Introduction to Topological Data Analysis

Persistent Homology

Sheridan B. Green

Yale University

October 22, 2018

Finding Cosmic Voids and Filament Loops Using Topological Data Analysis

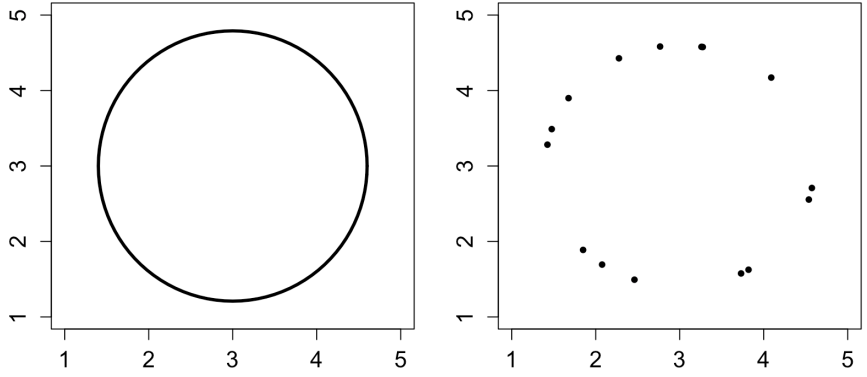
Xin Xu[a], Jessi Cisewski[a,*], Sheridan B. Green[b]

[a] *Department of Statistics and Data Science, Yale University, New Haven, CT 06520, USA*
[b] *Department of Physics, Yale University, New Haven, CT 06520, USA*

Introduction

- **Topological data analysis** (TDA) is a current field of research that has grown out of algebraic topology and computational geometry
- **Persistent homology** is a useful tool from TDA that allows for identifying topological features of a dataset
  - Toplogical features: components, holes, graph structure
  - Dataset: point cloud in a Euclidean/general metric space (just need pairwise distances between the points)
- PH also allows one to place statistical confidence levels on these topological features, enabling one to filter out noise
- PH generalizes clustering algorithms, allowing you to identify clusters (H_0 homology group generators), rings (H_1), voids (H_2) and so on...

Example



(a) True manifold

(b) Sampled manifold

Problem

How do we identify the topological features of the true manifold from the sampled manifold?

How to connect the discrete points?

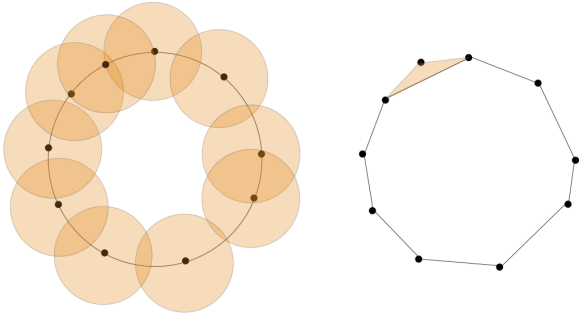


Figure 2: Rips Complex $V_{2d}(\mathbb{X})$

Imagine expanding balls of radius d out from each point $x \in \mathbb{X}$. If the balls surrounding two points overlap, they are connected.
Clearly, the set of points forms a continuous cluster, but how can we identify the hole?

We're going to need simplices to do this...

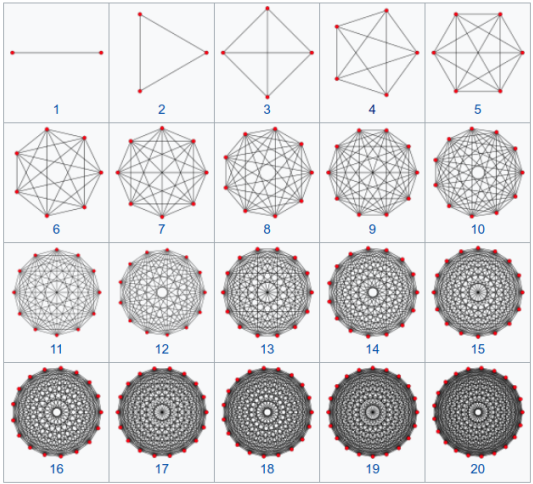


Figure 3: k-simplices

Combinations of simplices form a simplicial complex

- Any combination of simplices (whose intersections are either empty or common faces) forms a **simplicial complex**
- Homology allows us to count the components, holes, voids, etc. from a simplicial complex generated from a dataset (can be done with linear algebra!)
- A hole is an empty 1-cycle, a void is an empty 2-cycle, etc.
- Rips complex in earlier slide is an example of a simplicial complex

