A Primer on Multi-Scale Topological Kernels

Bastian Rieck (@Pseudomanifold)

What is algebraic topology?

Develop invariants that classify topological spaces up to homeomorphism.

What is algebraic topology?

Develop invariants that classify topological spaces up to homeomorphism.

Use tools from algebra to study topological spaces.

What is algebraic topology?

Develop invariants that classify topological spaces up to homeomorphism.

Use tools from algebra to study topological spaces.

Understand shapes through calculations.

A first taste

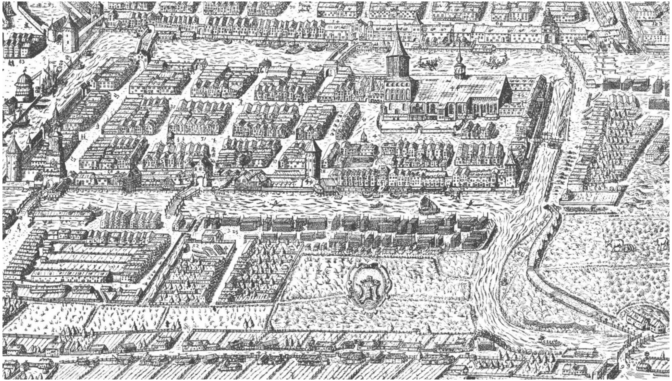
Seven Bridges of Königsberg

Is there a walk through the city that crosses every bridge *exactly* once?

A first taste

Seven Bridges of Königsberg

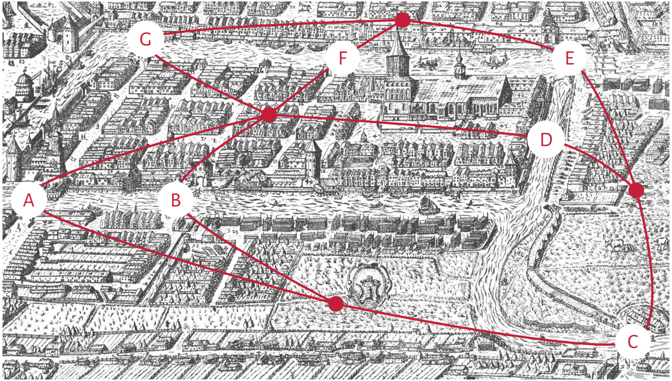
Is there a walk through the city that crosses every bridge *exactly* once?



A first taste

Seven Bridges of Königsberg

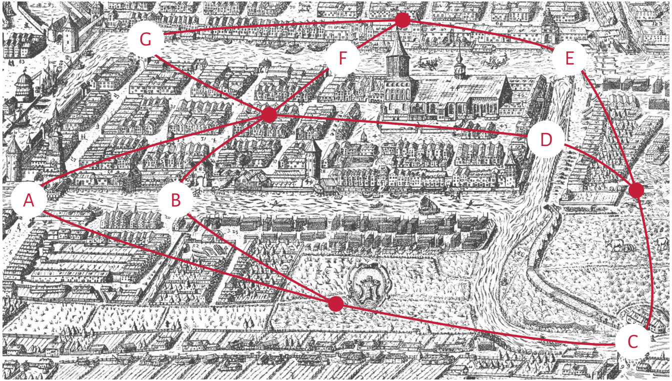
Is there a walk through the city that crosses every bridge *exactly* once?



A first taste

Seven Bridges of Königsberg

Is there a walk through the city that crosses every bridge *exactly* once?



No such walk can exist because there are more than two vertices with *odd* degree!

Simple invariants

Betti numbers

Space	β_0	β_1	β_2

The d^{th} Betti number counts the number of d-dimensional holes. It can be used to distinguish between spaces.

- $d = 0$: connected components
- $d = 1$: cycles
- $d = 2$: voids

Simple invariants

Betti numbers

The d^{th} Betti number counts the number of d-dimensional holes. It can be used to distinguish between spaces.

- $d = 0$: connected components
- $d = 1$: cycles
- $d = 2$: voids

Space		β_0	β_1	β_2
Point		1	0	0

Simple invariants

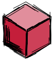
Betti numbers

The d^{th} Betti number counts the number of d-dimensional holes. It can be used to distinguish between spaces.

