Proximal Riemannian Pursuit for Large-scale Trace-norm Minimization

Mingkui Tan[1], Shijie Xiao[2], Junbin Gao[3], Dong Xu[4], Anton van den Hengel[1], Qinfeng Shi[1]
[1]The School of Computer Science, University of Adelaide, Australia
[2]OmniVision Technologies Singapore Pte. Ltd., Singapore
[3]The University of Sydney Business School, University of Sydney, Australia
[4]School of Electrical and Information Engineering, University of Sydney, Australia

Abstract

Trace-norm regularization plays an important role in many areas such as computer vision and machine learning. When solving general large-scale trace-norm regularized problems, existing methods may be computationally expensive due to many high-dimensional truncated singular value decompositions (SVDs) or the unawareness of matrix ranks. In this paper, we propose a proximal Riemannian pursuit (PRP) paradigm which addresses a sequence of trace-norm regularized subproblems defined on nonlinear matrix varieties. To address the subproblem, we extend the proximal gradient method on vector space to nonlinear matrix varieties, in which the SVDs of intermediate solutions are maintained by cheap low-rank QR decompositions, therefore making the proposed method more scalable. Empirical studies on several tasks, such as matrix completion and low-rank representation based subspace clustering, demonstrate the competitive performance of the proposed paradigms over existing methods.

1. Introduction

Trace-norm regularization has widely appeared in many problems, such as matrix recovery (MR) [8, 43], robust principal component analysis (RPCA) [7], low-rank representation (LRR) [30, 55, 56], and robust multi-task learning [4]. Most of the trace-norm based problems can be formulated into the following general form [29, 33]:

$$\min_{\mathbf{X},\mathbf{E}} \ ||\mathbf{X}||_* + \lambda\Upsilon(\mathbf{E}), \ \text{s.t.}\ \mathcal{A}(\mathbf{X}) + \mathcal{B}(\mathbf{E}) = \mathbf{D}, \qquad (1)$$

where λ is a regularization parameter, $||\mathbf{X}||_*$ is the trace-norm (or the nuclear-norm) of a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, $\mathcal{A}$ and $\mathcal{B}$ are linear operators depending on specific applications [29], $\mathbf{D}$ represents data or observations, and $\mathbf{E}$ represents the fitting error. The minimization of problem (1) would encourage $\mathbf{X}$ to be low-rank [9, 12, 14]. In general, $\Upsilon(\mathbf{E})$ is a non-smooth regularizer on $\mathbf{E}$, such as the ℓ_1-norm regularization (*i.e.*, $||\mathbf{E}||_1$) or $\ell_{2,1}$-norm regularization (*i.e.*, $||\mathbf{E}||_{2,1}$) [7, 29, 30].

Problem (1) has been involved in many computer vision tasks recently, such as image restoration [19, 21, 34, 46], multi-label image classification problems [13, 10, 20], video segmentation [25, 60], and so on. Note that by removing the term $\Upsilon(\mathbf{E})$, problem (1) is reduced to a simple form:

$$\min_{\mathbf{X}} \ \|\mathbf{X}\|_* \ \text{s.t.}\ \mathcal{A}(\mathbf{X}) = \mathbf{D}, \qquad (2)$$

or a slightly relaxed *matrix lasso* problem [18, 50]:

$$\min_{\mathbf{X}} \ \|\mathbf{X}\|_* + \frac{\gamma}{2}\|\mathcal{A}(\mathbf{X}) - \mathbf{D}\|_2^2, \qquad (3)$$

where γ is a regularization parameter.

In the last decade, many algorithms have been proposed to solve problem (2) or (3) [17, 14, 24, 50, 58], for example, the singular value thresholding (SVT) method [6], augmented Lagrangian method (ALM) and alternating direction method (ADM) [27, 45, 48], proximal gradient (PG) [18], and accelerated proximal gradient (APG) [18, 50]. Due to the non-smoothness of $\Upsilon(\mathbf{E})$, the optimization of problem (1) is more challenging. Recently, aforementioned methods (*e.g.*, ADM and APG) have been extended to solve this problem by minimizing $\mathbf{X}$ and $\mathbf{E}$ in an alternative way [54, 28, 29].

These methods have shown great success in practice. However, their optimization usually requires many singular value decompositions (SVDs), which can be very expensive for large-scale problems. With continuation strategy and rank prediction techniques, the convergence can be accelerated by applying truncated SVDs [27, 29, 50]. However, the truncated SVDs are often cold-started, that is, when updating $\mathbf{X}$, one has to compute the SVD of a new intermediate matrix in order to compute the thresholding operations involved in the optimizations [35, 51, 33]. As a result, the computation cost can be very high on large-scale problems with large ranks [52]. Since variables between iterations may be very close to each other, a warm-start SVD could be applied to accelerate the speed [52], but this is not very stable and may incur convergence issues.

In this paper, we make the following contributions.

- We propose a *proximal Riemannian gradient* (PRG) method to address trace-norm regularized problems defined on $\mathcal{M}_{\leq r} = \{\mathbf{X} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{X}) \leq r\}$, and provide the convergence result of the algorithm. By exploiting geometries on $\mathcal{M}_{\leq r}$, the SVDs of intermediate solutions are maintained by cheap low-rank QR decompositions, making the method very scalable.
- To address general trace-norm regularized problems in (1), we present a simple proximal Riemannian pursuit (PRP) scheme, which addresses a sequence of subproblems defined on $\mathcal{M}_{\leq r}$, where r increases monotonically with simple update rules. Therefore, unlike existing fixed-rank methods [1, 5, 35, 26], this paradigm does not require the knowledge of matrix ranks.

2. Notations and Preliminaries

Let the superscript $^{\mathsf{T}}$ denote the transpose of a vector/matrix, $\mathbf{0}$ be a vector/matrix with all zeros, diag($\mathbf{v}$) be a diagonal matrix with diagonal elements equal to $\mathbf{v}$, $\langle \mathbf{A}, \mathbf{B}\rangle = \text{tr}(\mathbf{A}\mathbf{B}^{\mathsf{T}})$ be the inner product of $\mathbf{A}$ and $\mathbf{B}$, and $\|\mathbf{v}\|_p$ be the ℓ_p-norm of a vector $\mathbf{v}$. Let $\mathcal{A}$ be a linear operator with $\mathcal{A}^*$ being its adjoint operator. The operator $\max(\boldsymbol{\sigma}, \mathbf{v})$ operates on each dimension of $\boldsymbol{\sigma}$. Let $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^{\mathsf{T}}$ be the SVD of $\mathbf{X} \in \mathbb{R}^{m\times n}$. 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 $\|\mathbf{X}\|_F = \|\boldsymbol{\sigma}\|_2$. Lastly, for any convex function $\Omega(\mathbf{X})$, let $\partial\Omega(\mathbf{X})$ denote its subdifferential at $\mathbf{X}$.

We also need some basics of the Geometries of fixed-rank matrices and matrix varieties. Due to page limitation, more details are presented in the supplementary file.

