A Riemannian Proximal Newton Method

Speaker: Wutao Si

Xiamen University & UCLouvain

June 3, 2023

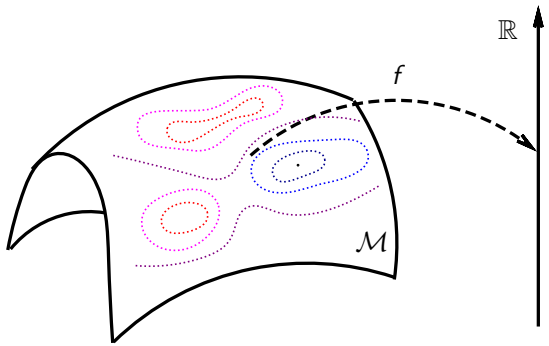
Joint work with Wen Huang, P.-A. Absil, Rujun Jiang, Simon Vary

SIAM Conference on Optimization (OP23)

Problem Statement

Optimization on Manifolds with Structure:

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

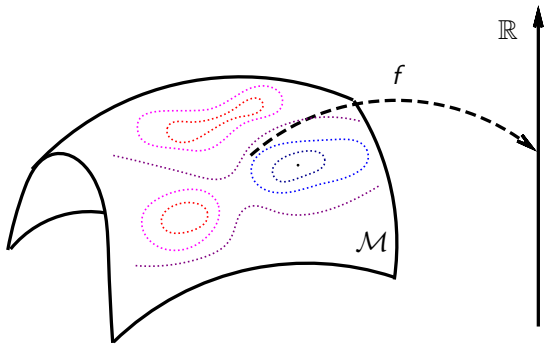


- $\mathcal{M}$ is a finite-dimensional Riemannian manifold;
- f is smooth and may be nonconvex; and
- $h(x)$ is continuous and convex but may be nonsmooth;

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].

Outline

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

Euclidean Proximal Gradient Method and its variants

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

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

Euclidean Proximal Gradient Method and its variants

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

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

- Proximal Gradient
- Proximal inexact Newton
- Proximal quasi-Newton

Euclidean Proximal Gradient Method and its variants

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

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

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

Euclidean Proximal Gradient Method and its variants

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

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

- Proximal Gradient
- 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;
- For convex f and h, $O\left(\frac{1}{k}\right)$ convergence rate , $O\left(\frac{1}{k^2}\right)$ for its accelerated version;
- Linear convergence rate for strongly convex f and convex h;
- Local convergence rate by KL property;

Euclidean Proximal Gradient Method and its variants

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

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

- Proximal Gradient
- Proximal inexact Newton
- Proximal quasi-Newton

Given x_0;

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

Euclidean Proximal Gradient Method and its variants

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

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

- Proximal Gradient
- Proximal inexact Newton
- Proximal quasi-Newton

Given x_0;

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

- H_k is Hessian or a positive definite approximation to Hessian [LSS14, MYZZ23];
- t_k is one for sufficiently large k;
- Quadratic/Superlinear convergence rate for strongly convex f and convex h;

[LSS14] 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.

[MYZZ23] Boris S Mordukhovich, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang. A globally convergent proximal Newton-type method in nonsmooth convex optimization. Mathematical Programming, 198(1):899-936, 2023.

Euclidean Proximal Gradient Method and its variants

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

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

- Proximal Gradient
- Proximal inexact Newton
- Proximal quasi-Newton

Given x_0, H_0;

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

Euclidean Proximal Gradient Method and its variants

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

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

- Proximal Gradient
- Proximal inexact Newton
- Proximal quasi-Newton

Given x_0, H_0;

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

- Dennis-Moré condition $\Longrightarrow$ superlinear convergence rate for strongly convex f and convex h [LSS14];

[LSS14] 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.

Outline

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

Riemannian proximal gradient method and its variants

Optimization with Structure:

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

Riemannian proximal gradient method and its variants

Optimization with Structure:

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

- Proximal Gradient 1
- Proximal Gradient 2
- Inexact version

Riemannian proximal gradient method and its variants

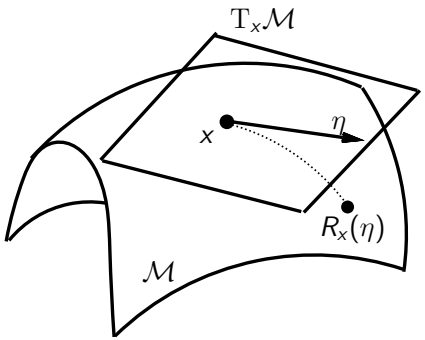
Optimization with Structure:

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

- Proximal Gradient 1
- Proximal Gradient 2
- Inexact version

[CMSZ20]: Given x_0,

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



[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.

