Riemannian Pursuit for Big Matrix Recovery

Mingkui Tan[1], Ivor W. Tsang[2], Li Wang[3],
Bart Vandereycken[4], Sinno Jialin Pan[5]

[1]School of Computer Science, University of Adelaide, Australia
[2]Center for Quantum Computation & Intelligent Systems, UTS, Australia
[3]Department of Mathematics, University of California, USA
[4]Department of Mathematics, Princeton University, USA
[5]Institute for Infocomm Research, SG

August 18, 2014

Outline

Notations

We will use the following notations:

- Given a linear operator $\mathcal{A}$, denote its adjoint operator by $\mathcal{A}^*$. For instance, if $\mathcal{A}$ is a matrix $\mathbf{A}$, its adjoint operator is $\mathbf{A}^\top$
- Let $\mathbf{X} = \mathbf{U}(\text{diag}(\boldsymbol{\sigma}))\mathbf{V}^\top$ be the SVD of $\mathbf{X} \in \mathbb{R}^{m\times n}$. The nuclear norm of $\mathbf{X}$ is defined as

$$\|\mathbf{X}\|_* = \|\boldsymbol{\sigma}\|_1 = \sum_i |\sigma_i|.$$

- The condition number $\kappa_r(\mathbf{X})$ of $\mathbf{X}$ w.r.t. a given number r is defined as $\kappa_r(\mathbf{X}) = \sigma_1/\sigma_r$

Low Rank Assumption

- Recovering $\widehat{\mathbf{X}}$ from partial observations $\mathbf{b}$ is impossible in general.
- Low-rank assumption: $\text{rank}(\widehat{\mathbf{X}}) \leq r$, where $r \ll \min(m, n)$.
  - Faces from a same person lie on a low-dimensional manifold (Yan, *et al.* , 2007).
  - In collaborative filtering tasks, items from same group may have similar actions.

Problem Formulation

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

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

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

The definition of the operator $\mathcal{A}$ depends on specific application:

- Matrix completion [Recht et al.(2010)]:
- Quantum state tomography [Candés & Plan(2010a)]
- Matrix learning & factorizations [Laue(2012)]
- Low rank structure learning or clustering [Deng et al.(2013)]
- ....

Existing MR Methods

MR by minimizing $f(\mathbf{X})$ with rank constraint

$$\text{rank}(\mathbf{X}) \leq r$$

is known to be NP-hard.

- How to estimate r?
- Estimate $\text{rank}(\mathbf{X})$ by cross-validation?
  A good idea, but, how to measure the performance of a specific parameter? Very difficult!

Existing methods:

1. Nuclear-norm convex relaxations [Candés & Plan(2010a)]:
   Replace the rank constraint by $||\mathbf{X}||_* \leq v$ for some $v > 0$.
2. Fixed-rank methods (relaxations):
   Assume $\text{rank}(\mathbf{X}) = r$, where r is supposed to be known.
3. Other methods: Max-norm based methods, p-norm non-convex methods $(0 < p < 1)$, and so on.

Nuclear-norm Convex Relaxations

Two kinds of trace-norm relaxations are usually studied:

- Nuclear-norm minimization with equality constraint

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

  Typical methods:
  - Singular Value Thresholding (SVT) [Cai et al.(2010)]
  - Augmented Lagrangian method (ALM) [Lin et al.(2010)]

- Matrix lasso problem:

$$\min_{\mathbf{X},\boldsymbol{\xi}} \lambda\|\mathbf{X}\|_* + \frac{1}{2}\|\boldsymbol{\xi}\|_2^2 \ : \ \boldsymbol{\xi} = \mathbf{b} - \mathcal{A}(\mathbf{X}), \tag{3}$$

  Typical methods:
  - Accelerated Proximal Gradient (APG) [Toh & Yun(2010)]
  - Stochastic Gradient methods

Nuclear-norm Convex Relaxations

Advantages:

- A global solution can be obtained by convex optimization methods (but may not be unique).
- Good RIP guarantees for exact matrix recovery, namely, $\gamma_r \leq 1/3$ (Cai *et al.* , 2013).

Disadvantages:

- Require many high-dimensional SVDs.
- Very expensive for large-scale problems.
- Has solution bias due to the nuclear-norm regularization [Vandereycken(2013)].

