Optimization Algorithms on Riemannian Manifolds with Applications

Wen Huang

Coadvisor: Kyle A. Gallivan — Florida State University
Coadvisor: Pierre-Antoine Absil — Catholic University of Louvain

November 5, 2013

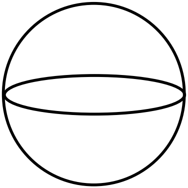
Problem Statements

- Finding an optimum of a real-valued function f on a Riemannian manifold, i.e.,

$$\min f(x), x \in \mathcal{M}$$

- Finite dimensional manifold
- Roughly speaking, a manifold is a set endowed with coordinate patches that overlap smoothly, e.g.,

sphere: $\{x \in \mathbb{R}^n | \|x\|_2 = 1\}$.



Motivations

Optimization on manifolds is used in many areas [AMS08].

- Numerical linear algebra
- Signal processing
- Data mining
- Statistical image analysis

Frameworks of Optimization

- Line search optimization methods
  - Find a search direction,
  - Apply a line search algorithm and obtain a next iterate.
- Trust region optimization methods
  - Build a local model that approximates the objective function f,
  - Optimize the local model and obtain a candidate of next iterate,
  - If the local model is close to f, then accept the candidate to be next iterate, otherwise, reject the candidate,
  - Update the local model.

Existing Euclidean Optimization Algorithms

There are many algorithms developed for problems in Euclidean space. (see e.g. [NW06]) e.g.,

- Newton-based (requires gradient and Hessian)
- gradient-based (requires gradient only)
  - Steepest descent
  - Quasi-Newton
    - Restricted Broyden Family (BFGS, DFP)
    - Symmetric rank-1 update
- These ideas can be combined with line search or trust region strategies.

Existing Riemannian Optimization Algorithms

The algorithmic and theoretical work on Riemannian manifolds is quite limited.

- Trust region with Newton-Steihaug CG (C. G. Baker [Bak08])
- Riemannian BFGS (C. Qi [Qi11])
- Riemannian BFGS (W. Ring and B. Wirth [RW12])

Quadratic:

$$\lim_{k\to\infty} \frac{dist(x_{k+1}, x^*)}{dist(x_k, x^*)^2} < \infty$$

Superlinear:

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

Linear:

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

Framework of Line Search Optimization Methods

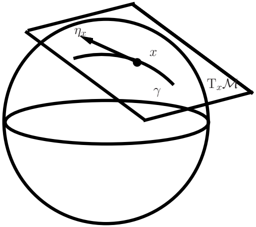
- Line search optimization methods on Euclidean space

$$x_+ = x + \alpha d,$$

  where d is a descent direction and α is a step size.
- Cannot apply to problems on Riemannian manifold directly
  - direction?
  - addition?
- Riemannian concepts can be found in [O'N83, AMS08].

Tangent Space

- γ is a curve on $\mathcal{M}$. The tangent vector shows the direction along γ at x, for which is $\gamma'(0)$, where $\gamma(0) = x$.
- Tangent space at x is the set of all tangent vectors(directions) at x, denoted by $\mathrm{T}_x \mathcal{M}$.
- Tangent space is a linear space.



Riemannian Metric

A Riemannian metric g is defined on each $\mathrm{T}_x\,\mathcal{M}$ as an inner product $g_x : \mathrm{T}_x\,\mathcal{M} \times \mathrm{T}_x\,\mathcal{M} \to \mathbb{R}$. A Riemannian manifold is the combination $(\mathcal{M}, g)$. This results in:

- angle between directions and length of directions
- distance:

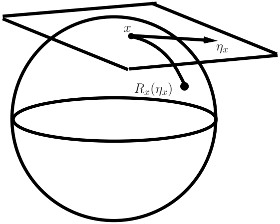
$$d(x, y) = \inf_{\gamma}\{\int_0^1 \|\dot{\gamma}(t)\|_{g_{\gamma(t)}} dt\},$$

  where γ is a curve on $\mathcal{M}$ with $\gamma(0) = x$ and $\gamma(1) = y$.
- neighborhood:

$$\mathcal{B}_\delta(x) = \{y \in \mathcal{M} : d(x, y) < \delta\}.$$

Retraction

Retraction is a mapping from a tangent vector to a point on $\mathcal{M}$, denoted by $R_x(\eta_x)$ where $x \in \mathcal{M}$ and $\eta_x \in \mathrm{T}_x\,\mathcal{M}$.



Framework of Line Search Optimization Methods

- Line search optimization methods on Riemannian manifolds

$$x_+ = R_x(\alpha d),$$

where $d \in \mathrm{T}_x \mathcal{M}$ and α is a step size.

