Introduction to Topological Data Analysis
Part II: Comparing Topological Signatures

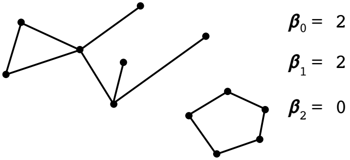
Tom Needham
The Ohio State University

Cőde Bootcamp
The Ohio State University
May 28, 2019

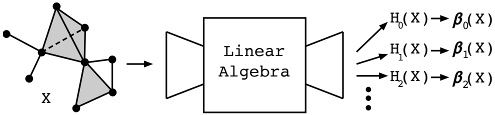
Review: Algebraic Topology

Algebraic topology is a field of math where one computes invariants of a topological space X in order to distinguish it from other spaces.

One family of invariants are the Betti numbers $\beta_k(X)$ which count "k-dimensional holes" in X.

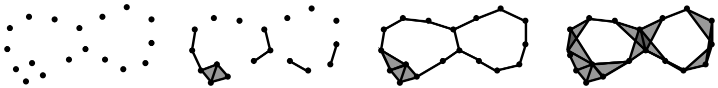


These come from the homology $H_k(X)$ of the space X, which is easily computed if X is a simplicial complex.



Review: Persistent Homology

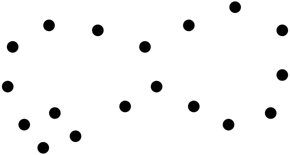
To apply algebraic topology to point cloud data $X = \{\vec{x}_1, \ldots, \vec{x}_N\}$, $\vec{x}_j \in \mathbb{R}^d$, we construct a filtered simplicial complex.

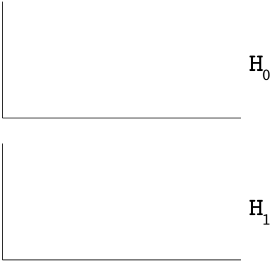


We keep track of "births" and "deaths" of topological features (i.e., "holes" of various dimensions) to produce a barcode or persistence diagram for X.

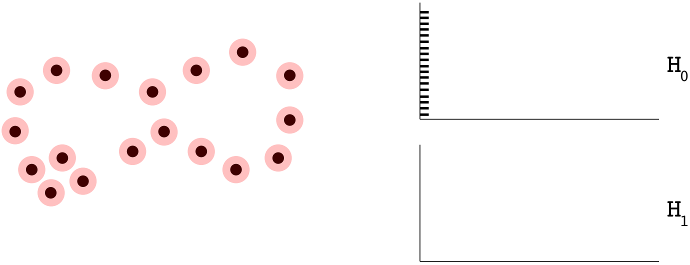
These are representations of the persistent homology of X.

