Geometric mean of symmetric positive definite matrices

Wen Huang

Xiamen University

Tianyuan Mathematical Center in Southwest China, Sichuan University, Chengdu

November 20, 2020

This is joint work with Xinru Yuan, Kyle A. Gallivan and Pierre-Antoine Absil.

Computing a geometric mean of SPD matrices: *Riemannian methods;*

Computing a geometric mean of SPD matrices: *Riemannian methods;*

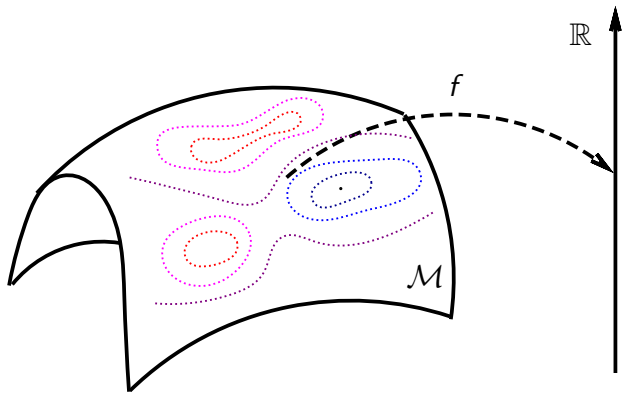
A difference between Riemannian optimization and Euclidean constrained optimization: *choice of metric;*

Riemannian Optimization

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

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

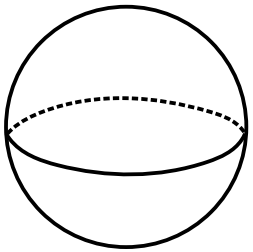
where $\mathcal{M}$ is a Riemannian manifold.



Riemannian manifold = manifold + Riemannian metric

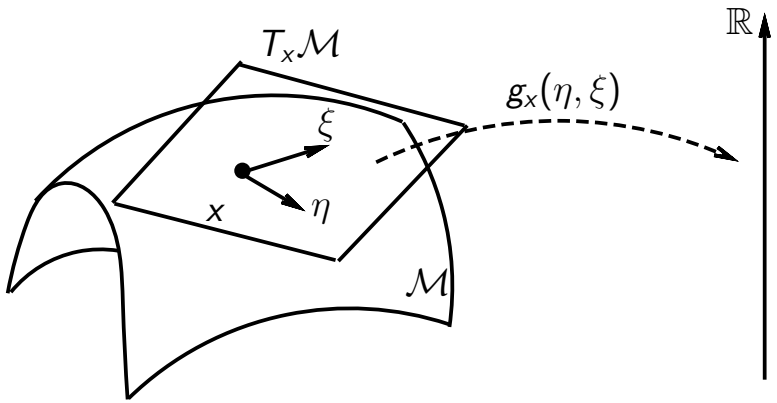
Riemannian Manifold

Manifolds:


Sphere

- 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 $\mathbb{R}^n$;
- And many more.

Riemannian metric:



A Riemannian metric, denoted by g, is a smoothly-varying inner product on the tangent spaces;

Riemannian Metric

Euclidean metric:

$$g(U, V) = \mathrm{trace}(U^T V).$$

- Oblique manifold: $\mathrm{Ob}(p, n) = \{X \in \mathbb{R}^{n\times p} \mid \mathrm{diag}(X^T X) = \mathbf{1}_p\}$;
  ICA [SAGQ12], Sparse PCA [GHT15];
- Stiefel manifold: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n\times p} \mid X^T X = I_p\}$;
  SVD [SI13], Electron structure calculation [HJL+19];
- Fixed-rank manifold: $\mathbb{R}_r^{n\times p} = \{X \in \mathbb{R}^{n\times p} \mid \mathrm{rank}(X) = r\}$;
  Matrix completion [Van13], Weighted low-rank approximation [ZHG+15]

Riemannian Metric

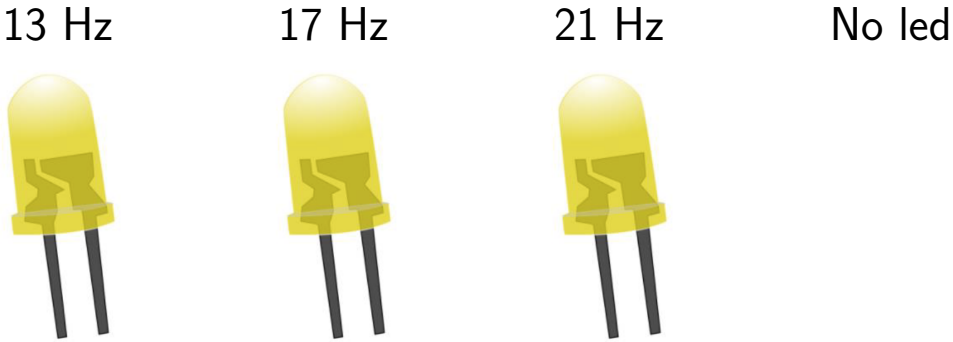
Euclidean metric:

$$g(U, V) = \mathrm{trace}(U^T V).$$

- Oblique manifold: $\mathrm{Ob}(p, n) = \{X \in \mathbb{R}^{n\times p} \mid \mathrm{diag}(X^T X) = \mathbf{1}_p\}$;
  ICA [SAGQ12], Sparse PCA [GHT15];
- Stiefel manifold: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n\times p} \mid X^T X = I_p\}$;
  SVD [SI13], Electron structure calculation [HJL+19];
- Fixed-rank manifold: $\mathbb{R}_r^{n\times p} = \{X \in \mathbb{R}^{n\times p} \mid \mathrm{rank}(X) = r\}$;
  Matrix completion [Van13], Weighted low-rank approximation [ZHG+15]

Euclidean metric or other Riemannian metric?

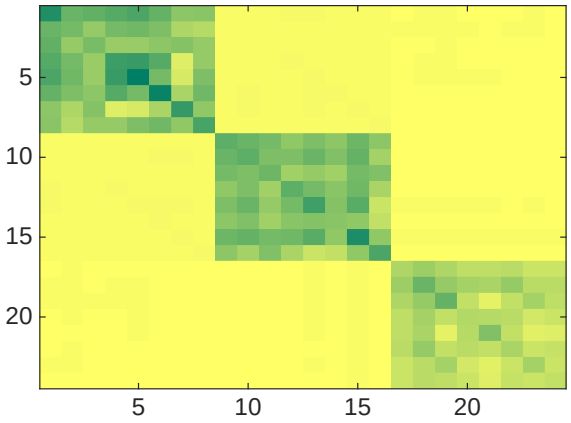
Application: Electroencephalography (EEG) Classification



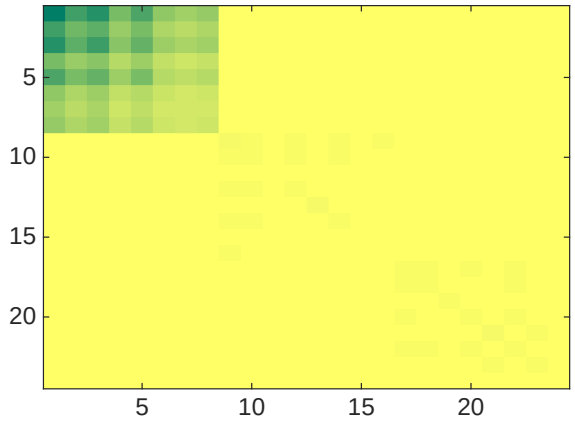
- The subject is either asked to focus on one specific blinking LED or a location without LED
- EEG system is used to record brain signals
- Covariance matrices of size 24×24 are used to represent EEG recordings [KCB$^+$15, MC17]
- Covariance matrices in $\mathcal{S}_{++}^{n} = \{A \in \mathbb{R}^{n \times n} : A = A^T, A \succ 0\}$

