Available online at www.ijpe-online.com

International Journal of Performability Engineering

Vol. 13, No. 5, September 2017, pp. 643-656
DOI: 10.23940/ijpe.17.05.p9.643656

A Stochastic Sub-gradient Method for Low Rank Matrix Completion of Collaborative Recommendation

WeihuaYuan[a,b], Hong Wang[a,*], Baofang Hu[a] , Qian Sun[b]

[a]*School of Information Science and Engineering, Shandong Normal University, Jinan,250358,China*
[b]*School of Computer Science and Technology, Shandong Jianzhu University,Jinan,250101, China*

Abstract

In this paper, we focus on nuclear norm regularized matrix completion model in large matrices, and propose a new model named stochastic sub-gradient method for low rank matrix completion (SS-LRMC). To the problem of traditional SVT algorithm that would use one fixed threshold to shrink all the singular values during iterations, and the enormous computation burden when faced with large matrices, we define an adaptive singular value thresholding operator, and put forward a kind of matrix completion model applicable for user-item rating matrix of collaborative filtering. During iterations, we combine stochastic sub-gradient descent techniques with the adaptive singular value thresholding operator to obtain low rank intermediate solutions. Empirical results confirm that our proposed model and algorithm outperform several state-of-the-art matrix completion algorithms and the application to collaborative filtering recommendation can effectively solve the sparse problem of the user-item rating matrix and can significantly improve recommendation accuracy.

Keywords: Stochastic sub-gradient; Convex optimization; Collaborative filtering; Nuclear norm; Matrix completion

(Submitted on April 14, 2017; Revised on June 28, 2017; Accepted on August 12, 2017)

1. Introduction

The processing of large-scale and high dimensional data plays an increasingly important role in today's social life and scientific research. These data may contain a large number of missing values or various errors. For example, in facial recognition images may be shadowed, or may contain occlusion and deformation; and user-item rating matrix of the famous Netflix problem involves a large amount of unknown items. Matrix completion is one of the most commonly used techniques to solve such problems. The recovery of all elements from a partially observed set of entries is called matrix completion [15,21]. This is mission impossible because there are numerous possibilities of recovery for the items in the matrix, and without any additional constraints it is hard to determine which completion is optimal. However, if the original matrix is low rank or approximately low rank, it is possible to do accurate reconstruction for the matrix because the essential dimensions are significantly smaller than it looks. Matrix completion technology is widely used in computer vision, machine learning, recommender systems and sparse channel estimation etc. Candès and Tao [5] proved that most matrices can be accurately recovered by solving the nuclear norm minimization problem when the number of samplings meets some given conditions.

How to make efficient recommendations for customers has become a crucial step for most e-commerce companies [12]. Collaborative filtering as a popular recommender technique mainly relies on the user-item rating matrix to make recommendations. This rating matrix is ordinarily characterized by high dimensionality and high sparseness due to a large proportion of unknown elements. For instance, in user-movie rating matrix of Netflix, the number of known ratings only constitutes about one percent of the total. However it is commonly believed that only a few factors may contribute to one's preference to a film, such as film genres, cast, plot setting, performance skills and shooting effect etc. Consequently, it is reasonable to assume the matrix low rank or approximately low rank and this is a typical matrix completion problem.

* Corresponding author.
E-mail address: 1456029328@qq.com.

In this paper, we focus on stochastic sub-gradient method based nuclear norm regularized matrix completion problems, as well as its application to recommender systems. Our objective is to dispose of the sparseness of user-item rating matrix and to improve recommendation accuracy. The problem to be solved can be formulated as (1):

$$F(X) = \min_{X \in R^{m \times n}} f(X) + \tau \|X\|_* \tag{1}$$

Among which $f(X)$ is a convex function over $X \in R^{m \times n}$, $\|X\|_*$ denotes the nuclear norm of X, and τ is the regularization parameter. The contribution of the paper is as follows:

- The definition of varied iterative threshold τ_t and the adaptive singular value thresholding shrinkage operator $D_{\tau_t}(.)$

To the problem of traditional SVT that would use one fixed threshold τ to do shrinkage on all singular values of intermediate solutions $X^{(t)}$, we modify fixed τ to varied threshold τ_t suitable for user-item rating matrix. Based on this we also define an adaptive singular value thresholding shrinkage operator $D_{\tau_t}(.)$ and the corresponding matrix completion model applicable for collaborative filtering.

- The combination of stochastic sub-gradient methods with $D_{\tau_t}(.)$ to obtain low rank intermediate solutions.

To the large computational cost of traditional matrix completion algorithms, we obtain low-rank intermediate solutions by combining stochastic sub-gradient descent technique with the adaptive singular value thresholding shrinkage operator $D_{\tau_t}(.)$ during optimization process, to decrease computation cost as well as to increase its convergence speed.

- A kind of improved collaborative recommendation algorithm based on SS-LRMC.

It demonstrates empirically the proposed model and algorithms can effectively solve the sparse problem of the rating matrix and can enhance the prediction accuracy remarkably when applied to collaborative filtering recommendations.

Organization of this paper is as follows. Section 2 is mainly about related work, and we also introduce some fundamentals and some classical matrix completion models and algorithms. Section 3 is about our proposed model SS-LRMC, algorithms and the analysis of the algorithms. We demonstrate performance and effectiveness of the proposed algorithms through numerical experiments in Section 4. Finally, we conclude the paper in Section 5.