Riemannian manifold fixed-rank matrices. The fixed rank-s matrices lie on a smooth submanifold defined below $\mathcal{M}_s = \{\mathbf{X} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{X}) = s\} = \{\mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^{\mathsf{T}} : \mathbf{U} \in \text{St}_s^m, \mathbf{V} \in \text{St}_s^n, ||\boldsymbol{\sigma}||_0 = s\}$, where $\text{St}_s^m = \{\mathbf{U} \in \mathbb{R}^{m\times s} : \mathbf{U}^{\mathsf{T}}\mathbf{U} = \mathbf{I}\}$ denotes the Stiefel manifold of $m \times s$ real and orthonormal matrices, and the entries in $\boldsymbol{\sigma}$ are in descending order [51]. The tangent space $T_{\mathbf{X}}\mathcal{M}_s$ at $\mathbf{X}$ is given by $T_{\mathbf{X}}\mathcal{M}_s = \{\mathbf{U}\mathbf{M}\mathbf{V}^{\mathsf{T}} + \mathbf{U}_p\mathbf{V}^{\mathsf{T}} + \mathbf{U}\mathbf{V}_p^{\mathsf{T}} : \mathbf{M} \in \mathbb{R}^{s\times s}, \mathbf{U}_p \in \mathbb{R}^{m\times s}, \mathbf{U}_p^{\mathsf{T}}\mathbf{U} = \mathbf{0}, \mathbf{V}_p \in \mathbb{R}^{n\times s}, \mathbf{V}_p^{\mathsf{T}}\mathbf{V} = \mathbf{0}\}$. Given $\mathbf{X} \in \mathcal{M}_s$ and $\mathbf{A}, \mathbf{B} \in T_{\mathbf{X}}\mathcal{M}_s$, by defining a metric $g_{\mathbf{X}}(\mathbf{A}, \mathbf{B}) = \langle \mathbf{A}, \mathbf{B}\rangle$, $\mathcal{M}_s$ is a **Riemannian manifold** by restricting $\langle \mathbf{A}, \mathbf{B}\rangle$ to the *tangent bundle* [2]. Here, the *tangent bundle* is defined as the disjoint union of all tangent spaces $T\mathcal{M}_s = \bigcup_{\mathbf{X}\in\mathcal{M}_s}\{\mathbf{X}\} \times T_{\mathbf{X}}\mathcal{M}_s$. The norm of a tangent vector $\boldsymbol{\zeta}_{\mathbf{X}} \in T_{\mathbf{X}}\mathcal{M}_s$ at $\mathbf{X}$ is defined as $||\boldsymbol{\zeta}_{\mathbf{X}}|| = \sqrt{\langle\boldsymbol{\zeta}_{\mathbf{X}}, \boldsymbol{\zeta}_{\mathbf{X}}\rangle}$.

As $\mathcal{M}_s$ is embedded in $\mathbb{R}^{m\times n}$, the Riemannian gradient of f at $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^{\mathsf{T}}$, denoted by grad$f(\mathbf{X})$, is given as the orthogonal projection of the gradient of $\nabla f(\mathbf{X})$ onto the tangent space $T_{\mathbf{X}}\mathcal{M}_s$.

$$P_{T_{\mathbf{X}}\mathcal{M}_s}(\mathbf{Z}) : \mathbf{Z} \mapsto P_U\mathbf{Z}P_V + P_U^{\perp}\mathbf{Z}P_V + P_U\mathbf{Z}P_V^{\perp}. \tag{4}$$

where $P_U = \mathbf{U}\mathbf{U}^{\mathsf{T}}$ and $P_U^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\mathsf{T}}$. Letting $\mathbf{G} = \nabla f(\mathbf{X})$ be the gradient of $f(\mathbf{X})$, it follows that

$$\text{grad}f(\mathbf{X}) = P_{T_{\mathbf{X}}\mathcal{M}_s}(\mathbf{G}). \tag{5}$$

Moreover, define $P_{T_{\mathbf{0}}\mathcal{M}_s}(\mathbf{Z}) = \mathbf{0}$ when $\mathbf{X} = \mathbf{0}$.

Varieties of low-rank matrices [44]. Now we consider the closure of $\mathcal{M}_r$, which is defined by

$$\mathcal{M}_{\leq r} = \{\mathbf{X} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{X}) \leq r\}, \tag{6}$$

which is a real-algebraic variety. Let $\text{ran}(\mathbf{X})$ be the column space of $\mathbf{X}$. In the singular points where $\text{rank}(\mathbf{X}) = s < r$, we will construct search directions in the tangent cone [44] (instead of the tangent space)

$$T_{\mathbf{X}}\mathcal{M}_{\leq r} = T_{\mathbf{X}}\mathcal{M}_s \oplus \{\Xi_{r-s} \in \mathcal{U}^{\perp} \otimes \mathcal{V}^{\perp}\}, \tag{7}$$

where $\mathcal{U} = \text{ran}(\mathbf{X})$ and $\mathcal{V} = \text{ran}(\mathbf{X}^{\mathsf{T}})$, and Ξ_{r-s} is a best rank-$(r-s)$ approximation of $\mathbf{G} - P_{T_{\mathbf{X}}\mathcal{M}_s}(\mathbf{G})$ which can be cheaply computed with truncated SVD of rank $(r - s)$. This implies that a tangent vector on $T_{\mathbf{X}}\mathcal{M}_{\leq r}$ can be represented by

$$\boldsymbol{\xi} = \mathbf{U}\mathbf{M}\mathbf{V}^{\mathsf{T}} + \mathbf{U}_p\mathbf{V}^{\mathsf{T}} + \mathbf{U}\mathbf{V}_p^{\mathsf{T}} + \mathbf{U}_s\text{diag}(\boldsymbol{\sigma}_s)\mathbf{V}_s^{\mathsf{T}}, \tag{8}$$

where $\Xi_{r-s} = \mathbf{U}_s\text{diag}(\boldsymbol{\sigma}_s)\mathbf{V}_s^{\mathsf{T}}$. Let grad$f(\mathbf{X}) \in T_{\mathbf{X}}\mathcal{M}_{\leq r}$ be the projection of $\mathbf{G}$ on $T_{\mathbf{X}}\mathcal{M}_{\leq r}$. It can be computed by

$$\text{grad}f(\mathbf{X}) = P_{T_{\mathbf{X}}\mathcal{M}_s}(\mathbf{G}) + \Xi_{r-s}. \tag{9}$$

Retraction. Given a search direction $\boldsymbol{\xi} \in T_{\mathbf{X}}\mathcal{M}_{\leq r}$, the retraction finds the best approximation by a matrix with rank at most r as measured in terms of the Frobenius norm,

$$R_{\mathbf{X}}^{\leq r}(\boldsymbol{\xi}) = \arg\min\nolimits_{\mathbf{Y}\in\mathcal{M}_{\leq r}} ||\mathbf{Y} - (\mathbf{X} + \boldsymbol{\xi})||_F. \tag{10}$$

Algorithm 1: Computation of $R_{\mathbf{X}}^{\leq r}(\boldsymbol{\xi})$.

```
Require: $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^{\mathsf{T}} \in \mathcal{M}_{\leq r}$, the tangent vector
    $\boldsymbol{\xi} = \mathbf{U}\mathbf{M}\mathbf{V}^{\mathsf{T}} + \mathbf{U}_p\mathbf{V}^{\mathsf{T}} + \mathbf{U}\mathbf{V}_p^{\mathsf{T}} + \mathbf{U}_s\text{diag}(\boldsymbol{\sigma}_s)\mathbf{V}_s^{\mathsf{T}}$.
1: Compute $(\mathbf{Q}_u, \mathbf{R}_u)$=qr$(\mathbf{U}_p, 0)$, $(\mathbf{Q}_v,\mathbf{R}_v)$=qr$(\mathbf{V}_p,0)$.
2: Let $\mathbf{Z} = [\text{diag}(\boldsymbol{\sigma}) + \mathbf{M}\ \ \mathbf{R}_v^{\mathsf{T}}; \mathbf{R}_u\ \ \mathbf{0}]$.
3: Compute $(\mathbf{U}_z, \boldsymbol{\sigma}_z, \mathbf{V}_z) = \text{svd}(\mathbf{Z})$.
4: Let $\bar{\boldsymbol{\sigma}} = [\boldsymbol{\sigma}_z; \boldsymbol{\sigma}_s]$.
5: Let $\bar{\mathbf{U}}$=$[[\mathbf{U}\ \ \mathbf{Q}_u]\mathbf{U}_z\ \mathbf{U}_s]$, $\bar{\mathbf{V}}$=$[[\mathbf{V}\ \ \mathbf{Q}_v]\mathbf{V}_z\ \ \mathbf{V}_s]$.
6: Arrange $\bar{\boldsymbol{\sigma}}$ in descending order, and $\bar{\mathbf{U}}$ and $\bar{\mathbf{V}}$ accordingly.
7: Let $\mathbf{U}_+ = \bar{\mathbf{U}}(:, 1 : r)$, $\mathbf{V}_+ = \bar{\mathbf{V}}(:, 1 : r)$ and
   $\boldsymbol{\sigma}_+ = \bar{\boldsymbol{\sigma}}(1 : r)$.
8: Output $R_{\mathbf{X}}^{\leq r}(\boldsymbol{\xi}) = \mathbf{U}_+\text{diag}(\boldsymbol{\sigma}_+)\mathbf{V}_+^{\mathsf{T}}$.
```

In general, problem (10) can be addressed by performing truncated SVD on $\mathbf{X}+\boldsymbol{\xi}$, which, however, can be very expensive for high-dimensional matrices when r is large. Fortunately, as summarized in the following, by exploiting geometries over $\mathcal{M}_{\leq r}$, $R^{\leq r}_{\mathbf{X}}(\boldsymbol{\xi})$ can be efficiently computed by slightly modifying the retraction on fixed-rank manifold $\mathcal{M}_r$ (see details in [51]).

