

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

3. TS1 thresholding algorithms

Next we will utilize the fixed point Equation (2.19) to derive two thresholding algorithms for TS1 regularized problem (2.3). As in [30, 31], 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 kth step of iteration as:

$$\begin{aligned} X^k &= G_{\lambda\mu,a}(B_\mu(X^{k-1})) \\ &= U^{k-1}\,\mathrm{Diag}\left(g_{\lambda\mu,a}(\sigma^{k-1})\right)V^{k-1,T}, \end{aligned} \tag{3.1}$$

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 Equation (2.9), and

$$g_{\lambda\mu,a}(w) = \begin{cases} 0, & \text{if } |w| < t; \\ h_{\lambda\mu}(w), & \text{if } |w| \geq t. \end{cases} \tag{3.2}$$

Recall that the thresholding parameter t is:

$$t = \begin{cases} t_2^* = \lambda\mu\frac{a+1}{a}, & \text{if } \lambda \leq \frac{a^2}{2(a+1)\mu}; \\ t_3^* = \sqrt{2\lambda\mu(a+1)} - \frac{a}{2}, & \text{if } \lambda > \frac{a^2}{2(a+1)\mu}. \end{cases} \tag{3.3}$$

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.

3.1. 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 (2.3), 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 (2.19), the following inequalities hold:

$$\begin{aligned} &\sigma_i^* > t \Leftrightarrow i \in \{1,2,...,r\}, \\ &\sigma_j^* \leq t \Leftrightarrow j \in \{r+1, r+2, ..., m\}, \end{aligned} \tag{3.4}$$

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

$$\begin{aligned} &\sigma_r^* \geq t \geq t_3^* = \sqrt{2\lambda\mu(a+1)} - \tfrac{a}{2}; \\ &\sigma_{r+1}^* \leq t \leq t_2^* = \lambda\mu\tfrac{a+1}{a}. \end{aligned} \tag{3.5}$$

It follows that

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

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

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

(I) When $\lambda_1 \le \frac{a^2}{2(a+1)\mu}$, set $\lambda^* = \lambda_1$.
Then we will have $\lambda^* \le \frac{a^2}{2(a+1)\mu}$ 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 ith singular value σ_i^* is approximated by σ_i^n at the nth 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 nth step as

$$\lambda_n = \begin{cases} \lambda_{1,n}, & \text{if } \lambda_{1,n} \le \frac{a^2}{2(a+1)\mu}, \\ \lambda_{2,n}, & \text{if } \lambda_{1,n} > \frac{a^2}{2(a+1)\mu}. \end{cases} \tag{3.6}$$

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 1.

Algorithm 1 : TS1-s1 thresholding algorithm

Initialize: Given X^0 and parameter μ 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 λ_n by (3.6),
then obtain the thresholding value t_n by (3.3);
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. 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:

$$a = \lambda\mu + \sqrt{(\lambda\mu)^2 + 2\lambda\mu}. \tag{3.7}$$

The threshold value is:

$$t = \lambda\mu\frac{a+1}{a} = \frac{\lambda\mu}{2} + \frac{\sqrt{(\lambda\mu)^2 + 2\lambda\mu}}{2}. \tag{3.8}$$

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

$$\begin{aligned} &\sigma_i^* > t \Leftrightarrow i \in \{1,2,...,r\}, \\ &\sigma_j^* \leq t \Leftrightarrow j \in \{r+1,r+2,...,m\}. \end{aligned} \tag{3.9}$$

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 (3.7).

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:

$$\begin{aligned} &\lambda_n = \frac{2(\sigma_{r+1}^n)^2}{1+2\sigma_{r+1}^n}; \\ &a_n = \lambda_n\mu_n + \sqrt{(\lambda_n\mu_n)^2 + 2\lambda_n\mu_n}. \end{aligned} \tag{3.10}$$

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 2.

Algorithm 2 : TS1-s2 thresholding algorithm

Initialize: Given X^0 and parameter μ.
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 λ^n and a^n by (3.10), 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

4. 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-the-art solvers on matrix completion problems [1]. The comparison solvers include:

- LMaFit [28],
- FPCA [23],
- sIRLs-q [24],
- IRucLq-M [20],
- LRGeomCG [34]

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 [10, 11]. FPCA is a

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

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 [11] 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 *i*7-4770 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).

