A Tutorial on Riemannian Optimization

赖志坚
北京国际数学研究中心
April 29, 2024

Riemannian optimization

o

(Un)constrained Optimization Problem

Given an objective $f : \mathbb{R}^n \to \mathbb{R}$, the general form of a (Euclidean) optimization problem is

$$\begin{aligned} &\min\ f(x) \\ &\text{s.t. } x \in S, \end{aligned} \tag{1}$$

where $x = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^n$, and feasible region $S \subset \mathbb{R}^n$ consists of all possible solutions.

Classically, we consider it as

- unconstrained optimization problem if $S = \mathbb{R}^n$;
- constrained optimization problem if $S \subsetneq \mathbb{R}^n$, e.g., $S = \{x \in \mathbb{R}^n : g_i(x) = 0, i = 1, 2, \ldots, m$ and $h_j(x) \le 0, j = 1, 2, \ldots, l\}$.

Line Search Framework for $S = \mathbb{R}^n$

Algorithm 1 Line Search Framework for $S = \mathbb{R}^n$

An initial point $x_0 \in \mathbb{R}^n$; $k \leftarrow 0$;
repeat
  Choose a search direction $d_k \in \mathbb{R}^n$;
  Choose a step size $t_k > 0$;
  Update new point by $x_{k+1} := x_k + t_k d_k$;
  Set $k \rightarrow k + 1$;
until stopping criterion are satisfied;

It should be noted that:

- By using *local information* of objective f at x_k, we can select
  - steepest descent direction: $d_k = -\nabla f(x_k)$
  - Newton direction: $d_k = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$
- For *arbitrary* d_k and t_k, the new point x_{k+1} is always in $\mathbb{R}^n$. (unconstrained!)

Questions

Why cannot the line search framework be used for constrained optimization problems, i.e., $S \subsetneq \mathbb{R}^n$? Because $x_{k+1} := x_k + t_k d_k$ may not be feasible.

Recall the general form of a (Euclidean) optimization problem is

$$\begin{aligned} &\min \ f(x) \\ &\text{s.t. } x \in S. \end{aligned} \tag{2}$$

- $S = \mathbb{R}^n$. Formally, x is still subject to the *real* (not complex) Euclidean space $\mathbb{R}^n$.
- $S \subsetneqq \mathbb{R}^n$. Assume that we can generate a sequence $\{x_k\} \subset S$ by the formula

$$x_{k+1} := \mathrm{UPDATE}\left(x_k, d_k, t_k\right), \tag{3}$$

where $\mathrm{UPDATE}\colon S \times D \times \mathbb{R}^+ \to S$, and D consist of all meaningful search direction.

A new insight

The essential difference between constrained and unconstrained problems is not determined by the problem itself, but by the algorithm we adopt to solve the problems.

Riemannian optimization

Given an objective $f : \mathcal{M} \to \mathbb{R}$ where $\mathcal{M}$ is a Riemannian manifold, we want to solve

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

40+ manifolds $\mathcal{M}$ available in the Riemannian optimization solver "Manopt" [BMAS14]:

- $\mathbb{R}^n, \mathbb{R}^{m \times n}$ (any vector space) are trivial manifolds.
- Sphere manifold, $\{x \in \mathbb{R}^n : \|x\|_2 = 1\}$.
- Stiefel manifold, $\{X \in \mathbb{R}^{n \times p} : X^T X = I_p\}$.
- Grassmann manifold, the set of all p-dimensional subspaces of $\mathbb{R}^n$.
- Fixed rank manifold, $\{X \in \mathbb{R}^{n \times m} : \mathrm{rank}(X) = r\}$.
- Oblique manifold, $\{X \in \mathbb{R}^{n \times m} : \|X_{:1}\| = \cdots = \|X_{:m}\| = 1\}$.
- Hyperbolic manifold, $\left\{x \in \mathbb{R}^{n+1} : x_0^2 = x_1^2 + \cdots + x_n^2 + 1\right\}$.
- In most cases, the $\mathbb{R}$ above can be replaced by $\mathbb{C}$.

Riemannian optimization

Given an objective $f : \mathcal{M} \to \mathbb{R}$ where $\mathcal{M}$ is a Riemannian manifold, we want to solve

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

Applications of Riemannian optimization [HLWY20]:

- p-harmonic flow
- low-rank nearest correlation matrix estimation
- phase retrieval
- Bose-Einstein condensates
- cryoelectron microscopy (cryo-EM)
- linear eigenvalue problem
- nonlinear eigenvalue problem from electronic structure calculations
- combinatorial optimization
- deep learning, etc.

Application I: Extreme Eigenvalue or Singular Value

For a matrix $A \in \mathrm{Sym}(n)$, we have

$$\text{the smallest eigenvalue of } A = \min_{x \in \mathbb{S}^{n-1}} x^T A x. \tag{4}$$

Similarly, for a matrix $M \in \mathbb{R}^{m \times n}$, we have

$$\text{the largest singular value of } M = \max_{x \in \mathbb{S}^{m-1}, y \in \mathbb{S}^{n-1}} x^T M y. \tag{5}$$

- Unit sphere manifold, $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : \|x\|_2 = 1\}$.
- $\mathbb{S}^{m-1} \times \mathbb{S}^{n-1}$ is a product manifold.

Spare PCA wants to find principle eigenvectors with few nonzero elements.

$$\min_{x \in \mathrm{St}(n,p)} -\mathrm{tr}\left(X^T A^T A X\right) + \rho \|X\|_1. \tag{6}$$

where $\|X\|_1 = \sum_{ij} |X_{ij}|$ and $\rho \geq 0$ is a parameter to promote sparsity.

- Stiefel manifold, $\mathrm{St}(n,p) = \left\{X \in \mathbb{R}^{n \times p} : X^T X = I_p\right\}$.
- Grassmann manifold, $\mathrm{Gr}(n,p) = \left\{\mathrm{span}(X) : X \in \mathbb{R}^{n \times p}, X^T X = I_p\right\}$. (See Appendix for more.)

Let Ω denote the set of pairs (i,j) such that M_{ij} is observed. We want to recover a low-rank matrix M by

$$\begin{array}{ll} \min_X & \operatorname{rank}(X) \\ \text{s.t.} & X_{ij} = M_{ij}, \quad (i,j) \in \Omega. \end{array} \tag{7}$$

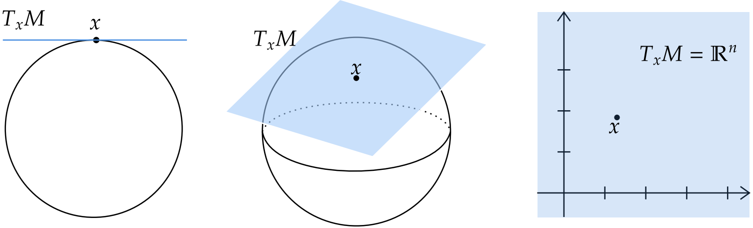
If $\operatorname{rank}(M) = r$ is known, an alternative model is

$$\min_{X \in \mathrm{Fr}(m,n,r)} \sum_{(i,j)\in\Omega} (X_{ij} - M_{ij})^2 . \tag{8}$$

- Fixed rank manifold, $\mathrm{Fr}(m,n,r) = \{X \in \mathbb{R}^{m\times n} : \operatorname{rank}(X) = r\}$.