Riemannian proximal gradient method and its variants

Optimization with Structure:

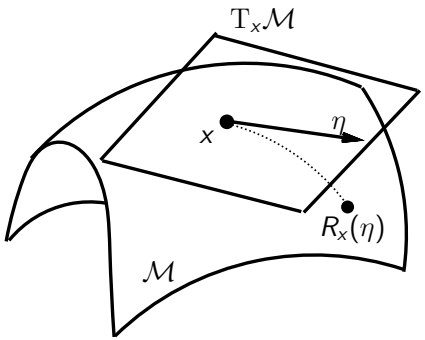
$$\min_{x\in\mathcal{M}} F(x) = f(x) + h(x),$$

- Proximal Gradient 1
- Proximal Gradient 2
- Inexact version

[CMSZ20]: Given x_0,

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

- Direction in the tangent space;
- Ambient space must be linear;
- Solved by a semismooth Newton method;



[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.

Riemannian proximal gradient method and its variants

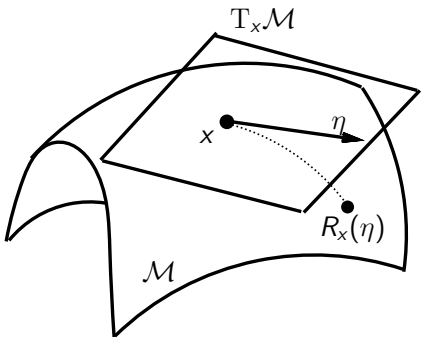
Optimization with Structure:

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

- Proximal Gradient 1
- Proximal Gradient 2
- Inexact version

[CMSZ20]: Given x_0,

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



- Direction in the tangent space;
- Ambient space must be linear;
- Solved by a semismooth Newton method;
- Any limit point is a critical point;
- No local convergence rate results;

[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.

Riemannian proximal gradient method and its variants

Optimization with Structure:

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

- Proximal Gradient 1
- Proximal Gradient 2
- Inexact version

[HW22]: Given x_0,

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

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

Riemannian proximal gradient method and its variants

Optimization with Structure:

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

- Proximal Gradient 1
- Proximal Gradient 2
- Inexact version

[HW22]: Given x_0,

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

- Direction in the tangent space;
- Well-defined for general manifold;
- Subproblem is difficult in general (simple for sphere);
- Any limit point is a critical point;
- $O\left(\frac{1}{k}\right)$ rate for retraction convex f and h;
- Local convergence rate by Riemannian KL property;

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

Riemannian proximal gradient method and its variants

Optimization with Structure:

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

- Proximal Gradient 1
- Proximal Gradient 2
- Inexact version

[HW23]: Given x_0,

$$\begin{cases} \text{Let } \ell_{x_k}(\eta) = \langle \mathrm{grad} f(x_k), \eta\rangle_{x_k} + \frac{L}{2}\|\eta\|^2_{x_k} + h(R_{x_k}(\eta)); \\ \hat{\eta}_k \text{ approximately solves } \min_{\eta\in \mathrm{T}_{x_k}\mathcal{M}} \ell_{x_k}(\eta) \text{ in the sense} \\ \text{its distance to a stationary point } \eta_k^* \text{ can be control, and} \\ \ell_{x_k}(0) \geq \ell_k(\hat{\eta}_k); \\ x_{k+1} = R_{x_k}(\hat{\eta}_k); \end{cases}$$

[HW23] W. Huang and K. Wei. An inexact Riemannian proximal gradient method. Computational Optimization and Applications, 2023:1-32.

Riemannian proximal gradient method and its variants

Optimization with Structure:

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

- Proximal Gradient 1
- Proximal Gradient 2
- Inexact version

[HW23]: Given x_0,

$$\begin{cases} \text{Let } \ell_{x_k}(\eta) = \langle \mathrm{grad} f(x_k), \eta\rangle_{x_k} + \frac{L}{2}\|\eta\|^2_{x_k} + h(R_{x_k}(\eta)); \\ \hat{\eta}_k \text{ approximately solves } \min_{\eta\in \mathrm{T}_{x_k}\mathcal{M}} \ell_{x_k}(\eta) \text{ in the sense} \\ \text{its distance to a stationary point } \eta_k^* \text{ can be control, and} \\ \ell_{x_k}(0) \geq \ell_k(\hat{\eta}_k); \\ x_{k+1} = R_{x_k}(\hat{\eta}_k); \end{cases}$$

- the search direction η_k in [CMSZ20] can be viewed as an inexact solution;
- Well-defined for general manifold;
- Local convergence rate by Riemannian KL property;

[HW23] W. Huang and K. Wei. An inexact Riemannian proximal gradient method. Computational Optimization and Applications, 2023:1-32.

