Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Riemannian Submanifold Tracking on Low-Rank Algebraic Variety

Qian Li
Chinese Academy of Sciences
Beijing, China
qianli.charlene@gmail.com

Zhichao Wang*
Tsinghua University
Beijing, China
wzchary@gmail.com

Abstract

Matrix recovery aims to learn a low-rank structure from high dimensional data, which arises in numerous learning applications. As a popular heuristic to matrix recovery, convex relaxation involves iterative calling of singular value decomposition (SVD). Riemannian optimization based method can alleviate such expensive cost in SVD for improved scalability, which however is usually degraded by the unknown rank. This paper proposes a novel algorithm RIST that exploits the algebraic variety of low-rank manifold for matrix recovery. Particularly, RIST utilizes an efficient scheme that automatically estimate the potential rank on the real algebraic variety and tracks the favorable Riemannian submanifold. Moreover, RIST utilizes the second-order geometric characterization and achieves provable superlinear convergence, which is superior to the linear convergence of most existing methods. Extensive comparison experiments demonstrate the accuracy and efficiency of RIST algorithm.

Introduction

Matrix recovery is becoming a fundamental problem arising in many applications ranging from machine learning, data mining to computer vision (Vandereycken 2013; Liu et al. 2011; Candès et al. 2011). In particularly, low-rank matrix recovery decomposes a possibly noisy data $\mathbf{Z}$ into a low-rank component $\mathbf{X}$ and a sparse component $\mathbf{E}$, and this can be formalized as an optimization problem:

$$\min_{\mathbf{X},\,\mathbf{E}} : \operatorname{rank}(\mathbf{X}) + \lambda\Upsilon(\mathbf{E}) \quad \text{s.t. } \mathbf{Z} = \mathcal{A}(\mathbf{X}) + \mathcal{B}(\mathbf{E}) \quad (1)$$

where $\operatorname{rank}(\cdot)$ encourages variable $\mathbf{X}$ to be low-rank, $\Upsilon(\cdot)$ refers to a non-smooth regularizer and λ is the regularization parameter. Both $\mathcal{A}(\cdot)$ and $\mathcal{B}(\cdot)$ are linear operators determined by applications (Candès et al. 2011; Candès and Recht 2009; Recht 2011).

Early attempts adopt the convex relaxation to make the non-convex low-rank matrix recovery tractable (e.g., the trace norm as the convex surrogates for the rank function), such as EALM (Lin, Chen, and Ma 2010), IALM (Lin, Chen, and Ma 2010) and LSADM (Goldfarb, Ma, and Scheinberg 2013). Above algorithms are greatly influenced by the cost of iterative calling of singular value decomposition (SVD) or truncated SVD (Bouwmans and Zahzah 2014). To reduce this cost, many algorithms transform the problem into a convex programming on the positive semi-definite matrices and use the approximate SDP solver (Hazan 2008; Jaggi, Sulovsk, and others 2010). However, these algorithms either converge slowly (Wang et al. 2014; Jaggi 2013) or are memory inefficient on large-scale problems (Laue 2012; Tan et al. 2014). More significantly, these algorithms resolve the convex-relaxed problem of low-rank matrix recovery instead of the original non-convex one, which may produce the solutions that seriously deviate from the true ones (Papamakarios, Panagakis, and Zafeiriou 2011).

An alternative heuristic resorts to Riemannian optimization by exploiting the smooth geometry for optimum searching, which has shown superior scalability than the convex relaxation methods (Vandereycken 2013; Candès et al. 2011). Most existing Riemannian optimization based algorithms differ in their specific choice of Riemannian manifold. For instance, RTRMC (Boumal and Absil 2011) and Lingo (Li et al. 2015) constrain the searching space over Grassmannian manifolds after adopting matrix factorization. RP (Tan et al. 2014), LRGeomCG (Vandereycken 2013) and ScGrassMC (Candès et al. 2011)) exploit the rank constraint as the fixed-rank matrix manifold and directly optimize the matrix recovery problem via Riemannian theory. Though most Riemannian optimization based algorithms show superior performance than classical heuristics on numerical experiments, two main limitations still exist. First, the rank of the matrix to be recovered determines the geometry of Riemannian manifold, which is usually unknown and non-trivial to specify. Moreover, most Riemannian optimization approaches establish the linear convergence rate at best. In addition, as Riemannian manifold is open without boundary when the rank is fixed, this convergence analysis may be limited due to the neglect of manifold boundary (Schneider and Uschmajew 2015; Agarwal, Negahban, and Wainwright 2010; Yan et al. 2015).

To address above issues, this paper proposes an effective matrix recovery solver named *Riemannian Submanifold Tracking* (RIST), which optimizes on the closure of fixed-rank matrix manifolds and searches for the optimal solution with the second-order information of Riemannian geometry.

*Corresponding author

The main contributions of this work are as follows:

- Rather than using unbounded manifolds, RIST exploits the searching space over the real-algebraic variety that is closed with boundary. Theoretical analysis proves that RIST achieves a super-linear convergence rate, which is superior to the linear convergence rate of most state-of-the-art methods.
- By tracking the submanifold of real-algebraic variety, the rank can be automatically estimated, and this alleviates the difficulty of rank estimation in most existing fixed-rank matrix manifolds.
- RIST utilizes the second-order geometric characterization and optimizes over the submanifold of the real-algebraic variety, which benefits for achieving an accurate optimum with fast convergence. Numerical experiments on the task of Robust PCA and matrix completion also verify the effectiveness of RIST algorithm.

