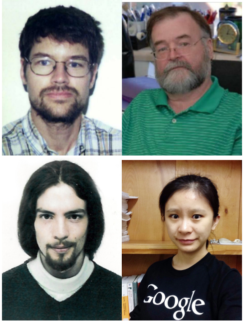
Riemannian Optimization and its Application to Computations on Symmetric Positive Definite Matrices

Wen Huang

Université catholique de Louvain

Joint work with:

- Pierre-Antoine Absil, Professor of Mathematical Engineering, *Université catholique de Louvain*
- Kyle A. Gallivan, Professor of Mathematics, *Florida State University*
- Sylvain Chevallier, Assistant Professor, *Université de Versailles*
- Xinru Yuan, Ph.D candidate in Applied and Computational Mathematics, *Florida State University*

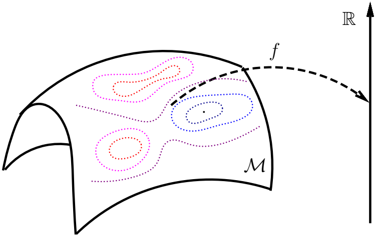


Riemannian Optimization

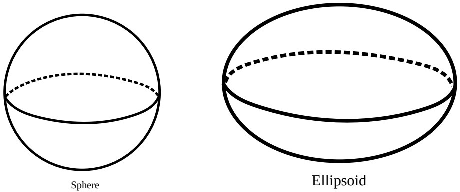
Problem: Given $f(x) : \mathcal{M} \rightarrow \mathbb{R}$, solve

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

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



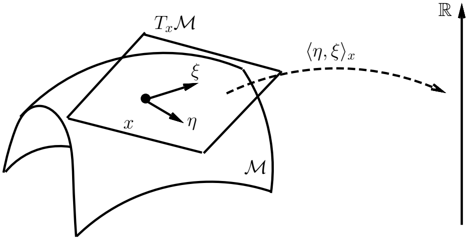
Examples of Manifolds



- Stiefel manifold: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n \times p} | X^T X = I_p\}$
- Grassmann manifold: Set of all p-dimensional subspaces of $\mathbb{R}^n$
- Set of fixed rank m-by-n matrices
- And many more

Riemannian Manifolds

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

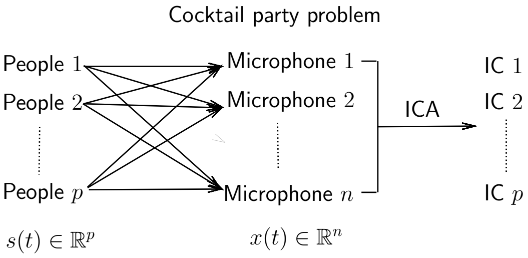


Applications

Four applications are used to demonstrate the importances of the Riemannian optimization:

- Independent component analysis [CS93]
- Matrix completion problem [Van12]
- Geometric mean of symmetric positive definite matrices [ALM04, JVV12]
- Dictionary learning of symmetric positive definite matrices [CS15]

Application: Independent Component Analysis



- Observed signal is $x(t) = As(t)$
- One approach:
  - Assumption: $E\{s(t)s(t+\tau)\}$ is diagonal for all τ
  - $C_\tau(x) := E\{x(t)x(x+\tau)^T\} = AE\{s(t)s(t+\tau)^T\}A^T$

Application: Independent Component Analysis

- Minimize joint diagonalization cost function on the Stiefel manifold [TI06]:

$$f : \mathrm{St}(p,n) \to \mathbb{R} : V \mapsto \sum_{i=1}^{N} \|V^T C_i V - \mathrm{diag}(V^T C_i V)\|_F^2.$$

- $C_1, \ldots, C_N$ are covariance matrices and $\mathrm{St}(p,n) = \{X \in \mathbb{R}^{n \times p} | X^T X = I_p\}$.

