Ann. Appl. Math.
doi: 10.4208/aam.OA-2023-0010

Vol. **39**, No. 2, pp. 181-205
May 2023

Nonnegative Low Rank Matrix Completion by Riemannian Optimalization Methods

Guang-Jing Song[1] and Michael K. Ng[2,*]

[1] *School of Mathematics and Information Sciences, Weifang University, Weifang, Shandong 261061, China*
[2] *Department of Mathematics, The University of Hong Kong, Hong Kong, China*

Received 18 April 2023; Accepted (in revised version) 5 May 2023

Dedicated to the memory of Professor Zhongci Shi

Abstract. In this paper, we study Riemannian optimization methods for the problem of nonnegative matrix completion that is to recover a nonnegative low rank matrix from its partial observed entries. With the underlying matrix incohence conditions, we show that when the number m of observed entries are sampled independently and uniformly without replacement, the inexact Riemannian gradient descent method can recover the underlying n_1-by-n_2 nonnegative matrix of rank r provided that m is of $\mathcal{O}(r^2 s\log^2 s)$, where $s=\max\{n_1,n_2\}$. Numerical examples are given to illustrate that the nonnegativity property would be useful in the matrix recovery. In particular, we demonstrate the number of samples required to recover the underlying low rank matrix with using the nonnegativity property is smaller than that without using the property.

AMS subject classifications: 15A23, 65f22

Key words: Manifolds, tangent spaces, nonnegative matrices, low rank.

---

*Corresponding author.
Emails: `sgjshu@163.com` (G. Song), `mng@maths.hku.hk` (M. Ng)

1 Introduction

Matrix completion, the problem of filling the missing elements by partially observed matrices became popular after the Netflix prize competition which was held in 2006. In order to avoiding being an underdetermined and intractable problem, low rank is often a necessary hypothesis to restrict the degree of freedoms of the missing entries. The matrix completion problem can be formulated as the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \operatorname{rank}(X) \\ \text{subject to} \quad & P_\Omega(X)=P_\Omega(A), \end{aligned} \tag{1.1}$$

where $X\in\mathbb{R}^{n_1\times n_2}$ is the decision variable, the set Ω of locations corresponding to the observed entries($(i,j)\in\Omega$ if A_{ij} is observed) is a set of cardinality m sampled uniformly at random, and the corresponding sampling operator P_Ω is defined by

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

In general, the rank minimization problem listed in (1.1) is NP-hard and computationally intractable. Many methods were proposed to solve the matrix completion problem, see for instance [1–4, 6–14]. In general, it can be divided into two categories: convex and non-convex optimization methods. Under the framework of convex optimization, the nuclear norm minimization problem

$$\begin{aligned} \text{minimize} \quad & \|X\|_* \\ \text{subject to} \quad & P_\Omega(X)=P_\Omega(A), \end{aligned} \tag{1.2}$$

is often applied to recover the unknown matrix entries, where the nuclear norm $\|X\|_*$ of a matrix X is defined as the sum of its singular values. With some suitable assumptions (incoherence conditions), it has been shown that if the number of observed entries satisfies $m\sim\mathcal{O}(sr^2\log^\alpha s)$ for some $\alpha\geq 0$, the underlying rank r matrix can be exactly recovered with high probability, where $s=\max\{n_1,n_2\}$. Meanwhile, many computationally efficient algorithms are designed to solve model (1.2), see [15–18] and references therein. On the other hand, there are non-convex optimization methods for solving (1.1) by parameterizing in a factorization form or studying in a set of fixed rank matrices. The computational cost of most non-convex algorithms are shown to be cheaper than that of the convex methods. The major issue is how to choose suitable initial guesses in non-convex optimization methods such that they can converge to the underlying low rank solution.

Nonnegative data matrices appear in many data analysis applications. For instance, in image analysis, image pixel values are nonnegative and the associated nonnegative image data matrices can be formed for clustering and recognition, see for instance [19–30]. In text mining, the frequencies of terms in documents are nonnegative and the resulted nonnegative term-to-document data matrices can be constructed for clustering, see for example [31–34]. In bioinformatics, nonnegative gene expression values are studied and non- negative gene expression data matrices are generated for diseases and genes classification, see [35–39]. The main aim of this paper is to study the problem of nonnegative matrix completion by introducing nonnegativity requirement for X in (1.1). In the literature, Xu et al. [42] proposed to solve the nonnegative matrix completion problem by using an algorithm based on the classical alternating direction augmented Lagrangian method. In [43], Xu and Yin designed a block coordinate descent method to study the nonnegative matrix completion problem. However, there is no theoretical result for the exact recovery of the underlying nonnegative low rank matrix.

In this paper, we study the Riemannian optimization methods for the problem of nonnegative matrix completion:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|P_\Omega(X)-P_\Omega(A)\|_F^2 \\ &\text{subject to} \quad \text{rank}(X)=r, \quad X\geq 0, \end{aligned} \tag{1.3}$$

that is to recover a nonnegative low rank matrix from its partial observed entries. With the underlying low rank matrix incohence conditions, we show that when the number m of observed entries are sampled independently and uniformly without replacement, the inexact Riemannian gradient descent method can recover the underlying n_1-by-n_2 nonnegative matrix of rank r provided that m is of $\mathcal{O}(r^2 s\log^2 s)$, where $s=\max\{n_1,n_2\}$. Numerical examples are given to illustrate that the nonnegativity property would be useful in the matrix recovery. In particular, we demonstrate the number of samples required to recover the underlying low rank matrix with using the nonnegativity property is smaller than that without using the property.

The rest of this paper is organized as follows. In Section 2, we study the Riemannian gradient descent method for solving (1.3). In Section 3, we provide the bounds on the number of sampled entries required for nonnegative low rank matrix completion. In Section 4, numerical examples are given to show the advantages of the proposed methods. Finally, some concluding remarks are given in Section 5.

2 The proposed algorithm

2.1 Mathematical preliminaries

Denote

$$\mathcal{M}_r := \left\{X \in \mathbb{R}^{n_1\times n_2},\ \mathrm{rank}(X)=r\right\}, \tag{2.1}$$

as the set of all $n_1\times n_2$ rank r matrices. It is well known that $\mathcal{M}_r$ forms an embedded submanifold of the set of matrices with size $n_1\times n_2$. And when the matrices in $\mathcal{M}_r$ be endowed with the usual trace inner product, then $\mathcal{M}_r$ forms a Riemannian submanifold of the embedding space $\mathbb{R}^{n_1\times n_2}$. Denote

$$\mathcal{M}_n := \left\{X \in \mathbb{R}^{n_1\times n_2},\ X_{i,j}\geq 0,\ i=1,\cdots,n_1,\ j=1,\cdots,n_2\right\} \tag{2.2}$$

as the $n_1\times n_2$ nonnegativity matrices set. The projection onto the fixed rank matrix set $\mathcal{M}_r$ is derived by the Eckart-Young-Mirsky theorem [40] which can be expressed as

$$\pi_r(X)=\sum_{i=1}^{r}\sigma_i(X)u_i(X)v_i^T(X), \tag{2.3}$$

where $\sigma_i(X)$, $i=1,\cdots,r$ are first r singular values of X, and $u_i(X)$, $v_i(X)$ are first r columns of the unitary matrices of $U(X)$ and $V(X)$. The projection onto the nonnegative matrix set $\mathcal{M}_n$ is expressed as

$$\pi_+(X)=\begin{cases} X_{ij}, & \text{if } \ X_{ij}\geq 0,\\ 0, & \text{if } \ X_{ij}<0. \end{cases} \tag{2.4}$$

Let $X\in\mathbb{R}^{n_1\times n_2}$ be an arbitrary matrix in the manifold $\mathcal{M}_r$, and $X = U(X)\Sigma(X)V(X)^T$ be a skinny SVD decomposition of X. It follows from [41, Proposition 2.1] that the tangent space of $\mathcal{M}_r$ at X can be expressed as

$$T_{\mathcal{M}_r}(X)=\{U(X)W^T+ZV(X)^T \mid W\in\mathbb{R}^{n_2\times r},\ Z\in\mathbb{R}^{n_1\times r} \text{ are arbitrary}\}. \tag{2.5}$$

For a given matrix Y, the projection of Y onto the subspace $T_{\mathcal{M}_r}(X)$ can be written as

$$P_{T_{\mathcal{M}_r}(X)}(Y)=U(X)U(X)^TY+YV(X)V(X)^T-U(X)U(X)^TYV(X)V(X)^T. \tag{2.6}$$

2.2 The inexact Riemannian gradient descent method

In this subsection, an Inexact Riemannian gradient descent method (IRGD) is studied to solve the nonnegative low rank matrix completion problem. Different from

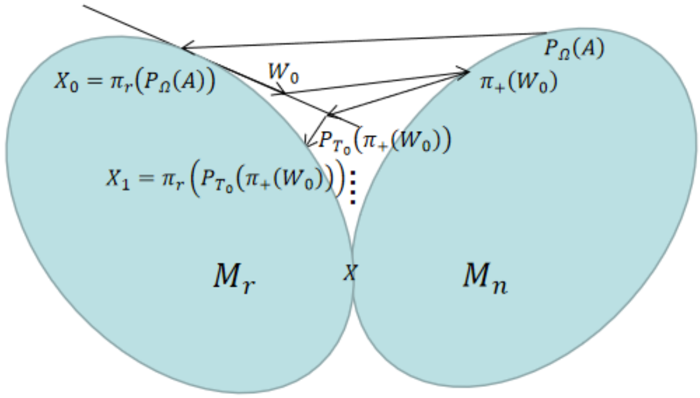


Figure 1: Inexact Riemannian gradient decent method using tangent spaces.

the Riemannian gradient descent method [17], there is an additional step in the inexact version that projects the iterate onto the nonnegative matrix manifold $\mathcal{M}_n$, see Fig. 1.

