Stochastic Geometric Days 2023

Wednesday 14, 2023

Clustering of discrete measures via mean measure quantization with applications to Topological Data Analysis

Frédéric Chazal

DataShape team

Inria & Laboratoire de Mathématiques d'Orsay

Institut DATAIA Université Paris-Saclay

Joint work with C. Levrard (Univ. Paris Cité) and M. Royer (Inria DataShape and System X)

Framework and general picture

Input :

Measure Sample $\mathbb{X}_n = \{X_1, \ldots, X_n\}$, X_i's i.i.d. $\sim X \in \mathcal{M}(\mathbb{R}^D)$.

$\mathcal{M}(\mathbb{R}^D)$ is the space of measures on $\mathbb{R}^D$ (not of constant total mass).

Examples :

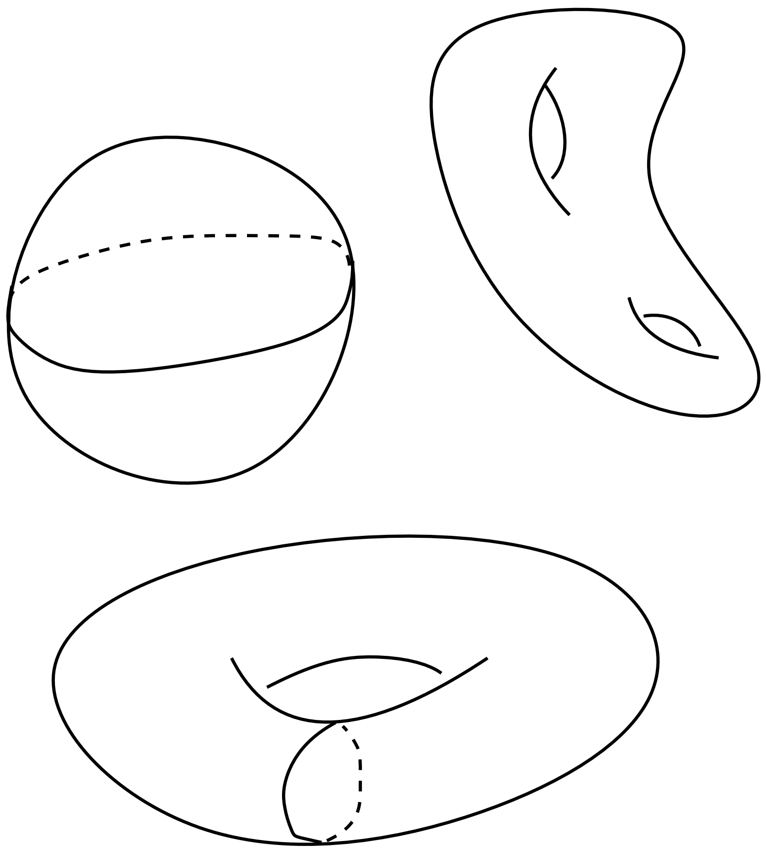
- Samples of persistence diagrams $(D = 2)$.
- Sample of realizations of a point processes in $\mathbb{R}^D$.

Objective :

Clusterize the set of measures $\mathbb{X}_n$.

TDA motivation

Input : Samples of persistence diagrams (discrete measures in $\mathbb{R}^2$), e.g. computed from point clouds sampled on submanifolds of $\mathbb{R}^N$.

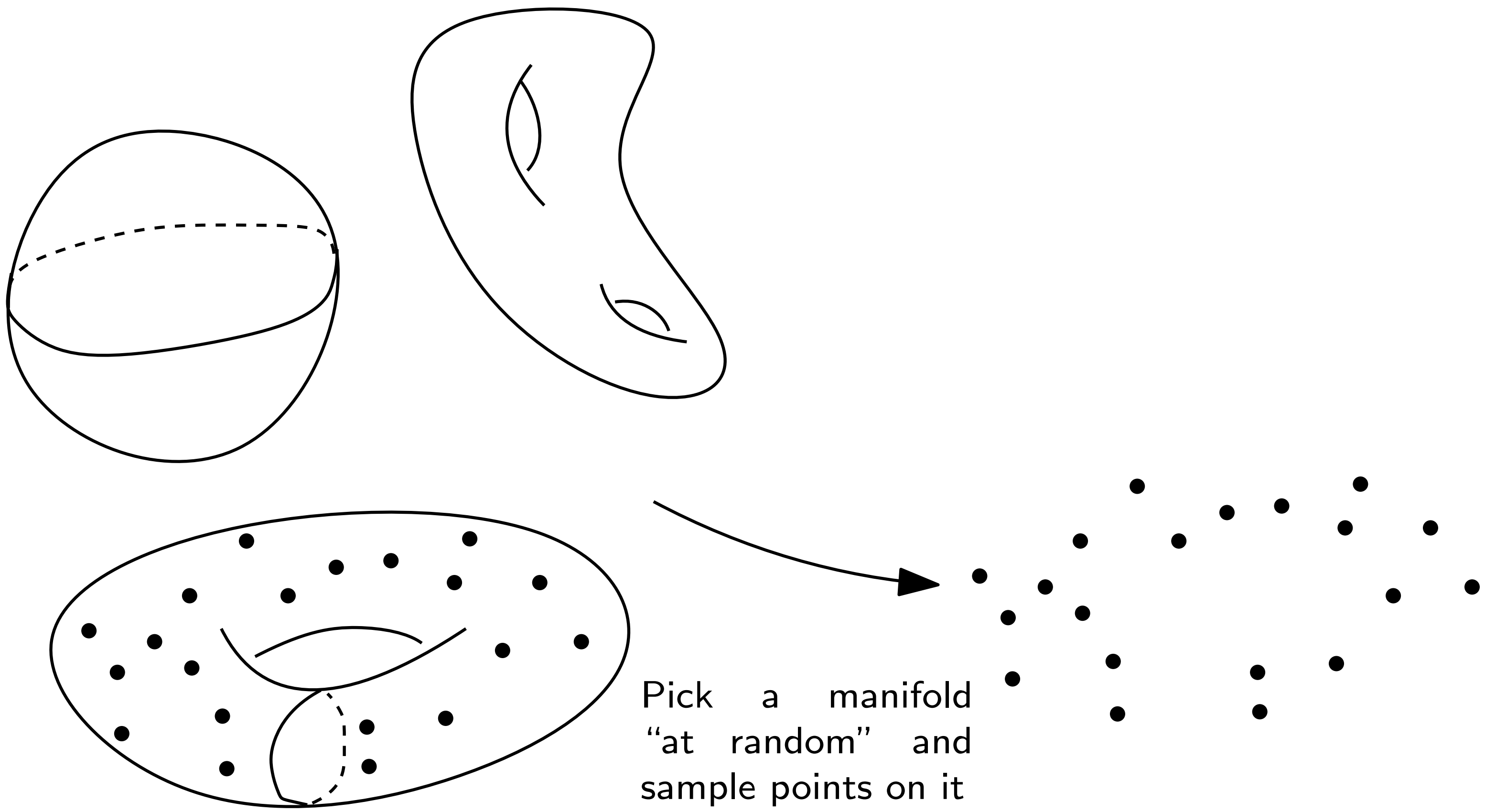


Objective :

Clusterize the set of persistence diagrams.

TDA motivation

Input : Samples of persistence diagrams (discrete measures in $\mathbb{R}^2$), e.g. computed from point clouds sampled on submanifolds of $\mathbb{R}^N$.

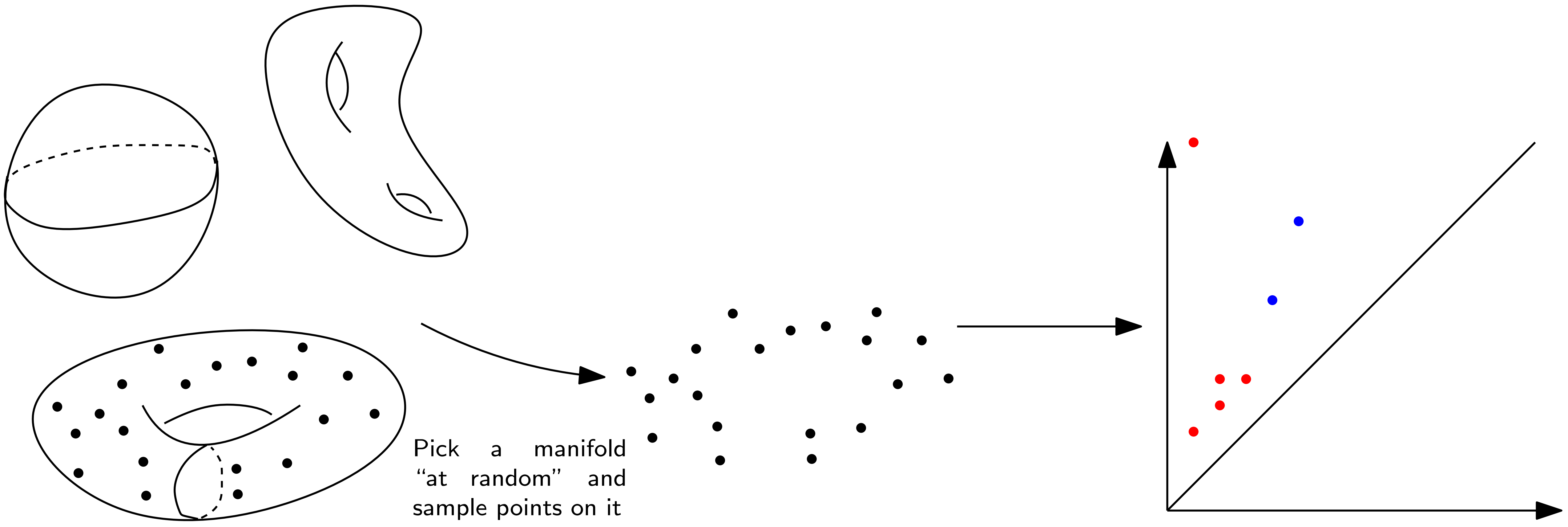


Objective :

Clusterize the set of persistence diagrams.

TDA motivation

Input : Samples of persistence diagrams (discrete measures in $\mathbb{R}^2$), e.g. computed from point clouds sampled on submanifolds of $\mathbb{R}^N$.

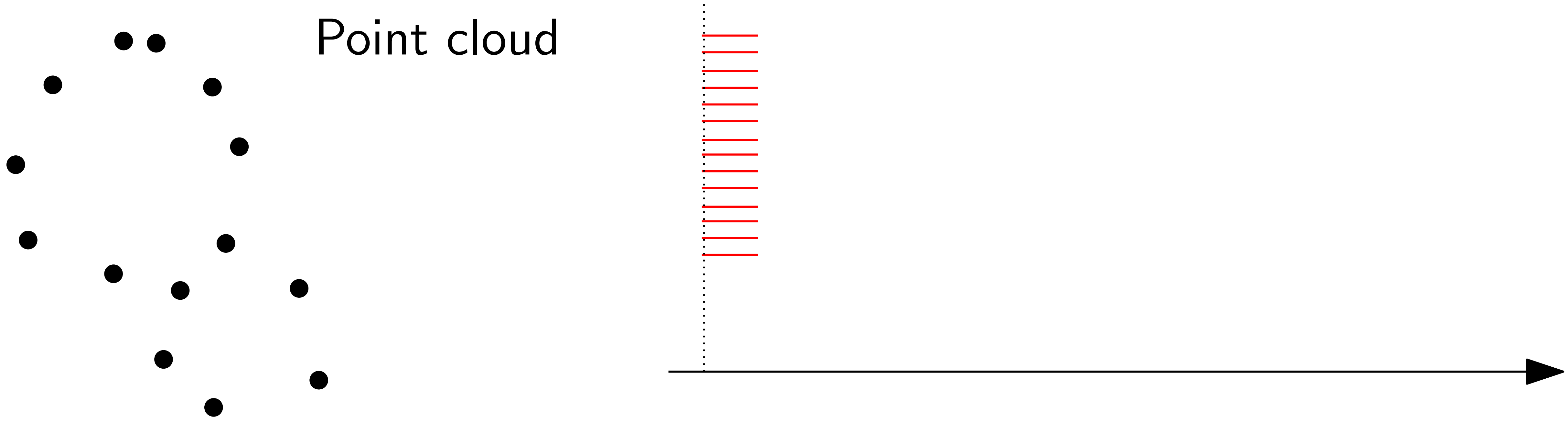


Observed data : a topological descriptor, the persistence diagram of the sample

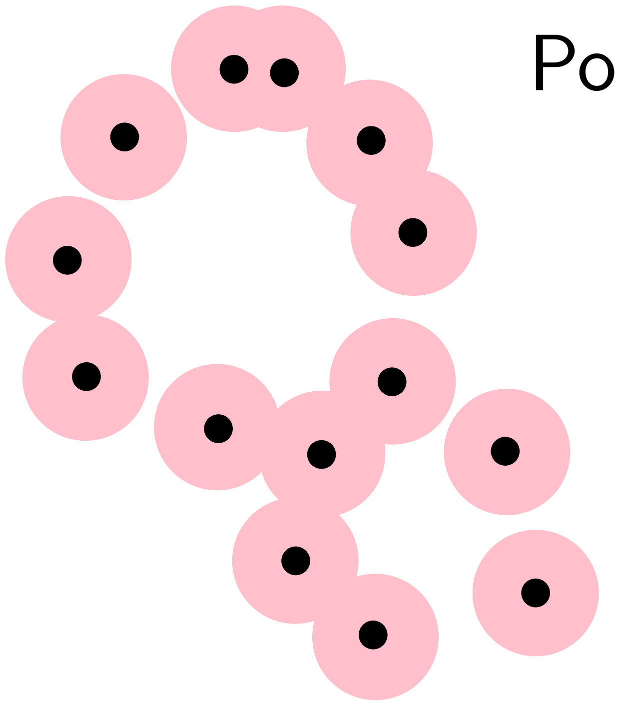
Objective :

Clusterize the set of persistence diagrams.

