Riemannian Pursuit for Big Matrix Recovery

Mingkui Tan [1] TANMINGKUI@GMAIL.COM
Ivor W. Tsang [2] IVOR.TSANG@GMAIL.COM
Li Wang [3] LIW022@UCSD.EDU
Bart Vandereycken [4] BARTV@PRINCETON.EDU
Sinno Jialin Pan [5] JSPAN@I2R.A-STAR.EDU.SG

[1] School of Computer Science, The University of Adelaide, Ingkarni Wardli North Terrace Campus 5005, Australia
[2] Center for Quantum Computation & Intelligent Systems, University of Technology Sydney, Australia
[3] Department of Mathematics, University of California, San Diego, USA
[4] Department of Mathematics, Princeton University, Fine Hall, Washington Road, Princeton NJ 08544-1000, USA
[5] Institute for Infocomm Research, 1 Fusionopolis Way, #21-01 Connexis (South) 138632, Singapore

Abstract

Low rank matrix recovery is a fundamental task in many real-world applications. The performance of existing methods, however, deteriorates significantly when applied to ill-conditioned or large-scale matrices. In this paper, we therefore propose an efficient method, called Riemannian Pursuit (RP), that aims to address these two problems simultaneously. Our method consists of a sequence of fixed-rank optimization problems. Each subproblem, solved by a nonlinear Riemannian conjugate gradient method, aims to correct the solution in the most important subspace of increasing size. Theoretically, RP converges linearly under mild conditions and experimental results show that it substantially outperforms existing methods when applied to large-scale and ill-conditioned matrices.

1. Introduction

Matrix recovery (MR) has attracted a lot of attention from various research communities, such as statistical machine learning, collaborative filtering, image and signal processing (Candès & Recht, 2009; Negahban & Wainwright, 2012). With the fast development of Web 2.0 in the last decade, big MR problems have been widely involved in many practical applications, leading to great challenges in computation. For instance, in collaborative filtering tasks, the Netflix Prize problem involves 10^8 ratings of 480,189 users on 17,770 movies (KDDCup, 2007). The MR problem is defined as follows:

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

Definition 1. *Given a linear operator $\mathcal{A}: \mathbb{R}^{m\times n} \to \mathbb{R}^l$, let $\mathbf{b} = \mathcal{A}(\widehat{\mathbf{X}}) + \mathbf{e}$ be l linear measurements of an* unknown *rank-$\widehat{r}$ matrix $\widehat{\mathbf{X}} \in \mathbb{R}^{m\times n}$, where $\mathbf{e}$ denotes noise. Then the task of MR is to recover $\widehat{\mathbf{X}}$ by solving*

$$\min_{\mathbf{X}} f(\mathbf{X}), \quad \text{s.t.} \ \ \operatorname{rank}(\mathbf{X}) \le r, \tag{1}$$

where $l \ll mn$, $r \ge \widehat{r}$, and $f(\mathbf{X}) = \frac{1}{2}\|\mathbf{b} - \mathcal{A}(\mathbf{X})\|_2^2$.

The definition of $\mathcal{A}$ depends on the application context, such as matrix completion, quantum state tomography, matrix factorizations; see, e.g., (Recht et al., 2010; Candès & Plan, 2010a; Recht, 2011; Keshavan et al., 2010b; Laue, 2012). Although our derivation is valid for any $\mathcal{A}$ that allows for MR, in the numerical experiments, we will focus on *matrix completion* (MC) as a specific application. In this case, $\mathcal{A}(\mathbf{X})$ is defined as the element-wise restriction of $\mathbf{X}$ on Ξ, a subset of the complete set of entries of $\mathbf{X}$.

Problem (1) is known to be NP-hard. To address it, many researchers proposed to solve the nuclear-norm convex relaxation (Fazel, 2002; Hazan, 2008; Recht et al., 2010): $\min_{\mathbf{X}} \|\mathbf{X}\|_*$, s.t. $\mathcal{A}(\mathbf{X}) = \mathbf{b}$, where $\|\cdot\|_*$ denotes the matrix nuclear norm. A number of algorithms have been proposed to solve this relaxation, such as the singular value thresholding (SVT) (Cai et al., 2010), the augmented Lagrangian method (ALM) (Lin et al., 2010; 2011; Yang & Yuan, 2013). In practice, the following *matrix lasso* problem is also often studied (Toh & Yun, 2010): $\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{b} - \mathcal{A}(\mathbf{X})\|_2^2 + \lambda\|\mathbf{X}\|_*$, where λ is a regularization parameter. Regarding this problem, the accelerated proximal gradient (APG) method has been proven to be effective (Toh & Yun, 2010; Mishra et al., 2013). While nuclear-norm based methods have shown some success in practice, their applicability to large-scale problems is rather limited because of their necessity to compute high-dimensional SVDs. Recently, Mishra et al. (2013) proposed a method to solve the

matrix lasso problem by avoiding high-dimensional SVDs. However, empirically, this method still performs similarly to APG.

To improve the scalability of MR, a different solution is to relax problem (1) to a fixed-rank optimization problem (Mitra et al., 2010; Keshavan et al., 2010a):

$$\min_{\mathbf{X}} f(\mathbf{X}), \quad \text{s.t.} \quad \text{rank}(\mathbf{X}) = r, \tag{2}$$

where r is an estimated rank. This problem is non-convex, but it can be solved efficiently by local-search methods. Particularly, since the fixed-rank matrices belong to a smooth matrix manifold, many efficient methods based on manifold optimization have been proposed (Meyer et al., 2011; Boumal & Absil, 2012; Vandereycken, 2013).

By exploiting the smooth geometry of fixed-rank matrices, fixed-rank based methods have shown superior scalability compared with the nuclear-norm based methods (Boumal & Absil, 2012; Mishra et al., 2012; Vandereycken, 2013). However, there are two deficiencies for the existing fixed-rank methods. *Firstly*, since the ground-truth rank of the matrix to be recovered is usually unknown, it is nontrivial to set the value of r in (2). *Secondly*, as observed in (Ngo & Saad, 2012; Boumal & Absil, 2012), when solving (2) with ill-conditioned $\widehat{\mathbf{X}}$, existing fixed-rank based methods may converge slowly.

To address the above issues, we develop an efficient and scalable algorithm for MR that iteratively increases the rank of the matrix to be recovered by a fixed integer ρ. The main contributions of this paper are as follows:

1) We propose the *Riemannian Pursuit* (RP) method, which essentially solves a sequence of fixed-rank minimization problems using a nonlinear Riemannian Conjugate Gradient (NRCG) method. The convergence of NRCG is guaranteed by the application of a strong Wolfe line search.

2) We prove that RP converges linearly under mild conditions. Compared with other fixed-rank based methods, the proposed optimization scheme can effectively address the convergence issues that occur with ill-conditioned and large rank problems.

3) RP automatically estimates the rank under proper stopping conditions, which avoids the difficulty of the rank estimation in most existing fixed-rank based methods.