In the algorithm, $X_0=\pi_r(P_\Omega(A))$ is a SVD (singular value decomposition) truncation of the observed matrix $P_\Omega(A)$. T_l refers to the tangent space of $\mathcal{M}_r$ at X_l defined as in (2.5), and $P_{T_l}(G_l)$ refers to the projection of G_l onto T_l defined as in (2.6). Here the subscript refers to the iteration index. The summary of Algorithm 2.1 is given as follows.

Algorithm 2.1 Inexact Riemannian gradient decent method.

Initilization: $X_0=\pi_r(P_\Omega(A)),\Omega$ is a set of cardinality m sampled uniformly at random.

for $l=0,1,\cdots,$ do

1: $G_l=P_\Omega(A-X_l)$;

2: $\alpha_l=\frac{\langle P_{T_l}(G_l),P_{T_l}(G_l)\rangle}{\langle P_{T_l}(G_l),P_\Omega P_{T_l}(G_l)\rangle}$;

3: $W_l=X_l+\alpha_l P_{T_l}(G_l)$;

4: $W_l'=\pi_+(W_l)$;

5: $X_{l+1}=\pi_r(W_l')$;

end for

Output: X_l when the stopping criterion is satisfied.

We see from Step 4 that we projects the iterate W_l (which is derived by updating from X_l along the gradient descent direction on T_l) onto the nonnegative matrix manifold $\mathcal{M}_n$ to get W_l' via π_2. In Step 5, X_{l+1} is updated by using the svd truncation of W_l'.

In order to show the new sequence generated by Algorithm 2.1 is convergent, the following lemma proposed in [17] will be used in the sequel.

Lemma 2.1 (Lemma 4 in [17])**.** *Let X_l be a rank r matrix and T_l be the tangent space of the rank r matrix manifold at X_l. Suppose that X is another rank r matrix on the manifold $\mathcal{M}_r$, and $\sigma_{\min}(X)$ denotes the minimum singular value of X. Then*

$$(i)\quad \|(I-P_{T_l})X\|_F \le \frac{\|X_l-X\|_F^2}{\sigma_{\min}(X)}, \qquad (ii)\quad \|P_{T_l}-P_T\| \le \frac{2\|X_l-X\|_F}{\sigma_{\min}(X)}.$$

Next we need to prove the following results about the error in the projections.

Lemma 2.2. *Let X and X_l be two rank r matrices. Denote p as the sampling rate and suppose T and T_l are the tangent spaces of the fixed rank r matrix manifold at X and X_l, respectively. Assume*

$$\|P_T - p^{-1}P_T P_\Omega P_T\| \le \varepsilon_0 \tag{2.7}$$

and

$$\frac{\|X_l-X\|_F}{\sigma_{\min}(X)} \le \frac{\varepsilon_0 p^{\frac{1}{2}}}{4(1+\varepsilon_0)} \tag{2.8}$$

are satisfied for some $0<\varepsilon_0<1$. Then

$$\|P_\Omega P_{T_l}\| \le (1+\varepsilon_0)p^{\frac{1}{2}} \quad \textit{and} \quad \|P_{T_l} - p^{-1}P_{T_l}P_\Omega P_{T_l}\| \le \frac{5\varepsilon_0}{2}. \tag{2.9}$$

Proof. For the first inequality in (2.9). It follows from (2.7) that

$$\|P_T P_\Omega P_T\| \le (1+\varepsilon_0)p.$$

Then for any matrix $Z\in\mathbb{R}^{n_1\times n_2}$, we have

$$\begin{aligned}\|P_\Omega P_T(Z)\|_F^2 &= \langle P_\Omega P_T(Z), P_\Omega P_T(Z)\rangle \\ &= \langle P_T(Z), P_T P_\Omega P_T(Z)\rangle \\ &\le (1+\varepsilon_0)p\|P_T(Z)\|_F^2.\end{aligned}$$

Thus $\|P_\Omega P_T\| \le \sqrt{(1+\varepsilon_0)p}$. Moreover,

$$\begin{aligned}\|P_\Omega P_{T_l}\| &\le \|P_\Omega(P_{T_l}-P_T)\| + \|P_\Omega P_T\| \le \frac{2\|X_l-X\|_F}{\sigma_{\min}(X)} + \|P_\Omega P_T\| \\ &\le \frac{\varepsilon_0 p^{\frac{1}{2}}}{2(1+\varepsilon_0)} + \sqrt{(1+\varepsilon_0)p} \le (1+\varepsilon_0)p^{\frac{1}{2}}.\end{aligned}$$

The second inequality can proved by (ii) of Lemma 2.1. Setting $t=\sqrt{(1+\varepsilon_0)}>1$, then $2t^4-2t^3-t^2+1=(t^2-1)(2t^2-t-1)>0$ is satisfied and the last inequality can be derived.

For the second inequality in (2.9). With the above tools in hand we have

$$\begin{aligned}\|P_{T_l}-p^{-1}P_{T_l}P_\Omega P_{T_l}\| \le & \|P_{T_l}-P_T\|+p^{-1}\|P_TP_\Omega P_T-P_{T_l}P_\Omega P_T\| \\ & +p^{-1}\|P_{T_l}P_\Omega P_{T_l}-P_{T_l}P_\Omega P_T\|+\|P_T-p^{-1}P_TP_\Omega P_T\| \\ \le & \|P_{T_l}-P_T\|+p^{-1}\|P_T-P_{T_l}\|\|P_\Omega P_T\| \\ & +p^{-1}\|P_{T_l}P_\Omega\|\|P_{T_l}-P_T\|+\|P_T-p^{-1}P_TP_\Omega P_T\| \\ \le & \frac{2\varepsilon_0 p^{\frac{1}{2}}}{4(1+\varepsilon_0)}+\frac{2\varepsilon_0}{4\sqrt{(1+\varepsilon_0)}}+\frac{\varepsilon_0}{2}+\varepsilon_0\le\frac{5\varepsilon_0}{2}.\end{aligned}$$

The proof is completed. □

Lemma 2.3 (Lemma 4.6 in [17]). *Assume the second inequality given in* (2.9) *is satisfied. Then the stepsize* α_l *in Algorithm* 2.1 *can be bounded as*

$$\frac{2}{(2+5\varepsilon_0)p}\le\alpha_l=\frac{\|P_{T_l}(G_l)\|_F^2}{\langle P_{T_l}(G_l),P_\Omega P_{T_l}(G_l)\rangle}\le\frac{2}{(2-5\varepsilon_0)p}.$$

Then we can prove the following theorem about errors of the iterates.

Theorem 2.1. *Suppose the inequality given in* (2.7) *and*

$$\frac{\|X_0-A\|_F}{\sigma_{\min}(A)}\le\frac{\varepsilon_0 p^{\frac{1}{2}}}{4(1+\varepsilon_0)} \tag{2.10}$$

are satisfied with ε_0 *being a positive numerical constant such that* $\beta_1=\frac{22\varepsilon_0}{2-5\varepsilon_0}<1$. *Then the iterates* X_l *generated by Algorithm* 2.1 *satisfy*

$$\|X_l-A\|_F\le\beta_1^l\|X_0-A\|_F,\quad l=1,2,\cdots. \tag{2.11}$$

Proof. We can prove (2.11) by induction. Suppose that in the l-th interaction X_l satisfies

$$\frac{\|X_l-A\|_F}{\sigma_{\min}(A)}\le\frac{p^{\frac{1}{2}}\varepsilon_0}{4(1+\varepsilon_0)}, \tag{2.12}$$

with $\sigma_{\min}(A)$ is the minimum nonzero singular of A. Recall that (2.7) is satisfied, then by Lemma 2.2, we have

$$\|P_\Omega P_{T_l}\|\le(1+\varepsilon_0)p^{\frac{1}{2}}, \tag{2.13a}$$

$$\|P_{T_l}-p^{-1}P_{T_l}P_\Omega P_{T_l}\|\le\frac{5\varepsilon_0}{2}. \tag{2.13b}$$

For the third step in Algorithm 2.1, denote $W_{l+1}=\pi_+(X_1+\alpha_l P_{T_l}(G_l))$, we have

$$\begin{aligned}\|X_{l+1}-A\|_F=&\|X_{l+1}-W_l+W_{l+1}-A\|_F\\ \leq&\|X_{l+1}-W_{l+1}\|_F+\|W_{l+1}-A\|_F\\ \leq&2\|W_{l+1}-A\|_F.\end{aligned}\tag{2.14}$$

The second inequality is derived by noting that X_{l+1} is the optimal rank r approximation of W_{l+1}. Noting that the underling matrix A is nonnegative, then after adding a nonnegative projection π_+ in the proposed Algorithm 2.1, the iterative result of each step will be closer to the underling matrix, i.e.,

$$\|W_{l+1}-A\|_F=\|\pi_+(W_l)-A\|_F\leq\|W_l-A\|_F.\tag{2.15}$$

The projection π_+ can help us to improve the effectiveness of the proposed algorithms which can be seen from the experiments results given in Section 4. Combing (2.14) and (2.15) and plugging $W_l=X_1+\alpha_l P_{T_l}(G_l)$ gives

$$\begin{aligned}\|X_{l+1}-A\|_F\leq&2\|X_l+\alpha_l P_{T_l}(G_l)-A\|_F\\ =&2\|X_l-A-\alpha_l P_{T_l}P_\Omega(X_l-A)\|_F\\ \leq&2\|(P_{T_l}-\alpha_l P_{T_l}P_\Omega P_{T_l})(X_l-A)\|_F+2\|(I-P_{T_l})(X_l-A)\|_F\\ &+2|\alpha_l|\|P_{T_l}P_\Omega(I-P_{T_l})(X_l-A)\|_F\\ =:&I_1+I_2+I_3.\end{aligned}\tag{2.16}$$

For I_1, applying Lemma 2.2 gives

$$I_1=2\|(P_{T_l}-\alpha_l P_{T_l}P_\Omega P_{T_l})(X_l-A)\|_F\leq\frac{20\varepsilon_0}{2-5\varepsilon_0}\|X_l-A\|_F.$$