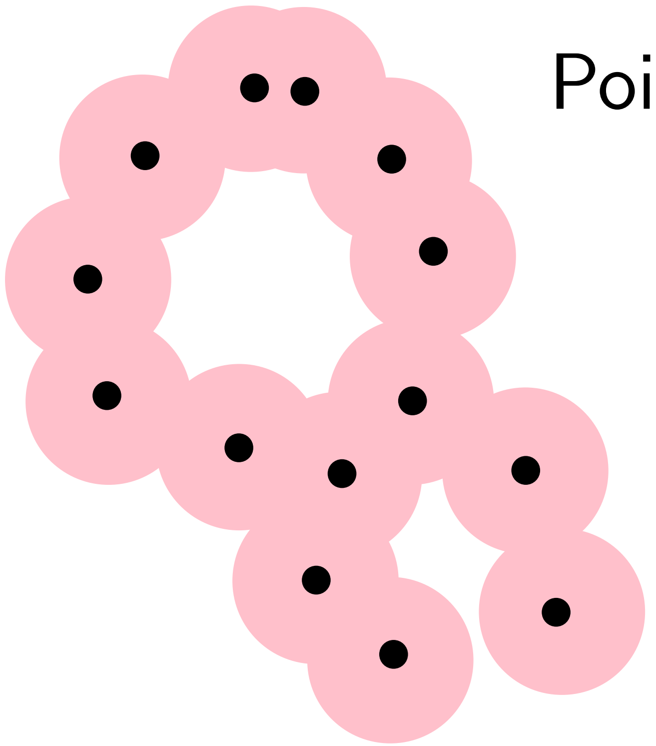
Persistent diagrams for distance functions



Persistent diagrams for distance functions



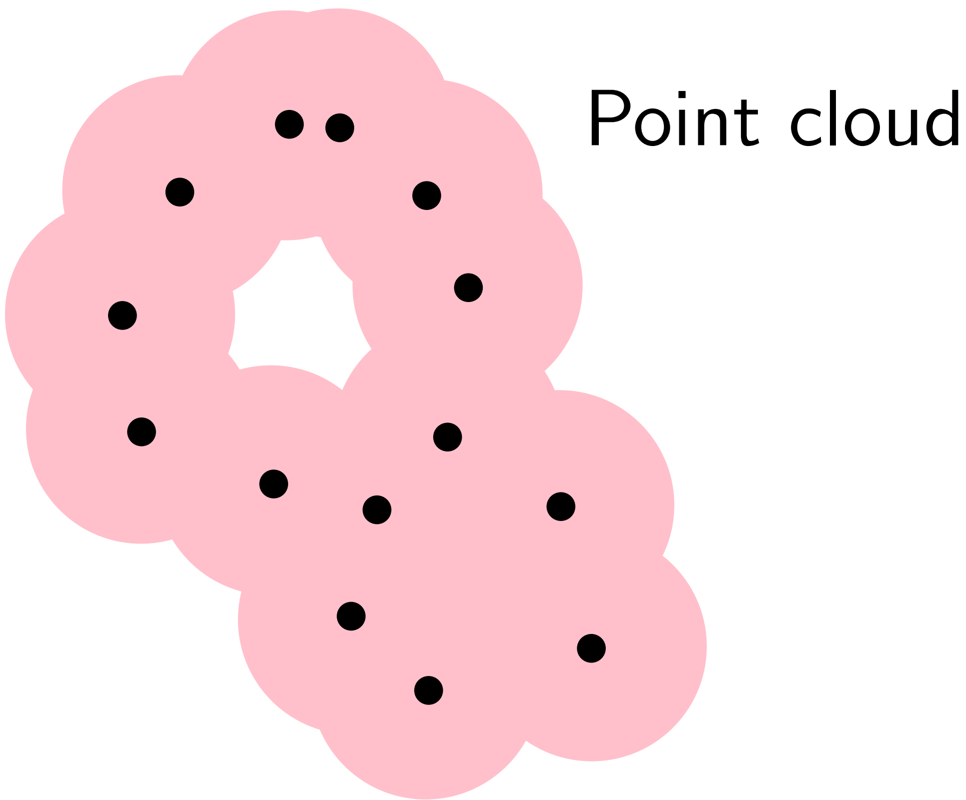
Persistent diagrams for distance functions

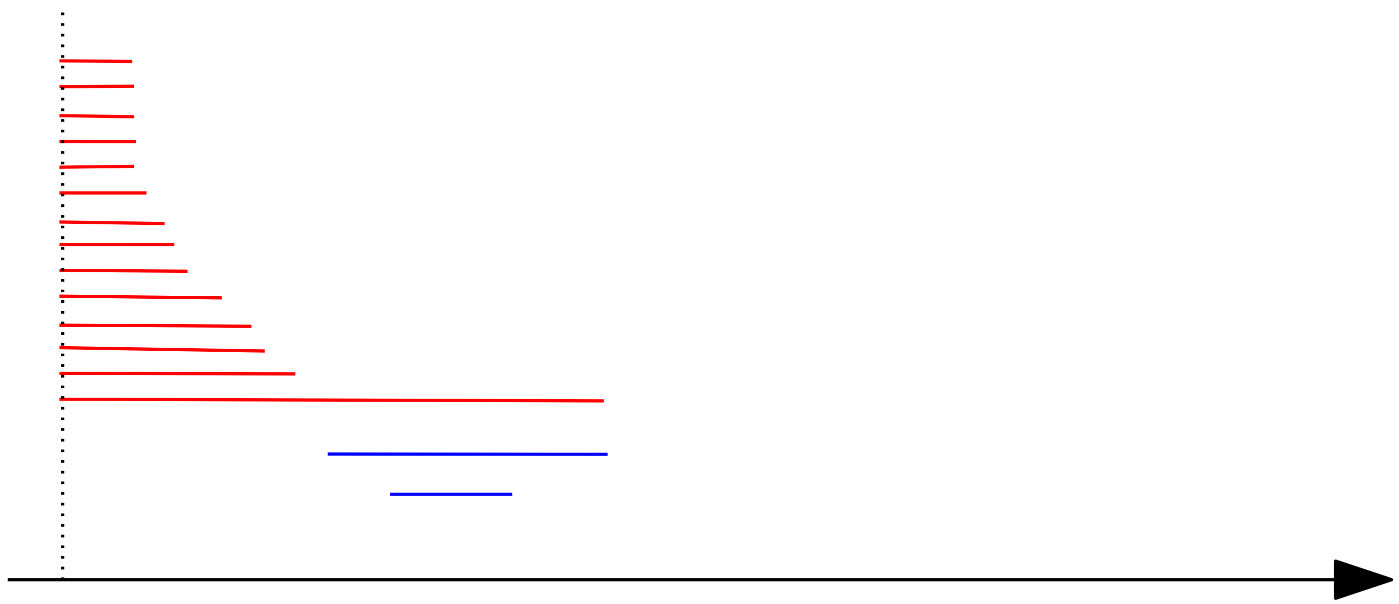


Point cloud

1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

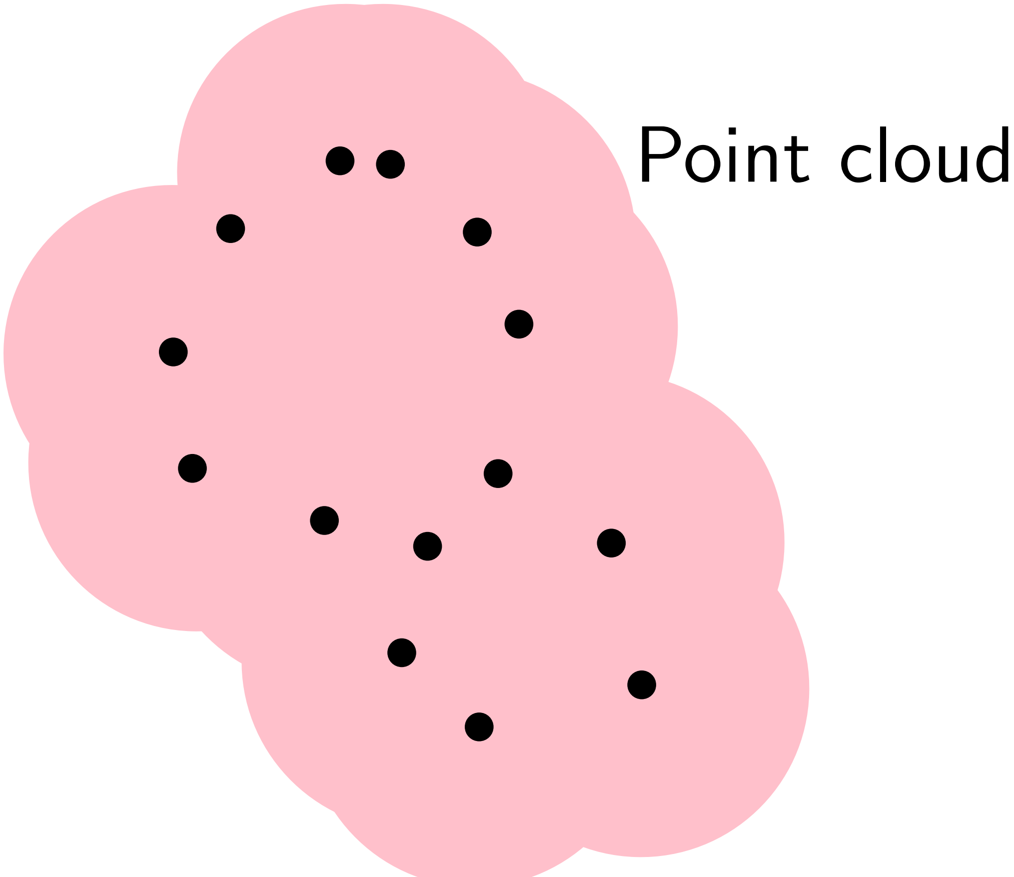
Persistent diagrams for distance functions

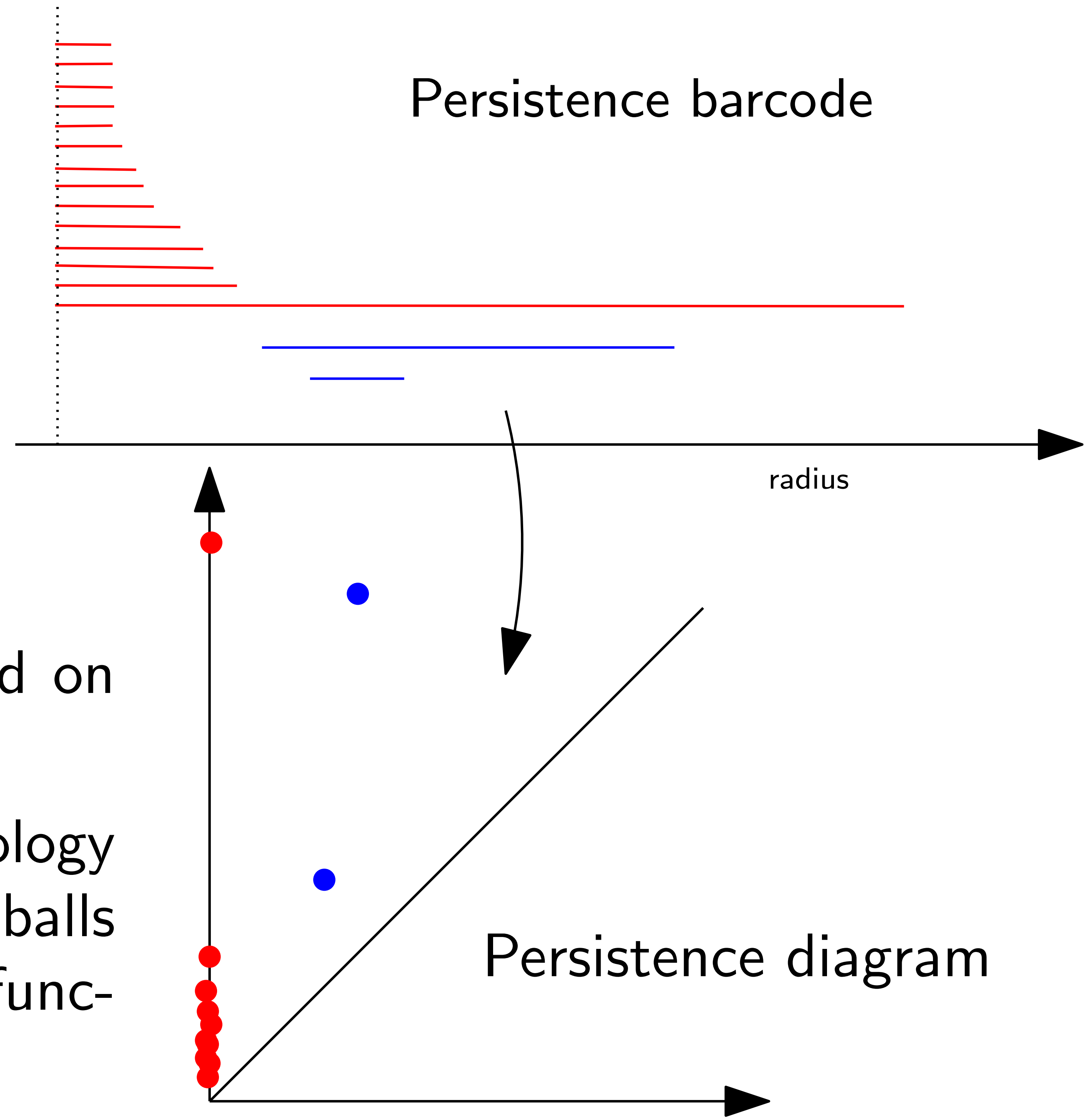




1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

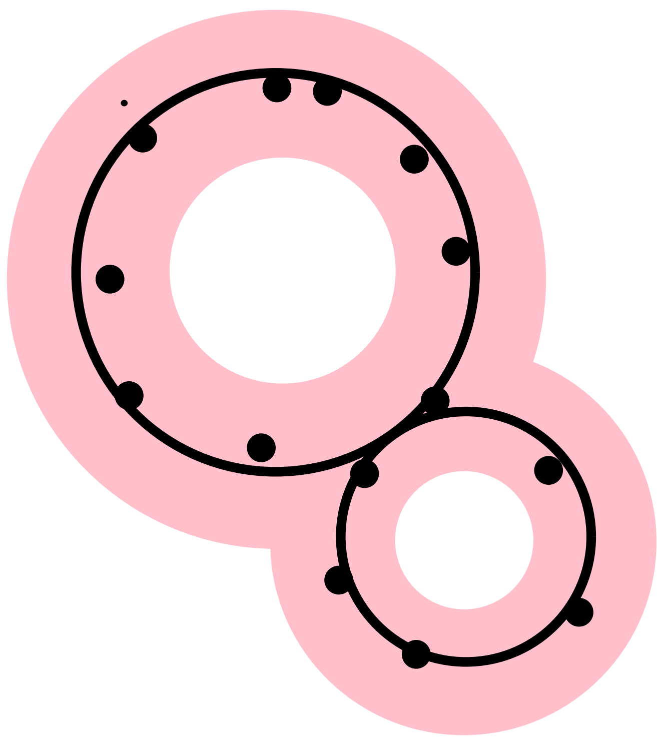
Persistent diagrams for distance functions