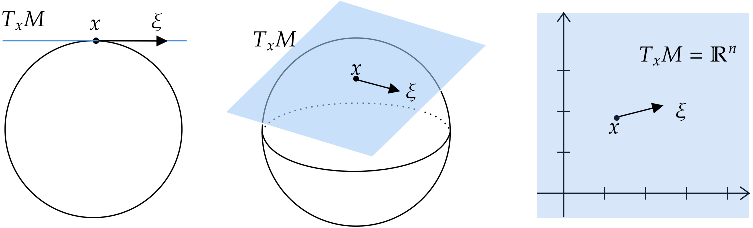
Riemannian Manifold = Manifold + Riemannian Metric

- A manifold $\mathcal{M}$ is a set that can be locally linearized.[1]
  - $T_x\mathcal{M}$ is tangent space at x.
  - $\xi \in T_x\mathcal{M}$ is tangent vector at x.
- A Riemannian metric $\langle \cdot, \cdot \rangle$ assigns an inner product $\langle \cdot, \cdot \rangle_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ to each tangent space of the manifold in a way that varies smoothly from point to point.



[1]Exactly, it is a topological space that is locally homeomorphic to some open subset of Euclidean space.

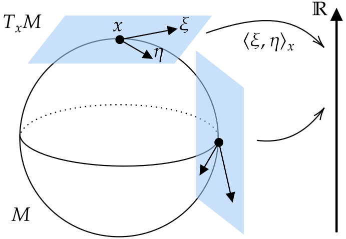
- A manifold $\mathcal{M}$ is a set that can be locally linearized.[2]
  - $T_x\mathcal{M}$ is tangent space at x.
  - $\xi \in T_x\mathcal{M}$ is tangent vector at x.
- A Riemannian metric $\langle \cdot, \cdot \rangle$ assigns an inner product $\langle \cdot, \cdot \rangle_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ to each tangent space of the manifold in a way that varies smoothly from point to point.



[2] Exactly, it is a topological space that is locally homeomorphic to some open subset of Euclidean space.

Riemannian Manifold = Manifold + Riemannian Metric

- A manifold $\mathcal{M}$ is a set that can be locally linearized.[3]
  - $T_x\mathcal{M}$ is tangent space at x.
  - $\xi \in T_x\mathcal{M}$ is tangent vector at x.
- A Riemannian metric $\langle \cdot, \cdot \rangle$ assigns an inner product $\langle \cdot, \rangle_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ to each tangent space of the manifold in a way that varies smoothly from point to point.



[3]Exactly, it is a topological space that is locally homeomorphic to some open subset of Euclidean space.

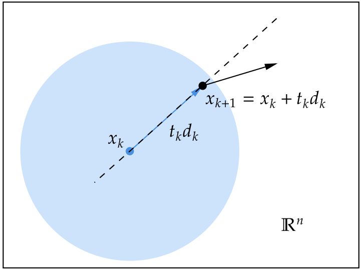
Euclidean Optimization v.s. Riemannian Optimization

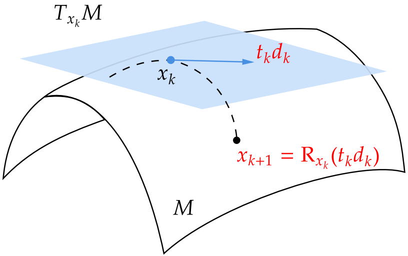
Algorithm 2 Line Search Framework for $S = \mathbb{R}^n$

Choose a search direction $d_k \in \mathbb{R}^n$;
Choose a step size $t_k > 0$;
Update new point by $x_{k+1} := x_k + t_k d_k$;

Algorithm 3 Line Search Framework for $S = \mathcal{M}$

Choose a search direction $d_k \in T_{x_k}\mathcal{M}$;
Choose a step size $t_k > 0$;
Update new point by $x_{k+1} := \mathrm{R}_{x_k}(t_k d_k)$;





Advantages in Comparison to Euclidean Optimization

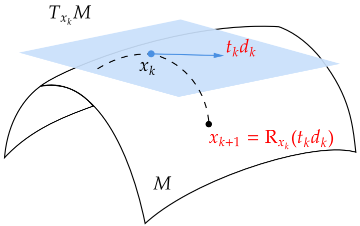
Riemannian version of classical methods:

- Riemannian steepest decent [Bou23]
- Riemannian conjugate gradient [Sat22]
- Riemannian trust region [ABG07]
- Riemannian Newton [Bou23]
- Riemannian BFGS [HGSA16]
- Riemannian proximal gradient [CMMCSZ20]
- Riemannian stochastic algorithms [ZJRS16]
- Riemannian ADMM [KGB16]
- and more

Almost all algorithms in Euclidean setting can be extended to Riemannian setting.

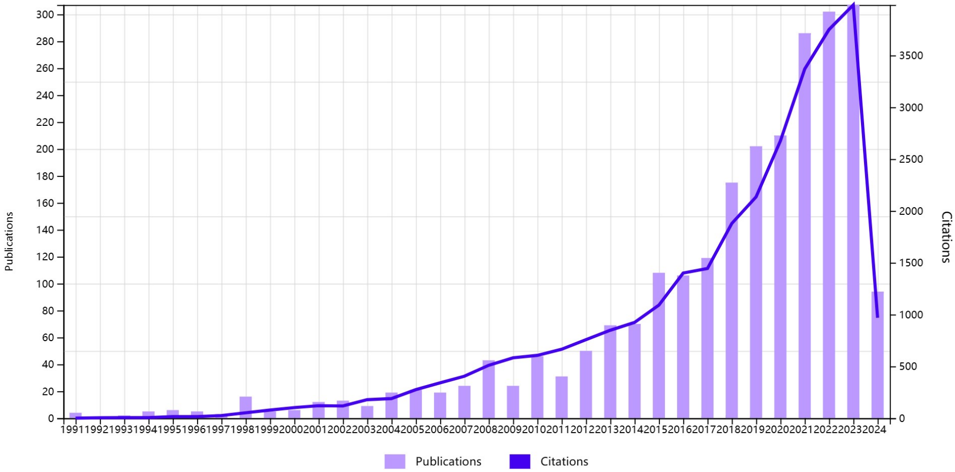
Advantages of Riemannian optimization:

1. All iterates on the manifold.
2. Transform constrained problems into unconstrained ones.
3. Use of the geometric structure of the feasible region.
4. Convergence properties of like optimization on Euclidean space.



