Geomstats: A Python Package for Riemannian Geometry in Machine Learning

Nina Miolane nmiolane@stanford.edu
Nicolas Guigui nicolas.guigui@inria.fr
Alice Le Brigant alice.le-brigant@univ-paris1.fr
Johan Mathe johan@froglabs.ai
Benjamin Hou benjamin.hou11@imperial.ac.uk
Yann Thanwerdas yann.thanwerdas@inria.fr
Stefan Heyder stefan.heyder@tu-ilmenau.de
Olivier Peltre opeltre@gmail.com
Niklas Koep niklas.koep@gmail.com
Hadi Zaatiti hadi.zaatiti@irt-systemx.fr
Hatem Hajri hatem.hajri@irt-systemx.fr
Yann Cabanes yann.cabanes@gmail.com
Thomas Gerald thomas.gerald@lip6.fr
Paul Chauchat pchauchat@gmail.com
Christian Shewmake cshewmake2@gmail.com
Daniel Brooks jimrivers75@gmail.com
Bernhard Kainz b.kainz@imperial.ac.uk
Claire Donnat cdonnat@stanford.edu
Susan Holmes susan@stat.stanford.edu
Xavier Pennec xavier.pennec@inria.fr

Editor: Cheng Soon Ong

Abstract

We introduce Geomstats, an open-source Python package for computations and statistics on nonlinear manifolds such as hyperbolic spaces, spaces of symmetric positive definite matrices, Lie groups of transformations, and many more. We provide object-oriented and extensively unit-tested implementations. Manifolds come equipped with families of Riemannian metrics with associated exponential and logarithmic maps, geodesics, and parallel transport. Statistics and learning algorithms provide methods for estimation, clustering, and dimension reduction on manifolds. All associated operations are vectorized for batch computation and provide support for different execution backends—namely NumPy, PyTorch, and TensorFlow. This paper presents the package, compares it with related libraries, and provides relevant code examples. We show that Geomstats provides reliable building blocks to both foster research in differential geometry and statistics and democratize the use of Riemannian geometry in machine learning applications. The source code is freely available under the MIT license at `geomstats.ai`.

Keywords: differential geometry, Riemannian geometry, statistics, machine learning, manifold

1. Introduction

Data on manifolds naturally arise in different fields. **Hyperspheres** model directional data in molecular and protein biology (Kent and Hamelryck, 2005) and are used through Principal Nested Spheres to model some aspects of 3D shapes (Jung et al., 2012; Hong et al., 2016). Computations on **hyperbolic spaces** arise for impedance density estimation (Huckemann et al., 2010), geometric network comparison (Asta and Shalizi, 2014), and the analysis of reflection coefficients extracted from a radar signal (Chevallier et al., 2015). **Symmetric positive definite (SPD) matrices** characterize data from diffusion tensor imaging (DTI) (Pennec et al., 2006; Yuan et al., 2012) and functional magnetic resonance imaging (fMRI) (Sporns et al., 2005). Covariance matrices, which are also SPD matrices, appear in applications such as automatic speech recognition (ASR) systems (Shinohara et al., 2010), image and video descriptors (Harandi et al., 2014), or air traffic complexity representation (Brigant and Puechmorel, 2019). The **Lie groups of transformations** SO(3) and SE(3) appear naturally when dealing with articulated objects like the human spine (Arsigny, 2006; Boisvert et al., 2006) or the pose of a camera (Kendall and Cipolla, 2017; Hou et al., 2018). **Stiefel manifolds** are used to process video action data or to analyze two vector-cardiograms (Chakraborty and Vemuri, 2019). **Grassmannians** appear in computer vision to perform video-based face recognition and shape recognition (Turaga et al., 2008). Statistics on **landmark spaces** are used in anthropology and many applied fields to describe biological shapes (Richtsmeier et al., 1992). More generally, Kendall shape spaces gave rise to a very important literature in shape statistics (Dryden and Mardia, 1998). A variety of applications use tools from infinite-dimensional Riemannian geometry to study the shapes of **discretized curves** such as those sampled from closed two or three-dimensional curves defining the contours of organs in computational anatomy (Younes, 2012). In addition, open curves that describe the temporal evolution of physical phenomena are embedded in various manifolds, for example in a Lie group (Celledoni et al., 2015) or in the hyperbolic plane (Le Brigant, 2017).