Remark 1. *$R^{\leq r}_{\mathbf{X}}(\boldsymbol{\xi})$ can be efficiently computed in Algorithm 1 with efficient QR decompositions on low rank matrices $\mathbf{U}_p$ and $\mathbf{V}_p$. The corresponding time complexity is $14(m+n)r^2+C_{SVD}r^3$, where $r \ll \min(m,n)$ and C_{SVD} is a moderate constant (say less than 200) [51].*

Most Riemannian-optimization based algorithms focus on fixed-rank manifolds $\mathcal{M}_r$ [1, 5, 35, 26]. In the trace-norm minimization, the rank degeneration (i.e. $\text{rank}(\mathbf{X}) < r$) may happen and even inevitable. Therefore, the introduction of $\mathcal{M}_{\leq r}$ is important and necessary in the trace-norm minimization of our paper. Note that the fixed-rank manifold $\mathcal{M}_r$ is open, thus the manifold properties break down at the boundary where $\text{rank}(\mathbf{X}) < r$, making the convergence analysis of the algorithm difficult accordingly [44].

3. Proximal Riemannian pursuit

In this paper, similar to [54, 28], we solve a slightly relaxed form of (1) with the equality constraint replaced with a penalty term:

$$\min_{\mathbf{X},\mathbf{E}} \ ||\mathbf{X}||_* + \lambda\Upsilon(\mathbf{E}) + \frac{\gamma}{2}||\mathcal{A}(\mathbf{X})+\mathcal{B}(\mathbf{E})-\mathbf{D}||_F^2, \qquad (11)$$

where γ is often a large penalty parameter. For convenience, let us define

$$\Psi(\mathbf{X},\mathbf{E}) = ||\mathbf{X}||_* + \lambda\Upsilon(\mathbf{E}) + \frac{\gamma}{2}||\mathcal{A}(\mathbf{X})+\mathcal{B}(\mathbf{E})-\mathbf{D}||_F^2.$$

Problem (11) can be addressed by proximal gradient (PG) method and accelerated proximal gradient (APG) [54, 28]. However, they are not computationally efficient due to many truncated SVDs. To address this, in this work, we propose to address problem (11) by iteratively solving a series of subproblems (indexed by $t = 1,\ldots,T$) with *progressively relaxed* rank constraint on $\mathbf{X}$. Specifically, each of the subproblems is in the form of

$$\min\nolimits_{\mathbf{X},\mathbf{E}} \ \Psi(\mathbf{X},\mathbf{E}), \ \text{ s.t. } \text{rank}(\mathbf{X}) \leq r, \qquad (12)$$

where the rank constraint is progressively relaxed with $r := t\kappa$ when t increases. The parameter κ is an integer that is *several times smaller* than the true rank, and a simple and reasonable setting of this parameter will be discussed later.

The proposed paradigm is referred to as the proximal Riemannian pursuit (PRP) method and is illustrated in Algorithm 2, where a continuation strategy is applied for γ to accelerate the convergence as in [54]. The key step of the proposed approach is to address the subproblem in (12) which is defined on $\mathcal{M}_{\leq r}$. In this paper, we solve problem (12) by a proximal Riemannian gradient method.

Algorithm 2: General PRP scheme for solving problem (1).

```
Input: Parameters $\kappa$, $\lambda$, $\gamma^0$, $\gamma^{\text{tar}}$, and $\chi \in (1,+\infty)$.
Initialize $\mathbf{X}^0 = \mathbf{0}$ and $\mathbf{E}^0 = \mathbf{0}$;
for $t = 1 : T$ do
    Let $r := r + \kappa$ and $\gamma^t \leftarrow \max(\gamma^{t-1}\chi, \gamma^{\text{tar}})$.
    Update $(\mathbf{X}^t, \mathbf{E}^t)$ by addressing (12) with $r$ and $\gamma^t$;
    Terminate if the stopping condition is achieved.
```

Before continuing, we discuss the determination of κ and stopping conditions.

Determination of κ. Let $\boldsymbol{\sigma}$ be the singular vector of $\mathcal{A}^*(\mathbf{D})$ in descending order. We choose κ such that

$$\sigma_i \geq \eta\,\sigma_1, \ \ \forall i \leq \kappa, \qquad (13)$$

where $\eta \in (0.5, 1]$. In other words, κ denotes the number of sufficiently large singular values of $\mathcal{A}^*(\mathbf{D})$.

Besides the progressive update of r, one may choose a large κ such that $\kappa > \text{rank}(\mathbf{X}^*)$, where $\mathbf{X}^*$ is the optimal solution of (1). However, this strategy has two drawbacks. First, the $\mathbf{X}^*$ is unknown, thus the setting of r is difficult. Second, if r is too large, it may incur severely ill-conditioning issues [49] and the computational complexity will increase dramatically.

Stopping condition: Since PRP increase the rank by κ iteratively, anyway it will be stopped in limited steps due to limited size of $\mathbf{X}$, but we may stop it earlier in practice. Basically, we can stop the algorithm if the objective value cannot be decreased significantly at some iteration.

Remark 2. *Let $\{\mathbf{X}^t, \mathbf{E}^t\}$ be the sequence generated by Algorithm 2. Then we have*

$$\Psi(\mathbf{X}^t,\mathbf{E}^t) \leq \Psi(\mathbf{X}^{t-1},\mathbf{E}^{t-1}) - \beta||\boldsymbol{\Xi}^{t-1}_{\kappa}||_F^2, \qquad (14)$$

where β is some number, $\boldsymbol{\Xi}^{t-1}_{\kappa}$ can be computed according to equation (7).

The proof relies on some results from later sections, and can be found in the supplementary file. According to Remark 2, if $||\boldsymbol{\Xi}^{t-1}_{\kappa}||_F^2$ is very small, then there is no need to proceed. We therefore set the convergence condition as follows:

$$\tfrac{\Psi(\mathbf{X}^{t-1},\mathbf{E}^{t-1})-\Psi(\mathbf{X}^{t},\mathbf{E}^{t})}{\kappa\Psi(\mathbf{X}^{t-1},\mathbf{E}^{t-1})} \leq \varepsilon, \qquad (15)$$

where ε denotes a tolerance value. In practice, a large γ may be required to prevent from the solution bias incurred by the regularization. However, in this case, the optimal solution $\mathbf{X}^*$ may not be an exact low-rank matrix. In this case, the early stopping will help to obtain a low-rank solution, thus it is very important.

4. Proximal Riemannian gradient methods

In this section, we present the proximal Riemannian gradient (PRG) method to address problem (12),[1] which extends the classical proximal method over vector space to $\mathcal{M}_{\leq r}$ [39, 50]. For simplicity, we first consider a simpler case where $\Upsilon(\mathbf{E})$ is not considered.