4.1. 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 [23] , 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

$$\text{rel.err} = \frac{\|X_{opt} - M\|_F}{\|M\|_F} \tag{4.1}$$

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

4.1.1. 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 [20, 28], 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 $X^T X$ 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 $(K+1)$st eigenvalues of $X^T X$, 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

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

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

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

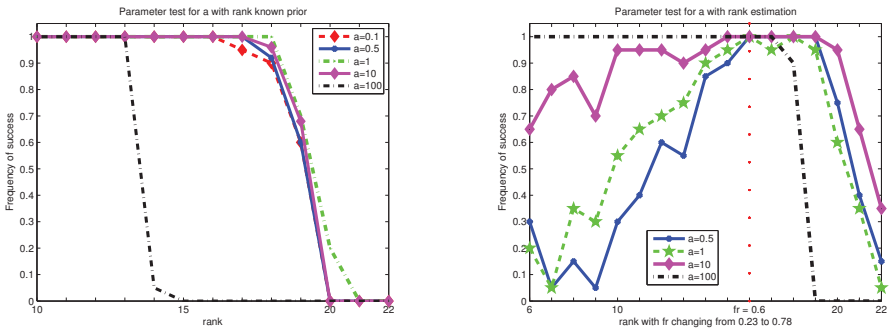
we adjust our rank estimator from K to $\widetilde{K}$. During numerical simulations, we did this adjustment only once for each problem. In most cases, this estimation adjustment is quite satisfactory and the adjusted estimate is very close to the true rank r.

4.1.2. 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 (4.1) was less than 5×10^{-3}. The test results for known rank scheme and rank estimation scheme are both shown in Figure 4.1. 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 $= 0.6$.

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 [30]. 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



Rank is known Rank is estimated

Fig. 4.1: *Optimal parameter test for semi-adaptive method: TS1-s1*

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

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

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

4.2. 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 4.2.1. 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 4.2.2. 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.

4.2.1. 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.

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

Table 4.1: *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* $tol = 10^{-6}$.

* 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.

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 4.1) and high dimensional matrices with $m = n = 1000$ (Table 4.2). Tests on non-square matrices with $m \neq n$ show similar results.

In Table 4.1, rank r varies from 5 to 18, while FR increases from 0.2437 up to 0.8190.

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

Table 4.2: *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	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

Table 4.3: *Numerical experiments on multivariate Gaussian matrices with varying covariance at known rank,* $m = n = 100$*,* $SR = 0.4$*.*

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 [34] with possible reason that we generate M with mean μ equal to 1, instead of 0 in [34]. 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 4.1, the two TS1 algorithms performed very well

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

Table 4.4: *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	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 4.5: *Comparison with random matrices generated from* (0,1) *uniform distribution. Rank* r *is given and* $m = n = 100$*,* $SR = 0.4$*, with stopping criterion* $tol = 10^{-6}$*.*

and remained stable for different FR values. At similar order of accuracy, the TL1s are faster than IRucL-q.

For large size matrices ($m = n = 1000$), rank r is varied from 50 to 110, see Table 4.2. 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 [20], it was carried out by high speed-performance CPU with many cores. Here we

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 4.6: *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		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

Table 4.7: *Numerical experiments for low rank matrix completion algorithms under rank estimation. True matrices are uncorrelated multivariate Gaussian,* $m=n=100$*,* $SR=0.4$*.*

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.

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 4.3, 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 4.4. 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.

