A Riemannian Proximal Newton-CG Method

Speaker: Wen Huang

Xiamen University

November 23, 2024

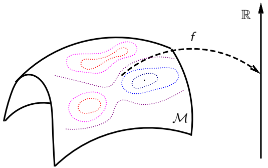
Joint work with Wutao Si in University Grenoble Alpes

2024年广西数学优化前沿研讨会

Problem Statement

Optimization on Manifolds with Structure:

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

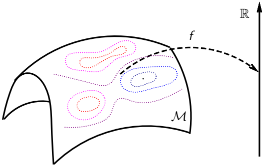


- $\mathcal{M}$ is a finite-dimensional Riemannian manifold;
- f is smooth and may be nonconvex; and
- $h(x)$ is continuous and convex but may be nonsmooth;

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 modes [OLCO13], sparse partial least squares regression [CSG+18], sparse inverse covariance estimation [BESS19], sparse blind deconvolution [ZLK+17], and clustering [HWGVD22].

Outline

- Proximal gradient method and its variants;
- A Riemannian proximal Newton method;
- A Riemannian proximal Newton-CG method;
- Numerical experiments;

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

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

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

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

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

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

Given x_0[1],

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

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

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

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

Given x_0,

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

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

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

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

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

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

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

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

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

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

- A representative one: FISTA [BT09];
- Based on the Nesterov momentum technique;
- $O\left(\frac{1}{k^2}\right)$ sublinear convergence rate for convex f and h;

[BT09] 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.

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

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

Given x_0;

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

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

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

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];
- t_k is one for sufficiently large k;
- Quadratic/Superlinear convergence rate for strongly convex f and convex h;
- Josephy-Newton algorithm[Jos79];

[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.

[Jos79] N. Josephy, Newton's method for generalized equations. Technical Summary Report 1965, Mathematics Research Center, University of Wisconsin, Madison, Wisconsin (1979)

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

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

Given x_0, H_0;

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

[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.

[ST16] K. Scheinberg and X. Tang. Practical inexact proximal quasi-Newton method with global complexity analysis. Mathematical Programming, (160):495-529, 2016.

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

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

Given x_0, H_0;

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

- Dennis-Moré condition $\Longrightarrow$ superlinear convergence rate for strongly convex f and convex h [LSS14];
- Sublinear without the accuracy assumption on H_k [ST16];

[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.

[ST16] K. Scheinberg and X. Tang. Practical inexact proximal quasi-Newton method with global complexity analysis. Mathematical Programming, (160):495-529, 2016.

Proximal Gradient Method and its variants

Euclidean to Riemannian

Optimization with Structure:

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

Riemannian versions

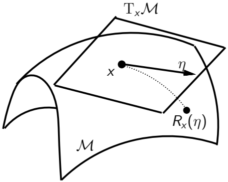
Optimization with Structure:

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

[CMSZ20], ManPG: Given x_0,

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton



[CMSZ20] S. Chen, S. Ma, A. Man-Cho So, and T. Zhang. Proximal gradient method for nonsmooth optimization over the Stiefel manifold. SIAM Journal on Optimization, 30(1):210-239, 2020.

Optimization with Structure:

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

[CMSZ20], ManPG: Given x_0,

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

[HW21a], RPG: Given x_0,

$$\begin{cases} \text{Let } \ell_{x_k}(\eta) = \langle \mathrm{grad} f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + h(R_{x_k}(\eta)); \\ \eta_k \text{ is a stationary point of } \ell_{x_k} \text{ and } \ell_{x_k}(0) \geq \ell_k(\eta_k); \\ x_{k+1} = R_{x_k}(\eta_k); \end{cases}$$

[CMSZ20] S. Chen, S. Ma, A. Man-Cho So, and T. Zhang. Proximal gradient method for nonsmooth optimization over the Stiefel manifold. SIAM Journal on Optimization, 30(1):210-239, 2020.
[HW21a] W. Huang and K. Wei. Riemannian proximal gradient methods. Mathematical Programming, 194, p.371-413, 2022.

Optimization with Structure:

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

[CMSZ20], ManPG: Given x_0,

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

[HW21a], RPG: Given x_0,

$$\begin{cases} \text{Let } \ell_{x_k}(\eta) = \langle \mathrm{grad} f(x_k), \eta \rangle_{x_k} + \frac{L}{2} \|\eta\|_{x_k}^2 + h(R_{x_k}(\eta)); \\ \eta_k \text{ is a stationary point of } \ell_{x_k} \text{ and } \ell_{x_k}(0) \geq \ell_k(\eta_k); \\ x_{k+1} = R_{x_k}(\eta_k); \end{cases}$$

- [CMSZ20]: numerical aspect;
- [HW21a]: theoretical aspect;

[CMSZ20] S. Chen, S. Ma, A. Man-Cho So, and T. Zhang. Proximal gradient method for nonsmooth optimization over the Stiefel manifold. SIAM Journal on Optimization, 30(1):210-239, 2020.
[HW21a] W. Huang and K. Wei. Riemannian proximal gradient methods. Mathematical Programming, 194, p.371-413, 2022.

Optimization with Structure:

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

[HW21b], AManPG: Given x_0, set $y_0 = x_0$

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

[HW21b] 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, p.e2409, 2021.

Optimization with Structure:

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

[HW21b], AManPG: Given x_0, set $y_0 = x_0$

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

- A representative on in [HW21b], also see [HW21a];
- Observe acceleration empirically;
- No theoretical guarantee for acceleration;

[HW21b] 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, p.e2409, 2021.

Optimization with Structure:

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

[WY23, WY24], ManRQN, ARPQN, ARPN: Given x_0

$$\begin{cases} \eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \\ \qquad \frac{1}{2}\langle \eta, \mathcal{H}_k \eta \rangle + h(x_k + \eta) \quad \left(\text{or } h(R_{x_k}(\eta))\right) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

[WY23] Q. Wang and W. Yang. Proximal Quasi-Newton Method for Composite Optimization over the Stiefel Manifold, 95:39, 2023.
[WY24] Q. Wang and W. Yang. An adaptive regularized proximal Newton-type methods for composite optimization over the Stiefel manifold, Computational Optimization and Applications, 2024

Optimization with Structure:

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

[WY23, WY24], ManRQN, ARPQN, ARPN: Given x_0

$$\begin{cases} \eta_k = \arg\min_{\eta\in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta\rangle + \\ \qquad\qquad \frac{1}{2}\langle \eta, \mathcal{H}_k \eta\rangle + h(x_k+\eta) \quad \left(\text{or } h(R_{x_k}(\eta))\right) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

- $\mathcal{H}_k$: an approximation of quasi-Newton update or Riemannian Hessian;
- Local superlinear convergence results: $h(R_{x_k}(\eta))$;
- Only use diagonal $\mathcal{H}_k$ and $h(x_k+\eta)$ numerically.

