Accelerated Singular Value Thresholding for Matrix Completion

Yao Hu[1] Debing Zhang[1] Jun Liu[3] Jieping Ye[2] Xiaofei He[1]

[1]Zhejiang University, HangZhou, China
[2]Arizona State University, Tempe, AZ 85287
[3]Imaging and Computer Vision Department, Siemens Corporate Research, Princeton, NJ 08540
{huyao001, debingzhangchina, junliu.nt, jieping, xiaofeihe}@gmail.com

ABSTRACT

Recovering a large matrix from a small subset of its entries is a challenging problem arising in many real world applications, such as recommender system and image inpainting. These problems can be formulated as a general matrix completion problem. The Singular Value Thresholding (SVT) algorithm is a simple and efficient first-order matrix completion method to recover the missing values when the original data matrix is of low rank. SVT has been applied successfully in many applications. However, SVT is computationally expensive when the size of the data matrix is large, which significantly limits its applicability. In this paper, we propose an Accelerated Singular Value Thresholding (ASVT) algorithm which improves the convergence rate from $O(\frac{1}{N})$ for SVT to $O(\frac{1}{N^2})$, where N is the number of iterations during optimization. Specifically, the dual problem of the nuclear norm minimization problem is derived and an adaptive line search scheme is introduced to solve this dual problem. Consequently, the optimal solution of the primary problem can be readily obtained from that of the dual problem. We have conducted a series of experiments on a synthetic dataset, a distance matrix dataset and a large movie rating dataset. The experimental results have demonstrated the efficiency and effectiveness of the proposed algorithm.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining

General Terms

Algorithms

Keywords

Matrix Completion, Singular Value Thresholding, Nesterov's Method, Adaptive Line Search Scheme

KDD'12, August 12–16, 2012, Beijing, China.

1. INTRODUCTION

Estimating missing values from very limited information of an unknown matrix has received considerable attention recently. This problem occurs in both theoretical studies [4, 5, 7], and real world applications such as recommender system [13, 14, 22] and image/video analysis [17, 11]. Since the completion of arbitrary matrices is an ill-posed problem, it is usually assumed that the underlying matrix comes from a restricted class. One of the most natural assumption is that the matrix has a low-rank or approximately low-rank structure. Specifically, given the incomplete data matrix $M \in \mathbb{R}^{m\times n}$, the matrix completion problem can be formulated as follows:

$$\begin{array}{ll}\min\limits_{X} & rank(X) \\ s.t. & X_{ij} = M_{ij}, (i,j) \in \Omega,\end{array} \tag{1}$$

where Ω is the set of locations corresponding to the observed entries.

However, the rank minimization problem (1) is NP-hard in general due to the non-convexity and discontinuous nature of the rank function. The existing algorithms can not directly solve the rank minimization problem efficiently. It is known that the nuclear norm is the tightest convex lower bound of the rank function of matrices on the unit ball $\{X \in \mathbb{R}^{m\times n} | \|X\|_2 \leq 1\}$ [20], where the spectral norm, $\|\cdot\|_2$, of a matrix is equal to the largest singular value of the matrix. Therefore, a widely used approach is to apply the nuclear norm $\|\cdot\|_*$ (i.e., the summation of all the singular values) as a convex surrogate of the non-convex matrix rank function. Thus, the rank minimization problem can be approximated by the nuclear norm minimization problem as its convex relaxation:

$$\begin{array}{ll}\min\limits_{X} & \|X\|_* \\ s.t. & X_{ij} = M_{ij}, (i,j) \in \Omega.\end{array} \tag{2}$$

Recent theoretical breakthrough in matrix completion [4, 5] shows that, under some general constraints, the solution of nuclear norm minimization problems is unique and, with a high probability, is equal to the solution of rank minimization problems if the number of observed entries is large enough.

Many algorithms have been proposed to solve nuclear norm minimization problems (2). Fazel [8] firstly shows that problem (2) can be expressed as a Semi-Definite Programming (SDP) problem, which can be solved by conventional SDP solvers such as SDPT3 and SeDuMi [28, 23]. However, such solvers are usually based on interior point methods, and do

not scale to large matrices. This limits the usage of the matrix completion technique in real world applications. Recently, to solve the rank minimization problem for large scale matrices, Toh et al. apply an accelerated proximal gradient optimization technique (NNLS) [27] for solving nuclear regularized least squares problems. It has been shown theoretically that NNLS can terminate within $O(\frac{1}{\sqrt{\epsilon}})$ iterations with an ϵ-optimal solution. Keshavan et al. [12] consider the matrix completion problem by formulating it in a matrix factorization view. Their theoretical analysis shows that one could reconstruct a low-rank matrix by observing a set of entries of size at most a polylogarithmic factor larger than the intrinsic dimension of the variety of rank r matrices. The Singular Value Thresholding algorithm (SVT) [3] is a simple and efficient algorithm for nuclear norm minimization problems proposed by Cai et al., which has been shown to achieve superior performance in practice. However, as a special case of gradient method, SVT has a global convergence rate of $O(\frac{1}{N})$, where N is the number of iterations during optimization. This is too slow especially when dealing with large scale datasets.

In this paper, we propose a novel matrix completion algorithm called Accelerated Singular Value Thresholding (ASVT) for speeding up the standard SVT algorithm. Our basic idea is to obtain the solution of the nuclear norm minimization problem in SVT by solving its dual problem whose objective function can be shown to be continuously differentiable with Lipschitz continuous gradient. Specifically, we exploit the relationship between the optimal solution of the primal problem and that of its dual problem, based on which, the optimal solution of the primary problem can be readily obtained from the optimal solution of the dual problem. We show that the dual problem can be efficiently solved by using an adaptive line search algorithm with a convergence rate of $O(\frac{1}{N^2})$. Moreover, compared with the standard SVT algorithm, our approach can tune the step size adaptively for each iteration. This can further improve the efficiency of the proposed algorithm.

The rest of the paper is organized as follows. We provide a brief review of the standard SVT algorithm in Section 2. In Section 3, we detail our proposed approach and provide some theoretical analysis. We propose an adaptive line search algorithm to solve the optimization problem in Section 4. Experimental results on both synthetic and real world datasets are presented in Section 5. Finally, we provide some concluding remarks and suggestions for future work in Section 6.

