CCF-CV 2017, Guangzhou, 4-28, 2017

Mingkui Tan

Riemannian Optimization and Its Application in Machine Learning and Computer Vision

Mingkui Tan

School of Software Engineering,
South China University of Technology, China

April 30, 2017

Outline

Unconstrained Optimization Revisited

Find a minimizer of a function without any constraints:

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathbb{R}^n. \tag{1}$$

In general, $f(\mathbf{x})$ is a differentiable function.

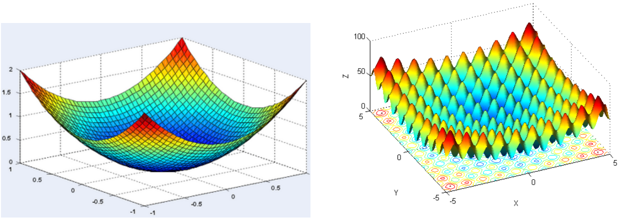


Figure: Examples of minimization

Unconstrained Optimization

General optimization scheme takes 3 iterative steps:

Algorithm 1 General Iterative Optimization Scheme

Given an initial point $\mathbf{x}_0$.

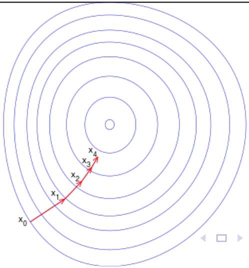
for $k = 0, 1, ...$ **do**

  Find a descent search direction $\mathbf{p}_k$;

  Find a suitable step size θ;

  Set $\mathbf{x}_{k+1} := \mathbf{x}_k + \theta_k \mathbf{p}_k$.

end



Unconstrained Optimization

- General optimization scheme contains 3 iterative steps:
- Two things need to consider:
  - Find a feasible search direction or descent direction $\mathbf{p}$
  - Find a suitable step size θ?
- $\mathbf{p}_k$ is a descent direction, if

$$\mathbf{p}_k^\top \nabla f(\mathbf{x}) < 0. \quad \text{Why?}$$

- Methods for adjusting step size
  - Exact line search
  - Fixed step step size
  - Diminishing step size: $\alpha_k \to 0$ and $\sum_k \alpha_k = \infty$
  - Backtracking Armijo line-search
  - Strong Wolfe-line Search

Constrained Optimization

- Find a minimizer with constraints:

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \Delta. \tag{2}$$

- Optimization over non-convex domain is very difficult:

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \theta_k \mathbf{p}_k$$

(a) Convex set (b) Non-convex set

Figure: Convex set: The line segment lies completely within the set. Non-convex set: Red part of the line segment lies outside of the set.

Non-convex but Simple Constraints

- Δ can be following non-convex sets:
  - Sphere constraint

$$\mathcal{S}^{n-1} = \{\mathbf{x} : ||\mathbf{x}||_2 = 1, \mathbf{x} \in \mathbb{R}^n\}$$

  - Orthogonal constraint

$$\mathcal{O} = \{\mathbf{X} : \mathbf{X}^\top \mathbf{X} = \mathbf{I}, \mathbf{X} \in \mathbb{R}^{n \times p}\}$$

  - Oblique constraint

$$\mathcal{OB} = \{\mathbf{X} : \mathrm{diag}(\mathbf{X}^\top \mathbf{X}) = \mathbf{I}, \mathbf{X} \in \mathbb{R}^{n \times p}\}$$

  - Low-rank constraint

$$\mathcal{M}_r = \{\mathbf{X} : \mathrm{rank}(\mathbf{x}) = r, \mathbf{X} \in \mathbb{R}^{n \times p}\}$$

- These constraints are hard to handle by traditional methods, e.g. penalty methods

Typical Applications

- Rayleigh quotient:

$$\min_{\mathbf{x}} \quad \mathbf{x}^\top \mathbf{A}\mathbf{x}, \quad \text{s.t.,} \quad ||\mathbf{x}|| = 1. \tag{3}$$

- Sparse PCA:

$$\min_{\mathbf{X}} \quad -\mathrm{tr}(\mathbf{X}^\top \mathbf{A}\mathbf{X}) + \lambda ||\mathbf{X}||_1, \quad \text{s.t.,} \quad \mathbf{X}^\top \mathbf{X} = \mathbf{I}. \tag{4}$$