Citation Report: Riemannian Optimization (Topic)

Publication Years: 1990-2024. Data Set: Web of Science Core Collection



Riemannian Optimization Libraries

Survey:

- A Brief Introduction to Manifold Optimization [HLWY20]
- A Survey of Geometric Optimization for Deep Learning: From Euclidean Space to Riemannian Manifold [FWL+23]
- History of Riemannian Optimization
  https://www.math.fsu.edu/~whuang2/pdf/NanjingUniversity_2019-10-23.pdf

Monographs of Riemannian Optimization:

- An Introduction to Optimization on Smooth Manifolds [Bou23] (the best textbook for beginners)
  https://www.nicolasboumal.net/book/
- Riemannian Optimization and Its Applications [Sat21]
  https://link.springer.com/book/10.1007/978-3-030-62391-3

Riemannian Optimization Libraries

- Optimization Algorithms on Matrix Manifolds [AMS08]
  https://press.princeton.edu/absil
- Convex Functions and Optimization Methods on Riemannian Manifolds [Udr94]
  https://link.springer.com/book/10.1007/978-94-015-8390-9
- Multivariate Data Analysis on Matrix Manifolds [TG21]
  https://link.springer.com/book/10.1007/978-3-030-76974-1
- Population-Based Optimization on Riemannian Manifolds [FT22a]
  https://link.springer.com/book/10.1007/978-3-031-04293-5

Libraries of General-purpose Riemannian Optimization Toolboxes:

- Manopt [BMAS14] in Matlab (the most comprehensive toolbox)
  https://www.manopt.org/

- Pymanopt [TKW16] in Python
  https://pymanopt.org/
- ROPTLIB [HAGH18] in C++
  https://www.math.fsu.edu/~whuang2/Indices/index_ROPTLIB.html
- ManifoldOptim [MRHA20] in R (a R wrapper of ROPTLIB)
  https://cran.r-project.org/web/packages/ManifoldOptim/index.html
- Manopt.jl [Ber22] in Julia
  https://manoptjl.org/

Libraries of Riemannian Packages for Various Goals:

- Geoopt [KKK20] is a Python library bringing Riemannian optimization tools to PyTorch.
  https://geoopt.readthedocs.io/en/latest/index.html

Riemannian Optimization Libraries

- McTorch [MJK+18] is also a Python library bringing Riemannian optimization tools to PyTorch.
  https://github.com/mctorch/mctorch
- TensorFlow RiemOpt [Smi21] is a library for Riemannian optimization in TensorFlow.
  https://github.com/master/tensorflow-riemopt
- Rieoptax [UHJM22] is a library for Riemannian Optimization in JAX.
  https://github.com/SaitejaUtpala/rieoptax
- CDOpt [XHLT22] is a Python toolbox for optimization on Riemannian manifolds with support for deep learning.
  https://cdopt.github.io/md_files/intro.html
- QGOpt [LRFO21] is an extension of TensorFlow optimizers on Riemannian manifolds that often arise in quantum mechanics.
  https://qgopt.readthedocs.io/en/latest

- Geomstats [MGLB+20] is a Python package for computations and statistics on manifolds. https://geomstats.github.io/

How to Optimize a Function on Manifold?

Consider the Riemannian optimization problem,

$$\begin{aligned} &\min \ f(x) \\ &\text{s.t. } x \in \mathcal{M}, \end{aligned} \tag{9}$$

where $f : \mathcal{M} \to \mathbb{R}$.

Goal: To find a local optimal solution $x^* \in \mathcal{M}$. (In general, $\mathcal{M}$ is nonconvex.)

Method: The iterative methods can still be used. But there are questions that we need to address:

- Q1: What is the direction of movement? Tangent vector
- Q2: How to move on manifolds? Retraction map
- Q3: What is a good direction to move? Riemannian gradient
- Q4: What is the optimal condition? Vector field

Q1: What is the Direction of Movement? Tangent Vector

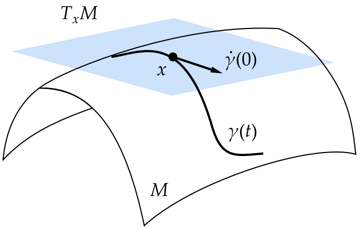
Remark

Here, it is sufficient to consider — embedded submanifold $\mathcal{M}$ of $\mathbb{R}^n$ = manifold + subset of $\mathbb{R}^n$.

Imagine a particle moving on a manifold $\mathcal{M}$ with a trajectory $\gamma : I \subseteq \mathbb{R} \to \mathcal{M}$ that passes through the point x at time $t = 0$. Then, the velocity

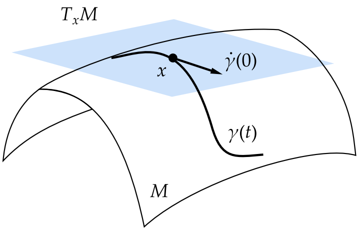
$$\dot{\gamma}(0) := \lim_{t \to 0} \frac{\gamma(t) - \gamma(0)}{t} = \frac{d}{dt}\gamma(t)\bigg|_{t=0}$$

is called a tangent vector belonging to x.



Q1: What is the Direction of Movement? Tangent Vector (Cont'd)

The tangent space at x is the set of all possible tangent vectors at that point, i.e., $T_x\mathcal{M} := \{\dot{\gamma}(0) : \gamma : I \to \mathcal{M}$ is a smooth curve, $\gamma(0) = x\}$.



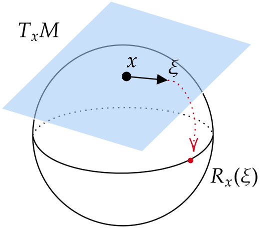
(1) For any $x \in \mathcal{M}$, $T_x\mathcal{M}$ are linear spaces sharing the same dimension.
(2) In general, $T_x\mathcal{M}$ is determined by x, except for $T_x\mathbb{R}^n \cong \mathbb{R}^n$.
(3) For embedded submanifold, $T_x\mathcal{M}$ is a subspace of $\mathbb{R}^n$, e.g., $T_x\mathbb{S}^{n-1} = \{u \in \mathbb{R}^n : x^T u = 0\}$.

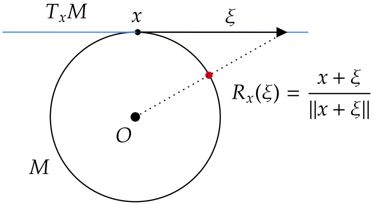
Q2: How to Move on Manifolds? Retraction to Create a Curve

$T\mathcal{M} = \{(x, \xi) : x \in \mathcal{M} \text{ and } \xi \in T_x\mathcal{M}\}$ is called the tangent bundle.
A retraction is a smooth map

$$\mathrm{R} : T\mathcal{M} \to \mathcal{M} : (x, \xi) \mapsto \mathrm{R}_x(\xi)$$

such that for each $(x, \xi) \in T\mathcal{M}$, the corresponding curve $t \mapsto \gamma(t) := \mathrm{R}_x(t\xi)$ has $\dot{\gamma}(0) = \xi$.





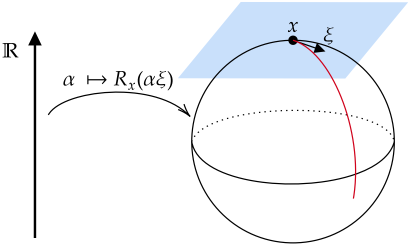
A retraction R yields a map $R_x : T_x\mathcal{M} \to \mathcal{M}$ for any x.

Q2: How to Move on Manifolds? Using Retraction to Create a Curve (Cont'd)

Retractions are not uniquely determined. E.g., on the unit sphere $\mathbb{S}^{n-1}$,

$$\mathrm{R}_x(\xi) = \frac{x+\xi}{\|x+\xi\|}, \quad \text{or} \quad \mathrm{R}_x(\xi) = \cos(\|\xi\|)x + \frac{\sin(\|\xi\|)}{\|\xi\|}\xi.$$

Given a tangent vector ξ at point x, $\alpha \mapsto R_x(\alpha\xi)$ defines a curve along this direction.

