NTNU

Norwegian University of Science and Technology

Nonsmooth Optimization on Riemannian Manifolds in Manopt.jl

Ronny Bergmann

Optimization Research Seminar
Heidelberg, December 8, 2022

Overview

Task. We aim to solve

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

where

- $\mathcal{M}$ is a Riemannian manifold
- $f\colon \mathcal{M} \to \mathbb{R}$ is nonsmooth and possibly high-dimensional

Roadmap.

1. Motivation
2. Algorithms
3. Numerical examples in `Manopt.jl`

Intuition: Embedded Manifolds

Consider $h\colon \mathbb{R}^n \to \mathbb{R}^k$, $1 \leq k \leq n$ as an equality constraint $h(p) = 0$.
If $\operatorname{rank} Dh(p) = k$ for all p with $h(p) = 0$, then

$$\mathcal{M} := \left\{ p \in \mathbb{R}^n \,\middle|\, h(p) = 0 \right\}$$

is a (smooth embedded sub-)manifold of $\mathbb{R}^n$ of dimension $m = n - k$,
cf. Definition 3.10. [Boumal 2022]

Example. The Sphere $\mathbb{S}^m \subset \mathbb{R}^n$ has $h(p) = \|p\| - 1 = 0$
$\Rightarrow$ We have $k = 1$ and $m = n - 1$.

Actually. It is enough to find such a function h locally around every p.

Intuition: Retractions – "Walking on Manifolds"

Interpretation. With $\operatorname{rank} Dh(p) = k$ we get $\dim\ker Dh(p) = m = n - k$
$\Rightarrow$ we have m "directions", where $Dh(p)[X] = 0$.
We call the set of these "directions" the Tangent space $T_p\mathcal{M}$.
The (disjoint) union of all tangent spaces is called the tangent bundle $T\mathcal{M}$.

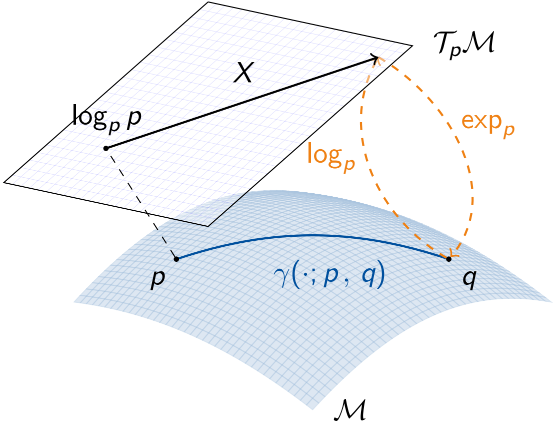
Goal. We would like to "walk" into these directions while staying on the manifold.

Definition. A function $R\colon T\mathcal{M} \to \mathcal{M}$, also denoted by $R_p\colon T_p\mathcal{M} \to \mathcal{M}$ for each $p \in \mathcal{M}$, is called a retraction
if each curve $c(t) = R_p(tX)$ satisfies $c(0) = p$ and $c'(0) = X$.

A Riemannian Manifold $\mathcal{M}$

A d-dimensional Riemannian manifold can be informally defined as a set $\mathcal{M}$ covered with a 'suitable' collection of charts, that identify subsets of $\mathcal{M}$ with open subsets of $\mathbb{R}^d$ and a continuously varying inner product on the tangent spaces.

[Absil, Mahony, and Sepulchre 2008]



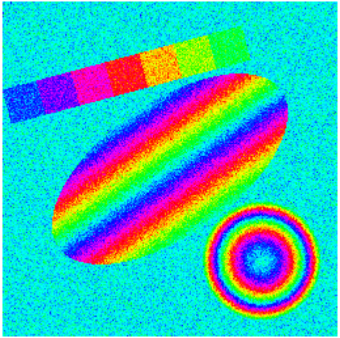
Notation.

