Optimization on manifolds: applications and algorithms

Bamdev Mishra
Our interest is the optimization problem of the form

\[
\min_{x \in \mathbb{R}^n} f(x)
\]
subject to \( x^\top x = 1 \).

Assume that \( f \) is differentiable.
There exist two complementary views of optimization with constraint $x^\top x = 1$.
Optimization differ in what is the search space

Embedding constraint in the linear Euclidean space

$$\min_{x \in \mathbb{R}^n} f(x)$$
subject to \( x^\top x = 1 \)

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

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

where \( \mathcal{M} =: \{ x \in \mathbb{R}^n : x^\top x = 1 \} \) is a differentiable manifold.
Manifold optimization generalizes unconstrained optimization to manifolds

Solving

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

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 $M$ 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

Set of fixed-rank matrices is a manifold.
$\mathbf{X} = \mathbf{G}\mathbf{H}^T$.
$\mathcal{M} =: \{\mathbf{X} \in \mathbb{R}^{n \times m} : \text{rank}(\mathbf{X}) = r\}$. 
Tensors are multiarray matrices which generalize matrices

movie × user × 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.

\[ \mathcal{X} = \mathcal{G} \times_1 U_1 \times_2 U_2 \times_3 U_3 \]

\[ \mathcal{M} = \{ \mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3} : \text{rank}(\mathcal{X}) = (r_1, r_2, r_3) \} \].
Many structured constraints are manifolds

Screenshots from https://Manopt.org.

<table>
<thead>
<tr>
<th>Name</th>
<th>Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean space (complex)</td>
<td>$\mathbb{R}^{m \times n}, \mathbb{C}^{m \times n}$</td>
</tr>
<tr>
<td>Symmetric matrices</td>
<td>${X \in \mathbb{R}^{n \times n} : X = X^T}^k$</td>
</tr>
<tr>
<td>Skew-symmetric matrices</td>
<td>${X \in \mathbb{R}^{n \times n} : X + X^T = 0}^k$</td>
</tr>
<tr>
<td>Centered matrices</td>
<td>${X \in \mathbb{R}^{m \times n} : X1_n = 0_m}$</td>
</tr>
<tr>
<td>Sphere</td>
<td>${X \in \mathbb{R}^{n \times m} : |X|_F = 1}$</td>
</tr>
<tr>
<td>Symmetric sphere</td>
<td>${X \in \mathbb{R}^{n \times n} : |X|_F = 1, X = X^T}$</td>
</tr>
<tr>
<td>Complex sphere</td>
<td>${X \in \mathbb{C}^{n \times m} : |X|_F = 1}$</td>
</tr>
<tr>
<td>Oblique manifold</td>
<td>${X \in \mathbb{R}^{n \times m} : |X1_1| = \cdots = |X1_m| = 1}$</td>
</tr>
</tbody>
</table>
Many structured constraints are manifolds

<table>
<thead>
<tr>
<th>Generalized Stiefel manifold</th>
<th>( { X \in \mathbb{R}^{n \times p} : X^T B X = I_p } ) for some ( B &gt; 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stiefel manifold, stacked</td>
<td>( { X \in \mathbb{R}^{md \times k} : (X X^T)_{ii} = I_d } )</td>
</tr>
<tr>
<td>Grassmann manifold</td>
<td>( { \text{span}(X) : X \in \mathbb{R}^{n \times p}, X^T X = I_p } ) (^k)</td>
</tr>
<tr>
<td>Complex Grassmann manifold</td>
<td>( { \text{span}(X) : X \in \mathbb{C}^{n \times p}, X^T X = I_p } ) (^k)</td>
</tr>
<tr>
<td>Generalized Grassmann manifold</td>
<td>( { \text{span}(X) : X \in \mathbb{R}^{n \times p}, X^T B X = I_p } ) for some ( B &gt; 0 )</td>
</tr>
<tr>
<td>Rotation group</td>
<td>( { R \in \mathbb{R}^{n \times n} : R^T R = I_n, \det(R) = 1 } ) (^k)</td>
</tr>
<tr>
<td>Special Euclidean group</td>
<td>( {(R, t) \in \mathbb{R}^{n \times n} \times \mathbb{R}^n : R^T R = I_n, \det(R) = 1 } ) (^k)</td>
</tr>
<tr>
<td>Essential manifold</td>
<td>Epipolar constraint between projected points in two perspectives in two cameras, ( \mathbb{R}^{n \times n} \times \mathbb{R}^2 )</td>
</tr>
</tbody>
</table>
Many structured constraints are manifolds