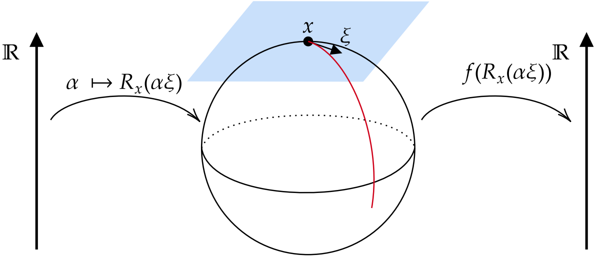


Euclidean setting	Riemannian setting
$x_{k+1} = x_k + \alpha_k d_k$	$x_{k+1} = R_{x_k}(\alpha_k \xi_k)$

Table: Two types of update formulas

Q3: What is a Good Direction? Riemannian Gradient

Moreover, the real function $\alpha \mapsto f(R_x(\alpha\xi))$ evaluates how the objective value changes along the given direction ξ.

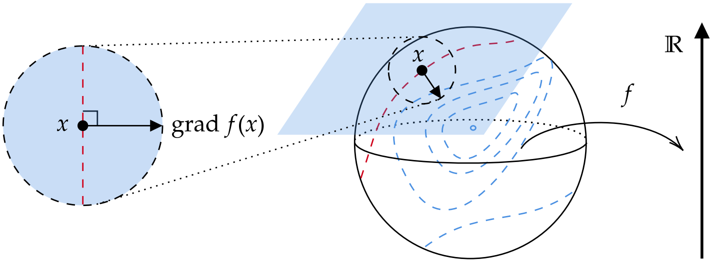


The Riemannian gradient, $\operatorname{grad} f(x)$, is the tangent vector at x such that:

$$\frac{\operatorname{grad} f(x)}{\|\operatorname{grad} f(x)\|} = \underset{\xi \in T_x\mathcal{M}:\|\xi\|=1}{\arg\max} \left(\lim_{\alpha \to 0} \frac{f(R_x(\alpha\xi)) - f(x)}{\alpha} \right).$$

Q3: What is a Good Direction? Riemannian Gradient (Cont'd)

Intuitively, $\operatorname{grad} f(x)$ should be approximately perpendicular to the contour line of f on the surface.



Also, $-\operatorname{grad} f(x)$ is the direction of steepest descent at x.

Q3: What is a Good Direction? Riemannian Gradient (Cont'd)

For embedded submanifold $\mathcal{M}$, Riemannian gradient of $f : \mathcal{M} \to \mathbb{R}$ is the orthogonal projection onto $T_X\mathcal{M}$ of the Euclidean gradient:

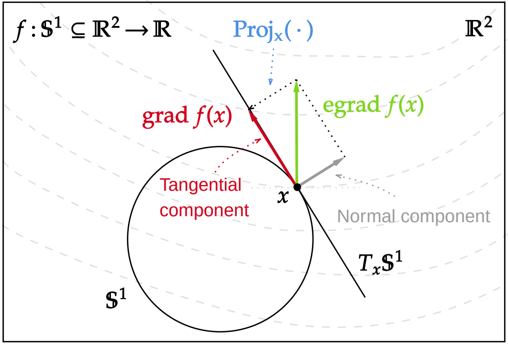
$$\operatorname{grad} f(x) = \operatorname{Proj}_x(\nabla f(x)).$$

Example

For $f(x) = \frac{1}{2}x^T Ax$, $\nabla f(x) = Ax$. On sphere $\mathbb{S}^{n-1}$, we have

$$\operatorname{Proj}_x(u) = (I_n - xx^T)u.$$

It follows that
$\operatorname{grad} f(x) = \operatorname{Proj}_x(\nabla f(x)) = (I_n - xx^T)Ax.$



Q4: What is the Optimal Condition? Singularity of Gradient Vector Field

A vector field on $\mathcal{M}$ is a map $V : \mathcal{M} \to T\mathcal{M}$ such that $V(x) \in T_x\mathcal{M}$ for all $x \in \mathcal{M}$.

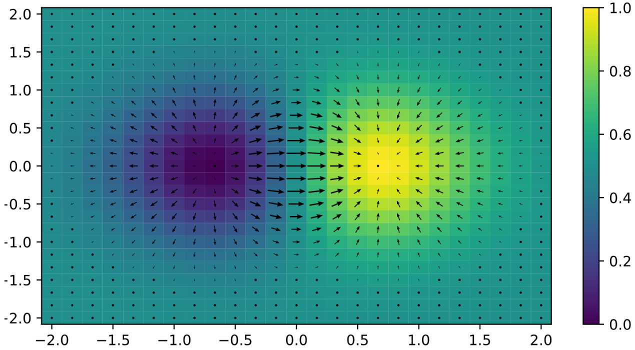


Figure: Let $\mathcal{M} = \mathbb{R}^2$. Gradient of the 2D function $f(x, y) = xe^{-(x^2+y^2)}$. Source: Wikipedia.

Q4: What is the Optimal Condition? Singularity of Gradient Vector Field

A vector field on $\mathcal{M}$ is a map $V : \mathcal{M} \to T\mathcal{M}$ such that $V(x) \in T_x\mathcal{M}$ for all $x \in \mathcal{M}$.

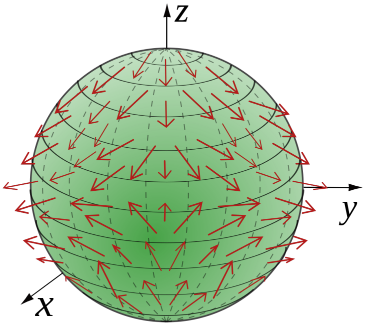


Figure: A vector field on a sphere $\mathbb{S}^2$. Source: Wikipedia.

Q4: What is the Optimal Condition? Singularity of Gradient Vector Field (Cont'd)

Riemannian gradient, $x \mapsto \operatorname{grad} f(x)$, is a special vector field generated by a scalar field f.
If x^* is a local minimizer/maximizer, then $\operatorname{grad} f(x^*) = 0_{x^*}$

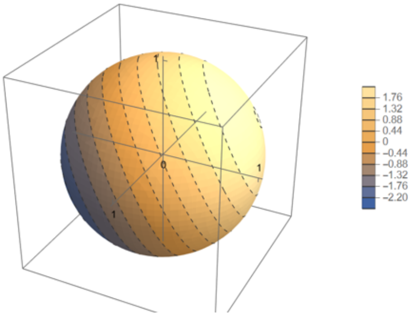


Figure: Contours of $f(x) = -x_1 + 2x_2 + x_3$ on $\mathbb{S}^2$.

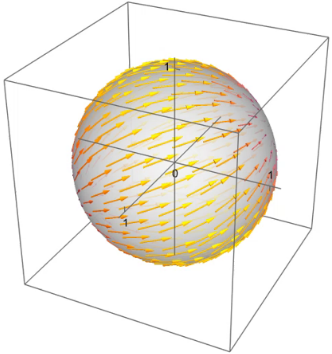
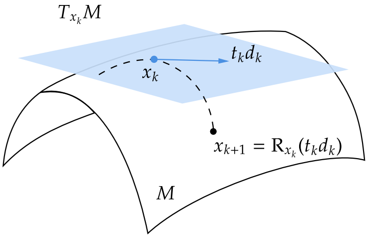


Figure: Gradient field of $f(x) = -x_1 + 2x_2 + x_3$ on $\mathbb{S}^2$.

Summary

Algorithm 4 Line Search Framework for solving $\min_{x \in \mathcal{M}} f(x)$.

