Journal of Machine Learning Research 15 (2014) 1455-1459 Submitted 8/13; Revised 1/14; Published 4/14

Manopt, a Matlab Toolbox for Optimization on Manifolds

Nicolas Boumal NICOLASBOUMAL@GMAIL.COM
Department of Mathematical Engineering,
Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

Bamdev Mishra BAMDEVM@GMAIL.COM
Department of Electrical Engineering and Computer Science
Université de Liège, B-4000 Liège, Belgium

P.-A. Absil ABSIL@INMA.UCL.AC.BE
Department of Mathematical Engineering
Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

Rodolphe Sepulchre R.SEPULCHRE@ENG.CAM.AC.UK
Department of Engineering (Control Group)
University of Cambridge, Trumpington Street, Cambridge CB2 1PZ, UK

Editor: Antti Honkela

Abstract

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.

Keywords: Riemannian optimization, nonlinear programming, non convex, orthogonality constraints, rank constraints, optimization with symmetries, rotation matrices

1. Introduction

Optimization on manifolds, or Riemannian optimization, is a fast growing research topic in the field of nonlinear optimization. Its purpose is to provide efficient numerical algorithms to find (at least local) optimizers for problems of the form

$$\min_{x \in \mathcal{M}} f(x), \tag{1}$$

where the search space $\mathcal{M}$ is a smooth space: a differentiable manifold which can be endowed with a Riemannian structure. In a nutshell, this means $\mathcal{M}$ can be linearized locally at each point x as a tangent space $\mathrm{T}_x\mathcal{M}$ and an inner product $\langle \cdot, \cdot \rangle_x$ which smoothly depends on x is available on $\mathrm{T}_x\mathcal{M}$. For example, when $\mathcal{M}$ is a submanifold of $\mathbb{R}^{n\times m}$, a typical inner product is $\langle H_1, H_2 \rangle_X = \mathrm{trace}(H_1^\top H_2)$. Many smooth search spaces arise often in applications.

©2014 Nicolas Boumal, Bamdev Mishra, P.-A. Absil and Rodolphe Sepulchre.

For example, the **oblique manifold** $\mathcal{M} = \{X \in \mathbb{R}^{n \times m} \colon \operatorname{diag}(X^\top X) = \mathbb{1}_m\}$ is a product of spheres. That is, $X \in \mathcal{M}$ if each column of X has unit 2-norm in $\mathbb{R}^n$. Absil and Gallivan (2006) show how independent component analysis can be cast on this manifold as non-orthogonal joint diagonalization.

When furthermore it is only the product $Y = X^\top X$ which matters, matrices of the form QX are equivalent for all orthogonal Q. This suggests a quotient geometry for the **fixed-rank elliptope** $\mathcal{M} = \{Y \in \mathbb{R}^{m \times m} : Y = Y^\top \succeq 0, \operatorname{rank}(Y) = n, \operatorname{diag}(Y) = \mathbb{1}_m\}$. Grubišić and Pietersz (2007) optimize over this set to produce low-rank approximations of covariance matrices.

The (compact) **Stiefel manifold** is the Riemannian submanifold of orthonormal matrices, $\mathcal{M} = \{X \in \mathbb{R}^{n \times m} \colon X^\top X = I_m\}$. Theis et al. (2009) formulate independent component analysis with dimensionality reduction as optimization over the Stiefel manifold. Journée et al. (2010b) frame sparse principal component analysis over this manifold as well.

The **Grassmann manifold** $\mathcal{M} = \{\operatorname{col}(X) \colon X \in \mathbb{R}_*^{n \times m}\}$, where X is a full-rank matrix and $\operatorname{col}(X)$ denotes the subspace spanned by its columns, is the set of subspaces of $\mathbb{R}^n$ of dimension m. Among other things, optimization over the Grassmann manifold is useful in low-rank matrix completion, where it is observed that if one knows the column space spanned by the sought matrix, then completing the matrix according to a least-squares criterion is easy (Boumal and Absil, 2011; Keshavan et al., 2010).

The **special orthogonal group** $\mathcal{M} = \{X \in \mathbb{R}^{n \times n} \colon X^\top X = I_n \text{ and } \det(X) = 1\}$ is the group of rotations, typically considered as a Riemannian submanifold of $\mathbb{R}^{n \times n}$. Optimization problems involving rotation matrices occur in robotics and computer vision, when estimating the attitude of vehicles or the pose of cameras (Boumal et al., 2013).

