Riemannian Interior Point Methods for Constrained Optimization on Manifolds

赖志坚

北京大学　北京国际数学研究中心

June 6, 2024

Riemannian Interior Point Methods (RIPM)

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 $h_j(x) \leq 0, j = 1, 2, \ldots, l\}$.

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 $k \to k + 1$;
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.

New Insight on (Un)constrained Optimization Problem

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 $\mathbb{R}^n$.
- $S \subsetneq \mathbb{R}^n$. 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 $\mathrm{UPDATE}\colon S \times D_k \times \mathbb{R}^+ \to S$, and D_k consists 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 algorithms 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, $\{X \in \mathbb{R}^{n \times m} : \operatorname{rank}(X) = r\}$.
- 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

- 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.

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

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

Application II: Sparse PCA

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

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

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

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

Application III: Low-Rank Matrix Completion

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

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

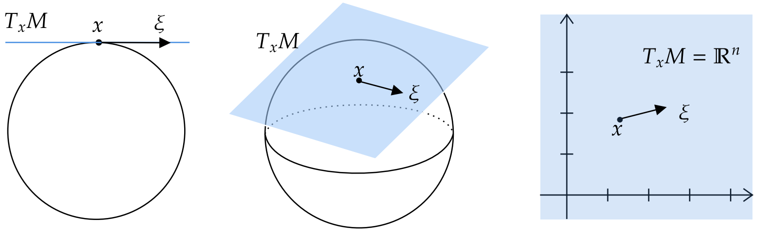
$$\begin{array}{ll} \min_X & \mathrm{rank}(X) \\ \text{s.t.} & X_{ij} = M_{ij}, \quad (i,j) \in \Omega. \end{array} \tag{7}$$

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

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

Riemannian Manifold = Manifold + Riemannian Metric

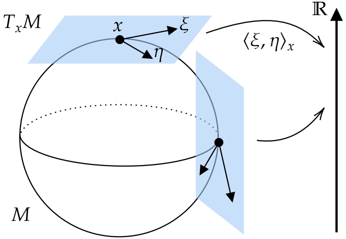
- A manifold $\mathcal{M}$ is a set that can be locally linearized.[1]
  - $T_x\mathcal{M}$ is tangent space at x.
  - $\xi \in T_x\mathcal{M}$ is tangent vector at x.
- A Riemannian metric $\langle\cdot,\cdot\rangle$ assigns an inner product $\langle\cdot,\rangle_x : T_x\mathcal{M} \times T_x\mathcal{M}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

Riemannian Manifold = Manifold + Riemannian Metric

- 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 $\langle \cdot, \rangle_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ 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.

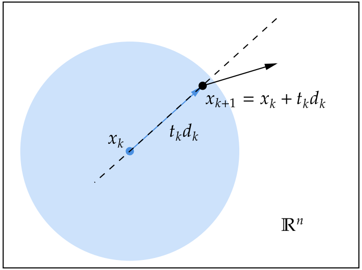
Euclidean Optimization v.s. Riemannian Optimization

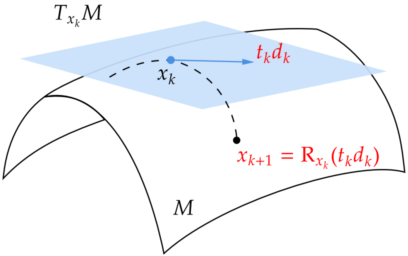
Algorithm 2 Line Search Framework for $S = \mathbb{R}^n$

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 $x_{k+1} := R_{x_k}(t_k d_k)$;





Advantages in Comparison to Euclidean Optimization

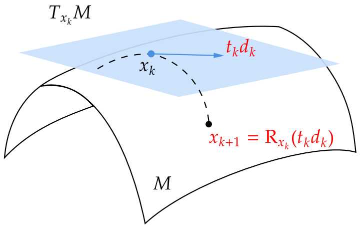
Riemannian version of classical methods:

- Riemannian steepest decent
- Riemannian conjugate gradient
- Riemannian trust region
- Riemannian Newton
- Riemannian BFGS
- Riemannian proximal gradient
- Riemannian stochastic algorithms
- Riemannian ADMM
- and more

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

Advantages of Riemannian optimization:

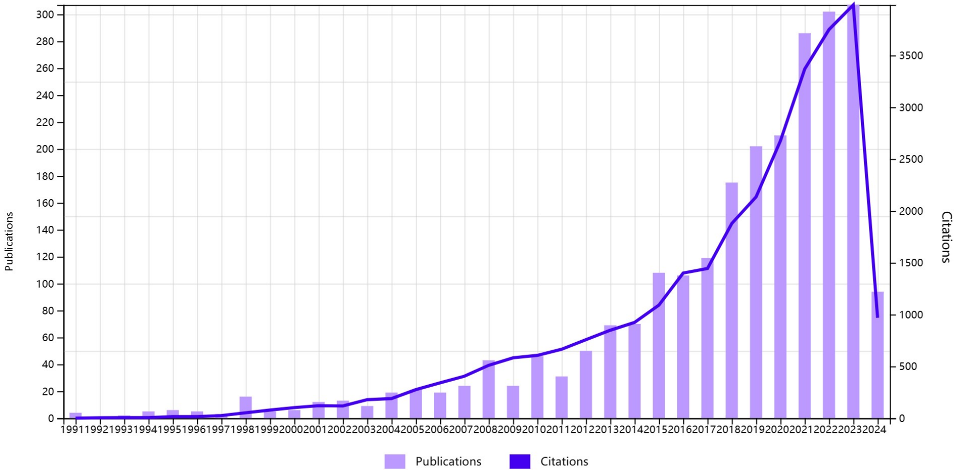
1. All iterates on the manifold.
2. Transform constrained problems into unconstrained ones.
3. Use of the geometric structure of the feasible region.
4. Convergence properties of like optimization on Euclidean space.



