Riemannian Proximal Gradient Methods

Wen Huang

Xiamen University

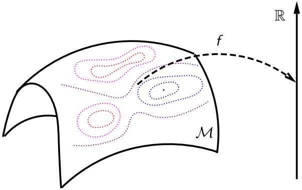
Dec. 18, 2019

This is joint work with Ke Wei at Fudan University.

Problem Statement

Optimization on Manifolds with Structure:

$$\min_{x \in \mathcal{M}} F(x) = f(x) + g(x),$$

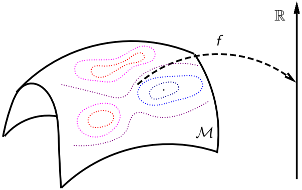


- $\mathcal{M}$ is a Riemannian manifold;
- f is Lipschitz continuously differentiable and may be nonconvex; and
- g is continuous and convex, but may be not differentiable.

Problem Statement

Optimization on Manifolds with Structure:

$$\min_{x \in \mathcal{M}} F(x) = f(x) + g(x),$$



- $\mathcal{M}$ is a Riemannian manifold;
- f is Lipschitz continuously differentiable and may be nonconvex; and
- g is continuous and convex, but may be not differentiable.

Applications: sparse PCA, sparse blind deconvolution, sparse low rank image representation, etc [JTU03, GHT15, SQ16, ZLK$^+$17]

Existing Nonsmooth Optimization on Manifolds

$F : \mathcal{M} \to \mathbb{R}$ is Lipschitz continuous

- Huang (2013), Gradient sampling method without convergence analysis.
- Grohs and Hosseini (2015), Two ϵ-subgradient-based optimization methods using line search strategy and trust region strategy, respectively. Any limit point is a critical point.
- Hosseini and Uschmajew (2017), Gradient sampling method and any limit point is a critical point.
- Hosseini and Huang and Yousefpour (2018), Merge ϵ-subgradient-based and quasi-Newton ideas and show any limit point is a critical point.

Existing Nonsmooth Optimization on Manifolds

$$F : \mathcal{M} \to \mathbb{R} \text{ is convex}$$

- Zhang and Sra (2016), subgradient-based method and function value converges to the optimal $O(1/\sqrt{k})$.
- Ferreira and Oliveira (2002) and Bento, Ferreira and Melo (2017), proximal point method and function value converges to the optimal $O(1/k)$ on Hadamard manifold.
- Liu, Shang, Cheng, Cheng, and Jiao (2017), F is Lipschitz-continuously differentiable, function value converges to the optimal $O(1/k^2)$

Existing Nonsmooth Optimization on Manifolds

$F = f + g$, where f is L-con, and g is non-smooth

- Chen, Ma, So, and Zhang (2018), A proximal gradient method with global convergence
- Huang and Wei (2019), A FISTA on manifolds with global convergence
- Huang and Wei (2019), A Riemannian proximal gradient method and its invariant with acceleration. Convergence rate analyses are given

A Euclidean Proximal Gradient Method

Optimization with Structure: $\mathcal{M} = \mathbb{R}^{n \times m}$

$$\min_{x \in \mathbb{R}^{n \times m}} F(x) = f(x) + g(x), \tag{1}$$

Proximal gradient method and its invariants are excellent methods for solving (1).

1

A Euclidean Proximal Gradient Method

Optimization with Structure: $\mathcal{M} = \mathbb{R}^{n\times m}$

$$\min_{x\in\mathbb{R}^{n\times m}} F(x) = f(x) + g(x), \qquad (1)$$

Proximal gradient method and its invariants are excellent methods for solving (1).

A proximal gradient method[1]:

initial iterate:x_0,

$$\begin{cases} d_k = \arg\min_{p\in\mathbb{R}^{n\times m}} \langle \nabla f(x_k), p\rangle + \frac{L}{2}\|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

[1]The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k\rangle + \frac{L}{2}\|x - x_k\|^2 + g(x)$.

A Euclidean Proximal Gradient Method

Optimization with Structure: $\mathcal{M} = \mathbb{R}^{n \times m}$

$$\min_{x \in \mathbb{R}^{n \times m}} F(x) = f(x) + g(x), \qquad (1)$$

Proximal gradient method and its invariants are excellent methods for solving (1).

A proximal gradient method[1]:

initial iterate: x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

A Euclidean Proximal Gradient Method

Optimization with Structure: $\mathcal{M} = \mathbb{R}^{n \times m}$

$$\min_{x \in \mathbb{R}^{n \times m}} F(x) = f(x) + g(x), \qquad (1)$$

Proximal gradient method and its invariants are excellent methods for solving (1).

A proximal gradient method[1]:

initial iterate:x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2}\|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;
- L: greater than the Lipschitz constant of ∇f;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2}\|x - x_k\|^2 + g(x)$.

A Euclidean Proximal Gradient Method

Optimization with Structure: $\mathcal{M} = \mathbb{R}^{n \times m}$

$$\min_{x \in \mathbb{R}^{n \times m}} F(x) = f(x) + g(x), \qquad (1)$$

Proximal gradient method and its invariants are excellent methods for solving (1).

A proximal gradient method[1]:

initial iterate: x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;
- L: greater than the Lipschitz constant of ∇f;
- Proximal mapping: easy to compute;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

A Euclidean Proximal Gradient Method

Optimization with Structure: $\mathcal{M} = \mathbb{R}^{n \times m}$

$$\min_{x \in \mathbb{R}^{n \times m}} F(x) = f(x) + g(x), \qquad (1)$$

Proximal gradient method and its invariants are excellent methods for solving (1).

A proximal gradient method[1]:

initial iterate: x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p), & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k. & \text{(Update iterates)} \end{cases}$$

- $g = 0$: reduce to steepest descent method;
- L: greater than the Lipschitz constant of ∇f;
- Proximal mapping: easy to compute;
- Any limit point is a critical point;

[1] The update rule: $x_{k+1} = \arg\min_x \langle \nabla f(x_k), x - x_k \rangle + \frac{L}{2} \|x - x_k\|^2 + g(x)$.

Convergence Rates

Assumption

$\min_{x \in \mathbb{R}^{n \times m}} F(x) = f(x) + g(x)$, with convex f;

- $O(1/k)$ sublinear convergence rate:

$$F(x_k) - F(x_*) \leq C/k, \text{ for a constant } C;$$

Convergence Rates

Assumption

$\min_{x \in \mathbb{R}^{n \times m}} F(x) = f(x) + g(x)$, with convex f;

- $O(1/k)$ sublinear convergence rate:

$$F(x_k) - F(x_*) \leq C/k, \text{ for a constant } C;$$

- Optimal gradient method: $O(1/k^2)$ [Dar83, Nes83]

Convergence Rates

Assumption

$\min_{x \in \mathbb{R}^{n \times m}} F(x) = f(x) + g(x)$, with convex f;

- $O(1/k)$ sublinear convergence rate:

$$F(x_k) - F(x_*) \leq C/k, \text{ for a constant } C;$$

- Optimal gradient method: $O(1/k^2)$ [Dar83, Nes83]
- Here, we consider FISTA [BT09]

$$\text{initial iterate: } x_0 \text{ and let } y_0 = x_0,\ t_0 = 1,$$

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(y_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(y_k + p), \\ x_{k+1} = y_k + d_k, \\ t_{k+1} = \frac{1 + \sqrt{4t_k^2 + 1}}{2}, \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}} (x_{k+1} - x_k). \end{cases}$$

Difficulties in the Riemannian Setting

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

In the Riemannian setting:

- How to define the proximal mapping?
- Can be solved cheaply?
- Share the same convergence rate?

