

Sending persistence diagrams into (Hilbert) space

Ulrich Bauer

TUM

July 7, 2015

ACAT final meeting
IST Austria

Joint work with: Jan Reininghaus, Stefan Huber, Roland Kwitt



||2



?

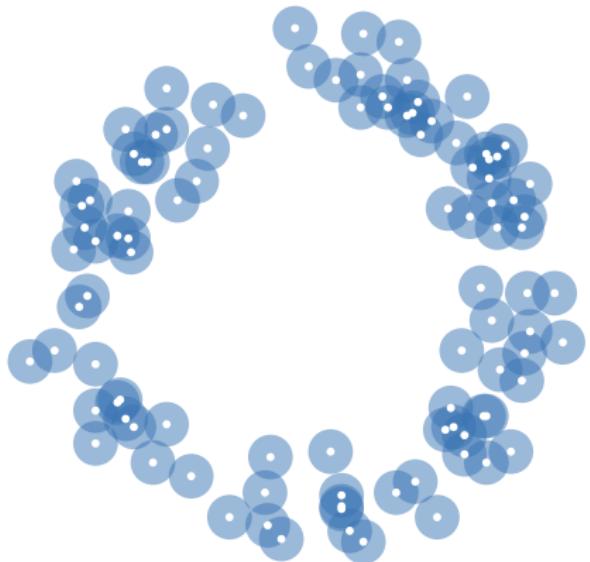


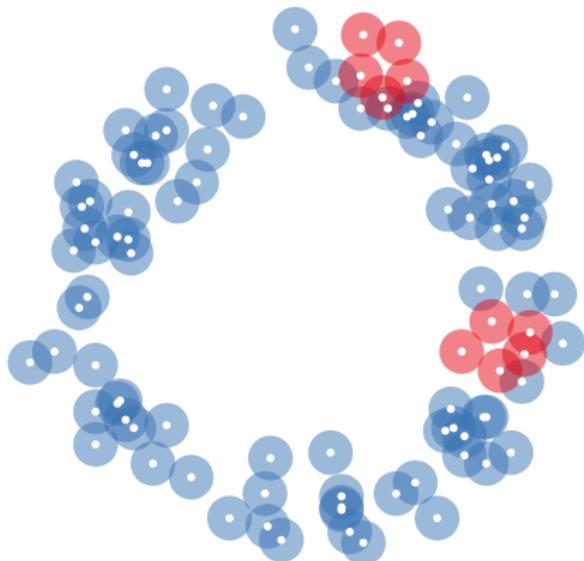
12



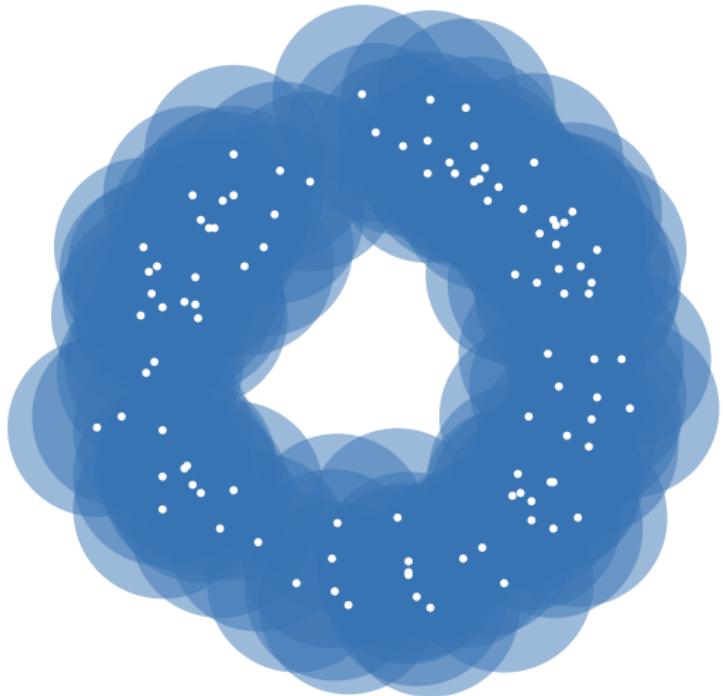
?

