Riemannian Optimization with an Application in Elastic Shape Analysis

W. Huang[2], K. A. Gallivan[1], A. Srivastava[1] and P.-A. Absil[2]

[1] Florida State University

[2] Université Catholique de Louvain

26 May 2015

Riemannian Optimization

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

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

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

Goal

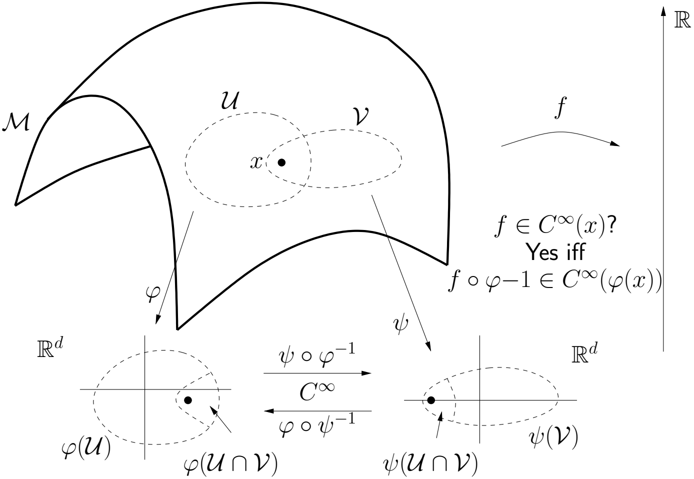
Adapt unconstrained Euclidean algorithms to function on $\mathcal{M}$.

Motivation

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

Riemannian Manifolds

Roughly, a Riemannian manifold is a smooth set with a smoothly-varying inner product on the tangent spaces.



$f \in C^\infty(x)$?

Yes iff

$f \circ \varphi{-}1 \in C^\infty(\varphi(x))$

Noteworthy Manifolds and Applications

- Grassmann manifold

$$\mathrm{Gr}(p,n) = \text{Set of all } p\text{-dimensional subspaces of } \mathbb{R}^n \;\; d = np - p^2$$

- Stiefel manifold

$$\mathrm{St}(p,n) = \{X \in \mathbb{R}^{n\times p} : X^T X = I_p\} \;\; d = np - \frac{p(p+1)}{2}$$

- Oblique manifold

$$\mathbb{R}_*^{n\times p}/\mathcal{S}_{\mathrm{diag}+} \simeq \{Y \in \mathbb{R}_*^{n\times p} : \mathrm{diag}(Y^T Y) = I_p\} = S^{n-1} \times \cdots \times S^{n-1}$$
$$d = (n-1)p^2$$

Noteworthy Manifolds and Applications

- Set of fixed-rank PSD matrices $\mathrm{S}_+(p, n)$. A quotient representation:

$$X \sim Y \Leftrightarrow \exists Q \in O_p : Y = XQ$$

- Set of $m \times n$ matrices with rank p $\mathrm{S}(p, m, n)$. A representation:

$$\mathrm{Gr}(p, n) \times \mathbb{R}^{m \times p}$$

Iterations on the Manifold

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

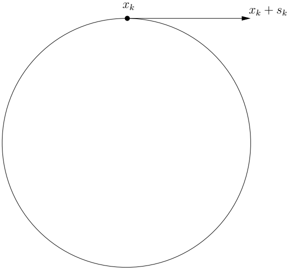
$$x_{k+1} = x_k + \Delta x_k = x_k + \alpha_k s_k \ .$$

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.

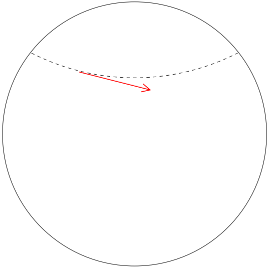
Riemannian Manifolds Provide

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



Tangent Vectors

- The concept of direction is provided by tangent vectors.
- Intuitively, tangent vectors are tangent to curves on the manifold.
- Tangent vectors are an intrinsic property of a differentiable manifold.

