Applied Mathematics, 2014, 5, 3445-3451
Published Online December 2014 in SciRes. http://www.scirp.org/journal/am
http://dx.doi.org/10.4236/am.2014.521322

Scientific Research

On Accelerated Singular Value Thresholding Algorithm for Matrix Completion

Li Wang, Jianfeng Hu*, Chuanzhong Chen

School of Mathematics and Statistics, Hainan Normal University, Haikou, China
Email: wlyoume@163.com, *lakerhjf@163.com, czchen@hainnu.edu.cn

Received 16 October 2014; revised 8 November 2014; accepted 16 November 2014

Abstract

An accelerated singular value thresholding (SVT) algorithm was introduced for matrix completion in a recent paper [1], which applies an adaptive line search scheme and improves the convergence rate from $O(1/N)$ for SVT to $O(1/N^2)$, where N is the number of iterations. In this paper, we show that it is the same as the Nemirovski's approach, and then modify it to obtain an accelerate Nemirovski's technique and prove the convergence. Our preliminary computational results are very favorable.

Keywords

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

1. Introduction

In many practical problems of interest, such as recommender system [2]-[4], one would like to recover a matrix from a small sampling of its entries. These problems can be formulated as the following matrix completion problem:

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

where $M \in \mathbb{R}^{m\times n}$ is the given incomplete data matrix, Ω is the set of locations corresponding to the observed entries. The problem (1) is NP-hard because the rank function is non-convex and discontinuous. However, it is known that the rank of a matrix is equal to the number of nonvanishing singular values, and the nuclear norm

*Corresponding author.

How to cite this paper: Wang, L., Hu, J.F. and Chen, C.Z. (2014) On Accelerated Singular Value Thresholding Algorithm for Matrix Completion. *Applied Mathematics*, **5**, 3445-3451. http://dx.doi.org/10.4236/am.2014.521322

$\|\cdot\|_*$ stand for the sum of the singular values. Thus, the rank minimization problem (1) can be relaxed as following convex program problem:

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

and Candès and Recht [5] proved that under suitable conditions, the program (1) and (2) are formally equivalent in the sense that they have exactly the same unique solution.

Now, there are many algorithms for solving the model (2), such as singular value thresholding (SVT) algorithm [6], accelerated proximal gradient (APG) algorithm [7] and so on. The singular value thresholding (SVT) algorithm solves the following problem:

$$\begin{cases} \min\limits_{X} & \tau\|X\|_* + \frac{1}{2}\|X\|_F^2 \\ s.t. & P_\Omega(X) = P_\Omega(M) \end{cases} \tag{3}$$

where P_Ω denotes the orthogonal projector onto the span of matrices vanishing outside of Ω so that the (i,j) th component of $P_\Omega(X)$ is equal to X_{ij} if $(i,j) \in \Omega$ and zero otherwise. In [6], Cai *et al.* showed that the optimal solution to the problem (3) converges to that of (2) as $\tau \to \infty$.

However, SVT has only a global convergence rate of $O(1/N)$, where N is the number of iterations. Then Liu Jun *et al.* [1] proposed an accelerated SVT algorithm by considering the Lagrange dual problem of (1.3) as follows:

$$\min_{Y \in \mathbb{R}^{m \times n}} h(Y) \tag{4}$$

where $h(Y) = -\tau\|D_\tau(P_\Omega(Y))\|_* - \frac{1}{2}\|D_\tau(P_\Omega(Y))\|_F^2 - \langle Y, P_\Omega(M - D_\tau(P_\Omega(Y)))\rangle$, D_τ is the soft-thresholding operator [1], and $h(Y)$ is convex and continuously differentiable with Lipschitz continuous gradient.

