

A tutorial on persistent homology with GUDHI and giotto-tda

Wojciech Reise

July 6, 2022

The 2nd Young Topologist Seminar, BIMSA



Persistent homology in practice

Persistent homology¹ has witnessed numerous applications, including networks² and time series³.

The growing interest in the use of topological features increased the efforts of the community in high-quality implementations of the persistence algorithm (ripser, giotto-ph, PHAT, perseus, dionysus) and its integration in machine-learning (ML) frameworks (scikit-tda, perslay, persformer...).

In this talk, we will focus on GUDHI and giotto-tda.

¹Frdric Chazal and Bertrand Michel (Feb. 2021). *An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists*. arXiv:1710.04019 [cs, math, stat]. DOI: 10.48550/arXiv.1710.04019. URL: <http://arxiv.org/abs/1710.04019>.

²Mehmet E. Aktas, Esra Akbas, and Ahmed El Fatmaoui (Dec. 2019). "Persistence homology of networks: methods and applications". en. In: *Applied Network Science* 4.1, p. 61. ISSN: 2364-8228. DOI: 10.1007/s41109-019-0179-3. URL: <https://appliednetsci.springeropen.com/articles/10.1007/s41109-019-0179-3>.

³Jose A. Perea (May 2019). "Topological Time Series Analysis". en. In: *Notices of the American Mathematical Society* 66.05, p. 1. ISSN: 0002-9920, 1088-9477. DOI: 10.1090/noti1869. URL: <http://www.ams.org/notices/201905/rnoti-p686.pdf>.

Persistent homology in practice

Persistent homology¹ has witnessed numerous applications, including networks² and time series³.

The growing interest in the use of topological features increased the efforts of the community in high-quality implementations of the persistence algorithm (ripser, giotto-ph, PHAT, perseus, dionysus) and its integration in machine-learning (ML) frameworks (scikit-tda, perslay, persformer...).

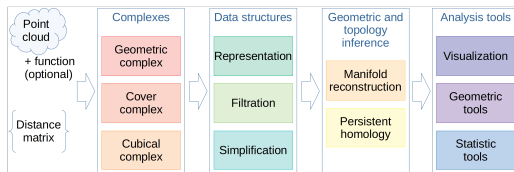
In this talk, we will focus on GUDHI and giotto-tda. The persistent homology interface is only a part of these libraries and both have more to offer!

¹Frdric Chazal and Bertrand Michel (Feb. 2021). *An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists*. arXiv:1710.04019 [cs, math, stat]. DOI: 10.48550/arXiv.1710.04019. URL: <http://arxiv.org/abs/1710.04019>.

²Mehmet E. Aktas, Esra Akbas, and Ahmed El Fatmaoui (Dec. 2019). "Persistence homology of networks: methods and applications". en. In: *Applied Network Science* 4.1, p. 61. ISSN: 2364-8228. DOI: 10.1007/s41109-019-0179-3. URL: <https://appliednetsci.springeropen.com/articles/10.1007/s41109-019-0179-3>.

³Jose A. Perea (May 2019). "Topological Time Series Analysis". en. In: *Notices of the American Mathematical Society* 66.05, p. 1. ISSN: 0002-9920, 1088-9477. DOI: 10.1090/noti1869. URL: <http://www.ams.org/notices/201905/rnoti-p686.pdf>.

GUDHI is a C++ library for Topological Data Analysis (TDA) and for Higher Dimensional Geometry Understanding⁴. It has a Python interface and an interface for R is under development. Some functionalities are available in the TDA package.



Ownership and Licensing

Gudhi is governed by an editorial board, comprised of academic researchers, mostly affiliated with DataShape. The code is available under MIT license to external contributions, but depends on GPLv3 and LGPL modules.

⁴Clement Maria et al. (2014). "The Gudhi Library: Simplicial Complexes and Persistent Homology". en. In: *Mathematical Software ICMS 2014*. Ed. by Hoon Hong and Chee Yap. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 167–174. ISBN: 978-3-662-44199-2. DOI: 10.1007/978-3-662-44199-2_28.

