A Series of Talks on Riemannian Optimization

Nonsmooth Optimization: Difficulties from Euclidean to Riemannian

Wen Huang

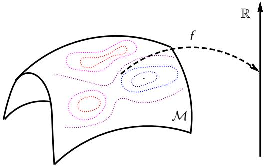
Xiamen University

September 17, 2025

Hunan University

Problem Statement

Optimization on Manifolds with Structure:



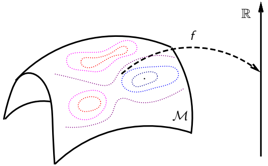
$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x),$$

- $\mathcal{M}$ is a finite-dimensional Riemannian manifold;
- f is smooth and may be nonconvex; and
- $h(x)$ is continuous and convex but may be nonsmooth;

Problem Statement

Optimization on Manifolds with Structure:

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



- $\mathcal{M}$ is a finite-dimensional Riemannian manifold;
- f is smooth and may be nonconvex; and
- $h(x)$ is continuous and convex but may be nonsmooth;

Applications: sparse PCA [ZHT06], compressed model [OLCO13], sparse partial least squares regression [CSG$^+$18], sparse inverse covariance estimation [BESS19], sparse blind deconvolution [ZLK$^+$17], and clustering [HWGVD22].

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, 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

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

- Zhang and Sra (2016), subgradient-based method and function value converges to the optimal $O(1/\sqrt{k})$.
- Ferreira and Oliveira (2002) proximal point method, convergence using convexity
  Bento, da Cruz Neto and Oliveira (2011), convergence using Kurdyka-Łojasiewicz (KL); and
  Bento, Ferreira, and Melo (2017), function value converges to the optimal $O(1/k)$ on Hadamard manifold using convexity

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
- Xiao, Liu, and Yuan (2021), Infeasible approach over the Stiefel manifold
- Zhou, Bao, and Ding (2022), An augmented Lagrangian method on matrix manifolds
- Huang and Wei (2021-2023), A Riemannian proximal gradient method, an inexact Riemannian proximal gradient method, and a modified FISTA on embedded manifolds
- Wang and Yang (2023), A proximal quasi-Newton method on manifolds on the Stiefel manifold
- Huang, Meng, Gallivan, and Van Dooren (2023), An inexact proximal gradient method on embedded submanifolds
- Beck and Rosset (2023), A dynamic smoothing technique

Content

Optimization with Structure:

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x).$$

- Proximal gradient methods
- Accelerated proximal gradient methods
- A proximal Newton method

[HW2021]: W. Huang and K. Wei, Riemannian proximal gradient methods, Mathematics Programming, 194, 371-413, 2022.
[HW2023]: An inexact Riemannian proximal gradient method, Computational Optimization and Applications, 85, 1-32, 2023
[HWGV2023]: A Riemannian optimization approach to clustering problems, arxiv, 2023
[SAHJV2023]: A Riemannian proximal Newton method, SIAM Journal on Optimization, 34:1, pp. 654-681, 2024

Content

Optimization with Structure:

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x).$$

- Proximal gradient methods
  - Euclidean version
  - Riemannian version in [CMSZ20]
  - Riemannian version in [HW21a]
- Accelerated proximal gradient methods
- A proximal Newton method

Proximal Gradient Method

Euclidean version

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

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x).$$

Proximal Gradient Method

Euclidean version

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

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x).$$

---

initial iterate: x_0,

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

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

Proximal Gradient Method

Euclidean version

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

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x).$$

---

initial iterate: x_0,

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

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

Proximal Gradient Method

Euclidean version

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

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x).$$

---

initial iterate: x_0,

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

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

Proximal Gradient Method

Euclidean version

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

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x).$$

---

initial iterate: x_0,

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

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

Proximal Gradient Method

Euclidean version

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

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x).$$

---

initial iterate: x_0,

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

- $h = 0$: 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;

Proximal Gradient Method

Euclidean version

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

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x).$$

---

initial iterate: x_0,

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

- $h = 0$: 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;
- $O\left(\frac{1}{k}\right)$ sublinear convergence rate for convex f and h;

Proximal Gradient Method

Euclidean version

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

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x).$$

---

initial iterate: x_0,

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

- $h = 0$: 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;
- $O\left(\frac{1}{k}\right)$ sublinear convergence rate for convex f and h;
- Linear convergence rate for strongly convex f and convex h;

Proximal Gradient Method

Euclidean version

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

$$\min_{x \in \mathbb{R}^n} F(x) = f(x) + h(x).$$

---

initial iterate: x_0,

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

- $h = 0$: 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;
- $O\left(\frac{1}{k}\right)$ sublinear convergence rate for convex f and h;
- Linear convergence rate for strongly convex f and convex h;
- Local convergence rate by KL property;

Proximal Gradient Method

Riemannian versions

Optimization with Structure: $\mathcal{M}$

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x).$$

Proximal Gradient Method

Riemannian versions

Optimization with Structure: $\mathcal{M}$

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x).$$

Euclidean proximal mapping

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

In the Riemannian setting:

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

Proximal Gradient Method

Riemannian version in [CMSZ20]

A Riemannian proximal mapping [CMSZ20]

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

- Only works for embedded submanifold;

[CMSZ18]: S. Chen, S. Ma, M. C. So, and T. Zhang, Proximal gradient method for nonsmooth optimization over the Stiefel manifold. SIAM Journal on Optimization, 30(1):210-239, 2020.

Proximal Gradient Method

Riemannian version in [CMSZ20]

A Riemannian proximal mapping [CMSZ20]

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

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

Proximal Gradient Method

Riemannian version in [CMSZ20]

A Riemannian proximal mapping [CMSZ20]

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

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

Proximal Gradient Method

Riemannian version in [CMSZ20]

[CMSZ20]

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

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

[XLWZ18]: X. Xiao, Y. Li, Z. Wen, and L. Zhang, A regularized semi-smooth Newton method with projection steps for composite convex programs. Journal of Scientific Computing, 76(1):364-389, 2018

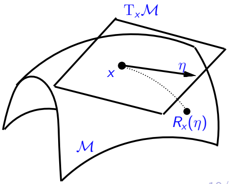
Proximal Gradient Method

Riemannian version in [CMSZ20]

ManPG [CMSZ20]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2} \|\eta\|_F^2 + h(x_k + \eta)$;
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-smooth Newton algorithm [XLWZ18];
- Step size 1 is not necessary decreasing;



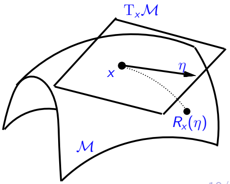
Proximal Gradient Method

Riemannian version in [CMSZ20]

ManPG [CMSZ20]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + h(x_k + \eta)$;
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-smooth Newton algorithm [XLWZ18];
- Step size 1 is not necessary decreasing;
- Convergence to a stationary point;



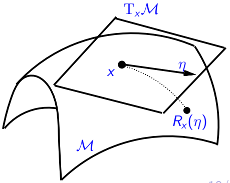
Proximal Gradient Method

Riemannian version in [CMSZ20]

ManPG [CMSZ20]

1. $\eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + h(x_k + \eta)$;
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-smooth Newton algorithm [XLWZ18];
- Step size 1 is not necessary decreasing;
- Convergence to a stationary point;
- No convergence rate analysis;



Proximal Gradient Method

Riemannian version in [HW21a]

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

Proximal Gradient Method

Riemannian version in [HW21a]

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

A Riemannian Proximal Gradient Method (RPG)

$$\text{Let } \ell_{x_k}(\eta) = \underbrace{\langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2}_{\text{Riemannian metric}} + h(\underbrace{R_{x_k}(\eta)}_{\text{replace } x_k + \eta});$$

1. $\eta_k \in \mathrm{T}_{x_k} \mathcal{M}$ is a stationary point of $\ell_{x_k}(\eta)$, and $\ell_{x_k}(0) \geq \ell_k(\eta_k)$;
2. $x_{k+1} = R_{x_k}(\eta_k)$;

- General framework for Riemannian optimization;

Proximal Gradient Method

Riemannian version in [HW21a]

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

A Riemannian Proximal Gradient Method (RPG)

Let $\ell_{x_k}(\eta) = \underbrace{\langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2}_{\text{Riemannian metric}} + h(\underbrace{R_{x_k}(\eta)}_{\text{replace } x_k + \eta});$

1. $\eta_k \in \mathrm{T}_{x_k} \mathcal{M}$ is a stationary point of $\ell_{x_k}(\eta)$, and $\ell_{x_k}(0) \geq \ell_k(\eta_k)$;
2. $x_{k+1} = R_{x_k}(\eta_k)$;

- General framework for Riemannian optimization;
- Step size can be fixed to be 1;

Proximal Gradient Method

Riemannian version in [HW21a]

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

A Riemannian Proximal Gradient Method (RPG)

$$\text{Let } \ell_{x_k}(\eta) = \underbrace{\langle \nabla f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2}_{\text{Riemannian metric}} + h(\underbrace{R_{x_k}(\eta)}_{\text{replace } x_k + \eta});$$

1. $\eta_k \in \mathrm{T}_{x_k} \mathcal{M}$ is a stationary point of $\ell_{x_k}(\eta)$, and $\ell_{x_k}(0) \geq \ell_k(\eta_k)$;
2. $x_{k+1} = R_{x_k}(\eta_k)$;

- General framework for Riemannian optimization;
- Step size can be fixed to be 1;
- Convergence rate results;

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. The function F is bounded from below and the sublevel set $\Omega_{x_0} = \{x \in \mathcal{M} \mid F(x) \leq F(x_0)\}$ is compact;

This assumption hold if, for example, F is continuous and $\mathcal{M}$ is compact.

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

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. The function F is bounded from below and the sublevel set $\Omega_{x_0} = \{x \in \mathcal{M} \mid F(x) \leq F(x_0)\}$ is compact;
2. The function f is L-retraction-smooth with respect to the retraction R in the sublevel set Ω_{x_0}.

Definition

A function $h : \mathcal{M} \to \mathbb{R}$ is called L-retraction-smooth 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 $R_x(\mathcal{S}_x) \subseteq \mathcal{N}$, we have that

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

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. The function F is bounded from below and the sublevel set $\Omega_{x_0} = \{x \in \mathcal{M} \mid F(x) \leq F(x_0)\}$ is compact;
2. The function f is L-retraction-smooth with respect to the retraction R in the sublevel set Ω_{x_0}.

If the following conditions hold, then f is L-retraction-smooth with respect to the retraction R in the manifold $\mathcal{M}$ [BAC18, Lemma 2.7]

- $\mathcal{M}$ is a compact Riemannian submanifold of a Euclidean space $\mathbb{R}^n$;
- the retraction R is globally defined;
- $f : \mathbb{R}^n \to \mathbb{R}$ is L-smooth in the convex hull of $\mathcal{M}$;

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

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. The function F is bounded from below and the sublevel set $\Omega_{x_0} = \{x \in \mathcal{M} \mid F(x) \leq F(x_0)\}$ is compact;
2. The function f is L-retraction-smooth with respect to the retraction R in the sublevel set Ω_{x_0}.

Theoretical results:

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

Proximal Gradient Method

Riemannian version in [HW21a]

Additional Assumptions:

- f and g are retraction-convex in $\Omega \supseteq \Omega_{x_0}$;

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

$$q_x(\eta) \geq q_x(\xi) + \langle \zeta, \eta - \xi \rangle_x \quad \forall \eta, \xi \in \mathcal{S}_x. \tag{1}$$

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

Proximal Gradient Method

Riemannian version in [HW21a]

Additional Assumptions:

- f and g are retraction-convex in $\Omega \supseteq \Omega_{x_0}$;

Lemma

Given $x \in \mathcal{M}$ and a twice continuously differentiable function $h : \mathcal{M} \to \mathbb{R}$, if one of the following conditions holds:

- $\mathrm{Hess}\, h$ *is positive definite at x, and the retraction is second order;*
- *The manifold $\mathcal{M}$ is an embedded submanifold of $\mathbb{R}^n$ endowed with the Euclidean metric; $\mathcal{W}$ is an open subset of $\mathbb{R}^n$; $x \in \mathcal{W}$; $h : \mathcal{W} \subset \mathbb{R}^n \to \mathbb{R}$ is a μ-strongly convex function in the Euclidean setting for a sufficient large μ; the retraction is second order;*

then there exists a neighborhood of x, denoted by $\mathcal{N}_x$, such that the function $h : \mathcal{M} \to \mathbb{R}$ is retraction-convex in $\mathcal{N}_x$.

