Optimization on manifolds: applications and algorithms

Bamdev Mishra

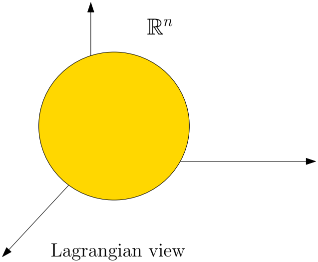
Microsoft

Our interest is the optimization problem of the form

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ \text{subject to} & x^\top x = 1. \end{array}$$

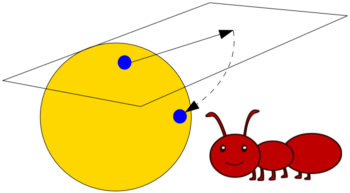
Assume that f is differentiable.

There exist two complementary views of optimization with constraint $x^\top x = 1$



Lagrangian view

(Constrained viewpoint)



Manifold optimization view

(An ant-on-manifold view: unconstrained viewpoint)

Optimization differ in what is the search space

Embedding constraint in the linear Euclidean space

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & x^\top x = 1 \end{aligned}$$

is equivalent to solving the Lagrangian
$\max_{\lambda \in \mathbb{R}} \min_{x \in \mathbb{R}^n} L(x, \lambda) = f(x) - \lambda(x^\top x - 1)$.

OR

Manifold optimization is on the nonlinear search space

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

where $\mathcal{M} =: \{x \in \mathbb{R}^n : x^\top x = 1\}$ is a differentiable manifold.

Manifold optimization generalizes unconstrained optimization to manifolds

Solving

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

where $\mathcal{M} =: \{x \in \mathbb{R}^n : x^\top x = 1\}$

is equivalent to

Unconstrained optimization over the manifold $\mathcal{M}$.

Outline

- Applications
- Manifold algorithms
- Manopt: a toolbox of optimization on manifolds

Applications

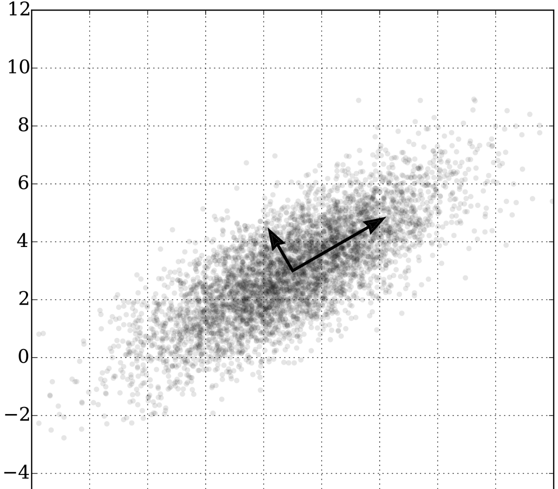
The Euclidean space $\mathbb{R}^n$ is a manifold.

Manifold $\mathcal{M}$ is a generalization of $\mathbb{R}^n$.

Not interesting, but trivially, all applications in $\mathbb{R}^n$ are on manifolds.

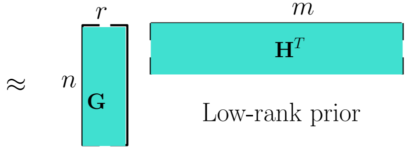
Principal components analysis (PCA) is on manifold of orthogonal matrices

Figure from Wikipedia.org.



Recommender system $\equiv$ low-rank matrix completion

$$
\begin{array}{cc}
 & m \text{ Users} \\
n \text{ Movies} & \begin{bmatrix} ? & ? & * & ? \\ * & * & ? & * \\ ? & * & * & ? \\ * & ? & * & ? \end{bmatrix}
\end{array}
$$



$$(n + m - r)r,\ r \ll (m, n)$$

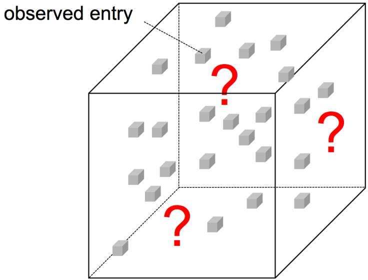
Set of fixed-rank matrices is a manifold.