In the accelerated SVT algorithm, an adaptive line search scheme was adopted based on Nemirovski's technique [8]. For reference purpose, in the following we list the related key steps of the algorithm in [1]:

Step 7: compute $Y_{k+1} = S_k - \frac{1}{L_k} h'(S_k)$;

Step 8: if $h(Y_{k+1}) \le h(S_k) - \frac{1}{2L_k}\|h'(S_k)\|_F^2$ then;

Step 9: go to step 14;
Step 10: else;
Step 11: $L_k = 2L_k$;
Step 12: end if;
Step 13: end while;

Step 14: set $\omega = 2L_k \dfrac{h(S_k) - h(Y_{k+1})}{\|h'(S_k)\|_F^2}$, $L_{k+1} = h(\omega)L_k$, $h(\omega) = \begin{cases} 1, & 1 \le \omega \le 5, \\ 0.8, & \omega > 5. \end{cases}$

Comparing with Nemirovski's scheme for updating L_k, *i.e.* $L_{k+1} = L_k$ in step 14, Liu Jun *et al.* [1] [9] used a more flexible scheme, in which L_k is not required to monotonically increase. As L_k is decreased, the step size $1/L_k$ is increased, it is expected that the number of iterations may be reduced. The idea is really attractive. However, it is found that the approach is just the same as the Nemirovski's algorithm in the following section.

2. Main Procedure

First, we declare that the value of ω (in step 14) is always not bigger than 2, this means that it is just the same as Nemirovski's line search scheme.

Since $h(Y)$ is convex and continuously differentiable with Lipschitz continuous gradient, we have

$$h(Y_{k+1}) \ge h(V) + \langle h'(S_k), Y_{k+1} - S_k \rangle \tag{5}$$

$$h(Y_{k+1}) \le h(S_k) + \langle h'(S_k), Y_{k+1} - S_k \rangle + \frac{L}{2}\|Y_{k+1} - S_k\|_F^2 \tag{6}$$

where L is the Lipschitz gradient constant [10].

Define function $\phi_k(Y) = h(S_k) + \langle h'(S_k), Y - S_k \rangle + \frac{L}{2}\|Y - S_k\|_F^2$, then we have

$$\phi_k(Y_{k+1}) = h(S_k) + \langle h'(S_k), Y_{k+1} - S_k \rangle + \frac{L_k}{2}\|Y_{k+1} - S_k\|_F^2 \tag{7}$$

which along with $Y_{k+1} = S_k - \frac{1}{L_k}h'(S_k)$ gives

$$\phi_k(Y_{k+1}) = h(S_k) - \frac{1}{2L_k}\|h'(S_k)\|_F^2 \tag{8}$$

From (5), (6) and (7), we have

$$h(Y_{k+1}) - \frac{L}{2}\|Y_{k+1} - S_k\|_F^2 \le h(S_k) + \langle h'(S_k), Y_{k+1} - S_k \rangle \le h(Y_{k+1})$$

and

$$h(S_k) + \langle h'(S_k), Y_{k+1} - S_k \rangle = \phi_k(Y_{k+1}) - \frac{L_k}{2}\|Y_{k+1} - S_k\|_F^2$$

It follows that

$$\phi_k(Y_{k+1}) - \frac{L_k}{2}\|Y_{k+1} - S_k\|_F^2 \le h(Y_{k+1}) \le \phi_k(Y_{k+1}) + \frac{L - L_k}{2}\|Y_{k+1} - S_k\|_F^2$$

that is

$$\frac{L_k - L}{2} \le \frac{\phi_k(Y_{k+1}) - h(Y_{k+1})}{\|Y_{k+1} - S_k\|_F^2} \le \frac{L_k}{2} \tag{9}$$

Combining (8), (9), $\omega = 2L_k \frac{h(S_k) - h(Y_{k+1})}{\|h'(S_k)\|_F^2}$ and $Y_{k+1} = S_k - \frac{1}{L_k}h'(S_k)$, we obtain

$$\max\left\{1, \frac{2L_k - L}{L_k}\right\} \le \omega \le 2 \tag{10}$$

Then, we make a few improvement based on the original algorithm to obtain a revised algorithm. The overall steps can be organized as follows:

Algorithm 1 The Modified ASVT Algorithm

Step 1: Input: L_1, $t_{-1} = 0$, $t_0 = 1$, $Y_1 = Y_0 = 0$, $\eta \in (0,1)$;

Step 2: Output: Y_{N+1}, $X_{N+1} = D_\tau(P_\Omega(Y_{N+1}))$;

Step 3: for $k = 1, 2, \cdots, N$ do;

Step 4: compute $\beta_k = \frac{t_{k-2} - 1}{t_{k-1}}$;

Step 5: compute $S_k = Y_k + \beta_k(Y_k - Y_{k-1})$;

Step 6: while 1 do;

Step 7: compute $Y_{k+1} = S_k - \frac{1}{L_k}h'(S_k)$;

Step 8: if $h(Y_{k+1}) \le h(S_k) - \frac{1}{2L_k}\|h'(S_k)\|_F^2$ then;

Step 9: $L_{k+1} = \eta L_k$, go to step 3;

Step 10: else;
Step 11: $L_{k+1} = 2L_k$;
Step 12: end if;
Step 13: end while;
Step 14: set $t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$;
Step 15: end for.

