# Sending persistence into space

## On the metric distortion of embedding persistence diagrams into reproducing kernel Hilbert spaces

Ulrich Bauer

TUM

June 28, 2018

Curves & Surfaces, Arcachon

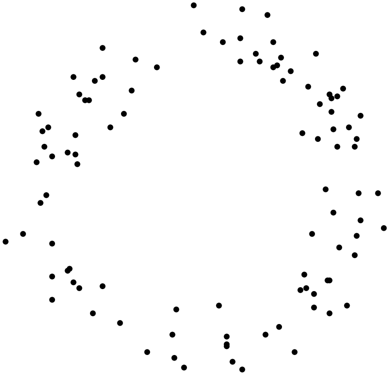
Joint work with Mathieu Carrière (INRIA/Columbia)

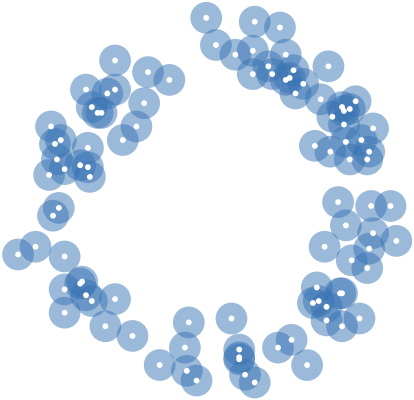
$\cong$

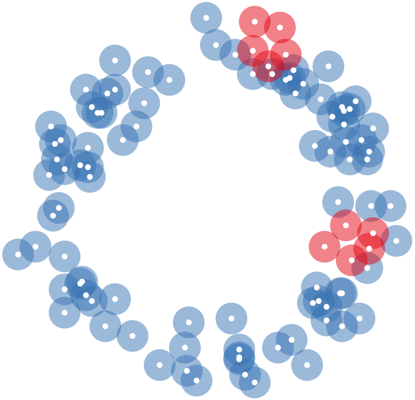

?

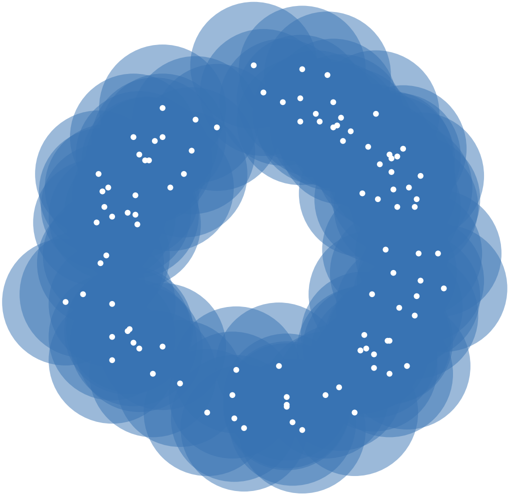
$\cong$

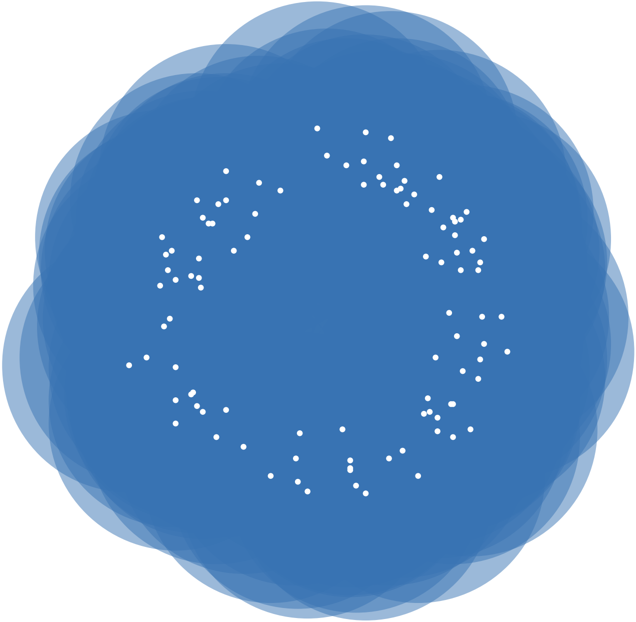

?

