Riemannian Optimization: A Proximal Newton Method

Speaker: Wen Huang

Xiamen University

December 10, 2023

ASA2023

Outline

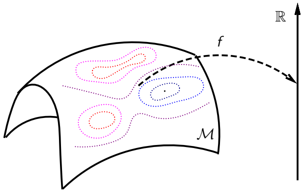
- Riemannian optimization;
- Applications;
- Smooth optimization framework;
- Research foci of Riemannian optimization;
- A Riemannian proximal Newton method;
- Summary;

Riemannian Optimization

Problem: Given $f(x) : \mathcal{M} \rightarrow \mathbb{R}$, solve

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

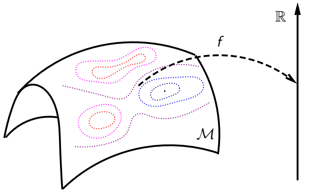
where $\mathcal{M}$ is a Riemannian manifold.



Problem: Given $f(x) : \mathcal{M} \to \mathbb{R}$, solve

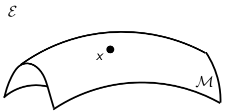
$$\min_{x \in \mathcal{M}} f(x)$$

where $\mathcal{M}$ is a Riemannian manifold.

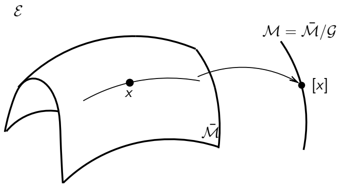


Two kinds of commonly-encountered manifolds

Embedded submanifold of a Euclidean space



Quotient manifold from an embedded submanifold

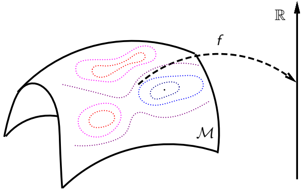


Riemannian Optimization

Problem: Given $f(x) : \mathcal{M} \to \mathbb{R}$, solve

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

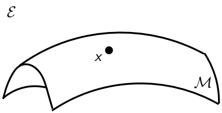
where $\mathcal{M}$ is a Riemannian manifold.



Examples:

- Sphere: $\{x \in \mathbb{R}^n \mid \|x\| = 1\}$;
- Stiefel manifold: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n \times p} \mid X^T X = I_p\}$;
- Fixed rank: $\mathbb{R}_r^{m \times n} = \{X \in \mathbb{R}^{m \times n} : \mathrm{rank}(X) = r\}$;
- etc;

Embedded submanifold of a Euclidean space

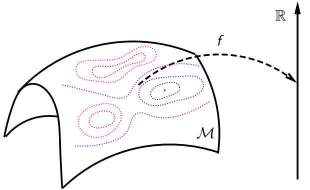


Riemannian Optimization

Problem: Given $f(x) : \mathcal{M} \to \mathbb{R}$, solve

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

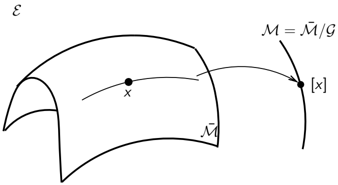
where $\mathcal{M}$ is a Riemannian manifold.



Examples:

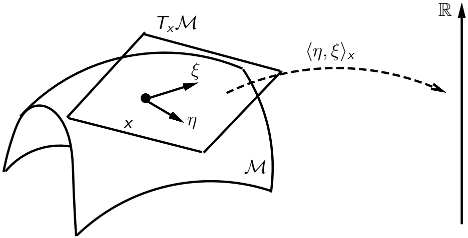
- Grassmann manifold: the set of p dimensional linear spaces in $\mathbb{R}^n$
  $\mathrm{Gr}(p, n) = \mathrm{St}(p, n)/\mathcal{O}_p$;
- Shape space;
- etc;

Quotient manifold from an embedded submanifold



Riemannian Optimization

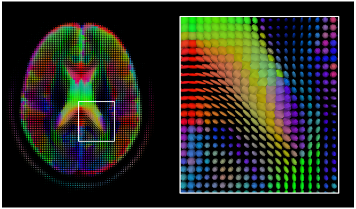
Roughly, a Riemannian manifold $\mathcal{M}$ is a smooth set with a smoothly-varying inner product on the tangent spaces.