2. Notations and Preliminaries

Throughout the paper, we denote by the superscript $^\mathsf{T}$ the transpose of a vector/matrix, $\mathbf{0}$ a vector/matrix with all zeros, $\text{diag}(\mathbf{v})$ a diagonal matrix with a vector of diagonal entries equal to $\mathbf{v}$, and $\|\mathbf{v}\|_p$ the ℓ_p-norm of a vector $\mathbf{v}$. We denote by $[n]$ the list $\{1, ..., n\}$. Given a linear operator $\mathcal{A}$, its adjoint operator is denoted by $\mathcal{A}^*$. Let $\mathbf{A} \odot \mathbf{B}$ and $\langle \mathbf{A}, \mathbf{B} \rangle = tr(\mathbf{A}\mathbf{B}^\mathsf{T})$ be the element-wise product and inner product of $\mathbf{A}$ and $\mathbf{B}$, respectively. Denote the SVD of $\mathbf{X} \in \mathbb{R}^{m\times n}$ as $\mathbf{X} = \mathbf{U}(\text{diag}(\boldsymbol{\sigma}))\mathbf{V}^\mathsf{T} = \sum_{i=1}^{q} \sigma_i \mathbf{u}_i \mathbf{v}_i^\mathsf{T}$, where $q = \min\{m, n\}$ and σ_i is arranged in descending order. The nuclear norm of $\mathbf{X}$ is defined as $\|\mathbf{X}\|_* = \|\boldsymbol{\sigma}\|_1 = \sum_i |\sigma_i|$ and the Frobenius norm of $\mathbf{X}$ is defined as $\|\mathbf{X}\|_F = \|\boldsymbol{\sigma}\|_2$. The condition number $\kappa_r(\mathbf{X})$ of $\mathbf{X}$ w.r.t. a given number r is defined as $\kappa_r(\mathbf{X}) = \sigma_1/\sigma_r$.

2.1. Matrix RIP Condition

To discuss convergence, we introduce the matrix restricted isometry property (RIP) condition (Recht et al., 2010). Specifically, the matrix RIP condition describes a property of a linear operator $\mathcal{A}$ as the smallest number γ_r such that

$$(1 - \gamma_r)||\mathbf{X}||_F^2 \leq ||\mathcal{A}(\mathbf{X})||_F^2 \leq (1 + \gamma_r)||\mathbf{X}||_F^2 \tag{3}$$

holds for all matrices of rank at most r. *Observe that the RIP condition does not hold for MC*. To study the exact recovery condition of MC, the incoherence of matrices is introduced (Candès & Recht, 2009; Candès & Plan, 2010b). Specifically, a matrix $\mathbf{X}$ of rank r with SVD $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T}$ is μ-incoherent ($\mu \geq 1$) if $\forall i \in [r]$

$$||\mathbf{u}_i||_\infty \leq \sqrt{\mu/m} \text{ and } ||\mathbf{v}_i||_\infty \leq \sqrt{\mu/n}. \tag{4}$$

Under these conditions, the RIP conditions of MR has been extended to MC (Meka et al., 2009a).

Proposition 1. *In MC, suppose the observed entry set Ξ is sampled according to the Bernoulli model with each entry $(i, j) \in \Xi$ being independently drawn from a probability p. There exists a constant $C > 0$, for all $\gamma_r \in (0, 1)$, $\mu \geq 1$, $n \geq m \geq 3$, if $p \geq C\mu^2 r^2 \log(n)/(\gamma_r^2 m)$, the following RIP condition holds*

$$(1 - \gamma_r)p||\mathbf{X}||_F^2 \leq ||\mathcal{A}(\mathbf{X})||_F^2 \leq (1 + \gamma_r)p||\mathbf{X}||_F^2, \tag{5}$$

for any μ-incoherent matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ of rank at most r with probability at least $1 - \exp(-n \log n)$.

2.2. Differential Geometry of Fixed-Rank Matrices

Given a positive integer r, consider the smooth submanifold of fixed rank-r matrices,

$$\begin{aligned} \mathcal{M}_r &= \{\mathbf{X} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{X}) = r\} \\ &= \{\mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T} : \mathbf{U} \in \text{St}_r^m, \mathbf{V} \in \text{St}_r^n, ||\boldsymbol{\sigma}||_0 = r\} \end{aligned}$$

with $\text{St}_r^m = \{\mathbf{U} \in \mathbb{R}^{m\times r} : \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}\}$ the Stiefel manifold of $m \times r$ real and orthonormal matrices. The tangent space $T_\mathbf{X}\mathcal{M}_r$ of $\mathcal{M}_r$ at $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T} \in \mathbb{R}^{m\times n}$ is given by $T_\mathbf{X}\mathcal{M}_r = \{\mathbf{U}\mathbf{M}\mathbf{V}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{U}\mathbf{V}_p^\mathsf{T} : \mathbf{M} \in \mathbb{R}^{r\times r}, \mathbf{U}_p \in \mathbb{R}^{m\times r}, \mathbf{U}_p^\mathsf{T}\mathbf{U} = \mathbf{0}, \mathbf{V}_p \in \mathbb{R}^{n\times r}, \mathbf{V}_p^\mathsf{T}\mathbf{V} = \mathbf{0}\}$.

Define a metric $g_{\mathbf{X}}(\mathbf{A},\mathbf{B}) = \langle \mathbf{A},\mathbf{B}\rangle$, where $\mathbf{X} \in \mathcal{M}_r$ and $\mathbf{A},\mathbf{B} \in T_{\mathbf{X}}\mathcal{M}_r$, then $\mathcal{M}_r$ is a Riemannian manifold by restricting $\langle \mathbf{A},\mathbf{B}\rangle$ to the *tangent bundle*. Here the *tangent bundle* is defined as the disjoint union of all tangent spaces $T\mathcal{M}_r = \bigcup_{\mathbf{X}\in\mathcal{M}_r}\{\mathbf{X}\} \times T_{\mathbf{X}}\mathcal{M}_r = \{(\mathbf{X},\mathbf{E}) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} : \mathbf{X} \in \mathcal{M}_r, \mathbf{E} \in T_{\mathbf{X}}\mathcal{M}_r\}$. By restricting $f(\mathbf{X}) = \frac{1}{2}\|\mathbf{b} - \mathcal{A}(\mathbf{X})\|_2^2$ to $\mathcal{M}_r$ we obtain a smooth function on $\mathcal{M}_r$. Its Riemannian gradient is given as the orthogonal projection onto the tangent space of the gradient of f. Define $P_U = \mathbf{U}\mathbf{U}^\mathsf{T}$ and $P_U^\perp = \mathbf{I} - \mathbf{U}\mathbf{U}^\mathsf{T}$ for any $\mathbf{U} \in \mathrm{St}_r^m$. The orthogonal projection of any $\mathbf{Z} \in \mathbb{R}^{m\times n}$ onto the tangent space at $\mathbf{X} = \mathbf{U}\mathrm{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T}$ is defined as

$$P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{Z}) : \mathbf{Z} \mapsto P_U\mathbf{Z}P_V + P_U^\perp\mathbf{Z}P_V + P_U\mathbf{Z}P_V^\perp. \tag{6}$$

Let $\mathbf{G} = \mathcal{A}^*(\mathcal{A}(\mathbf{X}) - \mathbf{b})$. Then, the Riemannian gradient of $f(\mathbf{X})$ on $\mathcal{M}_r$ can be calculated as

$$\mathrm{grad} f(\mathbf{X}) = P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G}). \tag{7}$$

For convenience, we define $P_{T_{\mathbf{0}}\mathcal{M}_r}(\mathbf{Z}) = \mathbf{0}$ when $\mathbf{X} = \mathbf{0}$. Moreover, the *Retraction* mapping on $\mathcal{M}_r$ is to go back from an element in the tangent space to the manifold, which can be computed in a closed form as

$$R_{\mathbf{X}}(\mathbf{E}) = P_{\mathcal{M}_r}(\mathbf{X}+\mathbf{E}) = \sum_{i=1}^{r}\sigma_i\mathbf{p}_i\mathbf{q}_i^\mathsf{T}, \tag{8}$$

where $\sum_{i=1}^{r}\sigma_i\mathbf{p}_i\mathbf{q}_i^\mathsf{T}$ denotes a best rank-r approximation to $\mathbf{X}+\mathbf{E}$. The norm of a tangent vector $\boldsymbol{\zeta}_{\mathbf{X}} \in T_{\mathbf{X}}\mathcal{M}_r$ evaluated at $\mathbf{X}$ is defined as $\|\boldsymbol{\zeta}_{\mathbf{X}}\| = \sqrt{\langle \boldsymbol{\zeta}_{\mathbf{X}}, \boldsymbol{\zeta}_{\mathbf{X}}\rangle}$. We refer to (Vandereycken, 2013) and references therein for more details on the geometry of $\mathcal{M}_r$.