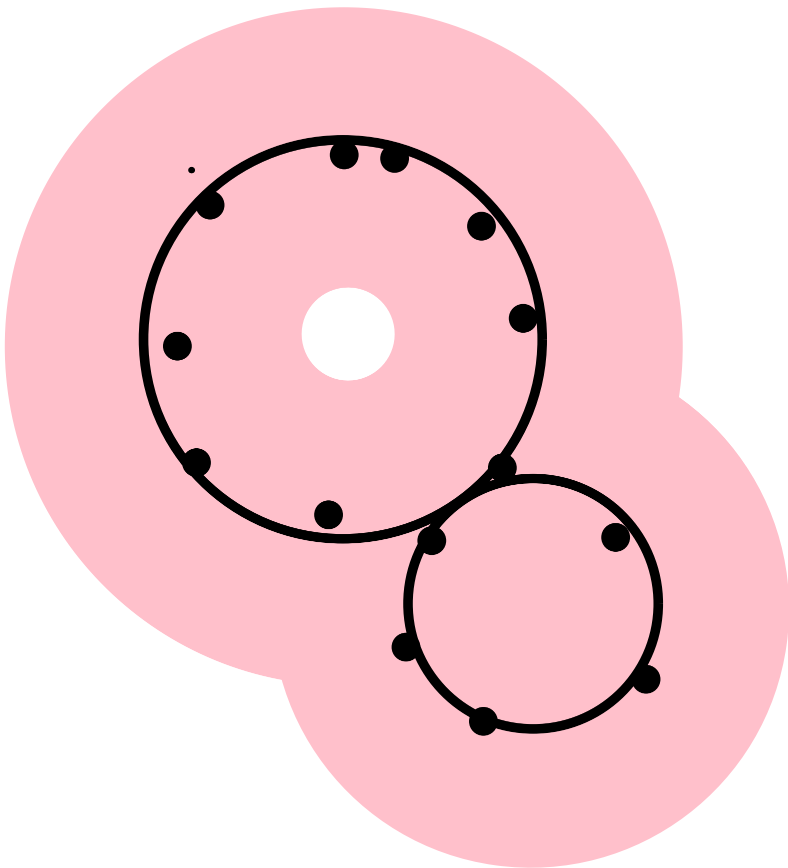
1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

Persistent diagrams for distance functions



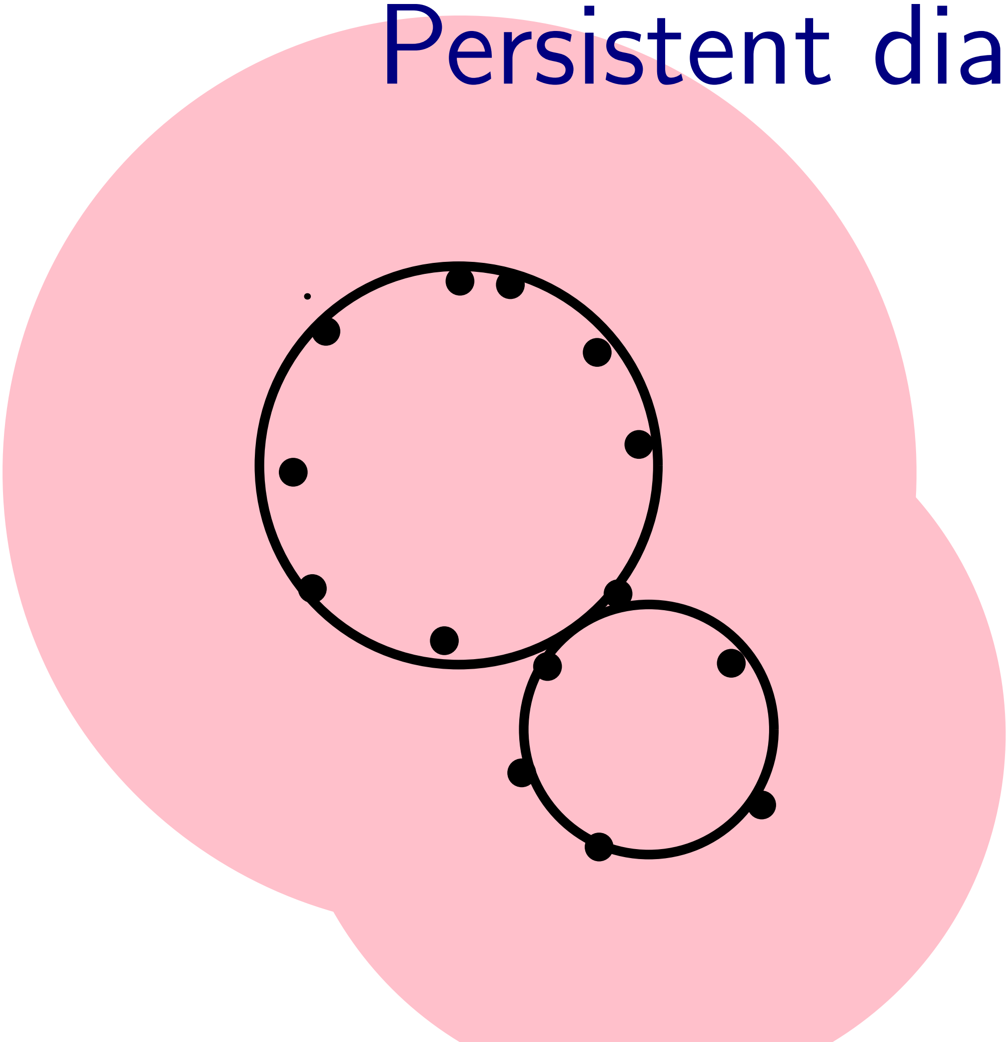
1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

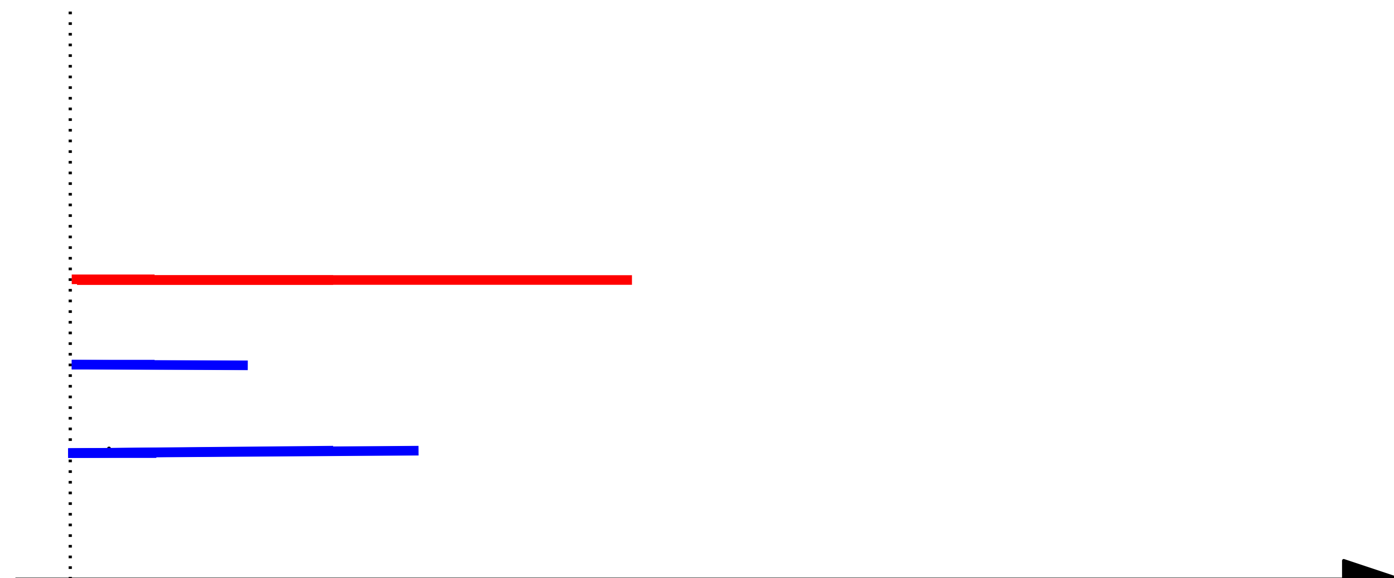
Persistent diagrams for distance functions



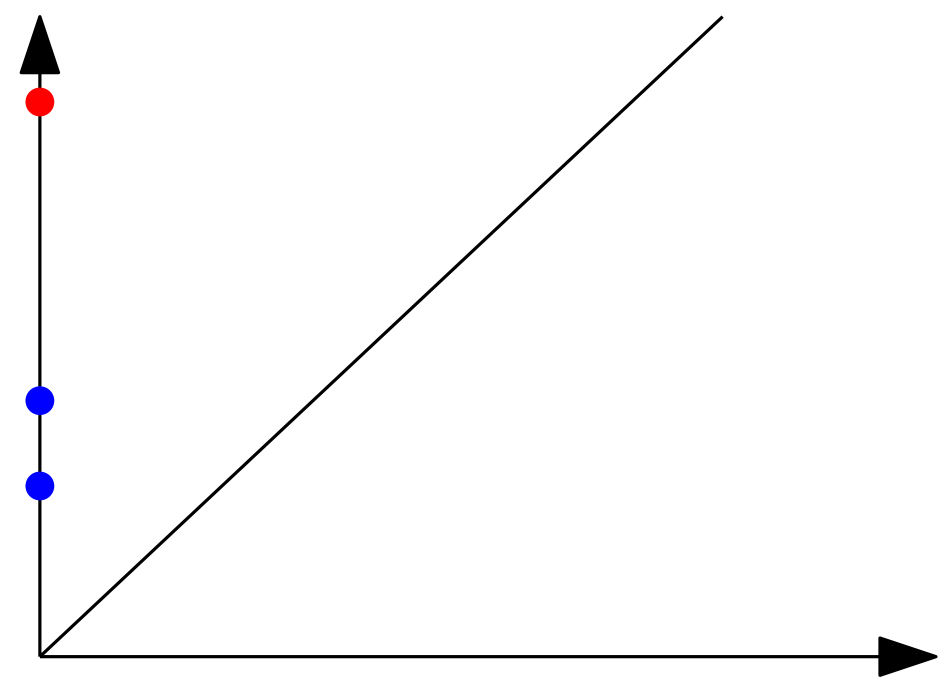
1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

Persistent diagrams for distance functions

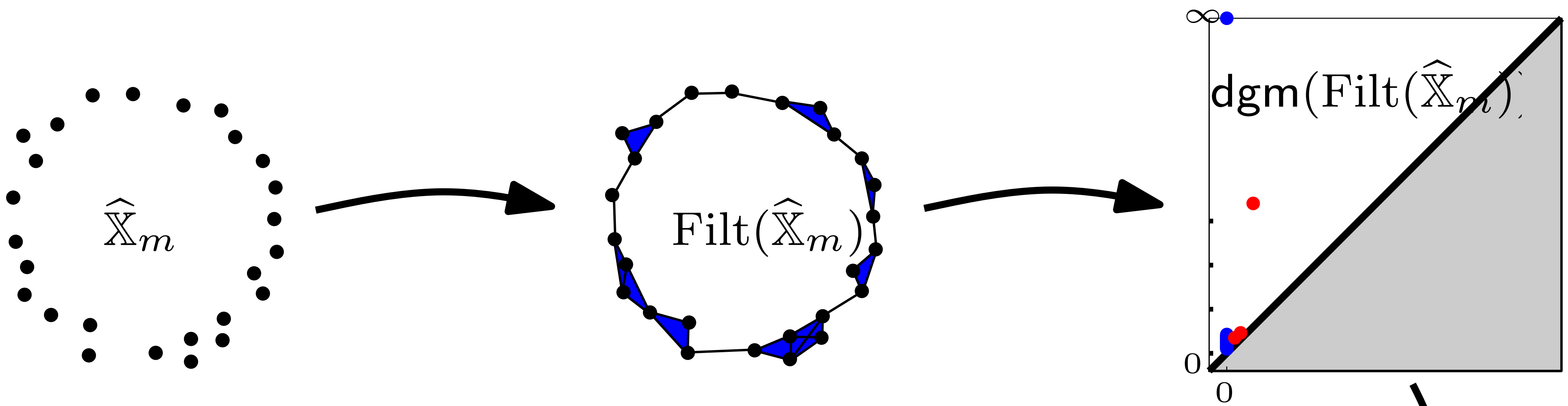




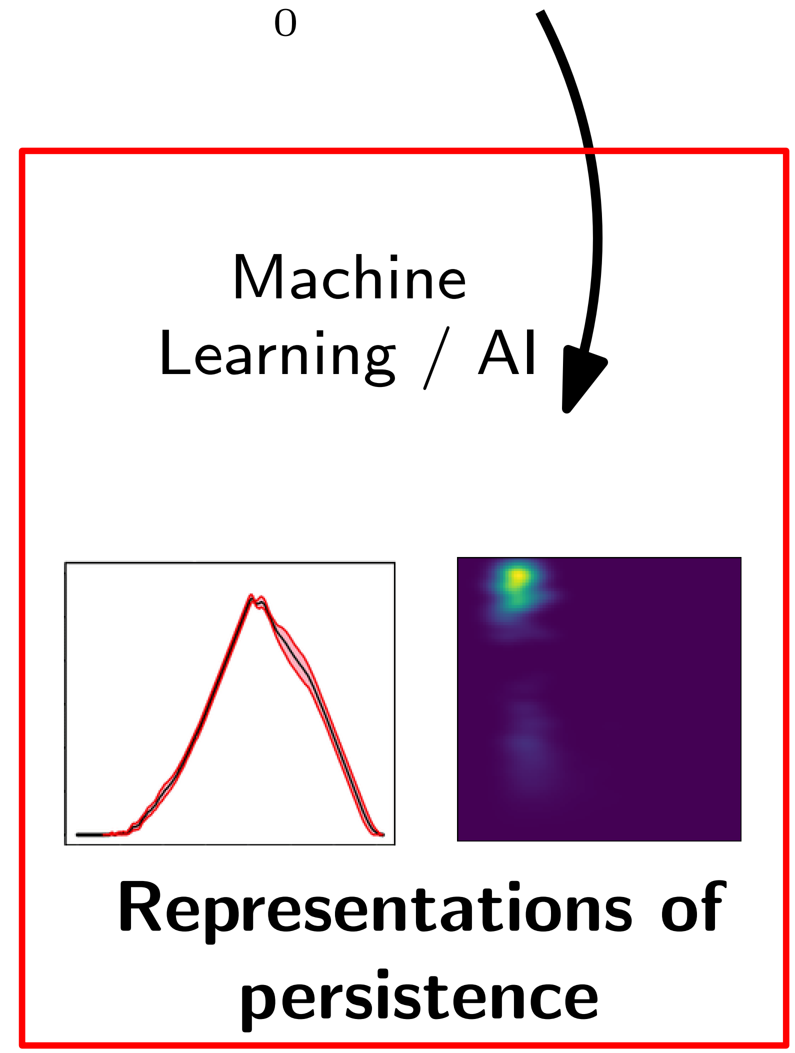
1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.



