




TDAvec: Computing Vector Summaries of Persistence Diagrams for Topological Data Analysis in R and Python

Umar Islambekov^{1*} and Aleksei Luchinsky^{1*¶}

¹ Bowling Green State University, USA ¶ Corresponding author * These authors contributed equally.

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: Ujjwal Karn 

Reviewers:

- [@adityapt](#)
- [@shravanngoswamii](#)
- [@zplus11](#)

Submitted: 29 April 2025

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/))

Summary

The theory of *persistent homology* is one of the popular tools in *topological data analysis* (TDA) to analyze data with underlying shape structure (Carlsson, 2009; Chazal & Michel, 2021; Edelsbrunner et al., 2002). In this context, a single data observation could be a collection of points lying in a metric space, an image, a graph or a time series. The basic idea behind persistent homology is to build a nested sequence (or *filtration*) of *simplicial complexes* (indexed by a scale parameter) on top of data points and keep a record of the appearance and disappearance of various topological features at different scale values. Here, these topological features are “holes” of different dimensions – connected components, loops, voids, and their higher-dimensional versions whose emergence and subsequent disappearance are tracked using a concept of homology from algebraic topology. From a geometric point of view, simplicial complexes consist of vertices, edges, triangles, tetrahedra etc., glued together and serve as a means for recovering (at least partially) the underlying shape information which is lost during sampling (Nanda & Sazdanovic, 2013).

A topological descriptor outputted by the persistent homology encoding the shape of data is called a *persistence diagram* (PD). Mathematically, a k -dimensional PD is a multi-set of points $D = \{(b_i, d_i)\}_{i=1}^N$, where each point (b_i, d_i) corresponds to a topological feature of homological dimension k (0 if a connected component, 1 if a loop, 2 if a void, etc) with the x -coordinate representing the scale at which this feature is born (or created), and the y -coordinate representing the scale at which it dies (or disappears). In practice, one is usually interested in applying a machine learning method to PDs to make further inferences from data. However, the fact that PDs do not form a Hilbert space, which is a feature (or an input) space for a wide class of machine learning methods, limits their direct of use in applications. To overcome this challenge, kernel methods and vectorization techniques are commonly used (Chung & Lawson, 2022). The kernel approach involves defining a notion of similarity between pairs of PDs, whereas the vectorization methods aim to transform PDs into finite-dimensional feature vectors that can be used as input for many standard machine learning models.

Statement of need

The problem of transforming PDs into finite dimensional vectors for machine learning purposes has attracted considerable attention in the TDA research community over the past decade. Early vector summaries of PDs - such as the persistence landscape (Bubenik, 2015), persistence silhouette (Chazal et al., 2014), Betti curve¹ (Chazal & Michel, 2021), and persistence image (Adams et al., 2017) - have been implemented in both Python and R packages. However, there remains a need for a unified package that brings these methods (both the classical

¹Also known as the Betti function

approaches and those developed more recently) together using consistent syntax and efficient implementation. The TDAvec package, available in both R and Python, is designed to meet this need. Its contributions can be summarized in the following three areas:

1. It extends the list of implemented vector summaries for PDs by incorporating 13 vectorization methods used in TDA. These methods can be grouped into three broad categories:

- Functional vector summaries - based on summary functions:
 - Betti curve (Chazal & Michel, 2021)
 - Euler characteristic curve (Richardson & Werman, 2014)
 - Normalized life curve (Chung & Lawson, 2022)
 - Persistence block (Chan et al., 2022)
 - Persistence surface (Adams et al., 2017)
 - Persistence landscape function (Bubenik, 2015)
 - Persistence silhouette function (Chazal et al., 2014)
 - Persistent entropy summary function (Atienza et al., 2020)
 - Template function (Perea et al., 2023)
- Algebraic vector summaries - based on polynomial maps:
 - Algebraic functions (Adcock et al., 2013)
 - Complex polynomial coefficients (Di Fabio & Ferri, 2015; Ferri & Landi, 1999)
 - Tropical coordinate function (Kališnik, 2019)
- Statistical vector summaries - based on descriptive statistics:
 - Basic descriptive statistics (Ali et al., 2023)

2. A univariate summary function f of a PD is commonly vectorized by evaluating it at a sequence of points on a one-dimensional grid, then organizing the resulting values into a vector:

$$(f(t_1), f(t_2), \dots, f(t_n)) \in \mathbb{R}^n, \quad (1)$$

where t_1, t_2, \dots, t_n form an increasing sequence of scale values. For instance, the `landscape()` and `silhouette()` functions in the TDA (Fasy et al., 2021) package produce such vector summaries for persistence landscapes and silhouettes, respectively. In addition to this standard approach, the TDAvec package introduces an alternative vectorization scheme that captures the average behavior of f between consecutive scale values t_i and t_{i+1} through integration:

$$\left(\frac{1}{\Delta t_1} \int_{t_1}^{t_2} f(t), dt, \frac{1}{\Delta t_2} \int_{t_2}^{t_3} f(t), dt, \dots, \frac{1}{\Delta t_{n-1}} \int_{t_{n-1}}^{t_n} f(t), dt \right) \in \mathbb{R}^{n-1}, \quad (2)$$

where $\Delta t_i = t_{i+1} - t_i$. Unlike the method in (1), this approach retains information about the behavior of f between neighboring scale points. It is applicable to any univariate summary function that is integrable in closed form, such as the persistence silhouette, persistent entropy summary function, Euler characteristic curve, normalized life curve, and Betti function. Users have the flexibility to choose between the two vectorization methods based on their application needs.