$\mathbf{X} = \mathbf{G}\mathbf{H}^\top$.

$\mathcal{M} =: \{\mathbf{X} \in \mathbb{R}^{n \times m} : \mathrm{rank}(\mathbf{X}) = r\}$.

Tensors are multiarray matrices which generalize matrices

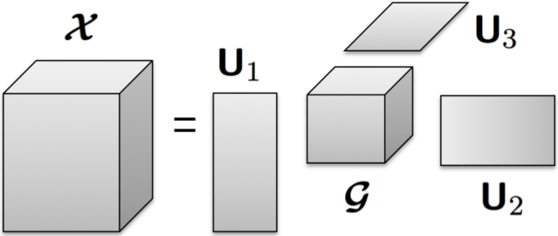
movie$\times$user$\times$time



Low-rank tensor decomposition / completion problem appears in forecasting, prediction, and multitask problems.

Fixed-(multilinear)rank tensors form a manifold

Tensor decomposition

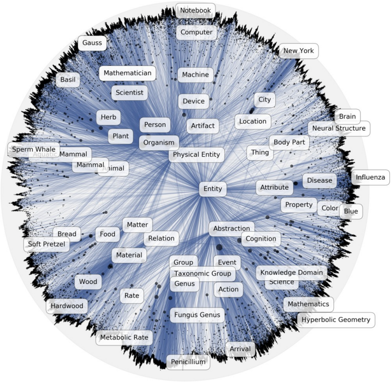


Set of fixed-rank tensors is a manifold.

$\mathcal{X} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$

$\mathcal{M} =: \{\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3} : \mathrm{rank}(\mathcal{X}) = (r_1, r_2, r_3)\}.$

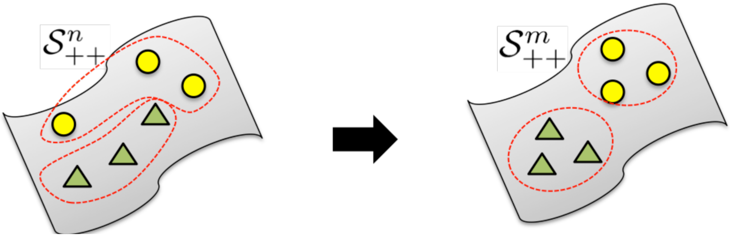
Learning continuous representations of hierarchies



Hyperbolic manifold is used to model hypernymy relationships

Figure from mnick.github.io.

Metric learning



Symmetric positive definite matrices form a manifold

Many structured constraints are manifolds

Screenshots from https://Manopt.org.

Name	Set
Euclidean space (complex)	$\mathbb{R}^{m\times n}$, $\mathbb{C}^{m\times n}$
Symmetric matrices	$\{X \in \mathbb{R}^{n\times n} : X = X^T\}^k$
Skew-symmetric matrices	$\{X \in \mathbb{R}^{n\times n} : X + X^T = 0\}^k$
Centered matrices	$\{X \in \mathbb{R}^{m\times n} : X\mathbf{1}_n = 0_m\}$
Sphere	$\{X \in \mathbb{R}^{n\times m} : \|X\|_\mathrm{F} = 1\}$
Symmetric sphere	$\{X \in \mathbb{R}^{n\times n} : \|X\|_\mathrm{F} = 1, X = X^T\}$
Complex sphere	$\{X \in \mathbb{C}^{n\times m} : \|X\|_\mathrm{F} = 1\}$
Oblique manifold	$\{X \in \mathbb{R}^{n\times m} : \|X_{:1}\| = \cdots = \|X_{:m}\| = 1\}$