Riemannian Gradient

- The Riemannian gradient $\mathrm{grad}\, f$ of f at x is the unique tangent vector such that

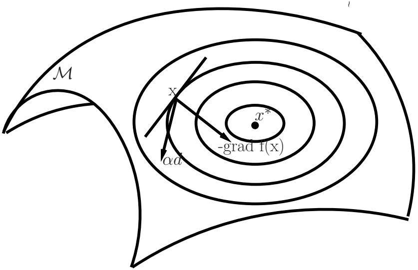
$$\langle \mathrm{grad}\, f(x), \eta \rangle_x = \mathrm{D}\, f(x)[\eta], \forall \eta \in \mathrm{T}_x\, \mathcal{M},$$

  where $\mathrm{D}\, f(x)[\eta]$ denotes the derivative of f along η.
- $\mathrm{grad}\, f(x)$ is the steepest ascent direction.

Search Direction and Step Size

- Search direction
  The angle between $-\operatorname{grad} f$ and d does not approach $\pi/2$.
- Step size
  - f decreases sufficiently,
  - Step size is not too small,
  - e.g., the Wolfe conditions, the Armijo-Goldstein conditions.
- Above conditions are sufficient to guarantee convergence.

- Example: The figure shows the contour curves of f around a minimizer x^*.



Steepest Descent

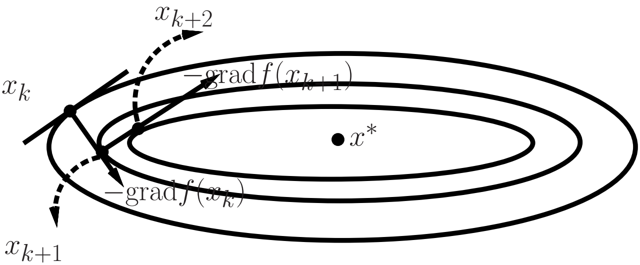
- Riemannian steepest descent (RSD): $d = -\operatorname{grad} f(x)$,
- Converges slowly, i.e., linearly

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

- The Riemannian Hessian of f at x is a linear operator on $\mathrm{T}_x \mathcal{M}$.
- Let $\operatorname{Hess} f(x^*)$ denote the Hessian at the minimizer x^* and $\lambda_{\min}$ and $\lambda_{\max}$ respectively denote the smallest and largest eigenvalue of $\operatorname{Hess} f(x^*)$. The smaller $\lambda_{\min}/\lambda_{\max}$ is, the more slowly steepest descent converges. [AMS08, Theorem 4.5.6]

An Example for Steepest Descent

- f is a function defined on a Euclidean space.
- x^* is a minimizer and $\lambda_{\min}/\lambda_{\max}$ is small.
- The following figure shows contour curves of $f(x)$ around x^* and iterates generated by an exact line search algorithm.

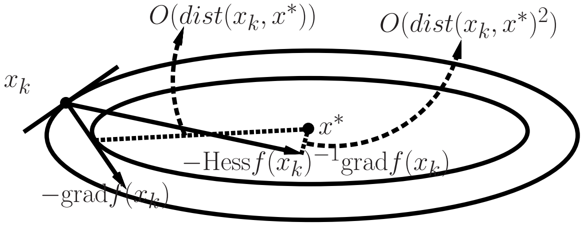


Newton Method

- Riemannian Newton update formula:

$$x_+ = R_x(\alpha[-\operatorname{Hess} f(x)^{-1} \operatorname{grad} f(x)]),$$

  where α is chosen to be 1 when x is close enough to x^*.
- The search direction is not necessarily descent.
- When x_k is close enough to x^*, the search direction is descent.
- Riemannian Newton method converges quadratically [AMS08, Theorem 6.3.2], i.e.,$\lim_{k\to\infty} \frac{\operatorname{dist}(x_{k+1},x^*)}{\operatorname{dist}(x_k,x^*)^2} < \infty$.

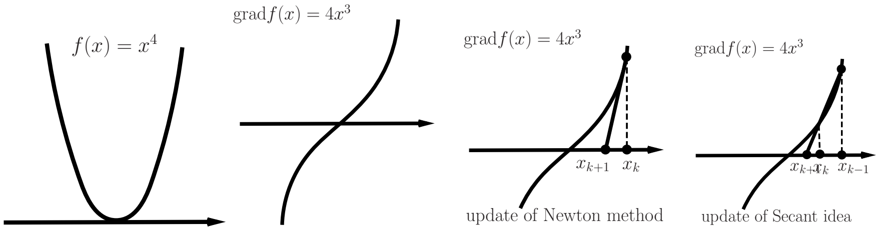


Quasi-Newton Methods

- Steepest descent method
    - Converge slowly
- Newton method
    - Requires the action of the Hessian which may be expensive or unavailable
    - Search direction may be not descent. Therefore, extra considerations are required.
- Quasi-Newton method
    - Approximate the action of the Hessian or its inverse and therefore accelerate the convergent rate
    - Provide an approach to produce a descent direction

Secant Condition

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



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

Riemannian Secant Conditions

- 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 in different tangent spaces. A method of comparing tangent vectors in different tangent spaces is required.