Yet, the adoption of methods from differential geometry has been inhibited by the lack of a reference implementation. Code sequences are often custom-tailored for specific problems and are not easily reused. Some Python packages do exist, but they often focus on optimization: Pymanopt (Townsend et al., 2016), Geoopt (Bécigneul and Ganea, 2018; Kochurov et al., 2019), and McTorch (Meghwanshi et al., 2018). Others are dedicated to a single manifold: PyRiemann on SPD matrices (Barachant, 2015), PyQuaternion on 3D rotations (Wynn, 2014), and PyGeometry on spheres, toruses, 3D rotations and translations (Censi, 2012). Lastly, others lack unit-tests and continuous integration: TheanoGeometry (Kühnel and Sommer, 2017). There is a need for an open-source implementation of differential geometry and associated learning algorithms for manifold-valued data.

We present Geomstats, an open-source Python package for computations and statistics on nonlinear manifolds. Geomstats has three main objectives: (i) foster research in differential geometry and geometric statistics by providing low-level code to gain intuition or test a theorem and a platform to share algorithms; (ii) democratize the use of geometric statistics by implementing user-friendly geometric learning algorithms using Scikit-Learn API; and (iii) provide educational support to learn "hands-on" differential geometry and geometric statistics, through its examples, notebooks and visualizations.

2. Implementation Overview

The package `geomstats` is organized into two main modules: `geometry` and `learning`. **The module `geometry` implements concepts in Riemannian geometry** with an object-oriented approach. Manifolds mentioned in the introduction are available as classes that inherit from the base class `Manifold`. The base class `RiemannianMetric` provides methods such as the geodesic distance between two points, the exponential and logarithm maps at a base point, etc. As examples, `HyperbolicMetric`, `StiefelCanonicalMetric`, the Lie groups' `InvariantMetric` s, or the curves' `L2Metric` and `SRVMetric` (Srivastava et al., 2011) inherit from `RiemannianMetric`. Going beyond Riemannian geometry, the class `Connection` implements affine connections. The implementation uses automatic differentiation with autograd to allow computations on manifolds when closed-form formulae do not exist. The API of the module `geometry` uses terms from differential geometry to foster contributions of researchers from this field. **The module `learning` implements statistics and learning algorithms** for data on manifolds. The code is object-oriented and classes inherit from Scikit-Learn base classes and mixins: `BaseEstimator`, `ClassifierMixin`, `RegressorMixin`, etc. This module provides classes such as `FrechetMean`, `KMeans`, `TangentPCA`, for implementations of Fréchet mean estimators (Fréchet, 1948), K-means, and principal component analysis (PCA) designed for manifold data. The API of the module `learning` follows Scikit-Learn's API, being therefore user-friendly to machine learning researchers and engineers.

The code follows international standards for readability and ease of collaboration, is vectorized for batch computations, undergoes unit-testing with continuous integration using Travis, relies on either Numpy, TensorFlow or PyTorch backends. The package comes with a `visualization` module to provide intuition on differential geometry (see Figure 1), and with a `datasets` module that provides toy data sets on manifolds. The repositories `examples` and `notebooks` provide convenient starting points to get familiar with geomstats.

The GitHub repository at `github.com/geomstats/geomstats` offers a convenient way to ask for help or request features by raising issues. The website `geomstats.github.io` provides documentation for users and important guidelines for those wishing to contribute to the project.

3. Comparison and Interaction with Existing Packages

This section compares the package `geomstats` with related Python implementations on differential geometry and learning. Table 1 compares the geometric operations and Table 2 compares the engineering infrastructures.

The library TheanoGeometry (Kühnel and Sommer, 2017) is the most closely related to Geomstats and provides nonlinear statistics and stochastic equations on Riemannian manifolds. The differential geometric tensors are computed with automatic differentiation. However, this library does not provide statistical learning algorithms and lacks engineering maintenance. Several other packages focus on optimization on Riemannian manifolds. Pymanopt (Townsend et al., 2016) computes gradients and Hessian-vector products on Riemannian manifolds with automatic differentiation and provides the following solvers: steepest descent, conjugate gradient, the Nelder-Mead algorithm, particle swarm optimization, and the Riemannian trust regions. Geoopt (Kochurov et al., 2019) focuses on stochastic adap-