The convergence of algorithm 1 is given by the following theorem. And in this algorithm, the upper bounds of the order of convergence have nothing to do with the initial value L_1, which means that the modified scheme improved and accelerated the Nemirovski's technique.

Theorem 1 For large enough k, the approximate solution Y_k obtained by the modified algorithm satisfies

$$h(Y_{k+1}) - h^* \le \frac{4LR_h^2}{(k+1)^2}$$

where R_h is the distance from the starting point to the optimal solution set.

Proof. According to the convergence of Nemirovski's algorithm ([8] Theorem 10.2.2), it has

$$h(Y_{k+1}) - h^* \le \frac{2L_{k+1}R_h^2}{(k+1)^2}$$

where $L_{k+1} \le \hat{L} = \max\{2L, L_1\}$. In the following we declare that $L_{k+1} \le 2L$ for large enough k in our modified algorithm.

Suppose that there exists a positive integer k such that $L_{k+1} > 2L$. Obviously, $L_{k+1} = \eta L_k$ or $L_{k+1} = \eta \times 2^l L_k$, where l denotes the number of implementing step 11 in kth iteration. Since the test condition in step 8 is satisfied when $L_k \ge L$, it easily follows that $2^l L_k < 2L$. Therefore, we have

$$L_{k+1} = \eta L_k > 2L$$

and then

$$L_k > \frac{2L}{\eta} > 2L$$

So we have a recurrence relation of the form $L_k = \frac{1}{\eta} L_{k+1}$, then by recurrence, we obtain

$$L_1 = \frac{1}{\eta} L_2 = \cdots = \left(\frac{1}{\eta}\right)^k L_{k+1} > 2 \times \left(\frac{1}{\eta}\right)^k L$$

Obviously, as $k \to \infty$, we can obtain $L_1 \to \infty$, which is impossible by finiteness of the initial value L_1. Thus, when k is large enough, we have $L_{k+1} \le 2L$. This completes the proof.

3. Computational Results

In this section, numerical experiments with MATLAB were performed to compare the performance of the Nemirovski's technique algorithm and further to gain an insight into the behavior of our approach on synthetic dataset.

Code Ne-SVT and M-ASVT, based on the Nemirovski's technique SVT method and our modified algorithm, respectively. Specific problems as follows: similar to the paper [1], we generate $m \times n$ matrices M of rank r by randomly selecting two matrices of M_L and M_R, with the dimension of $m \times r$ and $r \times n$, respectively. Each having independent identically distributed Gaussian entries, and setting $M = M_L M_R$. Suppose M_Ω is the observed part of M, and the set of observed indices $\Omega = \{(i, j)\}$ is sampled uniformly at random. Let p be the ratio, that is $p = n_z / mn$, where n_z is the number of the observed entries. Then different algorithms will be used to recover the missing entries from the partially observed information by solving the optimization problem (3) with a given parameter τ. In our tests, τ was set to $t\sqrt{mn}$ on the basis of Cai *et al.* [6], where $2 \le t \le 5$. We adopt the

relative reconstruction error defined by following:

$$\text{error} = \frac{\|X - M\|_F}{\|M\|_F} \tag{11}$$

where X is the computed solution of an algorithm, then there is using the "error" to evaluate the quality of the algorithm. In addition, we set $L_1 = 1$, $\eta = 0.8$ for all test problems. Compiled using MATLAB2010, both Ne-SVT and M-ASVT were run under a Windows XP system on a AMD Fusion APU E-450 1.65 Ghz personal computer with 1.98 GB of memory and about 16 digits of precision.