2. Related Work

2.1. Symbols used in this paper and their meanings

Firstly of all, we provide here a brief summary of the notations used in the paper. Matrices are bold capital, vectors are bold lowercase and scalars or entries are not bold. For a rank r matrix $X \in R^{m \times n}$, $m \le n$, X^T denotes its transpose, X^{opt} is its optimal and $X^{(t)}$ denotes the intermediate solutions in the t^{th} iteration. The singular value decomposition of X is represented as $X = U\Sigma V^T, U \in R^{m \times r}, V \in R^{n \times r}$, such that $U^T U = V^T V = I^r$, I^r is the $r-order$ identity matrix and $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_r)$ with diagonal elements satisfying $\sigma_1 \ge \sigma_2 \ge ... \ge \sigma_r$, among which σ_i is the i^{th} singular value of X, $i = 1, 2, ..., r$. X depends on $dr(X)$ degrees of freedom, that is, $dr(X) = (m + n - r)r$. The nuclear norm of X is represented as $\|X\|_*$ and the sub-gradient of the nuclear norm is expressed as $U_{1:m,1:r} V_{1:n,1:r}^T \in \partial \|X\|_*$ [17].

If the known entries of X is represented by matrix $M \in R^{m \times n}$, and its index set Ω is a uniformly random sampling set indicating positions of the non-zeroes in X, then, P_Ω is called projection selection operator of Ω, that is, P_Ω equals to X_{ij} for items belonging to Ω, and 0 otherwise.

$$P_\Omega(X) = \begin{cases} X_{ij}, & if\ (i, j) \in \Omega \\ 0, & otherwise \end{cases} \tag{2}$$

2.2. Related work

If rank r matrix X is low rank or approximately low rank, the complexity of matrix completion algorithms will be remarkably decreased if $r \ll \min(m,n)$. The problem can be formulated as:

$$\min rank(X) \text{ s.t. } P_\Omega(X) = P_\Omega(M) \tag{3}$$

Problem (3) has been proved to be NP-Hard, and Candès et al. [5] used nuclear norm instead as a convex relaxation to formulate the problem as (4) and proved that problem (3) and (4) are formally equivalent because they have the same unique solution.

$$\min \|X\|_* \text{ s.t. } P_\Omega(X) = P_\Omega(M) \tag{4}$$

When solving nuclear norm minimization problems like (1), traditional semi-definite tools such as SDPT could be done in polynomial time, but they would become invalid with matrix size exceeding 100×100. Jaggi [8] mapped the problem to convex optimization on a set of positive semi definite matrices with unit trace by invoking the Sparse-SDP [7]. However, this approach produces ε-accurate solutions with rank $\Theta(1/\varepsilon)$ orders of magnitude, which makes it challenging to hold the factorization in memory and to generate predictions in applications.

SVT [4] proposed by Cai and Candès, is a simple first order singular value thresholding algorithm. It carries out soft thresholding iterative shrinkage operations based on linear Bregman iteration. However, the running cost of SVT would increase rapidly as the rank becomes larger, which might even be prohibitively expensive when iterations needs to pass through a region where the spectrum is dense. Ma et al. [13] presented the fixed point continuation (FPC) algorithm based on continuous technologies, following the mathematical theories of SVT, with some slight difference in both the operating process and the given solution. M Chen et al. [6] proposed the accelerated neighbourhood gradient (APG) algorithm to solve matrix recovery problem by gradient method, which gains some improvement in both speed and accuracy compared to SVT and FPC. Because stochastic sub-gradient descent approaches can ensure reduction of both time and space complexity, Avron [1] combined it with non-convex projection M_r to (1) to keep the low rankness of intermediate solutions. However, the non-convexity of M_r would introduce cumulative errors and result in no guarantee of convergence.

Another category of optimization algorithms is based on matrix decomposition such as Low-rank Matrix Fitting (LMaFit) [18] and Riemannian Trust-Region for MC (RTRMC) [2]. This method usually approximates the original matrix with a product of two or more low-rank matrices, and the missing entries are recovered by solving the non-convex low rank approximation problems. These algorithms would obtain better prediction accuracy if we set the matrix rank with a larger value, but it is hard to make good explanations for their recommendations. In addition, these algorithms are sensitive to the initial solutions and the iterations could only converge to local optimum.

The algorithms mentioned above could run smoothly for matrices with moderate size considering both precision and speed; however with matrix size growing increasingly larger, time needs to solve such problems will grow exponentially. Therefore, it is of great significance to study efficient low rank recovery algorithms for large scale matrices in big data area.

3. Stochastic sub-gradient based matrix completion algorithms

3.1. The SVT algorithm

SVT [4] reconstructs low rank matrices by solving a convex optimization based nuclear norm minimization problem as (5):

$$\min_{X \in R^{m \times n}} \frac{1}{2}\|X\|_F^2 + \tau \|X\|_* \; s.t. P_\Omega(X) = P_\Omega(M) \tag{5}$$

For each singular value threshold $\tau \geq 0$, the singular value thresholding operator $D_\tau(.)$ is expressed as (6):

$$\begin{aligned} D_\tau(.) &= UD_\tau(\Sigma)V^T \\ D_\tau(\Sigma) &= diag(\max(0,\sigma_1-\tau),...,\max(0,\sigma_r-\tau)) \end{aligned} \tag{6}$$

$D_\tau(.)$ is a proximity operator associated with the nuclear norm of X, which simply applies a soft thresholding rule to all singular values of X. We have Theorem 2.1 for nuclear norm minimization problem:

Theorem 2.1 [4]: For $Y \in R^{m\times n}$, and $\tau \geq 0$, the singular value shrinkage operator obeys (7):

$$D_\tau(Y) = \arg\min_X \frac{1}{2}\|X-Y\|_F^2 + \tau\|X\|_* \tag{7}$$

Based on Theorem 2.1, the iterative sequence for solving (5) begins with $Y^{(0)} = 0^{m\times n} \in R^{m\times n}$ and is defined as (8):

$$\begin{cases} X^{(k)} = D_\tau(Y^{(k-1)}) \\ Y^{(k)} = Y^{(k-1)} + \delta_k P_\Omega(M - X^{(k)}) \end{cases} \tag{8}$$

With $\tau \to \infty$ and step size δ_k chosen properly, problem (5) will immensely converge to solutions of (4) by soft thresholding operation $D_\tau(.)$ on singular values of X.

