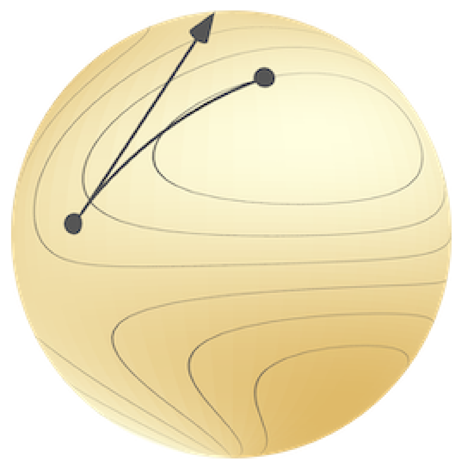
Manopt.jl



Recent new features and algorithms

Ronny Bergmann
ronny.bergmann@ntnu.no kellertuer
Dept. of Mathematical Sciences, NTNU, Trondheim, Norway.

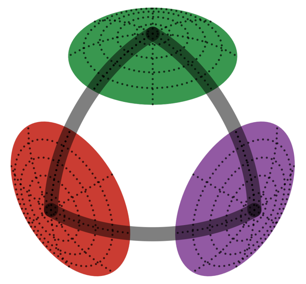
Motivation.

Optimization problems of the form

$$\operatorname*{arg\,min}_{p\in\mathcal{M}} f(p)$$

where $\mathcal{M}$ is a Riemannian manifold appear often, e. g.,

- the set of $n \times n$ symmetric positive definite matrices
- the set of $\mathbb{R}^{n\times m}$ matrices of fixed rank k
- a best subspace basis of some $\mathbb{R}^k \subset \mathbb{R}^n$
  $\Rightarrow \mathrm{St}(n,k) = \{ p \in \mathbb{R}^{n\times k} \,|\, p^{\mathrm{T}}p = I \}$
- a best k-dimensional subspace of $\mathbb{R}^n$
  $\Rightarrow \mathrm{Gr}(n,k) = \mathrm{St}(n,k)/\mathrm{St}(k,k)$



- the sphere, hyperbolic space, rotations, symplectic matrices or shape space,… see `Manifolds.jl`

Replace constraints by Riemannian geometry

Unconstrained algorithms via differential geometry

Provide easy access to Numerical Algorithms.

Features.

- generic algorithm interface with
  - a problem for the `AbstractManifold` & an allocating or mutating objective
  - a solver state specifying the algorithm

- Algorithms work on arbitrary manifolds that are implemented using `ManifoldsBase.jl`
- a library of stopping criteria including `&` and `|` to combine them
- a library of step sizes, debug, recording & statistics
- high-level interfaces, e. g.
  `gradient_descent(M, f, grad_f, p0)`
- can be used from within `JuMP.jl`
- Gradient & Hessian conversion from embedding
- use Automatic Differentiation via `ManifoldDiff.jl`

Turn Constraints into Geometry

and use

tutorials/getstarted/

Efficient Algorithms for Optimization on Riemannian Manifolds

Algorithms.

Derivative Free. Nelder–Mead, Particle Swarm, Mesh-adaptive direct Search (MADS), CMA-ES.

Gradient. Gradient descent including Averaged, Momentum, preconditioned Nesterov's gradient Conjugate gradient descent with several coefficients, Quasi Newton with several rules including L-BFGS.

Subgradient. Subgradient method, Convex Bundle Method, Proximal Bundle Method.

Hessian. Adaptive regularization with cubics (ARC) with a Lanczos sub solver, Trust Region with, e. g., a Steihaug-Toint sub solver.

Splitting based.

smooth. Levenberg-Marquardt, Stochastic & alternating gradient descent,

non-smooth. Proximal Point, Proximal Gradient, Chambolle-Pock, Cyclic Proximal Point, Difference of Convex (DC), DC proximal point, Douglas-Rachford, Primal-dual semi-smooth Newton.

Constrained. Augmented Lagrangian Method, Exact Penalty Method, Frank-Wolfe, Projected Gradient, Interior Point Newton.

Example.

The Riemannian center of mass.

```
using Manifolds, Manopt
M = Sphere(2); N = 500;
Q = [rand(M) for _ ∈ 1:N]
f(M,p)=sum(distance(M,p,q)^2 for q∈Q)/N
g(M,p) = sum(-2log(M,p,q) for q∈Q)/N
m = gradient_descent(M, f, g, Q[1])

# with debug every 10th iter. & record
s = gradient_descent(M, f, g, Q[1];
  debug = [:Iteration, :Cost, 10],
  record = [:Iterate, :Cost, :Change],
  return_state = true, # to access
)                      # recorded values:
r = get_record(s)
m2 = get_solver_result(s) # same as m
```

References.

RB (2022). Manopt.jl: Optimization on Manifolds in Julia. Journal of Open Source Software 7.70, p. 3866. DOI: 10.21105/joss.03866.

Axen, S. D., M. Baran, RB, and K. Rzecki (2023). Manifolds.jl: An Extensible Julia Framework for Data Analysis on Manifolds. ACM Transactions on Mathematical Software 49.4. DOI: 10.1145/3618296.

NTNU | Norwegian University of Science and Technology