$d = 0$: connected components

$d = 1$: cycles

$d = 2$: voids

	Space	β_0	β_1	β_2
Point		1	0	0
Cube		1	0	1

Simple invariants

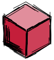
Betti numbers

The d^{th} Betti number counts the number of d-dimensional holes. It can be used to distinguish between spaces.

$d = 0$: connected components

$d = 1$: cycles

$d = 2$: voids

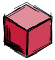
Space		β_0	β_1	β_2
Point		1	0	0
Cube		1	0	1
Sphere		1	0	1

Simple invariants

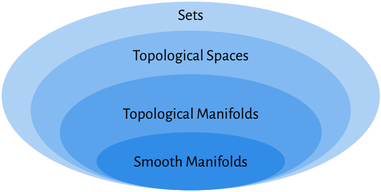
Betti numbers

The d^{th} Betti number counts the number of d-dimensional holes. It can be used to distinguish between spaces.

- $d = 0$: connected components
- $d = 1$: cycles
- $d = 2$: voids

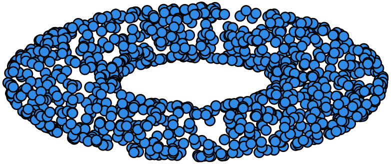
Space		β_0	β_1	β_2
Point		1	0	0
Cube		1	0	1
Sphere		1	0	1
Torus		1	2	1

Why topology?

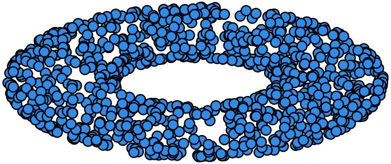


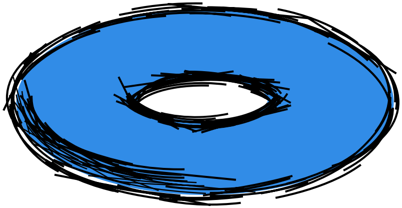
Most of machine learning happens at the level of smooth manifolds. A topological perspective is *more general* but also *coarser*.

Reality is often messy...



Reality is often messy...

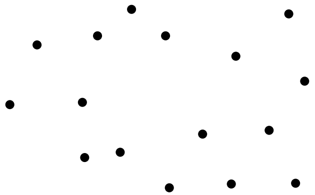




Persistent homology

Track topological features across different scales

Approximate a point cloud at different scales and observe how topological features appear and disappear as the scale changes.

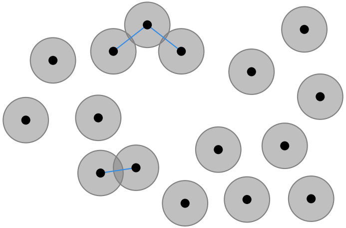


$$\mathcal{V}_\epsilon := \{\{x_1, x_2, \ldots\} \mid \mathrm{dist}(x_i, x_j) \leqslant \epsilon \text{ for all } i \neq j\}$$

Persistent homology

Track topological features across different scales

Approximate a point cloud at different scales and observe how topological features appear and disappear as the scale changes.

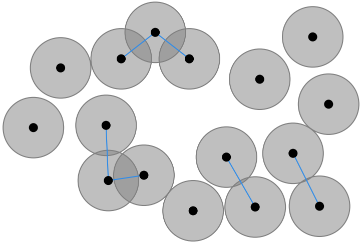


$$\mathcal{V}_\epsilon := \{\{x_1, x_2, \ldots\} \mid \mathrm{dist}(x_i, x_j) \leqslant \epsilon \text{ for all } i \neq j\}$$

Persistent homology

Track topological features across different scales

Approximate a point cloud at different scales and observe how topological features appear and disappear as the scale changes.

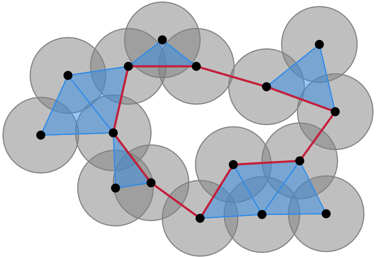


$$\mathcal{V}_\epsilon := \left\{\{x_1, x_2, \ldots\} \mid \mathrm{dist}(x_i, x_j) \leqslant \epsilon \text{ for all } i \neq j\right\}$$

Persistent homology

Track topological features across different scales

Approximate a point cloud at different scales and observe how topological features appear and disappear as the scale changes.