EEG Classification: Examples of Covariance Matrices

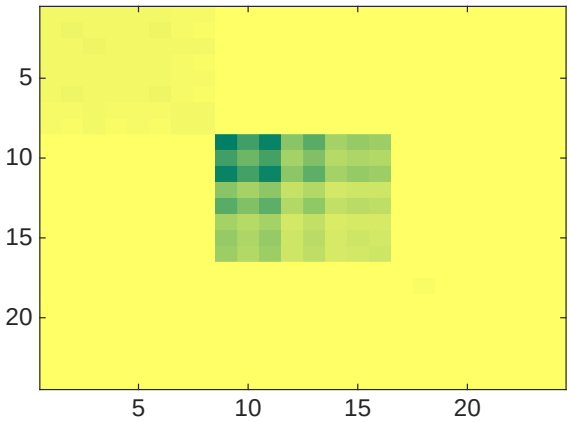
Resting Class



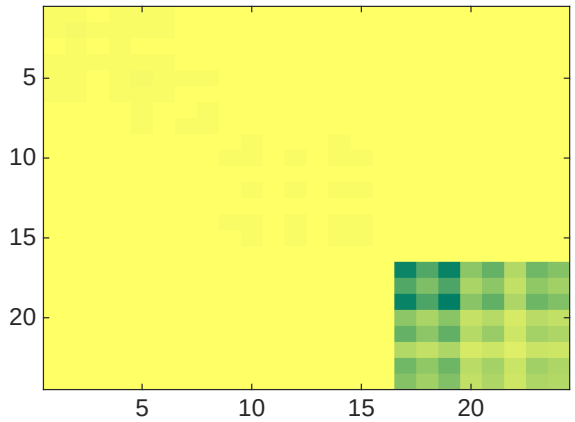
13 Hz Class



17 Hz Class

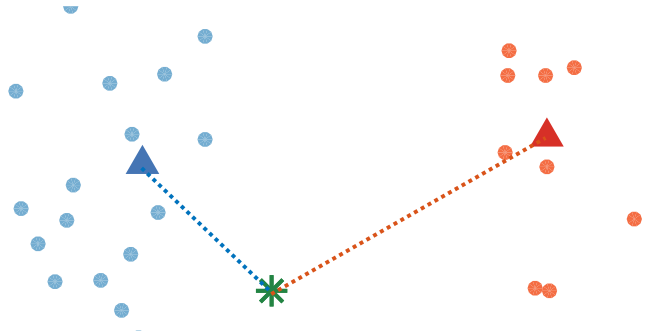


21 Hz Class



EEG Classification: Minimum Distance to Mean classier

Goal: classify new covariance matrix using Minimum Distance to Mean Classifier

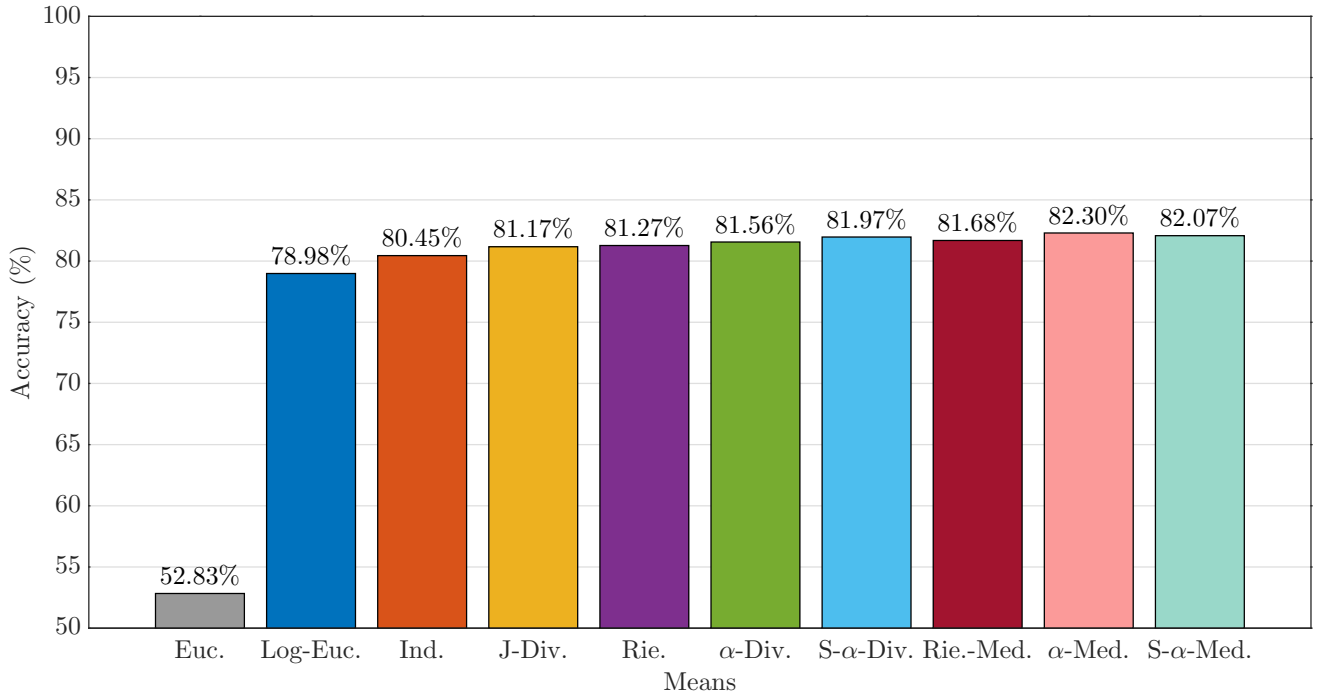


- For each class $k = 1, \ldots, K$, compute the center μ_k of the covariance matrices in the training set that belong to class k
- Classify a new covariance matrix X according to

$$\hat{k} = \underset{1 \leq k \leq K}{\operatorname{argmin}} \, \delta(X, \mu_k)$$