3. Riemannian Pursuit

Solving problem (1) is hard because there is little knowledge about the rank of the matrix to be recovered; otherwise (1) is reduced to a fixed-rank minimization problem. Considering that fixed-rank methods have gained great success in solving big MR problems with explicit knowledge of the rank, we propose to iteratively increase the rank by a fixed integer ρ and then solve a series of fixed-rank minimization subproblems until a proper stopping condition is achieved. Once this procedure is finished, the final rank returned is our rank estimation and we can perform a final, more accurate fixed-rank optimization step. Based on this motivation, the proposed method is presented in Algorithm 1. Since Riemannian optimization is a core element of the algorithm, we refer to it as the Riemannian Pursuit (RP) in the sequel. *The parameter ρ in RP is crucial for both rank estimation and convergence.* For ease of presentation, we leave the detailed setting of choosing ρ for Section 3.2.

As shown in Algorithm 1, $\boldsymbol{\xi}^t = \mathbf{b} - \mathcal{A}(\mathbf{X}^t)$ represents the residual at iteration t. Starting with $\mathbf{X}^0 = \mathbf{0}$, RP iterates with two main steps: 1) identifying the most-active subspace through an *active-subspace search* in Step 2; and 2) solving a *master problem optimization* regarding a fixed-rank minimization problem in Step 3. In Algorithm 1 and in the rest of the paper, we use the notation

$$P_t := P_{T_{\mathbf{X}^t}\mathcal{M}_{t\rho}}.$$

The parameter ϵ_{out} is a tolerance on the stopping condition of Algorithm 1 and will be detailed in Section 3.1.

Algorithm 1 RP: Riemannian Pursuit for MR.

Require: Rank increase ρ. Inner and outer iteration tolerance ϵ_{in} and ϵ_{out}.

1: Initialize $\mathbf{X}^0 = \mathbf{0}$, $\boldsymbol{\xi}^0 = \mathbf{b}$, $\mathbf{G} = \mathcal{A}^*(\boldsymbol{\xi}^0)$, and $t = 1$.

2: Perform an active-subspace search as follows.
2a: Compute $\mathbf{Q} = \mathbf{G} - P_{t-1}(\mathbf{G})$.
2b: Compute a best rank ρ approximation of $\mathbf{Q}$:
$$\mathbf{H}_2^{t-1} = \mathbf{U}_\rho\mathrm{diag}(\boldsymbol{\sigma}_\rho)\mathbf{V}_\rho^\mathsf{T}$$

3: Let $\mathbf{H}_1^{t-1} = P_{t-1}(\mathbf{G})$ and $\mathbf{H}^{t-1} = \mathbf{H}_1^{t-1} + \mathbf{H}_2^{t-1}$.
3a: Choose a proper step size τ_t from (10) and set
$$\mathbf{X}^{\text{intial}} = R_{\mathbf{X}^{t-1}}(-\tau_t\mathbf{H}^{t-1}). \qquad \text{(Warm Start)}$$
3b: Using $\mathbf{X}^{\text{initial}}$ as initial guess, call
$$\mathbf{X}^t = \text{NRCG}(\mathbf{X}^{\text{initial}}, \epsilon_{in}).$$

4: Update $\boldsymbol{\xi}^t = \mathbf{b} - \mathcal{A}(\mathbf{X}^t)$ and $\mathbf{G} = \mathcal{A}^*(\boldsymbol{\xi}^t)$.

5: Quit if stopping condition on ϵ_{out} is achieved; otherwise, let $t := t+1$ and go to Step 2.

In Step 2, the active-subspace search determines the most-active subspace that is orthogonal to $\mathbf{X}^{t-1}$ from $\mathbf{G} = \mathcal{A}^*(\boldsymbol{\xi}^{t-1})$. Such a subspace is obtained by computing the top ρ singular values and vectors of $\mathbf{G} - P_{t-1}(\mathbf{G})$, which is orthogonal to $T_{\mathbf{X}^{t-1}}\mathcal{M}_{(t-1)\rho}$ and thus also to $\mathbf{X}^{t-1}$. Remark that, due to computational reasons, the master problem in step 3b is not solved exactly, thus $P_{t-1}(\mathbf{G})$ is not necessarily zero (or negligible).

After the active-subspace search, we have $\mathrm{rank}(\mathbf{X}^t) = t\rho$, namely $\mathbf{X}^t \in \mathcal{M}_{t\rho}$. The master problem optimization in Step 3 is to solve a fixed-rank problem

$$\min_{\mathbf{X}} \ f(\mathbf{X}), \ \text{ s.t. } \mathrm{rank}(\mathbf{X}) = t\rho, \tag{9}$$

where $f(\mathbf{X}) = \frac{1}{2}\|\mathcal{A}(\mathbf{X}) - \mathbf{b}\|_2^2$ is a smooth function. In particular, we solve it using LRGeomCG from (Vandereycken, 2013), which is a nonlinear Riemannian Conjugate Gradient (NRCG) method.

As shown in Algorithm 1, NRCG involves two parameters, namely the initial point $\mathbf{X}^{\text{intial}}$ and its stopping tolerance ϵ_{in}. Here, $\mathbf{X}^{\text{intial}}$ of rank $t\rho$ in Step (3a) is used as a *warm-start* in NRCG, which is important for improving the overall efficiency of the algorithm. To be more specific, we use

$R_{\mathbf{X}^{t-1}}(-\tau_t \mathbf{H}^{t-1})$ as an initial point for NRCG, where τ_t is a step size that is obtained by a line search on the condition

$$f(R_{\mathbf{X}^{t-1}}(-\tau_t \mathbf{H}^{t-1})) \leq f(\mathbf{X}^{t-1}) - \frac{\tau_t}{2}\langle \mathbf{H}^{t-1}, \mathbf{H}^{t-1}\rangle. \quad (10)$$

Main Theoretical Results: Before presenting the details of NRCG, we summarize two major theoretical results regarding the convergence of Algorithm 1.

Firstly, let $\{\mathbf{X}^t\}$ be the sequence generated by RP, then $f(\mathbf{X}^t)$ decreases monotonically w.r.t. t.

Lemma 1. *Let $\{\mathbf{X}^t\}$ be the sequence generated by RP, then*

$$f(\mathbf{X}^t) \leq f(\mathbf{X}^{t-1}) - \frac{\tau_t}{2}||\mathbf{H}_2^{t-1}||_F^2, \quad (11)$$

where τ_t satisfies condition (10).