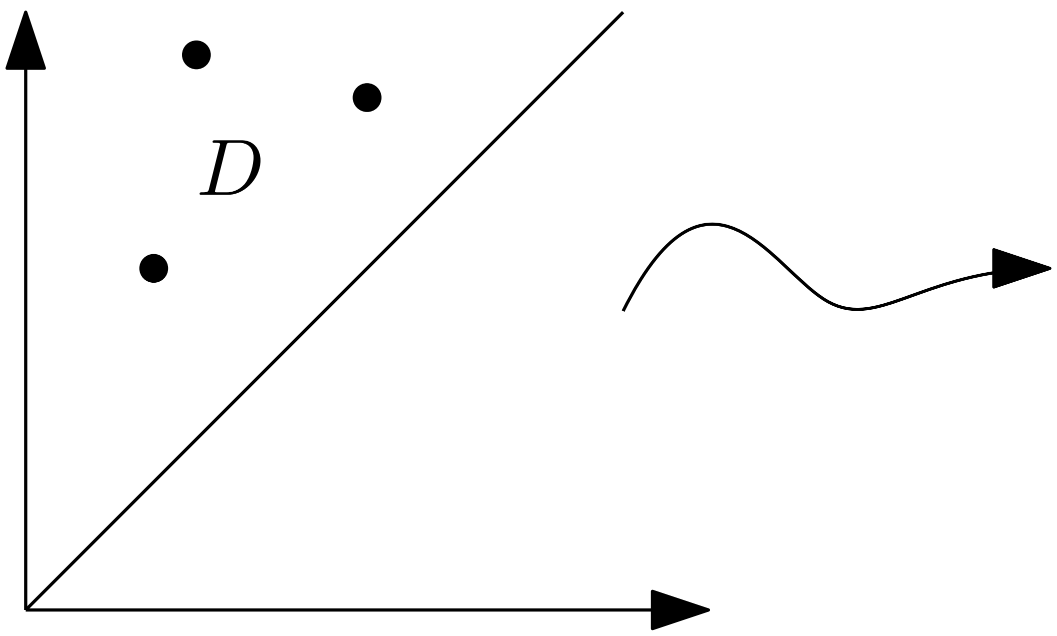
The problem of representation of persistence in ML



- Persistence diagrams are not well-suited for classical ML algorithms (the space of PD is highly non linear).
  $\rightarrow$ Need of linear representations of persistence
- Not always clear which part of the diagrams carries the relevant information.
  $\rightarrow$ How to (automatically) build relevant representations ?



Persistence diagrams as discrete measures



$$D := \sum_{p \in D} \delta_p$$

Motivations :

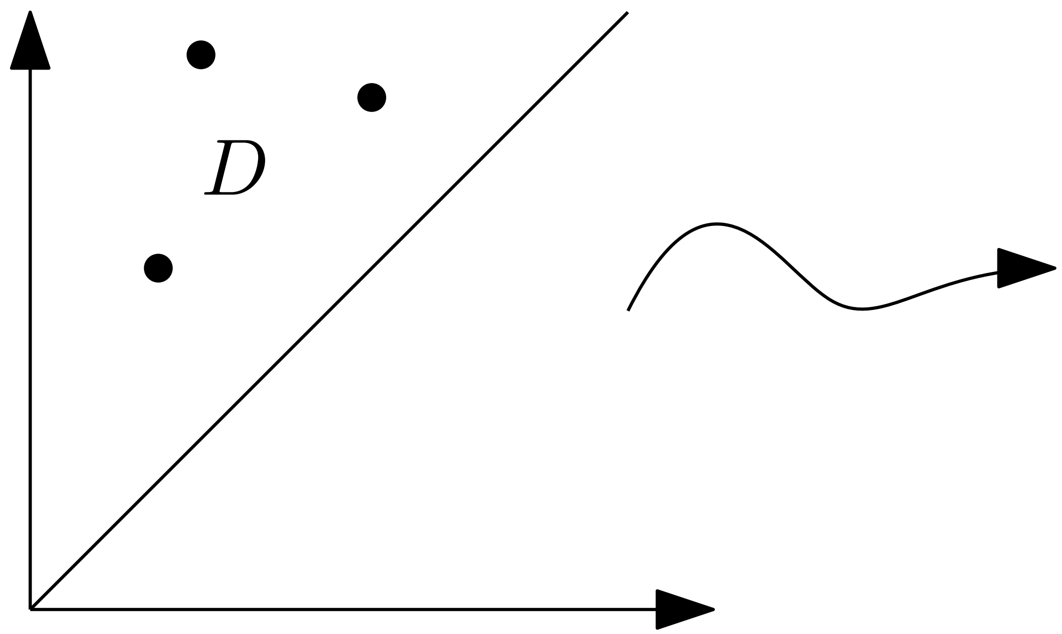
- The space of measures is much nicer that the space of P. D. !
- In the general algebraic persistence theory, persistence diagrams naturally appears as discrete measures in the plane. [C., de Silva, Glisse, Oudot 16]

- Many persistence representations can be expressed as

$$D(f) = \sum_{p \in D} f(p) = \int f dD$$

for well-chosen functions $f : \mathbb{R}^2 \to \mathcal{H}$.

Persistence diagrams as discrete measures



$$D := \sum_{p \in D} \delta_p$$

Benefits :

- Interesting statistical properties
- Data-driven selection of well-adapted representations from distributions of diagrams (mainly supervised, coming with guarantees : a whole zoo of methods)
- Optimisation of persistence-based functions

Objective of the talk : the non supervised case

Simple and efficient clustering of distributions of measures (in particular persistence diagrams) and unsupervised learning of linear representations with guarantees.

Framework and general picture

Measure Sample $\mathbb{X}_n = \{X_1, \ldots, X_n\}$, X_i's i.i.d. $\sim X \in \mathcal{M}(\mathbb{R}^D)$.

$\mathcal{M}(\mathbb{R}^D)$ is the space of measures on $\mathbb{R}^D$ (not of constant total mass).

Examples :

- Samples of persistence diagrams ($D = 2$).
- Sample of realizations of a point processes in $\mathbb{R}^D$.

Objective :

Clusterize the set of measures $\mathbb{X}_n$.

Framework and general picture

Measure Sample $\mathbb{X}_n = \{X_1, \ldots, X_n\}$, X_i's i.i.d. $\sim X \in \mathcal{M}(\mathbb{R}^D)$.

The direct approach

- endow $\mathcal{M}(\mathbb{R}^D)$ with a metric (e.g. Wasserstein),
- use standard metric clustering algorithms (k-means, hierarchical) :

  $\rightarrow$ may require $X_i(\mathbb{R}^D) = cte$ a.s. (Wasserstein metrics),

  $\rightarrow$ intractable for discrete measures with large number of support points.

Framework and general picture

Measure Sample $\mathbb{X}_n = \{X_1, \ldots, X_n\}$, X_i's i.i.d. $\sim X \in \mathcal{M}(\mathbb{R}^D)$.

The vectorization approach

$$X_i \in \mathcal{M}(R^D) \quad \Rightarrow \quad v_i = v(X_i) \in \mathbb{R}^k,$$

perform clustering on v_i's.

Not : $X(du)f := \int f X(du)$

- **Integral vectorization** : $v(X) = (X(du)f_1(u), \ldots, X(du)f_k(u))$ (Persistence Image, Silhouette, etc.)

- **Kernel vectorization** :

$$f_j(u) = \psi(\|u - c_j\|/\sigma),$$

kernel ψ, centers c_j, bandwidth σ.

$\rightarrow$ Fixed grid : $(c_j)'s$ covering of the ambient space.
$\rightarrow$ "Sample" grid : (c_j)'s drawn from the X_i's.

Theoretical setting

Choice of kernel

- **Requirements :** close to 1 around 0, decreases fast enough, 1-Lipschitz.
- **In practice :** $\Psi_{AT}(u) = \exp(-u)$.

Choice of centers

- **Mean measure** : $\mathbb{E}(X)(A) = \mathbb{E}(X(A))$, for a measurable A (intensity function).
- **Optimal codebook** :

$$\mathbf{c}^* \in \arg\min_{\mathbf{c} \in (\mathbb{R}^D)^k} \int \min_{j=1,\dots,k} \|u - c_j\|^2 \mathbb{E}(X)(du) = \arg\min_{\mathbf{c} \in (\mathbb{R}^D)^k} W_2^2(\mathbb{E}(X), P_{\mathbf{c}})$$

Choice of k, σ

- Theory in "for k large enough there exists σ".
- Practical calibration of $\sigma = \frac{B}{4}$, $B = \min_{i \neq j} \|c_i^* - c_j^*\|$.

Optimal codebook and clustering for persistence diagrams

Mixture of sampled shapes

- $S^{(1)}, \dots, S^{(L)}$ compact d_ℓ-dimensional submanifolds of $\mathbb{R}^D$, hidden labels $Z_i \in [\![1, L]\!]$, weights π_ℓ.
- Distance functions : $\mathrm{d}_{S^{(\ell)}} : \mathbb{R}^D \to \mathbb{R}_+$, $\mathrm{d}_{S^{(\ell)}}(x) = \min_{yinS^{(\ell)}} \|x - y\|$.

- "True" thresholded persistence diagrams at scale s (for $\mathrm{d}_{S^{(\ell)}}$) :

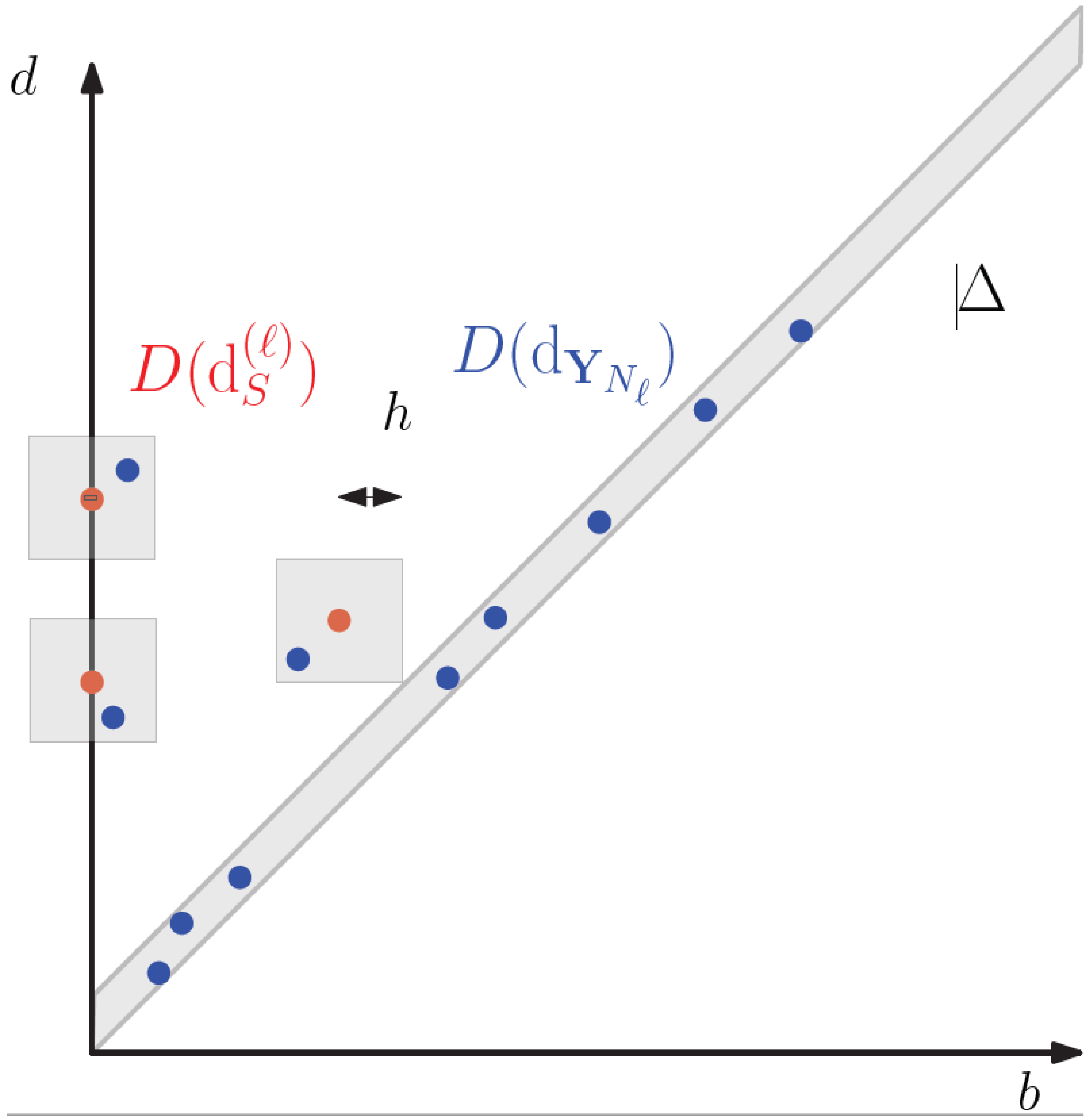
$$D^{(\ell)}_{\geq s} = \sum_{\{(b,d) \in D^{(\ell)} | d-b \geq s\}} n(b,d)\delta_{(b,d)} := \sum_{j=1}^{k_0^{(\ell)}} n(m_j^{(\ell)})\delta_{m_j^{(\ell)}}.$$

- For $\ell \in [\![1, L]\!]$, a $\mathbb{Y}_{N_\ell}$ sample uniformly enough on $S^{(\ell)}$, with $N_\ell^{-1/d_\ell} \lesssim h \leq s$.
- Component distribution : thresholded persistence diagram from $\mathbb{Y}_{N_\ell}$

$$X_i \mid \{Z_i = \ell\} \sim X^{(\ell)} \sim \hat{D}^{(\ell)}_{\geq s-h}.$$

Idea 1 (stability of persistence diagrams)

"If h is small enough (enough sample points on every shape), then X_i is close to the true diagram $D^{(\ell)}_{\geq s}$ (w.h.p)"



Idea 2

"If two shapes differ by at least one true diagram point, then those points can be approximated via quantization provided k is large enough."