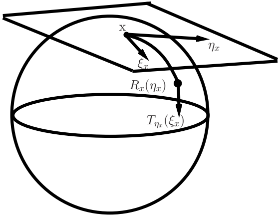
Vector Transport

Vector transport

- Transport a tangent vector from one tangent space to another.
- notation: $\mathcal{T}_{\eta_x}\xi_x$, denotes transport of ξ_x to tangent space of $R_x(\eta_x)$. R is a retraction associated with $\mathcal{T}$.
- An isometric vector transport, denoted by $\mathcal{T}_S$, additionally satisfies

$$g_x(\eta_x, \xi_x) = g_y(\mathcal{T}_{S_{\zeta_x}}\eta_x, \mathcal{T}_{S_{\zeta_x}}\xi_x),$$

where $x, y \in \mathcal{M}$, $y = R_x(\zeta_x)$ and $\eta_x, \xi_x, \zeta_x \in \mathrm{T}_x\,\mathcal{M}$.



Riemannian Secant Conditions

The secant condition of Qi [Qi11]:

$$\operatorname{grad} f(x_{k+1}) - P_{\gamma_k}^{1\leftarrow 0} \operatorname{grad} f(x_k) = \mathcal{B}_{k+1}(P_{\gamma_k}^{1\leftarrow 0} \operatorname{Exp}_{x_k}^{-1} x_{k+1}),$$

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]:

$$(\mathrm{grad}\, f(x_{k+1})^\flat \mathcal{T}_{R_{\xi_k}} - \mathrm{grad}\, f(x_k)^\flat)\mathcal{T}_{S_{\xi_k}}^{-1} = (\mathcal{B}_{k+1}\mathcal{T}_{S_{\xi_k}}\xi_k)^\flat$$

where $\mathcal{T}_R$ is differentiated retraction of R, i.e.,

$$\mathcal{T}_{R_{\eta_x}}\zeta_x = \frac{d}{dt}R_x(\eta_x + t\zeta_x)|_{t=0}$$

and $\eta_x^\flat$ denotes a function from $\mathrm{T}_x\,\mathcal{M}$ to $\mathbb{R}$, i.e., $\eta_x^\flat \xi_x = g_x(\eta_x, \xi_x)$. Their work is on infinite dimensional manifolds. It is rewritten in a finite dimensional form so that it can be compared to our secant condition.

Riemannian Secant Conditions

- We use

$$\operatorname{grad} f(x_{k+1})/\beta_k - \mathcal{T}_{S_{\xi_k}} \operatorname{grad} f(x_k) = \mathcal{B}_{k+1} \mathcal{T}_{S_{\xi_k}} \xi_k,$$

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

$$\mathcal{T}_{S_\xi} \xi = \beta \mathcal{T}_{R_\xi} \xi.$$

Euclidean DFP

The Euclidean secant condition and some additional constraints are imposed.

$$\min_B \|B - B_k\|_{W_B}$$
$$\text{s.t. } B = B^T,$$

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

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

where $s_k = x_{k+1} - x_k$ and $y_k = \mathrm{grad}\, f(x_{k+1}) - \mathrm{grad}\, f(x_k)$. This is called Davidon-Fletcher-Powell(DFP) update.

Euclidean BFGS

Let $H_k = B_k^{-1}$.

$$\min_{H} \|H - H_k\|_{W_H}$$
$$\text{s.t. } H = H^T,$$

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

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

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

Euclidean Broyden Family

The linear combination of BFGS update and DFP update is called Broyden Family update, $(1-\phi_k)BFGS + \phi_k DFP$:

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

where

$$v_k = \frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k}.$$

If $\phi_k \in [0,1]$, then it is restricted Broyden Family update.

- Properties
  - If $y_k^T s_k > 0$, then B_{k+1} is positive definite if and only if B_k is positive definite.
  - $y_k^T s_k > 0$ is guaranteed by the Wolfe second condition.