Riemannian manifold = Manifold + Riemannian metric (inner products)

Embedded submanifold: Computation on SPD manifold

- SPD manifold:
  $\mathcal{S}_{++}^{n} = \{X \in \mathbb{R}^{n \times n} : X = X^T, X \succ 0\}$;
- Applications of SPD matrices
  - Diffusion tensors in medical imaging [CSV12, FJ07, RTM07]
  - Describing images and video [LWM13, SFD02, ASF$^+$05, TPM06, HWSC15]
- Motivation of averaging SPD matrices
  - denoising / interpolation
  - clustering / classification



Embedded submanifold: Computation on SPD manifold

One averaging SPD matrices method:

$$G(A_1, \ldots, A_k) = \arg \min_{X \in \mathcal{S}_{++}^n} \frac{1}{2k} \sum_{i=1}^{k} \mathrm{dist}^2(X, A_i),$$

where $\mathrm{dist}(X, Y) = \| \log(X^{-1/2} Y X^{-1/2}) \|_F$ is the distance under the Riemannian metric $\langle \eta_X, \xi_X \rangle_X = \mathrm{trace}(\eta_X X^{-1} \xi_X X^{-1})$.

Embedded submanifold: Computation on SPD manifold

One averaging SPD matrices method:

$$G(A_1, \dots, A_k) = \arg \min_{X \in \mathcal{S}^n_{++}} \frac{1}{2k} \sum_{i=1}^{k} \mathrm{dist}^2(X, A_i),$$

where $\mathrm{dist}(X, Y) = \| \log(X^{-1/2} Y X^{-1/2}) \|_F$ is the distance under the Riemannian metric $\langle \eta_X, \xi_X \rangle_X = \mathrm{trace}(\eta_X X^{-1} \xi_X X^{-1})$.

Why shall we use Riemannian optimization approach?

Metric: $\langle \eta_X, \xi_X \rangle_X = \mathrm{trace}(\eta_X X^{-1} \xi_X X^{-1})$

Metric: $\langle \eta, \xi \rangle_X = \mathrm{trace}(\eta^T \xi)$

Condition number of the Riemannian Hessian [YHAG2020]

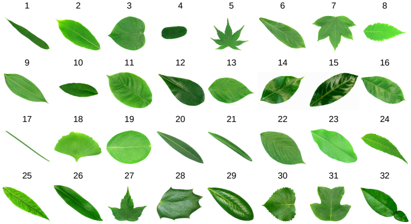
- $\kappa(H^R) \leq 1 + \frac{\ln(\max \kappa_i)}{2}$, where $\kappa_i = \kappa(\mu^{-1/2} A_i \mu^{-1/2})$
- $\kappa(H^R) \leq 20$ if $\max(\kappa_i) = 10^{16}$

- $\frac{\kappa^2(\mu)}{\kappa(H^{\mathrm{R}})} \leq \kappa(H^{\mathrm{E}}) \leq \kappa(H^{\mathrm{R}}) \kappa^2(\mu)$
- $\kappa(H^E) \geq \kappa^2(\mu)/20$

[YHAG2020]: X. Yuan, W. Huang*, P.-A. Absil, K. A. Gallivan. "Computing the matrix geometric mean: Riemannian vs Euclidean conditioning, implementation techniques, and a Riemannian BFGS method", *Numerical Linear Algebra with Applications*, 27:5, 1-23, 2020.

Applications

Quotient manifold: Computation on shape space



- Classification [LKS+12, HGSA15]
- Face recognition [DBS+13]

Quotient manifold: Computation on shape space

- Elastic shape analysis invariants:
  - Rescaling
  - Translation
  - Rotation
  - Reparametrization
- The shape space is a quotient space

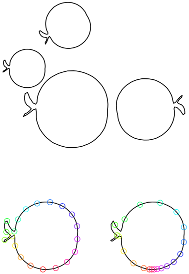
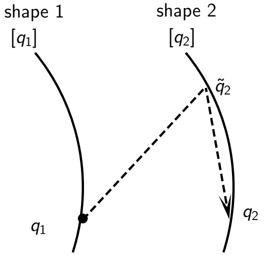
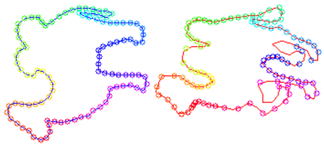


Figure: All are the same shape.