Outline

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

Outline

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

Note that we focus on:

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

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:

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

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:

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

Does it converge superlinearly locally?

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:

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

Does it converge superlinearly locally?

Not necessarily!

Consider the Sparse PCA over sphere:

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

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

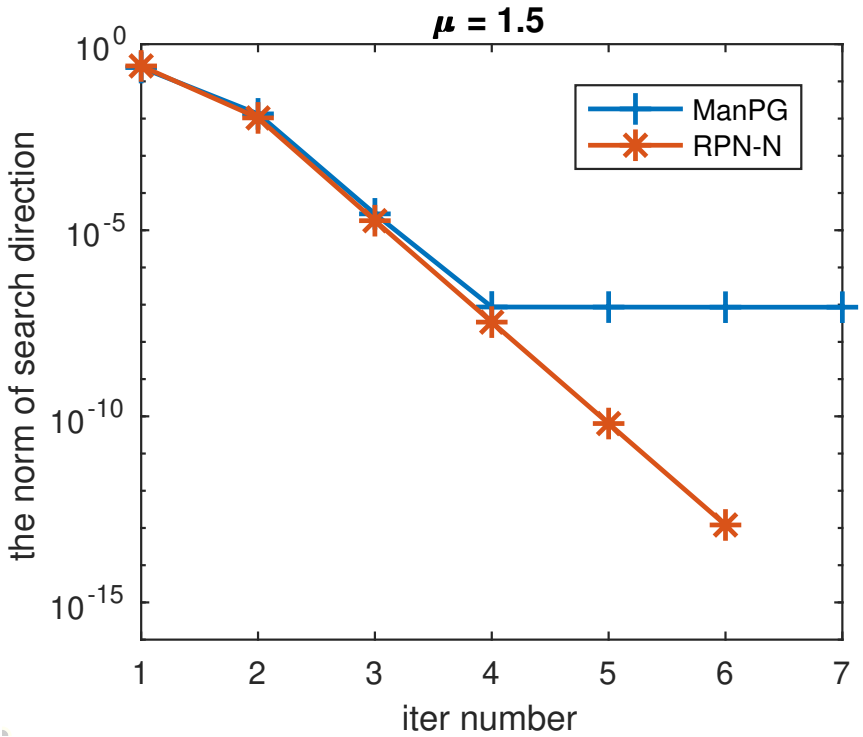


Figure: Comparisons of native generalization (RPN-N) and the proximal gradient method (ManPG) in [CMSZ20].

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:

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

- For $R_{x_k}(\eta)$, $x_k + \eta$ in h is only a first order approximation;

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:

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

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

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

For the smooth case, where $h(x) \equiv 0$.

Riemannian Newton method

Given $x_0 \in \mathcal{M}$,

1. Solve $\operatorname{Hess} f(x_k)[u_k] = -\operatorname{grad} f(x_k)$ for $u_k \in \mathrm{T}_{x_k}\mathcal{M}$;
2. $x_{k+1} = R_{x_k}(u_k)$.

For the smooth case, where $h(x) \equiv 0$.

Riemannian Newton method

Given $x_0 \in \mathcal{M}$,

1. Solve $\operatorname{Hess} f(x_k)[u_k] = -\operatorname{grad} f(x_k)$ for $u_k \in \mathrm{T}_{x_k}\mathcal{M}$;
2. $x_{k+1} = R_{x_k}(u_k)$.

- Let $v(x_k) = -\operatorname{grad} f(x_k)$, then the search direction u_k can be written in terms of $v(x_k)$, i.e., $\operatorname{Proj}_{\mathrm{T}_{x_k}\mathcal{M}}(\mathrm{D}v(x_k)[u_k]) = -v(x_k)$.

For the smooth case, where $h(x) \equiv 0$.

Riemannian Newton method

Given $x_0 \in \mathcal{M}$,

1. Solve $\operatorname{Hess} f(x_k)[u_k] = -\operatorname{grad} f(x_k)$ for $u_k \in \mathrm{T}_{x_k}\mathcal{M}$;
2. $x_{k+1} = R_{x_k}(u_k)$.