4.1. Case 1: $\lambda = 0$

When $\Upsilon(\mathbf{E})$ is not considered, problem (12) is equivalent to the following problem

$$\min_{\mathbf{X}\in\mathcal{M}_{\leq r}} \ ||\mathbf{X}||_* + \tfrac{\gamma}{2}||\mathcal{A}(\mathbf{X}) - \mathbf{D}||_F^2, \qquad (16)$$

which is known as the matrix lasso problem [50]. For convenience, let us define

$$f(\mathbf{X}) := \frac{\gamma}{2}||\mathcal{A}(\mathbf{X}) - \mathbf{D}||_F^2.$$

To introduce the proximal method on $\mathcal{M}_{\leq r}$, similarly as in [27, 50], we introduce a local model of $\Psi(\mathbf{X})$ on the tangent cone $T_{\mathbf{X}}\mathcal{M}_{\leq r}$ around $\mathbf{Y} \in \mathcal{M}_{\leq r}$ but keeping $||\mathbf{X}||_*$ intact as follows:

$$m_L(\mathbf{Y}; \mathbf{X}, \boldsymbol{\xi}) := ||\mathbf{X}||_* + Q(\mathbf{X}), \qquad (17)$$

$$Q(\mathbf{X}) := f(\mathbf{Y}) + \langle \text{grad} f(\mathbf{Y}), \boldsymbol{\xi}\rangle + \frac{L}{2}\langle \boldsymbol{\xi}, \boldsymbol{\xi}\rangle, \qquad (18)$$

where $\boldsymbol{\xi} \in T_{\mathbf{Y}}\mathcal{M}_{\leq r}$ and $\mathbf{X} := \mathbf{Y} + \boldsymbol{\xi}$, and L is the Lipschitz constant. Note that the above local model is very different from that in classical proximal gradient methods (*e.g.,* [39, 50]) in the sense that the operation $\mathbf{X} := \mathbf{Y} + \boldsymbol{\xi}$ is restricted on the tangent cone $T_{\mathbf{Y}}\mathcal{M}_{\leq r}$. On the tangent cone, it is valid for the operation $\mathbf{Y} + \boldsymbol{\xi}$ for any $\boldsymbol{\xi} \in T_{\mathbf{Y}}\mathcal{M}_{\leq r}$.

With the introduction of the local model, the proximal Rimannian gradient method addresses problem (16) in an iterative way (similar to [39, 50]), which updates $\mathbf{X}_k$ in the kth iteration by minimizing $m_L(\mathbf{X}_{k-1}; \mathbf{X}, \boldsymbol{\xi})$ on $\mathcal{M}_{\leq r}$, *i.e.,*

$$\mathbf{X}_k = \arg\min_{\mathbf{X}\in\mathcal{M}_{\leq r}} \ m_L(\mathbf{Y}; \mathbf{X}, \boldsymbol{\xi}), \qquad (19)$$

where $\mathbf{Y}$ is set to $\mathbf{X}_{k-1}$ and L is set to L_k. Let $T_L(\mathbf{Y})$ be the minimizer of problem (19), i.e. $T_L(\mathbf{Y}) := \arg\min_{\mathbf{X}\in\mathcal{M}_{\leq r}} m_L(\mathbf{Y}; \mathbf{X}, \boldsymbol{\xi})$. Then it can be computed by first computing the solution to $\arg\min_{\mathbf{X}\in\mathcal{M}_{\leq r}} \ Q(\mathbf{X})$, and then performing a singular value thresholding operation on the solution. Let $R_{\mathbf{Y}}(\boldsymbol{\xi}) = \arg\min_{\mathbf{X}\in\mathcal{M}_{\leq r}} \ Q(\mathbf{X})$. The computation of $R_{\mathbf{Y}}(\boldsymbol{\xi})$ and $T_L(\mathbf{Y})$ is shown in the following lemma whose proof is available in the supplementary file.

Lemma 1. *Given $\mathbf{Y}$ and the parameter L, $R_{\mathbf{Y}}(\boldsymbol{\xi})$ can be computed by equation (10) with $\boldsymbol{\xi} = -\text{grad}f(\mathbf{Y})/L$. Denoting the SVD of $R_{\mathbf{Y}}(\boldsymbol{\xi})$ as $R_{\mathbf{Y}}(\boldsymbol{\xi})$=$\mathbf{U}_+ \text{diag}(\boldsymbol{\sigma}_+)\mathbf{V}_+^\top$, we have $T_L(\mathbf{Y}) = \mathbf{U}_+ \text{diag}(\max(\boldsymbol{\sigma}_+ - 1/L, 0))\mathbf{V}_+^\top$.*

From Lemma 1, $R_{\mathbf{Y}}(\boldsymbol{\xi})$ is exactly the Retraction $R_{\mathbf{X}}^{\leq r}(\boldsymbol{\xi})$, and it can be cheaply computed according to Algorithm 1 whose complexity is $14(m+n)r^2 + C_{SVD}r^3$, where $r \ll \min(m, n)$. Since the decomposition $R_{\mathbf{Y}}(\boldsymbol{\xi}) = \mathbf{U}_+\text{diag}(\boldsymbol{\sigma}_+)\mathbf{V}_+^\top$ can be maintained by QR decompositions on low-rank matrices, we do not have to compute the truncated SVDs as in the classical proximal gradient methods [39, 50]. After obtaining $R_{\mathbf{Y}}(\boldsymbol{\xi})$, $T_L(\mathbf{Y})$ can be easily computed. The general scheme of PRG is shown in Algorithm 3.

Algorithm 3: Proximal Riemannian gradient method for solving problem (16).

```
Input: $\mathbf{X}_0$, parameter $\gamma$ and $r$, stopping tolerance $\epsilon$.
for $k = 1, ..., K$ do
    Compute grad$f(\mathbf{X}_{k-1})$;
    Let $\boldsymbol{\xi}_k = -\text{grad}f(\mathbf{X}_{k-1})/L_k$ where $L_k$ is determined by Armijo line search in (20);
    Set $\mathbf{X}_k = T_{L_k}(\mathbf{X}_{k-1})$;
    Terminate if stopping conditions are achieved;
Return $\mathbf{X}_k$.
```

Determining L_k: In practice, it is critical to find a good parameter L_k to make a sufficient decrease of the objective. Let $\Psi(\mathbf{X}) = ||\mathbf{X}||_* + \frac{\gamma}{2}||\mathcal{A}(\mathbf{X}) - \mathbf{D}||_F^2$ be the objective function. In the kth iteration, given a descent direction $\boldsymbol{\zeta}_k \in T_{\mathbf{X}_k}\mathcal{M}_{\leq r}$, and L_k can be determined using Armijo line search to satisfy

$$\Psi(T_{L_k}(\mathbf{X}_k)) \leq \Psi(\mathbf{X}_k) + \beta\langle \text{grad}f(\mathbf{X}_k), \boldsymbol{\zeta}_k\rangle / L_k, \qquad (20)$$

where $\beta \in (0, 1)$. Here, $1/L_k$ can be considered as the step size. The existence of L_k is guaranteed by the following lemma, whose proof is available in the supplementary file.

Lemma 2. *Let $\mathbf{X}_k \in \mathcal{M}_{\leq r}$, and $\boldsymbol{\zeta}_k \in T_{\mathbf{X}}\mathcal{M}_{\leq r}$ be a descent direction. Then there exists an L_k that satisfies the condition in (20).*

In the following, we discuss the convergence and stopping conditions of the algorithm.

Convergence and optimality: Optimization methods on Riemannian manifolds are often locally convergent. Whereas, for PRG, the limit point $\mathbf{X}^*$ will be a global solution if rank$(\mathbf{X}^*) < r$, as shown in the following proposition whose proof can be found in the supplementary file.

Proposition 1. *Let $\{\mathbf{X}_k\}$ be an infinite sequence of iterates generated by Algorithm 3. Then every accumulation point of $\{\mathbf{X}_k\}$ is a critical point of f over $\mathcal{M}_{\leq r}$. Furthermore, $\lim_{k\to\infty} ||\text{grad}f(\mathbf{X}_k) + \boldsymbol{\zeta}||_F = 0$, where $\boldsymbol{\zeta}$ is the subdifferential of $||\mathbf{X}||_*$ at $\mathbf{X}$ [18]. If rank$(\mathbf{X}^*) < r$, then we have $\nabla f(\mathbf{X}^*) + \boldsymbol{\zeta} = \mathbf{0}$,* i.e., *$\mathbf{X}^*$ is a global optimum to (16).*

[1] In fact, $\mathcal{M}_{\leq r}$ is a closure of the Riemannian submanifold $\mathcal{M}_r$. We abuse "Riemannian" here for simplicity.