[WY23] Q. Wang and W. Yang. Proximal Quasi-Newton Method for Composite Optimization over the Stiefel Manifold, 95:39, 2023.
[WY24] Q. Wang and W. Yang. An adaptive regularized proximal Newton-type methods for composite optimization over the Stiefel manifold, Computational Optimization and Applications, 2024

Optimization with Structure:

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

- Proximal Gradient
- Accelerated versions
- Proximal inexact Newton
- Proximal quasi-Newton

[WY23, WY24], ManRQN, ARPQN, ARPN: Given x_0

$$\begin{cases} \eta_k = \arg\min_{\eta \in \mathrm{T}_{x_k}\mathcal{M}} \langle \nabla f(x_k), \eta \rangle + \\ \qquad\qquad \frac{1}{2}\langle \eta, \mathcal{H}_k \eta \rangle + h(x_k + \eta) \quad \left(\text{or } h(R_{x_k}(\eta))\right) \\ x_{k+1} = R_{x_k}(\eta_k) \end{cases}$$

- $\mathcal{H}_k$: an approximation of quasi-Newton update or Riemannian Hessian;
- Local superlinear convergence results: $h(R_{x_k}(\eta))$;
- Only use diagonal $\mathcal{H}_k$ and $h(x_k + \eta)$ numerically.

Good theoretical results

but not practical algorithms with a local superlinear convergence rate

Outline

- Proximal gradient method and its variants;
- A Riemannian proximal Newton method;
- A Riemannian proximal Newton-CG method;
- Numerical experiments;

A practical algorithm with a local superlinear convergence rate

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

Outline

- Proximal gradient method and its variants;
- A Riemannian proximal Newton method;
- A Riemannian proximal Newton-CG method;
- Numerical experiments;

Note that this method focuses on:

- $\mathcal{M}$ is an Riemannian embedded submanifold of a Euclidean space;
- $h(x) = \mu \|x\|_1$;

A Riemannian proximal Newton method

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

A Riemannian proximal Newton method (RPN)

1. Compute the ManPG direction
   $v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v)$;
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
   $$J(x_k)[u(x_k)] = -v(x_k),$$
   where $J(x_k) = -\left[I_n - \Lambda_{x_k} + t \Lambda_{x_k} (\nabla^2 f(x_k) - \mathcal{L}_{x_k}) \right]$, 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

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

A Riemannian proximal Newton method (RPN)

1. Compute the ManPG direction
$v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v)$;
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
$$J(x_k)[u(x_k)] = -v(x_k),$$
where $J(x_k) = -\left[I_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})\right]$, 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

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

A Riemannian proximal Newton method (RPN)

1. Compute the ManPG direction
   $v(x_k) = \mathrm{argmin}_{v\in \mathrm{T}_{x_k}\mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v\rangle + \frac{1}{2t}\|v\|_F^2 + h(x_k + v)$;
2. Find $u(x_k) \in \mathrm{T}_{x_k}\mathcal{M}$ by solving
   $$J(x_k)[u(x_k)] = -v(x_k),$$
   where $J(x_k) = -\left[I_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})\right]$, 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

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

A Riemannian proximal Newton method (RPN)

1. Compute the ManPG direction
   $v(x_k) = \operatorname{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v)$;
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
   $$J(x_k)[u(x_k)] = -v(x_k),$$
   where $J(x_k) = -\left[I_n - \Lambda_{x_k} + t\Lambda_{x_k}(\nabla^2 f(x_k) - \mathcal{L}_{x_k})\right]$, 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;

Without loss of generality, we assume that the nonzero entries of x_* are in the first part, i.e., $x_* = [\bar{x}_*^T, 0^T]^T$. B_x denotes an orthonormal basis of $\mathrm{T}_x^{\perp} \mathcal{M}$ at x.

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;

Without loss of generality, we assume that the nonzero entries of x_* are in the first part, i.e., $x_* = [\bar{x}_*^T, 0^T]^T$. B_x denotes an orthonormal basis of $\mathrm{T}_x^\perp \mathcal{M}$ at x.

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 $x = [\bar{x}^T, \hat{x}^T]^T \in \mathcal{U}$, it holds that $\bar{x} + \bar{v} \neq 0$ and $\hat{x} + \hat{v} = 0$.

A Riemannian proximal Newton method

Local superlinear convergence rate

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 superlinearly to x_*.

The convergence rate is improved to quadratically convergence in [SAH+24a]

A Riemannian proximal Newton method

- Similar to the Riemannian Newton method, this Riemannian proximal Newton method does not guarantee global convergence;

A Riemannian proximal Newton method

- Similar to the Riemannian Newton method, this Riemannian proximal Newton method does not guarantee global convergence;
- A hybrid method that merges ManPG with RPN is proposed in [SAH+24b];

Input: $x_0 \in \mathcal{M}$, $t > 0$, $\epsilon > 0$;

for $k = 0, 1, \ldots$ **do**
    Compute a ManPG direction v_k;
    If $\|v_k\| \leq \epsilon$, then $K = k$ and break;
    $x_{k+1} = R_{x_k}(\alpha v_k)$ with an appropriate step size;
end for
for k = K+1, K+2, ... **do**
    Compute u_k by solving $J(x_k) u_k = -v_k$ with v_k being the ManPG direction;
    $x_{k+1} = R_{x_k}(u_k)$;
end for

A Riemannian proximal Newton method

- Similar to the Riemannian Newton method, this Riemannian proximal Newton method does not guarantee global convergence;
- A hybrid method that merges ManPG with RPN is proposed in [SAH+24b];

Input: $x_0 \in \mathcal{M}$, $t > 0$, $\epsilon > 0$;

1: **for** $k = 0, 1, \ldots$ **do**
2: Compute a ManPG direction v_k;
3: If $\|v_k\| \leq \epsilon$, then $K = k$ and break;
4: $x_{k+1} = R_{x_k}(\alpha v_k)$ with an appropriate step size;
5: **end for**
6: **for** k = K+1, K+2, ... **do**
7: Compute u_k by solving $J(x_k) u_k = -v_k$ with v_k being the ManPG direction;
8: $x_{k+1} = R_{x_k}(u_k)$;
9: **end for**

The switching parameter ϵ is crucial for the performance.

