

(Largely pictorial) Intro to Topological Data Analysis



Corporate needs you to find the differences
between this picture and this picture.

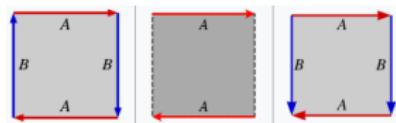


They're the same

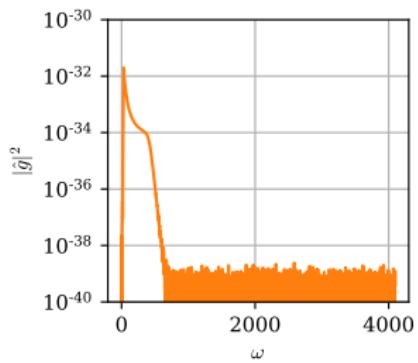
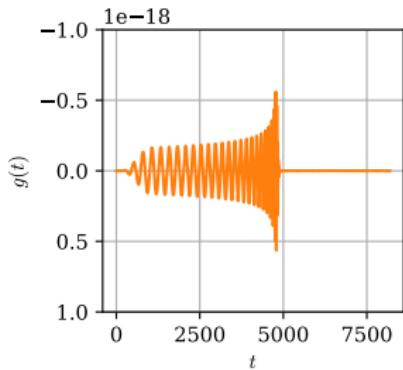
Outline

- ▶ motivation: picking up signal from **noise**?
 - **topological features** i.e., **shape** of data
- ▶ time-series data and **Takens embedding**
- ▶ aspects of TDA
 - aspects of topology
 - **(persistent) homology**
 - topological features for machine learning pipelines
- ▶ extras if time/appetite...

Figure: Fluid flow on a Möbius strip.
From Vanneste (2021), JFM.