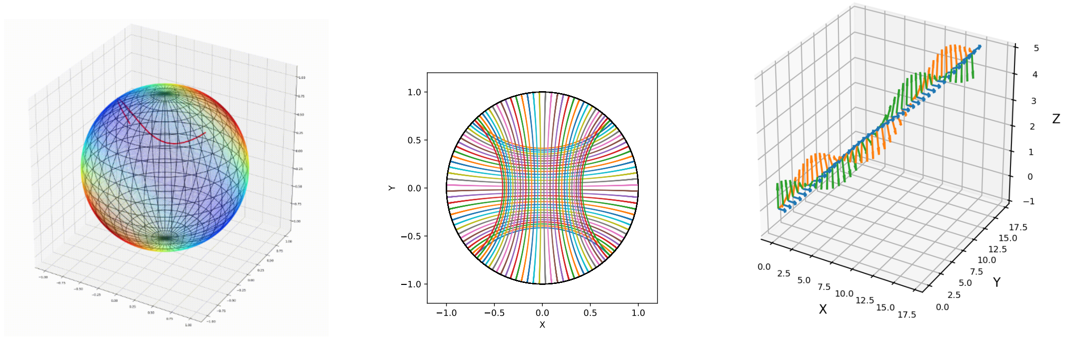

Figure 1: Left: Minimization of a scalar field on the sphere $\mathbb{S}^2$ using Riemannian gradient descent. Middle: Regular geodesic grid on the hyperbolic space $\mathbb{H}^2$ in Poincaré disk representation. Right: Geodesic on the Lie group SE(3) for the canonical left-invariant metric. These examples and more are available at `geomstats.ai`.

tive optimization on Riemannian manifolds, for machine learning problems. The library provides stochastic solvers, stochastic gradient descent and Adam, as well as the following samplers: Stochastic Gradient Langevin Dynamics, Hamiltonian Monte-Carlo, Stochastic Gradient Hamiltonian Monte-Carlo. Lastly, McTorch (Meghwanshi et al., 2018) provides optimization on Riemannian manifold for deep learning by adding a "Manifold" parameter to PyTorch's network layers and optimizers. The library provides the following solvers: stochastic gradient descent, AdaGrad and conjugate gradients. As these libraries focus on optimization, they substitute potentially computationally expensive operations by practical proxies, for example, by replacing exponential maps by so-called retractions. However, they are less modular than Geomstats in terms of the Riemannian geometry. For example, each manifold comes with a single Riemannian metric, in contrast to Geomstats where families of Riemannian metrics are implemented. Furthermore, they do not provide statistical learning algorithms.

The optimization libraries are complementary to Geomstats and interact easily with it. Geomstats provides low-level implementations of Riemannian geometry that can be used to define optimization costs. In turn, an optimization library can provide an efficient solver to use within the implementation of Geomstats' learning algorithms. An example of such interactions, between Pymanopt and Geomstats can be found in Geomstats' `examples` folder.

4. Usage: Examples of Learning on Riemannian manifolds

Three steps are needed to run learning algorithms on manifolds with Geomstats: (i) instantiate the manifold of interest, (ii) instantiate the learning algorithm of interest, and (iii) run the algorithm. The following code snippet illustrates the use of K-means on the hypersphere.

```
sphere = Hypersphere(dim=5)
```

	Manifolds	**Geometry**
Pymanopt	Euclidean manifold, symmetric matrices, sphere, complex circle, SO(n), Stiefel, Grassmannian, oblique manifold, SPD(n), elliptope, fixed-rank PSD matrices	Exponential and logarithmic maps, retraction, vector transport, `egrad2rgrad`, `ehess2rhess`, inner product, distance, norm
Geoopt	Euclidean manifold, sphere, Stiefel, Poincaré ball	Same as Pymanopt
McTorch	Stiefel, SPD(n)	Same as Pymanopt
TheanoGeometry	Sphere, ellipsoid, SPD(n), Landmarks, GL(n), SO(n), SE(n)	Inner product, exponential and logarithmic maps, parallel transport, Christoffel symbols, Riemann, Ricci and scalar curvature, geodesics, Fréchet mean
Geomstats	Euclidean, Minkowski, hyperbolic space, Poincaré polydisk, hypersphere, SO(n), SE(n), GL(n), Stiefel, Grassmannian, SPD(n), symmetric matrices, skew-symmetric matrices, discretized curves on manifolds, landmarks on manifolds	Levi-Civita connection, Christoffel symbols, parallel transport, exponential and logarithmic maps, inner product, distance, norm, geodesics, group invariant metrics, Fréchet means and learning algorithms on manifolds