A Riemannian Proximal Gradient Method in [CMSZ18]

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

A Riemannian proximal mapping [CMSZ18]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;

- Only works for embedded submanifold;

[1][CMSZ18]: S. Chen, S. Ma, M. C. So, and T. Zhang, Proximal gradient method for nonsmooth optimization over the Stiefel manifold. arXiv:1811.00980v2

A Riemannian Proximal Gradient Method in [CMSZ18]

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

A Riemannian proximal mapping [CMSZ18]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;

- Only works for embedded submanifold;
- Proximal mapping is defined in tangent space;

[1][CMSZ18]: S. Chen, S. Ma, M. C. So, and T. Zhang, Proximal gradient method for nonsmooth optimization over the Stiefel manifold. arXiv:1811.00980v2

A Riemannian Proximal Gradient Method in [CMSZ18]

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

A Riemannian proximal mapping [CMSZ18]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;

- Only works for embedded submanifold;
- Proximal mapping is defined in tangent space;
- Convex programming;

[1][CMSZ18]: S. Chen, S. Ma, M. C. So, and T. Zhang, Proximal gradient method for nonsmooth optimization over the Stiefel manifold. arXiv:1811.00980v2

A Riemannian Proximal Gradient Method in [CMSZ18]

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

ManPG [CMSZ18]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;

- Only works for embedded submanifold;
- Proximal mapping is defined in tangent space;
- Convex programming;
- Solved efficiently for the Stiefel manifold by a semi-Newton algorithm [XLWZ18];

A Riemannian Proximal Gradient Method in [CMSZ18]

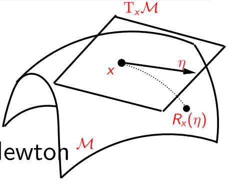
Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

ManPG [CMSZ18]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;
2. $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$ with an appropriate step size α_k;

- Only works for embedded submanifold;
- Proximal mapping is defined in tangent space;
- Convex programming;
- Solved efficiently for the Stiefel manifold by a semi-Newton algorithm [XLWZ18];



A Riemannian Proximal Gradient Method in [CMSZ18]

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

ManPG [CMSZ18]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;
2. $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$ with an appropriate step size α_k;

- Convergence to a stationary point [HW19];

A Riemannian Proximal Gradient Method in [CMSZ18]

Euclidean proximal mapping

$$d_k = \arg \min_{p \in \mathbb{R}^{n \times m}} \langle \nabla f(x_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(x_k + p)$$

ManPG [CMSZ18]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$;
2. $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$ with an appropriate step size α_k;

- Convergence to a stationary point [HW19];
- No convergence rate analysis (expect rate $O(1/k)$ if f is convex);

New Riemannian Proximal Gradient Methods

GOAL:

1. **Numerical aspect**:
   An accelerated Riemannian proximal gradient method with good numerical performance

2. **Theoretical aspect**:
   An accelerated Riemannian proximal gradient method with convergence rate analysis and good numerical performance for some instances

Numerical aspect: A New Riemannian Proximal Gradient

GOAL: Develop an accelerated Riemannian proximal gradient method with good numerical performance

Numerical aspect: A New Riemannian Proximal Gradient

GOAL: Develop an accelerated Riemannian proximal gradient method with good numerical performance

A Riemannian FISTA with a safeguard

initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$;

1. Invoke a safeguard every N iterations;
2. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{y_k}\mathcal{M}} \langle \mathrm{grad}\, f(y_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + g(y_k + \eta)$;
3. $x_{k+1} = R_{y_k}(\eta_k)$;
4. $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
5. Compute $y_{k+1} = R_{x_{k+1}}\left(\frac{1-t_k}{t_{k+1}} R_{x_{k+1}}^{-1}(x_k)\right)$;

Numerical aspect: A New Riemannian Proximal Gradient

A Riemannian FISTA with a safeguard

A Riemannian FISTA with a safeguard

initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$;

1. Invoke a safeguard every N iterations;
2. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{y_k}\mathcal{M}} \langle \mathrm{grad}\, f(y_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + g(y_k + \eta)$;
3. $x_{k+1} = R_{y_k}(\eta_k)$;
4. $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
5. Compute $y_{k+1} = R_{x_{k+1}}\left(\frac{1-t_k}{t_{k+1}} R_{x_{k+1}}^{-1}(x_k)\right)$;

- Run proximal gradient method every N iterations
- If the iterate by FISTA has larger function value than that by proximal gradient, then the safeguard takes effect.

Numerical aspect: A New Riemannian Proximal Gradient

A Riemannian FISTA with a safeguard

A Riemannian FISTA with a safeguard

initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$;

1. Invoke a safeguard every N iterations;
2. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{y_k}\mathcal{M}} \langle \mathrm{grad} f(y_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + g(y_k + \eta)$;
3. $x_{k+1} = R_{y_k}(\eta_k)$;
4. $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
5. Compute $y_{k+1} = R_{x_{k+1}}\left(\frac{1-t_k}{t_{k+1}} R_{x_{k+1}}^{-1}(x_k)\right)$;

FISTA initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^{n\times m}} \langle \nabla f(y_k), p \rangle + \frac{L}{2}\|p\|_F^2 + g(y_k + p), \\ x_{k+1} = y_k + d_k, \\ t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}, \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k). \end{cases}$$

Numerical aspect: A New Riemannian Proximal Gradient

A Riemannian FISTA with a safeguard

A Riemannian FISTA with a safeguard

initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$;

1. Invoke a safeguard every N iterations;
2. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{y_k} \mathcal{M}} \langle \mathrm{grad} f(y_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(y_k + \eta)$;
3. $x_{k+1} = R_{y_k}(\eta_k)$;
4. $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
5. Compute $y_{k+1} = R_{x_{k+1}} \left(\frac{1-t_k}{t_{k+1}} R_{x_{k+1}}^{-1}(x_k) \right)$;

A Riemannian generalization: $R_x(\eta) = x + \eta$, $R_x^{-1}(y) = y - x$:

$$y_{k+1} = \underbrace{x_{k+1} + \frac{1-t_k}{t_{k+1}} \underbrace{(x_k - x_{k+1})}_{\text{replaced by } R_{x_{k+1}}^{-1}(x_k)}}_{\text{replaced by } R_{x_{k+1}}\left(\frac{1-t_k}{t_{k+1}} R_{x_{k+1}}^{-1}(x_k)\right)},$$

Numerical aspect: A New Riemannian Proximal Gradient

A Riemannian FISTA with a safeguard

A Riemannian FISTA with a safeguard

initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$;

1. Invoke a safeguard every N iterations;
2. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{y_k}\mathcal{M}} \langle \mathrm{grad} f(y_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + g(y_k + \eta)$;
3. $x_{k+1} = R_{y_k}(\eta_k)$;
4. $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
5. Compute $y_{k+1} = R_{x_{k+1}}\left(\frac{1-t_k}{t_{k+1}} R_{x_{k+1}}^{-1}(x_k)\right)$;

- Works well in practice
- Convergence globally
- No convergence rate analysis

Theoretical aspect: A New Riemannian Proximal Gradient

GOAL: Develop an accelerated Riemannian proximal gradient method with convergence rate analysis and good numerical performance for some instances

Theoretical aspect: A New Riemannian Proximal Gradient

GOAL: Develop an accelerated Riemannian proximal gradient method with convergence rate analysis and good numerical performance for some instances

