Toward multi-parameter persistence for ML

David Loiseaux

PhD supervisors : Mathieu Carrière, Frédéric Cazals

3IA Chair : Topological and Geometrical Data Analysis
3IA Chair holder : Jean-Daniel Boissonnat

Inria

3iA Côte d'Azur
Interdisciplinary Institute
for Artificial Intelligence

Curse of Dimension

In high dimensional spaces,

- Gaussian measures are grossly *uniform laws on a sphere*,

Curse of Dimension

In high dimensional spaces,

- Gaussian measures are grossly *uniform laws on a sphere*,
- Distances *tends to infinity* w.r.t. the dimension,

Curse of Dimension

In high dimensional spaces,

- Gaussian measures are grossly *uniform laws on a sphere*,
- Distances *tends to infinity* w.r.t. the dimension,
- Volumes of objects *concentrates in their crust*,
- ...

Curse of Dimension

In high dimensional spaces,

- Gaussian measures are grossly *uniform laws on a sphere*,
- Distances *tends to infinity* w.r.t. the dimension,
- Volumes of objects *concentrates in their crust*,
- ...

$\longrightarrow$ Building statistics *directly* from such spaces is thus not really an option, without tricks to *reduce their variance*.

Variance reduction technics

Examples on images :

- Locally destroy information : convolutions / resolution reduction

Variance reduction technics

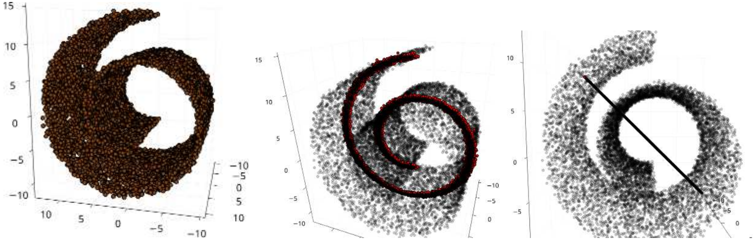
Examples on images :

- Locally destroy information : convolutions / resolution reduction
- Projection on a smaller linear space : PCA

Variance reduction technics

Examples on images :

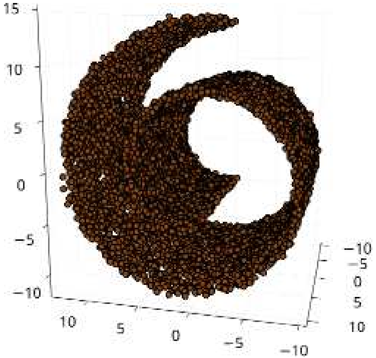
- Locally destroy information : convolutions / resolution reduction
- Projection on a smaller linear space : PCA

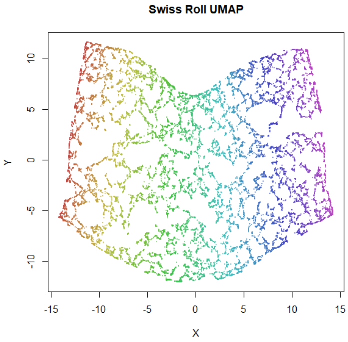


Variance reduction technics

Examples on images :

- Locally destroy information : convolutions / resolution reduction
- Projection on a smaller linear space : PCA
- Projection on a smaller space : Manifold Learning (UMAP, ISOMAP)





Variance reduction technics

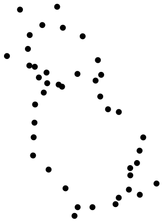
Examples on images :

- Locally destroy information : convolutions / resolution reduction
- Projection on a smaller linear space : PCA
- Projection on a smaller space : Manifold Learning (UMAP, ISOMAP)

Can we “forget” more ?
With weaker assumptions / more guarantees ?

Forget more geometry, keep the topology!

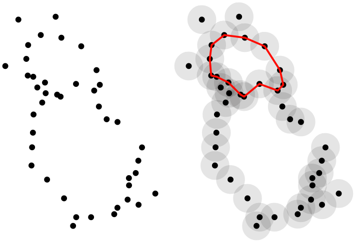
Persistent Homology in a nutshell



$\mathcal{K}_1$

Forget more geometry, keep the topology!

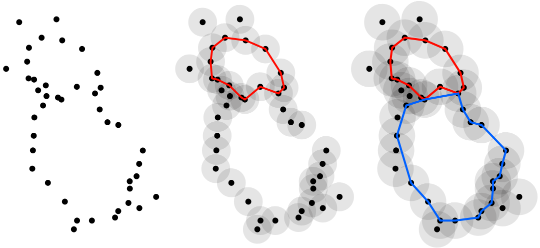
Persistent Homology in a nutshell



$$\mathcal{K}_1 \overset{\iota}{\hookrightarrow} \mathcal{K}_2$$

Forget more geometry, keep the topology!

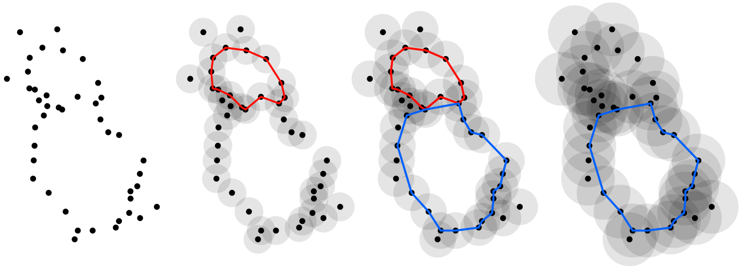
Persistent Homology in a nutshell



$$\mathcal{K}_1 \overset{\iota}{\hookrightarrow} \mathcal{K}_2 \overset{\iota}{\hookrightarrow} \mathcal{K}_3$$

Forget more geometry, keep the topology!

Persistent Homology in a nutshell

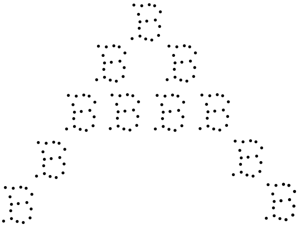


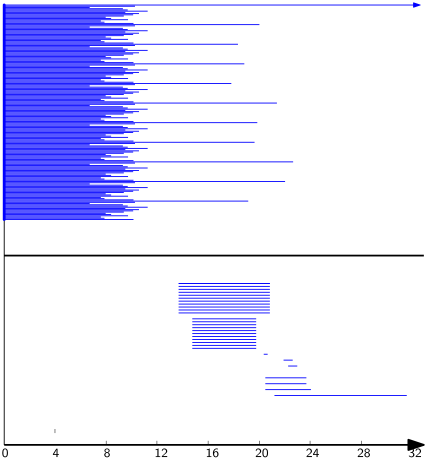
$$\mathcal{K}_1 \overset{\iota}{\hookrightarrow} \mathcal{K}_2 \overset{\iota}{\hookrightarrow} \mathcal{K}_3 \overset{\iota}{\hookrightarrow} \mathcal{K}_4$$

Forget more geometry, keep the topology!