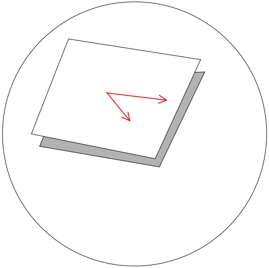


Definition

The tangent space $T_x M$ is the vector space comprised of the tangent vectors at $x \in M$. The Riemannian metric is an inner product on each tangent space.

Tangent Vectors

- The concept of direction is provided by tangent vectors.
- Intuitively, tangent vectors are tangent to curves on the manifold.
- Tangent vectors are an intrinsic property of a differentiable manifold.



Definition

The tangent space $T_x M$ is the vector space comprised of the tangent vectors at $x \in M$. The Riemannian metric is an inner product on each tangent space.

Example Tangent Spaces and Metrics

Stiefel manifold embedded in $\mathbb{R}^{n \times p}$

$$\mathrm{St}(p,n) = \{X \in \mathbb{R}^{n \times p} : X^T X = I_p\}$$

$$V \in T_X \mathrm{St}(p,n) \quad X^T V + V^T X = 0, \quad \text{i.e., } X^T V \text{ is skew symmetric}$$

$$V = XA + X_{\perp} B \quad -A = A^T, \quad \text{i.e., } A \text{ is skew symmetric}$$

Euclidean metric: $g_e(V,V) = trV^T V$

Canonical metric: $g_c(V,V) = trV^T(I - \frac{1}{2}XX^T)V = \frac{1}{2}trA^T A + trB^T B$

Riemannian gradient and Riemannian Hessian

Definition

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

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

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

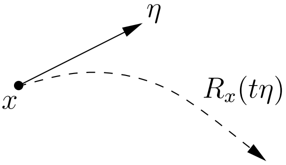
$$\mathrm{Hess}\, f(x) : T_x M \to T_x M : \eta \to \nabla_\eta \mathrm{grad}\, f$$

Retractions

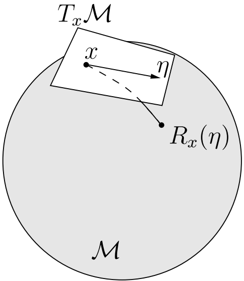
Definition

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
- lifts objective function f from M to T_xM, via the pullback $\hat{f}_x = f \circ R_x$
- defines curves in a direction
- exponential map $\mathrm{Exp}(t\eta)$ defines “straight lines” geodesic

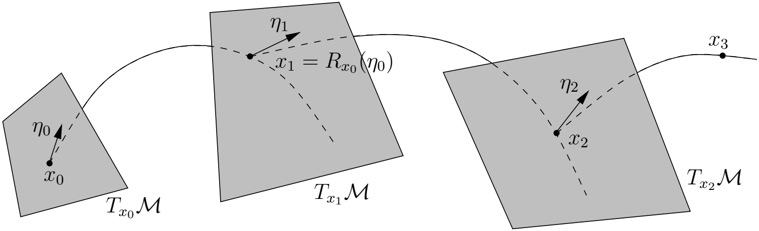


Generic Riemannian Optimization Algorithm

1. At iterate $x \in M$, define $\hat{f}_x = f \circ R_x$.
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: geodesic in the direction $-\mathrm{grad}\, f(x)$
- Newton

Local model-based: series of flat space problems

- Riemannian Trust region (RTR)
- Riemannian Adaptive Cubic Overestimation (RACO)

Retraction and transport-based: information from multiple tangent spaces

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

Basic principles

All/some elements required for optimizing a cost function (M, g):

- an efficient numerical representation for points x on M , for tangent spaces $T_x M$, and for the inner products $g_x(\cdot, \cdot)$ on $T_x M$;
- choice of a retraction $R_x : T_x M \to M$;
- formulas for $f(x), \mathrm{grad}\ f(x)$ and $\mathrm{Hess}\ f(x)$ (or its action);
- formulas for combining information from multiple tangent spaces.