Proximal Gradient Method

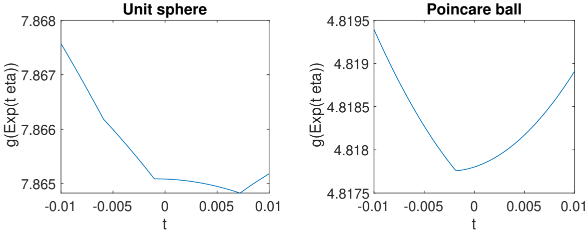
Riemannian version in [HW21a]

Additional Assumptions:

- f and g are retraction-convex in $\Omega \supseteq \Omega_{x_0}$;

Nonsmooth? Example: $h(x) = \|x\|_1$ with exponential mapping

- unit sphere: $\{x \in \mathbb{R}^n \mid x^T x = 1\}$, $n = 100$
- Poincaré ball model [GBH18]: $\{x \in \mathbb{R}^n \mid x^T x < 1\}$, $n = 100$
- $h(\mathrm{Exp}_x(t\eta_x))$ versus t



[GBH18] Ganea et al., Hyperbolic entailment cones for learning hierarchical embedding,

Proximal Gradient Method

Riemannian version in [HW21a]

Additional Assumptions:

- f and g are retraction-convex in $\Omega \supseteq \Omega_{x_0}$;
- Retraction approximately satisfies the triangle relation in Ω: for all $x, y, z \in \Omega$,

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

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

- In the Euclidean setting: $\eta_x = R_x^{-1}(y) = y - x$, $\xi_x = R_x^{-1}(z) = z - x$, $\zeta_y = R_y^{-1}(z) = z - y$:

$$\xi_x - \eta_x = (z - x) - (y - x) = z - y = \zeta_y.$$

- Holds for compact set $\overline{\Omega}$ with the exponential mapping;

Proximal Gradient Method

Riemannian version in [HW21a]

Additional Assumptions:

- f and g are retraction-convex in $\Omega \supseteq \Omega_{x_0}$;
- Retraction approximately satisfies the triangle relation in Ω: for all $x, y, z \in \Omega$,

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

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

Theoretical results:

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

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

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. Assumptions for the global convergence

---

1. The function F is bounded from below and the sublevel set $\Omega_{x_0} = \{x \in \mathcal{M} \mid F(x) \leq F(x_0)\}$ is compact;
2. The function f is L-retraction-smooth with respect to the retraction R in the sublevel set Ω_{x_0}.

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

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. Assumptions for the global convergence
2. f is locally Lipschitz continuously differentiable

Definition ([AMS08, 7.4.3])

A function f on $\mathcal{M}$ is Lipschitz continuously differentiable if it is differentiable and if there exists β_1 such that, for all x, y in $\mathcal{M}$ with $\mathrm{dist}(x, y) < i(\mathcal{M})$, it holds that

$$\|\mathcal{P}_\gamma^{0 \leftarrow 1} \operatorname{grad} f(y) - \operatorname{grad} f(x)\|_x \leq \beta_1 \operatorname{dist}(x, y),$$

where γ is the unique minimizing geodesic with $\gamma(0) = x$ and $\gamma(1) = y$.

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. Assumptions for the global convergence
2. f is locally Lipschitz continuously differentiable

If f is smooth and the manifold $\mathcal{M}$ is compact, then the function f is Lipschitz continuously differentiable. [AMS08, Proposition 7.4.5 and Corollary 7.4.6].

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

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. Assumptions for the global convergence
2. f is locally Lipschitz continuously differentiable
3. F satisfies the Riemannian KL property [BdCNO11]

Definition

A continuous function $f : \mathcal{M} \to \mathbb{R}$ is said to have the Riemannian KL property at $x \in \mathcal{M}$ if and only if there exists $\varepsilon \in (0, \infty]$, a neighborhood $U \subset \mathcal{M}$ of x, and a continuous concave function $\varsigma : [0, \varepsilon] \to [0, \infty)$ such that

- $\varsigma(0) = 0$, ς is C^1 on $(0, \varepsilon)$, and $\varsigma' > 0$ on $(0, \eta)$,
- For every $y \in U$ with $f(x) < f(y) < f(x) + \varepsilon$, we have

$$\varsigma'(f(y) - f(x)) \operatorname{dist}(0, \partial f(y)) \geq 1,$$

where $\operatorname{dist}(0, \partial f(y)) = \inf\{\|v\|_y : v \in \partial f(y)\}$ and ∂ denotes the Riemannian generalized subdifferential. The function ς is called the desingularising function.

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. Assumptions for the global convergence
2. f is locally Lipschitz continuously differentiable
3. F satisfies the Riemannian KL property [BdCNO11]

Theoretical results:

- it holds that

$$\sum_{k=0}^{\infty} \mathrm{dist}(x_k, x_{k+1}) < \infty.$$

Therefore, there exists only a unique accumulation point.

Proximal Gradient Method

Riemannian version in [HW21a]

Assumption:

1. Assumptions for the global convergence
2. f is locally Lipschitz continuously differentiable
3. F satisfies the Riemannian KL property [BdCNO11]

Theoretical results:

- If the desingularising function has the form $\varsigma(t) = \frac{C}{\theta} t^{\theta}$ for $C > 0$ and $\theta \in (0, 1]$ for all $x \in \Omega_{x_0}$, then
  - if $\theta = 1$, then the Riemannian proximal gradient method terminates in finite steps;
  - if $\theta \in [0.5, 1)$, then $\|x_k - x_*\| < C_1 d^k$ for $C_1 > 0$ and $d \in (0, 1)$;
  - if $\theta \in (0, 0.5)$, then $\|x_k - x_*\| < C_2 k^{\frac{-1}{1-2\theta}}$ for $C_2 > 0$;

Proximal Gradient Method

Numerical experiments

Sparse PCA problem

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

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

Proximal Gradient Method

Numerical experiments

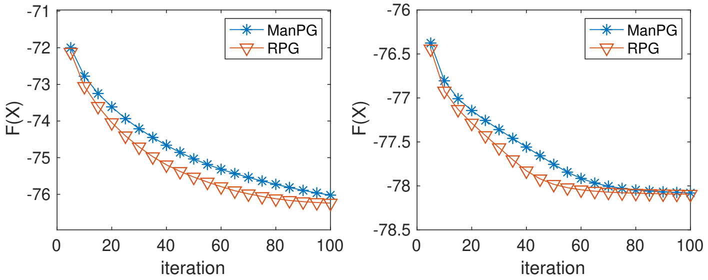


Figure: Two typical runs of ManPG, RPG, A-ManPG, and A-RPG for the Sparse PCA problem. $n = 1024$, $p = 4$, $\lambda = 2$, $m = 20$.

Summary of RPG

Generalizing the proximal mapping to manifolds is nontrivial

- Multiple Riemannian proximal mapping
- Theoretical results
- Numerical experiments

W. Huang and K. Wei, Riemannian proximal gradient methods, Mathematics Programming, 194, 371-413, 2022.

Summary of RPG

[BJJP25]: Given x_0,

$$\begin{cases} \text{Let } H_{x_k}(x) = h(x) + \frac{1}{2\lambda} d^2\left(x, R_{x_k}(-\lambda \mathrm{grad} f(x_k))\right); \\ x_{k+1} \text{ is a stationary point of } H_{x_k}(x); \\ \text{and } H_{x_k}(x_k) \geq H_{x_k}(x_{k+1}); \end{cases}$$

- x_{k+1} can be viewed as a Riemannian proximal point of h on manifold;
- Any limit point is a critical point by Exponential map;

[BJJP25] R. Bergmann, H. Jasa, P. John, M. Pfeffer. The intrinsic Riemannian proximal gradient method for nonconvex optimization. arXiv:2506.09775, 2025.

Content

Optimization with Structure:

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x).$$

- Proximal gradient methods
- Accelerated proximal gradient methods
  - Accelerated version of ManPG [HW21b];
  - Accelerated version of RPG [HW21a];
  - Accelerated version with theoretical guarantee [FJHY25];
- A proximal Newton method

[HW21a] W. Huang and K. Wei. An extension of fast iterative shrinkage-thresholding algorithm to Riemannian optimization for sparse principal component analysis. Numerical Linear Algebra with Applications, 29(1): e2409, 2022.
[HW21b] W. Huang and K. Wei. Riemannian proximal gradient methods. Mathematical Programming, 194(1-2):371-413,2022.
[FJHY25] S. Feng, Y. Jiang, W. Huang, and S. Ying. A Riemannian Accelerated Proximal Gradient Method, 2025.

Related Work

Euclidean Setting

A **proximal gradient** method, initial iterate x_0:

$$\begin{cases} d_k = \arg\min_p \langle \nabla f(x_k), p \rangle + \frac{L}{2}\|p\|_{\mathrm{F}}^2 + h(x_k + p) & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k & \text{(Update iterates)} \end{cases}$$

Related Work

Euclidean Setting

A **proximal gradient** method, initial iterate x_0:[1]

$$\begin{cases} d_k = \arg\min_p \langle \nabla f(x_k), p \rangle + \frac{L}{2}\|p\|_F^2 + h(x_k + p) & \text{(Proximal mapping)} \\ x_{k+1} = x_k + d_k & \text{(Update iterates)} \end{cases}$$

FISTA in convex [BT09]:

Given x_0, let $y_0 = x_0, t_0 = 1$;

$$\begin{cases} d_{y_k} = \mathrm{argmin}_p \langle \nabla f(y_k), p \rangle + \frac{L}{2}\|p\|_F^2 + h(y_k + p) \\ x_{k+1} = y_k + d_{y_k} \\ t_{k+1} = \frac{\sqrt{4t_k^2+1}+1}{2} \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k). \end{cases}$$

- Based on the Nesterov momentum technique;
- Two-point iterative sequence: x_k and y_k;
- $O\left(\frac{1}{k^2}\right)$ sublinear convergence rate for convex f and h;

Related Work

Euclidean Setting

FISTA in strongly convex [dST+21]:

$$\text{Given } x_0, \ \text{let } z_0 = x_0, A_0 = 0, q = \frac{\mu}{L} \ (\mu \geq 0);$$

$$\begin{cases} A_{k+1} = \frac{2A_k+1+\sqrt{4A_k+4qA_k^2+1}}{2(1-q)} \\ \tau_k = \frac{(A_{k+1}-A_k)(1+qA_k)}{A_{k+1}+2qA_kA_{k+1}-qA_k^2}, \quad \gamma_k = \frac{A_{k+1}-A_k}{1+qA_{k+1}} \\ y_k = x_k + \tau_k(z_k - x_k) \\ d_{y_k} = \mathrm{argmin}_p \langle \nabla f(y_k), p \rangle + \frac{L}{2}\|p\|_{\mathrm{F}}^2 + h(y_k + p) \\ x_{k+1} = y_k + d_{y_k} \\ z_{k+1} = (1 - q\gamma_k)z_k + q\gamma_k y_k + \gamma_k d_k. \end{cases}$$

- Three-point iterative sequence: x_k, y_k and z_k;
- $\min\{O\left(\frac{1}{k^2}\right), \ O\left(1-\sqrt{q}\right)^k\}$ convergence rate for strongly convex f and convex h;

[dST+21] A. d'Aspremont, D. Scieur and A. Taylor. Acceleration methods. Foundations and Trends in Optimization, 5(1-2): 1–245, 2021.