Citation Report: Riemannian Optimization (Topic)

1 A Tutorial on Riemannian Optimization

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 [FT22]
  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/

Riemannian Optimization Libraries

- 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

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

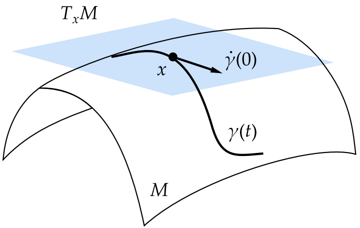
Remark

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

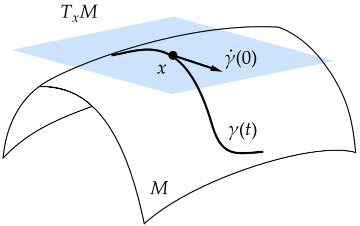
$$\dot{\gamma}(0) := \lim_{t \to 0} \frac{\gamma(t) - \gamma(0)}{t} = \frac{d}{dt}\gamma(t)\Big|_{t=0}$$

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, $\gamma(0) = x\}$.



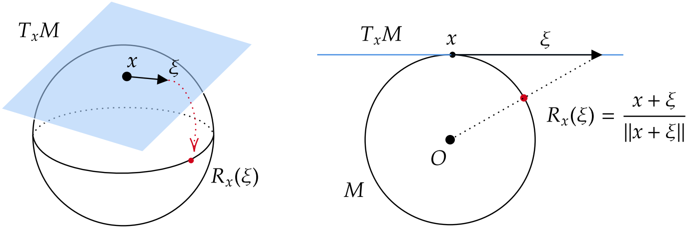
(1) For any $x \in \mathcal{M}$, $T_x\mathcal{M}$ are linear spaces sharing the same dimension.
(2) In general, $T_x\mathcal{M}$ is determined by x, except for $T_x\mathbb{R}^n \cong \mathbb{R}^n$.
(3) For embedded submanifold, $T_x\mathcal{M}$ is a subspace of $\mathbb{R}^n$, e.g., $T_x\mathbb{S}^{n-1}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

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

$T\mathcal{M} = \{(x, \xi) : x \in \mathcal{M}$ 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 $\dot{\gamma}(0) = \xi$.



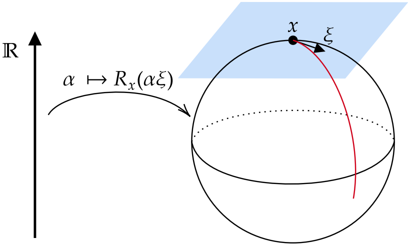
A retraction R yields a map $\mathrm{R}_x : T_x\mathcal{M} \to \mathcal{M}$ for any x.

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, $\alpha \mapsto \mathrm{R}_x(\alpha\xi)$ defines a curve along this direction.

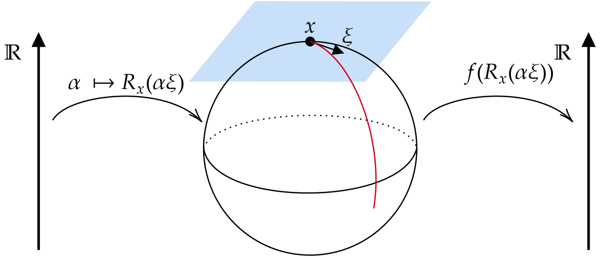


Euclidean setting	Riemannian setting
$x_{k+1} = x_k + \alpha_k d_k$	$x_{k+1} = R_{x_k}(\alpha_k \xi_k)$

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.