Stopping conditions. A natural stopping condition would be $||\text{grad}f(\mathbf{X}_k) + \boldsymbol{\zeta}||_F \le \epsilon$ with some predefined ϵ. In practice, we do not have to solve the subproblem exactly. For simplicity, we stop PRG early if the following condition is achieved:

$$\frac{(\Psi(\mathbf{X}_{k-1})-\Psi(\mathbf{X}_k))}{\Psi(\mathbf{X}_{k-1})} \le \epsilon, \tag{21}$$

where ϵ denotes a tolerance value.

4.2. Case 2: $\lambda \neq 0$

Now, we are ready to extend PRG to minimize problem (12) in which $\lambda \neq \mathbf{0}$. Following [29], we optimize the two variables $\mathbf{X}$ and $\mathbf{E}$ using an alternating approach. Let the pair $(\mathbf{X}_k, \mathbf{E}_k)$ denote the variables obtained from the k-iteration. At the $(k+1)$th iteration, we update $\mathbf{X}$ and $\mathbf{E}$ as below.

To update $\mathbf{X}$, we fix $\mathbf{E} = \mathbf{E}_k$ and define a local model of $\Psi(\mathbf{X}, \mathbf{E})$ on the tangent cone $T_{\mathbf{X}}\mathcal{M}_{\le r}$ around $\mathbf{X}_k \in \mathcal{M}_{\le r}$:

$$\begin{aligned} m_L(\mathbf{X}; \mathbf{X}_k, \mathbf{E}_k, \boldsymbol{\xi}) := {} & ||\mathbf{X}||_* + f(\mathbf{X}_k, \mathbf{E}_k) \\ & + \langle \text{grad}f(\mathbf{X}_k, \mathbf{E}_k), \boldsymbol{\xi} \rangle + L/2 \langle \boldsymbol{\xi}, \boldsymbol{\xi} \rangle, \end{aligned}$$

where $\boldsymbol{\xi} \in T_{\mathbf{X}_k}\mathcal{M}_{\le r}$ and L is the Lipschitz constant. Since $\boldsymbol{\xi} \in T_{\mathbf{X}_k}\mathcal{M}_{\le r}$, it is valid to have $\mathbf{X} = \mathbf{X}_k + \boldsymbol{\xi}$. Let $T_L(\mathbf{X}_k, \mathbf{E}_k)$ be the minimizer of $m_L(\mathbf{X}; \mathbf{X}_k, \mathbf{E}_k, \boldsymbol{\xi})$ subjected to rank$(\mathbf{X}) \le r$. Similar to Lemma 1, $T_L(\mathbf{X}_k, \mathbf{E}_k)$ can be computed with two steps, where L can be determined by Armijo line search to make a sufficient decrease of the objective.

To update $\mathbf{E}$, we fix $\mathbf{X} = \mathbf{X}_{k+1}$ and solve a problem:

$$\min_{\mathbf{E}} \ \ \lambda \Upsilon(\mathbf{E}) + \tfrac{\gamma}{2} ||\mathcal{A}(\mathbf{X}_{k+1}) + \mathcal{B}(\mathbf{E}) - \mathbf{D}||_F^2. \tag{22}$$

Solving problem (22) with general $\mathcal{B}$ may be difficult. Fortunately, for MR and LRR, where $\mathcal{B}(\mathbf{E}) = \mathbf{E}$ and $\Upsilon(\mathbf{E})$ is either $\|\mathbf{E}\|_1$ or $\|\mathbf{E}\|_{2,1}$, problem (22) has a closed-form solution. Let us define $\mathbf{B}_k = \mathbf{D} - \mathcal{A}(\mathbf{X}_{k+1})$. Then $\mathbf{B}_k$ is a vector for MR and a matrix in the form of $\mathbf{B}_k = [\mathbf{b}_1^k, \ldots, \mathbf{b}_n^k]$ for LRR. The closed-form solution, denoted by $S_\lambda(\mathbf{B}_k)$, can be found in supplementary file. In cases where the problem (22) cannot be solved in closed-form, one may adopt iterative procedures to solve it.

The detailed algorithm, which is referred to as robust PRG (PRG(R)), is shown in Algorithm 4. Due to the possible ill-conditioning issues,[2] we again apply a homotopy continuation technique for λ to accelerate the convergence speed. Starting from an initial guess λ_0, we set $\lambda_k = \min(\lambda_0 \rho^{k-1}, \lambda)$ and compute $\mathbf{E}_k = S_{\lambda_k}(\mathbf{X}_k, \mathbf{E}_{k-1})$, where ρ is chosen from $(0, 1)$. Clearly, λ_k is non-increasing w.r.t. the iteration index k.

We discuss the convergence as follows.

Algorithm 4: Robust PRG for solving problem (12).

```
Input: Initial $(\mathbf{X}_0, \mathbf{E}_0)$, parameter $\lambda$, $\gamma$ and $r$, initial
       $\lambda_0 > \lambda$, $\rho \in (0, 1)$, tolerance $\epsilon$.
for $k = 1, ..., K$ do
    Let $\lambda_k = \max(\lambda_{k-1}\rho, \lambda)$;
    Compute grad$f(\mathbf{X}_{k-1}, \mathbf{E}_{k-1})$ by (5) or (9);
    Choose $L_k$ by Armijo line search. Set
    $\mathbf{X}_k = T_{L_k}(\mathbf{X}_{k-1}, \mathbf{E}_{k-1})$;
    Compute $\mathbf{E}_k = S_{\lambda_k}(\mathbf{B}_{k-1})$ with
    $\mathbf{B}_{k-1} = \mathbf{D} - \mathcal{A}(\mathbf{X}_k)$;
    Terminate if stopping conditions are achieved;
Return $(\mathbf{X}_k, \mathbf{E}_k)$.
```

Proposition 2. *Let* $\Psi(\mathbf{X}_k, \mathbf{E}_k) = ||\mathbf{X}_k||_* + \lambda_k \Upsilon(\mathbf{E}_k) + f(\mathbf{X}_k, \mathbf{E}_k)$*, and* $\{(\mathbf{X}_k, \mathbf{E}_k)\}$ *be an infinite sequence of iterates generated by Algorithm 4. It follows that* $\Psi(\mathbf{X}_{k+1}, \mathbf{E}_{k+1}) \le \Psi(\mathbf{X}_k, \mathbf{E}_k)$*, and* $\{(\mathbf{X}_k, \mathbf{E}_k)\}$ *converges to a limit point* $(\mathbf{X}^*, \mathbf{E}^*)$.

The proof of Proposition 2 can be found in the supplementary file. The stopping condition in (21) can be extended to PRG(R) by replacing $\Psi(\mathbf{X})$ with $\Psi(\mathbf{X}, \mathbf{E})$.

4.3. Complexity analysis

The complexity of PRP includes two main folds, *i.e.*, the computation of Ξ_κ^t which can be done by truncated SVD of rank κ and the subproblem optimization by PRG or PRG(R). Here, we focus on the complexity of PRP on MR. At the tth iteration of PRP, the complexity of PRG or PRG(R) is $O((m+n)(\kappa t)^2 + l\kappa t)$, where $\kappa t \le r + \kappa$. For sufficiently sparse matrices like in MR, the truncated SVD of rank κ in PRP can be completed in $O((m+n)\kappa)$ using PROPACK [23]; while the truncated SVD in existing proximal gradient based methods takes $O((m+n)r)$, where κ is several times smaller than r. The complexity comparison on LRR and RPCA can be found in the supplementary file.