Related Work

Euclidean Setting

FISTA in strongly convex [dST+21]:

$$\text{Given } x_0, \text{ let } z_0 = x_0, A_0 = 0, q = \frac{\mu}{L} \ (\mu \geq 0);$$

$$\begin{cases} A_{k+1} = \frac{2A_k+1+\sqrt{4A_k+4qA_k^2+1}}{2(1-q)} \\ \tau_k = \frac{(A_{k+1}-A_k)(1+qA_k)}{A_{k+1}+2qA_kA_{k+1}-qA_k^2}, \quad \gamma_k = \frac{A_{k+1}-A_k}{1+qA_{k+1}} \\ y_k = x_k + \tau_k(z_k - x_k) \\ d_{y_k} = \mathrm{argmin}_p \langle \nabla f(y_k), p \rangle + \frac{L}{2}\|p\|_{\mathrm{F}}^2 + h(y_k + p) \\ x_{k+1} = y_k + d_{y_k} \\ z_{k+1} = (1 - q\gamma_k)z_k + q\gamma_k y_k + \gamma_k d_k. \end{cases}$$

- Three-point iterative sequence: x_k, y_k and z_k;
- $\min\{O\left(\frac{1}{k^2}\right), \ O\left(1-\sqrt{q}\right)^k\}$ convergence rate for strongly convex f and convex h;
- A unified accelerated method;

[dST+21] A. d'Aspremont, D. Scieur and A. Taylor. Acceleration methods. Foundations and Trends in Optimization, 5(1-2): 1–245, 2021.

Riemannian Version of FISTA

Euclidean version:

[BT09] convex: Given x_0, let $y_0 = x_0, t_0 = 1$;

$$\begin{cases} d_{y_k} = \mathrm{argmin}_p \langle \nabla f(y_k), p \rangle + \frac{L}{2} \|p\|_{\mathrm{F}}^2 + h(y_k + p) \\ x_{k+1} = y_k + d_{y_k} \\ t_{k+1} = \frac{\sqrt{4t_k^2+1}+1}{2} \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k). \end{cases}$$

- Riemannian version 1
- Riemannian version 2

Riemannian Version of FISTA

Euclidean version:

[BT09] convex: Given x_0, let $y_0 = x_0, t_0 = 1$;

$$\begin{cases} d_{y_k} = \mathrm{argmin}_p \langle \nabla f(y_k), p \rangle + \frac{L}{2}\|p\|_{\mathrm{F}}^2 + h(y_k + p) \\ x_{k+1} = y_k + d_{y_k} \\ t_{k+1} = \frac{\sqrt{4t_k^2+1}+1}{2} \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k). \end{cases}$$

- Riemannian version 1
- Riemannian version 2

[HW21b], AManPG: Given x_0, let $y_0 = x_0, t_0 = 1$;

$$\begin{cases} \eta_{y_k} = \arg\min_{\eta \in \mathrm{T}_{y_k}\mathcal{M}} \langle \nabla f(y_k), \eta \rangle + \frac{L}{2}\|\eta\|_F^2 + h(y_k + \eta) \\ x_{k+1} = R_{y_k}(\eta_{y_k}) \\ t_{k+1} = \frac{\sqrt{4t_k^2+1}+1}{2} \\ y_{k+1} = R_{x_{k+1}}\left(\frac{1-t_k}{t_{k+1}} R_{x_{k+1}}^{-1}(x_k)\right). \end{cases}$$

[HW22a] W. Huang and K. Wei. An extension of fast iterative shrinkage-thresholding algorithm to Riemannian optimization for sparse principal component analysis. Numerical Linear Algebra with Applications, 29(1): e2409, 2022.

Riemannian Version of FISTA

Euclidean version:

[BT09] convex: Given x_0, let $y_0 = x_0, t_0 = 1$;

$$\begin{cases} d_{y_k} = \mathrm{argmin}_p \langle \nabla f(y_k), p\rangle + \frac{L}{2}\|p\|_{\mathrm{F}}^2 + h(y_k + p) \\ x_{k+1} = y_k + d_{y_k} \\ t_{k+1} = \frac{\sqrt{4t_k^2+1}+1}{2} \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}}(x_{k+1} - x_k). \end{cases}$$

- Riemannian version 1
- Riemannian version 2

[HW21a]: Given x_0, let $y_0 = x_0, t_0 = 1$;

$$\begin{cases} \ell_{y_k}(\eta) = \langle \mathrm{grad} f(y_k), \eta\rangle_{y_k} + \frac{L}{2}\|\eta\|_{y_k}^2 + h(R_{y_k}(\eta)) \\ \eta_{y_k} \text{ is a stationary point of } \ell_{y_k} \text{ and } \ell_{y_k}(0) \geq \ell_{y_k}(\eta_{y_k}) \\ x_{k+1} = R_{y_k}(\eta_{y_k}) \\ t_{k+1} = \frac{1+\sqrt{4t_k^2+1}}{2} \\ y_{k+1} = R_{y_k}\left(\frac{t_{k+1}+t_k-1}{t_{k+1}}\eta_{y_k} - \frac{t_k-1}{t_{k+1}} R_{y_k}^{-1}(x_k)\right). \end{cases}$$

[HW22a] W. Huang and K. Wei. An extension of fast iterative shrinkage-thresholding algorithm to Riemannian optimization for sparse principal component analysis. Numerical Linear Algebra with Applications, 29(1): e2409, 2022.

[HW22b] W. Huang and K. Wei. Riemannian proximal gradient methods. Mathematical Programming,

Riemannian Version of FISTA

Euclidean version:

[BT09] convex: Given x_0, let $y_0 = x_0, t_0 = 1$;

$$\begin{cases} d_{y_k} = \mathrm{argmin}_p \langle \nabla f(y_k), p \rangle + \frac{L}{2} \|p\|_{\mathrm{F}}^2 + h(y_k + p) \\ x_{k+1} = y_k + d_{y_k} \\ t_{k+1} = \frac{\sqrt{4t_k^2+1}+1}{2} \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}} (x_{k+1} - x_k). \end{cases}$$

- Riemannian version 1
- Riemannian version 2

$$\frac{2}{\theta L}\left(t_k^2\left(F(x_{k+1}) - F(x_*)\right) - t_{k-1}^2\left(F(x_k) - F(x_*)\right)\right) \leqslant$$

$$\| \underbrace{(t_k - 1)R_{y_k}^{-1}(x_k) + R_{y_k}^{-1}(x_*)}_{\hat{W}_k} \|^2 - \| \underbrace{(t_k - 1)R_{y_k}^{-1}(x_k) + R_{y_k}^{-1}(x_*) - t_k \eta_{y_k}}_{\tilde{W}_{k+1}} \|^2$$

- $\hat{W}_k \neq \tilde{W}_k$ in general;
- How to control the difference?

[HW22a] W. Huang and K. Wei. An extension of fast iterative shrinkage-thresholding algorithm to Riemannian optimization for sparse principal component analysis. Numerical Linear Algebra with Applications, 29(1): e2409, 2022.

[HW22b] W. Huang and K. Wei. Riemannian proximal gradient methods. Mathematical Programming,

Riemannian Version of FISTA

Euclidean version:

[BT09] convex: Given x_0, let $y_0 = x_0, t_0 = 1$;

$$\begin{cases} d_{y_k} = \mathrm{argmin}_p \langle \nabla f(y_k), p \rangle + \frac{L}{2} \|p\|_{\mathrm{F}}^2 + h(y_k + p) \\ x_{k+1} = y_k + d_{y_k} \\ t_{k+1} = \frac{\sqrt{4t_k^2 + 1} + 1}{2} \\ y_{k+1} = x_{k+1} + \frac{t_k - 1}{t_{k+1}} (x_{k+1} - x_k). \end{cases}$$

- Riemannian version 1
- Riemannian version 2
  - Observe acceleration empirically;
  - No theoretical guarantee for acceleration;

[HW22a] W. Huang and K. Wei. An extension of fast iterative shrinkage-thresholding algorithm to Riemannian optimization for sparse principal component analysis. Numerical Linear Algebra with Applications, 29(1): e2409, 2022.
[HW22b] W. Huang and K. Wei. Riemannian proximal gradient methods. Mathematical Programming,

Related Work

Riemannian Setting

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

In smooth case: $h = 0$, **Riemannian Accelerated Gradient Methods**

- [LSC+17] [ZS18] [AS20] [JS22] [AOBL21] [MR22] [KY22] [MRP23]

Related Work

Riemannian Setting

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

In smooth case: $h = 0$, **Riemannian Accelerated Gradient Methods**

- [LSC+17] [ZS18] [AS20] [JS22] [AOBL21] [MR22] [KY22] [MRP23]

[KY22] Given x_0, let $z_0 = x_0$;

$$\begin{cases} y_k = \mathrm{Exp}_{x_k}\left(\tau_k \mathrm{Exp}_{x_k}^{-1}(z_k)\right) \\ x_{k+1} = \mathrm{Exp}_{y_k}\left(-\alpha_k \operatorname{grad} f(y_k)\right) \\ v_{y_k} = \beta_k \mathrm{Exp}_{y_k}^{-1}(z_k) - \gamma_k \operatorname{grad} f(y_k) \\ z_{k+1} = \mathrm{Exp}_{x_{k+1}}\left(\Gamma_{y_k}^{x_{k+1}}\left(v_{y_k} - \mathrm{Exp}_{y_k}^{-1}(x_{k+1})\right)\right). \end{cases}$$

[KY22] J. Kim and I. Yang. Accelerated gradient methods for geodesically convex optimization: tractable algorithms and convergence analysis. PMLR, 162: 11255–11282, 2022.

Related Work

Riemannian Setting

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

In smooth case: $h = 0$, **Riemannian Accelerated Gradient Methods**

- [LSC+17] [ZS18] [AS20] [JS22] [AOBL21] [MR22] [KY22] [MRP23]

  [KY22] Given x_0, let $z_0 = x_0$;

$$\begin{cases} y_k = \mathrm{Exp}_{x_k}\left(\tau_k \mathrm{Exp}_{x_k}^{-1}(z_k)\right) \\ x_{k+1} = \mathrm{Exp}_{y_k}\left(-\alpha_k \operatorname{grad} f(y_k)\right) \\ v_{y_k} = \beta_k \mathrm{Exp}_{y_k}^{-1}(z_k) - \gamma_k \operatorname{grad} f(y_k) \\ z_{k+1} = \mathrm{Exp}_{x_{k+1}}\left(\Gamma_{y_k}^{x_{k+1}}\left(v_{y_k} - \mathrm{Exp}_{y_k}^{-1}(x_{k+1})\right)\right). \end{cases}$$

- Accelerated convergence rates for geodesically convex f and geodesically strongly convex f, respectively;

[KY22] J. Kim and I. Yang. Accelerated gradient methods for geodesically convex optimization: tractable algorithms and convergence analysis. PMLR, 162: 11255–11282, 2022.

Related Work

Riemannian Setting

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

In smooth case: $h = 0$, **Riemannian Accelerated Gradient Methods**

- [LSC+17] [ZS18] [AS20] [JS22] [AOBL21] [MR22] [KY22] [MRP23]

  [KY22] Given x_0, let $z_0 = x_0$;

$$\begin{cases} y_k = \mathrm{Exp}_{x_k}\left(\tau_k \mathrm{Exp}_{x_k}^{-1}(z_k)\right) \\ x_{k+1} = \mathrm{Exp}_{y_k}\left(-\alpha_k \operatorname{grad} f(y_k)\right) \\ v_{y_k} = \beta_k \mathrm{Exp}_{y_k}^{-1}(z_k) - \gamma_k \operatorname{grad} f(y_k) \\ z_{k+1} = \mathrm{Exp}_{x_{k+1}}\left(\Gamma_{y_k}^{x_{k+1}}\left(v_{y_k} - \mathrm{Exp}_{y_k}^{-1}(x_{k+1})\right)\right). \end{cases}$$

- Accelerated convergence rates for geodesically convex f and geodesically strongly convex f, respectively;
- No unified parameters for accelerated gradient methods that works for both geodesically convex and geodesically strongly convex functions;

[KY22] J. Kim and I. Yang. Accelerated gradient methods for geodesically convex optimization: tractable algorithms and convergence analysis. PMLR, 162: 11255–11282, 2022.