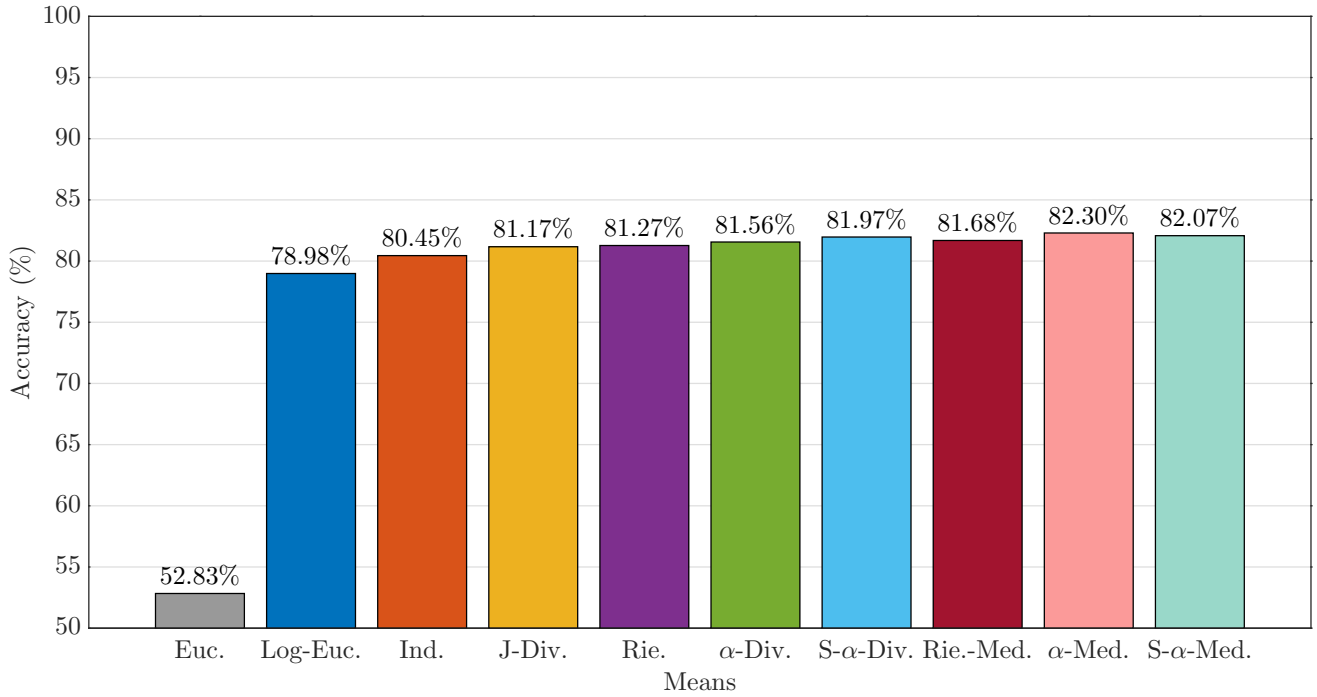
EEG Classfification: Accuracy

- Accuracy comparison



EEG Classfification: Accuracy

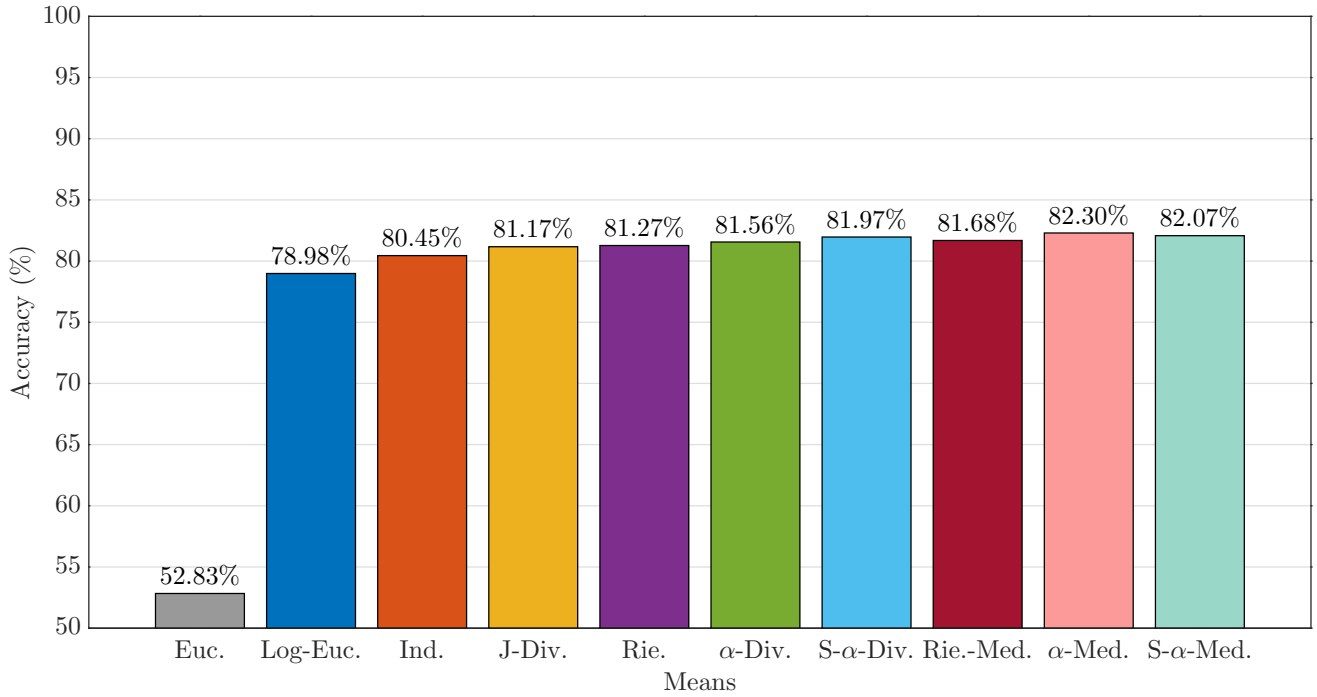
- Accuracy comparison



Euclidean metric is not appropriate to define the problem!

EEG Classfification: Accuracy

- Accuracy comparison



Euclidean metric is not appropriate to define the problem!

Is Euclidean metric appropriate for optimization? Averaging SPD matrices.

Averaging Schemes: from Scalars to Matrices

Let $A_1, \ldots, A_K$ be SPD matrices.

- Generalized arithmetic mean: $\frac{1}{K} \sum_{i=1}^{K} A_i$

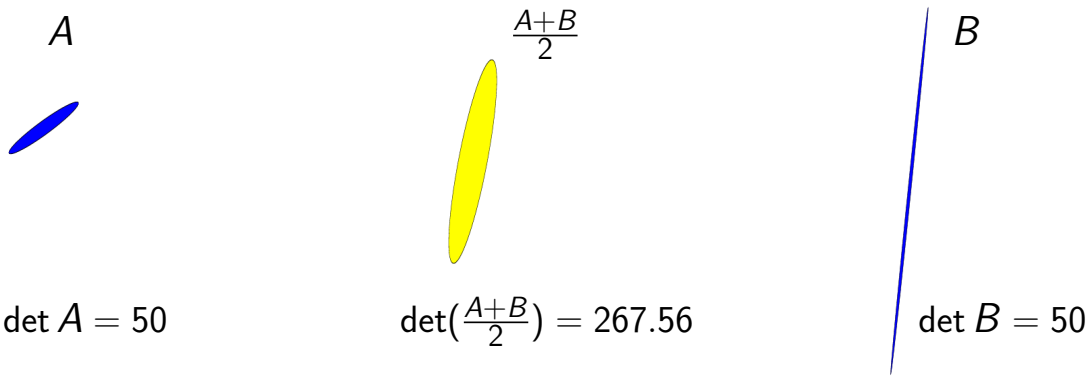
  $\rightarrow$ Not appropriate in many practical applications

Averaging Schemes: from Scalars to Matrices

Let $A_1, \ldots, A_K$ be SPD matrices.

