Introduction to Riemannian BFGS Methods

Speaker: Wen Huang

Rice University

May 23, 2017

Collaborators

Prof. Pierre-Antoine Absil
Université catholique de Louvain

Prof. Kyle A. Gallivan
Florida State University

Prof. Paul Hand
Rice University

Xinru Yuan
Florida State University

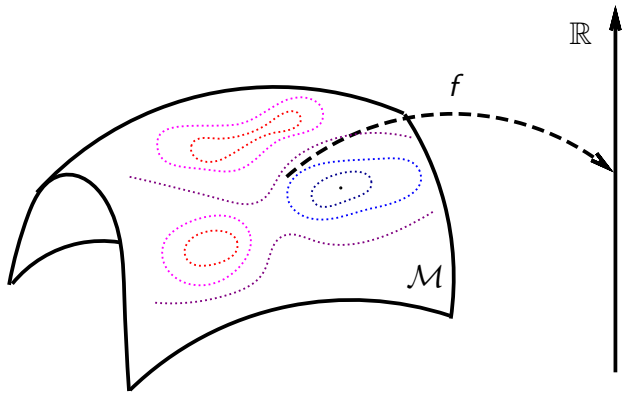
- What problems does Riemannian optimization consider?
- What does a Riemannian optimization algorithm look like?
- How do Riemannian optimization methods perform?

Problem Statement

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

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

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

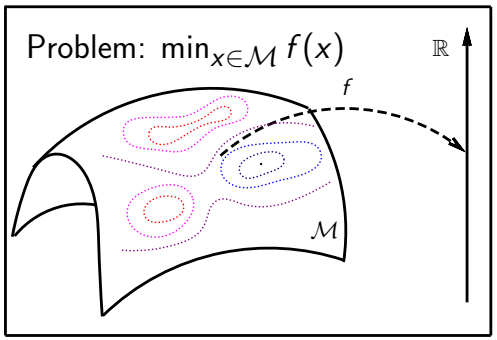


Unconstrained optimization problem on a constrained space.

What is a Riemannian manifold?

$$\text{Riemannian manifold} = \text{manifold} + \text{Riemannian metric}$$

Manifolds



Two kinds of manifolds:

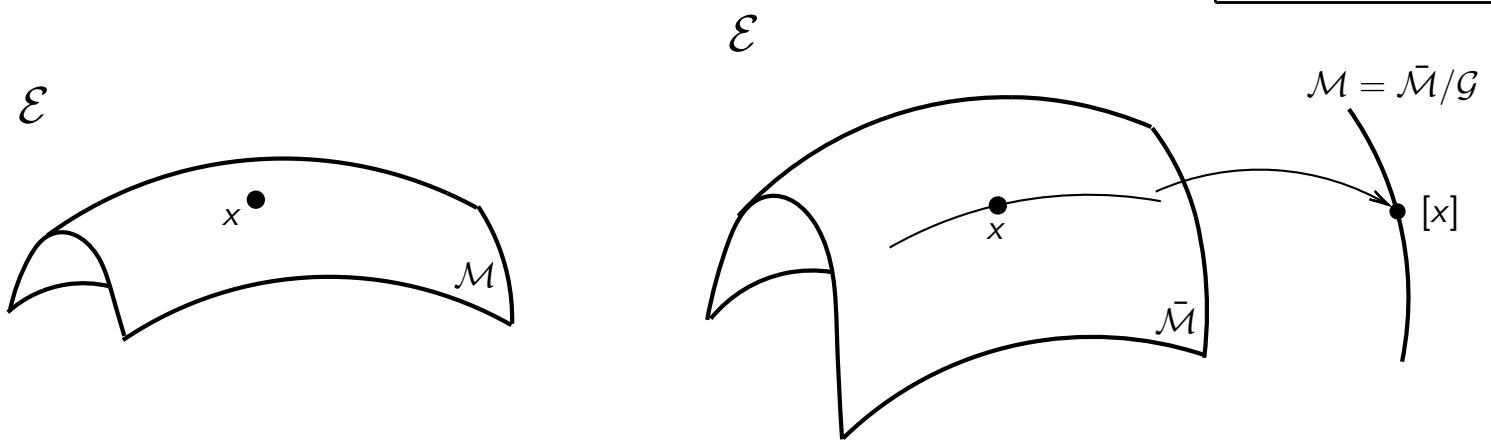
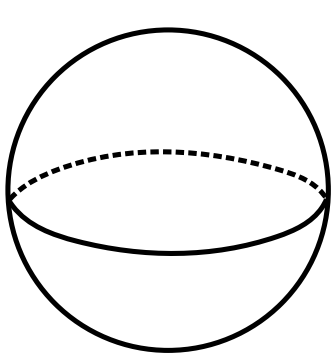
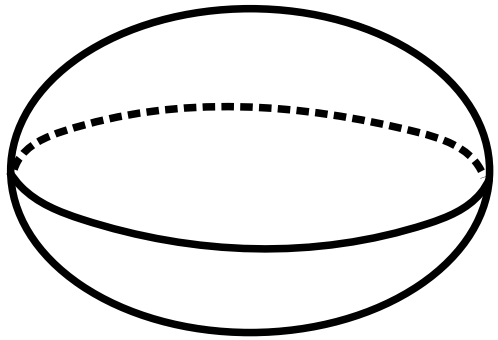


Figure: Left: an embedded submanifold; right: a quotient manifold

Manifolds: Examples



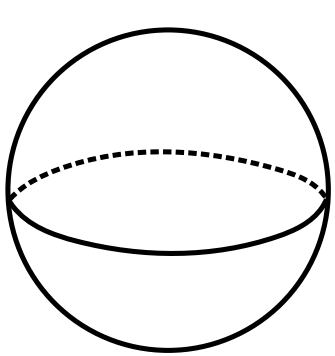
Sphere



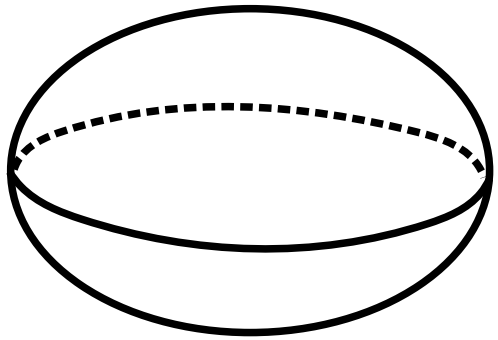
Ellipsoid

- Stiefel manifold: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n \times p} | X^T X = I_p\}$;
- Grassmann manifold $\mathrm{Gr}(p, n)$: all p-dimensional subspaces of $\mathbb{R}^n$;
- All r-by-r symmetric positive definite matrices, $\mathrm{S}_{++}(r)$;
- All rank-r m-by-n matrices $\mathbb{R}_r^{n \times m}$;
- And many more.