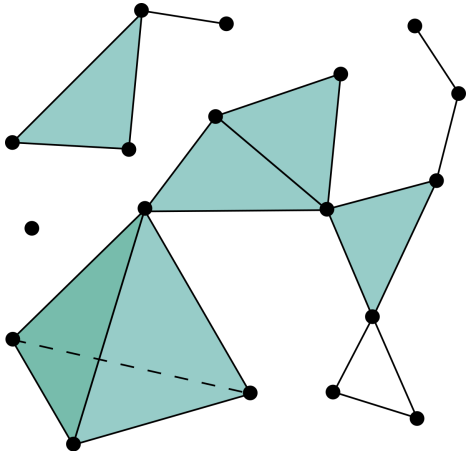


Figure 4: Simplicial complex

Rips complex

Definition (Vietoris-Rips Complex)

If we have a set of points $\mathbb{X}$ in a metric space (M, ρ) of dimension d, and $\mathbb{X} \subseteq \mathbb{R}^d$, then the Vietoris-Rips (VR) complex $V_d(\mathbb{X})$ at scale d (the VR complex over the point cloud $\mathbb{X}$ with parameter d) is defined as:

$$V_d(\mathbb{X}) = \{\sigma \subseteq \mathbb{X} \,|\, d_{\mathbb{X}}(u, v) \leq d, \forall u \neq v \in \sigma\}$$

where σ represents any k-simplex generated from the points $x \in \mathbb{X}$. The number of points $k + 1$ in a simplex determine the dimension (a k-simplex).

How do we choose the right connection distance?

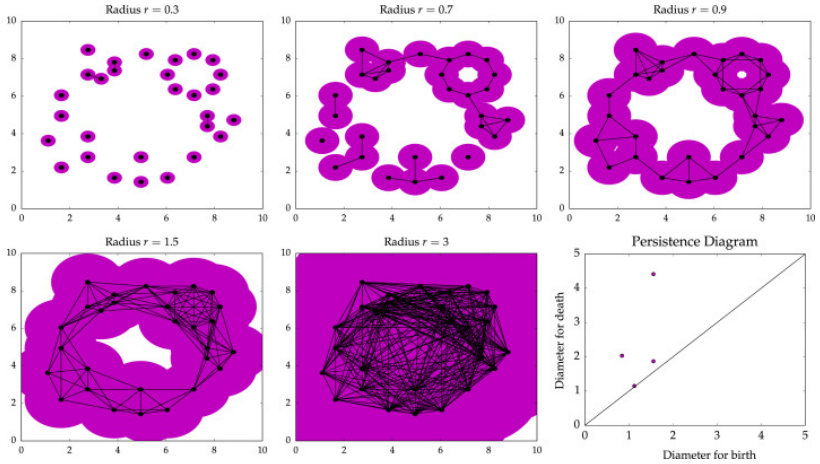


Figure 5: Varying the connectivity distance. Note that the simplicial complex K_1 generated for some $d_1 < d_2$ is a subcomplex of K_2 generated from d_2.

Persistence

- How to choose right d?
- At some connectivity scales, there are clearly holes/voids
- How to discriminate between noise and *real* topological features?
- Imagine expanding balls from $d = 0$ to ∞ and keeping track of the number of connected clusters, holes, voids at each d_i.
- Each topological feature has a "birth time" d_1 and a "death time" d_2, representing when the feature appeared in the simplicial complex and when it was filled in.
- The set of all such birth/death times for topological features in the dataset defines a **barcode**
- The set of nested simplicial complexes for the values of d used is called a **filtration**.

Barcode

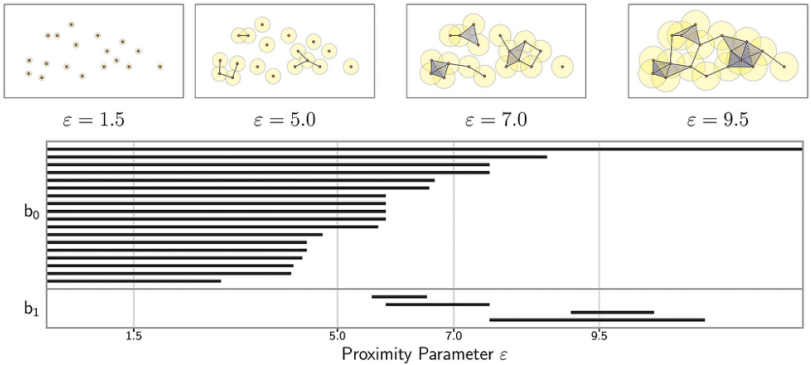


Figure 6: Example barcode from a filtration of Rips complexes. *Interpretation*: short bars represent noise, long bars represent features.