Preliminaries and Notations

Riemannian Manifolds can be considered as low dimensional smooth surfaces embedded in a higher dimensional Euclidean space. Given a positive integer k, the fixed-rank matrix manifold is denoted as

$$\begin{aligned}\mathcal{M}_k &= \{\mathbf{X} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{X}) = k\} \\ &= \{\mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^T : \mathbf{U} \in \text{St}_k^m, \mathbf{V} \in \text{St}_k^n, \|\boldsymbol{\sigma}\|_0 = k\}\end{aligned} \tag{2}$$

where $\text{St}_k^m = \{\mathbf{U} \in \mathbb{R}^{m\times k} : \mathbf{U}^T\mathbf{U} = \mathbf{I}\}$ specifies the Stiefel manifold of $m \times k$ semi-orthogonal matrices. The set $\mathcal{M}_k$ is a submanifold of dimension $(m+n-k)k$ embedded in $\mathbb{R}^{m\times n}$, and its tangent space at $\mathbf{X} \in \mathcal{M}_k$ is given by

$$T_{\mathbf{X}}\mathcal{M}_k = \left\{ [\mathbf{U}\,\mathbf{U}_\perp] \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} [\mathbf{V}\,\mathbf{V}_\perp]^T \right\} \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{k\times k}$, $\mathbf{B} \in \mathbb{R}^{k\times(n-k)}$, $\mathbf{C} \in \mathbb{R}^{(m-k)\times k}$, and $\mathbf{0} \in \mathbb{R}^{(m-k)\times(n-k)}$. Given a smooth function f on manifold, its Riemannian gradient grad $f(\mathbf{X})$ is given as the orthogonal projection of the Euclidean gradient $\nabla f(\mathbf{X})$ onto the tangent space $T_{\mathbf{X}}\mathcal{M}_k$ at $\mathbf{X}$. Let $P_{\mathbf{U}} = \mathbf{U}\mathbf{U}^T, P_{\mathbf{V}} = \mathbf{V}\mathbf{V}^T$, the orthogonal projection is computed as

$$P_{T_{\mathbf{X}}\mathcal{M}_k}(\mathbf{B}) = \{P_{\mathbf{U}}\mathbf{B}P_{\mathbf{V}} + P_{\mathbf{U}}^{\perp}\mathbf{B}P_{\mathbf{V}} + P_{\mathbf{U}}\mathbf{B}P_{\mathbf{V}}^{\perp}\} \tag{4}$$

The Riemannian gradient grad $f(\mathbf{X})$ satisfies Eq. (5):

$$\text{grad}\, f(\mathbf{X}) = P_{T_{\mathbf{X}}\mathcal{M}_k}(\nabla f(\mathbf{X})) \tag{5}$$

For any $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \in \mathcal{M}_k$ and $\boldsymbol{\xi} \in T_{\mathbf{X}}\mathcal{M}_k$, Riemannian Hessian in the direction of tangent vector $\boldsymbol{\xi}$ satisfies

$$\begin{aligned}\text{Hess} f(\mathbf{X})[\boldsymbol{\xi}] =& P_{\mathbf{U}}\boldsymbol{\xi}P_{\mathbf{V}} + P_{\mathbf{U}}^{\perp}(\boldsymbol{\xi} + \nabla f(\mathbf{X})\mathbf{V}_p\Sigma^{-1}\mathbf{V}^T)P_{\mathbf{V}} \\ &+ P_{\mathbf{U}}(\boldsymbol{\xi} + \mathbf{U}\boldsymbol{\Sigma}^{-1}\mathbf{U}_p^T\nabla f(\mathbf{X}))P_{\mathbf{V}}\end{aligned} \tag{6}$$

where $\mathbf{U}_p^T\mathbf{U} = \mathbf{0}$ and $\mathbf{V}_p^T\mathbf{V} = \mathbf{0}$.

The real-algebraic variety $\mathcal{M}_{\leq k}$ specifies the set of matrices with rank less than k, which can be also considered as the closure of fixed-rank matrix manifolds:

$$\begin{aligned}\mathcal{M}_{\leq k} =& \{\mathbf{X} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{X}) \leq k\} \\ =& \mathcal{M}_1 \cup \mathcal{M}_2 \cup \cdots \cup \mathcal{M}_k\end{aligned} \tag{7}$$

Suppose that $\mathcal{U} = \text{ran}(\mathbf{X})$ and $\mathcal{V} = \text{ran}(\mathbf{X}^T)$, the tangent cone with respect to $\mathcal{M}_{\leq k}$ is as follows

$$T_{\mathbf{X}}\mathcal{M}_{\leq k} = T_{\mathbf{X}}\mathcal{M}_{k-\kappa} \oplus \{\Xi_\kappa \in \mathcal{U}^\perp \otimes \mathcal{V}^\perp\} \tag{8}$$

Noted that Ξ_κ is the best rank-κ approximation of $\nabla f(\mathbf{X}) - P_{T_{\mathbf{X}}\mathcal{M}_{k-\kappa}}(\nabla f(\mathbf{X}))$. Then the Riemannian gradient can be computed by

$$\text{grad}\, f(\mathbf{X}) = P_{T_{\mathbf{X}}\mathcal{M}_{k-\kappa}}(\nabla f(\mathbf{X})) + \Xi_\kappa \tag{9}$$

Given the tangent vector $\boldsymbol{\xi}$ defining a direction over the tangent space, retraction operator smoothly maps the element from tangent spaces to manifold as follows:

$$R_{\mathcal{M}} : \boldsymbol{\xi} \to R(\mathbf{X}, \boldsymbol{\xi}) = P_{\mathcal{M}}(\mathbf{X} + \boldsymbol{\xi}) = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{10}$$

It is worth noting that the projection $P_{\mathcal{M}}(\cdot)$ is the rank-k approximation operation. Both $\mathbf{u}_i$ and $\mathbf{v}_i$ are the (ordered) singular values and vectors from $\mathbf{X} + \boldsymbol{\xi}$, and for references therein please refer to (Schneider and Uschmajew 2015).

RIST Algorithm

This section describes the RIST algorithm for non-convex matrix recovery in detail, which exploits the geometry of real-algebraic variety and adopts a trust-region method for optimization. Moreover, this section also gives theoretical justification on the superlinear convergence of RIST algorithm.