Application: Matrix Completion Problem

Matrix completion problem

	Movie 1	Movie 2				Movie n
User 1		1			4	
User 2	3	5			4	
			5	1		
User m		2			5	3

Rate matrix M

- The matrix M is sparse
- The goal: complete the matrix M

Application: Matrix Completion Problem

$$
\overset{\text{movies}}{\begin{pmatrix} a_{11} & & & a_{14} \\ & & & a_{24} \\ & & a_{33} & \\ a_{41} & & & \\ & a_{52} & a_{53} & \end{pmatrix}} = \overset{\text{meta-user}}{\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \end{pmatrix}} \overset{\text{meta-movie}}{\begin{pmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \end{pmatrix}}
$$

- Minimize the cost function

$$f : \mathbb{R}_r^{m\times n} \to \mathbb{R} : X \mapsto f(X) = \|P_\Omega M - P_\Omega X\|_F^2.$$

- $\mathbb{R}_r^{m\times n}$ is the set of m-by-n matrices with rank r. It is known to be a Riemannian manifold.

Application: Geometric Mean of Symmetric Positive Definite (SPD) Matrices

Computing the mean of a population of SPD matrices is important in medical imaging, image processing, radar signal processing, and elasticity. The desired properties are given in the ALM[1] list, some of which are

- if $A_1, \dots, A_k$ commute, then $G(A_1, \dots, A_k) = (A_1 \dots A_k)^{\frac{1}{k}}$;
- $G(A_{\pi(1)}, \dots, A_{\pi(k)}) = G(A_1, \dots, A_k)$, with π a permutation of $(1, \dots, k)$;
- $G(A_1, \dots, A_k) = G\left(A_1^{-1}, \dots A_k^{-1}\right)^{-1}$;
- $\det G(A_1, \dots, A_k) = (\det A_1 \dots \det A_k)^{\frac{1}{k}}$;

where $A_1, \dots, A_k$ are SPD matrices, and $G(\cdot, \dots, \cdot)$ denotes the geometric mean of arguments.

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

Application: Geometric Mean of Symmetric Positive Definite Matrices

One geometric mean is the Karcher mean of the manifold of SPD matrices with the affine invariant metric, i.e.,

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

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

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

Application: Dictionary learning of symmetric positive definite (SPD) matrices

Dictionary learning can be applied for classification and denoising.

- Euclidean dictionary learning problem (one formulation):

$$\min_{\|d_i\|_2 \leq 1, r_i \in \mathbb{R}^n} \sum_{i=1}^{N} \|x_i - [d_1, d_2, \ldots, d_n] \, r_i\|_2^2 + \lambda \|r_i\|_1, \qquad (1)$$

  where $x_i \in \mathbb{R}^s, i = 1, \ldots k$ are given data points, $d_i \in \mathbb{R}^s, i = 1, \ldots, n$ and $r_i \in \mathbb{R}^n, i = 1, \ldots, N$ are dictionary and sparse codes respectively.

- Problem (1) is usually solved by alternatively optimizing over $D := [d_1, \ldots, d_n]$ and $R := [r_1, \ldots, r_N]$.

Application: Dictionary learning of symmetric positive definite (SPD) matrices

- Dictionary learning problem of SPD matrices (one formulation):

$$\min_{\mathbf{B}\in\mathcal{M}_n^d, R\in\mathbb{R}_+^{n\times N}} \frac{1}{2}\sum_{i=1}^{N}\left(\mathrm{dist}^2(X_i, \mathbf{B}r_i) + \|r_i\|_1\right) + \mathrm{trace}\left(\mathbf{B}\right),$$

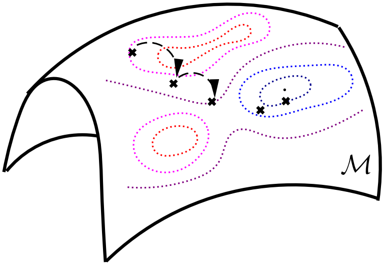
  where $\mathcal{M}_n^d$ denotes the product of n manifolds of SPD matrices $\mathbb{S}_+^d$, i.e., $\mathcal{M}_n^d := \left(\mathbb{S}_+^d\right)^n$.
- Problem (1) also can be solved by alternatively optimizing over $\mathbf{B}$ and R.
- Optimizing over $\mathbf{B}$ is a Riemannian optimization problem.