To summarize thus far

- With a dataset and a distance metric, we can construct a filtration of simplicial complexes and identify clusters, holes, voids, etc. in the topology with some notion of *persistence* (i.e. $d_{\text{death}} - d_{\text{birth}}$)
- Can summarize the persistence of all topological features in a dataset with a persistence diagram

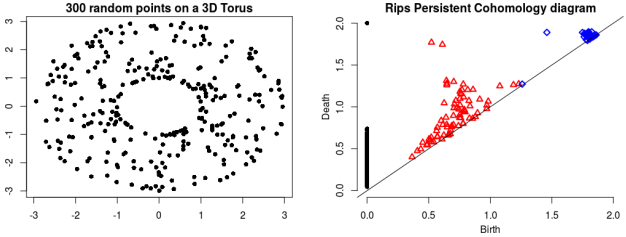


Figure 7: Persistence diagram, different colors for the different order homology groups H_i. When two features merge, the one with an earlier birth time survives and the other dies.

Confidence levels

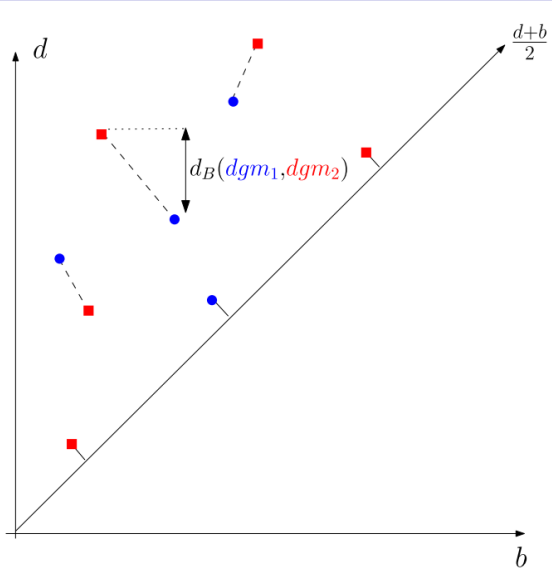
- We would like to know the **Betti numbers** of the topology of the underlying space that the data was sampled from
  - Betti b_0 is number of connected components, b_1 is number of holes, b_2 is number of voids
  - This is a useful topological trace that can be used for applications
- To estimate the Betti numbers, we need to be able to place confidence levels on each individual topological feature in the persistence diagram so that we can only concern ourselves with the most significant features in the dataset (i.e. the ones that "live the longest" in the filtration)
- How to compute confidence levels?

Bottleneck distances

- Persistence diagram (for a particular order of homology groups, say H_0) is a multi-set of $(d_{\text{birth}}, d_{\text{death}})$ tuples combined with the diagonal Δ (where we say there are infinite points)
  - Call this structure $\text{dgm}(\mathbb{X}, d_{\mathbb{X}})$ which depends on the dataset and the distance function/complex type used
- Define a matching between two persistence diagrams dgm_1 and dgm_2 as a subset $m \subseteq \text{dgm}_1 \times \text{dgm}_2$ s.t. all off-diagonal points from the two diagrams appear only once
- For any $p \in \text{dgm}_1 \backslash \Delta$ and $q \in \text{dgm}_2 \backslash \Delta$, the set $(\{p\} \times \text{dgm}_2) \cap m$ and $(\text{dgm}_1 \times \{q\}) \cap m$ contains only one element
- Define the **bottleneck distance** between these two diagrams as:

$$d_B(\text{dgm}_1, \text{dgm}_2) = \inf_{\text{matching} m} \sup_{(p,q) \in m} \|p - q\|_\infty$$

Bottleneck distance cont.