Persistent Homology in a nutshell

$$f_P : \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

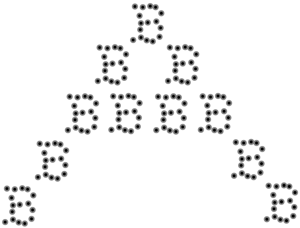


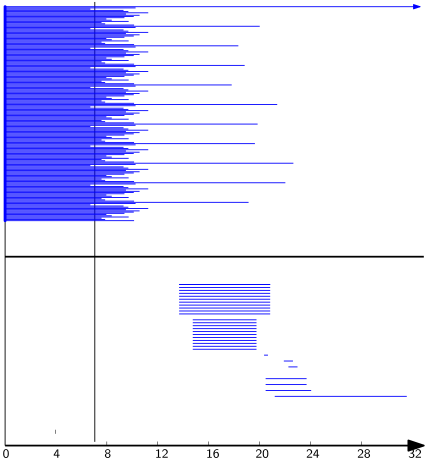


Forget more geometry, keep the topology!

Persistent Homology in a nutshell

$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

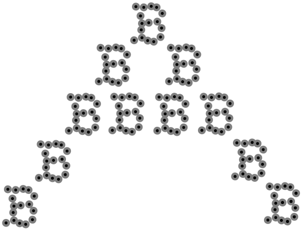


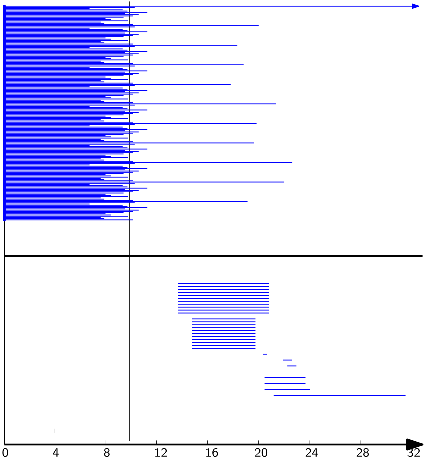


Forget more geometry, keep the topology!

Persistent Homology in a nutshell

$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

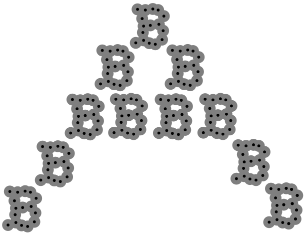


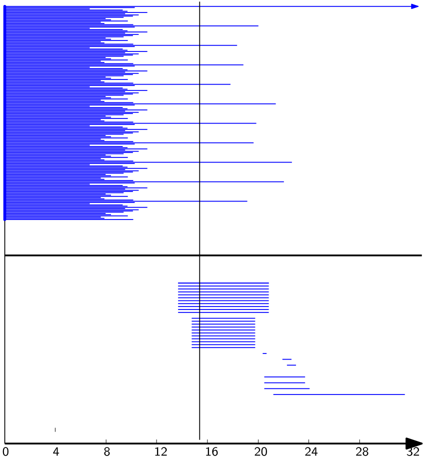


Forget more geometry, keep the topology!

Persistent Homology in a nutshell

$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

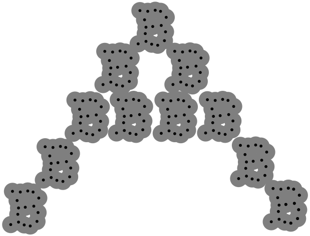


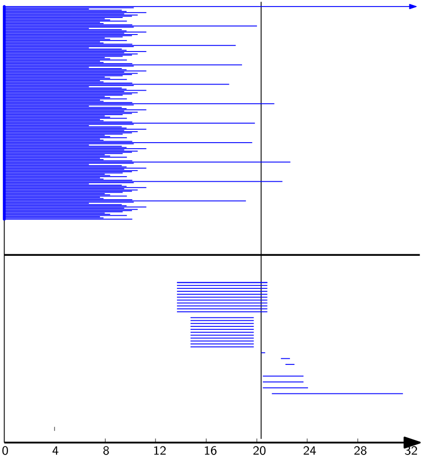


Forget more geometry, keep the topology!

Persistent Homology in a nutshell

$$f_P : \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

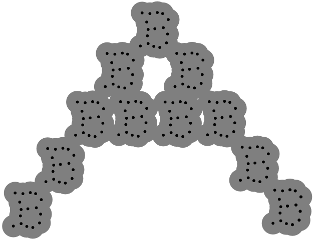


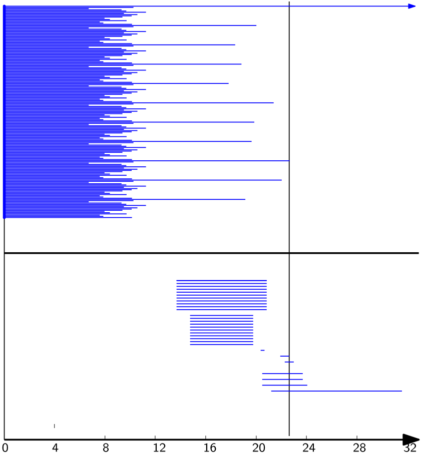


Forget more geometry, keep the topology!

Persistent Homology in a nutshell

$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

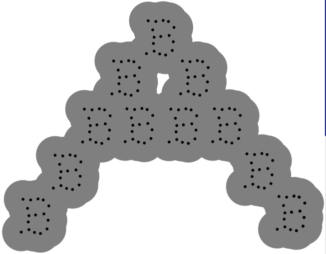


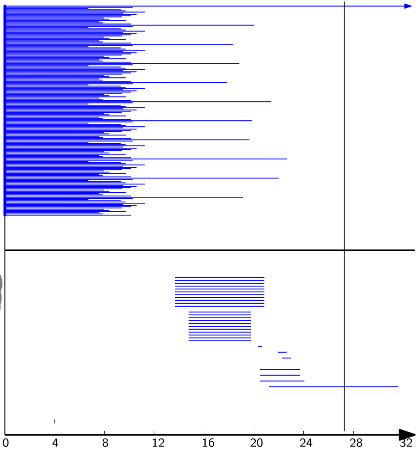


Forget more geometry, keep the topology!