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

Table 4.8: *Numerical experiments on low rank matrix completion algorithms under rank estimation. True matrices are multivariate Gaussian with different covariance,* $m = n = 100$*, and* $SR = 0.4$*.*

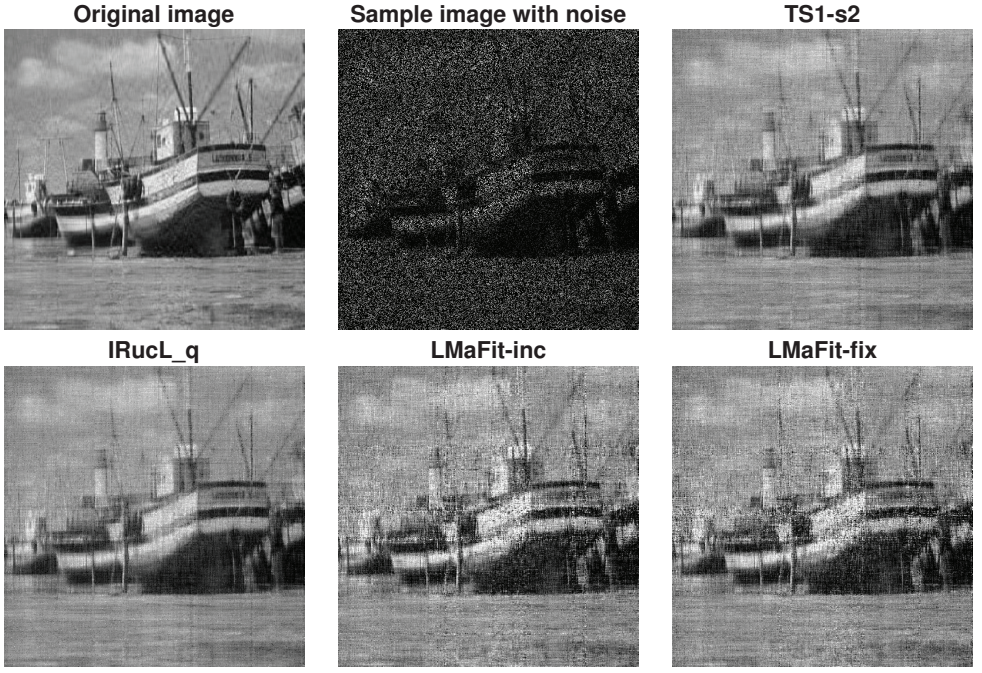


Fig. 4.2: *Image inpainting experiments with* $\mathrm{SR} = 0.3, \sigma = 0.15$.

In Table 4.4, 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.

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 4.1. The results are displayed at Table 4.5 (uniform distribution) and Table 4.6 (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 $r > 10$.

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

Table 4.9: *Numerical experiments on boat image inpainting with algorithms TS1, IRcuL-q and LMaFit under different sampling ratio and noise levels.*

4.2.2. 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 [23] 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 4.7 and Table 4.8. 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}.

4.3. Image inpainting. As in [20,28], 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 4.2) is used to produce ground truth as in [20] 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 [20] 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 4.9 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). 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.

5. Conclusion

We presented the transformed Schatten-1 penalty (TS1), and derived the closed form thresholding representation formula for global minimizers of TS1 regularized rank minimization problem. We studied two adaptive iterative TS1 schemes (TS1-s1 and TS1-s2) computationally for matrix completion in comparison with several state-of-the-art methods, in particular IRucL-q. In case of low rank matrix recovery under known rank, TS1-s2 performs the best in accuracy and computational speed. In low rank matrix recovery under rank estimation, TS1-s1 is almost on par with IRucL-q except when both the matrix covariance and rank rise to certain level. In future work, we shall study rank estimation techniques to further improve on TS1-s1 and explore other applications for TS1 penalty.

Acknowledgments. The authors would like to thank Prof. Wotao Yin for his helpful suggestions on low rank matrix completion methods and numerical experiments. We also thank the anonymous referees for their constructive comments.

Appendix A. Proof of Ky Fan k-norm inequality.