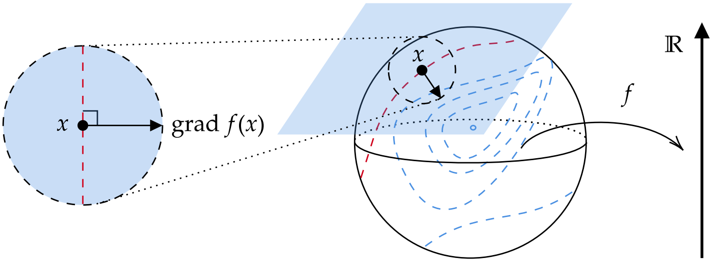


The Riemannian gradient, $\operatorname{grad} f(x)$, is the tangent vector at x such that:

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

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:

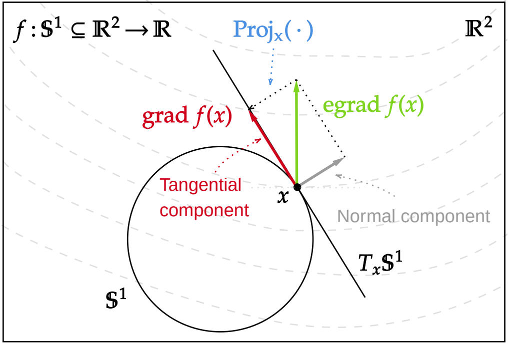
$$\operatorname{grad} f(x) = \operatorname{Proj}_x(\nabla f(x)).$$

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 $x \in \mathcal{M}$.

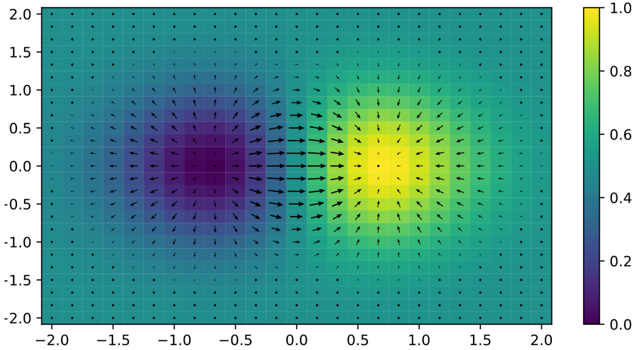


Figure: Let $\mathcal{M} = \mathbb{R}^2$. Gradient of the 2D function $f(x, y) = xe^{-(x^2+y^2)}$. Source: Wikipedia.

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 $x \in \mathcal{M}$.

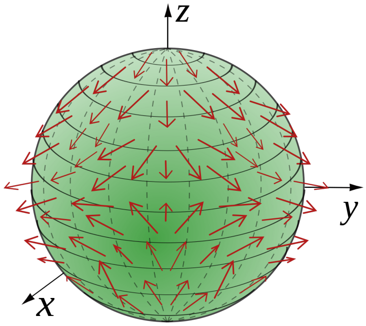


Figure: A vector field on a sphere $\mathbb{S}^2$. Source: Wikipedia.

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 $\operatorname{grad} f(x^*) = 0_{x^*}$

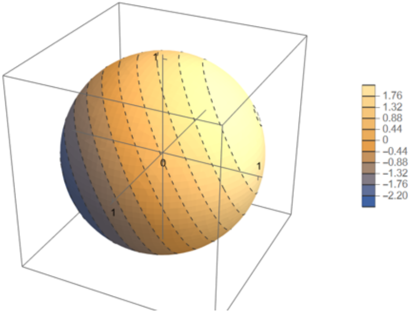


Figure: Contours of $f(x) = -x_1 + 2x_2 + x_3$ on $\mathbb{S}^2$.

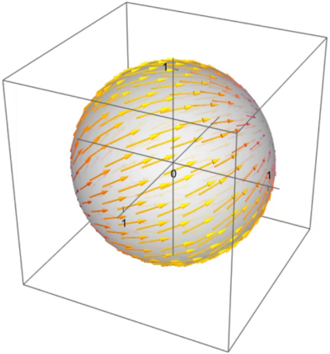


Figure: Gradient field of $f(x) = -x_1 + 2x_2 + x_3$ on $\mathbb{S}^2$.

Summary: Framework of Riemannian Optimization

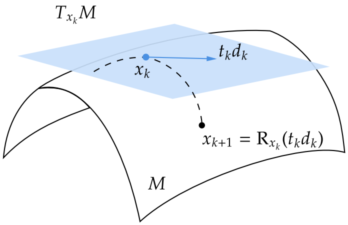
Riemannian optimization

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 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



Second Order Geometry: Covariant Derivative

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

Riemannian connection

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

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.

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

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.

Second Order Geometry: Riemannian Hessian

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

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

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

[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.

Riemannian Interior Point Methods (RIPM)

More Requirements from Applications

- Nonnegative PCA on Stiefel manifold, $\mathrm{St}(n,k) = \{X \in \mathbb{R}^{n\times k} : X^\top X = I\}$:

$$\min_{X\in \mathrm{St}(n,k)} \; -\operatorname{trace}(X^\top A^\top AX) \quad \text{s.t. } X \geq 0$$

- Nonnegative matrix completion on fixed rank manifold, $\mathbb{R}_r^{m\times n} = \{X \in \mathbb{R}^{m\times n} : \operatorname{rank}(X) = r\}$:

$$\min_{X\in \mathbb{R}_r^{m\times n}} \sum_{(i,j)\in\Omega} (X_{ij} - A_{ij})^2 \quad \text{s.t. } X \geq 0$$

What should we do now?

New Challenges

Unconstrained Riemannian Optimization (URO):

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

where $\mathcal{M}$ is a Riemannian manifold and $f\colon \mathcal{M} \to \mathbb{R}$.