Riemannian Broyden Family

Riemannian Restricted Broyden Family update is

$$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k s_k (\tilde{\mathcal{B}}_k^* s_k)^\flat}{(\tilde{\mathcal{B}}_k^* s_k)^\flat s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k} + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) v_k v_k^\flat,$$

where $\phi_k \in [0,1]$, $\eta_x^\flat$ denotes a function from $\mathrm{T}_x\,\mathcal{M}$ to $\mathbb{R}$, i.e., $\eta_x^\flat \xi_x = g_x(\eta_x, \xi_x)$, $s_k = \mathcal{T}_{S_{\alpha_k \eta_k}} \alpha_k \eta_k$ and $y_k = \mathrm{grad}\, f(x_{k+1})/\beta_k - \mathcal{T}_{S_{\alpha_k \eta_k}} \mathrm{grad}\, f(x_k)$, $\tilde{\mathcal{B}}_k = \mathcal{T}_{S_{\alpha_k \eta_k}} \circ \mathcal{B}_k \circ \mathcal{T}_{S_{\alpha_k \eta_k}}^{-1}$ and

$$v_k = \frac{y_k}{g(y_k, s_k)} - \frac{\tilde{\mathcal{B}}_k s_k}{g(s_k, \tilde{\mathcal{B}}_k s_k)}.$$

Riemannian Broyden Family

Properties

- If $g(y_k, s_k) > 0$, then $\mathcal{B}_{k+1}$ is positive definite if and only if $\mathcal{B}_k$ is positive definite.
- $g(y_k, s_k) > 0$ is not guaranteed by the most natural way of generalizing the Wolfe second condition for arbitrary retraction and isometric vector transport.
- We impose another condition called the 'locking condition'

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

where $\mathcal{T}_R$ is differentiated retraction.

Line Search Riemannian Broyden Family Method

(1) Given initial x_0 and symmetric positive definite 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

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

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

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

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

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

Euclidean Theoretical Results

- If $f \in C^2$ and strongly convex, then the sequence $\{x_k\}$ generated by a Broyden family algorithm with $\phi_k \in [0, 1-\delta)$ converges to the minimizer x^*, where $\delta > 0$. Furthermore, the convergence rate is linear.
- If additionally, $\mathrm{Hess}\, f$ is Hölder continuous at the minimizer x^*, i.e., there exist $p > 0$ and $L > 0$ such that

$$\|\mathrm{Hess}\, f(x) - \mathrm{Hess}\, f(x^*)\| \leq L\|x - x^*\|^p,$$

for all x in a neighborhood of x^*, then step size $\alpha_k = 1$ satisfies the Wolfe conditions eventually. Moreover, if 1 is chosen to be the step size whenever it satisfies the Wolfe conditions, $\{x_k\}$ converges to x^* superlinearly, i.e.,

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|_2}{\|x_k - x^*\|_2} = 0.$$

Riemannian Theoretical Results

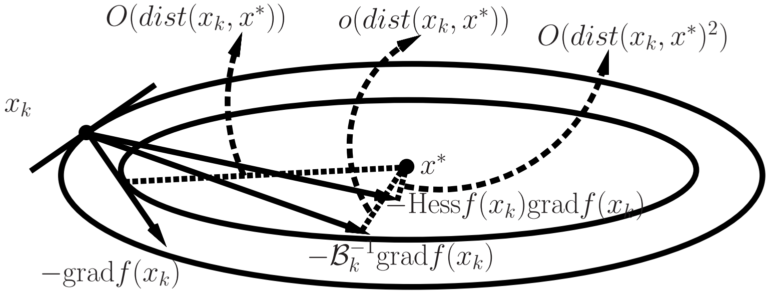
1. The (strong) convexity of a function is generalized to the Riemannian setting and is called (strong) retraction-convexity.
2. Suppose some reasonable assumptions hold. If $f \in C^2$ and strongly retraction-convex, then the sequence $\{x_k\}$ generated by a Riemannian Broyden family algorithm with $\phi_k \in [0, 1-\delta)$ converges to the minimizer x^*, where $\delta > 0$. Furthermore, the convergence rate is linear.
3. If additionally, $\mathrm{Hess}\, f$ satisfies a generalization of Hölder continuity at the minimizer x^*, then step size $\alpha_k = 1$ satisfies the Wolfe conditions eventually. Moreover, if 1 is chosen to be the step size whenever it satisfies the Wolfe conditions, $\{x_k\}$ converges to x^* superlinearly, i.e.,

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

Convergence Rate

Step size $\alpha_k = 1$:

- Eventually works for Riemannian Broyden family algorithm and Riemannian quasi-Newton algorithm
- Does not work for RSD in general.



Limited-memory RBFGS

Riemannian Restricted Broyden Family requires computing $\tilde{\mathcal{B}}_k = \mathcal{T}_{S_{\alpha_k \eta_k}} \circ \mathcal{B}_k \circ \mathcal{T}_{S_{\alpha_k \eta_k}}^{-1}$.

- Explicit form of $\mathcal{T}_S$ may not exist.
- Even though it exists, matrix multiplication is needed.

Limited-memory

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

We only consider limited-memory RBFGS algorithm.

