STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **4**, June 2016, pp 174–182.
Published online in International Academic Press (www.IAPress.org)

Two-Step Proximal Gradient Algorithm for Low-Rank Matrix Completion

Qiuyu Wang [1,*], Wenjiao Cao [2], Zheng-Fen Jin [3]

[1] *School of Software, Henan University, Kaifeng 475000, China*
[2] *School of Mathematics and Statistics, Henan University, Kaifeng 475000, China*
[3] *School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang 471023, China*

(Received: 15 February 2016; Accepted: 9 May 2016)

Abstract In this paper, we propose a two-step proximal gradient algorithm to solve nuclear norm regularized least squares for the purpose of recovering low-rank data matrix from sampling of its entries. Each iteration generated by the proposed algorithm is a combination of the latest three points, namely, the previous point, the current iterate, and its proximal gradient point. This algorithm preserves the computational simplicity of classical proximal gradient algorithm where a singular value decomposition in proximal operator is involved. Global convergence is followed directly in the literature. Numerical results are reported to show the efficiency of the algorithm.

Keywords Matrix nuclear norm minimization, Matrix completion, Proximal gradient algorithm, Singular value decomposition

AMS 2010 subject classifications 90C30, 65K05, 90C06, 90C25

DOI: 10.19139/soic.v4i2.201

1. Introduction

Over the past few years, finding low-rank matrix subject to some given constraints has attracted significant attentions due to its wide range of engineering applications. The general rank minimization problem can be expressed as

$$\begin{array}{c}\min_{X\in\mathbb{R}^{m\times n}}\ \mathrm{Rank}(X)\\ \text{s.t. } X\in\mathcal{C},\end{array} \tag{1}$$

where $X\in\mathbb{R}^{m\times n}$ is a decision variable, and $\mathcal{C}$ is a convex set. As a special case, the affine rank minimization problem can be expressed as follows:

$$\begin{array}{c}\min_{X\in\mathbb{R}^{m\times n}}\ \mathrm{Rank}(X)\\ \text{s.t. } \mathcal{A}(X)=b,\end{array} \tag{2}$$

where $\mathcal{A}:\mathbb{R}^{m\times n}\to\mathbb{R}^p$ is a closed linear operator and $\mathcal{A}^*$ is its adjoint, and $b\in\mathbb{R}^p$ is a known observation value. The rank minimization problem is generally known to be computationally intractable (NP-hard) [1, 2].

When the matrix variable is restricted to be diagonal, the problem (2) reduces to the linearly constrained nonsmooth cardinality minimization problem. One common approach is to replace the cardinality term by ℓ_1-norm and then to solve a convex minimization problem. Similarly, to solve (2) via a tractable problem, it is natural

*Correspondence to: Qiuyu Wang (Email: wqy@henu.edu.cn). School of Mathematics and Statistics, Henan University, Kaifeng, 475000, China.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

and common to replace "Rank(X)" by the so-called nuclear norm $\|\cdot\|_*$, that is

$$\begin{array}{l}\min_{X\in\mathbb{R}^{m\times n}} \|X\|_* \\ \text{s.t. } \mathcal{A}(X)=b,\end{array} \tag{3}$$

where $\|X\|_*$ is defined as the sum of its singular values. That is, assume that X has r positive singular values of $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, then $\|X\|_* = \sum_{i=1}^{r} \sigma_i$. A frequent alternative to (3) is the following nuclear norm regularized linear least squares problem

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|\mathcal{A}(X)-b\|_F^2 + \mu\|X\|_*, \tag{4}$$

where $\|\cdot\|_F$ is the Frobenius norm induced by the standard trace inner product, and parameter $\mu > 0$ is used to trade off both terms for minimization. The nuclear norm is alternatively known as the Schatten ℓ_1-norm and the Ky Fan norm [2], and it is the best convex approximation of the rank function over the unit ball of matrices [3].