Choose an initial point $x_0 \in \mathcal{M}$, a retraction R, and $k \leftarrow 0$;
repeat
  Compute a direction $d_k \in T_{x_k}\mathcal{M}$, e.g., $d_k = -\operatorname{grad} f(x)$;
  Compute a step length $t_k > 0$, e.g., Armijo condition;
  Compute the next point $x_{k+1} := \mathrm{R}_{x_k}(t_k d_k)$; ▷ update formula on manifold
until $\|\operatorname{grad} f(x_k)\|$ is close to 0



The covariant derivative of a vector field F on $\mathcal{M}$ is $\rightsquigarrow$

Riemannian connection

$$\nabla F(x) : T_x M \to T_x M, \ \textit{linear operator}.$$

general vector field

Example

If $\mathcal{M} = \mathbb{R}^n$, for a vector field $F : \mathbb{R}^n \to \mathbb{R}^n$, at $x \in \mathbb{R}^n$,

$$\nabla F(x) : T_x\mathbb{R}^n \equiv \mathbb{R}^n \to T_x\mathbb{R}^n \equiv \mathbb{R}^n, u \mapsto \mathbf{J}(x)u,$$

where $\mathbf{J}(x)$ is the $n \times n$ Jacobian matrix of F at x.

Second Order Algorithm: Riemannian Newton Method I

The covariant derivative of a vector field F on $\mathcal{M}$ is $\rightsquigarrow$

Riemannian connection

$$\nabla F(x) : T_x M \to T_x M, \ \textit{linear operator}.$$

general vector field

Algorithm 5 Riemannian Newton Method

Goal: To find singularity $x^* \in \mathcal{M}$ such that $F(x^*) = 0_{x^*} \in T_{x^*}\mathcal{M}$.
Take $x_0 \in \mathcal{M}$, and set $k = 0$.
repeat
  Solve a linear system on $T_{x_k}\mathcal{M} \ni v_k : \nabla F(x_k) v_k = -F(x_k)$,
  Compute $x_{k+1} := R_{x_k}(v_k)$;
until $\|F(x_{k+1})\|$ is efficiently close to zero

- It is a natural extension of the famous Newton method.
- Well-known convergence: the local superlinear/quadratic convergence also hold.

Specially, $\operatorname{Hess} f(x) \triangleq \nabla \operatorname{grad} f(x)$ is called Riemannian Hessian of $f : \mathcal{M} \to \mathbb{R}$ when $F = \operatorname{grad} f$.

(Proposition.) For any embedded submanifold $\mathcal{M}$, $\operatorname{Hess} f(x)[u] = \operatorname{Proj}_x(D \operatorname{grad} f(x)[u])$.

Example

For $f(x) = \frac{1}{2} x^T A x$ on $\mathbb{S}^{n-1}$, we have $\operatorname{grad} f(x) = (I_n - xx^T)Ax$. Its differential[a] is

$$D \operatorname{grad} f(x)[u] = Au - (u^T A x + x^T A u)x - (x^T A x)u;$$

project to the tangent space at x to reveal $\operatorname{Hess} f(x)[u] = Au - (x^T A u)x - (x^T A x)u$.

[a] Let $h : \mathcal{E} \to \mathcal{E}'$, the differential of h at x is $\mathrm{D}h(x) : \mathcal{E} \to \mathcal{E}'$, $\mathrm{D}h(x)[u] = \lim_{t \to 0} \frac{h(x+tu) - h(x)}{t}$.

- $\operatorname{Hess} f(x)$ is defined only on $T_x \mathbb{S}^{n-1}$ (not on all of $\mathbb{R}^n$).
- $\operatorname{Hess} f(x)$ is self-adjoint (i.e., symmetric) because $\operatorname{Hess} f(x) = \operatorname{Hess} f(x)^*$.

Second Order Algorithm: Riemannian Newton Method II

Recall: the optimal condition of $\min_{x\in\mathcal{M}} f(x)$ is

$$\operatorname{grad} f(x^*) = 0_{x^*} \in T_{x^*}\mathcal{M}.$$

Algorithm 6 Riemannian Newton Method for solving optimization problem $\min_{x\in\mathcal{M}} f(x)$

Take $x_0 \in \mathcal{M}$, and set $k = 0$.
repeat
  Solve a linear system on $T_{x_k}\mathcal{M} \ni \xi_k : \operatorname{Hess} f(x)\xi_k = -\operatorname{grad} f(x)$,
  Compute $x_{k+1} := R_{x_k}(\xi_k)$;
until $\|\operatorname{grad} f(x_{k+1})\|$ is efficiently close to zero

- It is a natural extension of the famous Newton method.
- Well-known convergence: the local superlinear/quadratic convergence also hold.

Summary: Framework of Riemannian Optimization

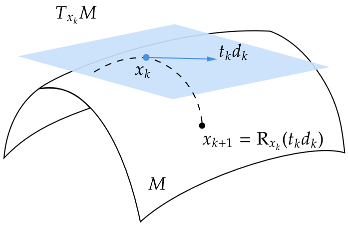
Riemannian optimization

Given an objective $f : \mathcal{M} \to \mathbb{R}$ where $\mathcal{M}$ is a Riemannian manifold, we want to solve

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

Algorithm 7 Line Search Framework for solving $\min_{x \in \mathcal{M}} f(x)$.

Choose an initial point $x_0 \in \mathcal{M}$, a retraction R, and $k \leftarrow 0$;
repeat
  Compute a direction $d_k \in T_{x_k}\mathcal{M}$;
  Compute a step length $t_k > 0$;
  Compute the next point $x_{k+1} := \mathrm{R}_{x_k}(t_k d_k)$;
until $\|\mathrm{grad} f(x_k)\|$ is close to 0



Summary: Unit Sphere Manifold

The set of all unit vectors, i.e., unit sphere,

$$\mathbb{S}^{n-1} := \{x \in \mathbb{R}^n : \|x\|_2 = 1\},$$

is an embedded submanifold of $\mathbb{R}^n$. Its tangent space at any $x \in \mathbb{S}^{n-1}$ is given by

$$T_x\mathbb{S}^{n-1} = \left\{u \in \mathbb{R}^n : x^T u = 0\right\},$$

and $\dim \mathbb{S}^{n-1} := \dim T_x\mathbb{S}^{n-1} = n - 1$. Then, the orthogonal projector to the tangent space at x is

$$\mathrm{Proj}_x : \mathbb{R}^n \to T_x\mathbb{S}^{n-1} : u \mapsto \mathrm{Proj}_x(u) = \left(I_n - xx^T\right) u = u - (x^T u)x.$$

One possible retraction on $\mathbb{S}^{n-1}$ is

$$\mathrm{R}_x(v) = \frac{x + v}{\|x + v\|} = \frac{x + v}{\sqrt{1 + \|v\|^2}}.$$

The Riemannian gradient of a smooth function $f : \mathbb{S}^{n-1} \to \mathbb{R}$ is given as

$$\mathrm{grad}\, f(x) = \mathrm{Proj}_x(\mathrm{egrad}\, f(x)) = \mathrm{egrad}\, f(x) - (x^T\, \mathrm{egrad}\, f(x))x.$$

For integers $p \leq n$, the set of all orthonormal matrices, i.e., Stiefel manifold,

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

is an embedded submanifold of $\mathbb{R}^{n\times p}$. Its tangent space at any $X \in \mathrm{St}(n,p)$ is given by

$$T_X \mathrm{St}(n,p) = \left\{V \in \mathbb{R}^{n\times p} : X^T V + V^T X = O\right\} = \{X\Omega + X_\perp B : \Omega \in \mathrm{Skew}(p), B \in \mathbb{R}^{(n-p)\times p}\},$$

and $\dim \mathrm{St}(n,p) := \dim T_X \mathrm{St}(n,p) = np - \frac{p(p+1)}{2}$. Then, the orthogonal projector is

$$\mathrm{Proj}_X : \mathbb{R}^{n\times p} \to T_X \mathrm{St}(n,p) : U \mapsto \mathrm{Proj}_X(U) = U - X\,\mathrm{sym}(X^T U),$$

where $\mathrm{sym}(Z) = \frac{Z+Z^T}{2}$ extracts the symmetric part of a matrix Z.