Discriminable shapes

The shapes $S^{(1)}, \ldots, S^{(\ell)}$ are discriminable at scale $\mathbf{s}$ if for any $1 \leq \ell_1 < \ell_2 \leq L$ there exists $m_{\ell_1,\ell_2} \in \mathbb{R}^2$ such that

$$D^{(\ell_1)}_{\geq s}(\{m_{\ell_1,\ell_2}\}) \neq D^{(\ell_2)}_{\geq s}(\{m_{\ell_1,\ell_2}\}).$$

Idea 2

"If two shapes differ by at least one true diagram point, then those points can be approximated via quantization provided k is large enough."

Covering property of optimal codebooks

Let $M_\ell = D^{(\ell)}_{\geq s}(\mathbb{R}^2)$, $\bar{M} = \sum_{\ell=1}^{L} \pi_\ell M_\ell$, and $\pi_{min} = \min_{\ell \leq L} \pi_\ell$.
Assume that $S^{(1)}, \ldots, S^{(L)}$ are discriminable at scale s, and let $m_1, \ldots, m_{k_0}$ denote the discrimination points. Let $K_0(h)$ denote

$$\inf\{k \geq 0 \mid \exists t_1, \ldots, t_k \quad \bigcup_{\ell=1}^{L} D^{(\ell)}_{\geq s} \setminus \{m_1, \ldots, m_{k_0}\} \subset \bigcup_{s=1}^{k} \mathrm{B}_\infty(t_s, h)\}.$$

Let $k \geq k_0 + K_0(h)$, and $(c_1^*, \ldots, c_k^*)$ denote an optimal k-points quantizer of $\mathbb{E}(X)$. Then, provided that h is small enough, we have

$$\forall j \in [\![1, k_0]\!] \quad \exists p \in [\![1, k]\!] \quad \|c_p^* - m_j\|_\infty \leq \frac{5\sqrt{\bar{M}}h}{\sqrt{\pi_{min}}}.$$

A coarse bound

Recall :

$$v_i = (X_i(du)\exp(-\|u - c_1^*\|/\sigma), \ldots, X_i(du)\exp(-\|u - c_k^*\|/\sigma)).$$

- Scale parameters : $\tilde{B} = \min_{i=1,\ldots,k_0, j=1,\ldots,K_0, j\neq i} \|m_i - m_j\|_\infty \wedge s$,

$$\sigma \in \left[\frac{\tilde{B}}{128M}, \frac{\tilde{B}}{64M}\right].$$

- Centers : $k \geq k_0 + K_0(h)$.

Proposition : Provided h is small enough, it holds, with high probability,

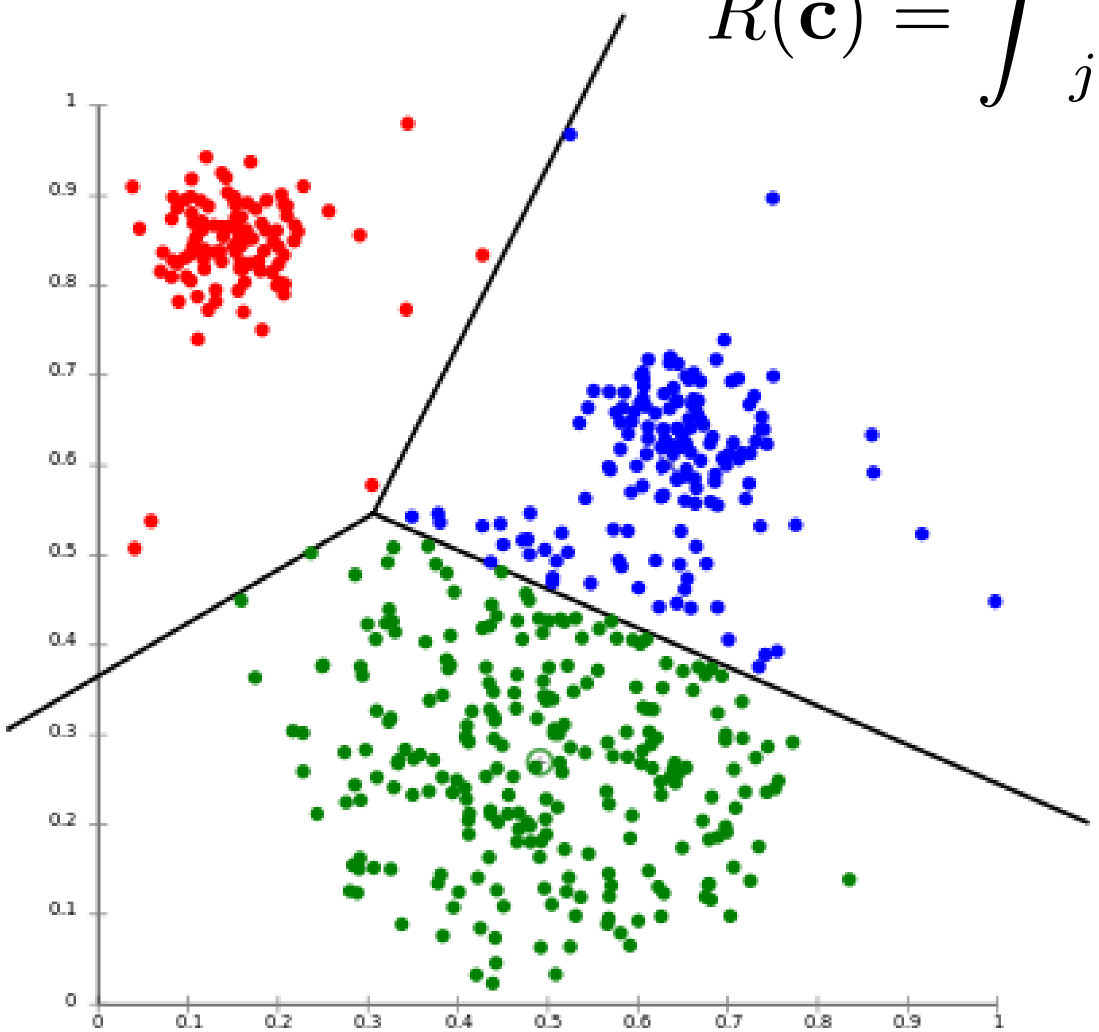
$$\begin{array}{lll} Z_{i_1} = Z_{i_2} & \Rightarrow & \|v_{i_1} - v_{i_2}\|_\infty \leq \frac{1}{4}, \\ Z_{i_1} \neq Z_{i_2} & \Rightarrow & \|v_{i_1} - v_{i_2}\|_\infty \geq \frac{1}{2}. \end{array}$$

Sample optimization of optimal codebooks

k-means like algorithm

Objective : minimize true risk

$$R(\mathbf{c}) = \int \min_{j=1,\dots,k} \|u - c_j\|^2 \mathbb{E}(X)(du).$$



Lloyd algorithm (point sample case) :

- Initialization at random
- Iteration t :
  - $c_j^t \leftarrow \frac{\bar{X}_n(du)[u \mathbb{1}_{W_j(\mathbf{c}^{t-1})}]}{\bar{X}_n[W_j(\mathbf{c}^{t-1})]}$.
- Stop when stabilized.

$W_j(\mathbf{c}^t)$: Voronoi cell of c_j^t,
$\bar{X}_n$ empirical distribution $\frac{1}{n}\sum_{i=1}^n \delta_{X_i}$ (sample case).

k-means like algorithm

Straightforward extensions

Batch algorithm (Lloyd's type)

- Initialization $\mathbf{c}^{(0)}$ at random.
- Iteration t :

$$c_j^t \leftarrow \frac{\bar{X}_n(du)[u\mathbb{1}_{W_j(\mathbf{c}^{t-1})}]}{\bar{X}_n[W_j(\mathbf{c}^{t-1})]}, \quad \bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

- Stop when stabilized.

Mini-batch algorithm (McQueen's type) Split $[\![1, n]\!]$ into T equally sized mini-batches $B_1, \dots, B_T$.

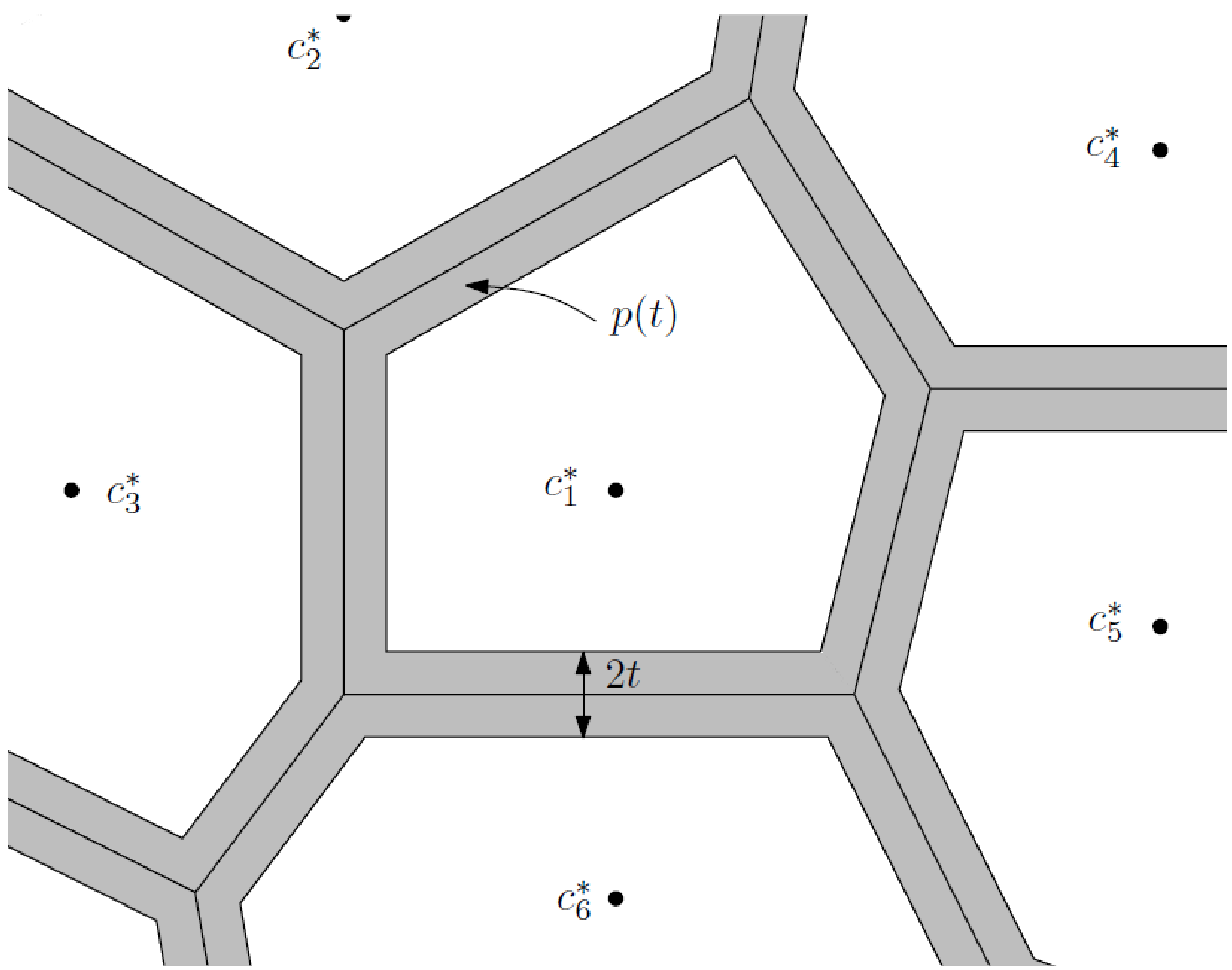
- Initialization $\mathbf{c}^{(0)}$ at random.
- For $t = 1, \dots, T$:

$$c_j^{(t)} \leftarrow \left(1 - \frac{1}{t}\right) c_j^{(t-1)} + \frac{1}{t}\frac{\bar{X}_{B_t}(du)[u\mathbb{1}_{W_j(\mathbf{c}^{t-1})}]}{\bar{X}_{B_t}[W_j(\mathbf{c}^{t-1})]}$$

Margin condition on $\mathbb{E}(X)$

For $X \in \mathcal{M}(\Lambda, M)$ a.s. ($\mathrm{Supp}(X) \subset \mathrm{B}(0,\Lambda)$ and $X(\mathbb{R}^D) \leq M$).

- $B = \inf_{\mathbf{c}^* \in \mathcal{C}_{opt}} \min_{i \neq j} \|c_i^* - c_j^*\| (> 0)$.
- $p_{\min} = \inf_{\mathbf{c}^* \in \mathcal{C}_{opt}} \min_i \mathbb{E}(X)(W_i(\mathbf{c}^*))(> 0)$.
- For $\mathbf{c}^* \in \mathcal{C}_{opt}$, $N(\mathbf{c}^*) = \bigcup_{i \neq j} \bar{W}_j(\mathbf{c}^*) \cap \bar{W}_i(\mathbf{c}^*)$ (skeleton of the Voronoi Diagram).



Margin condition on $\mathbb{E}(X)$

For $X \in \mathcal{M}(\Lambda, M)$ a.s. ($\mathrm{Supp}(X) \subset \mathrm{B}(0,\Lambda)$ and $X(\mathbb{R}^D) \leq M$).

- $B = \inf_{\mathbf{c}^* \in \mathcal{C}_{opt}} \min_{i \neq j} \|c_i^* - c_j^*\| (> 0)$.
- $p_{\min} = \inf_{\mathbf{c}^* \in \mathcal{C}_{opt}} \min_i \mathbb{E}(X)(W_i(\mathbf{c}^*)) (> 0)$.
- For $\mathbf{c}^* \in \mathcal{C}_{opt}$, $N(\mathbf{c}^*) = \bigcup_{i \neq j} \bar{W}_j(\mathbf{c}^*) \cap \bar{W}_i(\mathbf{c}^*)$ (skeleton of the Voronoi Diagram).