4.4. Parameter settings

For convenience of parameter setting, we suggest choosing the penalty parameter γ in (12) according to $\gamma = 1/(\nu\sigma_1)$, where ν is a scaling factor and σ_1 denotes the largest singular value of $\mathcal{A}^*(\mathbf{D})$. Note that this setting is consistent with the regularization parameter setting in *matrix lasso* in [50]. For robust cases, the parameter λ in (12) is chosen by $\lambda = \delta d_m$, where d_m denotes the mean of $|\mathbf{D}|$. Without loss of generality, we suggest setting $\nu \in (0.0001, 0.01)$ and $\delta \in (0.01, 1]$. One may also apply the cross-validation to choose ν and δ, but it is not considered in this paper.

[2] When λ is very small, $\|\mathbf{E}\|_1$ can be very large at the beginning, making the initial point far from the optimum.

5. Related studies

The proposed PRG methods over $\mathcal{M}_{\leq r}$ is closely related to fixed-rank methods defined on nonlinear fixed-rank manifolds [1, 5, 35, 36, 26], which have shown great advantages in computation for solving large-scale matrix completion problem [1, 5, 35], such as the low-rank geometric conjugate gradient method (LRGeomCG) [51], the quotient geometric matrix completion method (qGeomMC) [37], scaled gradients on Grassmann manifolds for matrix completion (ScGrassMC) method [40]. However, these methods can only deal with smooth objectives, and cannot handle trace-norm regularized problems. Some researchers proposed to solve a variational form of trace-norm regularized problem [17, 41]: $\min_{\mathbf{G},\mathbf{H}} \|\mathbf{G}\|_F^2 + \|\mathbf{H}\|_F^2$, s.t. $\mathcal{A}(\mathbf{G}\mathbf{H}^\top) = \mathbf{D}$, where $\mathbf{G} \in \mathbb{R}^{m\times r}$ and $\mathbf{H} \in \mathbb{R}^{n\times r}$. This problem can be addressed by either gradient based methods [17, 41] or stochastic gradient methods [42, 53]. However, these methods may suffer from slow convergence speeds [37, 51].

Recently, the authors in [38] exploited Riemannian structures and presented a trust-region algorithm to address trace-norm minimizations. The proposed method, denoted by MMBS, alternates between fixed-rank optimization and rank-one updates. However, empirically this method shows slower speed even than APG on large-scale problems [38]. The authors in [32] proposed a Grassmann manifold method based on a fixed-rank manifold. In general, this method has similar complexity to ScGrassMC that also operates on Grassmann manifold [40]. More recently, a new Retraction for accelerating the Riemannian three-factor low-rank matrix completion problem [26].

Active subspace methods or greedy methods, which increase the rank by one per iteration, have gained great attention in recent years [14, 17, 47, 24]. However, these methods usually involve expensive subproblems, and might be very expensive when the true rank is high. For example, Laue's method [24] needs to solve nonlinear master problems using the BFGS method, which is not scalable for large-scale problems. More recently, [16] proposed a novel active subspace selection method for solving trace-norm regularized problems, but this method may suffer from slow convergence speed due to the approximated SVDs and inefficient solvers for the subproblem optimization. The authors in [49, 57] proposed a Riemanian pursuit algorithm which increases the rank more than one, but this method cannot deal with trace-norm regularized problems. In [59], Zhou *et al.* proposed an algorithm which also operated on $\mathcal{M}_{\leq r}$ and adjusted the rank iteratively. However, our method is different from Zhou's method. Specifically, we address trace-norm regularized problems which are non-smooth; while Zhou's method focuses on smooth objectives. The trace-norm regularizer naturally encourages *low-rank solution*; while Zhou's method resorts to heuristics to reduce ranks.

6. Experimental Results

For convenience, we refer PRP with PRG(R) and PRG to as **PRP(R)** and **PRP**, respectively. We evaluate proposed methods on two classical tasks, namely matrix completion and LRR based clustering. All the experiments are conducted in Matlab (R2012b) on a PC installed a 64-bit operating system with an Intel(R) Core(TM) i7 CPU (3.2GHz with single-thread mode) and 64GB memory.

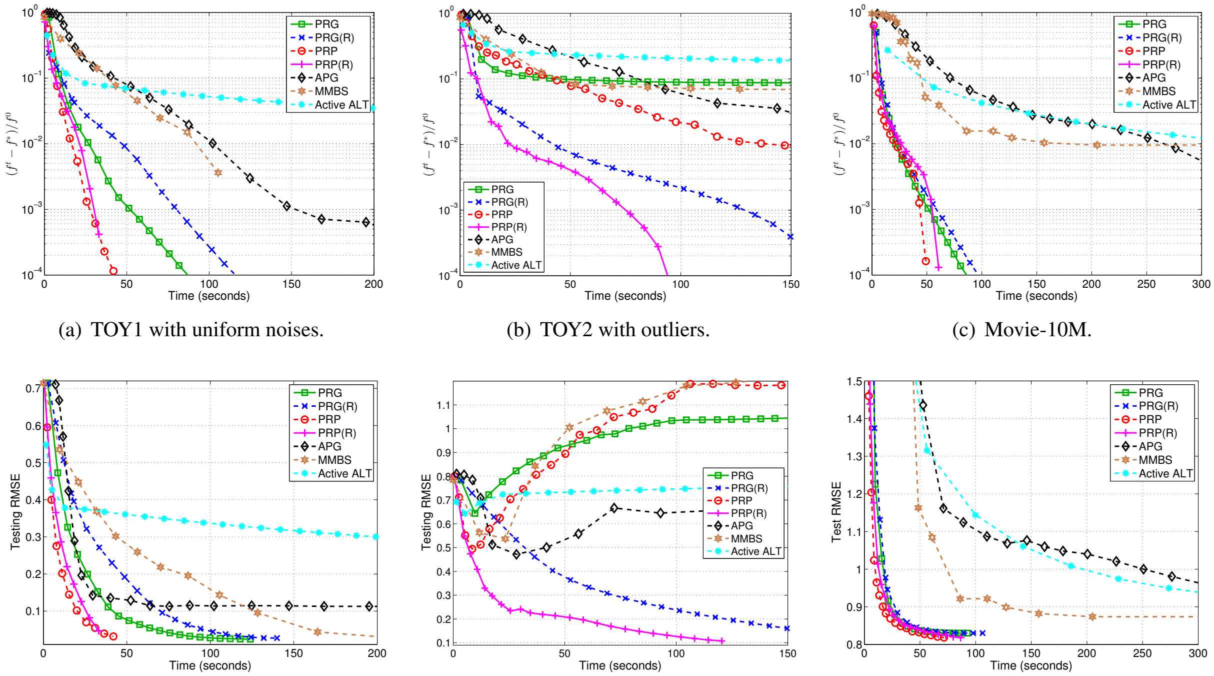
6.1. Experiments on Matrix Completion

We compare the proposed methods, *i.e.*, PRG, PRG(R), PRP and PRP(R), on several matrix completion tasks. Three state-of-the-art trace-norm based methods, e.g. APG [50], MMBS [38], and Active ALT [16], are adopted as baselines. We also compare with several efficient fixed-rank methods operating on manifolds, such as LRGeomCG [51], RP [49], LMaFit [53], ScGrassMC [40], A-R3MC1 [26], and qGeomMC [37]. We do not report the results of some methods (*e.g.*, IALM [27] and the method in [32]), since they are either slower than other compared methods or the sources are not available. We adopt the root-mean-square error (RMSE) as a major evaluation metric: RMSE $= \|\mathcal{P}_\Omega(\mathbf{D} - \mathbf{X}^*)\|_F/\sqrt{(|\Omega|)}$, where $\mathbf{X}^*$ denotes the recovered matrix, and Ω denotes the index set for testing, and $\mathcal{P}_\Omega$ denotes the orthogonal projection onto Ω [51].

