Giotto-tda – A python™ library for Topological Machine Learning

Umberto Lupo, EPFL
Wojciech Reise, Univ. Paris-Saclay

My awesome collaborators

Giotto-tda


Kathryn Hess


Guillaume Tauzin


Anibal Medina-Mardones


Alberto Dassatti


Julian Burella Pérez


Wojciech Reise


Lewis Tunstall


Matteo Caorsi

EPFL

ReDS
Reconfigurable & embedded Digital Systems

L2F

INAIT

Sponsored by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Innosuisse – Swiss Innovation Agency

... and many **community** contributors!

Topology and the ML workflow

Our objective: Place topological learning algorithms firmly *alongside* established machine learning techniques

ML ethos: Select the best combinations of techniques in a **data-driven** way. The best ones may *include* a number of topological steps as part of a larger **ML pipeline**

Featurization: Produce "**features**" (scalars or vectors) which are amenable to processing by ML algorithms: **set functions**, **explicit vectorisations**, **learned representations**, …

Hyperparameters: Typically, several are involved within each choice of featurization technique (example: pixel size for persistence images)

Large-scale cross-validation routines: Must involve all hyperparameters and model choices at once, topological or not.

giotto-tda: Pillars

Giotto-tda

Seamless integration with widely used ML frameworks: inherit their strengths and allow for creation of heterogeneous ML pipelines. Python + *scikit-learn*

Code modularity: "Lego blocks" approach. Topological algorithms as *scikit-learn transformers*

User-friendliness and **familiarity** to the broad data science community

Standardisation: Allow for integration of most available TDA techniques into a generic framework

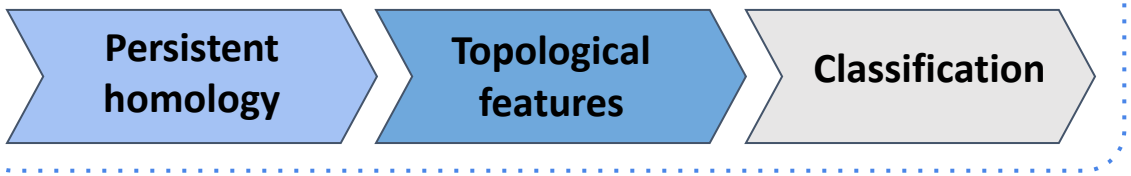
Performance within the language constraints. Parallelism and state-of-the-art C++ code for intensive steps (*ripser**, *GUDHI*, *flagser*, *Hera*, …)

Data structures: Support for **point clouds**, **time series**, **graphs** and **images**

Parallel implementation coming up very soon, stay tuned!

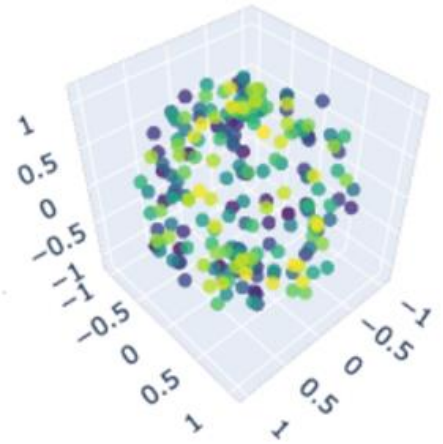
Persistent Homology

Pipeline



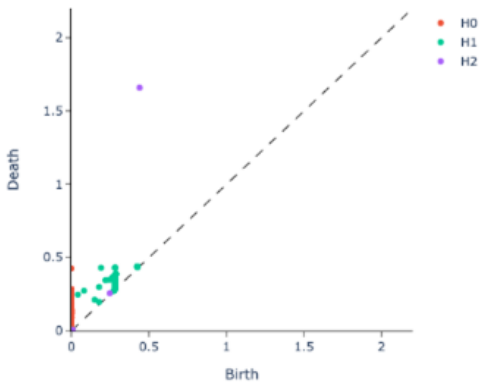

```
# Generate point clouds
clouds, labels = make_point_clouds(n_samples_per_shape=100,
                                   n_points=200,
                                   noise=0.2)

# Split between training set and test set
clouds_train, clouds_test, labels_train, labels_test = \
    train_test_split(clouds, labels)

# Define an end-to-end classification pipeline
persistence_pipeline = make_pipeline(
    VietorisRipsPersistence(),
    PersistenceEntropy(),
    RandomForestClassifier()
)

# Fit the whole pipeline on the training set
persistence_pipeline.fit(clouds_train, labels_train)

# Evaluate the model on the test set
persistence_pipeline.score(clouds_test, labels_test)
```