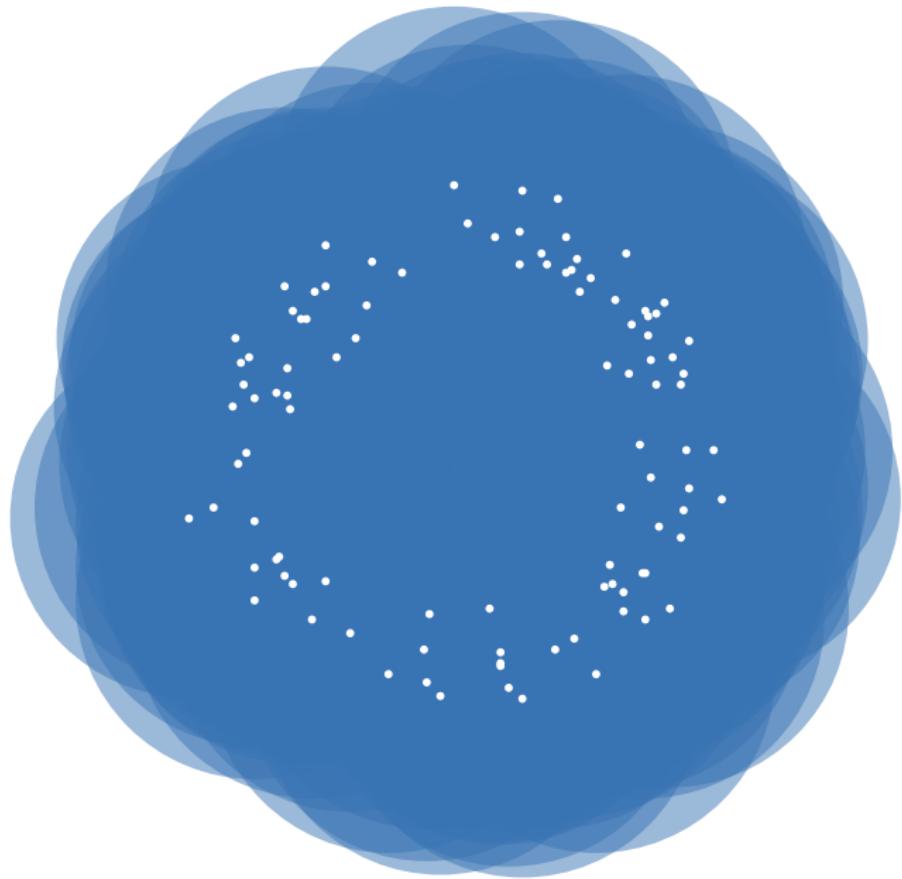


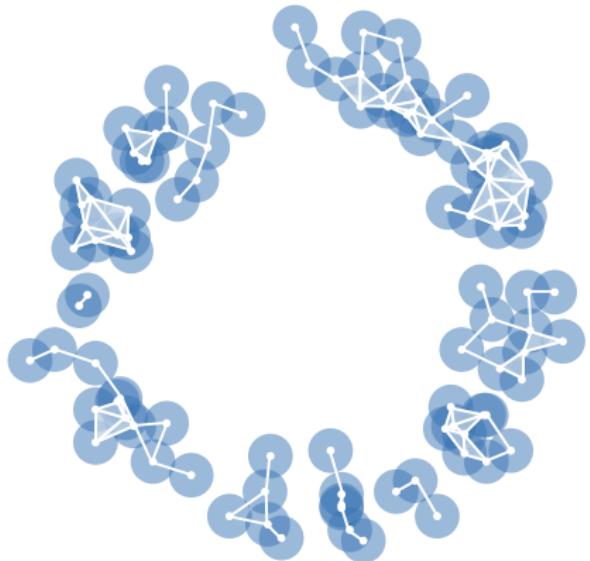


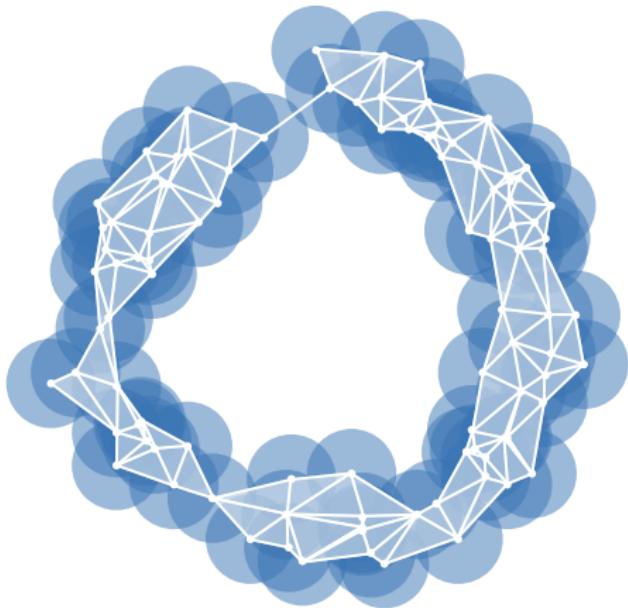


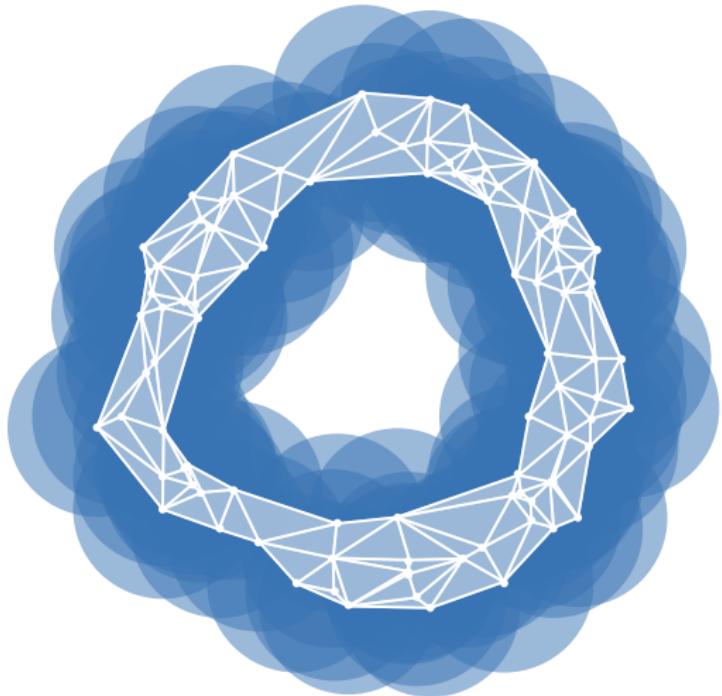


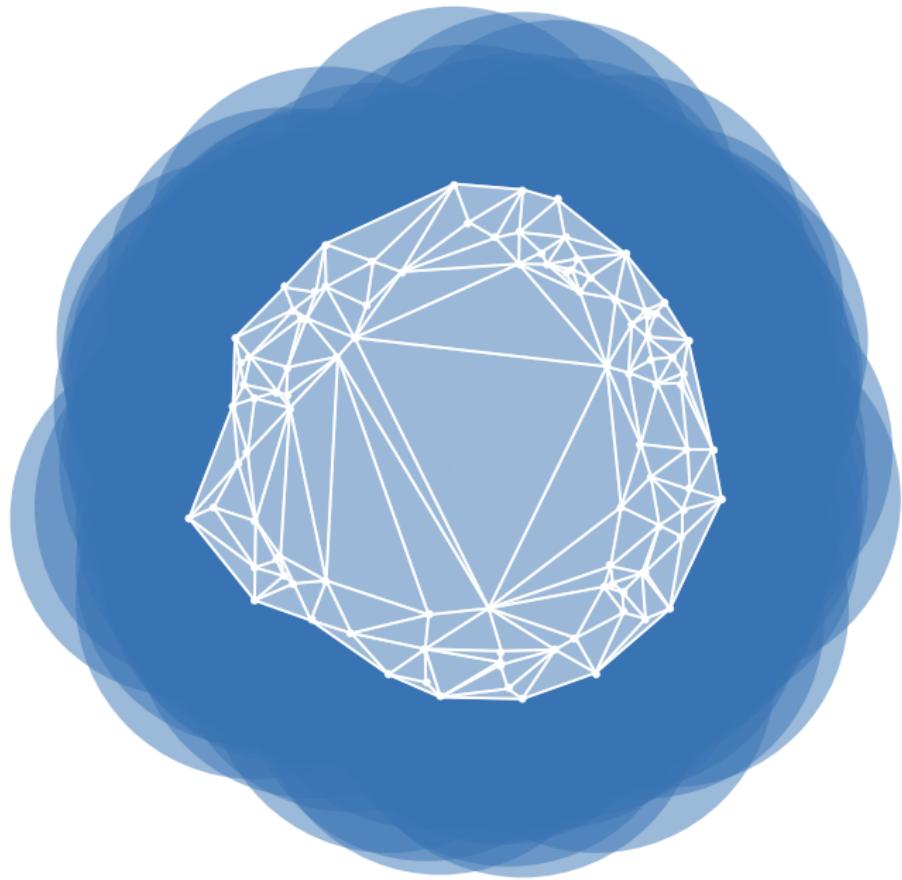


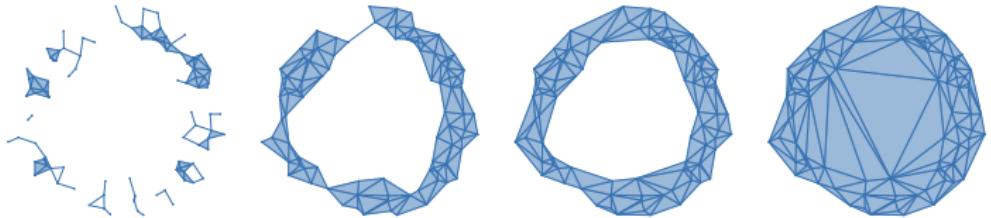