Persistent Homology in a nutshell

$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

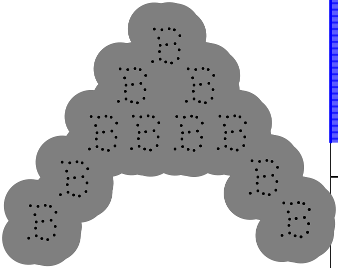


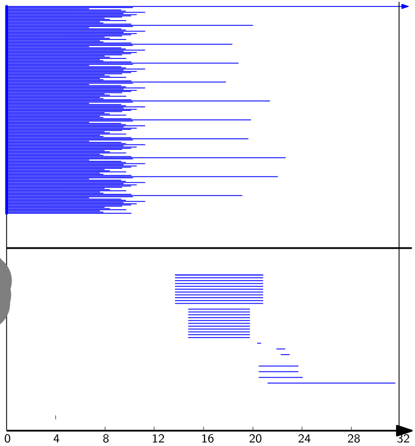


Forget more geometry, keep the topology!

Persistent Homology in a nutshell

$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

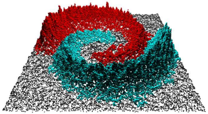




Position in statistical learning

Advantages:

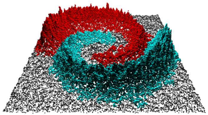
- *Computable*, and we know what we compute,



Position in statistical learning

Advantages:

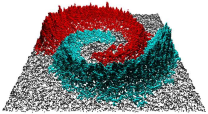
- *Computable*, and we know what we compute,
- This only cares about *intrinsic dimensions*,



Position in statistical learning

Advantages:

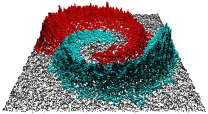
- *Computable*, and we know what we compute,
- This only cares about *intrinsic dimensions*,
- *Compact, dense* in information,



Position in statistical learning

Advantages:

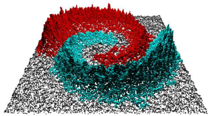
- *Computable*, and we know what we compute,
- This only cares about *intrinsic dimensions*,
- *Compact, dense* in information,
- *Stable* w.r.t. the input,



Position in statistical learning

Advantages:

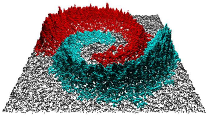
- *Computable*, and we know what we compute,
- This only cares about *intrinsic dimensions*,
- *Compact, dense* in information,
- *Stable* w.r.t. the input,
- *All scales* at once,



Position in statistical learning

Advantages:

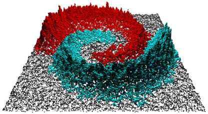
- *Computable*, and we know what we compute,
- This only cares about *intrinsic dimensions*,
- *Compact, dense* in information,
- *Stable* w.r.t. the input,
- *All scales* at once,
- Multiple properties can be *translated to* topological features (e.g. time series),



Position in statistical learning

Advantages:

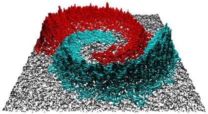
- *Computable*, and we know what we compute,
- This only cares about *intrinsic dimensions*,
- *Compact, dense* in information,
- *Stable* w.r.t. the input,
- *All scales* at once,
- Multiple properties can be *translated to* topological features (e.g. time series),
- Various representations and tools available for ML.

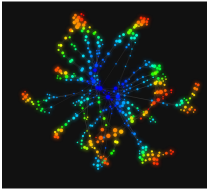


Position in statistical learning

Advantages:

- *Computable*, and we know what we compute,
- This only cares about *intrinsic dimensions*,
- *Compact, dense* in information,
- *Stable* w.r.t. the input,
- *All scales* at once,
- Multiple properties can be *translated to* topological features (e.g. time series),
- Various representations and tools available for ML.





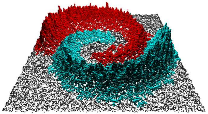
Disadvantages:

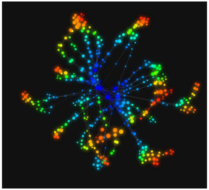
- "Only" computes topological information,

Position in statistical learning

Advantages:

- *Computable*, and we know what we compute,
- This only cares about *intrinsic dimensions*,
- *Compact, dense* in information,
- *Stable* w.r.t. the input,
- *All scales* at once,
- Multiple properties can be *translated to* topological features (e.g. time series),
- Various representations and tools available for ML.





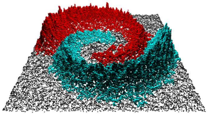
Disadvantages:

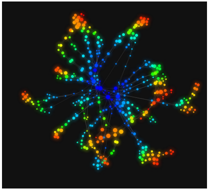
- "Only" computes topological information,
- (Can be) sensible to outliers,

Position in statistical learning

Advantages:

- *Computable*, and we know what we compute,
- This only cares about *intrinsic dimensions*,
- *Compact, dense* in information,
- *Stable* w.r.t. the input,
- *All scales* at once,
- Multiple properties can be *translated to* topological features (e.g. time series),
- Various representations and tools available for ML.

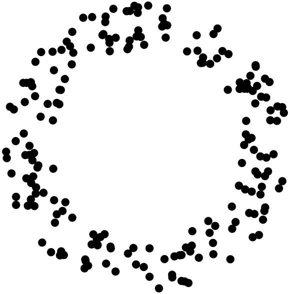




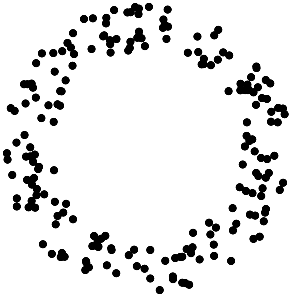
Disadvantages:

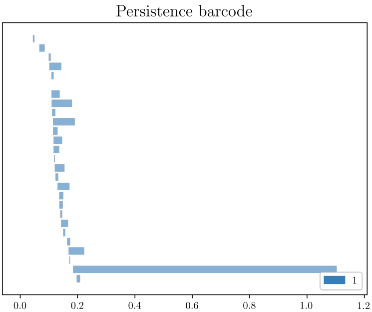
- “Only” computes topological information,
- (Can be) sensible to outliers,
- Can only consider one parameter.