Review: Persistent Homology

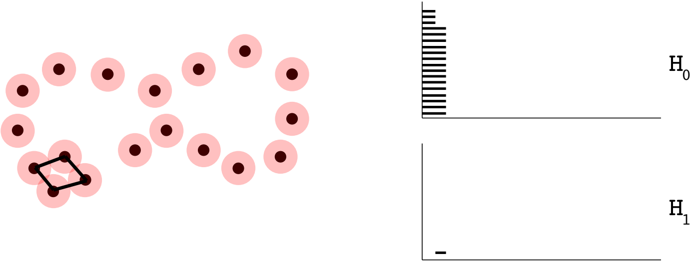




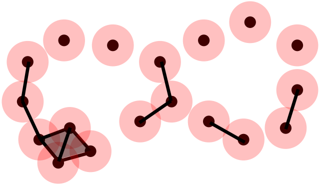
Review: Persistent Homology

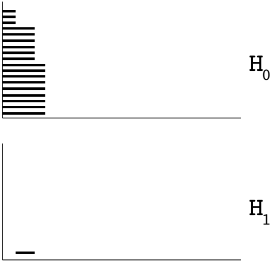


Review: Persistent Homology

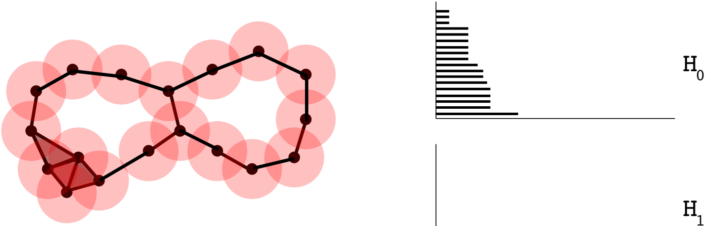


Review: Persistent Homology

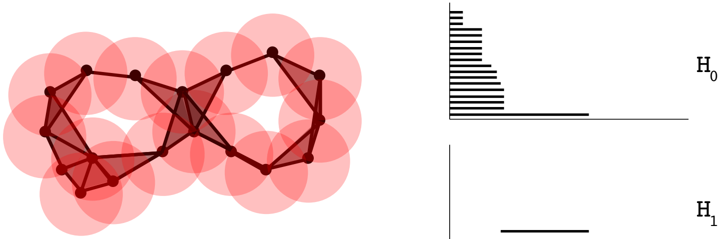




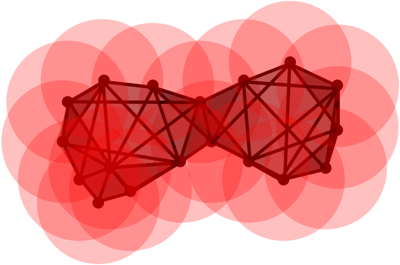
Review: Persistent Homology

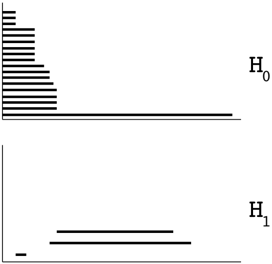


Review: Persistent Homology

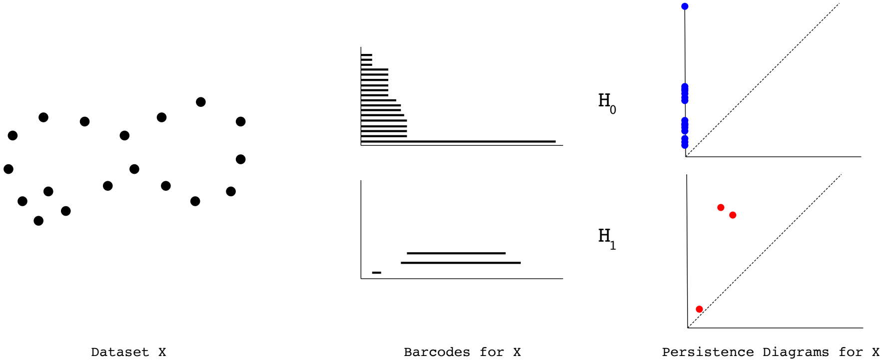


Review: Persistent Homology



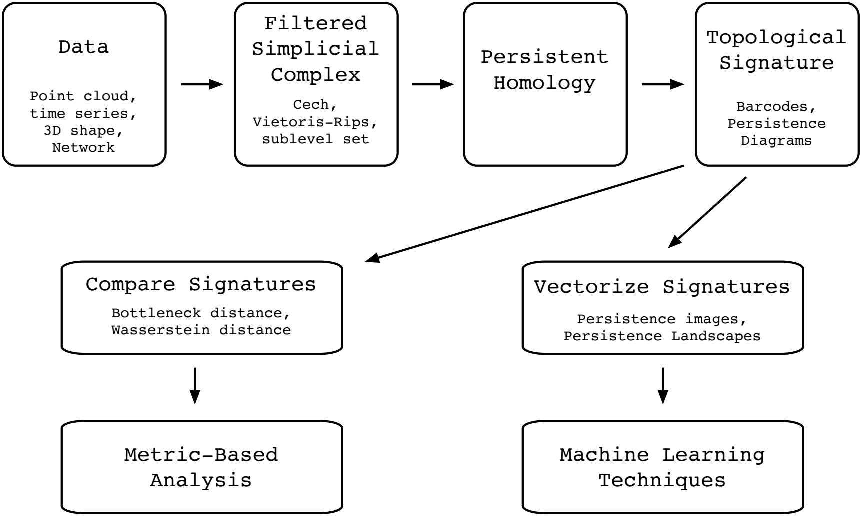


Review: Barcodes and Persistence Diagrams



See Example 1 in accompanying Jupyter notebook.

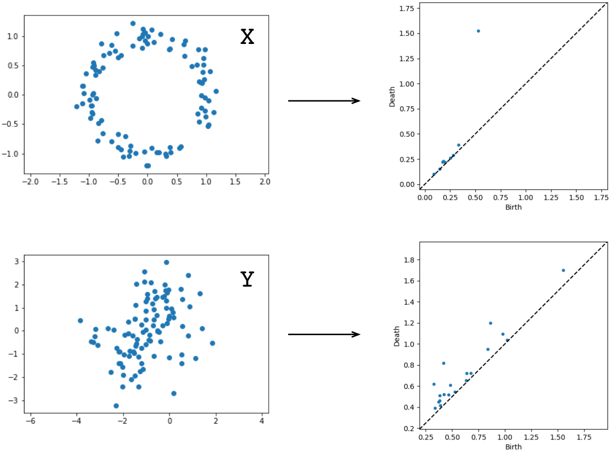
Review: TDA Pipeline



Comparing PDs: Bottleneck Distance

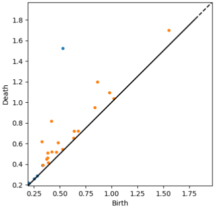
Question

How to we quantitatively compare the PDs for different datasets?