<table>
<thead>
<tr>
<th>Fixed-rank tensor</th>
<th>Tensors of fixed multilinear rank in Tucker format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrices with strictly positive entries</td>
<td>( { X \in \mathbb{R}^{m \times n} : X_{ij} &gt; 0 \ \forall i, j } )</td>
</tr>
<tr>
<td>Symmetric, positive definite matrices</td>
<td>( { X \in \mathbb{R}^{n \times n} : X = X^T, X &gt; 0 }^k )</td>
</tr>
<tr>
<td>Symmetric positive semidefinite, fixed-rank</td>
<td>( { X \in \mathbb{R}^{n \times n} : X = X^T \succeq 0, \text{rank}(X) = k } )</td>
</tr>
</tbody>
</table>
Many structured constraints are manifolds

<table>
<thead>
<tr>
<th>Multinomial manifold (strict simplex elements)</th>
<th>( { X \in \mathbb{R}^{n \times m} : X_{ij} &gt; 0 \forall i, j \text{ and } X^T 1_m = 1_n } )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multinomial doubly stochastic manifold</td>
<td>( { X \in \mathbb{R}^{n \times n} : X_{ij} &gt; 0 \forall i, j \text{ and } X 1_n = 1_n, X^T 1_n = 1_n } )</td>
</tr>
<tr>
<td>Multinomial symmetric and stochastic manifold</td>
<td>( { X \in \mathbb{R}^{n \times n} : X_{ij} &gt; 0 \forall i, j \text{ and } X 1_n = 1_n, X = X^T } )</td>
</tr>
</tbody>
</table>
## Applications

<table>
<thead>
<tr>
<th>Manifold $\mathcal{M}$</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grassmann manifold</td>
<td>Dimensionality reduction</td>
</tr>
<tr>
<td>Unit norm vector</td>
<td>Independent comp. analysis</td>
</tr>
<tr>
<td>Orthonormal matrix</td>
<td>Sparse and robust PCA</td>
</tr>
<tr>
<td>Rotation matrix $\text{SO}(3)$</td>
<td>Synchronization of rotations</td>
</tr>
<tr>
<td>Positive definite matrix</td>
<td>Diffusion tensor imaging</td>
</tr>
<tr>
<td>Fixed-rank matrix / tensor</td>
<td>Recommender systems</td>
</tr>
<tr>
<td>Positive definite matrix</td>
<td>Domain adaptation</td>
</tr>
<tr>
<td>$\text{SE}(3)$</td>
<td>Robotic movements</td>
</tr>
<tr>
<td>Hyperbolic space</td>
<td>NLP</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Manifold algorithms
Optimization on manifold framework has gained much attention lately

\[
\min_{x \in \mathcal{M}} \, f(x)
\]

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 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

- A function $f : \mathcal{M} \rightarrow \mathbb{R}$, smooth in the sense of the manifold structure.

- 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:
  \[ x_+ = R_x(-t \text{grad}_x f). \]

$\text{grad}_x f$ 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

\[ \min_{x \in \mathcal{M}} f(x), \]

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

The Riemannian gradient
\[
\text{grad}_x f = \nabla_x f - (x^\top \nabla_x f)x,
\]
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
\[ x_+ = (x - t\text{grad}_x f)/\|x - t\text{grad}_x f\|. \]
Riemannian Newton algorithm on $\mathcal{M}$

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

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

$\text{grad}_x f$ is the Riemannian gradient, $\text{Hess}_x f$ 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

\[
\min_{x \in \mathcal{M}} f(x),
\]

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

Expressions for \( \nabla_x f \) and \( R_x \) are known.

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

Solve the linear system for \( \xi \)

\[
D^2 f[\xi] - (x^\top \nabla_x f)\xi - (x^\top D^2 f[\xi])x = - (\nabla_x f - (x^\top \nabla_x f)x).
\]

Riemannian Newton method is the iteration

\[
x_+ = (x + \xi)/\|x + \xi\|.
\]
Computing the maximum eigenvalue value is optimization on a hypersphere

$$\min_{x^\top x = 1} f(x) = -x^\top Ax,$$

$A$ is a given symmetric matrix.

$$\nabla_x f = -2Ax$$
$$D^2f[\xi] = -2A\xi.$$

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

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

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

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 well-suited 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.

Cited by 342

Manopt, a Matlab toolbox for optimization on manifolds
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

\[
\min_{x^\top x = 1} f(x) = -x^\top Ax,
\]

\(A\) is a given symmetric matrix.

Manopt requires:
\[
\begin{align*}
  f(x) &= -x^\top Ax \\
  \nabla_x f &= -2Ax \\
  D^2 f[\xi] &= -2A\xi
\end{align*}
\]

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, xc, info] = trustregions(problem);
Convergence of the trust-regions algorithm on the sphere

Gradient norm

Iteration #
Manopt comes with a comprehensive list of solvers and manifolds

- More than 35 manifold descriptions for different structured constraints.
- 9 solvers both deterministic 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