Riemannian BFGS methods and its Applications

Wen Huang

April 5, 2016

Joint work with:

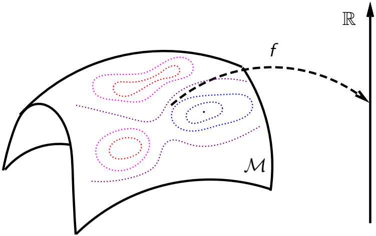
- Pierre-Antoine Absil, Professor of Mathematical Engineering, *Université catholique de Louvain*
- Kyle A. Gallivan, Professor of Mathematics, *Florida State University*
- Anuj Srivastava, Professor of Statistics, *Florida State University*

Riemannian Optimization

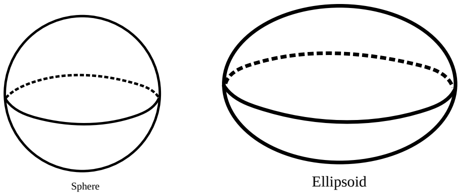
Problem: Given $f(x) : \mathcal{M} \to \mathbb{R}$, solve

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

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



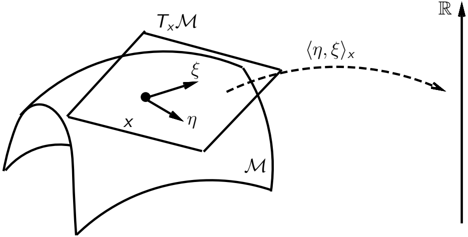
Examples of Manifolds



- 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.



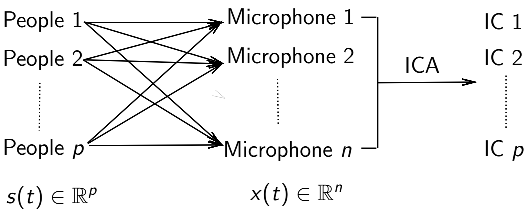
Applications

Two 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, CS15]
- Elastic shape analysis of curves [SKJJ11, HGSA15]

Application: Independent Component Analysis

Cocktail party problem



- 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

$$\begin{array}{cc} & \begin{array}{cccccc} \text{Movie 1} & \text{Movie 2} & & & & \text{Movie } n \end{array} \\ \begin{array}{c}\text{User 1}\\ \text{User 2}\\ \\ \text{User } m\end{array} & \begin{pmatrix} & 1 & & & 4 & \\ 3 & 5 & & & 4 & \\ & & 5 & 1 & & \\ & 2 & & & 5 & 3 \end{pmatrix}\end{array}$$

Rate matrix M

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

Application: Matrix Completion Problem

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

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

where $A_1, \ldots, A_k$ are SPD matrices, and $G(\cdot, \ldots, \cdot)$ 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

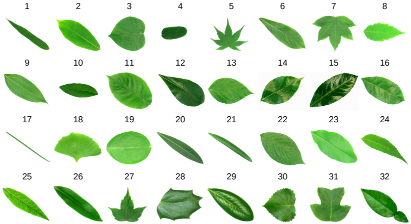
One geometric mean is the Karcher mean of the manifold of SPD matrices with the affine invariant metric, i.e.,

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

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: Elastic Shape Analysis of Curves



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

Application: Elastic Shape Analysis of Curves

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

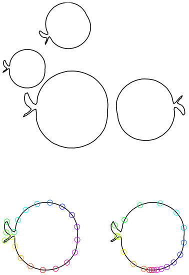
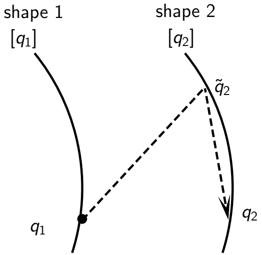


Figure: All are the same shape.

Application: Elastic Shape Analysis of Curves



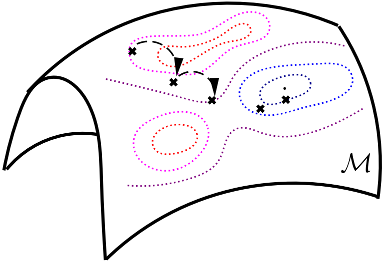
- Optimization problem $\min_{q_2 \in [q_2]} \mathrm{dist}(q_1, q_2)$ is defined on a Riemannian manifold
- Computation of a geodesic between two shapes
- Computation of Karcher mean of a population of shapes

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

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 a Euclidean line search-based optimization algorithm:

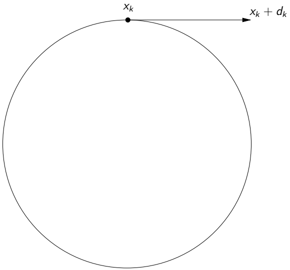
$$x_{k+1} = x_k + \alpha_k d_k \; = x_k - \alpha_k \mathcal{B}_k^{-1} \nabla f(x_k).$$

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

- Steepest descent: $\mathcal{B}_k = \mathrm{id}$;
- Newton's method: $\mathcal{B}_k = \nabla^2 f(x_k)$.