Notions: Let $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ be an $m \times n$ matrix, $\Omega \subset \{1, ..., m\} \times \{1, ..., n\}$ denote the indices of the observed entries of X, and let Ω^c denote the indices of the missing entries. The Frobenius norm of X is defined as $\|X\|_F^2 = \sqrt{\Sigma_{(i,j)} X_{ij}^2}$. Let $\mathcal{P}_\Omega$ be the orthogonal projection operator onto the span of matrices vanishing outside of Ω so that the (i,j)-th component of $\mathcal{P}_\Omega(X)$ is equal to X_{ij} when $(i,j) \in \Omega$ and zero otherwise. Let $X = U\Sigma V^T$ be the singular value decomposition for X, where $\Sigma = diag(\sigma_i), 1 \leq i \leq \min\{m,n\}$ and σ_i is the i-th largest singular value of X. The "shrinkage" operator $D_\tau(X)$ is defined as [3]:

$$D_\tau(X) = U\Sigma_\tau V^T,$$

where $\Sigma_\tau = diag(\max\{\sigma_i - \tau, 0\})$, $1 \leq i \leq \min\{m,n\}$.

Let $S_{\mu,L}^{1,1}(\mathbb{R}^{m\times n})$ be the class of convex functions with Lipschitz gradient [19]. A continuous differentiable function $f(Y)$ belongs to $S_{\mu,L}^{1,1}(\mathbb{R}^{m\times n})$ for some $L \geq \mu \geq 0$ if for any $X, Y \in \mathbb{R}^{m\times n}$ we have both of the following:

$$\|f'(X) - f'(Y)\|_F \leq L\|X - Y\|_F, \tag{3}$$

$$\langle f'(X) - f'(Y), X - Y\rangle \geq \mu\|X - Y\|_F^2. \tag{4}$$

2. A BRIEF REVIEW OF SVT

The Singular Value Thresholding (SVT) algorithm solves the following problem:

$$\begin{array}{ll} \min\limits_X & \tau\|X\|_* + \frac{1}{2}\|X\|_F^2 \\ s.t. & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M). \end{array} \tag{5}$$

Cai et al. [3] give a theoretical analysis that when $\tau \to \infty$, the optimal solution of problem (5) converges to that of problem (2).

With a given $\tau > 0$ and starting with $Y_0 \in \mathbb{R}^{m\times n}$, the SVT algorithm operates as follows

$$\begin{cases} X^k = D_\tau(Y^{k-1}) \\ Y^k = Y^{k-1} + \delta_k \mathcal{P}_\Omega(M - X^k), \end{cases} \tag{6}$$

until a stopping criterion is reached, where $\{\delta_k\}$ is a positive step size sequence. It has been shown that, when the step sequence obeys $0 < \inf \delta_k \leq \sup \delta_k < 2$, the sequence $\{X^k\}$ obtained via (6) exactly converges to the unique solution of the problem in (5).

Actually the iteration scheme (6) is the linearized Bregman iteration, which is a special instance of Uzawa's algorithm [3]. Furthermore, by defining the Lagrangian function of problem (5) as

$$\mathcal{L}(X, Y) = \tau\|X\|_* + \frac{1}{2}\|X\|_F^2 + \langle Y, \mathcal{P}_\Omega(M - X)\rangle, \tag{7}$$

where Y is the Lagrangian dual variable, we can derive its dual function as

$$f(Y) = \inf_X \mathcal{L}(X, Y). \tag{8}$$

Cai et al. show that SVT indeed optimizes the dual function $f(Y)$ via the gradient ascent method.

The SVT algorithm is shown to be an efficient algorithm for matrix completion, especially for large low rank matrices. However, SVT has a global convergence rate of $O(\frac{1}{N})$, which is still too slow for real world applications such as recommender systems.

3. THE OBJECTIVE FUNCTION

In this section we first examine the properties of the dual function $f(Y)$. We then analyze the relationship between the optimal solution of the problem (5) and that of its dual problem. Based on these results, we show how to achieve the optimal solution of the problem (5) from its dual optimum directly.

As the nuclear norm $\|\cdot\|_*$ is not differentiable, it is difficult to optimize the dual function $f(Y)$ directly. However, by using the Moreau-Yosida regularization technique [10], we can obtain some interesting and useful properties of the dual function $f(Y)$.

In the SVT method, the shrinkage operator plays a critical role in the whole iteration scheme. We have the following result about the nuclear norm minimization problem (5).

THEOREM 1. *[3] For each $\tau \geq 0$ and $Y \in \mathbb{R}^{m\times n}$, we have*

$$\mathcal{D}_\tau(Y) = \arg\min_X \tau\|X\|_* + \frac{1}{2}\|X - Y\|_F^2. \quad (9)$$

Theorem 1 tells us that the shrinkage operator is the proximal point mapping associated with the nuclear norm. Based on the properties of Moreau-Yosida regularization (see the Theorem 4.1.4 of [10]), we obtain the following results:

LEMMA 2. *For any $X, Y \in \mathbb{R}^{m\times n}$, we have*

$$\langle \mathcal{D}_\tau(X) - \mathcal{D}_\tau(Y), X - Y\rangle \geq \|\mathcal{D}_\tau(X) - \mathcal{D}_\tau(Y)\|_F^2. \quad (10)$$

It follows that $\mathcal{D}_\tau(Y)$ is globally Lipschitz continuous with modulus 1.

The main result of this section is summarized in the following theorem:

THEOREM 3. *$\forall \tau \geq 0$, the dual function $f(Y)$ in (8) is continuously differentiable with Lipschitz continuous gradient at most 1. Furthermore, when the dual optimal Y^* of the problem (5) is obtained, the primal optimal X^* of the problem (5) is given by:*

$$X^* = \mathcal{D}_\tau(\mathcal{P}_\Omega(Y^*)). \quad (11)$$

PROOF.

$$\begin{aligned}
& f(Y) && (12)\\
= & \inf_X \mathcal{L}(X, Y) && (13)\\
= & \inf_X (\tau\|X\|_* + \frac{1}{2}\|X\|_F^2 + \langle Y, \mathcal{P}_\Omega(M - X)\rangle) && (14)\\
= & \inf_X (\tau\|X\|_* + \frac{1}{2}\|X - \mathcal{P}_\Omega(Y)\|_F^2 + \langle Y, \mathcal{P}_\Omega(M)\rangle && (15)\\
& -\frac{1}{2}\|\mathcal{P}_\Omega(Y)\|_F^2) && (16)\\
= & \inf_X (\tau\|X\|_* + \frac{1}{2}\|X - \mathcal{P}_\Omega(Y)\|_F^2) + \langle Y, \mathcal{P}_\Omega(M)\rangle && (17)\\
& -\frac{1}{2}\|\mathcal{P}_\Omega(Y)\|_F^2 && (18)\\
= & g(Y) + \langle Y, \mathcal{P}_\Omega(M)\rangle - \frac{1}{2}\|\mathcal{P}_\Omega(Y)\|_F^2, && (19)
\end{aligned}$$

where the first part of (19), i.e. $g(Y)$, is the Moreau-Yosida Regularization of the nuclear norm $\|\cdot\|_*$ [10]. Using the well known properties of Moreau-Yosida Regularization [10] and Theorem 1, we get the following results

