Robust geometric filtrations for time-series

October 4, 2021

Abstract

The purpose of this project is to implement a geometric filtration (similar to Vietoris-Rips and Čech) that is robust to outliers. For this, we will use the so-called *distance-to-measure*, which acts as a density estimator, and combine it with standard filtrations. This implementation will then be tested against various noise models and on real time series data.

- Download and read this article (section 3):

  `https://drive.google.com/file/d/1a3KXbVi-5MKKaiuozZDtePcZNb-saKVx/view?usp=sharing`

- Using `Gudhi`'s interface, implement an algorithm that computes the DTM-weighted Vietoris-Rips filtration of a point cloud.

- Test your algorithm by running it (using relevant DTM parameters) on point clouds made from sampling a geometric shape (circle, sphere, torus, etc) and adding some outlier noise (using uniform or normal noise models). Compare the output with the standard Vietoris-Rips filtration.

- Implement the Hausdorff distance between point clouds.

- Illustrate the stability theorem (proposition 13) by running your algorithm on several point clouds, and by computing boxplots of the ratio between the corresponding bottleneck distances (for PDs) and Wasserstein (using Python package `POT`) and Hausdorff distances (for point clouds).

- Apply your algorithm on some time-series data from the UCR dataset:

  `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`

  For this, you will have to turn time series into point clouds with time-delay embedding:

  `https://giotto-ai.github.io/gtda-docs/0.3.0/notebooks/time_series_classification.html`

- Classify the time-series using classifier models trained on persistence diagram vectorizations seen in class and compare the performance with standard Vietoris-Rips filtration. If you want to use PersLay, you can either implement it yourself, or compile the `perslay` branch of `Gudhi`:

  `https://github.com/MathieuCarriere/gudhi/tree/perslay`