Limited-memory RBFGS

Consider the update of inverse Hessian approximation of RBFGS, $\mathcal{H}_k = \mathcal{B}_k^{-1}$. We have

$$\mathcal{H}_{k+1} = \mathcal{V}_k^{\flat} \tilde{\mathcal{H}}_k \mathcal{V}_k + \rho_k s_k s_k^{\flat}, \text{ where } \rho_k = \frac{1}{g(y_k, s_k)} \text{ and } \mathcal{V}_k = \mathrm{id} - \rho_k y_k s_k^{\flat}.$$

If the number of latest s_k and y_k we use is $m+1$, then

$$\begin{aligned} \mathcal{H}_{k+1} = & \tilde{\mathcal{V}}_k^{\flat} \tilde{\mathcal{V}}_{k-1}^{\flat} \cdots \tilde{\mathcal{V}}_{k-m}^{\flat} \tilde{\mathcal{H}}_{k+1}^0 \tilde{\mathcal{V}}_{k-m} \cdots \tilde{\mathcal{V}}_{k-1} \tilde{\mathcal{V}}_k \\ & + \rho_{k-m} \tilde{\mathcal{V}}_k^{\flat} \tilde{\mathcal{V}}_{k-1}^{\flat} \cdots \tilde{\mathcal{V}}_{k-m+1}^{\flat} s_{k-m}^{(k+1)} s_{k-m}^{(k+1)^{\flat}} \tilde{\mathcal{V}}_{k-m+1} \cdots \tilde{\mathcal{V}}_{k-1} \tilde{\mathcal{V}}_k \\ & + \cdots \\ & + \rho_k s_k^{(k+1)} s_k^{(k+1)^{\flat}}, \end{aligned}$$

where $\tilde{\mathcal{V}}_i = \mathrm{id} - \rho_i y_i^{(k+1)} s_i^{(k+1)^{\flat}}$ and $\mathcal{H}_{k+1}^0 = \frac{g(s_k, y_k)}{g(y_k, y_k)} \mathrm{id}$.

Construct $\mathcal{T}_S$

Methods 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.

Framework of Trust Region Optimization Methods

Euclidean trust region method is to build a local model

$$m_k(\eta) = f(x_k) + \mathrm{grad}\, f(x_k)^T \eta + \frac{1}{2}\eta^T B_k \eta$$

and finds

$$\eta_k = \arg \min_{\|\eta\|_2 \leq \delta_k} m_k(\eta),$$

where δ_k is the radius of trust region. The candidate of next iterate is

$$\tilde{x}_{k+1} = x_k + \eta_k.$$

If $(f(x_k) - f(\tilde{x}_k))/(m_k(0) - m_k(\eta_k)$ is big enough, then accept the candidate $x_{k+1} = \tilde{x}_{k+1}$, otherwise, reject the candidate. Finally, update the local model.

Framework of Trust Region Optimization Methods

Riemannian trust region builds a model on the tangent space of current iterate x_k,

$$m_k(\eta) = f(x_k) + g(\mathrm{grad}\, f(x_k), \eta) + \frac{1}{2} g(\eta, \mathcal{B}_k \eta)$$

and finds

$$\eta_k = \arg \min_{\|\eta\| \leq \delta_k} m_k(\eta),$$

where δ_k is the radius of trust region. The candidate of next iterate is

$$\tilde{x}_{k+1} = R_{x_k}(\eta_k).$$

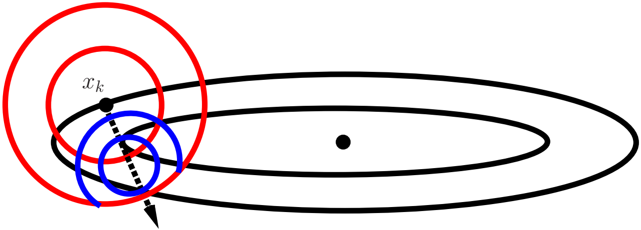
If $(f(x_k) - f(\tilde{x}_k))/(m_k(0) - m_k(\eta_k)$ is big enough, then accept the candidate $x_{k+1} = \tilde{x}_{k+1}$, otherwise, reject the candidate. Finally, update the local model.

Steepest Descent

- Riemannian trust region steepest descent(SD)
  - $\mathcal{B}_k = \mathrm{id}$,
  - If the local model is solved exactly, then

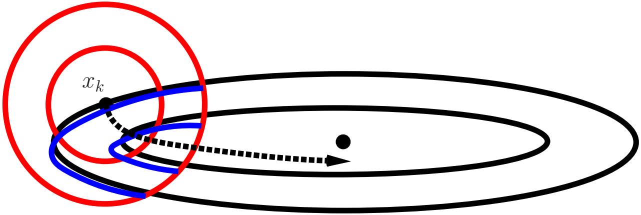
$$\eta_k = -\min(1, \delta_k / \| \mathrm{grad}\, f(x_k) \|)\, \mathrm{grad}\, f(x_k),$$

  - Converges linearly.