Outline

- Proximal gradient method and its variants;
- A Riemannian proximal Newton method;
- A Riemannian proximal Newton-CG method;
- Numerical experiments;

A practical and robust algorithm with
global convergence and local superlinear convergence guarantee

W. Huang, and W. Si. A Riemannian Proximal Newton-CG Method, arxiv:2405.08365, 2024.

Outline

- Proximal gradient method and its variants;
- A Riemannian proximal Newton method;
- A Riemannian proximal Newton-CG method;
- Numerical experiments;

Also focus on:

- $\mathcal{M}$ is an Riemannian embedded submanifold of a Euclidean space;
- $h(x) = \mu \|x\|_1$;

A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method (RPN)

1. Compute the ManPG direction
   $v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
   $$J(x_k)[u(x_k)] = -v(x_k);$$
3. $x_{k+1} = R_{x_k}(u(x_k));$

Smooth case:

- $v(x_k) = -t \,\mathrm{grad}\, f(x_k);$
- $J(x_k) = -t \,\mathrm{Hess}\, f(x_k);$
- $J(x_k)[u(x_k)] = -v(x_k) \Longrightarrow \underbrace{\mathrm{Hess}\, f(x_k)[u(x_k)] = -\,\mathrm{grad}\, f(x_k)}_{\text{truncated conjugate gradient (tCG)}}.$

A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method (RPN)

1. Compute the ManPG direction
$v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
$$J(x_k)[u(x_k)] = -v(x_k);$$
3. $x_{k+1} = R_{x_k}(u(x_k));$

Smooth case:

- $v(x_k) = -t \, \mathrm{grad} \, f(x_k);$
- $J(x_k) = -t \, \mathrm{Hess} \, f(x_k);$
- $J(x_k)[u(x_k)] = -v(x_k) \Longrightarrow \underbrace{\mathrm{Hess} \, f(x_k)[u(x_k)] = -\mathrm{grad} \, f(x_k)}_{\text{truncated conjugate gradient (tCG)}}.$

Nonsmooth case:

- $v(x_k)$: ManPG direction;
- $J(x_k)$: Generalized Jacobi of v;
- $u(x_k)$: solving a linear system by $\underbrace{J(x_k)[u(x_k)] = -v(x_k)}_{tCG?}$

A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method (RPN)

1. Compute the ManPG direction
   $v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v);$
2. Find $u(x_k) \in \mathrm{T}_{x_k} \mathcal{M}$ by solving
   $$J(x_k)[u(x_k)] = -v(x_k);$$
3. $x_{k+1} = R_{x_k}(u(x_k));$

Smooth case:

- $v(x_k) = -t \,\mathrm{grad}\, f(x_k);$
- $J(x_k) = -t \,\mathrm{Hess}\, f(x_k);$
- $J(x_k)[u(x_k)] = -v(x_k) \Longrightarrow$
  $\underbrace{\mathrm{Hess}\, f(x_k)[u(x_k)] = -\,\mathrm{grad}\, f(x_k)}_{\text{truncated conjugate gradient (tCG)}}.$

Nonsmooth case:

- $v(x_k)$: ManPG direction;
- $J(x_k)$: Generalized Jacobi of v;
- $u(x_k)$: solving a linear system by $\underbrace{J(x_k)[u(x_k)] = -v(x_k)}_{tCG?}$

Problem: $J(x_k)$ is not symmetric!

A Riemannian proximal Newton-CG method

Notation:

$$\mathfrak{B}_{x_k} = \nabla^2 f(x_k) - \mathcal{L}_{x_k} = \begin{pmatrix} \mathfrak{B}_{x_k}^{(11)} & \mathfrak{B}_{x_k}^{(12)} \\ \mathfrak{B}_{x_k}^{(21)} & \mathfrak{B}_{x_k}^{(22)} \end{pmatrix}, \mathcal{B}_{x_k} = \mathfrak{B}_{x_k}^{(11)}.$$

$$J(x_k) = -\begin{pmatrix} \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger} + t(I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger})\mathcal{B}_{x_k} & t(I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger})\mathfrak{B}_{x_k}^{(12)} \\ 0_{(n-j_k)\times j_k} & I_{n-j_k} \end{pmatrix}$$

$$\begin{cases} [\bar{B}_{x_k}\bar{B}_{x_k}^{\dagger} + t(I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger})\mathcal{B}_{x_k}]\bar{u}(x_k) = \bar{v}(x_k) - t(I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger})\mathfrak{B}_{x_k}^{(12)}\hat{u}(x_k) \\ \hat{u}(x_k) = \hat{v}(x_k) \end{cases}.$$

$$\Longrightarrow \bar{u}(x_k) = \bar{v}(x_k) - \{I_{j_k} + (I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger})N_{x_k}\}^{-1}(I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger})\ell_{x_k}$$

where $\ell_{x_k} = \frac{1}{t_k}(-I_{j_k} + t_k\mathcal{B}_{x_k})\bar{v}(x_k) + \mathfrak{B}_{x_k}^{(12)}\hat{v}(x_k)$ and $N_{x_k} = -I_{j_k} + t\mathcal{B}_{x_k}$ is symmetric.

A Riemannian proximal Newton-CG method

$$\bar{u}(x_k) = \bar{v}(x_k) - \{I_{j_k} + (I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger}) \underbrace{N_{x_k}}_{symmetric} \}^{-1}(I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger})\ell_{x_k}$$

Lemma

Let $N \in \mathbb{R}^{j\times j}$ and $B \in \mathbb{R}^{j\times m}$ with $m \leq j$. Suppose that $I_j + N$ is symmetric positive definite on $\{w \mid B^T w = 0\}$ and that B is full column rank. Then it holds that the unique solution of the problem

$$\min_{B^T w = 0} \ell^T w + \frac{1}{2} w^T (I_j + N) w$$

is given by

$$w_* = -\left[I_j + (I_j - BB^{\dagger})N\right]^{-1}\left[I_j - BB^{\dagger}\right]\ell.$$

A Riemannian proximal Newton-CG method

$$\bar{u}(x_k) = \bar{v}(x_k) - \{I_{j_k} + (I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger}) \underbrace{N_{x_k}}_{symmetric} \}^{-1}(I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger})\ell_{x_k}$$

Corollary

Suppose $\bar{B}_{x_k}$ has full column rank, $\mathcal{B}_{x_k}$ is symmetric positive definite on $\{w \mid B^T w = 0\}$. Then the proximal Newton equation $J(x_k)[u(x_k)] = -v(x_k)$ can be computed by

$$u(x_k) = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix},$$

where $w(x_k) = \mathrm{argmin}_{\bar{B}_{x_k}^T w = 0}\, \ell_{x_k}^T w + \frac{1}{2} w^T \mathcal{B}_{x_k} w$.

A Riemannian proximal Newton-CG method

$$\bar{u}(x_k) = \bar{v}(x_k) - \{I_{j_k} + (I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger}) \underbrace{N_{x_k}}_{symmetric} \}^{-1}(I_{j_k} - \bar{B}_{x_k}\bar{B}_{x_k}^{\dagger})\ell_{x_k}$$

Corollary

Suppose $\bar{B}_{x_k}$ has full column rank, $\mathcal{B}_{x_k}$ is symmetric positive definite on $\{w \mid B^T w = 0\}$. Then the proximal Newton equation $J(x_k)[u(x_k)] = -v(x_k)$ can be computed by

$$u(x_k) = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix},$$

where $w(x_k) = \mathrm{argmin}_{\bar{B}_{x_k}^T w = 0}\, \ell_{x_k}^T w + \frac{1}{2} w^T \mathcal{B}_{x_k} w$.

tCG can be used for the computation of $w(x_k)$.