A New Riemannian Proximal Gradient Method

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \underbrace{\langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2}\|\eta\|_{x_k}^2}_{\text{Riemannian metric}} + g(\underbrace{R_{x_k}(\eta)}_{\text{replace } x_k + \eta});$
2. $x_{k+1} = R_{x_k}(\eta_k);$

- General framework for Riemannian optimization;

Theoretical aspect: A New Riemannian Proximal Gradient

GOAL: Develop an accelerated Riemannian proximal gradient method with convergence rate analysis and good numerical performance for some instances

---

A New Riemannian Proximal Gradient Method

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \underbrace{\langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2}_{\text{Riemannian metric}} + g(\underbrace{R_{x_k}(\eta)}_{\text{replace } x_k + \eta});$
2. $x_{k+1} = R_{x_k}(\eta_k);$

- General framework for Riemannian optimization;
- The tangent space may be too rough to approximate manifold for convergence analysis;

Theoretical aspect: A New Riemannian Proximal Gradient

GOAL: Develop an accelerated Riemannian proximal gradient method with convergence rate analysis and good numerical performance for some instances

A New Riemannian Proximal Gradient Method

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \underbrace{\langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2}_{\text{Riemannian metric}} + g(\underbrace{R_{x_k}(\eta)}_{\text{replace } x_k + \eta});$
2. $x_{k+1} = R_{x_k}(\eta_k)$;

- General framework for Riemannian optimization;
- The tangent space may be too rough to approximate manifold for convergence analysis;
- Step size can be fixed to be 1;

Assumptions and Convergence Result

Assumption:

1. f is Lipschitz continuously differentiable in a Riemannian sense (L-retraction-smooth);

Assumptions and Convergence Result

Assumption:

1. f is Lipschitz continuously differentiable in a Riemannian sense (L-retraction-smooth);

Definition

A function $h : \mathcal{M} \to \mathbb{R}$ is called L-retraction-smooth with respect to a retraction R in $\mathcal{N} \subset \mathcal{M}$ if for any $x \in \mathcal{N}$ and any $\mathcal{S}_x \subset \mathrm{T}_x \mathcal{M}$ such that $R_x(\mathcal{S}_x) \subset \mathcal{N}$, we have that $q_x = h \circ R_x$ satisfies

$$q_x(\eta) \leq q_x(\xi) + \langle \mathrm{grad}\, q_x(\xi), \eta - \xi \rangle_x + \frac{L}{2} \|\eta - \xi\|_x^2 \quad \forall \eta, \xi \in \mathcal{S}_x.$$

Assumptions and Convergence Result

Assumption:

1. f is Lipschitz continuously differentiable in a Riemannian sense (L-retraction-smooth);

Theoretical results:

- For any accumulation point x_* of $\{x_k\}$, x_* is a stationary point, i.e., $0 \in \partial F(x_*)$.

Assumptions and Convergence Rate

Additional Assumptions:

- f is convex in a Riemannian sense (retraction-convex);

Assumptions and Convergence Rate

Additional Assumptions:

- f is convex in a Riemannian sense (retraction-convex);

Definition

A function $h : \mathcal{M} \to \mathbb{R}$ is called retraction-convex with respect to a retraction R in $\mathcal{N} \subseteq \mathcal{M}$ if for any $x \in \mathcal{N}$ and any $\mathcal{S}_x \subseteq \mathrm{T}_x \mathcal{M}$ such that $R_x(\mathcal{S}_x) \subseteq \mathcal{N}$, there exists a tangent vector $\zeta \in \mathrm{T}_x \mathcal{M}$ such that $q_x = h \circ R_x$ satisfies

$$q_x(\eta) \geq q_x(\xi) + \langle \zeta, \eta - \xi \rangle_x \;\; \forall \eta, \xi \in \mathcal{S}_x. \tag{2}$$

Note that $\zeta = \operatorname{grad} q_x(\xi)$ if h is differentiable; otherwise, ζ is any subgradient of q_x at ξ.

Assumptions and Convergence Rate

Additional Assumptions:

- f is convex in a Riemannian sense (retraction-convex);
- Retraction approximately satisfies the triangle relation:

$$\left| \|\xi_x - \eta_x\|_x^2 - \|\zeta_y\|_y^2 \right| \leq \kappa \|\eta_x\|_x^2, \text{ for a constant } \kappa$$

where $\eta_x = R_x^{-1}(y)$, $\xi_x = R_x^{-1}(z)$, $\zeta_y = R_y^{-1}(z)$.

Assumptions and Convergence Rate

Additional Assumptions:

- f is convex in a Riemannian sense (retraction-convex);
- Retraction approximately satisfies the triangle relation:

$$\left| \|\xi_x - \eta_x\|_x^2 - \|\zeta_y\|_y^2 \right| \leq \kappa \|\eta_x\|_x^2, \text{ for a constant } \kappa$$

where $\eta_x = R_x^{-1}(y)$, $\xi_x = R_x^{-1}(z)$, $\zeta_y = R_y^{-1}(z)$.

Table: Exponential mapping on the Stiefel manifold with the Euclidean metric $\langle \eta_x, \xi_x \rangle_x = \mathrm{trace}(\eta_x^T \xi_x)$. Left $= \left| \|\xi_x - \eta_x\|_x^2 - \|\zeta_y\|_y^2 \right|$

$(n,p)=(10,1)$		$(n,p)=(10,4)$		$(n,p)=(10,10)$	
$\|\eta_x\|$	Left	$\|\eta_x\|$	Left	$\|\eta_x\|$	Left
5.00_{-2}	7.83_{-5}	5.00_{-2}	1.83_{-5}	5.00_{-2}	2.14_{-6}
2.50_{-2}	1.80_{-5}	2.50_{-2}	4.27_{-6}	2.50_{-2}	4.72_{-7}
1.25_{-2}	4.25_{-6}	1.25_{-2}	1.01_{-6}	1.25_{-2}	1.11_{-7}
6.25_{-3}	1.03_{-6}	6.25_{-3}	2.46_{-7}	6.25_{-3}	2.68_{-8}
3.12_{-3}	2.54_{-7}	3.12_{-3}	6.05_{-8}	3.13_{-3}	6.61_{-9}

Assumptions and Convergence Rate

Additional Assumptions:

- f is convex in a Riemannian sense (retraction-convex);
- Retraction approximately satisfies the triangle relation:

$$\left| \|\xi_x - \eta_x\|_x^2 - \|\zeta_y\|_y^2 \right| \leq \kappa \|\eta_x\|_x^2, \text{ for a constant } \kappa$$

where $\eta_x = R_x^{-1}(y)$, $\xi_x = R_x^{-1}(z)$, $\zeta_y = R_y^{-1}(z)$.