For I_2, by (i) of Lemma 2.1 and the assumption given in (2.12), we have

$$\begin{aligned}I_2=&2\|(I-P_{T_l})(A)\|_F\leq\frac{2\|X_l-A\|_F^2}{\sigma_{\min}(X)}\\ \leq&\frac{p^{1/2}\varepsilon_0}{2(1+\varepsilon_0)}\|X_l-A\|_F\leq\frac{\varepsilon_0}{2-5\varepsilon_0}\|X_l-A\|_F.\end{aligned}$$

For I_3, by the bound of α_l given in Lemma 2.3, the bound of the spectral norm of $P_\Omega P_{T_l}$ given in Lemma 2.2 and the assumption given in (2.12), we have

$$\begin{aligned}I_3\leq&2|\alpha_l|\|P_\Omega P_{T_l}\|\|(I-P_{T_l})(A)\|_F\\ \leq&\frac{4}{(2-5\varepsilon_0)}(1+\varepsilon_0)p^{1/2}\frac{\|X_l-A\|_F^2}{\sigma_{\min}(A)}\\ \leq&\frac{\varepsilon_0}{2-5\varepsilon_0}\|X_l-A\|_F.\end{aligned}$$

Taking the bounds of I_1, I_2 and I_3 into (2.16) gives

$$\|X_{l+1}-A\|_F \le \beta_1 \|X_l-A\|_F,$$

where $\beta_1 = \frac{22\varepsilon_0}{2-5\varepsilon_0} < 1$. Note that (2.16) is satisfied for $l=0$ by the assumption of Theorem 2.1, then the sequence derived by Algorithm 2.1 is contractive. $\square$

In Algorithm 2.1, a SVD truncation is needed to project W_{l+1} back to the manifold $\mathcal{M}_r$. Such computational procedure can be expensive. Here we further modify the step by using the tangent space of the iterate to find an approximation of the $\pi_r(W_{l+1})$ on the manifold such that the computational cost can be reduced. The idea is shown as in Fig. 1 for illustration. Similar to the results in [17], the overall computational cost of $\pi_r(P_{T_l}(W_l'))$ in Algorithm 2.2 can be expressed as two matrix-matrix multiplications. In addition, the calculation procedure involves the QR decomposition of two matrices of sizes $n_1\times r$ and $n_2\times r$ matrices, and the SVD of a matrix of size $2r\times 2r$. The total cost per iteration is of $4n_1n_2r+O(n_1r^2+n_2r^2+r^3)$. In contrast, the computation of the best rank-r approximation of a non-structured $n_1\times n_2$ matrix costs $\mathcal{O}(n_1n_2r)+n_1n_2$ flops where the constant in front of n_1n_2r can be very large. In practice, the cost per iteration of the proposed Inexact Riemannian gradient decent method using tangent spaces (InRGD-TS) is less than that of original Inexact Riemannian gradient decent method (InRGD). In Section 4, numerical examples are given to demonstrate the total computational time of the proposed InRGD-TS method is less than that of the InRGD method. The resulting algorithm is listed in Algorithm 2.2. Here $\mathcal{M}_r$,X_0,T_l and P_{T_l} are defined as in Algorithm 2.1. $\mathcal{M}_n$ denotes the nonnegative matrices manifold. For Algorithm 2.2, we can show its convergence stated in Theorem 2.2.

Algorithm 2.2 Inexact Riemannian gradient decent method using tangent spaces.

Initilization: $X_0=\pi_r(P_\Omega(A))$,Ω is a set of cardinality m sampled uniformly at random.

for $l=0,1,\cdots,$ do

1: $G_l=P_\Omega(A-X_0)$;

2: $\alpha_l=\frac{\langle P_{T_l}(G_l),P_{T_l}(G_l)\rangle}{\langle P_{T_l}(G_l),P_\Omega P_{T_l}(G_l)\rangle}$;

3: $W_l=X_l+\alpha_l P_{T_l}(G_l)$;

4: $W_l'=\pi_+(W_l)$;

5: $Y_l=P_{T_l}(W_l')$;

6: $X_{l+1}=\pi_r(Y_l)$;

end for

Output: X_l when the stopping criterion is satisfied.

Theorem 2.2. *Let $\mathcal{M}_r$,A and T be given as in Theorem* 2.1*. Assume* (2.7) *and* (2.10) *are satisfied with ε_0 being a positive numerical constant such that $\beta_2=\frac{132\varepsilon_0}{2-5\varepsilon_0}<1$. Then the iterates X_l generated by Algorithm* 2.2 *satisfy*

$$\|X_l-A\|_F\le\beta_2^l\|X_0-A\|_F,\quad l=1,2,\cdots. \tag{2.17}$$

Proof. Similar to the proof of Theorem 2.1, we can prove (2.17) by induction. Suppose in the l-th interation X_l in Algorithm 2.2 satisfies (2.12). Then (2.13a) and (2.13b) can be derived. Denote $W_l=X_l+\alpha_l P_{T_l}(G_l)$, and $Y_l=P_{T_l}(\pi_+(W_l))$. Note that X_{l+1} is the optimal rank r approximation of Y_l, and Y_l is the closest point of $\pi_+(W_l)$ on the tangent space T_l, we have

$$\begin{aligned}
\|X_{l+1}-X\|_F&=\|X_{l+1}-Y_l+Y_l-A\|_F\\
&\le\|X_{l+1}-Y_l\|_F+\|Y_l-A\|_F\le 2\|Y_l-A\|_F\\
&=2\|P_{T_l}(\pi_+(W_l))-\pi_+(W_l)+\pi_+(W_l)-A\|_F\\
&\le 2\|P_{T_l}(\pi_+(W_l))-\pi_+(W_l)\|_F+2\|\pi_+(W_l)-A\|_F\\
&\le 2\|\pi_+(W_l)-W_l\|_F+2\|\pi_+(W_l)-A\|_F\\
&=2\|\pi_+(W_l)-A+A-X_l-\alpha_l P_{T_l}(G_l)\|_F+2\|\pi_+(W_l)-A\|_F\\
&\le 2\|\pi_+(W_l)-A\|_F+2\|X_l+\alpha_l P_{T_l}(G_l)-A\|_F+2\|\pi_+(W_l)-A\|_F\\
&\le 6\|X_l+\alpha_l P_{T_l}(G_l)-A\|_F.
\end{aligned}$$

Analogous to (2.15),

$$\|\pi_+(W_l)-A\|_F\le\|W_l-A\|_F=\|X_l+\alpha_l P_{T_l}(G_l)-A\|_F$$

can be derived and the last inequality follows. After choosing some suitable ε_0 such that $\beta_2=\frac{132\varepsilon_0}{2-5\varepsilon_0}<1$, and by the proof of Theorem 2.1 we have (2.17) is satisfied for $l+1$. Then, by the assumption of Theorem 2.2, (2.17) is satisfied when $l=0$. Combine them together, the iterates X_l generated by Algorithm 2.2 is convergent. $\square$

Remark 2.1. Besides Inexact Riemannian gradient descent methods, we can employ Inexact Riemannian conjugate gradient methods to solve the nonnegative matrix completion problem. In the Inexact Riemannian conjugate gradient method, the search direction is a linear combination of the projected gradient descent direction and the past search direction projected onto the tangent space of the current iterate. Similar to Theorems 2.1 and 2.2, the underlying nonnegative matrix can be exactly recovered by the Inexact Riemannian conjugate gradient method.

3 The initialization and sampling complexity

In this section, we mainly study the number of observed entries required to exactly recover the underling nonnegative low rank matrix. In this case, we need to introduce the following definitions firstly.

Definition 3.1 (Definition 1.2 in [1]). *Let $X\in\mathbb{R}^{n_1\times n_2}$ be a rank r matrix with the skinny singular value decomposition (SVD) as $X=USV^T$. We assume X is μ_0-incoherent, that is, there exists an absolute numerical constant $\mu_0>0$ such that*

$$\|P_U(e_i)\|\leq\sqrt{\frac{\mu_0 r}{n_1}}\quad \textit{and}\quad \|P_V(e_j)\|\leq\sqrt{\frac{\mu_0 r}{n_2}}$$

for $1\leq i\leq n_1$, $1\leq j\leq n_2$. Here e_l $(l=i,j)$ is the l-th canonical basis of $\mathbb{R}^{n_1}$ and $\mathbb{R}^{n_2}$, P_U and P_V are the orthogonal projections onto the column and row spaces of X, respectively.

The two conditions given in (2.7) and (2.10) in Theorem 2.1 are used to guarantee the convergence of Algorithm 2.1. For (2.7), it is a local restricted isometry property which is saying that the operator $P_TP_\Omega P_T$ is close to an isometry on T if the number of the observed entries is big enough. Under the framework of Bernoulli sampling model, Candes and Retha [1] demonstrated that (2.7) plays a key role in nuclear norm minimization for matrix completion problem. In the following discussion, we set $s=\max\{n_1,n_2\}$, $t=\min\{n_1,n_2\}$.

Lemma 3.1 (Theorem 4.1 in [1]). *Let $X\in\mathbb{R}^{n_1\times n_2}$ be a μ_0-incoherent matrix with rank r. Suppose Ω with $|\Omega|=m$ is sampled according to the Bernoulli model. Then for all $\beta>1$*

$$\left\|\frac{n_1n_2}{m}P_TP_\Omega P_T-P_T\right\|_{op}\lesssim\sqrt{\frac{\mu_0 rs\beta\log s}{m}},$$

holds with probability at least $1-3s^{-\beta}$ provided that $m\gtrsim\mu_0 r\beta s\log s$.

It follows from Lemma 3.1 that (2.7) is satisfied with probability at least $1-3s^{-\beta}$, as long as $m\gtrsim\mu_0 r\beta s\log s$, where μ_0 is the incoherence condition constant stated in Definition 3.1.