A particular category of (3) is the so-called matric completion problem: given a random subset of entries of low-rank matrix, it would like to recover all the missing entries. Specially, according to Candès & Recht [4], it can be formulated as

$$\begin{array}{ll}\min_{X\in\mathbb{R}^{m\times n}} & \|X\|_* \\ \text{s.t.} & X_{i,j}=M_{i,j}, \quad (i,j)\in\Omega,\end{array} \tag{5}$$

where M is the unknown matrix with some available sampled entries and Ω is a set of index pairs (i,j). Throughout our discussion, the solution set of (5) is assumed to be nonempty. It is known that when the observations are corrupted by Gaussian noise, the following nuclear norm regularized linear least squares problem is preferable,

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|P_\Omega(X)-P_\Omega(M)\|_F^2 + \mu\|X\|_*, \tag{6}$$

where $P_\Omega(\cdot)$ is defined as

$$P_\Omega(X)_{i,j} = \begin{cases} X_{i,j} & \text{if } (i,j)\in\Omega, \\ 0 & \text{otherwise.}\end{cases} \tag{7}$$

Over the past few years, the numerical approaches for solving the formulations (3) and its different variants have been draw increasingly attention, e.g. [7, 8, 10]. Cai, Candès & Shen [3] proposed a singular value thresholding algorithm and showed that if the parameter goes infinity, then the sequence of optimal solutions converges to the one of (3) with minimum Frobenius norm. Ma, Goldfarb & Chen [11] introduced a fixed point continuation algorithm (FPCA), which is a matrix extension of the fixed point continuation algorithm by Hale et al. [12]. Liu, Sun & Toh [13] developed a proximal point algorithm from the primal, dual, and primal-dual forms, in which the inner sub-problems are solved by gradient projection method or accelerated proximal gradient method. Toh and Yun [5] extended Beck & Teboulle's algorithm [6] to solve matrix completion problems and designed an accelerated proximal gradient algorithm.

Due to its simplicity, the proximal gradient algorithm is popular to solve a closed proper convex optimization problem. However, the classical proximal gradient algorithm has been regarded as a slower until the exciting progress in recent years. The "accelerated" approaches mainly base on an extrapolation step which relies on not only the current point, but also two or more previous computed iterations (e.g., [14, 15]). Other dramatic algorithms can refer to Nesterov [16], Beck & Teboulle [6], Tseng [17], O'Donoghue & Candès [18], and references therein.

The main contribution of the paper is to further investigate the efficiency of the proximal gradient algorithms in solving the matric completion problem (6). The distinguished characters of the proposed algorithm is that each generated iteration is a combination of the previous point, the current point, and its proximal gradient point. As a result, the combination improves the algorithm's flexibility and practicability. The algorithm is also closely related to the well-known two-step iterative shrinkage/thresholding algorithm (TwIST) of the Bioucas-Dias & Figueiredo's algorithm [14] for sparse signal recovering. In other words, the proposed algorithm can be regarded as an extension or application of TwIST to solve nuclear norm regularized least squares for the purpose of testing its efficiency to find low rank matrices. The proposed algorithm preserves the computational simplicity of classical proximal

gradient algorithm and mainly involves a singular value decomposition in proximal operator. We give its global convergence property without proof under some appropriate conditions. Finally, we test the algorithm to show its numerical performance.

We organize the rest of this paper as follows. In Section 2, we summarize some preliminaries which are useful for further analysis, and quickly review some closed related approaches in the literature. In Section 3, we develop a two-step proximal gradient algorithm and present its convergence theorem. In Section 4, in order to investigate the benefit of the proposed algorithm, we test the algorithm by a series of numerical experiments. Finally, we conclude our paper in Section 5.

We summarize the notation used in this paper. Matrices are written as uppercase letters. Vectors are written as lowercase letters. For matrix $X \in \mathbb{R}^{m\times n}$, its Frobenius norm is defined as $\|X\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{i,j}^2}$, where $x_{i,j}$ is the (i,j)-th component of X. For any two matrices $X, Y \in \mathbb{R}^{m\times n}$, we define $\langle X, Y\rangle = \mathrm{tr}(X^\top Y)$ (the standard trace inner product), so that $\|X\|_F = \sqrt{\langle X, X\rangle}$. For any $x \in \mathbb{R}^n$, we denote by $\mathrm{Diag}(x)$ the diagonal matrix possessing the components of vector x on the diagonal. For any $X \in \mathbb{R}^{n\times n}$, we use $\lambda_{\max}(X)$ to denote its largest eigenvalue. We define "$\top$" as the transpose of a vector or a matrix. Additional notation will be introduced when it occurs.