Riemannian Submanifold Tracking

RIST algorithm reformulates the non-convex matrix recovery of problem (1) as a Riemannian optimization over the searching space of real-algebraic variety $\mathcal{M}_{\leq k}$ as follows:

$$\min_{\mathbf{X},\mathbf{E}} : f(\mathbf{X}, \mathbf{E}) + \lambda\Upsilon(\mathbf{E}) \quad \text{s.t. } \mathbf{X} \in \mathcal{M}_{\leq k}, \tag{11}$$

Let $f(\mathbf{X}, \mathbf{E}) = \|\mathcal{A}(\mathbf{X}) + \mathcal{B}(\mathbf{E}) - \mathbf{Z}\|_F^2$. The following part describes the details of RIST algorithm for solving problem (11).

The rank determines the real-algebraic variety $\mathcal{M}_{\leq k}$ in some sense. Most previous approaches initialize the rank empirically or resort to full SVD, which is inaccurate and expensive for large-scale problem. Motivated by RandomSVD (Halko, Martinsson, and Tropp 2011), this paper proposes a simple and effective schema to initialize the rank through randomized partial matrix decompositions, as shown in Algorithm 1. Given the observed matrix $\mathbf{Z}$, only few $\theta \geq 1$ singular values are computed sequentially by using a random matrix Θ. Algorithm 1 finally outputs the κ as the initial rank until the following condition is satisfied:

$$\frac{\sigma_\kappa}{\sum_{i=1}^{\kappa}\sigma_i} \leq \eta, \quad 0 \leq \eta \leq 0.05 \tag{12}$$

where $\sigma_i \in \boldsymbol{\sigma} = \{\sigma_1, \sigma_2, \cdots, \sigma_i, \sigma_\kappa\}$ are arranged in descending order. Moreover, the condition (12) is also significant for updating the rank by summing up a dynamic integer κ, which tracks the submanifold $\mathcal{M}_k$ within its closure $\mathcal{M}_{\leq k}$.

Algorithm 1 Rank Initialization

Input: observed matrix $\mathbf{Z} \in \mathbb{R}^{m\times n}$, $\theta \geq 1$.
Output: κ
$\mathbf{U} = \mathbf{0} \in \mathbb{R}^{m\times\theta}$, $\mathbf{\Sigma} = \mathbf{0} \in \mathbb{R}^{\theta\times\theta}$, $\mathbf{V} = \mathbf{0} \in \mathbb{R}^{n\times\theta}$;
repeat
  Generate standard Guassian matrix $\Theta \in \mathbb{R}^{n\times\theta}$.
  Compute the random SVD by calling: $[\mathbf{U}_\theta, \mathbf{\Sigma}_\theta, \mathbf{V}_\theta] = \text{RandomSVD}(\mathbf{Z} - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \theta, \Theta)$
  $\mathbf{U} \leftarrow [\mathbf{U}\ \ \mathbf{\Sigma}_\theta]$, $\mathbf{\Sigma} \leftarrow [\mathbf{\Sigma}\ \ \mathbf{\Sigma}_\theta]$, $\mathbf{V} \leftarrow [\mathbf{V}\ \ \mathbf{V}_\theta]$.
  Terminate if singular values in diag($\mathbf{\Sigma}$) satisfies Eq. (12); otherwise let $\kappa = \kappa + \theta$.
until convergence
return κ.

To accelerate the convergence, RIST implements a warm-start strategy over $\mathcal{M}_{\leq k}$ to yield initial point $(\mathbf{X}_*, \mathbf{E}_*)$ for a series of second-order Riemannian subproblems over fixed-rank manifold $\mathcal{M}_k$. To update $\mathbf{X}$ as in Algorithm 2, we fix $\mathbf{E} = \mathbf{E}_t$ and minimize a local model of $f(\mathbf{X}, \mathbf{E}_t)$ with respect to $\mathbf{X}$. Given the search direction $\boldsymbol{\xi} = \text{grad}\, f(\mathbf{X}, \mathbf{E}_t)$ and the step size α obtained by Armijo rule, $\mathbf{X}_*$ can be updated by smooth line-search Riemannian algorithm as follows:

$$f(\mathbf{X}_{t-1}, \mathbf{E}_t) - f(R(\mathbf{X}_t, -\alpha_t\boldsymbol{\xi}_t), \mathbf{E}_t) \geq \alpha_t\langle\boldsymbol{\xi}_t, \boldsymbol{\xi}_t\rangle \quad (13)$$

For updating $\mathbf{E}$, RIST fixes $\mathbf{X}$ and solves the objective function (11). The potential optimum $\mathbf{E}$ may vary. For Robust PCA (RPCA) (Candès et al. 2011), the optimum $\mathbf{E}_*$ satisfies

$$\mathbf{E}_* = \mathcal{S}[\mathbf{Z} - \mathcal{A}(\mathbf{X}_0)] \quad (14)$$

Algorithm 2 RIST: Riemann Submanifold Tracking

Input: estimated rank κ
Output: $\mathbf{X}$, $\mathbf{E}$
Initialize $\mathbf{X}_1 = \mathbf{0}$, $\mathbf{E}_1 = \mathbf{0}$, $k = \kappa$;
for $t = 1, 2, \ldots$ **do**
  Compute the grad$f(\mathbf{X}_t, \mathbf{E}_t)$ on $\mathcal{M}_{\leq k}$ by Eq. (9).
  Update $\mathbf{X}_* = R(\mathbf{X}_t, \alpha_t\text{grad}\, f(\mathbf{X}_t, \mathbf{E}_t))$ on $\mathcal{M}_{\leq k}$ by Eq. (10).
  Compute $\mathbf{E}_*$ by Eq. (14).
  Let $k = k + \kappa$.
  Use $(\mathbf{X}_*, \mathbf{E}_*)$ as warm-start point on $\mathcal{M}_k$ and call: $(\mathbf{X}_{t+1}, \mathbf{E}_{t+1}) = \text{ROM}(\mathbf{X}_*, \mathbf{E}_*, k)$
  Let $\boldsymbol{\sigma} = \text{diag}(\mathbf{X}_{t+1})$ and update κ by Eq. (12).
end for
return $\mathbf{X}$ and $\mathbf{E}$.