For (2.10), it is required to determine an initial guess that falls within a certain small area of the underlying nonnegative matrix. If it is valid, the sequence generated by Algorithm 2.1 can guarantee to converge linearly to the underling nonnegaive low rank matrix. Here we adapt the trimming scheme proposed in [6,17] to construct an initial guess. More precisely, the scheme is implemented by dividing the sampling set Ω into $L+1$ parts, such that $\Omega=\bigcup_{i=0}^{L}\Omega_i$ and the initialization matrix was constructed by values from the $L+1$ subsets of Ω independently. In our setting, there is a nonnegative projection as given in (2.4) to make sure the elements of the trimming results are nonnegative. The scheme is given in Algorithm 3.1.

Algorithm 3.1 Initialization via resampled method and trimming method.

Partition Ω into $L+1$ equal groups, i.e., $\Omega=\bigcup_{i=0}^{L}\Omega_i$, $|\Omega_0|=\cdots=|\Omega_L|=\frac{m}{L+1}=\hat{m}$.
Set $Z_0=\pi_r(\frac{n_1n_2}{\hat{m}}P_{\Omega_0}(A))$
for $l=0,\cdots,L-1$ do
1:$Z_l=U_l\Sigma_rV_l^T$;
2: $\hat{Z}_l=A_l^{(i)}\Sigma_r(B_l^{(i)})^T$, where

$$A_l^{(i)}=\frac{U_l^{(i)}}{\|U_l^{(i)}\|}\min\Big\{\|U_l^{(i)}\|,\sqrt{\frac{\mu_0 r}{n_1}}\Big\},\quad B_l^{(i)}=\frac{V_l^{(i)}}{\|V_l^{(i)}\|}\min\Big\{\|V_l^{(i)}\|,\sqrt{\frac{\mu_0 r}{n_2}}\Big\};$$

3: $Z'_{l+1}=\pi_r(\hat{Z}_l+\frac{n_1n_2}{\hat{m}}P_{\hat{T}_l}P_{\Omega_{l+1}}(X-\hat{Z}_l))$;
4: $Z_{l+1}=\pi_+(Z'_{l+1})$;
end for
Output: $X_0=Z_L$

Next we show the output of Algorithm 3.1 falls within the neighborhood required by (2.10). Let us first state the following lemma.

Lemma 3.2 (Theorem 6.3 in [1])**.** *Suppose $X\in\mathbb{R}^{n_1\times n_2}$, and Ω with $|\Omega|=m$ is a set of indices sampled according to the Bernoulli model. Then for all $\beta>2$*

$$\Big\|\Big(I-\frac{n_1n_2}{m}P_\Omega\Big)X\Big\|\lesssim\sqrt{\frac{s^2t\beta\log s}{m}}\|X\|_\infty,$$

holds with probability at least $1-s^{-\beta}$ provided $m\gtrsim\beta s\log s$.

By Lemma 3.2 and the incoherence conditions listed in Definition 3.1, the distance between the initiation value Z_0 in Algorithm 3.1 and the underlying matrix X can be estimated as follows.

Lemma 3.3. *Suppose $X\in\mathbb{R}^{n_1\times n_2}$ satisfies the incoherence conditions given in Definition 3.1, Ω with $|\Omega|=m$ is a set of indices sampled according to the Bernoulli model. Let Ω_i, $i=0,\cdots,L$ be a division of Ω given in Algorithm 3.1 and $Z_0=\pi_r(\frac{n_1n_2}{\hat{m}}P_{\Omega_0}(X))$. Then for all $\beta>2$,*

$$\|Z_0-X\|_F\le\frac{\sigma_{\min}(X)}{256\kappa^2} \tag{3.1}$$

holds with probability at least $1-s^{-\beta}$ *provided that*

$$\hat{m} \gtrsim \mu_0^2 \beta \kappa^6 r^2 s \log s.$$

Proof. Set $W_0 = \frac{n_1 n_2}{\hat{m}} P_{\Omega_0}(X)$. First,

$$\|Z_0 - X\| \le \|Z_0 - W_0\| + \|W_0 - X\| \le 2\|W_0 - X\| \le 2\sqrt{\frac{s^2 t \beta \log s}{\hat{m}}} \|X\|_\infty,$$

where the third inequality can be derived by Lemma 3.2 which holds with probability at least $1-s^{-\beta}$. And then

$$\begin{aligned} \|Z_0 - X\|_F &\le \sqrt{2r}\|Z_0 - X\| \le \sqrt{\frac{8 s^2 t r \log s}{\hat{m}}} \|X\|_\infty \\ &\le \sqrt{\frac{8\mu_0^2 s r^2 \log s}{\hat{m}}} \|X\| \le \frac{\sigma_{\min}(X)}{256\kappa^2}. \end{aligned}$$

The third inequality is followed from the fact that

$$\|X\|_\infty = \|U \Sigma V^T\|_\infty \le \frac{\mu_0 r}{t} \sigma_{\max},$$

the fourth inequality is derived by $\hat{m} \gtrsim \mu_0^2 \beta \kappa^6 r^2 s \log s$. □

Lemma 3.4. *Suppose* $X \in \mathbb{R}^{n_1 \times n_2}$ *with* $\operatorname{rank}(X) = r$, κ *is the condition number of* X *and* L *is defined as in Algorithm* 3.1. *Then for all* $\beta > 1$, *the output of Algorithm* 3.1 *satisfies*

$$\|X_0 - X\|_F \le \left(\frac{5}{6}\right)^L \frac{\sigma_{\min}(X)}{256\kappa^2}$$

with high probability provided

$$\hat{m} \gtrsim \mu_0^2 r^2 \kappa^6 s \beta \log s.$$

Proof. For $l=0$, $Z_0 = \pi_r(\frac{n_1 n_2}{\hat{m}} P_{\Omega_0}(X))$, then by Lemma 3.3 we have

$$\|Z_0 - X\|_F \le \frac{\sigma_{\min}(X)}{256\kappa^2}$$

satisfied with high probability provided

$$\hat{m} \gtrsim \mu_0^2 \beta \kappa^6 r^2 s \log s. \tag{3.2}$$

Assume that

$$\|Z_l - X\|_F \leq \left(\frac{5}{6}\right)^l \frac{\sigma_{\min}(X)}{256\kappa^2}. \tag{3.3}$$

Then by Lemma 13 in [17], we have $\hat{Z}_l$ is an incoherent matrix with incoherence parameter $\frac{100}{81}\mu_0$ and

$$\|\hat{Z}_l - X\|_F \leq 8\kappa\|Z_l - X\|_F.$$

Denote

$$W_l = \hat{Z}_l + \frac{n_1 n_2}{\hat{m}} P_{\hat{T}_l} P_{\Omega_{l+1}}(X - \hat{Z}_l) \quad \text{and} \quad Z'_{l+1} = \pi_1(W_l).$$

Note that X is nonnegative and Z'_{l+1} is the optimal rank r approximation of W_l, then the approximation error at the $(l+1)$th iteration can be decomposed as

$$\begin{aligned}
\|Z_{l+1} - X\|_F &= \|\pi_+(Z'_{l+1}) - X\|_F \leq \|Z'_{l+1} - X\|_F \\
&= \|Z'_{l+1} - W_l + W_l - X\|_F \\
&\leq 2\left\|\hat{Z}_l + \frac{n_1 n_2}{\hat{m}} P_{\hat{T}_l} P_{\Omega_{l+1}}(X - \hat{Z}_l) - X\right\|_F \\
&\leq 2\left\|\left(P_{\hat{T}_l} - \frac{n_1 n_2}{\hat{m}} P_{\hat{T}_l} P_{\Omega_{l+1}} P_{\hat{T}_l}\right)(\hat{Z}_l - X)\right\|_F + 2\|(I - P_{\hat{T}_l})(\hat{Z}_l - X)\|_F \\
&\quad\ + 2\left\|\frac{n_1 n_2}{\hat{m}} P_{\hat{T}_l} P_{\Omega_{l+1}}(I - P_{\hat{T}_l})(\hat{Z}_l - X)\right\|_F \\
&:= I_5 + I_6 + I_7.
\end{aligned}$$

Applying Lemma 3.1 to

$$\left\|P_{\hat{T}_l} - \frac{n_1 n_2}{\hat{m}} P_{\hat{T}_l} P_{\Omega_{l+1}} P_{\hat{T}_l}\right\|$$

in I_5 gives

$$I_5 \leq \kappa\sqrt{\frac{100\mu_0\beta r s \log s}{81\hat{m}}}\|Z_l - X\|_F$$

holds with high probability.

By applying (i) of Lemma 2.1 and recall the assumption (3.3), we have

$$I_6 \leq \frac{2\|\hat{Z}_l - X\|_F^2}{\sigma_{\min}(X)} \leq \frac{128\kappa^2\|Z_l - X\|_F^2}{\sigma_{\min}(X)} \leq \frac{1}{2}\|Z_l - X\|_F.$$

Note that $\hat{Z}_l$ is independent of Ω_{l+1} with the incoherence parameter $\frac{100}{81}\mu_0$, then it follows from Lemma 6 in [17] that

$$\left\|\frac{n_1 n_2}{\hat{m}} P_{T_l} P_{\Omega_{l+1}}(P_U - P_{U_l}) - P_{T_l}(P_U - P_{U_l})\right\| \leq \sqrt{\frac{4800\mu_0 s\beta r \log s}{81\hat{m}}}$$

holds with high probability. Moreover, due to $X=UU^TX$and $P_{\hat{T}_l}(\hat{Z}_l)=\hat{Z}_l$, we have

$$\begin{aligned}(I-P_{\hat{T}_l})(\hat{Z}_l-X)=&-(I-P_{\hat{T}_l})(X)\\=&-UU^TX+\hat{U}_l\hat{U}_l^TX+UU^TX\hat{V}_l\hat{V}_l^T-\hat{U}_l\hat{U}_l^TX\hat{V}_l\hat{V}_l^T\\=&-(UU^T-\hat{U}_l\hat{U}_l^T)X(I-\hat{V}_l\hat{V}_l^T)\\=&(P_U-P_{\hat{U}_l})(\hat{Z}_l-X)(I-P_{\hat{V}_l}).\end{aligned}$$