A Riemannian proximal Newton-CG method

A Riemannian proximal Newton method (RPN)

1. $v(x_k) = \mathrm{argmin}_{v \in \mathrm{T}_{x_k} \mathcal{M}} \; f(x_k) + \langle \nabla f(x_k), v \rangle + \frac{1}{2t} \|v\|_F^2 + h(x_k + v)$;
2. $d(x_k) = \begin{pmatrix} \bar{d}(x_k) \\ \hat{d}(x_k) \end{pmatrix} = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix}$, where $w(x_k)$ is an output of tCG for solving $\min_{\bar{B}_{x_k}^T w = 0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle$.
3. $x_{k+1} = R_{x_k}(\alpha_k d(x_k))$ with an appropriate step size α_k;

Question:

- Is $\mathcal{B}_{x_k}$ symmetric positive definite near a local minimizer x_*?
- What is the early termination conditions for tCG?
  - Guarantee global convergence;
  - Guarantee local superlinear convergence;

Is $\mathcal{B}_{x_k}$ symmetric positive definite near x_*?

Is $\mathcal{B}_{x_k}$ symmetric positive definite near x_*?

Assumption:

1. The function f is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
2. 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;
3. 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 $\hat{x} + \hat{v} = 0$;
4. The linear operator $\mathcal{B}_{x_*}$ is positive definite on the subspace $\mathfrak{L}_{x_*} = \{w \mid \bar{B}_{x_*}^T w = 0\}$.

Is $\mathcal{B}_{x_k}$ symmetric positive definite near x_*?

Assumption:

1. The function f is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
2. 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;
3. 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 $\hat{x} + \hat{v} = 0$;
4. The linear operator $\mathcal{B}_{x_*}$ is positive definite on the subspace $\mathfrak{L}_{x_*} = \{w \mid \bar{B}_{x_*}^T w = 0\}$.

- Under the second assumption, the intersection of the manifold and the sparsity constraints forms an embedded submanifold around x_*;
- $\mathcal{B}_{x_*}$ is the Riemannian Hessian of F at x_* for the submanifold;
- $\mathcal{B}_{x_*}$ is symmetric positive semidefinite on $\mathfrak{L}_{x_*}$;

Is $\mathcal{B}_{x_k}$ symmetric positive definite near x_*?

Assumption:

1. The function f is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
2. 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;
3. 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 $\hat{x} + \hat{v} = 0$;
4. The linear operator $\mathcal{B}_{x_*}$ is positive definite on the subspace $\mathfrak{L}_{x_*} = \{w \mid \bar{B}_{x_*}^T w = 0\}$.

Lemma

Suppose the above Assumption holds. Then there exists a neighborhood of x_*, denoted by $\mathcal{V}_2$, and a positive constant χ_ϵ such that the smallest eigenvalue of $\mathcal{B}_x$ on $\mathfrak{L}_x$ is greater than χ_ϵ for all $x \in \mathcal{V}_2$. This implies $\mathcal{B}_x$ is positive definite on $\mathfrak{L}_x$ for all $x \in \mathcal{V}_2$.

Early termination conditions in tCG

tCG step

2. $d(x_k) = \begin{pmatrix} \bar{d}(x_k) \\ \hat{d}(x_k) \end{pmatrix} = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix}$, where $w(x_k)$ is an output of tCG for solving $\min_{\bar{B}_{x_k}^T w = 0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle$.

A Riemannian proximal Newton-CG method

Early termination conditions in tCG

tCG step

2. $d(x_k) = \begin{pmatrix} \bar{d}(x_k) \\ \hat{d}(x_k) \end{pmatrix} = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix}$, where $w(x_k)$ is an output of tCG for solving $\min_{\bar{B}_{x_k}^T w = 0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle$.

Difficulty

- Smooth:

  $$\text{approximately} \quad \min_{d \in \mathrm{T}_{x_k} \mathcal{M}} \langle \mathrm{grad}\, f(x_k), d \rangle + \frac{1}{2} \langle \mathrm{Hess}\, f(x_k)[d], d \rangle,$$

  find $d(x_k)$ such that $\langle d(x_k), \mathrm{grad}\, f(x_k) \rangle < 0$;

- Nonsmooth:

  $$\text{approximately} \quad \min_{\bar{B}_{x_k}^T w = 0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle,$$

  find $w(x_k)$ such that $d(x_k)$ is a descent direction;

Early termination conditions in tCG

tCG step

2. $d(x_k) = \begin{pmatrix} \bar{d}(x_k) \\ \hat{d}(x_k) \end{pmatrix} = \begin{pmatrix} \bar{v}(x_k) + w(x_k) \\ \hat{v}(x_k) \end{pmatrix}$, where $w(x_k)$ is an output of tCG for solving $\min_{\bar{B}_{x_k}^T w = 0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle$.

Difficulty

- Smooth:

  $$\text{approximately} \quad \min_{d \in \mathrm{T}_{x_k} \mathcal{M}} \langle \operatorname{grad} f(x_k), d \rangle + \frac{1}{2} \langle \operatorname{Hess} f(x_k)[d], d \rangle,$$

  find $d(x_k)$ such that $\langle d(x_k), \operatorname{grad} f(x_k) \rangle < 0$;

- Nonsmooth:

  $$\text{approximately} \quad \min_{\bar{B}_{x_k}^T w = 0} \langle \ell_{x_k}, w \rangle + \frac{1}{2} \langle w, \mathcal{B}_{x_k} w \rangle,$$

  find $w(x_k)$ such that $d(x_k)$ is a descent direction;

The early termination conditions for the smooth case are not sufficient.

Early termination conditions in tCG

Algorithm: Truncated conjugate gradient (tCG)

Input: $\vartheta > 0$, $\gamma > 0$, $\tau > 0$, $\theta > 0$, and $\kappa \in (0, 1)$;
Output: $(w(x), \mathrm{status})$;

if $G_x(v(x)) > G_x(0)$ **then**
  return $w(x) = 0$ and $\mathrm{status} =' early1'$;
end if
$z = \mathfrak{B} v(x)$;
if $\langle v(x), z \rangle + \tau \|\hat{v}(x)\|_F^2 < \gamma \|v(x)\|_F^2$ **then**
  return $w(x) = 0$ and $\mathrm{status} =' early2'$;
end if
$w_0 = 0$, $r_0 = P_x(\ell_x)$, $o_0 = -r_0$, $\delta_0 = \langle r_0, r_0 \rangle$, $t_0 = z$;
...... (CG iterations)

Omit subscript k for simplicity

Early termination conditions in tCG

Algorithm: Truncated conjugate gradient (tCG)

Input: $\vartheta > 0$, $\gamma > 0$, $\tau > 0$, $\theta > 0$, and $\kappa \in (0, 1)$;
Output: $(w(x), \mathrm{status})$;