Objects

- Direction: d_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

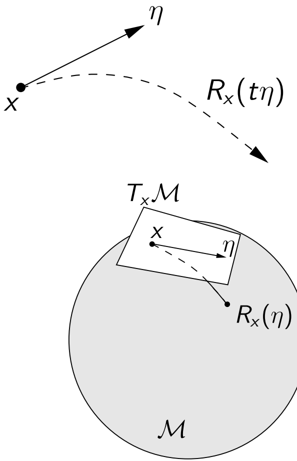
The Riemannian Hessian of f at x is a symmetric linear operator from $T_x M$ to $T_x M$ defined as

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

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

Riemannian Line Search-based Methods

Riemannian Optimization Algorithm

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

- Riemannian steepest descent [AMS08]: $\eta = -\operatorname{grad} f(x)$
- Riemannian Newton [AMS08]: $\eta = -\operatorname{Hess} f(x)^{-1} \operatorname{grad} f(x)$

Quasi-Newton Methods

Motivations:

- Steepest descent: $\mathcal{B}_k = \mathrm{id}$
  - Converge slowly
- Newton method: $\mathcal{B}_k = \mathrm{Hess}\, f(x_k)$
  - Require the Hessian which may be expensive or unavailable
  - Search direction may not be descent
- Quasi-Newton methods: $\mathcal{B}_k$ is obtained by a recursive formula.
  - Use gradient to approximate the action of the Hessian and therefore accelerate the convergent rate
  - Provide an approach to produce descent direction

Among the quasi-Newton methods, only the BFGS method is considered in this talk.

Existing Riemannian BFGS methods

- Euclidean BFGS search direction:

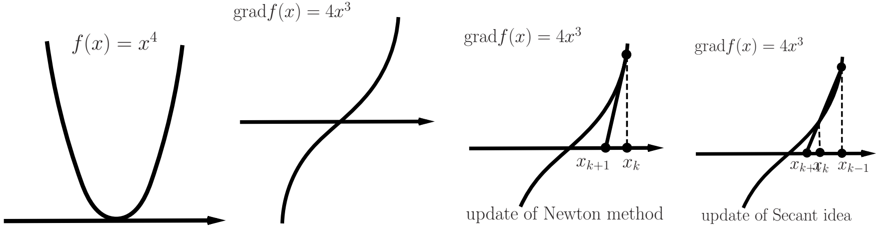
$$d_k = -B_k^{-1} \operatorname{grad} f(x_k),$$

where $B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$, $s_k = x_{k+1} - x_k$, and $y_k = \operatorname{grad} f(x_{k+1}) - \operatorname{grad} f(x_k)$.

- Existing Riemannian BFGS methods
  - Brace and Manton [BM06] / Savas and Lim [SL10]: Riemannian BFGS methods, Grassmann manifold
  - Qi [Qi11]: geodesic
  - Ring and Wirth [RW12] / Huang et. al. [HGA15]: retraction

Secant Condition

An 1 dimension example to show the idea of secant condition.



- Newton: $x_{k+1} = x_k - (\mathrm{Hess}\, f(x_k))^{-1}\, \mathrm{grad}\, f(x_k)$
- Secant: $x_{k+1} = x_k - B_k^{-1}\, \mathrm{grad}\, f(x_k)$,
  $B_k(x_k - x_{k-1}) = \mathrm{grad}\, f(x_k) - \mathrm{grad}\, f(x_{k-1})$

Euclidean and Riemannian Secant Conditions

Secant condition provides us an idea to approximate the action of Hessian.

- Euclidean:

$$\mathrm{grad}\, f(x_{k+1}) - \mathrm{grad}\, f(x_k) = B_{k+1}(x_{k+1} - x_k).$$

- Riemannian:
  - $x_{k+1} - x_k$ can be replaced by $R_{x_k}^{-1}(x_{k+1})$
  - $\mathrm{grad}\, f(x_{k+1})$ and $\mathrm{grad}\, f(x_k)$ are on different tangent space. A method of comparing tangent vectors in different tangent space is required.