Submanifold Optimization over $\mathcal{M}_k$

After the initial points $(\mathbf{X}_*, \mathbf{E}_*)$ are computed, RIST calls an accurate second-order Riemannian optimization (ROM) over $\mathcal{M}_k$ submanifold. We fix $\mathbf{E} = \mathbf{E}_j$ as in Algorithm 3 of ROM. The second-order approximation over the tangent space is

$$m(\mathbf{X}, \mathbf{E}_j; \boldsymbol{\delta}) = f(\mathbf{X}, \mathbf{E}_j) + \langle\text{grad}\, f(\mathbf{X}, \mathbf{E}_j), \boldsymbol{\delta}\rangle + \frac{1}{2}\langle\text{Hess} f(\mathbf{X}, \mathbf{E}_j)[\boldsymbol{\delta}], \boldsymbol{\delta}\rangle \quad (15)$$

where $\boldsymbol{\delta} \in T_{\mathbf{X}}\mathcal{M}_k$, grad $f(\mathbf{X}, \mathbf{E}_j)$ and Hess$f(\mathbf{X}, \mathbf{E}_j)[\boldsymbol{\delta}]$ are yielded by equation (5) and (6) respectively. The low-rank problem (11) with respect to $\mathbf{X}$ can be reformulated as follows:

$$\min_{\boldsymbol{\delta}} : m(\mathbf{X}, \mathbf{E}_j; \boldsymbol{\delta}) + \Upsilon(\mathbf{E}_j) \text{s.t. } (\mathbf{X}, \boldsymbol{\delta}) \in T_{\mathbf{X}}S,\ \mathbf{X} \in \mathcal{M}_k \quad (16)$$

It is worth mentioning that the tangent bundle $T_{\mathbf{X}}S$ is as follows:

$$T_{\mathbf{X}}S = \{(\mathbf{X}, \boldsymbol{\delta}) | \mathbf{X} \in \mathcal{M}_k, \boldsymbol{\delta} \in T_{\mathbf{X}}\mathcal{M}_k, \langle\boldsymbol{\delta}, \boldsymbol{\delta}\rangle \leq \Delta\} \quad (17)$$

where $\Delta > 0$ is called the trust-region radius determining how far the movement can be made. RIST updates Δ iteratively by measuring the quotient as follows:

$$\phi_j = \frac{f(\mathbf{X}_j) - f(R_{\mathcal{M}}(\mathbf{X}_j))}{m(\mathbf{X}_j; \mathbf{0})) - m(R_{\mathcal{M}}(\mathbf{X}_j); \boldsymbol{\delta}_j)} \quad (18)$$

Depending on the value of ϕ_j, the new iterate $\mathbf{X}_j$ will be accepted or rejected. When ϕ_j is exceedingly small by $\phi_j < \mu_1$, the step-length and direction are not appropriate, the $\mathbf{X}_j$ should be rejected and the ϕ_j is reduced by a factor $d_1 < 1$. Moreover, the $\phi_j \in (\mu_1, \mu_2)$ demonstrates a quite appropriate model producing the acceptable $\mathbf{X}_j$ and radius ϕ_j. A third case is that the model is relatively appropriate by $\phi_j > \mu_2$, thus we accept $\mathbf{X}_j$ and enlarge ϕ by a factor $d_2 > 1$. After yielding $\mathbf{X}$, the updating process for $\mathbf{E}$ is similar to the Step 5 in Algorithm 2.

Algorithm 3 ROM: Riemann Optimization over $\mathcal{M}_k$

Input: k, $\Delta_{\text{max}} > 0$, $\Delta_0 \in (0, \Delta_{\text{max}})$, $\mathbf{X}_*$, $\mathbf{E}_*$.
Output: $\mathbf{X}$, $\mathbf{E}$
Initialize $\mathbf{X}_0 = \mathbf{X}_*$, $\mathbf{E}_0 = \mathbf{E}_*$, $\mu_1 = 0.25$, $\mu_2 = 0.75$, $d_1 = 0.5$, $d_2 = 3$.
for $j = 0, 1, \ldots$ **do**
  Compute $\mathbf{G} = \text{grad}\, f(\mathbf{X}_j)$ over $\mathcal{M}_k$ by Eq. (5).
  Let $\mathbf{H} = \text{Hess}\, f(\mathbf{X}_j)[\mathbf{G}]$ via (6) and compute

$$\boldsymbol{\delta}_j = \begin{cases} -\dfrac{\|\mathbf{G}\|_F^2}{\langle\mathbf{H}, \mathbf{G}\rangle}\mathbf{G} & \langle\mathbf{H}, \mathbf{G}\rangle > 0 \\ -\mathbf{G} & \langle\mathbf{H}, \mathbf{G}\rangle \leq 0 \end{cases}$$

  Update $\mathbf{X}_{j+1} = R(\mathbf{X}_j, \beta_j\boldsymbol{\delta}_j)$ on $\mathcal{M}_k$ by Eq. (10).
  Compute ϕ_j by Eq. (18).
  if $\phi_j < \mu_1$ **then**
    $\Delta_{j+1} = d_1\Delta_j$
    $\mathbf{X}_{j+1} = \mathbf{X}_j$
  else if $\mu_1 < \phi_j < \mu_2$ **then**
    $\Delta_{j+1} = \Delta_j$
  else
    $\Delta_{j+1} = \min(d_2\Delta_j, \Delta_{\text{max}})$
  end if
  Compute $\mathbf{E}_{j+1}$ via Eq. (14).
end for
return $\mathbf{X}$ and $\mathbf{E}$.