- Let $v(x_k) = -\operatorname{grad} f(x_k)$, then the search direction u_k can be written in terms of $v(x_k)$, i.e., $\operatorname{Proj}_{\mathrm{T}_{x_k}\mathcal{M}}(\mathrm{D}v(x_k)[u_k]) = -v(x_k)$.

Riemannian Newton method

Given $x_0 \in \mathcal{M}$,

1. Let $v(x_k) = -\operatorname{grad} f(x_k)$;
2. Solve $\operatorname{Proj}_{\mathrm{T}_{x_k}\mathcal{M}}(\mathrm{D}v(x_k)[u_k]) = -v(x_k)$ for $u_k \in \mathrm{T}_{x_k}\mathcal{M}$;
3. $x_{k+1} = R_{x_k}(u_k)$.

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 (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. $x_{k+1} = R_{x_k}(u(x_k))$;

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

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. $x_{k+1} = R_{x_k}(u(x_k))$;

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 (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. $x_{k+1} = R_{x_k}(u(x_k))$;

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 (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. $x_{k+1} = R_{x_k}(u(x_k))$;

Next, we will show:

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

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

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

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

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

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

Above problem can be rewritten as

$$\underset{B_x^T v = 0}{\operatorname{argmin}} \langle \nabla f(x), v \rangle + \frac{1}{2t}\|v\|_F^2 + h(x+v)$$

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 $\mathrm{T}_x^{\perp} \mathcal{M}$.

The Lagrangian function:

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

Therefore

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

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

Define

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

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

Define

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

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

[Gow04] M.S. 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. Semismooth homeomorphisms and strong stability of semidefinite and Lorentz complementarity problems. Mathematics of Operations Research, 28(1):39-63, 2003.

Lemma (G-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 $\partial_B F$ 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

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

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

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.

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 G-semismooth implicit function theorem is used

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 $J(x) = \mathrm{I}_n - \Lambda_x + t\Lambda_x(\nabla^2 f(x) - \mathcal{L}_x) : \mathrm{T}_x\mathcal{M} \to \mathrm{T}_x\mathcal{M}$, since $B_x^{\mathrm{T}} J(x) = 0$;
- The Riemannian proximal Newton direction: $J(x)u(x) = -v(x)$;

A Riemannian proximal Newton method

Local superlinear convergence rate

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;

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 $\hat{x} + \hat{v} = 0$.

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

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 superlinearly to x_*.

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

- KKT conditions:

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

- Closed form solutions:

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

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

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

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

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

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;