Parallel transport

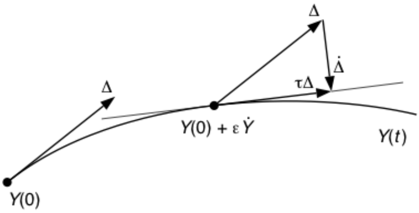


Figure: Parallel transport

- Parallel transport one tangent vector along some curve $Y(t)$.
- It is often along the geodesic $\gamma_\eta(t) : \mathbb{R} \to M : t \to \mathrm{Exp}_x(t\eta_x)$.
- In general, geodesics and parallel translation require solving an ordinary differential equation.

Vector transport

Definition

We define a **vector transport** on a manifold M to be a smooth mapping

$$TM \oplus TM \to TM : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}(\xi_x) \in TM$$

satisfying the following properties for all $x \in M$.

- (Consistency) $\mathcal{T}_{0_x}\xi_x = \xi_x$ for all $\xi_x \in T_xM$;
- (Linearity) $\mathcal{T}_{\eta_x}(a\xi_x + b\zeta_x) = a\mathcal{T}_{\eta_x}(\xi_x) + b\mathcal{T}_{\eta_x}(\zeta_x)$.
- There exists a retraction R that is associated with $\mathcal{T}$ and satisfies the following illustrated relationship:

Vector transport

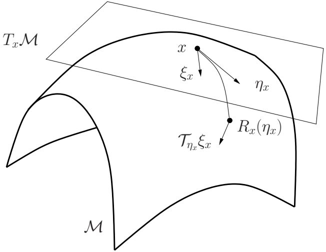


Figure: Vector transport.

Retraction/Transport-based Riemannian Optimization

Benefits

- Increased generality does not compromise the important theory
- Can easily employ classical optimization techniques
- Less expensive than or similar to previous approaches
- May provide theory to explain behavior of algorithms in 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 CG 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 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`

Many people Application interests start to increase noticeably

Other Applications

- Large-scale Generalized Symmetric Eigenvalue Problem and SVD (RTR, Absil, Baker, Gallivan 2004-08)
- Blind source separation on both Orthogonal group and Oblique manifold (RTR, Absil and Gallivan 2006, Selvan et al. 2012)
- Low-rank approximate solution symmetric positive definite Lyapanov $AXM + MXA = C$ (RTR, Vandereycken and Vanderwalle 2009)
- Best low-rank approximation to a tensor (RTR, Ishteva, Absil, Van Huffel, De Lathauwer 2010)
- rotation synchronization (RTR, Absil, Boumal, 2012)
- ICA (RTR, RQN, LM-RQN, Absil, Gallivan, Huang 2013)
- rotation synchronization (RTR, RQN, LM-RQN, Absil, Gallivan, Huang 2013)
- phase retrieval (RQN, LM-RQN, Gallivan, Huang, Zhang 2014)
- graph similarity and community detection (Gallivan, Huang, Van Dooren, Zhou 2015)

Current UCL/FSU Methods

- Riemannian Steepest Descent
- Riemannian Trust Region Newton: global quadratic convergence
- Riemannian Broyden Family : local superlinear convergence using Riemannian Wolf conditions
- Riemannian Trust Region SR-1: local $(d+1)-$superlinear convergence
- For large problems
    - Limited memory RTRSR1
    - Limited memory RBFGS
- Riemannian CG (much more work to do on analysis)

Future Riemannian Work

- relaxing conditions on vector transport, retractions, Wolfe conditions etc.
- 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

Elastic Shape Analysis of Curves

- Elastic shape analysis invariants:
  - Rescaling
  - Translation
  - Rotation
  - Reparametrization
- Younes 1998, and Younes, Michor, Shah, Mumford 2008 [10, 11]
- Square Root Velocity Function framework used (Srivastava, Klassen, Joshi, and Jermyn [7]).
- extensive analysis and application of elastic shape
- much less work on understanding efficient and robust algorithms

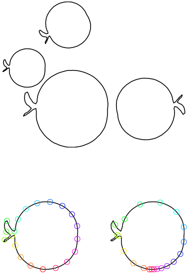


Figure: All are the same shape.