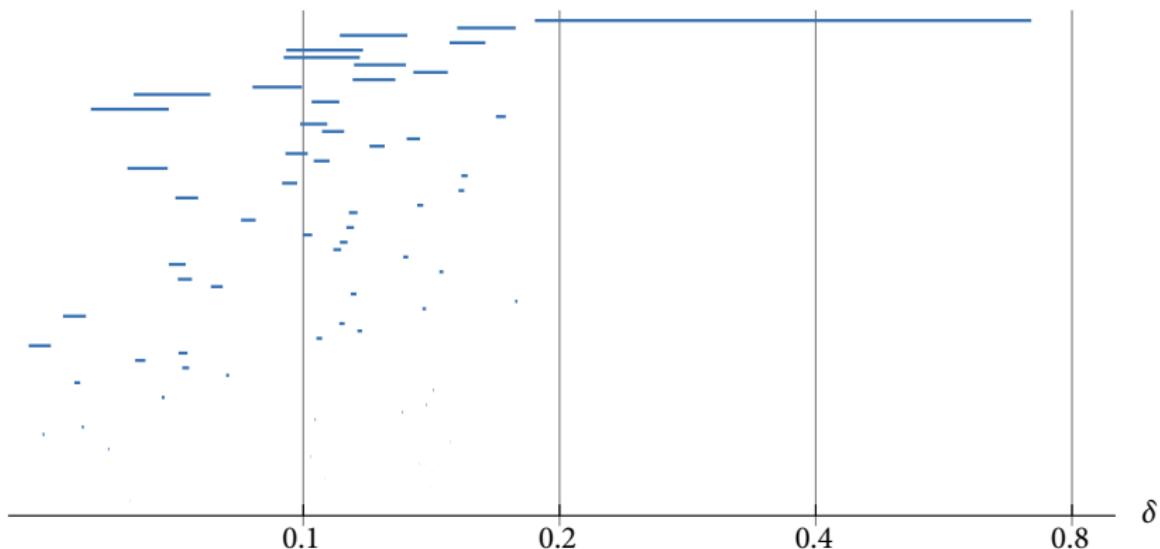
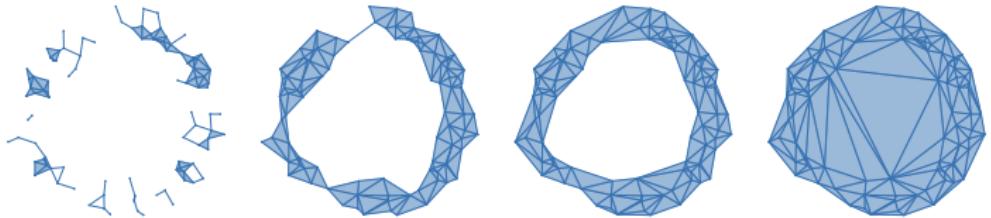


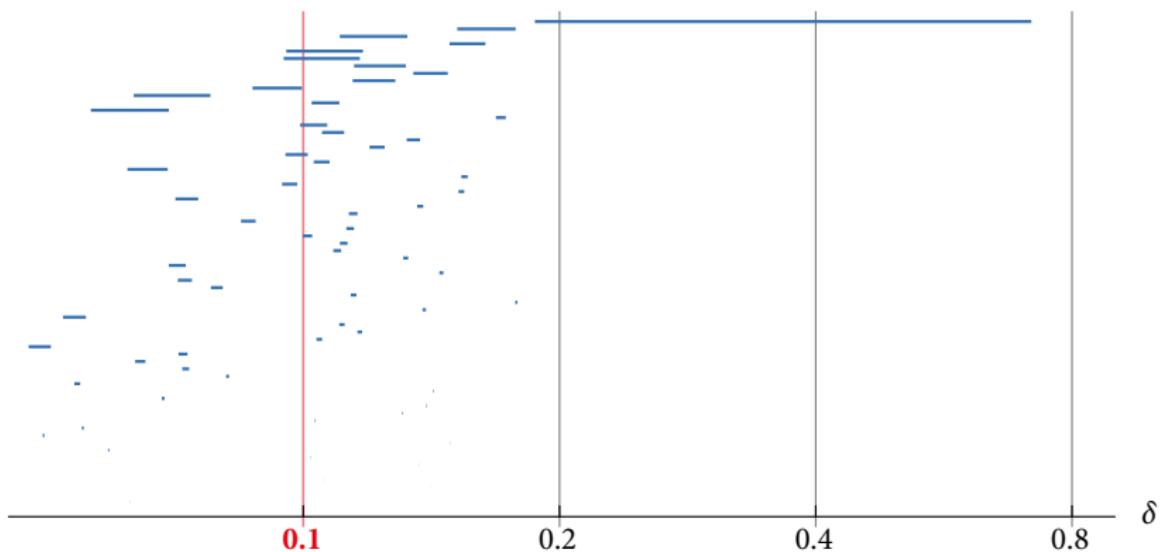
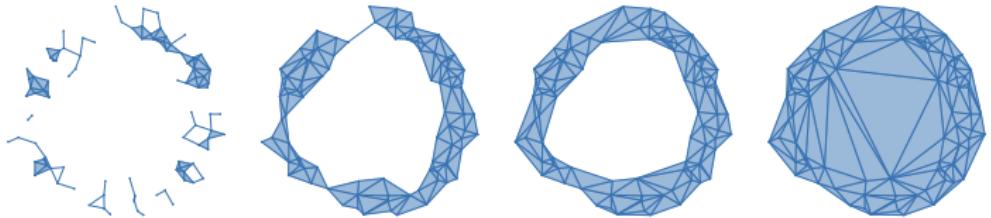


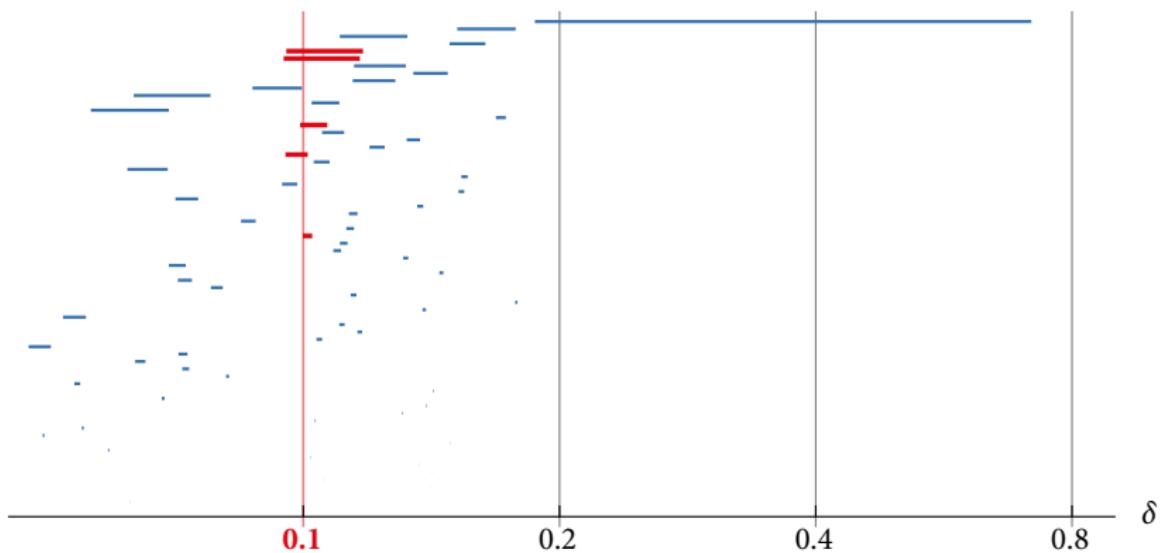
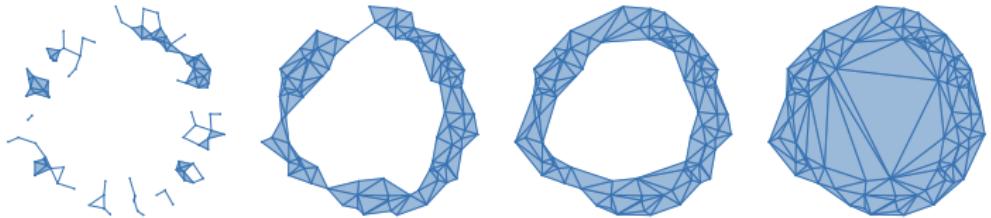