Weakness of (URO):

1. It requires the entire feasible region to form exactly one manifold.
2. Adding new constraints does not necessarily guarantee that the entire feasible region is still a manifold.
3. Even if the entire feasible region is proven to be a manifold, there are no available software packages.

We are attempting to develop a new model, called (CRO), to address these issues.

New Model — Constrained Riemannian Optimization (CRO)

Constrained Riemannian Optimization (CRO):

$$\min_{x \in \mathcal{M}} \quad f(x) \qquad \text{s.t.} \quad h(x) = 0, \text{ and } g(x) \leq 0, \tag{CRO}$$

where $h : \mathcal{M} \to \mathbb{R}^l$, and $g : \mathcal{M} \to \mathbb{R}^m$.

Advantages of (CRO):

1. Still using the geometric structure of $\mathcal{M}$. The advantages of Riemannian optimization are maintained.
2. Very flexible, even if the constraints of h, g cannot form a new manifold.

Existing Riemannian Algorithms for (CRO)

- **Riemannian Augmented Lagrangian Method** [LB20, YS22]:

$$\mathcal{L}_\rho(x,\lambda,\gamma) := f(x) + \frac{\rho}{2}\left(\sum_j \left(h_j(x) + \frac{\gamma_j}{\rho}\right)^2 + \sum_i \max\left\{0, \frac{\lambda_i}{\rho} + g_i(x)\right\}^2\right),$$

where $x \in \mathcal{M}$, $\rho > 0$ is a penalty parameter and $\gamma \in \mathbb{R}^l, \lambda \in \mathbb{R}^m, \lambda \geq 0$ are Lagrangian multipliers. It alternates between updating x and updating (λ, γ, ρ).

- **Riemannian Exact Penalty Method** [LB20]:

$$\min_{x \in \mathcal{M}} f(x) + \rho\left(\sum_i \max\{0, g_i(x)\} + \sum_j |h_j(x)|\right)$$

- **Riemannian Sequential Quadratic Programming method** [SO21, OOT20]: At each iteration, we solve

$$\begin{array}{ll} \min_{\Delta x_k \in T_{x_k}\mathcal{M}} & \frac{1}{2}\langle B_k[\Delta x_k], \Delta x_k\rangle + \langle \operatorname{grad} f(x_k), \Delta x_k\rangle \\ \text{s.t.} & g_i(x_k) + \langle \operatorname{grad} g_i(x_k), \Delta x_k\rangle \leq 0, i = 1,2,\ldots,m, \\ & h_j(x_k) + \langle \operatorname{grad} h_j(x_k), \Delta x_k\rangle = 0, j = 1,2,\ldots,l. \end{array}$$

where $B_k : T_{x_k}\mathcal{M} \to T_{x_k}\mathcal{M}$.

Riemannian Interior Point Methods (RIPM)

Intuition of Interior Point Methods on Manifolds

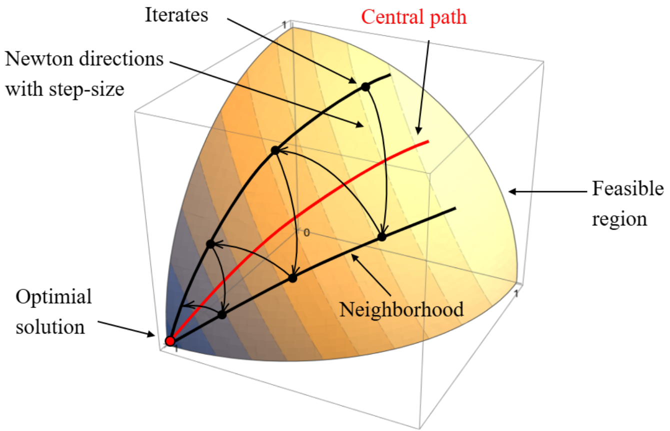


Figure: Image of *primal* interior point method, not primal-dual.

Formulation of RIPM

Constrained Riemannian Optimization (CRO):

$$\begin{aligned} \min_{x \in \mathcal{M}} \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0, \text{ and } g(x) \leq 0, \end{aligned} \tag{CRO}$$

where $h : \mathcal{M} \to \mathbb{R}^l$, and $g : \mathcal{M} \to \mathbb{R}^m$.

Lagrangian function is

$$\mathcal{L}(x, y, z) \triangleq f(x) + y^T h(x) + z^T g(x). \tag{10}$$

Fixing y and z, $x \mapsto \mathcal{L}(x, y, z)$ is a real-valued function on $\mathcal{M}$, then we have

- $\operatorname{grad}_x \mathcal{L}(x, y, z) = \operatorname{grad} f(x) + \sum_{i=1}^{l} y_i \operatorname{grad} h_i(x) + \sum_{i=1}^{m} z_i \operatorname{grad} g_i(x)$,
- $\operatorname{Hess}_x \mathcal{L}(x, y, z) = \operatorname{Hess} f(x) + \sum_{i=1}^{l} y_i \operatorname{Hess} h_i(x) + \sum_{i=1}^{m} z_i \operatorname{Hess} g_i(x)$.