Persistence and noise

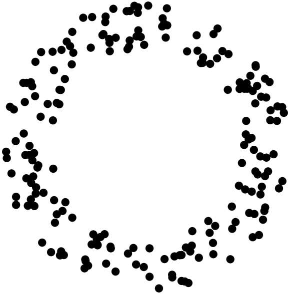


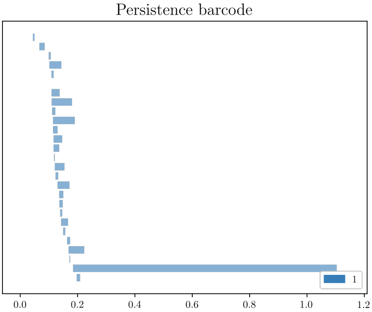
Persistence and noise





Persistence and noise

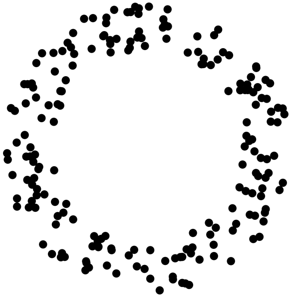


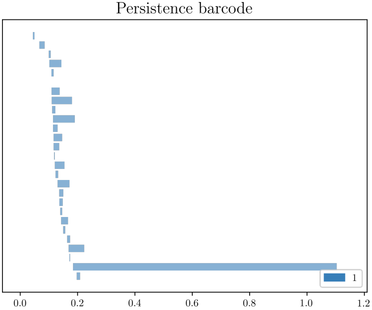


Theorem [Cohen-Steiner, et al.] Given two *tame* spaces X, Y,

$$d_I(\text{Persistence}(X), \text{Persistence}(Y)) = d_b(\text{dgm}(X), \text{dgm}(Y)) \leq 2d_{\text{GH}}(X, Y).$$

Persistence and noise

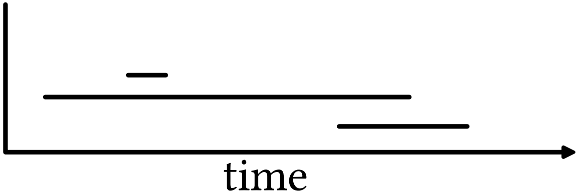




Theorem [Chazal, et al.] Given an (a, b)-standard measure μ of support X_μ, and an n-sampling X_n of μ, we have, for any $\varepsilon > 0$,

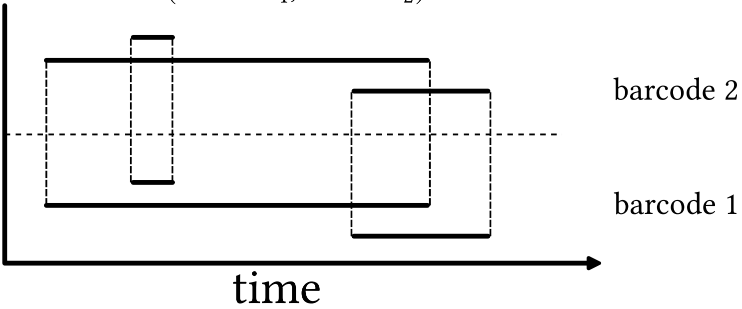
$$\mathbb{P}(d_H(X_\mu, X_n) > \varepsilon) \leq \frac{2^b}{a\varepsilon^b} e^{-na\varepsilon^b}.$$

Bottleneck distance



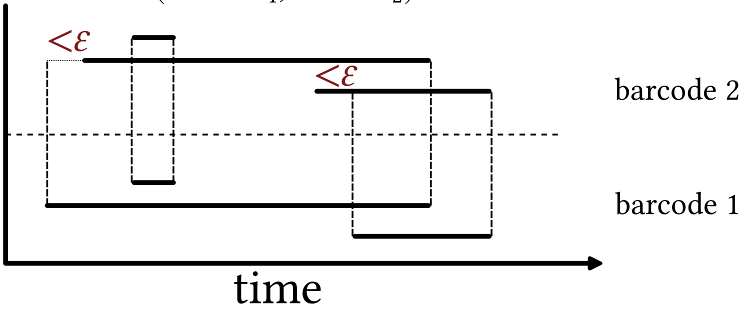
Bottleneck distance

$$d(\text{barcode}_1, \text{barcode}_2) = 0$$

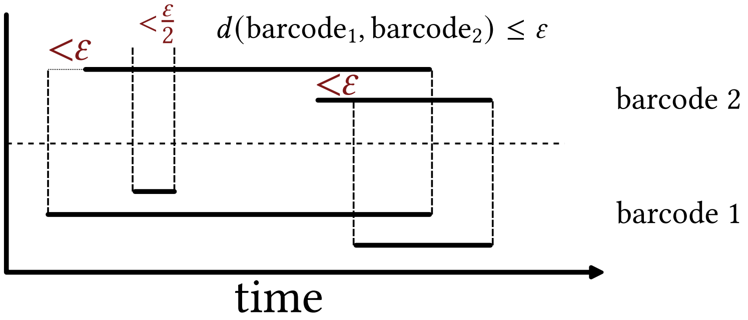


Bottleneck distance

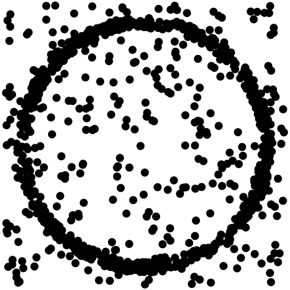
$$d(\text{barcode}_1, \text{barcode}_2) \leq \varepsilon$$

