

Unsupervised anomaly detection in time series with topological data analysis

Laurent Oudre

Centre Borelli, ENS Paris-Saclay, University Paris-Saclay, France
joint work with Alexandre Bois and Brian Tervil

DASS Workshop
September 8th, 2026

Contents

1. Introduction and motivation

2. Model and problem statement

3. Background on TDA

4. Algorithm

5. Experiments and results

6. Conclusion

The Centre Borelli



The Centre Borelli

ENS Paris-Saclay, University Paris-Saclay, France

Fusion of two labs :

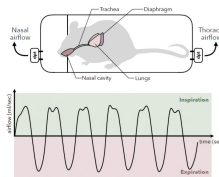
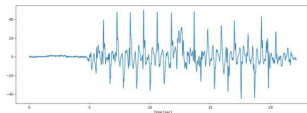
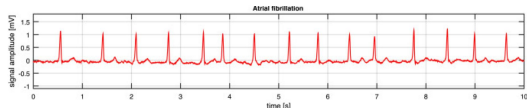
- ▶ The Centre de mathématiques et de leurs applications (CMLA) : applied mathematics for the study of complex phenomena and data
- ▶ The Cognition & Action Group (CognacG) : quantification and study of human and animal behavior

Main scientific questions

How to quantify human and animal behavior

- ▶ Adventure launched since 2012 : interdisciplinary collaboration between mathematicians, physicians, neuroscientists, engineers, biologists, etc...
- ▶ Implementation of measurement chains “pipelines”, platforms and intelligent tools but also of procedures for analysis, measurement and processing of data
- ▶ Creation of tools for diagnostic assistance, inter-individual comparison and longitudinal follow-up
- ▶ Integration into a clinical environment and interaction between algorithms and medical/neuroscience experts

Protocols and signals



Different protocols and signals: electrocardiogram (heart activity), accelerometry (gait analysis), plethysmography (respiratory activity)

What is an anomaly?

- ▶ Complex question: each method has its own (often implicit) definition
- ▶ Take home messages from large benchmarks [Schmidl et al., 2022; Paparrizos et al., 2025]: no consensus, supervised methods struggle to generalize (scarce annotations and difficulty to detect new types of anomalies), performances of unsupervised techniques dramatically depend on the datasets
- ▶ **Our conclusion:** It is impossible to create a completely generic anomaly detection algorithm: it must be tailored to the generative models of our data. What matters is to know which type of anomalies we are looking for.

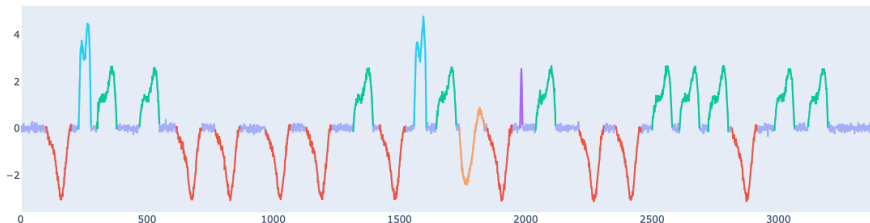
Our signals, our anomalies

- ▶ In monitoring and healthcare, the signal is made of **repetitive patterns** of possibly varying length (e.g. a heartbeat in an ECG)
- ▶ Some patterns occur frequently (the **normal** patterns), while others are rare (the **anomalies**)
- ▶ Mathematically, our data mixes a **normal behaviour** with **anomalies**: i.e. points or sub-sequences that depart from it

Unsupervised setting

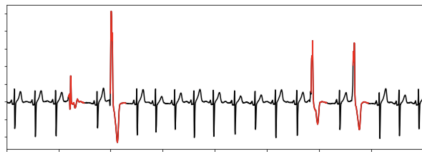
We work with no labels and no annotations. Each point will receive an **anomaly score**: an application-dependent threshold turns it into a binary answer.

Normal patterns and rare anomalies

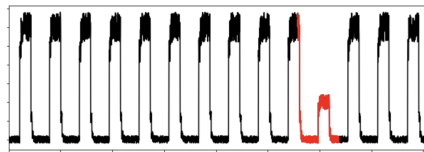


- ▶ Each colour is a recurring *atom*. The **green** and **red** patterns each occur ten times: **normal**
- ▶ The **blue**, **orange** and purple ones occur once or twice: **anomalies**
- ▶ Atoms can differ widely in length

Some real life examples



ECG - ventricular premature contractions appear as **anomalous beats**



NAB - server metrics, traffic & web data with labelled **events**

Our definition: a structured, repetitive activity in which pathologies or incidents manifest as deviations from the recurring pattern.

State-of-the-art for unsupervised anomaly detection

Many unsupervised detectors, each encoding a different notion of **normality**

Density / distance

LOF, HBOS, Isolation Forest: flag points in low-density regions [Breunig 2000; Liu 2008]

Subsequence-based

Matrix Profile, DAMP, NORMA: compare subsequences to a normal shape [Yeh 2018]

Reconstruction

Auto-encoders, LSTM, CNN: prediction / reconstruction error as a score [Malhotra 2015]

Open questions: varying lengths, multiple normal behaviours, repeated anomalies and noise

Contents

1. Introduction and motivation

2. Model and problem statement

2.1 Continuous model

2.2 Discrete model

2.3 Problem statement

3. Background on TDA

4. Algorithm

5. Experiments and results

6. Conclusion

Continuous model

- ▶ Let $\mathbf{x} = (\mathbf{x}[t])_{t \in [0,1]}$ and $\epsilon = (\epsilon[t])_{t \in [0,1]}$ be real-valued functions defined on $[0, 1]$. We call $\mathbf{x}$ the *signal* and ϵ is called the *noise*.
- ▶ We use a **convolutional sparse coding** model, where $\mathbf{x}$ is composed of atoms (some normal, some abnormal)

$$\mathbf{x} = \underbrace{\sum_{i=1}^{M_n} n_i * \eta_i}_{\text{normal}} + \underbrace{\sum_{i=1}^{M_a} a_i * \alpha_i}_{\text{anomalies}}$$

where $*$ is the convolution operator.

- ▶ The observed signal is

$$\mathbf{y} = \mathbf{x} + \epsilon$$

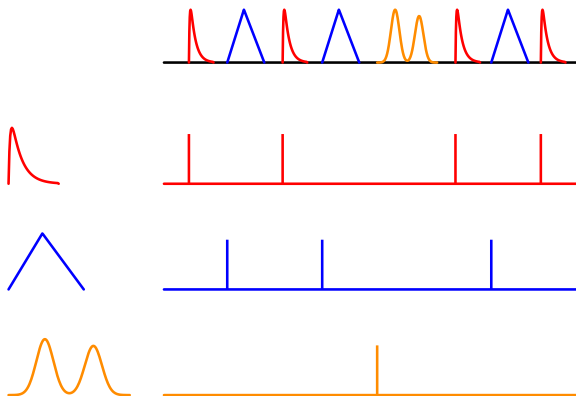
where ϵ is a noise term.

Continuous model

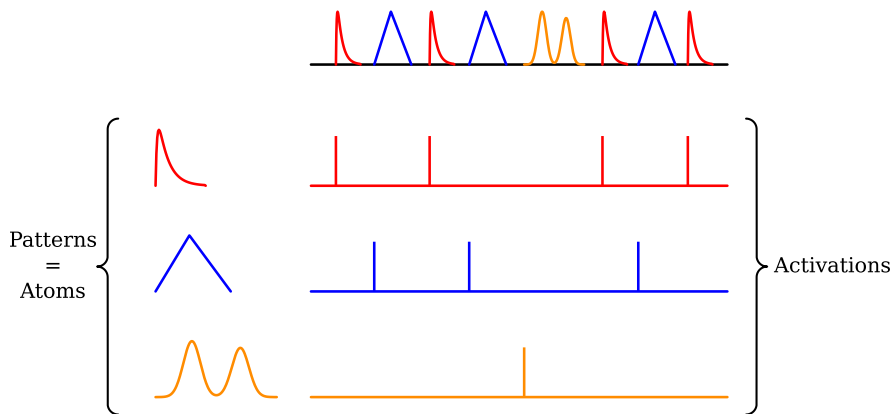
$$\mathbf{x} = \underbrace{\sum_{i=1}^{M_n} n_i * \eta_i}_{\text{normal}} + \underbrace{\sum_{i=1}^{M_a} a_i * \alpha_i}_{\text{anomalies}}$$