KKT Vector Field — F(w)

Riemannian KKT conditions [LB20] are

$$\begin{cases} \operatorname{grad}_x \mathcal{L}(x,y,z) = 0_x, \\ h(x) = 0, \\ g(x) \leq 0, \\ Zg(x) = 0, \ (Z := \operatorname{diag}(z_1,\ldots,z_m)) \\ z \geq 0. \end{cases} \tag{11}$$

Using $s := -g(x)$, the above becomes

$$F(w) \triangleq \begin{pmatrix} \operatorname{grad}_x \mathcal{L}(x,y,z) \\ h(x) \\ g(x) + s \\ ZSe \end{pmatrix} = 0_w := \begin{pmatrix} 0_x \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{and } (z,s) \geq 0, \tag{12}$$

where $w := (x,y,z,s) \in \mathscr{M} \triangleq \mathcal{M} \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}^m$. Note that $T_w\mathscr{M} \equiv T_x\mathcal{M} \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}^m$.

Covariant Derivative of KKT Vector Field

For each $x \in \mathcal{M}$, we define

$$H_x : \mathbb{R}^l \to T_x\mathcal{M}, \quad H_x v \triangleq \sum_{i=1}^{l} v_i \operatorname{grad} h_i(x). \tag{13}$$

Hence, the adjoint operator is

$$H_x^* : T_x\mathcal{M} \to \mathbb{R}^l, \quad H_x^* \xi = \left[\langle \operatorname{grad} h_1(x), \xi \rangle_x, \cdots, \langle \operatorname{grad} h_l(x), \xi \rangle_x\right]^T. \tag{14}$$

Lemma (Lai. 2024)

The linear operator $\nabla F(w) : T_w\mathscr{M} \to T_w\mathscr{M}$ is given by

$$\nabla F(w)\Delta w = \begin{pmatrix} \operatorname{Hess}_x \mathcal{L}(w)\Delta x + H_x\Delta y + G_x\Delta z \\ H_x^* \Delta x \\ G_x^* \Delta x + \Delta s \\ Z\Delta s + S\Delta z \end{pmatrix}, \tag{15}$$

where $\Delta w = (\Delta x, \Delta y, \Delta s, \Delta z) \in T_x\mathcal{M} \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}^m \equiv T_w\mathscr{M}$.

Riemannian Interior Point Method (RIPM)

Step 0. Initial w_0 with $(z_0, s_0) > 0$.
Step 1. Solve

$$\nabla F(w_k)\Delta w_k = -F(w_k) + \mu_k \hat{e}, \tag{16}$$

where $\hat{e} \triangleq (0_x, 0, 0, e)$.
Step 2. Compute the step sizes α_k such that $(z_{k+1}, s_{k+1}) > 0$.
Step 3. Update:

$$w_{k+1} = \bar{R}_{w_k}(\alpha_k \Delta w_k). \tag{17}$$

Step 4. Let $\mu_k \to 0$. Return to 1.

Theorem (Local Convergence, Lai. 2024)

Under some standard assumptions.

(1) *If $\mu_k = o(\|F(w_k)\|)$, $\alpha_k \to 1$, then $\{w_k\}$ locally, superlinearly converges to w^*.*

(2) *If $\mu_k = O(\|F(w_k)\|^2)$, $1 - \alpha_k = O(\|F(w_k)\|)$, then $\{w_k\}$ locally, quadratically converges to w^*.*

Global Line Search RIPM Algorithm

Merit function: Choose $\varphi(w) \triangleq \|F(w)\|^2$.

Backtracking for step size α_k:

1. Centrality conditions.
2. Sufficient decreasing condition.

With a slight abuse of notation, we also let

$$\varphi(\alpha) \triangleq \varphi(\underbrace{\bar{R}_{w_k}(\alpha \Delta w_k)}_{\text{new iterate}}) \text{ for fixed } w_k \text{ and } \Delta w_k, \tag{18}$$

then $\varphi(0) = \varphi(w_k) =: \varphi_k$ and $\varphi'(0) = \langle \operatorname{grad} \varphi(w_k), \Delta w_k \rangle$. Sufficient decreasing asks

$$\varphi(\alpha_k) - \varphi(0) \leq \alpha_k \beta \varphi'(0).$$

Descent direction: Let Δw_k be the solution of $\nabla F(w_k)\Delta w_k = -F(w_k) + \rho_k \sigma_k \hat{e}$, then $\varphi'(0) < 0$ if we set $\rho_k := s_k^T z_k / m, \sigma_k \in (0,1)$. Then, $\{\varphi_k\}$ is monotonically decreasing.

Global Convergence

Assumptions:[3]

1. the functions $f(x), h(x), g(x)$ are smooth; the set $\{\operatorname{grad} h_i(x)\}_{i=1}^{l}$ is linearly independent in $T_x\mathcal{M}$ for all x; and $w \mapsto \nabla F(w)$ is Lipschitz continuous;
2. the sequences $\{x_k\}$ and $\{z_k\}$ are bounded;
3. the operator $\nabla F(w)$ is nonsingular.