3.2. Problems about the singular value threshold τ in SVT and our proposed model

In problem (5), parameter τ has double roles. Firstly, τ acts as singular value threshold, and the selection of singular values larger than τ by $D_\tau(.)$ is equivalent to selecting the principal components, which will keep the vector composed of singular values of X sparse. Higher values of τ will lead to lower rank solutions of (5).

Secondly, according to works in [19], problem (5) can be reformulated as the following unconstrained problem (9):

$$\min_X \frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 + \tau\|X\|_* \tag{9}$$

Based on projection operator P_Ω and the iterative process (8) of SVT, problem (9) and (5) have the same optimal solution X^{opt} [19]. Consequently, τ could also be used to balance the role of the loss term $\frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2$ and the regularization term $\|X\|_*$. Larger values of τ suggest that the latter is more important than the former, as a result X^{opt} must firstly meet the constraint of low rankness, then less training errors. The role of observed entries becomes smaller as values of τ get increasingly large. However, in recommender systems, the loss term does play a tremendously import role in the computation of root mean squared error (RMSE). RMSE usually acts as an important prediction index in collaborative recommendations, which must be low enough if we want to improve the prediction accuracy.

Consequently, both the items of (9) are essential when it comes to solve nuclear norm regularized matrix completion models. That is, to decrease the time and space cost of iterations, we need to keep the low rank property of the final solutions, and to improve prediction accuracy, we should simultaneously improve the effect of the loss term in (9). Based on the above analysis, model (9) is modified to (10) as our nuclear norm regularized low rank matrix completion model in collaborative filtering:

$$\min_X \frac{\alpha}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 + \tau_t\|X\|_* \tag{10}$$

Parameter α and τ_t are set to balance the role between minimizing the rank of intermediate solutions and improving the accuracy of recommendation. Here τ_t is no longer a fixed constant, but an adaptive variable during iterations.

3.3. The value of α and τ_t

When conducting matrix completion in collaborative filtering, we usually initialize all the entries of X with zero. In the iterations of (10), if τ_t is set to very large values as τ in SVT, to fit M the unknown positions might eventually be filled with zeroes with high probability, which deviates from our application requirements of predicting and recommending with these unknowns. Consequently, the whole model might fail if values of τ_t are chosen improperly.

Our experiments in section 4 show that decreasing τ_t always benefits the accuracy of results as far as RMSE is concerned. Hence, the choice of τ_t should reflect a trade-off between minimizing the matrix rank and enhancing prediction accuracy. Therefore, we divide the work of setting τ_t into two phases. Phase1 is about strengthening low rank constraints, that is, we qualify the value of threshold τ_t a larger constant to approximate the original objective of matrix completion--minimizing the nuclear norm. Stage 2 is about continuing to enhance the influence of prediction accuracy, that is, τ_t is gradually decreased to relax threshold limit to enhance the influence of the loss term in (10). Suppose the initial value of τ_t is represented as τ_0, T as the total number of iterations, and t the t^{th} iteration, then the varied threshold τ_t is defined as (11):

$$\tau_t = \begin{cases} \tau_0 & 1 \le t \le \left\lfloor \frac{T}{2} \right\rfloor \\ \tau_0 e^{(\frac{1}{2} - \frac{t}{T})} & \left\lceil \frac{T}{2} \right\rceil \le t \le T \end{cases} \tag{11}$$

Definition 3.1. Adaptive singular value thresholding shrinkage operator $D_{\tau_t}(.)$

Based on formula (11), our adaptive singular value thresholding shrinkage operator $D_{\tau_t}(.)$ can be redefined as (12):

$$\begin{gathered} D_{\tau_t}(.) = diag(\max(0, \sigma_1 - \tau_t), ..., \max(0, \sigma_r - \tau_t)) \\ D_{\tau_t}(M) = U D_{\tau_t}(\Sigma) V^T \end{gathered} \tag{12}$$

The value of τ_t will be gradually decreased, but its lower bound is ensured--the final threshold will not become too small according to (11), which illustrates the calculation of τ_t is reasonable when we need to follow low-rankness as a fundamental principle.

Parameter α is a tunable and auxiliary parameter to help increase contributions of the loss term. When iteration sequence $\{X^{(t)}\}$ must be kept very low rank based on characteristics of the data or application requirements, then τ_t might stop at a very large minimum even if it is gradually decreased according to (11). Under such circumstances, we can increase the value of α appropriately to help enhance the role of the loss term in (10).

3.4. Low rank solutions of the algorithm during the optimization process

According to traditional SVT, the singular value thresholding shrinkage operator $D_{\tau_t}(.)$ is the most computationally intensive part because it needs the singular value decomposition of matrix X, and the running cost

of SVT would increase rapidly as the rank becomes larger, which might even be prohibitively expensive when iterations need to pass through a region where the spectrum is dense. There will be more singular values larger that τ_t when it is calculated according to (11), which means the vectors composed of these singular values will not be as sparse as before. The running cost of the whole iterations might accordingly increase rapidly. In this case, the whole iterations might suffer from heavy computation if iterations are still done according to (8). Therefore, we combine stochastic sub-gradient descent technique with the adaptive singular value thresholding shrinkage operator $D_{\tau_t}(.)$, to obtain low-rank intermediate solutions.

3.4.1. Low rank intermediate solutions by stochastic sub-gradient descent technology

It has been proved that stochastic sub-gradient descent technology could reduce the time and space complexity, and that the final solution could converge to the optimal solution of the algorithm [16].