Firstly, we compare the relative error between Ne-SVT and M-ASVT for solving the randomly generated low rank matrix problem. The initial parameters as follows: m = 100, n = 50, r = 5, p = 0.9, so meaning that 90% entries are observed. We will recover the other 10% entries by running Ne-SVT and M-ASVT separately. **Table 1** reports the relative reconstruction error of different methods after 30, 60, 90 and 120 iterations. We can observe that the convergence rate of M-ASVT is really faster than that of the other, which is consistent with our analysis, and the further shows we can see **Figure 1**.

Then we can still test these algorithms with different settings on the following different low rank matrix completion problems: 1) 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 = 100, n = 50, r = 3, p = 0.9, and let τ change from $2\sqrt{mn}$ to $5\sqrt{mn}$; 2) Fix some number remains the same, and only let p change from 0.5 to 0.9; 3) Fix some number remains the same, let rank r change from 3 to 15; 4) Fix some number remains the same, only change the size of M.

Finally, **Table 2** shows the comparative results of randomly generated matrix completion problems, in which we choose error < 10^{-6} as the stop condition. Clearly, we can know that the more smaller τ, the more bigger p,

Table 1. Relative error comparison between Ne-SVT and M-ASVT.

Iterations	30	60	90	120
Ne-SVT	1.69e−04	7.48e−06	3.98e−07	1.09e−07
M-ASVT	3.80e−05	2.78e−07	2.95e−08	2.95e−08

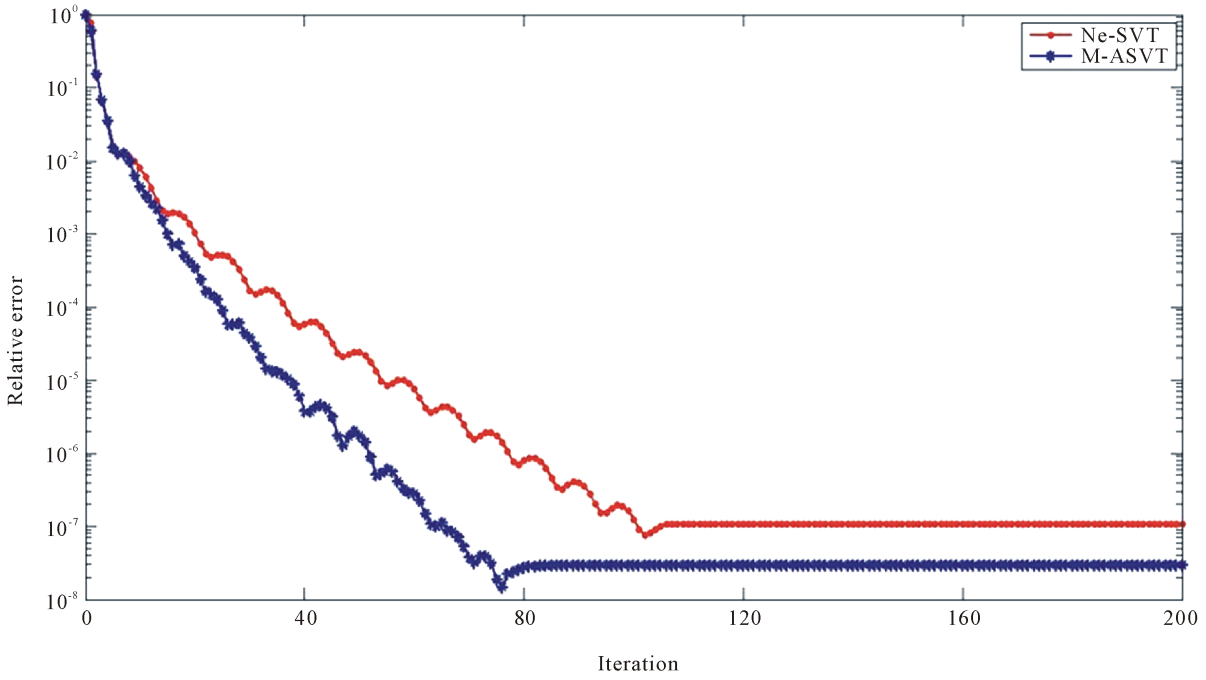

Figure 1. Convergence rate of Ne-SVT and M-ASVT on synthetic data.

Table 2. Comparisons between Ne-SVT and M-ASVT on the synthetic dataset with different settings.