Quotient manifold: Computation on shape space

Registration

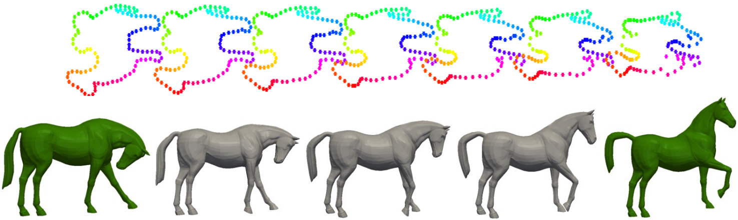




- Optimization problem $\min_{q_2 \in [q_2]} \mathrm{dist}(q_1, q_2)$ is defined on a Riemannian manifold

Applications

Quotient manifold: Computation on shape space
Geodesic / Interpolation

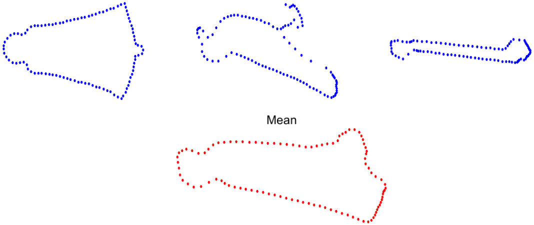


$$\min_{\alpha \in \mathcal{H}_{x,y}} \frac{1}{2} \int_0^1 \langle \dot{\alpha}(\tau), \dot{\alpha}(\tau) \rangle_{\alpha(\tau)} \mathrm{d}\tau$$

- Computation of a geodesic between two shapes
- Interpolation in shape space

Quotient manifold: Computation on shape space

Karcher mean

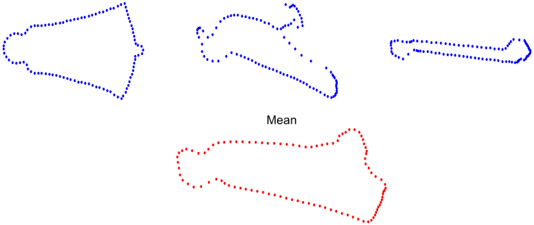


$$\min_{X \text{ is a shape}} \frac{1}{2k} \sum_{i=1}^{k} \mathrm{dist}^2(X, S_i),$$

- Computation of Karcher mean of a population of shapes

Quotient manifold: Computation on shape space

Karcher mean



$$\min_{X \text{ is a shape}} \frac{1}{2k} \sum_{i=1}^{k} \operatorname{dist}^2(X, S_i),$$

- Computation of Karcher mean of a population of shapes

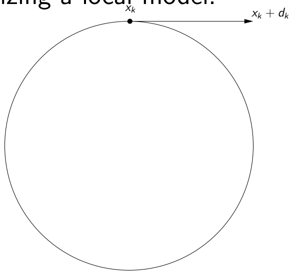
Riemannian optimization is used since these problems naturally involve a Riemannian manifold

Consider the following generic update for an iterative Euclidean optimization algorithm:

$$x_{k+1} = x_k + \Delta x_k = x_k + \alpha_k s_k \ .$$

This iteration is implemented in numerous ways, e.g.:

- Steepest descent: $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$
- Newton's method: $x_{k+1} = x_k - \left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$
- Trust region method: Δx_k is set by optimizing a local model.



Riemannian Manifolds Provide

- Riemannian concepts describing directions and movement on the manifold
- Riemannian analogues for gradient and Hessian

Definition

The Riemannian gradient of f at x is the unique tangent vector in $\mathrm{T}_x \mathcal{M}$ satisfying $\forall \eta \in \mathrm{T}_x \mathcal{M}$, the directional derivative

$$\mathrm{D} f(x)[\eta] = \langle \mathrm{grad} f(x), \eta \rangle$$

and $\mathrm{grad} f(x)$ is the direction of steepest ascent.