Definition 3.2. Sub-gradient of $g: R^n \to R$ at x is defined as the following set:

$$\partial g(x) = \{y \in R^n : \forall z \in R^n < y, z - x > \leq g(z) - g(x)\} \tag{13}$$

Sub-gradient descent is a method of solving such problems as $\min_X F(X)$. And the iterations are usually expressed as (14):

$$\hat{X}^{(t+1)} = \Pi_K(X^{(t)} - \eta^{(t)} g^{(t)}) \tag{14}$$

Solutions of the stochastic sub-gradient descent expressed as (14) will converge to the optimal as long as it is an unbiased estimator of the sub-gradient, that is, $E(g^{(t)}) \in \partial F(X^{(t)})$. As described before, Avron [1] combined stochastic sub-gradient descent technology with non-convex projection M_r to keep intermediate solutions low rank. M_r would retain the first r singular values of $X^{(t)}$ by conducting a truncating operation to the SVD of $\hat{X}^{(t)}$. Due to the non-convexity of M_r, a cumulative error will be produced and there will be no guarantee of convergence.

For solving the above problems, we firstly obtain a low rank stochastic sub-gradient of (10) and then adopt $D_{\tau_t}(.)$ defined in (12) instead of M_r, to do soft thresholding on singular values of matrix $\hat{X}^{(t)}$, and to obtain the low rank intermediate solutions $X^{(t)}$. The following is the detail, and we need the definition of probing matrix [1].

Definition 3.3. Probing matrix

Stochastic matrix $Y \in R^{m \times n}$ is a probing matrix, if it satisfies that $E\left[YY^T\right] = I_{n \times n}$, among which $I_{n \times n}$ is the identity matrix and the expectation is over the choice of Y.

There are three cases of commonly used sub-gradient probing techniques, and we use the scaled identity vectors (Case 3 of Lemma 3.1 in [1]) as our probing matrix, that is:

Supposing $Y = Z/\sqrt{k}$, $k \ll n$, Z is a stochastic matrix if it obeys such distributions as each column of Z is drawn uniformly randomly from $\left\{\sqrt{n}e_1, ..., \sqrt{n}e_n\right\}$ (scaled identity vectors) and independently of each other. And the corresponding low rank stochastic sub-gradient $g^{(t)}$ is represented as (15):

$$g^{(t)} = g(F(X^{(t)}))YY^T \tag{15}$$

Theorem 3.1. $g^{(t)}$ in (15) is a low rank and unbiased estimator of $g(F(X^{(t)}))$.

Proof.

By the choice of probing matrix Y composed of scaled identity vectors, we get $E(AYY^T) = AE(YY^T) = A$ for any matrix A. So we conclude that $g^{(t)}$ in (15) is an unbiased estimator of $g(F(X^{(t)}))$, that is, $E(g^{(t)}) \in \partial F(X^{(t)})$, and $g^{(t)}$ can be proved to be low rank:

$$rank(g^{(t)}) = rank(g(F(X^{(t)}))YY^T) =$$
$$rank((g(F(X^{(t)}))Y)Y^T) \le rank(g(F(X^{(t)}))Y) \le k$$

Consequently, $g^{(t)}$ is an unbiased and low rank estimator of $g(F(X^{(t)}))$.

3.4.2. Perform $D_{\tau_t}(.)$ operation to enforce low-rankness of intermediate solutions

After the sub-gradient descent operation of (14) and (15), the rank of intermediate solution $\hat{X}^{(t+1)}$ in the t^{th} iteration, represented as $rank(\hat{X}^{(t+1)})$, might be added to $rank(X^{(t)}) + k$, which might make the intermediate matrices full rank after $\lceil n/k \rceil$ iterations. Therefore we decided to conduct $D_{\tau_t}(.)$ to the SVD of $\hat{X}^{(t+1)}$ based on (12), to separate its singular values into two categories to enforce intermediate solutions low rank: those exceeding τ_t are preserved because they represent principle components of the matrix, while those smaller than τ_t are discarded because they might merely be noise. If the SVD of $\hat{X}^{(t+1)}$ is represented as $U^{(t+1)}\hat{\Sigma}^{(t+1)}(V^{(t+1)})^T$, then:

$$X^{(t+1)} = D_{\tau_t}(\hat{X}^{(t+1)}) = U^{(t+1)}D_{\tau_t}(\hat{\Sigma}^{(t+1)})(V^{(t+1)})^T \tag{16}$$

And we get the following inequality (17):

$$rank(X^{(t+1)}) = rank(D_{\tau_t}(\hat{X}^{(t+1)})) \le rank(\hat{X}^{(t+1)}) \tag{17}$$

In particular, if some singular values of $\hat{X}^{(t+1)}$ are smaller than τ_t, we get $rank(D_{\tau_t}(\hat{X}^{(t+1)})) < rank(\hat{X}^{(t+1)})$, which reduces the rank of $\hat{X}^{(t+1)}$ and enforces the convergence of iterations. Furthermore, (17) also indicates that our algorithm is well suitable for large scale matrix completion problems because the factorization of $X^{(t)}$ can be held efficiently in memory due to its low rankness.

3.5. Algorithms

The operations of the two steps described above will guarantee each intermediate solution $X^{(t)}$ low rank. The whole algorithm SS-LRMC is described as Algorithm 1. Firstly we represent the gradient of (10) as (18):

$$g(F(X)) = \alpha(P_\Omega(X) - P_\Omega(M)) + \tau_t(UV^T) \in \partial_X F(X) \tag{18}$$

Algorithm 1 Stochastic sub-gradient based low rank matrix completion (SS-LRMC)

Input: $X^{(0)} = 0^{m\times n}$, step size of stochastic sub-gradient descent $\eta^{(t)}$, initial of singular value threshold τ_0, dimension of probing matrix k, maximum iterative steps T, tuneable parameters α, and $\varepsilon = 10^{-4}$