Challenges

The key is how to avoid high dimensional SVDs and estimate the correct rank

Fixed-rank Methods

The fixed-rank methods solve the following problem:

$$\min_{\mathbf{X}} f(\mathbf{X}), \qquad \text{s.t.} \quad \text{rank}(\mathbf{X}) = r, \tag{4}$$

where r is supposed to be known.

- Non-convex since the constraint $\text{rank}(\mathbf{X}) = r$ defines a nonlinear smooth matrix manifold [Meyer et al.(2011)].
- Many efficient local-search methods have been proposed:
  - Greedy methods:
    - Singular Value Projection (SVP) [Meka et al.(2009b)]
    - Atomic Decomposition for Minimum Rank Approximation (ADMiRA) [Lee & Bresler(2010)]
  - Stochastic Gradient methods [Wen et al.(2012)].
  - Manifold optimization methods:
    - Low-rank geometric conjugate gradient method (LRGeomCG) [Vandereycken(2013)]
    - The quotient geometric matrix completion method (qGeomMC) [Mishra et al.(2012)]

Differential Geometry of Fixed-rank Matrices

- Smooth manifold $\mathcal{M}_r$ of fixed-rank matrices:

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

  Stiefel manifold: $m \times r$ real and orthonormal matrices: $\text{St}_r^m = \{\mathbf{U} \in \mathbb{R}^{m\times r} : \mathbf{U}^\top\mathbf{U} = \mathbf{I}\}$.

- The tangent space $T_{\mathbf{X}}\mathcal{M}_r$ of $\mathcal{M}_r$ at $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^\top \in \mathbb{R}^{m\times n}$: $T_{\mathbf{X}}\mathcal{M}_r = \{\mathbf{U}\mathbf{M}\mathbf{V}^\top + \mathbf{U}_p\mathbf{V}^\top + \mathbf{U}\mathbf{V}_p^\top : \mathbf{M} \in \mathbb{R}^{r\times r}, \mathbf{U}_p \in \mathbb{R}^{m\times r}, \mathbf{U}_p^\top\mathbf{U} = \mathbf{0}, \mathbf{V}_p \in \mathbb{R}^{n\times r}, \mathbf{V}_p^\top\mathbf{V} = \mathbf{0}\}$.

Differential Geometry of Fixed-rank Matrices

Riemannian gradient of f on $\mathcal{M}_r$:

- Given a smooth function $f : \mathcal{M}_r \to \mathbb{R}$, the Riemannian gradient is the orthogonal projection of the gradient of f onto the tangent space.
- Define $P_U = \mathbf{U}\mathbf{U}^\top$ and $P_U^\perp = \mathbf{I} - \mathbf{U}\mathbf{U}^\top$ for any $\mathbf{U} \in \mathrm{St}_r^m$, and the orthogonal projection of any $\mathbf{Z} \in \mathbb{R}^{m\times n}$ onto the tangent space at $\mathbf{X} = \mathbf{U}\mathrm{diag}(\boldsymbol{\sigma})\mathbf{V}^\top$ is defined as

$$\mathbf{E} = P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{Z}) : \mathbf{Z} \mapsto P_U\mathbf{Z}P_V + P_U^\perp\mathbf{Z}P_V + P_U\mathbf{Z}P_V^\perp. \tag{5}$$

Retraction:

- The *Retraction* mapping lets a point $\mathbf{X} + \mathbf{E}$ in the tangent space go back to the manifold:

$$R_\mathbf{X}(\mathbf{E}) = P_{\mathcal{M}_r}(\mathbf{X} + \mathbf{E}) = \sum_{i=1}^{r} \sigma_i \mathbf{p}_i \mathbf{q}_i^\top, \tag{6}$$

  where $\sum_{i=1}^{r} \sigma_i \mathbf{p}_i \mathbf{q}_i^\top$ denotes the best rank-r approximation to $\mathbf{X} + \mathbf{E}$ by the SVD.
- *Retraction* can be calculated with $O((m+n)r^2)$ cost.

Fixed-rank Methods

Advantages of Fixed-rank Methods:

- Fixed-rank Methods have superior scalability compared with the nuclear-norm based methods [Boumal & Absil(2012), Mishra et al.(2012), Vandereycken(2013)].
- Greedy methods need truncated SVD of known low rank only.
- Some manifold optimization methods involve simple matrix-products, which is particularly important for parallel computing.