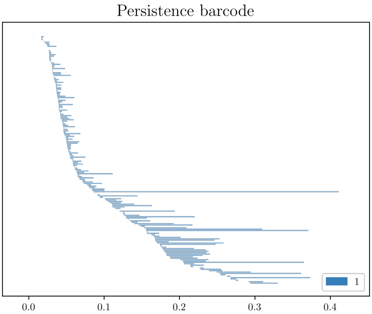


Bottleneck distance

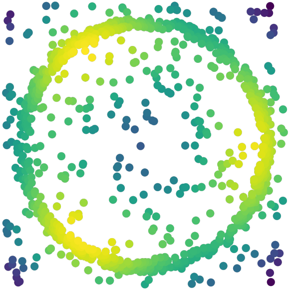


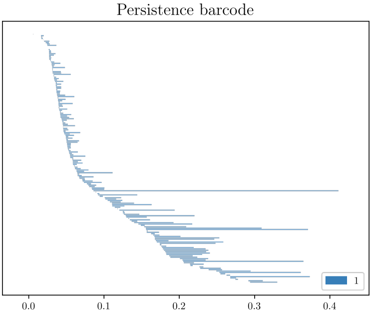
Persistence and noise



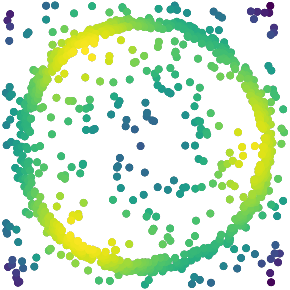


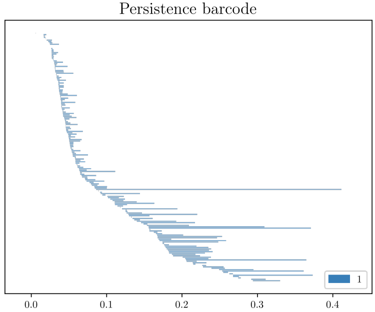
Persistence and noise





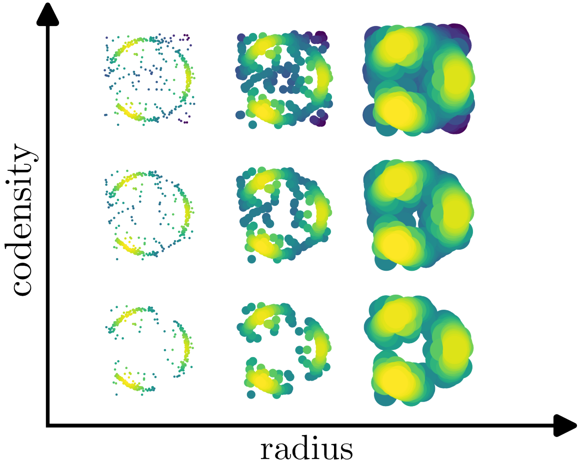
Persistence and noise



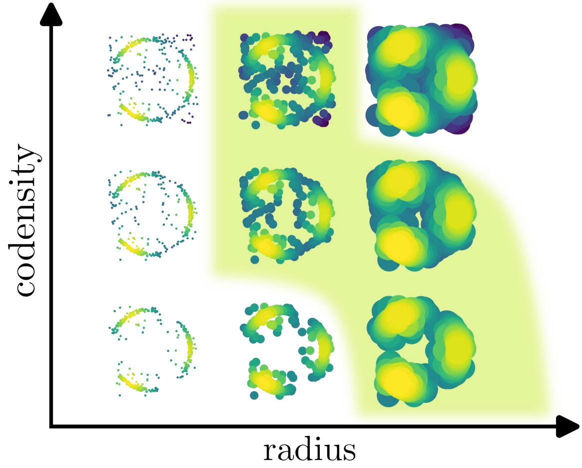


⟶ Threshold ?

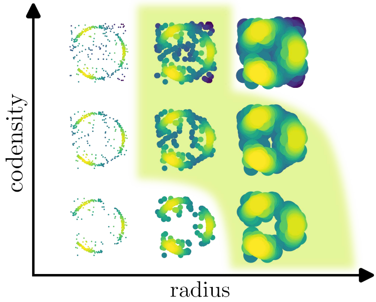
First example

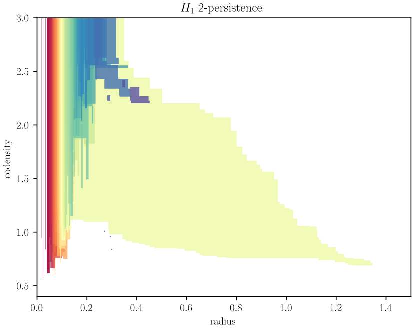


First example

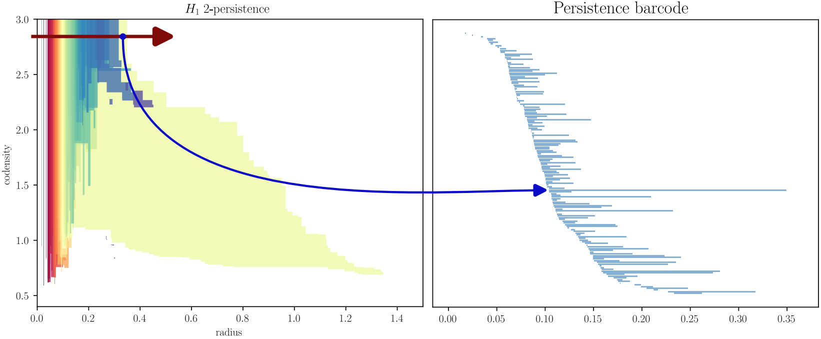


First example

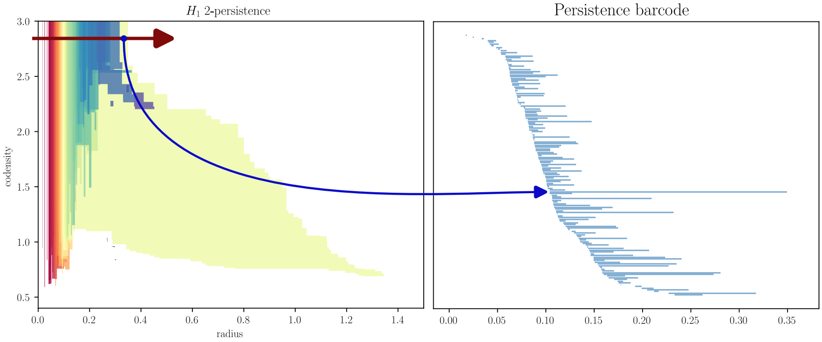




First example



First example



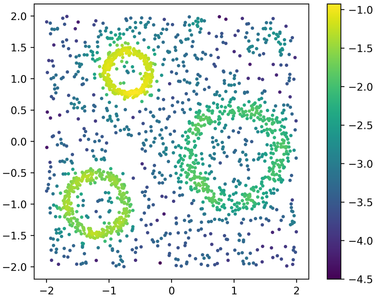
Theorem.[L., Blumberg, Carrière]

If the true multiparameter persistence module M is *simple enough*, this representation M^δ is a *δ-approximation*, i.e.,

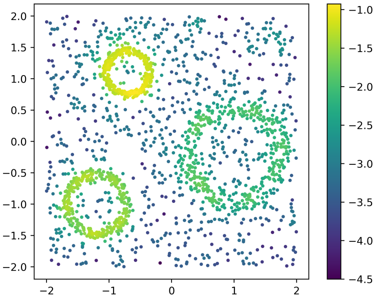
$$d_{\text{matching}}(M^\delta, M) \leq d_{\text{interleaving}}(M^\delta, M) \leq d_{\text{bottleneck}}(M^\delta, M) \leq \delta.$$

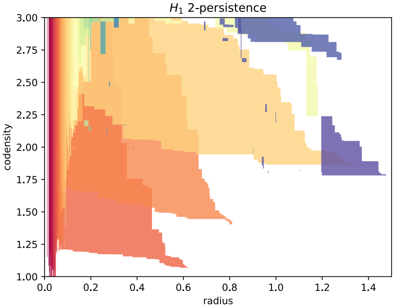
And is exact on *discrete data*, if δ is *small enough*, i.e. $d_{\text{bottleneck}}(M^\delta, M) = 0$.

