# A Tutorial in Topological Data Analysis

## Fall 2018 MAA NCS


Matthew Zabka

Mathematics and Computer Science
Southwest Minnesota State University


2018-10-13

Please clone or download the following folder:

https://github.com/MatthewZabka/MAA-NCS18

# Overview

# A short review

As we saw yesterday:

- Topology provides many descriptors about the 'shape' of a space.
- One of these descriptors – homology – is computable, which makes it particularly useful in applications.
- Given a topological space $T$ the $n$-th homology group $H_n(T)$ – in very basic terms – has a generator for each $n$-dimensional hole in $T$.
- So the rank of the $n$-th homology group, called the $n$-th **Betti number**, denoted

  $$\beta_n := \mathrm{rank}(H_n(T)),$$

  is the number of $n$-dimensional holes in $T$.
- $\beta_1$ counts the number of holes in $T$.
- $\beta_2$ counts the number of voids in $T$ ...

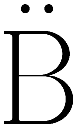
# A short review

- The $n$-th Betti number tells us about the number of $n$-dimensional holes in a space.
  - What is a 0-th dimensional hole?
  - $\beta_0$ counts the number of connected components in a space.
  - What is $\beta_0$ of the following space?

  - $\beta_0 = 2$

# A short review

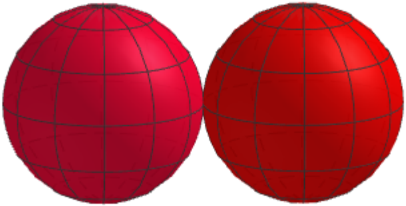
- The $n$-th Betti number tells us about the number of $n$-dimensional holes in a space.
  - $\beta_1$ counts the number of ‘holes’ in a space.
  - You can think of a hole as something through which you can stick you arm.
  - What are $\beta_0$ and $\beta_1$ of the following space?



  - $\beta_0 = 3$ and $\beta_1 = 2$

# A short review

- The $n$-th Betti number tells us about the number of $n$-dimensional holes in a space.
  - $\beta_2$ counts the number of voids.
  - What is $\beta_2$ of the following space?



  - $\beta_2 = 2$
  - What are $\beta_0$ and $\beta_1$?
  - $\beta_0 = 1$ and $\beta_1 = 0$.

- Recall: Given a set of points $X$ and a parameter $r$, we can create a formal simplicial complex.
- Let the $n$-simplex $\{x_0, x_1, \ldots, x_n\}$ exist if and only if $d(x_i, x_j) < r$ for all $0 \leq i, j \leq n$.
- We can visualize this process using InteractiveJPDwB.

# InteractiveJPDwB

- InteractiveJPDwB [?] lets one visualize the generated simplicial complex for data in $\mathbb{R}^2$ and different values of $r$.
- In other words, it demonstrates 0-th and 1-st degree persistence homology via barcodes.
- Thanks to Michael Catanzaro, you can easily install this program onto your computer.
- You can download this program from:

  https://github.com/MatthewZabka/MAA-NCS18

# Discuss with your groupmates!

https://github.com/MatthewZabka/MAA-NCS18

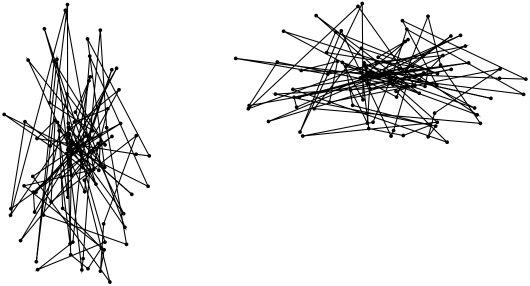
- What is the minimum number of points required so that $\beta_1 = 1$ for some value of $r$?
- What is the minimum number of points required so that, for some $r$, we have $\beta_0 = 3$ and $\beta_1 = 2$?
- What is the largest degree of homology that is geometrically feasible in $\mathbb{R}^2$?
- What is the minimum number of points required to have $\beta_2 = 1$? In what dimension must the points lie?
- What is the minimum number of points required to have $\beta_n = 1$? In what dimension must the points lie?

# Discussion

# Persistent Homology in Dimension 0 (Clustering)

- Start with a set of data in a metric space. Set $r = 0$.
- Increase $r$. Create an edge between two points whenever the distance between them is less than $r$. This creates a graph.
- The graph defines a simplicial complex via Vietoris-Rips.
- If a topological property (like a Betti number) *persist* over a large range of $r$, we can conclude something about the structure of the data.
- In this case, we should expect to see $\beta_0 = 2$ over a large range of $r$.

# Persistent Homology in Dimension 0 (Clustering)



- A graph similar to this should persist over a large range of $r$.
- How could we see the clusters if these data did not lie in $\mathbb{R}^2$?

# RIPSER

- There are several programs that do persistence.
- We shall first look at RIPSER[**?**].
- Developed by Ulrich Bauer, RIPSER is a very fast C++ program for computing Vietoris-Rips persistence barcodes.
- Let's try this on our data with two clusters.