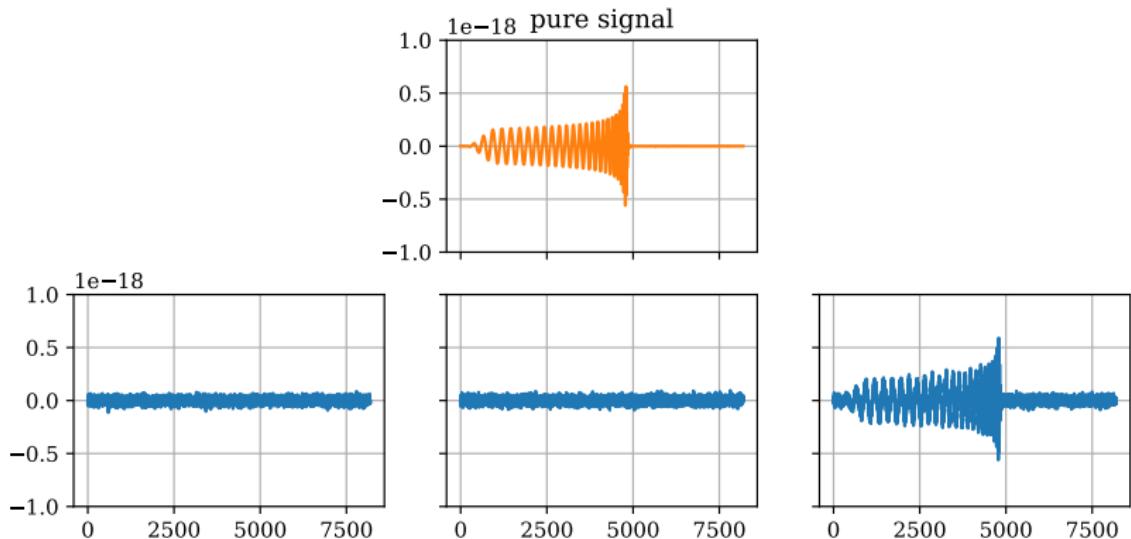
Synthetic examples



$$f = g + \frac{10^{-19}}{R} \xi, \quad \xi \sim N(0, 1)$$

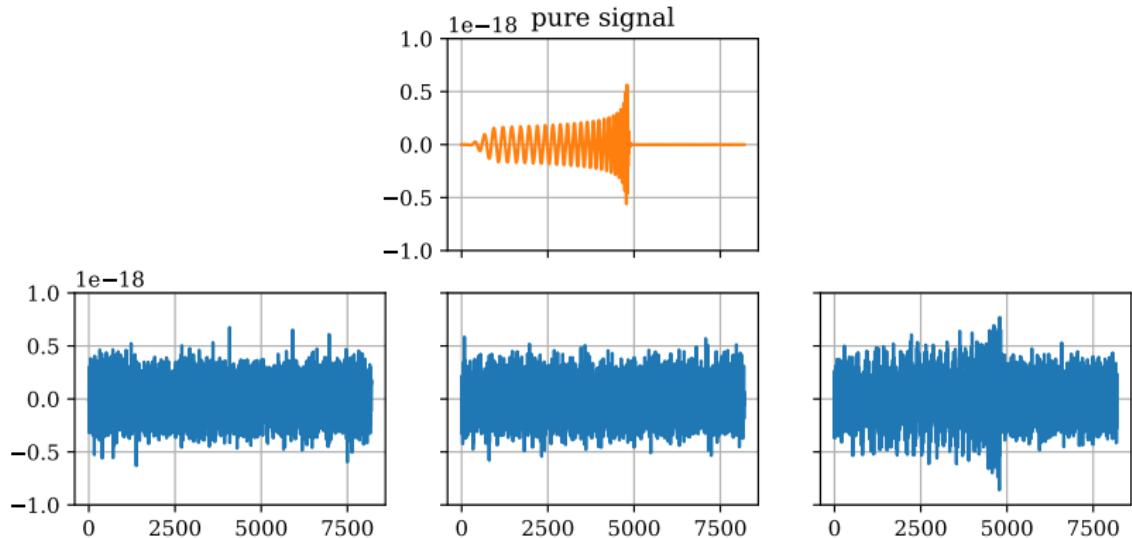
- ▶ f the noisy signal
- ▶ g a synthetic signal
- ▶ $R \in (0.075, 0.65)$ controls the **Signal to Noise Ratio (SNR)**
→ SNR ≈ 10 or below

Synthetic signal



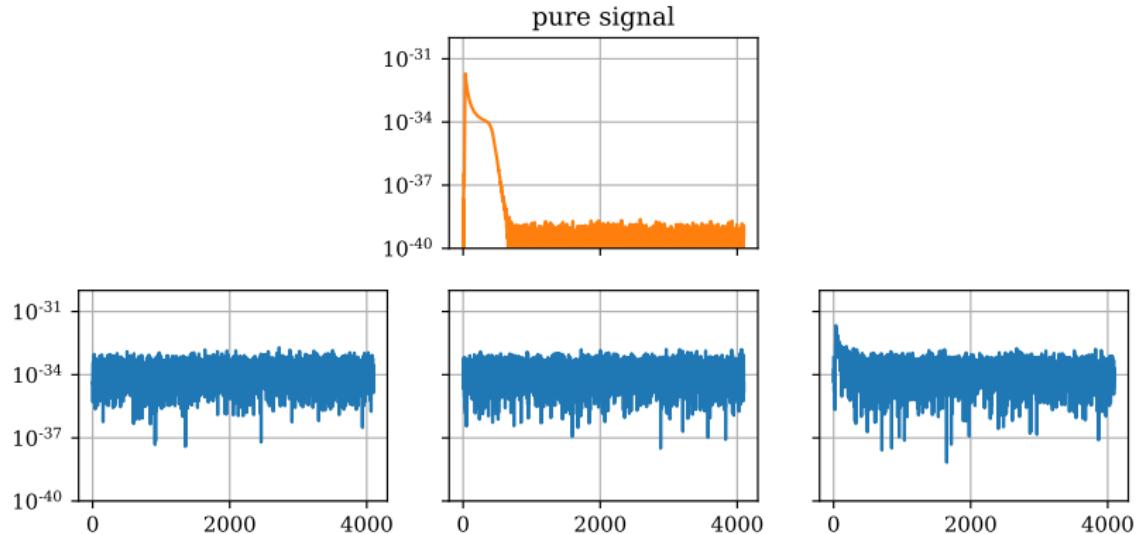
- ▶ Low noise regime ($R = 4$ because why not)
→ “obvious”