Table: Exponential mapping on the Stifel manifold with the canonical metric $\langle \eta_x, \xi_x \rangle_x = \mathrm{trace}(\eta_x^T (I - XX^T/2)\xi_x)$. Left $= \left| \|\xi_x - \eta_x\|_x^2 - \|\zeta_y\|_y^2 \right|$

$(n,p) = (10,2)$		$(n,p) = (10,4)$		$(n,p) = (10,9)$	
$\|\eta_x\|$	Left	$\|\eta_x\|$	Left	$\|\eta_x\|$	Left
5.00_{-2}	3.55_{-5}	5.00_{-2}	1.15_{-5}	5.00_{-2}	8.39_{-6}
2.50_{-2}	8.06_{-6}	2.50_{-2}	2.58_{-6}	2.50_{-2}	1.89_{-6}
1.25_{-2}	1.90_{-6}	1.25_{-2}	6.08_{-7}	1.25_{-2}	4.45_{-7}
6.25_{-3}	4.61_{-7}	6.25_{-3}	1.47_{-7}	6.25_{-3}	1.08_{-7}
3.13_{-3}	1.13_{-7}	3.13_{-3}	3.63_{-8}	3.12_{-3}	2.66_{-8}

Assumptions and Convergence Rate

Additional Assumptions:

- f is convex in a Riemannian sense (retraction-convex);
- Retraction approximately satisfies the triangle relation:

$$\left| \|\xi_x - \eta_x\|_x^2 - \|\zeta_y\|_y^2 \right| \leq \kappa \|\eta_x\|_x^2, \text{ for a constant } \kappa$$

where $\eta_x = R_x^{-1}(y)$, $\xi_x = R_x^{-1}(z)$, $\zeta_y = R_y^{-1}(z)$.

Theoretical results:

- Convergence rate $O(1/k)$:

$$F(x_k) - F(x_*) \leq \frac{1}{k} \left(\frac{L}{2} \|R_{x_0}^{-1}(x_*)\|_{x_0}^2 + \frac{L\kappa C}{2} (F(x_0) - F(x_*)) \right).$$

A Riemannian FISTA

A Riemannian FISTA

initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$;

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{y_k} \mathcal{M}} \langle \mathrm{grad}\, f(y_k), \eta \rangle_{y_k} + \frac{L}{2} \|\eta\|_{y_k}^2 + g(R_{y_k}(\eta))$;
2. $x_{k+1} = R_{y_k}(\eta_k)$;
3. $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
4. Compute $y_{k+1} = R_{y_k}\left(\frac{t_{k+1}+t_k-1}{t_{k+1}} \eta_{y_k} - \frac{t_k-1}{t_{k+1}} R_{y_k}^{-1}(x_k) \right)$;

FISTA initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^{n\times m}} \langle \nabla f(y_k), p \rangle + \frac{L}{2} \|p\|_F^2 + g(y_k + p), \\ x_{k+1} = y_k + d_k, \\ t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}, \\ y_{k+1} = x_{k+1} + \frac{t_k-1}{t_{k+1}} (x_{k+1} - x_k). \end{cases}$$

A Riemannian FISTA

A Riemannian FISTA

initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$;

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{y_k}\mathcal{M}} \langle \mathrm{grad}\, f(y_k), \eta \rangle_{y_k} + \frac{L}{2}\|\eta\|_{y_k}^2 + g(R_{y_k}(\eta))$;
2. $x_{k+1} = R_{y_k}(\eta_k)$;
3. $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
4. Compute $y_{k+1} = R_{y_k}\left(\frac{t_{k+1}+t_k-1}{t_{k+1}}\eta_{y_k} - \frac{t_k-1}{t_{k+1}} R_{y_k}^{-1}(x_k)\right)$;

A Riemannian generalization:

$$\begin{aligned} y_{k+1} =& y_k + \frac{t_{k+1} + t_k - 1}{t_{k+1}}(x_{k+1} - y_k) - \frac{t_k - 1}{t_{k+1}}(x_k - y_k) \\ =& x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k), \end{aligned}$$

Assumptions and Convergence Rate

Additional Assumptions:

- There exists a constant $\tilde{\kappa}$ such that

$$\Big| \|(t_{k+1}-1)(R_{y_k}^{-1}(x_{k+1}) - R_{y_k}^{-1}(y_{k+1})) + R_{y_k}^{-1}(x_*) - R_{y_k}^{-1}(y_{k+1})\|_{y_k}^2 - \|(t_{k+1}-1)R_{y_{k+1}}^{-1}(x_{k+1}) + R_{y_{k+1}}^{-1}(x_*)\|_{y_{k+1}}^2 \Big| \leq \tilde{\kappa} \|R_{y_k}^{-1}(y_{k+1})\|_{y_k}^2.$$

- $\phi(k) := \sum_{i=0}^{k} \|R_{y_k}^{-1}(y_{k+1})\|_{y_k}^2$ increases on the order of $O((k+1)^\theta)$ for $\theta \in [0,1]$, i.e., $\frac{\phi(k)}{(k+1)^\theta} < C_\phi$ for all k.

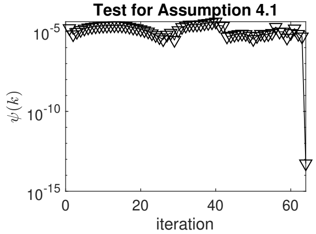
Assumptions and Convergence Rate

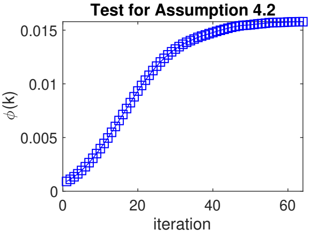
Additional Assumptions:

- There exists a constant $\tilde{\kappa}$ such that

$$\Big| \|(t_{k+1} - 1)(R_{y_k}^{-1}(x_{k+1}) - R_{y_k}^{-1}(y_{k+1})) + R_{y_k}^{-1}(x_*) - R_{y_k}^{-1}(y_{k+1})\|_{y_k}^2 - \|(t_{k+1} - 1)R_{y_{k+1}}^{-1}(x_{k+1}) + R_{y_{k+1}}^{-1}(x_*)\|_{y_{k+1}}^2 \Big| \leq \tilde{\kappa} \|R_{y_k}^{-1}(y_{k+1})\|_{y_k}^2.$$

- $\phi(k) := \sum_{i=0}^{k} \|R_{y_k}^{-1}(y_{k+1})\|_{y_k}^2$ increases on the order of $O((k+1)^\theta)$ for $\theta \in [0, 1]$, i.e., $\frac{\phi(k)}{(k+1)^\theta} < C_\phi$ for all k.





Assumptions and Convergence Rate

Additional Assumptions:

- There exists a constant $\tilde{\kappa}$ such that

$$\Big| \|(t_{k+1}-1)(R_{y_k}^{-1}(x_{k+1}) - R_{y_k}^{-1}(y_{k+1})) + R_{y_k}^{-1}(x_*) - R_{y_k}^{-1}(y_{k+1})\|_{y_k}^2 - \|(t_{k+1}-1)R_{y_{k+1}}^{-1}(x_{k+1}) + R_{y_{k+1}}^{-1}(x_*)\|_{y_{k+1}}^2 \Big| \leq \tilde{\kappa}\|R_{y_k}^{-1}(y_{k+1})\|_{y_k}^2.$$

- $\phi(k) := \sum_{i=0}^{k} \|R_{y_k}^{-1}(y_{k+1})\|_{y_k}^2$ increases on the order of $O((k+1)^{\theta})$ for $\theta \in [0,1]$, i.e., $\frac{\phi(k)}{(k+1)^{\theta}} < C_{\phi}$ for all k.