Manifolds: Examples



Sphere



Ellipsoid

- Stiefel manifold: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n\times p} | X^T X = I_p\}$;
- Grassmann manifold $\mathrm{Gr}(p, n)$: all p-dimensional subspaces of $\mathbb{R}^n$;
- All r-by-r symmetric positive definite matrices, $\mathrm{S}_{++}(r)$;
- All rank-r m-by-n matrices $\mathbb{R}_r^{n\times m}$;
- And many more.

A manifold may have multiple representations.

Manifolds: Representations

A manifold may have multiple representations. Representations influence complexities of optimization algorithms.

An important question:

How to represent points on a manifold?

1

Manifolds: Representations

A manifold may have multiple representations. Representations influence complexities of optimization algorithms.

An important question:

How to represent points on a manifold?

The fixed rank manifold: $\mathbb{R}_r^{n\times m} = \{X \in \mathbb{R}^{n\times m} \mid \mathrm{rank}(X) = r\}$[1]

Embedded manifold

- $\mathbb{R}_r^{n\times m}$: submanifold of $\mathbb{R}^{n\times m}$

Quotient manifold

- $\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r}$, where the star $*$ means full rank
- $\mathrm{St}(r,n) \times \mathrm{S}_{++}(r) \times \mathrm{St}(r,m)/\mathrm{St}(r,r)$;
- $\mathrm{St}(r,n) \times \mathbb{R}_*^{m\times r}/\mathrm{St}(r,r)$;

[1] See details in A Riemannian approach for large-scale constrained least-squares with symmetries, by B. Mishra, Ph.D thesis, 2014.

Manifolds: The Fixed-rank Manifold

Embedded manifold

- $\mathbb{R}_r^{n\times m}$

Quotient manifold

- $\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r}$;

Definition of $\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r}$

$\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r} = \{[(H, M)] \mid H \in \mathbb{R}_*^{n\times r}, M \in \mathbb{R}_*^{m\times r}\}$, where $[(H, M)] = \{(HP^{-1}, MP^T) \mid P \in \mathbb{R}_*^{r\times r}\}$.

- $\mathcal{I} : \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r} \to \mathbb{R}_r^{n\times m}$: $[(H, M)] = HM^T$; $(X = HM^T)$

Manifolds: The Fixed-rank Manifold

Embedded manifold

- $\mathbb{R}_r^{n\times m}$

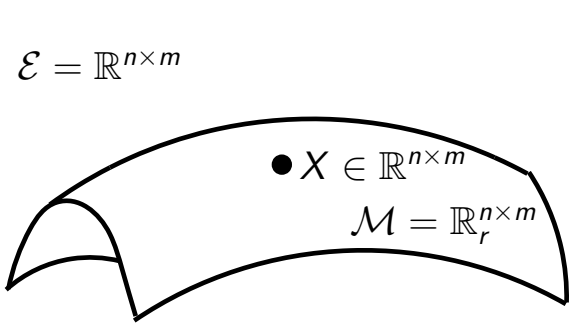
Quotient manifold

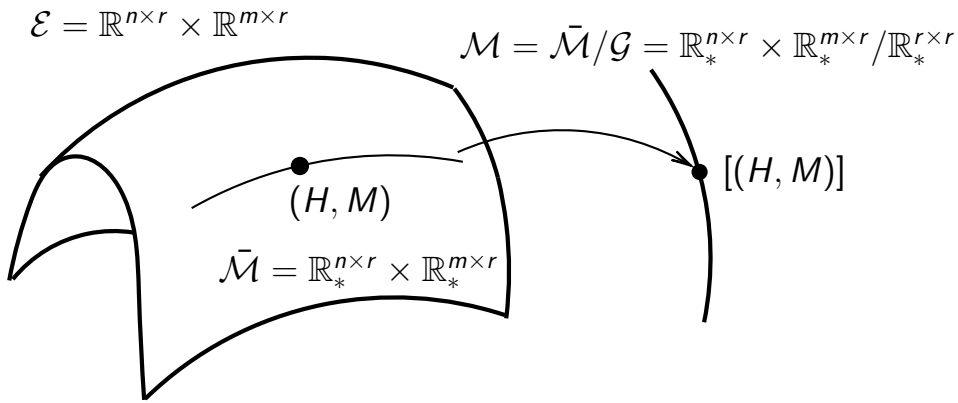
- $\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r}$;

Definition of $\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r}$

$\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r} = \{[(H, M)] \mid H \in \mathbb{R}_*^{n\times r}, M \in \mathbb{R}_*^{m\times r}\}$, where $[(H, M)] = \{(HP^{-1}, MP^T) \mid P \in \mathbb{R}_*^{r\times r}\}$.

- $\mathcal{I} : \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r} \to \mathbb{R}_r^{n\times m}$: $[(H, M)] = HM^T$; $(X = HM^T)$

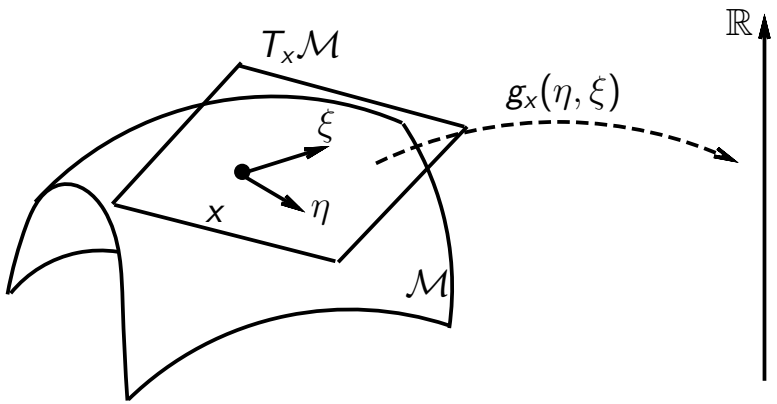




- $\mathbb{R}_r^{n\times m}$: naturally use $\mathbb{R}^{n\times m}$ (inefficient)
- $\mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}/\mathbb{R}_*^{r\times r}$: naturally use $\mathbb{R}^{n\times r} \times \mathbb{R}^{m\times r}$ (efficient)