1: **if** $G_x(v(x)) > G_x(0)$ **then**
2:     return $w(x) = 0$ and $\mathrm{status} =' early1'$;
3: **end if**
4: $z = \mathfrak{B} v(x)$;
5: **if** $\langle v(x), z\rangle + \tau \|\hat{v}(x)\|_F^2 < \gamma \|v(x)\|_F^2$ **then**
6:     return $w(x) = 0$ and $\mathrm{status} =' early2'$;
7: **end if**
8: $w_0 = 0$, $r_0 = P_x(\ell_x)$, $o_0 = -r_0$, $\delta_0 = \langle r_0, r_0\rangle$, $t_0 = z$;
9: ...... (CG iterations)

- $G_x(u) = f(x) + \langle \nabla f(x), u\rangle + \frac{1}{2}\langle u, \mathfrak{B}_x u\rangle + \frac{\tau}{2}\|\hat{u}(x)\|_F^2 + h(x+u)$;
- Use to guarantee global convergence;
- $\frac{\tau}{2}\|\hat{u}(x)\|_F^2$ is added for the condition in Step 5;

Early termination conditions in tCG

Algorithm: Truncated conjugate gradient (tCG)

Input: $\vartheta > 0$, $\gamma > 0$, $\tau > 0$, $\theta > 0$, and $\kappa \in (0, 1)$;
Output: $(w(x), \mathrm{status})$;

```
if  G_x(v(x)) > G_x(0)  then
    return w(x) = 0 and status = 'early1';
end if
z = 𝔅 v(x);
if  ⟨v(x), z⟩ + τ‖v̂(x)‖_F^2 < γ‖v(x)‖_F^2  then
    return w(x) = 0 and status = 'early2';
end if
w_0 = 0, r_0 = P_x(ℓ_x), o_0 = −r_0, δ_0 = ⟨r_0, r_0⟩, t_0 = z;
...... (CG iterations)
```

- Use to guarantee global convergence;
- $\tau \|\hat{v}(x)\|_F^2$ is used since $\mathfrak{B}_x \succ 0$ may not hold;

A Riemannian proximal Newton-CG method

Early termination conditions in tCG

Algorithm: Truncated conjugate gradient (tCG)

Input: $\vartheta > 0$, $\gamma > 0$, $\tau > 0$, $\theta > 0$, and $\kappa \in (0, 1)$;
Output: $(w(x), \mathrm{status})$;

...... (See the previous slide)
$w_0 = 0$, $r_0 = P_x(\ell_x)$, $o_0 = -r_0$, $\delta_0 = \langle r_0, r_0 \rangle$, $t_0 = z$;
for $i = 0, 1, \ldots$ **do**
    $p_i = \mathcal{B} o_i$ and $q_i = P_x(p_i)$;
    if $\langle o_i, q_i \rangle \leq \vartheta \delta_i$ **then**
        return $w(x) = w_i$ and $\mathrm{status} =' neg'$;
    end if
    ...... (Remaining CG iterations)
end for

An existing early termination condition

Early termination conditions in tCG

Algorithm: Truncated conjugate gradient (tCG)

Input: $\vartheta > 0$, $\gamma > 0$, $\tau > 0$, $\theta > 0$, and $\kappa \in (0, 1)$;
Output: $(w(x), \mathrm{status})$;

...... (See previous slides)
for $i = 0, 1, \ldots$ **do**
    ...... (See previous slides)
    $\alpha_i = \frac{\langle r_i, r_i \rangle}{\langle o_i, q_i \rangle}$; $w_{i+1} = w_i + \alpha_i o_i$; $r_{i+1} = r_i + \alpha_i q_i$;
    $d_{i+1} = \begin{pmatrix} \bar{v}(x) + w_{i+1} \\ \hat{v}(x) \end{pmatrix}$, $t_{i+1} = t_i + \alpha_i \begin{pmatrix} p_i \\ \mathfrak{B}_{21} o_i \end{pmatrix}$;
    if $\langle d_{i+1}, t_{i+1} \rangle + \tau \|\hat{v}(x)\|_F^2 < \gamma \|d_{i+1}\|_F^2$ or $G_x(d_{i+1}) > G_x(0)$ **then**
        return $w(x) = w_i$ and $\mathrm{status} = ' early3'$;
    end if
    ...... (Remaining CG iterations)
end for

Use to guarantee global convergence

Early termination conditions in tCG

Algorithm: Truncated conjugate gradient (tCG)

Input: $\vartheta > 0$, $\gamma > 0$, $\tau > 0$, $\theta > 0$, and $\kappa \in (0, 1)$;
Output: $(w(x), \mathrm{status})$;

...... (See previous slides)
for $i = 0, 1, \dots$ **do**
  ...... (See previous slides)
  $\beta_{i+1} = \frac{\langle r_{i+1}, r_{i+1} \rangle}{\langle r_i, r_i \rangle}$; $o_{i+1} = -r_{i+1} + \beta_{i+1} o_i$;
  $\delta_{i+1} = \langle r_{i+1}, r_{i+1} \rangle + \beta_{i+1}^2 \delta_i$; (Note that $\delta_{i+1} = \langle o_{i+1}, o_{i+1} \rangle$)
  $i = i + 1$;
  if $\|r_i\|_F \leq \|r_0\|_F \min(\|r_0\|_F^\theta, \kappa)$ **then**
    return $w(x) = w_i$, and $\mathrm{status} = '\mathit{lin}'$ if $\|r_0\|_F^\theta > \kappa$ and $\mathrm{status} = '\mathit{sup}'$ otherwise;
  end if
end for

An existing early termination condition

A Riemannian proximal Newton-CG method

Assumption:

1. The function f is twice continuously differentiable with a Lipschitz continuous gradient;

Theorem

Suppose the above Assumption holds and the parameters are appropriately chosen. Then it holds that

$$\lim_{k \to \infty} \|v(x_k)\|_F = 0.$$

A Riemannian proximal Newton-CG method

Assumption:

1. The function f is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
2. 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;
3. 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 $\hat{x} + \hat{v} = 0$;
4. The function F is ς-geodesically strongly convex at x_*, i.e., there exists a neighborhood $\tilde{\mathcal{U}}_{x_*}$ of x_* in $\mathcal{M}$ such that

$$F(y) \geq F(x_*) + \frac{\varsigma}{2} \| \mathrm{Exp}_{x_*}^{-1}(y) \|_F^2$$

holds for any $y \in \tilde{\mathcal{U}}_{x_*}$.

A Riemannian proximal Newton-CG method

Assumption:

1. The function f is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
2. 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;
3. 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 $\hat{x} + \hat{v} = 0$;
4. The function F is ς-geodesically strongly convex at x_*, i.e., there exists a neighborhood $\tilde{\mathcal{U}}_{x_*}$ of x_* in $\mathcal{M}$ such that
$$F(y) \geq F(x_*) + \frac{\varsigma}{2} \| \mathrm{Exp}_{x_*}^{-1}(y) \|_F^2$$
holds for any $y \in \tilde{\mathcal{U}}_{x_*}$.

Lemma

Suppose the last Assumption holds, that is, the function $F = f + h$ is ς-geodesically strongly convex at x_*. Then the linear operator $\mathcal{B}_{x_*}$ is positive definite on $\mathfrak{L}_{x_*}$.