SRVF and Preshape Space

Preshape space, denoted $\mathcal{C}_n$, removes translation and rescaling for $\mathbb{L}_2$.

- A shape is represented by a function $\beta : \mathbb{D} \to \mathbb{R}^n$, where $\mathbb{D}$ is $[0, 1]$ for open curves and unit circle $\mathbb{S}^1$ for closed curves.
- Square Root Velocity (SRV) function of the shape β is

$$q(t) = \begin{cases} \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|_2}} & \text{if } \|\dot{\beta}(t)\|_2 \neq 0; \\ 0 & \text{if } \|\dot{\beta}(t)\|_2 = 0. \end{cases}$$

- Translation is removed by derivative.
- Rescaling is removed by forcing the length of the curve to be 1, i.e., $\int_0^1 \|\dot{\beta}(t)\|_2 dt = \int_0^1 \|q(t)\|_2^2 dt = 1$.
- Preshape spaces (closure condition added for closed curves)

$$\mathcal{C}_n^o = \{q : [0, 1] \to \mathbb{R}^n | \int_0^1 \|q(t)\|_2^2 dt = 1\}$$

$$\mathcal{C}_n^c = \{q : \mathbb{S}^1 \to \mathbb{R}^n | \int_{\mathbb{S}^1} \|q(t)\|_2^2 dt = 1, \int_{\mathbb{S}^1} q(t)\|q(t)\|_2 dt = 0\}$$

Shape Space

Shape space removes rotation and reparameterization. Inherits metric from $\mathbb{L}_2$

$$\mathrm{SO}(n) = \{O \in \mathbb{R}^{n\times n} | O^T O = I_n, \det(O) = 1\}$$
$$\mathrm{SO}(n) \times \mathfrak{l}_n \to \mathfrak{l}_n : (O, q) \to Oq$$

$$\Gamma = \{\gamma : \mathbb{D} \to \mathbb{D} | \gamma \text{ is a diffeomorphism.}\}$$
$$\mathfrak{l}_n \times \Gamma \to \mathfrak{l}_n : (q, \gamma) \to (q \circ \gamma)\sqrt{\dot{\gamma}}$$

$$[q] = \{(O, (q, \gamma)) | O \in \mathrm{SO}(n), \gamma \in \Gamma\}$$

$$\mathfrak{L}_n = \mathfrak{l}_n / \mathrm{SO}(n) \times \Gamma = \{[q] | q \in \mathfrak{l}_n\}.$$

both are isometric group actions. Note: diffeomorphism is $\gamma, \gamma^{-1} \in \mathcal{C}^\infty$

Best Rotation and Reparameterization

An important and basic task is to find the best rotation O_* and reparameterization γ_*.

- Geodesic
- Mean
- Other statistical analysis

Best Rotation and Reparameterization

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

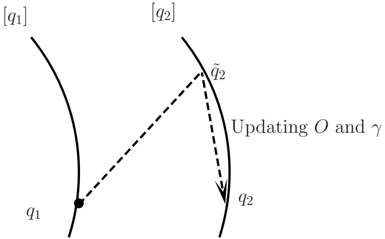


Figure: Align representation of $[q_2]$ with q_1.