- Generalized arithmetic mean: $\frac{1}{K} \sum_{i=1}^{K} A_i$

  $\rightarrow$ Not appropriate in many practical applications

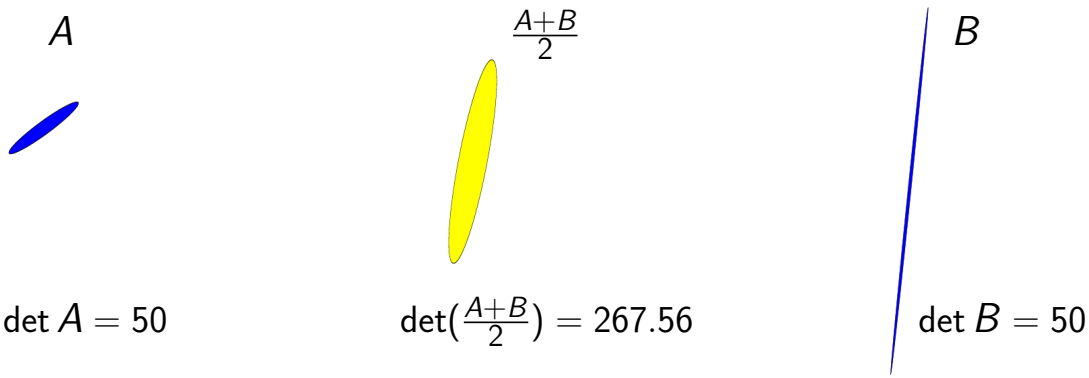


Averaging Schemes: from Scalars to Matrices

Let $A_1, \ldots, A_K$ be SPD matrices.

- Generalized arithmetic mean: $\frac{1}{K}\sum_{i=1}^{K} A_i$

  $\rightarrow$ Not appropriate in many practical applications



- Generalized geometric mean: $(A_1 \cdots A_K)^{1/K}$

  $\rightarrow$ Not appropriate due to non-commutativity

  $\rightarrow$ How to define a matrix geometric mean?

Desired Properties of a Matrix Geometric Mean

The desired properties are given in the ALM list[1], some of which are:

- $G(A_{\pi(1)}, \ldots, A_{\pi(K)}) = G(A_1, \ldots, A_K)$ with π a permutation of $(1, \ldots, K)$
- if $A_1, \ldots, A_K$ commute, then $G(A_1, \ldots, A_K) = (A_1, \ldots, A_K)^{1/K}$
- $G(A_1, \ldots, A_K)^{-1} = G(A_1^{-1}, \ldots, A_K^{-1})$
- $\det(G(A_1, \ldots, A_K)) = (\det(A_1) \cdots \det(A_K))^{1/K}$

[1] T. Ando, C.-K. Li, and R. Mathias, *Geometric means*, Linear Algebra and Its Applications, 385:305-334, 2004

Geometric Mean of SPD Matrices

- A well-known mean on the manifold of SPD matrices is the **Karcher mean** [Kar77]:

$$G(A_1, \ldots, A_K) = \arg \min_{X \in \mathcal{S}^n_{++}} \frac{1}{2K} \sum_{i=1}^{K} \delta^2(X, A_i), \tag{1}$$

where $\delta(X, Y) = \| \log(X^{-1/2} Y X^{-1/2}) \|_F$ is the geodesic distance under the affine-invariant metric

$$g(\eta_X, \xi_X) = \text{trace}(\eta_X X^{-1} \xi_X X^{-1})$$

- The Karcher mean defined in (1) satisfies all the geometric properties in the ALM list [LL11]

Algorithms

$$G(A_1, \ldots, A_k) = \operatorname*{argmin}_{X \in \mathcal{S}_{++}^n} \frac{1}{2K} \sum_{i=1}^{K} \delta^2(X, A_i),$$

- Riemannian steepest descent [RA11, Ren13]
- Riemannian Barzilai-Borwein method [IP15]
- Riemannian Newton method [RA11]
- Richardson-like iteration [BI13]
- Riemannian steepest descent, conjugate gradient, BFGS, and trust region Newton methods [JVV12]
- Limited-memory Riemannian BFGS method [YHAG19]

Algorithms

$$G(A_1, \ldots, A_k) = \operatorname*{argmin}_{X \in \mathcal{S}_{++}^n} \frac{1}{2K} \sum_{i=1}^{K} \delta^2(X, A_i),$$

- Riemannian steepest descent [RA11, Ren13]
- Riemannian Barzilai-Borwein method [IP15]
- Riemannian Newton method [RA11]
- Richardson-like iteration [BI13]
- Riemannian steepest descent, conjugate gradient, BFGS, and trust region Newton methods [JVV12]
- Limited-memory Riemannian BFGS method [YHAG19]

Riemannian gradient is used in all the above methods!

Algorithms

$$G(A_1, \ldots, A_k) = \operatorname*{argmin}_{X \in \mathcal{S}_{++}^n} \frac{1}{2K} \sum_{i=1}^{K} \delta^2(X, A_i),$$

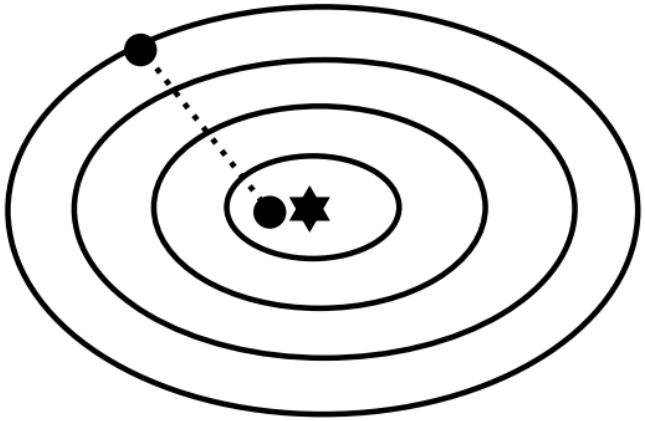
- Riemannian steepest descent [RA11, Ren13]
- Riemannian Barzilai-Borwein method [IP15]
- Riemannian Newton method [RA11]
- Richardson-like iteration [BI13]
- Riemannian steepest descent, conjugate gradient, BFGS, and trust region Newton methods [JVV12]
- Limited-memory Riemannian BFGS method [YHAG19]

Riemannian gradient is used in all the above methods!

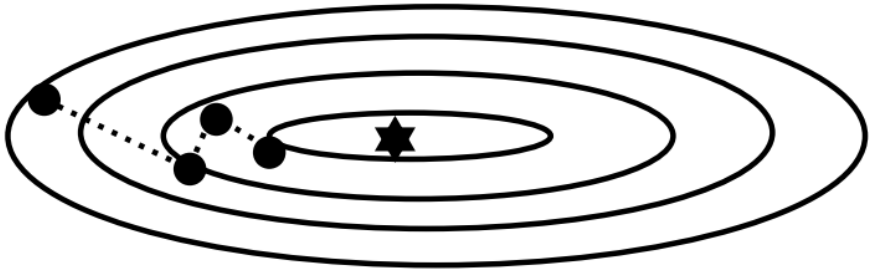
The LRBFGS in [YHAG19] is preferred.