- ▶ The $\mathbf{n}_i$ and $\mathbf{a}_i$ are respectively **normal** and **abnormal** atoms of length $l_{n_i} < 1$ (resp. l_{a_i}). Note that atoms have *a priori* different lengths.
- ▶ There are M_n normal atoms and M_a abnormal atoms.
- ▶ The η_i and α_i are the **activations**: binary time series that indicate the starting time of each occurrence.
- ▶ The number of occurrences of atom $\mathbf{n}_i$ (resp. $\mathbf{a}_i$) is denoted by $k_{n_i} = \|\eta_i\|_0$ (resp. $k_{a_i} = \|\alpha_i\|_0$)

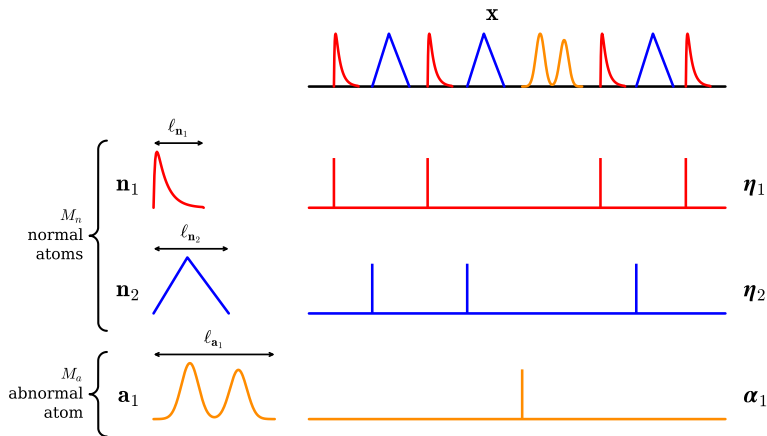
Illustration



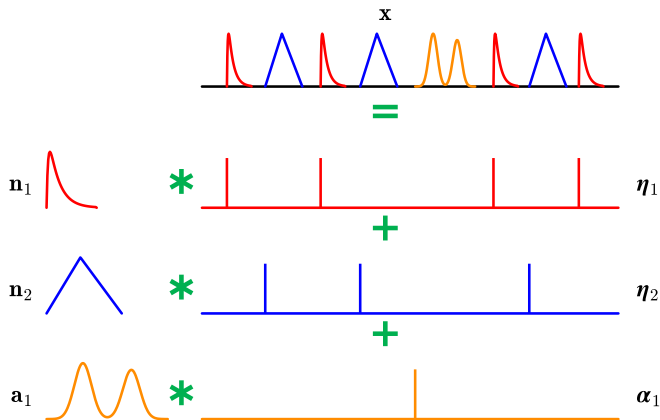
Illustration



Illustration



Illustration



Discrete model

- ▶ In practice we do not work with continuous functions in $[0, 1]$ but on sampled time series
- ▶ We will denote $\hat{\mathbf{x}}$ the uniform sampling of size n of signal $\mathbf{x}$, thus giving the discrete model

$$\hat{\mathbf{x}} = \sum_{i=1}^{M_n} \hat{\mathbf{n}}_i * \hat{\boldsymbol{\eta}}_i + \sum_{i=1}^{M_a} \hat{\mathbf{a}}_i * \hat{\boldsymbol{\alpha}}_i$$

- ▶ The observed time series is $\hat{\mathbf{y}} = \hat{\mathbf{x}} + \hat{\boldsymbol{\varepsilon}}$

Main assumptions

Disjoint supports

Atom occurrences do not overlap: every point is *either* normal *or* abnormal

Noise

The infinite norm of the noise $\|\epsilon\|_\infty$ is small

Boundary conditions

Atoms always start and end at 0. $\mathbf{n}_i(0) = \mathbf{n}_i(l_{\mathbf{n}_i}) = 0$ (resp. $\mathbf{a}_i(0) = \mathbf{a}_i(l_{\mathbf{a}_i}) = 0$)

Rare anomalies

$\max_i k_{\mathbf{a}_i} \ll \min_i k_{\mathbf{n}_i}$: abnormal atoms are **rare**

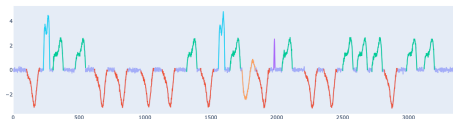
The anomaly detection problem

Anomaly detection

Find every index $i \in [1, n]$ such that $\hat{\mathbf{x}}_a[i] \neq 0$, i.e. recover all the **abnormal activations and atom lengths**.

- ▶ The ground-truth answer is a **binary vector** of length n
- ▶ The model allows **several** normal behaviours and **repeating** anomalies
- ▶ Like most detectors, we output a continuous **anomaly score** per point
- ▶ A threshold (left to the application) gives the decision

How to solve it



- ▶ The problem seems to be connected to **pattern detection** and classical methods such as Matrix Profile should work...
- ▶ But those methods **struggle to adapt to the variability of the data** (varying lengths, dilation/contraction of the shapes)
- ▶ How could we **automatically detect recurring patterns and shapes** in our time series?

Geometric properties of the time series → Topological Data Analysis (TDA)

Contents

1. Introduction and motivation

2. Model and problem statement

3. Background on TDA

3.1 Delay embeddings

3.2 Simplices, simplicial complexes and filtrations

3.3 Persistent homology and diagrams

3.4 DTM-weighted filtrations

4. Algorithm

5. Experiments and results

6. Conclusion

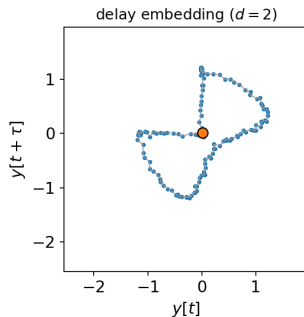
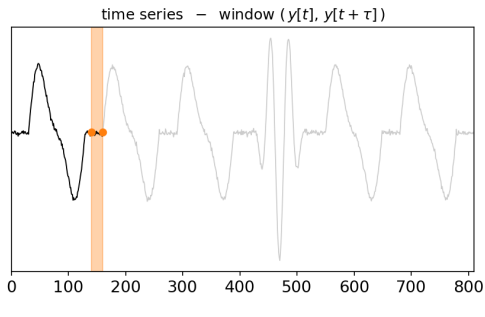
Delay embeddings

- ▶ The first step is to change the representation domain of the time series
- ▶ **Delay embedding:** turn a 1-D signal into a curve in $\mathbb{R}^d$ by stacking time-shifted copies

$$\hat{X}_{d,\hat{\tau}}[i] = (\hat{x}[i], \hat{x}[i + \hat{\tau}], \dots, \hat{x}[i + (d - 1)\hat{\tau}]) \in \mathbb{R}^d$$

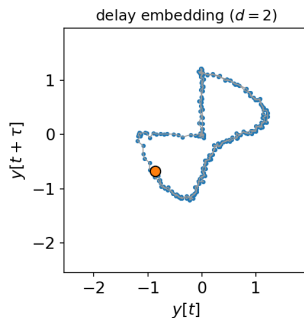
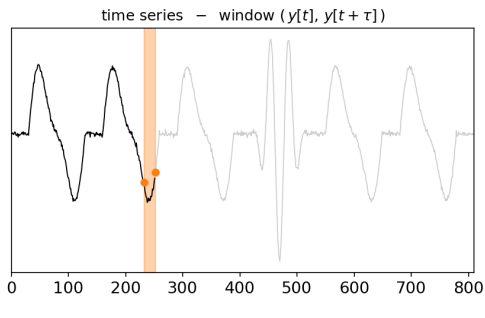
- ▶ **Two parameters:** dimension d and delay $\hat{\tau}$
- ▶ Classical tool from dynamical systems: the embedding reconstructs the underlying geometry of the dynamics [Takens 1981]
- ▶ A **periodic** signal traces a closed curve: a **loop** [Sauer 1991]
- ▶ Each atom returning to 0 gives **at least one loop**

Delay embedding in action



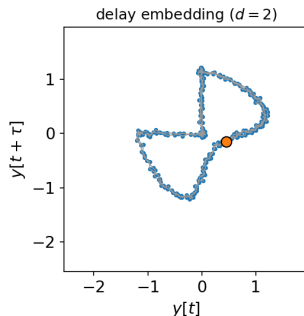
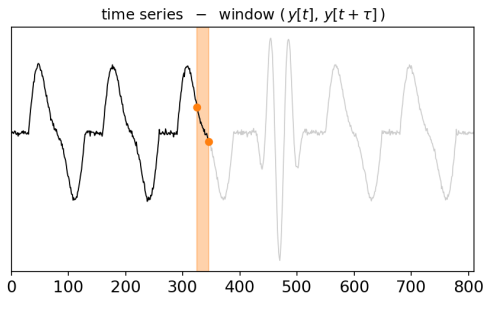
A sliding window $(y[t], y[t + \tau])$ maps each time t to a point in $\mathbb{R}^d$ (here $d = 2$).