Theoretical results:

- Convergence rate $O(1/k^2)$ if $\theta = 0$:

$$F(x_k) - F(x_*) \leq \frac{2L}{k^2}\|R_{x_0}^{-1}(x_*)\|_{x_0}^2 + \frac{2L\tilde{\kappa}C_{\phi}}{k^{2-\theta}}(F(x_0) - F(x_*)).$$

The Proposed Algorithm

A Riemannian FISTA with a safeguard

initial iterate: x_0 and let $y_0 = x_0$, $t_0 = 1$;

1. Invoke a safeguard every N iterations;
2. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{y_k} \mathcal{M}} \langle \mathrm{grad}\, f(y_k), \eta \rangle_{y_k} + \frac{L}{2} \|\eta\|_{y_k}^2 + g(R_{y_k}(\eta))$;
3. $x_{k+1} = R_{y_k}(\eta_k)$;
4. $t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2}$;
5. Compute $y_{k+1} = R_{y_k}\left(\frac{t_{k+1}+t_k-1}{t_{k+1}} \eta_{y_k} - \frac{t_k-1}{t_{k+1}} R_{y_k}^{-1}(x_k) \right)$;

- Convergence globally;
- Convergence rate $\frac{1}{k^{2-\theta}}$ if previous assumptions hold and safeguard takes effect for finite iterations;

Riemannian subproblem

$$\eta_u = \arg \min_{\eta \in \mathrm{T}_u \mathcal{M}} \ell_u(\eta) := \langle \nabla f(u), \eta \rangle_u + \frac{L}{2} \|\eta\|_u^2 + g(R_u(\eta))$$

Riemannian subproblem

$$\eta_u = \arg \min_{\eta \in \mathrm{T}_u \mathcal{M}} \ell_u(\eta) := \langle \nabla f(u), \eta \rangle_u + \frac{L}{2} \|\eta\|_u^2 + g(R_u(\eta))$$

In some cases, the subproblem can be solved by exploiting the structure of the manifold;

Riemannian subproblem

$$\eta_u = \arg \min_{\eta \in \mathrm{T}_u \mathcal{M}} \ell_u(\eta) := \langle \nabla f(u), \eta \rangle_u + \frac{L}{2} \|\eta\|_u^2 + g(R_u(\eta))$$

Solving the Riemannian Proximal Mapping

initial iterate: $\eta_0 \in \mathrm{T}_u \mathcal{M}$, $\sigma \in (0,1)$, $k = 0$;

1. $v_k = R_u(\eta_k)$;
2. Compute
$$\xi_k^* = \arg \min_{\xi \in \mathrm{T}_{v_k} \mathcal{M}} \langle \mathcal{T}_{R_{\eta_k}}^{-\sharp} (\mathrm{grad}\, f(u) + \tilde{L} \eta_k), \xi \rangle_u + \frac{\tilde{L}}{4} \|\xi\|_F^2 + g(v_k + \xi);$$
3. Find $\alpha > 0$ such that $\ell_u(\eta_k + \alpha \mathcal{T}_{R_{\eta_k}}^{-1} \xi_k^*) < \ell_u(\eta_k) - \sigma \alpha \|\xi_k^*\|_u^2$;
4. $\eta_{k+1} = \eta_k + \alpha \mathcal{T}_{R_{\eta_k}}^{-1} \xi_k^*$, $k \leftarrow k + 1$ and goto Step 1;

Above algorithm is used if the ambient space is $\mathbb{R}^n$

Riemannian subproblem

$$\eta_u = \arg \min_{\eta \in \mathrm{T}_u \mathcal{M}} \ell_u(\eta) := \langle \nabla f(u), \eta \rangle_u + \frac{L}{2} \|\eta\|_u^2 + g(R_u(\eta))$$

Solving the Riemannian Proximal Mapping

initial iterate: $\eta_0 \in \mathrm{T}_u \mathcal{M}$, $\sigma \in (0,1)$, $k = 0$;

1. $v_k = R_u(\eta_k)$;
2. Compute
$$\xi_k^* = \arg \min_{\xi \in \mathrm{T}_{v_k} \mathcal{M}} \langle \mathcal{T}_{R_{\eta_k}}^{-\sharp} (\mathrm{grad}\, f(u) + \tilde{L}\eta_k), \xi \rangle_u + \frac{\tilde{L}}{4} \|\xi\|_F^2 + g(v_k + \xi);$$
3. Find $\alpha > 0$ such that $\ell_u(\eta_k + \alpha \mathcal{T}_{R_{\eta_k}}^{-1} \xi_k^*) < \ell_u(\eta_k) - \sigma\alpha \|\xi_k^*\|_u^2$;
4. $\eta_{k+1} = \eta_k + \alpha \mathcal{T}_{R_{\eta_k}}^{-1} \xi_k^*$, $k \leftarrow k + 1$ and goto Step 1;

Above algorithm is used if the ambient space is $\mathbb{R}^n$

Riemannian subproblem

$$\eta_u = \arg \min_{\eta \in \mathrm{T}_u \mathcal{M}} \ell_u(\eta) := \langle \nabla f(u), \eta \rangle_u + \frac{L}{2} \|\eta\|_u^2 + g(R_u(\eta))$$

Solving the Riemannian Proximal Mapping

initial iterate: $\eta_0 \in \mathrm{T}_u \mathcal{M}$, $\sigma \in (0,1)$, $k = 0$;

1. $v_k = R_u(\eta_k)$;
2. Compute
$$\xi_k^* = \arg \min_{\xi \in \mathrm{T}_{v_k} \mathcal{M}} \langle \mathcal{T}_{R_{\eta_k}}^{-\sharp} (\mathrm{grad}\, f(u) + \tilde{L}\eta_k), \xi \rangle_u + \frac{\tilde{L}}{4} \|\xi\|_F^2 + g(v_k + \xi);$$
3. Find $\alpha > 0$ such that $\ell_u(\eta_k + \alpha \mathcal{T}_{R_{\eta_k}}^{-1} \xi_k^*) < \ell_u(\eta_k) - \sigma \alpha \|\xi_k^*\|_u^2$;
4. $\eta_{k+1} = \eta_k + \alpha \mathcal{T}_{R_{\eta_k}}^{-1} \xi_k^*$, $k \leftarrow k+1$ and goto Step 1;

Above algorithm is used if the ambient space is $\mathbb{R}^n$

Riemannian subproblem

$$\eta_u = \arg\min_{\eta \in \mathrm{T}_u \mathcal{M}} \ell_u(\eta) := \langle \nabla f(u), \eta \rangle_u + \frac{L}{2} \|\eta\|_u^2 + g(R_u(\eta))$$

Solving the Riemannian Proximal Mapping

initial iterate: $\eta_0 \in \mathrm{T}_u \mathcal{M}$, $\sigma \in (0,1)$, $k = 0$;

1. $v_k = R_u(\eta_k)$;
2. Compute
$$\xi_k^* = \arg\min_{\xi \in \mathrm{T}_{v_k} \mathcal{M}} \langle \mathcal{T}_{R_{\eta_k}}^{-\sharp} (\mathrm{grad}\, f(u) + \tilde{L} \eta_k), \xi \rangle_u + \frac{\tilde{L}}{4} \|\xi\|_F^2 + g(v_k + \xi);$$
3. Find $\alpha > 0$ such that $\ell_u(\eta_k + \alpha \mathcal{T}_{R_{\eta_k}}^{-1} \xi_k^*) < \ell_u(\eta_k) - \sigma \alpha \|\xi_k^*\|_u^2$;
4. $\eta_{k+1} = \eta_k + \alpha \mathcal{T}_{R_{\eta_k}}^{-1} \xi_k^*$, $k \leftarrow k + 1$ and goto Step 1;

An application of [CMSZ18] if $R_u^{-1}(\eta)$ exists.