Theorem (Global Convergence, Lai. 2024)

Let $\{\sigma_k\} \subset (0,1)$ bounded away from zero and one. If Assumptions 1~3 hold, then $\{F(w_k)\}$ converges to zero; and for any limit point $w^* = (x^*, y^*, z^*, s^*)$ of $\{w_k\}$, x^* is a Riemannian KKT point of problem (CRO).

[3] The Euclidean counterpart comes from El-Bakry, A., Tapia, R. A., Tsuchiya, T., and Zhang, Y. (1996). On the formulation and theory of the Newton interior-point method for nonlinear programming. *J Optim Theory Appl*, 1996.

Riemannian Interior Point Methods (RIPM)

Dominant Cost — Solving Newton Equation

Dominant cost of our RIPM is to solve (at each iteration)

$$\nabla F(w)\Delta w = -F(w) + \mu\hat{e}, \tag{19}$$

where

$$F(w) = \begin{pmatrix} F_x \triangleq \operatorname{grad}_x \mathcal{L}(x,y,z) \\ F_y \triangleq h(x) \\ F_z \triangleq g(x) + s \\ F_s \triangleq ZSe \end{pmatrix}, \quad \hat{e} = \begin{pmatrix} 0_x \\ 0 \\ 0 \\ e \end{pmatrix}. \tag{20}$$

Thus, we need to solve the following linear system on $T_xM \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}^m$:

$$\begin{pmatrix} \operatorname{Hess}_x \mathcal{L}(w)\Delta x + H_x\Delta y + G_x\Delta z \\ H_x^*\Delta x \\ G_x^*\Delta x + \Delta s \\ Z\Delta s + S\Delta z \end{pmatrix} = \begin{pmatrix} -F_x \\ -F_y \\ -F_z \\ -F_s + \mu e \end{pmatrix}. \tag{21}$$

Condensed form of Newton Equation

It suffices[a] to solve condensed form on $T_xM \times \mathbb{R}^l$:

$$\mathcal{T}(\Delta x, \Delta y) := \begin{pmatrix} \mathcal{A}_w \Delta x + H_x \Delta y \\ H_x^* \Delta y \end{pmatrix} = \begin{pmatrix} c \\ q \end{pmatrix}, \tag{22}$$

where

$$\begin{aligned} \mathcal{A}_w &:= \operatorname{Hess}_x \mathcal{L}(w) + G_x S^{-1} Z G_x^*, \\ c &:= -F_x - G_x S^{-1}(Z F_z + \mu e - F_s), \quad q := -F_y. \end{aligned} \tag{23}$$

[a] We recover $\Delta s = Z^{-1}(\mu e - F_s - S\Delta z)$, $\Delta z = S^{-1}[Z(G_x^* \Delta x + F_z) + \mu e - F_s]$.

$\mathcal{T}$ is self-adjoint (but may indefinite) operator on $T_x\mathcal{M} \times \mathbb{R}^l$. This is a saddle point problems on Hilbert space.

The difficulty lies in...

- the Riemannian setting leaves us with no explicit matrix form available.
- a natural way is to find the representing matrix $\hat{\mathcal{T}}$ under some basis of tangent space. (Expensive!)

Krylov Subspace Methods on Tangent Space

An ideal approach is to use iterative methods, such as **Krylov subspace methods** (e.g., Conjugate Gradients method), on $T_xM \times \mathbb{R}^l$ directly.

For simplicity, we consider the case of only inequality constraints, where Δy vanishes, thus we only needs to

$$\text{solve } \mathcal{A}_w \Delta x = c \text{ for } \Delta x \in T_xM. \tag{24}$$

- It only needs to call an abstract linear operator $v \mapsto \mathcal{A}_w v$. (matrix-vector product)
- All the iterates v_k are in T_xM.
- Since operator $\mathcal{A}_w$ is self-adjoint but indefinite, we use Conjugate Residual (CR) method to solve it.

The discussion of above can be naturally extended to the general case.

Conjugate Gradients (CG) on a Tangent Space

Input: positive definite map H on $\mathrm{T}_x\mathcal{M}$ and $b \in \mathrm{T}_x\mathcal{M}$, $b \neq 0$

Set $v_0 = 0, r_0 = b, p_0 = r_0$

For $n = 1, 2, \ldots$

Compute Hp_{n-1} (this is the only call to H)

$\alpha_n = \frac{\|r_{n-1}\|_x^2}{\langle p_{n-1}, Hp_{n-1}\rangle_x}$

$v_n = v_{n-1} + \alpha_n p_{n-1}$

$r_n = r_{n-1} - \alpha_n H p_{n-1}$

If $r_n = 0$, **output** $s = v_n$: the solution of $Hs = b$

$\beta_n = \frac{\|r_n\|_x^2}{\|r_{n-1}\|_x^2}$

$p_n = r_n + \beta_n p_{n-1}$

1. Exactly the same in form of usual CG.
2. Every vectors v_n, r_n, p_n belong to tangent space $V \equiv T_x M$.
3. Converges very fast if H is PD with small condition number.