Delay embedding in action



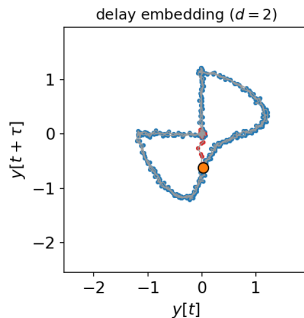
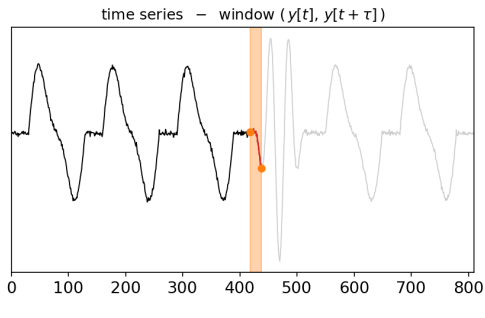
A sliding window $(y[t], y[t + \tau])$ maps each time t to a point in $\mathbb{R}^d$ (here $d = 2$).

Delay embedding in action



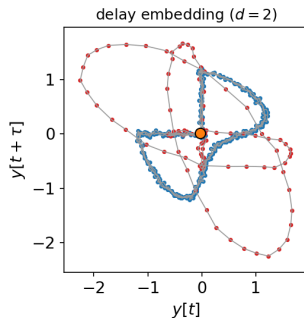
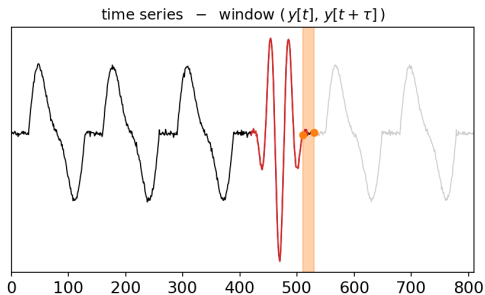
Each occurrence of the **normal pattern** retraces the **same loop**: points accumulate, the loop becomes **dense**.

Delay embedding in action



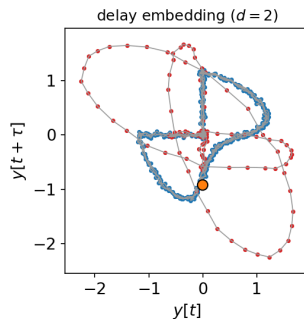
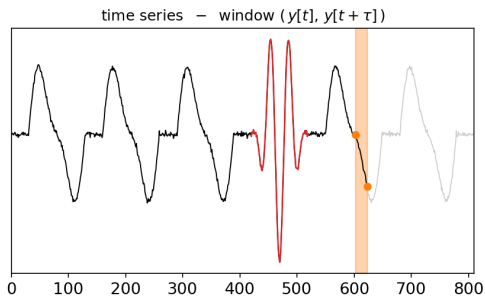
The **anomaly** draws a **different, sparse** curve, visited only once.

Delay embedding in action



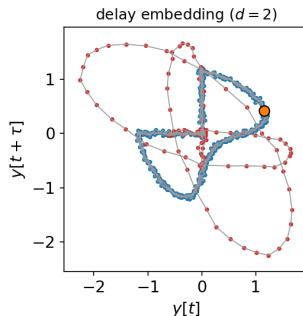
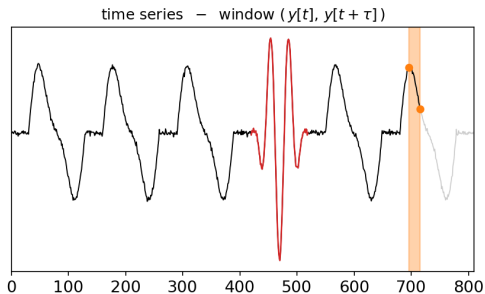
The **anomaly** draws a **different, sparse** curve, visited only once.

Delay embedding in action



The **anomaly** draws a **different, sparse** curve, visited only once.

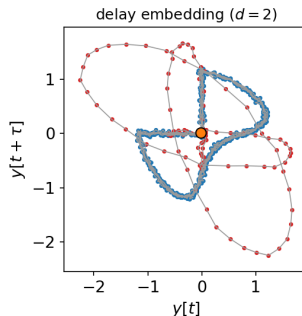
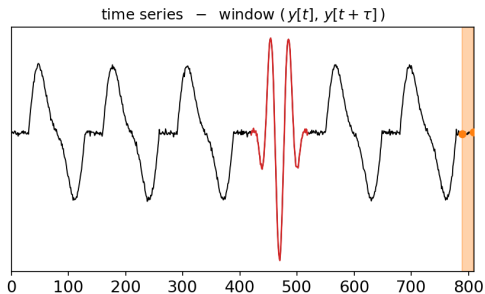
Delay embedding in action



Atoms start/end at 0 $\Rightarrow$ every pattern yields a loop through the origin.

Normal = dense loop, abnormal = sparse excursion.

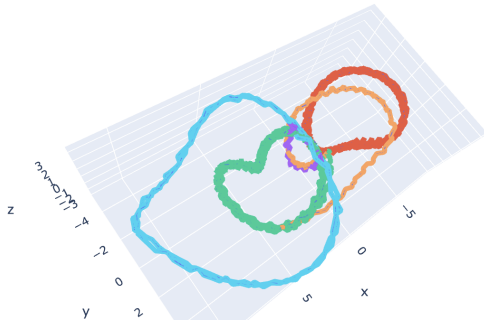
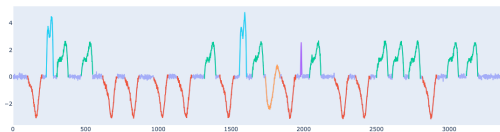
Delay embedding in action



Atoms start/end at 0 $\Rightarrow$ every pattern yields a loop through the origin.

Normal = dense loop, abnormal = sparse excursion.

Patterns are loops



Delay embedding of the toy signal (PCA to 3-D), $d = 10$, $\hat{\tau} = 4$.

Patterns are loops

Each **pattern** in the data forms a **loop** in the $\mathbb{R}^d$ space

Density separates

Green and **red** loops (many occurrences) are **thick & dense**. Rare loops are **thin & sparse**.

This will be the core of our approach: study the geometrical properties of the point cloud in $\mathbb{R}^d$

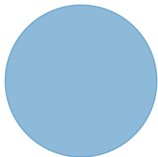
Topological data analysis

- ▶ **Topological data analysis** (TDA) is a set of techniques derived from algebraic topology that describe the structure of a set of points through a nested family of **simplicial complexes**, called a **filtration** and indexed by a scale parameter α
- ▶ The **persistence diagram** records, for each topological feature, the value of α at which it appears and the value at which it disappears: connected components in dimension 0, **loops** in dimension 1
- ▶ These techniques provide theoretical stability guarantees: the diagram is controlled under sampling, noise and outliers, with explicit bounds

More info on [\[Chazal 2021, Bois, 2026\]](#)

TDA: studying the shape of data

disk: $\beta_0 = 1, \beta_1 = 0$



annulus: $\beta_0 = 1, \beta_1 = 1$



$\beta_0 = 2, \beta_1 = 1$



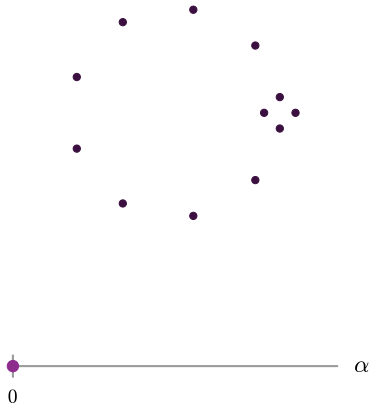
$\beta_0 = \# \text{ connected components}, \beta_1 = \# \text{ independent loops (1-cycles) = holes}$