Riemannian Metric

- A Riemannian metric, denoted by g, is a smoothly-varying inner product on the tangent spaces;
- A Riemannnian metric defines angles and lengths;



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

Riemannian Metric

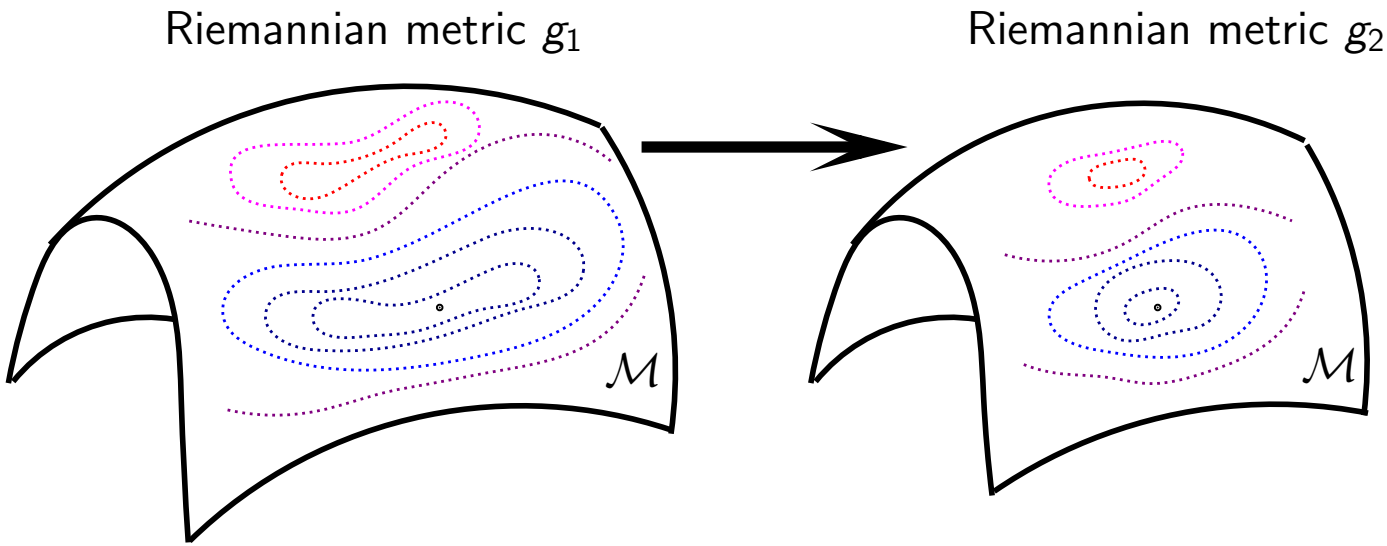


Figure: Changing metric may influence the difficulty of a problem.

Riemannian metric influences

- Riemannian gradient
- Riemannian Hessian

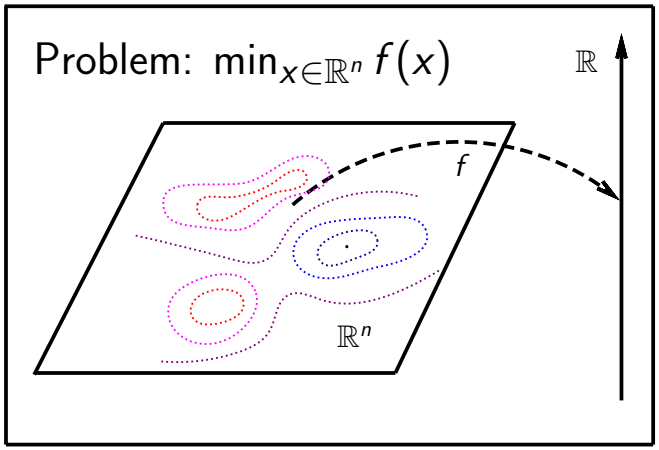
- What problems does Riemannian optimization consider?
- What does a Riemannian optimization algorithm look like?
- How do Riemannian optimization methods perform?

Line Search-based Methods

Euclidean Optimization:

- Update:
  $x_{k+1} = x_k + \alpha_k d_k \ = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$;
- Steepest descent: $B_k = \mathrm{id}$;
- Newton's method: $B_k = \nabla^2 f(x_k)$.

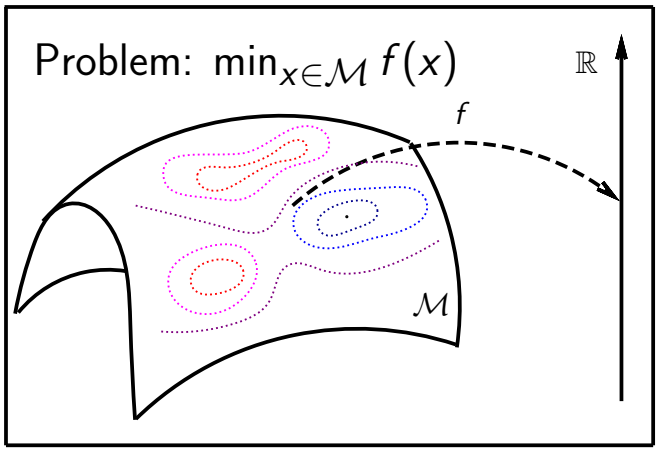
Euclidean



Riemannian Optimization:

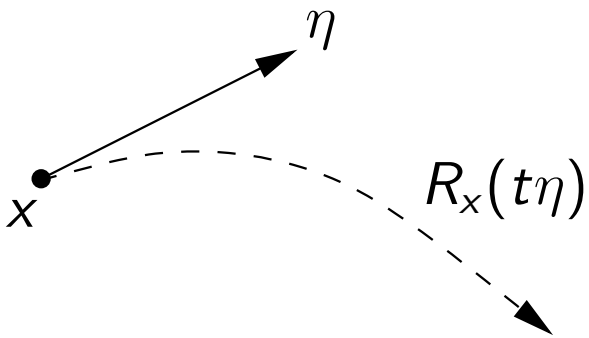
- Riemannian gradient
- Riemannian Hessian
- How to update?