Assumption:

1. The function f is twice continuously differentiable with a Lipschitz continuous Euclidean Hessian;
2. 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;
3. 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 $\hat{x} + \hat{v} = 0$;
4. The function F is ς-geodesically strongly convex at x_*, i.e., there exists a neighborhood $\tilde{\mathcal{U}}_{x_*}$ of x_* in $\mathcal{M}$ such that
$$F(y) \geq F(x_*) + \frac{\varsigma}{2} \| \mathrm{Exp}_{x_*}^{-1}(y) \|_F^2$$
holds for any $y \in \tilde{\mathcal{U}}_{x_*}$.

Theorem

Suppose the previous assumptions hold. If x is sufficiently close x_* and the parameters are appropriately chosen, then tCG terminates only due to the accurate condition, i.e., $\|r_i\|_F \leq \|r_0\|_F \min(\|r_0\|_F^{\theta}, \kappa)$.

Theorem

Suppose the previous Assumptions hold and the parameters are appropriately chosen. Then there exists a neighborhood of x_*, *denoted by* $\mathcal{V}_8$, *such that if the step size one is used, then the convergence rate is* $\min(1+\theta, 2)$, *i.e.,* $\|R_x(d(x)) - x_*\|_F \leq C_{\mathrm{up}} \|x - x_*\|_F^{\min(1+\theta,2)}$ *holds for any* $x \in \mathcal{V}_8$ *and a constant* $C_{\mathrm{up}} > 0$.

A Riemannian proximal Newton-CG method

Theorem

Suppose the previous Assumptions hold and the parameters are appropriately chosen. Then there exists a neighborhood of x_*, denoted by $\mathcal{V}_8$, such that if the step size one is used, then the convergence rate is $\min(1+\theta, 2)$, i.e., $\|R_x(d(x)) - x_*\|_F \leq C_{\mathrm{up}} \|x - x_*\|_F^{\min(1+\theta,2)}$ holds for any $x \in \mathcal{V}_8$ and a constant $C_{\mathrm{up}} > 0$.

Is step size one acceptable for x sufficiently close to x_*?
That is to make objective function sufficiently descent.

A Riemannian proximal Newton-CG method

Theorem

Suppose the previous Assumptions hold and the parameters are appropriately chosen. Then there exists a neighborhood of x_*, denoted by $\mathcal{V}_8$, such that if the step size one is used, then the convergence rate is $\min(1+\theta, 2)$, *i.e.,* $\|R_x(d(x)) - x_*\|_F \leq C_{\mathrm{up}} \|x - x_*\|_F^{\min(1+\theta,2)}$ *holds for any* $x \in \mathcal{V}_8$ *and a constant* $C_{\mathrm{up}} > 0$.

Is step size one acceptable for x sufficiently close to x_*?
That is to make objective function sufficiently descent.

- For smooth Riemannian optimization problem, step size one is acceptable eventually for Riemannian Newton method;
- For Euclidean nonsmooth optimization problem $F = f + g$, step size one is also acceptable eventually for proximal Newton method [LSS14];

Example

- Consider $F : \mathbb{R}^2 \to \mathbb{R} : (x_1, x_2)^T \mapsto \underbrace{x_1^2 - 3x_1 + 1 + x_2^2}_{f(x)} + \underbrace{|x_1| + |x_2|}_{g(x)}$;
- The unique minimizer: $x_* = (1, 0)^T$;
- $x = (1 + \epsilon, 0)^T$ with $|\epsilon|$ being arbitrarily small;
- Proximal Newton direction: $u(x) = -(\epsilon, 0)^T$;
- Retraction: $R : \mathrm{T}\,\mathcal{M} \to \mathcal{M} : \eta_x \mapsto x + \eta_x + \begin{pmatrix} 0 \\ 2\eta_x^T \eta_x \end{pmatrix}$;
- $R(u(x)) = (1, 2\epsilon^2)^T$;
- $F(R_x(u(x))) - F(x) = 4\epsilon^4 + \epsilon^2 > 0$;
- Step size one is not acceptable for any $\epsilon > 0$;

Example

- Consider $F : \mathbb{R}^2 \to \mathbb{R} : (x_1, x_2)^T \mapsto \underbrace{x_1^2 - 3x_1 + 1 + x_2^2}_{f(x)} + \underbrace{|x_1| + |x_2|}_{g(x)};$
- The unique minimizer: $x_* = (1, 0)^T$;
- $x = (1 + \epsilon, 0)^T$ with $|\epsilon|$ being arbitrarily small;
- Proximal Newton direction: $u(x) = -(\epsilon, 0)^T$;
- Retraction: $R : \mathrm{T}\,\mathcal{M} \to \mathcal{M} : \eta_x \mapsto x + \eta_x + \begin{pmatrix} 0 \\ 2\eta_x^T \eta_x \end{pmatrix};$
- $R(u(x)) = (1, 2\epsilon^2)^T$;
- $F(R_x(u(x))) - F(x) = 4\epsilon^4 + \epsilon^2 > 0$;
- Step size one is not acceptable for any $\epsilon > 0$;

The answer is negative for nonsmooth Riemannian problems.

Difficulty comes from the nonsmoothness and the curvature.

A Riemannian proximal Newton-CG method

Two consecutive iterations near x_* guarantee sufficient descent.

Theorem

Suppose that the previous Assumptions hold and that there exists a neighborhood of x_*, denoted by $\mathcal{V}_9$, such that for any $x \in \mathcal{V}_9$, it holds that $\|R_x(d(x)) - x_*\|_F \leq C_{\mathrm{up}}\|x - x_*\|_F^{\varkappa}$ for a $\varkappa > \sqrt{2}$ and $R_x(d(x)) \in \mathcal{V}_9$. Then there exists a neighborhood of x_*, denoted by $\mathcal{V}_{10}$, and a constant $\rho_1 > 0$ such that for any $x \in \mathcal{V}_{10}$, it holds that

$$F(x_{++}) \leq F(x) - \rho_1 \|v(x)\|_F^2,$$

where $x_+ = R_x(d(x))$ and $x_{++} = R_{x_+}(d(x_+))$.

A Riemannian proximal Newton-CG method

Two consecutive iterations near x_* guarantee sufficient descent.

Theorem

Suppose that the previous Assumptions hold and that there exists a neighborhood of x_*, denoted by $\mathcal{V}_9$, such that for any $x \in \mathcal{V}_9$, it holds that $\|R_x(d(x)) - x_*\|_F \leq C_{\mathrm{up}} \|x - x_*\|_F^{\varkappa}$ for a $\varkappa > \sqrt{2}$ and $R_x(d(x)) \in \mathcal{V}_9$. Then there exists a neighborhood of x_*, denoted by $\mathcal{V}_{10}$, and a constant $\rho_1 > 0$ such that for any $x \in \mathcal{V}_{10}$, it holds that

$$F(x_{++}) \leq F(x) - \rho_1 \|v(x)\|_F^2,$$

where $x_+ = R_x(d(x))$ and $x_{++} = R_{x_+}(d(x_+))$.