- ▶ A point cloud is just a discrete set: its topology is trivial! ($\beta_0 = \text{number of points}$)
- ▶ We need to build **shapes** from it by connecting the dots

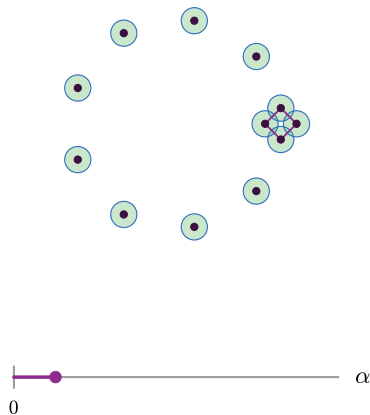
Notion of filtration

- ▶ Roughly speaking, in our context a **filtration** is a strategy to **connect the dots** at every scale at once: for each value of a parameter α , a rule says which points are close enough to be joined, and the shapes obtained **grow with** α , each one containing all the previous ones.
- ▶ Popular choice: **Vietoris-Rips filtration**: a ball around each point, growing with α . As α grows:
 - ▶ Two points get **linked** as soon as their balls touch.
 - ▶ Three points that are pairwise linked get **filled in**: the triangle they form becomes solid.

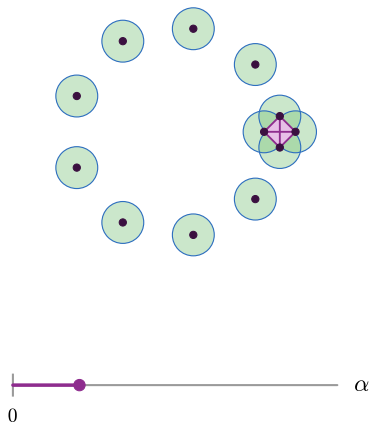
Connecting the dots at every scale



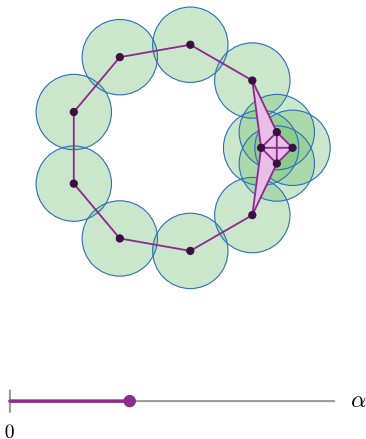
Connecting the dots at every scale



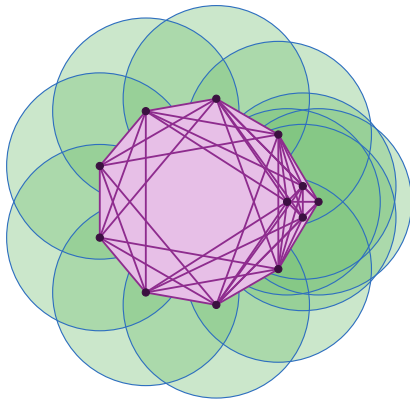
Connecting the dots at every scale



Connecting the dots at every scale



Connecting the dots at every scale

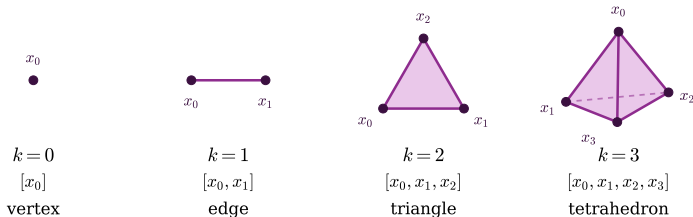


Formal definition: simplices

The animation only used three kinds of objects: **points**, **segments** between two points, and **filled triangles** between three.

k -simplex

A **k -simplex** on $X \subset \mathbb{R}^d$ is an unordered tuple $\sigma = [x_0, \dots, x_k]$ of $k + 1$ distinct points of X , called its **vertices**. A **face** of σ is the simplex spanned by a subset of its vertices.



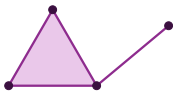
A simplex is entirely described by the **list of its vertices**: no geometry is stored, only which points go together.

Formal definition: simplicial complexes

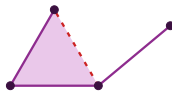
At a fixed scale α , the animation produced a **collection** of such simplices. Not any collection will do: a filled triangle cannot be there without its three edges.

Simplicial complex

A **simplicial complex** K is a set of simplices such that any face of a simplex of K is a simplex of K .



every face is there



one edge is missing

In our previous example, filtration gives this **for free**: we fill a triangle only once its three edges are already present.

From simplices to simplicial complex

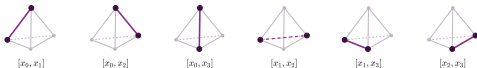
$k = 0$

4 vertices



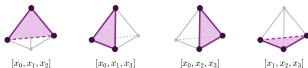
$k = 1$

6 edges



$k = 2$

4 triangles



$k = 3$

1 tetrahedron



Take the tetrahedron

$\sigma = [x_0, x_1, x_2, x_3]$ and list the simplices spanned by the subsets of its vertices:

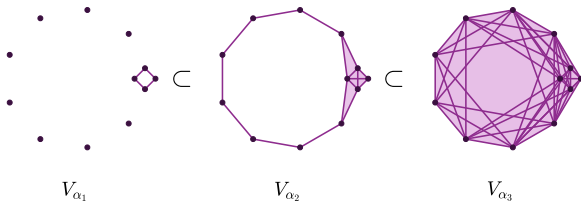
- ▶ Taken alone, $\{\sigma\}$ is **not** a simplicial complex.
- ▶ The set of all its faces, drawn here, is one.

Formal definition: filtrations

One complex per scale, and the animation showed that **nothing built at a given scale is ever undone** at a larger one.

Filtration

A **filtration** is a family $V = (V_\alpha)_{\alpha \in \mathbb{R}^+}$ of simplicial complexes such that $\alpha \leq \beta \Rightarrow V_\alpha \subset V_\beta$.



Every simplex therefore has a **filtration value**: the smallest α at which it enters, and it never leaves.

Formal definition: the Vietoris-Rips filtration

The one we have been using: grow a ball of radius $\alpha/2$ around each point, and link two points as soon as their balls meet.

Vietoris-Rips filtration

$$[x_0, \dots, x_k] \in \text{Rips}(X, \alpha) \iff \forall i, j, \|x_i - x_j\| \leq \alpha$$

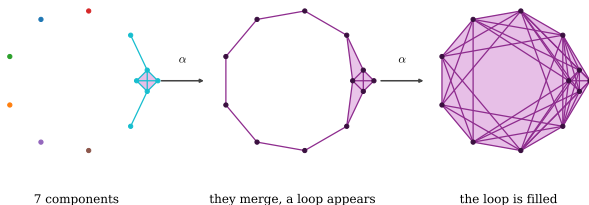
- ▶ Only the **edges** have to be checked: a tetrahedron enters as soon as its 6 edges are there, its 4 triangles follow automatically.

Birth and death

We now have **one simplicial complex V_α per scale**. On each of them we can count the number β_0 of connected components and β_1 of loops.