Riemannian



Retractions

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

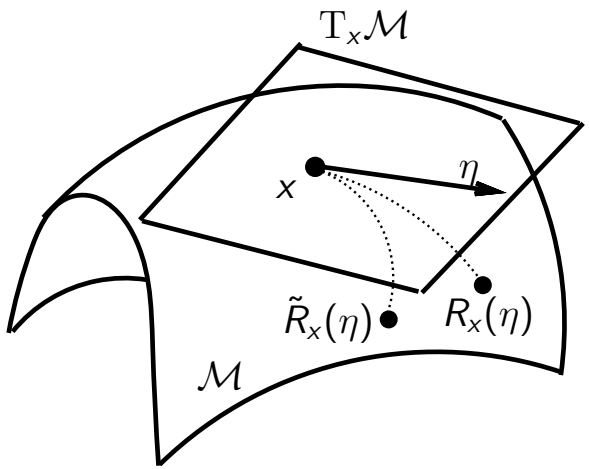


Definition

A retraction is a mapping R from TM to M satisfying the following:

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

- Retraction influences convergence speed



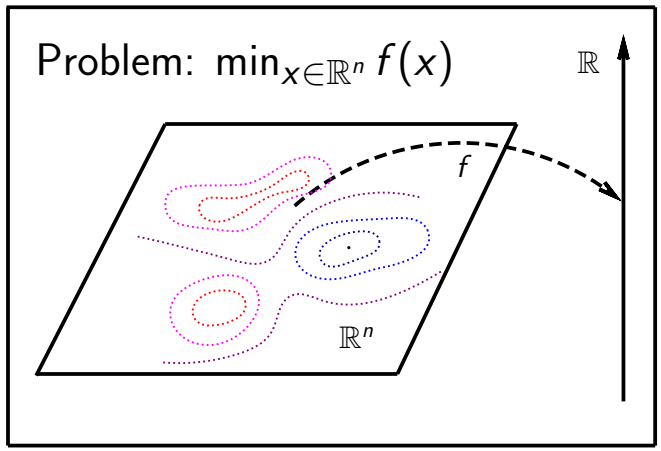
Two retractions: R and $\tilde{R}$

Line Search-based Methods

Euclidean Optimization:

1. Given $x_0 \in \mathbb{R}^n$, $k = 0$;
2. Repeat: $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$ for some α_k and $\mathcal{B}_k$;
3. $k \leftarrow k + 1$ and goto 2;

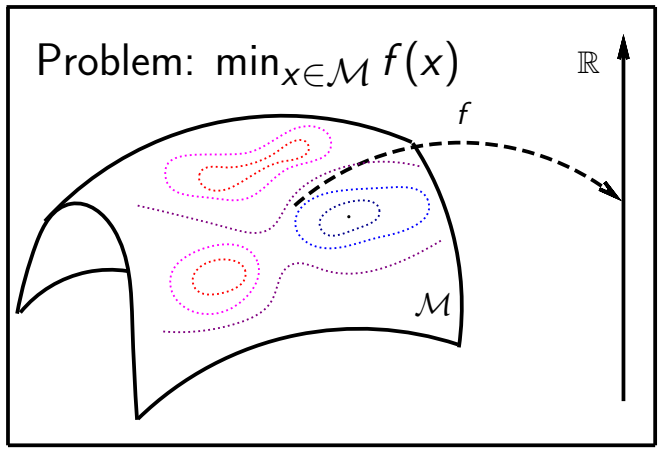
Euclidean



Riemannian Optimization:

1. Given $x_0 \in \mathcal{M}$, $k = 0$;
2. Repeat: $x_{k+1} = R_{x_k}\left(-\alpha_k \mathcal{B}_k^{-1} \operatorname{grad} f(x_k)\right)$ for some α_k and $\mathcal{B}_k$;
3. $k \leftarrow k + 1$ and goto 2;

Riemannian



BFGS Methods

Euclidean BFGS method:

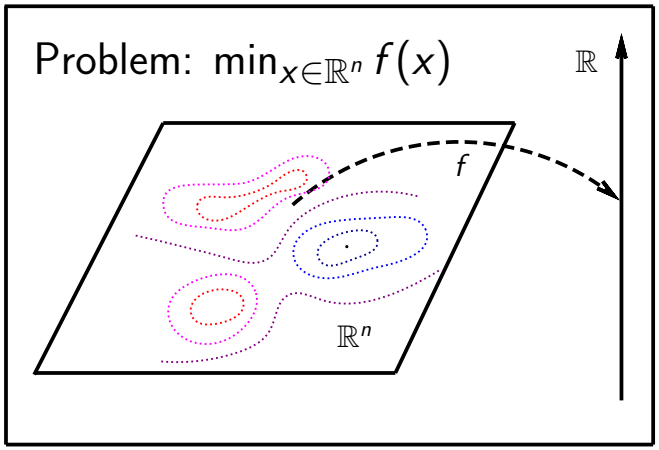
1. Given $x_0 \in \mathbb{R}^n$ and B_0, $k = 0$;
2. Repeat: $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k)$ for some α_k and $\mathcal{B}_k$;
3. Compute B_{k+1} by (1)
4. $k \leftarrow k + 1$ and goto 2;

Euclidean BFGS update

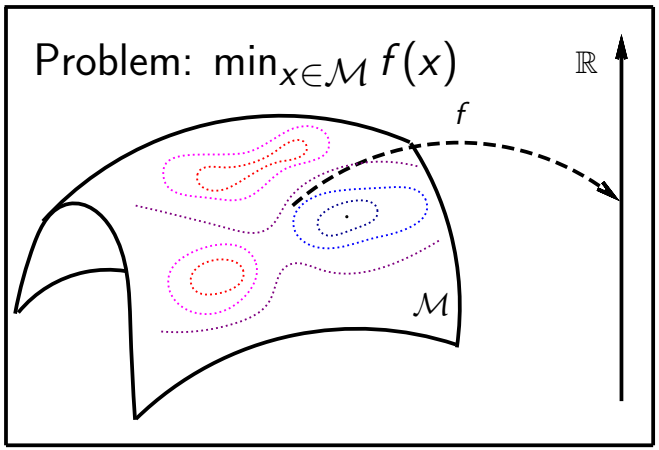
$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \qquad (1)$$

where $s_k = x_{k+1} - x_k$, and
$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