More Applications

- Large-scale Generalized Symmetric Eigenvalue Problem and SVD
- Blind source separation on both Orthogonal group and Oblique manifold
- Low-rank approximate solution symmetric positive definite Lyapanov $AXM + MXA = C$
- Best low-rank approximation to a tensor
- Rotation synchronization
- Graph similarity and community detection
- Low rank approximation to role model problem
- Shape analysis

Comparison with Constrained Optimization

- 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



Iterations on the Manifold

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

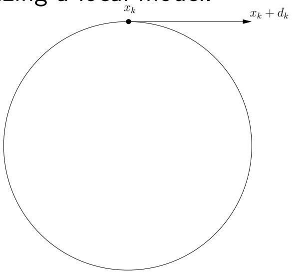
$$x_{k+1} = x_k + \Delta x_k = x_k + \alpha_k s_k \ .$$

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.

Objects

- Direction/movement: $s_k / \Delta x_k$
- Gradient: $\nabla f(x_k)$
- Hessian: $\nabla^2 f(x_k)$
- Addition: $+$



Riemannian gradient and Riemannian Hessian

Definition

The Riemannian gradient of f at x is the unique tangent vector in $T_x M$ satisfying $\forall \eta \in T_x M$, 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 $T_x M$ to $T_x M$ defined as

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

where ∇ is the affine connection.

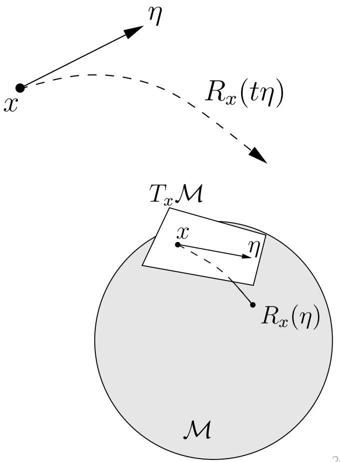
Retractions

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



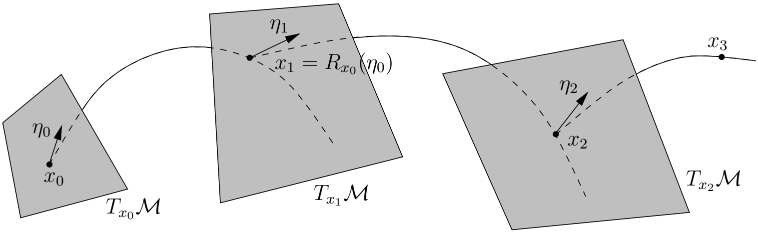
- maps tangent vectors back to the manifold
- defines curves in a direction

Generic Riemannian Optimization Algorithm

1. At iterate $x \in M$
2. Find $\eta \in T_x M$ which satisfies certain condition.
3. Choose new iterate $x_+ = R_x(\eta)$.
4. Goto step 1.

A suitable setting

This paradigm is sufficient for describing many optimization methods.



Categories of Riemannian optimization methods

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)

Categories of Riemannian optimization methods

Elements required for optimizing a cost function (M, g):

- an representation for points x on M, for tangent spaces T_xM, and for the inner products $g_x(\cdot, \cdot)$ on T_xM;
- choice of a retraction $R_x : T_xM \to M$;
- formulas for $f(x), \mathrm{grad}\ f(x)$ and $\mathrm{Hess}\ f(x)$ (or its action);
- Computational and storage efficiency;

Categories of Riemannian optimization methods

Retraction and transport-based: information from multiple tangent spaces

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

Additional element required for optimizing a cost function (M, g):

- formulas for combining information from multiple tangent spaces.

Vector Transports

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 $\mathcal{T}$
- Isometric vector transport $\mathcal{T}_{\mathrm{S}}$ preserve the length of tangent vector

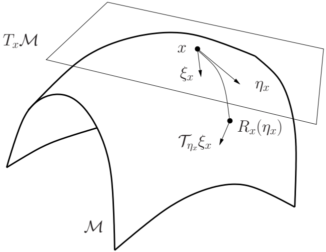


Figure: Vector transport.

Retraction/Transport-based Riemannian Optimization

Benefits

- Increased generality does not compromise the important theory
- Less expensive than or similar to previous approaches
- May provide theory to explain behavior of algorithms specifically developed for a particular application – or closely related ones

Possible Problems

- May be inefficient compared to algorithms that exploit application details

Some History of Optimization On Manifolds (I)

Luenberger (1973), *Introduction to linear and nonlinear programming*. Luenberger mentions the idea of performing line search along geodesics, "which we would use if it were computationally feasible (which it definitely is not)". Rosen (1961) essentially anticipated this but was not explicit in his Gradient Projection Algorithm.