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

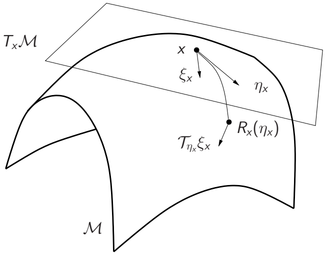


Figure: Vector transport.

Riemannian Secant Conditions

In the Riemannian setting, a naive secant condition is

$$\mathrm{grad}\, f(x_{k+1}) - \mathcal{T}_{\xi_k} \mathrm{grad}\, f(x_k) = B_{k+1} \mathcal{T}_{\xi_k} \xi_k,$$

where $\xi_k = R_{x_k}^{-1}(x_{k+1})$.

It is not clear whether this secant condition is sufficient to give a well-defined and convergent Riemannian BFGS method.

Riemannian Secant Conditions

- The Euclidean secant condition can be written as

$$y_k = B_{k+1} s_k \text{ or } y_k^T = s_k^T B_{k+1}$$

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

- Riemannian secant condition can be

$$\mathfrak{y}_k = \mathcal{B}_{k+1} \mathfrak{s}_k \text{ or } \mathfrak{y}_k^\flat = \mathfrak{s}_k^\flat \mathcal{B}_{k+1},$$

where $\eta_x^\flat : \mathrm{T}_x \mathcal{M} \to \mathbb{R} : \xi_x \mapsto g_x(\eta_x, \xi_x)$.

- $\mathfrak{y}_k$? and $\mathfrak{s}_k$?
- Three Riemannian BFGS secant conditions are discussed [Qi11, RW12, HGA15]

Riemannian Secant Conditions

The secant condition of Qi [Qi11] uses

$$\mathfrak{y}_k = \mathcal{B}_{k+1}\mathfrak{s}_k$$

and chooses

$$\mathfrak{y}_k = \mathrm{grad}\, f(x_{k+1}) - P_{\gamma_k}^{1\leftarrow 0}\,\mathrm{grad}\, f(x_k) \text{ and } \mathfrak{s}_k = (P_{\gamma_k}^{1\leftarrow 0}\,\mathrm{Exp}_{x_k}^{-1} x_{k+1}),$$

where Exp is a particular retraction, called the exponential mapping and P is a particular vector transport, called the parallel translation.

Riemannian Secant Conditions

The secant condition of Ring and Wirth [RW12] uses

$$\mathfrak{y}_k^\flat = \mathfrak{s}_k^\flat \mathcal{B}_{k+1}$$

and chooses

$$\mathfrak{y}_k^\flat = (\mathrm{grad}\, f(x_{k+1})^\flat \mathcal{T}_{R_{\xi_k}} - \mathrm{grad}\, f(x_k)^\flat)\mathcal{T}_{S_{\xi_k}}^{-1} \text{ and } \mathfrak{s}_k^\flat = (\mathcal{T}_{S_{\xi_k}} \xi_k)^\flat$$

where $\xi_k = R_{x_k}^{-1}(x_{k+1})$ and $\mathcal{T}_R$ is differentiated retraction of R, i.e., $\mathcal{T}_{R_{\eta_x}} \zeta_x = \frac{d}{dt} R_x(\eta_x + t\zeta_x)|_{t=0}$.

Riemannian Secant Conditions

In [HGA15], we use

$$\mathfrak{y}_k = \mathcal{B}_{k+1}\mathfrak{s}_k$$

and choose

$$\mathfrak{y}_k = \mathrm{grad}\, f(x_{k+1})/\beta_k - \mathcal{T}_{S_{\xi_k}} \mathrm{grad}\, f(x_k) \text{ and } \mathfrak{s}_k = \mathcal{T}_{S_{\xi_k}} \xi_k,$$

where $\xi_k = R_{x_k}^{-1}(x_{k+1})$ and $\beta_k = \|\xi_k\| / \|\mathcal{T}_{R_{\xi_k}} \xi_k\|$, and $\mathcal{T}_S$ is an isometric vector transport that satisfies $\mathcal{T}_{S_\xi}\xi = \beta \mathcal{T}_{R_\xi}\xi$.

Euclidean BFGS

$$\min_{B} \|B^{-1} - B_k^{-1}\|_{W_H}$$
$$\text{s.t. } B = B^T, Bs_k = y_k$$

where W_H is any positive definite matrix satisfying $W_H s_k = y_k$ and $\|A\|_{W_H} = \|W_H^{1/2} A W_H^{1/2}\|_F$.