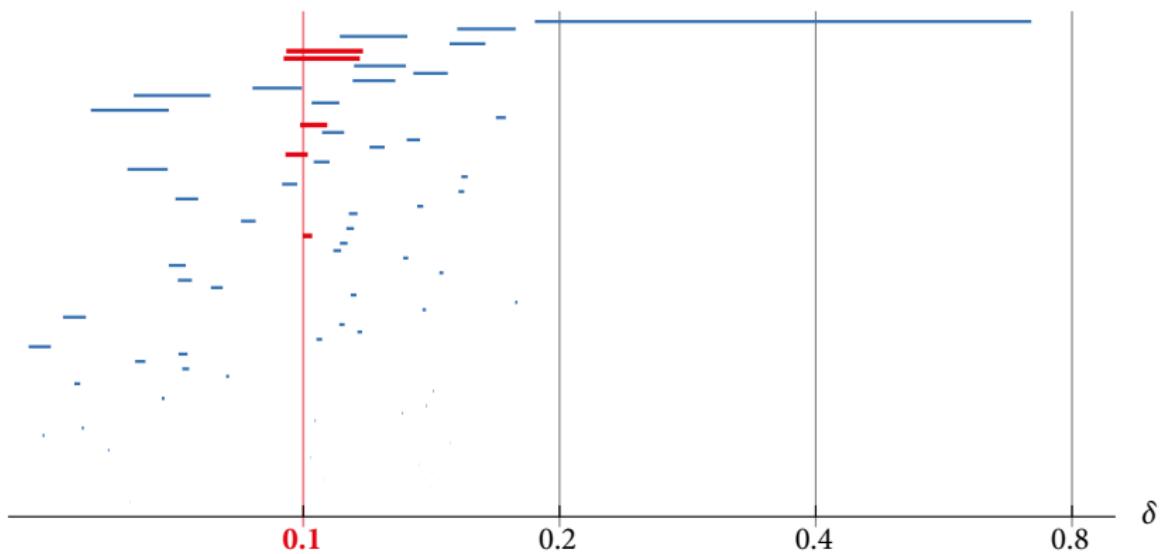
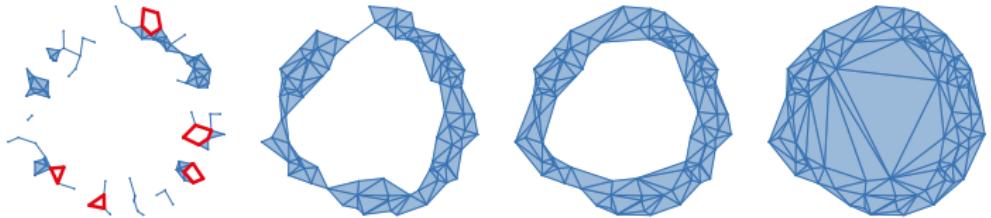


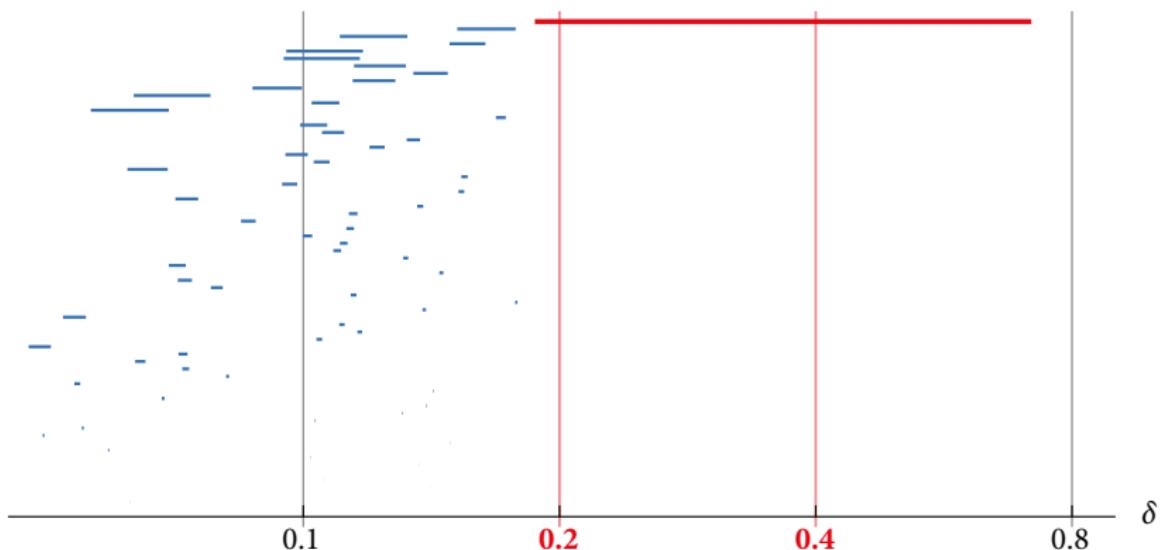
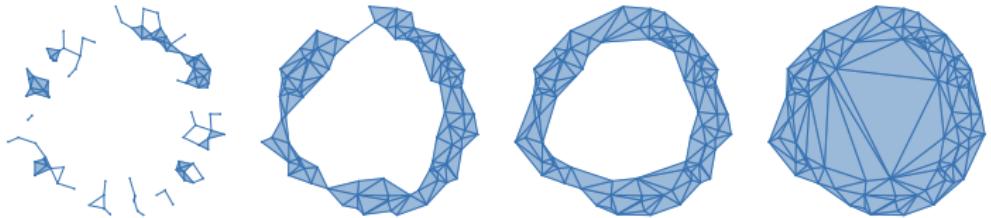


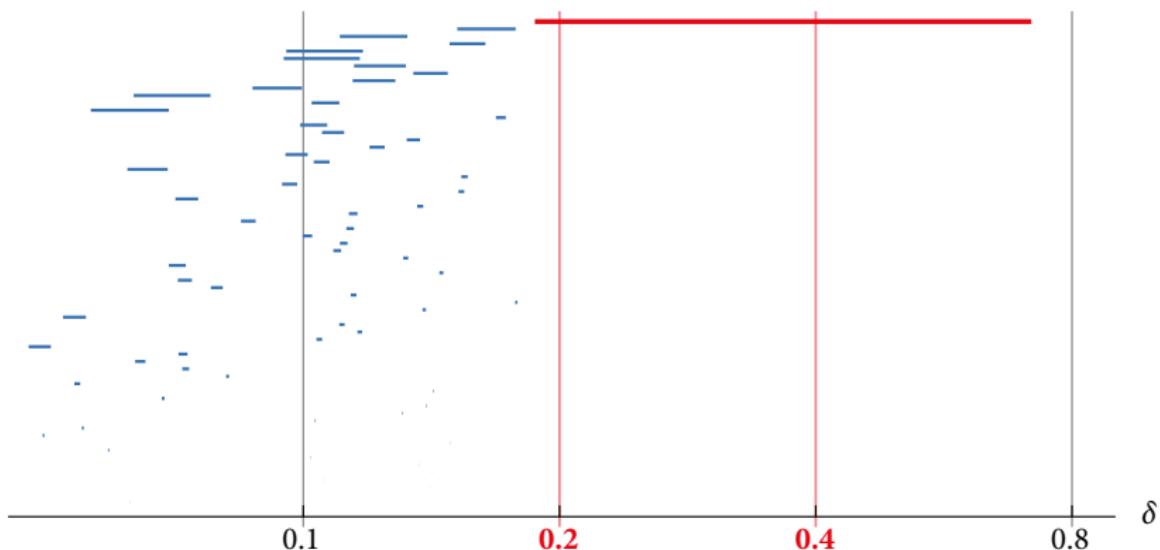
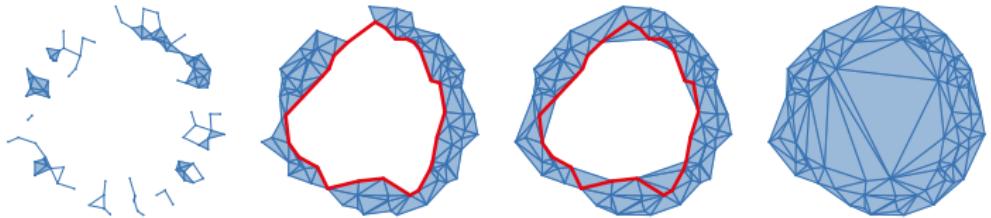