The Proposed Approach

Riemannian accelerated proximal gradient method (RAPG)

- Riemannian proximal mapping [HW21a];
- Nesterov's acceleration;
- A three-point iterative method;

The Proposed Approach

Riemannian accelerated proximal gradient method (RAPG)

Initial iterate x_0, let $z_0 = x_0$;

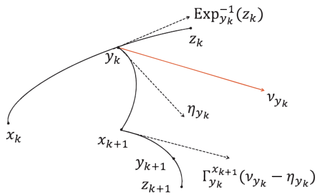
1. $y_k = \mathrm{Exp}_{x_k}\left(\tau_k \mathrm{Exp}_{x_k}^{-1}(z_k)\right)$;
2. η_{y_k} is a stationary point of $\ell_{y_k}(\eta)$ on $\mathrm{T}_{y_k}\mathcal{M}$ with $\ell_{y_k}(0) \geqslant \ell_{y_k}(\eta_{y_k})$, where $\ell_{y_k}(\eta) = \langle \mathrm{grad}\, f(y_k), \eta \rangle + \frac{\theta L}{2} \|\eta\|_{y_k}^2 + h\left(\mathrm{Exp}_{y_k}(\eta)\right)$;
3. $x_{k+1} = \mathrm{Exp}_{y_k}(\eta_{y_k})$;
4. $v_{y_k} = \beta_k \mathrm{Exp}_{y_k}^{-1}(z_k) + \gamma_k \eta_{y_k}, \quad z_{k+1} = \mathrm{Exp}_{x_{k+1}}\left(\Gamma_{y_k}^{x_{k+1}}(v_{y_k} - \eta_{y_k})\right)$;

The Proposed Approach

Riemannian accelerated proximal gradient method (RAPG)

Initial iterate x_0, let $z_0 = x_0$;

1. $y_k = \mathrm{Exp}_{x_k}\left(\tau_k \mathrm{Exp}_{x_k}^{-1}(z_k)\right)$;
2. η_{y_k} is a stationary point of $\ell_{y_k}(\eta)$ on $\mathrm{T}_{y_k}\mathcal{M}$ with $\ell_{y_k}(0) \geqslant \ell_{y_k}(\eta_{y_k})$, where $\ell_{y_k}(\eta) = \langle \mathrm{grad}\, f(y_k), \eta \rangle + \frac{\theta L}{2}\|\eta\|_{y_k}^2 + h\left(\mathrm{Exp}_{y_k}(\eta)\right)$;
3. $x_{k+1} = \mathrm{Exp}_{y_k}(\eta_{y_k})$;
4. $v_{y_k} = \beta_k \mathrm{Exp}_{y_k}^{-1}(z_k) + \gamma_k \eta_{y_k}, \quad z_{k+1} = \mathrm{Exp}_{x_{k+1}}\left(\Gamma_{y_k}^{x_{k+1}}(v_{y_k} - \eta_{y_k})\right)$;



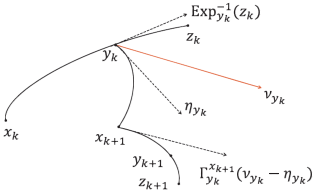
1. Step 1: compute y_k; note that x_k, y_k and z_k are on a geodesic;

The Proposed Approach

Riemannian accelerated proximal gradient method (RAPG)

Initial iterate x_0, let $z_0 = x_0$;

1. $y_k = \mathrm{Exp}_{x_k}\left(\tau_k \mathrm{Exp}_{x_k}^{-1}(z_k)\right)$;
2. η_{y_k} is a stationary point of $\ell_{y_k}(\eta)$ on $\mathrm{T}_{y_k}\mathcal{M}$ with $\ell_{y_k}(0) \geqslant \ell_{y_k}(\eta_{y_k})$, where $\ell_{y_k}(\eta) = \langle \mathrm{grad}\, f(y_k), \eta \rangle + \frac{\theta L}{2}\|\eta\|_{y_k}^2 + h\left(\mathrm{Exp}_{y_k}(\eta)\right)$;
3. $x_{k+1} = \mathrm{Exp}_{y_k}(\eta_{y_k})$;
4. $v_{y_k} = \beta_k \mathrm{Exp}_{y_k}^{-1}(z_k) + \gamma_k \eta_{y_k}, \quad z_{k+1} = \mathrm{Exp}_{x_{k+1}}\left(\Gamma_{y_k}^{x_{k+1}}(v_{y_k} - \eta_{y_k})\right)$;



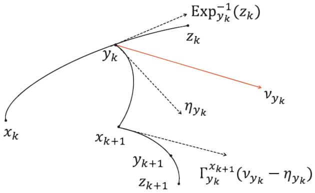
1. Step 1: compute y_k; note that x_k, y_k and z_k are on a geodesic;
2. Step 2: compute a Riemannian proximal gradient direction η_{y_k};

The Proposed Approach

Riemannian accelerated proximal gradient method (RAPG)

Initial iterate x_0, let $z_0 = x_0$;

1. $y_k = \mathrm{Exp}_{x_k}\left(\tau_k \mathrm{Exp}_{x_k}^{-1}(z_k)\right)$;
2. η_{y_k} is a stationary point of $\ell_{y_k}(\eta)$ on $\mathrm{T}_{y_k}\mathcal{M}$ with $\ell_{y_k}(0) \geqslant \ell_{y_k}(\eta_{y_k})$, where $\ell_{y_k}(\eta) = \langle \mathrm{grad}\, f(y_k), \eta \rangle + \frac{\theta L}{2}\|\eta\|_{y_k}^2 + h\left(\mathrm{Exp}_{y_k}(\eta)\right)$;
3. $x_{k+1} = \mathrm{Exp}_{y_k}(\eta_{y_k})$;
4. $v_{y_k} = \beta_k \mathrm{Exp}_{y_k}^{-1}(z_k) + \gamma_k \eta_{y_k}, \quad z_{k+1} = \mathrm{Exp}_{x_{k+1}}\left(\Gamma_{y_k}^{x_{k+1}}(v_{y_k} - \eta_{y_k})\right)$;



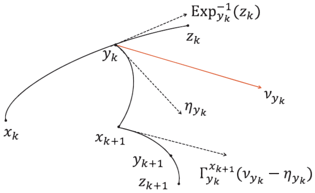
1. Step 1: compute y_k; note that x_k, y_k and z_k are on a geodesic;
2. Step 2: compute a Riemannian proximal gradient direction η_{y_k};
3. Step 3: update x_{k+1} by exponential map;

The Proposed Approach

Riemannian accelerated proximal gradient method (RAPG)

Initial iterate x_0, let $z_0 = x_0$;

1. $y_k = \mathrm{Exp}_{x_k}\left(\tau_k \mathrm{Exp}_{x_k}^{-1}(z_k)\right)$;
2. η_{y_k} is a stationary point of $\ell_{y_k}(\eta)$ on $\mathrm{T}_{y_k}\mathcal{M}$ with $\ell_{y_k}(0) \geqslant \ell_{y_k}(\eta_{y_k})$, where $\ell_{y_k}(\eta) = \langle \mathrm{grad}\, f(y_k), \eta \rangle + \frac{\theta L}{2}\|\eta\|_{y_k}^2 + h\left(\mathrm{Exp}_{y_k}(\eta)\right)$;
3. $x_{k+1} = \mathrm{Exp}_{y_k}(\eta_{y_k})$;
4. $v_{y_k} = \beta_k \mathrm{Exp}_{y_k}^{-1}(z_k) + \gamma_k \eta_{y_k}, \quad z_{k+1} = \mathrm{Exp}_{x_{k+1}}\left(\Gamma_{y_k}^{x_{k+1}}(v_{y_k} - \eta_{y_k})\right)$;



1. Step 1: compute y_k; note that x_k, y_k and z_k are on a geodesic;
2. Step 2: compute a Riemannian proximal gradient direction η_{y_k};
3. Step 3: update x_{k+1} by exponential map;
4. Step 4: update z_{k+1} by exponential map and parallel transport;

The Proposed Approach

Riemannian accelerated proximal gradient method (RAPG)

Initial iterate x_0, let $z_0 = x_0$;

1. $y_k = \mathrm{Exp}_{x_k}\left(\tau_k \mathrm{Exp}_{x_k}^{-1}(z_k)\right)$;
2. η_{y_k} is a stationary point of $\ell_{y_k}(\eta)$ on $\mathrm{T}_{y_k}\mathcal{M}$ with $\ell_{y_k}(0) \geqslant \ell_{y_k}(\eta_{y_k})$, where $\ell_{y_k}(\eta) = \langle \mathrm{grad}\, f(y_k), \eta \rangle + \frac{\theta L}{2}\|\eta\|_{y_k}^2 + h\left(\mathrm{Exp}_{y_k}(\eta)\right)$;
3. $x_{k+1} = \mathrm{Exp}_{y_k}(\eta_{y_k})$;
4. $v_{y_k} = \beta_k \mathrm{Exp}_{y_k}^{-1}(z_k) + \gamma_k \eta_{y_k}, \quad z_{k+1} = \mathrm{Exp}_{x_{k+1}}\left(\Gamma_{y_k}^{x_{k+1}}(v_{y_k} - \eta_{y_k})\right)$;

Next, we will show:

1. Assumptions on Manifolds and functions;
2. Parameter expressions for τ_k, β_k, γ_k;
3. Convergence rate of RAPG;

Assumptions on Manifolds and Functions

Assumption on Manifold:

1. Let Ω be a geodesically uniquely convex subset of $\mathcal{M}$. The diameter of Ω satisfies $\mathrm{diam}(\Omega) \leqslant D < \infty$;
2. The sectional curvature of Ω is bounded below by $\kappa_{\min}$ and above by $\kappa_{\max}$. If $\kappa_{\max} > 0$, it is additionally assumed that $D < \frac{\pi}{\sqrt{\kappa_{\max}}}$;

Assumptions on Manifolds and Functions

Assumption on Manifold:

1. Let Ω be a geodesically uniquely convex subset of $\mathcal{M}$. The diameter of Ω satisfies $\operatorname{diam}(\Omega) \leqslant D < \infty$;
2. The sectional curvature of Ω is bounded below by $\kappa_{\min}$ and above by $\kappa_{\max}$. If $\kappa_{\max} > 0$, it is additionally assumed that $D < \frac{\pi}{\sqrt{\kappa_{\max}}}$;

For the eigenvalues of the Hessian matrix of the squared distance function $\frac{1}{2}\mathrm{d}^2(\cdot, p)$ on $\Omega \subset \mathcal{M}$, where $p \in \Omega$:

- the upper bound:
$$\zeta = \begin{cases} \sqrt{-\kappa_{\min}} D \coth\left(\sqrt{-\kappa_{\min}} D\right), & \text{if } \kappa_{\min} < 0 \\ 1, & \text{if } \kappa_{\min} \geqslant 0 \end{cases}$$
- the lower bound:
$$\delta = \begin{cases} 1, & \text{if } \kappa_{\max} \leqslant 0 \\ \sqrt{\kappa_{\max}} D \cot\left(\sqrt{\kappa_{\max}} D\right), & \text{if } \kappa_{\max} > 0 \end{cases}$$

Assumptions on Manifolds and Functions

Assumption on Manifold:

1. Let Ω be a geodesically uniquely convex subset of $\mathcal{M}$. The diameter of Ω satisfies $\mathrm{diam}(\Omega) \leqslant D < \infty$;
2. The sectional curvature of Ω is bounded below by $\kappa_{\min}$ and above by $\kappa_{\max}$. If $\kappa_{\max} > 0$, it is additionally assumed that $D < \frac{\pi}{\sqrt{\kappa_{\max}}}$;

For the eigenvalues of the Hessian matrix of the squared distance function $\frac{1}{2}\mathrm{d}^2(\cdot, p)$ on $\Omega \subset \mathcal{M}$, where $p \in \Omega$:

- the upper bound:

$$\zeta = \begin{cases} \sqrt{-\kappa_{\min}} D \coth\left(\sqrt{-\kappa_{\min}} D\right), & \text{if } \kappa_{\min} < 0 \\ 1, & \text{if } \kappa_{\min} \geqslant 0 \end{cases}$$

- the lower bound:

$$\delta = \begin{cases} 1, & \text{if } \kappa_{\max} \leqslant 0 \\ \sqrt{\kappa_{\max}} D \cot\left(\sqrt{\kappa_{\max}} D\right), & \text{if } \kappa_{\max} > 0 \end{cases}$$

Choose $\xi \geq \zeta$.