Riemannian IPM (RIPM) vs. Euclidean IPM (EIPM)

Constrained Riemannian Optimization (CRO):

$$\begin{aligned} \min_{x \in \mathcal{M}} \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0, \text{ and } g(x) \leq 0, \end{aligned} \tag{CRO}$$

where $h : \mathcal{M} \to \mathbb{R}^l$, and $g : \mathcal{M} \to \mathbb{R}^m$.

Riemannian IPM (RIPM) vs. Euclidean IPM (EIPM)

1. RIPM inherits the advantages of Riemannian optimization and can exploit the geometric structure of the constraints.
2. EIPM is a special case of RIPM when $\mathcal{M} = \mathbb{R}^n$ or $\mathbb{R}^{n \times k}$.
3. RIPM solves Newton equation (25) of smaller order on $\mathrm{T}_x\mathcal{M} \times \mathbb{R}^l$:

$$\mathcal{T}(\Delta x, \Delta y) := \begin{pmatrix} \mathcal{A}_w \Delta x + H_x \Delta y \\ H_x^* \Delta x \end{pmatrix} = \begin{pmatrix} c \\ q \end{pmatrix}. \tag{25}$$

4. RIPM can solve some problems that EIPM cannot. For example, $\operatorname{rank}(X) = r$ is not continuous, we can not apply EIPM.

MATLAB Code

Our Solver `RIPM.m` available at `https://github.com/GALVINLAI/RIPM`

MATLAB code (.\examples\)	M	f	g	h
Euc_linear_nonnega_sphereEq	Euclidean	linear	nonnegative	sphere x'*x=1
Euc_linear_or_projection_nonnega_orthogonalEq	Euclidean	linear/projection	nonnegative	orthogonality X'*X-I=0
Euc_projection_nonnega	Euclidean	projection	nonnegative	-
Euc_projection_nonnega_symmetricEq	Euclidean	projection	nonnegative	symmetry X-X'=0
Fixedrank_MatrixCompletion_nonnega	fixedrank	matrix completion	nonnegative	-
Fixedrank_MatrixCompletion_nonnega_reliableEq	fixedrank	matrix completion	nonnegative	reliable sampled data
Fixedrank_projection_nonnega	fixedrank	projection	nonnegative	-
Ob_ONMF_StiefelEq	oblique	-trace(X'*AAt*X)	nonnegative	norm(X*V,'fro')^2-1
Ob_linear_or_projection_nonnega_StiefelEq	oblique	linear/projection	nonnegative	norm(X*V,'fro')^2-1
Sp_linear_nonnega	sphere	linear	nonnegative	-
Sp_linear_nonnega_linearEq	sphere	linear	nonnegative	linear
Sp_quadratic_nonnega	sphere	quadratic	nonnegative	-
Stiefel_linear_or_projection_nonnega	stiefel	linear/projection	nonnegative	-

Future Works I

(1) How to use preconditioner method for solving Newton equation? Due to complementary condition $S^*Z^* = 0$, as $k \to \infty$, the values of $S_k^{-1}Z_k = \begin{pmatrix} \frac{(z_k)_1}{(s_k)_1} \to 0 & & 0 \\ & \ddots & \\ & & \frac{(z_k)_n}{(s_k)_n} \to \infty \end{pmatrix}$ display a huge difference in magnitude.

Condensed form on $T_x\mathcal{M} \times \mathbb{R}^l$:

$$\mathcal{T}(\Delta x, \Delta y) := \begin{pmatrix} \mathcal{A}_w \Delta x + H_x \Delta y \\ H_x^* \Delta x \end{pmatrix} = \begin{pmatrix} c \\ q \end{pmatrix},$$

where

$$\mathcal{A}_w := Hess_x \mathcal{L}(w) + \Theta,$$

Hence, the operator $\Theta := G_x S^{-1} Z G_x^*$ in the condensed system (Above) makes it ill-conditioned, so the iterative method will likely fail unless it is carefully preconditioned.

Example of This Problem

To minimize $f(X) = \langle X, -E\rangle$ with $X \geq 0$ on $\mathrm{St}(n,p) = \left\{X \in \mathbb{R}^{n\times p} : X^\top X = I\right\}$. Recall that $U \mapsto \mathrm{Proj}_X(U) = U - X\,\mathrm{sym}\left(X^T U\right)$, where $\mathrm{sym}(Z) = \frac{Z+Z^T}{2}$ for any Z.

Using RIPM, at k-th iteration, given current tuple $W_k = (X_k, Z_k, S_k)$, we must solve the equation:

$$\mathcal{A}_k(\Delta X) = C_k \text{ for some constant } C_k \in \mathrm{T}_{X_k}\mathrm{St}(n,k), \tag{26}$$

with

$$\Delta X \mapsto \mathcal{A}_k(\Delta X) = \mathrm{Hess}_x \mathcal{L}(W_k)[\Delta X] + \Theta_k(\Delta X) = \mathrm{Proj}_{X_k}\left[\Delta X \cdot D_1\right] + \mathrm{Proj}_{X_k}\left[\Delta X \odot D_2\right],$$

where $D_1 := \mathrm{sym}[X_k^\top (Z_k + E)]$, $D_2 = Z_k \odot S_k^{\circ(-1)}$ are constants.