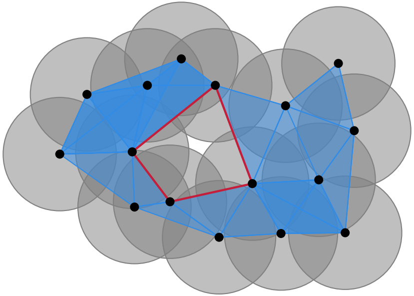


$$\mathcal{V}_\epsilon := \{\{x_1, x_2, \ldots\} \mid \mathrm{dist}(x_i, x_j) \leqslant \epsilon \text{ for all } i \neq j\}$$

Persistent homology

Track topological features across different scales

Approximate a point cloud at different scales and observe how topological features appear and disappear as the scale changes.

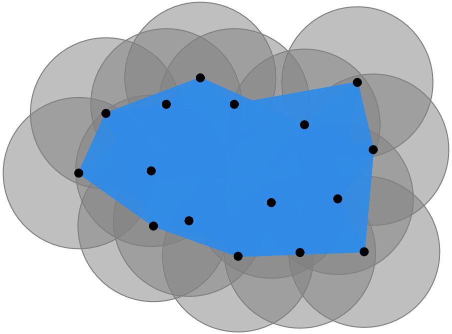


$$\mathcal{V}_\epsilon := \{\{x_1, x_2, \ldots\} \mid \mathrm{dist}(x_i, x_j) \leqslant \epsilon \text{ for all } i \neq j\}$$

Persistent homology

Track topological features across different scales

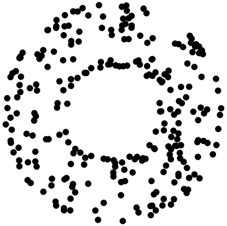
Approximate a point cloud at different scales and observe how topological features appear and disappear as the scale changes.



$$\mathcal{V}_\epsilon := \{\{x_1, x_2, \ldots\} \mid \mathrm{dist}(x_i, x_j) \leqslant \epsilon \text{ for all } i \neq j\}$$

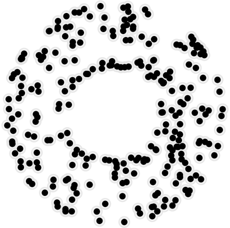
Persistent homology

Storing topological features in *persistence diagrams*



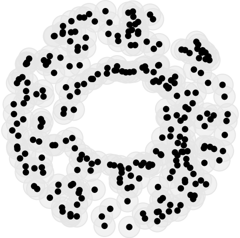
Persistent homology

Storing topological features in *persistence diagrams*



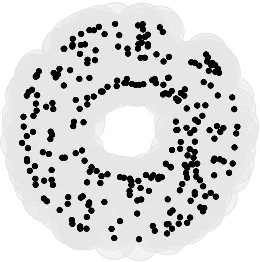
Persistent homology

Storing topological features in *persistence diagrams*



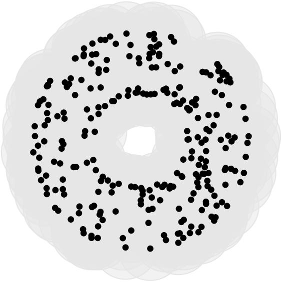
Persistent homology

Storing topological features in *persistence diagrams*



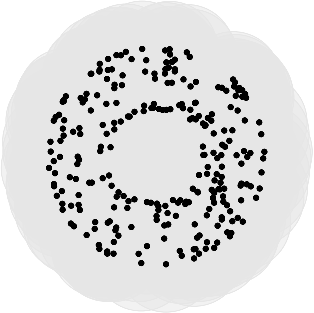
Persistent homology

Storing topological features in *persistence diagrams*



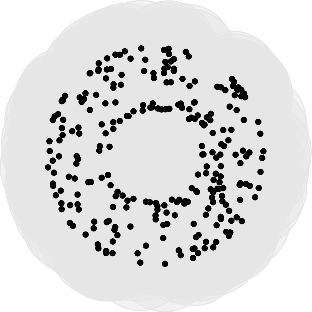
Persistent homology

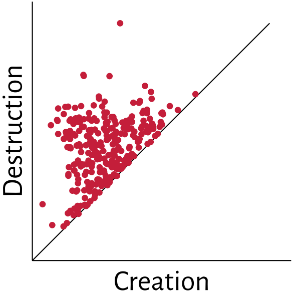
Storing topological features in *persistence diagrams*



Persistent homology

Storing topological features in *persistence diagrams*





Brief interlude

Persistent homology can also be considered as a *generic* way of associating a sequence of algebraic objects, such as (Abelian) groups to other objects, such as topological spaces.