The set of **fixed-rank matrices** $\mathcal{M} = \{X \in \mathbb{R}^{n \times m} \colon \operatorname{rank}(X) = k\}$ admits a number of different Riemannian structures. Vandereycken (2013) proposes an embedded geometry for $\mathcal{M}$ and exploits Riemannian optimization on that manifold to address the low-rank matrix completion problem. Shalit et al. (2012) use the same geometry to address similarity learning. Mishra et al. (2012) further cover a number of quotient geometries.

The set of **symmetric, positive semidefinite, fixed-rank matrices** is also a manifold, $\mathcal{M} = \{X \in \mathbb{R}^{n \times n} \colon X = X^\top \succeq 0, \operatorname{rank}(X) = k\}$. Meyer et al. (2011) exploit this to propose low-rank algorithms for metric learning. This space is tightly related to the space of **Euclidean distance matrices** X such that X_{ij} is the squared distance between two fixed points $x_i, x_j \in \mathbb{R}^k$. Mishra et al. (2011) leverage this geometry to formulate efficient low-rank algorithms for Euclidean distance matrix completion.

The rich geometry of Riemannian manifolds makes it possible to define gradients and Hessians of cost functions f, as well as systematic procedures (called *retractions*) to move on the manifold starting at a point x, along a specified tangent direction at x. Those are sufficient ingredients to generalize standard nonlinear optimization methods such as gradient descent, conjugate gradients, quasi-Newton, trust-regions, etc.

Building upon many earlier results not reviewed here, the recent monograph by Absil et al. (2008) sets an algorithmic framework to analyze problems of the form (1) when f is a smooth function, with a strong emphasis on building a theory that leads to efficient numerical algorithms on special manifolds. In particular, it describes the necessary ingredients to design first- and second-order algorithms on Riemannian submanifolds and quotient manifolds of linear spaces. These algorithms come with numerical costs and convergence guarantees essentially matching those of the Euclidean counterparts they generalize. For example, the Riemannian trust-region method converges globally toward critical points and converges locally quadratically when the Hessian of f is available.

The maturity of the theory of smooth Riemannian optimization, its widespread applicability and its excellent track record performance-wise prompted us to build the Manopt toolbox: a user-friendly piece of software to help researchers and practitioners experiment with these tools. Code and documentation are available at `www.manopt.org`.

2. Architecture and features of Manopt

The toolbox architecture is based on a separation of the manifolds, the solvers and the problem descriptions. For basic use, one only needs to pick a manifold from the library, describe the cost function (and possible derivatives) on this manifold and pass it on to a solver. Accompanying tools help the user in common tasks such as numerically checking whether the cost function agrees with its derivatives up to the appropriate order, approximating the Hessian based on the gradient of the cost, etc.

Manifolds in Manopt are represented as structures and are obtained by calling a factory. The manifold descriptions include projections on tangent spaces, retractions, helpers to convert Euclidean derivatives (gradient and Hessian) to Riemannian derivatives, etc. All the manifolds mentioned in the introduction work out of the box, and more can be added. Cartesian products of known manifolds are supported too.

Solvers in Manopt are functions that implement generic Riemannian minimization algorithms. Solvers log standard information at each iteration and comply with standard stopping criteria. Users may provide callbacks to log extra information or check custom stopping criteria. Currently available solvers include Riemannian trust-regions—based on work by Absil et al. (2007)—and conjugate gradients (both with preconditioning), as well as steepest descent and a couple of derivative-free schemes. More solvers can be added.

An optimization problem in Manopt is represented as a problem structure. The latter includes a field which contains a manifold, as obtained from a factory. Additionally, the problem structure hosts function handles for the cost function f and (possibly) its derivatives. An abstraction layer at the interface between the solvers and the problem description offers great flexibility in the cost function description. As the needs grow during the life-cycle of the toolbox and new ways of describing f become necessary (subdifferentials, partial gradients, etc.), it will be sufficient to update this interface.

Computing $f(x)$ typically produces intermediate results which can be reused in order to compute the derivatives of f at x. To prevent redundant computations, Manopt incorporates an (optional) caching system, which becomes useful when transitioning from a proof-of-concept draft of the algorithm to a convincing implementation.