Birth and death

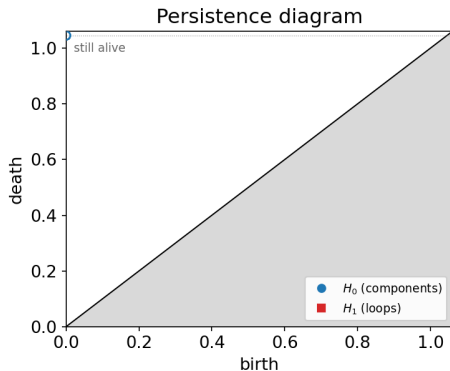
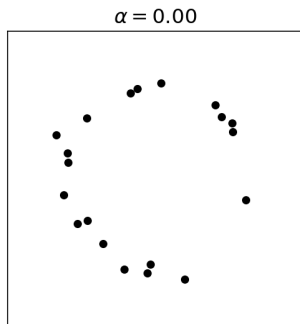
- ▶ **A connected component** is born at $\alpha = 0$, one per point. It dies when it **merges** with another one.
- ▶ **A loop** is born when the last of its edges enters the complex. It dies when enough triangles have entered to **pave** the surface it encloses



Persistence homology and diagram

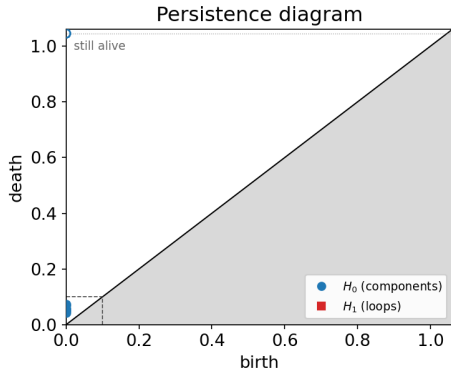
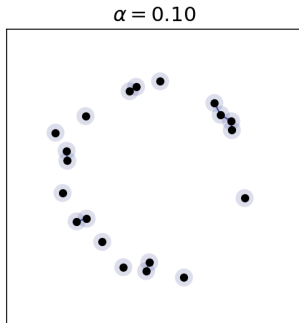
- ▶ Simplices are only added and never removed. What disappears is a **connected component** that is no longer alone, or a **loop** that is no longer empty.
- ▶ Along the filtration, each component and each loop spans an interval of scales. These intervals are the **persistent homology** of the filtration.
- ▶ The **persistence diagram** $\text{Diag}(\mathbb{V})$ is the multiset of points (birth, death), one per homology class.

Persistent homology in action



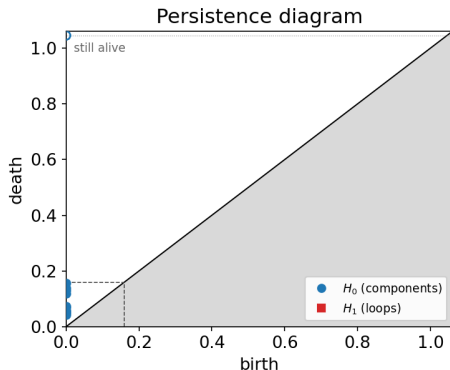
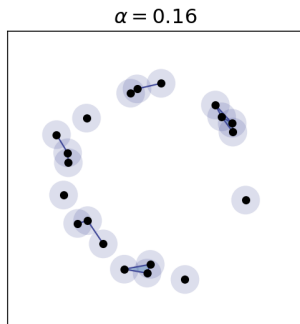
Small α : many components (H_0), no loop yet.

Persistent homology in action



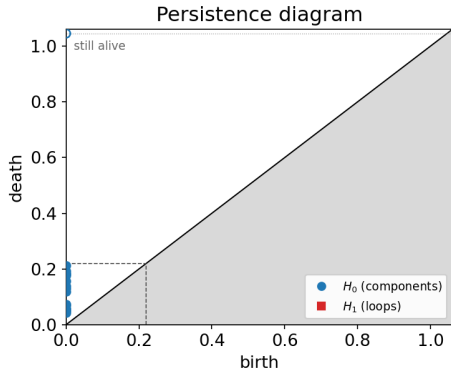
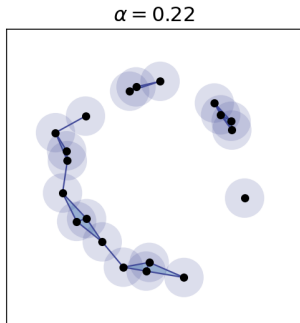
Small α : many components (H_0), no loop yet.

Persistent homology in action



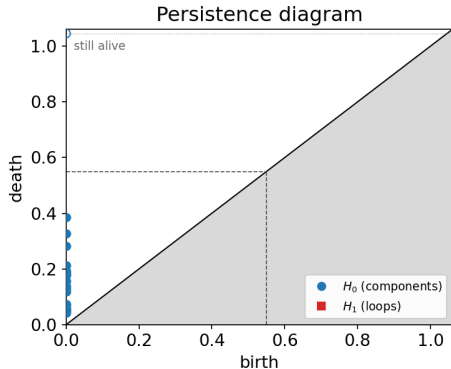
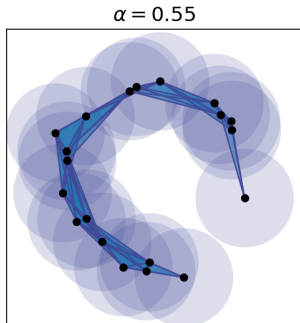
Small α : many components (H_0), no loop yet.

Persistent homology in action



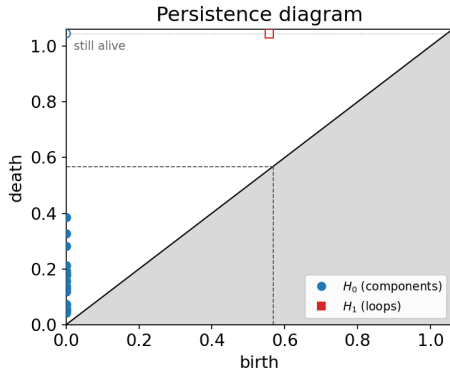
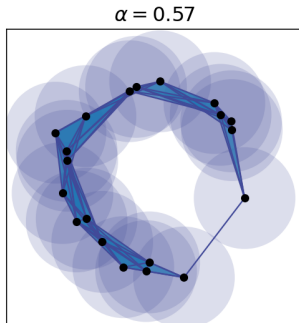
Small α : many components (H_0), no loop yet.

Persistent homology in action



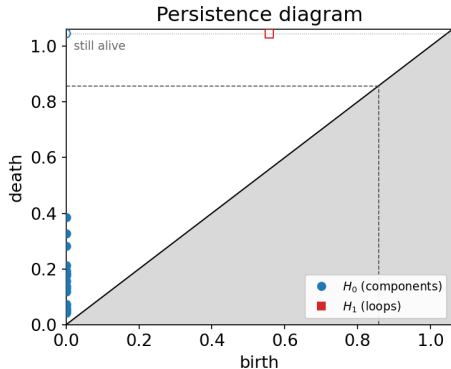
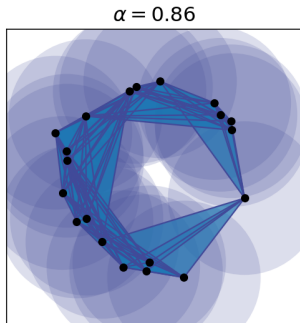
Small α : many components (H_0), no loop yet.

Persistent homology in action



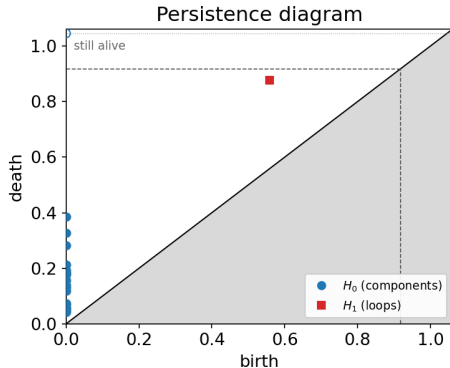
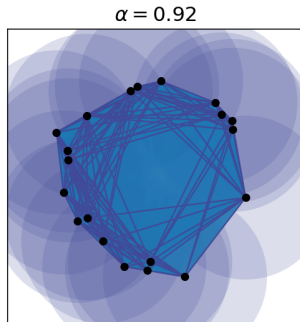
Components merge; the big loop is **born** when the circle closes up.

Persistent homology in action



The loop **persists** over a long range of scales.

Persistent homology in action



The loop **dies** when triangles fill it. One component never dies: past a certain scale the whole cloud is a single piece, so there is nothing left for it to merge with.

Why is it useful for us?

- ▶ **0-dim classes** = connected components, **1-dim classes** = loops;
- ▶ Each 1-dim class admits a **representative cycle**: an explicit list of edges (hence of data points!) realizing the loop.
- ▶ **Stability properties**: Small perturbation of the data $\Rightarrow$ small perturbation of the diagram.

TDA can help us to detect these loops (by tracking the H_1 features, i.e. the loops), but also to characterize them

Can we use this approach directly?