- Matrix recovery

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{b} - \mathcal{A}(\mathbf{X})\|_2^2, \qquad \text{s.t.} \quad \mathrm{rank}(\mathbf{X}) = r. \tag{5}$$

Connections to Manifold

- Unit sphere $\mathcal{S}^2 = \{\mathbf{x} : ||\mathbf{x}||_2 = 1, \mathbf{x} \in \mathbb{R}^3\}$

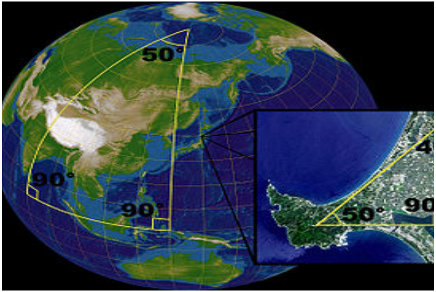


Figure: Example: The Earth surface is a two-dimensional Manifold

Connection to Manifolds

- Stiefel manifold $\mathcal{O} = \{\mathbf{X} : \mathbf{X}^\top \mathbf{X} = \mathbf{I}, \mathbf{X} \in \mathbb{R}^{n \times p}\}$
- Oblique manifold $\mathcal{OB} = \{\mathbf{X} : \mathrm{diag}(\mathbf{X}^\top \mathbf{X}) = \mathbf{I}, \mathbf{X} \in \mathbb{R}^{n \times p}\}$
- Low-rank manifold $\mathcal{M}_r = \{\mathbf{X} : \mathrm{rank}(\bar{\mathbf{x}}) = r, \mathbf{X} \in \mathbb{R}^{n \times p}\}$

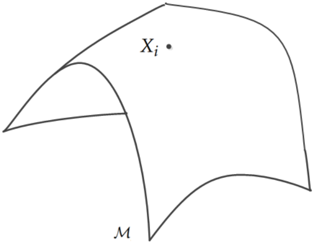


Figure: Demonstration of more general manifolds.

Riemannian Manifold

Riemannian Manifold

A Riemannian manifold $(\mathcal{M}, g)$ is a real smooth manifold $\mathcal{M}$ equipped with an Riemannian metric $g_{\mathbf{x}}$ on the tangent space $T_{\mathbf{x}}\mathcal{M}$ at each point $\mathbf{x}$ that varies smoothly from point to point.

- Smoothness
- Tangent space $T_{\mathbf{x}}\mathcal{M}$
- Riemannian metric, e.g.,

$$g_{\mathbf{x}}(\mathbf{A}, \mathbf{B}) = \mathrm{tr}(\mathbf{A}^\top \mathbf{B}), \quad \forall \mathbf{A}, \mathbf{B} \in \mathbf{T}_{\mathbf{x}}\mathcal{M}$$

Tangent Space

Tangent vector

A tangent vector on $\mathcal{M}$ at a point $\mathbf{X}$ is thought of as the "velocity" of a curve passing through the point $\mathbf{X}$.

Tangent space

The tangent space to $\mathcal{M}$ at $\mathbf{X}$, denoted by $T_{\mathbf{X}}\mathcal{M}$, is the set of all tangent vectors to $\mathcal{M}$ at $\mathbf{X}$.

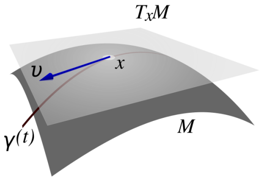


Figure: Tangent space

Riemannian gradient

Find a minimizer of a function under manifold constraints:

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{x} \in \mathcal{M}. \tag{6}$$

Gradient in vector space

Suppose $f(\mathbf{x})$ is differentiable, $\mathbf{G} = \nabla f(\mathbf{X}), \forall \mathbf{X} \in \mathcal{M}$

Riemannian Gradient

The Riemannian gradient is the orthogonal projection of $\mathbf{G}$ onto $T_{\mathbf{X}}M$,

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

Steepest Descent on Riemannian Manifold

Algorithm 2 Steepest Descent on Riemannian Manifold.

1: Initialize $\mathbf{X}_0$ and $\zeta_0 = \mathbf{0}$. Let $k = 0$.

2: Compute the Riemannian gradient $\mathrm{grad} f(\mathbf{X}_k)$.

3: Compute the decent direction $\zeta_k = -\mathrm{grad} f(\mathbf{X}_k)$.

4: Choose a suitable step size θ_k.