Assumptions on Manifolds and Functions

Assumption on Manifold:

1. Let Ω be a geodesically uniquely convex subset of $\mathcal{M}$. The diameter of Ω satisfies $\mathrm{diam}(\Omega) \leqslant D < \infty$;
2. The sectional curvature of Ω is bounded below by κ_{min} and above by κ_{max}. If $\kappa_{\mathrm{max}} > 0$, it is additionally assumed that $D < \frac{\pi}{\sqrt{\kappa_{\mathrm{max}}}}$;

Assumption on functions:

1. The function f is geodesically L-smooth and geodesically μ-strongly convex ($\mu \geqslant 0$) in Ω;
2. The function h is ρ-retraction-convex with respect to the exponential map in Ω;

Assumptions on Manifold and Functions

ρ-retraction-convex:

$\tilde{h}_x(\eta) = h(R_x(\eta)) + \frac{\rho}{2}\|\eta\|^2$ is convex in tangent space.

- $\rho > 0$, h is said to be ρ-weakly retraction-convex with respect to R;
- $\rho = 0$, h is said to be retraction-convex with respect to R;
- $\rho < 0$, h is said to be ρ-strongly retraction-convex with respect to R.

Assumptions on Manifold and Functions

ρ-retraction-convex:

$\tilde{h}_x(\eta) = h(R_x(\eta)) + \frac{\rho}{2}\|\eta\|^2$ is convex in tangent space.

- $\rho > 0$, h is said to be ρ-weakly retraction-convex with respect to R;
- $\rho = 0$, h is said to be retraction-convex with respect to R;
- $\rho < 0$, h is said to be ρ-strongly retraction-convex with respect to R.

Weakly Retraction-Convex: A Necessary Assumption

e.g. $\|x\|_1$ is locally weakly retraction-convex on the embedded submanifold of $\mathbb{R}^n$.

Parameter Expressions for β_k, γ_k, τ_k

Under assumptions on manifold and functions:

$$A_{k+1} = \frac{\xi + 2\xi A_k + \sqrt{\xi^2 + 4\xi^2 A_k + 4\frac{\mu-\rho}{\theta L-\rho}\xi A_k^2}}{2\left(\xi - \frac{\mu-\rho}{\theta L-\rho}\right)},$$

$$\beta_k = \frac{\xi(\theta L - \rho) + (\mu - \rho) A_k}{\xi(\theta L - \rho) + (\mu - \rho) A_{k+1}}, \gamma_k = \frac{(\theta L - \rho)(A_{k+1} - A_k)}{\xi(\theta L - \rho) + (\mu - \rho) A_{k+1}}, \tau_k = \frac{\beta_k A_{k+1}}{\gamma_k A_k + \beta_k A_{k+1}};$$

Parameter Expressions for β_k, γ_k, τ_k

Under assumptions on manifold and functions:

$$A_{k+1} = \frac{\xi + 2\xi A_k + \sqrt{\xi^2 + 4\xi^2 A_k + 4\frac{\mu-\rho}{\theta L-\rho}\xi A_k^2}}{2\left(\xi - \frac{\mu-\rho}{\theta L-\rho}\right)},$$

$$\beta_k = \frac{\xi(\theta L-\rho) + (\mu-\rho)A_k}{\xi(\theta L-\rho) + (\mu-\rho)A_{k+1}}, \gamma_k = \frac{(\theta L-\rho)(A_{k+1}-A_k)}{\xi(\theta L-\rho) + (\mu-\rho)A_{k+1}}, \tau_k = \frac{\beta_k A_{k+1}}{\gamma_k A_k + \beta_k A_{k+1}};$$

Reduce to Euclidean space:

- if $\xi = 1$, $\rho = 0$, RAPG is FISTA in strongly convex [dST+21];
- otherwise, it is new as far as we known;

On manifold:

- Our parameter settings apply to both convex and strongly convex cases on manifold, leading to a unified accelerated algorithm.

Convergence Rate of RAPG

Under assumptions on manifold and functions:

- Sublinear convergence for $\mu \geqslant \rho$: $O\left(\frac{1}{k^2}\right)$;
- Linear convergence for $\mu > \rho$:

$$\min \left\{ \left(1 - \sqrt{\frac{\mu - \rho}{(\theta L - \rho)\xi}}\right)^k C_1, \ \frac{2}{\left(k + 2\sqrt{A_0}\right)^2} C_2 \right\}.$$

Assumption on functions:

1. The function f is geodesically L-smooth and geodesically μ-strongly convex ($\mu \geqslant 0$) in Ω;
2. The function h is ρ-retraction-convex with respect to the exponential map in Ω;

$$F(x) = f(x) + h(x)$$

Convergence Rate of RAPG

Sketch of the analysis

The core of our analysis is the construction of a potential function (or Lyapunov function) Φ_k that combines:

1. the function value gap;
2. the distance from the iterate to the optimal point; and
3. distortion error from curvature;

$$\Phi_k = A_k(F(x_k) - F(x_*)) + \frac{\xi(\theta L - \rho) + (\mu - \rho)A_k}{2}\left(\left\|\mathrm{Exp}_{x_k}^{-1}(z_k) - \mathrm{Exp}_{x_k}^{-1}(x_*)\right\|^2 + (\xi - 1)\left\|\mathrm{Exp}_{x_k}^{-1}(z_k)\right\|^2\right)$$

Convergence Rate of RAPG

Sketch of the analysis

The core of our analysis is the construction of a potential function (or Lyapunov function) Φ_k that combines:

1. the function value gap;
2. the distance from the iterate to the optimal point; and
3. distortion error from curvature;

$$\Phi_k = A_k(F(x_k) - F(x_*)) + \frac{\xi(\theta L - \rho) + (\mu - \rho)A_k}{2}\left(\left\|\mathrm{Exp}_{x_k}^{-1}(z_k) - \mathrm{Exp}_{x_k}^{-1}(x_*)\right\|^2 + (\xi - 1)\left\|\mathrm{Exp}_{x_k}^{-1}(z_k)\right\|^2\right)$$

A convergence rate of $O(1/A_k)$ is achieved if $\Phi_{k+1} \leqslant \Phi_k$ is satisfied.

The limit of RAPG:

- RAPG is theoretically supported only under the convexity of both f and h on manifolds;
- What happens in the nonconvex case?

We develop an improved version of the method.

Adaptive Restart for RAPG

Adaptive Restart for Riemannian Accelerated Proximal Gradient Method (AR-RAPG)

Set $z_0 = x_0$, $\tilde{x}_0 = x_0$, $\theta \geqslant 1$, $L = L_{\text{init}}$, $i = 0$, and $j = N_0$;
for $k = 0, 1, 2, \cdots$ **do**
  if $k == j$ **then**
    $[\tilde{x}_{i+1}, x_k, z_k, A_k, N_{i+1}, L] =$ Safeguard$(\tilde{x}_i, x_k, z_k, A_k, N_i, L)$;
    Set $j = j + N_{i+1}$ and $i = i + 1$;
  end if
  $(A_{k+1}, \beta_k, \gamma_k, \tau_k)$ are derived from the same formulas as in RAPG;
  Compute y_k, x_{k+1}, z_{k+1} as in RAPG;
end for

Adaptive Restart for RAPG

Adaptive Restart for Riemannian Accelerated Proximal Gradient Method (AR-RAPG)

1: Set $z_0 = x_0$, $\tilde{x}_0 = x_0$, $\theta \geqslant 1$, $L = L_{\mathrm{init}}$, $i = 0$, and $j = N_0$;
2: **for** $k = 0, 1, 2, \cdots$ **do**
3:     **if** $k == j$ **then**
4:         $[\tilde{x}_{i+1}, x_k, z_k, A_k, N_{i+1}, L] =$ Safeguard$(\tilde{x}_i, x_k, z_k, A_k, N_i, L)$;
5:         Set $j = j + N_{i+1}$ and $i = i + 1$;
6:     **end if**
7:     $(A_{k+1}, \beta_k, \gamma_k, \tau_k)$ are derived from the same formulas as in RAPG;
8:     Compute y_k, x_{k+1}, z_{k+1} as in RAPG;
9: **end for**

- Safeguard strategy from [HW21a];
- The functions f and h are not required to be convex on manifold;
- If the convexity of the functions is not known, we simply set $\mu = 0$ and $\rho = 0$;

[HW22b] W. Huang and K. Wei. Riemannian proximal gradient methods. Mathematical Programming, 194(1-2):371-413,2022.

Adaptive Restart for RAPG

Safeguard

Require: $(\tilde{x}_i, x_k, z_k, A_k, N_i, L)$;
Ensure: $[\tilde{x}_{i+1}, x_k, z_k, A_k, N_{i+1}, L]$;
1: $\eta_{\tilde{x}_i}$ is a stationary point of $\ell_{\tilde{x}_i}(\eta)$ on $\mathrm{T}_{x_i}\mathcal{M}$ with $\ell_{\tilde{x}_i}(0) \geqslant \ell_{\tilde{x}_i}(\eta_{\tilde{x}_i})$;
2: Set $\alpha_i = 1,\ i_{ls} = 0$;
3: **while** $F(\mathrm{Exp}_{\tilde{x}_i}(\alpha_i \eta_{\tilde{x}_i})) > F(\tilde{x}_i) - \sigma\alpha_i \|\eta_{\tilde{x}_i}\|^2$ and $i_{ls} < N_{ls}$ **do**
4: $\quad \alpha_i = \rho\alpha_i,\ i_{ls} = i_{ls} + 1$;
5: **end while**
6: **if** $i_{ls} == N_{ls}$ **then**
7: $\quad L = \tau L$ and go to Step 1; The estimation of L is too small
8: **end if**
9: **if** $F(\mathrm{Exp}_{\tilde{x}_i}(\alpha_i \eta_{\tilde{x}_i})) < F(x_k)$ **then**
10: $\quad$ Safeguard takes effect
11: $\quad$ **if** $N_i \neq N_{\max}$ **then**
12: $\quad\quad L = \tau L$;
13: $\quad$ **end if**
14: $\quad x_k = \mathrm{Exp}_{\tilde{x}_i}(\alpha_i \eta_{\tilde{x}_i}),\ z_k = x_k,\ A_k = A_0$; {Restart}
15: $\quad N_{i+1} = \max\{N_i - 1, N_{\min}\}$;
16: **else**
17: $\quad x_k$, z_k, and A_k keep unchanged; No restart
18: $\quad N_{i+1} = \min\{N_i + 1, N_{\max}\}$;
19: **end if**
20: $\tilde{x}_{i+1} = x_k$.

- Adaptively update the smoothness parameter L;
- Guarantee a decrease in the function value after a finite number of iterations;

Adaptive Restart for RAPG

Theorem (Convergence)

Under assumptions of Manifolds, if

1. Ω is compact;
2. all iterates remian in Ω;
3. f is smooth, h is locally Lipschitz continuous,

then any accumulation point $\tilde{x}_*$ of the sequence $\{\tilde{x}_i\}$ generated by AR-RAPG is a stationary point.