Second example

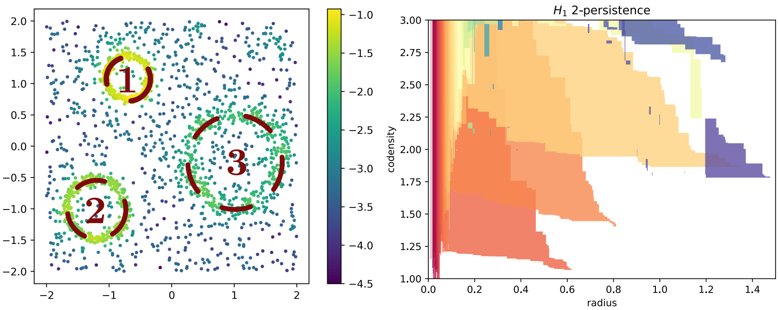


Second example

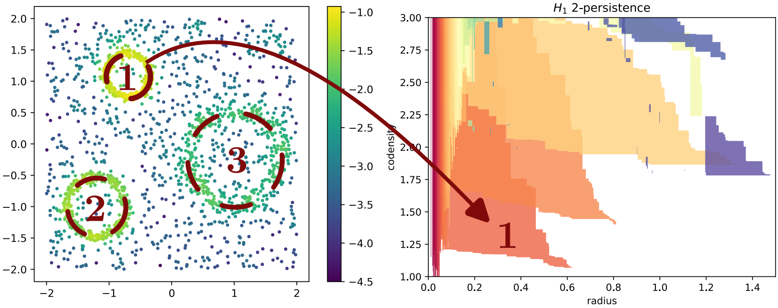




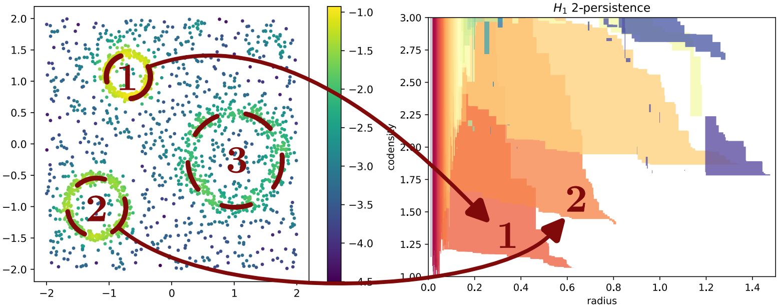
Second example



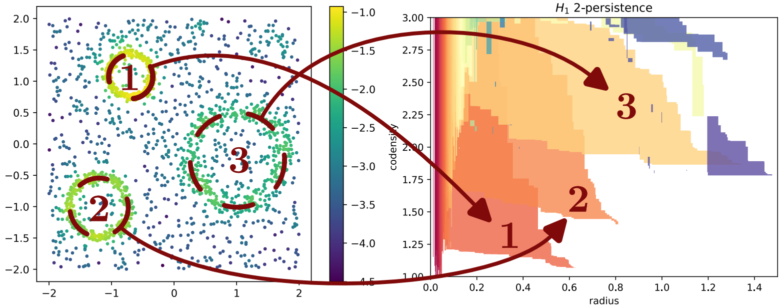
Second example



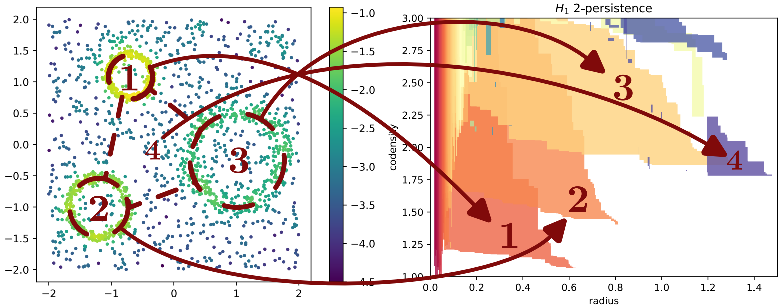
Second example



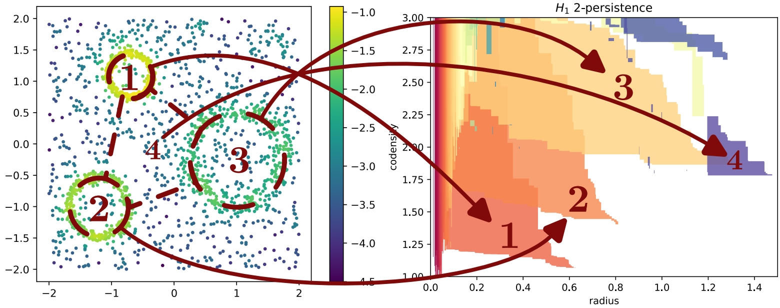
Second example



Second example

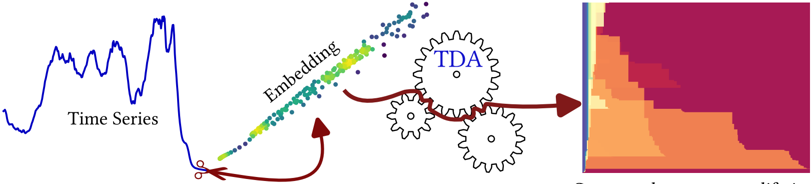


Second example



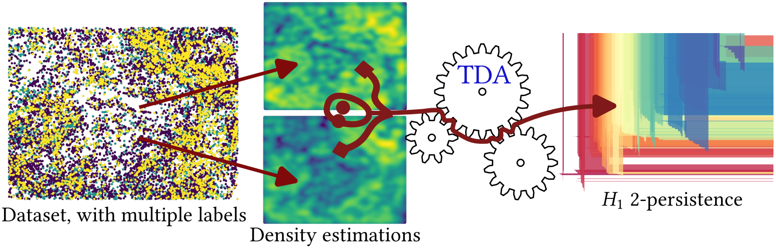
All scales at once, *and at **every** concentrations !*