Complexity Analysis

From above description, we can see that the complexity of RIST algorithm is dominated by the computation of Riemannian gradient, Riemannian Hessian and the retraction operator. Both Riemannian gradient and Hessian involves computing matrix-vector product and requires $O(mnk)$ flops. Rank estimation and retraction operator are implemented by Randomized SVD (Halko, Martinsson, and Tropp 2011) and PROPACK (Larsen 2004) respectively, both with complexity $O(mnk)$. Overall, RIST consumes a computational complexity of $O(mnk)$. It is a significant complexity reduction since the estimated rank k is typically very small compared to the problem size $m \times n$. More significantly, the above complexity will reduce to $O((m+n)k)$ when the matrix is sufficiently sparse.

Convergence Analysis

The optimum for $\mathbf{E}$ can either be zero in MC or obtained by the closed-form in RPCA, this section considers the convex function f with respect to $\mathbf{X}$, namely $f(\mathbf{X})$.

Lemma 1. *(Absil, Mahony, and Sepulchre 2009) The sequence $\{\mathbf{X}^t\}$ generated by Algorithm 2 converges to the stationary point $\mathbf{X}_*$ with grad $f(\mathbf{X}_*) = 0$.*

Theorem 1. *Suppose $\boldsymbol{\delta}_* \in T_{\mathbf{X}_j}\mathcal{M}_k$ is the unique solution for the subproblem (15) at point X_j and the first-order necessary condition grad$f(\mathbf{X}_j)$ + Hess$f(\mathbf{X}_j)[\boldsymbol{\delta}_*] = 0$ is satisfied. Assume that the Hess$f(\mathbf{X}_j)$ is positive definite, Algorithm 3 has q-superlinear convergence.*

Proof. Suppose the γ is the unique minimizing geodesic satisfying $\gamma(0) = \mathbf{X}_j$ and $\gamma(1) = \mathbf{X}_{j+1}$. Let $P_\gamma^{1\leftarrow 0}$ denote the parallel translator along γ by sending the vector of $T_{\mathbf{X}_j}\mathcal{M}_k$ to the vector of $T_{\mathbf{X}_{j+1}}\mathcal{M}_k$ and $\gamma'(0) = \boldsymbol{\delta}_*$.

$$P_\gamma^{0\leftarrow 1}\text{grad}f(\mathbf{X}_{j+1}) = \text{grad}f(\mathbf{X}_j) + \int_0^1 \frac{d}{d\tau}P_\gamma^{0\leftarrow\tau}\boldsymbol{\xi}d\tau$$

$$= \int_0^1 \left(\frac{d}{d\tau}P_\gamma^{0\leftarrow\tau}\text{grad}f(\mathbf{X}_j) - \text{Hess}f(\mathbf{X}_j)[\boldsymbol{\delta}_*]\right)d\tau$$

Considering the fact that the parallel translation is an isometry, we have $\left\|\mathcal{P}_\gamma^{0\leftarrow 1}\text{grad}f(\mathbf{X}_{j+1})\right\| = \|\text{grad}f(\mathbf{X}_{j+1})\|$. Also we have $\|\text{grad}f(\mathbf{X}_{j+1})\|$

$$= \left\|\int_0^1 \left(\mathcal{P}_\gamma^{0\leftarrow\tau}\text{Hess}f(\gamma)\left[\gamma'\right] - \text{Hess}f(\mathbf{X}_j)[\boldsymbol{\delta}_*]\right)d\tau\right\|$$

$$\leqslant \|\boldsymbol{\delta}_*\|\int_0^1 (\|P_\gamma^{0\leftarrow\tau}\text{Hess}f(\gamma)P_\gamma^{\tau\leftarrow 0} - \text{Hess}f(\gamma)P_\gamma^{\tau\leftarrow 0}\|$$

$$+ \|\text{Hess}f(\gamma)\|\|P_\gamma^{\tau\leftarrow 0} - \mathbf{I}\| + \|\text{Hess}f(\gamma) - \text{Hess}f(\mathbf{X}_j)\|)d\tau$$

$$\leq c_0\|\boldsymbol{\delta}_*\|^2(\|\mathbf{X}_\infty^3\| + c_1)$$

where γ is a function of τ, namely, $\gamma = \gamma(\tau)$.

Since $\boldsymbol{\delta}_*$ satisfies the first order necessary condition grad$f(\mathbf{X}_j)$ + Hess$f(\mathbf{X}_j)[\boldsymbol{\delta}_*] = 0$, then

$$\boldsymbol{\delta}_* = -\mathbf{U}_0(\mathbf{U}_0^T[\text{Hess}f(\mathbf{X}_j)]\mathbf{U}_0)^{-1}\mathbf{U}_0^T\text{grad}f(\mathbf{X}_j)$$

where $\mathbf{U}_0$ is an arbitrary orthonormal basis for $T_{\mathbf{X}}S$ (Sun, Qu, and Wright 2015). Therefore, it follows that

$$\|\text{grad}f(\mathbf{X}_{j+1})\| \leq o(\|\mathbf{X}\|_\infty^3)\|\text{grad}f(\mathbf{X}_j)\|^2 \qquad (19)$$

Applying the Taylors theorem to $f(\mathbf{X}_j)$ yields

$$\|\text{grad}f(\mathbf{X}_j)\| = \frac{1}{2}\text{Hess}f(\mathbf{X}_*)(\mathbf{X}_j - \mathbf{X}_*)^2 + o(\|\mathbf{X}_j - \mathbf{X}_*\|^2)$$

Let $\lambda_{\min}$ and $\lambda_{\max}$ be the eigenvalues of Hess$f(\mathbf{X}_*)$, then there exist $c_2 < \lambda_{\min}$ and $c_3 > \lambda_{\max}$ such that

$$c_2\|\mathbf{X}_* - \mathbf{X}_j\| \leq \|\text{grad}\, f(\mathbf{X}_j)\| \leq c_3\|\mathbf{X}_* - \mathbf{X}_j\| \qquad (20)$$