Margin condition with radius r_0 :

$\mathbb{E}(X) \in \mathcal{M}(\Lambda, M)$ satisfies a margin condition with radius $r_0 > 0$ if and only if, for all $0 \leq t \leq r_0$,

$$\sup_{\mathbf{c}^* \in \mathcal{C}_{opt}} \mathbb{E}(X) \left(\mathrm{B}(N(\mathbf{c}^*), t)\right) \leq \frac{B p_{min}}{128 \Lambda^2} t,$$

Convergence results

If $X \in \mathcal{M}(\Lambda, M)$ a. s. and $\mathbb{E}(X)$ satisfies a margin condition.

Batch algorithm.
If $|\mathrm{Supp}(X)| \leq N_{\max}$ a.s. and $\mathbf{c}^{(0)} \in \mathrm{B}(\mathcal{C}_{opt}, \Lambda_0)$, for $T \geq 2\log(n)$ and n large enough, with high probability $(1 - e^{-C_0 n} - e^{-x})$,

$$R(\mathbf{c}^{(T)}) - R^* \leq C \frac{M^3 \Lambda^2 k^2 D \log(k)}{n p_{\min}^2} (1 + x).$$

Mini-batch algorithm

If $\mathbf{c}^{(0)} \in \mathrm{B}(\mathcal{C}_{opt}, \Lambda_0)$ and $n/T = ckM^2 \log(n)/p_{\min}^2$ (size of batches), then

$$\mathbb{E}\left(R(\mathbf{c}^{(T)}) - R^*\right) \leq C \frac{k^2 M^4 \Lambda^2 \log(n)}{n p_{\min}^3}.$$

$\rightarrow$ minimax rates (in n).

Experiments

The ATOL procedure

https://gudhi.inria.fr/python/latest/representations.html

$X_1, \ldots, X_n$ a measure sample. User choice of k.

- **Quantization step** : build $\hat{\mathbf{c}} = (\hat{c}_1, \ldots, \hat{c}_k)$ via mini-batch Algorithm
- **Vectorization step** : convert X_i into v_i via

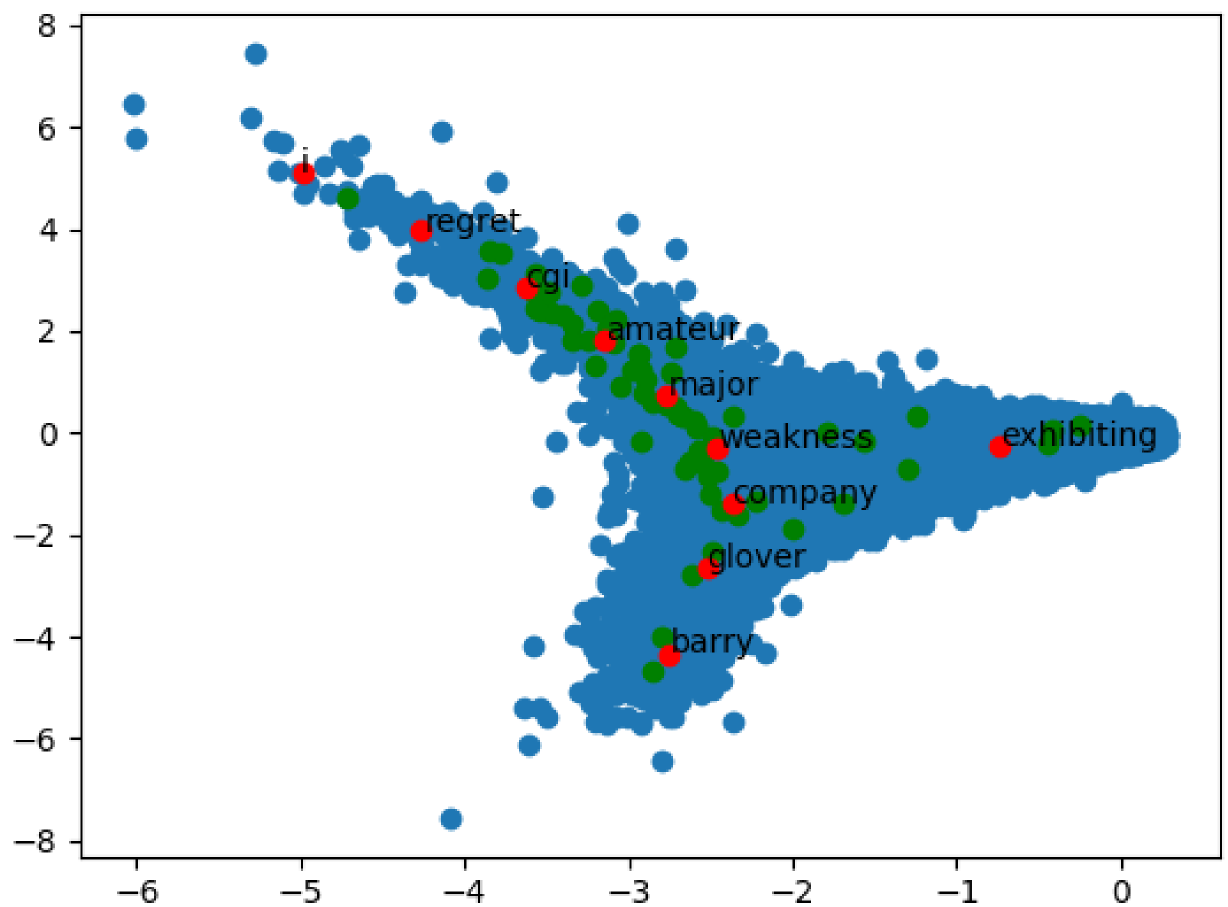
  $$v_i = (X_i(du)(\exp(-\|u - \hat{c}_1\|/\sigma)), \ldots, X_i(du)(\exp(-\|u - \hat{c}_k\|/\sigma))),$$

  where $\sigma = \hat{B}/2$.

Then use your favorite clustering/learning algorithm.

A high dimensional example : sentiment learning on texts

- Large Movie Review Dataset : 50000 reviews (texts), with labels (positive or negative)
- Review $=$ bag of words
- Word w embedded into $\mathbb{R}^{100}$ via `word2vec` (module `gensim`)



A high dimensional example : sentiment learning on texts

- Large Movie Review Dataset : 50000 reviews (texts), with labels (positive or negative)
- Review $=$ bag of words
- Word w embedded into $\mathbb{R}^{100}$ via `word2vec` (module `gensim`)

10-fold cross-validation, accuracies and computations times :

- ATOL with $k = 20 + 32$ units dense one layer NN : 85.6 ± 0.95, average times $5.5 + 208.3 + 351.2$ s.
- Recurrent NN (LSTM) with 64 units : 89.3 ± 0.44, average time about 1 hour
- `kaggle` winner 99.9, time 10379.3 s.

Large scale graph classification

$G(V, E)$ graph with set V of vertices and set E of edges, t a diffusion time.

- Heat Kernel Signature at time t : HKS_t set values on V.
- Filtration of G w.r.t. HKS_t : 4 types of topological features with life times via extended persistence.

$$G(V, E) \xrightarrow[\text{signatures}]{\text{heat kernel}} \mathsf{HKS}_t(G) \in \mathbb{R}^{|V|},$$

$$G(V, E) \xrightarrow[\text{persistence}]{\text{extended}} \mathsf{PD}(\mathsf{HKS}_t(G), G) \in (\mathcal{M}(\mathbb{R}^2))^4.$$

Large scale graph classification

$$G(V,E) \xrightarrow[\text{signatures}]{\text{heat kernel}} \mathsf{HKS}_t(G) \in \mathbb{R}^{|V|},$$

$$G(V,E) \xrightarrow[\text{persistence}]{\text{extended}} \mathsf{PD}(\mathsf{HKS}_t(G), G) \in (\mathcal{M}(\mathbb{R}^2))^4.$$

Vectorization : For two diffusion times t_1 and t_2, ATOL on each $\mathcal{M}(\mathbb{R}^2)$ coordinate, with $k = 10$: $\rightarrow$ embedding in $\mathbb{R}^{(10\times 4\times 2)}$.
Classification : Random Forest (100 trees).

method		SF	NetLSD	FGSD	GeoScat	ATOL
`reddit threads`	(203K)	81.4±.2	**82.7±.1**	82.5±.2	80.0±.1	80.7±.1
`twitch egos`	(127K)	67.8±.3	63.1±.2	**70.5±.3**	69.7±.1	69.7±.1
`github stargazers`	(12.7K)	55.8±.1	63.2±.1	65.6±.1	54.6±.3	**72.3±.4**
`deezer ego nets`	(9.6K)	50.1±.1	52.2±.1	**52.6±.1**	52.2±.3	51.0±.6

Mean ROC-AUC and standard deviations (100 repetitions of $0.8/0.2$ train/test).

Large scale graph classification

Alternative approach

$$G(V,E) \xrightarrow[\text{signatures}]{\text{heat kernel}} \mathsf{HKS}_{t_1,t_2,t_3,t_4}(G) \in \mathbb{R}^{4|V|} \approx \mathcal{M}(\mathbb{R}^4).$$

Vectorization : ATOL with $k = 80$ (embedding in $\mathbb{R}^{80}$).
Classification : Random Forest $(100$ trees$)$.

method	RetGK	FGSD	WKPI	GNTK	PersLay	Atol (PD)	Atol (Direct)
REDDIT (5K, 5 classes)	56.1±.5	47.8	59.5±.6	—	55.6±.3	**67.1±.3**	**66.1±.2**
REDDIT (12K, 11 classes)	48.7±.2	—	48.5±.5	—	47.7±.2	**51.4±.2**	**50.7±.3**
COLLAB (5K, 3 classes)	81.0±.3	80.0	—	83.6±.1	76.4±.4	**88.3±.2**	**88.5±.1**
IMDB-B (1K, 2 classes)	71.9±1.	73.6	75.1±1.1	**76.9±3.6**	71.2±.7	74.8±.3	73.9±.5
IMDB-M (1.5K, 3 classes)	47.7±.3	**52.4**	48.4±.5	**52.8±4.6**	48.8±.6	47.8±.7	47.0±.5

Mean accuracies and standard deviations.

Recap

A coarse and unsupervised measure vectorization scheme

- but fast,
- yields not that bad results in further clustering and classification tasks,
- comes with a few theoretical insights.

Recap

A coarse and unsupervised measure vectorization scheme

- but fast,
- yields not that bad results in further clustering and classification tasks,
- comes with a few theoretical insights.

Perspectives (on-going work) :

- (time-)dependent data.
- supervised learning.

Thanks for your attention

References :

[1] M. Royer, F.Chazal, C.Levrard, Y. Umeda, Y. Ike. ATOL : Measure Vectorization for Automatic Topologically-Oriented Learning. AISTAT 2021

[2] F. Chazal, C. Levrard and M. Royer. Clustering of measures via mean measure quantization. Electronic Journal of Statistics 2021.

A small recap

Vectorization : $v(X_i) = (X_i(du)(\psi(\|u - c_1^*\|/\sigma)), \ldots, X_i(du)(\psi(\|u - c_k^*\|/\sigma)))$.

Quantization : $\mathbf{c}^* \in \arg\min_{\mathbf{c} \in (\mathbb{R}^D)^k} \mathbb{E}(X)(du) \min_{j=1,\ldots,k} \|u - c_j\|^2$.

Relevant when

- distributions from two different clusters differ on an area of size r (choose $\sigma \lesssim r$).
- $\mathbf{c}^*$ has codepoints on these areas.
  - $\rightarrow$ (Theoretical worst case) $k \gtrsim r^{-d}$, d "dimension" of the support of $\mathbb{E}(X)$.
  - $\rightarrow$ Worst-case guarantees are the same as for deterministic grid (for $d = D$).

Major advantage : fast approximation of $\mathbf{c}^*$ from sample.

A small recap 2

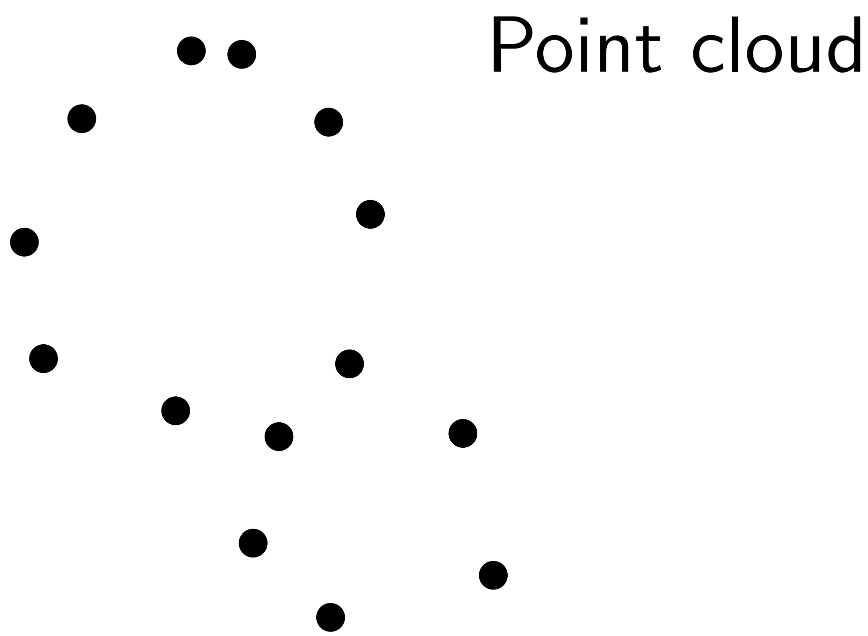
Sample approximation of $\mathbf{c}^*$ with fast algorithms and optimal rates, but :

- stringent dependency on the initialization (volume arguments for repeated initializations deprecates for large D's),

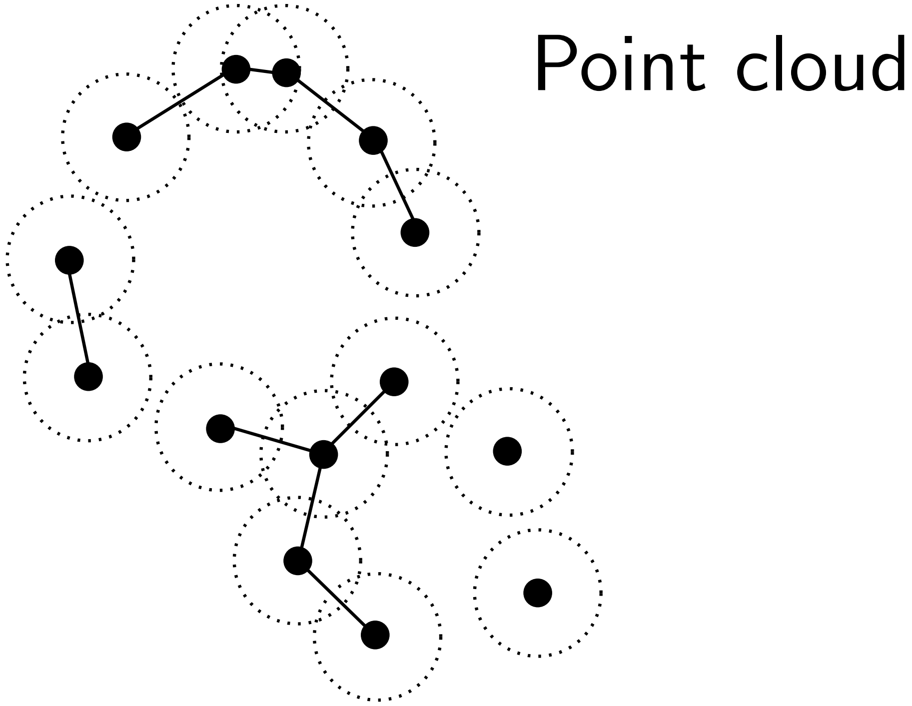
- margin condition far too demanding (uniform distributions do not satisfy it for instance).