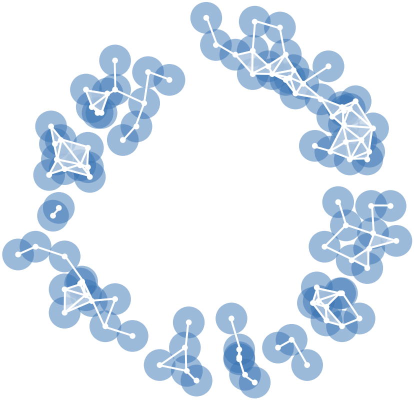


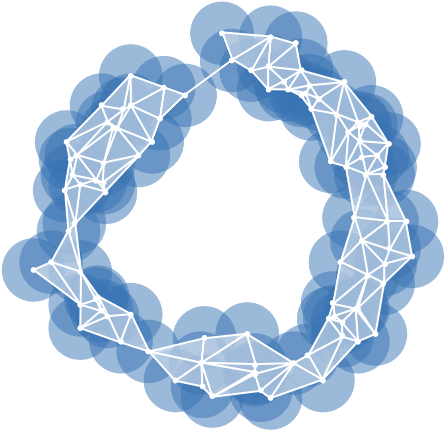


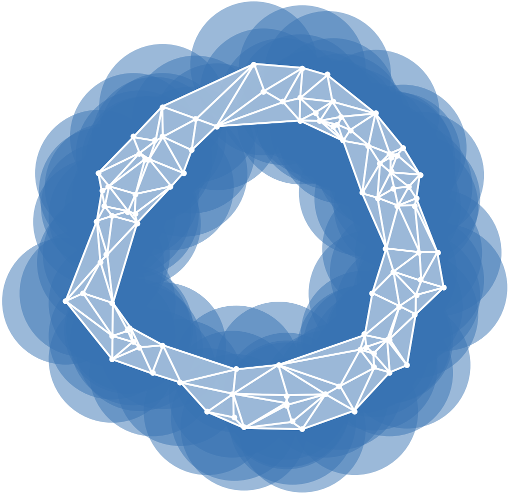


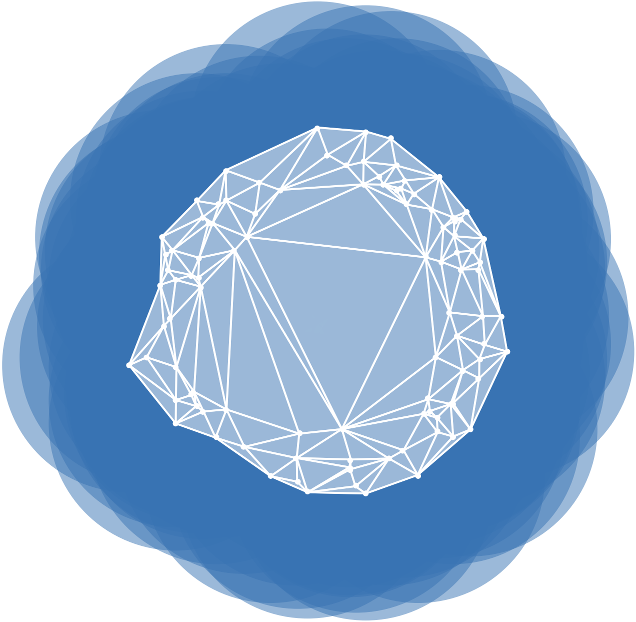


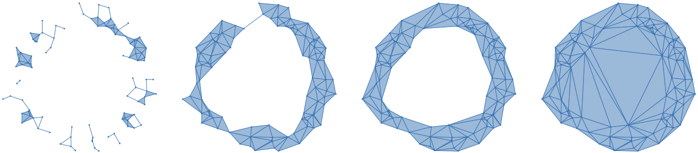


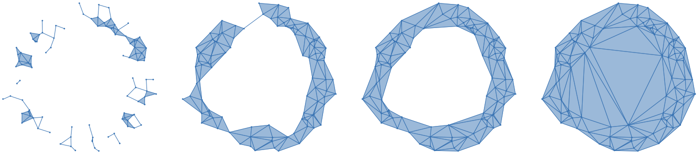


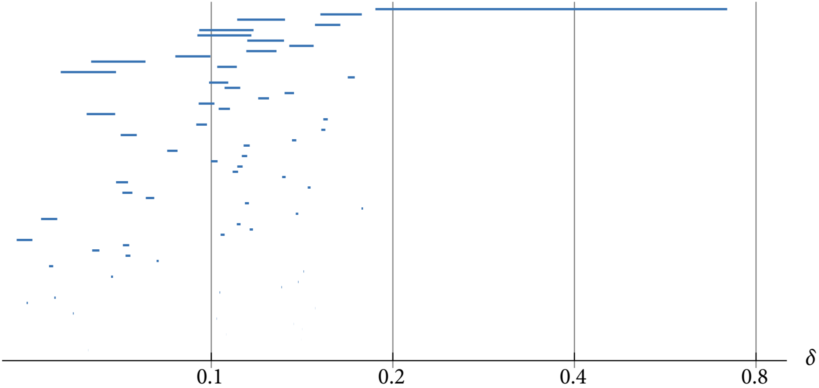


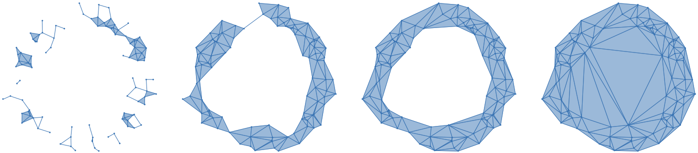


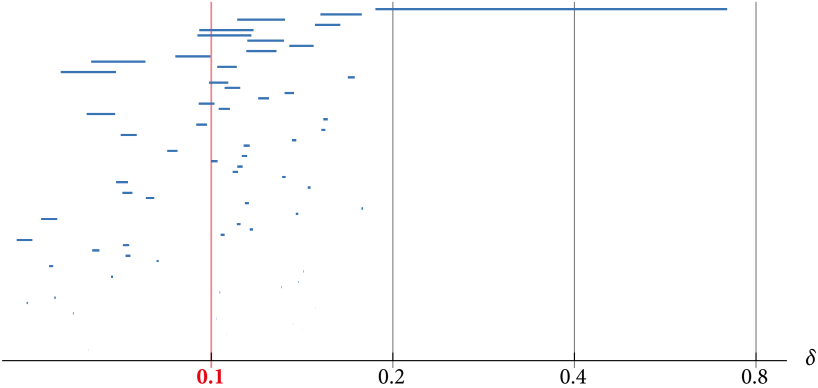


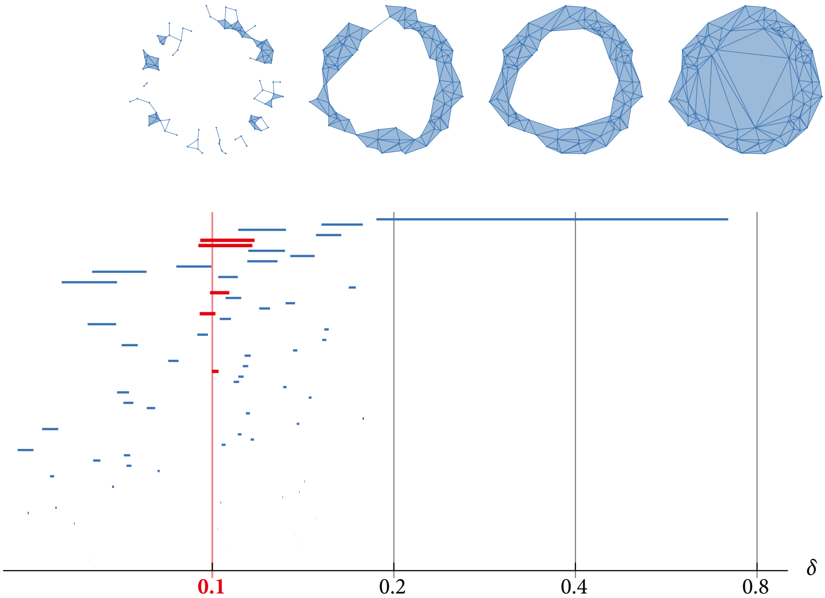


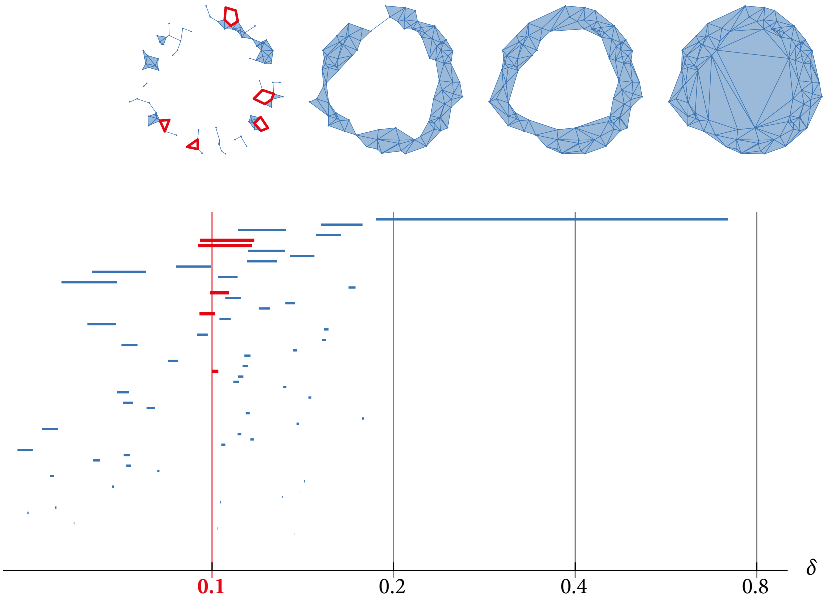


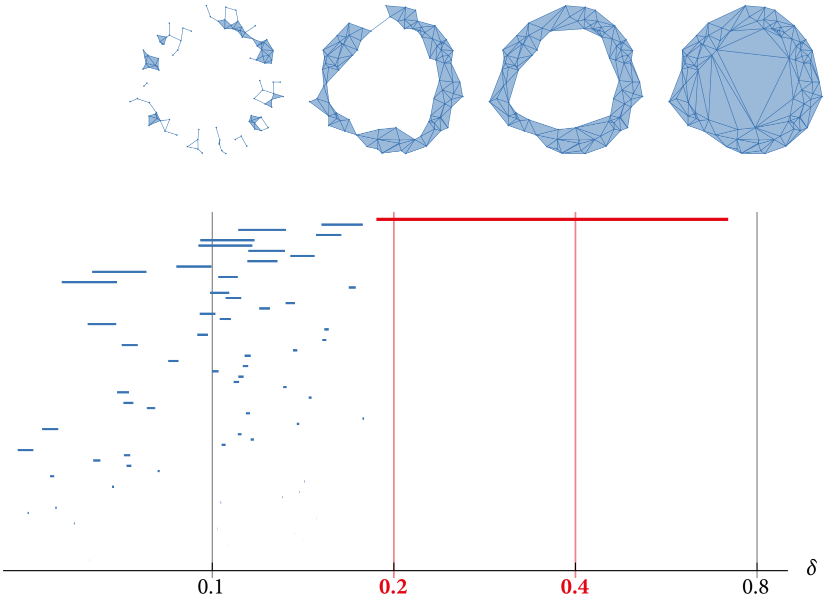

δ
0.1
0.2
0.4
0.8



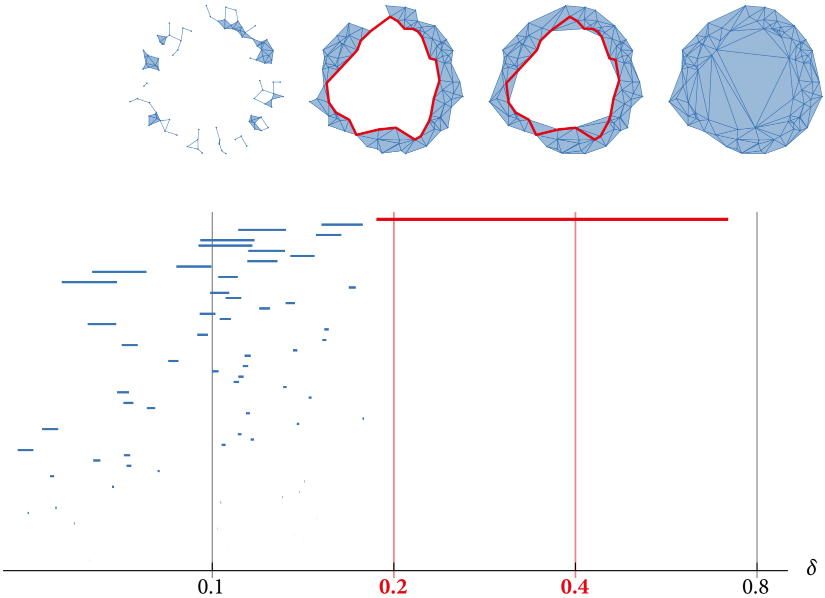

δ
0.1
0.2
0.4
0.8


δ
0.1
0.2
0.4
0.8


δ
0.1
0.2
0.4
0.8

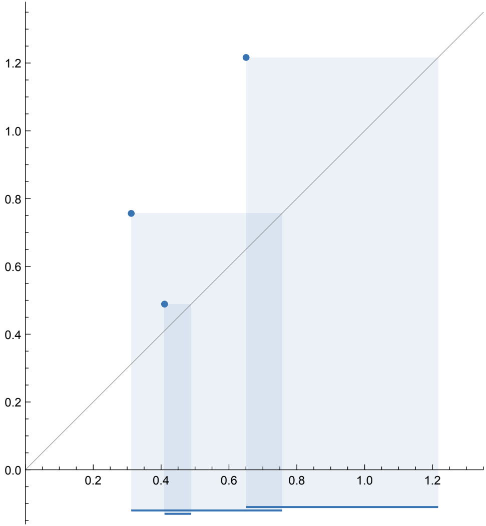


δ
0.1
0.2
0.4
0.8



δ
0.1
0.2
0.4
0.8

# Persistence barcodes and persistence diagrams



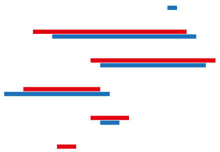
# The pipeline of topological data analysis

$$
\begin{tikzcd}