Application areas

- Graphs
- Point clouds
- Time series

Distances between persistence diagrams

Bottleneck distance

Given two persistence diagrams $\mathcal{D}$ and $\mathcal{D}'$, their *bottleneck* distance is defined as

$$W_\infty(\mathcal{D}, \mathcal{D}') := \inf_{\eta\colon \mathcal{D} \to \mathcal{D}'} \sup_{x \in \mathcal{D}} \|x - \eta(x)\|_\infty,$$

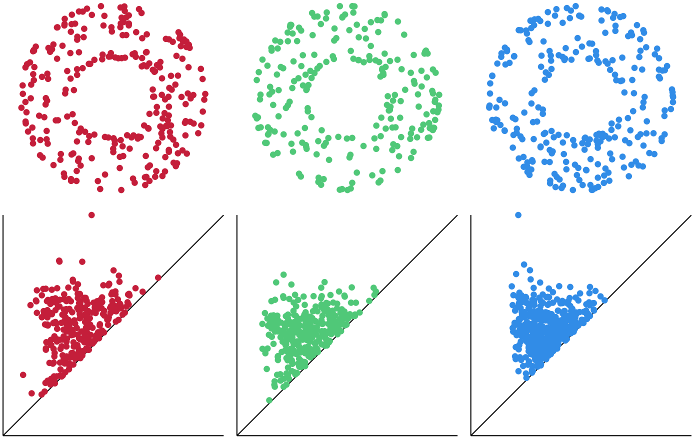
where $\eta\colon \mathcal{D} \to \mathcal{D}'$ denotes a bijection between the point sets of $\mathcal{D}$ and $\mathcal{D}'$ and $\|\cdot\|_\infty$ refers to the L_∞ distance between two points in $\mathbb{R}^2$.

Wasserstein distance

$$W_p(\mathcal{D}_1, \mathcal{D}_2) := \left(\inf_{\eta\colon \mathcal{D}_1 \to \mathcal{D}_2} \sum_{x \in \mathcal{D}_1} \|x - \eta(x)\|_\infty^p \right)^{\frac{1}{p}}$$

Stability properties of persistence diagrams

Intuitive view



Stability properties of persistence diagrams

Formal view

Let $\mathcal{M}$ be a triangulable space with continuous tame functions $f, g \colon \mathcal{M} \to \mathbb{R}$. Then the corresponding persistence diagrams satisfy $W_\infty(\mathcal{D}_f, \mathcal{D}_g) \leqslant \|f - g\|_\infty$.

Topological features in the context of machine learning

Topological features constitute an additional set of **inductive biases**.

Topological features are **complementing** machine learning algorithms.

Topological features have advantageous **theoretical properties**.

Examples

D. J. E. Waibel, S. Atwell, M. Meier, C. Marr and **B. Rieck**, 'Capturing Shape Information with Multi-Scale Topological Loss Terms for 3D Reconstruction', *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2022, pp. 150–159

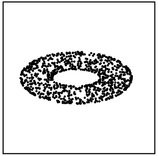
M. Horn, E. De Brouwer, M. Moor, Y. Moreau, B. Rieck and K. Borgwardt, 'Topological Graph Neural Networks', *International Conference on Learning Representations*, 2022

L. O'Bray*, **B. Rieck*** and K. Borgwardt, 'Filtration Curves for Graph Representation', *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2021, pp. 1267–1275

B. Rieck* et al., 'Uncovering the Topology of Time-Varying fMRI Data using Cubical Persistence', *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 6900–6912

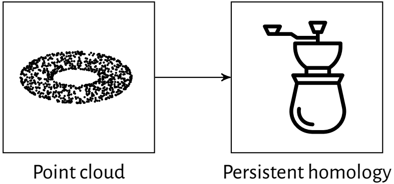
M. Moor*, M. Horn*, **B. Rieck**† and K. Borgwardt†, 'Topological Autoencoders', *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 7045–7054