Not that useful theoretical results for the moment...

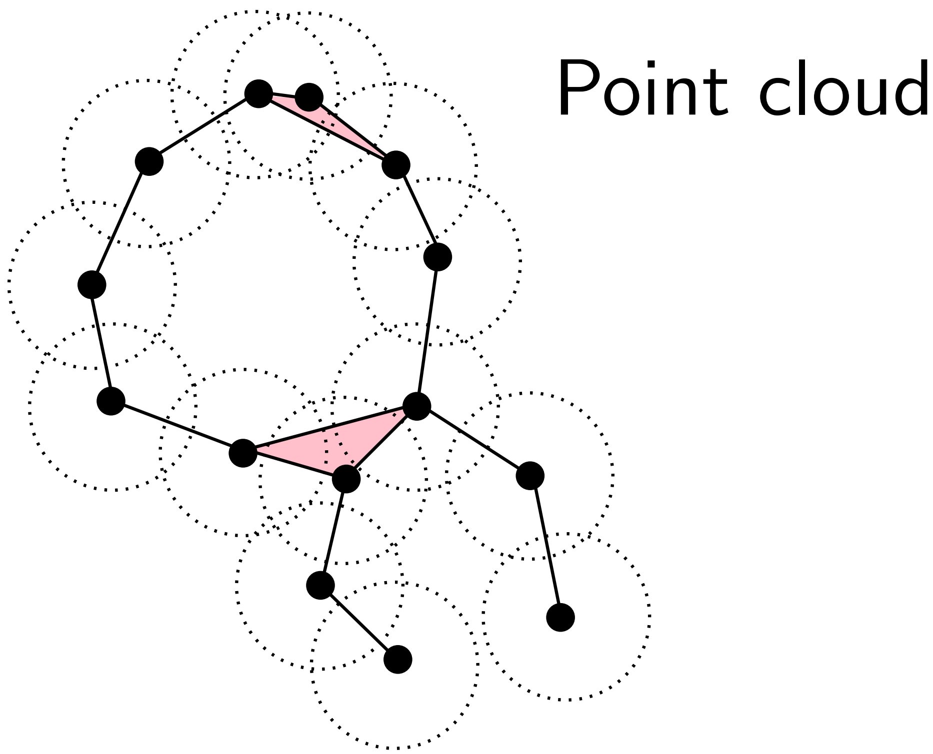
Persistent diagrams for distance functions



Persistent diagrams for distance functions

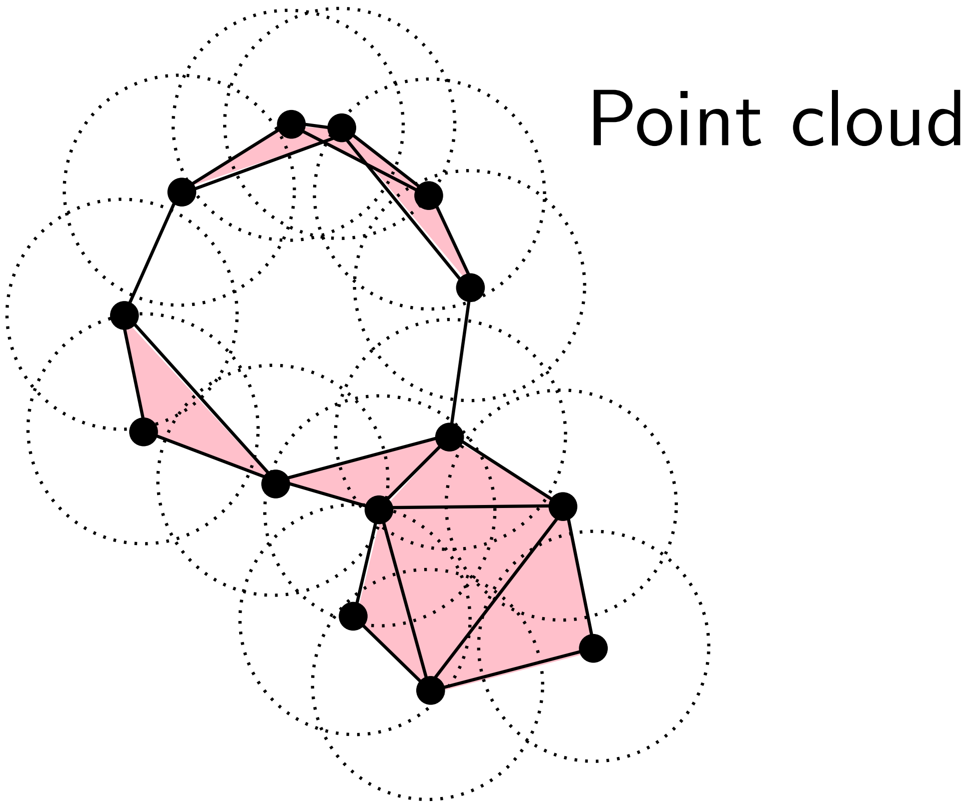


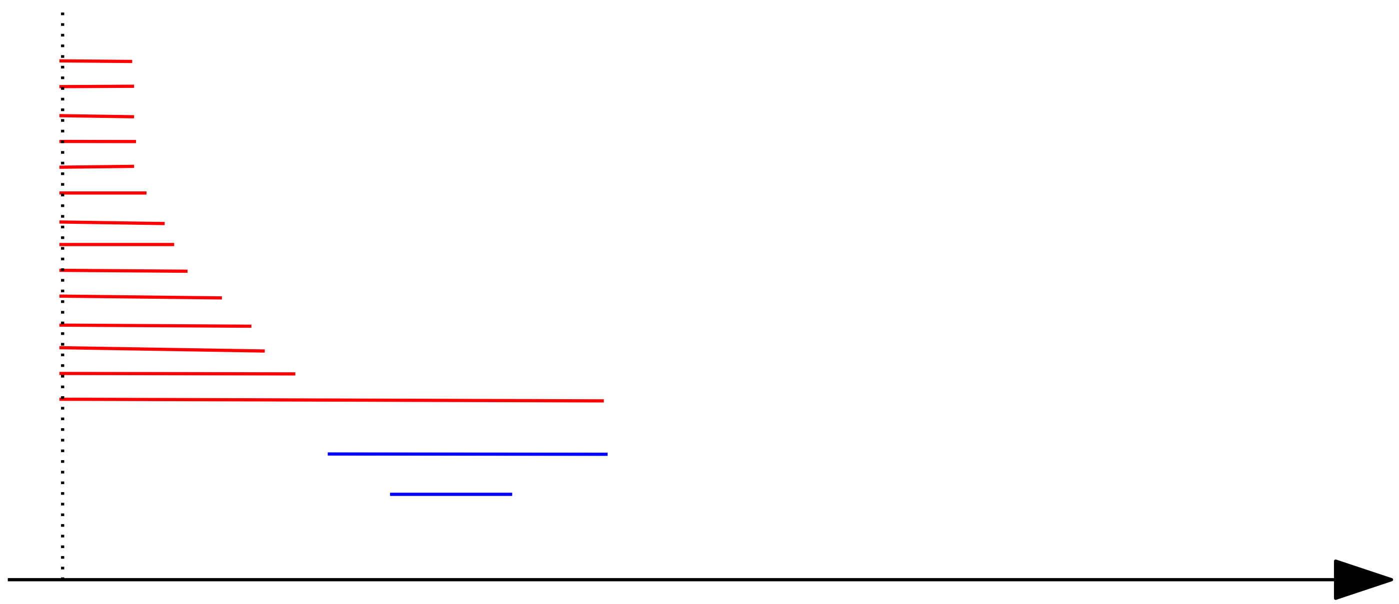
Persistent diagrams for distance functions



1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

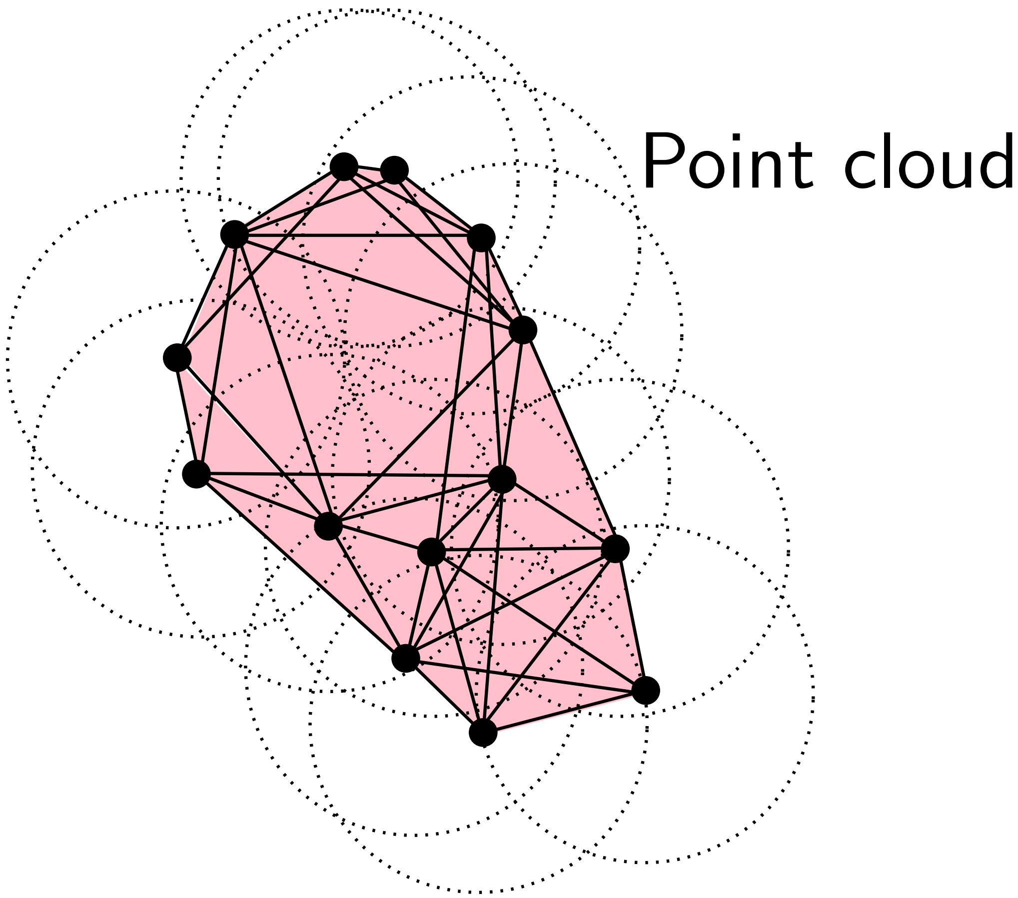
Persistent diagrams for distance functions