Comparing PDs: Bottleneck Distance

- Plot PDs for X and Y on the same axes.



- Try to "match" points in $PD(X)$ with points in $PD(Y)$.
- For each pair of points $x \in PD(X)$ and $y \in PD(Y)$ that are matched, assign a "cost". For theoretical reasons, the most natural choice is

$$\|x - y\|_\infty = \max\{|x_1 - y_1|, |x_2 - y_2|\}.$$

Comparing PDs: Bottleneck Distance

- We allow any point in $PD(X)$ or $PD(Y)$ to be matched to the nearest point on the diagonal.
- For any choice of matching between the points, we consider the maximum cost incurred for any pair of points.
- The bottleneck distance between $PD(X)$ and $PD(Y)$ is the optimal max cost, where we search over all possible matchings.

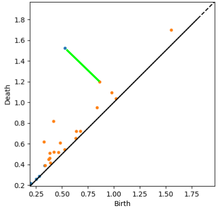


Figure: Optimal matching. Green line indicates max cost for this matching. All other points matched to the diagonal.

See Example 2 in accompanying Jupyter notebook.

Vectorizing PDs: Persistence Images

Problem

Main hurdle to using PDs in standard machine learning pipelines:

- ML algorithms require feature vectors as inputs.
- A PD is an unordered set of points in $\mathbb{R}^2$ — it is not a vector!

Solution

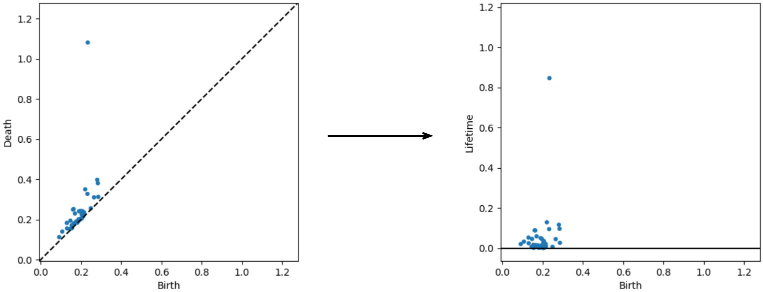
Turn a PD into a vector in some principled way — many options have been proposed.

We will explore persistence images [Adams et. al., 2017].

Given a PD, we construct its persistence image as follows:

Vectorizing PDs: Persistence Images

Transform the PD into the "lifetime" representation
(each point $(b, d) \mapsto (b, d - b)$).



Vectorizing PDs: Persistence Images

- Use each point in the PD as the center of a symmetric Gaussian.
- Take the sum of Gaussians to get a function $f(x, y)$ on $\mathbb{R}^2$.
- Multiply this function by the function $w(x, y) = y$, which weights points farther from the x-axis.

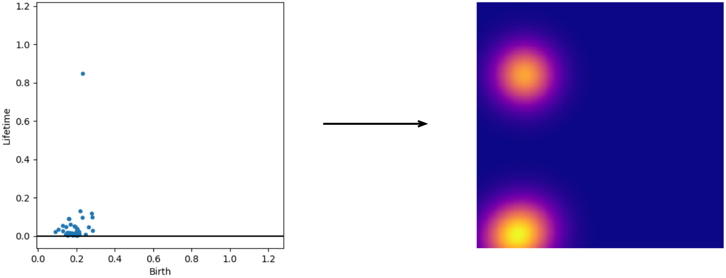
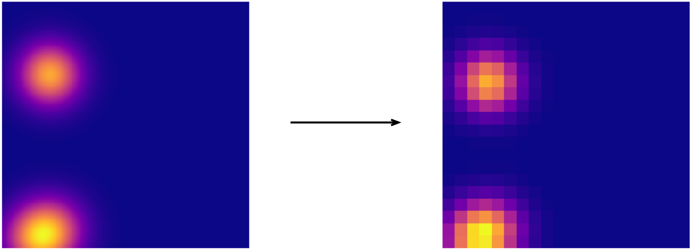


Figure: Visualizing the result as a "heat map".

Vectorizing PDs: Persistence Images

- Pixelate the image: slice the domain into a grid, then take the average of the function over each square.
- Reshaping the result gives a vector!



The result can be used in any ML algorithm.
See Example 3 in accompanying Jupyter notebook.

Conclusions

- TDA is a flexible theory — can give insights not accessible by traditional methods.
- TDA is a very active field — lots to do in theory and applications.
- Some survey articles if you want to read more:
  - *Topological Pattern Recognition of Point Cloud Data* by Gunnar Carlsson
  - *Introduction to Applied Algebraic Topology* — shameless plug for work-in-progress lecture notes available on my website `sites.google.com/site/tneedhammath`.