Output: Optimal solution X^{opt}

1. For $t = 1$ to T do
2. Generate the $n \times k$ probing matrix Y
3. Calculate singular value thresholding τ_t according to (11)
4. Compute gradient $g(F(X^{(t)}))$ of $F(X^{(t)})$ according to (18)
5. Evaluate a low rank unbiased estimator $g^{(t)}$ of $g(F(X^{(t)}))$ according to (15)
6. Do stochastic sub-gradient descent according to (14)
7. Do Incremental SVD factorization of (14) to get SVD of $\hat{X}^{(t+1)}$, expressed as $U^{(t+1)}\hat{\Sigma}^{(t+1)}(V^{(t+1)})^T$
8. Do $D_{\tau_t}(.)$ operation on SVD of $\hat{X}^{(t+1)}$ according to (12) to get $X^{(t+1)}$, represented as (16)
9. If $\left\|P_\Omega(X^{(t+1)} - M)\right\|_F \Big/ \left\|P_\Omega(M)\right\|_F < \varepsilon$
10. then break
11. end for
12. return $X^{(t+1)}$

The reconstruction solution X^{opt} by SS-LRMC (represented as $\hat{M}$) can regarded as an approximately complete sampling set, based on which we can make reliable collaborative filtering recommendations. Firstly, we ought to calculate user similarities between pairs of users according to $\hat{M}$ and select the first h most similar users of target user u_i (usually called h nearest neighbors). Then a recommendation list is produced and can be recommended to u_i from his h nearest neighbors in the light of the first N most favorite films or books etc. Here we choose consine similarity (19) to measure the similarities between user u_i and u_j.

$$sim(u_i, u_j) = \frac{\sum_{h=1}^{n} \hat{M}[i,h] \bullet \hat{M}[j,h]}{\sqrt{\sum_{h=1}^{n} \hat{M}[i,h]^2} \bullet \sqrt{\sum_{h=1}^{n} \hat{M}[j,h]^2}} \tag{19}$$

Other similarity indices such as Jaccard similarity could also be used except the consine similarity in (19). The whole process is called collaborative filtering recommendation based on SS-LRMC (abbreviated as SS-LRMC-CF) and described in Algorithm 2.

Algorithm 2 Collaborative filtering recommendation based on SS-LRMC (SS-LRMC-CF)

Input: Reconstruction solution $\hat{M}$, number of neighbours h, parameter N

Output: Recommendation list $List_i$

1. Calculate the similarity for each pair of users based on (19)
2. Sort node pairs in descending order based on the consine value, represented as $L = (i, j)$
3. List top h elements in L, denoted as KL
4. for EACH element (i, j) in KL do
5. Produce the recommendation list for u_i, expressed as $List_i$
6. end for
7. return $List_i$

3.6. Algorithm analysis of SS-LRMC

The following is mainly about some properties of SS-LRMC and the time complexity analysis.

- Incremental SVD of $\hat{X}^{(t+1)}$

In SS-LRMC, our proposed $D_{\tau_t}(.)$ has the same property as $D_{\tau}(.)$, so $D_{\tau_t}(.)$ will also act just on the singular values of the matrix, without changing the appropriate singular vectors, which can also acquire the same distinct values although there might exist various SVD factorizations. Therefore, the SVD computation is essential to both SVT and our proposed SS-LRMC. Unfortunately, time complexity of SVD on $X \in R^{m\times n}$ is usually $O(mn^2)$, so executing traditional SVD operation to $\hat{X}^{(t+1)}$ might lead to heavy computations. However, incremental SVD factorization such as economy SVD [3] will substantially cut down the cost of the process, and it can reduce the time complexity to $O((m+n)(r+k)^2)$, among which parameter r is the matrix rank, $k \ll n$ is dimension of probing matrix Y. Therefore, we exploit incremental SVD on $\hat{X}^{(t+1)}$ after the stochastic gradient descent operation.

- Termination condition of the algorithm

For any $X \in R^{m\times n}$, $\|P_{\Omega}(X)\|_F \le \|X\|_F$ holds. During the optimization process we set $\left\|P_{\Omega}(X^{(t+1)} - M)\right\|_F \Big/ \|P_{\Omega}(M)\|_F < \varepsilon$ as termination condition, and we set $\varepsilon = 10^{-4}$ as proposed in [3].

- Time complexity analysis

For rank r matrix X, $k \ll n$, time complexity for Step 4 of Algorithm 1 is $O(mr)$, and the time cost for step 5 is $O(m(k+r)^2)$. The main computation cost in step 7, namely, the incremental SVD of (16), based on step 4 and economy SVD introduced in [3] is $O((m+n)(r+k)^2)$. According to the above analysis, the running time is dominated by economy SVD by $D_{\tau_t}(.)$. Consequently, the time complexity of the algorithm is $O((m+n)(r+k)^2)$.

4. Experiments and analysis

4.1. Datasets

- Real datasets

The dataset in our experiments is Movielens 10M, which contains about 10^7 integer ratings scale from 1 to 5 applied to 10681 movies by 71567 users of the online movie recommender service MovieLens, each user has rated at least 15 items. The sparse degree of the dataset is about 98.7%. We partition the dataset into training and test sets according to works by [1]. We also evaluated our algorithms on the Netflix data of more than 100 million movie ratings with integer values ranging from 1 to 5, performed by 480,189 anonymous Netflix customers on 17,770 films with nearly 99% unknown entries.

- Randomly generated datasets

Suppose $n \times n$ matrix M of rank r, generated by the product of two factor matrices $U \in R^{n\times r}$ and $V \in R^{n\times r}$, namely $M = UV^T$. Each entry of factor matrix U or V is sampled independently from the standard normal distribution $N(0,1)$, respectively. Ω is the uniformly randomly sampled index set with cardinality $|\Omega|$, $dr(M)$ is the degrees of freedom mentioned in Section 2.1, $|\Omega|/n^2$ is the fraction of observations of M, and $|\Omega|/dr(M)$ is ratio between number of sampled entries of M and its degrees of freedom.