\text{Data} \arrow[d] & \text{point cloud} \arrow[d, "\text{distance}"] \\
\text{Geometry} \arrow[d] & \text{function} \arrow[d, "\text{sublevel sets}"] \\
\text{Topology} \arrow[d] & \text{topological spaces} \arrow[d, "\text{homology}"] \\
\text{Algebra} \arrow[d] & \text{vector spaces} \arrow[d, "\text{barcode}"] \\
\text{Combinatorics} & \text{intervals}
\end{tikzcd}
$$

# Stability of persistence barcodes for functions

## Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

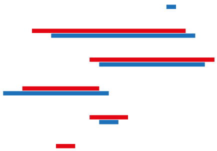
*If two functions $f, g : K \to \mathbb{R}$ have distance $\|f - g\|_\infty \leq \delta$, then there exists a $\delta$-matching of their barcodes.*



# Stability of persistence barcodes for functions

## Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

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

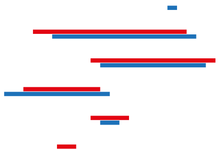


- *matching* of sets $X, Y$: bijection of subsets $X' \subseteq X$, $Y' \subseteq Y$

# Stability of persistence barcodes for functions

## Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

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

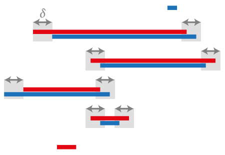


- *matching* of sets $X, Y$: bijection of subsets $X' \subseteq X$, $Y' \subseteq Y$
- *$\delta$-matching of barcodes*:

# Stability of persistence barcodes for functions

## Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

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

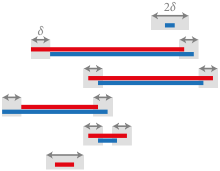



- *matching* of sets $X, Y$: bijection of subsets $X' \subseteq X$, $Y' \subseteq Y$
- *$\delta$-matching of barcodes*:
  - matched intervals have endpoints within distance $\leq \delta$

# Stability of persistence barcodes for functions

## Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

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

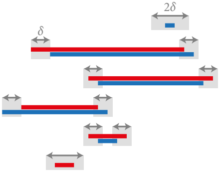



- *matching* of sets $X, Y$: bijection of subsets $X' \subseteq X$, $Y' \subseteq Y$
- *$\delta$-matching of barcodes*:
  - matched intervals have endpoints within distance $\leq \delta$
  - unmatched intervals have length $\leq 2\delta$

# Stability of persistence barcodes for functions

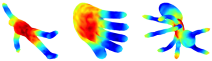
## Theorem (Cohen-Steiner, Edelsbrunner, Harer 2005)

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



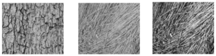

- *matching* of sets $X, Y$: bijection of subsets $X' \subseteq X$, $Y' \subseteq Y$
- *$\delta$-matching of barcodes*:
  - matched intervals have endpoints within distance $\leq \delta$
  - unmatched intervals have length $\leq 2\delta$
- *Bottleneck distance $d_\infty$*: infimum $\delta$ admitting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