Numerical Experiments

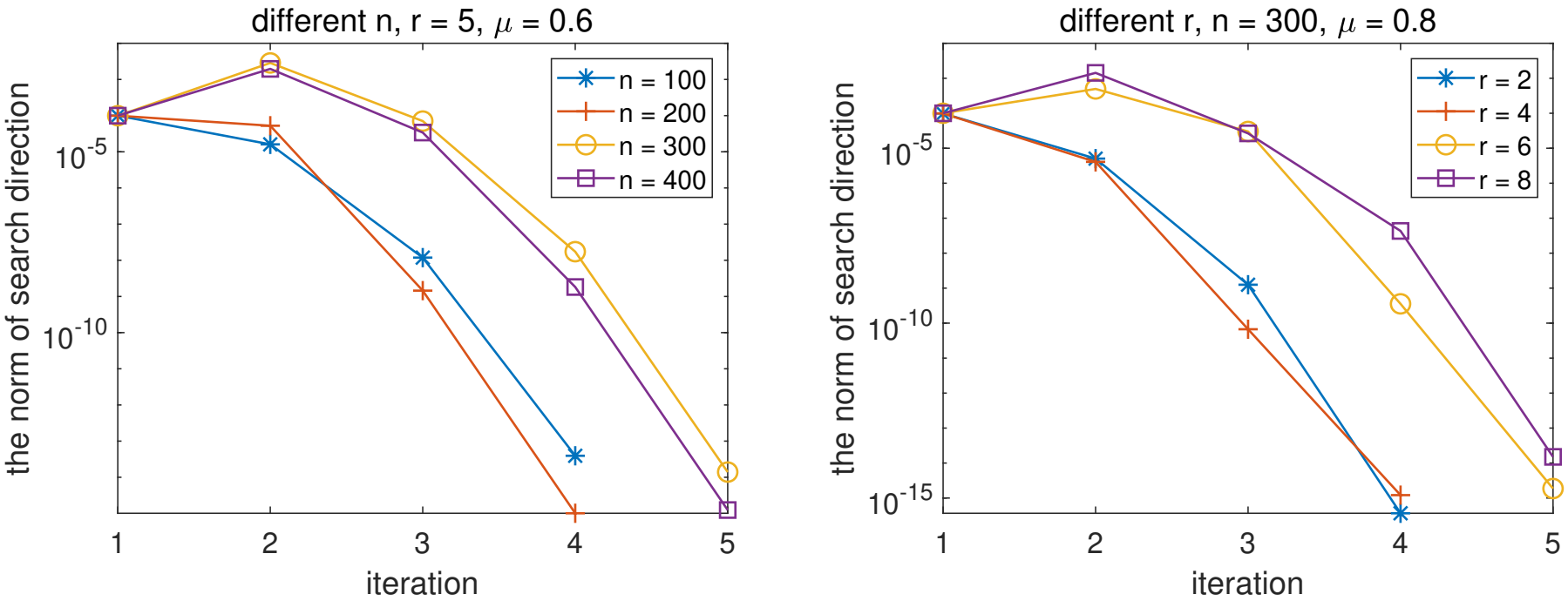


Figure: Random data. Left: different $n = \{100, 200, 300, 400\}$ with $r = 5$ and $\mu = 0.6$; Right: different $r = \{2, 4, 6, 8\}$ with $n = 300$ and $\mu = 0.8$

A Hybrid version of ManPG and RPN

Require: $x_0 \in \mathcal{M}$, $t > 0$, $\rho \in (0, \frac{1}{2}]$, $\epsilon > 0$;

for $k = 0, 1, \ldots$ **do**
  Compute v_k by solving the Riemannian proximal gradient subproblem;
  if $\|v_k\| > \epsilon$ **then**
    Set $\alpha = 1$;
    while $F(R_{x_k}(\alpha v_k)) > F(x_k) - \frac{1}{2}\alpha\|v_k\|^2$ **do**
      $\alpha = \rho\alpha$;
    end while
    $x_{k+1} = R_{x_k}(\alpha v_k)$;
  else
    Compute u_k by solving $J(x_k)u_k = -v_k$;
    $x_{k+1} = R_{x_k}(u_k)$;
  end if
end for

Numerical Experiments

Consider the simple version of sparse PCA with $r = 1$, i.e.,

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

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

Table: An average result of 5 random runs for random data with different setting of (n, μ). The subscript k indicates a scale of 10^k. iter-u denotes the number of using the new search direction u_k.

(n, μ)	Algo	iter	iter-v	iter-u	f	sparsity	$\|v_k\|$
(5000,1.5)	ManPG	3000	897	-	-4.59_1	0.37	7.41_{-8}
(5000,1.5)	RPN	334	-	5	-4.59_1	0.37	4.53_{-16}
(10000,1.8)	ManPG	3000	1736	-	-1.02_2	0.32	2.19_{-8}
(10000,1.8)	RPN	580	-	6	-1.02_2	0.32	5.69_{-16}
(30000,2.0)	ManPG	3000	1283	-	-3.98_2	0.22	1.19_{-8}
(30000,2.0)	RPN	347	-	5	-3.98_2	0.22	5.25_{-15}
(50000,2.2)	ManPG	3000	1069	-	-7.06_2	0.18	4.56_{-7}
(50000,2.2)	RPN	789	-	5	-7.06_2	0.18	1.41_{-14}
(80000,2.5)	ManPG	3000	834	-	-1.17_3	0.17	1.41_{-6}
(80000,2.5)	RPN	839	-	6	-1.17_3	0.17	1.94_{-15}

Stopping criteria: ManPG does not terminate until iteration attains the maximal iteration (3000), RPN terminate until $\|v_k\| \leq 10^{-12}$