Numerical Experiments

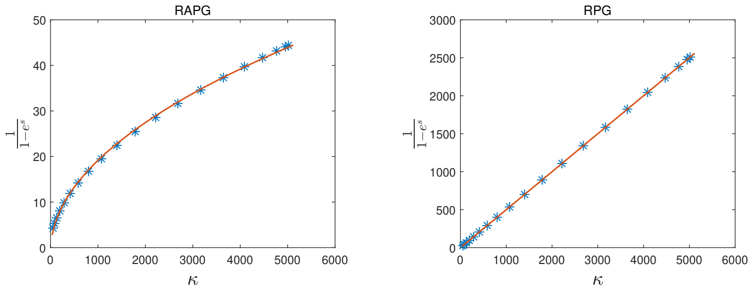
Convergence rate verification of RAPG and RPG

$$\min_{x \in \mathbb{S}^{n-1}} F(x) = \underbrace{-x^T A^T A x}_{f_1(x)} + \underbrace{\lambda \|x\|_1}_{h(x)},$$

- $A = USV^T + e$;
- $S \in \mathbb{R}^{m \times n}$: first m columns are $\mathrm{diag}(m + c, m, m - 1, \cdots, 2)$ with c varying from 0.01 to 1, and the remaining columns are zero;
- e is a small noise;

Numerical Experiments

Convergence rate verification of RAPG and RPG



(a) RAPG (b) RPG

Figure: Empirical relationship between κ and $\frac{1}{1-e^s}$ for RAPG and RPG. $m = 20, n = 1000, \lambda = 10^{-4}$.

Numerical Experiments

Effectiveness of the safeguard in AR-RAPG

$$\min_{X \in \mathrm{OB}(p,n)} F(X) = \underbrace{\|X^T A^T A X - D^2\|_F^2}_{f_2(X)} + \underbrace{\lambda \|X\|_1}_{h(X)}$$

- Oblique manifold: $\mathrm{OB}(p,n) = \{X \in \mathbb{R}^{n \times p} \mid x_i^T x_i = 1, \ i = 1, \cdots, p\}$;
- Entries of A: standard normal distribution $\mathcal{N}(0,1)$;
- Each column of A: zero mean and unit 2-norm;

Numerical Experiments

Effectiveness of the safeguard in AR-RAPG

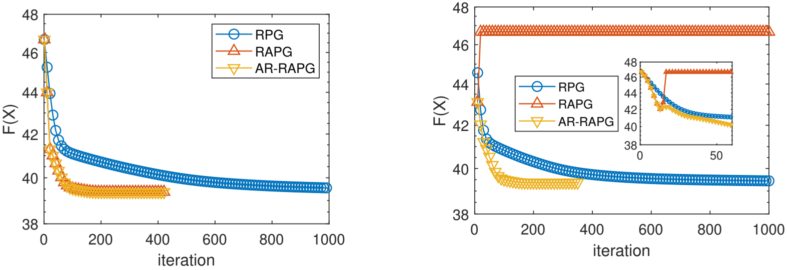


Figure: Comparison of RPG, RAPG, and AR-RAPG for the SPCA problem on oblique manifold. $\lambda = 1$, $m = 20$, $n = 200$, $p = 4$. Left: $L = 2\|D^2\|_F^2$; Right: $L = 1.2\|D^2\|_F^2$.

Numerical Experiments

Sparse PCA problem:

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

- $\mathrm{OB}(p, n) = \{X \in \mathbb{R}^{n \times p} \mid x_i^T x_i = 1,\ i = 1, \ldots, p\}$ denotes the oblique manifold;
- x_i being the i-th column of X;
- $A \in \mathbb{R}^{m \times n}$ is the data matrix and $p \leq m$;
- D is a diagonal matrix with the dominant singular values of A on the diagonal;

Compared with:

- **ManPG, ManPG-Ada**: in [CMSZ20];
- **RPG:** in [HW21a];

Numerical Experiments

Left: the norm of search direction;
Right: function value.

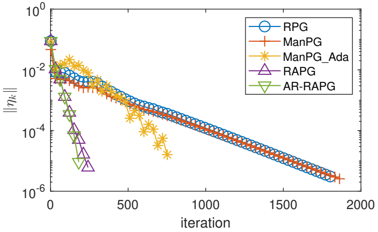


Figure: SPCA problem on oblique manifold. $n = 200$, $m = 20$, $p = 4$.

Numerical Experiments

For $m = 20, \ p = 4, \ n = \{32, 64, 128, 256\}$.
Left: number of iterations;
Right: CPU time.

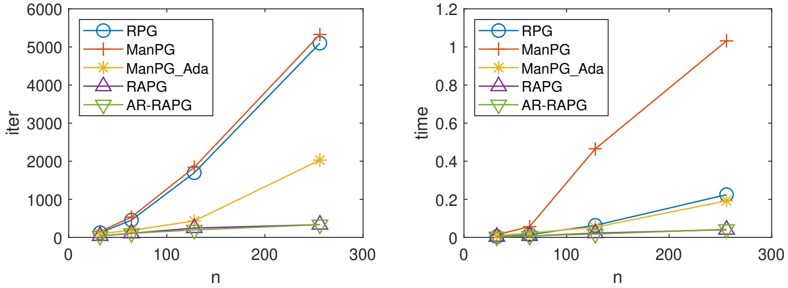


Figure: SPCA problem on oblique manifold.

Numerical Experiments

For $m = 20$, $n = 128$, $p = \{1, 2, 3, 4\}$.
Left: number of iterations;
Right: CPU time.

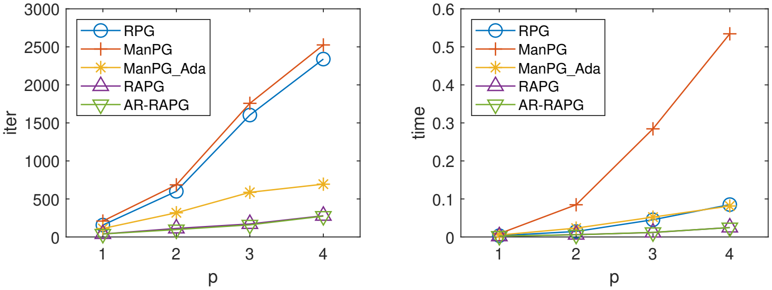


Figure: SPCA problem on oblique manifold.

Content

Optimization with Structure:

$$\min_{x \in \mathcal{M}} F(x) = f(x) + h(x).$$

- Proximal gradient methods
- Inexact proximal gradient methods
- A proximal Newton method
    - Related proximal Newton methods
    - A Riemannian proximal Newton method

Related Proximal Newton Methods

Euclidean version

Given x_0;

$$\begin{cases} d_k = \operatorname{argmin}_p \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + t_k d_k, \text{ for a step size } t_k \end{cases}$$

Related Proximal Newton Methods

Euclidean version

Given x_0;

$$\begin{cases} d_k = \mathrm{argmin}_p \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + t_k d_k, \text{ for a step size } t_k \end{cases}$$

- H_k is Hessian or a positive definite approximation to Hessian [LSS14, MYZZ22];

[LLS14] Jason D Lee, Yuekai Sun, and Michael A Saunders. Proximal newton-type methods for minimizing composite functions. SIAM Journal on Optimization, 24(3):1420-1443, 2014.

[MYZZ22] Boris S Mordukhovich, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A globally convergent proximal newton-type method in nonsmooth convex optimization. Mathematical Programming, pages 1-38, 2022.

Related Proximal Newton Methods

Euclidean version

Given x_0;

$$\begin{cases} d_k = \mathrm{argmin}_p \langle \nabla f(x_k), p \rangle + \frac{1}{2}\langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + t_k d_k, \text{ for a step size } t_k \end{cases}$$

- H_k is Hessian or a positive definite approximation to Hessian [LSS14, MYZZ22];
- t_k is one for sufficiently large k;

[LLS14] Jason D Lee, Yuekai Sun, and Michael A Saunders. Proximal newton-type methods for minimizing composite functions. SIAM Journal on Optimization, 24(3):1420-1443, 2014.
[MYZZ22] Boris S Mordukhovich, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A globally convergent proximal newton-type method in nonsmooth convex optimization. Mathematical Programming, pages 1-38, 2022.

Related Proximal Newton Methods

Euclidean version

Given x_0;

$$\begin{cases} d_k = \mathrm{argmin}_p \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + t_k d_k, \text{ for a step size } t_k \end{cases}$$

- H_k is Hessian or a positive definite approximation to Hessian [LSS14, MYZZ22];
- t_k is one for sufficiently large k;
- Quadratic/Superlinear convergence rate for strongly convex f and convex h;

[LLS14] Jason D Lee, Yuekai Sun, and Michael A Saunders. Proximal newton-type methods for minimizing composite functions. SIAM Journal on Optimization, 24(3):1420-1443, 2014.
[MYZZ22] Boris S Mordukhovich, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A globally convergent proximal newton-type method in nonsmooth convex optimization. Mathematical Programming, pages 1-38, 2022.

Related Proximal Newton Methods

Riemannian version: a naive generalization

Focus on embedded submanifolds

Euclidean version:

$$\begin{cases} d_k = \mathrm{argmin}_p \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, \nabla^2 f(x_k) p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad}\, f(x_k), \eta \rangle + \frac{1}{2} \langle \eta, \mathrm{Hess}\, f(x_k) \eta \rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

Related Proximal Newton Methods

Riemannian version: a naive generalization

Focus on embedded submanifolds

Euclidean version:

$$\begin{cases} d_k = \mathrm{argmin}_p \langle \nabla f(x_k), p\rangle + \frac{1}{2}\langle p, \nabla^2 f(x_k) p\rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \mathrm{grad}\, f(x_k), \eta\rangle + \frac{1}{2}\langle \eta, \mathrm{Hess}\, f(x_k)\eta\rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

Does it converge superlinearly locally?

Related Proximal Newton Methods

Riemannian version: a naive generalization

Focus on embedded submanifolds

Euclidean version:

$$\begin{cases} d_k = \operatorname{argmin}_p \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, \nabla^2 f(x_k) p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad}\, f(x_k), \eta \rangle + \frac{1}{2} \langle \eta, \mathrm{Hess}\, f(x_k) \eta \rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

Does it converge superlinearly locally?

Not necessarily!

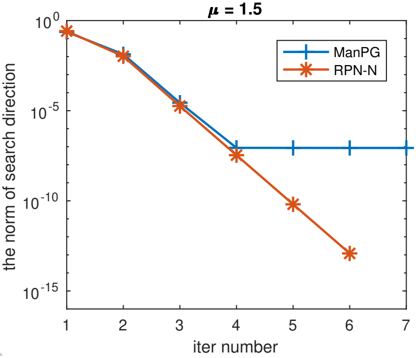
Related Proximal Newton Methods

Riemannian version: a naive generalization

Consider the Sparse PCA over sphere:

$$\min_{x \in \mathbb{S}^{n-1}} -x^{\mathrm{T}} A^{\mathrm{T}} A x + \mu \|x\|_1,$$

where $f(x) = -x^{\mathrm{T}} A^{\mathrm{T}} A x$, $h(x) = \mu \|x\|_1$.