Numerical Experiments

Sparse PCA problem [GHT15]

$$\min_{X \in OB(p,n)} \|X^T A^T A X - D^2\|_F^2 + \lambda \|X\|_1,$$

where $A \in \mathbb{R}^{m \times n}$, D is the diagonal matrix with dominant singular values of A, $OB(p, n) = \{X \in \mathbb{R}^{n \times p} \mid \mathrm{diag}(X^T X) = I_p\}$, $p \leq m$;

Numerical Experiments

Solve the proximal mapping:

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_x \mathcal{M}} \langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta));$$

- Exponential mapping (each column): $R_x(\eta_x) = x \cos(\|\eta_x\|) + \frac{\eta_x}{\|\eta_x\|} \sin(\|\eta_x\|)$;

Numerical Experiments

Solve the proximal mapping:

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_x \mathcal{M}} \langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta));$$

- Exponential mapping (each column): $R_x(\eta_x) = x \cos(\|\eta_x\|) + \frac{\eta_x}{\|\eta_x\|} \sin(\|\eta_x\|)$;
- Explore the fact that the following problem has a closed solution:

$$\min_{x \in OB(p,n)} \|x - y\|_F^2 + \frac{1}{2\lambda} \|x\|_1 \text{ for any } y \in \mathbb{R}^{n \times p}.$$

Numerical Experiments

Solve the proximal mapping:

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_x \mathcal{M}} \langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta));$$

- Exponential mapping (each column):
  $R_x(\eta_x) = x \cos(\|\eta_x\|) + \frac{\eta_x}{\|\eta_x\|} \sin(\|\eta_x\|)$;
- Explore the fact that the following problem has a closed solution:

$$\min_{x \in OB(p,n)} \|x - y\|_F^2 + \frac{1}{2\lambda} \|x\|_1 \text{ for any } y \in \mathbb{R}^{n \times p}.$$

- A conditional gradient (Frank-Wolfe) method is used;

Numerical Experiments

Solve the proximal mapping:

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_x \mathcal{M}} \langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta));$$

- Exponential mapping (each column): $R_x(\eta_x) = x \cos(\|\eta_x\|) + \frac{\eta_x}{\|\eta_x\|} \sin(\|\eta_x\|)$;
- Explore the fact that the following problem has a closed solution:

$$\min_{x \in OB(p,n)} \|x - y\|_F^2 + \frac{1}{2\lambda} \|x\|_1 \text{ for any } y \in \mathbb{R}^{n \times p}.$$

- A conditional gradient (Frank-Wolfe) method is used;
- Numerically, using approximate 2 iterations is enough for high accuracy;

Numerical Experiments

Solve the proximal mapping:

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_x \mathcal{M}} \langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta));$$

- Exponential mapping (each column): $R_x(\eta_x) = x \cos(\|\eta_x\|) + \frac{\eta_x}{\|\eta_x\|} \sin(\|\eta_x\|)$;
- Explore the fact that the following problem has a closed solution:

$$\min_{x \in OB(p,n)} \|x - y\|_F^2 + \frac{1}{2\lambda} \|x\|_1 \text{ for any } y \in \mathbb{R}^{n \times p}.$$

- A conditional gradient (Frank-Wolfe) method is used;
- Numerically, using approximate 2 iterations is enough for high accuracy;

Numerical Experiments

Table: An average result of 10 random tests. $n = 128$, $m = 20$, $r = 4$. $\delta = (L\|x_{k+1} - x_k\|)^2$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	δ	spar.	navar
3	ManPG	11791	1.40	8.33_1	5.11_{-6}	0.54	0.86
	RPG	11679	0.94	8.33_1	5.11_{-6}	0.54	0.86
	ManPG-Ada	1398	0.30	8.33_1	1.67_{-3}	0.54	0.86
	A-ManPG	273	0.09	8.33_1	9.19_{-4}	0.54	0.86
	A-RPG	263	0.06	8.33_1	1.12_{-3}	0.54	0.86

- ManPG: the method in [CMSZ18];
- RPG: the new Riemannian proximal gradient without acceleration;
- A-ManPG: Use similar technique to accelerate ManPG;
- A-RPG: the new Riemannian proximal gradient with acceleration;

Numerical Experiments

Table: An average result of 10 random tests. $n = 128$, $m = 20$, $r = 4$. $\delta = (L\|x_{k+1} - x_k\|)^2$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	δ	spar.	navar
3	ManPG	11791	1.40	8.33_1	5.11_{-6}	0.54	0.86
	RPG	11679	0.94	8.33_1	5.11_{-6}	0.54	0.86
	ManPG-Ada	1398	0.30	8.33_1	1.67_{-3}	0.54	0.86
	A-ManPG	273	0.09	8.33_1	9.19_{-4}	0.54	0.86
	A-RPG	263	0.06	8.33_1	1.12_{-3}	0.54	0.86

ManPG-Ada:

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{\tilde{L}}{2}\|\eta\|_F^2 + g(x_k + \eta)$;
2. $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$ with an appropriate step size α_k;
3. Update $\tilde{L}$;

Numerical Experiments

Table: An average result of 10 random tests. $n = 128$, $m = 20$, $r = 4$. $\delta = (L\|x_{k+1} - x_k\|)^2$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	δ	spar.	navar
3	ManPG	11791	1.40	8.33_1	5.11_{-6}	0.54	0.86
	RPG	11679	0.94	8.33_1	5.11_{-6}	0.54	0.86
	ManPG-Ada	1398	0.30	8.33_1	1.67_{-3}	0.54	0.86
	A-ManPG	273	0.09	8.33_1	9.19_{-4}	0.54	0.86
	A-RPG	263	0.06	8.33_1	1.12_{-3}	0.54	0.86

- ManPG and RPG: Stop when $\delta < 10^{-8} nr$;
- A-ManPG and A-RPG: Stop when F is smaller than the minimum of ManPG and RPG;

Numerical Experiments

Table: An average result of 10 random tests. $n = 128$, $m = 20$, $r = 4$. $\delta = (L\|x_{k+1} - x_k\|)^2$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	δ	spar.	navar
3	ManPG	11791	1.40	8.33_1	5.11_{-6}	0.54	0.86
	RPG	11679	0.94	8.33_1	5.11_{-6}	0.54	0.86
	ManPG-Ada	1398	0.30	8.33_1	1.67_{-3}	0.54	0.86
	A-ManPG	273	0.09	8.33_1	9.19_{-4}	0.54	0.86
	A-RPG	263	0.06	8.33_1	1.12_{-3}	0.54	0.86

- spar.: sparsity of the solution;
- navar: the adjusted variance normalized by the variance from the standard PCA;

Numerical Experiments

Table: An average result of 10 random tests. $n = 128$, $m = 20$, $r = 4$. $\delta = (L\|x_{k+1} - x_k\|)^2$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	δ	spar.	navar
3	ManPG	11791	1.40	8.33_1	5.11_{-6}	0.54	0.86
	RPG	11679	0.94	8.33_1	5.11_{-6}	0.54	0.86
	ManPG-Ada	1398	0.30	8.33_1	1.67_{-3}	0.54	0.86
	A-ManPG	273	0.09	8.33_1	9.19_{-4}	0.54	0.86
	A-RPG	263	0.06	8.33_1	1.12_{-3}	0.54	0.86

- PG without acceleration is slower than PG with acceleration;
- RPG is slightly faster ManPG in term of computational time;

Numerical Experiments

Table: An average result of 10 random tests. $n = 128$, $m = 20$, $r = 4$. $\delta = (L\|x_{k+1} - x_k\|)^2$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	δ	spar.	navar
3	ManPG	11791	1.40	8.33_1	5.11_{-6}	0.54	0.86
	RPG	11679	0.94	8.33_1	5.11_{-6}	0.54	0.86
	ManPG-Ada	1398	0.30	8.33_1	1.67_{-3}	0.54	0.86
	A-ManPG	273	0.09	8.33_1	9.19_{-4}	0.54	0.86
	A-RPG	263	0.06	8.33_1	1.12_{-3}	0.54	0.86