The global convergence result becomes: $\liminf_{k \to \infty} \|v(x_k)\|_F = 0$.

A new interpretation of RPN

Lemma

Suppose the previous Assumptions hold. Then there exists a neighborhood of x_*, denoted by $\mathcal{V}_5$, such that

$$u(x) = \operatorname*{argmin}_{u \in \mathrm{T}_x \mathcal{M}, \hat{u} = \hat{v}(x)} G_x(u) = \frac{1}{2}\langle u, \mathfrak{B}_x u\rangle + \nabla f(x)^T u + \mu\|x + u\|_1 \quad (1)$$

holds for any $x \in \mathcal{V}_5$.

- First, find the ManPG search direction $v(x)$;
- Fixed the entries that corresponds to the zero of $x + v$;
- Solve (1) for $u(x)$;

A new interpretation of RPN

Lemma

Suppose the previous Assumptions hold. Then there exists a neighborhood of x_*, denoted by $\mathcal{V}_5$, such that

$$u(x) = \operatorname*{argmin}_{u \in \mathrm{T}_x \mathcal{M}, \hat{u} = \hat{v}(x)} G_x(u) = \frac{1}{2}\langle u, \mathfrak{B}_x u\rangle + \nabla f(x)^T u + \mu \|x + u\|_1 \quad (1)$$

holds for any $x \in \mathcal{V}_5$.

- $\mathcal{M}_{sub}$: submanifold of the intersection of $\mathcal{M}$ and the sparse constraints;
- $\mathfrak{B}_x^{(11)}$ is the Riemannian Hessian at x with respect to $\mathcal{M}_{sub}$;
- $u(x)$ is the Riemannian Newton direction on $\mathcal{M}_{sub}$;

A new interpretation of RPN

Lemma

Suppose the previous Assumptions hold. Then there exists a neighborhood of x_*, denoted by $\mathcal{V}_5$, such that

$$u(x) = \operatorname*{argmin}_{u \in \mathrm{T}_x \mathcal{M}, \hat{u} = \hat{v}(x)} G_x(u) = \frac{1}{2}\langle u, \mathfrak{B}_x u\rangle + \nabla f(x)^T u + \mu \|x + u\|_1 \quad (1)$$

holds for any $x \in \mathcal{V}_5$.

- $\mathcal{M}_{sub}$: submanifold of the intersection of $\mathcal{M}$ and the sparse constraints;
- $\mathfrak{B}_x^{(11)}$ is the Riemannian Hessian at x with respect to $\mathcal{M}_{sub}$;
- $u(x)$ is the Riemannian Newton direction on $\mathcal{M}_{sub}$;

No counterpart in the Euclidean space.

Outline

- Proximal gradient method and its variants;
- A Riemannian proximal Newton method;
- A Riemannian proximal Newton-CG method;
- Numerical experiments;

Sparse PCA problem

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

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

Table: An average result of 20 random runs for random data. Multiple values of n, p, and μ are used. The subscript k indicates a scale of 10^k.

(n, p, μ)	Algo	iter	Fval	$\|v(x_k)\|_F$	time	sparsity
(400, 8, 0.8)	ManPG	3416.15	-2.16_1	3.66_{-9}	2.69	0.63
(400, 8, 0.8)	ManPG-Ada	1281.55	-2.16_1	1.06_{-10}	1.21	0.63
(400, 8, 0.8)	ManPQN	1260.40	-2.16_1	9.83_{-11}	0.72	0.63
(400, 8, 0.8)	RPN-CG	204.85	-2.16_1	1.16_{-11}	0.37	0.63
(800, 8, 0.8)	ManPG	4232.80	-5.92_1	1.84_{-7}	3.56	0.48
(800, 8, 0.8)	ManPG-Ada	1867.05	-5.92_1	2.57_{-10}	1.80	0.48
(800, 8, 0.8)	ManPQN	1883.80	-5.92_1	1.22_{-10}	1.43	0.48
(800, 8, 0.8)	RPN-CG	215.05	-5.92_1	1.07_{-11}	0.60	0.48

Table: An average result of 20 random runs for random data. Multiple values of n, p, and μ are used. The subscript k indicates a scale of 10^k.

(n, p, μ)	Algo	iter	Fval	$\|v(x_k)\|_F$	time	sparsity
(400, 8, 0.8)	ManPG	3416.15	-2.16_1	3.66_{-9}	2.69	0.63
(400, 8, 0.8)	ManPG-Ada	1281.55	-2.16_1	1.06_{-10}	1.21	0.63
(400, 8, 0.8)	ManPQN	1260.40	-2.16_1	9.83_{-11}	0.72	0.63
(400, 8, 0.8)	RPN-CG	204.85	-2.16_1	1.16_{-11}	0.37	0.63
(800, 8, 0.8)	ManPG	4232.80	-5.92_1	1.84_{-7}	3.56	0.48
(800, 8, 0.8)	ManPG-Ada	1867.05	-5.92_1	2.57_{-10}	1.80	0.48
(800, 8, 0.8)	ManPQN	1883.80	-5.92_1	1.22_{-10}	1.43	0.48
(800, 8, 0.8)	RPN-CG	215.05	-5.92_1	1.07_{-11}	0.60	0.48

- Proximal gradient on Stiefel manifold: ManPG, ManPG-Ada [CMSZ20];
- Proximal quasi-Newton on Stiefel manifold: ManPQN [WY23];
- The proposed method: RPN-CG;

Table: An average result of 20 random runs for random data. Multiple values of n, p, and μ are used. The subscript k indicates a scale of 10^k.

(n, p, μ)	Algo	iter	Fval	$\|v(x_k)\|_F$	time	sparsity
(400, 8, 0.8)	ManPG	3416.15	-2.16_1	3.66_{-9}	2.69	0.63
(400, 8, 0.8)	ManPG-Ada	1281.55	-2.16_1	1.06_{-10}	1.21	0.63
(400, 8, 0.8)	ManPQN	1260.40	-2.16_1	9.83_{-11}	0.72	0.63
(400, 8, 0.8)	RPN-CG	204.85	-2.16_1	1.16_{-11}	0.37	0.63
(800, 8, 0.8)	ManPG	4232.80	-5.92_1	1.84_{-7}	3.56	0.48
(800, 8, 0.8)	ManPG-Ada	1867.05	-5.92_1	2.57_{-10}	1.80	0.48
(800, 8, 0.8)	ManPQN	1883.80	-5.92_1	1.22_{-10}	1.43	0.48
(800, 8, 0.8)	RPN-CG	215.05	-5.92_1	1.07_{-11}	0.60	0.48

- Stop criterion: iter≥ 5000 or $\|v(x)\|_F \leq 10^{-10}$;
- The entries of A are drawn from the standard normal distribution;
- Runs that converges to the same minimizer are reported;
- Support estimation: $(x + v(x))_i$ nonzero and $|(x)_i| \geq \|v(x)\|_F$;

Table: An average result of 20 random runs for random data. Multiple values of n, p, and μ are used. The subscript k indicates a scale of 10^k.