Summary: Stiefel Manifold (Cont'd)

Two possible retractions on $\mathrm{St}(n,p)$ are

- Retraction based on the polar decomposition of $X+V$:

$$\mathrm{R}_X(V) = (X+V)\left(I+V^TV\right)^{-1/2}.$$

  This is a projection retraction, namely, $\mathrm{R}_x(v) = \underset{x' \in \mathcal{M}}{\arg\min} \|x' - (x+v)\|$.

- Retraction based on the QR factorization of $X+V$:

$$R_X(V) = \mathrm{qf}(X+V),$$

  where $\mathrm{qf}(A)$ denotes the Q factor of the QR factorization.

The Riemannian gradient of a smooth function $f : \mathrm{St}(n,p) \to \mathbb{R}$ is given as

$$\mathrm{grad}\, f(X) = \mathrm{Proj}_X(\mathrm{egrad}\, f(X)) = \mathrm{egrad}\, f(X) - X\,\mathrm{sym}(X^T\,\mathrm{egrad}\, f(X)).$$

1. 想系统地学习流形优化的话，Nicolas Boumal 的教科书 “An introduction to optimization on smooth manifolds (2023)” 这一本书就足够了，并且不需微分几何作为前置知识。初次学习的阅读建议如下：

- 第 3 章 Embedded geometry: first order
- 第 4 章 First-order optimization algorithms
- 第 7 章 Embedded submanifolds: examples

如果研究只涉及一阶算法，这几章基本够用。

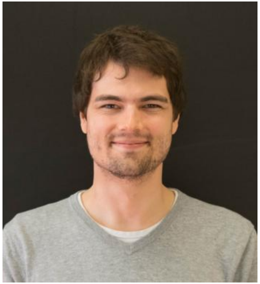


Figure: Nicolas Boumal, EPFL

2. Manopt 是最标准的流形优化软件，也是由 Nicolas Boumal 的团队开发的。可以配套地玩一玩。

3. Hiroyuki Sato 的教科书 “Riemannian Optimization and Its Applications (2021)” 着重介绍了黎曼共轭梯度法。其中，第 6 章总结了一些流形优化的前沿研究方向可供大家参考。

Recent Developments in Riemannian Optimization

- Stochastic Optimization
  - Riemannian Stochastic Gradient Descent Method
  - Riemannian Stochastic Variance Reduced Gradient Method
- Constrained Optimization on Manifolds
- Other Emerging Methods and Related Topics
  - Second-Order Methods
  - Nonsmooth Riemannian Optimization
  - Geodesic and Retraction Convexity
  - Multi-objective Optimization on Riemannian Manifolds

Figure: Hiroyuki Sato, Kyoto University

There have been some derivative-free optimization techniques specifically for manifolds.

- [Dre07] extended three popular direct search methods, namely, the Nelder-Mead simplex algorithm, the Mesh-Adapted Direct Search (MADS) algorithm, and frame-based methods, to Riemannian manifolds.
- [BIA10] proposed to adapt the particle swarm optimization algorithm on Grassmann manifolds to find the best low multilinear rank approximation for a given tensor.
- A Derivative-Free Riemannian Powell's Method, Minimizing Hartley-Entropy-Based ICA Contrast. [CSA15]
- Stochastic Derivative-Free Optimization on Riemannian Manifolds. [FT22b]
- Learning-to-Learn to Guide Random Search: Derivative-Free Meta Blackbox Optimization on Manifold. [STD+23]
- Stochastic zeroth-order Riemannian derivative estimation and optimization. [LBM23]

A Tutorial on Riemannian Optimization

Thank you for listening!
Any questions?

[ABG07] P-A Absil, Christopher G Baker, and Kyle A Gallivan.
Trust-region methods on Riemannian manifolds.
Foundations of Computational Mathematics, 7:303–330, 2007.

[AMS08] P-A Absil, Robert Mahony, and Rodolphe Sepulchre.
Optimization Algorithms on Matrix Manifolds.
Princeton University Press, 2008.

[Ber22] Ronny Bergmann.
Manopt.jl: Optimization on manifolds in Julia.
Journal of Open Source Software, 7(70):3866, 2022.

Reference

[BIA10] Pierre B Borckmans, Mariya Ishteva, and Pierre-Antoine Absil.
A modified particle swarm optimization algorithm for the best low multilinear rank approximation of higher-order tensors.
In Marco Dorigo, Mauro Birattari, Gianni A. Di Caro, René Doursat, Andries P. Engelbrecht, Dario Floreano, Luca Maria Gambardella, Roderich Groß, Erol Şahin, Hiroki Sayama, and Thomas Stützle, editors, *Swarm Intelligence*, pages 13–23, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[BMAS14] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre.
Manopt, a Matlab toolbox for optimization on manifolds.
Journal of Machine Learning Research, 15(42):1455–1459, 2014.

[Bou23] Nicolas Boumal.
An Introduction to Optimization on Smooth Manifolds.
Cambridge University Press, 1 edition, March 2023.

[CMMCSZ20] Shixiang Chen, Shiqian Ma, Anthony Man-Cho So, and Tong Zhang.
Proximal gradient method for nonsmooth optimization over the stiefel manifold.
SIAM Journal on Optimization, 30(1):210–239, 2020.

[CSA15] Amit Chattopadhyay, Suviseshamuthu Easter Selvan, and Umberto Amato.
A derivative-free Riemannian Powell's method, minimizing Hartley-entropy-based ICA contrast.
IEEE Transactions on Neural Networks and Learning Systems, 27(9):1983–1990, 2015.

[Dre07] David W Dreisigmeyer.
Direct search algorithms over Riemannian manifolds.
Optimization Online, 2007.

[FT22a] Robert Simon Fong and Peter Tino.
Population-Based Optimization on Riemannian Manifolds.
Springer, 2022.

[FT22b] Robert Simon Fong and Peter Tino.
Stochastic derivative-free optimization on riemannian manifolds.
In *Population-Based Optimization on Riemannian Manifolds*, pages 105–137. Springer, 2022.

[FWL+23] Yanhong Fei, Xian Wei, Yingjie Liu, Zhengyu Li, and Mingsong Chen.
A survey of geometric optimization for deep learning: From euclidean space to riemannian manifold.
arXiv preprint arXiv:2302.08210, 2023.

[HAGH18] Wen Huang, P-A Absil, Kyle A Gallivan, and Paul Hand.
ROPTLIB: An object-oriented C++ library for optimization on Riemannian manifolds.
ACM Transactions on Mathematical Software, 44(4):1–21, 2018.

[HGSA16] Wen Huang, Kyle A Gallivan, Anuj Srivastava, and Pierre-Antoine Absil.
Riemannian optimization for registration of curves in elastic shape analysis.
Journal of Mathematical Imaging and Vision, 54:320–343, 2016.

[HLWY20] Jiang Hu, Xin Liu, Zai-Wen Wen, and Ya-Xiang Yuan.
A brief introduction to manifold optimization.
Journal of the Operations Research Society of China, 8(2):199–248, June 2020.