Disadvantages

- How do we know the rank r? It is nontrivial.
- Greedy methods require restricted conditions to converge.
- Convergence Issues: If $\widehat{\mathbf{X}}$ is in **ill-conditioning**, existing fixed-rank methods may converge very slowly [Ngo & Saad(2012)].

Motivations

- Fixed-rank methods have gained great success in solving big MR problems, but it requires the explicit knowledge of the rank r.
- Three Questions:
  1. Can we avoid the rank estimation by iteratively increase the rank by a fixed integer ρ?
  2. If using this procedure, how can we stop it?
  3. Is this procedure helpful to avoid the convergence issue on ill-conditioned problems?

Main Contributions

- We propose a *Riemannian Pursuit* (RP) method, which indeed increases the rank by ρ and essentially solves a sequence of fixed-rank minimization problems.
- RP converges linearly under mild conditions.
- RP can effectively address the convergence issues that occur with ill-conditioned and large rank problems.
- RP can automatically estimates the rank under proper stopping conditions.

Active Subspace Search

- Let $\mathbf{G} = \mathcal{A}^*(\mathcal{A}(\mathbf{X}) - \mathbf{b})$. The gradient of $f(\mathbf{X})$ on $\mathcal{M}_r$ can be calculated by

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

- The gradient direction orthogonal to $\mathcal{M}_r$ is $\mathbf{Q} = \mathbf{G} - P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G})$
- Select the subspace of rank ρ from $\mathbf{Q}$
- Add this subspace into the original manifold

Riemannian Pursuit

Algorithm 1 Riemannian Pursuit for MR.

1: Inner iteration tolerance ϵ_{in}, and outer iteration tolerance ϵ_{out}. Initialize $\mathbf{X}^0 = \mathbf{0}$, $\boldsymbol{\xi}^0 = \mathbf{b}$ and $\mathbf{G} = \mathcal{A}^*(\boldsymbol{\xi}^0)$. Let $t = 1$ and $r = \rho$.

2: Perform an active-subspace search as follows.

(2a): Compute $\mathbf{Q} = \mathbf{G} - P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G})$.

(2b): Compute a best rank ρ approximation of $\mathbf{Q}$:

$$\mathbf{H}_2^{t-1} = \mathbf{U}_\rho \text{diag}(\boldsymbol{\sigma}_\rho) \mathbf{V}_\rho{}^\top$$

3: Let $\mathbf{H}_1^{t-1} = P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G})$ and $\mathbf{H}^{t-1} = \mathbf{H}_1^{t-1} + \mathbf{H}_2^{t-1}$.

(3a): Choose a proper step size τ_t and set

$$\mathbf{X}^{\text{intial}} = R_{\mathbf{X}}(-\tau_t \mathbf{H}^{t-1}). \quad (\text{Warm Start})$$

(3b): Update $\mathbf{X}^t$ by $\mathbf{X}^t$ =NRCG($\mathbf{X}^{\text{intial}}$, ϵ_{in}).

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

5: Quit if stopping conditions achieve, otherwise, let $t = t+1$, and go to Step 2.

Main Theoretical Results

- LEMMA 1: Firstly, let $\{\mathbf{X}^t\}$ be the sequence generated by RP, then $f(\mathbf{X}^t)$ decreases monotonically w.r.t. t.
- THEOREM 1: Let $\{\mathbf{X}^t\}$ be the sequence generated by RP, as long as $f(\mathbf{X}^t) > f(\widehat{\mathbf{X}}) = \frac{C}{2}||\mathbf{e}||^2$ (where $C > 1$), and there exists an integer $\iota > 0$ such that $\gamma_{r+2\iota\rho} < 1/2$, then RP decreases linearly in objective values when $t < \iota$, namely $f(\mathbf{X}^{t+1}) \leq \nu f(\mathbf{X}^t)$, where
$$\nu = \left(1 - \frac{\rho\zeta}{2r}\left(\frac{C(1-2\gamma_{(r+2\iota\rho)})^2}{(\sqrt{C}+1)^2(1-\gamma_{(r+2\iota\rho)})}\right)(1-\frac{1}{\sqrt{C}})^2\right).$$