Gabay (1982), *Minimizing a differentiable function over a differential manifold*. Steepest descent along geodesics; Newton's method along geodesics; Quasi-Newton methods along geodesics. On Riemannian submanifolds of $\mathbb{R}^n$.

Smith (1993-94), *Optimization techniques on Riemannian manifolds*. Levi-Civita connection ∇; Riemannian exponential mapping; parallel translation.

Some History of Optimization On Manifolds (II)

The "pragmatic era" begins:

Manton (2002), *Optimization algorithms exploiting unitary constraints*
"The present paper breaks with tradition by not moving along geodesics". The geodesic update $\mathrm{Exp}_x \eta$ is replaced by a projective update $\pi(x + \eta)$, the *projection* of the point $x + \eta$ onto the manifold.

Adler, Dedieu, Shub, et al. (2002), *Newton's method on Riemannian manifolds and a geometric model for the human spine*. The exponential update is relaxed to the general notion of *retraction*. The geodesic can be replaced by any (smoothly prescribed) curve tangent to the search direction.

Absil, Mahony, Sepulchre (2007) Nonlinear conjugate gradient using retractions.

Some History of Optimization On Manifolds (III)

Theory, efficiency, and library design improve dramatically:

Absil, Baker, Gallivan (2004-07), Theory and implementations of Riemannian Trust Region method. Retraction-based approach. Matrix manifold problems, software repository

`http://www.math.fsu.edu/~cbaker/GenRTR`

Anasazi Eigenproblem package in Trilinos Library at Sandia National Laboratory

Absil, Gallivan, Qi (2007-10), Basic theory and implementations of Riemannian BFGS and Riemannian Adaptive Cubic Overestimation. Parallel translation and Exponential map theory, Retraction and vector transport empirical evidence.

Some History of Optimization On Manifolds (IV)

Ring and With (2012), combination of differentiated retraction and isometric vector transport for convergence analysis of RBFGS

Absil, Gallivan, Huang (2009-2015), Complete theory of Riemannian Quasi-Newton and related transport/retraction conditions, Riemannian SR1 with trust-region, RBFGS on partly smooth problems, A C++ library: `http://www.math.fsu.edu/~whuang2/ROPTLIB`

Sato, Iwai (2013-2015), Global convergence analysis using the differentiated retraction for Riemannian conjugate gradient methods

Many people Application interests start to increase noticeably

Current UCL/FSU Methods

- Riemannian Steepest Descent
- Riemannian Trust Region Newton: global, quadratic convergence
- Riemannian Broyden Family : global (convex), superlinear convergence
- Riemannian Trust Region SR1: global, $(d+1)-$superlinear convergence
- For large problems
  - Limited memory RTRSR1
  - Limited memory RBFGS
- Riemannian conjugate gradient (much more work to do on local analysis)
- A library is available at `www.math.fsu.edu/~whuang2/ROPTLIB`

Current/Future Work on Riemannian methods

- Manifold and inequality constraints
- Discretization of infinite dimensional manifolds and the convergence/accuracy of the approximate minimizers – specific to a problem and extracting general conclusions
- Partly smooth cost functions on Riemannian manifold

Geometric Mean and Dictionary Learning

Computations of SPD matrices are used to show the performance of Riemannian methods.

- Geometric mean of SPD matrices [ALM04]

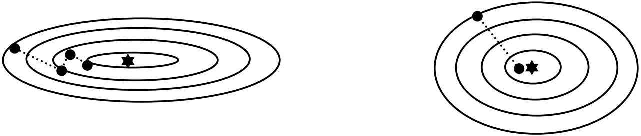
$$\min_{X \in S_+^n} \frac{1}{2k} \sum_{i=1}^{k} \mathrm{dist}^2(X, A_i) = \frac{1}{2k} \sum_{i=1}^{k} \| \log(A_i^{-1/2} X A_i^{-1/2}) \|_F^2.$$

- Dictionary learning for SPD matrices [CS15]

$$\min_{\mathbf{B} \in \mathcal{M}_n^d, R \in \mathbb{R}_+^{n \times N}} \frac{1}{2} \sum_{i=1}^{N} \left(\mathrm{dist}^2(X_i, \mathbf{B} r_i) + \|r_i\|_1 \right) + \mathrm{trace}\,(\mathbf{B})\,.$$

Geometric Mean of SPD matrices

Hemstitching phenomenon



Condition Number at the minimizer [YHAG15]

- For the cost function $F(X) = \frac{1}{2k} \sum_{i=1}^{k} \mathrm{dist}^2(A_i, X)$, we have

$$1 \leq \frac{\mathrm{Hess} F_A(X)[\Delta X, \Delta X]}{\|\Delta X\|^2} \leq 1 + \frac{\log(\max \kappa_i)}{2}.$$

- If $\max \kappa_i = 10^{10}$, then $1 + \frac{\log(\max \kappa_i)}{2} \approx 12.51$.