Newton Method

- Riemannian trust region Newton method
  - $\mathcal{B}_k = \mathrm{Hess}\, f(x_k)$,
  - Converges quadratically [Bak08],
  - In [Bak08], the local model is not required to be solved exactly and a Riemannian truncated conjugate gradient is proposed.



Quasi-Newton Method

Symmetric rank-1 update

- Euclidean: $B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}$,
- Riemannian: $\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k + \frac{(y_k - \tilde{\mathcal{B}}_k s_k)(y_k - \tilde{\mathcal{B}}_k s_k)^\flat}{g(s_k, y_k - \tilde{\mathcal{B}}_k s_k)}$,
  where $\tilde{\mathcal{B}}_k = \mathcal{T}_{S_{\eta_k}} \circ \mathcal{B}_k \circ \mathcal{T}_{S_{\eta_k}}^{-1}$.
- Properties:
  - It does not preserve positive definiteness of $\mathcal{B}_k$,
  - It produces better Hessian approximation as an operator.

  These properties suggest we use trust region.

Riemannian Trust region with symmetric rank-1 method (RTR-SR1)

(1) Given $\tau_1, c \in (0,1)$, $\tau_2 > 1$, initial x_0, and symmetric B_0. Let $k = 0$.

(2) Obtain η_k by (approximately) solving the local model $m_k(\eta)$

(3) Set the candidate of next iterate $\tilde{x}_{k+1} = R_{x_k}(\eta_k)$.

(4) Let $\rho_k = (f(x_k) - f(\tilde{x}_k))/(m_k(0) - m_k(\eta_k)$. If $\rho_k > c$, then $x_{k+1} = \tilde{x}_{k+1}$, otherwise $x_{k+1} = x_k$.

(5) Update the local model by first using update formula to obtain $\mathcal{B}_{k+1}$ and setting

$$\delta_{k+1} = \begin{cases} \tau_2 \delta_k, & \text{if } \rho_k > 0.75 \text{ and } \|\eta\| \geq 0.8\delta_k; \\ \tau_1 \delta_k, & \text{if } \rho_k < 0.1; \\ \delta_k, & \text{otherwise.} \end{cases}$$

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

Euclidean Theoretical Results

- If f is Lipschitz continuously differentiable and bounded below and the $\|B_k\| \leq C$ for some constant C, then the sequence $\{x_k\}$ generated by trust region with symmetric rank-1 update method converges to a stationary point x^*. [NW06]
- Suppose some reasonable assumptions hold. If $f \in C^2$ and the $\mathrm{Hess}\, f$ is Lipschitz continuous around the minimizer x^*, then the sequence $\{x_k\}$ converges to x^* $n+1$-step superlinearly, i.e.,

$$\lim_{k \to \infty} \frac{\|x_{k+n+1} - x^*\|_2}{\|x_k - x^*\|_2} = 0,$$

where n is the dimension of the domain. [BKS96]

Riemannian Theoretical Results

- Global convergence property has been proved in [Bak08] and is applicable for RTR-SR1.
- Suppose some reasonable assumptions hold. If $f \in C^2$ and the $\mathrm{Hess}\, f$ satisfies a Riemannian generalization version of Lipschitz continuity around the minimizer x^*, then the sequence $\{x_k\}$ converges to x^* $d+1$-step superlinearly, i.e.,

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

where d is the dimension of the manifold.

Limited-memory RTR-SR1

- Same motivation as limited-memory RBFGS
  - Less storage complexity,
  - Avoid some expensive operations.
- Similar techniques
  - Use a few previous s_k and y_k to approximate the action of the Hessian.

Important Theorems

Dennis and Moré conditions give necessary and sufficient conditions for a sequence $\{x_k\}$ converging superlinearly to x^* [DM77]. We have generalized to

- Riemannian Dennis Moré conditions for root solving
- Riemannian Dennis Moré conditions for optimization

Optimization for Partly Smooth Functions

- f is called partly smooth on $\mathcal{S}$ if it is continuously differentiable on an open dense subset.
- Gradient sampling algorithm (GS) [BLO05],
  - Global convergence analysis.
  - Works for non-Lipschitz continuous functions empirically.
- BFGS [LO13],
  - Modify the line search algorithm,
  - Modify the stopping criterion,
  - No convergence analysis.
  - Does not work for non-Lipschitz continuous functions empirically.