Secondly, let $\widehat{\mathbf{X}}$ be the ground-truth low-rank matrix and $\mathbf{e}$ be the additive noise, the following theorem indicates that $f(\mathbf{X}^t)$ decreases linearly when $f(\mathbf{X}^t) > f(\widehat{\mathbf{X}}) = \frac{1}{2}||\mathbf{e}||^2$.

Theorem 1. *Let $\{\mathbf{X}^t\}$ be the sequence generated by RP and $\zeta = \min\{\tau_1, \cdots, \tau_\iota\}$. As long as $f(\mathbf{X}^t) \geq \frac{C}{2}||\mathbf{e}||^2$ (where $C > 1$) and if there exists an integer $\iota > 0$ such that $\gamma_{(\widehat{r}+2\iota\rho)} < \frac{1}{2}$, then* RP *decreases linearly in objective values when $t < \iota$, namely $f(\mathbf{X}^{t+1}) \leq \nu f(\mathbf{X}^t)$, where*

$$\nu = 1 - \frac{\rho\zeta}{2\widehat{r}}\left(\frac{C(1-2\gamma_{(\widehat{r}+2\iota\rho)})^2}{(\sqrt{C}+1)^2(1-\gamma_{(\widehat{r}+2\iota\rho)})}\right)\left(1-\frac{1}{\sqrt{C}}\right)^2.$$

This theorem illustrates the convergence speed of RP under the RIP condition $\gamma_{(\widehat{r}+2\iota\rho)} < \frac{1}{2}$. To apply it to the matrix completion problem, we need to adapt a variant of the RIP condition in (5) and assume that the $\mathbf{X}^t$ are incoherent, uniformly in t.

3.1. Stopping Conditions for RP

According to Lemma 1, RP monotonically decreases in objective value. Without proper stopping conditions, RP may increase the rank until $t\rho \geq h = \min(m, n)$, where $\boldsymbol{\xi}^t = \mathbf{b} - \mathcal{A}(\mathbf{X}^t) = \mathbf{0}$, and the solution will likely be over-fitted. To avoid this issue, one can terminate on a small relative residual,

$$\|\boldsymbol{\xi}^t\|_F / ||\mathbf{b}||_F \leq \lambda_F. \quad (12)$$

In real-world problems, the matrix to be recovered is not exactly low-rank and then (12) may not be adequate. Since RP decreases the objective values monotonically, we propose to use a difference in function values as the stopping condition,

$$2(f(\mathbf{X}^{t-1}) - f(\mathbf{X}^t))/(\rho\|\mathbf{b}\|_F^2) \leq \epsilon_{out},$$

where ϵ_{out} is a predefined tolerance value. This condition is based on the assumption that over-fitting will happen if increasing the rank does not significantly decrease the objective value. In practice, $\epsilon_{out} = 10^{-5}$ is usually a good choice.

3.2. Parameter Selection on ρ

As shown in Theorem 1, RP with a larger ρ converges faster. However, a small ρ is required in order to make an accurate estimation to the rank. Particularly, when dealing with problems of ill-conditioning, ρ should be small enough. We present a simple and effective method to set an appropriate ρ. Let $\boldsymbol{\sigma}$ be the singular vector of $\mathcal{A}^*(\mathbf{b})$, where σ_i is arranged in descending order. Motivated by the thresholding strategy in StOMP for sparse recovery (Donoho et al., 2012), we choose ρ such that for $0 < \eta \leq 1$

$$\sigma_i \geq \eta\,\sigma_1, \ \ \forall i \leq \rho. \quad (13)$$

In other words, ρ denotes the number of sufficiently large singular values of $\mathcal{A}^*(\mathbf{b})$. In general, a smaller η leads to a larger ρ. When setting $\eta = 1$, we trivially have $\rho = 1$, which is not efficient when the exact rank $\widehat{r}$ is large.[1] We refer to the Supplementary Materials for an efficient computational strategy to compute ρ given η.

3.3. Nonlinear Riemannian Conjugate Gradient

In this section, we detail the NRCG method for solving the fixed-rank problem (9) in Step 3b of RP. To differentiate from the outer iteration variable $\mathbf{X}^t$ of RP, we use $\mathbf{X}_k$ for the inner iteration index k of NRCG. In Euclidean space, the search direction $\boldsymbol{\zeta}_k$ of nonlinear CG is calculated as

$$\boldsymbol{\zeta}_k = -\text{grad}f(\mathbf{X}_k) + \beta_k \boldsymbol{\zeta}_{k-1},$$

where β_k is calculated from, for example, the Fletcher–Reeves (FR) rule:

$$\beta_t = \frac{\langle \text{grad}f(\mathbf{X}_k), \text{grad}f(\mathbf{X}_k)\rangle}{\langle \text{grad}f(\mathbf{X}_{k-1}), \text{grad}f(\mathbf{X}_{k-1})\rangle}. \quad (14)$$

Different from Euclidean space, the search direction on a manifold are adapted to follow a path on the manifold (Absil et al., 2008). Particularly, since $\text{grad}f(\mathbf{X}_k) \in T_{\mathbf{X}_k}\mathcal{M}_r$, $\text{grad}f(\mathbf{X}_{k-1}) \in T_{\mathbf{X}_{k-1}}\mathcal{M}_r$, and $\boldsymbol{\zeta}_{k-1}$ are in different tangent spaces of the manifold, the above two equations are not applicable on Riemannian manifolds. To extend nonlinear CG of Euclidean space to Riemannian manifolds, two additional operators, namely *retraction* and *vector transport* are necessary. With the previously defined *retraction* mapping in (8), one can move points towards the direction of a tangent vector and make them stay on the manifold. A *vector transport* $\mathcal{T}$ on a manifold $\mathcal{M}_r$ is a smooth map that transports tangent vectors from one tangent space to another (Absil et al., 2008). Denoting such a vector transport by $\mathcal{T}_{\mathbf{X}\to\mathbf{Y}} : T_{\mathbf{X}}\mathcal{M}_r \to T_{\mathbf{Y}}\mathcal{M}_r$, the conjugate direction can be calculated as

$$\boldsymbol{\zeta}_k = -\mathbf{E}_k + \beta_t \mathcal{T}_{\mathbf{X}_{k-1}\to\mathbf{X}_k}(\boldsymbol{\zeta}_{k-1}), \quad (15)$$

[1] In practice, we suggest setting $\eta \geq 0.60$.

where $\mathbf{E}_k = \text{grad}f(\mathbf{X}_k)$ and β_k is determined from (14).

The NRCG method is presented in Algorithm 2, which includes two major steps: 1) calculating the conjugate search direction in Step 3, and 2) updating $\mathbf{X}_{k+1}$ by retraction, namely $\mathbf{X}_{k+1} = R_{\mathbf{X}_k}(\theta_k\boldsymbol{\zeta}_k)$, where θ_k denotes the step size satisfying the strong Wolfe conditions. Specifically, given a descent direction $\boldsymbol{\zeta}_k \in T_{\mathbf{X}_k}\mathcal{M}_r$, θ_k is determined such that

$$f(R_{\mathbf{X}_k}(\theta_k\boldsymbol{\zeta}_k)) \leq f(\mathbf{X}_k) + c_1\theta_k\langle \text{grad}f(\mathbf{X}_k), \boldsymbol{\zeta}_k\rangle, \tag{16}$$

$$|\langle \text{grad}f(R_{\mathbf{X}_k}(\theta_k\boldsymbol{\zeta}_k)), \mathcal{T}_{\mathbf{X}_{k-1}\to\mathbf{X}_k}(\boldsymbol{\zeta}_k)\rangle| \leq c_2|\langle \text{grad}f(\mathbf{X}_k), \boldsymbol{\zeta}_k\rangle|, \tag{17}$$

where $0 < c_1 < c_2 < 1/2$.