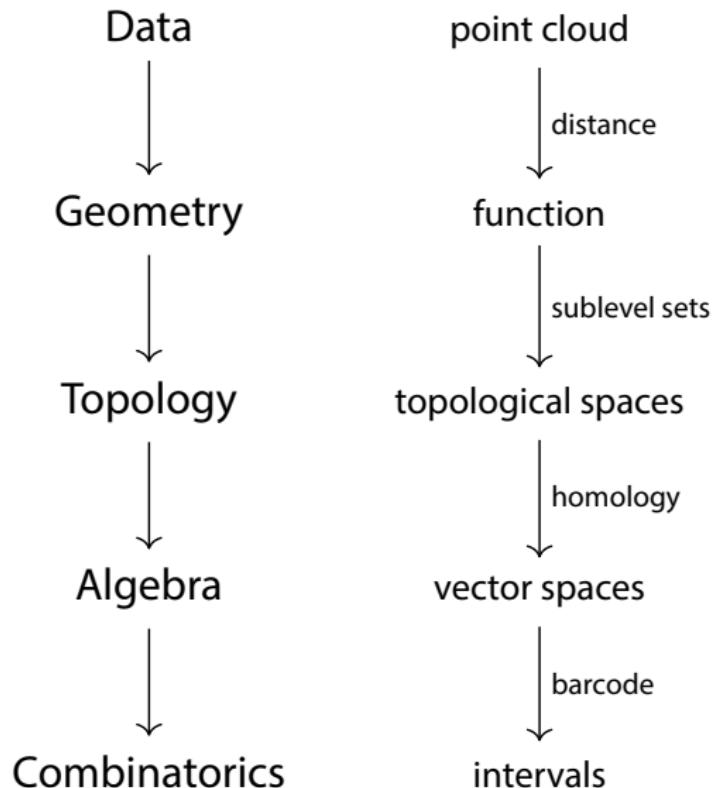








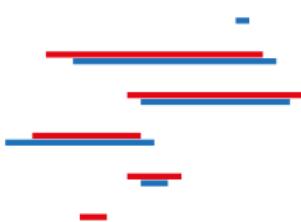
The pipeline of topological data analysis



Stability of persistence barcodes for functions

Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

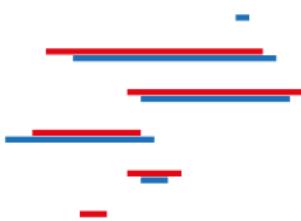
If two functions $f, g : K \rightarrow \mathbb{R}$ have
distance $\|f - g\|_\infty \leq \delta$, then there exists
a δ -matching of their barcodes.



Stability of persistence barcodes for functions

Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

If two functions $f, g : K \rightarrow \mathbb{R}$ have
distance $\|f - g\|_\infty \leq \delta$, then there exists
a δ -matching of their barcodes.

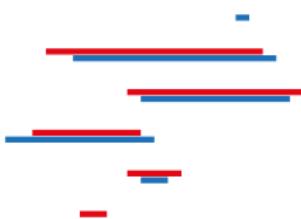


- ▶ matching of sets X, Y : bijection of subsets $X' \subseteq X, Y' \subseteq Y$

Stability of persistence barcodes for functions

Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

If two functions $f, g : K \rightarrow \mathbb{R}$ have distance $\|f - g\|_\infty \leq \delta$, then there exists a δ -matching of their barcodes.

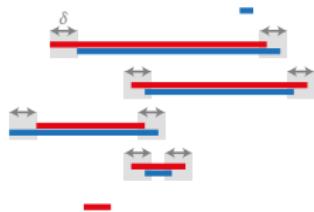


- ▶ matching of sets X, Y : bijection of subsets $X' \subseteq X, Y' \subseteq Y$
- ▶ δ -matching of barcodes:

Stability of persistence barcodes for functions

Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

If two functions $f, g : K \rightarrow \mathbb{R}$ have distance $\|f - g\|_\infty \leq \delta$, then there exists a δ -matching of their barcodes.

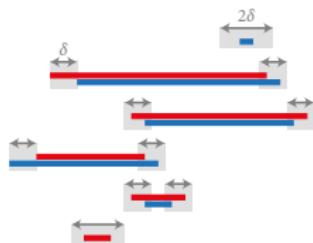


- ▶ matching of sets X, Y : bijection of subsets $X' \subseteq X, Y' \subseteq Y$
- ▶ δ -matching of barcodes:
 - ▶ matched intervals have endpoints within distance $\leq \delta$

Stability of persistence barcodes for functions

Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

If two functions $f, g : K \rightarrow \mathbb{R}$ have distance $\|f - g\|_\infty \leq \delta$, then there exists a δ -matching of their barcodes.

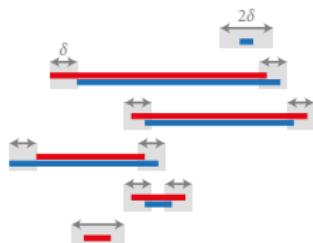


- ▶ matching of sets X, Y : bijection of subsets $X' \subseteq X, Y' \subseteq Y$
- ▶ δ -matching of barcodes:
 - ▶ matched intervals have endpoints within distance $\leq \delta$
 - ▶ unmatched intervals have length $\leq 2\delta$

Stability of persistence barcodes for functions

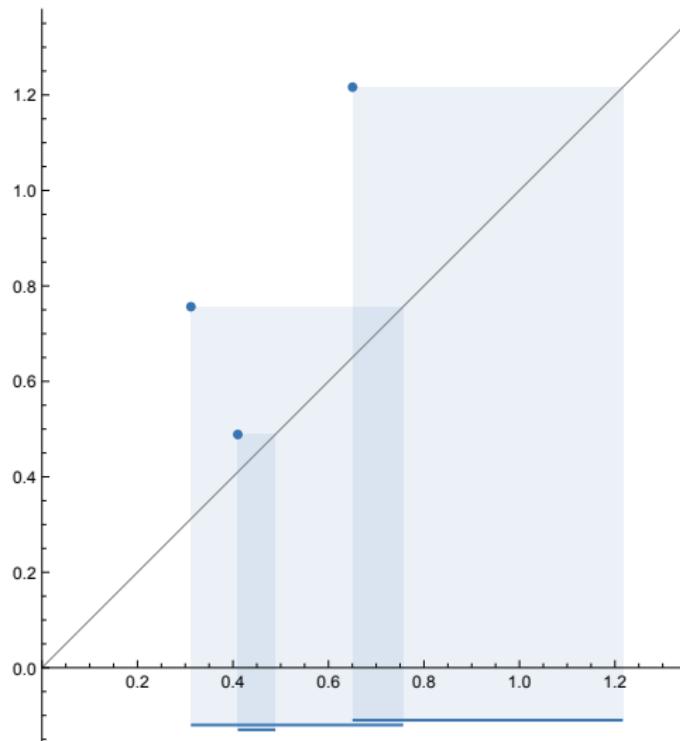
Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

If two functions $f, g : K \rightarrow \mathbb{R}$ have distance $\|f - g\|_\infty \leq \delta$, then there exists a δ -matching of their barcodes.