- $g(Y)$ is a globally continuously differentiable convex function,
- $g'(Y) = \mathcal{P}_\Omega(Y - \mathcal{D}_\tau(\mathcal{P}_\Omega(Y)))$,
- $g'(Y)$ is continuously differentiable with Lipschitz continuous gradient 1. That is, for any $Y_1, Y_2 \in \mathbb{R}^{m\times n}$,

$$\|g'(Y_1) - g'(Y_2)\|_F \leq \|Y_1 - Y_2\|_F.$$

Then the gradient of $f(Y)$ can be obtained as follows:

$$\begin{aligned}
f'(Y) &= g'(Y) + \mathcal{P}_\Omega(M) - \mathcal{P}_\Omega(Y)\\
&= \mathcal{P}_\Omega(Y) - \mathcal{P}_\Omega(\mathcal{D}_\tau(\mathcal{P}_\Omega(Y))) + \mathcal{P}_\Omega(M) - \mathcal{P}_\Omega(Y)\\
&= \mathcal{P}_\Omega(M - \mathcal{D}_\tau(\mathcal{P}_\Omega(Y))).
\end{aligned} \quad (20)$$

It follows that for any $Y_1, Y_2 \in \mathbb{R}^{m\times n}$, we have

$$\begin{aligned}
&\|f'(Y_1) - f'(Y_2)\|_F\\
=&\|\mathcal{P}_\Omega(\mathcal{D}_\tau(\mathcal{P}_\Omega(Y_1))) - \mathcal{P}_\Omega(\mathcal{D}_\tau(\mathcal{P}_\Omega(Y_2)))\|_F\\
\leq&\|\mathcal{D}_\tau(\mathcal{P}_\Omega(Y_1)) - \mathcal{D}_\tau(\mathcal{P}_\Omega(Y_2))\|_F \quad (21)\\
\leq&\|\mathcal{P}_\Omega(Y_1) - \mathcal{P}_\Omega(Y_2)\|_F\\
\leq&\|Y_1 - Y_2\|_F,
\end{aligned}$$

where the second inequality follows from Lemma 2. Therefore, $f(Y)$ is continuously differentiable with Lipschitz continuous gradient at most 1.

When the dual optimal Y^* is obtained, by using the result of (19), we can get

$$\begin{aligned}
X^* &= \arg\min_X \mathcal{L}(X, Y^*)\\
&= \arg\min_X \tau\|X\|_* + \frac{1}{2}\|X - \mathcal{P}_\Omega(Y^*)\|_F^2 \quad (22)\\
&= \mathcal{D}_\tau(\mathcal{P}_\Omega(Y^*)),
\end{aligned}$$

where the third equality follows from Theorem 1. □

Since $f(Y)$ is the dual function of the objective function (5), $f(Y)$ is concave. Define

$$\begin{aligned}
h(Y) &= -f(Y)\\
&= -(\tau\|\mathcal{D}_\tau(\mathcal{P}_\Omega(Y))\|_* + \frac{1}{2}\|\mathcal{D}_\tau(\mathcal{P}_\Omega(Y))\|_F^2 \quad (23)\\
&\quad + \langle Y, \mathcal{P}_\Omega(M - \mathcal{D}_\tau(\mathcal{P}_\Omega(Y)))\rangle),
\end{aligned}$$

which is convex. Thus, the following holds for any $Y_1, Y_2 \in \mathbb{R}^{m\times n}$:

$$\langle h(Y_1) - h(Y_2), Y_1 - Y_2\rangle \geq 0. \quad (24)$$

From (20) and (21), it is easy to see $h(Y)$ belongs to the class $S_{0,1}^{1,1}(\mathbb{R}^{m\times n})$ and

$$h'(Y) = -\mathcal{P}_\Omega(M - \mathcal{D}_\tau(\mathcal{P}_\Omega(Y))). \quad (25)$$

Furthermore, in view of the relationship between the primal and dual optimal solutions of problem (5), we can solve problem (5) by firstly minimizing the objective function $h(Y)$, i.e.,

$$\min_{Y\in\mathbb{R}^{m\times n}} h(Y). \quad (26)$$

4. OPTIMIZATION METHOD

The key step in our proposed algorithm is to solve the dual problem (26). In this section, we develop an efficient optimization algorithm to solve this problem. Because the objective function $h(Y)$ is continuously differentiable with Lipschitz continuous gradient 1, in the following we propose to solve the smooth convex optimization problem (26) using the Nesterov's method.

It has been shown that Nesterov's method is a very powerful optimization technique for class $S_{\mu,L}^{1,1}(\mathbb{R}^{m\times n})$ [19]. However, how to choose the step size in each iteration is a critical issue in the Nesterov's method. To overcome this problem, we propose an Accelerated Singular Value Thresholding (ASVT) method with an adaptive line search scheme to solve problem (26).

The Nesterov's method attempts to find the optimal solution of (26) by utilizing two sequence $\{Y_k\}$ and $\{S_k\}$, where $\{Y_k\}$ is the sequence of approximate solutions, and $\{S_k\}$ is

the sequence of searching points. The searching point S_k is the affine combination of Y_k and Y_{k-1} as

$$S_k = Y_k + \beta_k(Y_k - Y_{k-1}), \tag{27}$$

where β_k is a tuning parameter. The approximate solution Y_{k+1} can be computed as a gradient step of S_k as

$$Y_{k+1} = S_k - \frac{1}{L_k}h'(S_k), \tag{28}$$

where $1/L_k$ is the step size. Starting from an initial point Y_0, we compute S_k and Y_{k+1} according to (27) and (28), and arrive at the optimal solution Y^*.