- Logarithmic map $\log_p q = \dot\gamma(0; p, q)$
- Exponential map $\exp_p X = \gamma_{p,X}(1)$
- Geodesic $\gamma(\cdot; p, q)$
- Tangent space $\mathcal{T}_p\mathcal{M}$
- inner product $(\cdot, \cdot)_p$

Manifold-valued Signal & Image Processing

Tasks in image processing are often phrased as an optimisation problem.
Here. The pixel take values on a manifold

- phase-valued data ($\mathbb{S}^1$)
- wind-fields, GPS ($\mathbb{S}^2$)
- DT-MRI ($\mathcal{P}(3)$)
- EBSD, (grain) orientations ($\mathrm{SO}(n)$)



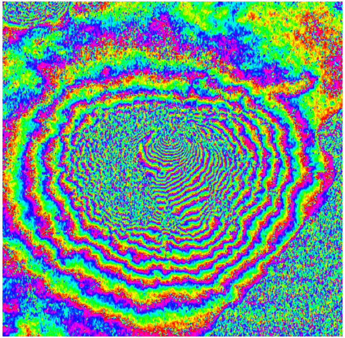
Artificial noisy phase-valued data.

Tasks. Denoising, Inpainting, labeling (classification), deblurring,...

Manifold-valued Signal & Image Processing

Tasks in image processing are often phrased as an optimisation problem.
Here. The pixel take values on a manifold

- phase-valued data ($\mathbb{S}^1$)
- wind-fields, GPS ($\mathbb{S}^2$)
- DT-MRI ($\mathcal{P}(3)$)
- EBSD, (grain) orientations ($\mathrm{SO}(n)$)



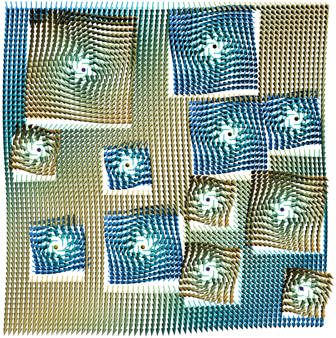
InSAR-Data of Mt. Vesuvius.
[Rocca, Prati, and Guarnieri 1997]

Tasks. Denoising, Inpainting, labeling (classification), deblurring,...

Manifold-valued Signal & Image Processing

Tasks in image processing are often phrased as an optimisation problem.
Here. The pixel take values on a manifold

- phase-valued data ($\mathbb{S}^1$)
- wind-fields, GPS ($\mathbb{S}^2$)
- DT-MRI ($\mathcal{P}(3)$)
- EBSD, (grain) orientations ($\mathrm{SO}(n)$)



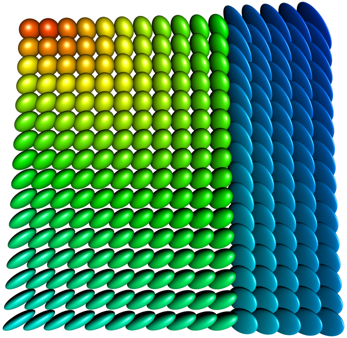
Artificial noisy data on the sphere $\mathbb{S}^2$.

Tasks. Denoising, Inpainting, labeling (classification), deblurring,...

Manifold-valued Signal & Image Processing

Tasks in image processing are often phrased as an optimisation problem.
Here. The pixel take values on a manifold

- phase-valued data ($\mathbb{S}^1$)
- wind-fields, GPS ($\mathbb{S}^2$)
- DT-MRI ($\mathcal{P}(3)$)
- EBSD, (grain) orientations ($\mathrm{SO}(n)$)



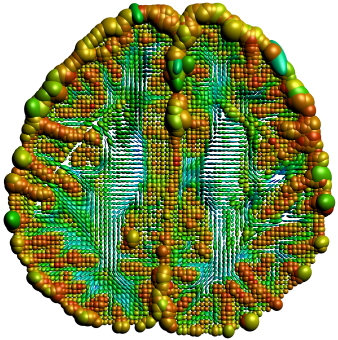
Artificial diffusion data,
each pixel is a symmetric positive matrix.