Definition

The Riemannian Hessian of f at x is a symmetric linear operator from $\mathrm{T}_x \mathcal{M}$ to $\mathrm{T}_x \mathcal{M}$ defined as

$$\mathrm{Hess} f(x) : \mathrm{T}_x \mathcal{M} \to \mathrm{T}_x \mathcal{M} : \eta \to \nabla_\eta \mathrm{grad} f,$$

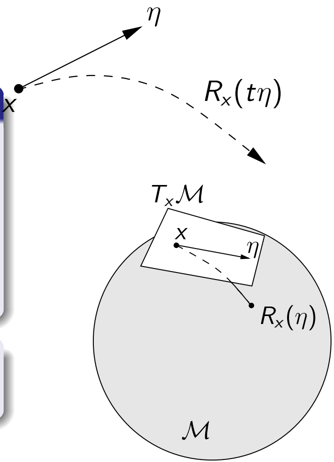
where ∇ is the affine connection.

Euclidean	Riemannian
$x_{k+1} = x_k + \alpha_k d_k$	$x_{k+1} = R_{x_k}(\alpha_k \eta_k)$

Definition

A retraction is a mapping R from $\mathrm{T}\,\mathcal{M}$ to $\mathcal{M}$ satisfying the following:

- R is continuously differentiable
- $R_x(0) = x$
- $\mathrm{D}\,R_x(0)[\eta] = \eta$



- maps tangent vectors back to the manifold
- defines curves in a direction

Retraction-based: local information only

Line search-based: use local tangent vector and $R_x(t\eta)$ to define line

- Steepest decent
- Newton

Local model-based: series of flat space problems

- Riemannian trust region Newton (RTR)
- Riemannian adaptive cubic overestimation (RACO)

Smooth Optimization Framework

Categories of Riemannian smooth optimization methods

Retraction and transport-based: information from multiple tangent spaces

- Nonlinear conjugate gradient: multiple tangent vectors
- Quasi-Newton e.g. Riemannian BFGS: transport operators between tangent spaces

Additional element required for optimizing a cost function;

- formulas for combining information from multiple tangent spaces.

Retraction and transport-based: information from multiple tangent spaces

- Nonlinear conjugate gradient: multiple tangent vectors
- Quasi-Newton e.g. Riemannian BFGS: transport operators between tangent spaces

Additional element required for optimizing a cost function;

- formulas for combining information from multiple tangent spaces.

Vector Transport:

- Vector transport: Transport a tangent vector from one tangent space to another;
- $\mathcal{T}_{\eta_x}\xi_x$, denotes transport of ξ_x to tangent space of $R_x(\eta_x)$. R is a retraction associated with $\mathcal{T}$;

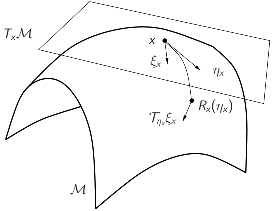


Figure: Vector transport.

Given a retraction and a vector transport, we can generalize classical unconstrained smooth optimization methods from Euclidean space to the Riemannian manifold.

Given a retraction and a vector transport, we can generalize classical unconstrained smooth optimization methods from Euclidean space to the Riemannian manifold.

Do the Riemannian versions of those methods work well?

Given a retraction and a vector transport, we can generalize classical unconstrained smooth optimization methods from Euclidean space to the Riemannian manifold.

Do the Riemannian versions of those methods work well?

No, generally

- Lose many theoretical results and important properties;
- Impose restrictions on retraction/vector transport;

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

- Manifold recognition
- Riemannian metric
- Retraction / Geodesic
- Vector transport / Parallel translation

[EAS1998] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998

[CMV2017] T Carson, D. G. Mixon, and S. Villar. Manifold optimization for k-means clustering. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, 73–77. IEEE, 2017

[SDN2021] G. Song, W. Ding, and M. K. Ng, Low rank pure quaternion approximation for pure quaternion matrices, *SIAM Journal on Matrix Analysis and Applications*, 42, pp. 58–82, 2021

[VAV2013] B. Vandereycken, P.-A. Absil, and S. Vandewalle. A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank, *IMA Journal of Numerical Analysis*, 33.2, 481–514, 2013.

[Zim2017] R. Zimmermann. A matrix-algebraic algorithm for the Riemannian logarithm on the Stiefel manifold under the canonical metric. *SIAM Journal on Matrix Analysis and Applications*, 38.2, 322–342, 2017.