Stopping Conditions of RP

- RP monotonically decreases the objective values. Without proper stopping conditions, it may increase the rank until $t\rho = \min(m, n)$, leading to over-fitting issue.
- To avoid this, we propose to use the averaged function value difference between iterations as the stopping condition, namely

$$2(f(\mathbf{X}^{t-1}) - f(\mathbf{X}^{t}))/(\rho\|\mathbf{b}\|_2^2) \leq \epsilon_{out}, \tag{8}$$

  where ϵ_{out} is a predefined tolerance value.
- If the increasing of rank does not significantly reduce the objective value, we stop it to avoid over-fitting.
- When it is stopped, a rank is returned!

Riemannian Pursuit

Algorithm 2 Riemannian Pursuit for MR.

1: Inner iteration tolerance ϵ_{in}, and outer iteration tolerance ϵ_{out}. Initialize $\mathbf{X}^0 = \mathbf{0}$, $\boldsymbol{\xi}^0 = \mathbf{b}$ and $\mathbf{G} = \mathcal{A}^*(\boldsymbol{\xi}^0)$. Let $t = 1$ and $r = \rho$.

2: Perform an active-subspace search as follows.

(2a): Compute $\mathbf{Q} = \mathbf{G} - P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G})$.

(2b): Compute a best rank ρ approximation of $\mathbf{Q}$:

$$\mathbf{H}_2^{t-1} = \mathbf{U}_\rho \text{diag}(\boldsymbol{\sigma}_\rho) \mathbf{V}_\rho{}^\top$$

3: Let $\mathbf{H}_1^{t-1} = P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G})$ and $\mathbf{H}^{t-1} = \mathbf{H}_1^{t-1} + \mathbf{H}_2^{t-1}$.

(3a): Choose a proper step size τ_t and set

$$\mathbf{X}^{\text{intial}} = R_{\mathbf{X}}(-\tau_t \mathbf{H}^{t-1}). \quad \text{(Warm Start)}$$

(3b): Update $\mathbf{X}^t$ by $\mathbf{X}^t$ =NRCG($\mathbf{X}^{\text{intial}}$, ϵ_{in}).

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

5: Quit if stopping conditions achieve, otherwise, let $t = t+1$, and go to Step 2.

Fixed-rank subproblem

Step (3b) (Update $\mathbf{X}^t$ by $\mathbf{X}^t$ =NRCG($\mathbf{X}^{\text{intial}}$, ϵ_{in})) is to solving the following problem:

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

- ρ is much smaller than $\widehat{r}$. When t is small, $t\rho < \widehat{r}$.
- Smaller condition number:

$$\kappa_{t\rho}(\mathbf{X}) = \sigma_1/\sigma_{t\rho} < \kappa_r(\mathbf{X}) \leq \infty.$$

- Faster convergence speed with a small condition number.
- Selection of ρ is important.
  Setting $\rho = 1$ is the simplest way, but $\rho > 1$ is better.