	Settings					Iterations		CPU time (s)	
	m	n	r	p	τ	Ne-SVT	M-ASVT	Ne-SVT	M-ASVT
τ	100	50	3	0.9	$5\sqrt{mn}$	158	114	3.02	2.65
	100	50	3	0.9	$4\sqrt{mn}$	119	87	2.31	2.00
	100	50	3	0.9	$3\sqrt{mn}$	89	74	1.78	1.67
	100	50	3	0.9	$2\sqrt{mn}$	66	46	1.32	1.02
p	100	50	3	0.5	$2\sqrt{mn}$	200	122	4.02	2.87
	100	50	3	0.6	$2\sqrt{mn}$	200	112	3.88	2.60
	100	50	3	0.7	$2\sqrt{mn}$	147	96	2.89	2.24
	100	50	3	0.8	$2\sqrt{mn}$	94	61	1.85	1.38
r	100	50	5	0.9	$2\sqrt{mn}$	86	56	1.70	1.22
	100	50	8	0.9	$2\sqrt{mn}$	97	75	1.91	1.73
	100	50	10	0.9	$2\sqrt{mn}$	130	97	2.58	2.26
	100	50	15	0.9	$2\sqrt{mn}$	197	145	3.98	3.41
m, n	80	40	3	0.9	$2\sqrt{mn}$	69	51	0.81	0.70
	160	80	3	0.9	$2\sqrt{mn}$	65	45	3.61	2.85
	200	100	3	0.9	$2\sqrt{mn}$	56	39	5.45	4.32
	300	150	3	0.9	$2\sqrt{mn}$	52	38	13.37	10.84
	1000	500	3	0.9	$2\sqrt{mn}$	44	33	57.15	49.76

the more smaller rank r and the more bigger the size, then the more smaller the error and the more better efficiently. And we can observe that the M-ASVT performance surpasses Ne-SVT in all cases.

4. Conclusion

In this paper, we point out that the ASVT algorithm [1] is essentially the same as the Nemirovski's approach, then modify it to obtain the accelerate Nemirovski's approach and prove the convergence. We also give the comparative results of the convergence rate of Ne-SVT and M-ASVT. The preliminary computational results show that our approach is really more efficient. We empirically choose $\eta = 0.8$ $(0 < \eta < 1)$ in our test, and then we plan to study the better choice of η and develop the adaptive line search scheme to further improve the algorithm.

Acknowledgements

The research is supported by Natural Science Foundation of Hainan Province of China (No. 114006).

References

[1] Hu, Y., Zhang, D. and Liu, J. (2012) Accelerated Singular Value Thresholding for Matrix Completion. *Proceedings of the* 18*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, 12-16 August 2012, 298-306. http://dx.doi.org/10.1145/2339530.2339581

[2] Koren, Y. (2008) Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. *Proceedings of the* 14*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, 24-27

August 2008, 426-434. http://dx.doi.org/10.1145/1401890.1401944

[3] Koren, Y. (2010) Collaborative Filtering with Temporal Dynamics. *Communications of the ACM*, **53**, 89-97. http://dx.doi.org/10.1145/1721654.1721677

[4] Steck, H. (2010) Training and Testing of Recommender Systems on Data Missing Not at Random. *Proceedings of the* 16*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington DC, 25-28 July 2010, 713-722.

[5] Candès, E.J. and Recht, B. (2009) Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics*, **9**, 717-772. http://dx.doi.org/10.1007/s10208-009-9045-5

[6] Cai, J.F., Candès, E.J. and Shen, Z. (2010) A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization*, **20**, 1956-1982. http://dx.doi.org/10.1137/080738970

[7] Toh, K.C. and Yun, S. (2010) An Accelerated Proximal Gradient Algorithm for Nuclear Norm Regularized Least Squares Problems. *Pacific Journal of Optimization*, **6**, 615-640.

[8] Nemirovski, A. (1994) Lecture Notes on Efficient Methods in Convex Programming. http://www2.isye.gatech.edu/~nemirovs/Lect_EMCO.pdf

[9] Liu, J., Chen, J. and Ye, J. (2009) Large-Scale Sparse Logistic Regression. *Proceedings of the* 15*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, 28 June-1 July 2009, 547-556. http://dx.doi.org/10.1145/1557019.1557082

[10] Nesterov, Y. (2003) Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers, Norwell.