In the Nesterov's method, β_k and L_k are two key parameters. When they are set properly, the sequence $\{Y_k\}$ can converge to the optimal Y^* at a certain convergence rate. The Nesterov's constant scheme [19] and Nemirovski's line search scheme [18] usually need to set β_k and L_k. However, the Nesterov's constant scheme assumes L_k and β_k to be constant, while in Nemirovski's line search scheme, L_k is required to monotonically increase, $L_k \leq L_{k+1}$ and β_k is independent on L_k [18], resulting in slow convergence.

In the following, we assume that $\tilde{\mu}$ is the lower bound of μ, and $\tilde{\mu}$ is known in advance. This assumption is reasonable, since 0 is always a lower-bound for μ in (4). With this assumption, we adopt an adaptive line search scheme proposed by Liu et al. [16] for the Nesterov's method. This adaptive line search scheme is built upon the estimate sequence [19], which is defined as follows:

Definition 1. [19] A pair of sequences $\{\phi_k(Y)\}$ and $\{\lambda_k \geq 0\}$ is called an *estimate sequence* of function $h(Y)$ if the following two conditions hold:

1. $\lim_{k\to\infty} \lambda_k = 0$.
2. $\phi_k(Y) \leq (1-\lambda_k)h(Y) + \lambda_k\phi_0(Y), \forall Y \in \mathbb{R}^{m\times n}$.

The following theorem provides a systematic way for constructing the estimate sequence:

THEOREM 4. *[19] Let us assume that:*

1. *$h(Y)$ is smooth and strongly convex with Lipschitz gradient L. Moreover we know the value of $\tilde{\mu}$, which satisfies $\mu \geq \tilde{\mu} \geq 0$.*
2. *$\phi_0(Y)$ is an arbitrary function on $\mathbb{R}^{m\times n}$.*
3. *S_k is an arbitrary searching sequence on $\mathbb{R}^{m\times n}$.*
4. *α_k satisfies: $\alpha_k \in (0,1)$ and $\sum_{k=0}^{\infty}\alpha_k = \infty$.*
5. *$\lambda_0 = 1$.*

Then $\{\phi_k(Y), \lambda_k\}$ can be defined as follows:

$$\begin{aligned} \lambda_{k+1} &= (1-\alpha_k)\lambda_k, & (29)\\ \phi_{k+1}(Y) &= (1-\alpha_k)\phi_k(Y) + \alpha_k[h(S_k) + \\ &\quad \langle h'(S_k), Y - S_k\rangle + \frac{\tilde{\mu}}{2}\|Y - S_k\|_F^2]. & (30)\end{aligned}$$

If we choose a simple quadratic function for $\phi_0(Y)$ as

$$\phi_0(Y) = \phi_0^* + \frac{\gamma_0}{2}\|Y - V_0\|_F^2, \tag{31}$$

Algorithm 1 The Adaptive Line Search Scheme

1: **Input:** $\tilde{\mu}, \alpha_{-1} = 0.5, Y_{-1} = Y_0, L_{-1} = L_0, \gamma_0 \geq \tilde{\mu}, \lambda_0 = 1$.
2: **Output:** Y_N
3: **for** $k = 0, 1, 2, \cdots, N$ **do**
4: **while** 1 **do**
5: compute $\alpha_k \in (0,1)$ as the root of $L_k\alpha_k^2 = (1-\alpha_k)\gamma_k + \alpha_k\tilde{\mu}, \gamma_{k+1} = (1-\alpha_k)\gamma_k + \alpha_k\tilde{\mu}, \beta_k = \frac{\gamma_k(1-\alpha_{k-1})}{\alpha_{k-1}(\gamma_k + L_k\alpha_k)}$;
6: compute $S_k = Y_k + \beta_k(Y_k - Y_{k-1})$
7: compute $Y_{k+1} = S_k - \frac{1}{L_k}h'(S_k)$
8: **if** $h(Y_{k+1}) \leq h(S_k) - \frac{1}{2L_k}\|h'(S_k)\|_F^2$ **then**
9: goto Step 14
10: **else**
11: $L_k = 2L_k$
12: **end if**
13: **end while**
14: set $\omega = 2L_k\frac{h(S_k)-h(Y_{k+1})}{\|h'(S_k)\|_F^2}, L_{k+1} = h(\omega)L_k$
15: set $\lambda_{k+1} = (1-\alpha_k)\lambda_k$
16: **end for**

then we can specify the estimation sequence defined in Theorem 4 as [19]:

$$\phi_k(Y) = \phi_k^* + \frac{\gamma_k}{2}\|Y - V_k\|_F^2, \tag{32}$$

where the sequences γ_k, V_k and ϕ_k^* satisfy:

$$\begin{aligned} V_{k+1} &= \frac{1}{\gamma_{k+1}}[(1-\alpha_k)\gamma_k V_k + \tilde{\mu}\alpha_k S_k - \alpha_k h'(S_k)], & (33)\\ \gamma_{k+1} &= (1-\alpha_k)\gamma_k + \alpha_k\tilde{\mu}, & (34)\\ \phi_{k+1}^* &= \frac{\alpha_k(1-\alpha_k)\gamma_k}{\gamma_{k+1}}(\langle h'(S_k), V_k - S_k\rangle + \frac{\tilde{\mu}}{2}\|S_k - V_k\|_F^2) \\ &\quad +(1-\alpha_k)\phi_k^* + \alpha_k h(S_k) - \frac{\alpha_k^2}{2\gamma_{k+1}}\|h'(S_k)\|_F^2. & (35)\end{aligned}$$

The estimate sequence defined in Definition 1 has the following important property:

THEOREM 5. *[19] Let $\{\phi_k(Y)\}$ and $\{\lambda_k \geq 0\}$ be an estimate sequence. For any sequence $\{Y_k\}$, if*

$$h(Y_k) \leq \phi_k^* \equiv \min_{Y\in\mathbb{R}^{m\times n}} \phi_k(Y), \tag{36}$$

we have

$$h(Y_k) - h^* \leq \lambda_k[\phi_0(Y^*) - h^*] \to 0. \tag{37}$$

Based on Theorem 5, we propose an Accelerated Singular Value Thresholding (ASVT) algorithm based on Nesterov's method with an adaptive line search scheme. The complete procedure is summarized in Algorithm 1. In this scheme, the proper adaptive step size $1/L_k$ is designed to look for the approximate solution sequence $\{Y_k\}$ satisfying the condition (36). Then according to Theorem 5, the convergence rate of the solution sequence can be analyzed using the sequence $\{\lambda_k\}$.