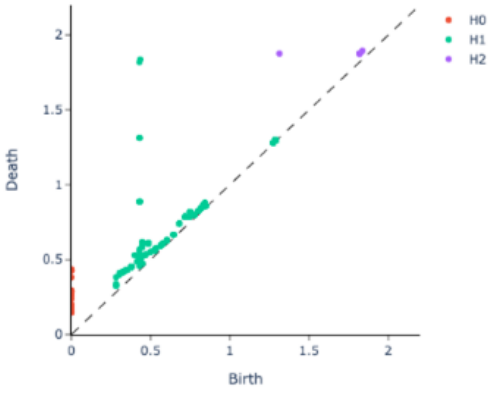
Sampled sphere



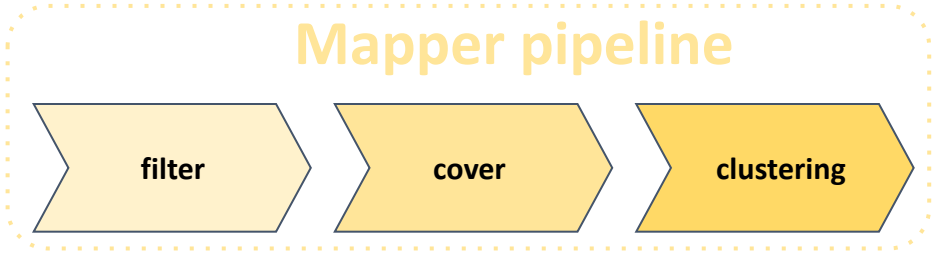
Sampled torus

Persistence diagrams





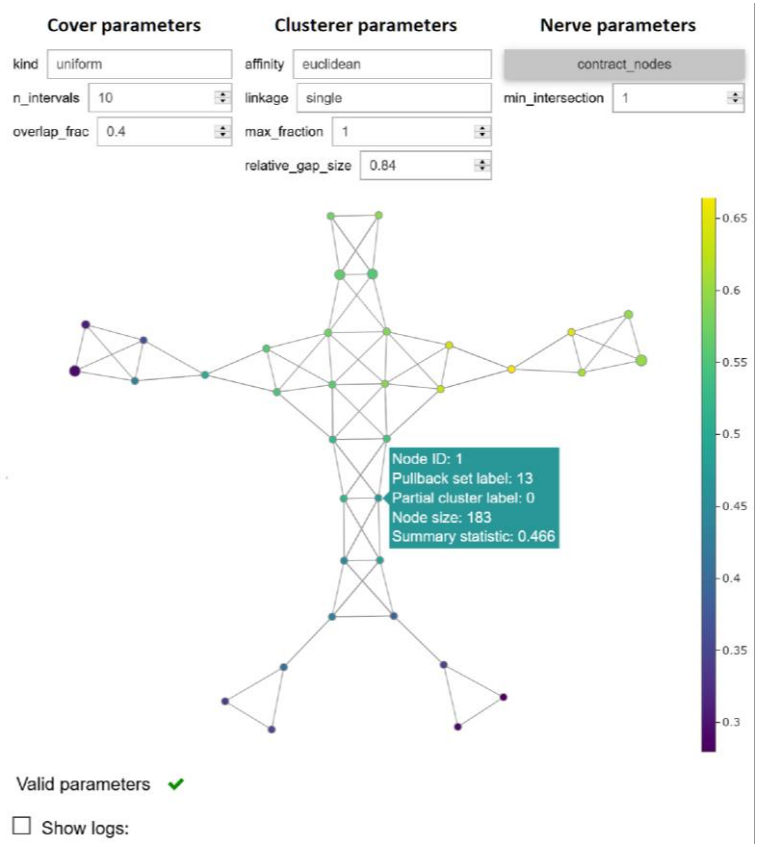
Mapper



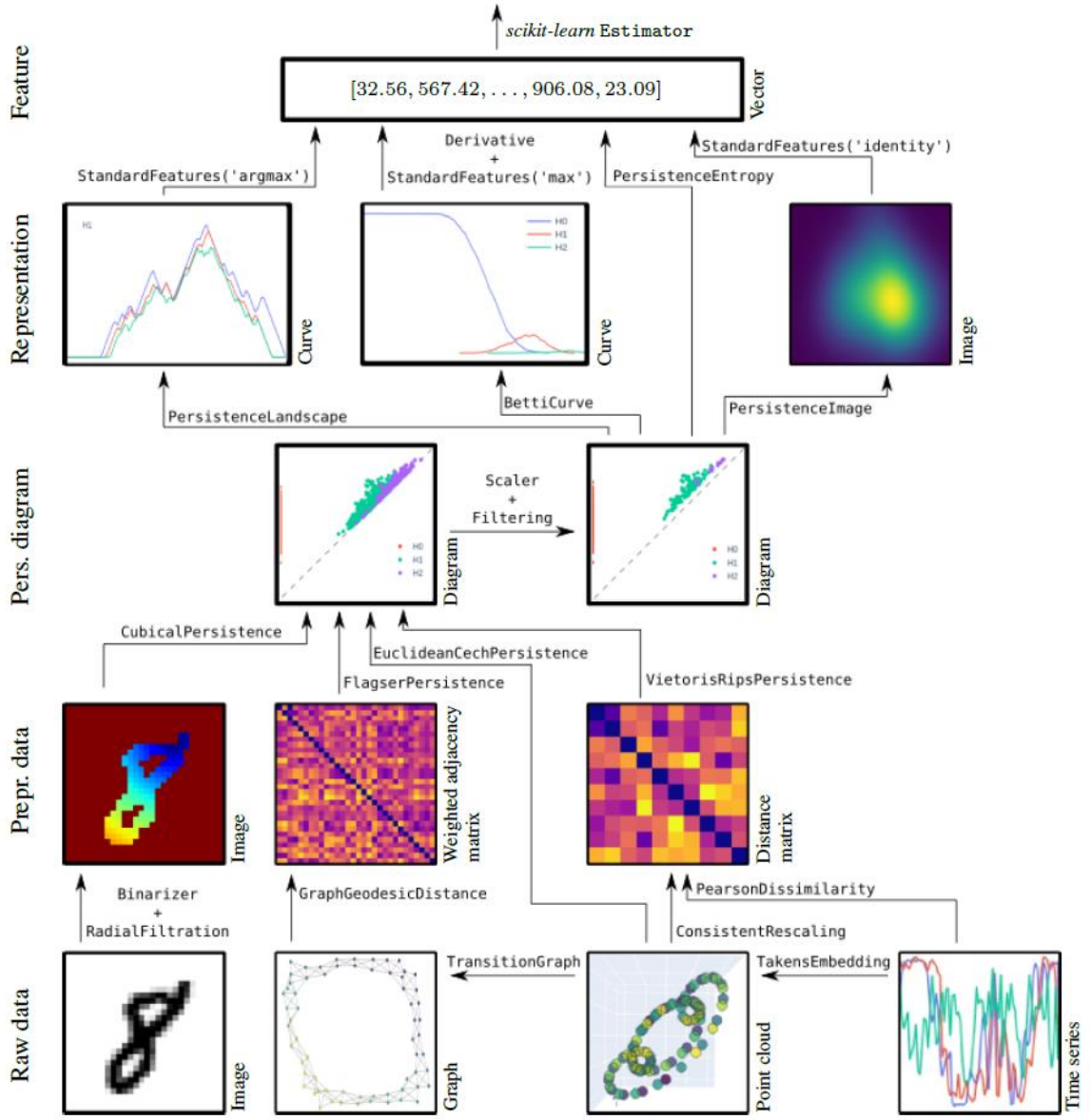

```
# Filter function can be any sklearn Transformer
filter_func = Projection(columns=[0, 1])
# Define cover
cover = CubicalCover()
# Choose clustering algorithm
clusterer = DBSCAN()

# Initialise pipeline
pipe = make_mapper_pipeline(
    filter_func=filter_func,
    cover=cover,
    clusterer=clusterer
)

# Generate interactive plot (Jupyter required)
plot_interactive_mapper_graph(pipe, alien)
```

Point cloud/metric space

Topological summary



“Endless” possibilities



References

Our paper: G. Tauzin et al. ***giotto-tda*: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration** (extended NeurIPS version, JMLR version)

Source on GitHub: github.com/giotto-ai/giotto-tda

Docs: giotto-ai.github.io/gtda-docs

Tutorials & examples: giotto-ai.github.io/gtda-docs/0.5.0/notebooks

API reference: giotto-ai.github.io/gtda-docs/0.5.0/modules

What would you like to do with *giotto-tda*?

Your help is welcome on GitHub: https://github.com/giotto-ai/giotto-tda

We are always **looking to integrate:**

- New algorithmic developments
- More preprocessing techniques
- More kernel and vectorization methods
- … And other important features!

Chat with us!

- Slack: https://slack.giotto.ai/
- GitHub discussions: https://github.com/giotto-ai/giotto-tda/discussions

Today's tutorials

Navigate to https://github.com/ulupo/giotto-tda_demo_POSTECH_MINDS and follow the installation instructions there

Interrupt us & ask questions!

Docs: giotto-ai.github.io/gtda-docs