A generic topology-driven machine learning pipeline

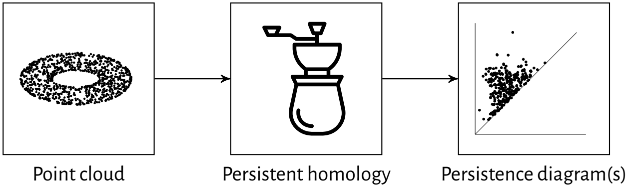


Point cloud

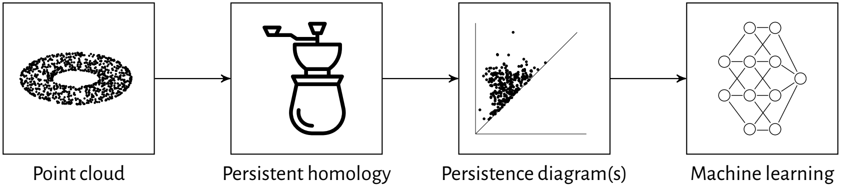
A generic topology-driven machine learning pipeline



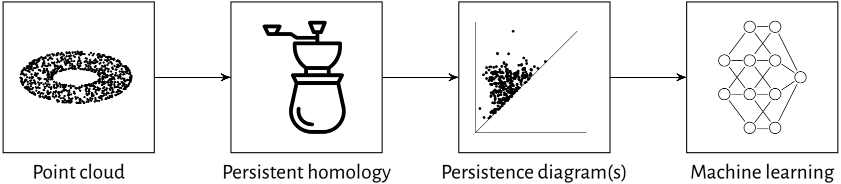
A generic topology-driven machine learning pipeline



A generic topology-driven machine learning pipeline



A generic topology-driven machine learning pipeline



Some caveats

Persistence diagrams are cumbersome to work with due to their multiset structure.

Bottleneck and Wasserstein distances may be computationally inefficient.

A multi-scale kernel

The *first* kernel between persistence diagrams; it is simple to implement and expressive.

Kernel and feature map definition

$$k_\sigma(\mathcal{D}, \mathcal{D}') := \frac{1}{8\pi\sigma} \sum_{p \in \mathcal{D},\, q \in \mathcal{D}'} \exp(-8^{-1}\sigma^{-1}\|p - q\|^2) - \exp(-8^{-1}\sigma^{-1}\|p - \overline{q}\|^2)$$

$$\Phi(x) := \frac{1}{4\pi\sigma} \sum_{p \in \mathcal{D}} \exp(-4^{-1}\sigma^{-1}\|x - p\|^2) - \exp(-4^{-1}\sigma^{-1}\|x - \overline{p}\|^2)$$

Here, $\overline{p} := (d, c)$ for $p = (c, d)$, i.e. the *mirror image* of a point across the diagonal.

J. Reininghaus, S. Huber, U. Bauer and R. Kwitt, 'A stable multi-scale kernel for topological machine learning', *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4741–4748

Universality

Theorem

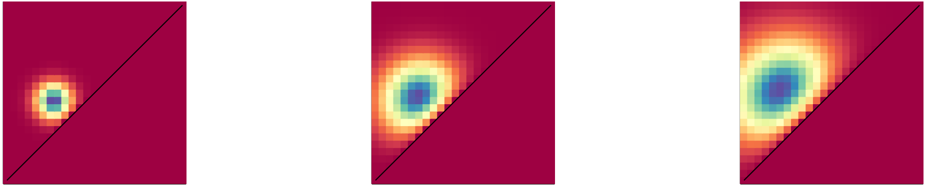
The kernel $k(\mathcal{D}, \mathcal{D}') := \exp(k_\sigma(\mathcal{D}, \mathcal{D}'))$ is *universal* with respect to the first Wasserstein distance W_1.

(This means that we should be able to use it with MMD!)