Two choices for vector transport are orthogonal projection

$$\mathcal{T}_{\mathbf{X}\to\mathbf{Y}}(\boldsymbol{\zeta}) = P_{T_{\mathbf{Y}}\mathcal{M}_r}(\boldsymbol{\zeta}) \tag{18}$$

and the scaled differentiated retraction

$$\mathcal{T}_{\mathbf{X}\to\mathbf{Y}}(\boldsymbol{\zeta}) = \alpha \cdot \frac{d}{dt} R_{\mathbf{X}}(\boldsymbol{\nu} + t\boldsymbol{\zeta})|_{t=0}\,, \tag{19}$$

where $\boldsymbol{\nu} = R_{\mathbf{X}}^{-1}(\mathbf{Y})$ and α is such that $\|\mathcal{T}_{\mathbf{X}\to\mathbf{Y}}(\boldsymbol{\zeta})\|_F = \|\boldsymbol{\zeta}\|_F$; see (Sato & Iwai, 2013).

The standard choice (18) from (Vandereycken, 2013) is cheaper to evaluate, but (19) is required for proving the convergence of NRCG.

Proposition 2 (Sato & Iwai (2013))**.** *Given the retraction (8) and vector transport (19) on $\mathcal{M}_r$, there exists a step size θ_k that satisfies the strong Wolfe conditions (16) and (17).*

Lemma 2 (Sato & Iwai (2013))**.** *The search direction $\boldsymbol{\zeta}_k$ generated by* NRCG *using the vector transport (19) and the strong Wolfe conditions (16) and (17) satisfies*

$$-\frac{1}{1-c_2} \leq \frac{\langle \text{grad}f(\mathbf{X}_k), \boldsymbol{\zeta}_k\rangle}{\langle \text{grad}f(\mathbf{X}_{k-1}), \text{grad}f(\mathbf{X}_{k-1})\rangle} \leq \frac{2c_2-1}{1-c_2}. \tag{20}$$

Using (Dieci & Eirola, 1999), (19) can be implemented efficiently even though it will be more expensive than (18). One can show that the difference between (18) and (19) is $O(\|\boldsymbol{\zeta}\|^2)$, hence they have the same behavior near the optimizer where $\|\boldsymbol{\zeta}\| \to 0$. For convenience, we therefore use (18) in the numerical experiments.

Global convergence of NCRG is obtained if the functions $f(R_{\mathbf{X}_k}(\theta\boldsymbol{\zeta}_k))$ are Lipschitz continuously differentiable in θ. Since the manifold $\mathcal{M}_{t\rho}$ is open at points where $\text{rank}(\mathbf{X}) < t\rho$, such a condition cannot hold uniformly on $\mathcal{M}_{t\rho}$. In (Vandereycken, 2013), the additional term $\frac{1}{2}\mu^2(||\mathbf{X}^\dagger||_F^2 + ||\mathbf{X}||_F^2)$ was added to the objective function to penalize rank drops. For simplicity, we assume that f satisfies the necessary Lipschitz conditions throughout the iteration.

Assumption 1. *Let $\{\mathbf{X}_k\}$ and $\{\boldsymbol{\zeta}_k\}$ be the sequence of iterates and search directions generated by the NRCG method in Step (3b). We assume that $\theta \mapsto f(R_{\mathbf{X}_k}(\theta\boldsymbol{\zeta}_k))$ is Lipschitz continuously differentiable with a uniform Lipschitz constant $L > 0$.*

Theorem 2. *Let $\{\mathbf{X}_k\}$ be the sequence generated by the NRCG method in Step (3b) of Algorithm 1 with the strong Wolfe line search, where $0 < c_1 < c_2 < 1/2$, then we have $\lim_{k\to\infty} \inf \text{grad}f(\mathbf{X}_k) = \mathbf{0}$.*

Proof: Combining Lemma 2 and Assumption 1 as in (Sato & Iwai, 2013).

Algorithm 2 NRCG($\mathbf{X}^{\text{intial}}$,$r$, ϵ_{in}) for solving (2).

1: Initialize $\mathbf{X}_1 = \mathbf{X}^{\text{intial}}$ and $\boldsymbol{\zeta}_0 = \mathbf{0}$. Let $k = 1$.
2: Compute the gradient $\mathbf{E}_k = \text{grad}f(\mathbf{X}_k)$ by (7).
3: Compute a conjugate direction $\boldsymbol{\zeta}_k$ according to (15).
4: Choose a step size θ_k satisfying the strong Wolfe conditions (16) and (17), and set $\mathbf{X}_{k+1} = R_{\mathbf{X}_k}(\theta_k\boldsymbol{\zeta}_k)$.
5: Terminate and output $\mathbf{X}_{k+1}$ if the stopping conditions are achieved; otherwise, let $k = k+1$ and go to step 1.

3.4. Computational Advantages of RP

The proposed RP algorithm has several computational advantages. First of all, it is useful for rank detection. Specifically, under the stopping conditions in Section 3.1, RP will automatically estimate the rank as $r = t\rho$.

Secondly, RP converges well on ill-conditioned problems. Even when the condition number $\kappa_r(\mathbf{X})$ is very large, $\kappa_{t\rho}(\mathbf{X})$ will be small for t small. Thus NCRG can converge well and consequently, the total convergence of RP is improved significantly.

Thirdly, RP has good scaling characteristics for solving large-scale problems. For example, the only SVD calculations required are the best rank $t\rho$ approximation of $\mathbf{X}^t + \boldsymbol{\xi}$ for the retraction and the best rank ρ of the matrix $\mathbf{Q}$ in Step 2b. In all cases, the matrices involved are highly structured. Specifically, $\mathbf{X}_k + \theta_k\boldsymbol{\zeta}_k$ is a rank $2t\rho$ matrix, and thus the full SVD can be computed very efficiently; see (Vandereycken, 2013). For $\mathbf{Q} = \mathbf{G} - P_t(\mathbf{G})$, we can use PROPACK (Larsen, 2004) or randomized low-rank approximation (Halko et al., 2011) at a cost of $O((|\mathbf{G}| + t\rho n)\rho)$ where $|\mathbf{G}|$ is the cost of one matrix-vector product with $\mathbf{G}$. For example, $|\mathbf{G}| = O(\widehat{r}n\log^2 n)$ is highly sparse for MC.

Finally, thanks to the warm-start, each application of NRCG requires only a few iterations to achieve a sufficiently accurate solution.

4. Related Studies

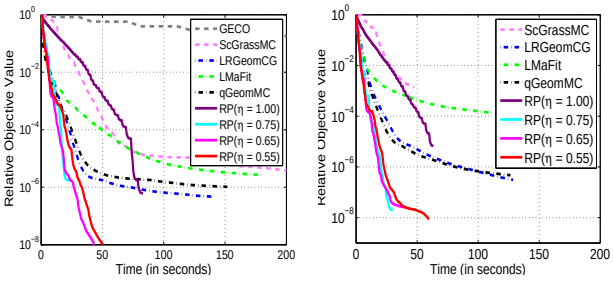
Fixed-rank methods such as the low-rank geometric conjugate gradient method (LRGeomCG) (Vandereycken, 2013), the quotient geometric matrix completion method (qGeomMC) (Mishra et al., 2012), and the method of scaled gradients on Grassmann manifolds for matrix completion (ScGrassMC) (Ngo & Saad, 2012), have shown promising performance. Gradient methods and stochastic gradient methods have been developed to address the fixed-rank problems (Jaggi & Sulovsky, 2010; Wen et al., 2012). All these methods require the estimate of r. Greedy-like algorithms have also been proposed to solve the fixed-rank problem, such as the Singular Value Projection (SVP) (Meka et al., 2009b), Atomic Decomposition for Minimum Rank Approximation (ADMiRA) (Lee & Bresler, 2010), SpaRCS (Waters et al., 2011), and so on. For these methods, they are guaranteed to converge under restricted conditions.