Cost Functions

- Minimization problem

$$\min_{O \in \mathrm{SO}(n), \gamma \in \Gamma} \mathrm{dist}_{\mathfrak{l}_n}(Oq_1, (q_2, \gamma)).$$

- Open curve

$$d_{l_n^o}(Oq_1, (q_2 \circ \gamma)\sqrt{\dot{\gamma}}) = \cos^{-1} \langle Oq_1, (q_2 \circ \gamma)\sqrt{\dot{\gamma}} \rangle_{\mathbb{L}^2}$$

$$H^o(O, \gamma(t)) = \int_0^1 \|Oq_1(t) - (q_2 \circ \gamma(t))\sqrt{\dot{\gamma}(t)}\|_2^2 dt \text{ same extreme points}$$

- Closed curve
  - Closed form of preshape space distance is unknown.
  - Preshape (Extrinsic) distance is used.

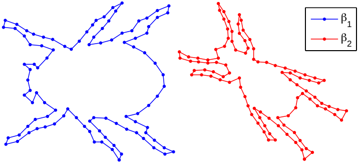
$$H^c(O, \gamma) = \int_{\mathbb{S}^1} \|Oq_1(t) - (q_2 \circ \gamma(t))\sqrt{\dot{\gamma}(t)}\|_2^2 dt$$

Coordinate Descent Method

Optimize rotation and reparameterization alternately.

- Open curves
    - Rotation: Procrustes problem solved using SVD
    - Reparameterization: Dynamic programming (DP) with slope constraints
- Closed curves
    - Choose a point on the closed curve and break it into an open curve
    - Apply coordinate descent method of open curves
    - Compare results for a sufficiently large number of break points

Two Shapes



Coordinate Descent Method

One iteration, denoted CD1, is used in [7].

- Complexity is $O(N^3)$, where N is the number of points in the curves.
- Note rotation and the correspondence of portions of the structures.
- Does iterating more improve results?

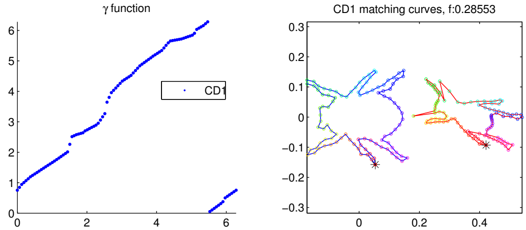
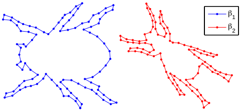


Figure: Results given by CD1

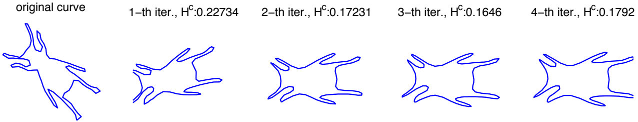
Representations and Implementation Difficulties

Representation Approach 1

- q_1 and q_2 are represented by points.
- Evaluation of (q_2, γ) over iterations on q
- $q_2^{(k+1)} = (q_2^{(k)}, \gamma^{(k)})$ computed on each iteration by evaluation of interpolating function of $q_2^{(k)}$.
- New interpolating function for $q_2^{(k+1)}$ → Shape of q_2 changes.



!



Representations and Implementation Difficulties

- Representation Approach 2
  - q_1 is represented by points and q_2 is represented by a fixed interpolating curve.
- Difficulty: Lack of compuational associativity may not reduce the cost function in practice
- Cost function evaluated in DP uses points $(q_2, \gamma^{(k)})$, and evaluates $((q_2, \gamma^{(k)}), \tilde{\gamma}^{(k+1)})$.
- Next q iterate is obtained using $(q_2, \gamma^{(k)} \circ \tilde{\gamma}^{(k+1)}) = (q_2, \gamma^{(k+1)})$ since fixed interpolation function for q_2
- Cost function values for the two forms of applying $\gamma^{(k+1)}$ can differ

iteration (k)	1	2	3
H^c iterate form	0.390583	0.378312	0.390114
H^c in DP	0.285534	0.248016	0.241679

Table: Computed cost function values. Difference continues growing with k.