More applications



Connected components lifetime

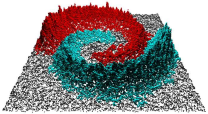
More applications

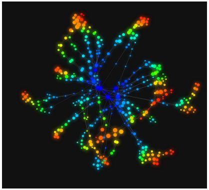


Position in statistical learning

Advantages:

- Computable, and we know what we compute,
- This only cares about *intrinsic dimensions*,
- *Compact, dense* in information,
- *Stable* w.r.t. the input,
- *All scales* at once, with *multiple scales*
- Multiple properties can be *translated to* topological features (e.g. time series)

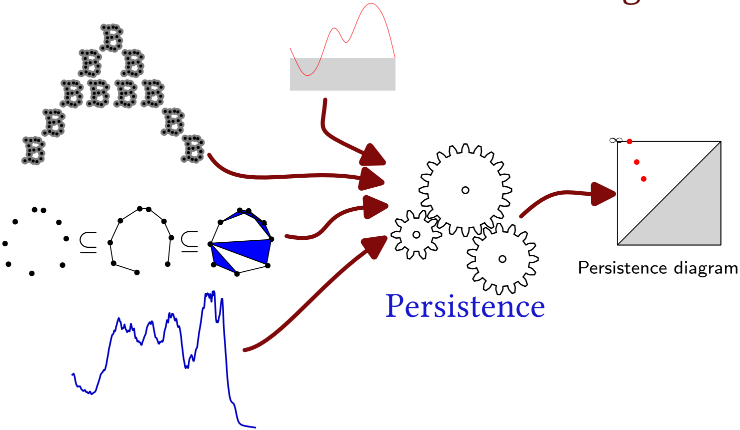




Disadvantages:

- "Only" computes topological information,
- Involved theoretical objects,
- Still a challenge to compute it,
- Room for improvement on the ML side.

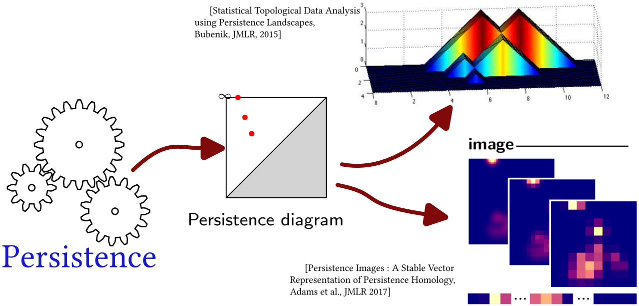
Persistence and machine learning



Persistence and machine learning

A few persistence transformations

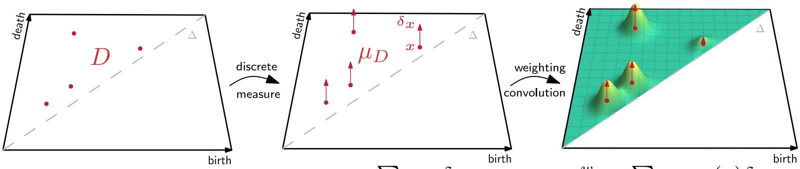
- Vectorizations,



Persistence and machine learning

A few persistence transformations

- Vectorizations,
- Optimal transport,



$\mu_{\mathrm{D}} := \sum_{x \in \mathrm{D}} \delta_x$

$\mu_{\mathrm{D}}^{w} := \sum_{x \in \mathrm{D}} w(x)\delta_{\bar{x}}$

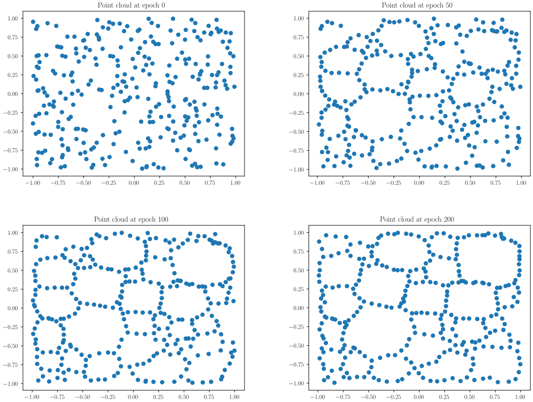
$\tilde{\mu}_{\mathrm{D}}^{w} := \mu_{\mathrm{D}}^{w} * \mathcal{N}(0, \sigma)$

[*Persistence weighted Gaussian kernel for topological data analysis*, Kisano, Hiraoka, Fukumizu, ICML, 2016]

Persistence and machine learning

A few persistence transformations

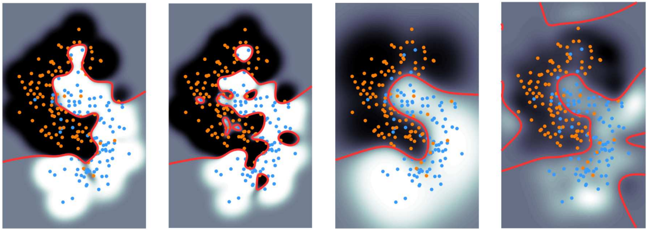
- Vectorizations,
- Optimal transport,
- Optimization, Regularization,



Persistence and machine learning

A few persistence transformations

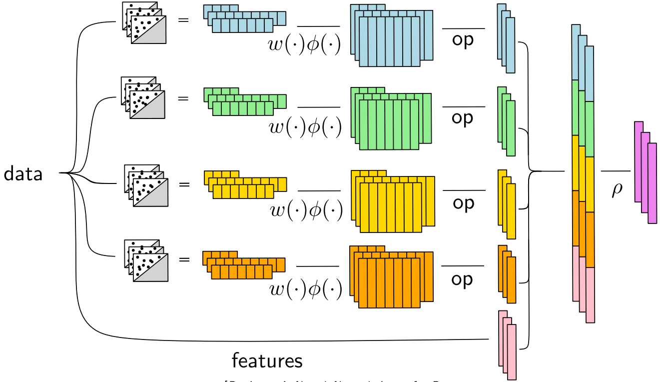
- Vectorizations,
- Optimal transport,
- Optimization, Regularization,



Persistence and machine learning

A few persistence transformations

- Vectorizations,
- Optimal transport,
- Optimization, Regularization,
- Deep Learning,