# RIPSER

- Go to github to get the data. These data are stored as a point cloud.

  https://github.com/MatthewZabka/MAA-NCS18

- Input `data1.txt` into RIPSER:

  https://live.ripser.org/

- Suppose we have data that lie on the circle.

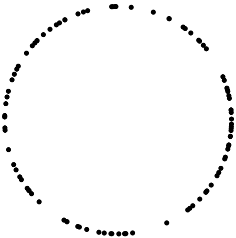


Figure: An unrealistic example `data2.txt`, where data lie perfectly on $S^1$.

- Input `data2.txt` into RIPSER.
- Wait ...data are never this nice.

# Persistent Homology

- Suppose we have data that **almost** lie on the circle.

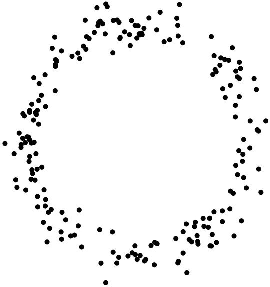


Figure: A slightly more realistic example `data3.txt`.

- Why is this also not realistic or impressive?

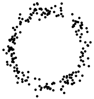


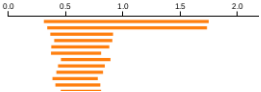
Figure: `data3.txt`.

- Now it is your turn!
- Input `data3.txt` into RIPSER. Confirm that one generator for $H_0$ and one generator for $H_1$ persist.
- Try inputting `data4.txt` into RIPSER up to distance 2. What can you say about the data's shape?
- Try some real data! `data5.txt` (up to distance 150) and `data6.txt` (up to distance 2)
- Data are located in the `data` folder that you have already downloaded!

# Discussion

# Combining Persistence and Other Methods

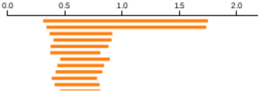
- Suppose we had a barcode in dimension 1 that looked as follows:




- What are are possibilities for the manifold on which the data lie?

# Combining Persistence and Other Methods

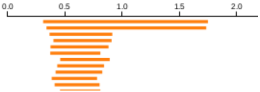
- Suppose we had a barcode in dimension 1 that looked as follows:



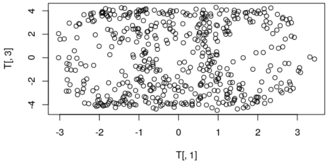

- What are are possibilities for the manifold on which the data lie?

# Combining Persistence and Other Methods

- Suppose we had a barcode in dimension 1 that looked as follows:

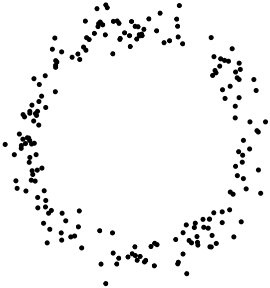



- Suppose we perform PCA and get the following projection




- What does PCA suggest about the dimension on the manifold?
- What does this suggest about the space on which the data lie?

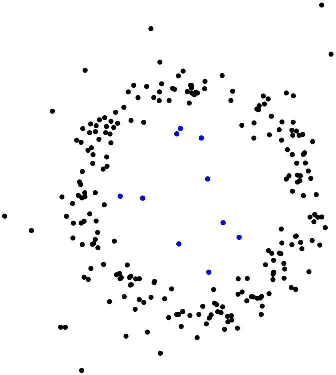
- Let's think about how this could go wrong and how we could fix it!
- Suppose that, instead of data that look like this:



- Let's think about how this could go wrong and how we could fix it!
- We had data that looked like this:

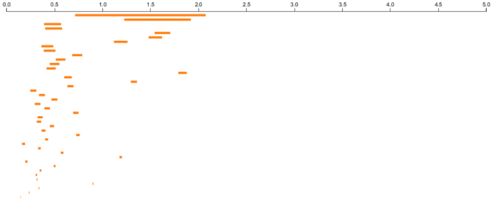
# RIVET

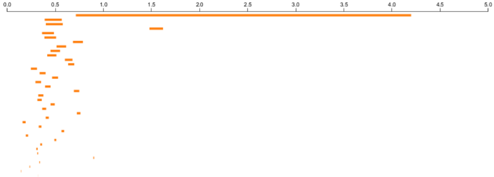
- This is a much more realistic example.
- The blue points – noise – will make it hard to see the generator of $H_1$.



# RIVET

The first barcode includes the entire data set. The second barcode eliminates the blue 'noise'.



# RIVET

- One way to deal with such situations: consider the density of the points.

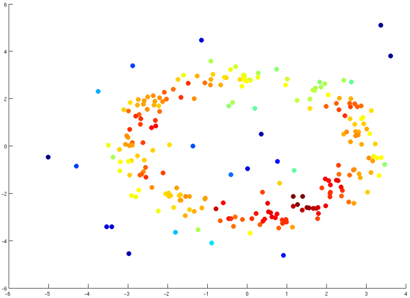



Figure: A density heat map of the data

- Letting the density threshold vary results in **multi-parameter persistence**!

- Michael Lesnick and Matthew Wright have written a program for visualizing multiparameter persistence.[?]
- Installation is not so simple – let us go through it together!