Euclidean



Riemannian

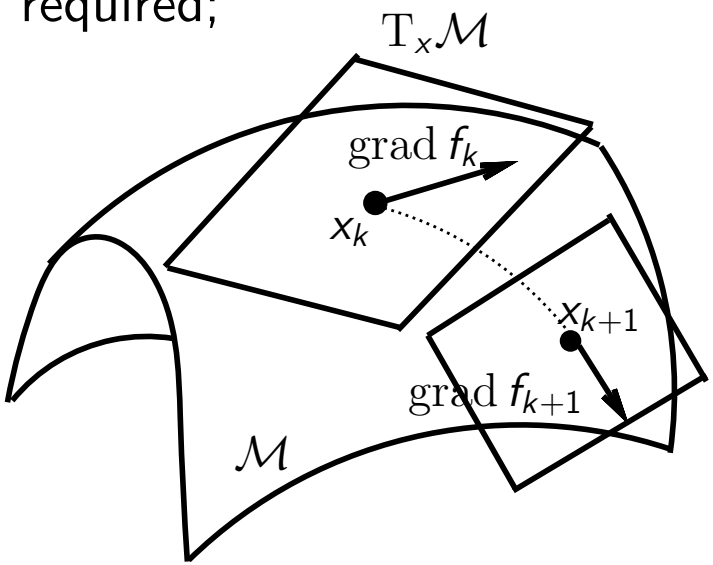


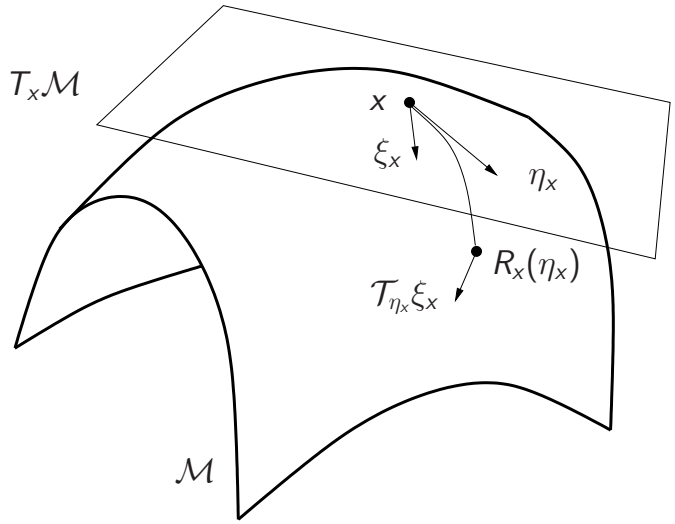
BFGS updates

$$\text{Euclidean: } B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k},$$

$$s_k = x_{k+1} - x_k, y_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

- Explore information in previous iterates to accelerate algorithm;
- A vector transport $\mathcal{T} : \mathrm{T}\mathcal{M} \times \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}\xi_x$ is required;





BFGS updates

A Riemannian BFGS update:

$$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k \mathfrak{s}_k (\tilde{\mathcal{B}}_k \mathfrak{s}_k)^\flat}{\mathfrak{s}_k^\flat \tilde{\mathcal{B}}_k \mathfrak{s}_k} + \frac{\mathfrak{y}_k \mathfrak{y}_k^\flat}{\mathfrak{y}_k^\flat \mathfrak{s}_k},$$

where $\tilde{\mathcal{B}}_k = \mathcal{T}_{\xi_k} \circ \mathcal{B}_k \circ \mathcal{T}_{\xi_k}^{-1}$, $\mathfrak{y}_k = \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\xi_k} \operatorname{grad} f(x_k)$, $\mathfrak{s}_k = \mathcal{T}_{\xi_k} \xi_k$, $\xi_k = R_{x_k}^{-1}(x_{k+1})$

- Above Riemannian BFGS method does not work in general;
- What fails? In the Euclidean setting,

  $B_k \succcurlyeq 0$

  $\Rightarrow$

  search direction $d_k = -B_k \nabla f(x_k)$ is descent

  $\Rightarrow$

  line search with Wolfe conditions can be done

  $\Rightarrow$

  $y_k^T s_k > 0$

  $\Rightarrow$

  $B_{k+1} \succcurlyeq 0$.

BFGS updates

A Riemannian BFGS update:

$$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k \mathfrak{s}_k (\tilde{\mathcal{B}}_k \mathfrak{s}_k)^\flat}{\mathfrak{s}_k^\flat \tilde{\mathcal{B}}_k \mathfrak{s}_k} + \frac{\mathfrak{y}_k \mathfrak{y}_k^\flat}{\mathfrak{y}_k^\flat \mathfrak{s}_k},$$

where $\tilde{\mathcal{B}}_k = \mathcal{T}_{\xi_k} \circ \mathcal{B}_k \circ \mathcal{T}_{\xi_k}^{-1}$, $\mathfrak{y}_k = \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\xi_k} \operatorname{grad} f(x_k)$, $\mathfrak{s}_k = \mathcal{T}_{\xi_k} \xi_k$, $\xi_k = R_{x_k}^{-1}(x_{k+1})$

- Above Riemannian BFGS method does not work in general;
- What fails? In the Euclidean setting,
  $B_k \succcurlyeq 0$
  $\Rightarrow$
  search direction $d_k = -B_k \nabla f(x_k)$ is descent
  $\Rightarrow$
  line search with Wolfe conditions can be done
  $\Rightarrow$ not true in the Riemannian setting
  $y_k^T s_k > 0$
  $\Rightarrow$
  $B_{k+1} \succcurlyeq 0$.