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

- Smooth unconstrained optimization algorithms
- Nonsmooth unconstrained optimization algorithms
- Constrained optimization algorithms

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

- Smooth unconstrained optimization algorithms
- Nonsmooth unconstrained optimization algorithms
- Constrained optimization algorithms

Riemannian optimization mainly focuses on this topic.
Discuss later.

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

- Computations on the SPD manifold;
- Computations on the shape space;
- Clustering and graph partitions;
- Beamforming in wireless communication;
- Blind source separation;
- etc

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

- Representation of a manifold and tangent spaces;
- Choose a Riemannian metric;
- Choose a retraction;
- Choose a vector transport;

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

- Representation of a manifold and tangent spaces;
- Choose a Riemannian metric;
- Choose a retraction;
- Choose a vector transport;

Above factors may influence algorithms significantly.

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

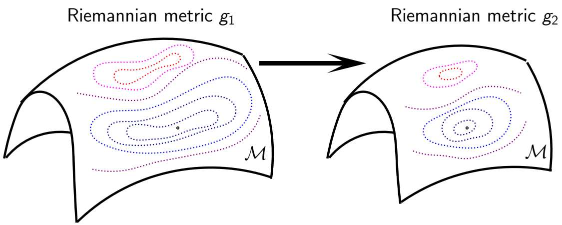


Figure: Changing Riemannian metric may influence the difficulty of a problem.

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

- Manopt (Matlab library) [Boumal, Mishra, Absil, Sepulchre(2014)]
- Pymanopt (Python version of Manopt) [Townsend, Koep, Weichwald (2016)]
- Manoptjl (Julia, nonsmooth methods) [Bergmann (2019)]
- ROPTLIB (C++ library, interfaces to Matlab and Julia) [Huang, Absil, Gallivan, Hand (2018)]
- ManifoldOptim (R wrapper of ROPTLIB) [Martin, Raim, Huang, Adragni (2018)]
- McTorch (Python, GPU acceleration) [Meghawanshi, Jawanpuria, Kunchukuttan, Kasai, Mishra (2018)]
- CDOpt (Python, embedded submanifold in the form of $c(x) = 0$) [Xiao, Hu, Liu, Toh (2022)]

Research Foci of Riemannian Optimization

1. Manifold recognition, geometry structure analyses and computations;
2. Generalization Euclidean algorithms to the Riemannian setting;
3. Algorithms specialization for applications;
4. Library developments;

Provide theories to explain behaviors of existing algorithms for particular applications

- [MBDG2023]: IRKA is a Riemannian gradient descent method;
- [YHAG2020]: Richardson-like iteration for matrix geometric mean is a Riemannian gradient descent method;
- [BM2006]: The improved BFGS method is a Riemannian BFGS method using vector transport by parallelization;

[MBDG2023] P. Mlinaric, C. Beattie, Z. Drmac, and S. Gugercin. IRKA is a Riemannian Gradient Descent Method. arxiv:2311.02031, 2023

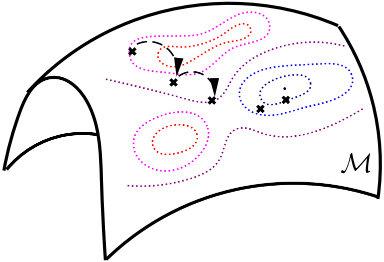
[YHAG2020] X. Yuan, W. Huang, P.-A. Absil, K. A. Gallivan. Computing the matrix geometric mean: Riemannian vs Euclidean conditioning, implementation techniques, and a Riemannian BFGS method, *Numerical Linear Algebra with Applications*, 27:5, 1-23, 2020

[BM2006] I. Brace and J. H. Manton. An improved BFGS-on-manifold algorithm for computing weighted low rank approximations. *Proceedings of 17th international Symposium on Mathematical Theory of Networks and Systems*, P.1735–1738, 2006

Comparison with Constrained Optimization

Not all Riemannian optimization problem can be formulated as constrained optimization problems, and vice versa.