Nonlinear Conjugate Gradient Descent

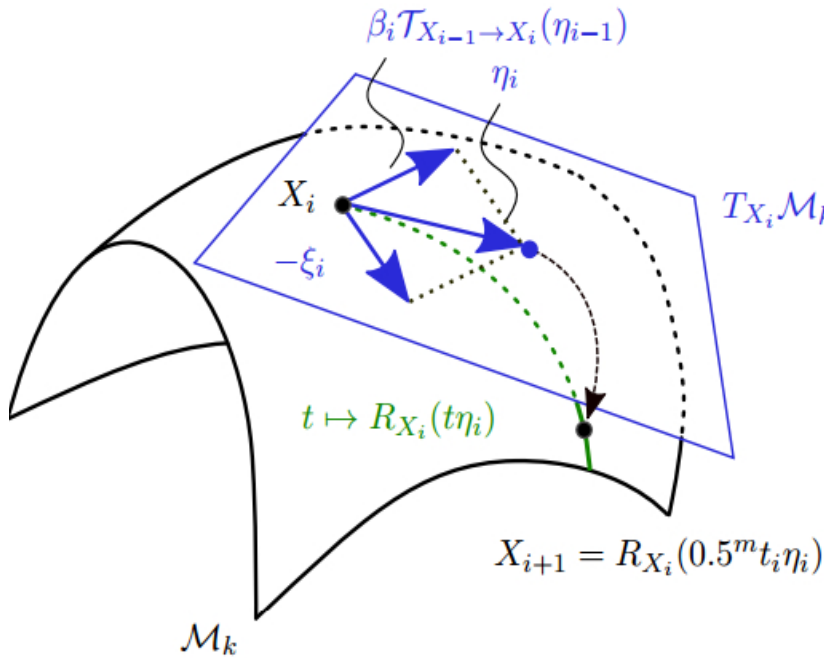


Figure: Basic elements of Riemannian manifold on matrices.

Nonlinear Conjugate Gradient Descent

The nonlinear conjugate gradient descent algorithm is shown as follows:

Algorithm 3 NRCG for solving the fixed-rank minimization problem.

1: Initialize $\mathbf{X}_1 = \mathbf{X}^{\text{intial}}$ and $\boldsymbol{\zeta}_0 = \mathbf{0}$. Let $k = 1$.
2: Compute the Riemannian gradient $\mathbf{E}_k = \text{grad} f(\mathbf{X}_k)$.
3: Compute the conjugate direction $\boldsymbol{\zeta}_k$ according to

$$\boldsymbol{\zeta}_k = \mathbf{P}_k + \beta_t \mathcal{T}_{\mathbf{X}_{k-1} \to \mathbf{X}_k}(\boldsymbol{\zeta}_{k-1}).$$

4: Choose a step size θ_k satisfying the strong Wolfe conditions, and set $\mathbf{X}_{k+1} = R_{\mathbf{X}_k}(\theta_k \boldsymbol{\zeta}_k)$.
5: Terminate and output $\mathbf{X}_{k+1}$ if the stopping conditions are achieved; otherwise, let $k = k + 1$ and go to step 1.

Experiments

- Toy Experiments for Convergence Comparison
- Real-world Experiments on Collaborative Filtering Tasks

Toy Experimental Settings

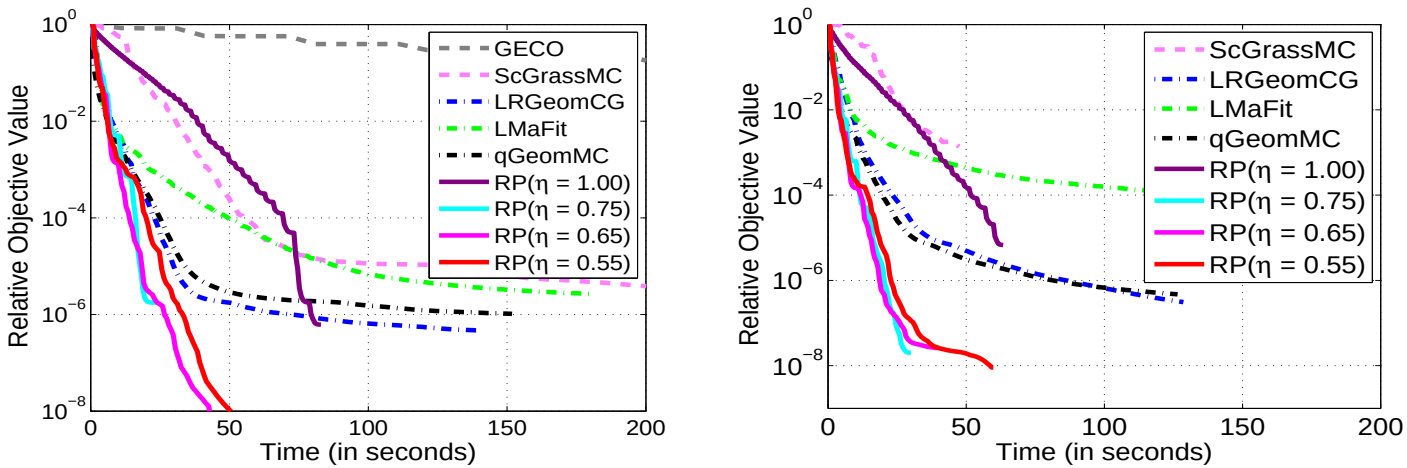
Methods for comparison:

- LRGeomCG, qGeomMC, ScGrassMC, SVP, ADMiRA, SpaRCS, GECO and APG.
- Except GECO and APG, others are fixed-rank methods.