- PG without acceleration is slower than PG with acceleration;
- RPG is slightly faster ManPG in term of computational time;

Numerical Experiments

Table: An average result of 10 random tests. $n = 128$, $m = 20$, $r = 4$. $\delta = (L\|x_{k+1} - x_k\|)^2$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	δ	spar.	navar
3	ManPG	11791	1.40	8.33_1	5.11_{-6}	0.54	0.86
	RPG	11679	0.94	8.33_1	5.11_{-6}	0.54	0.86
	ManPG-Ada	1398	0.30	8.33_1	1.67_{-3}	0.54	0.86
	A-ManPG	273	0.09	8.33_1	9.19_{-4}	0.54	0.86
	A-RPG	263	0.06	8.33_1	1.12_{-3}	0.54	0.86

- ManPG and RPG: similarly; and A-ManPG and A-RPG: similarly; in term of:
  - number of iterations;
  - function values;
  - sparsity;
  - adjusted variance;

Numerical Experiments

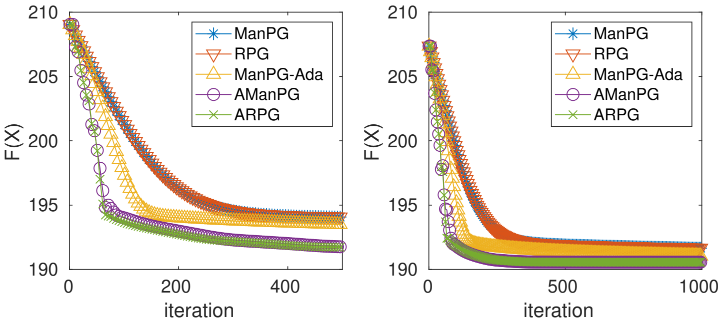


Figure: Two typical runs of ManPG, RPG, A-ManPG, and A-RPG for the Sparse PCA problem. $n = 1024$, $p = 4$, $\lambda = 2$, $m = 20$.

Numerical Experiments

Sparse PCA problem (Another model) [CMSZ18, HW19]

$$\min_{X \in \mathrm{St}(p,n)} -\mathrm{trace}(X^T A^T A X) + \lambda \|X\|_1,$$

where $A \in \mathbb{R}^{m \times n}$ is a data matrix.

Numerical Experiments

Solve the proximal mapping:

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_x \mathcal{M}} \langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta));$$

- Exponential mapping:

$$\mathrm{Exp}_X(\eta_X) = \begin{bmatrix} X & Q \end{bmatrix} \exp\left(\begin{bmatrix} \Omega & -R^T \\ R & 0 \end{bmatrix} \right) \begin{bmatrix} I_p \\ 0 \end{bmatrix},$$

where $\Omega = X^T \eta_X$, Q and R are from the compact QR factorization of $(I - XX^T)\eta_X$.

Numerical Experiments

Solve the proximal mapping:

$$\eta_k = \arg\min_{\eta \in \mathrm{T}_x \mathcal{M}} \langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + g(R_{x_k}(\eta));$$

- Exponential mapping:

$$\mathrm{Exp}_X(\eta_X) = \begin{bmatrix} X & Q \end{bmatrix} \exp\left(\begin{bmatrix} \Omega & -R^T \\ R & 0 \end{bmatrix} \right) \begin{bmatrix} I_p \\ 0 \end{bmatrix},$$

  where $\Omega = X^T \eta_X$, Q and R are from the compact QR factorization of $(I - XX^T)\eta_X$.
- Ingredients for the algorithm on Page 17:
  - R^{-1} by iterative methods [Zim17]
  - $\mathcal{T}_R^{-\sharp}$ by iterative methods

Numerical Experiments

Lemma

The adjoint operator of the inverse differentiated retraction is

$$\mathcal{T}_{\eta_X}^{-\sharp}\xi_X = \begin{bmatrix} X & Q_1 \end{bmatrix} \exp\left(\begin{bmatrix} \Omega_{\eta_X} & -R_1^T \\ R_1 & 0_{p\times p} \end{bmatrix}\right) \begin{bmatrix} X & Q_1 \end{bmatrix}^T \omega_x + \left(I - \begin{bmatrix} X & Q_1 \end{bmatrix} \begin{bmatrix} X & Q_1 \end{bmatrix}^T\right) \omega_x,$$

where $\omega_X = X\Omega_{\zeta_Y} + QR_2$, $Y = \mathrm{Exp}_X(\eta_X)$, $Q_1 R_1 = (I - XX^T)\eta_X$ *and* $Q_2 \tilde{R}_2 = (I - [XQ_1][XQ_1]^T)\xi_X$ *are qr decompositions,* $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$, $\tilde{M}_1 = \begin{bmatrix} \Omega_{\eta_X} & -R_1^T & 0_{p\times p} \\ R_1 & 0_{p\times p} & 0_{p\times p} \\ 0_{p\times p} & 0_{p\times p} & 0_{p\times p} \end{bmatrix}$, $\tilde{Z}\tilde{\Lambda}\tilde{Z}^H = \tilde{M}_1$, *and* Ω_{ζ_Y} *and* R_2 *are solutions of* $\tilde{Z}^H \begin{bmatrix} X & Q \end{bmatrix}^T \xi_X =$

$$\left(\left(\tilde{Z}^H \mathrm{Exp}_X(\tilde{M}_1) \begin{bmatrix} \Omega_{\zeta_Y} & -R_2^T \\ R_2 & 0_{2p\times 2p} \end{bmatrix} \tilde{Z}\right) \odot \overline{\Phi}\right) \tilde{Z}^H \begin{bmatrix} I_p \\ 0_{2p\times p} \end{bmatrix}.$$

Numerical Experiments

Table: An average result of 10 random tests. $n = 1024$, $m = 20$, $r = 4$. $\delta = (L\|x_{k+1} - x_k\|)^2$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	δ	spar.	navar
3	ManPG	1572	0.92	-7.28	4.76_{-5}	0.64	0.74
	RPG	1464	5.46	-7.28	4.06_{-5}	0.64	0.74
	ManPG-Ada	376	0.22	-7.28	3.99_{-4}	0.64	0.74
	A-ManPG	110	0.20	-7.28	1.06_{-3}	0.64	0.74
	A-RPG	88	1.61	-7.28	2.05_{-4}	0.64	0.74

- Same notation, same stopping criterion, same parameter setting;
- New approaches take more time due to excessive cost on R^{-1} and $\mathcal{T}^{-\sharp}$;
- New approaches take less iterations;

Numerical Experiments

Table: An average result of 10 random tests. $n = 1024$, $m = 20$, $r = 4$. $\delta = (L\|x_{k+1} - x_k\|)^2$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	δ	spar.	navar
3	ManPG	1572	0.92	-7.28	4.76_{-5}	0.64	0.74
	RPG	1464	5.46	-7.28	4.06_{-5}	0.64	0.74
	ManPG-Ada	376	0.22	-7.28	3.99_{-4}	0.64	0.74
	A-ManPG	110	0.20	-7.28	1.06_{-3}	0.64	0.74
	A-RPG	88	1.61	-7.28	2.05_{-4}	0.64	0.74

- Same notation, same stopping criterion, same parameter setting;
- New approaches take more time due to excessive cost on R^{-1} and $\mathcal{T}^{-\sharp}$;
- New approaches take less iterations;

Numerical Experiments

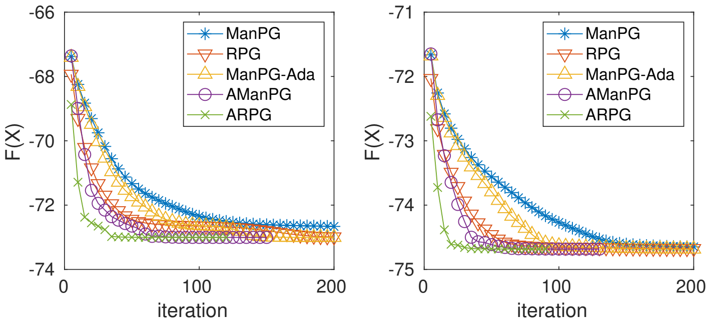


Figure: Two typical runs of ManPG, RPG, A-ManPG, and A-RPG for the Sparse PCA problem. $n = 1024$, $p = 4$, $\lambda = 2$, $m = 20$.