BFGS updates

A Riemannian BFGS update:

$$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k \mathfrak{s}_k (\tilde{\mathcal{B}}_k \mathfrak{s}_k)^\flat}{\mathfrak{s}_k^\flat \tilde{\mathcal{B}}_k \mathfrak{s}_k} + \frac{\mathfrak{y}_k \mathfrak{y}_k^\flat}{\mathfrak{y}_k^\flat \mathfrak{s}_k},$$

where $\tilde{\mathcal{B}}_k = \mathcal{T}_{\xi_k} \circ \mathcal{B}_k \circ \mathcal{T}_{\xi_k}^{-1}$, $\mathfrak{y}_k = \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\xi_k} \operatorname{grad} f(x_k)$, $\mathfrak{s}_k = \mathcal{T}_{\xi_k} \xi_k$, $\xi_k = R_{x_k}^{-1}(x_{k+1})$

- Above Riemannian BFGS method does not work in general;
- Restrictions on retraction and vector transport;
  - Qi [Qi11]: exponential mapping and parallel translation
  - Ring and Wirth [RW12]: differentiated retraction and an isometric vector transport
  - Huang et. al. [HGA15]: differentiated retraction along a direction and an isometric vector transport
  - Huang et. al. [HAG17]: an isometric vector transport
- Complete convergence analysis exists for above Riemannian BFGS

Complexities of Vector Transport

Problems of computing $\tilde{\mathcal{B}}_k = \mathcal{T}_{\xi_k} \circ \mathcal{B}_k \circ \mathcal{T}_{\xi_k}^{-1}$.

- Explicit form of $\mathcal{T}$ may not exist;
- Maybe too expensive: matrix multiplications or matrix-vector multiplications

Intrinsic representation of tangent vectors and vector transport by parallelization [HAG16]

- Represent a tangent vector by $\mathbb{R}^d$, where d is the dimension of the manifold
- Vector transport by parallelization reduces to an identity (efficient)

ROPTLIB

A C++ library is available at `www.math.fsu.edu/~whuang2/ROPTLIB`

- BLAS and LAPACK;
- Interfaces with R, Matlab, and Julia;
- Windows, Linux, Mac

Included Methods:

- Line-search: RBB, RBFGS, LRBFGS, RCG, RNewton;
- Trust-region: RTRSR1, RTRNewton, LRTRSR1;

Included Manifolds:

- Euclidean space;
- Stiefel manifold, Orthogonal group, Sphere in $\mathbb{R}^n$, Sphere in $\mathbb{L}^2$;
- Grassmann manifold;
- Elastic shape space;
- Manifold of symmetric positive semidefinite matrices with rank fixed;
- The fixed-rank manifold;
- Any power or products of above manifolds;

- What problems does Riemannian optimization consider?
- What does a Riemannian optimization algorithm look like?
- How do Riemannian optimization methods perform?

Experiments

- Karcher mean on symmetric positive definite (SPD) manifold
- The blind deconvolution problem

Karcher mean on SPD manifold

$$\text{Problem:} \quad \min_{X \in \mathrm{S}_{++}(n)} F(X) = \frac{1}{2K} \sum_{i=1}^{K} \| \log(A_i^{-1/2} X A_i^{-1/2}) \|_F^2$$

- $A_i, i = 1, \ldots, K$ are given SPD matrices;
- Domain is the SPD manifold $\mathrm{S}_{++}(n)$;
- Geodesic distance under the affine invariant metric

$$\text{Distance:} \quad \mathrm{dist}(X, Y) = \| \log(Y^{-1/2} X Y^{-1/2}) \|_F$$
$$\text{Metric:} \quad g_X(\xi_X, \eta_X) = \mathrm{trace}(\xi_X X^{-1} \eta_X X^{-1})$$

- Geodesic convex and a unique minimizer;

Karcher mean on SPD manifold: Metrics

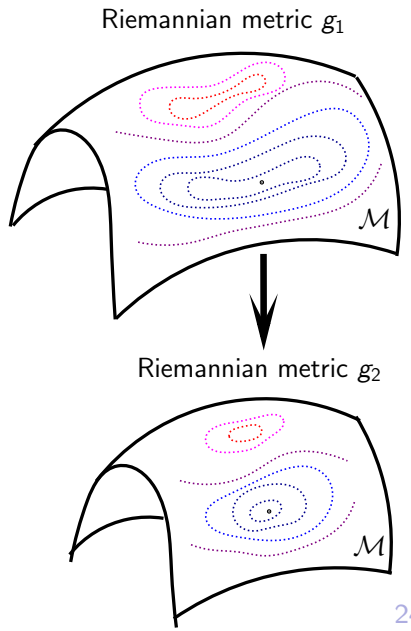
$$\text{Problem: } \min_{X \in \mathrm{S}_{++}(n)} F(X) = \frac{1}{2K} \sum_{i=1}^{K} \| \log(A_i^{-1/2} X A_i^{-1/2}) \|_F^2$$

- Euclidean metric:
  $g_X(\xi_X, \eta_X) = \mathrm{trace}(\xi_X^T \eta_X)$;
- Affine invariant metric:
  $g_X(\xi_X, \eta_X) = \mathrm{trace}(\xi_X X^{-1} \eta_X X^{-1})$;

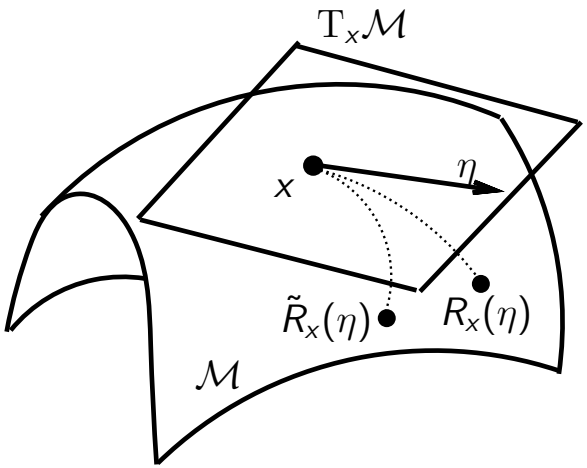
Condition number κ of Hessian at the minimizer:

- Hessian of Euclidean metric
  - $\kappa \approx$ square of the condition number of the minimizer
- Hessian of affine invariant metric
  - $\kappa \leq 20$ in double precision theoretically;
  - $\kappa \leq 4$ usually in our experiments;

Riemannian metric is important.