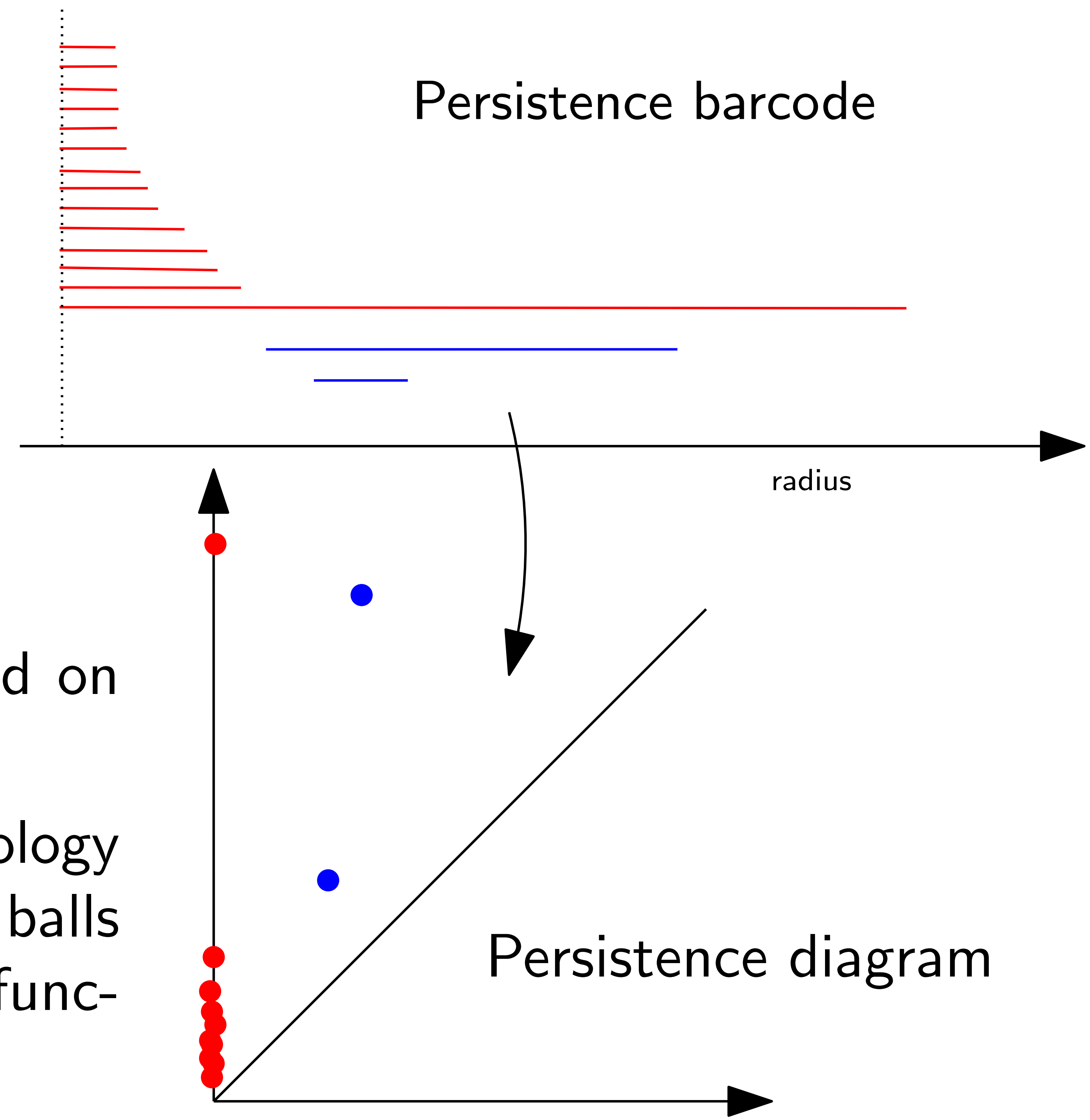




1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

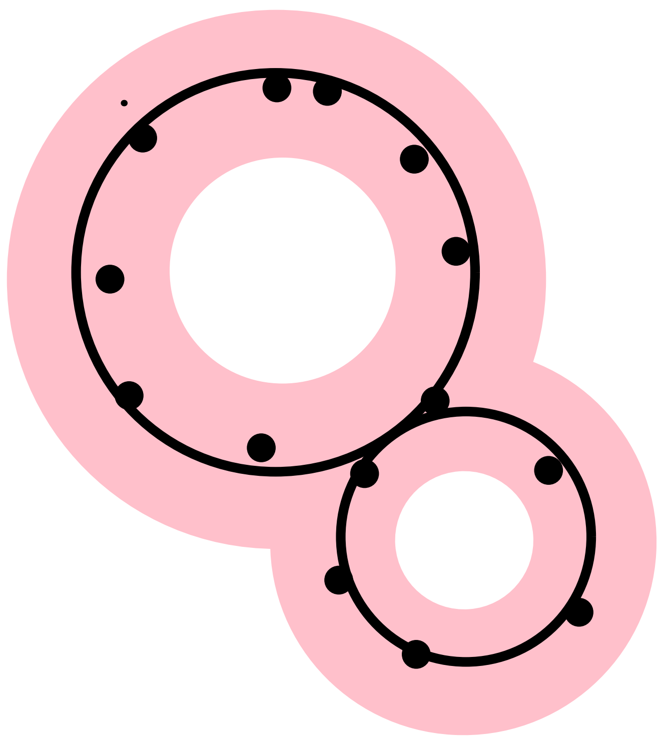
Persistent diagrams for distance functions





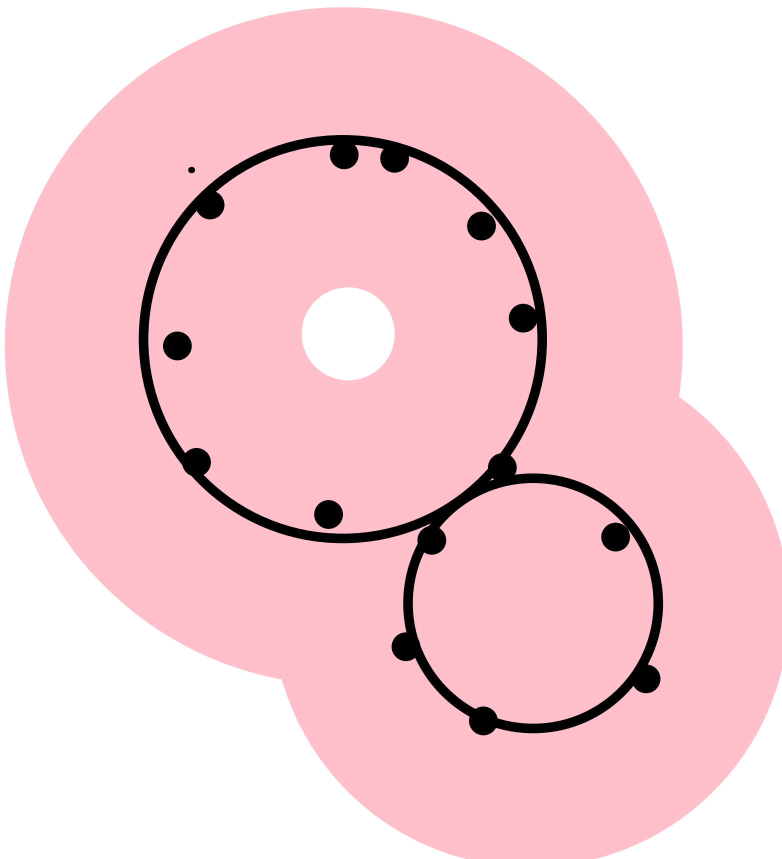
1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

Persistent diagrams for distance functions



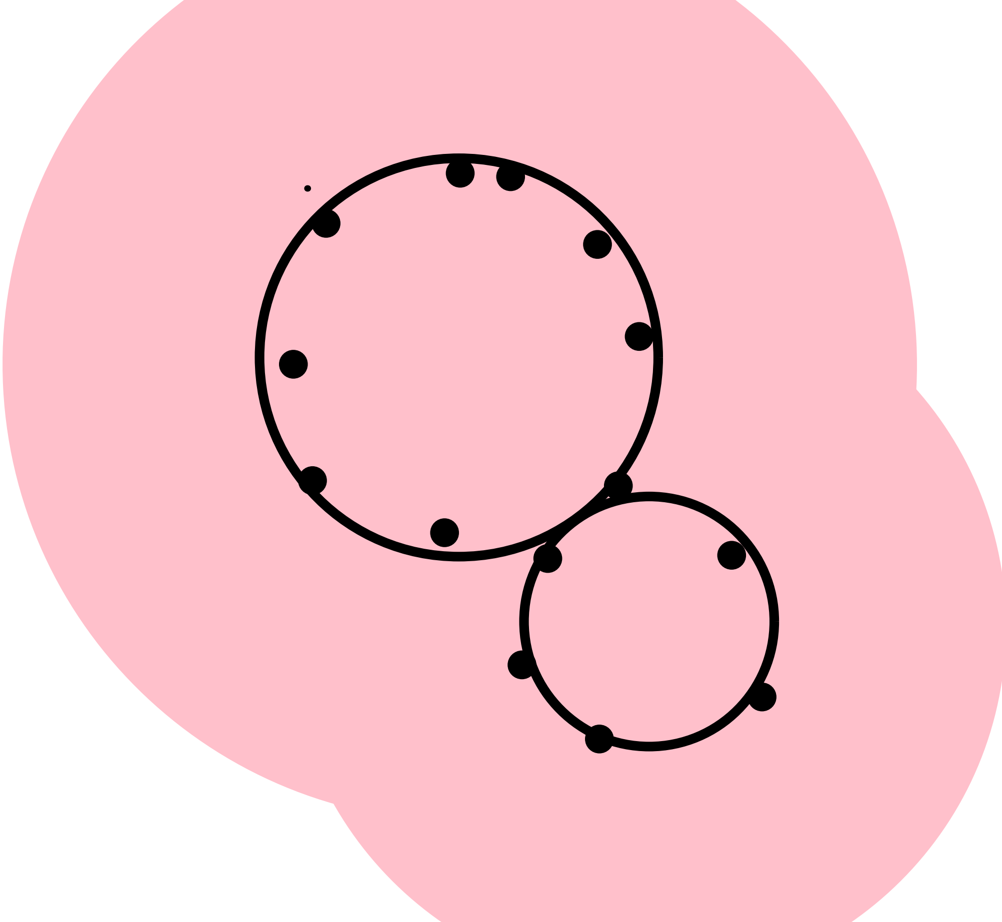
1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

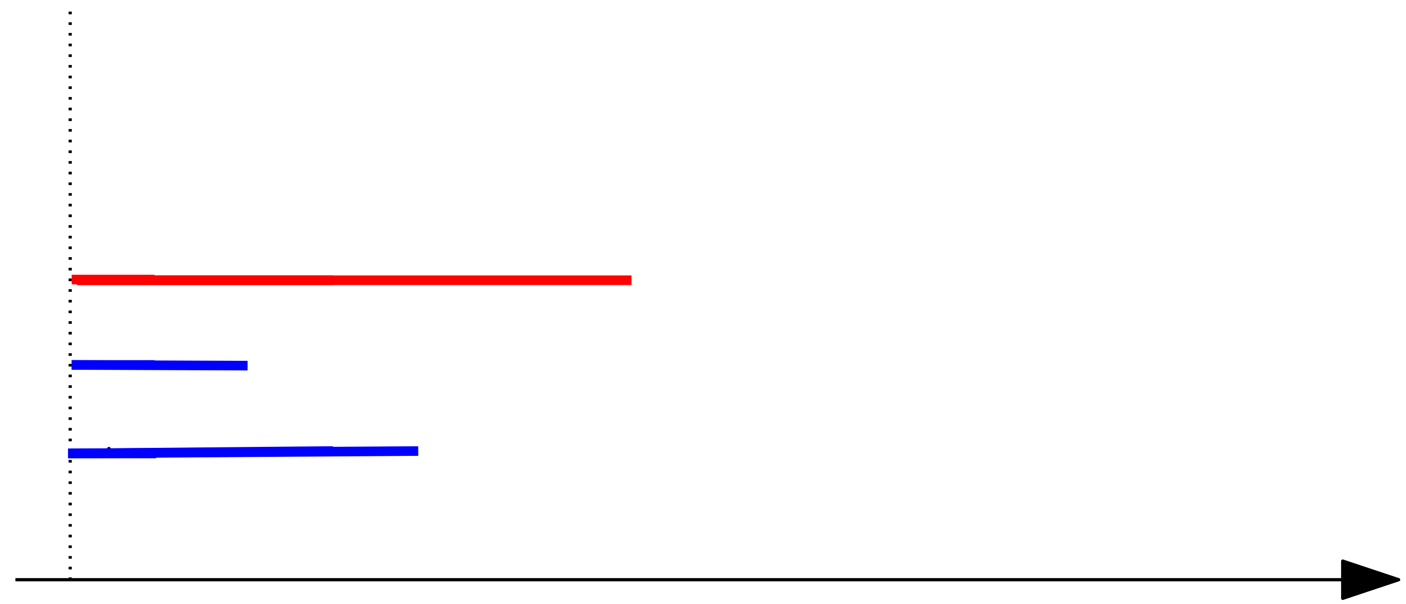
Persistent diagrams for distance functions

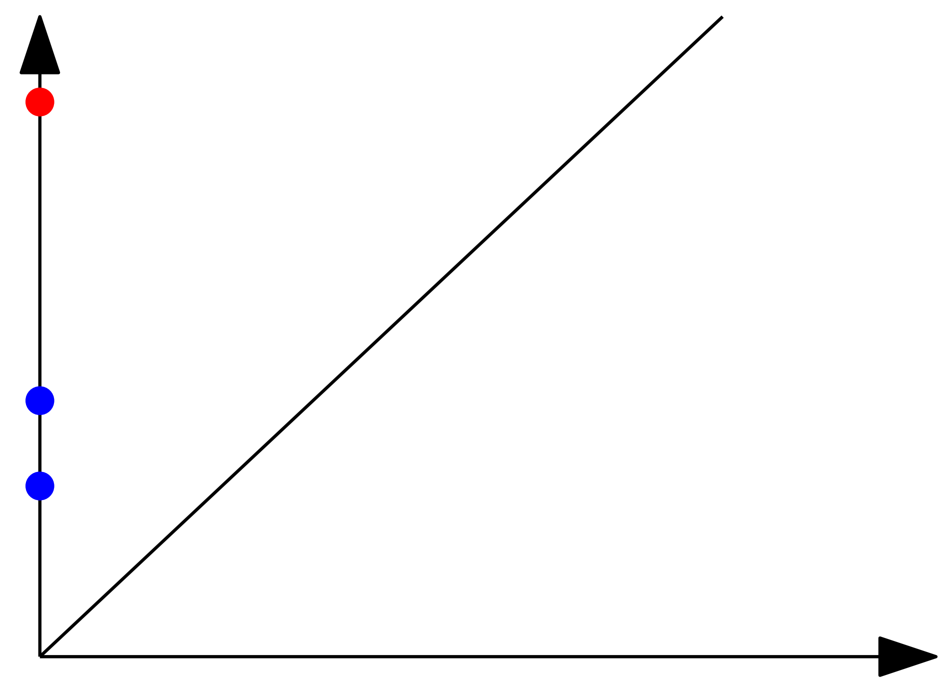


1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

Persistent diagrams for distance functions







1. Grow a family of balls centered on the data (set) of interest.
2. Track the evolution of the topology (homology) of the union of balls (sublevel sets of the distance function).
3. Persistence barcodes/diagrams : encode the topological information.

A zoo of representations of persistence

(non exhaustive list)

- Collections of 1D functions

  $\rightarrow$ landscapes [Bubenik 2012]

  $\rightarrow$ Betti curves [Umeda 2017]

- discrete measures : (interesting statistical properties [Chazal, Divol 2018])

  $\rightarrow$ persistence images [Adams et al 2017]

  $\rightarrow$ convolution with Gaussian kernel [Reininghaus et al. 2015] [Chepushtanova et al. 2015] [Kusano Fukumisu Hiraoka 2016-17] [Le Yamada 2018]

  $\rightarrow$ sliced on lines [Carrière Oudot Cuturi 2017]

- finite metric spaces [Carrière Oudot Ovsjanikov 2015]

- polynomial roots or evaluations [Di Fabio Ferri 2015] [Kališnik 2016]

- etc...