2. Preliminaries

In this section, we review some preliminaries on proximal gradient algorithms which are useful in the later analysis. Let $f : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ be a closed proper convex function. The proximal operator $\mathrm{prox}_{\lambda f} : \mathbb{R}^n \to \mathbb{R}^n$ of the scaled function λf is defined by

$$\mathrm{Prox}_{\lambda f}(y) = \arg\min_{x\in\mathbb{R}^n} \Big(\lambda f(x) + \frac{1}{2}\|x - y\|_2^2\Big), \tag{8}$$

which is also called the proximal operator of f with parameter λ. The definition indicates that $\mathrm{Prox}_{\lambda f}(y)$ is a point that compromises between minimizing f and being near to y. Proximal gradient algorithm is to solve the convex separable minimization problem

$$\min_{x\in\mathbb{R}^n} \; F(x) := f(x) + g(x), \tag{9}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ are closed proper convex and f is differentiable. The proximal gradient algorithm is

$$x_{k+1} = \mathrm{Prox}_{\lambda_k g}\Big(x_k - \lambda_k \nabla f(x_k)\Big), \tag{10}$$

where $\lambda_k > 0$ is a step size. It is well-known that when ∇f is Lipschitz continuous with constant L, this method converges with rate $O(1/k)$ when a fixed step size $\lambda_k \in (0, 1/L]$ is used.

The accelerated version of the basic proximal gradient algorithm mainly includes an extrapolation. One popular approach is the well-known fast iterative shrinkage-thresholding algorithm (FISTA) of Beck & Teboulle [6]. Dramatically, the iteration complexity can achieve $O(1/k^2)$ with proper choice on steplength λ_k. Another accelerated approach is the so-called two-step iterative shringkage/thresholding algorithm (TwIST) of Bioucas-Dias & Figueiredo [14]. TwIST is motivated from [19] for solving a linear system in case that the coefficient matrix can be split into two terms and one of them is positive definite and easy to invert. Compared to the standard proximal gradient iteration (10), a general formulation with scalars β and α is formulated as

$$x_{k+1} = \big(1-\alpha\big)x_{k-1} + \big(\alpha - \beta\big)x_k + \beta\,\mathrm{Prox}_{\lambda_k g}\Big(x_k - \lambda_k \nabla f(x_k)\Big). \tag{11}$$

Clearly, the case of $\alpha = 1$ and $\beta = 1$ reduces to (10). It was shown in [14] that, the iteration (11) converges globally at the case of $f(x) = \|Ax - b\|^2$ and $g(x) = \|x\|_1$ by using some proper values of α and β.

3. Two-step proximal gradient algorithm

Based on the above preliminaries, we are now ready to construct the two-step proximal gradient algorithm to solve problem (6). Comparing the formulations of (6) and (9), we can take

$$f(X) = \frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 \quad \text{and} \quad g(X) = \mu\|X\|_*.$$

It's easy to know that $g : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a nonsmooth and continuous convex function and $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a continuously differentiable convex function with Lipschitz continuous gradient

$$\|\nabla f(X) - \nabla f(Y)\|_F \le L\|X - Y\|_F, \quad \forall\, X, Y \in \mathbb{R}^{m\times n},$$

where L is a Lipschitz constant.

For convenience, we denote $P_\Omega(X) = \mathcal{A}(X)$ and $P_\Omega(M) = b$. Using the notation, we consider the following quadratic approximation of $F(x)$ at a given point X_k

$$Q_k(X) := f(X_k) + \langle X - X_k, \nabla f(X_k)\rangle + \frac{L}{2}\|X - X_k\|^2 + g(X). \tag{12}$$

