An Enchiridion for Topological Data Analysis

Bastian Rieck

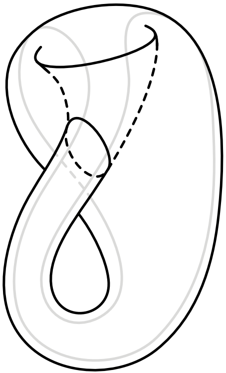
This talk

And now for something completely different…

Enchiridion, derived from the two Greek words ἐν (en, “in”) and χείρ (kheír, “hand”) refers to a small manual, in particular one that contains *practical advice*.

Here: practical advice on topological data analysis

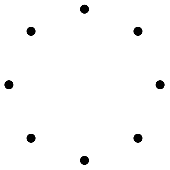
What is topological data analysis?



- Many data sets "in the wild" are assumed to have a smooth underlying structure
- This referred to as the *manifold hypothesis*
- Topological data analysis aims to describe these manifolds
- Invariant to stretching and bending

A simple example

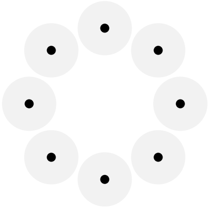
What is the shape of this set of points?



Technically, a set of points does not have a “shape”. Still, we *perceive* the points to be arranged in a circle. How can we quantify this?

Squinting leads to topology

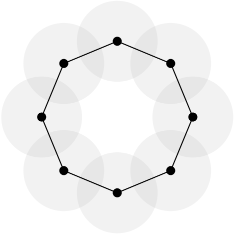
Vietoris–Rips complex construction



We can “squint” our eyes and look at how the connectivity of the points changes. The more we squint, the more connections we see.

Squinting leads to topology

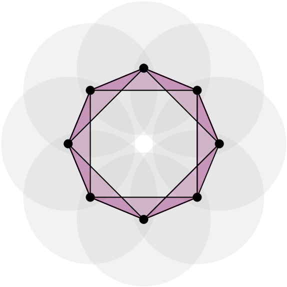
Vietoris–Rips complex construction



We can “squint” our eyes and look at how the connectivity of the points changes. The more we squint, the more connections we see.

Squinting leads to topology

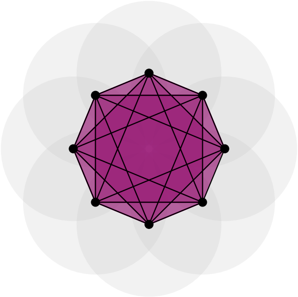
Vietoris–Rips complex construction



We can "squint" our eyes and look at how the connectivity of the points changes. The more we squint, the more connections we see.

Squinting leads to topology

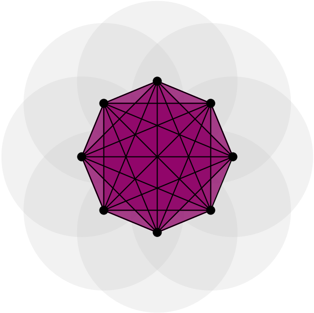
Vietoris–Rips complex construction



We can “squint” our eyes and look at how the connectivity of the points changes. The more we squint, the more connections we see.

Squinting leads to topology

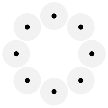
Vietoris–Rips complex construction



We can "squint" our eyes and look at how the connectivity of the points changes. The more we squint, the more connections we see.

Formalization

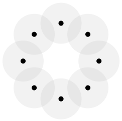
- “Squinting” leads to different *scales* at which we look at the data
- For small scales, we see only points
- For medium scales, we see a circle
- For large scales, we see a blob



Small scale

Formalization

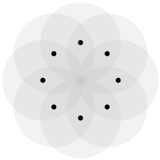
- “Squinting” leads to different *scales* at which we look at the data
- For small scales, we see only points
- For medium scales, we see a circle
- For large scales, we see a blob



Medium scale

Formalization

- "Squinting" leads to different *scales* at which we look at the data
- For small scales, we see only points
- For medium scales, we see a circle
- For large scales, we see a blob



Large scale

Even more formalization

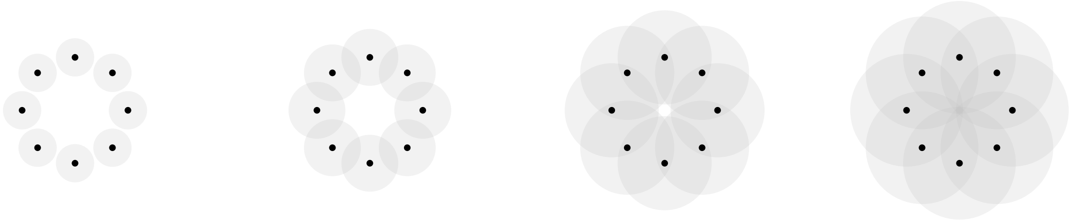
- Algebraic topology finds *invariant properties* of high-dimensional objects
- The k^{th} *Betti number* β_k counts the number of k-dimensional "holes" in ~~a manifold $\mathcal{M}$~~ your data
- Can be generalized to (almost) any ~~mathematical object~~ data set
- Key word: simplicial homology

$$0 \xrightarrow{\partial_{n+1}} C_n \xrightarrow{\partial_n} C_{n-1} \xrightarrow{\partial_{n-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$$

$$\begin{aligned} Z_d &:= \ker \partial_d \\ B_d &:= \operatorname{im} \partial_{d+1} \\ H_p &:= Z_p / B_p = \ker \partial_p / \operatorname{im} \partial_{p+1} \end{aligned}$$

Connecting Betti numbers and squinting

Letting ϵ denote our squinting parameter, i.e. our scale, let's track Betti numbers! Here, only β_0 (connected components) and β_1 (circles).