Conditioning of the Objective Function

Hemstitching phenomenon
for steepest descent



well-conditioned Hessian



ill-conditioned Hessian

- Small condition number $\Rightarrow$ fast convergence
- Large condition number $\Rightarrow$ slow convergence

Conditioning of the Karcher Mean Objective Function

- **Riemannian metric**:

$$g_X(\xi,\eta) = \mathrm{trace}(\xi X^{-1}\eta X^{-1})$$

- **Euclidean metric**:

$$g_X(\xi,\eta) = \mathrm{trace}(\xi\eta)$$

Condition number κ of Hessian at the minimizer μ:

- Hessian of Riemannian metric:
  - $\kappa(H^R) \leq 1 + \dfrac{\ln(\max \kappa_i)}{2}$, where $\kappa_i = \kappa(\mu^{-1/2} A_i \mu^{-1/2})$
  - $\kappa(H^R) \leq 20$ if $\max(\kappa_i) = 10^{16}$
- Hessian of Euclidean metric:
  - $\dfrac{\kappa^2(\mu)}{\kappa(H^{\mathrm{R}})} \leq \kappa(H^{\mathrm{E}}) \leq \kappa(H^{\mathrm{R}})\kappa^2(\mu)$
  - $\kappa(H^E) \geq \kappa^2(\mu)/20$

BFGS: from Euclidean to Riemannian

- Update formula:

$$x_{k+1} = x_k + \alpha_k \eta_k$$

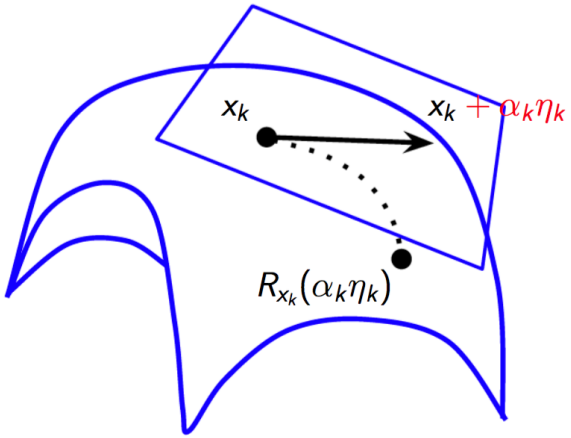
- Search direction:

$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$

- B_k update:

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

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



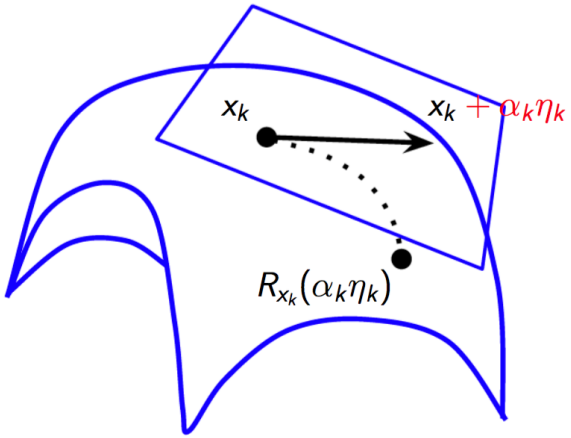
Optimization on a Manifold

BFGS: from Euclidean to Riemannian

- Update formula:

replace by $R_{x_k}(\eta_k)$

$$x_{k+1} = x_k + \alpha_k \eta_k$$



Optimization on a Manifold

- Search direction:

$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$

- B_k update:

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

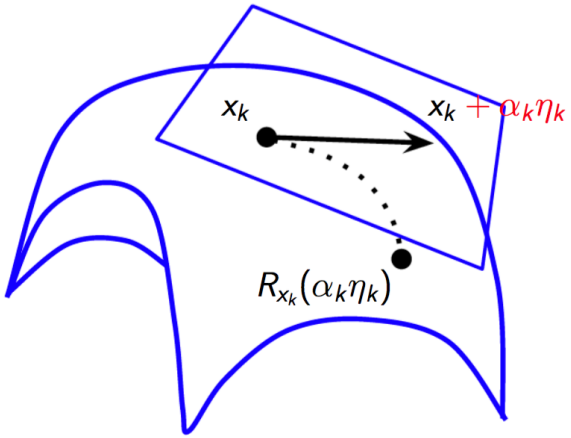
where $s_k = x_{k+1} - x_k$, and $y_k = \operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k)$

BFGS: from Euclidean to Riemannian

- Update formula:

replace by $R_{x_k}(\eta_k)$

$$x_{k+1} = x_k + \alpha_k \eta_k$$



Optimization on a Manifold

- Search direction:

$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$

- B_k update:

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

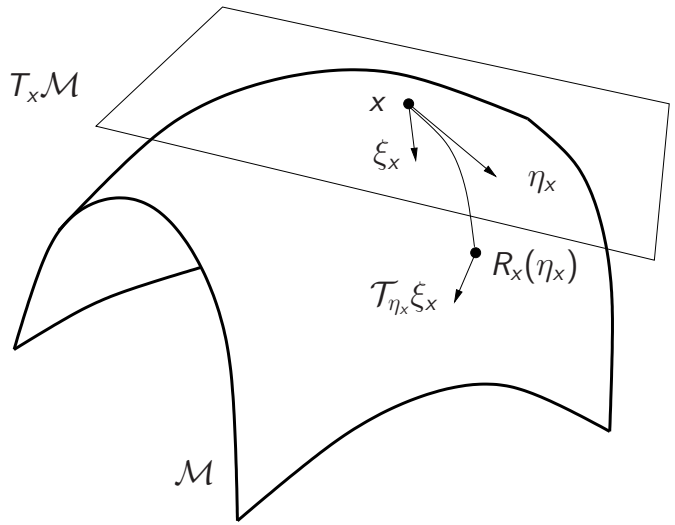
where $s_k = x_{k+1} - x_k$, and $y_k = \operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k)$

replaced by $R_{x_k}^{-1}(x_{k+1})$

on different tangent spaces

BFGS: from Euclidean to Riemannian

A vector transport: $\mathcal{T} : \mathrm{T}\mathcal{M} \times \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}\xi_x$:



- Euclidean: $y_k = \underline{\operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k)}$
- Riemannian: $y_k = \underline{\operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\alpha_k \eta_k} \operatorname{grad} f(x_k)}$

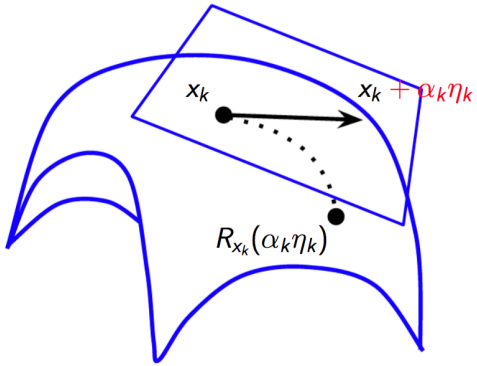
BFGS: from Euclidean to Riemannian

- Update formula:

$$x_{k+1} = \underline{R_{x_k}(\alpha_k \eta_k)}$$

- Search direction:

$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$



Optimization on a Manifold

- B_k update:

$$\tilde{B}_k = \underline{\mathcal{T}_{\alpha_k \eta_k} \circ B_k \circ \mathcal{T}^{-1}_{\alpha_k \eta_k}},$$

$$B_{k+1} = \underline{\tilde{B}_k - \frac{\tilde{B}_k s_k s_k^\flat \tilde{B}_k}{s_k^\flat \tilde{B}_k s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k}},$$

where $s_k = \underline{\mathcal{T}_{\alpha_k \eta_k}(\alpha_k \eta_k)}$, and $y_k = \underline{\operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\alpha_k \eta_k} \operatorname{grad} f(x_k)}$;

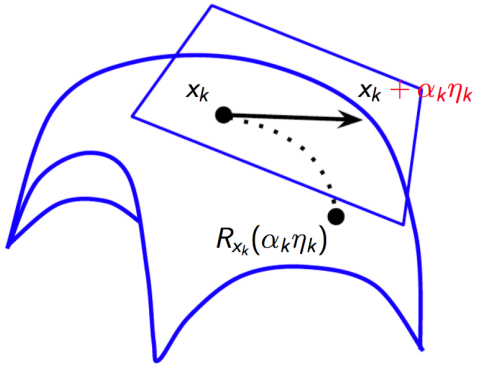
BFGS: from Euclidean to Riemannian

- Update formula:

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

- Search direction:

$$\eta_k = -B_k^{-1} \operatorname{grad} f(x_k)$$



Optimization on a Manifold

- B_k update:

$$\tilde{B}_k = \mathcal{T}_{\alpha_k \eta_k} \circ B_k \circ \mathcal{T}^{-1}_{\alpha_k \eta_k}, \leftarrow \text{matrix matrix multiplication}$$

$$B_{k+1} = \tilde{B}_k - \frac{\tilde{B}_k s_k s_k^\flat \tilde{B}_k}{s_k^\flat \tilde{B}_k s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k},$$

where $s_k = \mathcal{T}_{\alpha_k \eta_k}(\alpha_k \eta_k)$, and $y_k = \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\alpha_k \eta_k} \operatorname{grad} f(x_k)$;

matrix vector multiplication

matrix vector multiplication

Extra cost on vector transports!