Combining (19) and (20) gives $c_2\|\mathbf{X}_* - \mathbf{X}_{j+1}\| \leqslant \|\text{grad}f(\mathbf{X}_{j+1})\| \leqslant o(\|\mathbf{X}\|_\infty^3)c_3^2\|\mathbf{X}_* - \mathbf{X}_j\|^2$

Consequently, it can be deduced that $\|\mathbf{X}_* - \mathbf{X}_{j+1}\| \leq q\|\mathbf{X}_* - \mathbf{X}_j\|$ where $q = o(\|\mathbf{X}\|_\infty^3)c_3^2\|\mathbf{X}_j - \mathbf{X}_*\|/c_2$. □

Experiments

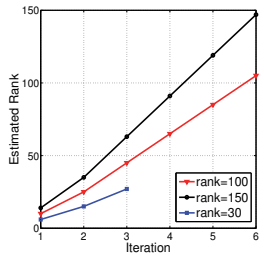
In this section, the tasks of Robust PCA and matrix completion are carried out to evaluate the proposed RIST algorithm. All comparison algorithms are implemented in Matlab and tested on a desktop computer with a 3.20 GHz CPU and 4.00 GB of memory.

Experiments on Matrix Completion

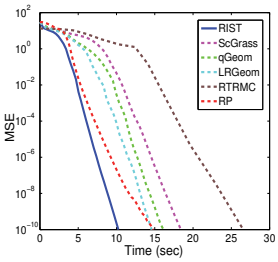
This section chooses five state-of-the-art manifold related completion algorithms including RTRMC (Boumal and Absil 2011), qGeom (Mishra, Apuroop, and Sepulchre 2012), RP (Tan et al. 2014), LRGeom (Vandereycken 2013) and ScGrass (Ngo and Saad 2012) for comparison.

Synthetic Experiments The low-rank matrix is generated $\mathbf{X} = \mathbf{L}\mathbf{R}^T \in \mathbb{R}^{m\times m}$ of rank k, where every entry in the matrix $\mathbf{L}$ or $\mathbf{R}$ follows the standard Gaussian distribution independently. We define the test set Ω by sampling $k(2m-k)\rho$ entries from $\mathbf{X}$ uniformly and the set size is $|\Omega| = k(2m-k)\rho$. Note that $m = 10000$ and ρ is the oversampling ratio that determines the number of entries that are observed. The parameters ρ and η of RIST are set as 1.5 and 0.04, respectively. We use the default parameters values for the comparison methods. The Mean Squared Error (MSE) is used as the comparison metric for synthetic experiments: MSE $= \|P_\Omega(\tilde{\mathbf{X}}) - P_\Omega(\mathbf{X})\|_F^2/|\Omega|$. For comparing the convergence of RIST with several baseline methods, we set rank $k = 15$, 25 and 35, respectively.

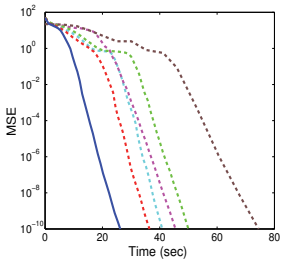
Fig. 1(a) first illustrates the numerical behavior of Algorithm 2 with respect to rank estimation. The rank estimated by RIST is 29, 103 and 148 receptively, which is approximate to the ground-truth rank setting (e.g., 30, 100, 150). More importantly, RIST takes less than six iterations to accurately estimate the ground-truth rank. Consequently, it can be concluded that RIST can accurately estimate the ground-truth rank with small number of iterations. In addition, performance evaluations in terms of MSE versus the run time are shown in Fig. 1(b), Fig. 1(c) and Fig. 1(d). RTRMC, qGeom, LRGeom and ScGrass optimize over fixed-rank manifold, which all need to predefine the rank in advance. We set the rank parameter of these comparison methods as the ground-truth, namely, 15, 25 and 35. It can be seen that RTRMC converges the slowest, while RP is faster than RTRMC, ScGrass, qGeom and LRGeom. Note that the RIST algorithm converges faster than other methods.



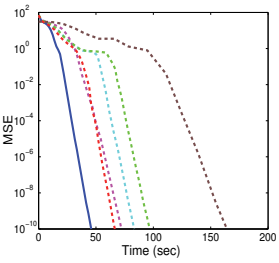
(a) estimated rank



(b) rank $k = 15$

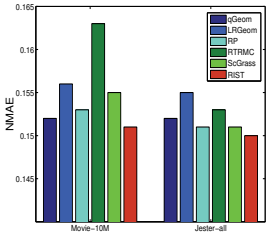


(c) rank $k = 25$

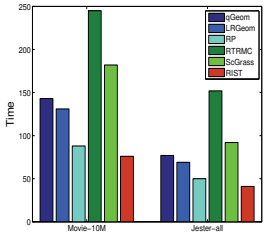


(d) rank $k = 35$

Figure 1: Performance of matrix completion methods on synthetic experiments: (a) shows the number of iterations required for estimating the rank. (b), (c) and (d) describe the results of MSE versus time.



(a) NMAE



(b) Time

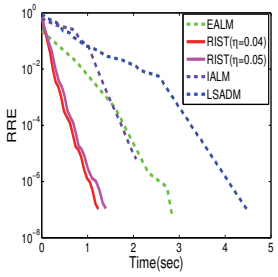
Figure 2: Performance of comparison methods on Jester-all and Movie-10M.