5: Update by doing *Retraction*:

$$\mathbf{X}_{k+1} = P_{\mathcal{M}_r}(\mathbf{X} + \theta\zeta) = R_{\mathbf{X}_k}(\theta_k \zeta_k)$$

6: Terminate if stopping conditions are achieved; otherwise, let $k = k + 1$ and go to step 1.

Two new terminologies:

- Riemannian gradient
- *Retraction*

Retraction

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

$$R_{\mathbf{X}}(\theta\zeta) = P_{\mathcal{M}_r}(\mathbf{X} + \theta\zeta) \quad (7)$$

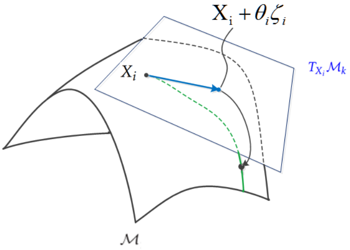
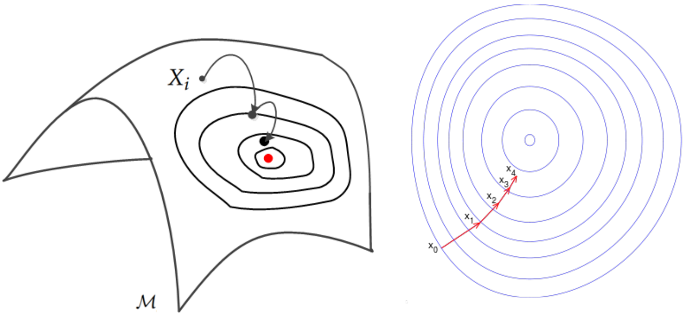


Figure: Retraction

Riemannian Optimization



(a) Riemannian optimization (b) Traditional optimization

Figure: Comparison of Riemanian optimization and traditional optimization

Conjugate Gradient Descent on Manifold

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

$$\zeta_k = -\text{grad} f(\mathbf{X}_k) + \beta_t \zeta_{k-1}, \tag{8}$$

where β_k can be calculated by Fletcher-Reeves (F-R) rule

- But notice:
  - $\text{grad} f(\mathbf{X}_k)$ and $\beta_t \zeta_{k-1}$ are in different tangent spaces! Thus, Eq. (8) is not valid!
  - We still need a new operator: *Vector Transport*:

$$\mathcal{T}_{\mathbf{X}_{k-1} \to \mathbf{X}_k}(\zeta_{k-1})$$

Vector transport

Vector Transport

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

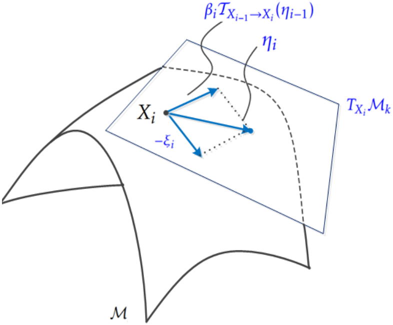


Figure: Conjugate search direction.

Nonlinear Conjugate Gradient Descent

Algorithm 3 Nonlinear Riemannian Conjugate Descent.

1: Initialize $\mathbf{X}_0$ and $\zeta_0 = \mathbf{0}$. Let $k = 0$.
2: Compute the Riemannian gradient $\mathbf{E_k} = \mathrm{grad} f(\mathbf{X}_k)$.
3: Compute the conjugate direction ζ_k according to

$$\zeta_k = -\mathbf{E_k} + \beta_k \mathcal{T}_{\mathbf{X}_{k-1}\to\mathbf{X}_k}(\zeta_{k-1}).$$

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

Matrix Recovery (MR)

Matrix Recovery (MR)

Given a matrix $\widehat{\mathbf{X}} \in \mathbb{R}^{m \times n}$, and linear measurements of by $\mathbf{b} = \mathcal{A}(\widehat{\mathbf{X}}) + \mathbf{e}$, where $\mathcal{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^{l}$ is a linear operator and $\mathbf{e}$ denotes noise. Matrix recover is to recover $\widehat{\mathbf{X}}$ by solving

$$\min_{\mathbf{X}} f(\mathbf{X}), \qquad \text{s.t.} \quad \text{rank}(\mathbf{X}) \leq r, \tag{9}$$

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

- The linear operator $\mathcal{A}$ measures partial information of $\widehat{\mathbf{X}}$.
- We use $\mathcal{A}^*$ as the adjoint operator.
- If $\mathcal{A}$ is a matrix $\mathbf{A}$, its adjoint operator is $\mathbf{A}^\top$

Application of Matrix Recovery

The definition of $\mathcal{A}$ depends on specific applications:

- Matrix completion [Recht et al.(2010)]:
- HDR imaging [Deng et al.(2013)]
- Matrix learning & factorizations [Laue(2012)]
- Low rank structure learning or clustering [Deng et al.(2013)]
- ....