4.2. Evaluation index

- Index for matrix completion

Supposing that $\hat{M}$ denotes the reconstruction solution of M, then the fit error and the prediction error are defined as (20) and (21):

$$fit-err=\left\|P_{\Omega}(\hat{M}-M)\right\|_F/n \tag{20}$$

$$pre-err=\left\|\hat{M}-M\right\|_F/n \tag{21}$$

Among which n is the dimension of our randomly generated matrix M .

- Index for Recommender systems

The accuracy of predicting ratings is expressed as closeness between a user's real rating and the prediction score of the algorithm. We use RMSE as the index of predicting accuracy, that is, to ratings not appearing in training set but in test set we will compute their RMSE:

$$RMSE=\sqrt{\frac{1}{|TEST|}\sum_{(u,v,R_{u,v})}(R_{u,v}-estimated)^2} \tag{22}$$

In which $TEST$ represents the test set, and $|TEST|$ is the number of people's ratings in test set. Smaller value of RMSE indicates higher predicting accuracy.

4.3. Comparison of SS-LRMC with several state of the art algorithms on random generated matrices

All the experiments in this section were done on a modest desktop computer with an Intel (R) Core (TM)i7-2600, 3.4GHZ processer with 4 GB of memory. The Operating System is Window 8.1, and all the programs are operated in MATLAB 2009b.We generated a 1000×1000 square matrix M with $r=10,25,50$, respectively. All the numerical results are averaged over five runs. In this section we compare our proposed SS-LRMC with the following methods SVT [4] and SSGD [1].

Figure 1 plots the varying curve of prediction error (20) and fit error (21) of our proposed algorithm as a function of iterations for rank 25square matrices with n = 1000. From the figure we can see that both the prediction error and fit error continue to fall with the number of iterations, and after about 90 iterations, the prediction error converges to a certain constant, which demonstrates the convergence of our proposed method. Furthermore, the figure also demonstrates that the prediction error is marginally above the fitness error, and they get very close after 70 iterations, which illustrates the validity of the stopping criteria in Algorithm 1.

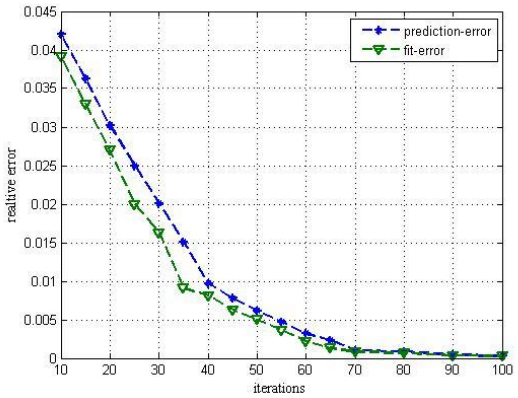

Figure. 1. Trend of prediction error and fitness error as a function of iterations for rank 25 square matrices with n = 1000 of SS-LRMC

Table 1 shows the comparative results between number of iterations and the corresponding running time to reach convergence among algorithms of SVT, SSGD and SS-LRMC with different values of rank r , $|\Omega|/dr(M)$ and $|\Omega|/n^2$. From Table 1, we could see that SS-LRMC performs better in terms of iterative numbers and running time. For the same values of rank r , $|\Omega|/dr(M)$ and $|\Omega|/n^2$, the proposed method will converge to the optimal after the least number of

iterations; and it runs much faster than the other two algorithms. If we fix $|\Omega|/dr(M)$ or $|\Omega|/n^2$, we find that the larger value of r is, the more number of iterations needed to achieve the optimum, among which SVT reports the largest increase in the number of iterations. And if we fix the rank, we find the larger the value of $|\Omega|/n^2$, which means the quantitative increase of elements in Ω, the number of iterations to reach the optimal shows a declining tendency for all the three algorithms.

Table 1. Performance comparison of three algorithms under different values of Rank, $|\Omega|/dr(M)$ and $|\Omega|/n^2$

r	$\|\Omega\|/dr(M)$	$\|\Omega\|/n^2$	SVT		SSGD		SS-LRMC	
			Iter#	**time**	**Iter#**	**time**	**Iter#**	**time**
10	5	0.019	112	270	102	280	99	241
10	3	0.011	129	450	110	509	101	309
25	5	0.049	123	2199	112	1830	89	1620
25	3	0.029	148	2380	124	2192	95	1993
50	5	0.099	196	6084	133	4090	118	3998
50	3	0.059	221	7391	149	5001	125	4002

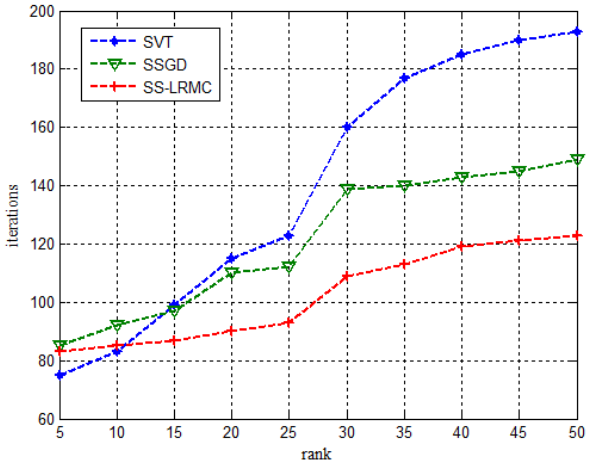

Figure. 2. Comparison of different number of iterations for SVT, SSGD and the proposed algorithm SS-LRMC needed to reach the optimal under same rank

Figure 2 shows the different number of iterations for SVT，SSGD and the proposed algorithm SS-LRMC needed to obtain the optimal under the same rank. When rank=10, number of iterations of the three algorithms to reach the optimal has only a little difference, indicating that SVT would work very well when the matrix to be reconstructed is very low rank. However, with rank becoming larger, the number of iterations needed for the three algorithms to reach the optimal increase gradually. SVT gets the largest increase in number of iterations, SSGD follows, and SS-LRMC gets a minor rise in number of iterations. This illustrates that the low-rank intermediate solutions $X^{(t)}$ of SS-LRMC, which incorporates the adaptive singular value thresholding shrinkage operator $D_{\tau_t}(.)$ with stochastic sub-gradient descent techniques, can reduce time and space cost and can accelerate algorithm convergence to acquire better performance.

