An Introduction to Topological Data Analysis

Matthew Zabka

Laber Labs
North Carolina State University

Southwest Minnesota State University

Outline

Crash course in algebraic topology

Persistent Homology and examples

JPDwB Tutorial

Persistence Homology and RIPSER

Multiparameter Persistence and RIVET

What is algebraic topology?

Topology is a branch of mathematics which is good at extracting global qualitative features from complicated geometric structures.

Algebraic topology provides a set of *algebraic* descriptors to topological objects.

What is algebraic topology?

Topology is a branch of mathematics which is good at extracting global qualitative features from complicated geometric structures.

Algebraic topology provides a set of *algebraic* descriptors to topological objects.

Questions and scope

Topological questions surround different notions of connectedness: connected components, loops, voids, etc.

Two topological spaces are equivalent through the lens of topology if one can be *continuously* deformed to the other.

A B C D

Algebraic Topology

What is algebraic topology?

Topology is a branch of mathematics which is good at extracting global qualitative features from complicated geometric structures.

Algebraic topology provides a set of *algebraic* descriptors to topological objects.

Questions and scope

Topological questions surround different notions of connectedness: connected components, loops, voids, etc.

Two topological spaces are equivalent through the lens of topology if one can be *continuously* deformed to the other.

A B C D

Invariants of topological spaces

- Algebraic Topology assigns *invariants* to topological spaces. These take the form of groups, rings, fields, vector spaces, etc.
- Our computations will be over the field $\mathbb{F}_2$, so it suffices to record only the *dimensions* of vector spaces.
- If two spaces are the same, then the invariants must be the same.
  If the invariants are not the same, then the two spaces are not the same.

Topological Data Analysis

Goal

Topological data analysis uses topology to summarize and study the 'shape' of data.

Topological Data Analysis

Goal

Topological data analysis uses topology to summarize and study the 'shape' of data.

Examples:

- motion tracking problems,
- analysis of brain arteries,
- analysis of social and spatial networks, including neuronal networks, Twitter, co-authorship,
- study of viral evolution,
- measurement of protein compressibility,
- analysis of phase transitions,
- financial crash analysis,
- piecewise constant signal analysis,
- study of cosmic web and its filamentary structure,
- identification of breast cancer subtypes,
- study of plant root systems,
- discrimination of EEG signals before and during epileptic seizures,
- steganalysis of images,
- sphere packings,
- population activity in the visual cortex,
- fMRI data

Overview of TDA

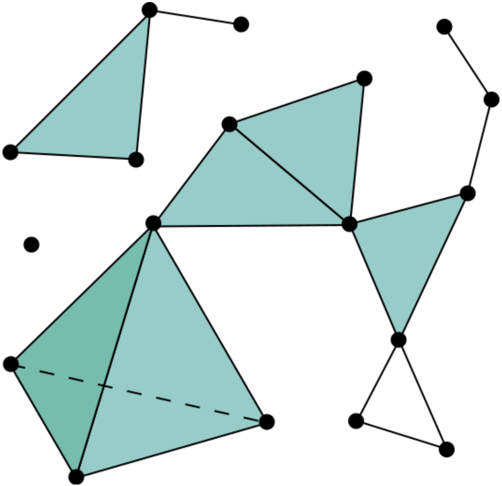
Topological data analysis comes in a variety of flavors.

The two most popular methods in TDA are:

1. Persistent Homology
2. Mapper

Simplicial Complexes

A *simplicial complex* is a combinatorial object, generalizing the notion of a graph.
Each simplicial complex is built out of *simplices* of varying dimensions.