Limited-memory RBFGS (LRBFGS)

Riemannian BFGS:

- Let $\mathcal{H}_{k+1} = \mathcal{B}_{k+1}^{-1}$
- $\mathcal{H}_{k+1} = (\mathrm{id} - \rho_k y_k s_k^\flat)\tilde{\mathcal{H}}_k(\mathrm{id} - \rho_k y_k s_k^\flat) + \rho_k s_k s_k^\flat$
  where $s_k = \mathcal{T}_{\alpha_k\eta_k}\alpha_k\eta_k$, $y_k = \operatorname{grad} f(x_{k+1}) - \mathcal{T}_{\alpha_k\eta_k} \operatorname{grad} f(x_k)$,
  $\rho_k = 1/g(y_k, s_k)$ and $\tilde{\mathcal{H}}_k = \mathcal{T}_{\alpha_k\eta_k} \circ \mathcal{H}_k \circ \mathcal{T}_{\alpha_k\eta_k}^{-1}$

Limited-memory Riemannian BFGS:

- Stores only the m most recent s_k and y_k
- Transports these vectors to the new tangent space rather than $\mathcal{H}_k$
- Computational and storage complexity depends upon m

Implementations

- Retraction
  - Exponential mapping: $\mathrm{Exp}_X(\xi) = X^{1/2}\exp(X^{-1/2}\xi X^{-1/2})X^{1/2}$
  - Second order approximation retraction [JVV12]:
    $$R_X(\xi) = X+\xi+\frac{1}{2}\xi X^{-1}\xi = \tfrac{1}{2}(\xi X^{-1/2}+X^{1/2})(\xi X^{-1/2}+X^{1/2})^T+\tfrac{1}{2}X$$
- Vector transport
  - Parallel translation: $\mathcal{T}_{p_\eta}(\xi) = Q\xi Q^T$, with $Q = X^{\frac{1}{2}}\exp(\frac{X^{-\frac{1}{2}}\eta X^{-\frac{1}{2}}}{2})X^{-\frac{1}{2}}$
  - Vector transport by parallelization [HAG16] : essentially an identity

Implementation

Vector Transport by Parallelization

- Vector transport by parallelization:

$$\mathcal{T}_{\eta_x}\xi_x = B_y B_x^{\dagger}\xi_x;$$

where $y = R_x(\eta_x)$ and $\dagger$ denotes pseudo-inverse, has identity implementation [HAG16]:

$$\mathcal{T}_{\tilde{\eta}_x}\tilde{\xi}_x = \tilde{\xi}_x.$$

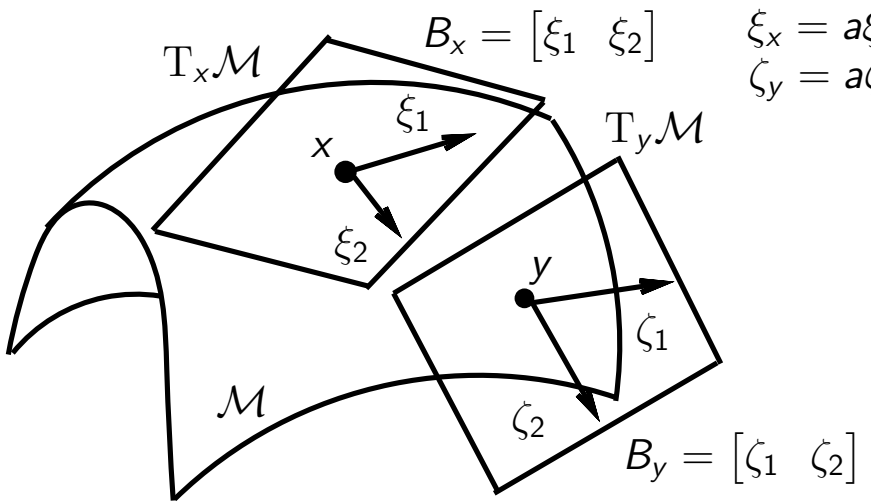
Example:

Extrinsic:

$$\zeta = \mathcal{T}_{\eta}\xi = B_y B_x^{\dagger}\xi$$

Intrinsic:

$$\begin{aligned}\tilde{\zeta} =& \widetilde{\mathcal{T}_{\eta}\xi} \\ =& B_y^{\dagger} B_y B_x^{\dagger} B_x \tilde{\xi} \\ =& \tilde{\xi}\end{aligned}$$



Implementations

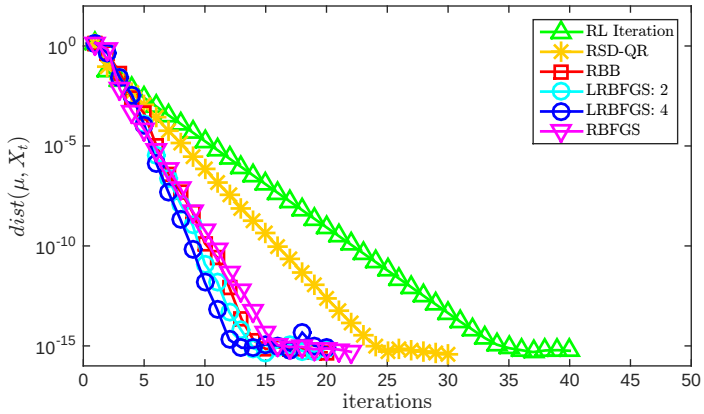
- Cholesky $X_k = L_k L_k^T$ assumed to be computed on each step
- B_X of $\mathrm{T}_X \mathcal{S}_{++}^n$, the orthonormal basis of $\mathrm{T}_X \mathcal{S}_{++}^n$

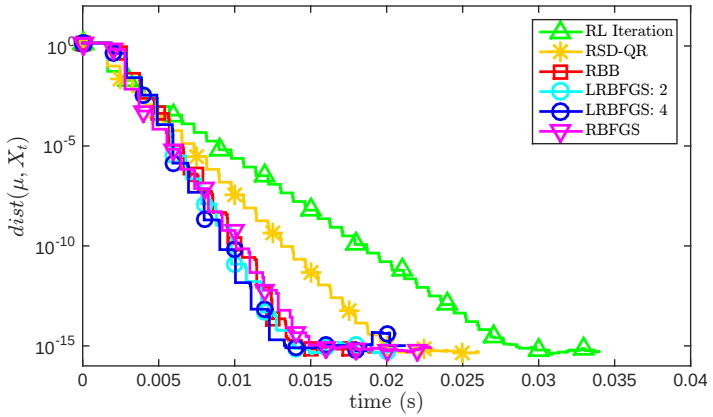
$$B_X = \{Le_i e_i^T L^T : i = 1, \ldots, n\} \cup \{\frac{1}{\sqrt{2}} L(e_i e_j^T + e_j e_i^T) L^T,$$
$$i < j,\ i = 1, \ldots, n,\ j = 1, \ldots, n\},$$

  where $\{e_i, \ldots, e_n\}$ is the standard basis of n-dimensional Euclidean space.
- orthonormal under $g_X(\xi_X, \eta_X)$.
- $\xi_X = B_X \hat{\xi}_X \leftrightarrow \xi_X = LSL^T$, where S is symmetric and constains scale coefficients.
- intrinsic representation of tangent vectors is easily maintained.