Together with

$$P_{\hat{T}_l}((P_U-P_{\hat{U}_l})(\hat{Z}_l-X)(I-P_{\hat{V}_l}))=P_{\hat{T}_l}((I-P_{\hat{T}_l})(\hat{Z}_l-X))=0,$$

I_7 can be bounded as follows,

$$\begin{aligned}I_7=&2\left\|\frac{n_1n_2}{\hat{m}}P_{\hat{T}_l}P_{\Omega_{l+1}}(I-P_{\hat{T}_l})(\hat{Z}_l-X)\right\|_F\\=&2\left\|\frac{n_1n_2}{\hat{m}}P_{\hat{T}_l}P_{\Omega_{l+1}}(I-P_{\hat{T}_l})(\hat{Z}_l-X)-P_{\hat{T}_l}(I-P_{\hat{T}_l})(\hat{Z}_l-X)\right\|_F\\\le&2\left\|\frac{n_1n_2}{\hat{m}}(P_{\hat{T}_l}P_{\Omega_{l+1}}-P_{\hat{T}})(P_U-P_{\hat{U}_l})\right\|\|\hat{Z}_l-X\|_F\\\le&2\sqrt{\frac{4800\mu_0\beta sr\log s}{81\hat{m}}}\|\hat{Z}_l-X\|_F\\\le&16\kappa\sqrt{\frac{4800\mu_0\beta sr\log s}{81\hat{m}}}\|Z_l-X\|_F.\end{aligned}$$

Combining the bounds of I_5, I_6 and I_7 together, we can get

$$\|Z_{l+1}-X\|_F\le\left(\frac{1}{2}+182\kappa\sqrt{\frac{\mu_0\beta sr\log s}{\hat{m}}}\right)\|Z_l-X\|_F\le\frac{5}{6}\|Z_l-X\|_F$$

holds with high probability provided

$$\hat{m}\gtrsim\mu_0\beta\kappa^2sr\log s. \tag{3.4}$$

Therefore taking a maximum of the right hand sides of (3.2) and (3.4) gives

$$\|Z_L-X\|_F\le\left(\frac{5}{6}\right)^L\frac{\sigma_{\min}(X)}{256\kappa^2}$$

with high probability provided $\hat{m}\gtrsim\mu_0^2\beta\kappa^6sr\log s$. □

Combining the above results, we can get the following results.

Theorem 3.1. *Suppose* $X\in\mathbb{R}^{n_1\times n_2}$ *is nonnegative with* $\mathrm{rank}(X)=r$, κ *is the condition number of* X*. Let* Ω ($|\Omega|=m$) *be a set of indices sampled according to the Bernoulli model. Let* X_0 *be the output of Algorithm* 3.1*. Then the iterates generated by Algorithm* 2.1 *converge to* X *with high probability provided*

$$m\gtrsim\frac{\mu_0^2}{\epsilon_0^2}\kappa^6\beta sr^2\log s\log\Big(\frac{s\log s}{24\epsilon_0}\Big).$$

Proof. This result follows from Lemma 2.2, Theorem 2.1, and Lemma 3.4. $\square$

4 Experimental results

In this section, numerical results are presented to show the effectiveness of the proposed inexact Riemannian gradient descent method (InRGD) and its version using tangent spaces (InRGD-TS) for nonnegative low rank matrix competition. We also make use of our results to derive the Inexact Riemannian conjugate gradient method without using tangent spaces (InRCG) and using tangent spaces (InRCG-TS) for comparison. On the other hand, we would like to compare low rank matrix completion methods without using nonnegativity. Both Riemannian gradient descent (RGD) and Riemannian Conjugate Gradient method (RCG) (see for example [17]) are employed in the comparison. All the experiments are performed under Windows 7 and MATLAB R2018a running on a desktop (Intel Core i7, @3.40GHz, 8.00G RAM).

The relative error (RES) is defined by

$$\mathrm{RES}=\frac{\|X-A\|_F}{\|A\|_F},$$

where X is the recovered solution and A is the ground-truth nonnegative matrix. Moreover, in order to evaluate the performance for real-world nonnegative matrices, the peak signal-to-noise ratio (PSNR) is used to measure the equality of the estimated nonnegative matrices, which is defined as:

$$\mathrm{PSNR}=10\log_{10}\frac{n_1n_2(X_{\max}-X_{\min})^2}{\|X-A\|_F},$$

where $X_{\max}$ and $X_{\min}$ are maximal and minimal entries of A, respectively. The stopping criterion of the algorithms are all set to

$$\frac{\|X_{l+1}-X_l\|_F}{\|X_l\|_F}\leq 10^{-5}.$$

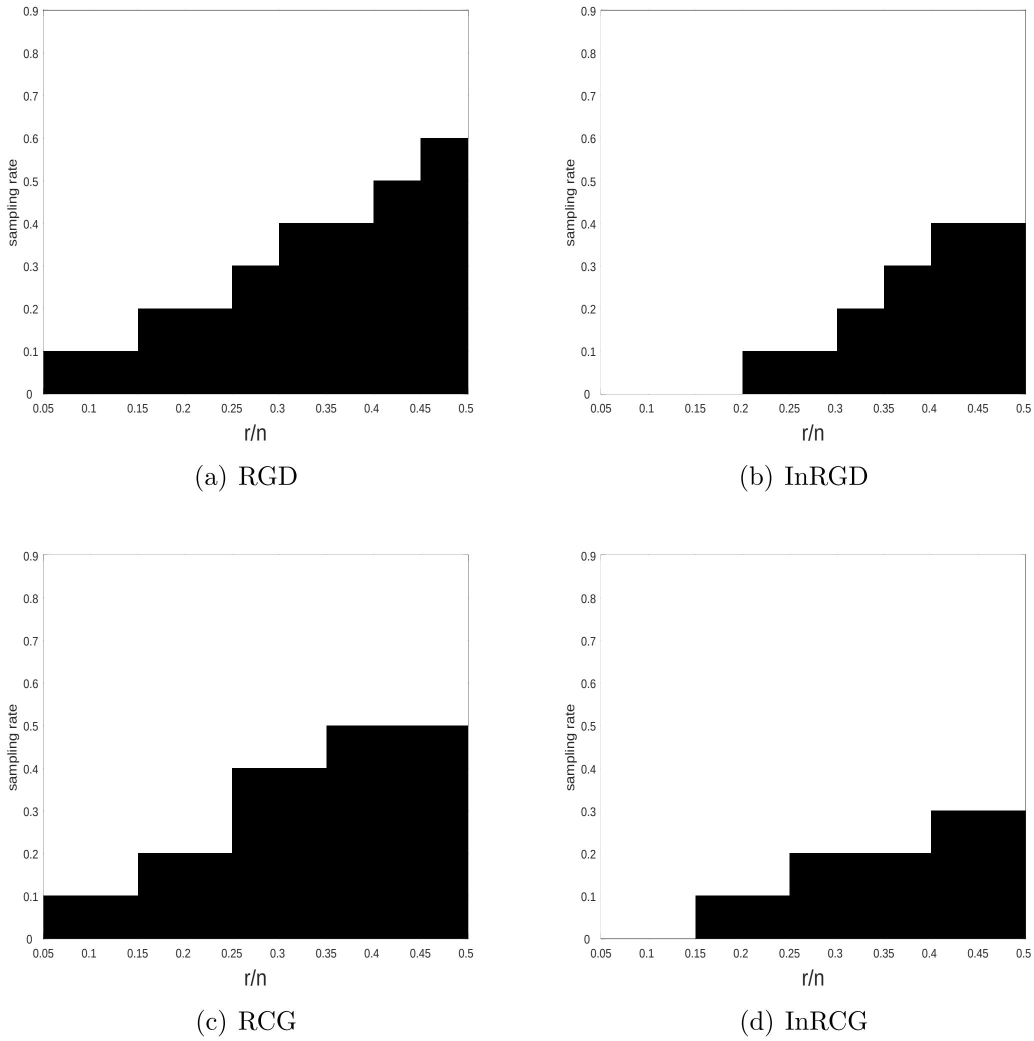
4.1 Synthetic data

We perform the proposed InRGD and InRGD-TS methods, InRCG and InRCG-TS methods, RGD and RCG methods on synthetic nonnegative low-rank matrix data. We randomly generate n_1-by-r matrix B and r-by-n_2 matrix C with entries uniformly distributed in the interval [0,1]. A is generated by normalizing BC to ensure that each element belongs to [0,1]. Notice that we have $\text{rank}(A_0)=r$. We set $n_1=500, n_2=800$ and $n_1=1000, n_2=800$, and choose $r=40$. We set the sampling rate (sr) from 0.1 to 0.9 with step size 0.1. Table 1 reports the average results over 10 tests of CPU Time (CPU) and the residual (RSE) $\frac{\|A-Y_c\|_F}{\|A\|_F}$, where Y_c is computed matrix. According to the table, when sr is in between 0.1 and 0.5, the RSEs of the proposed methods InRGD and InRGD-TS are smaller than those of RGD and RCG. The nonnegativity constraint should be useful in the matrix recovery. Also the computational times required by InRGD and InRCG are less than that required by RGD and RCG. Because the cost of SVD decomposition is avoided, the computational times required by InRGD-TS and InRCG-TS are always less than that required by InRGD and InRCG. Moreover, the performance of InRCG and InRCG-TS is comparable with InRGD and InRGD-TS, respectively.

Table 1: Average results of 10 tests related to the Cputimes and RES of the recovered matrices by RGD, InRGD, InRGD-TS, RCG, InRCG and InRCG-TS on synthetic data.