Shwartz et al. (2011) proposed a Greedy Efficient Component Optimization (GECO) algorithm to solve a convex relaxation of the rank-constrained problem by iteratively increasing the rank by 1. Jaggi & Sulovsky (2010) proposed a new approximation algorithm based on a sparse approximate SDP model (Hazan, 2008). Laue (2012) proposed a hybrid strategy to solve the MR problem by iteratively increases the rank by 1. Different from these methods, RP essentially solves a sequence of fixed-rank methods and incrementally increases the rank by $\rho \geq 1$. In practice, it is crucial to use a large ρ to accelerate the convergence speed on big matrices of large ranks. Moreover, unlike RP, GECO solves much more expensive regression subproblems; Laue's method solves the nonlinear master problems with a BFGS method, which is memory inefficient on large-scale problems. Finally, the importance of stopping conditions for rank estimation is absent in these methods.

5. Numerical Experiments

5.1. Baseline Methods and Performance Matric

Following state-of-the-art methods are adopted as baseline methods: SVP (Meka et al., 2009b), APG (Toh & Yun, 2010) (which uses a homotopy strategy to solve the *matrix lasso* problem), GECO (Shwartz et al., 2011), LMaFit (Wen et al., 2012), LMaFit-A (Wen et al., 2012) (which extends LMaFit by automatically estimating the rank). ScGrassMC (Ngo & Saad, 2012), a fixed-rank method which is claimed to alleviate the convergence issue over ill-conditioning problems. LRGeomCG (Vandereycken, 2013), a fixed-rank method that adopts the nonlinear Riemannian Conjugate Descent method with Armijo line search. qGeomMC, a fixed-rank method based on manifold optimization. [2] Note that some related methods, such as the IALM and OptSpace are not considered as baselines in this paper since the adopted baseline methods above have been shown to be state-of-the-art in MR (Wen et al., 2012; Vandereycken, 2013). For RP, we adopt the stopping criteria discussed in Section 3.1.



(a) Gaussian singular values $\mathbf{s}_g$ where ρ w.r.t. different η's is 1, 8, 14, 18, resp.

(b) χ^2 singular values $\mathbf{s}_{\chi}^2$ where ρ w.r.t. different η's is 1, 4, 6, 12, respectively.

Figure 1. Convergence of comparison methods on $\mathbf{s}_g$ and $\mathbf{s}_{\chi^2}$, where $\zeta_{os} = 3$ and $\widehat{r} = 50$. GECO cannot converge within 1 hour on $\mathbf{s}_g$, and we omit its results on $\mathbf{s}_{\chi^2}$. ScGrassMC gets numerical problems after 50 iterations on $\mathbf{s}_{\chi^2}$.

The relative testing error (RTE) is adopted as the comparison metric in synthetic experiments: RTE $= \|\mathcal{P}_{\Xi}(\widehat{\mathbf{X}} - \mathbf{X}^*)\|_F / \|\mathbf{X}^*_{\Xi}\|_F$. The testing root-mean-square error (RMSE) is used as the comparison metric in real-world applications: RMSE $= \|\mathcal{P}_{\Xi}(\widehat{\mathbf{X}} - \mathbf{X}^*)\|_F / \sqrt{(|\Xi|)}$. Here $\widehat{\mathbf{X}}$ denotes the observed matrix (with missing entries filled with '0'), $\mathbf{X}^*$ denotes the recovered matrix, and Ξ denotes the index set of observed entries.

All the experiments (except for GECO) are conducted in Matlab on a PC installed a 64-bit operating system with an Intel(R) Core(TM) i7 CPU (2.80GHz with single-thread mode) and 24GB memory.

5.2. Synthetic Problem Generation

In synthetic experiments, we focus on matrices with large condition numbers. Following (Ngo & Saad, 2012), we generate ground-truth low-rank matrices $\widehat{\mathbf{X}} = \widehat{\mathbf{U}}\mathrm{diag}(\widehat{\boldsymbol{\sigma}})\widehat{\mathbf{V}}^{\mathsf{T}} + \mathbf{e}$, where $\boldsymbol{\sigma}$ is a $\widehat{r}$-sparse vector, $\widehat{\mathbf{U}} \in \mathrm{St}_r^m$, and $\widehat{\mathbf{V}} \in \mathrm{St}_r^n$. Two types of singular values are studied: 1) *Gaussian* sparse singular value $\mathbf{s}_g$ with each nonzero entry

[2] LMaFit, ScGrassMC, qGeomMC and LRGeomCG are from: http://www.montefiore.ulg.ac.be/~mishra/fixedrank/fixedrank.html; APG, SVP and GECO are from: http://www.math.nus.edu.sg/~mattohkc/NNLS.html, http://www.cs.utexas.edu/~pjain/svp/, www.cs.huji.ac.il/~shais/code/index.html, respectively.

sampled from *Gaussian* distribution $N(0, 1000)$, and 2) χ^2 sparse singular values $\mathbf{s}_\chi^2$, where each entry is the square of $\mathbf{s}_g$. The two types of singular values are fast decaying, and their condition numbers $\kappa_{\widehat{r}}(\widehat{\mathbf{X}})$ are very large. Once $\widehat{\mathbf{X}}$ is generated, we sample $l = r(m+n-r) \times \zeta_{os}$ entries from $\widehat{\mathbf{X}}$ uniformly to produce $\mathbf{b}$, where ζ_{os} is the oversampling factor (Lin et al., 2010). In the noisy cases, each sampled entry is disturbed by a *Gaussian* noise with strength $\Delta\|\mathbf{b}\|_2/\|\mathbf{n}\|_2$, where Δ is a strength factor and $\mathbf{n} \in \mathbb{R}^l$ is generated by $\mathbf{n} = randn(l, 1)$ in Matlab.

5.3. Performance Comparison in Noiseless Cases

We compare the convergence of RP on $\mathbf{s}_g$ and $\mathbf{s}_{\chi^2}$ with several baseline methods. To show the impact of η from (13) (and thus ρ) on the convergence of RP, we test $\eta = 1.00, 0.75, 0.65, 0.55$, respectively. For simplicity, we set the rank parameter for the fixed-rank methods as the ground-truth rank, namely, $r = \widehat{r} = 50$, which is the best choice for r in the noiseless case. The relative objective value w.r.t. the **computational time** are shown in Fig. 1. As can be seen from Fig. 1, RP with different η generally converges faster than other methods on both types of matrices. The reason that the other methods converge slowly is due to the large condition numbers of $\mathbf{s}_g$ and $\mathbf{s}_{\chi^2}$. All the above observations justify that RP can converge well on ill-conditioned problems thanks to the reasons mentioned in Section 3.4.

5.4. Performance Comparison in Noisy Cases

In this section, we study the performance of the comparison methods on noisy problems of medium size (i.e., $m = n = 5,000$). We report the training time and RTE of various methods for comparison.