Many structured constraints are manifolds

Generalized Stiefel manifold	$\{X \in \mathbb{R}^{n\times p} : X^T B X = I_p\}$ for some $B \succ 0$
Stiefel manifold, stacked	$\{X \in \mathbb{R}^{md\times k} : (XX^T)_{ii} = I_d\}$
Grassmann manifold	$\{\mathrm{span}(X) : X \in \mathbb{R}^{n\times p}, X^T X = I_p\}^k$
Complex Grassmann manifold	$\{\mathrm{span}(X) : X \in \mathbb{C}^{n\times p}, X^T X = I_p\}^k$
Generalized Grassmann manifold	$\{\mathrm{span}(X) : X \in \mathbb{R}^{n\times p}, X^T B X = I_p\}$ for some $B \succ 0$
Rotation group	$\{R \in \mathbb{R}^{n\times n} : R^T R = I_n, \det(R) = 1\}^k$
Special Euclidean group	$\{(R, t) \in \mathbb{R}^{n\times n} \times \mathbb{R}^n : R^T R = I_n, \det(R) = 1\}^k$
Essential	Epipolar constraint between projected points in two

Many structured constraints are manifolds

Fixed-rank	$\{X \in \mathbb{R}^{m\times n} : \operatorname{rank}(X) = k\}$
Fixed-rank tensor	Tensors of fixed multilinear rank in Tucker format
Matrices with strictly positive entries	$\{X \in \mathbb{R}^{m\times n} : X_{ij} > 0\ \forall i, j\}$
Symmetric, positive definite matrices	$\{X \in \mathbb{R}^{n\times n} : X = X^T, X \succ 0\}^k$
Symmetric positive semidefinite, fixed-rank	$\{X \in \mathbb{R}^{n\times n} : X = X^T \succeq 0, \operatorname{rank}(X) = k\}$

Many structured constraints are manifolds

Multinomial manifold (strict simplex elements)	$\{X \in \mathbb{R}^{n \times m} : X_{ij} > 0 \forall i, j \text{ and } X^T \mathbf{1}_m = \mathbf{1}_n\}$
Multinomial doubly stochastic manifold	$\{X \in \mathbb{R}^{n \times n} : X_{ij} > 0 \forall i, j \text{ and } X\mathbf{1}_n = \mathbf{1}_n, X^T \mathbf{1}_n = \mathbf{1}_n\}$
Multinomial symmetric and stochastic manifold	$\{X \in \mathbb{R}^{n \times n} : X_{ij} > 0 \forall i, j \text{ and } X\mathbf{1}_n = \mathbf{1}_n, X = X^T\}$

Applications

Manifold $\mathcal{M}$	Applications
Grassmann manifold	Dimensionality reduction
Unit norm vector	Independent comp. analysis
Orthonormal matrix	Sparse and robust PCA
Rotation matrix SO(3)	Synchronization of rotations
Positive definite matrix	Diffusion tensor imaging
Fixed-rank matrix / tensor	Recommender systems
Positive definite matrix	Domain adaptation
SE(3)	Robotic movements
Hyperbolic space	NLP
...	

Manifold algorithms

Optimization on manifold framework has gained much attention lately

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

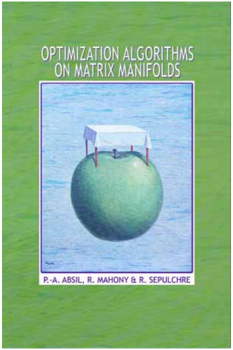
The PCA problem:

$$\min_{x \in \mathcal{M}} \quad -x^\top \mathbf{A} x$$

where $\mathcal{M} := \{x \in \mathbb{R}^n : x^\top x = 1\}$

Geometric methods: Optimization on manifolds

A monograph on optimization on matrix manifolds

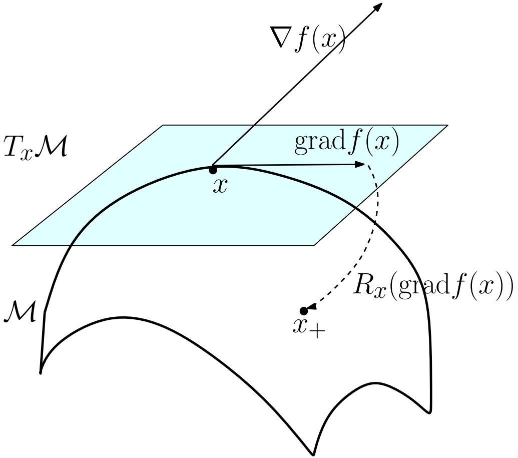


Optimization Algorithms on Matrix Manifolds
P.-A. Absil, R. Mahony, R. Sepulchre
Princeton University Press, January 2008