Related Proximal Newton Methods

Riemannian version: a naive generalization

Euclidean version:

$$\begin{cases} d_k = \mathrm{argmin}_p \langle \nabla f(x_k), p \rangle + \frac{1}{2}\langle p, \nabla^2 f(x_k) p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad}\, f(x_k), \eta \rangle + \frac{1}{2}\langle \eta, \mathrm{Hess}\, f(x_k) \eta \rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

- $x_k + \eta$ in h is only a first order approximation;

Related Proximal Newton Methods

Riemannian version: a naive generalization

Euclidean version:

$$\begin{cases} d_k = \mathrm{argmin}_p \langle \nabla f(x_k), p\rangle + \frac{1}{2}\langle p, \nabla^2 f(x_k) p\rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k \end{cases}$$

A native generalization by replacing the Euclidean gradient and Hessian by the Riemannian gradient and Hessian:

$$\begin{cases} \eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \mathrm{grad}\, f(x_k), \eta\rangle + \frac{1}{2}\langle \eta, \mathrm{Hess}\, f(x_k)\eta\rangle + h(x_k + \eta) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

$$\begin{cases} \eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \mathrm{grad}\, f(x_k), \eta\rangle + \frac{1}{2}\langle \eta, \mathrm{Hess}\, f(x_k)\eta\rangle + h(x_k + \eta + \frac{1}{2}\Pi(\eta, \eta)) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

- $x_k + \eta$ in h is only a first order approximation;
- If an second order approximation is used, then the subproblem is difficult to solve;

A Riemannian Proximal Newton Method

Riemannian version

A Riemannian proximal Newton method (RPN)

1. Compute
   $v(x_k) = \operatorname{argmin}_{v \in \mathrm{T}_{x_k}\mathcal{M}} \ f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$
2. Find $u(x_k) \in \mathrm{T}_{x_k}\mathcal{M}$ by solving
   $$J(x_k)[u(x_k)] = -v(x_k),$$
   where $J(x_k) = -\left[\mathrm{I}_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})\right]$, Λ_{x_k} and $\mathcal{L}_{x_k}$ are defined later ;
3. $x_{k+1} = R_{x_k}(u(x_k));$

A Riemannian Proximal Newton Method

Riemannian version

A Riemannian proximal Newton method (RPN)

1. Compute
$v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
$$J(x_k)[u(x_k)] = -v(x_k),$$
where $J(x_k) = -\left[\mathrm{I}_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})\right]$, Λ_{x_k} and $\mathcal{L}_{x_k}$ are defined later ;
3. $x_{k+1} = R_{x_k}(u(x_k));$

1. Step 1: compute a Riemannian proximal gradient direction (ManPG)

A Riemannian Proximal Newton Method

Riemannian version

A Riemannian proximal Newton method (RPN)

1. Compute
   $v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v)$;
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
   $$J(x_k)[u(x_k)] = -v(x_k),$$
   where $J(x_k) = -\left[\mathrm{I}_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})\right]$, Λ_{x_k} and $\mathcal{L}_{x_k}$ are defined later ;
3. $x_{k+1} = R_{x_k}(u(x_k))$;

1. Step 1: compute a Riemannian proximal gradient direction (ManPG)
2. Step 2: compute the Riemannian proximal Newton direction, where $J(x_k)$ is from a generalized Jacobi of $v(x_k)$;

A Riemannian Proximal Newton Method

Riemannian version

A Riemannian proximal Newton method (RPN)

1. Compute
$v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v)$;
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
$$J(x_k)[u(x_k)] = -v(x_k),$$
where $J(x_k) = -\left[\mathrm{I}_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k}) \right]$, Λ_{x_k} and $\mathcal{L}_{x_k}$ are defined later ;
3. $x_{k+1} = R_{x_k}(u(x_k))$;

1. Step 1: compute a Riemannian proximal gradient direction (ManPG)
2. Step 2: compute the Riemannian proximal Newton direction, where $J(x_k)$ is from a generalized Jacobi of $v(x_k)$;
3. Step 3: Update iterate by a retraction;

A Riemannian Proximal Newton Method

Riemannian version

A Riemannian proximal Newton method (RPN)

1. Compute
$$v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$$
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
$$J(x_k)[u(x_k)] = -v(x_k),$$
where $J(x_k) = -\left[\mathrm{I}_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})\right]$, Λ_{x_k} and $\mathcal{L}_{x_k}$ are defined later ;
3. $x_{k+1} = R_{x_k}(u(x_k));$

Next, we will show:

1. G-semismoothness of $v(x_k)$ and its generalized Jacobi;
2. Superlinear convergence rate;

A Riemannian Proximal Newton Method

Riemannian version

Definition (G-Semismoothness [Gow04])

Let $F : \mathcal{D} \to \mathbb{R}^m$ where $\mathcal{D} \subset \mathbb{R}^n$ be an open set, $\mathcal{K} : \mathcal{D} \rightrightarrows \mathbb{R}^{m \times n}$ be a nonempty set-valued mapping. We say that F is G-semismooth at $x \in \mathcal{D}$ with respect to $\mathcal{K}$ if for any $J \in \mathcal{K}(x + d)$,

$$F(x + d) - F(x) - Jd = o(\|d\|) \text{ as } d \to 0.$$

If F is G-semismooth at any $x \in \mathcal{D}$ with respect to $\mathcal{K}$, then F is called a G-semismooth function with respect to $\mathcal{K}$.

The standard definition of semismoothness additional requires:

- $\mathcal{K}$ is compact valued, upper semicontinuous set-valued mapping;
- F is a locally Lipschitz continuous function;
- F is directionally differentiable at x;

[Gow04] M Seetharama Gowda. Inverse and implicit function theorems for h-differentiable and semismooth functions. Optimization Methods and Software, 19(5):443-461, 2004.

A Riemannian Proximal Newton Method

Riemannian version

$v(x)$ (dropping the subscript for simplicity)

$$v(x) = \operatorname*{argmin}_{v \in \mathrm{T}_x \mathcal{M}} \; f(x) + \langle \nabla f(x), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x+v);$$

A Riemannian Proximal Newton Method

Riemannian version

$v(x)$ (dropping the subscript for simplicity)

$$v(x) = \operatorname*{argmin}_{v \in \mathrm{T}_x \mathcal{M}} \; f(x) + \langle \nabla f(x), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x+v);$$

Above problem can be rewritten as

$$\arg \min_{B_x^T v = 0} \langle \xi_x, v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x+v)$$

where $B_x^T v = (\langle b_1, v \rangle, \langle b_2, v \rangle, \ldots, \langle b_m, v \rangle)^T$, and $\{b_1, \ldots, b_m\}$ forms an orthonormal basis of $\mathrm{T}_x^{\perp} \mathcal{M}$.

A Riemannian Proximal Newton Method

Riemannian version

The Lagrangian function:

$$\mathcal{L}(v, \lambda) = \langle \xi_x, v \rangle + \frac{1}{2t} \langle v, v \rangle + h(X + v) - \langle \lambda, B_x^T v \rangle.$$

Therefore

$$\text{KKT: } \begin{cases} \partial_v \mathcal{L}(v, \lambda) = 0 \\ B_x^T v = 0 \end{cases} \Longrightarrow \begin{cases} v = \mathrm{Prox}_{th} \left(x - t(\xi_x - B_x \lambda) \right) - x \\ B_x^T v = 0 \end{cases}$$

where $\mathrm{Prox}_{tg}(z) = \mathrm{argmin}_{v \in \mathbb{R}^{n \times p}} \frac{1}{2} \|v - z\|_F^2 + th(v)$.

Define

$$\mathcal{F} : \mathbb{R}^n \times \mathbb{R}^{n+d} \mapsto \mathbb{R}^{n+d} : (x; v, \lambda) \mapsto \begin{pmatrix} v + x - \mathrm{Prox}_{th}(x - t[\nabla f(x) + B_x \lambda]) \\ B_x^T v \end{pmatrix}.$$

$v(x)$ is the solution of the system $\mathcal{F}(x, v(x), \lambda(x)) = 0$;

A Riemannian Proximal Newton Method

Riemannian version

Define

$$\mathcal{F} : \mathbb{R}^n \times \mathbb{R}^{n+d} \mapsto \mathbb{R}^{n+d} : (x; v, \lambda) \mapsto \begin{pmatrix} v + x - \mathrm{Prox}_{th}(x - t[\nabla f(x) + B_x \lambda]) \\ B_x^T v \end{pmatrix}.$$

- $\mathcal{F}$ is semismooth;
- $v(x)$ is G-semismooth by the G-semismooth Implicit Function Theorem in [Gow04, PSS03];

[Gow04] M Seetharama Gowda. Inverse and implicit function theorems for h-differentiable and semismooth functions. Optimization Methods and Software, 19(5):443-461, 2004.

[PSS03] Jong-Shi Pang, Defeng Sun, and Jie Sun. Semismo oth homeomorphisms and strong stability of semidefinite and Lorentz complementarity problems. Mathematics of Operations Research, 28(1):39-63, 2003.

A Riemannian Proximal Newton Method

Riemannian version

Lemma (Semismooth Implicit Function Theorem)

Suppose that $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ is a semismooth function with respect to $\partial_{\mathrm{B}} F$ in an open neighborhood of (x^0, y^0) with $F(x^0, y^0) = 0$. Let $H(y) = F(x^0, y)$, if every matrix in $\partial_C H(y^0)$ is nonsingular, then there exists an open set $\mathcal{V} \subset \mathbb{R}^n$ containing x^0, a set-valued fucntion $\mathcal{K} : \mathcal{V} \to \mathbb{R}^{m\times n}$, and a G-semismooth function $f : \mathcal{V} \to \mathbb{R}^m$ with respect to $\mathcal{K}$ satisfying $f(x^0) = y^0$, for every $x \in \mathcal{V}$,

$$F(x, f(x)) = 0,$$

and the set-valued function $\mathcal{K}$ is

$$\mathcal{K} : x \mapsto \{-(A_y)^{-1} A_x : [A_x \ A_y] \in \partial_{\mathrm{B}} F\big(x, f(x)\big)\},$$

where the map $x \mapsto \mathcal{K}(x)$ is compact valued and upper semicontinuous.

Not new but an arrangement of existing results.

A Riemannian Proximal Newton Method

Riemannian version

Without loss of generality, we assume that the nonzero entries of x_* are in the first part, i.e., $x_* = [\bar{x}_*^T, 0^T]^T$

Assumption

Let $B_{x_*}^{\mathrm{T}} = [\bar{B}_{x_*}^{\mathrm{T}}, \hat{B}_{x_*}^{\mathrm{T}}]$, where $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$ and $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$. It is assumed that $j \geq d$ and $\bar{B}_{x_*}$ is full column rank.

A Riemannian Proximal Newton Method

Riemannian version

Without loss of generality, we assume that the nonzero entries of x_* are in the first part, i.e., $x_* = [\bar{x}_*^T, 0^T]^T$

Assumption

Let $B_{x_*}^{\mathrm{T}} = [\bar{B}_{x_*}^{\mathrm{T}}, \hat{B}_{x_*}^{\mathrm{T}}]$, where $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$ and $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$. It is assumed that $j \geq d$ and $\bar{B}_{x_*}$ is full column rank.

$v(x)$ is a G-semismooth function of x in a neighborhood of x_*

Under the above Assumption, there exists a neighborhood $\mathcal{U}$ of x_* such that $v : \mathcal{U} \to \mathbb{R}^n : x \mapsto v(x)$ is a G-semismooth function with respect to $\mathcal{K}_v$, where

$$\mathcal{K}_v : x \mapsto \left\{ -[\mathrm{I}_n,\ 0] B^{-1} A : [A\ B] \in \partial_{\mathrm{B}} \mathcal{F}\big(x, v(x), \lambda(x)\big) \right\}.$$

For $x \in \mathcal{U}$, any element of $\mathcal{K}_v(x)$ is called a generalized Jacobi of v at x.