[KGB16] Artiom Kovnatsky, Klaus Glashoff, and Michael M Bronstein.
MADMM: A generic algorithm for non-smooth optimization on manifolds.
In Bastianand Leibe, Jiriand Matas, Nicuand Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 680–696, Cham, 2016. Springer International Publishing.

[KKK20] Max Kochurov, Rasul Karimov, and Serge Kozlukov.
Geoopt: Riemannian optimization in pytorch.
arXiv:2005.02819, 2020.

[LBM23] Jiaxiang Li, Krishnakumar Balasubramanian, and Shiqian Ma.
Stochastic zeroth-order riemannian derivative estimation and optimization.
Mathematics of Operations Research, 48(2):1183–1211, 2023.

Reference

4 Summary

[LRFO21] Ilia Luchnikov, Alexander Ryzhov, Sergey Filippov, and Henni Ouerdane.
QGOpt: Riemannian optimization for quantum technologies.
SciPost Physics, 10:79, 2021.

[MGLB+20] Nina Miolane, Nicolas Guigui, Alice Le Brigant, Johan Mathe, Benjamin Hou, Yann Thanwerdas, Stefan Heyder, Olivier Peltre, Niklas Koep, Hadi Zaatiti, et al.
Geomstats: A Python package for Riemannian geometry in machine learning.
Journal of Machine Learning Research, 21(223):1–9, 2020.

[MJK+18] Mayank Meghwanshi, Pratik Jawanpuria, Anoop Kunchukuttan, Hiroyuki Kasai, and Bamdev Mishra.
McTorch, a manifold optimization library for deep learning.
arXiv:1810.01811, 2018.

[MRHA20] Sean Martin, Andrew M. Raim, Wen Huang, and Kofi P. Adragni.
ManifoldOptim: An R interface to the ROPTLIB library for Riemannian manifold optimization.
Journal of Statistical Software, 93(1):1–32, 2020.

Reference

[Sat21] Hiroyuki Sato.
Riemannian Optimization and Its Applications.
SpringerBriefs in Electrical and Computer Engineering. Springer Cham, 2021.

[Sat22] Hiroyuki Sato.
Riemannian conjugate gradient methods: General framework and specific algorithms with convergence analyses.
SIAM Journal on Optimization, 32(4):2690–2717, 2022.

[Smi21] Oleg Smirnov.
TensorFlow RiemOpt: A library for optimization on Riemannian manifolds.
arXiv:2105.13921, 2021.

[STD+23] Bilgehan Sel, Ahmad Tawaha, Yuhao Ding, Ruoxi Jia, Bo Ji, Javad Lavaei, and Ming Jin.
Learning-to-learn to guide random search: Derivative-free meta blackbox optimization on manifold.
In *Learning for Dynamics and Control Conference*, pages 38–50. PMLR, 2023.

[TG21] Nickolay Trendafilov and Michele Gallo.
Multivariate Data Analysis on Matrix Manifolds.
Springer, 1 edition, 2021.

[TKW16] James Townsend, Niklas Koep, and Sebastian Weichwald.
Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation.
Journal of Machine Learning Research, 17(137):1–5, 2016.

[Udr94] Constantin Udriste.
Convex Functions and Optimization Methods on Riemannian manifolds.
Springer Science & Business Media, 1994.

[UHJM22] Saiteja Utpala, Andi Han, Pratik Jawanpuria, and Bamdev Mishra.
Rieoptax: Riemannian optimization in JAX.
In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022.

[Van13] Bart Vandereycken.
Low-rank matrix completion by Riemannian optimization.
SIAM Journal on Optimization, 23(2):1214–1236, 2013.

[XHLT22] Nachuan Xiao, Xiaoyin Hu, Xin Liu, and Kim-Chuan Toh.
CDOpt: A python package for a class of Riemannian optimization.
arXiv:2212.02698, 2022.

[ZJRS16] Hongyi Zhang, Sashank J Reddi, and Suvrit Sra.
Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds.
In *Advances in Neural Information Processing Systems*, volume 29, 2016.

- Appendix

Grassmannian Manifold as a Quotient Manifold

Grassmannian manifold is the set of linear subspaces of dimension p in $\mathbb{R}^n$,

$$\mathrm{Gr}(n,p) = \left\{\mathrm{span}(X) : X \in \mathbb{R}^{n\times p}, X^T X = I_p\right\}.$$

We define an equivalence relation $\sim$ over $\mathrm{St}(n,p) = \left\{X \in \mathbb{R}^{n\times p} : X^T X = I_p\right\}$ as below.

$$X \sim Y \Leftrightarrow \mathrm{span}(X) = \mathrm{span}(Y) \Leftrightarrow X = YQ \text{ for some } Q \in O(p),$$

where $\mathrm{O}(p)$ is the orthogonal group. Formally, if $L = \mathrm{span}(X)$, we identify L with

$$[X] = \{Y \in \mathrm{St}(n,p) : Y \sim X\}$$

This identification establishes a one-to-one correspondence between $\mathrm{Gr}(n,p)$ and the quotient set

$$\mathrm{St}(n,p)/\sim = \{[X] : X \in \mathrm{St}(n,p)\}.$$

Principal Component Analysis (PCA):
Given k points $y_1, \ldots, y_k \in \mathbb{R}^n$, the goal of PCA is to find a linear subspace $L \in \mathrm{Gr}(n, p)$ which fits the data $y_1, \ldots, y_k$ as well as possible, in the sense that it solves

$$\min_{L \in \mathrm{Gr}(n,p)} \sum_{i=1}^{k} \operatorname{dist}\left(L, y_i\right)^2 ,$$

where $\operatorname{dist}(L, y)$ is the Euclidean distance between y and the point in L closest to y.[4]
General objective function: We may need more general optimization algorithms to address:

$$\min_{L \in \mathrm{Gr}(n,p)} f(L),$$

where objective function $f : \mathrm{Gr}(n, p) \to \mathbb{R}$. Clearly, Euclidean optimization cannot solve these problems unless we convert the problem into some equivalent Euclidean problem.

[4] This objective function admits an explicit solution involving the SVD of the data matrix $M = [y_1, \ldots, y_k]$. However, this is not the case for other objective functions.

Riemannian Metric Induces the Distance Space

The norm of a tangent vector ξ at any point x on $\mathcal{M}$ can be defined as

$$\|\xi\|_x := \sqrt{\langle \xi, \xi \rangle_x}$$

Furthermore, the length $L(c)$ of a curve $c : [a, b] \to \mathcal{M}$ on $\mathcal{M}$ can be defined as

$$L(c) := \int_a^b \|c'(t)\|_{c(t)} \, dt.$$

A natural distance on $\mathcal{M}$, called the Riemannian distance,

$$\operatorname{dist}(x, y) := \inf_c L(c)$$

where the infimum is taken over all curve segments which connect x to y, and thus $\mathcal{M}$ becomes a distance space.

What is the Manifold? (Strict Definitions)

A $\boldsymbol{d}$-dimensional (smooth) manifold is a topological space $\mathcal{M}$ satisfying the following three properties:

(1) $\mathcal{M}$ is second-countable and Hausdorff.

(2) $\mathcal{M}$ is locally Euclidean of dimension d (i.e., each point of $\mathcal{M}$ has a neighborhood U and a homeomorphism $\varphi : U \to V$ from U to an open set V in $\mathbb{R}^d$).