- All iterates on the manifold
- Convergence properties of unconstrained optimization algorithms
- No need to consider Lagrange multipliers or penalty functions
- Exploit the structure of the constrained set



A Non-exhaustive Review

- Smooth unconstrained problems
  - Steepest descent: Smith 1994; Helmke-Moore 1994; Iannazzo-Porcelli 2019;
  - Conjugate gradient: Smith 1994; Gallivan-Absil 2010; Ring-Wirth 2012; Sato-Iwai 2015;
  - Quasi-Newton: Ring-Wirth 2012; Huang-Absil-Gallivan 2018; Huang-Gallivan 2022
  - Newton-CG: Absil-Baker-Gallivan 2007; Huang-Huang 2023
- Nonsmooth unconstrained problems
  - Proximal point method: Ferreira-Oliveira 2002;
  - Optimality conditions: Yang-Zhang-Song 2014;
  - Gradient sampling: Huang 2013; Hosseini and Uschmajew 2017;
  - ϵ-subgradient-based methods: Grohs-Hosseini 2015;
  - Proximal gradient methods: Huang-Wei 2022;
  - Proximal Newton method: Si-Absil-Huang-Jiang-Vary 2023;
- Constrained problems:
  - Augmented Lagrangian methods: Boumal-Liu 2019;
  - Sequential quadratic programming: Obara-Okuno-Takeda 2022;
  - Frank-Wolfe Methods: Weber-Sra 2023;

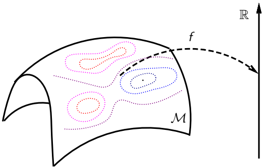
A Non-exhaustive Review

- Smooth unconstrained problems:
  - Stiefel manifold: Wen-Yin 2012; Jiang-Dai 2014; Xiao-Liu-Yuan 2020; Dai-Wang-Zhou 2020
  - Symplectic Stiefel manifold: Gao-Son-Absil-Stykel 2021
  - Symmetric positive definite manifold: Bini-Iannazzo 2013; Zhang 2017; Yuan-Huang-Absil-Gallivan 2020;
  - Fixed rank manifold: Wen-Yin-Zhang 2012; Mishra 2014; Sutti-Vandereycken 2021; Levin-Kileel-Boumal 2022
- Nonsmooth unconstrained problems:
  - Stiefel Manifold: Huang-Wei 2019; Chen-Ma-So-Zhang 2020; Xiao-Liu-Yuan 2020;
  - Fixed rank manifold: Cambier-Absil 2016;
  - Matrix manifolds: Zhou-Bao-Ding-Zhu 2022
  - Smooth equation constraints: Xiao-Liu-Toh 2023
- Constrained problems:
  - Stiefel + non-negativity: Jiang-Meng-Wen-Chen 2019;
  - Symmetric positive definite + zeros: Phan-Menickelly 2020;

A Riemannian Proximal Newton Method

Optimization on Manifolds with Structure:

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

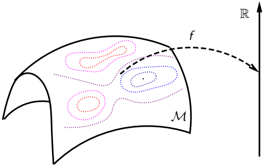


- $\mathcal{M}$ is a finite-dimensional Riemannian manifold;
- f is smooth and may be nonconvex; and
- $h(x)$ is continuous and convex but may be nonsmooth;

A Riemannian Proximal Newton Method

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].

Euclidean Proximal Gradient/Newton Method

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

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

Given x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k. \end{cases}$$

Euclidean Proximal Gradient/Newton Method

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

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

Given x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2}\langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k. \end{cases}$$

proximal gradient: $H_k = L I_n$

- $h \equiv 0 \Rightarrow$ Steepest descent;
- Linear convergence;

proximal Newton: $H_k = \nabla^2 f(x_k)$

- $h \equiv 0 \Rightarrow$ Newton;
- Superlinear convergence;

Euclidean Proximal Gradient/Newton Method

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

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

Given x_0,

$$\begin{cases} d_k = \arg\min_{p \in \mathbb{R}^n} f(x_k) + \langle \nabla f(x_k), p \rangle + \frac{1}{2} \langle p, H_k p \rangle + h(x_k + p) \\ x_{k+1} = x_k + d_k. \end{cases}$$

proximal gradient: $H_k = L I_n$

- $h \equiv 0 \Rightarrow$ Steepest descent;
- Linear convergence;

proximal Newton: $H_k = \nabla^2 f(x_k)$

- $h \equiv 0 \Rightarrow$ Newton;
- Superlinear convergence;

How to generalize to the Riemannian setting?