Collaborative Filtering We then test the proposed RIST algorithm on the real-world collaborative filtering task. Two collaborative filter datasets: the Jester-all dataset (Goldberg et al. 2001) and Movie-10M dataset (Herlocker et al. 1999) are used for the collaborative filtering. Among them, the Jester-all contains 4.1×10^6 ratings for 100 jokes from 73421 users, while the Movie-10M contains 10^7 ratings given by 69878 users on 10677 movies. The Normalized Mean Absolute Error (NMAE) is used as the comparison metric: $\text{NMAE} = \frac{\sum_{i,j\in\Omega}|\tilde{\mathbf{X}}_{ij}-\mathbf{X}_{ij}|}{(r_{max}-r_{min})|\Omega|}$, where $r_{\max}$ and $r_{\min}$ are the upper and the lower bounds for the ratings. 50% of ratings is randomly selected as the test set Ω with the size defined by $|\Omega|$. The estimated rank of Jester-all and Movie-10M by the proposed RIST algorithm is 19 and 16. For fair comparison, the same rank settings are assumed for ScGrass, LRGeom, qGeom, RTRMC and RP. The parameters η of RIST is 0.05. We use the default parameters values for other methods.

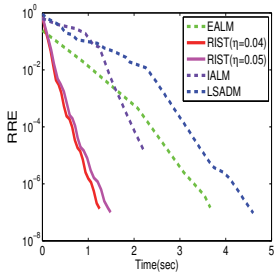
The averaged results of 20 independent experiments are reported in Fig. 2. RIST achieves the best NMAE values with least computational time, when compared with other methods. Noted that qGeom, RP and ScGrass perform comparable NMAE values with RIST on Jester-all and Movie-10M, but they consume more time to obtain such satisfactory results. These empirical results indicate that RIST is more efficient than competing algorithms while achieving the better or comparable NMAE for collaborative filtering.

Experiments on Robust PCA

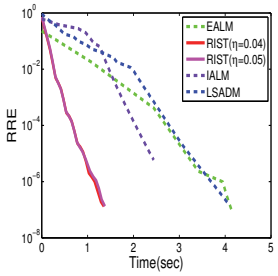
For a systematic evaluation, we compare the proposed RIST algorithm with three popular algorithms for Robust PCA problem, including EALM (Lin, Chen, and Ma 2010), IALM (Lin, Chen, and Ma 2010) and LSADM (Goldfarb, Ma, and Scheinberg 2013).



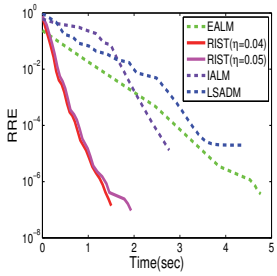
(a) rank $k = 0.06m$



(b) rank $k = 0.07m$



(c) rank $k = 0.08m$



(d) rank $k = 0.1m$

Figure 3: Performance of Robust PCA methods on synthetic experiments.

Synthetic Experiments Each synthetic data is characterized by the sum of low-rank matrix $\mathbf{X}$ and sparse matrix $\mathbf{E}$. For the sparse matrix $\mathbf{E}$, the percentage of nonzero entries are both set as 10%. Those non-zero entries are generated from a uniform distribution with the range $(-100, 100)$. The Relative Recovery Error (RRE) and time are used to measure the matrix recovery accuracy and efficiency: $\text{RRE} = \|\tilde{\mathbf{X}} - \mathbf{X}\|_F / \|\mathbf{X}\|_F$.

Performance evaluations in terms of RRE and computational time are shown in Figure 3, where m is set as 300×300 and the rank is defined as the percentage of matrix dimension. Note that RIST with different η generally converges faster than other three methods while LSADM per-

Table 1: Performance evaluation on synthetic data in terms of RRE and time.

size	rank	EALM		IALM		LSADM		RIST ($\eta = 0.05$)		RIST ($\eta = 0.04$)	
		Time	RRE	Time	RRE	Time	RRE	Time	RRE	Time	RRE
500	1%	11.0	3.4e-7	7.1	8.9e-8	49.7	8.4e-7	5.1	6.6e-8	5.0	**6.4e-8**
	2%	13.5	1.4e-7	8.7	5.6e-8	72.9	7.6e-7	8.2	4.4e-8	8.3	**4.3e-8**
	5%	13.8	5.1e-7	10.5	4.6e-6	77.6	2.7e-7	9.4	1.9e-7	9.8	**1.7e-7**
1000	1%	51.3	6.2e-7	19.5	9.1e-7	87.8	8.7e-7	17.0	7.4e-7	16.8	**7.3e-7**
	2%	51.8	1.0e-6	28.0	9.2e-7	92.9	5.5e-6	27.1	3.7e-7	26.8	**3.5e-7**
	5%	62.1	9.8e-7	30.6	3.3e-7	141.4	6.9e-7	30.3	2.8e-8	29.1	**2.7e-8**

forms the worst under three rank settings. Additionally, from larger matrices of 500×500 and 1000×1000 in Table 1, RIST outperforms all other three methods with fast convergence and comparable accuracy, both under the scenario of $\eta = 0.04$ and $\eta = 0.05$. IALM improves the convergence speed of EALM but achieves worse RRE, since the partial SVD in IALM usually produce approximate solutions. In contrast, LSADM consumes more running time to achieve the comparable RRE values with other methods.

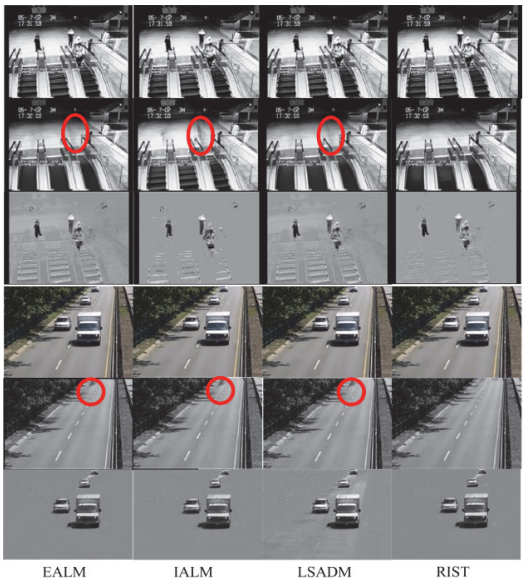


Figure 4: Subway Station and Highway video frames and their separated backgrounds and foregrounds.