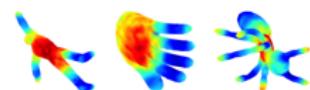


- ▶ matching of sets X, Y : bijection of subsets $X' \subseteq X, Y' \subseteq Y$
- ▶ δ -matching of barcodes:
 - ▶ matched intervals have endpoints within distance $\leq \delta$
 - ▶ unmatched intervals have length $\leq 2\delta$
- ▶ Bottleneck distance d_∞ : infimum δ admitting a δ -matching

Persistence barcodes and persistence diagrams



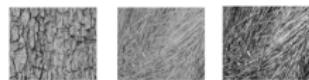
Topological machine learning



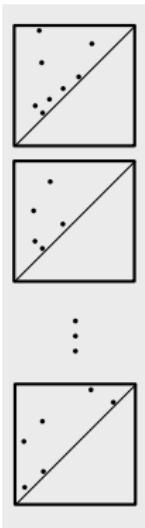
Task: shape retrieval



Task: object recognition



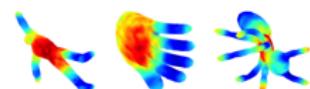
Task: texture recognition



⋮



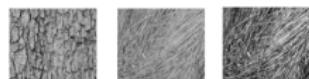
Topological machine learning



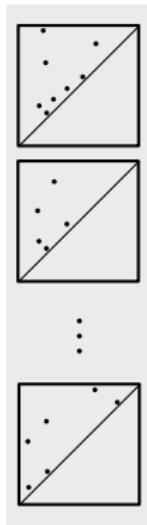
Task: shape retrieval



Task: object recognition



Task: texture recognition



SVM

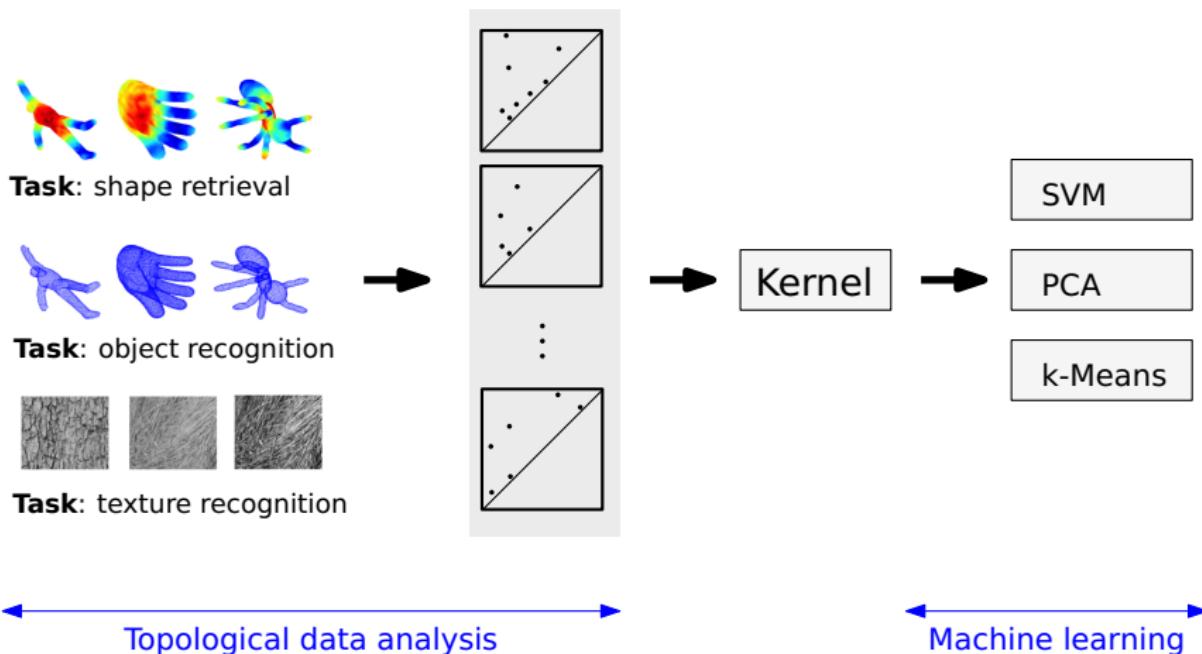
PCA

k-Means

↔ Topological data analysis ↔

↔ Machine learning ↔

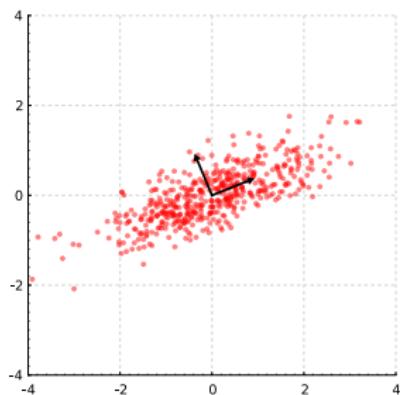
Topological machine learning



Linear machine learning methods

Many machine learning methods solve linear geometric problems on vector spaces

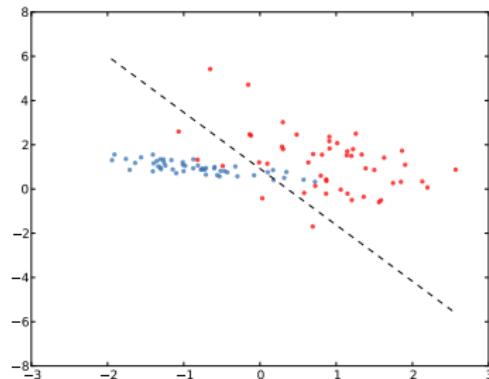
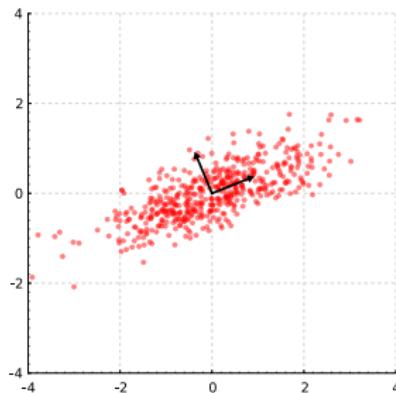
- ▶ Principal component analysis (PCA): find orthogonal directions of largest variance



Linear machine learning methods

Many machine learning methods solve linear geometric problems on vector spaces

- ▶ Principal component analysis (PCA): find orthogonal directions of largest variance
- ▶ Support vector machines (SVM): find best-separating hyperplane



Kernels and kernel methods

Let X be a set, and \mathcal{H} a vector space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.

- ▶ Consider a map $\Phi : X \rightarrow \mathcal{H}$
- ▶ Then

$$k(\cdot, \cdot) = \langle \Phi(\cdot), \Phi(\cdot) \rangle_{\mathcal{H}} : X \times X \rightarrow \mathbb{R}$$

is a *kernel* (and Φ is the associated *feature map*)

Kernels and kernel methods

Let X be a set, and \mathcal{H} a vector space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.