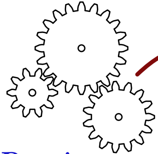


[*PersLay: A Neural Network Layer for Persistence Diagrams and New Graph Topological Signatures*, Carrière, Chazal, Ike, Lacombe, Royer, Umeda, AISTATS, 2019]

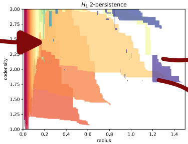
Multiparameter persistence and machine learning

A few persistence transformations

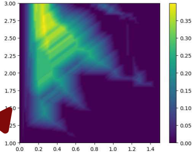
- Vectorizations,



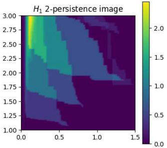
Persistence



[Multiparameter persistence landscapes Vipond, JMLR 2020]



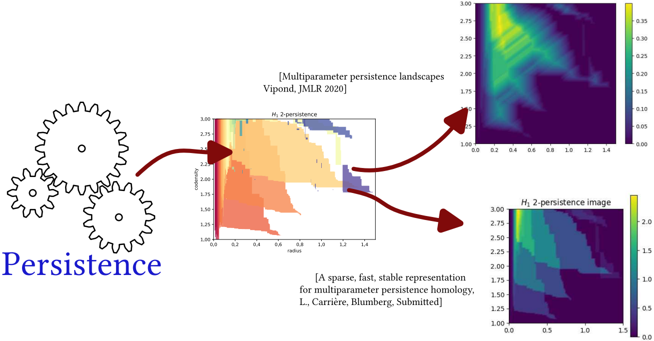
[A sparse, fast, stable representation for multiparameter persistence homology, L., Carrière, Blumberg, Submitted]



Multiparameter persistence and machine learning

A few persistence transformations

- Vectorizations,



Theorem [L., Blumberg, Carrière]

Given two n-persistence (interval decomposable) modules M, N, this vectorization V satisfies

$$\|V(M) - V(N)\|_\infty \lesssim d_{\text{bottleneck}}(M, N).$$

Multiparameter persistence and machine learning

A few persistence transformations

- Vectorizations,

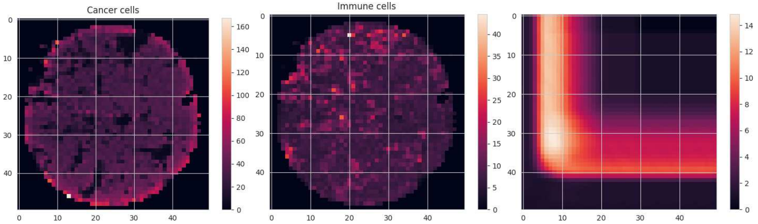


Figure: [Multiparameter Persistence Image for Topological Machine Learning, Carrière, Blumberg, NeurIPS 2020]

Multiparameter persistence and machine learning

A few persistence transformations

- Vectorizations,
- Optimal transport, *TODO*

Multiparameter persistence and machine learning

A few persistence transformations

- Vectorizations,
- Optimal transport, *TODO*
- Optimization, *TODO*

Multiparameter persistence and machine learning

A few persistence transformations

- Vectorizations,
- Optimal transport, *TODO*
- Optimization, *TODO*
- Deep Learning, *TODO*

Contributions

Papers :

- [Approximation of multiparameter persistence modules, L., Carrière, Blumberg] - On ArXiV, submitted.

Contributions

Papers :

- [Approximation of multiparameter persistence modules, L., Carrière, Blumberg] - On ArXiV, submitted.
- [A sparse, fast, stable representation for multiparameter persistence homology, L., Carrière, Blumberg] - submitted.

Contributions

Papers :

- [Approximation of multiparameter persistence modules, L., Carrière, Blumberg] - On ArXiV, submitted.
- [A sparse, fast, stable representation for multiparameter persistence homology, L., Carrière, Blumberg] - submitted.

Software :

- Python library (written in C++) for multiparameter persistence approximation, aswell as vectorization / representation.

Contributions

Papers :

- [Approximation of multiparameter persistence modules, L., Carrière, Blumberg] - On ArXiV, submitted.
- [A sparse, fast, stable representation for multiparameter persistence homology, L., Carrière, Blumberg] - submitted.

Software :

- Python library (written in C++) for multiparameter persistence approximation, aswell as vectorization / representation.
    - Link : github.com/DavidLapous/multipers
      (There are tutorial notebooks)

Contributions

Papers :

- [Approximation of multiparameter persistence modules, L., Carrière, Blumberg] - On ArXiV, submitted.
- [A sparse, fast, stable representation for multiparameter persistence homology, L., Carrière, Blumberg] - submitted.

Software :

- Python library (written in C++) for multiparameter persistence approximation, aswell as vectorization / representation.
  - Link : github.com/DavidLapous/multipers
    (There are tutorial notebooks)
  - Build on top of – and meant to be a part of – our team library
    गुढी GUDHI Geometry Understanding in Higher Dimensions .

Contributions

Papers :

- [Approximation of multiparameter persistence modules, L., Carrière, Blumberg] - On ArXiV, submitted.
- [A sparse, fast, stable representation for multiparameter persistence homology, L., Carrière, Blumberg] - submitted.

Software :

- Python library (written in C++) for multiparameter persistence approximation, aswell as vectorization / representation.
  - Link : github.com/DavidLapous/multipers
    (There are tutorial notebooks)
  - Build on top of – and meant to be a part of – our team library
    गुढी GUDHI Geometry Understanding in Higher Dimensions .
  - Linked with a few projects :
    - [Filtration-Domination in Bifiltered Graphs, Alonso, Kerber, Pritam]
    - [Euler transforms, Lebovici, Hacquard]

Contributions

Papers :

- [Approximation of multiparameter persistence modules, L., Carrière, Blumberg] - On ArXiV, submitted.
- [A sparse, fast, stable representation for multiparameter persistence homology, L., Carrière, Blumberg] - submitted.

Software :

- Python library (written in C++) for multiparameter persistence approximation, aswell as vectorization / representation.
  - Link : github.com/DavidLapous/multipers
    (There are tutorial notebooks)
  - Build on top of – and meant to be a part of – our team library
    गुढी GUDHI Geometry Understanding in Higher Dimensions .
  - Linked with a few projects :
    - [Filtration-Domination in Bifiltered Graphs, Alonso, Kerber, Pritam]
    - [Euler transforms, Lebovici, Hacquard]

Thank you !