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

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

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

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

Notice that,

$$|w_i^{(k)}|\le\sum_{j=1}^{k}|U_{j,i}||V_{j,i}|\le\|U(:,i)\|_2\|V(:,i)\|_2\le 1,\tag{A.3}$$

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

$$\sum_{i=1}^{m}|w_i^{(k)}|\le\sum_{i=1}^{m}\sum_{j=1}^{k}|U_{j,i}||V_{j,i}|=\sum_{j=1}^{k}\sum_{i=1}^{m}|U_{j,i}||V_{j,i}|$$

$$\leq \sum_{j=1}^{k} \|U(j,:)\|_2 \, \|V(j,:)\|_2 \leq k, \tag{A.4}$$

where $U(j,:)$ and $V(j,:)$ are the jth 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 (A.1), 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 ith 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 (A.2), 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

$$U_{i,i} = V_{i,i} = \pm 1; \quad U_{i,j} = U_{j,i} = V_{i,j} = V_{j,i} = 0 \text{ for any index } j \neq i. \tag{A.5}$$

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

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

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 (A.1) and inequalities (A.3) and (A.4), it is true that

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

Furthermore, by definition (A.2), $w_k^{(k)} = \sum_{j=1}^{k} U_{j,k}V_{j,k} = U_{k,k}V_{k,k} = 1$. This is because $U_{j,k} = V_{j,k} = 0$ for index $j < k$, by the assumption (A.5) . Thus vectors U_k and V_k are also standard basis vectors with the kth 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. ∎

REFERENCES

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

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

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

[4] E. Candès and B. Recht, *Exact matrix completion via convex optimization*, Found. Comput. Math., 9:717–772, 2009.

[5] W. Cao, J. Sun, and Z. Xu, *Fast image deconvolution using closed-form thresholding formulas of $L_q, (q = 1/2, 2/3)$ regularization*, J. Vis. Commun. Image Represent., 24(1):31–41, 2013.

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

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

[8] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. math., 57(11):1413-1457, 2004.

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

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

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

[12] J. Fan and R. Li, *Variable selection via nonconcave penalized likelihood and its oracle properties*, J. Am. Stat. Assoc., 96(456):1348–1360, 2001.

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

[14] M. Fazel, H. Hindi, and S. Boyd, *A rank minimization heuristic with application to minimum order system approximation*, in Proceedings of the American Control Conference, Arlington, VA, 2001.

[15] M. Fazel, H. Hindi, and S. Boyd, *Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices*, in Proceedings of the American Control Conference, Denver, CO, 2003.

[16] 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.

[17] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*, Cambridge University Press, 2012.

[18] S. Ji, K.-F. Sze, Z. Zhou, A. So, and 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.

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

[20] M. Lai, Y. Xu, and W. Yin, *Improved iteratively reweighted least squares for unconstrained smoothed l_q minimization*, SIAM J. Numer. Anal., 51(2):927–957, 2013.

[21] Z. Lu and Y. Zhang, *Iterative reweighted singular value minimization methods for l_p regularized unconstrained matrix minimization*, preprint, arXiv:1401.0869, 2014.

[22] J. Lv and Y. Fan, *A unified approach to model selection and sparse recovery using regularized least squares*, Ann. Stat., 37(6A):3498–3528, 2009.

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

[24] K. Mohan and M. Fazel, *Iterative reweighted algorithms for matrix rank minimization*, J. Mach. Learn. Res., 13(1):3441–3473, 2012.

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

[26] 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.

[27] T. Tao, *Topics in Random Matrix Theory*, American Mathematical Society, 2012.

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

[29] Z. Xu, X. Chang, F. Xu, and H. Zhang, *$L_{1/2}$ regularization: an iterative thresholding method*, IEEE Trans. Neural Netw. Learn. Syst., 23:1013–1027, 2012.

[30] S. Zhang and J. Xin, *Minimization of transformed l_1 penalty: Closed form representation and iterative thresholding algorithms*, Commun. Math. Sci., 15(2):513–539, 2017.

[31] S. Zhang and J. Xin, *Minimization of transformed l_1 penalty: Theory, difference of convex function algorithm, and robust application in compressed sensing*, preprint, arXiv:1411.5735, 2014.

[32] M. Friedlander, I. Macedo, and T-K Pong, *Gauge optimization and duality*, SIAM J. Optim., 24(4):1999–2022, 2014.

[33] M. Friedlander and I. Macedo, *Low-rank spectral optimization*, preprint, arXiv:1508.00315, 2015.

[34] B. Vandereycken, *Low-rank matrix completion by riemannian optimization*, SIAM J. Optim., 23(2):1214–1236, 2013.