Algorithms

$$\mathrm{grad} F(X) = -\frac{1}{k} \sum_{i=1}^{k} \mathrm{Log}(A_i X^{-1}) X.$$

First order approaches

- Riemannian steepest descent [RA11]
- Riemannian conjugate gradient [JVV12]
- Richardson-like iteration [BI13]
- Limited-memory Riemannian BFGS method [YHAG15]

Implementations

- Function: $\frac{1}{2k}\sum_{i=1}^{k} \|\log(A_i^{-1/2} X A_i^{-1/2})\|_F^2$;
- Gradient:

$$-\frac{1}{k}\sum_{i=1}^{k} \mathrm{Log}(A_i X^{-1}) X = \frac{1}{k}\sum_{i=1}^{k} A_i^{1/2} \mathrm{Log}(A_i^{-1/2} X A_i^{-1/2}) A_i^{-1/2} X^{1/2}$$

- $A_i^{-1/2}$ can be computed in advance.
- The dominated computational time is on the function evaluation.

Implementations

- Retraction [JVV12]
  - Exponential mapping: $R_X(\xi_X) = X^{1/2}\exp(X^{-1/2}\xi_X X^{-1/2})X^{1/2}$
  - Second order retraction: $R_X(\xi_X) = X + \xi_X + \xi_X X^{-1}\xi_x/2$
- Vector transports:
  - Parallel translation:
    $\mathcal{T}_{\eta_X}\xi_X = Q(X, \eta_X)\xi_X Q(X, \eta_X)^T$,
    $Q(X, \eta_X) = X^{1/2}\exp\left(\frac{X^{-1/2}\eta_X X^{-1/2}}{2}\right)X^{-1/2}$
  - Vector transport by parallelization: essentially an identity
- The dominated computational time is on the function evaluation.

Numerical Results

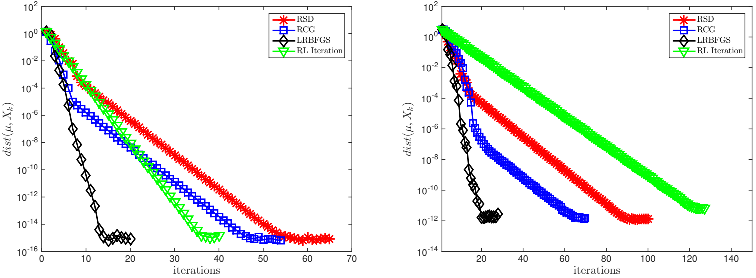


Figure: Evolution of averaged distance between current iterate and the exact Karcher mean with respect to time and iterations with $k = 100$ (the number of matrices) and $n = 3$ (the size of matrices); Left: $1 \leq \kappa(A_i) \leq 200$; Right: $10^3 \leq \kappa(A_i) \leq 2 \cdot 10^6$

Numerical Results

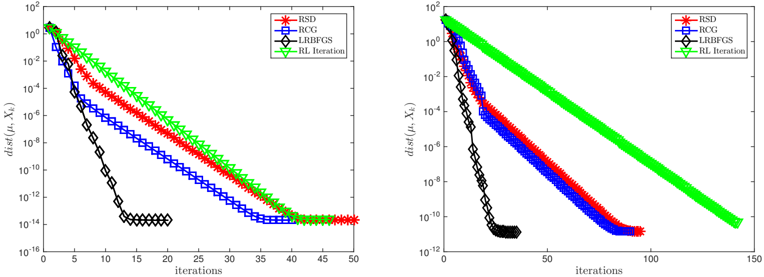


Figure: Evolution of averaged distance between current iterate and the exact Karcher mean with respect to time and iterations with $k = 30$ (the number of matrices) and $n = 100$ (the size of matrices); Left: $1 \leq \kappa(A_i) \leq 20$; Right: $10^4 \leq \kappa(A_i) \leq 2 \cdot 10^6$

Dictionary Learning for SPD matrices

The subproblem: given R, find $\mathbf{B}$.