“Giotto-tda is a Python library which integrates high-performance TDA with machine learning via a scikit-learn-compatible API and state-of-the-art C++ implementations.”⁵

- ▶ **Seamless integration with ML frameworks:** notion of a dataset; applying the transformer to a collection of point-clouds/graphs; specific treatment of diagrams by homology dimension.
- ▶ **Code modularity:** topological algorithms as scikit-learn transformers
- ▶ **Standardisation:** inclusion of many popular TDA techniques (recently, edge-collapse⁶ and weightedrips complexes⁷).
- ▶ **Performance:** relying on state-of-the-art implementations of persistent homology, including giotto-ph⁸ for Vietoris–Rips complexes.

For more details, see the documentation⁹ and the presentation of the library¹⁰.

Ownership and Licensing

The code is the result of a collaboration between L2F SA, UPHESS (EPFL) and REDS (HEIG-VD). The package is distributed under the AGPLv3 license.

⁵Guillaume Tauzin et al. (2021). “giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration”. In: *Journal of Machine Learning Research* 22.39, pp. 1–6. URL: <http://jmlr.org/papers/v22/20-325.html>.

⁶Jean-Daniel Boissonnat and Siddharth Pritam (2019). “Edge Collapse and Persistence of Flag Complexes”. en. In: *SoCG’19*, p. 17.

⁷Hirokazu Anai et al. (May 2020). “DTM-based Filtrations”. en. In: *arXiv:1811.04757 [cs, math]*. arXiv: 1811.04757. URL: <http://arxiv.org/abs/1811.04757>.

⁸Julin Burella Prez et al. (July 2021). “Giotto-ph: A Python Library for High-Performance Computation of Persistent Homology of Vietoris–Rips Filtrations”. en. In: *arXiv:2107.05412 [cs]*. arXiv: 2107.05412. URL: <http://arxiv.org/abs/2107.05412>.

⁹<https://giotto-ai.github.io/gtda-docs/>

¹⁰Presentation by Umberto Lupo https://github.com/ulupo/giotto-tda_demo/blob/main/giotto-tda.pdf

Giotto-tda workflow

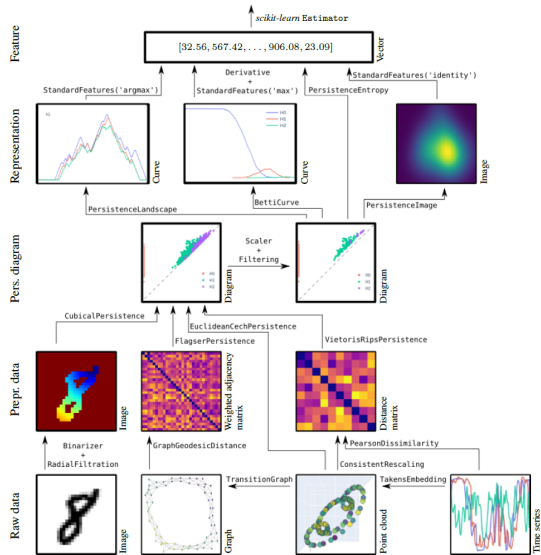


Figure from the documentation.

Shape of an attractor using GUDHI

The `SimplexTree` allows us to define new filtrations of particular complexes. It is based on code from the `velour` package¹¹.

Gravitational waves detection using giotto-tda

The `giotto-tda` interface allows to easily test a novel embedding method¹² in a supervised setting.

The slides and the notebooks are available at
https://github.com/wreise/gudhi_giotto_ph_demo/tree/tys22

¹¹<https://github.com/raphaeltinarrage/velour>

¹²Sean M. Kennedy, John D. Roth, and James W. Scrofani (Sept. 2018). "A Novel Method for Topological Embedding of Time-Series Data". en. In: *2018 26th European Signal Processing Conference (EUSIPCO)*. Rome: IEEE, pp. 2350–2354. ISBN: 978-90-827970-1-5. DOI: 10.23919/EUSIPCO.2018.8553502. URL: <https://ieeexplore.ieee.org/document/8553502/>.