In Algorithm 1, the while loop from Step 4 to Step 13 is designed to choose a proper step size to satisfy step 8. As the Lipschitz gradient of $h(Y)$ is 1, just like the Nemirovski's line search scheme, L_k is upper-bounded by 2, since the step 8 always holds when $L_k \geq 1$ [18]. Recall the fact that the iteration of L_k in Nemirovski's line search scheme is

Iteration	20	40	60	80
SVT	2.25e-04	7.36e-07	3.14e-09	1.52e-11
ASVT	**2.50e-06**	**5.94e-12**	**2.11e-14**	**2.40e-14**

Table 1: Relative error comparison between SVT and ASVT on solving the synthetic low rank matrix problem ($m = 1,000, n = 500$, $r = 15$, $p = 0.7$ and $\tau = 2\sqrt{mn}$). As can be seen, our proposed ASVT can accelerate the convergence by 2-5 orders of magnitude.

required to monotonically increase, in step 14 we adopt a more flexible iteration scheme of L_k [16] as

$$L_{k+1} = L_k \cdot h(\omega), h(\omega) = \begin{cases} 1, & 1 \le \omega \le 5 \\ 0.8, & \omega > 5 \end{cases} \quad (38)$$

where the parameter ω is computed as

$$\omega = 2L_k \frac{h(S_k) - h(Y_{k+1})}{\|h'(S_k)\|_F^2} \ge 1 \quad (39)$$

due to the condition in Step 8. When ω is too large, L_{k+1} is reduced to $0.8L_k$ to avoid the step size $\frac{1}{L_k}$ used in Step 7 becoming too small, which may slow down the convergence rate. Our experimental results show this particular choice of $h(\cdot)$ works well.

Although the step size $\frac{1}{L_k}$ does not monotonically decrease any more in our adaptive line search scheme, the proposed line search scheme preserves the convergence property, as summarized in the following theorem:

THEOREM 6. *[16] For Algorithm 1, we have*

$$\lambda_N \le \min\left\{\prod_{k=1}^{N}(1 - \sqrt{\frac{\tilde{\mu}}{L_k}}), \frac{1}{(1 + \sum_{k=1}^{N} \frac{1}{2}\sqrt{\frac{\gamma_0}{L_k}})^2}\right\}, \quad (40)$$

and

$$h(Y_N) - h^* \le \lambda_N \left[h(Y_0) - h^* + \frac{\gamma_0}{2}\|Y_0 - Y^*\|_F^2\right]. \quad (41)$$

Recall that $h(Y)$ is continuously differentiable with Lipschitz continuous gradient 1 and L_k is upper bounded by 2. When we set r_0 larger than 2, ASVT can get a global convergence rate $O(\frac{1}{N^2})$ from Theorem 6, which is significantly better than the original SVT algorithm.

5. EXPERIMENTS

In this section, we evaluate the performance of ASVT in comparison with SVT on both synthetic and real world datasets.

5.1 Experiments on Synthetic Data

We generate matrices $M \in \mathbb{R}^{m\times n}$ of rank r by sampling two matrices of $M_L \in \mathbb{R}^{m\times r}$ and $M_R \in \mathbb{R}^{r\times n}$, each having i.i.d. Gaussian entries, and setting $M = M_L M_R$. Suppose M_Ω is the observed part of M, and the set of observed indices Ω is sampled uniformly at random. Let p be the ratio between the observed entries and all $m \times n$ entries. Then different algorithms will be used to recover the missing entries from the partially observed information by solving the optimization problem in (5) with a given parameter τ. As suggested in [3], τ can be set to $t\sqrt{mn}$, where $2 \le t \le 5$.

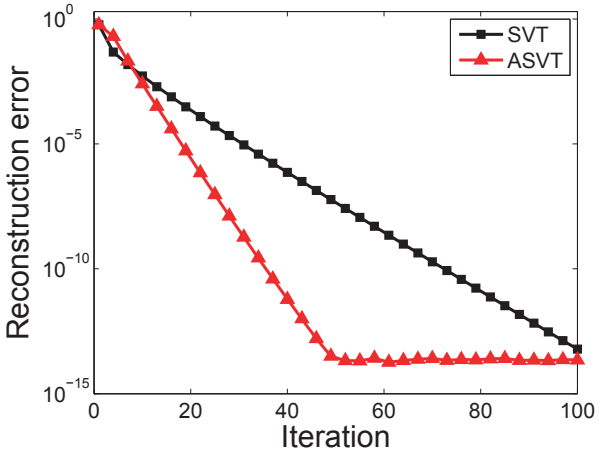

Figure 1: Convergence rate of SVT and ASVT on synthetic data ($m = 1,000, n = 500$, $r = 15$, $p = 0.7$ and $\tau = 2\sqrt{mn}$).

We evaluate the quality of the computed solution X of an algorithm by the relative reconstruction error defined by:

$$error = \|X - M\|_F / \|M\|_F, \quad (42)$$

which is a very commonly used criterion to evaluate the performance of a matrix completion algorithm [27]. For the parameter setting of SVT, we choose the constant step size as $\delta = 1.2/p$ which is recommended in [3]. For ASVT we set the initial L and $\tilde{\mu}$ as $L = p/1.2, \tilde{\mu} = 0.1$.

Firstly we conduct a simulation study with the following parameters: $m = 1,000, n = 500$, $r = 15$, $p = 0.7$ and $\tau = 2\sqrt{mn} \approx 1,414$. So 70% entries are observed. We will recover the other 30% entries by running SVT and ASVT separately. Fig. 1 illustrates the fast convergence rate of ASVT compared with SVT. Table 1 reports the relative reconstruction error of different methods after 20, 40, 60 and 80 iterations. We can observe that the convergence rate of ASVT is at least two orders of magnitude faster than that of SVT, which is consistent with our analysis.

To check how the performance of both algorithms changes with different settings, we test SVT and ASVT on the following different low rank matrix completion problems:

- Fix the matrix size (m, n), the rank r and the ratio of the observed entries p. Then test the performance with respect to different choices of the parameter τ. We fix $m = 1,000$, $n = 500$, $r = 15$, $p = 0.7$, and let τ change from $2\sqrt{mn}$ to $5\sqrt{mn}$.

- Fix the matrix size (m, n), the ratio of observed entries p and the parameter τ. Then test the performance with respect to different choices of the rank r. We fix $m = 1,000$, $n = 500$, $r = 15$, $\tau = 2\sqrt{mn}$, and let p change from 0.4 to 0.9.