Toy experimental setting

- We generate ground-truth $\widehat{\mathbf{X}}$ by $\widehat{\mathbf{X}} = \widehat{\mathbf{U}}\mathrm{diag}(\widehat{\boldsymbol{\sigma}})\widehat{\mathbf{V}}^\top + \mathbf{e}$, where $\boldsymbol{\sigma}$ is a $\widehat{r}$-sparse vector, $\widehat{\mathbf{U}} \in \mathrm{St}_r^m$, and $\widehat{\mathbf{V}} \in \mathrm{St}_r^n$.
- Two kinds of singular values are studied:
  1) *Gaussian* sparse singular value vector $\mathbf{s}_g$ sampled from the *Gaussian* distribution $N(0, 1000)$;
  2) χ^2 sparse singular value vector $\mathbf{s}_\chi^2$, where each entry is the square of $\mathbf{s}_g$.

Toy Experiments for Convergence Comparison



(a) Relative objective values on $\mathbf{s}_g$, where ρ w.r.t. different η's is 1, 1, 8, 14, 18, respectively.

(b) Relative objective values on $\mathbf{s}_\chi^2$, where ρ for different η's is 1, 4, 6, 12, respectively.

Figure: Convergence of comparison methods on $\mathbf{s}_g$ and $\mathbf{s}_{\chi^2}$.

- RP with different ρ converges well on ill-conditioned problems.
- Other algorithms have convergence issues. GECO cannot converge within 1 hour on $\mathbf{s}_g$, and we omit its results on $\mathbf{s}_{\chi^2}$. ScGrassMC gets numerical problems after 50 iterations on $\mathbf{s}_{\chi^2}$.

Experiments on Real-world Collaborative Filtering Tasks

Table: Experimental results on real-world datasets. Result of APG on Netflix is absent. The average ranks estimated by APG, Lmafit-A and RP on Movie-10M are 100, 77 and 10, respectively. The average ranks estimated by Lmafit-A and RP on Netflix are 81 and 12, respectively

Dataset	Movie-10M		Netflix-100M	
	RMSE	Time (seconds)	RMSE	Time (seconds)
APG	1.096	1048±17	-	-
LRGeomCG	0.824	338±11	0.867	3128±35
QgeomMC	0.850	189±7	0.880	3965 ± 74
Lmafit	0.837	307±1	0.875	3798±50
Lmafit-A	0.969	421±16	0.962	5286±165
RP	**0.817**	81±1	**0.859**	1332±27

- RP performs best among all the methods in terms of RMSE and computational efficiency.
- We use the rank returned by RP as the rank for LRGeomCG, qGeomMC and Lmafit, thus RP is much faster if the model selection cost is considered.

Comparison on MovieLens with 10M ratings

Table: Performance comparison on Movie-10M dataset.

Methods	Time (in seconds)	SpeedUp	RMSE	CPU(GHz)
GECO	784,941	**9,000x**	0.821	2.5
Laue	2,663	**30x**	0.815	2.5
Jaggi	3,120	**38x**	0.8617	2.4
RP	81	–	0.817	2.8

Conclusions

We propose a Riemannian Pursuit (RP) for tackling **Big Matrix Recovery** problems.

- By exploiting the Riemannian geometry on the fixed-rank manifolds, high-dimensional SVDs in the master problem of RP are avoided, which exhibits good scalability for large-scale problems.
- RP converges linearly under mild conditions.
- RP can automatically estimate the rank and effectively address the convergence issues on ill-conditioned and large rank matrices.
- Extensive experimental results show that RP achieves superb scalability on **Big Matrices** using a single PC; while it maintains similar or better MR performance.

Thank you for your attention!
Our Poster is at T4!

Conjugate Gradient Descent on Manifold

- Steepest gradient descent in general is very slow.
- Conjugate gradient descent is a good choice.
  Let $\boldsymbol{\zeta}_{k-1}$ be a search direction at $(k-1)^{\text{th}}$, we compute the conjugate search direction:

$$\boldsymbol{\zeta}_t = -\text{grad} f(\mathbf{X}_k) + \beta_t \boldsymbol{\zeta}_{k-1}, \tag{10}$$

  where β_t can be calculated by the Fletcher-Reeves (F-R) rule.
- But notice:
  - $\text{grad} f(\mathbf{X}_k)$ and $\beta_t \boldsymbol{\zeta}_{k-1}$ are in different tangent spaces!
    Therefore, Eq. (10) is not valid!
  - We still need an operator: *Vector Transport*.