6.1.1 Synthetic Experiments

Following [40, 49], we generate synthetic low-rank matrices $\mathbf{D} = \mathbf{U}\mathrm{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T} \in \mathbb{R}^{m\times m}$ of rank r, where $\mathbf{U} \in \mathrm{St}_r^m, \mathbf{V} \in \mathrm{St}_r^m$, $m = 5000$, $r = 50$ and $\boldsymbol{\sigma}$ is a 50-dimensional vector with its entries sampled from a uniform distribution $[0, 1000]$. We sample $l = \omega r(2m - r)$ entries from $\mathbf{D}$ uniformly as the observations stored in $\mathbf{d} \in \mathbb{R}^l$, where ω is an oversampling factor [27]. Here, we set $\omega = 2.5$. We study two toy data sets whose observations are perturbed by two kind of noises: In the first toy data set **TOY1**, each entry of $\mathbf{d}$ is perturbed by additive Gaussian noise of magnitude $0.01\|\mathbf{d}\|_2/\|\mathbf{n}\|_2$, where $\mathbf{n} \in \mathbb{R}^l$ is a Gaussian random vector sampled from $N(0, 1)$; The second toy data **TOY2** is constructed based on TOY1, by further perturbing 5% of the observations with outliers uniformly sampled from $[-10, 10]$. In synthetic experiments, we set $\nu = 0.005$, $\delta = 0.1$ and $\eta = 0.65$ (see equation (13)). The trace-norm based methods APG, MMBS and Active ALT, are adopted as the baselines. The *Relative objective difference* and *Testing RMSE* w.r.t. time on **TOY1** and **TOY2** are reported in Figure 1.

According to Figure 1(a), our proposed PRG, PRG(R), PRP and PRP(R) converge much faster than the comparators, and PRP and PRP(R) improve upon their counterparts (*i.e.*, PRG and PRG(R)) significantly. From Figure 1(d), the testing RMSE shows similar trends to the objective values.



(a) TOY1 with uniform noises.

(b) TOY2 with outliers.

(c) Movie-10M.

(d) TOY1 with uniform noises.

(e) TOY2 with outliers.

(f) Movie-10M.

Figure 1. Performance of various methods on **TOY1**, **TOY2** and **Movie-10M**, in terms of relative objective difference vs computational time (see (a)(b)(c)) and testing RMSE values vs computational time (see (d)(e)(f)).

Note that, our methods thus achieve low RMSE values in very short time. In general, the Active ALT method is slower than others, which may be due to the approximated SVDs and inefficient solvers for the subproblem optimization.

From Figure 1(b), on **TOY2** which is disturbed by outliers, our proposed PRG(R) and PRP(R) converge faster than the baselines. From Figure 1(e), only the proposed PRG(R) and PRP(R) achieve promising testing RMSE values; while other methods over-fit the data after several iterations due to outliers. Note that PRP(R) converges faster than its counterpart PRG(R). These observations demonstrate the effectiveness and efficiency of our proposed methods.

6.1.2 Experiments on Real-world Data

We study the performance of PRP and SR-PRG on three collaborative filtering data sets: MovieLens with 10M ratings (denoted by Movie-10M) [15], Netflix Prize dataset [22] and and Yahoo! Music Track 1 data set [11]. The statistics of these data sets are recorded in Table 1.

In the first experiment, we only compare with the three trace-norm based methods, e.g. APG [50], MMBS [38] and Active ALT [16] on Movie-10M. We report the change of *Relative objective difference* and *Testing RMSE* w.r.t. time in Figure 1. Here, we randomly choose 80% of the ratings as the training set and the remainder as the testing set.

From Figures 1(c) and 1(f), our proposed methods show much faster convergence speed as well as faster decreasing of testing RMSE values. In the second experiment, the baseline methods include APG [50],[3] MMBS [38], LRGeomCG [51], qGeomMC [37], LMaFit [53], ScGrassMC [40],[4] Active ALT [16],[5] RP [49][6] and A-R3MC1 [26].[7] Among them, A-R3MC1 is a recently developed method which applies a new retraction technique for solving fixed-rank problems. It is considered as the state-of-the-art method.

Table 1. Statistics of datasets.

Data set	m	n	$\|\Omega\|$
Movie-10M	71,567	10,677	10,000,054
Netflix	48,089	17,770	100,480,507
Yahoo	1,000,990	624,961	252,800,275

For fixed-rank methods, the rank parameter must be provided. In this paper, the ranks returned by PRP are used as the rank estimations for the fixed-rank methods, i.e., ScGrassMC, qGeomMC, LMaFit and A-R3MC1. Specifically, the ranks returned by PRP for the three data sets are 14,

[3] APG method is from http://www.math.nus.edu.sg/~mattohkc/NNLS.html.

[4] MMBS, LRGeomCG, qGeomMC, LMaFit, and ScGrassMC are available from http://www.montefiore.ulg.ac.be/~mishra/fixedrank/fixedrank.html.

[5] http://www.cs.utexas.edu/~cjhsieh/.

[6] http://www.tanmingkui.com/rp.html.

[7] https://github.com/innerlee/Publications.

16, 28, respectively, and PRP(R) returns the same ranks. Other parameters for the comparison methods are kept default. We set $\nu = 0.001$ $\eta = 0.65$, and $\delta = 0.7$ in our methods. Following [24, 47, 17], we report the testing RMSE of different methods over 10 random 80/20 training/testing partitions.

Comparison results are shown in Table 2. According to the table, PRP and PRP(R) generally perform the best among all the compared methods in terms of testing RMSE and time. The proposed PRP and PRP(R) methods also achieve slightly better testing RMSE than RP with comparable time. Note that RP relies on carefully designed stopping conditions to induce low-rank solutions, and cannot deal with outliers [49]. Maybe due to this reason, PRP and PRP(R) achieve significant improvements in terms of testing RMSE on the Yahoo data set.

Table 2. Experimental results on real-world datasets, where time is recorded in seconds. Some results on Netflix and Yahoo are left blank, since expensive computation cost makes them unavailable.

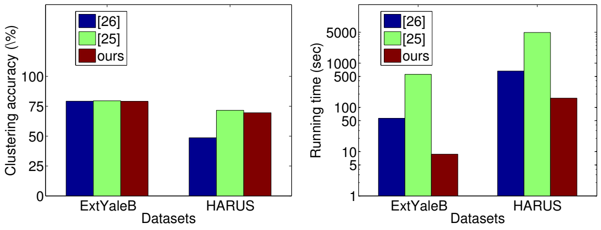
Method	Movie-10M		Netflix		Yahoo	
	RMSE	Time	RMSE	Time	RMSE	Time
APG	1.094	810.01	1.038	2883.80	–	–
LRGeomCG	0.823	57.67	0.860	2356.86	25.228	18319
QgeomMC	0.836	96.41	0.897	9794.75	24.167	82419
LMaFit	0.838	133.86	0.876	2683.73	24.368	24349
ALT	0.855	917.17	–	–	–	–
MMBS	0.821	441.10	–	–	–	–
ScGrassMC	0.845	216.07	0.892	4522.68	24.954	37705
RP	0.818	46.56	0.858	1143.02	23.451	12456
A-R3MC1(50)	0.8327	93.27	0.9003	3340.08	23.243	13157
A-R3MC1(100)	0.8276	175.87	0.8868	4999.45	22.497	21185
PRP	0.817	53.42	0.855	1057.35	22.644	15972
PRP(R)	0.815	67.73	0.857	1245.15	22.537	17263

6.2. Experiments on LRR Subspace Clustering

We compare our PRP(R) method with existing LRR solvers in [29, 30]. In [55], the authors proposed a LRR solver which is based on factorizing the data matrix with its truncated SVD. However, the computation of skinny SVD for large data matrix is usually too expensive. We thus exclude it for the comparison.