Tasks. Denoising, Inpainting, labeling (classification), deblurring,...

Manifold-valued Signal & Image Processing

Tasks in image processing are often phrased as an optimisation problem.
Here. The pixel take values on a manifold

- phase-valued data ($\mathbb{S}^1$)
- wind-fields, GPS ($\mathbb{S}^2$)
- DT-MRI ($\mathcal{P}(3)$)
- EBSD, (grain) orientations ($\mathrm{SO}(n)$)



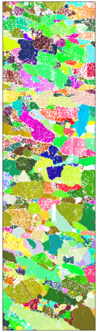
DT-MRI of the human brain.
Camino Profject: cmic.cs.ucl.ac.uk/camino

Tasks. Denoising, Inpainting, labeling (classification), deblurring,...

Manifold-valued Signal & Image Processing

Tasks in image processing are often phrased as an optimisation problem.
Here. The pixel take values on a manifold

- phase-valued data ($\mathbb{S}^1$)
- wind-fields, GPS ($\mathbb{S}^2$)
- DT-MRI ($\mathcal{P}(3)$)
- EBSD, (grain) orientations ($\mathrm{SO}(n)$)



Grain orientations in EBSD data.
MTEX toolbox: mtex-toolbox.github.io

Tasks. Denoising, Inpainting, labeling (classification), deblurring,...

Why Manifolds?

- A constrained problem on $\mathbb{R}^n$ $\Rightarrow$ an unconstrained problem on $\mathcal{M}$
- The "type of convexity" changes: Convexity is defined along geodesics γ
- If we can omit "working" in the embedding $\Rightarrow$ dimension reduction

! we need efficient ways to compute e. g. retractions.

The Smooth Case & Gradient Descent

For a smooth function $f\colon \mathcal{M} \to \mathbb{R}$ we have

- The differential $Df\colon T\mathcal{M} \to \mathbb{R}$, or phrased differently $Df(p)\colon T_p\mathcal{M} \to \mathbb{R}$
- the gradient $\operatorname{grad} f(p) \in T_p\mathcal{M}$ is the Riesz representer defined by the property

$$Df(p)[X] = (\operatorname{grad} f(p)\,,\, X)_p, \qquad \text{for all } X \in T_p\mathcal{M}$$

$\Rightarrow$ Like in $\mathbb{R}^n$: $Y = -\operatorname{grad} f(p)$ is the direction of steepest descent.

Algorithm. Gradient descent.
Given f and a retraction R we perform

$$p_{k+1} = R_{p_k}(-s_k \operatorname{grad} f(p))$$

for some step size(s) s_k – e. g. an Armijo backtracking linesearch, cf. Ch. 4.1 [Absil, Mahony, and Sepulchre 2008]

Proximal Map

For $f\colon \mathcal{M} \to \overline{\mathbb{R}}$ and $\lambda > 0$ we define the Proximal Map as

[Moreau 1965; Rockafellar 1970; Ferreira and Oliveira 2002]

$$\operatorname{prox}_{\lambda f}(p) := \operatorname*{arg\,min}_{u \in \mathcal{M}} d_{\mathcal{M}}(u,p)^2 + \lambda f(u).$$

! For a minimizer u^* of f we have $\operatorname{prox}_{\lambda f}(u^*) = u^*$.

- For f proper, convex, lsc:
  - the proximal map is unique.
  - Proximal-Point-Algorithm:
    $p_k = \operatorname{prox}_{\lambda f}(p_{k-1})$ converges to $\operatorname{arg\,min} f$

The Cyclic Proximal Point Algorithm

If we can split our nonsmooth $f(p) = \sum_{i=1}^{c} g_i(p)$, we can use the Cyclic Proximal Point-Algorithmus (CPPA): [Bertsekas 2011; Bačák 2014]

$$p_{k+\frac{i+1}{c}} = \operatorname{prox}_{\lambda_k g_i}(p_{k+\frac{i}{c}}), \quad i = 0, \ldots, c-1, \; k = 0, 1, \ldots$$

On a Hadamard manifold $\mathcal{M}$:
convergence to a minimizer of f if

- all g_i proper, convex, lower semi-continuous
- $\{\lambda_k\}_{k\in\mathbb{N}} \in \ell_2(\mathbb{N})\backslash\ell_1(\mathbb{N})$.