The solution is

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

This is called Broyden-Fletcher-Goldfarb-Shanno(BFGS) update.

Riemannian BFGS

$$\min_{\mathcal{B}} \|\mathcal{B}^{-1} - \tilde{\mathcal{B}}_k^{-1}\|_{W_{\mathcal{H}}}$$

$$\text{s.t. } \mathcal{B} \text{ is a self-adjoint operator}, \mathcal{B}\mathfrak{s}_k = \mathfrak{y}_k$$

where $\tilde{\mathcal{B}}_k = \mathcal{T}_{S_{\xi_k}} \circ \mathcal{B}_k \circ \mathcal{T}_{S_{\xi_k}}^{-1}$, $W_{\mathcal{H}}$ is any positive definite matrix satisfying $W_{\mathcal{H}} s_k = y_k$, $\|A\|_{W_{\mathcal{H}}} = \|W_{\mathcal{H}}^{1/2} G^{1/2} A G^{-1/2} W_{\mathcal{H}^{1/2}}\|_F$, and G is the matrix expression of the metric.

The solution is

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

Line Search Riemannian BFGS method

(1) Given initial x_0 and self-adjoint positive definite $\mathcal{B}_0$. Let $k = 0$.

(2) Obtain search direction by $\eta_k = -\mathcal{B}_k^{-1} \operatorname{grad} f(x_k)$

(3) Set next iterate $x_{k+1} = R_{x_k}(\alpha_k \eta_k)$, where α_k is set to satisfy the Wolfe conditions

$$f(x_{k+1}) \leq f(x_k) + c_1 \alpha_k g(\operatorname{grad} f(x_k), \eta_k), \tag{1}$$

$$\frac{d}{dt} f(R_{x_k}(t\eta_k))|_{t=\alpha_k} \geq c_2 \frac{d}{dt} f(R_{x_k}(t\eta_k)|_{t=0}. \tag{2}$$

where $0 < c_1 < 0.5 < c_2 < 1$.

(4) Use update formula to obtain $\mathcal{B}_{k+1}$.

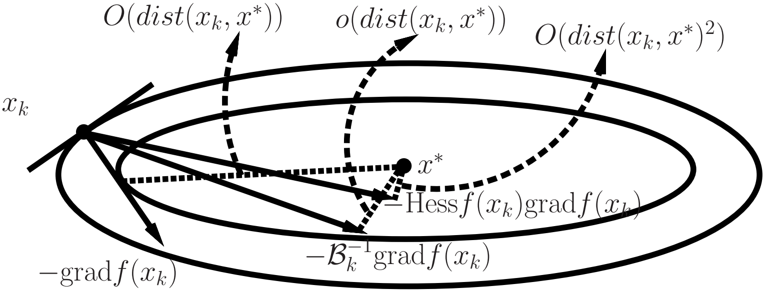
(5) If not converge, then $k \leftarrow k + 1$ and go to Step 2.

Convergence rate

1. Converges superlinearly:

$$\lim_{k\to\infty} \frac{dist(x_{k+1}, x^*)}{dist(x_k, x^*)} = 0$$

2. Step size $\alpha_k = 1$ satisfies the Wolfe conditions eventually.



Limited-memory RBFGS

Riemannian BFGS requires computing $\tilde{\mathcal{B}}_k = \mathcal{T}_{S_{\xi_k}} \circ \mathcal{B}_k \circ \mathcal{T}_{S_{\xi_k}}^{-1}$.

- Explicit form of $\mathcal{T}_S$ may not exist.
- Even though it exists, matrix multiplications or matrix-vector multiplications may be needed.

Limited-memory

- Similar to Euclidean case, it requires less memory.
- It avoids the requirement of explicit form of $\mathcal{T}_S$.

Construct $\mathcal{T}_S$

How to construct $\mathcal{T}_S$ satisfying the locking condition

$$\mathcal{T}_{S_\xi} \xi = \beta \mathcal{T}_{R_\xi} \xi, \quad \beta = \frac{\|\xi\|}{\|\mathcal{T}_{R_\xi} \xi\|},$$

for all $\xi \in \mathrm{T}_x \mathcal{M}$.

- Method 1: Modifying an existing isometric vector transport
- Method 2: Construct $\mathcal{T}_S$ when a smooth function of building orthonormal basis of tangent space is known.
- Both ideas use Householder reflection twice.
- Method 3: Given an isometric vector transport $\mathcal{T}_S$, a retraction is obtained by solving $\frac{d}{dt} R_x(t\eta_x) = \mathcal{T}_{S_{t\eta_x}} \eta_x$. In some cases, the closed form of the solution exists.