Its unique minimizer admits the following formulation:

$$\begin{aligned}
& \arg\min_{X\in\mathbb{R}^{m\times n}} Q_k(X) \\
= \;& \arg\min_{X\in\mathbb{R}^{m\times n}} f(X_k) + \langle X - X_k, \nabla f(X_k)\rangle + \tfrac{L}{2}\|X - X_k\|^2 + g(X) \\
= \;& \arg\min_{X\in\mathbb{R}^{m\times n}} g(X) + \tfrac{L}{2}\Big\|X - \Big(X_k - \tfrac{1}{L}\nabla f(X_k)\Big)\Big\|_F^2 \\
= \;& \mathrm{Prox}_{g/L}\Big(X_k - \tfrac{1}{L}\mathcal{A}^*\big[\mathcal{A}(X_k) - b\big]\Big).
\end{aligned}$$

By definition, evaluating a proximal operator involves solving a convex optimization problem generally. In some cases, exploiting special structure in the problem like sparsity may produce the simplest algorithm, even derive analytical solutions. Let $Y_k = X_k - \frac{1}{L}\mathcal{A}^*\big[\mathcal{A}(X_k) - b\big]$, and suppose that the rank of Y_k is r. The reduced singular value decomposition of Y_k can be expressed as

$$Y_k = U_k\Sigma_k V_k^\top, \quad \Sigma_k = \mathrm{Diag}(\{\sigma_i\}_{1\le i\le r}),$$

where U_k and V_k are respectively $m \times r$ and $n \times r$ matrix with orthogonal columns, $\sigma \in \mathbb{R}^r$ is the vector of positive singular values arranged in descending order $\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r > 0$. It follows [3] that the proximal operator can be expressed as

$$\begin{aligned}
& \mathrm{Prox}_{g/L}\Big(X_k - \tfrac{1}{L}\mathcal{A}^*\big[\mathcal{A}(X_k) - b\big]\Big) \\
= \;& \mathrm{Prox}_{g/L}\Big(Y_k\Big) \\
= \;& U_k\mathrm{Diag}\Big(\big[\sigma - \tfrac{\mu}{L}\big]_+\Big)V_k^\top,
\end{aligned}$$

where $[\cdot]_+ = \max\{\cdot, 0\}$ and "max" is interpreted as componentwise.

Based on the above analysis, the standard proximal gradient algorithm for solving (4) reduces to

$$X_{k+1} = \mathrm{Prox}_{g/L}(Y_k),$$

which means that the next iteration is only determined by the current X_k. For deriving an algorithm with more general form, as formula (11), we choose two positive scalars α and β, and set

$$X_{k+1} = \big(1 - \alpha\big)X_{k-1} + \big(\alpha - \beta\big)X_k + \beta\,\mathrm{Prox}_{g/L}(Y_k).$$

It is clear that the new iteration is a combination of the previous point X_{k+1}, the current point X_k, and its proximal gradient point $\mathrm{Prox}_{g/L}(Y_k)$.

For the general nuclear norm regularized linear least squares problem (4), the Lipschitz constant L which depends on the maximum eigenvalue of $\mathcal{A}^*\mathcal{A}$ is not always easily computable. Specially, in this case of the matrix completion problem (6), the linear operator $\mathcal{A}$ always satisfies $\mathcal{A}^*\mathcal{A} = I$ (i.e, the identity matrix), and then $L = 1$ from the fact that $L = \lambda_{\max}(\mathcal{A}^*\mathcal{A})$. Based on the above analysis, the standard version of the two-step proximal gradient algorithm (abbr. Ts_PGA) for solving separable convex minimization (6) can be outlined as follows.

Algorithm 1
(Ts_PGA)

Initialization. Choose positive constants α and β. Choose an initial point X_0 and set $X_{-1} = 0$. Let $k := 0$.
Step 1. Compute

$$\widetilde{X}_k = \text{Prox}_g\Big(X_k - \mathcal{A}^*\big[\mathcal{A}(X_k) - b\big]\Big)$$

where

$$X_{k+1} = \big(1-\alpha\big)X_{k-1} + \big(\alpha - \beta\big)X_k + \beta\widetilde{X}_k. \tag{13}$$

Step 2. If termination criterion is not met, let $k := k + 1$. Go to Step 1.