Synthetic signal



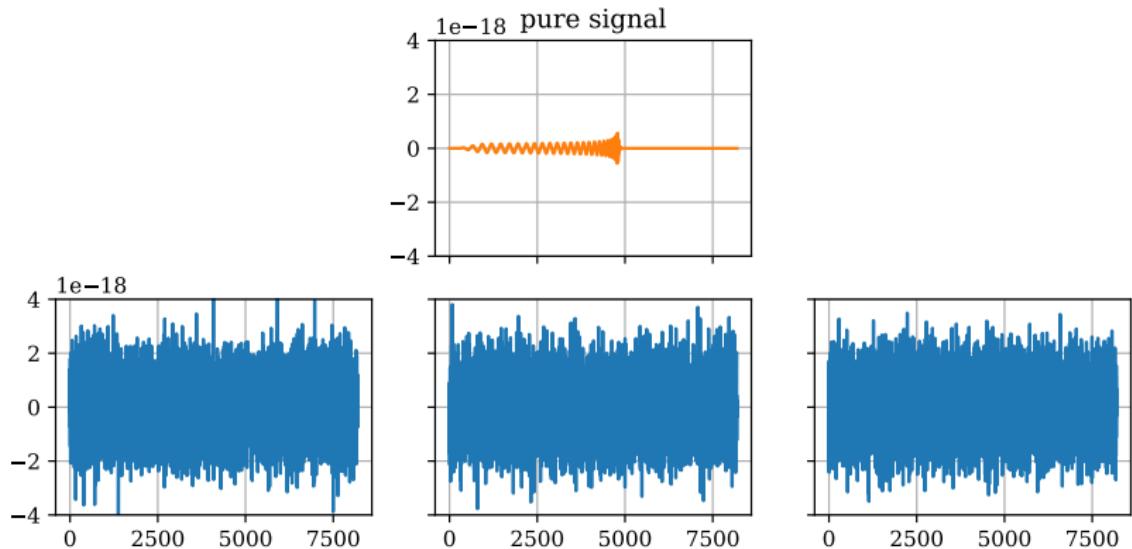
- ▶ Medium noise regime ($R = 0.65$ SNR $\approx 10?$)
→ got lucky here, something visual