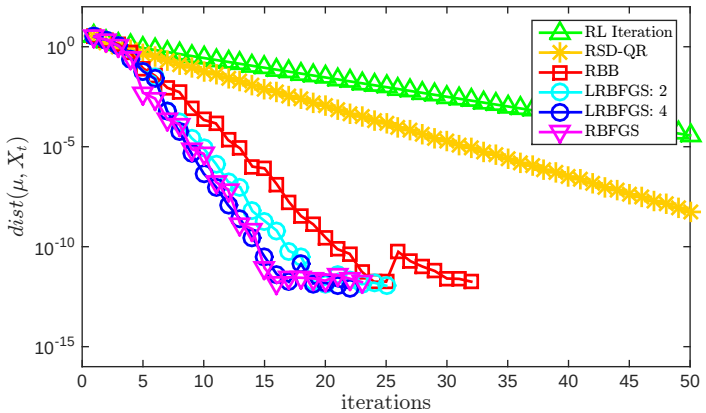
Numerical Results: $K = 100$, size $= 3 \times 3$, $d = 6$

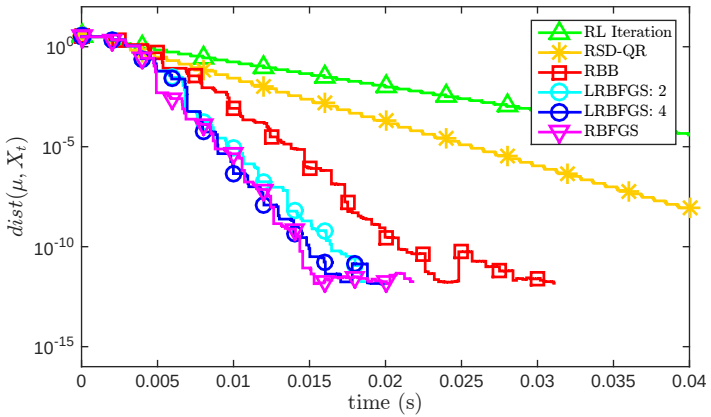
- $1 \leq \kappa(A_i) \leq 200$





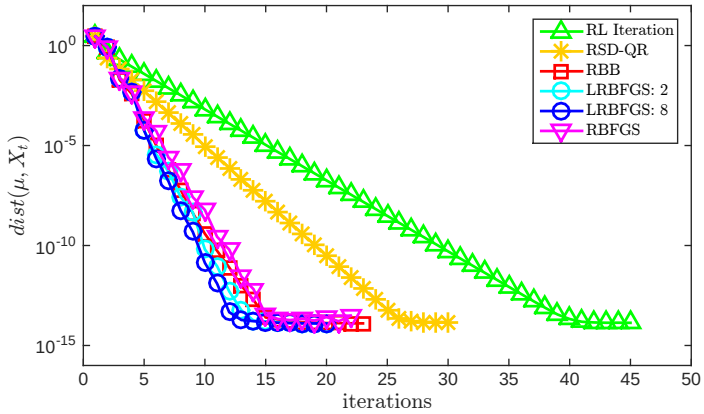
- $10^3 \leq \kappa(A_i) \leq 10^7$

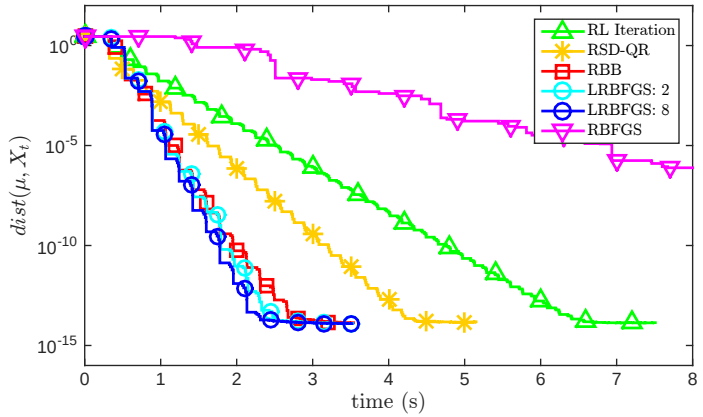




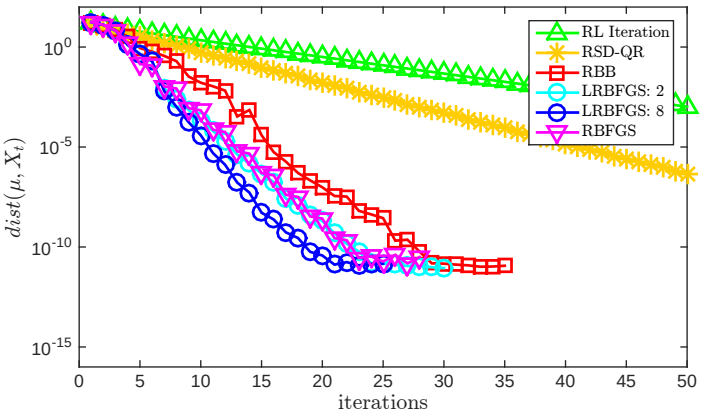
Numerical Results: $K = 30$, size $= 100 \times 100$, $d = 5050$