Manifold is a differentiable structure

- Manifold $\mathcal{M}$ is a differentiable structure that locally looks like Euclidean.
- We work with with a metric or an inner product g at every point.
- $T_x\mathcal{M}$ is the linearization of $\mathcal{M}$ at x and is called the tangent space.
- $\mathcal{M}$ and g together is called a Riemannian manifold.

Optimization on manifolds: algorithms



- Task: Compute a first-order critical point of f on $\mathcal{M}$.

Riemannian steepest descent on $\mathcal{M}$

t is stepsize.

- Euclidean:
  $x_+ = x - t\nabla_x f$.

- Manifold:
  $x_+ = R_x(-t\mathrm{grad}_x f)$.

  $\mathrm{grad}_x f$ is the Riemannian gradient and
  R_x is the retraction operator to ensure $x_+ \in \mathcal{M}$.

Guarantees:
Backtracking linesearch conditions exist.
Global convergence to first-order critical point exist.
Local rate analysis exists.

Riemannian manifold helps to derive concrete formulas for steepest descent iterations

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

where $\mathcal{M} =: \{x \in \mathbb{R}^n : x^\top x = 1\}$.

The Riemannian gradient
$\mathrm{grad}_x f = \nabla_x f - (x^\top \nabla_x f)x$,
where $\nabla_x f$ is the partial derivative of f at x.

R_x is "projection" onto manifold. $R_x(\xi) = (x + \xi)/\|x + \xi\|$.

Riemannian steepest descent method is the iteration
$x_+ = (x - t\mathrm{grad}_x f)/\|x - t\mathrm{grad}_x f\|$.

Riemannian Newton algorithm on $\mathcal{M}$

- Euclidean:
  $x_+ = x + \xi$, where $\mathrm{D}^2 f[\xi] = -\nabla_x f$.

- Manifold:
  $x_+ = R_x(\xi)$, where $\mathrm{Hess}_x f[\xi] = -\mathrm{grad}_x f$.

  $\mathrm{grad}_x f$ is the Riemannian gradient, $\mathrm{Hess}_x f$ is the Riemannian Hessian, and
  R_x is the retraction operator.

Guarantees:
Local quadratic rate analysis exists.
Global convergence to first-order critical point exist.

Riemannian manifold helps to derive concrete formulas for Newton iterations

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

where $\mathcal{M} =: \{x \in \mathbb{R}^n : x^\top x = 1\}$.

Expressions for $\mathrm{grad}_x f$ and R_x are known.

$\mathrm{Hess}_x f[\xi] = \mathrm{D}^2 f[\xi] - (x^\top \nabla_x f)\xi - (x^\top \mathrm{D}^2 f[\xi])x$.

Solve the linear system for ξ

$$\underbrace{\mathrm{D}^2 f[\xi] - (x^\top \nabla_x f)\xi - (x^\top \mathrm{D}^2 f[\xi])x}_{\mathrm{Hess}_x f[\xi]} = -\underbrace{(\nabla_x f - (x^\top \nabla_x f)x)}_{\mathrm{grad}_x f}.$$

Riemannian Newton method is the iteration

$x_+ = (x + \xi)/\|x + \xi\|$.

Computing the maximum eigenvalue value is optimization on a hypersphere

$$\min_{x^\top x = 1} \quad f(x) = -x^\top \mathbf{A} x,$$

$\mathbf{A}$ is a given symmetric matrix.