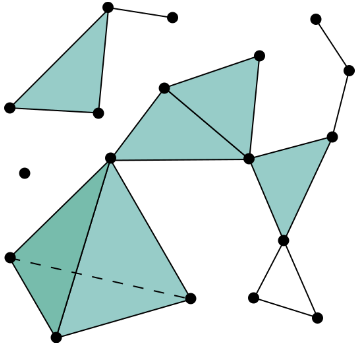


Betti numbers of simplicial complexes

$\beta_0 = \#$ of connected components

$\beta_1 = \#$ of holes

$\beta_2 = \#$ of voids



$\beta_0 \quad =$

$\beta_1 \quad =$

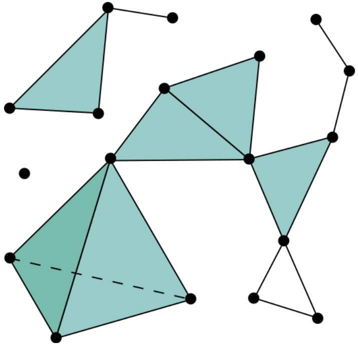
$\beta_2 \quad =$

Betti numbers of simplicial complexes

$\beta_0 = \#$ of connected components

$\beta_1 = \#$ of holes

$\beta_2 = \#$ of voids

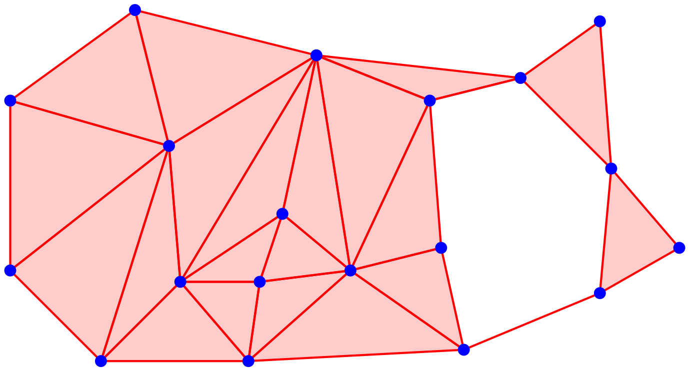


$$\beta_0 = 3$$
$$\beta_1 = 1$$
$$\beta_2 = 1$$

Homology of simplicial complexes

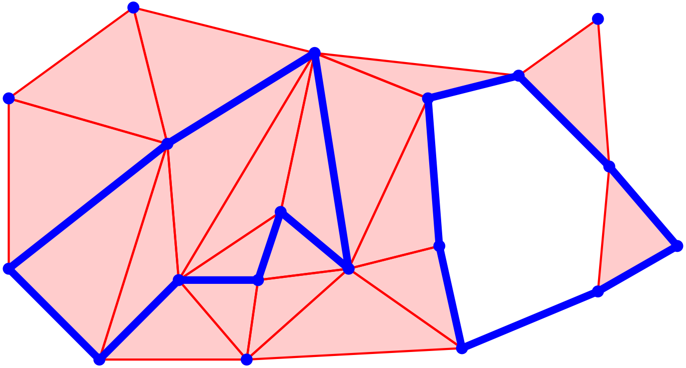
Definition

Homology in degree k is given by k-cycles modulo the k-boundaries.



Definition

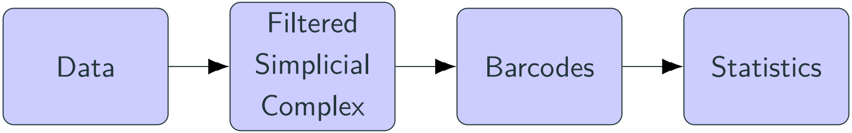
Homology in degree k is given by k-cycles modulo the k-boundaries.



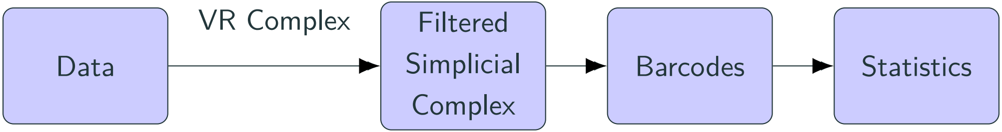
β_k = rank of homology in degree k

Overview of PH

Persistent homology consists of the following pipeline:



Overview of PH



Simplicial Complexes from Point data

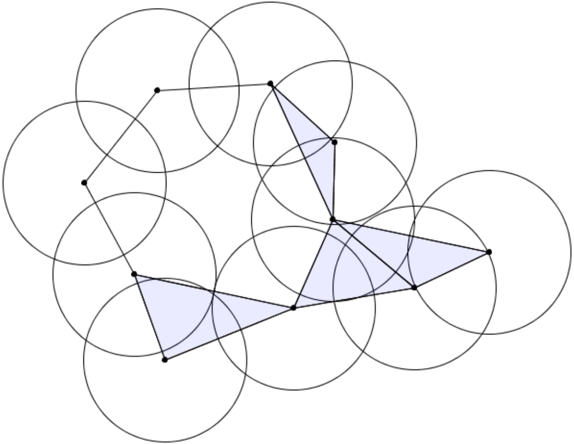
Definition

A *point cloud* P is a finite metric space.