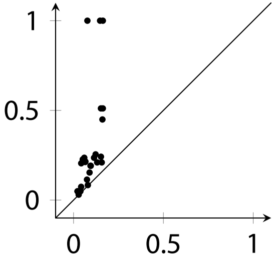


ϵ_1 ϵ_2 ϵ_3 ϵ_4

Both β_0 and β_1 change at certain values of ϵ.

Persistence diagrams

If a topological feature is *created* at ϵ_i and *destroyed* at ϵ_j, store a point (ϵ_i, ϵ_j) in the *persistence diagram*.



Persistence diagrams are diagrams in $\mathbb{R}^2$. They can serve as fingerprints of your data.

Some nice properties

- There are *metrics* for persistence diagrams[1]
- There are *kernels* for persistence diagrams[2]

[1] Bottleneck distance, Wasserstein distance
[2] Multi-scale kernel, indicator function kernel, Riemannian manifold kernel…

But what is it good for?

Basic recipe

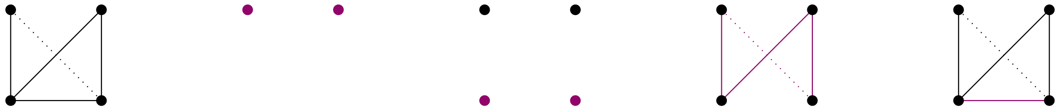
1. Define different levels of “squinting” at your data set: scales of distances, weights in networks, time, …
2. Calculate persistence diagrams (one for each dimension)
3. Use *kernel* or *distance* measure to assess dissimilarity

Applications

- Finding patterns in unstructured data sets (“point clouds” and general feature spaces)
- Describing structures in natural images
- Characterizing graphs

How does it work?

Calculate *degree* of graph nodes to obtain different scales at which the graph can be viewed (or use node/edge weights, if available). Track occurrence of β_0 (*connected components*) and β_1 (*cycles*).



Original unweighted graph

$\epsilon = 2, \beta_0 = 2, \beta_1 = 0$

$\epsilon = 3, \beta_0 = 4, \beta_1 = 0$

$\epsilon = 3, \beta_0 = 1, \beta_1 = 1$

$\epsilon = 3, \beta_0 = 1, \beta_1 = 2$

Experiments

Name	Graphs	Nodes (avg.)	Edges (avg.)	Node labels	Edge labels
DD	1178	284.32	715.66	✓	✗
IMDB-BINARY	1000	19.77	96.53	✗	✗
IMDB-MULTI	1500	13.00	65.94	✗	✗
MUTAG	188	17.93	19.79	✓	✓
NCI1	4110	29.87	32.30	✓	✗
NCI109	4127	29.68	32.13	✓	✗
PROTEINS	1113	39.06	72.82	✓	✗
REDDIT-BINARY	2000	429.63	497.75	✗	✗
REDDIT-MULTI-5k	4999	508.52	594.87	✗	✗

Compare topology-based network analysis against *graph kernels* (special algorithms for assessing the dissimilarity of two graphs).

Results

Name	Best known method	TDA	Difference
DD[1]	0.80	0.72	0.08
IMDB-BINARY[2]	0.67	0.73	−0.06
IMDB-MULTI[2]	0.45	0.52	−0.07
MUTAG[3]	0.88	0.95	−0.07
NCI1[2]	0.80	0.68	0.12
NCI109[2]	0.85	0.61	0.24
PROTEINS[2]	0.76	0.76	0.00
REDDIT-BINARY[2]	0.78	0.88	−0.10
REDDIT-MULTI-5k[2]	0.55	0.48	0.07

[1] Nino Shervashidze et al. "Weisfeiler–Lehman Graph Kernels". In: *Journal of Machine Learning Research* 12 (Nov. 2011), pp. 2539–2561

[2] Pinar Yanardag and S.V.N. Vishwanathan. "Deep Graph Kernels". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2015, pp. 1365–1374

[3] Mahito Sugiyama et al. "`graphkernels`: R and Python packages for graph comparison". In: *Bioinformatics* 34.3 (2018), pp. 530–532

Want to know more?

Find out if TDA can help with *your* data by visiting `https://is.gd/topology` and try your own graph data[3] or contact me at `bastian.rieck@bsse.ethz.ch`.

[3] Some setup is required, though…