R. Kwitt, S. Huber, M. Niethammer, W. Lin and U. Bauer, 'Statistical Topological Data Analysis — A Kernel Perspective', *Advances in Neural Information Processing Systems*, ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett, vol. 28, 2015, pp. 3070–3078

Example

Feature map illustration



$\sigma = 0.1$ $\sigma = 0.5$ $\sigma = 1.0$

More kernels

Not covered in detail

Alternative formulations exist, based on sliced Wasserstein distance calculations,[1] kernel embeddings,[2] or Riemannian geometry.[3]

[1] M. Carrière, M. Cuturi and S. Oudot, 'Sliced Wasserstein Kernel for Persistence Diagrams', *Proceedings of the 34th International Conference on Machine Learning*, ed. by D. Precup and Y. W. Teh, vol. 70, Proceedings of Machine Learning Research, 2017, pp. 664–673

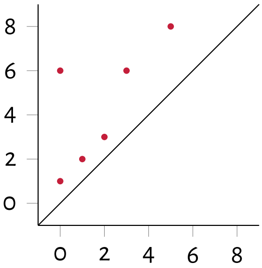
[2] G. Kusano, K. Fukumizu and Y. Hiraoka, 'Kernel Method for Persistence Diagrams via Kernel Embedding and Weight Factor', *Journal of Machine Learning Research* 18.189, 2018, pp. 1–41

[3] T. Le and M. Yamada, 'Persistence Fisher Kernel: A Riemannian Manifold Kernel for Persistence Diagrams', *Advances in Neural Information Processing Systems*, ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett, vol. 31, 2018, pp. 10007–10018

Betti curves

A simplified representation of persistence diagrams

Persistence diagram

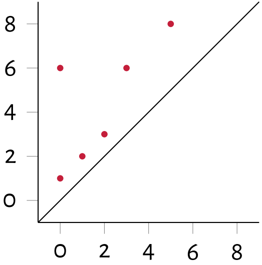


The Betti curve is a function mapping a persistence diagram to an integer-valued curve, i.e. each Betti curve is a function $\mathcal{B}\colon \mathbb{R} \to \mathbb{N}$.

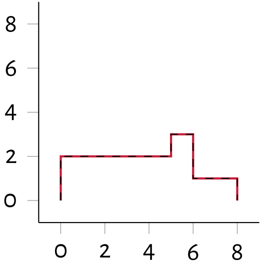
Betti curves

A simplified representation of persistence diagrams

Persistence diagram



Persistence barcode



The Betti curve is a function mapping a persistence diagram to an integer-valued curve, i.e. each Betti curve is a function $\mathcal{B}\colon \mathbb{R} \to \mathbb{N}$.

Betti curves

A simplified representation of persistence diagrams

The Betti curve is a function mapping a persistence diagram to an integer-valued curve, i.e. each Betti curve is a function $\mathcal{B}\colon \mathbb{R} \to \mathbb{N}$.

Betti curves

Properties

Easy to calculate

Simple representation, 'living' in the space of piecewise linear functions

Vector space operations are possible (addition, scalar multiplication)

Distances and kernels can be defined

We obtain a simple *kernel* via:

$$k(\mathcal{D}, \mathcal{D}') := \int_{\mathbb{R}} \mathcal{B}_{\mathcal{D}}(x)\mathcal{B}_{\mathcal{D}'}(x)dx$$

Open question

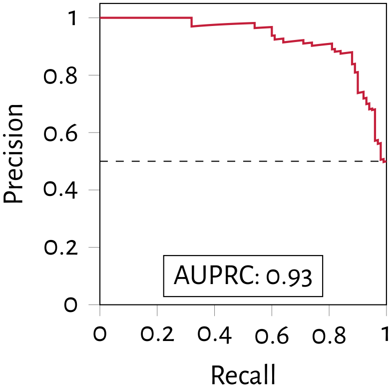
While this kernel can be evaluated quickly, can we do *better*?

B. Rieck, F. Sadlo and H. Leitte, 'Topological Machine Learning with Persistence Indicator Functions', *Topological Methods in Data Analysis and Visualization V*, ed. by H. Carr, I. Fujishiro, F. Sadlo and S. Takahashi, Cham, Switzerland: Springer, 2020, pp. 87–101, arXiv: `1907.13496 [math.AT]`

Betti curves

Classification scenario example

- Use `REDDIT-BINARY` data set (co-occurrence graphs)
- Calculate filtration based on *vertex degree*
- Calculate persistence diagrams for $d = 1$ (cycles)
- Given $p = 1$, use a kernel SVM for classification



Still state-of-the-art performance, but at a fraction of the (computational) cost of graph neural networks (GNNs).

Application

Classifying graphs with weighted edges

Pick function to induce a graph filtration $G_1 \subseteq G_2 \cdots \subseteq G_k = G$.