! no convergence rate

The Exact Riemannian Chambolle–Pock Algorithm

[RB, Herzog, Silva Louzeiro, Tenbrinck, and Vidal-Núñez 2021; Chambolle and Pock 2011]

Assume. $f(p) = F(p) + G(\Lambda(p))$, with $\Lambda\colon \mathcal{M} \to \mathcal{N}$.

Input: m, $p^{(0)} \in \mathcal{C} \subset \mathcal{M}$, $n = \Lambda(m)$, $\xi_n^{(0)} \in \mathcal{T}_n^*\mathcal{N}$, and parameters σ, τ, $\theta > 0$

1: $k \leftarrow 0$

2: $\bar{p}^{(0)} \leftarrow p^{(0)}$

3: **while** not converged **do**

4: $\quad \xi_n^{(k+1)} \leftarrow \operatorname{prox}_{\tau G_n^*}\Bigl(\xi_n^{(k)} + \tau\bigl(\log_n \Lambda(\bar{p}^{(k)})\bigr)^\flat\Bigr)$

5: $\quad p^{(k+1)} \leftarrow \operatorname{prox}_{\sigma F}\Biggl(\exp_{p^{(k)}}\Bigl(\mathrm{P}_{p^{(k)}\leftarrow m}\bigl(-\sigma D\Lambda(m)^*[\xi_n^{(k+1)}]\bigr)^\sharp\Bigr)\Biggr)$

6: $\quad \bar{p}^{(k+1)} \leftarrow \exp_{p^{(k+1)}}\bigl(-\theta \log_{p^{(k+1)}} p^{(k)}\bigr)$

7: $\quad k \leftarrow k+1$

8: **end while**

Output: $p^{(k)}$

Implementing Manifolds & Optimisation – in Julia.

- abstract definition of manifolds and properties thereon
  e. g. different metrics, retractions, embeddings

$\Rightarrow$ implement abstract algorithms for generic manifolds

- easy to implement own manifolds & easy to use
- well-documented and well-tested
- fast.

Why Julia?

- high-level language, properly typed
- multiple dispatch (cf. `f(x)`, `f(x::Number)`, `f(x::Int)`)
- just-in-time compilation, solves two-language problem
- I like the language – and the community.

NTNU

Implementing a Riemannian Manifold

`ManifoldsBase.jl` uses a `AbstractManifold{𝔽}` with type parameter $\mathbb{F} \in \{\mathbb{R}, \mathbb{C}, \mathbb{H}\}$ to provide an interface for implementing functions like

- `inner(M, p, X, Y)` for the Riemannian metric $(X, Y)_p$
- `exp(M, p, X)` and `log(M, p, q)`,
- more general: `retract(M, p, X, m)`, where `m` is a retraction method
- similarly: `parallel_transport(M, p, X, q)` and `vector_transport_to(M, p, X, q, m)`

for your manifold `M` a subtype of the abstract manifold `Manifold{𝔽}`.

☺ mutating version `exp!(M, q, p, X)` works in place in `q`

➔ basis for generic algorithms working on any `Manifold` and generic functions like `norm(M,p,X)`, `geodesic(M, p, X)` and `shortest_geodesic(M, p, q)`

🔗 juliamanifolds.github.io/ManifoldsBase.jl/

NTNU

Manifolds.jl – A library of manifolds in Julia

`Manifolds.jl` is based on the `ManifoldsBase.jl` interface. [Axen, Baran, RB, and Rzecki 2021]

Features.