$\nabla_x f = -2\mathbf{A}x$
$\mathrm{D}^2 f[\xi] = -2\mathbf{A}\xi.$

$\mathrm{grad}_x f = -2\mathbf{A}x + 2(x^\top \mathbf{A} x)x$
$\mathrm{Hess}_x f[\xi] = -2\mathbf{A}\xi + 2(x^\top \mathbf{A} x)\xi + 2(x^\top \mathbf{A}\xi)x.$

$R_x(\xi) = (x + \xi)/\|x + \xi\|.$

Most Euclidean optimization algorithms generalize well to manifolds

- Conjugate gradients.
- BFGS and Quasi-Newton methods.
- Non-smooth optimization on manifolds.
- Stochastic gradients w/o variance reduction.
- Preconditioning on manifolds.

The Riemannian theory is not only theoretically elegant but allows to write concrete numerical algorithms on both manifolds.

Manifold optimization toolbox Manopt

Manifold optimization tools are independent of the cost function

1. All the discussion revolves around manifolds and is fairly independent of the cost function f.

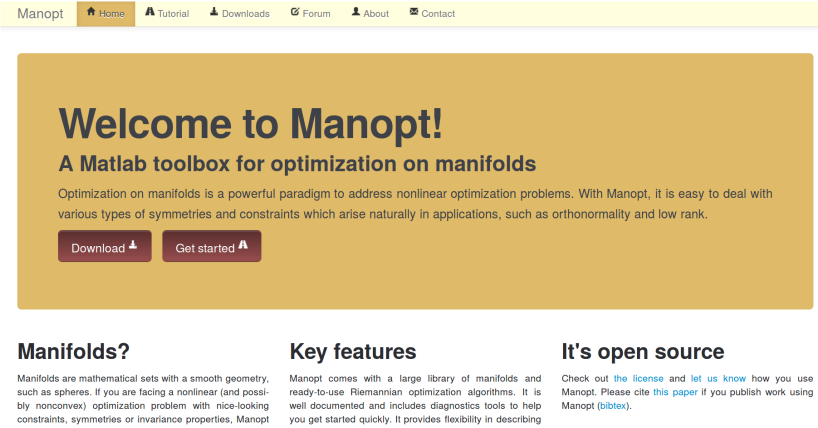
2. The solvers need only manifold notions like metric, tangent space characterization, retraction operations, transport of vectors.

Leveraging these two points, we design a modular manifold optimization toolbox Manopt.

https://Manopt.org

A Matlab toolbox to make optimization on manifolds feel as simple as unconstrained optimizaiton.

Nicolas Boumal and **BM**
P.-A. Absil, Y. Nesterov and R. Sepulchre



Manopt | Home | Tutorial | Downloads | Forum | About | Contact

Welcome to Manopt!

A Matlab toolbox for optimization on manifolds

Optimization on manifolds is a powerful paradigm to address nonlinear optimization problems. With Manopt, it is easy to deal with various types of symmetries and constraints which arise naturally in applications, such as orthonormality and low rank.

Download | Get started

Manifolds?

Manifolds are mathematical sets with a smooth geometry, such as spheres. If you are facing a nonlinear (and possibly nonconvex) optimization problem with nice-looking constraints, symmetries or invariance properties, Manopt

Key features

Manopt comes with a large library of manifolds and ready-to-use Riemannian optimization algorithms. It is well documented and includes diagnostics tools to help you get started quickly. It provides flexibility in describing

It's open source

Check out the license and let us know how you use Manopt. Please cite this paper if you publish work using Manopt (bibtex).

Manopt has grown in popularity

Description

Optimization on manifolds is a rapidly developing branch of nonlinear optimization. Its focus is on problems where the smooth geometry of the search space can be leveraged to design efficient numerical algorithms. In particular, optimization on manifolds is wellsuited to deal with rank and orthogonality constraints. Such structured constraints appear pervasively in machine learning applications, including low-rank matrix completion, sensor network localization, camera network registration, independent component analysis, metric learning, dimensionality reduction and so on.