- ▶ Consider a map $\Phi : X \rightarrow \mathcal{H}$
- ▶ Then

$$k(\cdot, \cdot) = \langle \Phi(\cdot), \Phi(\cdot) \rangle_{\mathcal{H}} : X \times X \rightarrow \mathbb{R}$$

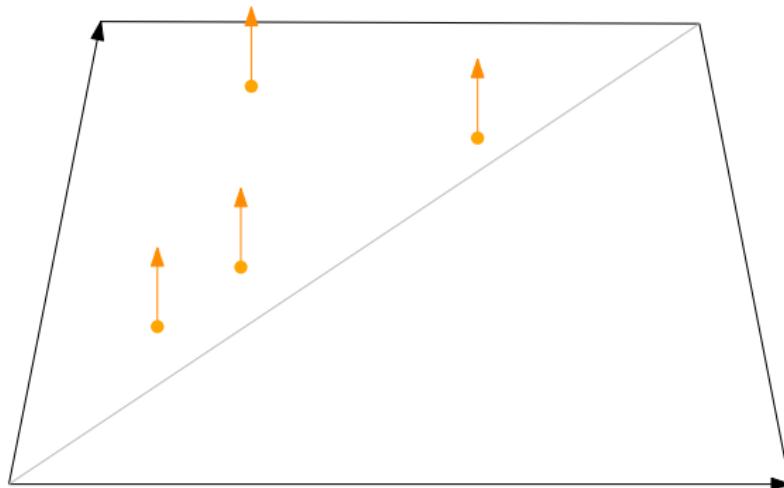
is a *kernel* (and Φ is the associated *feature map*)

For many linear methods:

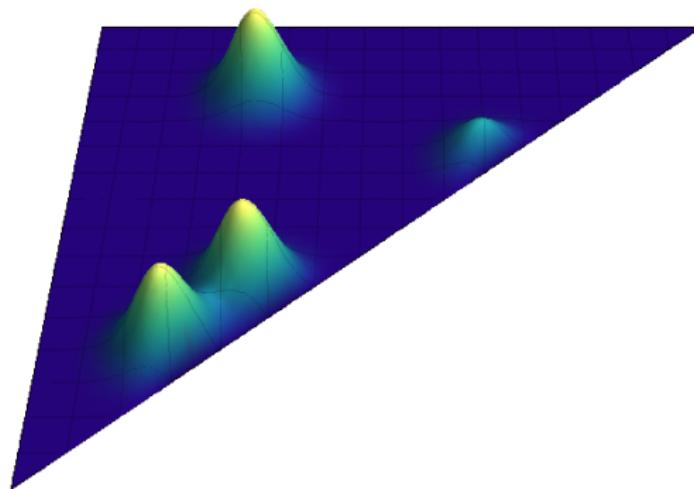
- ▶ Feature map Φ not required explicitly
- ▶ No coordinate basis for \mathcal{H} required
- ▶ Computations can be performed by evaluating $k(\cdot, \cdot)$

This is called the *kernel trick*.

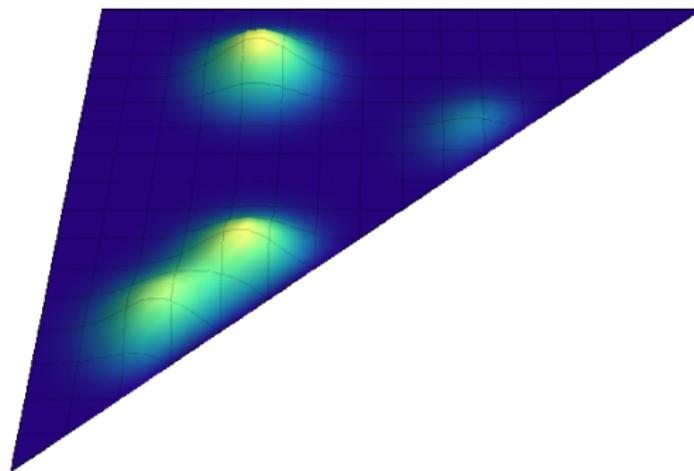
Feature maps for persistence diagrams by diffusion



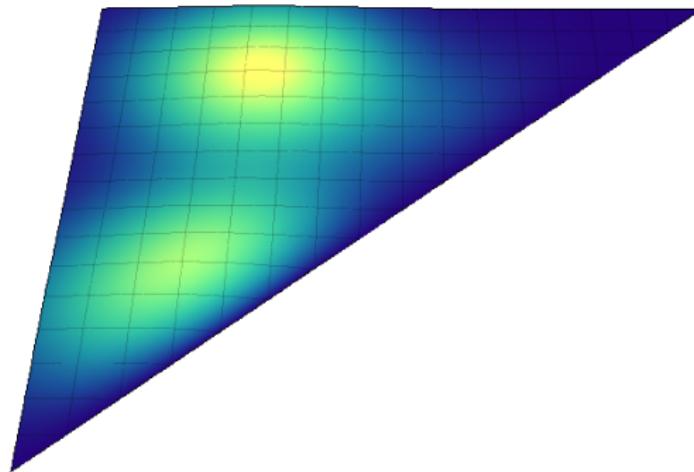
Feature maps for persistence diagrams by diffusion



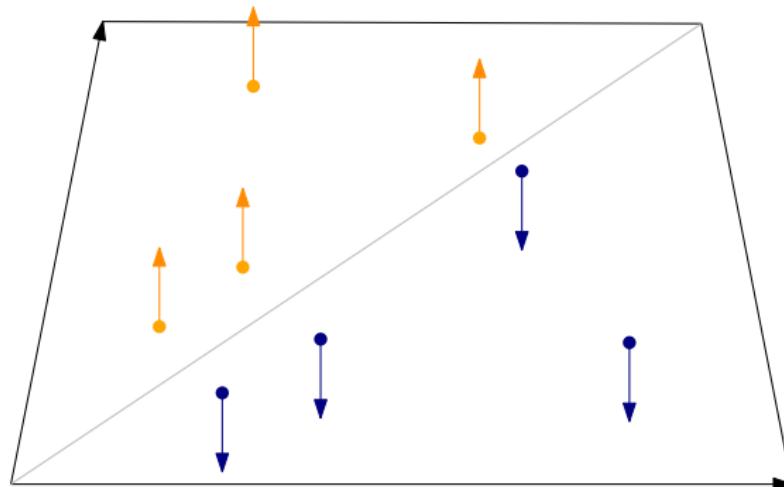
Feature maps for persistence diagrams by diffusion



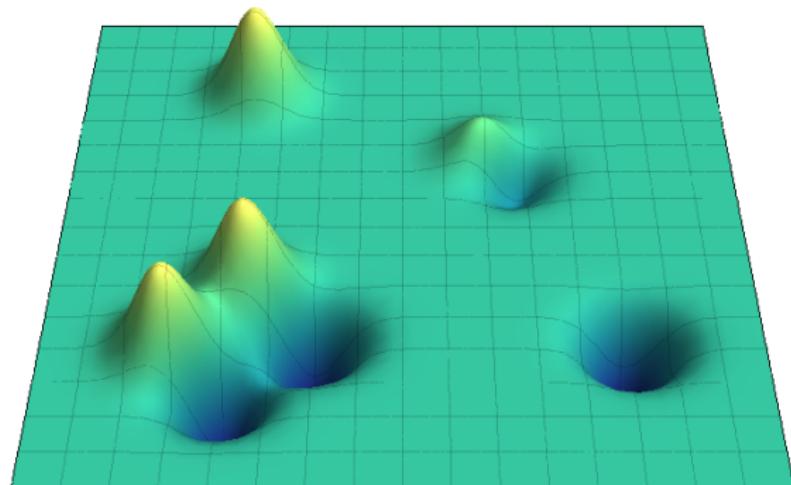
Feature maps for persistence diagrams by diffusion