Simplicial Complexes from Point data

Definition

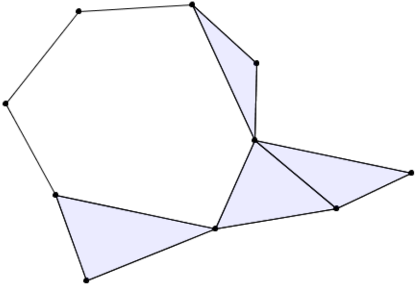
The Vietoris-Rips complex is a simplicial complex built out of a point cloud. Put a circle of radius r around each point. Add an edge whenever two circles overlap. Add a triangle whenever three circles overlap.



Simplicial Complexes from Point data

Definition

The Vietoris-Rips complex is a simplicial complex built out of a point cloud. Put a circle of radius r around each point. Add an edge whenever two circles overlap. Add a triangle whenever three circles overlap.

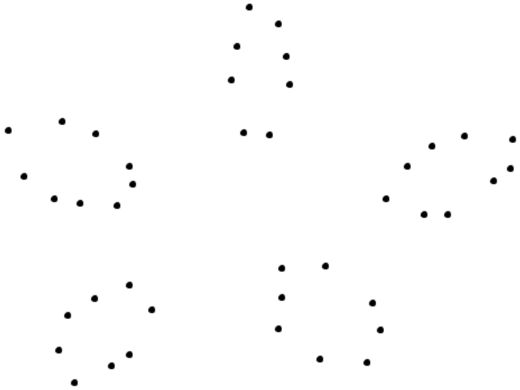


Vietoris-Rips parameter

Question

How do we choose the correct radius for the Vietoris-Rips construction?

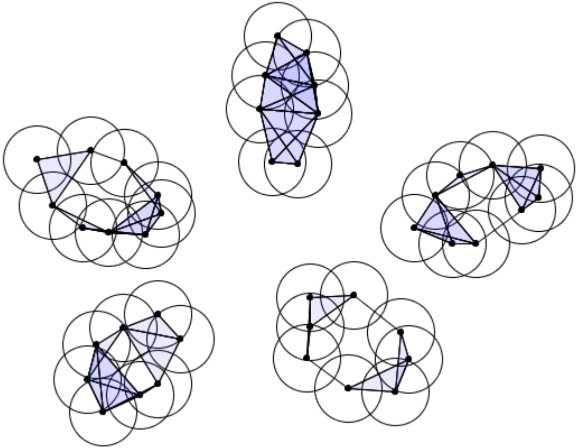
Often, there is no one "right" choice.



Question

How do we choose the correct radius for the Vietoris-Rips construction?

Often, there is no one “right” choice.

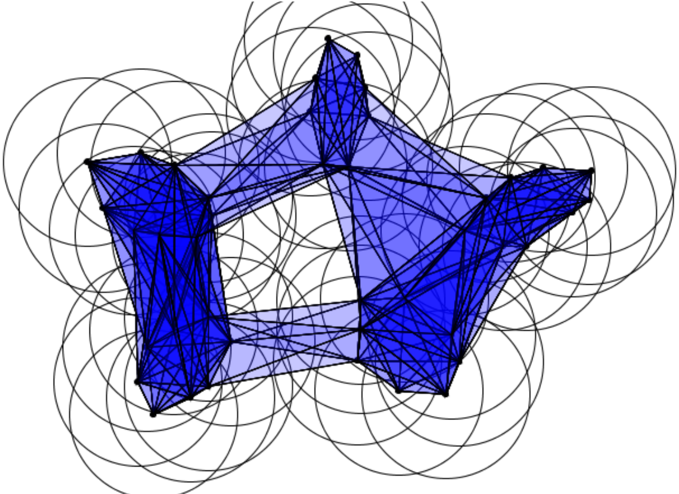


Vietoris-Rips parameter

Question

How do we choose the correct radius for the Vietoris-Rips construction?