- different metrics
- Lie groups
- Build manifolds using
  - Product manifold $\mathcal{M}_1 \times \mathcal{M}_2$
  - Power manifold $\mathcal{M}^{n \times m}$
  - Tangent bundle
- Embedded manifolds
- perform statistics
- well-documented, including formulae and references
- well-tested, >98 % code cov.

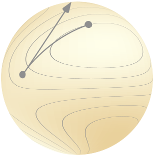
Manifolds. For example

- (unit) Sphere, Circle & Torus
- Fixed Rank Matrices
- (Generalized) Stiefel & Grassmann
- Hyperbolic space
- Rotations, O(n), SO(n), SU(n)
- Several further Lie groups
- Symmetric positive definite matrices
- Symplectic & Symplectic Stiefel
- ...

juliamanifolds.github.io/Manifolds.jl/

JuliaCon 2020 youtu.be/md-FnDGCh9M

Manopt.jl: Optimisation on Manifolds in Julia



Goal. Provide optimisation algorithms on Riemannian manifolds, based on `ManifoldsBase.jl` & works any manifold from `Manifolds.jl`.

Features.

- generic algorithm framework: With `Problem P` and `Options O`
  - `initialize_solver!(P,O)`
  - `step_solver!(P, O, i)`: ith step
- run algorithm: call `solve(P,O)`
- generic debug and recording
- step sizes and stopping criteria.

Manopt Family.

- manoptjl.org [RB 2022]
- manopt.org [Boumal, Mishra, Absil, and Sepulchre 2014]
- pymanopt.org [Townsend, Koep, and Weichwald 2016]

Algoirthms.

- Gradient Descent
  CG, Stochastic, Momentum, …
- Quasi-Newton
  BFGS, DFP, Broyden, SR1, …
- Nelder-Mead, Particle Swarm
- Subgradient Method
- Trust Regions
- Chambolle-Pock
- Douglas-Rachford
- Cyclic Proximal Point

Selected References

- Axen, S. D., M. Baran, RB, and K. Rzecki (2021). *Manifolds.jl: An Extensible Julia Framework for Data Analysis on Manifolds*. arXiv: 2106.08777.
- Bačák, M. (2014). "Computing medians and means in Hadamard spaces". In: *SIAM Journal on Optimization* 24.3, pp. 1542–1566. DOI: 10.1137/140953393.
- RB (2022). "Manopt.jl: Optimization on Manifolds in Julia". In: *Journal of Open Source Software* 7.70, p. 3866. DOI: 10.21105/joss.03866.
- RB, R. Herzog, M. Silva Louzeiro, D. Tenbrinck, and J. Vidal-Núñez (Jan. 2021). "Fenchel duality theory and a primal-dual algorithm on Riemannian manifolds". In: *Foundations of Computational Mathematics*. DOI: 10.1007/s10208-020-09486-5. arXiv: 1908.02022.
- Boumal, N., B. Mishra, P.-A. Absil, and R. Sepulchre (2014). "Manopt, a Matlab toolbox for optimization on manifolds". In: *The Journal of Machine Learning Research* 15, pp. 1455–1459. URL: https://www.jmlr.org/papers/v15/boumal14a.html.
- Boumal, N. (June 2022). *An introduction to optimization on smooth manifolds*. To appear with Cambridge University Press. URL: https://www.nicolasboumal.net/book.
- Chambolle, A. and T. Pock (2011). "A first-order primal-dual algorithm for convex problems with applications to imaging". In: *Journal of Mathematical Imaging and Vision* 40.1, pp. 120–145. DOI: 10.1007/s10851-010-0251-1.
- Townsend, J., N. Koep, and S. Weichwald (2016). "Pymanopt: A Python Toolbox for Optimization on Manifolds using Automatic Differentiation". In: *Journal of Machine Learning Research* 17.137, pp. 1–5. URL: http://jmlr.org/papers/v17/16-177.html.