We conduct experiments regarding the clustering task on the Extended Yale Face Database B (**ExtYaleB**) and the Human Activity Recognition Using Smartphones dataset (**HARUS**) [3] We have $\mathbf{D} \in \mathbb{R}^{2016\times 640}$ for the ExtYaleB data set and $\mathbf{D} \in \mathbb{R}^{561\times 10299}$ for the HARUS data set. The ExtYaleB contains $2,414$ frontal face images of 38 subjects with different lighting, poses and illumination conditions, where each subject has round 64 faces. Following [31], we use 640 faces from the first 10 subjects. Each face image is resized to 48×42 pixels and then reshaped as a 2016-dimensional gray-level intensity feature. The HARUS is a large-scale data set (containing 10,299 signals w.r.t. 6 activities) with data collected using embedded sensors on the smartphones carried by volunteers on their waists, when they are conducting daily activities (*e.g.*, walking, sitting, laying). The captured sensor signals are pre-processed to filter noise and post-processed.

Following [30], we measure the clustering performance by *clustering accuracy*. The best clustering accuracies and corresponding running times are reported in Figure 2. From the figure, our PRP(R) method outperforms the two existing LRR solvers in terms of efficiency, since our algorithm does not frequently involve SVDs w.r.t. large matrices. Moreover, our algorithm achieves comparable clustering performance with [29]. In contrast, the LRR solver in [30] achieves lower clustering accuracy on HARUS, possibly because that algorithm is not guaranteed to obtain a globally optimal solution.



(a) Clustering accuracy.

(b) Running time.

Figure 2. Clustering accuracies and Running time of different LRR solvers on two data sets.

7. Conclusion

Classical proximal gradient methods for addressing trace-norm regularized problems may suffer from high computational cost on large-scale problems [52]. To reduce the computational complexity, we have proposed in this paper a Proximal Riemannian Pursuit (PRP) strategy which addresses general trace-norm regularized problems by progressively activating a number of active subspaces. Moreover, we have proposed a Proximal Riemannian Gradient (PRG) method for addressing the trace-norm regularized subproblems defined over a matrix variety $\mathcal{M}_{\leq r}$, where r is adjusted automatically by PRP. By exploiting geometries on $\mathcal{M}_{\leq r}$, PRG maintains the SVDs of the intermediate solutions via cheaper low-rank QR decompositions, without solving truncated SVDs of large-ranks explicitly at high computational cost. Extensive experiments on multiple data sets have demonstrated the superior efficiency of the proposed methods over other methods.

Acknowledgement

This work was in part funded by the Data to Decisions Cooperative Research Centre, Australia, Australian Research Council grants DP140102270 and DP160100703, the National Natural Science Foundation of China under Grant No.61472267.

References

[1] P.-A. Absil, L. Amodei, and G. Meyer. Two Newton methods on the manifold of fixed-rank matrices endowed with Riemannian quotient geometries. *Comput. Stat.*, 29(3-4):569–590, 2014.

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

[3] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Ambient assisted living and home care*, pages 216–223. 2012.

[4] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Mach. Learn.*, 73(3):243–272, 2008.

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

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

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

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

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

[10] C.-H. Chen, V. M. Patel, and R. Chellappa. Matrix completion for resolving label ambiguity. In *CVPR*, 2015.

[11] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer. The Yahoo! Music Dataset and KDD-Cup'11. *JMLR Workshop and Conference Proceedings*, 2012.

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

[13] A. Goldberg, B. Recht, J. Xu, R. Nowak, and X. Zhu. Transduction with matrix completion: Three birds with one stone. In *Advances in neural information processing systems*, pages 757–765, 2010.

[14] E. Hazan. Sparse approximate solutions to semidefinite programs. *LATIN*, pages 306–316, 2008.

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

[16] C. Hsieh and P. Olsen. Nuclear norm minimization via active subspace selection. In *ICML*, 2014.

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

[18] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML*. ACM, 2009.

[19] F. Jiang, M. Oskarsson, and K. Astrom. On the minimal problems of low-rank matrix factorization. In *CVPR*, 2015.

[20] L. Jing, L. Yang, J. Yu, and M. K. Ng. Semi-supervised low-rank mapping learning for multi-label classification. In *CVPR*, 2015.

[21] X.-Y. Jing, X. Zhu, F. Wu, X. You, Q. Liu, D. Yue, R. Hu, and B. Xu. Super-resolution person re-identification with semi-coupled low-rank discriminant dictionary learning. In *CVPR*, 2015.

[22] KDDCup. ACM SIGKDD and netflix. In *Proceedings of KDD Cup and Workshop*, 2007.

[23] R. M. Larsen. Propack–software for large and sparse svd calculations. 2004.

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

[25] C. Li, L. Lin, W. Zuo, S. Yan, and J. Tang. Sold: Sub-optimal low-rank decomposition for efficient video segmentation. In *CVPR*, 2015.

[26] Z. Li, D. Zhao, Z. Lin, and E. Y. Chang. A new retraction for accelerating the riemannian three-factor low-rank matrix completion algorithm. In *CVPR*, 2015.

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

[28] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *CAMSAP*, 61, 2009.

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

[30] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *T-PAMI*, 35(1):171–184, 2013.

[31] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

[32] Y. Liu, F. Shang, H. Cheng, and J. Cheng. Nuclear norm regularized least squares optimization on grassmannian manifolds. In *UAI*, 2014.

[33] C. Lu, J. Tang, S. Yan, and Z. Lin. Generalized nonconvex nonsmooth low-rank minimization. In *CVPR*, 2014.

[34] S. Lu, X. Ren, and F. Liu. Depth enhancement via low-rank matrix completion. In *CVPR*, 2014.

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

[36] B. Mishra, , and R. Sepulchre. R3mc: A riemannian three-factor algorithm for low-rank matrix completion. In *53rd IEEE Conference on Decision and Control*, 2014.

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

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

[39] Y. Nesterov. Gradient methods for minimizing composite objective function. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL), 2007.

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

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

[42] B. Recht and C. Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Math. Programm. Comput.*, 5(2):201–226, 2013.

[43] R. Salakhutdinov and N. Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *NIPS*, 2010.

[44] R. Schneider and A. Uschmajew. Convergence results for projected line-search methods on varieties of low-rank matrices via Lojasiewicz inequality. *arXiv preprint arXiv:1402.5284*, 2014.

[45] Y. Shen, Z. Wen, and Y. Zhang. Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. *Optim. Method. and Softw.*, (ahead-of-print):1–25, 2012.

[46] X. Shu, F. Porikli, and N. Ahuja. Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices. In *CVPR*, 2014.

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

[48] S. Sra, S. Nowozin, and S. J. Wright. *Optimization for Machine Learning*. MIT Press, 2011.

[49] M. Tan, I. W. Tsang, L. Wang, B. Vandereycken, and S. J. Pan. Riemannian pursuit for big matrix recovery. In *ICML*, 2014.

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

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

[52] S. Wei and Z. Lin. Accelerating iterations involving eigenvalue or singular value decomposition by block lanczos with warm start. Technical report, Technical Report MSR-TR-2010-162, Microsoft Research, 2010.

[53] Z. Wen, W. Yin, and Y. Zhang. 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.

[54] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *NIPS*, 2009.

[55] S. Xiao, W. Li, D. Xu, and D. Tao. Falrr: A fast low rank representation solver. In *CVPR*, 2015.

[56] S. Xiao, D. Xu, and J. Wu. Automatic face naming by learning discriminative affinity matrices from weakly labeled images. *IEEE Trans. Neural Netw. Learning Syst.*, 26(10):2440–2452, 2015.

[57] Y. Yan, M. Tan, I. Tsang, Y. Yang, C. Zhang, and Q. Shi. S-calable maximum margin matrix factorization by active riemannian subspace search. In *IJCAI*, 2015.

[58] X. Zhang, D. Schuurmans, and Y. Yu. Accelerated training for matrix-norm regularization: A boosting approach. In *NIPS*, 2012.

[59] G. Zhou, W. Huang, K. A. Gallivan, P. V. Dooren, and P. A. Absil. Rank-constrainted optimization: A riemannian manifold approach. In *ESANN 2015 proceedings*, 2015.

[60] T. Zhou and D. Tao. Godec:randomized low-rank & sparse matrix decomposition in noisy case. In *ICML*, 2011.