Table 1: Comparison of libraries in terms of geometric operations

	Backends	**Continuous integration (CI) and coverage**
Pymanopt	Autograd, PyTorch, TensorFlow, Theano	CI, coverage 85%
Geoopt	PyTorch	75%
McTorch	PyTorch	CI, coverage 84%
TheanoGeometry	Theano	No CI, no unit tests
Geomstats	NumPy, PyTorch, TensorFlow	CI, coverage 92% (NumPy), 76% (TensorFlow), 79% (PyTorch)

Table 2: Comparison of code infrastructure

```
data = sphere.random_uniform(n_samples=10)
clustering = OnlineKMeans(metric=sphere.metric, n_clusters=4)
clustering = clustering.fit(data)
```

The following code snippet shows the use of tangent PCA on the 3D rotations.

```
so3 = SpecialOrthogonal(n=3, point_type='vector')
metric = so3.bi_invariant_metric
data = so3.random_uniform(n_samples=10)
tpca = TangentPCA(metric=metric, n_components=2)
tpca = tpca.fit(data)
tangent_projected_data = tpca.transform(data)
```

All geometric computations are performed behind the scenes. The user only needs a high-level understanding of Riemannian geometry. Each algorithm can be used with any of the manifolds and metric implemented in the package. The folders `examples` and `notebooks` provide many more code snippets that help users get started with Geomstats.

5. Conclusion

We presented the Python Package Geomstats, with the aim to provide the wider Machine Learning community with off-the-shelf geometric learning algorithms on a wide variety of manifolds, and flexibility in the choice of metrics, while being faithful to the mathematician's formulation of Riemannian geometry. This sometimes comes at cost of efficiency, and future contributions will be devoted to addressing this caveat.

Acknowledgments

This work is partially supported by the National Science Foundation, grant NSF DMS RTG 1501767, the Inria-Stanford associated team GeomStats, the European Research Council (ERC) under the EU Horizon 2020 research and innovation program (grant agreement G-Statistics No. 786854) and by the French Government through the 3IA Côte d'Azur Investments in the Future project (National Research Agency ANR-19-P3IA-0002).

References

Vincent Arsigny. *Processing Data in Lie Groups: An Algebraic Approach. Application to Non-Linear Registration and Diffusion Tensor MRI.* PhD thesis, École polytechnique, 11 2006.

Dena Asta and Cosma Rohilla Shalizi. Geometric Network Comparison. *Journal of Machine Learning Research*, 2014. ISSN 1939-1374. doi: 10.1109/PES.2006.1709566.

Alexandre Barachant. PyRiemann: Python package for covariance matrices manipulation and Biosignal classification with application in Brain Computer interface, 2015. URL `https://github.com/alexandrebarachant/pyRiemann`.

Gary Bécigneul and Octavian-Eugen Ganea. Riemannian Adaptive Optimization Methods. In *Proceedings of the International Conference on Learning Representations (ICLR) 2019*, pages 1–16, 2018.

Jonathan Boisvert, Xavier Pennec, Hubert Labelle, Farida Cheriet, and Nicholas Ayache. Principal spine shape deformation modes using riemannian geometry and articulated models. *Lecture Notes in Computer Science*, 4069 LNCS:346–355, 2006. ISSN 16113349.

Alice Le Brigant and Stéphane Puechmorel. Quantization and clustering on riemannian manifolds with an application to air traffic analysis. *Journal of Multivariate Analysis*, 173:685–703, 2019. doi: 10.1016/j.jmva.2019.05.008.

Elena Celledoni, Markus Eslitzbichler, and Alexander Schmeding. Shape analysis on Lie groups with applications in computer animation. *The Journal of Geometric Mechanics*, 8 (3):273–304, 2015.

Andrea Censi. PyGeometry: Library for handling various differentiable manifolds., 2012. URL `https://github.com/AndreaCensi/geometry`.

Rudrasis Chakraborty and Baba Vemuri. Statistics on the (compact) Stiefel manifold : Theory and Applications. *Annals of Statistics*, 47(1):415–438, 2019.

Emmanuel Chevallier, Frédéric Barbaresco, and Jesus Angulo. Probability density estimation on the hyperbolic space applied to radar processing. *Lecture Notes in Computer Science*, 9389:753–761, 2015. ISSN 16113349.

Ian Dryden and Kanti Mardia. *Statistical shape analysis, with Applications in R.* John Wiley & Sons, New York, 1998.