- Fix the matrix size (m, n), the matrix rank r and the parameter τ. Then test the performance with respect to different choices of the ratio of observed entries p. We fix $m = 1,000$, $n = 500$, $p = 0.7$, $\tau = 2\sqrt{mn}$, and let r change from 5 to 50.

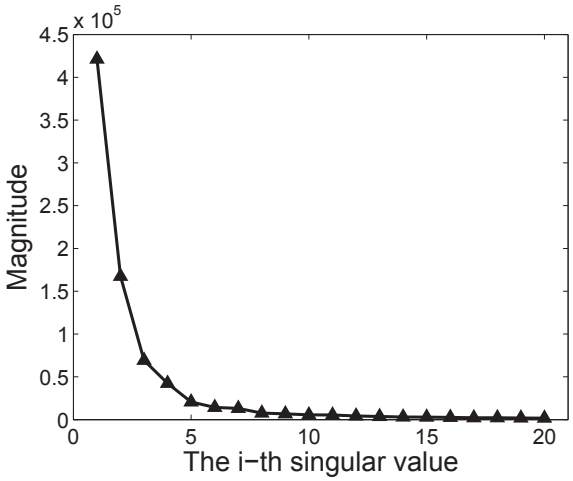

Figure 2: The largest 20 singular values of the distance matrix data.

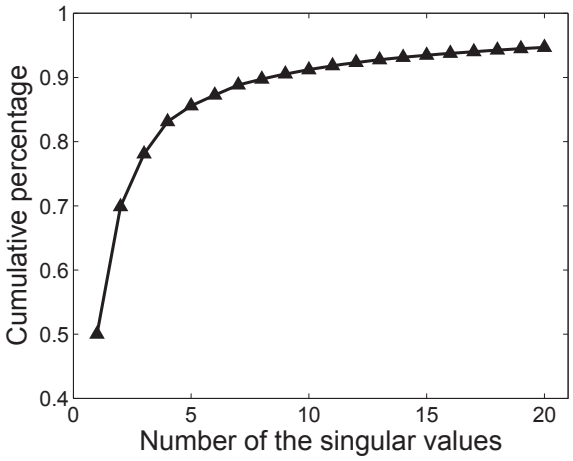

Figure 3: The normalized cumulative sums of the singular values of the distance matrix data.

- Fix the matrix rank r, the ratio of the observed entries p and the parameter τ. Then test the performance with respect to different choices of the matrix size (m, n). We fix $r = 15$, $p = 0.7$, $\tau = 2\sqrt{mn}$, and let (m, n) change from $(m = 400, n = 200)$ to $(m = 5,000, n = 2,500)$.

Table 2 reports the comparative results of randomly generated matrix completion problems. We can observe that ASVT converges much faster than SVT in all cases. ASVT's performance after 50 iterations surpasses SVT by several orders of magnitude. In average ASVT only needs 60% of the iterations required by SVT to achieve a given relative reconstruction error.

5.2 Experiments on Distance Matrix Data

In this experiment, we compare the performance of SVT and ASVT on a real world distance matrix dataset [2]. We consider the problem of recovering the real world distance

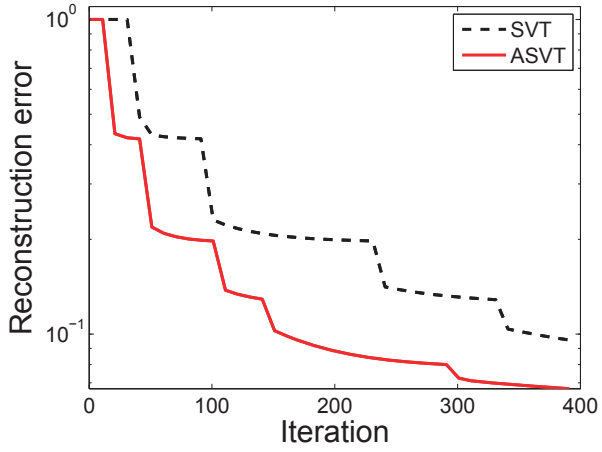

Figure 4: Convergence rates of SVT and ASVT on the real world distance matrix.

matrix M of 312 cities in the United States and Canada. The element M_{ij} of M measures the geodesic distance between the city i and the city j. Suppose some locations of M are unknown, we can recover the missing information by using the known part of the pairwise distance matrix. As mentioned in [3], the squared Euclidean distance matrix is a low rank matrix. With geodesic distances, the distance matrix M can also be well approximated by low rank matrices (the top several singular values of M are actually dominant as shown in Fig. 2 and Fig. 3). It is thus reasonable to recover the missing information using matrix completion algorithms.

Suppose the recovered matrix after k iterations is X^k. The rank of X^k has been shown empirically to be nondecreasing in the SVT algorithm [3]; we observe that ASVT has the same property. Since the complete data matrix is known, as suggested in [3], we also use the relative reconstruction error $\|X^k - M\|_F / \|M\|_F$ to measure the effectiveness of SVT and ASVT after the k-th iteration.

The reconstruction error of SVT and ASVT after each iteration is shown in Fig. 4. To achieve the same level of accuracy (measured by the reconstruction error), ASVT needs much fewer iterations than SVT. Fig. 5 shows the change of the rank with the iteration of SVT and ASVT. To reach a certain rank, ASVT needs much fewer iterations than SVT too. The iterations and computational times needed to reach the i-th rank are shown in Table 3. Moreover, we can see that the computational costs in one iteration of SVT and ASVT are similar (time/number of iterations). Note that, the last column of Table 3 ($\|M - M_r\|_F / \|M\|_F$) is the minimum relative error that we can achieve, where M_r is the best rank r approximation of M by computing the SVD of M.