Generalizations of Proximal Gradient Method

Euclidean Proximal gradient:

Given x_0,

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

Riemannian generalization 1: (for embedded submanifold)

$$\left. \begin{array}{c} \nabla f(x_k) \Longrightarrow \operatorname{grad} f(x_k) \\ x_{k+1} = x_k + d_k \Longrightarrow x_{k+1} = R_{x_k}(d_k) \\ p \in \mathbb{R}^n \Longrightarrow p \in \mathrm{T}_{x_k} \mathcal{M} \end{array} \right\} \Longrightarrow \text{ Converge globally}$$

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

Generalizations of Proximal Gradient Method

Euclidean Proximal gradient:

Given x_0,

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

Riemannian generalization 2: (for general manifold)

$$\left.\begin{array}{c} \nabla f(x_k) \Longrightarrow \mathrm{grad}\, f(x_k) \\ x_{k+1} = x_k + d_k \Longrightarrow x_{k+1} = R_{x_k}(d_k) \\ p \in \mathbb{R}^n \Longrightarrow p \in \mathrm{T}_{x_k}\mathcal{M} \\ h(x_k + p) \Longrightarrow h(R_{x_k}(p)) \end{array}\right\} \Longrightarrow \begin{array}{l} \text{Converge globally} \\ \text{Convergence rate analyses} \end{array}$$

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

Euclidean proximal Newton:

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

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

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

Euclidean proximal Newton:

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

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

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

Does it converge superlinearly locally?

Euclidean proximal Newton:

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

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

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

Does it converge superlinearly locally?

Not necessarily!

Consider the Sparse PCA over sphere:

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

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

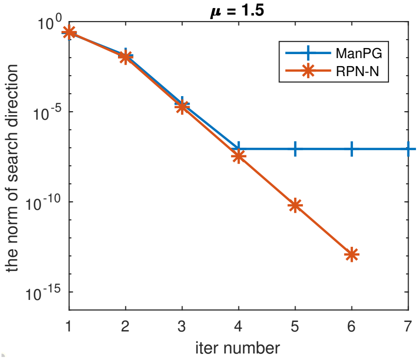


Figure: Comparisons of native generalization (RPN-N) and the proximal gradient method (ManPG) in [CMSZ20].

Euclidean version:

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

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

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

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

Euclidean version:

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

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

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

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

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

A Riemannian proximal Newton method (RPN)

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

A Riemannian proximal Newton method (RPN)

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

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

A Riemannian proximal Newton method (RPN)

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

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

A Riemannian proximal Newton method (RPN)

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

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

A Riemannian proximal Newton method (RPN)

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

Next, we will show:

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

Definition (G-Semismoothness [Gow04])

Let $F : \mathcal{D} \rightarrow \mathbb{R}^m$ 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 $J \in \mathcal{K}(x + d)$,

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

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

The standard definition of semismoothness additional requires:

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

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

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

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

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

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

Above problem can be rewritten as

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

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

The Lagrangian function:

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

Therefore

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

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

Define

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

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

Define

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

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

[Gow04] M Seetharama Gowda. Inverse and implicit function theorems for h-differentiable and semismooth functions. Optimization Methods and Software, 19(5):443-461, 2004.
[PSS03] Jong-Shi Pang, Defeng Sun, and Jie Sun. Semismo oth homeomorphisms and strong stability of semidefinite and Lorentz complementarity problems. Mathematics of Operations Research, 28(1):39-63, 2003.

Lemma (Semismooth Implicit Function Theorem)

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

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

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

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

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

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

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

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

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

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_*.

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_*.