For being easily understood, we make a remark on the close relationship between Ts_PGA and the solver TwIST of Bioucas-Dias & Figueiredo [14], and the solver FPCA of Ma, Goldfarb, and Chen [11].

Remark 1
It is well-known that TwIST is designed for solving ℓ_1-norm regularized minimization problems in compressive sensing. It is also easy to derive that Ts_PGA is equivalent to TwIST when it used to recover large sparse signal in compressive sensing. Generally speaking, the proposed algorithm Ts_PGA can be considered as an extension of TwIST to solve the problems of recovering corrupted low rank matrix. At the special case of $\alpha = \beta = 1$, the iteration (13) reduces to the classical proximal gradient algorithm, also named the fixed point continuation algorithm when it used to solve (4). In other words, Ts_PGA can also be regarded as a generalized variant of FPAC.

Given that Ts_PGA is an application of the well-known TwIST to solve nuclear norm least squares for matric completion, hence, its global convergence can be followed directly. For completeness of this paper, we list the convergence theorem without proof at the end of this section. For the proof of the theorem, one can refer to [14, Appendix II].

Theorem 1
Let ξ be a real number such that $0 < \xi \leq \lambda_i(\mathcal{A}^*\mathcal{A}) \leq 1$ where $\lambda_i(\cdot)$ is the i-th eigenvalue and let $\hat{\rho} = \frac{1-\sqrt{\xi}}{1+\sqrt{\xi}}$. Let $\bar{X}$ be the minimizer of (6) and define

$$E_k = X_k - \bar{X}_k \quad \& \quad W_k = [E_{k+1},\ E_k]^\top.$$

(1) There exists a matrix Q_k such that $W_{k+1} = Q_k W_k$. If $0 < \alpha < 2$ and $0 < \beta < 2\alpha$, then $\rho(Q_k) < 1$, where $\rho(Q_k)$ is the spectral radius of $Q(k)$.
(2) Setting $\alpha = \hat{\rho}^2 + 1$ and $\beta = \frac{2\alpha}{1+\xi}$ guarantees that $\rho(Q_k) < 1$.
(3) If $0 < \alpha \leq 1$, and $0 < \beta < 2\alpha$, then $\lim_{k\to\infty} W_k = 0$

4. Numerical Experiments

In this section, to illustrate the efficiency of Ts_PGA, we test it to recover some missing entries of a low-rank matrix at the case of different rank, sample radios, and noisy levels. At the meantime, we compare it with the closed related algorithm FPCA [11] for performance comparison. In the first place, we summary some useful parameters to make

our experiments more easy to follow:

m: the row number of matrix;
n: the column number of matrix;
r: the rank of original matrix, which is far less than $\min\{m, n\}$;
sr: the sample ratio;
p: the number of measurements, which is set to be $p = sr \times m \times n$;
dr: the number of degree of freedom for a rank r matrix, which is defined as $dr = r(m + n - r)$;
Fr: $Fr = r(m + nr)/p$ and $Fr \in (0, 1)$ is important to successfully recover M;
M: a real low rank matrix to be recovered, which is generated by $M = M_L M_R^\top$, where matrix $M_L \in \mathbb{R}^{m\times r}$ and $M_R \in \mathbb{R}^{n\times r}$ are generated via the Matlab script "$randn(m, r)$";
Ω: the index set of known elements, which are selected randomly;
b: the given measurement vector, which is $b = \mathcal{A}(X) + \omega$;
ω: Gaussian noise with mean zero and standard deviation σ generated by $\sigma \times randn(p, 1)$;
μ: a penalty parameter, which is updated by continuation technique.
X^*: an optimal solution.

All experiments are performed under Window 7 premium and MATLAB v7.8(2009a) running on a Lenovo laptop with an Intel core CPU at 2.4 GHz and 2 GB memory. The iterative process is terminated when the optimal solution X^* satisfies the following criterion:

$$RelErr = \frac{\|X^* - M\|_F}{\|M\|_F} \leq tol. \tag{14}$$

The relative error (RelErr) measures the quality of X^* to the original M. As usual [3, 20], we say that M is recovered successfully if $RelErr$ is less than $tol = 10^{-3}$. By the way, the terminated condition (14) is also used in [11]. On the other hand, for comparing in a relatively fair way, we also use the Matlab package as in FPCA for matrix singular value decomposition (SVD).