- ▶ Rips filtrations treat **every point equally**: a few outliers can create spurious components or destroy the structure.
- ▶ In our anomaly detection setting, anomalies *are* rare points: we need a filtration that **sees density**, not just geometry.

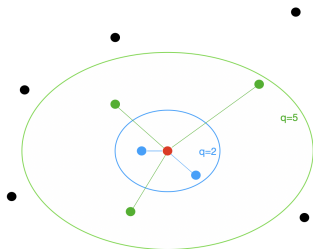
Idea

Weight each point by its **distance to a measure (DTM)**: balls around dense points grow first, balls around rare points are delayed.

The distance to a measure (DTM)

- For the empirical measure $\mu_{\hat{X}}$ of a finite cloud $\hat{X}$ we compute:

$$d_{\mu_{\hat{X}},m}(x) = \sqrt{\frac{1}{q} \sum_{i=1}^q \|x - \text{NN}^{(i)}(x)\|^2} \quad (\text{mean square distance to the } q \text{ NN})$$

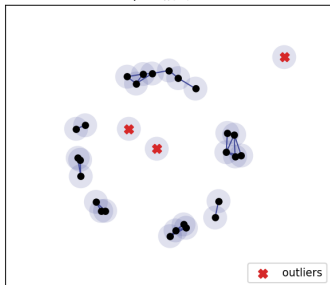


- $d_{\mu,m}(x)$ is **low** in dense regions, **high** for isolated points.
- Robust: moving a small mass of μ changes $d_{\mu,m}$ little
- Idea: create a new filtration!** The ball around point x stays empty until $\alpha = d_{\mu,m}(x)$ then grows

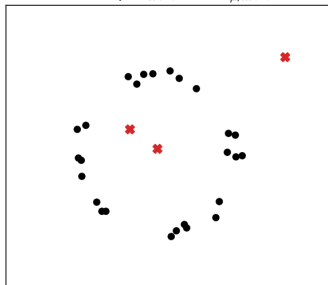
DTM-weighted filtrations

$\alpha = 0.15$

Rips: $r_x(\alpha) = \alpha$



DTM-Rips: $r_x(\alpha) = \alpha - d_{\mu,m}(x)$

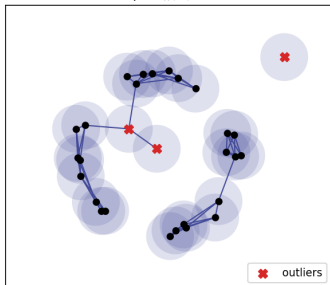


Left: all balls grow together, outliers included.

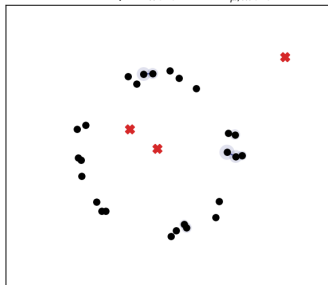
DTM-weighted filtrations

$$\alpha = 0.30$$

Rips: $r_x(\alpha) = \alpha$



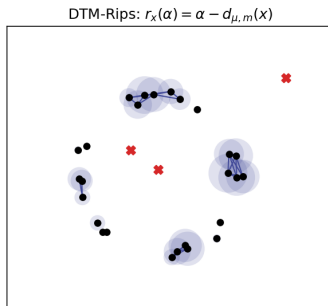
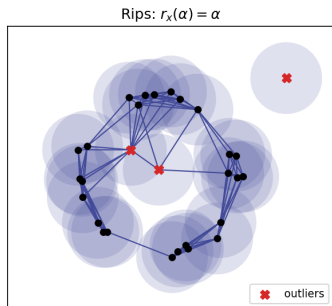
DTM-Rips: $r_x(\alpha) = \alpha - d_{\mu,m}(x)$



Left: all balls grow together, outliers included.

DTM-weighted filtrations

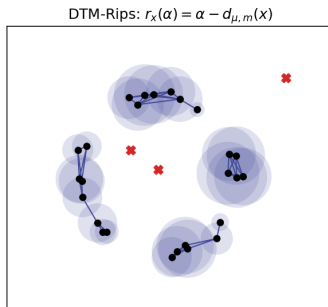
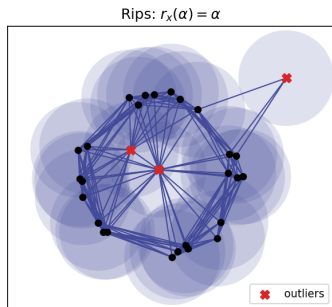
$\alpha = 0.45$



Right: dense circle points activate first; **outliers**, with high DTM, are still asleep.

DTM-weighted filtrations

$\alpha = 0.60$

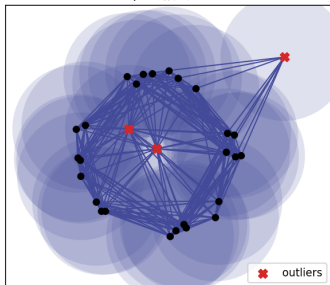


Right: dense circle points activate first; **outliers**, with high DTM, are still asleep.

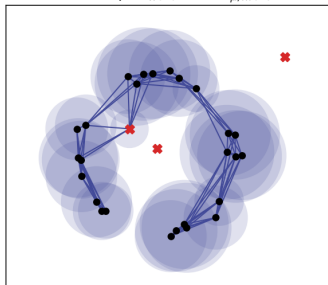
DTM-weighted filtrations

$$\alpha = 0.80$$

Rips: $r_x(\alpha) = \alpha$



DTM-Rips: $r_x(\alpha) = \alpha - d_{\mu,m}(x)$

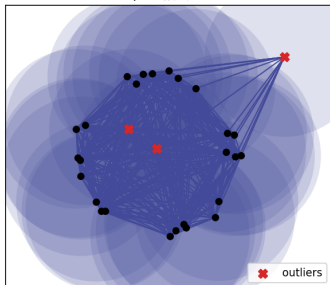


The loop forms from dense points only $\rightarrow$ its **birth date reflects density**. Outliers join (much) later.

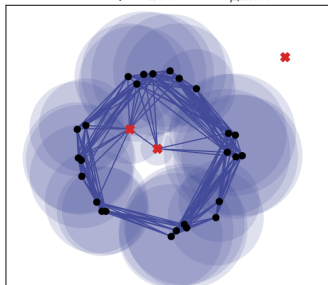
DTM-weighted filtrations

$$\alpha = 1.00$$

Rips: $r_x(\alpha) = \alpha$



DTM-Rips: $r_x(\alpha) = \alpha - d_{\mu,m}(x)$



The loop forms from dense points only $\rightarrow$ its **birth date reflects density**. Outliers join (much) later.

Filtration values

The whole **DTM-weighted filtration** is characterized by vertices and edges:

- ▶ vertex $[x]$ enters at $\alpha = d_{\mu,m}(x)$;
- ▶ edge $[x, y]$ enters at

$$\begin{cases} \max(d_{\mu,m}(x), d_{\mu,m}(y)) & \text{if } \|x - y\| < |d_{\mu,m}(x) - d_{\mu,m}(y)| \\ \frac{\|x - y\| + d_{\mu,m}(x) + d_{\mu,m}(y)}{2} & \text{otherwise.} \end{cases}$$

Why this matters for us

The **birth date** of a 1-cycle now encodes the **density** of the points forming it:

- ▶ loop made of frequent (normal) points → low birth date;
- ▶ loop involving rare (abnormal) points → high birth date.

Contents

1. Introduction and motivation

2. Model and problem statement

3. Background on TDA

4. Algorithm

4.1 Overview

4.2 Embedding, subsampling and filtration

4.3 Identifying normal cycles on the diagram

4.4 Extracting representative cycles

4.5 Computation of the anomaly scores

4.6 Theoretical guarantees

5. Experiments and results

6. Conclusion

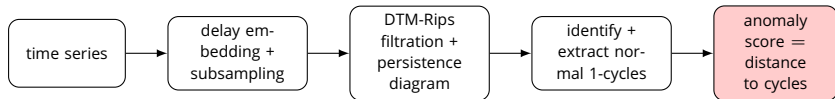
Principle of the algorithm

Principle of the algorithm

Detect the **dense, recurring loops** of the delay embedding as the normal behaviour and score every point by its **distance** to them.