4.4. Performance comparison of the algorithms on real datasets

In this section, we firstly run SVT, SSGD and SS-LRMC on real data set Movielens and apply their matrix completion results to recommender systems and calculate their RMSE, respectively, and we can see that all the algorithms acquire better RMSEs on Movielens.

The RMSE in Figure 3 demonstrates that low rank matrix completion technology is applicable for collaborative filtering recommendations. With the increase of the matrix rank, RMSE of SVT shows a downward trend, but still much higher than SSGD and SS-LRMC, and the difference in RMSE widens between SVT and SS-LRMC with the number of iterations.

This illustrates that although SVT works well in matrix completion with rank r taking very small values, but in terms of RMSE, it does not work so well because larger τ means making no full use of the loss item in model (10). On the other hand, our proposed method gets the lowest RMSE which means the best recommendation precision among the three algorithms, due to our strategy of using the varying singular value thresholding shrinkage operator $D_{\tau_t}(.)$, which will lower the threshold during iterations to get better RMSE. Consequently our proposed method is more applicable for low rank matrix completion in collaborative filtering recommendations.

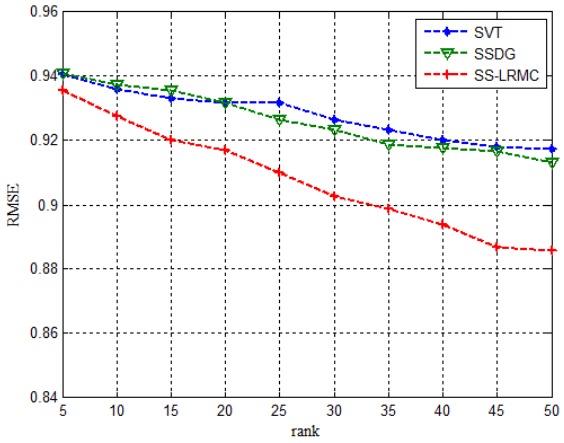

Figure. 3. RMSE comparison on Movielens of the three algorithms

Figure 4 depicts the RMSE comparison on Netflix among SVD [20], PMF [14], SS-LRMC-CF and the traditional neighbourhood method, which made recommendations according to (23).

$$\hat{r}_{ui} = b_{ui} + \sum_{j \in S^k(i;u)} S_{ij}(r_{uj} - b_{uj}) \Big/ \sum_{j \in S^k(i;u)} S_{ij} \tag{23}$$

For SVD and PMF, the rank in figure 4 corresponds to factors or dimensions of matrix feature vector, and it is irrelevant to the traditional neighborhood model (23). From the figure we can see that our proposed method delivers more accurate results with RMSE falls to 0.8831 for k=50, which is much lower than SVD with a RMSE of 0.9175 and PMF with a RMSE of 0.9130 for the same k value. In the literature, both the PMF and SVD would achieve lower RMSE values when the dimensions of their matrix feature vector raise up to rank=100 or rank=200, however higher rank would result in higher space and time cost. Our algorithm always benefits the accuracy of results as far as RMSE is concerned by decreasing the value of τ_t and producing low rank intermediate and final solutions in the iterations. And we are sure that RMSE will be further decreased if we integrate implicit or explicit feedback, such as trust relationships, into our matrix completion results to make recommendations, according to conclusions from works in [9,10,11].

5. Conclusions and future work

In this paper, we make a comparative study of several state-of-the-art matrix completion algorithms, and concern the matrix completion problem by convex optimization technologies. We also build the algorithmic frameworks according to the idea of low rankness and sparse assumption to efficiently solve nuclear norm regularized matrix completion for large-scale problems. The experiments indicate that our proposed method always delivers good prediction accuracy and decrease the time complexity, which works well in both low rank matrix completion and collaborative filtering. Future work contains comparison studies between work of low rank matrix completion and low rank matrix factorization; to combine matrix completion technology with explicit or implicit feedback, like the trust relationships or influence of fake reviews to further reduce the RMSE and to improve recommendation accuracy; as well as more efficient algorithms to further decrease intermediate time and space expenses and broader the application area of our proposed methods.

Acknowledgements

This work is supported by The National Natural Science Fund (No. 61672329, 61373149, 61472233), and the Technology Program of Shandong Province under Grant(No.2014GGB01617), Excellent course project of Shandong Province(No.2012BK294, 2013BK399, 2013BK402) and educational science planning project of Shandong Province(No.ZK1437B010).

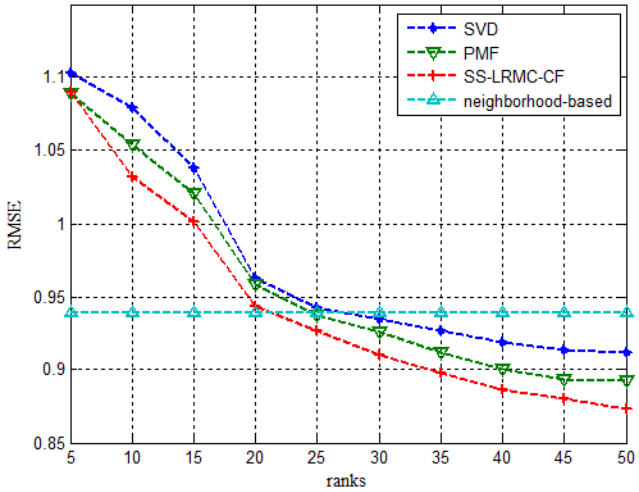

Figure. 4. RMSE comparison on Netflix among SVD, PMF, SS-LRMC-CF and neighbourhood-based method