Our tests are partitioned into four parts. In the first three parts, we test Ts_PGA and FPCA to solve different cases of matrix completion problems. In the forth part, we use two typical images to visibly show the efficiency of both tested algorithms. In each test, we use the same technique as in [14, (26) and (27)] to choose the parameters α and β for the purpose of guarantee the algorithm's convergence. The numerical results generated by Ts_PGA and FPCA for solving matrix completion problems with different sr and r are reported in Table 1, where "Time" denotes the CPU time required by the algorithm. As can be seen from the top part of the Table 1, both algorithms required more computing time as Fr is greater than 0.38. For each cases with different Fr, we can clearly see that Ts_PGA performs better than FPCA in terms of running time and relative error. The worse thing is that FPCA failed to obtain high accuracy solutions at the cases of $sr = 0.8, 0.9$. From the bottom part of this table, we observe that all the problems are successfully solved with different r when $Fr = 0.38$, and see that Ts_PGA is a winner in this case. From the simple test, it is concluded that Ts_PGA is more efficient than FPCA.

In the second test, we apply Ts_PGA and FPCA to solve some easy matrix completion problems with different dimension. The numerical results are listed in Table 2 where Fr restricts into $(0.01, 0.2)$ and the m and n vary from 100 to 2000. From this table, we see that both algorithms produce satisfied optimal solutions within almost equivalent running time. But the accuracy of the solutions obtained by FPCA looks slightly higher, which means that the performance of both algorithms are competitive.

In the third test, we use Ts_PGA and FPCA to recover low rank matrix which corrupted by Gaussian noise. The noisy level is set as $\sigma = 1e - 3$. The numerical results are reported in Table 3. From this table, it is clear to see that Ts_PGA requires less running time to get solutions with similar accuracy than FPCA. This test shows that Ts_PGA runs faster to solve noisy matrix completion problem.

To visibly illustrate the efficiency of Ts_PGA, in the last test, we use both algorithms to recover a couple of randomly corrupted images which widely used in the literature. In this test, we choose $sr = 0.5$ and $r = 40$. The original low-rank images, the corrupted images, and recovered images by each algorithm are presented in Figure

Table 1. Numerical results of Ts_PGA and FPCA

				Ts_PGA		FPCA	
(m, n)	r	sr	Fr	Time	RelErr	Time	RelErr
(100,100)	10	0.4	0.47	3.68	5.10e-006	4.35	2.66e-006
(100,100)	10	0.5	0.38	4.35	2.66e-006	9.06	1.37e-006
(100,100)	10	0.6	0.32	0.18	8.57e-005	0.26	4.70e-005
(100,100)	10	0.7	0.27	0.20	7.32e-005	0.85	4.91e-005
(100,100)	10	0.8	0.24	0.20	6.10e-005	1.35	3.36e-002
(100,100)	10	0.9	0.21	0.25	5.67e-005	1.32	2.27e-001
(100,100)	10	0.5	0.38	4.35	2.66e-006	9.06	1.37e-006
(200,200)	20	0.5	0.38	20.40	2.37e-006	44.27	1.21e-006
(300,300)	30	0.5	0.38	60.00	2.43e-006	132.96	1.02e-006
(400,400)	40	0.5	0.38	131.56	2.21e-006	284.10	8.81e-007
(500,500)	50	0.5	0.38	239.57	2.24e-006	510.55	9.21e-007