- ▶ Normal points: many occurrences $\Rightarrow$ **small DTM** $\Rightarrow$ loops **born early**
- ▶ Abnormal points: rare $\Rightarrow$ **large DTM** $\Rightarrow$ loops **born late**, or **no loop**
- ▶ Reading **1-D persistence** focuses on a few structured cycles
- ▶ An atom that varies little **from one window to the next** draws a **small** loop: it gets filled early, and the persistence threshold discards it.

Overview



1. **Embed & subsample:** delay embedding $\hat{Y}$; keep n_{points} well-spread points
2. **DTM Rips filtration:** density-weighted, q neighbours
3. **Identify cycles:** two thresholds on the diagram select **normal** loops
4. **Score:** anomaly score = distance to the extracted cycles

Step 1: embedding, subsampling, filtration

- ▶ Compute $\hat{Y} = \hat{Y}_{d, \hat{\tau}} (n - (d - 1)\hat{\tau})$ points: too large for persistent homology ($O(n^3)$ worst case).
- ▶ **Greedy subsampling** $\tilde{Y} \subset \hat{Y}$ with n_{points} points (farthest point sampling).
- ▶ Compute the **DTM**-Rips filtration $V = \text{Rips}[\tilde{Y}, d_{\mu_{\hat{Y}}, m}, 1]$

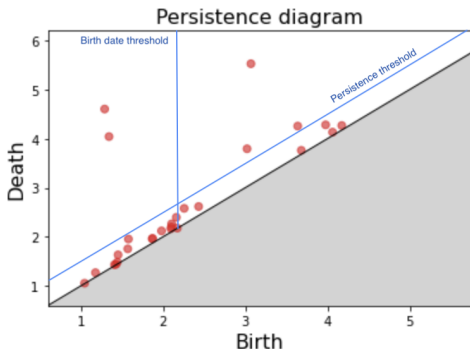
Key trick

The filtration is built on the **subsample** $\tilde{Y}$, but the DTM is computed w.r.t. the **full cloud** $\hat{Y}$: density information (number of occurrences!) is preserved.

Complexity: $O(n^3) \rightarrow O(n_{\text{points}}^3 + n^2)$. but good stability properties !

Intuition for q : if $\max_i(k_{a_i}) < q \leq \min_i(k_{n_i})$, the DTM is ≈ 0 on normal points and strictly higher on abnormal ones

Step 2: identifying normal cycles on the diagram



Here : two *normal* cycles

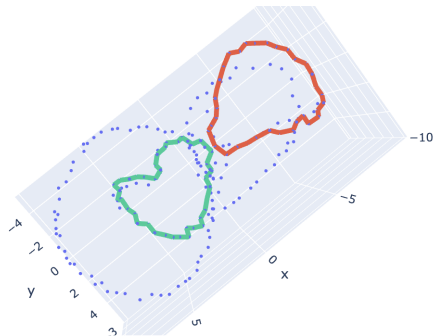
Two automatic thresholds:

1. **Persistence threshold:** sort persistences in decreasing order, cut at the largest gap (+ n_{diag} extra points kept).
2. **Birth date threshold:** among the persistent points, cut at the largest *birth date* gap (with safeguards: most persistent cycle always kept; threshold = $+\infty$ if all births are close).

Normal cycles = persistent **and** early-born (upper-left region).

Step 3: extracting representative cycles

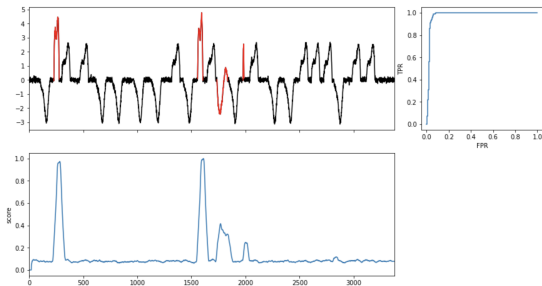
- ▶ The diagram gives the scale at which a hole appears and disappears, but not the points that form it.
- ▶ Extracting a **representative cycle** = recovering the list of edges that close the loop.
- ▶ Standard matrix reduction [Edelsbrunner et al., 2002] with no additional computation (the diagram already contains the cycles)
- ▶ Output: $L_{cycles} = (c_1, \dots, c_{n_{cycles}})$, the **normal set**.



Subsample ($n_{points} = 200$) and the two extracted normal cycles: the green and red atoms are recovered.

Step 4: anomaly scores

- ▶ Score of an embedded point $x \in \hat{Y}$: $d(x, L_{cycles})$
- ▶ Score of a time point $\mathbf{y}[i]$: average over all windows containing it
- ▶ Binary answer by thresholding: evaluation below via **AUC-ROC** (threshold-free)



Example series: anomaly score (bottom) and ROC curve, $AUC = 0.98$.

What can we guarantee?

- ▶ We never look for anomalies: we build a picture of **normality**, and score by distance to it.
- ▶ **Question:** is that picture right? The anomalies are in the cloud too.
- ▶ **Theorem:** We can prove that our filtration stays close to the one we would get on clean data, with an explicit bound.
- ▶ Three sources of error, all controlled: **sampling** ($\rightarrow 0$ with n), **noise** (proportional to its amplitude), **anomalies** (proportional to the *share* of the signal the anomalies occupy).
- ▶ Close filtrations $\rightarrow$ close diagrams $\rightarrow$ the right cycles

Important: what counts in the theorem is the *share* of the signal the anomalies occupy. One long anomaly is enough to pollute normality.

Contents

1. Introduction and motivation
2. Model and problem statement
3. Background on TDA
4. Algorithm
- 5. Experiments and results**
 - 5.1 Experimental setup**
 - 5.2 Results**
6. Conclusion

Experimental setup

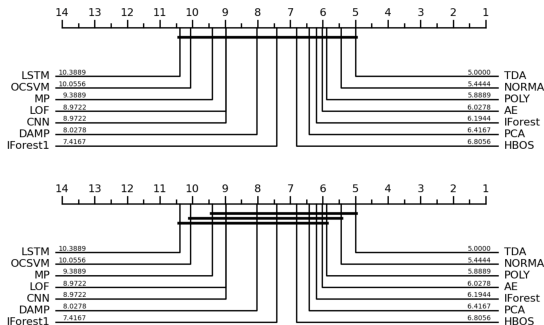
- ▶ **Real-world benchmark:** TSB-UAD suite (Paparrizos et al., 2022): 18 public datasets, 1980 labeled univariate series (ECG, server metrics, spacecraft telemetry, ...).
- ▶ **Competitors:** 13 methods (IForest, LOF, HBOS, Matrix Profile, DAMP, NORMA, PCA, AE, LSTM, CNN, POLY, OCSVM, ...).
- ▶ **Metric:** AUC-ROC per series, averaged per dataset (threshold-free).
- ▶ **Parameter heuristics** (same rule for all series):
 - ▶ period L estimated by autocorrelation; $\hat{\tau} = 6$; $d = \max(40, \min(120, \lfloor L/3 \rfloor))$ so that $d\hat{\tau} \approx$ pattern length;
 - ▶ $q = \lfloor n/L \rfloor$ ($\approx$ number of normal occurrences); $n_{points} = 400$.
- ▶ For additional experiments: **synthetic dataset:** 120 series generated from the model (punctual/sequential anomalies, single/multiple normal atoms, repeated anomalies).