Acceleration for SPCA on the Stiefel manifold

Scaled proximal mapping:

$$\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + g(x_k + \eta)$$

$$\Longrightarrow \eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2}\|\eta\|_{W}^2 + g(x_k + \eta)$$

where $\|\eta\|_W^2 = \mathrm{vec}(\eta)^T W \mathrm{vec}(\eta)$ and W is symmetric positive definite.

Acceleration for SPCA on the Stiefel manifold

Scaled proximal mapping:

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$$

$$\Longrightarrow \eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_W^2 + g(x_k + \eta)$$

where $\|\eta\|_W^2 = \mathrm{vec}(\eta)^T W \mathrm{vec}(\eta)$ and W is symmetric positive definite.

- Difficult to solve in general

Acceleration for SPCA on the Stiefel manifold

Scaled proximal mapping:

$$\eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + g(x_k + \eta)$$

$$\Longrightarrow \eta_k = \arg \min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_{W}^2 + g(x_k + \eta)$$

where $\|\eta\|_W^2 = \mathrm{vec}(\eta)^T W \mathrm{vec}(\eta)$ and W is symmetric positive definite.

- Difficult to solve in general
- Diagonal matrix W inspired by the Riemannian Hessian of the smooth term.

Acceleration for SPCA on the Stiefel manifold

The diagonal weight W

- Riemannian Hessian of $f : \mathrm{St}(p,n) \to \mathbb{R} : X \mapsto -\mathrm{trace}(X^T A^T A X)$:

$$\begin{aligned} \mathrm{Hess}\, f(X)[\eta_X] =& P_{\mathrm{T}_X \mathrm{St}(p,n)}(-2A^T A \eta_X + 2\eta_X (X^T A^T A X)), \\ & \forall \eta_X \in \mathrm{T}_X \mathrm{St}(p,n) \end{aligned}$$

- An np-by-np matrix representation of $\mathrm{Hess}\, f(X)$:

$$\begin{aligned} \langle \eta_X, \mathrm{Hess}\, f(X)[\eta_X] \rangle =& \langle \eta_X, -2A^T A \eta_X + 2\eta_X (X^T A^T A X) \rangle \\ =& \langle \mathrm{vec}(\eta_X), J \mathrm{vec}(\eta_X) \rangle, \end{aligned}$$

where $J = -2I_p \otimes (A^T A) + 2(X^T A^T A X) \otimes I_n$.

- The diagonal matrix $W = \max(\mathrm{diag}(J), \tau I_{np})$.

Acceleration for SPCA on the Stiefel manifold

Numerical experiments

Table: An average result of 20 random runs for the random data: $r = 4$, $n = 3000$ and $m = 40$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	$\|\eta_{z_k}\|$	sparsity	variance
2.5	ManPG-D	1538	1.67	-1.48_1	1.09_{-3}	0.65	0.72
2.5	ManPG	2155	2.20	-1.48_1	1.09_{-3}	0.65	0.72
2.5	ManPG-Ada-D	469	0.60	-1.48_1	1.03_{-3}	0.65	0.72
2.5	ManPG-Ada	508	0.60	-1.48_1	1.04_{-3}	0.65	0.72
2.5	AManPG-D	201	0.39	-1.48_1	1.02_{-3}	0.65	0.72
2.5	AManPG	237	0.43	-1.49_1	1.05_{-3}	0.65	0.72

Acceleration for SPCA on the Stiefel manifold

Numerical experiments

Table: The result for the DNA methylation data: $r = 4$, $n = 24589$ and $m = 113$. The subscript k indicates a scale of 10^k.

λ	Algo	iter	time	f	$\|\eta_{z_k}\|$	sparsity	variance
6.0	ManPG-D	706	7.37	-7.74_3	3.11_{-3}	0.29	0.96
6.0	ManPG	2206	20.10	-7.74_3	3.14_{-3}	0.29	0.96
6.0	ManPG-Ada-D	369	4.58	-7.74_3	3.03_{-3}	0.29	0.96
6.0	ManPG-Ada	957	10.18	-7.74_3	3.11_{-3}	0.29	0.96
6.0	AManPG-D	93	2.33	-7.74_3	2.91_{-3}	0.29	0.96
6.0	AManPG	183	3.46	-7.74_3	2.96_{-3}	0.29	0.96

Summary

- Propose first Riemannian proximal gradient methods with convergence rate analyses;
- Propose Riemannian proximal gradient methods with acceleration;
- Apply the methods to sparse PCA problems on the oblique manifold and the Stiefel manifold;
- Compare the new proximal gradient method with the existing proximal gradient method;

References I

A. Beck and M. Teboulle.
A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
SIAM Journal on Imaging Sciences, 2(1):183–202, January 2009.
doi:10.1137/080716542.

Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang.
Proximal gradient method for nonsmooth optimization over the Stiefel manifold.
arXiv:1811.00980, 2018.

John Darzentas.
Problem Complexity and Method Efficiency in Optimization.
1983.

Matthieu Genicot, Wen Huang, and Nickolay T. Trendafilov.
Weakly correlated sparse components with nearly orthonormal loadings.
In *Geometric Science of Information*, pages 484–490, 2015.

W. Huang and K. Wei.
Extending FISTA to Riemannian optimization for sparse PCA.
arXiv:1909.05485, 2019.

Ian T. Jolliffe, Nickolay T. Trendafilov, and Mudassir Uddin.
A modified principal component technique based on the Lasso.
Journal of Computational and Graphical Statistics, 12(3):531–547, 2003.

Y. E. Nesterov.
A method for solving the convex programming problem with convergence rate \$O(1/k^2)\$.
Dokl. Akas. Nauk SSSR (In Russian), 269:543–547, 1983.

References II

J. Shi and C. Qi.
Low-rank sparse representation for single image super-resolution via self-similarity learning.
In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 1424–1428, Sep. 2016.

Xiantao Xiao, Yongfeng Li, Zaiwen Wen, and Liwei Zhang.
A regularized semi-smooth newton method with projection steps for composite convex programs.
Journal of Scientific Computing, 76(1):364–389, Jul 2018.

R. Zimmermann.
A Matrix-Algebraic Algorithm for the Riemannian Logarithm on the Stiefel Manifold under the Canonical Metric.
SIAM Journal on Matrix Analysis and Applications, 38(2):322–342, 2017.

Y. Zhang, Y. Lau, H.-W. Kuo, S. Cheung, A. Pasupathy, and J. Wright.
On the global geometry of sphere-constrained sparse blind deconvolution.
In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.