# Topological machine learning

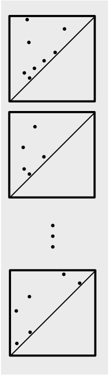


**Task**: shape retrieval

**Task**: object recognition

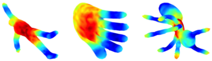


**Task**: texture recognition



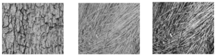
Topological data analysis

# Topological machine learning

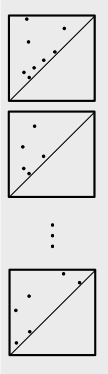


**Task**: shape retrieval

**Task**: object recognition



**Task**: texture recognition



SVM

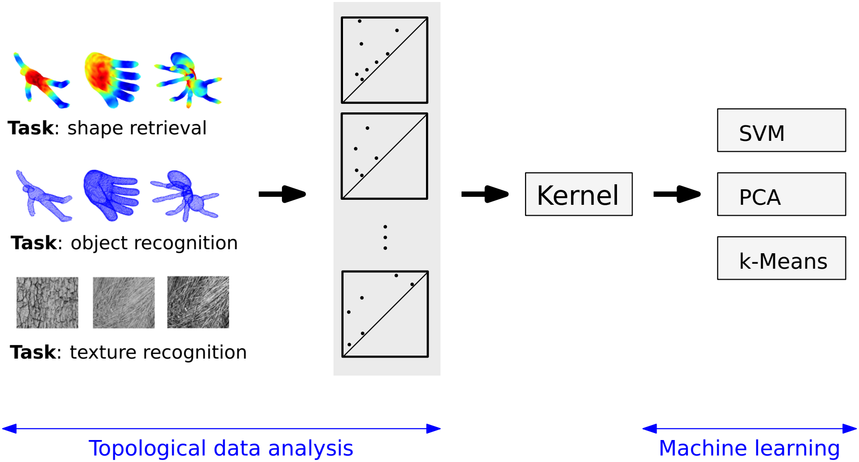
PCA

k-Means

Topological data analysis

Machine learning

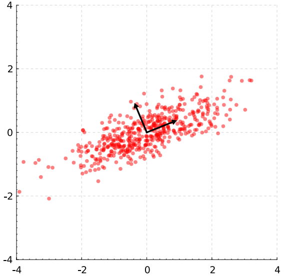
# Topological machine learning



# Linear machine learning methods

Many machine learning methods solve linear geometric problems on vector spaces

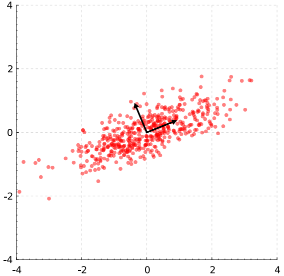
- Principal component analysis (PCA): find orthogonal directions of largest variance

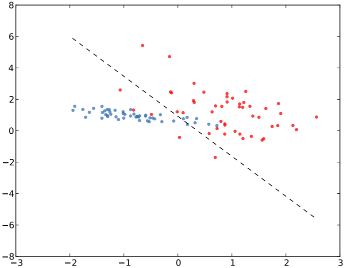


# Linear machine learning methods

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 $\Phi : X \to \mathcal{H}$
- Then

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

  is a *kernel* (and $\Phi$ 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 $\Phi : X \to \mathcal{H}$
- Then

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

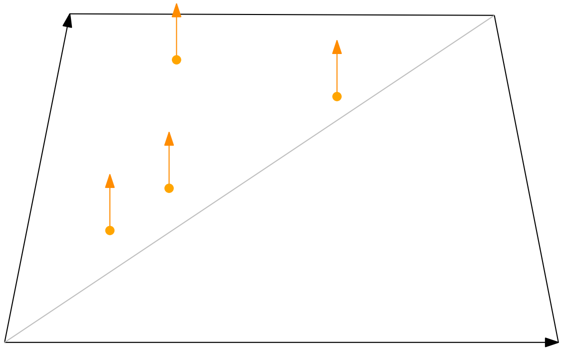
  is a *kernel* (and $\Phi$ is the associated *feature map*)

For many linear methods:

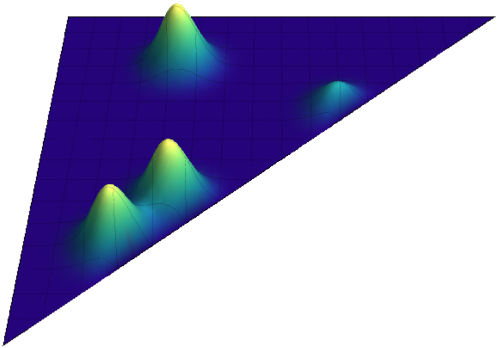
- Feature map $\Phi$ not required explicitly
- No explicit basis for $\mathcal{H}$ required
- Computations can be performed by evaluating $k(\cdot, \cdot)$

This is called the *kernel trick*.