Results on TSB-UAD (average AUC-ROC)

	TDA	IForest	IForest1	LOF	MP	PCA	NORMA	HBOS	POLY	OCSVM	AE	CNN	LSTM	DAMP
Dodgers	0.79	0.79	0.64	0.54	0.52	0.77	0.79	0.30	0.69	0.64	0.73	0.68	0.39	0.62
ECG	0.88	0.75	0.61	0.56	0.58	0.71	0.95	0.68	0.70	0.64	0.73	0.52	0.54	0.72
IOPS	0.82	0.54	0.78	0.50	0.72	0.74	0.76	0.64	0.68	0.71	0.63	0.61	0.61	0.50
MGAB	0.58	0.57	0.58	0.96	0.91	0.54	0.55	0.54	0.51	0.52	0.71	0.58	0.56	0.55
NAB	0.76	0.45	0.56	0.48	0.49	0.69	0.58	0.68	0.75	0.61	0.54	0.52	0.50	0.54
NASA-MSL	0.64	0.57	0.69	0.52	0.52	0.75	0.55	0.77	0.81	0.64	0.70	0.57	0.57	0.67
NASA-SMAP	0.83	0.72	0.68	0.68	0.62	0.74	0.80	0.77	0.80	0.65	0.77	0.68	0.64	0.77
SensorScope	0.52	0.56	0.56	0.55	0.50	0.54	0.59	0.56	0.62	0.51	0.52	0.52	0.53	0.56
YAHOO	0.64	0.62	0.81	0.86	0.86	0.57	0.92	0.57	0.76	0.50	0.79	0.96	0.94	0.78
KDD21	0.75	0.65	0.57	0.78	0.90	0.58	0.88	0.60	0.58	0.60	0.79	0.74	0.66	0.76
Daphnet	0.70	0.74	0.68	0.78	0.44	0.69	0.46	0.69	0.77	0.45	0.44	0.47	0.44	0.50
GHL	0.86	0.94	0.94	0.54	0.42	0.91	0.64	0.92	0.76	0.45	0.63	0.47	0.47	0.53
Genesis	0.84	0.78	0.66	0.68	0.35	0.85	0.60	0.59	0.87	0.70	0.72	0.73	0.53	0.71
MITDB	0.71	0.70	0.61	0.61	0.69	0.67	0.86	0.70	0.68	0.65	0.80	0.58	0.51	0.66
OPP	0.48	0.49	0.52	0.45	0.82	0.52	0.65	0.54	0.28	0.38	0.70	0.47	0.57	0.65
Occupancy	0.53	0.86	0.78	0.53	0.32	0.78	0.53	0.89	0.80	0.66	0.69	0.79	0.71	0.53
SMD	0.77	0.85	0.73	0.69	0.51	0.80	0.61	0.77	0.87	0.61	0.63	0.61	0.58	0.57
SVDB	0.77	0.72	0.58	0.59	0.74	0.68	0.92	0.71	0.67	0.68	0.79	0.58	0.55	0.70

Best score on **4 datasets**, top-5 on 13/18, best **average rank** overall.

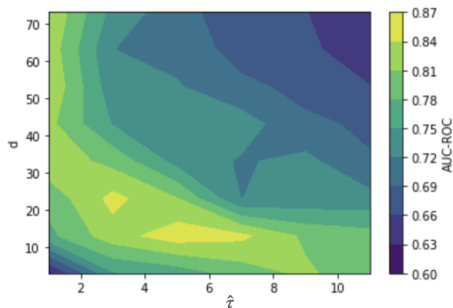
Statistical comparison



Critical diagrams ($\alpha = 0.05$). Top: Friedman + Wilcoxon. Bottom: Friedman + Nemenyi.

- ▶ Best average rank, but the difference with the best competitors is **not statistically significant**.
- ▶ **Failure modes:** data far from the model (no clear normal behavior, trend, binary series), parameter heuristics not universal, subsample too small for very long series.

Influence of the embedding parameters d and $\hat{\tau}$

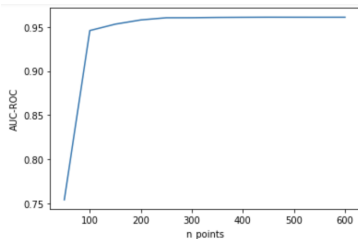


Average AUC-ROC on the synthetic dataset.

d : embedding dimension, τ : delay

- ▶ Best scores along $d\hat{\tau} \approx \text{constant}$: the window length should match the **atom length**
- ▶ Consistent with Perea & Harer (2015) for periodic signals
- ▶ Avoid extremes: $\hat{\tau}$ too small $\rightarrow$ cloud near the diagonal line; too large $\rightarrow$ local shape is lost.

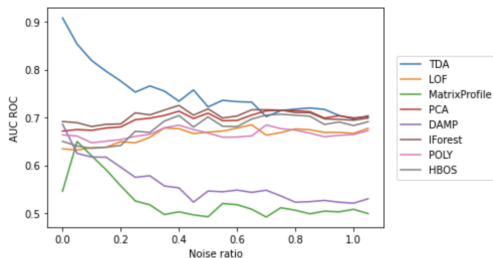
Influence of the subsampling



AUC-ROC as a function of n_{points}

- ▶ AUC plateaus from $n_{points} \approx 200$ ($\approx n/10$): computation $\sim 1000\times$ faster than full persistence, with near-optimal accuracy.
- ▶ $O(n_{points}^3 + n^2)$: n_{points} can grow much slower than n (series of length $> 10^5$ handled with $n_{points} = 400$)

Robustness to noise



Average AUC-ROC vs noise ratio $\text{Var}(\epsilon)/\text{Var}(\mathbf{x})$

- ▶ Best unsupervised method up to 30%, AUC-ROC above 0.8 up to 20%.
- ▶ **Failure mode:** too much noise **fills the loops** → cycles are less persistent → harder to read on the diagram.
- ▶ **Careful:** the synthetic data follow our exact model, so the comparison is biased in our favour.

Contents

1. Introduction and motivation
2. Model and problem statement
3. Background on TDA
4. Algorithm
5. Experiments and results
6. Conclusion

Take-home messages

- ▶ A time series with repetitive patterns becomes, via **delay embedding**, a point cloud where **patterns are loops** and **frequency is density**.
- ▶ The **DTM-Rips filtration** merges both pieces of information: normal patterns appear as **persistent, early-born 1-cycles** on the persistence diagram.
- ▶ Anomaly score = **distance to the extracted normal cycles**: a global, geometric notion of normality (no model fitting, unsupervised).
- ▶ **Theory**: explicit stability bound between the computed filtration and the ideal noiseless/anomaly-free one.
- ▶ **Practice**: competitive with 13 state-of-the-art methods on 18 datasets, robust to noise, scalable thanks to the subsampling trick.

Limitations and perspectives

Limitations

- ▶ Assumes repetitive, structured, continuous data: struggles with trends, binary series, absence of normal cycles
- ▶ Long or slowly-varying anomalies can be missed (visible in the theoretical bound)
- ▶ Parameter heuristics ($d, \hat{\tau}$) are not universal

Perspectives

- ▶ Faster cycle extraction, sparse filtrations (improve the time/accuracy trade-off)
- ▶ Extensions of the model: linear trend, multivariate time series.

A. Bois, B. Tervil, L. Oudre. *A persistent homology-based algorithm for unsupervised anomaly detection in time series*. Transactions on Machine Learning Research, 2024.

Code & data: <https://github.com/Alex-B-4/>

Questions?

References

- ▶ A. Bois, H. Henneuse, B. Tervil, L. Oudre. Persistent homology for time series: a selective review. *Transactions on Machine Learning Research*, 2026.
- ▶ M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander. LOF: identifying density-based local outliers. In *ACM SIGMOD*, pp. 93–104, 2000.
- ▶ F. Chazal, B. Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4:667963, 2021.
- ▶ H. Edelsbrunner, D. Letscher, A. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.
- ▶ F. T. Liu, K. M. Ting, Z.-H. Zhou. Isolation forest. In *IEEE International Conference on Data Mining*, pp. 413–422, 2008.
- ▶ P. Malhotra, L. Vig, G. Shroff, P. Agarwal. Long short term memory networks for anomaly detection in time series. In *ESANN*, 2015.
- ▶ J. Paparrizos, Y. Kang, P. Boniol, R. S. Tsay, T. Palpanas, M. J. Franklin. TSB-UAD: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proceedings of the VLDB Endowment*, 15(8):1697–1711, 2022.
- ▶ J. Paparrizos, P. Boniol, Q. Liu, et al. Advances in time-series anomaly detection: algorithms, benchmarks, and evaluation measures. In *ACM SIGKDD*, pp. 6151–6161, 2025.
- ▶ J. A. Perea, J. Harer. Sliding windows and persistence: an application of topological methods to signal analysis. *Foundations of Computational Mathematics*, 15(3):799–838, 2015.
- ▶ T. Sauer, J. A. Yorke, M. Casdagli. Embedology. *Journal of Statistical Physics*, 65(3):579–616, 1991.
- ▶ S. Schmidl, P. Wenig, T. Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.
- ▶ F. Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, pp. 366–381, Springer, 1981.
- ▶ C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, Z. Zimmerman, D. F. Silva, A. Mueen, E. Keogh. Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. *Data Mining and Knowledge Discovery*, 32:83–123, 2018.