(n, p, μ)	Algo	iter	Fval	$\|v(x_k)\|_F$	time	sparsity
(400, 8, 0.8)	ManPG	3416.15	-2.16_1	3.66_{-9}	2.69	0.63
(400, 8, 0.8)	ManPG-Ada	1281.55	-2.16_1	1.06_{-10}	1.21	0.63
(400, 8, 0.8)	ManPQN	1260.40	-2.16_1	9.83_{-11}	0.72	0.63
(400, 8, 0.8)	RPN-CG	204.85	-2.16_1	1.16_{-11}	0.37	0.63
(800, 8, 0.8)	ManPG	4232.80	-5.92_1	1.84_{-7}	3.56	0.48
(800, 8, 0.8)	ManPG-Ada	1867.05	-5.92_1	2.57_{-10}	1.80	0.48
(800, 8, 0.8)	ManPQN	1883.80	-5.92_1	1.22_{-10}	1.43	0.48
(800, 8, 0.8)	RPN-CG	215.05	-5.92_1	1.07_{-11}	0.60	0.48

RPN-CG always stops due to $\|v\|_F \leq 10^{-10}$
and is the most efficient one.

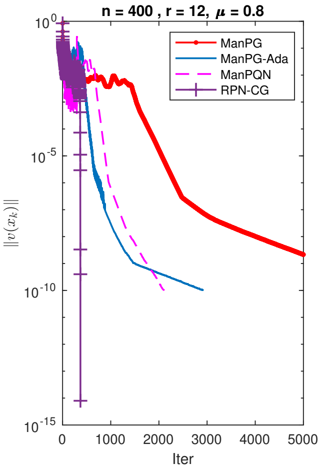


Figure: Sparse PCA: plots of $\|v(x_k)\|$ versus iterations and CPU times respectively.

The compressed modes (CM) problem aims to seek sparse solution of the independent-particle Schrödinger equation. It can be formulated as

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

where $H \in \mathbb{R}^{n \times n}$ denotes the discretized Schrödinger operator.

Table: An average result of 50 random runs for random data. Multiple values of n, p, and μ are used. The subscript k indicates a scale of 10^k.

(n, p, μ)	Algo	iter	Fval	$\|v(x_k)\|_F$	time	sparsity
(256, 4, 0.1)	ManPG	3000.00	2.49	4.03_{-5}	0.75	0.85
(256, 4, 0.1)	ManPG-Ada	3000.00	2.49	9.49_{-5}	0.88	0.85
(256, 4, 0.1)	ManPQN	3000.00	2.49	9.06_{-6}	1.22	0.84
(256, 4, 0.1)	RPN-CG	92.54	2.49	2.66_{-9}	0.20	0.86
(512, 4, 0.1)	ManPG	3000.00	3.29	3.83_{-5}	0.76	0.86
(512, 4, 0.1)	ManPG-Ada	3000.00	3.29	1.16_{-4}	0.88	0.86
(512, 4, 0.1)	ManPQN	3000.00	3.30	1.44_{-6}	2.98	0.86
(512, 4, 0.1)	RPN-CG	147.40	3.29	2.29_{-9}	0.48	0.88

- Stop criterion: iter$\geq$ 3000 or $\|v(x)\|_F \leq 10^{-8}$;
- Different runs may converge to different points;

Table: An average result of 50 random runs for random data. Multiple values of n, p, and μ are used. The subscript k indicates a scale of 10^k.

(n, p, μ)	Algo	iter	Fval	$\|v(x_k)\|_F$	time	sparsity
(256, 4, 0.1)	ManPG	3000.00	2.49	4.03_{-5}	0.75	0.85
(256, 4, 0.1)	ManPG-Ada	3000.00	2.49	9.49_{-5}	0.88	0.85
(256, 4, 0.1)	ManPQN	3000.00	2.49	9.06_{-6}	1.22	0.84
(256, 4, 0.1)	RPN-CG	92.54	2.49	2.66_{-9}	0.20	0.86
(512, 4, 0.1)	ManPG	3000.00	3.29	3.83_{-5}	0.76	0.86
(512, 4, 0.1)	ManPG-Ada	3000.00	3.29	1.16_{-4}	0.88	0.86
(512, 4, 0.1)	ManPQN	3000.00	3.30	1.44_{-6}	2.98	0.86
(512, 4, 0.1)	RPN-CG	147.40	3.29	2.29_{-9}	0.48	0.88

RPN-CG always stops due to $\|v\|_F \leq 10^{-8}$ and is the most efficient one.

None of other methods find a solution with $\|v\|_F \leq 10^{-8}$.

Numerical Experiments

Compressed modes

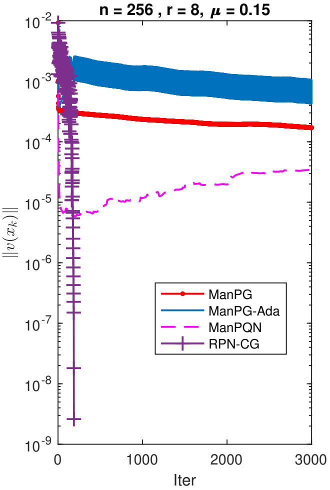


Figure: CM: plots of $\|v(x_k)\|$ versus iterations and CPU times respectively.

Summary

- Briefly review Euclidean and Riemannian proximal gradient method and its variants;
- Review the existing Riemannian proximal Newton method;
- Propose a Riemannian proximal Newton-CG method with global and local superlinear convergence gauranteed;
- Numerical experiments show its performance;

Future work

- Other types of $h(x)$;
- General manifold;
- Riemannian proximal quasi-Newton methods;
- Accelerated Riemannian proximal gradient method with theoretical guaranteed;

Thank you

Thank you!

References I

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.

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.

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.

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

References II

N. Josephy.
Newton's method for generalized equations.
Technical Summary Report, 1979.

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.

Vidvuds Ozolinš, 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.

Wutao Si, P. A. Absil, Wen Huang, Rujun Jiang, and Simon Vary.
A Riemannian Proximal Newton Method, 2024.
arXiv: 2304.04032v3.

Wutao Si, P.-A. Absil, Wen Huang, Rujun Jiang, and Simon Vary.
A Riemannian Proximal Newton Method.
SIAM Journal on Optimization, 34(1):654–681, 2024.

K. Scheinberg and X. Tang.
Practical inexact proximal quasi-newton method with global complexity analysis.
Mathematical Programming, (160):495–529, February 2016.

Qinsi Wang and Weihong Yang.
Proximal quasi-Newton method for composite optimization over the Stiefel manifold.
Journal of Scientific Computing, 95, 5 2023.

Qinsi Wang and Wei Hong Yang.
An adaptive regularized proximal Newton-type methods for composite optimization over the Stiefel manifold.
Computational Optimization and Applications, pages 1–39, 2024.

References III

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.