For medium-sized problems, we generate matrices with the ground-truth rank $\widehat{r} = 50$, and produce the observations with noise strength factor $\Delta = 0.01$ under oversampling rates $\zeta_{os} \in \{2, 2.3, 2.5, 2.8, 3.0, 3.3, 3.5, 3.8, 4.0\}$. We compare RP with SVP, APG, LMaFit, LMaFit-A, qGeomMC and LRGeomCG. Note that LMaFit-A can automatically adjust the rank. We set $\lambda_F = 0.01$, $\epsilon_{out} = 10^{-5}$ for the stopping conditions and $\eta = 0.65$ for RP. For APG, we set the trade-off parameter $\lambda = 10^{-3}\sigma_{\max}$, where $\sigma_{\max}$ is the largest singular value of $\mathcal{A}^*(\mathbf{b})$. For all the fixed-rank methods, we set $r = \widehat{r} = 50$. We use default settings for the other parameters of the baselines. For each oversampling rate, we run 10 independent experiments. The *Averaged* computational time, RTE and the estimated ranks are shown in Fig. 2(a), Fig. 2(b) and Fig. 2(c), respectively. From Fig. 2(a) and Fig. 2(b), under various oversampling factors, RP generally shows the least relative testing error and the best optimization efficiency among all methods. More importantly, from Fig. 2(c), only RP can estimate the target rank correctly under various oversampling factors; while APG can only correctly detect the rank with enough observations.

Table 1. Averaged training time (seconds) on big matrices.

Data Type	$\mathbf{s}_g$		$\mathbf{s}_{\chi^2}$	
r	50	100	50	100
LRGeomCG	316.8	992.2	564.3	1018.8
qGeomMC	216.1	1091.5	415.1	455.3
RP	**57.5**(48)	**205.6**(102)	**75.4**(48)	**150.8**(85)

* The number in bracket is the rank estimated by RP. Note that LRGeomCG and qGeomMC use the ground-truth rank ($r = \widehat{r}$).

5.5. Performance Comparison on Big Matrices

In the experiments on large-scale matrix completion problems, we vary the values of $\widehat{r}$ in $\{50, 100\}$, and set $m = n = 20,000$, $\zeta_{os} = 4$ and $\Delta = 0.05$. Here, we only compare the scalability of RP with that of LRGeomCG and qGeomMC since there two methods have shown better efficiency than other baselines. We set $\lambda_F = 0.05$, $\epsilon_{out} = 10^{-5}$ and $\eta = 0.65$ for RP. To demonstrate the superiority of RP over the baselines in terms of computational speed, we terminate the baselines **once they achieve the same objective value** of RP or a maximum of 400 iterations are achieved. In addition, we set $r = \widehat{r}$ for these two fixed-rank methods. The other parameters are the same as those in the experiments on the medium-sized problems.

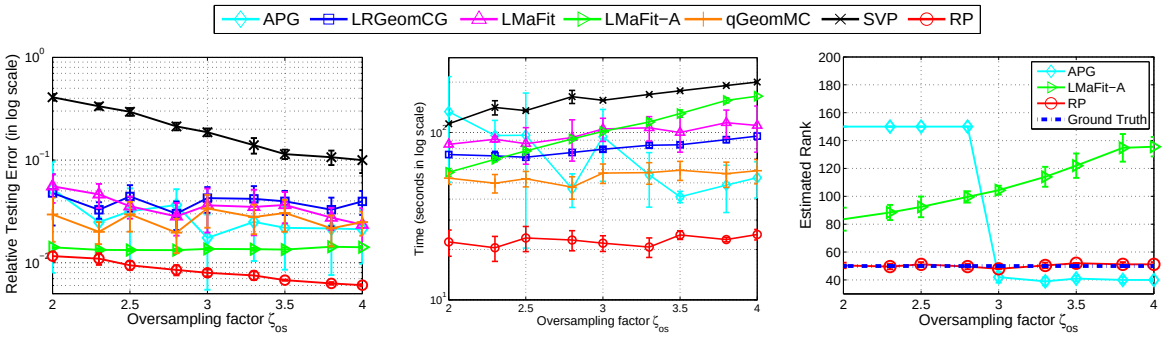
We run 10 independent experiments and record the averaged results for comparison. The averaged computational time and RTE are listed in Table 1 and Table 2, respectively. As can be found from Table 1, in general, the computational time of RP is 4 times faster than that of LRGeomCG, and 3 times faster than that of qGeomMC, respectively. In addition, as can be seen from Table 2, with the same training error, RP is also slightly better than the other two baseline methods in terms of the relative testing error. Notice that here we choose $r = \widehat{r}$ for LRGeomCG and qGeomMC, which is the optimal choice for rank estimation.

Table 2. Averaged relative testing error on big matrices.

Data Type	$\mathbf{s}_g$		$\mathbf{s}_{\chi^2}$	
r	50	100	50	100
LRGeomCG	0.0374	0.0388	0.074	0.0695
qGeomMC	0.0351	0.0316	0.120	0.0347
RP	**0.0316**	**0.0306**	**0.031**	**0.0275**

5.6. Real-World Experiments

In this section, we compare RP with the baseline methods on two real-world large-scale collaborative filtering datasets: MovieLens with 10M ratings (denoted by Movie-10M) (Herlocker et al., 1999) and Netflix Prize dataset (KDDCup, 2007). Movie-10M contains 10M ratings given by 71,567 users on 10,681 movies while Netflix Prize contains 100,480,507 ratings given by 480,189 users on 17,770 movies. The baseline methods include



(a) Relative testing error (in log scale) (b) Training time (seconds in log scale) (c) Estimated rank

Figure 2. Comparison on medium-sized problems of rank $\widehat{r} = 50$. The results are obtained by averaging over 10 independent trials.

APG, LRGeomCG, qGeomMC, Lmafit, Lmafit-A, GECO, Jaggi's method and Laue's method.

In general, collaborative filtering data are very noisy. As a result, the singular values of the matrices tend to be long-tail. Therefore, we need to set larger stopping tolerances to alleviate over-fitting. For this experiment, we set $\lambda_F = 0.2$ and $\epsilon_{out} = 10^{-4}$. We set $\eta = 0.65$, and the detected ranks by RP are used as the rank estimations for the fixed-rank methods, namely LRGeomCG, qGeomMC and LMaFIT. Finally, we constrain the maximum rank for all methods to 100. For comparison, we report the testing RMSE of different methods over 10 random 80/20 train/test partitions as explained in (Laue, 2012).

Comparison results are shown in Table 3. From the table, we can observe that RP performs the best among all the methods in terms of RMSE and computational efficiency. It is worth mentioning that we use the rank detected by RP as the rank estimation for LRGeomCG and qGeomMC. Therefore, RP can be much faster than these two methods if the cost of the model selection is considered.

Table 3. Experimental results on real-world datasets.

Dataset	Movie-10M		Netflix	
	RMSE	Time (seconds)	RMSE	Time (seconds)
APG	1.096	1048±17	-	-
LRGeomCG	0.824	338±11	0.867	3128±35
QgeomMC	0.850	189±7	0.880	3965 ± 74
Lmafit	0.837	307±1	0.875	3798±50
Lmafit-A	0.969	421±16	0.962	5286±165
RP	**0.817**	81±1	**0.859**	1332±27

* Result of APG on Netflix is absent due to out-of-memory issue. The standard variations of RMSE are not reported since they are not significant. The average ranks estimated by APG, Lmafit-A and RP on Movie are 100, 77 and 10, respectively. The average ranks estimated by Lmafit-A and RP on Netflix are 81 and 12, respectively.