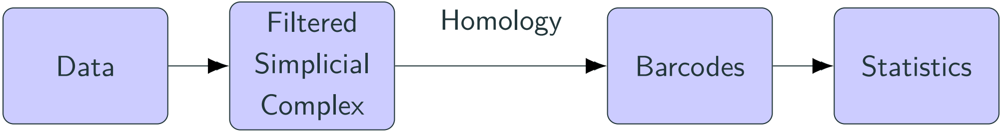
Often, there is no one “right” choice.



Processing demo

Processing demo

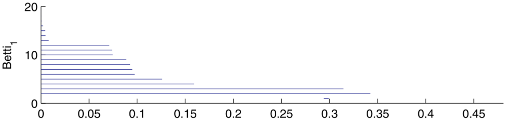
Overview of PH

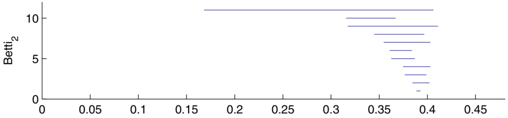


Barcodes

- The barcode provides a summary of how the homology changes as the radius varies in the Vietoris-Rips construction.
- We look for topological features which ‘persist’ over many values of radii.

Barcodes typically look like:





InteractiveJPDwB

- 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 use this program on your own computer.
- You can download this program from:

https://github.com/MatthewZabka/LaberLabs18

Anwer the following!

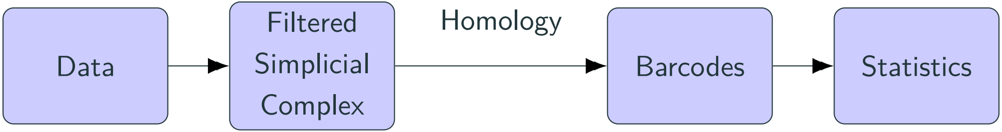
https://github.com/MatthewZabka/LaberLabs18

- 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

- 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?

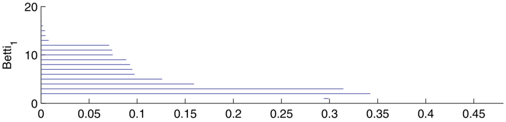
Overview of Persistence Homology

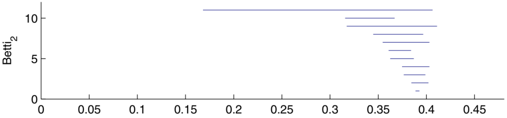


Barcodes

- The barcode provides a summary of how the homology changes as the radius varies in the Vietoris-Rips construction.
- We look for topological features that ‘persist’ over many values of radii.

Barcodes typically look like:





Persistent Homology in Dimension 0 (Clustering)

- Start with a set of data in a metric space. Set $r = 0$.

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.

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.

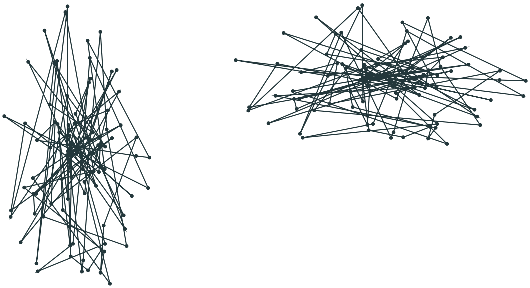
Persistent Homology in Dimension 0 (Clustering)

- 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.

Persistent Homology in Dimension 0 (Clustering)

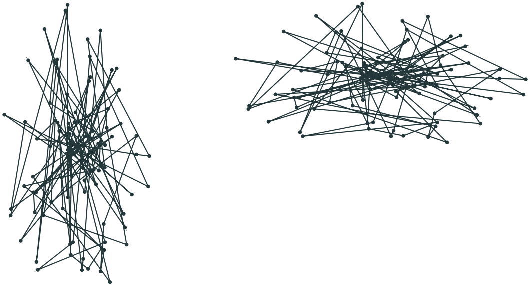
- Start with a set of data in a metric space. Set $r = 0$.
- Increase r. Create an edge between two points whenever the distance between them is less than r. This creates a graph.
- The graph defines a simplicial complex via Vietoris-Rips.
- If a topological property (like a Betti number) *persist* over 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.