Setting confidence levels

- Let dgm be the persistence diagram of the true data distribution and $\hat{\mathrm{dgm}}$ be the empirical persistence diagram from our dataset $\mathbb{X}$ with some distance function and across some filtration
- For a $1-\alpha$ confidence level, $\exists \eta_\alpha$ s.t.

$$P(d_B(\hat{\mathrm{dgm}}, \mathrm{dgm}) \geq \eta_\alpha) \leq \alpha$$

- Then, any given point in $\hat{\mathrm{dgm}}$ must be less than η_α in L_∞ norm away from a point in true dgm with $1-\alpha$ confidence
  - Center a box of side length $2\eta_\alpha$ around each feature i in $\hat{\mathrm{dgm}}$; if the diagonal does not intersect this box, then feature i can be interpreted as topological signal with significance α
  - Can also place a confidence band of width $\sqrt{2}\eta_\alpha$ out from diagonal s.t. all features outside of this band are α-significant topological features, and the rest are noise
- Every true distribution has its own relationship between α and η_α, so how do we get η_α?

Confidence band

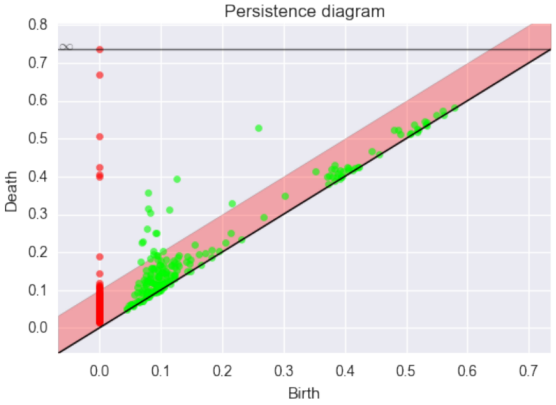


Figure 8: Example confidence band

Setting confidence levels cont.

- Since we don't know the true data distribution, we can estimate the confidence band width η_α by using bootstrap sampling
- For N_{boot} times, select a new subsample $\mathbb{X}^* \subseteq \mathbb{X}$ and compute its persistence diagram $\hat{\text{dgm}}^*$
- Then, compute $d_B(\hat{\text{dgm}}^*, \hat{\text{dgm}})$ for each subsample and generate a distribution of bottleneck distances
- This distribution approximates the distribution of $d_B(\hat{\text{dgm}}, \text{dgm})$ i.e. between the full observed dataset and the true data distribution
- The $1 - \alpha$ quantile of this distribution is an estimate of η_α
- Use this to generate a confidence band on your persistence diagram $\hat{\text{dgm}}$, which we will see in a later slide

Bottleneck distance distribution

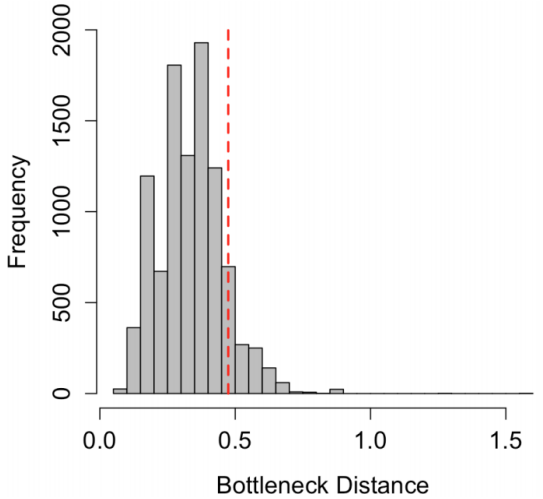


Figure 9: Dashed red line is η_α for $\alpha = 0.1$, i.e. 90% confidence level

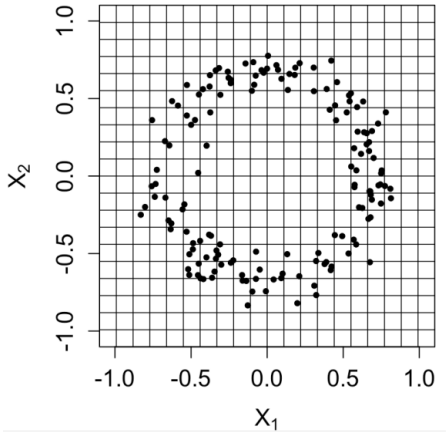
Application to galaxy redshift survey datasets

- Rather than generating Rips complexes using a standard Euclidean distance metric, we introduce a **distance-to-measure** (DTM):

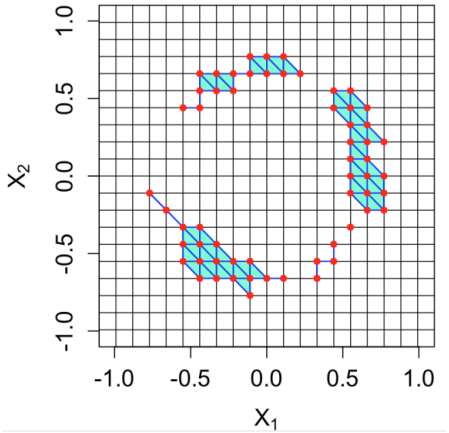
$$d(y \in \mathbb{R}^d) = \left(\frac{1}{k} \sum_{x_i \in N_k(y)} \|x_i - y\|^2 \right)^{1/2}$$