The Manopt toolbox, available at www. manopt. org, is a user-friendly, documented piece of software dedicated to simplify experimenting with state of the art Riemannian optimization algorithms. By dealing internally with most of the differential geometry, the package aims particularly at lowering the entrance barrier.

Total citations

Cited by 342

Scholar articles

Manopt, a Matlab toolbox for optimization on manifolds
N Boumal, B Mishra, PA Absil, R Sepulchre - The Journal of Machine Learning Research, 2014
Cited by 338 Related articles All 17 versions

Manopt has a modular structure

Basic codes structure of Manopt:

- Core tools
- Manifold definitions
- Manifold solvers.

Computing the maximum eigenvalue value is optimization on a hypersphere

$$\min_{x^\top x=1} \qquad f(x) = -x^\top \mathbf{A} x,$$

$\mathbf{A}$ is a given symmetric matrix.

Manopt requires:
$f(x) = -x^\top \mathbf{A} x$
$\nabla_x f = -2\mathbf{A}x$
$\mathrm{D}^2 f[\xi] = -2\mathbf{A}\xi$
Mention that we enforce $x^\top x = 1$.

The Riemannian notions are handled internally by Manopt.

Computing the maximum eigenvalue value with Manopt is optimization on a hypersphere

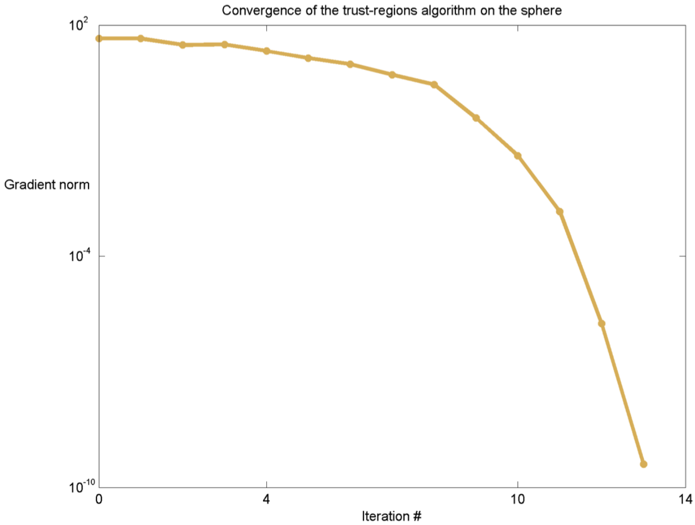
```
% Generate the problem data.
n = 1000;
A = randn(n);
A = .5*(A+A');

% Create the problem structure and specify the manifold.
problem.M = spherefactory(n);

% Define the problem cost function and its gradient.
problem.cost  = @(x) -x'*(A*x);
problem.egrad = @(x) -2*A*x;

% Numerically check gradient consistency.
checkgradient(problem);

% Solve.
[x, xcost, info] = trustregions(problem);
```



Convergence of the trust-regions algorithm on the sphere
10^2
10^{-4}
10^{-10}
Gradient norm
0
4
10
14
Iteration #

Manopt comes with a comprehensive list of solvers and manifolds

- More than 35 manifold descriptions for different structured constraints.
- 9 solvers both determinstic and stochastic solvers.
- We have a forum for discussions on Manopt and manifold optimization.

Manopt aims not only to be a platform for researchers to experiment with manifold optimization but also useful to have large-scale problems.

Manopt can be used for general unconstrained optimization as the Euclidean space is trivially a manifold.

There exist other independent toolboxes for optimization on manifolds

- Pymanopt: a Python toolbox for manifold optimization.
- McTorch: a PyTorch extension to do manifold optimization for deep learning applications.
- ROPTLIB: a C++ Library for optimization on manifolds with Python, R, and Julia wrappers.
- Geomstats: a Python package for computations and statistics on manifolds.

Optimization on manifolds: applications and algorithms

Bamdev Mishra

Microsoft