5.3 Experiments on Recommendation Data

We now focus on the application of the proposed algorithm on the recommendation problem. We show the results of SVT and ASVT on the MovieLens data which is a widely used recommendation dataset [27, 29] and can be downloaded from [9]. The dataset is collected by the Grou-

	Settings					iteration (1e-8)		error after 50 iterations	
	m	n	r	p	τ	SVT	ASVT	SVT	ASVT
τ	1,000	500	15	0.7	$5\sqrt{mn}$	121	**61**	8.18e-05	**1.22e-07**
	1,000	500	15	0.7	$4\sqrt{mn}$	99	**47**	1.97e-05	**3.70e-09**
	1,000	500	15	0.7	$3\sqrt{mn}$	77	**30**	1.96e-06	**1.94e-13**
	1,000	500	15	0.7	$2\sqrt{mn}$	56	**30**	4.11e-08	**2.81e-14**
p	1,000	500	15	0.4	$2\sqrt{mn}$	94	**62**	1.04e-05	**1.40e-07**
	1,000	500	15	0.5	$2\sqrt{mn}$	77	**43**	1.91e-06	**8.85e-10**
	1,000	500	15	0.6	$2\sqrt{mn}$	64	**30**	2.36e-07	**2.41e-14**
	1,000	500	15	0.8	$2\sqrt{mn}$	49	**29**	7.23e-09	**3.56e-14**
	1,000	500	15	0.9	$2\sqrt{mn}$	44	**29**	9.29e-10	**3.67e-14**
r	1,000	500	5	0.7	$2\sqrt{mn}$	46	**27**	2.43e-09	**1.61e-14**
	1,000	500	10	0.7	$2\sqrt{mn}$	53	**28**	1.85e-08	**2.37e-14**
	1,000	500	20	0.7	$2\sqrt{mn}$	61	**30**	1.43e-07	**3.72e-14**
	1,000	500	50	0.7	$2\sqrt{mn}$	95	**44**	1.43e-05	**1.19e-09**
m, n	400	200	15	0.7	$2\sqrt{mn}$	91	**41**	9.64e-06	**3.42e-10**
	700	350	15	0.7	$2\sqrt{mn}$	66	**31**	3.64e-07	**6.05e-14**
	1,500	750	15	0.7	$2\sqrt{mn}$	49	**28**	6.96e-09	**2.04e-14**
	2,000	1,000	15	0.7	$2\sqrt{mn}$	46	**27**	2.11e-09	**2.26e-14**
	5,000	2,500	15	0.7	$2\sqrt{mn}$	38	**26**	7.28e-11	**2.68e-14**

Table 2: Comparisons between SVT and ASVT on the synthetic dataset with different settings (matrix size (m, n), rank r, observed ratio p and parameter τ). As can be seen, our proposed ASVT can accelerate the convergence by 2-7 orders of magnitude.

Rank(r)	iteration(k)		time(second)		$\|M - X^k\|_F/\|M\|_F$		$\|M - M_r\|_F/\|M\|_F$
	SVT	ASVT	SVT	ASVT	SVT	ASVT	
0	39	**17**	1.96	**0.87**	1.0000	1.0000	1.0000
1	96	**42**	5.06	**2.23**	0.4173	0.4173	0.4091
2	231	**101**	12.40	**5.46**	0.1977	0.1976	0.1895
3	334	**145**	18.51	**7.88**	0.1284	0.1284	0.1159
4	692	**300**	38.39	**16.89**	0.0797	0.0797	0.0706

Table 3: Comparisons between SVT and ASVT on recovering the distance matrix data. $\|M - X^k\|_F/\|M\|_F$ is the relative reconstruction error after the k-th iteration. Note that, the last column ($\|M - M_r\|_F/\|M\|_F$) is the minimum relative error that we can achieve, where M_r is the best rank r approximation of M by computing the SVD of M.

pLens Research Project at the University of Minnesota and contains 100,000 rating information from 943 users on 1,682 movies. The data has been cleaned up such that users who had less than 20 ratings were removed. So each user in the data has rated at least 20 movies. The ratings are from 1 (strongly unsatisfactory) to 5 (strongly satisfactory). In the recommendation situation, the data matrix is highly sparse (only about 6.3% entries are known). In order to test SVT and ASVT, we split the ratings into training and test sets. In our experiment, 80,000 ratings (80%) are randomly chosen to be the training set and the test set contains the remaining 20,000 ratings (20%).

Evaluation of recommendation algorithms has long been divided between accuracy metrics (e.g. precision/recall) and error metrics (notably, RMSE and MAE). The mathematical convenience and fitness with formal optimization methods have made error matrics like RMSE more popular [6]. Suppose all the existing rating locations of the test set are denoted as Ω. Like many recent research papers [29, 1], we also take the Root Mean Squared Error (RMSE) to measure the effectiveness of an algorithm on solving recommendation problems:

$$RMSE = \sqrt{\frac{\Sigma_{(i,j)\in\Omega}(X_{ij} - M_{ij})^2}{\#|\Omega|}}, \tag{43}$$

where X_{ij} is the recovered value and M_{ij} is the ground truth, and $\#|\Omega|$ is exactly the number of test ratings, i.e. 20,000. So the setting here is $m = 943, n =$1,682, $p =$ 80,000/$(mn) \approx 0.05$, and $\#|\Omega| =$20,000. We empirically choose $\tau = 10^4$. Both SVT and ASVT automatically recover the rating matrix by estimating the rank from a small to a large value.

As expected, the RMSE value decreases very fast with the iterations of both SVT and ASVT at first. And when the estimated rank is large enough, RMSE value becomes stable. The comparison result in terms of RMSE is shown in Fig. 6. It is clear that ASVT has a much faster convergence rate than SVT on the MovieLens dataset. Fig. 7 shows the change of the rank after each iteration of SVT and ASVT. Similar to Fig. 5, less time is needed for ASVT to reach a certain rank. More detailed numerical results are listed in Table 4. For the MovieLens dataset, it costs ASVT only about half of the time to get a similar RMSE value compared with SVT.

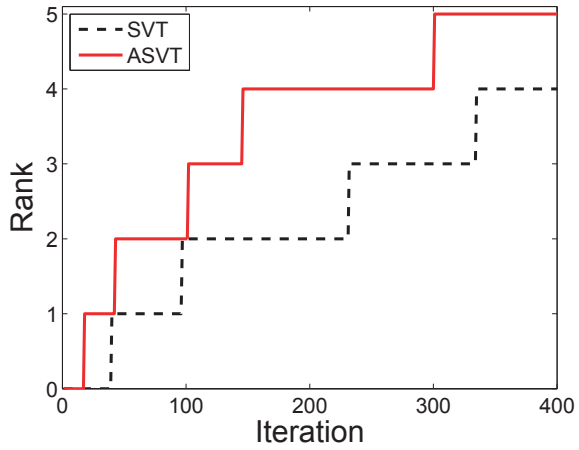

Figure 5: Rank vs. number of iterations of SVT and ASVT on the real world distance matrix.