Optimization for Partly Smooth Functions

Complexity

- GS,
  - Many gradient evaluations in each iteration
  - Each iteration needs to solve a convex quadratic program.
- BFGS,
  - Less gradient evaluations than GS
  - Solving a convex quadratic program is needed when the sequence is close to convergence.
- Solving a convex quadratical program is expensive.

Optimization for Riemannian Partly Smooth Functions

- Generalized the framework of GS to the Riemannian setting.
- Generalized the modifications of BFGS to the Riemannian setting.
- Empirical performance testing.

General Implementations

- All the discussions about Riemannian optimization algorithms are general,
- General implementations for Riemannian manifolds that can be represented by $\mathbb{R}^n$ are given,
    - $\mathcal{M}$ is a subset of $\mathbb{R}^n$,
    - $\mathcal{M}$ is a quotient manifold with total space be a subset of $\mathbb{R}^n$,
    - $\mathcal{M}$ is a product of two or more manifolds each of which is any of the first two types.

General Implementations

The discussions include

- Representation of metric, linear operator and vector transports,
    - n-dimensional representation,
    - d-dimensional representation (intrinsic approach),
- Constructions and implementations of the vector transports.

Implementations for Four Specific Manifolds

Providing detailed efficient implementations for four particular manifolds:

- the sphere,
- the compact Stiefel manifold,
- the orthogonal group,
- the Grassmann manifold.

Experiments

Four cost functions are tested:

- the Brockett cost function on the Stiefel manifold,
- the Rayleigh quotient function on the Grassmann manifold,
- the Lipschitz minmax function on the sphere,
- the non-Lipschitz minmax function on the sphere.

Experiments

Ten algorithms are compared

- RBFGS,
- Riemannian Broyden family using Davidon's update ϕ [Dav75],
- Riemannian Broyden family using a problem specific ϕ,
- Limited-memory RBFGS,
- Riemannian SD,
- Riemannian GS,
- RTR-SR1,
- Limited-memory RTR-SR1,
- RTR-SD,
- RTR-Newton [Bak08].

Experiments

Systematic comparisons are made. The following are shown in the dissertation.

- Performance of different retractions and vector transports,
- Performance of different choices of ϕ_k,
- Performance of different algorithms,
- The locking condition yield robustness and reliability of Riemannian Broyden family in our framework. Empirical evidence shows it is not necessary but behavior then is often difficult to predict.
- The value of limited-memory versions for large scaled problems,
- The value of Riemannian GS for non-Lipschitz continuous function on a manifold.

Applications

- Applications with smooth enough cost functions,
  - The joint diagonalization problem for independent component analysis,
  - The synchronization of rotation problem,
  - Rotation and reparameterization problem of closed curves in elastic shape analysis,
  - Secant-based nonlinear dimension reduction.
- Application with a partly smooth cost function.
  - Secant-based nonlinear dimension reduction.

The Joint Diagonalization Problem for Independent Component Analysis

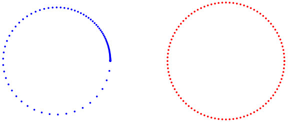
- Independent component analysis (ICA)
  - Determine an independent component form of a random vector,
  - Determine a few components of an independent component form of a random vector.
- Different cost functions are used [AG06], [TCA09]. We used the joint diagonalization cost function[TCA09].
  - The previous algorithm used is RTR-Newton. It is relatively slow when the number of samples are large.
  - RTR-SR1 and LRBFGS are the two fastest algorithms when the number of samples are large.

The Synchronization of Rotation Problem

- The Synchronization of Rotation Problem is to find N unknown rotations $R_1, \ldots R_N$ from M noisy measurements, H_{ij} of $\tilde{H}_{ij} = R_i R_j^T$.
- A review and a Riemannian approach for this problem can be found in [BSAB12].
- Using Riemannian optimization algorithms for the Riemannian approach, we showed that RBFGS and limited-memory RBFGS are the two fastest and reliable algorithms.

Rotation and Reparameterization Problem of Closed Curves in Elastic Shape Analysis

- In elastic shape analysis, a shape is invariant to
  - Scaling
  - Translation
  - Rotation
  - Reparametrization
    - shape1: $x = \cos(2\pi t^3), y = \sin(2\pi t^3), t \in [0, 1]$
    - shape2: $x = \cos(2\pi t), y = \sin(2\pi t), t \in [0, 1]$
- Our work is based on the framework of [SKJJ11].



Rotation and Reparameterization Problem of Closed Curves in Elastic Shape Analysis

- Elastic shape space is a quotient space. When two closed curves are compared, an important problem in elastic space analysis is to find the best rotation and reparametrization function.
- Previous algorithm is a coordinate relaxation of rotation and reparameterization.
  - Rotation: Singular value decomposition
  - Reparameterization: dynamic programming
  - One iteration
- Difficulties
  - High complexity.
  - Not robust when more iterations are used.