Synthetic signal



- ▶ Medium noise regime ($R = 0.65$ SNR $\approx 10?$)
→ signal still ok in PSD

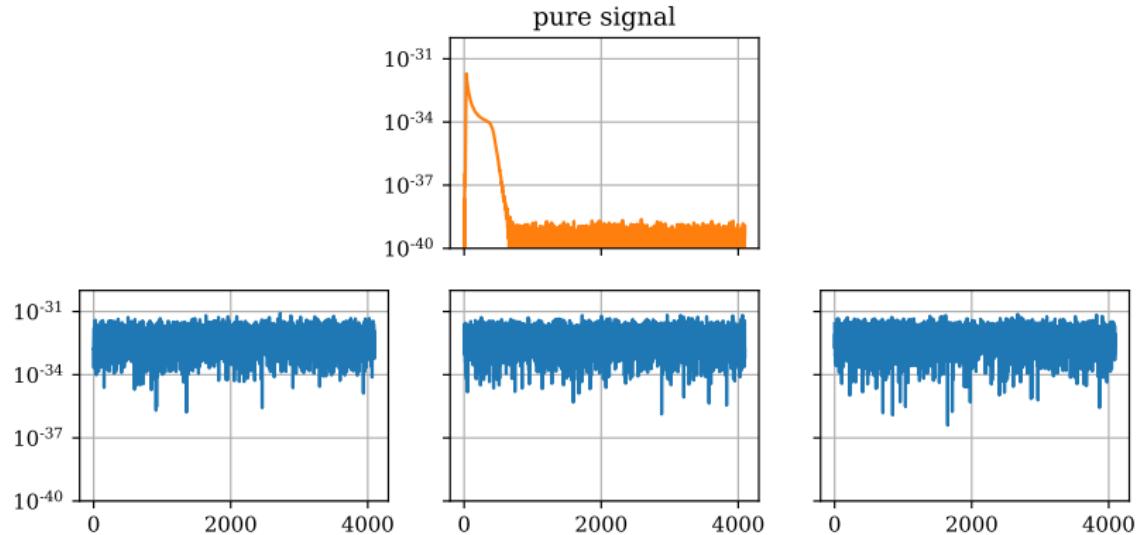
Synthetic signal



- ▶ Noisy regime ($R = 0.1$ SNR $\approx 3?$)

→ ...nope

Synthetic signal



- ▶ Noisy regime ($R = 0.1$ SNR $\approx 3?$)
→ ...still nope

Idea: shape of data

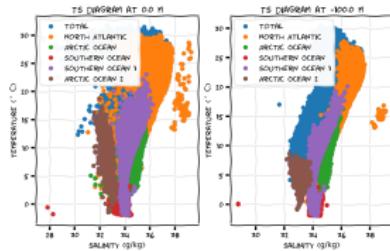


Figure: TS diagram from Argo data (with help from Fei Er Yan).

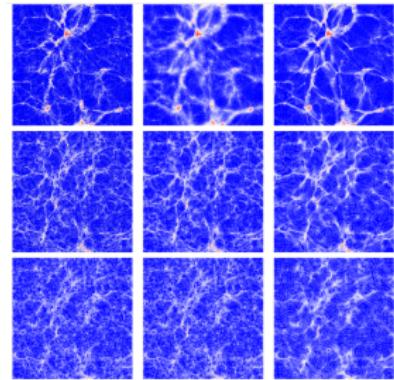


Figure: From Fig. 4 of Cheng, Chu & Tang (2015), JCAP.

- ▶ data might have a **shape**
 - noise might have one shape, structured data might have another
 - depends on **embedding** (a loop in 2d might be a curve in 3d)
 - **topology** \sim shape information!

Idea: shape of data

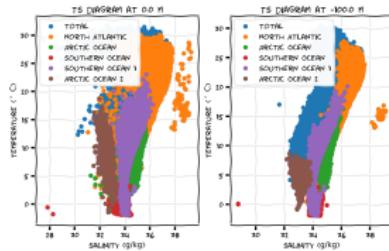


Figure: TS diagram from Argo data (with help from Fei Er Yan).

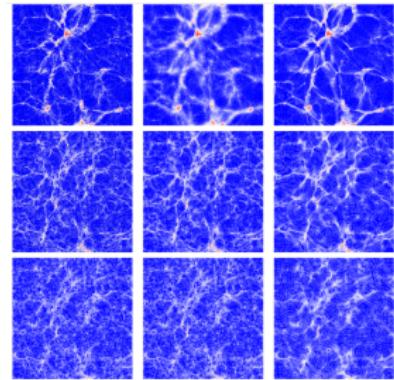


Figure: From Fig. 4 of Cheng, Chu & Tang (2015), JCAP.