Numerical Experiments

CPU Comparison

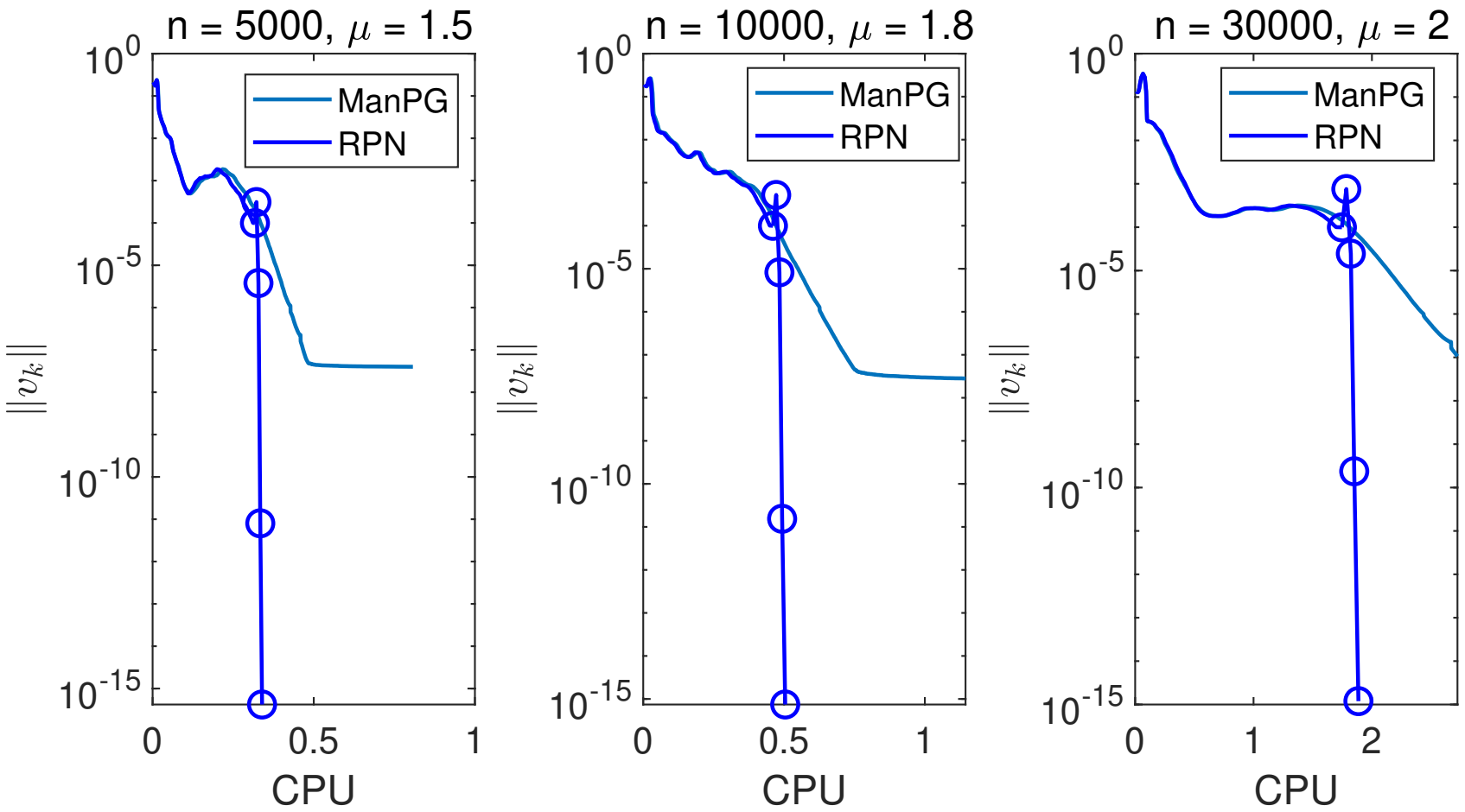


Figure: Random data: the norm of search direction v_k versus CPU for different (n, μ), where the blue circle indicates the use of the new direction u_k.

Synthetic Data [SCL$^+$18] : we first obtain an $m \times n$ noise-free matrix, then the data matrix A is generated by adding a random noise matrix, where each entry of the noise matrix is drawn form $\mathcal{N}(0, 0.25)$, we set $m = 400$, $n = 4000$ and $\mu = 1.2$.