Due to the absence of source codes, we record the published results of GECO (Shwartz et al., 2011), Jaggi's method (Jaggi & Sulovsky, 2010) and Laue's method (Laue, 2012), on the Movie-10M dataset. The experimental settings of these methods reported in the literatures are similar to ours, thus the comparison is fair. In addition, we list the training time, the times of speedup, RMSE and the CPU details in Table 4 for reference.

Table 4. Performance comparison on Movie-10M dataset.

Methods	Time (in seconds)	SpeedUp	RMSE	CPU(GHz)
GECO	784,941	**9,000x**	0.821	2.5
Laue	2,663	**30x**	0.815	2.5
Jaggi	3,120	**38x**	0.862	2.4
RP	81	–	0.817	2.8

From Table 4, we observe that on the Movie-10M dataset, RP obtains comparable or better performance to the baseline methods in terms of RMSE, but can achieve great speedup with similar CPUs. Particularly, RP is orders of magnitude faster than all the other methods. With these comparisons, we can conclude that RP can achieve much faster training speed over the comparison methods.

6. Conclusion

We propose a Riemannian Pursuit (RP) method for tackling big MR problems. In contrast to nuclear-norm based methods, RP only needs to compute rank-ρ truncated SVD with ρ very small per iteration, as opposed to APG which may take hundreds of high-dimensional SVDs. By exploiting the Riemannian geometry of the fixed-rank manifold, RP uses a more efficient master solver. Moreover, RP increases the rank of the matrix by $\rho > 1$ per iteration, thus it exhibits good scalability for big MR problems with large ranks. Finally, RP automatically detects the rank with appropriate stopping conditions, and performs well on ill-conditioned problems. Extensive experimental results show that RP achieves superb scalability and maintain similar or better MR performance compared with state-of-the-art methods.

Acknowledgments

This research was in part supported by the Singapore NTU A*SERC under Grant 112 172 0013, and the Australian Research Council Future Fellowship FT130100746.

References

Absil, P.-A. and Mahony, R. and Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Boumal, N. and Absil, P.-A. Rtrmc: A Riemannian trust-region method for low-rank matrix completion. In *NIPS*, 2012.

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

Candès, E. J. and Plan, Y. Tight oracle bounds for low-rank matrix recovery from a minimal number of random measurements. *IEEE Trans. on Inform. Theory*, 57(4):2342–2359, 2010a.

Candès, E. J. and Plan, Y. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010b.

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9:717–772, 2009.

Dieci, L. and Eirola, T. On smooth decompositions of matrices. *SIAM J. Numer. Anal.*, 20(3):800-819, 1999.

Donoho, D. L., Tsaig, Y., Drori, I., and Starck, J. L. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Trans. Info. Theory*, 58(2):1094–1121, 2012.

Fazel, M. Matrix rank minimization with applications. 2002. PhD thesis, Stanford University.

Golub, G. H. and Van Loan, C. F. *Matrix Computations*. JHU Press, 3rd edition, 1996.

Halko, N. and Martinsson, P. G. and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions *SIAM Review*, 53(2): 217–288, 2011

Hazan, E. Sparse approximate solutions to semidefinite programs. *LATIN*, pp. 306–316, 2008.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. An algorithmic framework for performing collaborative filtering. In *SIGIR*, 1999.

Jaggi, M. and Sulovsky, M. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010.

KDDCup. ACM SIGKDD and Netflix. In *Proceedings of KDD Cup and Workshop*, 2007.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from a few entries. *IEEE Trans. on Info. Theory*, 56:2980–2998, 2010a.

Keshavan, R. H, Montanari, A., and Oh, S. Matrix completion from noisy entries. *JMLR*, 99:2057–2078, 2010b.

Laue, S. A hybrid algorithm for convex semidefinite optimization. In *ICML*, 2012.

Lee, K. and Bresler, Y. ADMiRA: Atomic decomposition for minimum rank approximation. *IEEE Trans. on Inform. Theory*, 56(9):4402–4416, 2010.

Lin, Z., Chen, M., and Ma, Y. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, UIUC, 2010.

Lin, Z., Liu, R., and Su, Z. Linearized alternating direction method with adaptive penalty for low-rank representation. *arXiv preprint arXiv:1109.0367*, 2011.

Meka, R., Jain, P., and Dhillon, I. S. Guaranteed rank minimization via singular value projection. Technical report, 2009a.

Meka, R., Jain, P., and I.S.Dhillon. Guaranteed rank minimization via singular value projection. In *NIPS*, 2009b.

Meyer, G., Bonnabel, S., and Sepulchre, R. Linear regression under fixed-rank constraints: A Riemannian approach. In *ICML*, 2011.

Mishra, B., Apuroop, K. A., and Sepulchre, R. A Riemannian geometry for low-rank matrix completion. Technical report, 2012.

Mishra, B., Meyer, G., Bach, F., and Sepulchre, R. Low-rank optimization with trace norm penalty. *SIAM J. Optim.*, 23(4): 2124–2149, 2013.

Mitra, K., Sheorey, S., and Chellappa, R. Large-scale matrix factorization with missing data under additional constraints. In *NIPS*, 2010.

Negahban, S. and Wainwright, M. J. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *JMLR*, 13:1665–1697, 2012.

Ngo, T. T. and Saad, Y. Scaled gradients on Grassmann manifolds for matrix completion. In *NIPS*, 2012.

Larsen, R. M. PROPACK—Software for large and sparse SVD calculations `http://soi.stanford.edu/~rmunk/PROPACK`, 2004

Recht, B. A simpler approach to matrix completion. *JMLR*, pp. 3413–3430, 2011.

Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3), 2010.

Ring, W. and Wirth, B. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM J. Optim.*, 22(2):596–627, 2012.

Sato, H. and Iwai, T. A new, globally convergent Riemannian conjugate gradient method. *Optimization: A Journal of Mathematical Programming and Operations Research*, (ahead-of-print): 1–21, 2013.

Selvan, S. E., Amato, U., Gallivan, K. A., Qi, Ch., Carfora, M. F., Larobina, M., and Alfano, B. Descent algorithms on oblique manifold for source-adaptive ica contrast. *IEEE Trans. Neural Netw. Learning Syst.*, 23(12):1930–1947, 2012.

Shalit, U., Weinshall, D., and Chechik, G. Online learning in the embedded manifold of low-rank matrices. *JMLR*, 13:429–458, 2012.

Shwartz, S. S., Gonen, A., and Shamir, O. Large-scale convex minimization with a low-rank constraint. In *ICML*, 2011.

Toh, K.-C. and Yun, S. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pac. J. Optim.*, 6:615–640, 2010.

Vandereycken, B. Low-rank matrix completion by Riemannian optimization. *SIAM J. Optim.*, 23(2):1214–1236, 2013.

Waters, A. E., Sankaranarayanan, A. C., and Baraniuk, Richard G. Sparcs: Recovering low-rank and sparse matrices from compressive measurements. In *NIPS*, 2011.

Wen, Z., Yin, W., and Zhang, Y. Solving a low-rank factorization model for matrix completion by a non-linear successive over-relaxation algorithm. *Math. Program. Comput.*, 4(4):333–361, 2012.

Yang, J. and Yuan, X. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of Computation*, 82(281):301–329, 2013.