$n_1=500$, $n_2=800$, $r=40$												
sr	RGD		InRGD		InRGD-TS		RCG		InRCG		InRCG-TS	
	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE
0.1	700.7	0.593	709.7	0.380	175.8	0.380	242.6	0.583	445.5	0.294	173.6	0.294
0.2	595.5	0.197	701.6	0.105	179.2	0.105	887.9	2.994	905.3	0.021	365.3	0.021
0.3	683.6	0.048	488.4	9.17e-09	128.9	9.17e-09	1019.0	0.911	891.3	0.016	369.2	0.016
0.4	703.7	0.026	398.4	3.35e-09	102.3	3.30e-09	794.0	0.653	114.6	1e-09	46.9	8.99e-09
0.5	518.0	1.96e-09	234.9	1.90e-09	62.1	1.73e-09	162.1	3.85e-10	113.9	8.63e-10	43.8	4.53e-10
0.6	59.0	1.22e-09	52.3	8.82e-10	13.5	8.30e-10	46.5	3.06e-10	40.6	2.21e-10	17.0	2.21e-10
0.7	33.5	6.65e-10	32.6	5.14e-10	9.0	5.14e-10	32.1	1.72e-10	32.7	2.25e-10	14.0	2.17e-10
0.8	43.7	5.97e-10	42.1	4.67e-10	11.8	4.44e-10	38.3	1.77e-10	38.5	8.38e-11	16.8	3.36e-10
0.9	26.1	2.36e-10	25.0	2.82e-10	8.7	2.83e-10	27.3	9.70e-11	29.0	4.35e-11	12.9	4.58e-11
$n_1=1000$, $n_2=800$, $r=40$												
sr	RGD		InRGD		InRGD-TS		RCG		InRCG		InRCG-TS	
	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE
0.1	1508	0.332	1534	0.224	415.5	0.224	2057.0	1.57	2086.1	0.2047	997.9	0.2047
0.2	1425	0.058	1153	7.57e-09	333.6	7.31e-09	1974.8	1.79	558.9	1.21e-09	253.1	1.21e-09
0.3	830.8	2.30e-09	298.3	2.28e-09	85.2	2.49e-09	611.7	6.06e-10	115.1	7.92e-10	54.9	7.92e-10
0.4	219.8	1.13e-09	185.7	1.19e-09	54.8	1.43e-09	158.8	4.87e-10	96.1	5.65e-10	45.2	5.65e-10
0.5	77.6	7.22e-10	71.4	1.06e-09	22.1	6.34e-10	79.6	1.74e-10	67.3	2.54e-10	31.7	2.54e-10
0.6	49.8	6.91e-10	49.8	6.91e-10	16.2	6.91e-10	53.3	1.08e-10	55.1	1.74e-10	26.9	1.74e-10
0.7	41.4	5.24e-10	41.1	4.49e-10	14.1	4.50e-10	47.1	1.54e-10	49.1	1.91e-10	24.5	1.91e-10
0.8	33.9	3.74e-10	33.8	3.74e-10	12.2	3.74e-10	38.8	1.67e-10	43.0	1.79e-10	22.3	1.79e-10
0.9	43.6	4.91e-11	43.9	4.91e-11	15.4	4.91e-11	43.6	4.32e-11	47.8	2.67e-11	23.9	2.67e-11

Next we check the recovery ability of our algorithms as a function of rank(X) and the proportion ρ of observed nonnegative entries. The data sets are generated randomly similar to Table 1. We fix the matrix size to be $n=n_1=n_2=400$, and test different values of ρ and different values of rank$(X)/n$. For each pair (rank$(X)/n$,ρ), we simulate ten trials and declare a trial to be successful if the recovered matrix X satisfies $\frac{\|X-A\|_F}{\|A\|_F}\leq 10^{-5}$. Fig. 2 reports the recovery results of different methods (InRGD, InRCG, RGD and RCG). In the figure, a black pixel refers to failure case and a white pixel refers to a success case. It is clear from the results that



(a) RGD (b) InRGD

(c) RCG (d) InRCG

Figure 2: Recovery for varying matrix ranks and sampling numbers under the same matrix size $n=400$.

Table 2: Average results of 10 tests relate to the CPU and RES of the recovered matrices by RGD, InRGD, InRGD-TS, RCG, InRCG and InRCG-TS on synthetic data with Gaussian noise.

$n_1=400$, $n_2=400$, $r=20$, $\sigma=0.1$

sr	RGD		InRGD		InRGD-TS		RCG		InRCG		InRCG-TS	
	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE
0.1	1.56e+02	6.70e-01	1.78e+02	4.03e-01	6.08e+01	4.01e-01	2.17e+02	1.53e+00	2.37e+02	4.10e-01	1.21e+02	4.07e-01
0.2	1.59e+02	3.42e-01	1.82e+02	2.89e-01	6.31e+01	2.87e-01	2.19+02	4.66e-01	2.50e+02	2.90e-01	1.26e+02	2.91e-01
0.3	1.60e+02	2.15e-01	1.84e+02	2.09e-01	6.65e+01	2.09e-01	2.22e+02	2.16e-01	2.49e+02	2.08e-01	1.28e+02	2.08e-01
0.4	1.65e+02	1.63e-01	1.85e+02	1.61e-01	6.89e+01	1.63e-01	2.27e+02	1.63e-01	2.49e+02	1.63e-01	1.30e+02	1.63e-01
0.5	1.67e+02	1.32e-01	1.66e+02	1.32e-01	7.09e+01	1.32e-01	2.26e+02	1.32e-01	2.29e+02	1.32e-01	1.33e+02	1.32e-01
0.6	1.70e+02	1.11e-01	1.71e+02	1.11e-01	7.42e+01	1.11e-01	2.29e+02	1.11e-01	2.32e+02	1.11e-01	1.35e+02	1.11e-01
0.7	1.71e+02	9.69e-02	1.72e+02	9.69e-02	7.39e+01	9.69e-02	2.34e+02	9.69e-02	2.35e+02	9.70e-02	1.36e+02	9.70e-02
0.8	1.04e+02	1.07e+02	1.07e+02	8.72e-02	4.62e+01	8.72e-02	1.40e+02	8.72e-02	1.37e+02	8.72e-02	8.22e+01	8.72e-02
0.9	1.85e+02	1.86e+02	1.86e+02	7.89e-02	8.64e+01	7.98e-02	2.44e+02	7.98e-02	2.44e+02	7.98e-02	1.49e+02	7.98e-02

$n_1=400$, $n_2=400$, $r=20$, $\sigma=0.01$

sr	RGD		InRGD		InRGD-TS		RCG		InRCG		InRCG-TS	
	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE
0.1	1.53e+02	4.40e-01	1.58e+02	3.00e-01	5.97e+01	3.01e-01	2.18e+02	1.02e+00	2.75e+2	2.73e-01	1.50e+02	2.75e-01
0.2	1.83e+02	8.51e-02	1.55e+02	1.87e-02	6.75e+01	1.87e-02	2.76e+02	2.17e+00	6.32e+01	1.87e-01	3.48e+01	1.87e-02
0.3	1.11e+02	1.27e-02	3.27e+01	1.27e-02	1.31e+01	1.27e-02	2.70e+02	4.76e-01	2.00e+01	1.27e-02	1.08e+01	1.27e-02
0.4	1.95e+01	1.04e-02	1.65e+01	1.04e-02	6.88e+00	1.04e-02	1.53e+01	1.04e-02	1.03e+01	1.04e-02	5.70e+00	1.04e-02
0.5	2.69e+01	8.96e-03	1.62e+01	8.96e-03	6.64e+00	8.96e-03	2.47e+01	8.96e-03	1.16+01	8.96e-03	6.44e+00	8.96e-03
0.6	7.64e+00	7.99e-03	7.14e+00	7.99e-03	3.28e+00	7.99e-03	8.14e+00	7.99e-03	7.76e+00	7.99e-03	4.23e+00	7.99e-03
0.7	5.92e+00	7.29e-03	6.02e+00	7.29e-03	2.61e+00	7.29e-03	6.92e+00	7.29e-03	7.63e+00	7.29e-03	4.53e+00	7.29e-03
0.8	4.63e+00	6.78e-03	4.86e+00	6.78e-03	2.06e+00	6.78e-03	5.59e+00	6.78e-03	5.75e+00	6.78e-03	3.52e+00	6.78e-03
0.9	4.13e+00	6.35e-03	3.83e+00	6.35e-03	1.91e+00	6.35e-03	4.56e+00	6.35e-03	5.39e+00	6.35e-03	3.02e+00	6.35e-03

$n_1=400$, $n_2=400$, $r=20$, $\sigma=0.001$

sr	RGD		InRGD		InRGD-TS		RCG		In-RCG		InRCG-TS	
	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE	CPU	RSE
0.1	1.72e+02	4.50e-01	1.81e+02	3.03e-01	7.06e+01	3.02e-01	2.69e+02	1.12e+00	2.67e+02	2.67e-01	1.47e+02	2.73e-01
0.2	1.81e+02	8.87e-02	1.18e+02	1.79e-03	4.84e+01	1.79e-03	2.81e+02	2.42e+00	2.72e+02	2.52e-02	1.47e+02	2.52e-02
0.3	1.48e+02	1.26e-03	3.06e+01	1.26e-03	1.25e+01	1.26e-03	2.76e+02	3.21e-01	2.73e+01	1.26e-03	1.43e+01	1.26e-02
0.4	5.08e+01	1.02e-03	2.56e+01	1.02e-03	1.01e+01	1.02e-03	4.62e+01	1.02e-03	1.49e+01	1.02e-3	7.94e+00	1.02e-02
0.5	1.51e+01	8.09e-04	1.28e+01	8.90e-04	5.11e+00	8.90e-04	1.44e+01	8.90e-04	1.04e+01	8.90e-04	5.48e+00	8.90e-04
0.6	1.27e+01	7.92e-04	1.10e+01	7.92e-04	4.75e+00	7.92e-04	1.20e+01	7.92e-04	9.89e+00	7.92e-04	5.54e+00	7.92e-04
0.7	8.78e+00	9.69e-02	7.31e+00	7.31e-04	3.08e+00	7.31e-04	9.50e+00	7.31e-04	9.28e+00	7.31e-04	4.89e+00	7.31e-04
0.8	6.59e+00	8.72e-02	6.70e+00	6.76e-04	3.06e+00	6.76e-04	7.83e+00	6.76e-04	8.38e+00	6.76e-04	4.48e+00	6.76e-04
0.9	3.69e+00	7.89e-02	3.88e+00	6.35e-04	1.75e+00	6.35e-04	4.36e+00	6.35e-04	7.66e-01	6.35e-04	3.19e-01	6.35e-04

the nonnegativity projection used in InRGD and InRCG can help in the recovery underlying nonnegative low rank matrix.