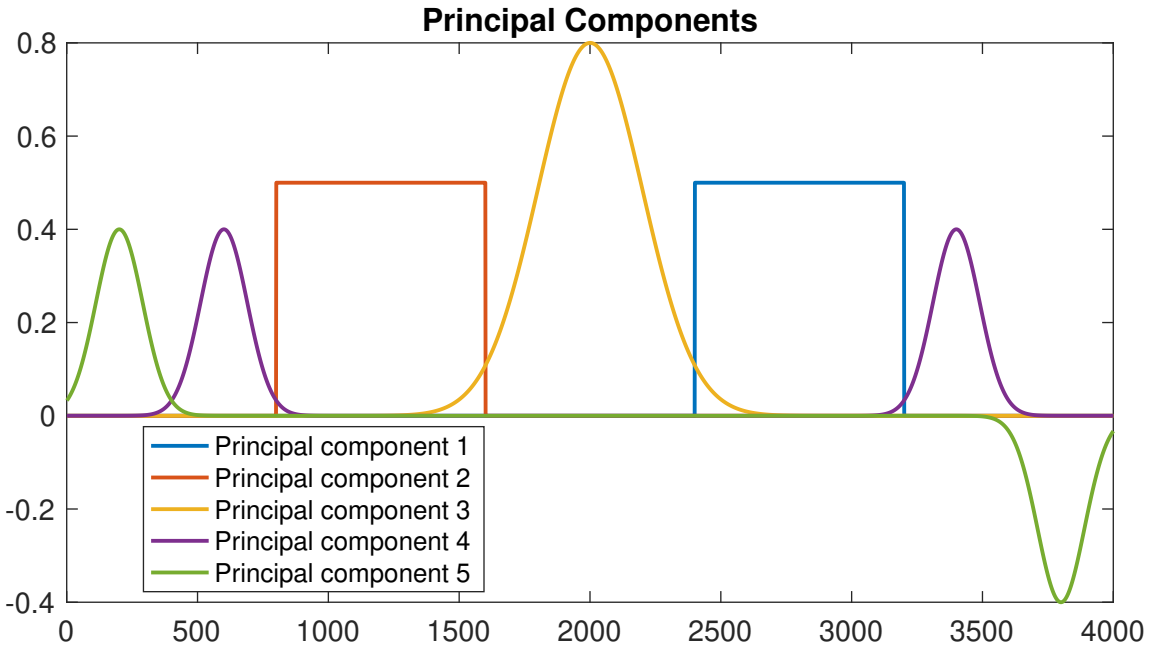


Figure: The five principal components used in the synthetic data.