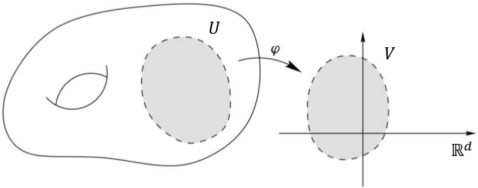


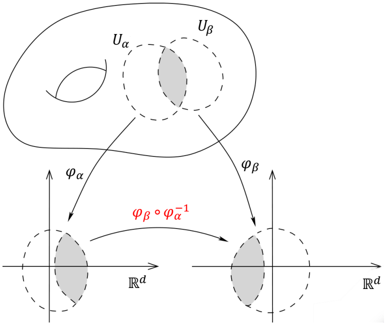
Figure: The pair (U, φ) is called a chart.

What is the Manifold? (Strict Definitions) (Cont'd)

(3) there is a family $\{(U_\lambda, \varphi_\lambda)\}_{\lambda\in\Lambda}$ with $\mathcal{M} = U_{\lambda\in\Lambda} U_\lambda$ such that for any $\alpha, \beta \in \Lambda$ with $U_\alpha \cap U_\beta \neq \emptyset$, the coordinate transformation

$$\varphi_\beta \circ \varphi_\alpha^{-1} : \varphi_\alpha \left(U_\alpha \cap U_\beta\right) \subseteq \mathbb{R}^d \to \varphi_\beta \left(U_\alpha \cap U_\beta\right) \subseteq \mathbb{R}^d$$

is of class C^∞.



The property (3) makes the consistent smoothness across all charts by $f \circ \varphi_\alpha^{-1} = (f \circ \varphi_\beta^{-1}) \circ (\varphi_\beta \circ \varphi_\alpha^{-1})$ because we say a function $f : \mathcal{M} \to \mathbb{R}$ is smooth at $p \in \mathcal{M}$ if there exists a chart (U, φ) such that $f \circ \varphi^{-1}$ is of class C^∞ at $\varphi(p)$.

click here for full doctoral thesis of zhijian lai

Table 1.1 Collection of some available manifolds in Manopt.

Name of Manifold	**Mathematical Formulation**
(Complex) Euclidean Space	$\mathbb{R}^{m\times n}, \mathbb{C}^{m\times n}$
Symmetric Matrices	$\{X \in \mathbb{R}^{n\times n} : X = X^T\}$
Skew-Symmetric Matrices	$\{X \in \mathbb{R}^{n\times n} : X + X^T = 0\}$
Centered Matrices	$\{X \in \mathbb{R}^{m\times n} : X\mathbf{1}_n = 0_m\}$
Sphere	$\{X \in \mathbb{R}^{m\times n} : \|X\|_{\mathrm{F}} = 1\}$
Symmetric Sphere	$\{X \in \mathbb{R}^{n\times n} : \|X\|_{\mathrm{F}} = 1, X = X^T\}$
Complex Sphere	$\{X \in \mathbb{C}^{m\times n} : \|X\|_{\mathrm{F}} = 1\}$
Oblique Manifold	$\{X \in \mathbb{R}^{m\times n} : \|X_{:,1}\|_{\mathrm{F}} = \cdots = \|X_{:,n}\|_{\mathrm{F}} = 1\}$
Complex Oblique Manifold	$\{X \in \mathbb{C}^{m\times n} : \|X_{:,1}\|_{\mathrm{F}} = \cdots = \|X_{:,n}\|_{\mathrm{F}} = 1\}$
Complex Circle	$\{z \in \mathbb{C}^n : \|z_1\| = \cdots = \|z_n\| = 1\}$
Phase of Real DFT	$\{z \in \mathbb{C}^n : \|z_k\| = 1, z_{1+\bmod (k,n)} = \bar{z}_{1+\bmod (n-k,n)}, \forall k\}$
Stiefel Manifold	$\{X \in \mathbb{R}^{n\times p} : X^TX = I\}$
Complex Stiefel Manifold	$\{X \in \mathbb{C}^{n\times p} : X^*X = I\}$
Generalized Stiefel Manifold	$\{X \in \mathbb{R}^{n\times p} : X^TBX = I\}$ for some $B \succ 0$
Grassmann Manifold	$\{\mathrm{span}(X) : X \in \mathbb{R}^{n\times p}, X^TX = I\}$
Complex Grassmann Manifold	$\{\mathrm{span}(X) : X \in \mathbb{C}^{n\times p}, X^*X = I\}$
Generalized Grassmann Manifold	$\{\mathrm{span}(X) : X \in \mathbb{R}^{n\times p}, X^TBX = I\}$ for some $B \succ 0$
Rotation Group	$\{R \in \mathbb{R}^{n\times n} : R^TR = I, \det(R) = 1\}$
Special Euclidean Group	$\{(R,t) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n : R^TR = I, \det(R) = 1\}$
Unitary Matrices	$\{U \in \mathbb{C}^{n\times n} : U^*U = I_n\}$
Hyperbolic manifold	$\{x \in \mathbb{R}^{n+1} : x_0^2 = x_1^2 + \cdots + x_n^2 + 1\}$ with Minkowski metric
Fixed-Rank Manifold	$\{X \in \mathbb{R}^{m\times n} : \mathrm{rank}(X) = k\}$
Fixed-Rank Tensor, Tucker	Tensors of fixed multilinear rank in Tucker format
Strictly Positive Matrices	$\{X \in \mathbb{R}^{m\times n} : X_{ij} > 0, \forall i,j\}$
Symmetric Positive Definite Matrices	$\{X \in \mathbb{R}^{n\times n} : X = X^T, X \succ 0\}$
-	$\{X \in \mathbb{R}^{n\times n} : X = X^T \succeq 0, \mathrm{rank}(X) = k\}$
-	$\{X \in \mathbb{R}^{n\times n} : X = X^T \succeq 0, \mathrm{rank}(X) = k, \mathrm{diag}(X) = \mathbf{1}\}$
-	$\{X \in \mathbb{R}^{n\times n} : X = X^T \succeq 0, \mathrm{rank}(X) = k, \mathrm{trace}(X) = 1\}$
Multinomial manifold	$\{X \in \mathbb{R}^{m\times n} : X_{ij} > 0, \forall i,j \text{ and } X\mathbf{1}_n = \mathbf{1}_m\}$
-	$\{X \in \mathbb{R}^{n\times n} : X_{ij} > 0, \forall i,j \text{ and } X\mathbf{1}_n = \mathbf{1}_n, X^T\mathbf{1}_n = \mathbf{1}_n\}$
-	$\{X \in \mathbb{R}^{n\times n} : X_{ij} > 0, \forall i,j \text{ and } X\mathbf{1}_n = \mathbf{1}_n, X = X^T\}$
Positive Definite Simplex	$\{(X_1, 2, \ldots, x_k) \in \mathbb{R}^{n\times n} : X_i \succ 0, \forall i \text{ and } X_1 + \cdots + x_k = I_n\}$
Complex Positive Definite Simplex	$\{(X_1, 2, \ldots, x_k) \in \mathbb{C}^{n\times n} : X_i \succ 0, \forall i \text{ and } X_1 + \cdots + x_k = I_n\}$
Sparse Matrices of Fixed Sparsity Pattern	$\{X \in \mathbb{R}^{m\times n} : X_{ij} = 0 \Leftrightarrow A_{ij} = 0\}$
Constant Manifold (singleton)	$\{A\}$