Rotation and Reparameterization Problem of Closed Curves in Elastic Shape Analysis

Gradient methods:

- Hessian is unknown,
- Infinite dimensional problem,
- Riemannian quasi-Newton algorithms can be applied,
  - Work for closed curves problem,
  - Reliable and much faster than the coordinate relaxation algorithm.

Secant-based Nonlinear Dimension Reduction

- Suppose $\mathcal{M}$ is a d-dimensional manifold embedded in $\mathbb{R}^n$. The idea is to find a projection $\pi_{[U]} = U(U^T U)^{-1} U^T$ such that $\pi_{[U]}|_{\mathcal{M}}$ is easy to invert, i.e., maximize $k_{\pi_{[U]}}$ where

$$k_{\pi_{[U]}} = \inf_{x,y \in \mathcal{M}, x \neq y} \frac{\|\pi_{[U]}(x-y)\|_2}{\|x-y\|_2}.$$

- The cost function $\phi([U]) = \|\pi_{[U]}(x-y)\|_2 / \|x-y\|_2$ is partly smooth.
- An alternative smooth cost function $F([U])$ is proposed in [BK05].
- Discretization is needed to approximate F and ϕ, called $\tilde{F}$ and $\tilde{\phi}$ respectively.

Secant-based Nonlinear Dimension Reduction

- Previous method used in [BK05] is Riemannian conjugate gradient algorithm for $\tilde{F}([U])$.
- An example (used in [BK00] and [BK05]) is tested
  - For the smooth function $\tilde{F}$, RBFGS and LRBFGS is the two fastest algorithms.
  - For the partly smooth function $\tilde{\phi}$, RBFGS is the fastest algorithm.
  - Even though Riemannian GS is relatively slow, it can escape from local optima and usually find the global optimum.
  - $\tilde{F}$ is a worse cost function than $\tilde{\phi}$ in the sense the global optimum of $\tilde{F}$ may be non-invertible.

Conclusions

- Generalized Broyden family update and symmetric rank-1 update to the Riemannian setting; combined them with line search and trust region strategy respectively and provided complete convergence analysis.
- Generalized limited-memory version of SR1 and BFGS to the Riemannian setting.
- The main work of generalizing quasi-Newton algorithms to Riemannian setting is finished.
- Generalized GS and modified version of BFGS to the Riemannian setting.
- Developed general, detailed and efficient implementations for Riemannian optimization.
- Empirical performances are accessed by experiments and four applications.

Thank you!

Thank you!

References I

P.-A. Absil and K. A. Gallivan.
Joint diagonalization on the oblique manifold for independent component analysis.
2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings., 5:V945–V948, 2006.

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.

References II

D. S. Broomhead and M. Kirby.
A new approach to dimensionality reduction: theory and algorithms.
SIAM Journal on Applied Mathematics, 60(6):2114–2142, 2000.

D. S. Broomhead and M. J. Kirby.
Dimensionality reduction using secant-based projection methods : the induced dynamics in projected systems.
Nonlinear Dynamics, 41(1-3):47–67, 2005.

R. H. Byrd, H. F. Khalfan, and R. B. Schnabel.
Analysis of a symmetric rank-one trust region method.
SIAM Journal on Optimization, 6(4):1025–1039, 1996.

References III

J. V. Burke, A. S. Lewis, and M. L. Overton.
A robust gradient sampling algorithm for nonsmooth, nonconvex optimization.
SIAM Journal on Optimization, 15(3):751–779, January 2005.
doi:10.1137/030601296.

N. Boumal, A. Singer, P.-A. Absil, and V. D. Blondel.
Cramer-Rao bounds for synchronization of rotations, 2012.
arXiv:1211.1621v1.

W. C. Davidon.
Optimally conditioned optimization algorithms without line searches.
Mathematical Programming, 9(1):1–30, 1975.

References IV

J. E. Dennis and J. J. Moré.
Quasi-Newton methods, motivation and theory.
SIAM Review, 19(1):46–89, 1977.

A. S. Lewis and M. L. Overton.
Nonsmooth optimization via quasi-Newton methods.
Mathematical Programming, 141(1-2):135–163, February 2013.
doi:10.1007/s10107-012-0514-2.

J. Nocedal and S. J. Wright.
Numerical optimization.
Springer, second edition, 2006.

References V

B. O'Neill.
Semi-Riemannian geometry.
Academic Press Incorporated [Harcourt Brace Jovanovich Publishers], 1983.

C. Qi.
Numerical optimization methods on Riemannian manifolds.
PhD thesis, Florida State University, 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 VI

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.

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.