Pick descriptor function $f\colon \mathcal{G} \to \mathbb{R}$.

Evaluate f alongside the filtration.

This turns a graph G into a *path*.

We can treat such paths as generalised Betti curves, which we call *filtration curves*.

L. O'Bray*, **B. Rieck*** and K. Borgwardt, 'Filtration Curves for Graph Representation', *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2021, pp. 1267–1275

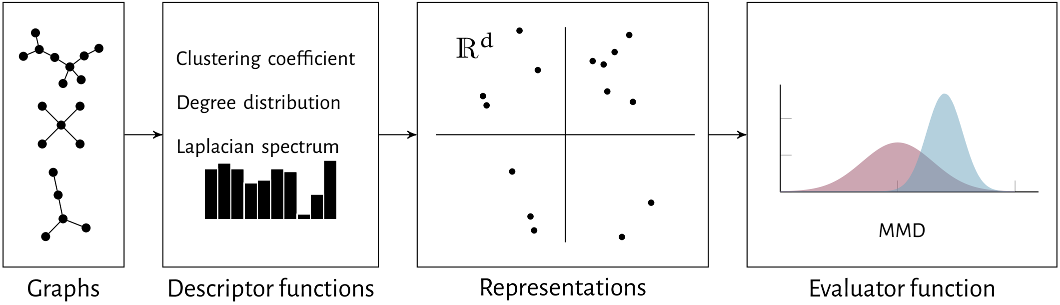
Experiments

Surprisingly competitive!

	Native edge weights				*Non-native edge weights*			
Method	BZR_MD	COX2_MD	DHFR_MD	ER_MD	BZR	COX2	DHFR	PROTEINS
CSM	**77.63 ± 1.29**	—	—	—	84.54 ± 0.65	79.78 ± 1.04	77.99 ± 0.96	—
HGK-SP	60.08 ± 0.88	59.92 ± 0.66	67.95 ± 0.00	59.42 ± 0.00	81.99 ± 0.30	78.16 ± 0.00	72.48 ± 0.65	74.53 ± 0.35
HGK-WL	52.64 ± 1.20	57.15 ± 1.20	66.08 ± 1.02	66.72 ± 1.28	81.42 ± 0.60	78.16 ± 0.00	75.35 ± 0.66	74.53 ± 0.35
MLG	51.46 ± 0.61	51.15 ± 0.00	67.95 ± 0.00	60.72 ± 0.69	**88.04 ± 0.70**	76.76 ± 0.87	**83.22 ± 0.94**	**75.55 ± 0.71**
WL	67.45 ± 1.40	60.07 ± 2.22	62.56 ± 1.51	70.35 ± 1.01	86.16 ± 0.97	79.67 ± 1.32	81.72 ± 0.80	73.06 ± 0.47
WL-OA	68.19 ± 1.09	62.37 ± 2.11	64.10 ± 1.70	70.96 ± 0.75	87.43 ± 0.81	**81.08 ± 0.89**	82.40 ± 0.97	73.50 ± 0.87
GNN	69.87 ± 1.29	66.05 ± 3.16	73.11 ± 1.59	75.38 ± 1.60	79.34 ± 2.43	76.53 ± 1.82	74.56 ± 1.44	70.31 ± 1.93
FC-V	75.61 ± 1.13	**73.41 ± 0.79**	**76.78 ± 0.69**	**82.51 ± 1.04**	85.61 ± 0.59	81.01 ± 0.88	81.43 ± 0.48	74.54 ± 0.48

More graph learning applications

Evaluating graph generative models



Some issues with the status quo

Kernels may not be valid (i.e. positive definite).

How to pick parameters?

Why use kernels on descriptor function representations?

L. O'Bray*, M. Horn*, **B. Rieck**† and K. Borgwardt†, 'Evaluation Metrics for Graph Generative Models: Problems, Pitfalls, and Practical Solutions', *International Conference on Learning Representations*, 2022, arXiv: `2106.01098 [cs.LG]`

Moving forward

Topological methods are versatile and can be calculated for different modalities.

Kernels are an integral part of modern computational topology!

Moving forward

Topological methods are versatile and can be calculated for different modalities.

Kernels are an integral part of modern computational topology!

Open questions

Can we use graph kernels for evaluating such models?

Are persistence diagrams the *right* structure to define kernels on?

Can we combine Bayesian optimisation with kernels?