- Here $N_k(y)$ is the set of k nearest neighbor datapoints to the point y in the metric space
- Can think of this DTM as a "root mean square distance to the k nearest neighbors"
- Can then define a filtration on a grid using the lower-level sets of the DTM: $L_t = \{x | d(x) \leq t\}$, i.e. all grid points with DTM below t, where t is called the *threshold*, are in the set and can form the vertices of simplices
- Can loop over values of t and construct a simplicial complex on the grid based on the DTM for each t, as we did before for the Rips complex with the Euclidean distance

Simplicial complex on a grid

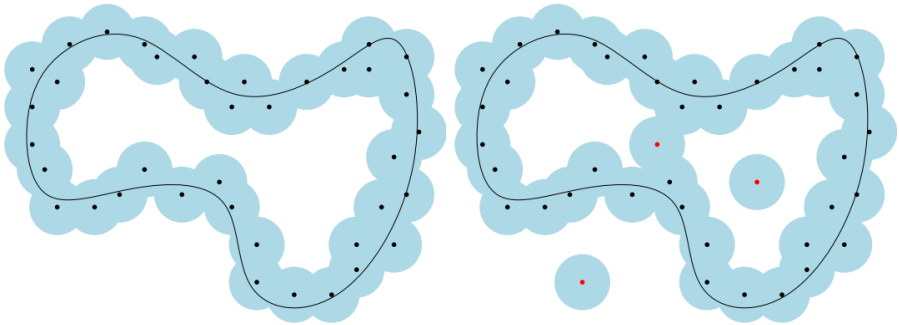


(a) Constructing a grid



(b) A simplicial complex on grid

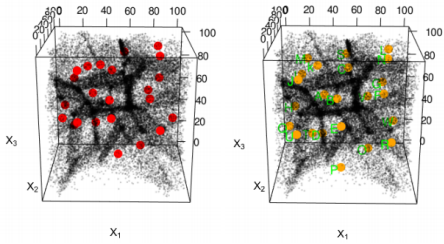
DTM Motivation



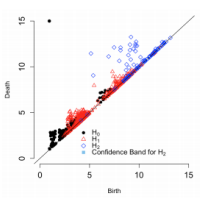
DTM is less sensitive to outliers than the ordinary distance function, since it takes the average over k nearest neighbors; it's also more smoothly varying than a single nearest-neighbor distance

PH applied to Voronoi foam data

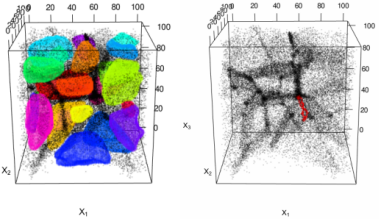
- Generate a set of points in $\mathbb{R}^3$ based on a Voronoi tessellation; looks similar to "cosmic web"
- We know where the "ground truth" voids are, since the tessellation is built around these points
- Allows us to compare the locations of the true voids to the ones identified using persistence homology
  - Requires one to pick a particular representation of the features in the data space... beyond scope of this lecture