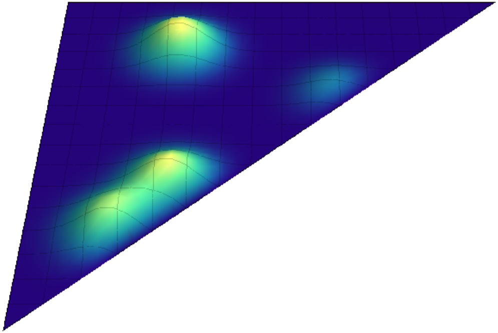
# Persistence scale space kernel



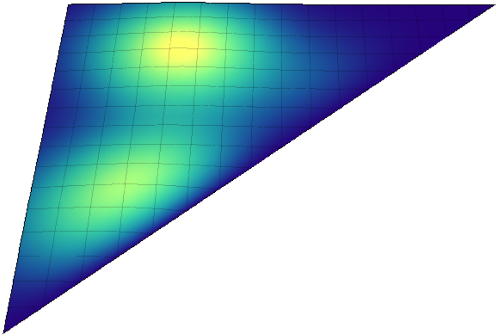
# Persistence scale space kernel



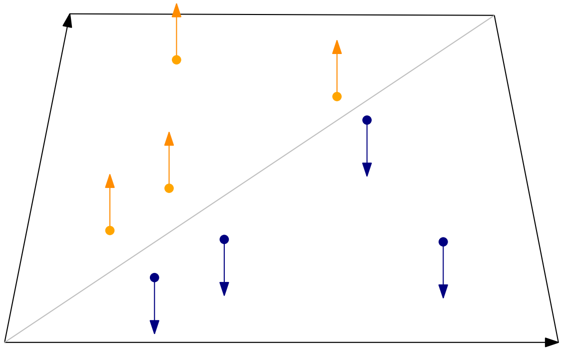
# Persistence scale space kernel



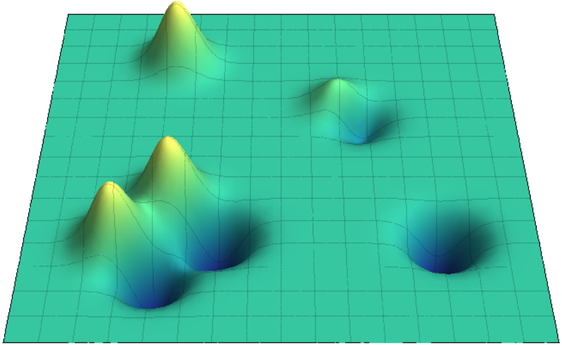
# Persistence scale space kernel



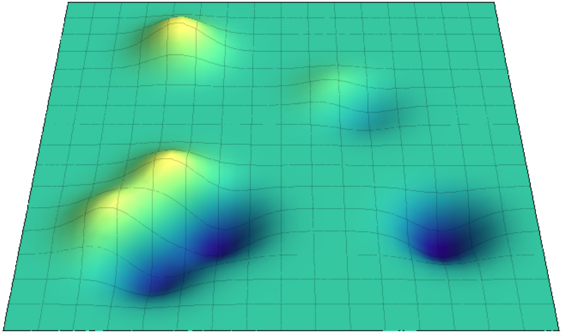
# Persistence scale space kernel



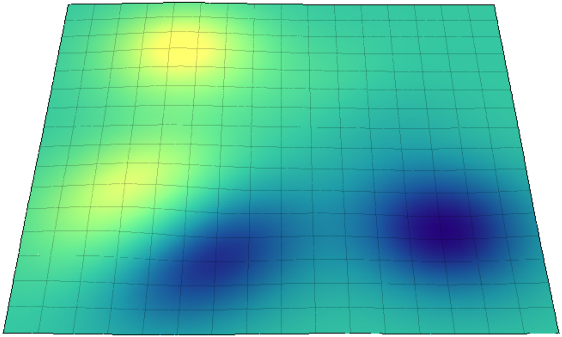
# Persistence scale space kernel



# Persistence scale space kernel



# Persistence scale space kernel



# 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

# Metric distortion of the PSS kernel

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

## Theorem (Reininghaus, Huber, B, Kwitt 2015)

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

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

# Metric distortion of the PSS kernel

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

## Theorem (Reininghaus, Huber, B, Kwitt 2015)

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

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

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

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

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

# Sliced Wasserstein distance and kermel

Another construction: the *sliced Wasserstein distance* $d_{\mathrm{SW}}$ between persistence diagrams.