3. Example: the maximum cut problem

Given an undirected graph with n nodes and weights $w_{ij} \geq 0$ on the edges such that $W \in \mathbb{R}^{n\times n}$ is the weighted adjacency matrix and $D \in \mathbb{R}^{n\times n}$ is the diagonal degree matrix with $D_{ii} = \sum_j w_{ij}$, the graph Laplacian is the positive semidefinite matrix $L = D - W$. The max-cut problem consists in building a partition $s \in \{+1, -1\}^n$ of the nodes in two classes such that $\frac{1}{4}s^\top Ls = \sum_{i<j} w_{ij}\frac{(s_i-s_j)^2}{4}$, that is, the sum of the weights of the edges connecting the two classes, is maximum. Let $X = ss^\top$. Then, max-cut is equivalent to:

$$\max_{X\in\mathbb{R}^{n\times n}} \operatorname{trace}(LX)/4$$
$$\text{s.t. } X = X^\top \succeq 0, \operatorname{diag}(X) = \mathbb{1}_n \text{ and } \operatorname{rank}(X) = 1.$$

Goemans and Williamson (1995) proposed and analyzed the famous relaxation of this problem which consists in dropping the rank constraint, yielding a semidefinite program. Alternatively relaxing the rank constraint to be $\operatorname{rank}(X) \le r$ for some $1 < r < n$ yields a tighter but nonconvex relaxation. Journée et al. (2010a) observe that fixing the rank with the constraint $\operatorname{rank}(X) = r$ turns the search space into a smooth manifold, the fixed-rank elliptope, which can be optimized over using Riemannian optimization. In Manopt, simple code for this reads (with $Y \in \mathbb{R}^{n\times r}$ such that $X = YY^\top$):

```
% The problem structure hosts a manifold structure as well as function handles
% to define the cost function and its derivatives (here provided as Euclidean
% derivatives, which will be converted to their Riemannian equivalent).
problem.M = elliptopefactory(n, r);
problem.cost  = @(Y)    -trace(Y'*L*Y)/4;
problem.egrad = @(Y)    -(L*Y)/2;
problem.ehess = @(Y, U) -(L*U)/2;          % optional

% These diagnostics tools help make sure the gradient and Hessian are correct.
checkgradient(problem); pause;
checkhessian(problem);  pause;

% Minimize with trust-regions, a random initial guess and default options.
Y = trustregions(problem);
```

Randomly projecting Y yields a cut: `s = sign(Y*randn(r, 1))`. The Manopt distribution includes advanced code for this example, where the caching functionalities are used to avoid redundant computations of the product LY in the cost and the gradient, and the rank r is increased gradually to obtain a global solution of the max-cut SDP (and hence a formal upperbound), following a procedure described by Journée et al. (2010a).

Acknowledgments

NB and BM are research fellows with the FNRS (aspirants). This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office. It was financially supported by the Belgian FRFC.

References

P.-A. Absil and K.A. Gallivan. Joint diagonalization on the oblique manifold for independent component analysis. In *Acoustics, Speech and Signal Processing, ICASSP 2006. IEEE International Conference on*, volume 5, 2006. doi: 10.1109/ICASSP.2006.1661433.

P.-A. Absil, C. G. Baker, and K. A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, July 2007.

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

N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. In J. Shawe-Taylor et al., editors, *Advances in Neural Information Processing Systems 24 (NIPS)*, pages 406–414. 2011.

N. Boumal, A. Singer, and P.-A. Absil. Robust estimation of rotations from relative measurements by maximum likelihood. *Proceedings of the 52nd Conference on Decision and Control, CDC 2013*, 2013.

M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

I. Grubišić and R. Pietersz. Efficient rank reduction of correlation matrices. *Linear algebra and its applications*, 422(2):629–653, 2007.

M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010a.

M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *The Journal of Machine Learning Research*, 11: 517–553, 2010b.

R.H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *The Journal of Machine Learning Research*, 99:2057–2078, 2010.

G. Meyer, S. Bonnabel, and R. Sepulchre. Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. *The Journal of Machine Learning Research*, 12:593–625, 2011.

B. Mishra, G. Meyer, and R. Sepulchre. Low-rank optimization for distance matrix completion. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 4455–4460. IEEE, 2011.

B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Arxiv preprint arXiv:1209.0430*, 2012.

U. Shalit, D. Weinshall, and G. Chechik. Online learning in the embedded manifold of low-rank matrices. *The Journal of Machine Learning Research*, 13:429–458, 2012.

F.J. Theis, T.P. Cason, and P.-A. Absil. Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold. In *Independent Component Analysis and Signal Separation*, pages 354–361. Springer, 2009.

B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013. doi: 10.1137/110845768.