Karcher mean on SPD manifold: Retractions



Two retractions: R and $\tilde{R}$

- $R_X^{(1)}(\eta_X) = X + \eta_X$;
  - Straight line;
  - Not preserve positive definiteness;
- $R_X^{(2)}(\eta_X) = X + \eta_X + \frac{1}{2}\eta_X X^{-1}\eta_X$;
  - Retracted curve is a second order approximation of the geodesic;
  - Preserve positive definiteness;

Karcher mean on SPD manifold: Methods

- $R_X^{(1)}(\eta_X) = X + \eta_X$;
- $R_X^{(2)}(\eta_X) = X + \eta_X + \frac{1}{2}\eta_X X^{-1}\eta_X$;

Compared methods:[2]

- Richardson-like (RL) iteration [BI13]: Riemannian metric + $R_X^{(1)}$ + carefully-chosen stepsize;
- Riemannian BB method: $R_X^{(2)}$
- Riemannian BFGS method: $R_X^{(2)}$
- Limited-memory RBFGS method: $R_X^{(2)}$

[2] See detailed implementation of Riemannian methods in [YHAG17].

Results

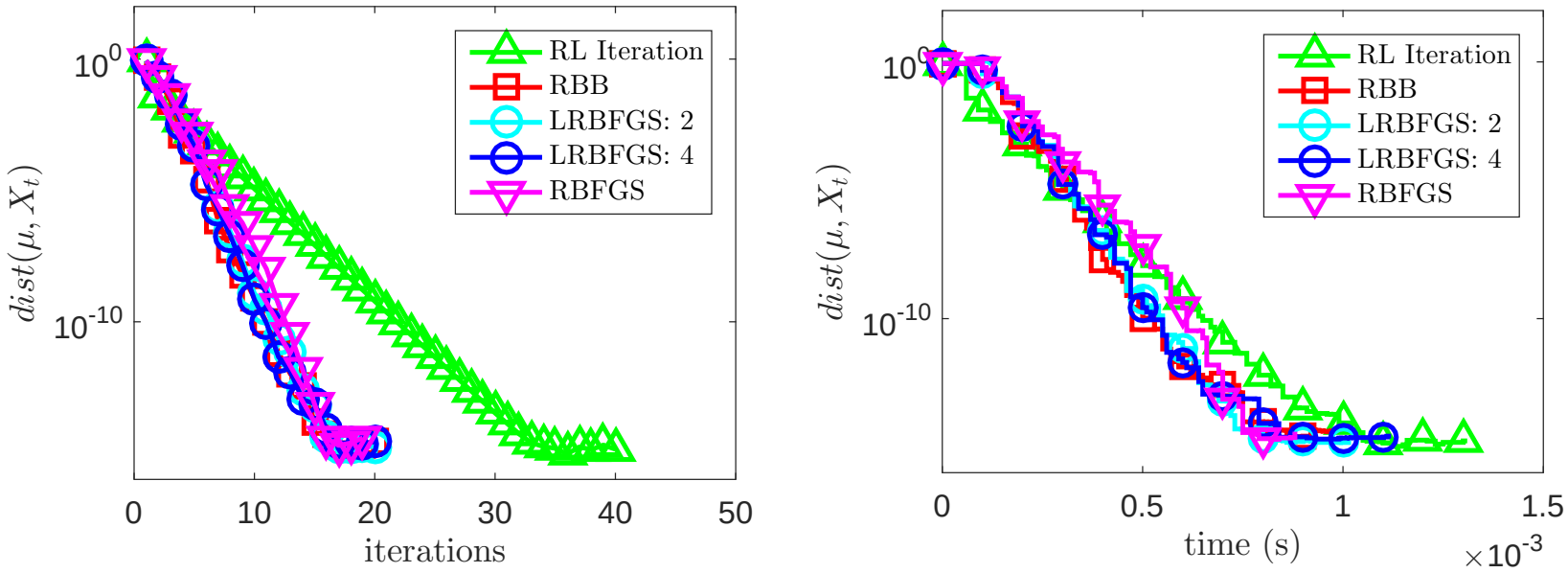


Figure: Evolution of averaged distance between current iterate and the exact Karcher mean with respect to time and iterations with $K = 3$ and $n = 3$; $1 \leq \kappa(A_i) \leq 20$

Results

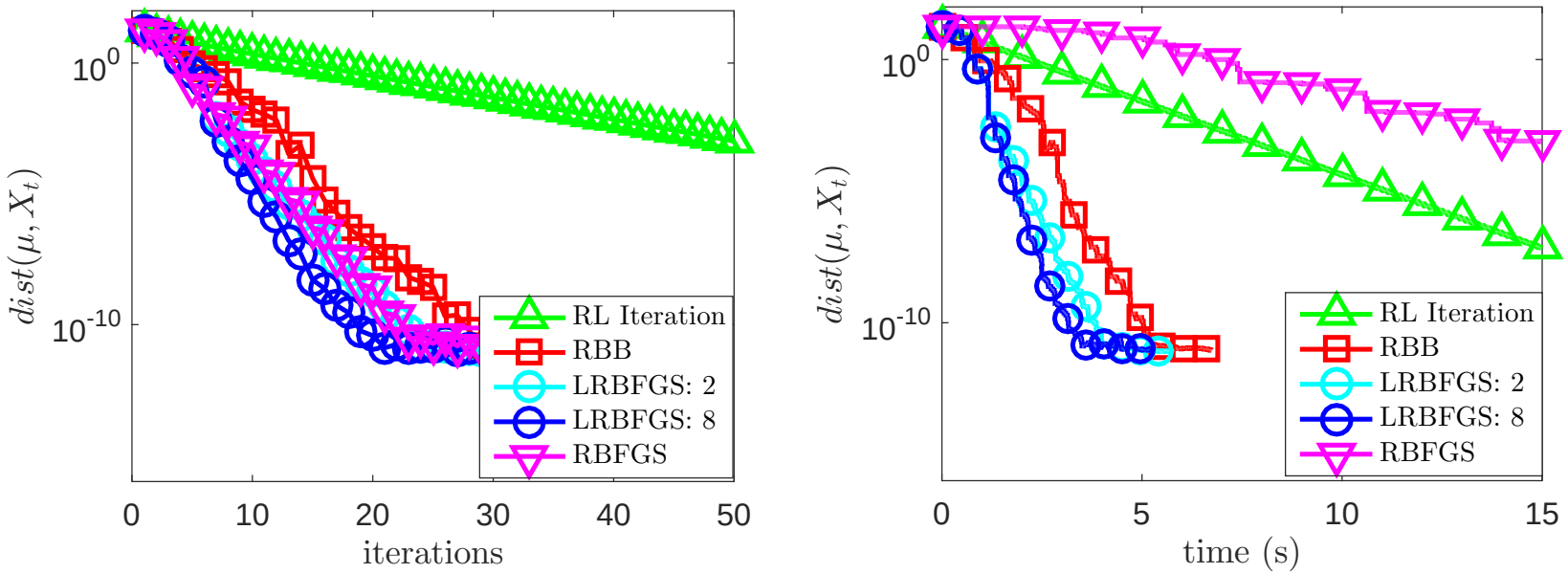


Figure: Evolution of averaged distance between current iterate and the exact Karcher mean with respect to time and iterations with $K = 30$ and $n = 100$; $10^4 \leq \kappa(A_i) \leq 10^7$.