Background Modeling Background modeling is an active application of RPCA problem, where the background sequence is modeled by a low-rank component $\mathbf{X}$ changing over time and the moving foreground objects $\mathbf{E}$ constitute the correlated sparse noises. Two surveillance videos including the Subway Station frames and Highway frames are used. Specifically, 195 frames with the resolution 160×130 are extracted from Subway Station frames and 90 frames with the resolution 320×240 are extracted from the Highway frames.

From Fig. 4 it is evident that IALM, EALM and LSADM fail to completely remove the moving shadows marked by the red circles. Especially, LSADM is affected by the variation of shadow and produces some white parts after the objects moving. In contrast, RIST is robust to the shadow changes and provides cleaner backgrounds without any remaining shadows. Moreover, the moving objects of Subway Station images separated by EALM and LSADM still involve some background shadow except human flows and moving escalators, which verifies that EALM and LSADM perform poorly than IALM and RIST. RIST algorithm is superior to IALM through more accurately capturing the moving people and separating the backgrounds.

Conclusion

Though Riemannian optimization based matrix recovery methods have shown superior scalability over convex relaxation methods, they are often degraded by inappropriate rank estimation in practice. Moreover, due to the neglect of the manifold boundary, their convergence analysis is not reliable. To alleviate these issues, this paper proposes an algorithm RIST that adopts Riemannian optimization to recover the matrix over low-rank algebraic variety. RIST represents an effective rank estimation strategy and utilizes the second-order geometric characterization, which achieves superlinear convergence rate. Both numerical experiments and theoretical analysis validate the effectiveness of RIST.

References

Absil, P.-A.; Mahony, R.; and Sepulchre, R. 2009. *Optimization algorithms on matrix manifolds*. Princeton University Press.

Agarwal, A.; Negahban, S.; and Wainwright, M. J. 2010. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, 37–45.

Boumal, N., and Absil, P.-a. 2011. Rtrmc: A riemannian trust-region method for low-rank matrix completion. In *Advances in neural information processing systems*, 406–414.

Bouwmans, T., and Zahzah, E. H. 2014. Robust pca via principal component pursuit: a review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding* 122:22–34.

Candès, E. J., and Recht, B. 2009. Exact matrix comple-

tion via convex optimization. *Foundations of Computational mathematics* 9(6):717–772.

Candès, E. J.; Li, X.; Ma, Y.; and Wright, J. 2011. Robust principal component analysis? *Journal of the ACM (JACM)* 58(3):11.

Goldberg, K.; Roeder, T.; Gupta, D.; and Perkins, C. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval* 4(2):133–151.

Goldfarb, D.; Ma, S.; and Scheinberg, K. 2013. Fast alternating linearization methods for minimizing the sum of two convex functions. *Mathematical Programming* 141(1-2):349–382.

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

Hazan, E. 2008. Sparse approximate solutions to semidefinite programs. In *LATIN 2008: Theoretical Informatics*. Springer. 306–316.

Herlocker, J. L.; Konstan, J. A.; Borchers, A.; and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 230–237. ACM.

Jaggi, M.; Sulovsk, M.; et al. 2010. A simple algorithm for nuclear norm regularized problems. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 471–478.

Jaggi, M. 2013. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 427–435.

Larsen, R. M. 2004. Propack-software for large and sparse svd calculations. *Available online. URL http://sun. stanford. edu/rmunk/PROPACK* 2008–2009.

Laue, S. 2012. A hybrid algorithm for convex semidefinite optimization. *arXiv preprint arXiv:1206.4608*.

Li, Q.; Niu, W.; Li, G.; Cao, Y.; Tan, J.; and Guo, L. 2015. Lingo: Linearized grassmannian optimization for nuclear norm minimization. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 801–809. ACM.

Lin, Z.; Chen, M.; and Ma, Y. 2010. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*.

Liu, R.; Lin, Z.; Wei, S.; and Su, Z. 2011. Solving principal component pursuit in linear time via l_1 filtering. *arXiv preprint arXiv:1108.5359*.

Mishra, B.; Apuroop, K. A.; and Sepulchre, R. 2012. A riemannian geometry for low-rank matrix completion. *arXiv preprint arXiv:1211.1550*.

Ngo, T., and Saad, Y. 2012. Scaled gradients on grassmann manifolds for matrix completion. In *Advances in Neural Information Processing Systems*, 1412–1420.

Papamakarios, G.; Panagakis, Y.; and Zafeiriou, S. 2011. Generalised scalable robust principal component analysis. *analysis* 58(3):11–37.

Recht, B. 2011. A simpler approach to matrix completion. *The Journal of Machine Learning Research* 12:3413–3430.

Schneider, R., and Uschmajew, A. 2015. Convergence results for projected line-search methods on varieties of low-rank matrices via lojasiewicz inequality. *SIAM Journal on Optimization* 25(1):622–646.

Sun, J.; Qu, Q.; and Wright, J. 2015. Complete dictionary recovery over the sphere. *arXiv preprint arXiv:1504.06785*.

Tan, M.; Tsang, I. W.; Wang, L.; Vandereycken, B.; and Pan, S. J. 2014. Riemannian pursuit for big matrix recovery. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 1539–1547.

Vandereycken, B. 2013. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization* 23(2):1214–1236.

Wang, Z.; Lai, M.-J.; Lu, Z.; Fan, W.; Davulcu, H.; and Ye, J. 2014. Rank-one matrix pursuit for matrix completion. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 91–99.

Yan, Y.; Mingkui, T.; Ivor, T.; Yi, Y.; Chengqi, Z.; and Qinfeng, S. 2015. Scalable maximum margin matrix factorization by active riemannian subspace search. In *International Joint Conference on Artificial Intelligence*, 3989–3994.