The Joint Diagonalization Problem

- The compact Stiefel manifold: $\mathrm{St}(p, n) = \{X \in \mathbb{R}^{n \times p} | X^T X = I_p\}$.
- The joint diagonalization problem in independent component analysis (JD) [TCA09]

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

where C_i are known symmetric matrices and $\mathrm{diag}(M)$ is a vector formed by diagonal entries of a matrix M.

Implementations

- (JD) The C_i matrices are selected as $C_i = \mathrm{diag}(n, n-1, \dots, 1) + 0.1(R_i + R_i^T)$, where the elements of $R_i \in \mathbb{R}^{n \times n}$ are independently drawn from the standard normal distribution.
- The line search algorithm is [DS83, Algorithm A6.3.1mod]. c_1 and c_2 in the Wolfe conditions are 10^{-4} and 0.999.
- Stopping criterion requires norm of final gradient over the norm of initial gradient to be less than 10^{-7}.

Parameters

- $n = 12$, $p = 6$.
- Retraction: $R_X(\eta) = \mathrm{qf}(X + \eta)$, where qf denotes the Q factor of the QR decomposition with nonnegative elements on the diagonal of R.
- Vector transport:
  - Vector transport by parallelization (essentially an identity).
  - Use Householder reflection twice.

Comparisons of RBFGS methods in [RW12] and [HGA15]

Table: An average of 100 random runs of RWRBFGS and RBFGS.

N	Method	iter	nf	ng	nV	t (millisecond)
128	RBFGS	99	113	100	198	21.55
128	RWRBFGS	104	121	107	209	20.16
256	RBFGS	101	115	102	201	34.47
256	RWRBFGS	106	125	110	215	34.57
512	RBFGS	102	117	103	204	62.82
512	RWRBFGS	106	125	110	215	62.80

- N: # of covariance matrices
- iter: # of iterations
- nf: # of function evaluations
- ng: # of gradient evaluations
- nV: # of vector transport
- t: the computational time

RBFGS in [HGA15] relaxes the requirement of differentiated retraction without losing efficiency.

Elastic Shape Analysis of Curves

- Elastic shape analysis invariants
    - Rescaling
    - Translation
    - Rotation
    - Reparameterization (difficult)

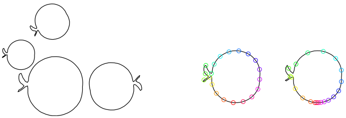
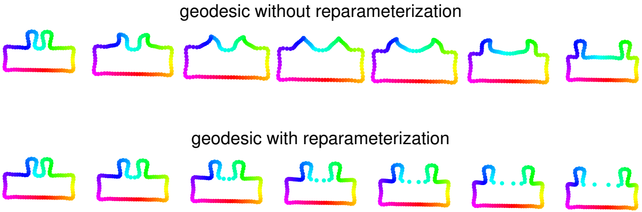


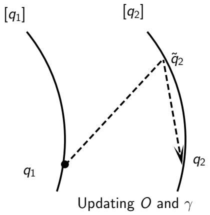
Figure: All are the same shape.

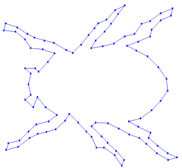


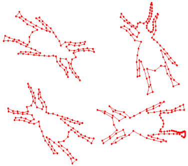
Best Rotation and Reparameterization

$$(O_*, \gamma_*) = \underset{(O,\gamma)\in \mathrm{SO}(n)\times\Gamma}{\operatorname{argmin}} \operatorname{dist}_{I_n}(q_1, O(q_2, \gamma)),$$

where $\mathrm{SO}(n)$ is the orthogonal group and Γ is the set of absolutely continuous bijection from $\mathbb{S}^1$ to $\mathbb{S}^1$.