Conjugate Gradient Descent on Manifold

A *Vector Transport* $\mathcal{T}$ on a manifold $\mathcal{M}_r$ is a smooth map which transports tangent vectors from one tangent space to another.

- A vector transport $\mathcal{T}$ on a manifold $\mathcal{M}_r$ is a smooth map [Absil et al.(2008)]:

$$T\mathcal{M}_r \oplus T\mathcal{M}_r \to T\mathcal{M}_r : (\boldsymbol{\zeta}_\mathbf{X}, \boldsymbol{\nu}_\mathbf{X}) \to \mathcal{T}_{\boldsymbol{\zeta}_\mathbf{X}}(\boldsymbol{\nu}_\mathbf{X})$$

  satisfying the following properties for all $\mathbf{X} \in \mathcal{M}_r$:
  - There exists a retraction R associated with $\mathcal{T}$ such that for all $(\boldsymbol{\zeta}_\mathbf{X}, \boldsymbol{\nu}_\mathbf{X}) \in T\mathcal{M}_r \oplus T\mathcal{M}_r$, it holds that $\mathcal{T}_{\boldsymbol{\zeta}_\mathbf{X}}(\boldsymbol{\nu}_\mathbf{X}) \in T_{R_\mathbf{X}}\mathcal{M}_r$.
  - $\mathcal{T}_\mathbf{0}(\boldsymbol{\nu}) = \boldsymbol{\nu}$ for all $\boldsymbol{\nu} \in T_\mathbf{X}\mathcal{M}_r$.
  - $\mathcal{T}_{\boldsymbol{\zeta}_\mathbf{X}}(a\boldsymbol{\nu}_\mathbf{X} + b\boldsymbol{\omega}_\mathbf{X}) = a\mathcal{T}_{\boldsymbol{\eta}_\mathbf{X}}(\boldsymbol{\nu}_\mathbf{X}) + b\mathcal{T}_{\boldsymbol{\eta}_\mathbf{X}}(\boldsymbol{\omega}_\mathbf{X})$ for $\boldsymbol{\nu}_\mathbf{X} \in T\mathcal{M}_r$ and $\boldsymbol{\omega}_\mathbf{X} \in T\mathcal{M}_r$.
- Very tedious definition, but the computation takes only $O(m+n)r^2$ cost.

Conjugate Gradient Descent on Manifold

With *Vector Transport* $\mathcal{T}$, the conjugate search direction can be calculated by

$$\boldsymbol{\zeta}_k = \mathbf{P}_k + \beta_t \mathcal{T}_{\mathbf{x}_{k-1} \to \mathbf{x}_k}(\boldsymbol{\zeta}_{k-1}), \tag{11}$$