Here, the semismooth implicit function theorem is used

A Riemannian Proximal Newton Method

Riemannian version

The generalized Jacobi of v at x is

$$\Big\{\mathcal{J}_x \,|\mathcal{J}_x[\omega] = -\left[\mathrm{I}_n - \Lambda_x + t\Lambda_x(\nabla^2 f(x) - \mathcal{L}_x)\right]\omega - M_x B_x H_x(\mathrm{D}B_x^{\mathrm{T}}[\omega])v, \forall \omega$$

$$M_x \in \partial_C \mathrm{prox}_{th}(x)\Big\},$$

where $\Lambda_x = M_x - M_x B_x H_x B_x^T M_k$, $H_x = \left(B_x^T M_x B_x\right)^{-1}$, $\mathcal{L}_x(\cdot) = \mathcal{W}_x(\cdot, B_x \lambda(x))$, and $\mathcal{W}_x$ denotes the Weingarten map;

- $v(x_*) = 0$;
- Set $J(x) = \mathrm{I}_n - \Lambda_x + t\Lambda_x(\nabla^2 f(x) - \mathcal{L}_x)$;
- The Riemannian proximal Newton direction: $J(x)u(x) = -v(x)$;
- Let $u(x) = (\bar{u}(x); \hat{u}(x))$, then

$$\hat{u}(x) = \hat{v} \text{ and } \bar{J}(x)\bar{u}(x) = -\bar{v}(x)$$

A Riemannian Proximal Newton Method

Riemannian version

Assumption:

1. Let $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$, where $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$ and and $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$. It is assumed that $j \geq d$ and $\bar{B}_{x_*}$ is full column rank;

A Riemannian Proximal Newton Method

Riemannian version

Assumption:

1. Let $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$, where $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$ and and $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$. It is assumed that $j \geq d$ and $\bar{B}_{x_*}$ is full column rank;
2. There exists a neighborhood $\mathcal{U}$ of $x_* = [\bar{x}_*^T, 0^T]^T$ on $\mathcal{M}$ such that for any $x = [\bar{x}^T, \tilde{x}^T]^T \in \mathcal{U}$, it holds that $\bar{x} + \bar{v} \neq 0$ and $\hat{x} + \hat{v} = 0$.

$$v(x) = \operatorname*{argmin}_{v \in \mathrm{T}_x \mathcal{M}} f(x) + \langle \nabla f(x), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x + v)$$

A Riemannian Proximal Newton Method

Riemannian version

Assumption:

1. Let $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$, where $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$ and and $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$. It is assumed that $j \geq d$ and $\bar{B}_{x_*}$ is full column rank;
2. There exists a neighborhood $\mathcal{U}$ of $x_* = [\bar{x}_*^T, 0^T]^T$ on $\mathcal{M}$ such that for any $x = [\bar{x}^T, \tilde{x}^T]^T \in \mathcal{U}$, it holds that $\bar{x} + \bar{v} \neq 0$ and $\hat{x} + \hat{v} = 0$.

Theorem

Suppose that x_* be a local optimal minimizer. Under the above Assumptions, assume that $J(x_*)$ is nonsingular. Then there exists a neighborhood $\mathcal{U}$ of x_* on $\mathcal{M}$ such that for any $x_0 \in \mathcal{U}$, RPN Algorithm generates the sequence $\{x_k\}$ converging quadratically to x_*.

A Riemannian Proximal Newton Method

Riemannian version

Assumption:

1. Let $B_{x_*}^T = [\bar{B}_{x_*}^T, \hat{B}_{x_*}^T]$, where $\bar{B}_{x_*} \in \mathbb{R}^{j \times d}$ and and $\hat{B}_{x_*} \in \mathbb{R}^{(n-j) \times d}$. It is assumed that $j \geq d$ and $\bar{B}_{x_*}$ is full column rank;
2. There exists a neighborhood $\mathcal{U}$ of $x_* = [\bar{x}_*^T, 0^T]^T$ on $\mathcal{M}$ such that for any $x = [\bar{x}^T, \tilde{x}^T]^T \in \mathcal{U}$, it holds that $\bar{x} + \bar{v} \neq 0$ and $\hat{x} + \hat{v} = 0$.

Theorem

Suppose that x_* be a local optimal minimizer. Under the above Assumptions, assume that $J(x_*)$ is nonsingular. Then there exists a neighborhood $\mathcal{U}$ of x_* on $\mathcal{M}$ such that for any $x_0 \in \mathcal{U}$, RPN Algorithm generates the sequence $\{x_k\}$ converging quadratically to x_*.

If the intersection of manifold and sparsity constraints forms an embedded manifold around x_*, then $\nabla^2 \bar{f}(x_*) - \bar{\mathcal{L}} \succeq 0$. If $\nabla^2 \bar{f}(x_*) - \bar{\mathcal{L}} \succ 0$, then $J(x_*)$ is nonsingular.

A Riemannian Proximal Newton Method

Riemannian version

$$\text{Smooth case:} \quad \min_{x \in \mathcal{M}} f(x)$$

- KKT conditions:

$$\nabla f(x) + \frac{1}{t} v + B_x \lambda = 0, \text{ and } B_x^T v = 0;$$

- Closed form solutions:

$$\lambda(x) = -B_x^{\mathrm{T}} \nabla f(x), \qquad v = -t \operatorname{grad} f(x);$$

- Action of $J(x)$: for $\omega \in \mathrm{T}_x \mathcal{M}$

$$J(x)[\omega] = -tP_{\mathrm{T}_x \mathcal{M}}(\nabla^2 f(x) - \mathcal{L}_x) P_{\mathrm{T}_x \mathcal{M}} \omega = -t \operatorname{Hess} f(x)[\omega]$$

- $J(x)u(x) = -v(x) \Longrightarrow \operatorname{Hess} f(x)[u(x)] = -\operatorname{grad} f(x)$;
- It is the Riemannian Newton method;

A Riemannian Proximal Newton Method

Numerical Experiments

Sparse PCA problem

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

where $A \in \mathbb{R}^{m \times n}$ is a data matrix and $\mathrm{St}(r, n) = \{X \in \mathbb{R}^{n \times r} \mid X^T X = I_r\}$ is the compact Stiefel manifold.

- $R_x(\eta_x) = (x + \eta_x)(I + \eta_x^T \eta_x)^{-1/2}$;
- $t = 1/(2\|A\|_2^2)$;
- Run ManPG until $\|v\|$ reaches 10^{-4}, i.e., it reduces by a factor of 10^3. The resulting x as the input of RPN;

A Riemannian Proximal Newton Method

Numerical Experiments

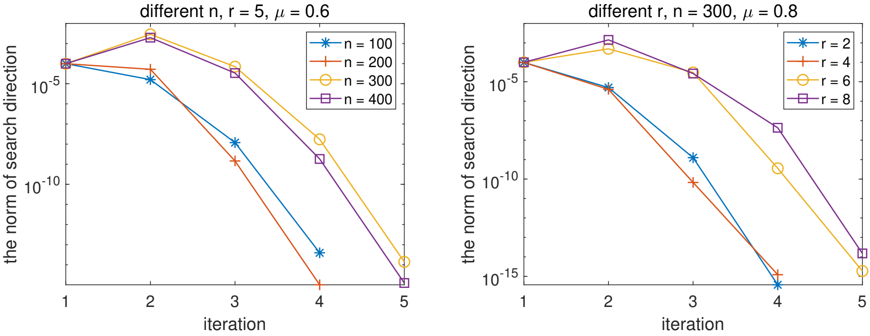


Figure: Random data. Left: different $n = \{100, 200, 300, 400\}$ with $r = 5$ and $\mu = 0.6$; Right: different $r = \{2, 4, 6, 8\}$ with $n = 300$ and $\mu = 0.8$

Summary

- Review Euclidean proximal Newton methods;
- Riemannian proximal Newton method;
- Convergence analysis;
- Numerical experiments;

W. Si, P.-A. Absil, W. Huang, R. Jiang, S. Vary, A Riemannian Proximal Newton Method, SIAM Journal on Optimization, 34:1, pp. 654-681, 2024.

Thank you

Thank you!

References I

P.-A. Absil, R. Mahony, and R. Sepulchre.
Optimization algorithms on matrix manifolds.
Princeton University Press, Princeton, NJ, 2008.

Foivos Alimisis, Antonio Orvieto, Gary Becigneul, and Aurelien Lucchi.
Momentum improves optimization on Riemannian manifolds.
In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1351–1359, 2021.

Kwangjun Ahn and Suvrit Sra.
From Nesterov's estimate sequence to Riemannian acceleration.
In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 84–118, 2020.

Nicolas Boumal, P-A Absil, and Coralia Cartis.
Global rates of convergence for nonconvex optimization on manifolds.
IMA Journal of Numerical Analysis, 39(1):1–33, 02 2018.

G. C. Bento, J. X. de Cruz Neto, and P. R. Oliveira.
Convergence of inexact descent methods for nonconvex optimization on Riemannian manifold.
arXiv preprint arXiv:1103.4828, 2011.

Matthias Bollh ofer, Aryan Eftekhari, Simon Scheidegger, and Olaf Schenk.
Large-scale sparse inverse covariance matrix estimation.
SIAM Journal on Scientific Computing, 41(1):A380–A401, 2019.

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.

References II

Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang.
Proximal gradient method for nonsmooth optimization over the Stiefel manifold.
SIAM Journal on Optimization, 30(1):210–239, 2020.

Haoran Chen, Yanfeng Sun, Junbin Gao, Yongli Hu, and Baocai Yin.
Fast optimization algorithm on riemannian manifolds and its application in low-rank learning.
Neurocomputing, 291:59 – 70, 2018.

Alexandre d'Aspremont, Damien Scieur, Adrien Taylor, et al.
Acceleration methods.
Foundations and Trends® in Optimization, 5(1-2):1–245, 2021.

Octavian Eugen Ganea, Gary Becigneul, and Thomas Hofmann.
Hyperbolic entailment cones for learning hierarchical embeddings.
35th International Conference on Machine Learning, ICML 2018, 4:2661–2673, 2018.

M Seetharama Gowda.
Inverse and implicit function theorems for h-differentiable and semismooth functions.
Optimization Methods and Software, 19(5):443–461, 2004.

W. Huang and K. Wei.
Riemannian proximal gradient methods.
Mathematical Programming, 2021.
published online, DOI:10.1007/s10107-021-01632-3.

Wen Huang and Ke Wei.
An extension of fast iterative shrinkage-thresholding algorithm to Riemannian optimization for sparse principal component analysis.
Numerical Linear Algebra with Applications, page e2409, 2021.

References III

Wen Huang, Meng Wei, Kyle A. Gallivan, and Paul Van Dooren.
A Riemannian Optimization Approach to Clustering Problems, 2022.

Jikai Jin and Suvrit Sra.
Understanding Riemannian acceleration via a proximal extragradient framework.
In *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 2924–2962, 2022.

Jungbin Kim and Insoon Yang.
Accelerated gradient methods for geodesically convex optimization: tractable algorithms and convergence analysis.
In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 11255–11282, 2022.

Yuan Yuan Liu, Fan Hua Shang, James Cheng, Hong Cheng, and Licheng Jiao.
Accelerated first-order methods for geodesically convex optimization on Riemannian manifolds.
In *Advances in Neural Information Processing Systems*, pages 4868–4877, 2017.

Jason D Lee, Yuekai Sun, and Michael A Saunders.
Proximal newton-type methods for minimizing composite functions.
SIAM Journal on Optimization, 24(3):1420–1443, 2014.

David Martínez-Rubio.
Global Riemannian acceleration in hyperbolic and spherical spaces.
In *Proceedings of the 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 768–826, 2022.

David Martínez-Rubio and Sebastian Pokutta.
Accelerated Riemannian optimization: Handling constraints with a prox to bound geometric penalties.
In *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 359–393, 2023.

References IV

Boris S Mordukhovich, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang.
A globally convergent proximal newton-type method in nonsmooth convex optimization.
Mathematical Programming, pages 1–38, 2022.

Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, and Stanley Osher.
Compressed modes for variational problems in mathematics and physics.
Proceedings of the National Academy of Sciences, 110(46):18368–18373, 2013.

Jong-Shi Pang, Defeng Sun, and Jie Sun.
Semismooth homeomorphisms and strong stability of semidefinite and lorentz complementarity problems.
Mathematics of Operations Research, 28(1):39–63, 2003.

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.

Hui Zou, Trevor Hastie, and Robert Tibshirani.
Sparse principal component analysis.
Journal of Computational and Graphical Statistics, 15(2):265–286, 2006.

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.

Hongyi Zhang and Suvrit Sra.
An estimate sequence for geodesically convex optimization.
In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1703–1723, 2018.