Recommender System

	4	?		?	5
2	?		5		
	4	2	?		
		2		5	?
?	3		1		5

Figure: An example of a user/item rating matrix on movies.

Low Rank Assumption

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

Existing MR Methods

Minimizing $f(\mathbf{X})$ with rank constraint $\text{rank}(\mathbf{X}) \leq r$ NP-hard.

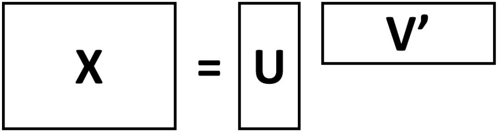


Figure: Matrix factorization: $\mathbf{X} = \mathbf{U}\mathbf{V}^\top$

Difficulty: the rank r is usually unknown.

- How to estimate r?

Some notations.

- 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 $\|\mathbf{X}\|_* = \|\boldsymbol{\sigma}\|_1 = \sum_i |\sigma_i|$.
- The condition number $\kappa_r(\mathbf{X})$ of $\mathbf{X}$ w.r.t. given rank r is $\kappa_r(\mathbf{X}) = \sigma_1 / \sigma_r$

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

$$\min_{\mathbf{X}} \|\mathbf{X}\|_*, \text{ s.t. } \mathcal{A}(\mathbf{X}) = \mathbf{b}, \tag{10}$$

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

$$\min_{\mathbf{X},\boldsymbol{\xi}} \lambda\|\mathbf{X}\|_* + \frac{1}{2}\|\boldsymbol{\xi}\|_2^2 \ : \ \boldsymbol{\xi} = \mathbf{b} - \mathcal{A}(\mathbf{X}), \tag{11}$$

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

Challenges

Avoid cold-started SVDs and estimate correct rank

Fixed-rank Methods

The fixed-rank methods solve the following problem:

$$\min_{\mathbf{X}} f(\mathbf{X}), \qquad \text{s.t.} \quad \text{rank}(\mathbf{X}) = r. \tag{12}$$

Here, r is supposed to be known.

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

Differential Geometry of Fixed-rank Matrices

CCF-CV 2017, Guangzhou, 4-28, 2017

Mingkui Tan

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

- Riemannian gradient is the orthogonal projection of $\mathbf{G} = \nabla f$ onto $T_{\mathbf{X}}\mathcal{M}_r$.
- The orthogonal projection of any $\mathbf{G} \in \mathbb{R}^{m\times n}$ onto $T_{\mathbf{X}}\mathcal{M}_r$ at $\mathbf{X} = \mathbf{U}\mathrm{diag}(\boldsymbol{\sigma})\mathbf{V}^\top$ is defined as

$$\mathrm{Grad} f(\mathbf{X}) = P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G}) : \mathbf{G} \mapsto P_U \mathbf{G} P_V + P_U^{\perp} \mathbf{G} P_V + P_U \mathbf{G} P_V^{\perp}, ($$

where $P_U = \mathbf{U}\mathbf{U}^\top$ and $P_U^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^\top$ for any $\mathbf{U} \in \mathrm{St}_r^m$.

Retraction:

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

$$R_{\mathbf{X}}(\mathbf{E}) = P_{\mathcal{M}_r}(\mathbf{X} + \mathbf{E}) = \sum_{i=1}^{r} \sigma_i \mathbf{p}_i \mathbf{q}_i^\top. \tag{14}$$

- $\sum_{i=1}^{r} \sigma_i \mathbf{p}_i \mathbf{q}_i^\top$ 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 nuclear-norm based methods [Vandereycken(2013)]
- Manifold optimization methods involve simple matrix-products, and easy for parallel computing

Disadvantages

- How do we know the rank r?
- 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

Motivations

- Fixed-rank methods require explicit knowledge of the rank
- Two Questions:
  1. How to avoid the rank estimation?
  2. How to avoid the convergence issue on ill-conditioned problems?