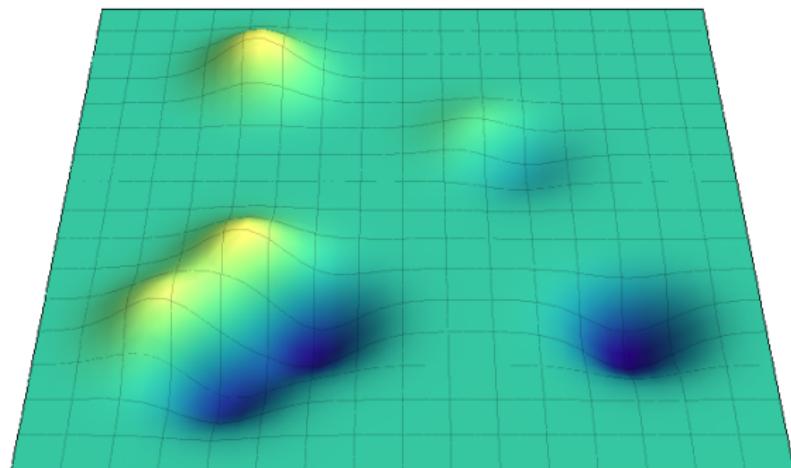
Feature maps for persistence diagrams by diffusion



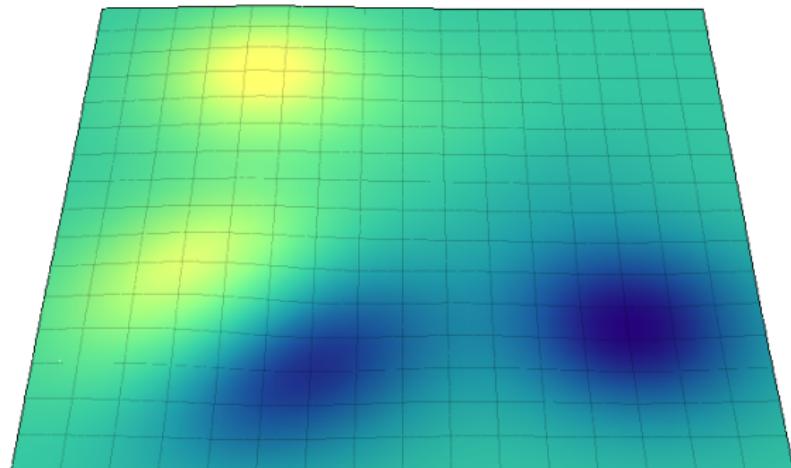
Feature maps for persistence diagrams by diffusion



Feature maps for persistence diagrams by diffusion



Feature maps for persistence diagrams by diffusion



Metric distortion of kernel feature maps

Let $d_{\text{PSS}}(D_f, D_g)$ be the L^2 distance between smoothings of the persistence diagrams D_f, D_g (i.e., a *kernel distance*).

Theorem

For two persistence diagrams D_f and D_g and $\sigma > 0$ we have

$$d_{\text{PSS}}(D_f, D_g) \leq \frac{C}{\sigma} d_1(D_f, D_g).$$

Metric distortion of kernel feature maps

Let $d_{\text{PSS}}(D_f, D_g)$ be the L^2 distance between smoothings of the persistence diagrams D_f, D_g (i.e., a *kernel distance*).

Theorem

For two persistence diagrams D_f and D_g and $\sigma > 0$ we have

$$d_{\text{PSS}}(D_f, D_g) \leq \frac{C}{\sigma} d_1(D_f, D_g).$$

- Here: stability with respect to *Wasserstein distance*:

$$d_p(D_f, D_g) = \left(\inf_{\mu: D_f \leftrightarrow D_g} \sum_{x \in D_f} \|x - \mu(x)\|_\infty^p \right)^{\frac{1}{p}}$$

- Note: bottleneck distance $d_B(D_f, D_g) = \lim_{p \rightarrow \infty} d_p(D_f, D_g)$

Sliced Wasserstein distance and kernel

Let SW be the *sliced Wasserstein distance*.

Theorem

For persistence diagrams D_f and D_g with at most N bars, we have

$$\frac{1}{8N^2} d_1(D_f, D_g) \leq d_{\text{SW}}(D_f, D_g) \leq \sqrt{8} d_1(D_f, D_g).$$

Sliced Wasserstein distance and kernel

Let SW be the *sliced Wasserstein distance*.

Theorem

For persistence diagrams D_f and D_g with at most N bars, we have

$$\frac{1}{8N^2} d_1(D_f, D_g) \leq d_{\text{SW}}(D_f, D_g) \leq \sqrt{8} d_1(D_f, D_g).$$

- ▶ SW is *conditionally negative definite*. This implies that $k_{\text{SW}}(\cdot, \cdot) = \exp(-\text{SW}(\cdot, \cdot)^2)$ is a kernel.
- ▶ Let d_{kSW} denote the corresponding RHKS distance.
- ▶ Then d_{kSW} is not the same as d_{SW} .
- ▶ But on bounded subspaces, the two are strongly equivalent (bi-Lipschitz).

Extending the TDA pipeline

Mapping barcodes into a Hilbert space?

- ▶ desirable for (kernel-based) machine learning methods and statistics
- ▶ stability (Lipschitz continuity): important for reliable predictions
- ▶ inverse stability (bi-Lipschitz): avoid loss of information

Extending the TDA pipeline

Mapping barcodes into a Hilbert space?

- ▶ desirable for (kernel-based) machine learning methods and statistics
- ▶ stability (Lipschitz continuity): important for reliable predictions
- ▶ inverse stability (bi-Lipschitz): avoid loss of information

Existing kernels for persistence diagrams:

- ▶ stability bounds only for 1–Wasserstein distance
- ▶ Lipschitz constants depend on bound on number and range of bars
- ▶ no bi-Lipschitz bounds known

Extending the TDA pipeline

Mapping barcodes into a Hilbert space?

- ▶ desirable for (kernel-based) machine learning methods and statistics
- ▶ stability (Lipschitz continuity): important for reliable predictions
- ▶ inverse stability (bi-Lipschitz): avoid loss of information

Existing kernels for persistence diagrams:

- ▶ stability bounds only for 1–Wasserstein distance
- ▶ Lipschitz constants depend on bound on number and range of bars
- ▶ no bi-Lipschitz bounds known

Can we hope for something better?

No bi-Lipschitz feature maps for persistence

Theorem (B, Carrière 2018)

*There is no bi-Lipschitz map from the persistence diagrams
(with the bottleneck or any p -Wasserstein distance)
into any Euclidean space \mathbb{R}^d ,
even when restricting to bounded range or number of bars.*

No bi-Lipschitz feature maps for persistence

Theorem (B, Carrière 2018)

*There is no bi-Lipschitz map from the persistence diagrams
(with the bottleneck or any p -Wasserstein distance)
into any Euclidean space \mathbb{R}^d ,
even when restricting to bounded range or number of bars.*

Theorem (B, Carrière 2018)

*If there was such a bi-Lipschitz map into any Hilbert space,
the ratio of the Lipschitz constants would have to go to ∞
if either bound on the number or range of bars go to ∞ .*

Properties of the space of persistence diagrams

Known properties:

- ▶ The space of persistence diagrams (with p -distance) is complete and separable [Mileyko et al. 2011]
- ▶ There are no upper bounds on curvature

These properties provide no immediate obstructions. But:

Persistence spaces and doubling spaces

Persistence spaces and doubling spaces

Definition

A metric space is *doubling* if every ball of radius r can be covered by a constant number of balls of radius $\frac{r}{2}$.

Proposition

Any Euclidean space is doubling. The image of a doubling space under a bi-Lipschitz map is again doubling.

Theorem (B, Carrière 2018)

The space of persistence diagrams is not doubling.