Finally, we would like to show the performance of the proposed algorithm when a Gaussian noise of zero mean and variance σ (=0.1,0.01,0.001) is added to nonnegative low rank matrices. The residuals of the computed solutions by the proposed algorithms (InRGD and InRGD-TS) are reported in Table 2. In the table, the results by the other Riemannian algorithms (InRCG and InRCG-TS) and the Riemannian algorithm without using nonnegativity projection (RGD and RCG) are also reported. Similar to Table 1, it is clear that the performance of the proposed algorithms (InRGD and InRGD-TS) is better than that of RGD and RCG when sr is small.

4.2 Real image data

In this experiment, we present image (nonnegative pixels) completion results. The original two images "Barbara" and "Pepper" with sizes $n_1 \times n_2 = 256 \times 256$ are shown in Fig. 3. Let Ω be the set of observed entries that are generated randomly. $\rho = \frac{|\Omega|}{n_1 \times n_2}$ is the percentage of observed entries. Similar to the synthetic data case, we compare our proposed InRGD and InRCG methods with the RGD and RCG matrix completion methods. Tables 3 and 4 show that the InRGD and InRCG methods perform better that the RGD and RCG methods in the RES, PSNR and SSIM values with different sampling rates, different ranks. We need to remark that if RSE is greater than 1, i.e., the algorithm failed to recover the underlying image, then "-" is used in Tables 3 and 4 to indicate these situations. The original images and some recover results under different sampling rates and ranks are given in Fig. 3.

Figure 3: Recovered images by RGD, In-RGD, RCG, In-RCG algorithms with different sampling ratios and rank choices. The original images, the observed images, the recovered images by RGD, In-RGD, RCG and In-RCG are respectively listed from the first column to the fifth column. The corresponding sampling rates, the rank assumptions, psnr and ssim values are listed at the bottom of every images.

Table 3: The RES, PSNR and SSIM values of the recovered results by RGD, InRGD, RCG and InRCG on "Barbara".

r	sr	RGD			InRGD			RCG			InRCG		
		RES	PSNR	SSIM	RES	PSNR	SSIM	RES	PSNR	SSIM	RES	PSNR	SSIM
20	0.2	0.509	10.78	0.089	0.320	14.83	0.141	-	-	-	0.322	14.77	0.142
	0.3	0.210	18.47	0.321	0.178	19.90	0.340	0.466	11.55	0.295	0.172	20.22	0.347
	0.4	0.123	23.12	0.453	0.122	23.21	0.456	0.123	23.10	0.455	0.122	23.21	0.456
	0.5	0.106	24.40	0.503	0.106	24.42	0.503	0.106	24.40	0.503	0.106	24.42	0.503
	0.6	0.100	24.96	0.524	0.100	24.96	0.524	0.100	24.96	0.524	0.100	24.96	0.524
	0.7	0.096	25.32	0.547	0.096	25.32	0.547	0.096	25.32	0.547	0.096	25.32	0.547
	0.8	0.093	25.53	0.556	0.093	25.53	0.556	0.093	25.53	0.556	0.093	25.53	0.556
25	0.2	0.621	9.06	0.047	0.375	13.45	0.092	-	-	-	0.375	13.44	0.098
	0.3	0.296	15.50	0.262	0.201	18.84	0.312	-	-	-	0.206	18.65	0.304
	0.4	0.139	20.05	0.452	0.132	22.54	0.459	-	-	-	0.132	22.50	0.458
	0.5	0.100	24.91	0.532	0.101	24.86	0.531	0.100	24.91	0.531	0.101	24.86	0.531
	0.6	0.089	25.89	0.571	0.089	25.89	0.571	0.089	25.89	0.571	0.089	25.89	0.571
	0.7	0.084	26.39	0.593	0.084	26.39	0.593	0.084	26.39	0.593	0.084	26.39	0.593
	0.8	0.082	26.70	0.608	0.082	26.70	0.608	0.082	26.70	0.608	0.082	26.70	0.608
30	0.2	0.681	8.25	0.033	0.408	12.71	0.078	-	-	-	0.399	12.89	0.079
	0.3	0.390	13.10	0.186	0.257	16.72	0.233	-	-	-	0.254	16.82	0.238
	0.4	0.181	19.75	0.412	0.152	21.26	0.429	-	-	-	0.151	21.36	0.432
	0.5	0.120	23.31	0.631	0.108	24.24	0.536	0.115	23.73	0.529	0.108	24.22	0.534
	0.6	0.085	26.29	0.598	0.085	26.30	0.598	0.085	26.29	0.598	0.085	26.30	0.598
	0.7	0.078	27.11	0.626	0.078	27.12	0.626	0.078	27.11	0.626	0.078	27.12	0.626
	0.8	0.073	27.62	0.645	0.073	27.62	0.645	0.073	27.62	0.645	0.073	27.62	0.645
35	0.2	0.699	8.03	0.034	0.447	11.91	0.072	-	-	-	0.422	12.42	0.077
	0.3	0.496	11.01	0.115	0.296	15.49	0.179	-	-	-	0.296	15.50	0.180
	0.4	0.235	17.50	0.358	0.175	20.05	0.387	-	-	-	0.181	19.78	0.391
	0.5	0.140	22.00	0.520	0.111	24.04	0.537	0.162	20.74	0.511	0.114	23.82	0.534
	0.6	0.086	26.22	0.607	0.085	26.38	0.608	0.086	26.22	0.607	0.091	25.76	0.605
	0.7	0.073	27.62	0.649	0.073	27.63	0.649	0.073	27.62	0.649	0.073	27.63	0.649
	0.8	0.068	28.30	0.674	0.068	28.30	0.674	0.068	28.30	0.674	0.068	28.30	0.674
40	0.2	0.738	7.57	0.028	0.489	11.13	0.062	-	-	-	0.454	11.77	0.069
	0.3	0.590	9.50	0.077	0.344	14.19	0.135	-	-	-	0.335	14.41	0.142
	0.4	0.316	14.93	0.282	0.201	18.84	0.340	-	-	-	0.198	18.97	0.343
	0.5	0.163	20.69	0.504	0.134	22.38	0.508	0.279	16.01	0.476	0.132	22.51	0.505
	0.6	0.091	25.73	0.607	0.089	25.97	0.612	0.095	25.37	0.607	0.089	25.96	0.609
	0.7	0.073	27.68	0.666	0.073	27.70	0.666	0.073	27.68	0.666	0.073	27.7	0.666
	0.8	0.064	28.85	0.697	0.064	28.85	0.697	0.064	28.85	0.697	0.064	28.85	0.697

5 Conclusions

In this paper, Riemannian optimization methods are proposed to recover a nonnegative low rank matrix from its partial observed entries. With the underlying matrix incoherence conditions, we show that when the number m of observed entries are sampled independently and uniformly without replacement, the inexact Riemannian gradient descent method can recover the underlying n_1-by-n_2 nonnegative matrix of rank r provided that m is of $\mathcal{O}(r^2 s\log^2 s)$ with $s=\max\{n_1,n_2\}$. Numerical examples

Table 4: The RES, PSNR and SSIM values of the recovered results by RGD, InRGD, RCG and InRCG on "Pepper".

r	sr	RGD			InRGD			RCG			InRCG		
		RES	PSNR	SSIM	RES	PSNR	SSIM	RES	PSNR	SSIM	RES	PSNR	SSIM
20	0.2	0.600	10.26	0.079	0.332	15.39	0.148	-	-	-	0.384	14.12	0.094
	0.3	0.267	17.27	0.277	0.192	20.13	0.319	0.556	10.91	0.258	0.228	18.66	0.273
	0.4	0.134	23.24	0.419	0.134	23.25	0.419	0.134	23.28	0.418	0.137	26.10	0.425
	0.5	0.118	24.39	0.472	0.117	24.41	0.473	0.118	24.39	0.472	0.111	24.93	0.489
	0.6	0.111	24.92	0.491	0.111	24.92	0.491	0.111	24.92	0.491	0.100	25.82	0.531
	0.7	0.106	25.27	0.511	0.106	25.27	0.511	0.106	25.27	0.511	0.093	26.43	0.559
	0.8	0.104	25.49	0.528	0.104	25.49	0.528	0.104	25.47	0.528	0.900	26.74	0.573
25	0.2	0.636	9.74	0.052	0.378	14.26	0.094	-	-	-	0.385	14.12	0.094
	0.3	0.380	14.22	0.212	0.225	18.76	0.283	-	-	-	0.228	18.66	0.274
	0.4	0.161	21.67	0.406	0.138	23.01	0.418	0.171	21.17	0.402	0.137	23.10	0.425
	0.5	0.112	24.86	0.488	0.110	24.95	0.489	0.112	24.81	0.488	0.111	24.93	0.489
	0.6	0.101	25.75	0.529	0.100	25.83	0.532	0.101	25.74	0.529	0.100	25.83	0.531
	0.7	0.093	26.42	0.559	0.093	26.43	0.559	0.093	26.42	0.560	0.093	26.43	0.559
	0.8	0.090	26.74	0.573	0.090	26.74	0.573	0.089	26.74	0.573	0.089	26.75	0.573
30	0.2	0.654	9.51	0.038	0.404	13.69	0.080	-	-	-	0.405	13.66	0.077
	0.3	0.456	12.69	0.175	0.260	17.50	0.234	-	-	-	0.259	17.56	0.243
	0.4	0.222	18.89	0.367	0.164	21.49	0.400	0.675	9.23	0.330	0.158	21.87	0.405
	0.5	0.116	24.55	0.496	0.111	24.91	0.502	0.118	24.36	0.495	0.115	24.89	0.503
	0.6	0.092	26.54	0.558	0.091	26.60	0.559	0.092	26.54	0.558	0.091	26.60	0.559
	0.7	0.084	27.33	0.585	0.084	27.34	0.585	0.083	27.33	0.585	0.083	27.34	0.585
	0.8	0.079	27.82	0.607	0.079	27.82	0.607	0.077	27.82	0.607	0.079	27.82	0.607
35	0.2	0.705	8.85	0.034	0.437	13.02	0.079	-	-	-	0.424	13.26	0.079
	0.3	0.524	11.44	0.113	0.300	16.26	0.185	-	-	-	0.298	16.32	0.185
	0.3	0.281	16.82	0.318	0.187	20.39	0.369	-	-	-	0.196	19.99	0.363
	0.5	0.135	23.18	0.485	0.168	24.72	0.505	0.293	16.47	0.459	0.114	24.69	0.499
	0.6	0.091	26.65	0.569	0.101	26.82	0.572	0.092	26.58	0.570	0.089	26.86	0.572
	0.7	0.077	28.09	0.613	0.078	28.09	0.613	0.077	28.08	0.614	0.077	28.10	0.614
	0.8	0.071	28.79	0.636	0.071	28.79	0.636	0.071	28.79	0.636	0.071	28.79	0.636
40	0.2	0.736	8.48	0.027	0.481	12.16	0.061	-	-	-	0.456	12.64	0.065
	0.3	0.567	10.75	0.070	0.338	15.25	0.137	-	-	-	0.336	15.28	0.136
	0.4	0.392	13.96	0.250	0.223	18.84	0.323	-	-	-	0.221	18.92	0.320
	0.5	0.183	20.55	0.451	0.138	20.04	0.473	0.361	14.67	0.422	0.131	23.44	0.477
	0.6	0.093	26.47	0.576	0.138	27.08	0.583	0.101	25.68	0.572	0.088	26.90	0.584
	0.7	0.071	28.69	0.631	0.071	28.77	0.631	0.072	28.69	0.631	0.071	28.76	0.631
	0.8	0.065	29.57	0.663	0.065	29.59	0.682	0.065	29.58	0.663	0.065	29.58	0.663