Numerical Experiments

Synthetic Data

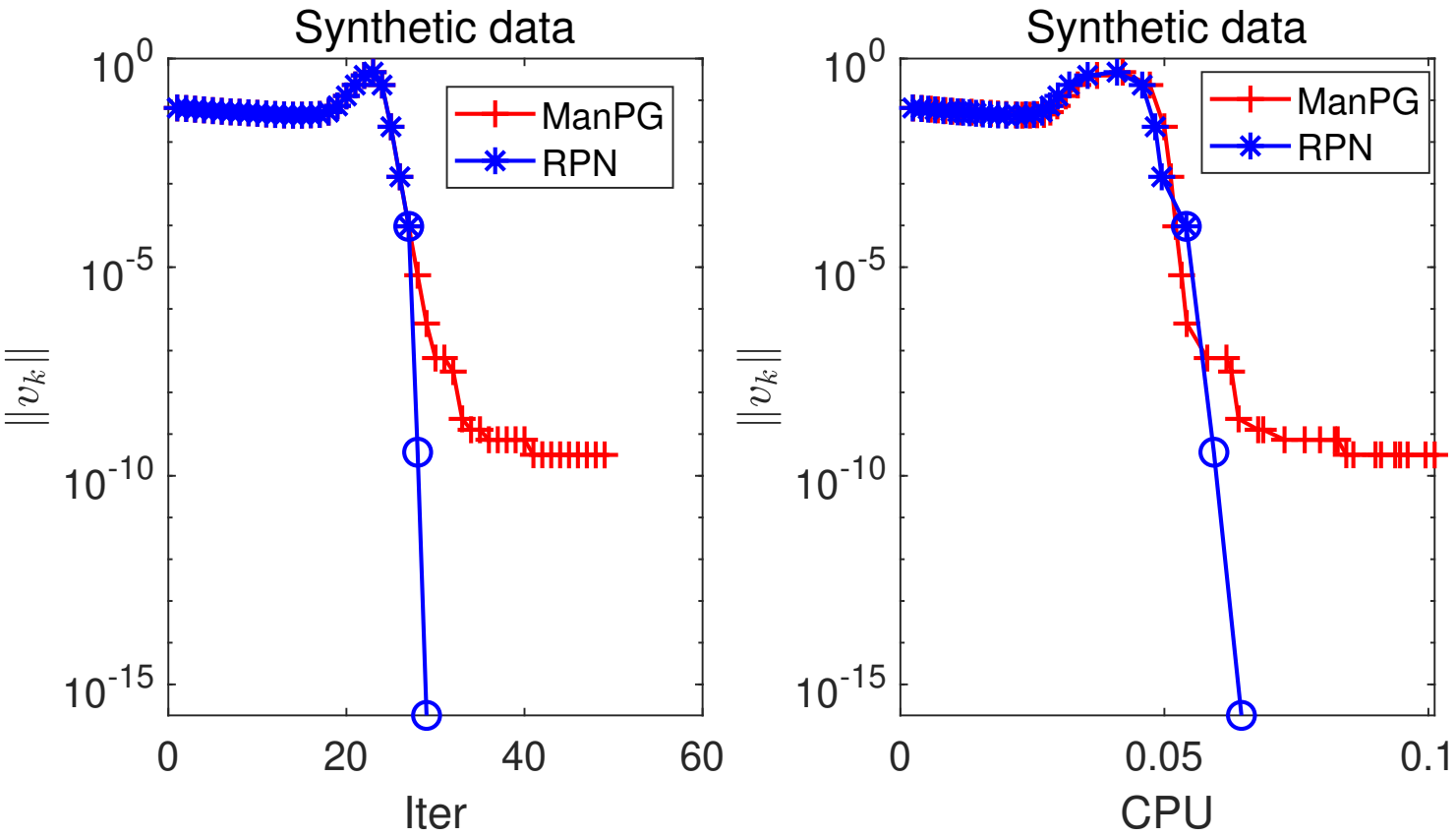


Figure: Plots of $\|v_k\|$ versus iterations and CPU times respectively, where $\|v_k\|$ is the norm of search direction, data matrix $A \in \mathbb{R}^{4000\times 400}$ is from the synthetic data, μ is set to be 1.2. Note that the blue circle indicates the use of the new direction u_k.

Summary

- Briefly review Euclidean and Riemannian proximal gradient method and its variants;
- Propose a Riemannian proximal Newton method [SAH+23]
- Local superlinear convergence rate is proven;
- Numerical experiments show its performance;

[SAH+23] W.Si, P.-A Absil, W. Huang, R. Jiang, and S. Vary (2023). A Riemannian Proximal Newton Method. arXiv preprint arXiv:2304.04032.

Future work

- Globalization;
- Other types of $h(x)$;
- General manifold;
- Riemannian proximal inexact-Newton methods;
- Riemannian proximal quasi-Newton methods;

Thank you

Thank you!

References I

Matthias Bollhofer, Aryan Eftekhari, Simon Scheidegger, and Olaf Schenk.

Large-scale sparse inverse covariance matrix estimation.
SIAM Journal on Scientific Computing, 41(1):A380–A401, 2019.

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.

Solving partial least squares regression via manifold optimization approaches.
IEEE transactions on neural networks and learning systems, 30(2):588–600, 2018.

M Seetharama Gowda.

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

Wen Huang and Ke Wei.

Riemannian proximal gradient methods.
Mathematical Programming, 194(1-2):371–413, 2022.

Wen Huang and Ke Wei.

An inexact Riemannian proximal gradient method.
Computational Optimization and Applications, pages 1–32, 2023.

Wen Huang, Meng Wei, Kyle A Gallivan, and Paul Van Dooren.

A Riemannian optimization approach to clustering problems.
arXiv preprint arXiv:2208.03858, 2022.

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.

References II

Boris S Mordukhovich, Xiaoming Yuan, Shangzhi Zeng, and Jin Zhang.

A globally convergent proximal Newton-type method in nonsmooth convex optimization.
Mathematical Programming, 198(1):899–936, 2023.

Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, and Stanley Osher.

Compressed models 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.

Wutao Si, P-A Absil, Wen Huang, Rujun Jiang, and Simon Vary.

A riemannian proximal newton method.
arXiv preprint arXiv:2304.04032, 2023.

Karl Sjöstrand, Line Harder Clemmensen, Rasmus Larsen, Gudmundur Einarsson, and Bjarne Ersbøll.

Spasm: A matlab toolbox for sparse statistical modeling.
Journal of Statistical Software, 84:1–37, 2018.

Hui Zou, Trevor Hastie, and Robert Tibshirani.

Sparse principal component analysis.
Journal of computational and graphical statistics, 15(2):265–286, 2006.

Yuqian Zhang, Yenson Lau, Han-wen Kuo, Sky Cheung, Abhay Pasupathy, and John Wright.

On the global geometry of sphere-constrained sparse blind deconvolution.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4894–4902, 2017.