1. $\mathcal{A}_k$ is self-adjoint but indefinite operator on $\mathrm{T}_{X_k}\mathrm{St}(n,k)$.
2. As $k \to \infty$, $\mathcal{A}_k$ becomes very ill-conditioned by Θ_k since complementary condition $Z^* \odot S^* = O$.

Future Works II

(2) Quasi-Newton RIPM. We can approximate the Hessian of Lagrangian in $\nabla F(w_k)$ with gradient information while ensuring its local convergence.

Algorithm of RIPM for (CRO)

Solve the perturbed Newton equation

$$\nabla F(w_k)\Delta w_k = -F(w_k) + \mu_k \hat{e},$$

Compute the step α_k such that $(z_{k+1}, s_{k+1}) > 0$;

Update $w_{k+1} = \bar{\mathrm{R}}_{w_k}(\alpha_k \Delta w_k)$;

Choose $0 < \mu_{k+1} < \mu_k$;

$$\nabla F(w)\Delta w = \begin{pmatrix} \mathrm{Hess}_x\, \mathcal{L}(w)\, \Delta x + H_x \Delta y + G_x \Delta z \\ H_x^* \Delta x \\ G_x^* \Delta x + \Delta s \\ Z\Delta s + S\Delta z \end{pmatrix}$$

Quasi-Newton method,
BFGS formulate, etc.

$$B(w)\Delta w = \begin{pmatrix} G(w)\Delta x + H_x \Delta y + G_x \Delta z \\ H_x^* \Delta x \\ G_x^* \Delta x + \Delta s \\ Z\Delta s + S\Delta z \end{pmatrix}$$

$$B(w_k)\Delta w_k = -F(w_k) + \mu_k \hat{e},$$

(3) Inexact Newton RIPM.

(4) Treatment of more state-of-the-art interior point methods. Our current global algorithm uses the simplest strategy. How about, e.g., the trust region?

Riemannian Interior Point Methods for Constrained Optimization on Manifolds

Thank you for listening!

Any questions?

Reference

[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.

[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.

[FT22] Robert Simon Fong and Peter Tino.
Population-Based Optimization on Riemannian Manifolds.
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.

[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.

Reference

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

[LB20] Changshuo Liu and Nicolas Boumal.
Simple algorithms for optimization on Riemannian manifolds with constraints.
Applied Mathematics & Optimization, 82:949–981, 2020.

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

[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.

[OOT20] Mitsuaki Obara, Takayuki Okuno, and Akiko Takeda.
Sequential quadratic optimization for nonlinear optimization problems on Riemannian manifolds.
SIAM Journal on Optimization, 32(2):822–853, May 2020.
Publisher: SIAM.

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

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

[SO21] Anton Schiela and Julian Ortiz.
An SQP method for equality constrained optimization on Hilbert manifolds.
SIAM Journal on Optimization, 31(3):2255–2284, 2021.

[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.

Reference

[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.

[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.

[YS22] Yuya Yamakawa and Hiroyuki Sato.
Sequential optimality conditions for nonlinear optimization on Riemannian manifolds and a globally convergent augmented Lagrangian method.
Computational Optimization and Applications, 81:1–25, March 2022.

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

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

Summary: Stiefel Manifold

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 $X+V$:

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

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

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

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

  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. Existence. There exists w^* satisfying the KKT conditions.
2. Smoothness. The functions f, g, h are smooth on M.
3. Regularity. The set $\{\operatorname{grad} h_i(x^*) : i = 1, \cdots, l\} \cup \{\operatorname{grad} g_i(x^*) : i \in \mathcal{A}(x)\}$ is linearly independent in $T_{x^*}M$.
4. Strict Complementarity. $(z^*)_i > 0$ if $g_i(x^*) = 0$ for all $i = 1, \cdots, m$.
5. Second-Order Sufficiency. $\langle \operatorname{Hess}_x \mathcal{L}(w^*)\xi, \xi \rangle > 0$ for all nonzero $\xi \in T_{x^*}M$ satisfying $\langle \xi, \operatorname{grad} h_i(x^*) \rangle = 0$ for $i = 1, \cdots, l$, and $\langle \xi, \operatorname{grad} g_i(x^*) \rangle = 0$ for $i \in \mathcal{A}(x^*)$.

Proposition (Lai. 2022)

If assumptions (1)-(5) hold, then standard Newton assumptions (N1)-(N3) hold for KKT vector field F.

流形优化入门自学建议

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。

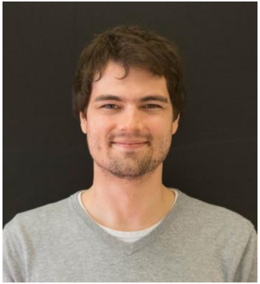


Figure: Nicolas Boumal, EPFL

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。

流形优化入门自学建议

3 Reference

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