3. To achieve higher computational efficiency, all code behind the vector summaries of TDAvec is written in C++ using the Rcpp (Eddelbuettel et al., 2024) and RcppArmadillo (Eddelbuettel & Sanderson, 2014) packages.

The TDAvec R package, along with a vignette demonstrating basic usage and run-time comparisons with other packages, is available on the CRAN repository². For Python examples, we refer the readers to this page.

²<https://cran.r-project.org/web/packages/TDAvec/index.html>

References

- Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., & Ziegelmeier, L. (2017). Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1), 218–252.
- Adcock, A., Carlsson, E., & Carlsson, G. (2013). The ring of algebraic functions on persistence bar codes. *Homology, Homotopy and Applications*, 18. <https://doi.org/10.4310/HHA.2016.v18.n1.a21>
- Ali, D., Asaad, A., Jimenez, M.-J., Nanda, V., Paluzo-Hidalgo, E., & Soriano-Trigueros, M. (2023). A survey of vectorization methods in topological data analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12), 14069–14080. <https://doi.org/10.1109/TPAMI.2023.3308391>
- Atienza, N., Gonzalez-Diaz, R., & Soriano-Trigueros, M. (2020). On the stability of persistent entropy and new summary functions for topological data analysis. *Pattern Recognition*, 107, 107509. <https://doi.org/10.1016/j.patcog.2020.107509>
- Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1), 77–102.
- Carlsson, G. (2009). Topology and data. *Bulletin of the American Mathematical Society*, 46(2). <https://doi.org/10.1090/S0273-0979-09-01249-X>
- Chan, K. C., Islambekov, U., Luchinsky, A., & Sanders, R. (2022). A computationally efficient framework for vector representation of persistence diagrams. *Journal of Machine Learning Research*, 23(268), 1–33.
- Chazal, F., Fasy, B. T., Lecci, F., Rinaldo, A., & Wasserman, L. (2014). Stochastic convergence of persistence landscapes and silhouettes. *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, 474–483. <https://doi.org/10.1145/2582112.2582128>
- Chazal, F., & Michel, B. (2021). An introduction to topological data analysis: Fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4. <https://doi.org/10.3389/frai.2021.667963>
- Chung, Y.-M., & Lawson, A. (2022). Persistence curves: A canonical framework for summarizing persistence diagrams. *Advances in Computational Mathematics*, 48(1), 1–42. <https://doi.org/10.1007/s10444-021-09893-4>
- Di Fabio, B., & Ferri, M. (2015). Comparing persistence diagrams through complex vectors. *Image Analysis and Processing—ICIAP 2015: 18th International Conference, Genoa, Italy, September 7–11, 2015, Proceedings, Part i 18*, 294–305. https://doi.org/10.1007/978-3-319-23231-7_27
- Eddelbuettel, D., Francois, R., Allaire, J., Ushey, K., Kou, Q., Russell, N., Ucar, I., Bates, D., & Chambers, J. (2024). *Rcpp: Seamless r and c++ integration*. <https://doi.org/10.32614/CRAN.package.Rcpp>
- Eddelbuettel, D., & Sanderson, C. (2014). RcppArmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis*, 71, 1054–1063. <https://doi.org/10.1016/j.csda.2013.02.005>
- Edelsbrunner, Letscher, & Zomorodian. (2002). Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4), 511–533. <https://doi.org/10.1007/s00454-002-2885-2>
- Fasy, B. T., Kim, J., Lecci, F., Maria, C., Millman, D. L., & Rouvreau, V. (2021). *TDA: Statistical tools for topological data analysis*. <https://doi.org/10.32614/CRAN.package.TDA>

- 130 Ferri, M., & Landi, C. (1999). Representing size functions by complex polynomials. *Proc.*
131 *Math. Met. In Pattern Recognition*, 9, 16–19.
- 132 Kališnik, S. (2019). Tropical coordinates on the space of persistence barcodes. *Foundations of*
133 *Computational Mathematics*, 19(1), 101–129. <https://doi.org/10.1007/s10208-018-9379-y>
- 134 Nanda, V., & Sazdanovic, R. (2013). Simplicial models and topological inference in biological
135 systems. In *Discrete and topological models in molecular biology* (pp. 109–141). Springer.
136 https://doi.org/10.1007/978-3-642-40193-0_6
- 137 Perea, J. A., Munch, E., & Khasawneh, F. A. (2023). Approximating continuous functions on
138 persistence diagrams using template functions. *Foundations of Computational Mathematics*,
139 23(4), 1215–1272. <https://doi.org/10.1007/s10208-022-09567-7>
- 140 Richardson, E., & Werman, M. (2014). Efficient classification using the euler characteristic.
141 *Pattern Recognition Letters*, 49, 99–106. <https://doi.org/10.1016/j.patrec.2014.07.001>

DRAFT