Active Subspace Search

- Ideally, we hope $\mathcal{A}(\mathbf{X}) - \mathbf{b} = \mathbf{0}$ or

$$\mathbf{G} = \mathcal{A}^*(\mathcal{A}(\mathbf{X}) - \mathbf{b}) = \mathbf{0}$$

- Residual: $\mathbf{Q} = \mathbf{G} - P_{T_{\mathbf{X}}\mathcal{M}_r}(\mathbf{G})$
- Select new subspaces of rank ρ from $\mathbf{Q}$
- Add new subspaces to the original manifold
- Essentially, we increase the rank by ρ

Algorithm 4 Riemannian Pursuit for MR.

1: Initialize $\mathbf{X}^0 = \mathbf{0}$ 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$(\mathbf{X}^{\text{intial}}, \epsilon_{in})$.

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

Convergence

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

Convergence speed

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. It may increase the rank until $t\rho = \min(m, n)$, leading to over-fitting issue.
- To avoid this, we propose to stop if

$$2(f(\mathbf{X}^{t-1}) - f(\mathbf{X}^{t}))/(\rho\|\mathbf{b}\|_2^2) \leq \epsilon_{out}, \qquad (15)$$

where ϵ_{out} is a predefined tolerance value.

- When it is stopped, a rank is returned!

Riemannian Pursuit

Algorithm 5 Riemannian Pursuit for MR.

1: Initialize $\mathbf{X}^0 = \mathbf{0}$ 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:

$$\min_{\mathbf{X}} \quad f(\mathbf{X}), \quad \text{s.t.} \quad \text{rank}(\mathbf{X}) = t\rho. \tag{16}$$

- ρ is much smaller than $\widehat{r}$, thus $t\rho < \widehat{r}$.
- 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.

Algorithm 6 NRCG for solving fixed-rank minimization problem.

1: Initialize $\mathbf{X}_1 = \mathbf{X}^{\text{intial}}$ and $\zeta_0 = \mathbf{0}$. Let $k = 1$.
2: Compute the Riemannian gradient $\mathbf{E_k} = \text{grad} f(\mathbf{X}_k)$.
3: Compute the conjugate direction ζ_k according to

$$\zeta_k = -\mathbf{E_k} + \beta_\mathbf{t} \mathcal{T}_{\mathbf{X_{k-1}} \to \mathbf{X_k}}(\zeta_{\mathbf{k-1}}).$$

4: Choose a step size θ_k satisfying the strong Wolfe conditions, and set $\mathbf{X}_{k+1} = R_{\mathbf{X}_k}(\theta_k \zeta_k)$.
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

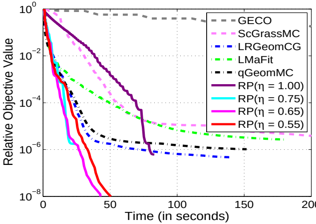
Methods for comparison:

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

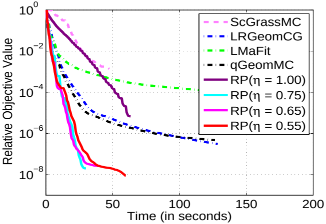
Toy experimental setting

- We generate ground-truth $\widehat{\mathbf{X}}$ by $\widehat{\mathbf{X}} = \widehat{\mathbf{U}}\text{diag}(\widehat{\boldsymbol{\sigma}})\widehat{\mathbf{V}}^\top + \mathbf{e}$, 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, 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.

Experiments on Collaborative Filtering Tasks

Table: Experimental results on real-world datasets. 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.
- RP is much faster if the model selection cost is considered.

HDR Imaging

HDR imaging [Oh, *et al.* , TPAMI 2015]

High dynamic range (HDR) imaging aims to

- Remove the outliers from a set of low dynamic range (LDR) images
- Generate a ghost-free HDR image

Compared Methods

- Ours with RP

$$\min_{\mathbf{X}} ||\mathcal{P}_\Omega(\mathbf{Y}) - \mathcal{P}_\Omega(\mathbf{X})||_1, \text{ s.t. } \operatorname{rank}(\mathbf{X}) = r, \tag{17}$$