Riemannian Approach

- Optimizing H is an optimization problem on $\mathrm{SO}(n) \times \Gamma$.
- Many Riemannian optimization algorithms have been systematically analyzed recently.
    - Riemannian trust-region Newton method (RTR-Newton) [2]
    - Riemannian Broyden family method including BFGS method and its limited-memory version (RBroyden family, RBFGS, LRBFGS) [6, 3, 5]
    - Riemannian trust-region symmetric rank-one update method and its limited-memory version (RTR-SR1, LRTR-SR1) [3, 4]
    - Riemannian Newton method (RNewton) [1]
- See Optimization algorithms on Riemannian manifolds with applications, FSU, Math Dept. [3] for details on analysis, applications and library design

Cost Function of Closed Curves

- Γ^c is represented by $\tilde{\Gamma} \times \mathbb{R}$ where

$$\tilde{\Gamma} = \{\gamma : [0,1] \to [0,1] | \gamma \text{ is diffeomorphism}\}.$$

and the $\tilde{\Gamma} \times \mathbb{R}$ group action on q is defined by

$$(q, (\gamma, m)) = (q(\gamma + m \text{ mod } 1))\sqrt{\dot{\gamma}}, \;\; (\gamma, m) \in \tilde{\Gamma} \times \mathbb{R}.$$

- The cost function on the manifold $\mathrm{SO}(n) \times \mathbb{R} \times \tilde{\Gamma}$ is

$$H^c(O, m, \gamma) = \int_0^1 \|Oq_1(t) - (q_2(\gamma(t) + m \mod 1))\sqrt{\dot{\gamma}(t)}\|_2^2 dt$$

where $\gamma(0) = 0$, $\int_0^1 \dot{\gamma}(t) dt = 1$, $\dot{\gamma} > 0$.

2-norm Sphere

- Optimization on the manifold $\tilde{\Gamma}$ directly has some difficulties
- $\tilde{\Gamma}$ can be replaced with the 2-norm sphere
- Replace the term $\sqrt{\dot{\gamma}(t)}$ in H^c by a function ℓ.
- $\ell > 0$ and $\ell \in \mathbb{S}_{\mathbb{L}_2}$, where $\mathbb{S}_{\mathbb{L}_2} = \{\ell \in \mathbb{L}^2([0,1], \mathbb{R}) | \int_0^1 \ell^2(t) dt = 1\}$.
- A constrained optimization is obtained

$$\min_{O \in \mathrm{SO}(n), m \in \mathbb{R}, \ell \in \mathbb{S}_{\mathbb{L}_2}, \ell > 0} \int_0^1 \|O q_1(t) - q_2(\int_0^t \ell^2(s) ds + m \mod 1) \ell(t)\|_2^2 dt.$$

Barrier Function

- A barrier function can be added to avoid the slope of γ being zero or going to ∞:

$$B(\gamma) = \int_0^1 (\dot{\gamma}(t) + \frac{1}{\dot{\gamma}(t)}) \sqrt{1 + \dot{\gamma}^2(t)} dt = \int_0^1 (\ell^2(t) + \frac{1}{\ell^2(t)}) \sqrt{1 + \ell^4(t)} dt$$

  which satisfies the symmetric property, i.e., $B(\gamma) = B(\gamma^{-1})$.
- The user can control the approach to a slope of 0 or ∞ by adding $\omega B(\gamma)$ to the cost function.

Riemmanian Algorithm

- q_1 is represented by points and q_2 is represented by an interpolating curve.
- Multiple values of m are used based on the variation of angle along the curve.
- Procrustes and DP on a coarse grid give initial ℓ_0 and O_0 for each m.