## Theorem (Carrière, Cuturi, Oudot 2017)

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

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

# Sliced Wasserstein distance and kermel

Another construction: the *sliced Wasserstein distance* $d_{\mathrm{SW}}$ between persistence diagrams.

## Theorem (Carrière, Cuturi, Oudot 2017)

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

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

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

# Conditional bounds

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

# Conditional bounds

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 $\infty$ if either bound on the number or range of bars go to $\infty$.*

# Properties of the space of persistence diagrams

Known properties:

- The space of persistence diagrams (with $p$-distance, $p < \infty$) is complete and separable [Mileyko et al. 2011]
- There are no upper bounds on curvature [Turner et al. 2014]

These properties provide no immediate obstructions to bi-Lipschitz embeddings. But:

# 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 $\frac{r}{2}$.

# 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.*

# Persistence spaces and doubling spaces

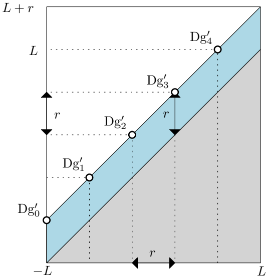
## Definition

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 (with $p$-distance) is not doubling.*

Thus, there is no bi-Lipschitz map to Euclidean space.

# Mapping from non-compact to compact?

What about embedding into Hilbert space?

- Consider the set $X = \{\mathrm{Dg}_n\}_n$ of persistence diagrams, each $\mathrm{Dg}_n$ having one interval of length $n$.
- Then $X$ is non-compact and metrically complete.

## Lemma

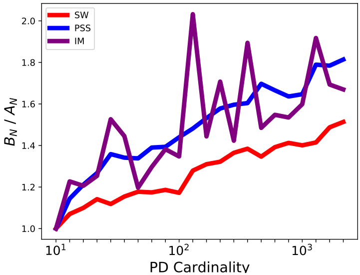
*Assume that there was a bi-Lipschitz map from persistence diagrams to some Hilbert space.*
*Such a map would send $X$ to a compact set.*

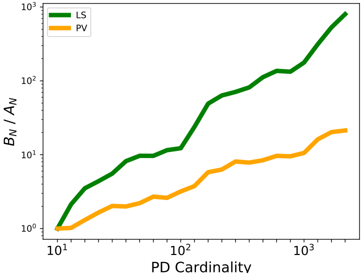
But this would contradict the fact that $X$ is not compact.

## Corollary

*If there is a bi-Lipschitz map from a bounded subset of persistence diagrams, the Lipschitz constants have to go to $\infty$ together with the bounds.*

# Examples

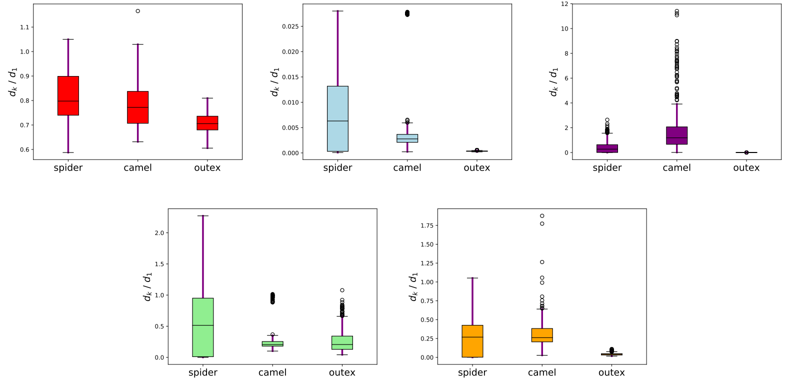



Empirical metric distortion bounds for persistence diagrams of different cardinalities (uniformly sampled points in upper half-square).

Kernels used: Sliced Wasserstein (red), Persistence Scale Space (blue), Persistence Images (purple); Persistence Landscapes (green), Persistence Vectors (orange)

# Examples




Empirical distribution of metric distortion ratios, for image (*spider*), shape (*camel*), and texture (*outex*) data sets.

Kernels used: Sliced Wasserstein (red), Persistence Scale Space (blue), Persistence Images (purple); Persistence Landscapes (green), Persistence Vectors (orange)