Maurice Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l'institut Henri Poincaré*, 10(4):215–310, 1948.

Mehrtash Tafazzoli Harandi, Richard Hartley, Brian Lovell, and Conrad Sanderson. Sparse Coding on Symmetric Positive Definite Manifolds using Bregman Divergences. *CoRR*, abs/1409.0, 2014.

Junpyo Hong, Jared Vicory, Jörn Schulz, Martin Styner, James Stephen Marron, and Stephen Pizer. Non-Euclidean Classification of Medically Imaged Objects via s-reps. *Medical Image Analysis*, 31:37–45, 2016.

Benjamin Hou, Nina Miolane, Bishesh Khanal, Mathew Lee, Amir Alansary, Steven McDonagh, Jo Hajnal, Daniel Rueckert, Ben Glocker, and Bernhard Kainz. Computing CNN loss and gradients for pose estimation with Riemannian geometry. In *Proceedings of the conference of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 11070 LNCS, 2018. ISBN 9783030009274.

Stephan Huckemann, Peter Kim, Ja Yong Koo, and Axel Munk. Möbius deconvolution on the hyperbolic plane with application to impedance density estimation. *Annals of Statistics*, 38(4):2465–2498, 2010. ISSN 00905364. doi: 10.1214/09-AOS783.

Sungkyu Jung, Ian Dryden, and James Stephen Marron. Analysis of principal nested spheres. *Biometrika*, 99(3):551–568, 2012. ISSN 00063444. doi: 10.1093/biomet/ass022.

Alex Kendall and Roberto Cipolla. Geometric Loss Functions for Camera Pose Regression with Deep Learning. In *Proceedings of the conference of Computer Vision and Pattern Recognition (CVPR)*, 2017. doi: 10.1109/CVPR.2017.694.

John Kent and Thomas Hamelryck. Using the Fisher-Bingham distribution in stochastic models for protein structure. *Quantitative Biology, Shape Analysis, and Wavelets*, 24(1): 57–60, 2005.

Maxim Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian Adaptive Optimization Methods with pytorch optim, 2019. URL `https://github.com/geoopt/geoopt`.

Line Kühnel and Stefan Sommer. Computational Anatomy in Theano. In *Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics*, pages 164–176, Cham, 2017. Springer International Publishing. URL `https://bitbucket.org/stefansommer/theanogeometry`.

Alice Le Brigant. Computing distances and geodesics between manifold-valued curves in the SRV framework. *Journal of Geometric Mechanics*, 2017.

Mayank Meghwanshi, Pratik Jawanpuria, Anoop Kunchukuttan, Hiroyuki Kasai, and Bamdev Mishra. McTorch, a manifold optimization library for deep learning, 2018. URL `https://github.com/mctorch/mctorch`.

Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A Riemannian Framework for Tensor Computing. *International Journal of Computer Vision*, 66(1):41–66, 1 2006.

Joan Richtsmeier, James Cheverud, and Subhash Lele. Advances in Anthropological Morphometrics. *Annual Review of Anthropology*, 21(5):283–305, 1992.

Yusuke Shinohara, Takashi Masuko, and Masami Akamine. Covariance clustering on Riemannian manifolds for acoustic model compression. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4326–4329, 3 2010.

Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: A structural description of the human brain. *PLoS Computational Biology*, 1(4):0245–0251, 2005. ISSN 15537358. doi: 10.1371/journal.pcbi.0010042.

Anuj Srivastava, Eric Klassen, Shantanu Joshi, and Ian Jermyn. Shape analysis of elastic curves in euclidean spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1415–1428, 2011.

James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research*, 17(137):1–5, 2016. ISSN 15337928. URL `https://github.com/pymanopt/pymanopt`.

Pavan Turaga, Ashok Veeraraghavan, and Rama Chellappa. Statistical analysis on Stiefel and Grassmann Manifolds with applications in Computer Vision. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

Kieran Wynn. PyQuaternions: A fully featured, pythonic library for representing and using quaternions, 2014. URL `https://github.com/KieranWynn/pyquaternion`.

Laurent Younes. Spaces and manifolds of shapes in computer vision: An overview. *Image and Vision Computing*, 30(6-7):389–397, 2012.

Ying Yuan, Hongtu Zhu, Weili Lin, and James Stephen Marron. Local polynomial regression for symmetric positive definite matrices. *Journal of the Royal Statistical Society Series B*, 74(4):697–719, 2012.