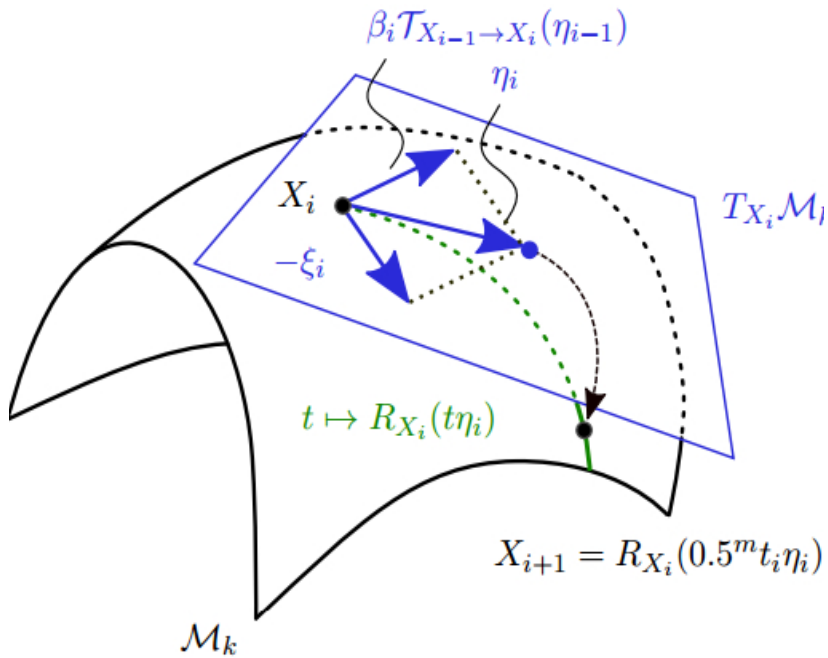


Figure: Conjugate search direction computation.

Other Issues in NRCG

- Line search for the step size θ:
  - Armijo line search: The convergence cannot be guaranteed.
  - Strong Wolfe line search: The convergence can be guaranteed.
- Initialization with Warm Start:
  Set $\mathbf{X}_1 = \mathbf{X}^{\text{intial}}$, where $\mathbf{X}^{\text{intial}}$ is from Algorithm 1.

Riemannian Pursuit

Algorithm 4 Riemannian Pursuit for MR.

1: Inner iteration tolerance ϵ_{in}, and outer iteration tolerance ϵ_{out}. Initialize $\mathbf{X}^0 = \mathbf{0}$, $\boldsymbol{\xi}^0 = \mathbf{b}$ and $\mathbf{G} = \mathcal{A}^*(\boldsymbol{\xi}^0)$. Let $t = 1$ and $r = \rho$.

2: Perform an active-subspace search as follows.

(2a): Compute $\mathbf{Q} = \mathbf{G} - P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G})$.

(2b): Compute a best rank ρ approximation of $\mathbf{Q}$:

$$\mathbf{H}_2^{t-1} = \mathbf{U}_\rho \text{diag}(\boldsymbol{\sigma}_\rho) \mathbf{V}_\rho{}^\top$$

3: Let $\mathbf{H}_1^{t-1} = P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G})$ and $\mathbf{H}^{t-1} = \mathbf{H}_1^{t-1} + \mathbf{H}_2^{t-1}$.

(3a): Choose a proper step size τ_t and set

$$\mathbf{X}^{\text{intial}} = R_{\mathbf{X}}(-\tau_t \mathbf{H}^{t-1}). \text{ (Warm Start)}$$

(3b): Update $\mathbf{X}^t$ by $\mathbf{X}^t$ =NRCG($\mathbf{X}^{\text{intial}}$, ϵ_{in}).

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

5: Quit if stopping conditions achieve, otherwise, let $t = t+1$, and go to Step 2.

Other Issues in NRCG

- Line search for the step size θ:
  - Armijo line search: The convergence cannot be guaranteed.
  - Strong Wolfe line search: The convergence can be guaranteed.
- Warm start for initialization:
  Set $\mathbf{X}_1 = \mathbf{X}^{\text{intial}}$, where $\mathbf{X}^{\text{intial}}$ is from Algorithm 1.
- Stopping condition of NRCG:

$$\frac{f(\mathbf{X}_{k-1}) - f(\mathbf{X}_k)}{f(\mathbf{X}_{k-1})} \leq \epsilon_{in}, \tag{12}$$

  - No need to solve the fixed-rank problem with high precision!
  - We set $\epsilon_{in} = 0.01$ in practice.

All these techniques are important to improve the efficiency.
Please find more details in the paper.


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


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


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


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


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


Candés, E. J. and Recht, B.

Exact matrix completion via convex optimization.
Found. Comput. Math., 9:717–772, 2009.


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


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


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


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


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

In *SIGIR*, 1999.


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


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


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


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


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


Lee, K. and Bresler, Y.
Admira: Atomic decomposition for minimum rank approximation.

IEEE Trans. on Inform. Theory, 56(9):4402–4416, 2010.


Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu.
Low-rank structure learning via nonconvex heuristic recovery.
IEEE Trans. Neural Netw. Learning Syst., 24(3):383–396, 2013.


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


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


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


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


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


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


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


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


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


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


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


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


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


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


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


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


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


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


Vandereycken, B.
Low-rank matrix completion by Riemannian optimization.
Technical report, 2012.

Siam J. Optim., 23(2):1214–1236, 2013.


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


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


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