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 $\mathbb{R}^2$?

- There are several programs that do persistence.

RIPSER

- There are several programs that do persistence.
- We shall first look at RIPSER.

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.

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.

- You should already have these data – stored as a point cloud.

https://github.com/MatthewZabka/LaberLabs18

RIPSER

- You should already have these data – stored as a point cloud.

https://github.com/MatthewZabka/LaberLabs18

- Input `cloud1.txt` into RIPSER:

https://live.ripser.org/

- Suppose we have data that lie on the circle.

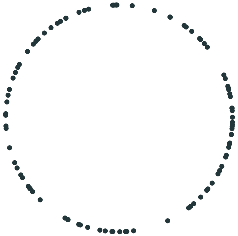


Figure 1: An unrealistic example `cloud2.txt`, where data lie perfectly on S^1.

- Suppose we have data that lie on the circle.

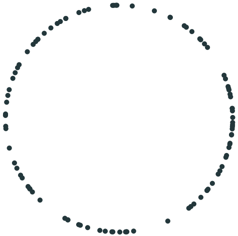


Figure 1: An unrealistic example `cloud2.txt`, where data lie perfectly on S^1.

- Input `cloud2.txt` into RIPSER.

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

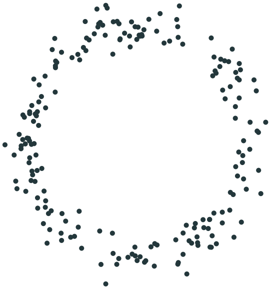


Figure 2: A slightly more realistic example `cloud3.txt`.

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

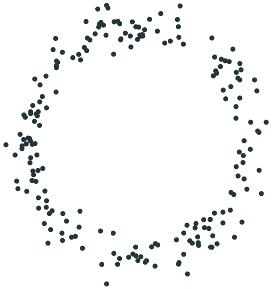


Figure 2: A slightly more realistic example `cloud3.txt`.

- We want to analyze data we cannot see!

https://github.com/MatthewZabka/LaberLabs18

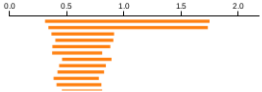
- Now it is your turn! Data are located in the `RIPSERdata` folder that you have already downloaded!
- Input `cloud1.txt` int RIPSER. Confirm that two generators of H_0 persist. (i.e. $\beta_0 = 2$)
- Input `cloud2.txt` into RIPSER. Confirm that one generator for H_0 and one generator for H_1 persist. (i.e. $\beta_0 = 1$ and $\beta_1 = 1$)
- Input `cloud3.txt` into RIPSER. Confirm that one generator for H_0 and one generator for H_1 persist. (i.e. $\beta_0 = 1$ and $\beta_1 = 1$)
- Try inputting `cloud4.txt` into RIPSER up to distance 2. What can you say about the data's shape?
- Try some actual data! `cloud5.txt` (Test up to distance 150.)
- More actual data! `cloud6.txt` (Test up to distance 2.)

Discussion

- What is the shape of `cloud4.txt`?
- What is the shape of `cloud5.txt`?
- What is the shape of `cloud6.txt`?

Combining Persistence and Other Methods

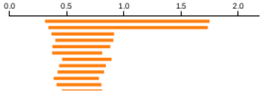
- Suppose we had a barcode in dimension 1 that looked as follows:



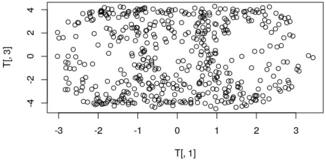
- What are are possibilities for the manifold on which the data lie?

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:



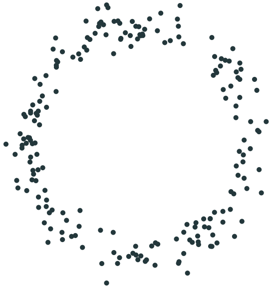
- What does PCA suggest about the dimension of the manifold?
- What does this suggest about the space on which the data lie?