Table 2. Comparison of Ts_PGA and FPCA for easy matrix completion problems

				Ts_PGA		FPCA	
(m, n)	r	sr	Fr	Time	RelErr	Time	RelErr
(100,100)	5	0.5	0.20	0.20	1.00e-004	0.21	2.69e-005
(200,200)	5	0.5	0.10	0.70	8.63e-005	0.83	6.95e-006
(300,300)	5	0.5	0.07	1.94	8.96e-005	2.19	3.64e-006
(400,400)	5	0.5	0.05	4.17	8.60e-005	5.06	1.83e-006
(500,500)	5	0.5	0.04	6.30	9.75e-005	6.94	1.39e-006
(600,600)	5	0.5	0.03	12.15	8.17e-005	11.77	9.12e-007
(700,700)	5	0.5	0.03	18.09	9.35e-005	19.68	7.75e-007
(800,800)	5	0.5	0.02	27.31	9.69e-005	29.83	3.21e-007
(900,900)	5	0.5	0.02	37.48	9.74e-005	41.24	2.88e-007
(1000,1000)	5	0.5	0.02	39.64	9.30e-005	44.53	2.52e-007
(1500,1500)	5	0.5	0.01	232.00	9.42e-005	262.56	1.62e-007
(2000,2000)	5	0.5	0.01	554.60	9.33e-005	554.60	9.33e-005

Table 3. Numerical Results of Ts_PGA and FPCA for (6) with Gaussian noise

				Ts_PGA		FPCA	
(m, n)	r	sr	Fr	Time	RelErr	Time	RelErr
(200,200)	10	0.5	0.10	5.81	1.86e-004	6.18	2.07e-004
(400,400)	10	0.5	0.05	3.95	1.31e-004	43.87	1.43e-004
(600,600)	10	0.5	0.03	16.77	1.19e-004	141.90	1.13e-004
(800,800)	10	0.5	0.02	34.86	1.15e-004	62.10	1.00e-004
(1000,1000)	10	0.5	0.02	55.01	1.04e-004	70.21	8.81e-005

1. From this figure, we see that Ts_PGA successfully recovered the both corrupted images, and the quality of each recovered image is competitive with the one derived by FPCA.

Taking the above four cases together, it illustrates that Ts_PGA provides an alternative approach to low rank matrix completion problems, and its performance is competitive with or slightly better than the widely used solver FPCA.

5. Conclusions

In the field of machine learning, one often meets the problem of exploiting the low-rank matrix from its given noisy partial entries. It has been shown that the task can be characterized as a matrix nuclear-norm minimization problem. However, the non-smoothness of the nuclear norm make the problem is challenging to solve. Due to the simplicity and effectiveness of proximal gradient algorithm for a wide range of convex minimization problems, the

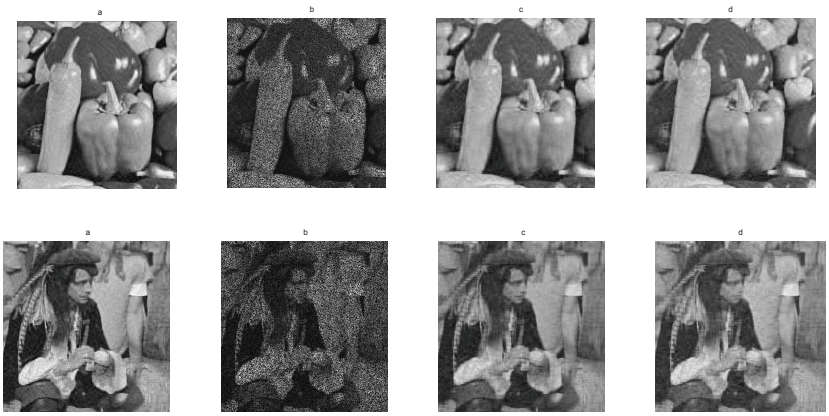

Figure 1. (first column): the original image with size 512×512 and rank $r = 40$; (second column): randomly masked image with $sr = 50\%$; (third column): recovered images by Ts_PGA with "RelErr" $6.88e-002$ (peppers) and $7.12e-002$ (man); (forth column): recovered images by FPCA, with "RelErr" $6.23e-002$ (peppers) and $7.30e-002$ (man).