are shown that the nonnegativity property would be useful in the matrix recovery. In particular, we demonstrate the number of samples required to recover the underlying low rank matrix with using the nonnegativity property is smaller than that without using the property. As a future research work, it would be interesting to show the convergence rate of the inexact Riemannian gradient descent with i.i.d. Gaussian noise.

Acknowledgements

G. Song was supported in part by the National Natural Science Foundation of China under Grant No. 12171369 and Key NSF of Shandong Province under Grant No. ZR2020KA008. M. Ng was supported in part by HKRGC GRF 12300519, 17201020 and 17300021, HKRGC CRF C1013-21GF and C7004-21GF, and Joint NSFC and RGC N-HKU769/21.

References

[1] E. J. Candès and B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math., 9 (2009), pp. 717–772.

[2] E. J. Candes and Y. Plan, Matrix completion with noise, Proc. IEEE, 98 (2010), pp. 925–936.

[3] E. J. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theor., 52 (2006), pp. 489–509.

[4] E. J. Candès and T. Tao, The power of convex relaxation: Near-optimal matrix completion, IEEE Trans. Infor. Theor., 56 (2010), pp. 2053–2080.

[5] E. J. Candès, X. Li, Y. Ma and Y, J. Wright, Robust principal component analysis?, J. ACM, 58(3) (2011), pp. 1–37.

[6] D. Gross, Recovering low-rank matrices from few coefficients in any basis, IEEE Trans. Infor. Theor., 57 (2011), pp. 1548–1566.

[7] B. Recht, A simpler approach to matrix completion, J. Machine Learn. Res., 12 (2011), pp. 3413–3430.

[8] B. Recht, M. Fazel, and P. A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev., 52 (2010), pp. 471–501.

[9] B. Vandereycken, Low-rank matrix completion by Riemannian optimization–extended version, Mathematics, 23 (2012), pp. 1214–1236.

[10] K. Wei, J. F. Cai, T. F. Chan and S. Leung, Guarantees of Riemannian optimization for low rank matrix recovery, SIAM J. Matrix Anal. Appl., 37(3) (2016), pp. 1198–1222.

[11] M. Fazel, Matrix Rank Minimization with Applications, Ph. D. dissertation, Stanford University, 2002.

[12] D. Gross, Recovering low-rank matrices from few coefficients in any basis, IEEE Trans. Infor. Theor., 57(3) (2011), pp. 1548–1566.

[13] H. Wang, J. F. Cai, T. Wang, and K. Wei, Fast cadzows algorithm and a gradient variant, J. Sci. Comput., 88(2) (2021), p. 41.

[14] J. F. Cai, T. Wang, and K. Wei Fast and provable algorithms for spectrally sparse signal reconstruction via low-rank hankel matrix completion. Appl. Comput. Harmonic Anal., 46(1) (2019), pp. 94–121.

[15] D. Goldfarb and S. Ma, Convergence of fixed-point continuation algorithms for matrix rank minimization, Found. Comput. Math., 11(2) (2011), pp. 183–210.
[16] J. F. Cai, E. J. Candes and Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim., 20(4) (2010), pp. 1956–1982.
[17] K. Wei, J. Cai, T. Chan and S. Leung, Guarantees of Riemanninan optimization for low rank matrix completion, Inverse Problems Imag., 14(2) (2020), pp. 233–265.
[18] H. Cai, J. Cai and K. Wei, Accelerated alternating projections for robust principal component analysis, J. Machine Learn. Res., 20 (2019), pp. 1–33.
[19] D. D. Lee and H. S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature, 401(6755) (1999), pp. 788–791.
[20] D. Guillamet and J. Vitria, Non-negative matrix factorization for face recognition, in Catalonian Conference on Artificial Intelligence, Springer, 2002, pp. 336–344.
[21] D. Guillamet, J. Vitria and B. Schiele, Introducing a weighted non-negative matrix factorization for image classification, Pattern Recog. Lett., 24(14) (2003), pp. 2447–2454.
[22] K. Chen, Matrix Preconditioning Techniques and Applications, volume 19. Cambridge University Press, 2005.
[23] Y. Wang, Y. Jia, C. Hu and M. Turk, Non-negative matrix factorization framework for face recognition, International Journal of Pattern Recognition and Artificial Intelligence, 19(04) (2005), pp. 495–511.
[24] D. Zhang, S. Chen and Z. Zhou, Two-dimensional non-negative matrix factorization for face representation and recognition, in International Workshop on Analysis and Modeling of Faces and Gestures, Springer, 2005, pp. 350–363.
[25] C. Ding, X. He and H. D. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, in Proceedings of the 2005 SIAM International Conference on Data Mining, SIAM, 2005, pp. 606–610.
[26] C. Ding, T. Li, W. Peng and H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006, pages 126–135.
[27] L. Jing, J. Yu, T. Zeng and Y. Zhu, Semi-supervised clustering via constrained symmetric non-negative matrix factorization, in International Conference on Brain Informatics, Springer, 2012, pp. 309–319.
[28] J. Liu, Z. Wu, Z. Wei, L. Xiao and L. Sun, A novel sparsity constrained nonnegative matrix factorization for hyperspectral unmixing, in 2012 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2012, pp. 1389–1392.
[29] M. Chen, W. Chen, B. Chen and B. Pan, Non-negative sparse representation based on block nmf for face recognition, in Chinese Conference on Biometric Recognition, Springer, 2013, pp. 26–33.
[30] Y. Liu, X. Pan, R. Shi, Y. Li, C. Wang and Z. Li, Predicting soil salt content over partially vegetated surfaces using non-negative matrix factorization, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 8(11) (2015), pp. 5305–5316.

[31] M. W. Berry and J. Kogan, Text Mining: Applications and Theory, John Wiley & Sons, 2010.

[32] T. Li and C. Ding, The relationships among various nonnegative matrix factorization methods for clustering, in Sixth International Conference on Data Mining (ICDM'06), IEEE, 2006, pp. 362–371.

[33] V. P. Pauca, F. Shahnaz, M. W. Berry and R. J. Plemmons, Text mining using non-negative matrix factorizations, in Proceedings of the 2004 SIAM International Conference on Data Mining, SIAM, 2004, pp. 452–456.

[34] W. Xu, X. Liu and Y. Gong, Document clustering based on non-negative matrix factorization, in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, 2003, pp. 267–273.

[35] P. M. Kim and B. Tidor, Subsystem identification through dimensionality reduction of large-scale gene expression data, Genome Res., 13(7) (2003), pp. 1706–1718.

[36] H. Kim and H. Park, Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis, Bioinformatics, 23(12) (2007), pp. 1495–1502.

[37] A. P. Montano, J. M. Carazo, K. K. D. Lehmann and R. D. P. Marqui, Nonsmooth nonnegative matrix factorization (nsNMF), IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(3) (2006), pp. 403–415.

[38] G. Wang, A. V. Kossenkov, and M. F. Ochs, LS-NMF: a modified non-negative matrix factorization algorithm utilizing uncertainty estimates, BMC Bioinfor., 7(1) (2006), p. 175.

[39] A. Cichocki, R. Zdunek, A. H. Phan and S. Amari, Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation, John Wiley & Sons, 2009.

[40] G. Golub and C. Van Loan, Matrix Computations, vol. 3, JHU Press, 2012.

[41] B. Vandereyckenl, Low-rank matrix completion by Rimannian optimization, SIAM J. Optim., 23(2013), pp. 1214–1236.

[42] Y. Xu, W. Yin, Z. Wen and Y. Zhang, An alternating direction algorithm for matrix completion with nonnegative factors, Front. Math. China, 7(2) (2012), pp. 365–384.

[43] Y. Xu and W. Yin, A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion, SIAM J. Imag. Sci., 6(3) (2013), pp. 1758–1789.