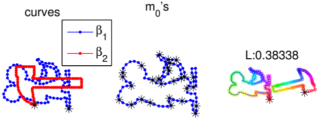
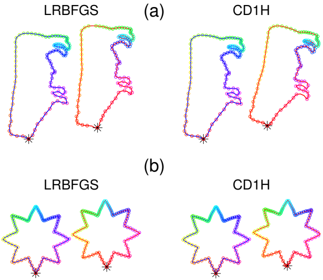


Figure: Choosing initial m for Riemannian algorithms.

Riemmanian Algorithm

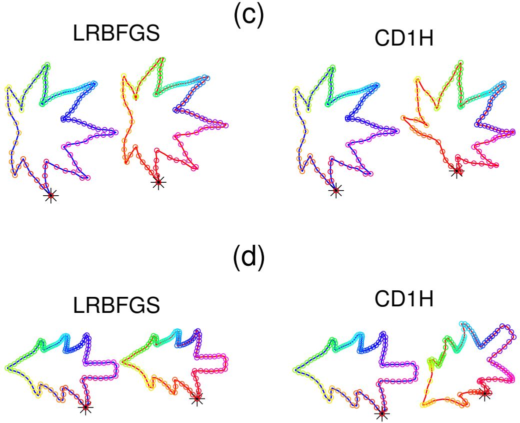
- Improvements
  - Keep the shape of q_2 constant
  - Avoid the problem with computational associativity of group action
  - Computational complexity reduces

Known γ_T: rotation and γ off



CD1 can be improved slightly by using one and half iteration in the coordinate descent method, denoted by CD1H.

Known γ_T: rotation and γ off significantly



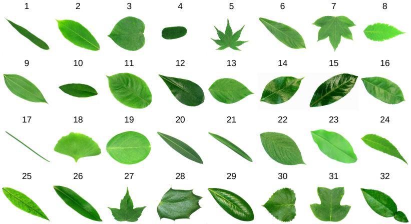
Data Sets

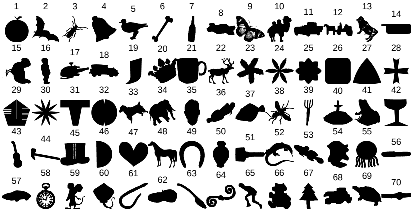
Flavia leaf dataset [9]

- 1907 images of leaves
- 32 species

MPEG-7 dataset [8]

- 1400 binary images
- 70 clusters





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

Representative of Riemannian Algorithm

- Five Riemannian methods are tested.
- 1000 pairs of shape in each data set are used.
- Based on the following table, LRBFGS is chosen to be the representative one.

	RBFGS	LRBFGS	RTR-SR1	LRTR-SR1	RSD	CD1H
L_{ave}(F)	0.16338	0.16182	0.16367	0.16723	0.20665	0.22323
t_{ave}(F)	0.08963	0.07954	0.11603	0.10862	0.06488	0.42895
L_{ave}(M)	0.33214	0.31893	0.33258	0.3418	0.48732	0.51664
t_{ave}(M)	0.19945	0.19318	0.24696	0.23102	0.14373	0.42817

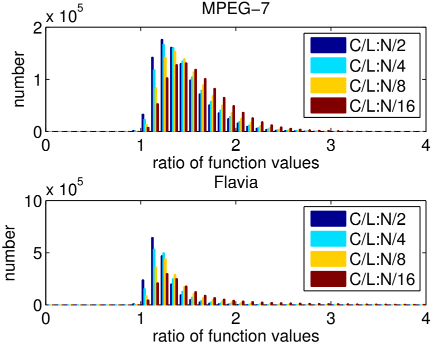
Table: Comparison of Riemannian Methods for representative sets from the Flavia and MPEG-7 datasets: average time per pair (t_{ave}) in seconds and average cost function per pair (L_{ave}).