Optimization Algorithms

- Coordinate Descent Method: Optimize rotation and reparameterization alternately.
  - Rotation: Procrustes problem solved using SVD
  - Reparameterization: $O(N)$ runs of Dynamic programming (DP) with slope constraints, where N is the number of points in the curves
  - Complexity is $O(N^3)$ per iteration.
- Riemannian Method
  - Domain: $\mathrm{SO}(n) \times \mathbb{R} \times \mathbb{S}^{\mathbb{L}^2}$, where $\mathbb{S}^{\mathbb{L}^2}$ is the unit sphere in $\mathbb{L}^2$.
  - Complexity is $O(N)$ per iteration.
- A global minimizer is desired

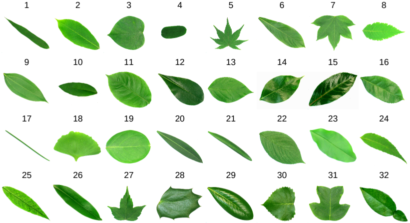
Examples (by Limited-memory Riemannian BFGS Method)

Figure: Applying best rotation and reparameterization to one of the curves. The colors indicate corresponding points on the two curves.

Data Sets

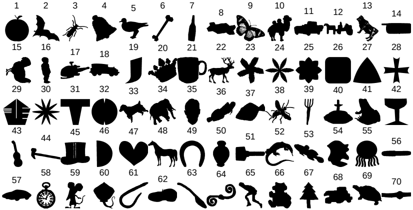
Flavia leaf dataset [WBX+07]

- 1907 images of leaves
- 32 species



MPEG-7 dataset [Uni]

- 1400 binary images
- 70 clusters



- Boundary curves: BWBOUNDARIES function in Matlab
- 100 points in $\mathbb{R}^2$ used for each boundary

Known $\gamma_T^{-1}(t) = (t + \sin(2\pi t))/(4\pi)$

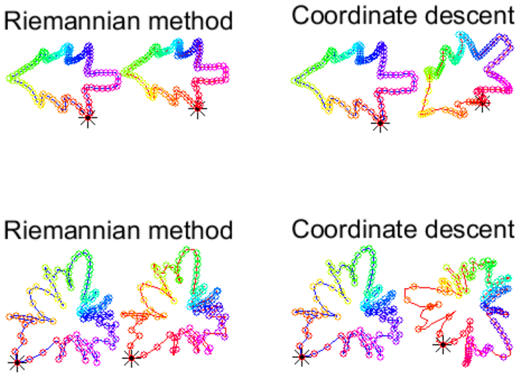


Figure: Apply random rotation and given γ_T^{-1} to a given shape to obtain the second shape. For the tested 1020 shapes, coordinate descent method may not find a global minimizer.

One Nearest Neighbor Results

- The 1NN metric, μ, computes the percentage of points whose nearest neighbor are in the same cluster, i.e.,

$$\mu = \frac{1}{n}\sum_{i=1}^{n} C(i), \quad C(i) = \begin{cases} 1 & \text{if point } i \text{ and its nearest neighbor} \\ & \text{are in the same cluster;} \\ 0 & \text{otherwise.} \end{cases}$$

	t_{ave}(F)	1NN(F)	t_{ave}(M)	1NN(M)
Riemannian LBFGS	0.088	89.51%	0.181	97.79%
Coordinate descent	0.897	87.52%	0.908	96.79%

Summary

- Introduced the framework of Riemannian optimization and the state-of-the-art Riemannian algorithms
- Used applications to show the importance of Riemannian optimization
- Used two experiments to show the numerical performance of Riemannian algorithms.

Thank you

Thank you!

References I

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

P.-A. Absil, R. Mahony, and R. Sepulchre.
Optimization algorithms on matrix manifolds.
Princeton University Press, Princeton, NJ, 2008.

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, pages 1735–1738, 2006.

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.

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.

J. E. Dennis and R. B. Schnabel.
Numerical methods for unconstrained optimization and nonlinear equations.
Springer, New Jersey, 1983.

References II

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.

Wen 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, 2015.
DOI:10.1007/s10851-015-0606-8.

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. 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.

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 III

A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn.
Shape analysis of elastic curves in Euclidean spaces.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(7):1415–1428, September 2011. doi:10.1109/TPAMI.2010.184.

B. Savas and L. H. Lim.
Quasi-Newton methods on Grassmannians and multilinear approximations of tensors.
SIAM Journal on Scientific Computing, 32(6):3352–3393, 2010.

F. J. Theis, T. P. Cason, and P.-A. Absil.
Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold.
Proceedings of the 8th International Conference on Independent Component Analysis and Signal Separation, 5441:354–361, 2009.

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.

Temple University.
Shape similarity research project.
www.dabi.temple.edu/ shape/MPEG7/dataset.html.

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

References IV

S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang.
A leaf recognition algorithm for plant classification using probabilistic neural network.
2007 IEEE International Symposium on Signal Processing and Information Technology, pages 11–16, 2007.
arXiv:0707.4289v1.