Retraction is important.

The Blind Deconvolution Problem

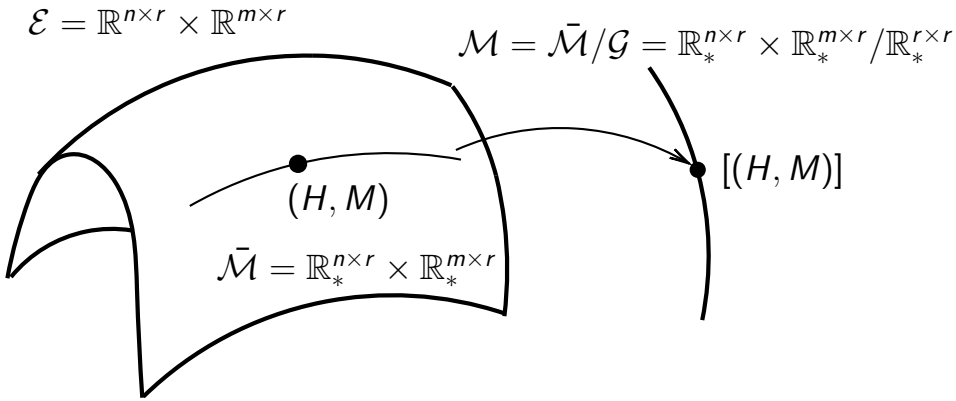
Problem:

$$\min_{X \in \mathbb{R}_1^{n \times m}} F(X) = \|y - \mathrm{diag}(BXC^*)\|_2^2.$$

- $y \in \mathbb{C}^\ell$, $B \in \mathbb{C}^{\ell \times n}$, and $C \in \mathbb{C}^{\ell \times m}$ are given;
- Domain is the manifold of n-by-m rank-1 matrices;
- The superscript star $*$ denotes the conjugate transpose operator.

The Blind Deconvolution Problem

$$\min_{X \in \mathbb{R}_1^{n \times m}} F(X) = \|y - \mathrm{diag}(BXC^*)\|_2^2$$



Two equivalent cost functions:[3]

- on the quotient manifold

$$f_1 : \mathbb{R}_*^n \times \mathbb{R}_*^m / \mathbb{R}_*^1 \to \mathbb{R}$$
$$: [(h, m)] \mapsto f_1([(h, m)]) = \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2.$$

- on ambient space

$$f_2 : \mathbb{R}^n \times \mathbb{R}^m : (h, m) \mapsto f_2(h, m) = \|y - \mathrm{diag}(Bhm^* C^*)\|_2^2;$$

[3] A penalty term [LLSW16] in the cost function is not added here for simplicity.

Results

Apply the same type of method (Newton method with truncated CG) for minimizing f_1 and f_2.

$$f_1 : \mathbb{R}^n_* \times \mathbb{R}^m_* / \mathbb{R}^1_* \to \mathbb{R}; f_2 : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$$

Table: An average of 50 runs. $\ell = 1024$.

m/n	180/340		260/260		340/180	
	f_1	f_2	f_1	f_2	f_1	f_2
time	0.124	0.213	0.139	0.198	0.164	0.219
nf	38.26	37.72	39.54	37	37.82	35.88
ng	33.48	32.72	34.12	32	32.84	31.3
nH	173.56	347.24	181.64	264.32	189.06	268.38
error	8.53_{-8}	1.52_{-7}	7.38_{-8}	1.24_{-7}	3.97_{-8}	5.00_{-8}

Optimizing for f_1 is faster than optimizing for f_2.

Summary

Points in this talk:

- Representation of a manifold changes complexity of an algorithm;
- Riemannian metric influences the condition number of Hessian;
- Retraction affects the number of iterations;
- Optimizing over quotient manifold can be better than optimizing over the ambient space;
- Generalizing algorithms to the Riemannian setting is not straightforward;

Thank you

Thank you!

References I

D. A. Bini and B. Iannazzo.

Computing the Karcher mean of symmetric positive definite matrices.
Linear Algebra and its Applications, 438(4):1700–1710, February 2013.
doi:10.1016/j.laa.2011.08.052.

Wen Huang, P.-A. Absil, and K. A. Gallivan.

Intrinsic representation of tangent vectors and vector transport on matrix manifolds.
Numerische Mathematik, 2016.

Wen Huang, P.-A. Absil, and K. A. Gallivan.

A riemannian bfgs method without differentiated retraction for nonconvex optimization problems.
Technical Report UCL-INMA-2017.04, U.C.Louvain, 2017.

Wen Huang, K. A. Gallivan, and P.-A. Absil.

A Broyden Class of Quasi-Newton Methods for Riemannian Optimization.
SIAM Journal on Optimization, 25(3):1660–1685, 2015.

Xiaodong Li, Shuyang Ling, Thomas Strohmer, and Ke Wei.

Rapid, robust, and reliable blind deconvolution via nonconvex optimization.
CoRR, abs/1606.04933, 2016.

C. Qi.

Numerical optimization methods on Riemannian manifolds.
PhD thesis, Florida State University, Department of Mathematics, 2011.

W. Ring and B. Wirth.

Optimization methods on Riemannian manifolds and their application to shape space.
SIAM Journal on Optimization, 22(2):596–627, January 2012.
doi:10.1137/11082885X.

References II

Xinru Yuan, Wen Huang, P.-A. Absil, and K. A. Gallivan.

A Riemannian quasi-newton method for computing the Karcher mean of symmetric positive definite matrices.
Technical Report FSU17-02, Florida State University, 2017.