(a) Voronoi foam data (b) Centers of mass



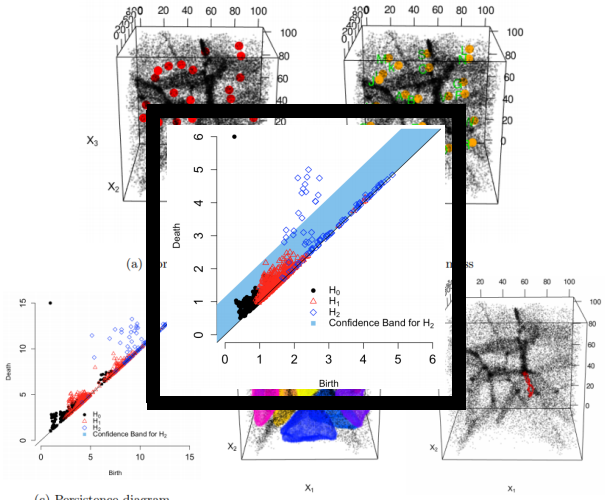
(c) Persistence diagram



(d) Example voids (e) Example filament loop

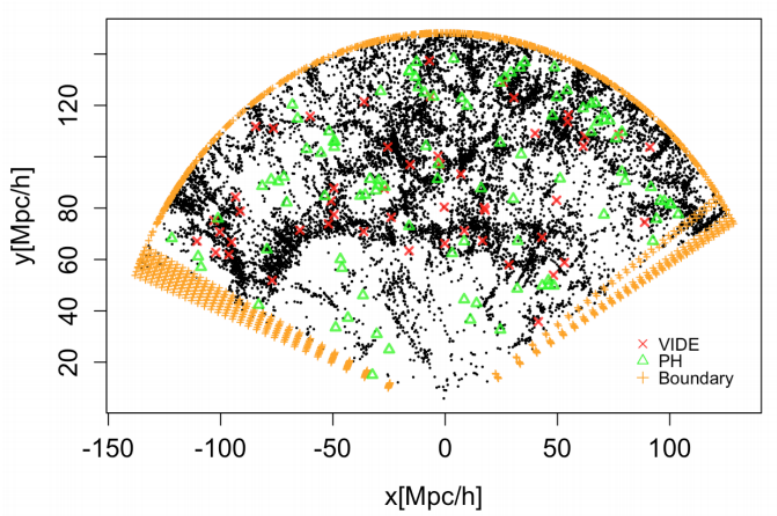
PH applied to Voronoi foam data

- Generate a set of points in $\mathbb{R}^3$ based on a Voronoi tessellation; looks similar to "cosmic web"
- We know where the "ground truth" voids are, since the tessellation is built around these points
- Allows us to compare the locations of the true voids to the ones identified using persistence homology
  - Requires one to pick a particular representation of the features in the data space... beyond scope of this lecture

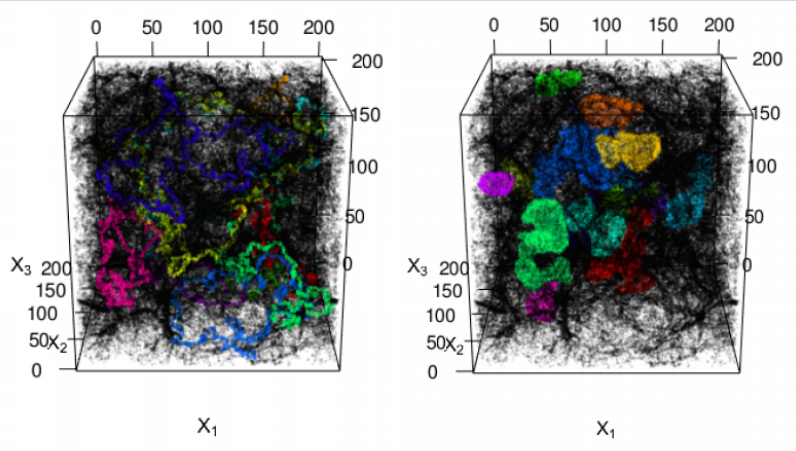


(c) Persistence diagram (d) Example voids (e) Example filament loop

PH applied to SDSS redshift survey



PH applied to cosmological simulation box



(a) Filament loops, H_1

(b) Cosmicvoids, H_2

Figure 10: Most statistically significant voids and loops in the simulation box.

Conclusion

- TDA/persistent homology provide a powerful way to identify topological features in a dataset
- Can be applied to a wide range of problems related to clustering, dimensionality reduction, exploratory data analysis and visualization, etc.
- Has been used extensively in cosmology, biophysics, multivariate time series analysis, 3D shape identification, among others
- Mathematical framework is still under active development (references given in final slide are quite recent, very mathematical statistics-heavy)

Software packages

- `TDA`: R software package for topological data analysis
- `GUDHI`: Geometry understanding in higher dimensions; C++ and Python package for computing Ĉech/Rips complexes, generating persistence diagrams, bottleneck distances, etc.
- `Dionysus`: C++ library for persistent homology calculations

References

- Algebraic Topology, Allen Hatcher
- Edelsbrunner et al. (2002): Topological Persistence and Simplification
- Chazal and Michel (2017): An Introduction to Topological Data Analysis
- Matthew Wright: Introduction to Persistent Homology
- TDA Part 1: Persistent Homology
- A User's Guide to Topological Data Analysis
- Figures came either from my paper, Chazal and Michel (2017), Wikipedia entries, documentation for one of the software packages, or a few from the references above