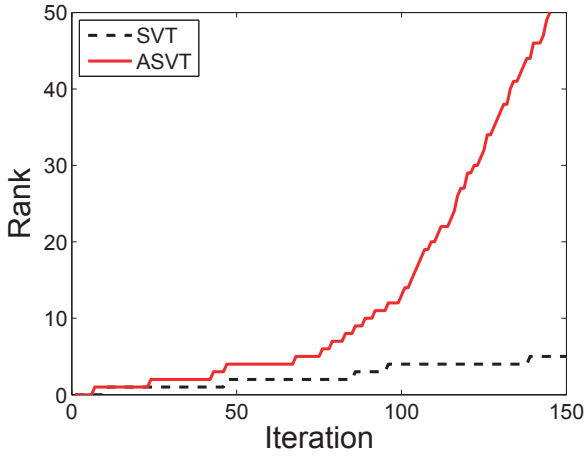

Figure 7: Rank vs. number of iterations of SVT and ASVT on MovieLens dataset.

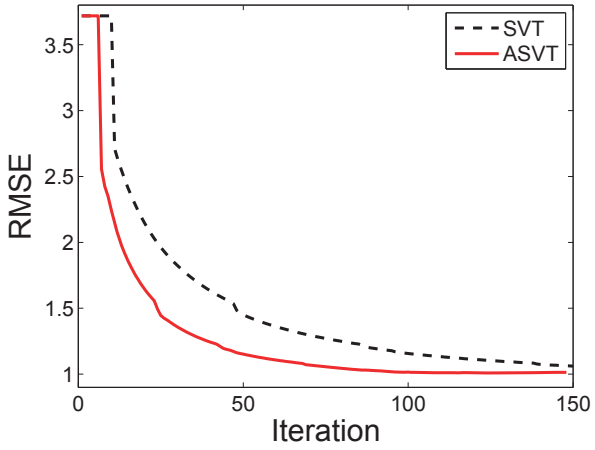

Figure 6: Convergence rates of SVT and ASVT on MovieLens dataset.

6. CONCLUSIONS

In this paper, we present an Accelerated Singular Value Thresholding method (ASVT) for estimating missing values for large scale matrix completion problems. The original SVT method solves the problem (5) with a global convergence rate $O(\frac{1}{N})$. We show how to speed up the original SVT algorithm using the Nesterov's method, which is an optimal first-order black-box method for the smooth convex optimization with a global convergence rate $O(\frac{1}{N^2})$. To further improve the efficiency, we adopt an adaptive line search scheme to tune the step size adaptively and in the meantime preserve the optimal convergence rate. We have conducted a series of experiments on synthetic and real world datasets. Experimental results show that ASVT is more efficient than the original SVT algorithm.

Recovering the missing values from limited information has been well studied for matrices. However, there is not much work on tensors, which are higher dimensional extensions of matrices. In many fields such as computer vision and biomedical signal progressing, it is more natural to represent the data as a tensor. Based on the recent development of tensor techniques [26, 25, 24, 17, 21, 15], it is promising to extend our work from matrix completion to tensor completion.

7. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No: 61125203).

8. REFERENCES

[1] D. Agarwal, B.-C. Chen, and B. Long. Localized factor models for multi-context recommendation. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego, California, USA, 2011.

[2] J. Burkardt. Cities – city distance datasets. `http://www.csit.fsu.edu/~burkardt/datasets/cities/cities.html`.

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

[4] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations on Computational Mathematics*, 9(6):717–772, 2009.

[5] E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transduction on Information Theory*, 56(5):2053–2080, 2009.

[6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, Barcelona, Spain, 2010.

[7] A. Eriksson and A. van den Hengel. Efficient computation of robust low-rank matrix approximations in the presence of missing data using the l1 norm. In *IEEE Conference on Computer Vision*

Rank	iteration		time(second)		RMSE	
	SVT	ASVT	SVT	ASVT	SVT	ASVT
1	46	**23**	24.37	**11.80**	1.5503	1.5574
2	85	**42**	47.40	**22.83**	1.2283	1.2273
3	95	**46**	53.54	**25.14**	1.1775	1.1818
4	138	**67**	80.45	**37.77**	1.0834	1.0841
5	152	**75**	89.28	**42.53**	1.0588	1.0564

Table 4: Comparisons between SVT and ASVT on the MovieLens dataset. We can see that ASVT only needs half of the time to get a similar RMSE value compared with SVT.

and Pattern Recognition, San Francisco, CA, USA, 2010.

[8] M. Fazel. Matrix rank minimization with applications. *PhD thesis, Stanford University*, 2002.

[9] GroupLens. Movielens. `http://www.grouplens.org/taxonomy/term/14`, 2009.

[10] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer Verlag, Heidelberg, 1996. Two volumes - 2nd printing.

[11] H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 2010.

[12] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[13] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, Las Vegas, Nevada, USA, 2008.

[14] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Paris, France, 2009.

[15] N. Li and B. Li. Tensor completion for on-board compression of hyperspectral images. In *Proceedings of the International Conference on Image Processing*, Hong Kong, China, 2010.

[16] J. Liu, J. Chen, and J. Ye. Large-scale sparse logistic regression. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, Paris, France, 2009.

[17] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. In *International Conference on Computer Vision*, pages 2114–2121, Kyoto, Japan, 2009.

[18] A. Nemirovski. *Efficient Methods in Convex Programming*. Lecture Notes, 1994.

[19] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2003.

[20] 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.

[21] B. R. Silvia Gandy and I. Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 2011.

[22] H. Steck. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, Washington, DC, USA, 2010.

[23] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones, 1998.

[24] D. Tao, X. Li, X. Wu, W. Hu, and S. J. Maybank. Supervised tensor learning. *Knowledge and Information Systems*, 13(1):1–42, 2007.

[25] D. Tao, X. Li, X. Wu, and S. J. Maybank. General tensor discriminant analysis and gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1700–1715, 2007.

[26] D. Tao, M. Song, X. Li, J. Shen, J. Sun, X. Wu, C. Faloutsos, and S. J. Maybank. Bayesian tensor approach for 3-d face modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(10):1397–1410, 2008.

[27] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6:615–640, 2010.

[28] R. H. Tutuncu, K. C. Toh, and M. J. Todd. Sdpt3 – a matlab software package for semidefinite quadratic linear programming, version 3.0, 2001.

[29] Y. Zhang and J. Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, Amsterdam, The Netherlands, 2007.