- *PSSV* [Oh, *et al.* , TPAMI, 2016]

$$\min_{\mathbf{X},\mathbf{E}} ||\mathbf{X}||_r + \lambda||\mathbf{E}||_{\mathbf{1}}, \quad \text{s.t. } \mathbf{Y} = \mathbf{X} + \mathbf{E}, \tag{18}$$

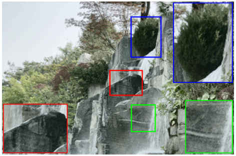
- *LMAFIT* [Shen, *et al.* , 2012]

$$\min_{\mathbf{X},\mathbf{U},\mathbf{V}} ||\mathbf{Y} - \mathbf{X}||_1, \quad \text{s.t. } \mathbf{X} = \mathbf{U}\mathbf{V}^\top. \tag{19}$$

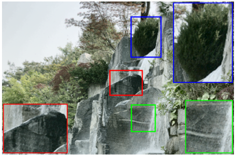
- *RegL1*

$$\min_{\mathbf{U},\mathbf{V}} ||\mathbf{Y} - \mathbf{U}\mathbf{V}^\top||_1 + \lambda||\mathbf{V}||_*, \quad \text{s.t. } \mathbf{U}^\top\mathbf{U} = \mathbf{I}. \tag{20}$$

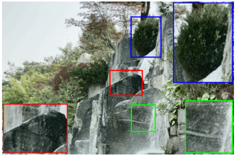
HDR Results



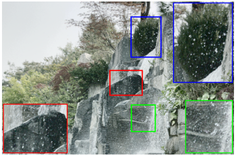
(a) Our method



(b) PSSV



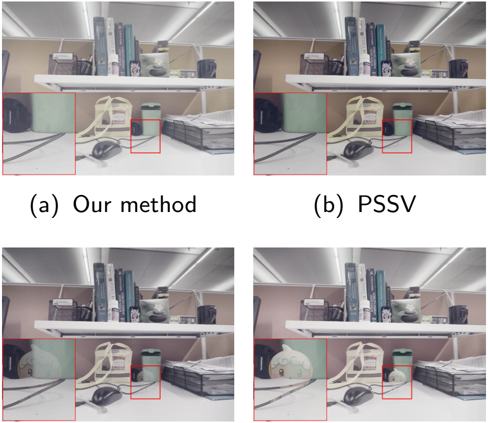
(c) LMAFIT



(d) RegL1

Figure: HDR imaging results on `Waterfall`. *Water drops* around the target waterfall are the target outliers to be removed.

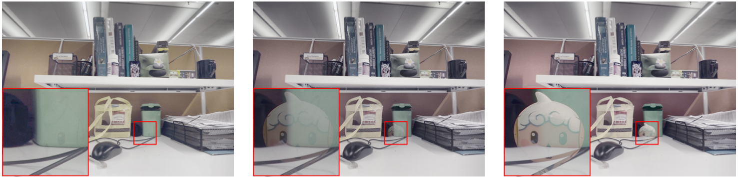
HDR Results



(a) Our method (b) PSSV (c) LMAFIT (d) RegL1

Figure: HDR imaging results on `Desk`. The *toy* is the target outlier to be removed.

HDR Results



(a) Our method (b) LMAFIT (c) RegL1

Figure: HDR imaging results on `Desk-full`. The *toy* is the target outlier to be removed.

References

- **Mingkui Tan**, Ivor W. Tsang, Li Wang, Bart Vandereycken, Sinno Jialin Pan. Riemannian Pursuit for Big Matrix Recovery, **ICML**, **2014**.
- Yan Yan, **Mingkui Tan**, Ivor W. Tsang, Yi Yang, Chengqi Zhang, Qinfeng Shi. Scalable Maximum Margin Matrix Factorization by Active Riemannian Subspace Search, **IJCAI**, **2015**.
- **Mingkui Tan**, Shijie Xiao, Junbin Gao, Dong Xu, Anton van den Hengel, Qinfeng Shi. Riemannian Proximal Gradient for Large-scale Trace-norm Minimization, **CVPR**, **2016**.
- Absil, P.-A. and Mahony, R. and Sepulchre, R. Optimization Algorithms on Matrix Manifolds. Princeton University Press, 2008.

Conclusions

- Riemannian optimization is effective in dealing with manifold constraints.
- We propose a Riemannian Pursuit (RP) for tackling Big Matrix Recovery problems.
- We apply RP to handle various machine learning and computer vision problems.

Thank you for your attention!

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.