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

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

- KKT conditions:

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

- Closed form solutions:

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

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

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

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

- 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

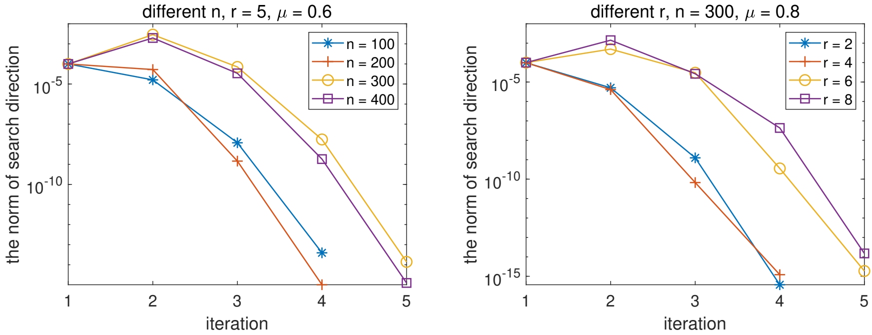


Figure: Random data. Left: different $n = \{100, 200, 300, 400\}$ with $r = 5$ and $\mu = 0.6$; Right: different $r = \{2, 4, 6, 8\}$ with $n = 300$ and $\mu = 0.8$

Collaborators

- Wutao Si, Xiamen University
- Pierre-Antoine Absil, Université catholique de Louvain
- Wen Huang, Xiamen University
- Rujun Jiang, Fudan University
- Simon Vary, Université catholique de Louvain

W. Si, P.-A. Absil, W. Huang*, R. Jiang, and S. Vary,
A Riemannian Proximal Newton Method, Accepted in *SIAM Journal on Optimization*.

Summary

- Riemannian optimization;
- Applications;
  - An example on an embedded submanifold;
  - An example on a quotient manifold;
- Smooth optimization framework;
  - Search direction/Riemannian metric;
  - Riemannian gradient/Hessian;
  - Retraction/vector transport;
- Research foci of Riemannian optimization;
  - Manifold recognition/structures;
  - Algorithm generalizations;
  - Applications/Libraries;
- A Riemannian proximal Newton method;
  - Naive generalization;
  - Superlinear convergence approach;
- Summary;

References I

Ognjen Arandjelovic, Gregory Shakhnarovich, John Fisher, Roberto Cipolla, and Trevor Darrell.
Face recognition with image sets using manifold density divergence.
In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 581–588. IEEE, 2005.

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.

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.

Guang Cheng, Hesamoddin Salehian, and Baba Vemuri.
Efficient recursive algorithms for computing the mean diffusion tensor and applications to DTI segmentation.
Computer Vision–ECCV 2012, pages 390–401, 2012.

H. Drira, B. Ben Amor, A. Srivastava, M. Daoudi, and R. Slama.
3D face recognition under expressions, occlusions, and pose variations.
Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(9):2270–2283, 2013.

P. T. Fletcher and S. Joshi.
Riemannian geometry for the statistical analysis of diffusion tensor data.
Signal Processing, 87(2):250–262, 2007.

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

References II

W. Huang, K. A. Gallivan, Anuj Srivastava, and P.-A. Absil.
Riemannian optimization for registration of curves in elastic shape analysis.
Journal of Mathematical Imaging and Vision, 54(3):320–343, 2015.
DOI:10.1007/s10851-015-0606-8.

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

Zhiwu Huang, Ruiping Wang, Shiguang Shan, and Xilin Chen.
Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning.
Pattern Recognition, 48(10):3113–3124, 2015.

H. Laga, S. Kurtek, A. Srivastava, M. Golzarian, and S. J. Miklavcic.
A Riemannian elastic metric for shape-based plant leaf classification.
2012 International Conference on Digital Image Computing Techniques and Applications (DICTA), pages 1–7, December 2012.
doi:10.1109/DICTA.2012.6411702.

Jiwen Lu, Gang Wang, and Pierre Moulin.
Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning.
In *Proceedings of the IEEE International Conference on Computer Vision*, pages 329–336, 2013.

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.

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.

Y. Rathi, A. Tannenbaum, and O. Michailovich.
Segmenting images on the tensor manifold.
In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

References III

Gregory Shakhnarovich, John W Fisher, and Trevor Darrell.
Face recognition from long-term observations.
In *European Conference on Computer Vision*, pages 851–865. Springer, 2002.

Oncel Tuzel, Fatih Porikli, and Peter Meer.
Region covariance: A fast descriptor for detection and classification.
In *European conference on computer vision*, pages 589–600. Springer, 2006.

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.

Thank you

Thank you!