RIVET

- Let's think about how this could go wrong and how we could fix it!

RIVET

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



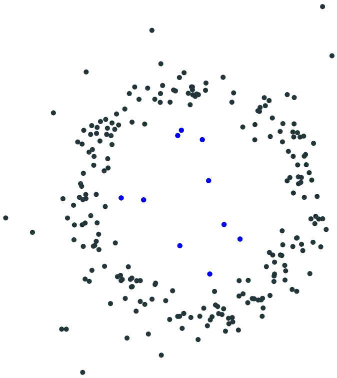
- Let's think about how this could go wrong and how we could fix it!
- We had data that looked like this:

RIVET

- This is a much more realistic example.

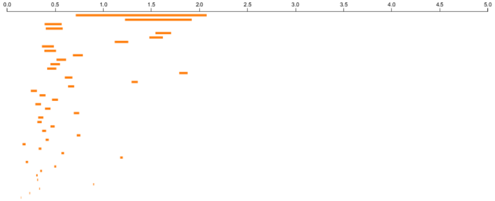
RIVET

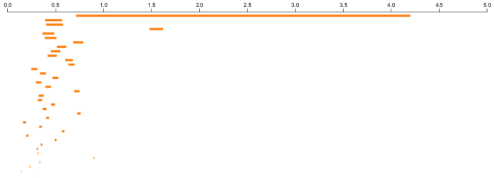
- This is a much more realistic example.
- The blue points – noise – will make it hard to see the generator of H_1.



RIVET

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





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

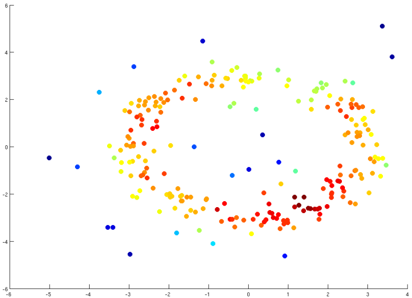


Figure 3: A density heat map of the data

RIVET

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

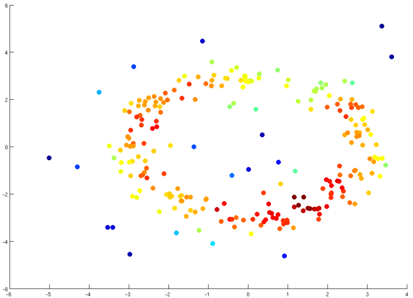


Figure 3: A density heat map of the data

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

RIVET

- Michael Lesnick and Matthew Wright have written a program for visualizing multiparameter persistence.
- Installation is not so simple.
- Let us try to use RIVET on this example together.

Thank you!

matthew.zabka@smsu.edu

References

General Overviews:

- Gunnar Carlsson. "Topology and data". *Bull. Amer. Math. Soc.* 46.2 (2009), pp. 255–308.
- Robert Ghrist. "Homological algebra and data". (2017). URL: https://www.math.upenn.edu/~ghrist/preprints/HAD.pdf.
- Jose A. Perea. "A Brief History of Persistence". (2018). URL: http://arxiv.org/abs/1809.03624.
- Matthew Wright. "Introduction to Persistent Homology - YouTube". (2016). URL: https://www.youtube.com/watch?v=h0bnG1Wavag.

More technical introduction:

- Peter Bubenik. "Statistical topological data analysis using persistence landscapes". *J. Mach. Lear. R.* 16.1 (2015).
- Steve Y Oudot. *Persistence theory: from quiver representations to data analysis*. Vol. 209. Amer. Math. Soc. Providence, RI, 2015.

Software

- Ulrich Bauer. "Ripser: a lean C++ code for the computation of Vietoris-Rips persistence barcodes". https://github. com/Ripser/ripser. 2017.
- Michael Lesnick and Matthew Wright. "Interactive visualization of 2-d persistence modules". *arXiv preprint arXiv:1512.00180* (2015). http://rivet.online/.
- Luke Wolcott. "InteractiveJPDwB: an interative program for persistence homology". https://github.com/lukewolcott/InteractiveJPDwB. 2016.