Comparisons of LRBFGS and CD1H

Test Environment and Tests Performed

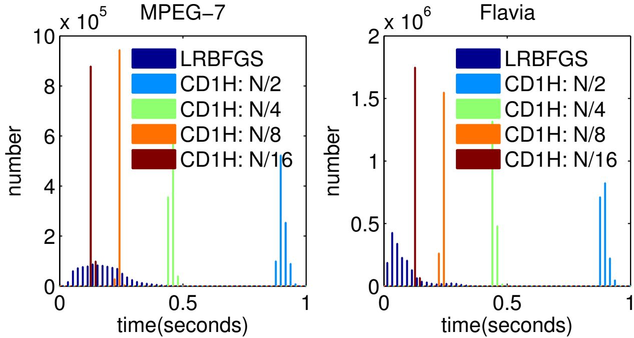
- Environment
  - All codes are written in C++ using BLAS and LAPACK, compiled with g++
  - Performs on a 64 bit Ubuntu system with 3.6 GHz CPU (Intel (R) Core (TM)). The code can be found on `www.math.fsu.edu/~whuang2/papers/RORCESA.htm`.
- Experiments
  - Compute all pairwise distances in the Flavia and MPEG-7 respectively
  - For CD1H method, the results of the breaking points chosen to be every 2, 4, 8, 16 point are reported.

Cost Function Ratios



- Percent of Flavia pairs reduced 99.2%, 99.4%, 99.7% and 99.8% for $N/i, i = 2, 4, 8, 16$
- Percent of MPEG-7 pairs reduced 99.7%, 99.8%, 99.9% and 99.9% for $N/i, i = 2, 4, 8, 16$

Computational Time Ratios



- LRBFGS computation time adjusts based on the complexity of shape based on number of m points.
- CD1H is essentially constant due to simple choice of number of break points.
- LRBFGS generically faster even with same number of initial points.

One Nearest Neighbor Results

- The quality of the extrinsic distance computations is assessed by the one nearest neighbor (1NN) metric
- The 1NN metric, μ, computes the percentage of points whose nearest neighbor are in the same cluster, i.e.,

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

One Nearest Neighbor Results

	LRBFGS	CD1H			
		$N/16$	$N/8$	$N/4$	$N/2$
t_{ave}(F)	0.088	0.126	0.233	0.448	0.897
1NN(F)	89.51%	79.55%	83.01%	85.95%	87.52%
t_{ave}(M)	0.181	0.127	0.236	0.454	0.908
1NN(M)	97.79%	90.29%	93.86%	96.07%	96.79%

Table: The average computation time and 1NN metric of LRBFGS and CD1H with break points chosen to be every 2, 4, 8 and 16 points for Flavia (F) and MPEG-7 (M) data sets.

Conclusion and Future Shape Work

- Conclusion
  - CD with multiple iterations unreliable; composition unreliable
  - CD1/CD1H may not be able to find an accurate solution
  - Riemannian approach is faster, better results, and more robust for more complicated shapes than CD1
- Future work
  - Analysis of effects of discretization on accuracy
  - Test the influence of the accuracy of distance in other shape analyses, e.g., geodesic, means
  - Combination with more robust global reparameterization optimization of Klassen et al.

References I

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

C. G. Baker.
Riemannian manifold trust-region methods with applications to eigenproblems.
PhD thesis, Florida State University, 2008.

W. Huang.
Optimization algorithms on Riemannian manifolds with applications.
PhD thesis, Florida State University, 2013.

W. Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian symmetric rank-one trust-region method.
Mathematical Programming, 2014.

References II

W. Huang, K. A. Gallivan, and P.-A. Absil.
A Broyden class of quasi-Newton methods for Riemannian optimization.
Submitted for publication, 2014.

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.

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.

Temple University.
Shape similarity research project.

References III

- 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.

- L. Younes.
  Computable elastic distances between shapes.
  SIAM Journal on Applied Mathematics, 58(2):565–586, April 1998.

- L. Younes, P. Michor, J. Shah, and D. Mumford.
  A metric on shape space with explicit geodesics.
  Rendiconti Lincei - Matematica e Applicazioni, 9(1):25–57, 2008.