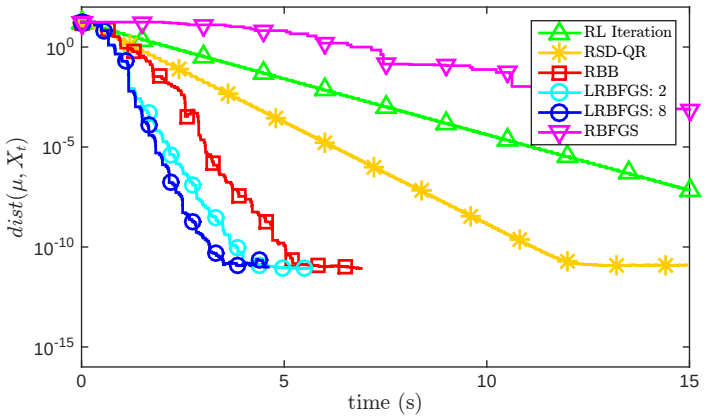
- $1 \leq \kappa(A_i) \leq 20$





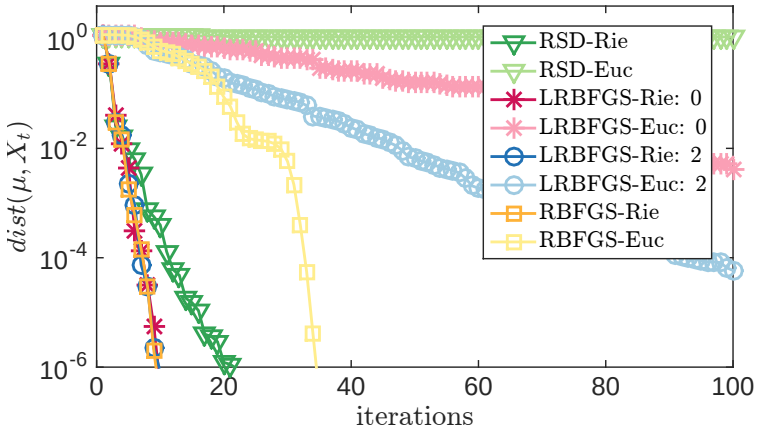
- $10^4 \leq \kappa(A_i) \leq 10^7$

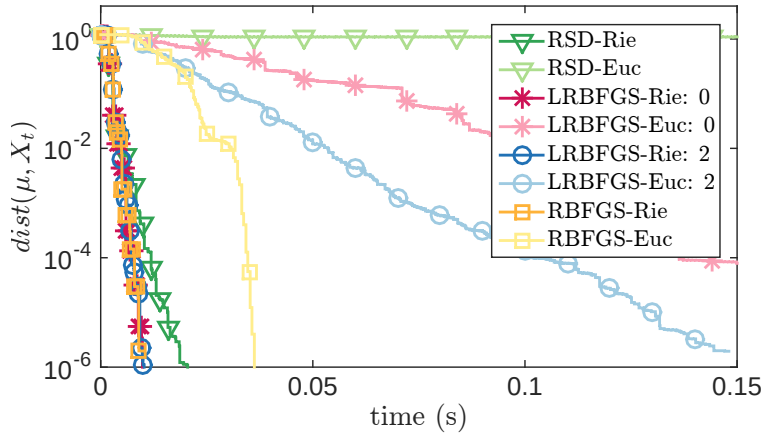




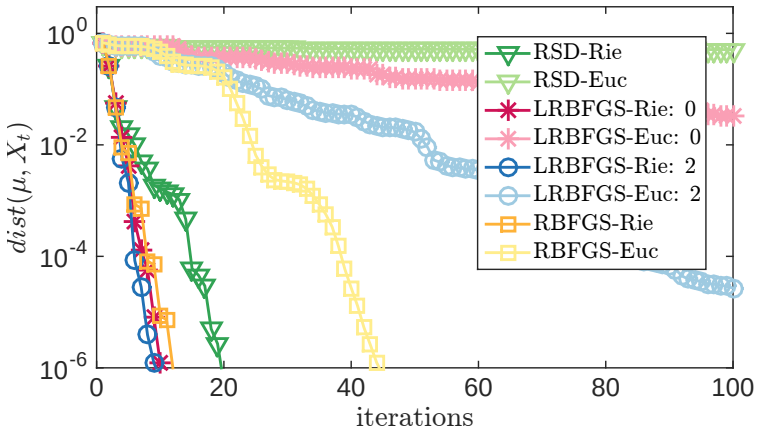
Numerical Results: Riemannian vs. Euclidean Metrics

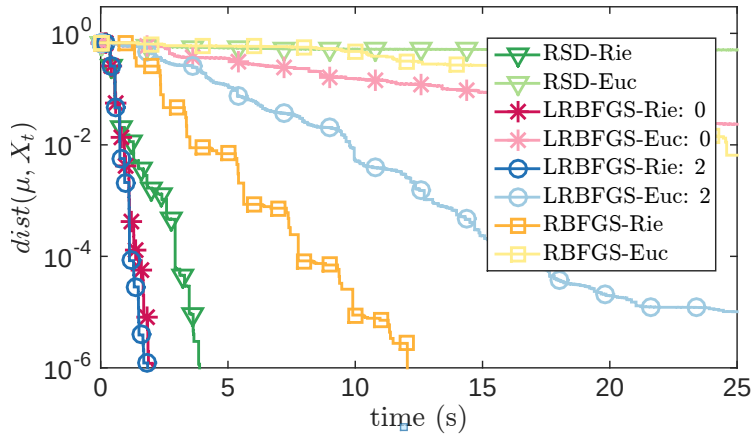
- $K = 100$, $n = 3$, and $1 \leq \kappa(A_i) \leq 10^6$.





- $K = 30$, $n = 100$, and $1 \leq \kappa(A_i) \leq 10^5$.





Summary

Non-Euclidean metric helps!

- Covariance matrices classification
- A geometric mean of SPD matrices
- Conditioner number of the Hessian
- Limited-memory Riemannian BFGS
- Numerical experiments

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, 2013.

Matthieu Genicot, Wen Huang, and Nickolay T. Trendafilov.

Weakly correlated sparse components with nearly orthonormal loadings.
In *Geometric Science of Information*, pages 484–490, 2015.

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

Intrinsic representation of tangent vectors and vector transport on matrix manifolds.
Numerische Mathematik, 136(2):523–543, 2016.

Jiang Hu, Bo Jiang, Lin Lin, Zaiwen Wen, and Ya-xiang Yuan.

Structured quasi-newton methods for optimization with orthogonality constraints.
SIAM Journal on Scientific Computing, 41(4):A2239–A2269, 2019.

Bruno Iannazzo and Margherita Porcelli.

The Riemannian Barzilai-Borwein method with nonmonotone line-search and the Karcher mean computation.
Optimization online, December, 2015.

B. Jeuris, R. Vandebril, and B. Vandereycken.

A survey and comparison of contemporary algorithms for computing the matrix geometric mean.
Electronic Transactions on Numerical Analysis, 39:379–402, 2012.

H. Karcher.

Riemannian center of mass and mollifier smoothing.
Communications on Pure and Applied Mathematics, 1977.

References II

Emmanuel K Kalunga, Sylvain Chevallier, Quentin Barthélemy, Karim Djouani, Yskandar Hamam, and Eric Monacelli.

From Euclidean to Riemannian means: Information geometry for SSVEP classification.
In *International Conference on Networked Geometric Science of Information*, pages 595–604. Springer, 2015.

J. Lawson and Y. Lim.

Monotonic properties of the least squares mean.
Mathematische Annalen, 351(2):267–279, 2011.

Estelle M Massart and Sylvain Chevallier.

Inductive means and sequences applied to online classication of EEG.
Technical report, ICTEAM Institute, Universite Catholiqué de Louvain, 2017.

Q. Rentmeesters and P.-A. Absil.

Algorithm comparison for Karcher mean computation of rotation matrices and diffusion tensors.
In *19th European Signal Processing Conference*, pages 2229–2233, Aug 2011.

Q. Rentmeesters.

Algorithms for data fitting on some common homogeneous spaces.
PhD thesis, Universite catholiqué de Louvain, 2013.

S. E. Selvan, U. Amato, K. A. Gallivan, and C. Qi.

Descent algorithms on oblique manifold for source-adaptive ICA contrast.
IEEE Transactions on Neural Networks and Learning Systems, 23(12):1930–1947, 2012.

H. Sato and T. Iwai.

A Riemannian optimization approach to the matrix singular value decomposition.
SIAM Journal on Optimization, 23(1):188–212, 2013.

References III

B. Vandereycken.

Low-rank matrix completion by Riemannian optimization—extended version.
SIAM Journal on Optimization, 23(2):1214–1236, 2013.

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

Computing the matrix geometric mean: Riemannian vs Euclidean conditioning, implementation techniques, and a Riemannian BFGS method.
Technical Report UCL-INMA-2019.05, U.C.Louvain, 2019.

G. Zhou, W. Huang, K. A. Gallivan, P. Van Dooren, and P.-A. Absil.

A Riemannian rank-adaptive method for low-rank optimization.
Neurocomputing, 2015.