References

1. H. Avron, S. Kale, and S. Kasiviswanathan, et al., "Efficient and Practical Stochastic Subgradient Descent for Nuclear Norm Regularization," in *Proceedings of the 29th International Conference on Machine Learning*, pp. 1231-1238, 2012
2. N. Boumal, and P. Absil, "RTRMC: a Riemannian Trust Region Method for Matrix Completion," in *Proceedings of the 25th Annual Conference on Neural Information Processing Systems*, Granada, Spain, pp. 406-414, 2011
3. M. Brand, "Fast Low-rank Modifications of the Thin Singular Value Decomposition," *Linear Algebra and its Applications*, 415(1), pp. 20-30, 2006
4. J. F. Cai, E. J. Candès, and Z. Shen, "A Singular Value Thresholding Algorithm for Matrix Completion," *SIAM Journal on optimization*, 20(4),pp. 1956-1982, 2010
5. E. J. Candès, and T, Tao, "The Power of Convex Relaxation: Near Optimal Matrix Completion," *IEEE Transactions on Information Theory*, 56(5), pp. 2053-2080, 2010
6. M. Chen, A. Ganesh, Z. Lin, and Y. Ma,etc., "Fast Convex Optimization Algorithms for Exact Recovery of a Corrupted Low-rank Matrix," *Journal of the Marine Biological Association of the Uk*, 56(3), pp. 707-722, 2015
7. E. Hazan, "Sparse Approximate Solutions to Semidefinite Programs," in *Latin American Conference on Theoretical Informatics*, Springer-Verlag, pp. 306-316, 2008
8. M. Jaggi, and M Sulovský, "A Simple Algorithm for Nuclear Norm Regularized Problems," in *International Conference on Machine Learning*, pp. 471-478, June 2010
9. M. Jamali, and M. Ester, "TrustWalker: a Random Walk Model for Combining Trust-based and Item-based Recommendation," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 397-406, 2009
10. F. KLIU, and H. J. LEE, "Use of Social Network Information to Enhance Collaborative Filtering Performance," *Expert Systems with Applications*, 37(7), pp. 4772-4778, 2010
11. Y. Koren, "Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA*, pp. 426-434, August 2008
12. L. Lü, M. Medo, H. Y. Chi, Y. C. Zhang, Z. K. Zhang and T. Zhou, "Recommender Systems," *Physics Reports*, 519(1), pp. 1-49, 2012
13. S. MA, D. Goldfarb, and L. Chen, "Fixed Point and Bregman Iterative Methods for Matrix Rank Minimization," *Mathematical Programming*, 128(1), pp. 321-353, 2011
14. A. Mnih, and R. Salakhutdinov, "Probabilistic Matrix Factorization", *Advances in neural information processing systems*, pp. 1257-1264, 2007
15. Y. G. Peng, J. L. Suo, Q. H. Dai, and W.L, Xu, "From Compressed Sensing to Low-rank Matrix Recovery: Theory and Applications," *Acta Automatica Sinica*, 39(7), pp. 981-994, 2013
16. N. Srebro, and A. Tewari, "Stochastic Optimization for Machine Learning," *ICML Tutorial*, 2010
17. G. A. Watson, "Characterization of the Subdifferential of some Matrix Norms," *Linear Algebra and its Applications*, 170(0), pp. 33-35, 1992
18. Z. W. Wen, W. T. Yin, and Y. Zhang, "Solving a Low-rank Factorization Model for Matrix Completion By a Nonlinear Successive Over-relaxation Algorithm," Rice University, Technical Report, pp. 1-24, 2010
19. J. Yang, X. Yuan, "Linearized Augmented Lagrangian and Alternating Direction Methods for Nuclear Norm Minimization", *Mathematics of Computation*, pp. 301-329, 2013
20. S. Zhang, W. Wang and J. Ford, "Using Singular Value Decomposition Approximation for Collaborative Filtering", *E-Commerce Technology*, pp. 257-264, 2005
21. Y. J. Zhao, B. Y. Zheng and S. N. Chen, "Matrix Completion and its Application in Signal Processing," *Journal of Signal processing*, pp. 423-436, 2015

Weihua Yuan was born in Qingdao, Shandong, China, in 1977. She gained her Bachelor's degree in management science and engineering from Shandong normal university in 2000, and Master's degree in computer application from Guizhou University in 2006. She has been working in school of computer science and technology, Shandong Jianzhu University since 2006, and she is also a PhD student working under supervision of Dr. Hong Wang in Shandong Normal University since 2012. She is author of two books, more than 10 articles. Her research interests include recommender systems, data mining, machine learning and big data cloud.

Hong Wang was born in Tianjin, China, in 1966. She gained her Master's and Bachelor's degree in computer software of Tianjin University in 1988, gained PHD of the Chinese academy of sciences in 2002. She has been working in Shandong Normal University since 1991. Currently she is a professor and doctoral supervisor of Computer Science department at School of Information Science and Engineering. She writes and publishes over 60 academic articles. Her main research interests include complex networks, workflow, mobile agent, social software and big data cloud.

Baofang Hu was born in Tai'an, Shandong, China, in 1979. She gained her Bachelor's degree from school of information science and engineering, Shandong Normal University in 2003, and Master's degree from school of information science and engineering, Shandong Normal University in 2006. She is now a PhD student in the same school under the supervision of Professor Hong Wang. She is author of seven articles, and her research interests include issues related to data mining and complex networks

Qian Sun was born in ZaoZhuang, Shandong, China, in1983. She gained Bachelor's degree in computer science from Tong ji University in 2004, and Master's degree in computer application from Tongji University in 2007. She has been working in school of computer science and technology, Shandong Jianzhu University since 2007. She is author of 3 articles. Her research interests include Social Networks, data mining and work flow.