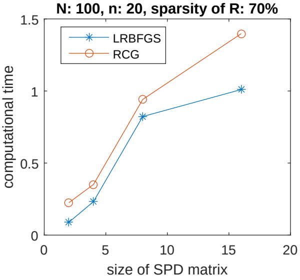
$$\min_{\mathbf{B} \in \mathcal{M}_n^d} \frac{1}{2} \sum_{i=1}^{N} \mathrm{dist}^2(X_i, \mathbf{B} r_i) + \mathrm{trace}\,(\mathbf{B}).$$

- Similar techniques, i.e., implementations for vector transport, retraction, function and gradient evaluation, can be applied;
- The dominated cost is on the function evaluations;
- The cost function is nonconvex;
- We set the initial iterate by $\mathbf{X}_0 = \mathbf{X}\left(R^{\dagger}\right)_+$, where $\mathbf{X}$ is a tensor whose i-th slice is X_i, † denotes the psudo-inverse and M_+ denotes a matrix forming by positive entries of M.

Numerical Results

Artificial tests:

- N: number of training points in $\mathbf{X}$
- n: number of atoms in dictionary $\mathbf{B}$
- d: size of SPD matrices
- R: the representation matrix



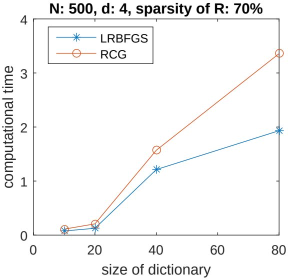


Figure: An average of 50 random runs for various parameter settings.

Dictionary Learning for SPD matrices

The subproblem: given $\mathbf{B}$, find R.

$$\min_{R=[r_1,\ldots,r_N]\in\mathbb{R}_+^{n\times N}} \frac{1}{2}\sum_{i=1}^{N}\left(\mathrm{dist}^2(X_i,\mathbf{B}r_i)+\|r_i\|_1\right).$$

The domain $\mathbb{R}_+^{n\times N}$ is NOT a manifold. A Riemannian optimization-like idea can be applied.

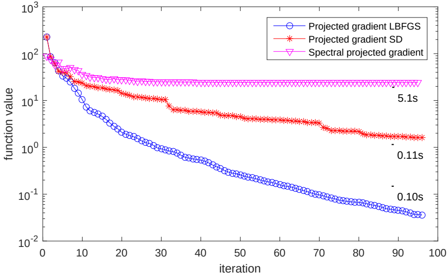


Figure: An representative result. $N = 100$, $d = 5$, $n = 20$

Summary

- Introduced the framework of Riemannian optimization and the state-of-the-art Riemannian algorithms
- Used applications to show the importance of Riemannian optimization
- Showed the performance of Riemannian optimization by geometric mean and dictionary learning of SPD matrices

Thanks!

References I

T. Ando, C. K. Li, and R. Mathias.
Geometric means.
Linear Algebra and Its Applications, 385:305–334, 2004.

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.

J. F. Cardoso and A. Souloumiac.
Blind beamforming for non-gaussian signals.
IEE Proceedings F Radar and Signal Processing, 140(6):362, 1993.

A. Cherian and S. Sra.
Riemannian dictionary learning and sparse coding for positive definite matrices.
CoRR, abs/1507.02772, 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.

Q. Rentmeesters and P.-A. Absil.
Algorithm comparison for karcher mean computation of rotation matrices and diffusion tensors.
19th European Signal Processing Conference (EUSIPCO 2011), (Eusipco):2229–2233, 2011.

F. J. Theis and Y. Inouye.
On the use of joint diagonalization in blind signal processing.
2006 IEEE International Symposium on Circuits and Systems, (2):7–10, 2006.

References II

B. Vandereycken.
Low-rank matrix completion by Riemannian optimization—extended version.
SIAM Journal on Optimization, 23(2):1214–1236, 2012.

X. Yuan, W. Huang, P.-A. Absil, and K. A. Gallivan.
A riemannian limited-memory bfgs algorithm for computing the matrix geometric mean.
Technical Report UCL-INMA-2015.12, U.C.Louvain, 2015.