algorithm and its accelerated variants have been intensively studied in the past few years. Particularly, the type of this algorithm was successfully used to solve matrix completion problem by Ma, Goldfarb, and Chen [11]. On the one hand, the proposed algorithm is actually an extension or generalization of the proximal gradient algorithm, in which the generated iteration is a combination of the previous point, the current point, and its proximal gradient point. On the other hand, our idea also comes from the effectiveness of the TwIST algorithm of Bioucas-Dias and Figueiredo [14] to recover large and sparse signal from the limited measurement. Hence, our algorithm can also be regarded as an extension of TwIST to solving matrix completion problems. Although both motivations are simple, the numerical experiments illustrated that the proposed algorithm performs well and is competitive with the well-known solver FPCA, which in turn showed the superiority of the proposed algorithm. Surely, this is the contribution of this paper. The paper was paid more attention on solving matrix completion problem where the Lipschitz constant of the gradient of the least square term is 1, which is essential to the convergence of the algorithm. However, its efficiency in solving general linear constrained nuclear norm minimization problems has not been studied. This should be our further task to investigate.

Acknowledgements

The work of Q. Wang was supported by the National Natural Science Foundation of China (Grant No. 11471101). The work of Z.-F. Jin was supported by the National Natural Science Foundation of China (Grant No. 11471102) and the Key Basic Research Foundation of the Higher Education Institutions of Henan Province(No.16A110012).

REFERENCES

1. L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Review, vol.38,pp.49–95,1996.
2. B. Recht, M. Fazel, and PA. Parrilo, *Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization*, SIAM Review,vol.52,pp.471–501,2010.
3. J.F. Cai, E.J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, preprint, SIAM J. Optim., vol.20, pp.1956–1982,2010.
4. E.J. Candès and B. Recht, *Exact matrix completion via convex optimization*, Foundations of Computational Mathematics, vol.9, pp.717–772,2009.
5. K.C. Toh, and S.W. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, Pacific J. Optim., vol.6, pp.615–640, 2010.
6. A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imag. Sci., vol.2, pp.183–202, 2009.
7. R.-P. Wen, X.-H. Yan, *A new gradient projection method for matrix completion*, App. Math. Comput., vol.258, pp.537–544, 2015.

8. H. Zhang, L.Z. Cheng, *Projected Landweber iteration for matrix completion*, J. Comput. Appl. Math., vol.235, pp.593–601,2010.
9. Y.-F. Li, Y.-J. Zhang, and Z.-H. Huang, *A reweighted nuclear norm minimization algorithm for low rank matrix recovery*, J. Comput. Appl. Math., vol.263, pp.338–350, 2014.
10. Y. Xiao, S.-Y. Wu, D.-H. Li, *Splitting and linearizing augmented Lagrangian algorithm for subspace recovery from corrupted observations*, Adv. Comput. Math., vol.38 , pp.837–858, 2013.
11. S. Ma, D. Goldfarb, and L. Chen, *Fixed point and Bregman iterative methods for matrix rank minimization*, Math. Program., vol.128, pp.321–353, (2011).
12. E.T. Hale, W. Yin, and Y. Zhang, *Fixed-point continuation for ℓ_1-minimization: methodology and convergence*, SIAM J. Optim., vol.19, pp.1107–1130, 2008.
13. Y.J. Liu, D. Sun, and K.C. Toh, *An implementable proximal point algorithmic frameword for nuclear norm minimization*, Math. Program., vol.133 , pp.399–436,2012.
14. J.M. Bioucas-Dias and M. Figueiredo, *A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoratin*, IEEE Trans. Image. Proces., vol.16, pp.2992–3004, 2007.
15. M. Elad, B. Matalon, and M. Zibulevsky, *Subspace optimization methods for linear least squares with non-quadratic regularization*, Appl. Comput. Harmon. Anal., vol.23, pp.346–367, 2007.
16. Y. Nesterov, *Gradient methods for minimizing composite objective function*, CORE Discussion Paper, Catholic University of Louvain, vol.76, 2007.
17. P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, SIAM J. Optim., available at http://pages.cs.wisc.edu/ brecht/cs726docs/Tseng.APG.pdf
18. B. O'Donoghue and E. Candès, *Adaptive restart for accelerated gradient schemes*, Found. Comput. Math., vol.15, pp.715–732, 2015.
19. O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, New York, 1996.
20. Z.-F. Jin, Z. Wan, Y. Jiao, and X. Lu, *An alternating direction method with continuation for nonconvex low rank minimization*, J. Sci. Comput., doi: 10.1007/s10915-015-0045-0.