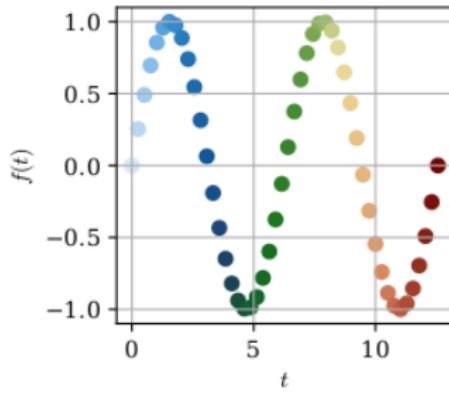
- ▶ data might have a **shape**
 - noise might have one shape, structured data might have another
 - depends on **embedding** (a loop in 2d might be a curve in 3d)
 - **topology** ~ shape information!

how to do this with time-series data?

Takens embedding: easy case

- ▶ semi-trivial example

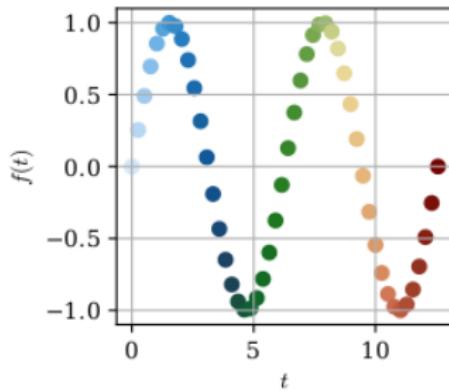
$$f(t) = \sin(t) = [f(0), f(\Delta t), f(2\Delta t), \dots] = [f_0, f_1, f_2, \dots]$$



Takens embedding: easy case

- ▶ semi-trivial example

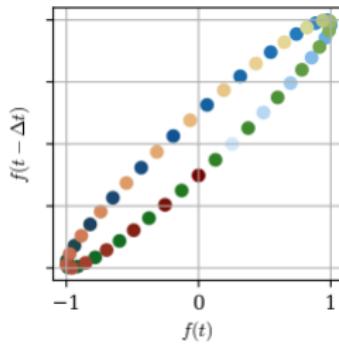
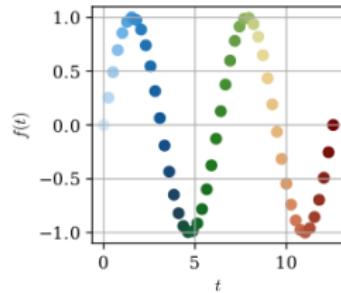
$$f(t) = \sin(t) = [f(0), f(\Delta t), f(2\Delta t), \dots] = [f_0, f_1, f_2, \dots]$$



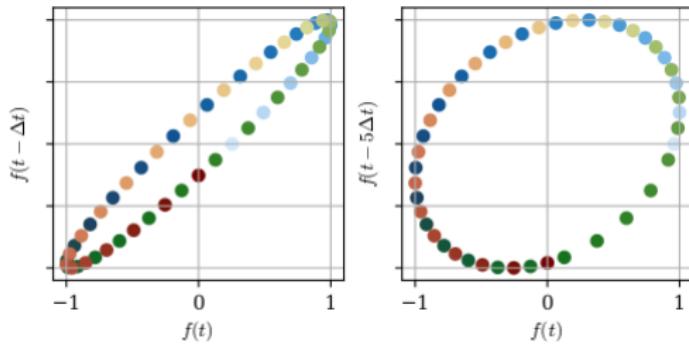
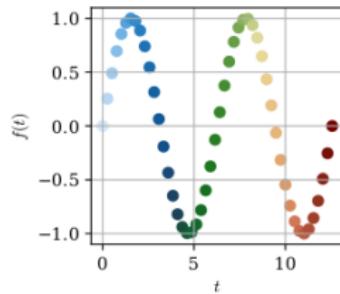
- ▶ why not

$$g(t) = \begin{pmatrix} f(t) \\ f(t - \Delta t) \end{pmatrix} = \begin{bmatrix} f_1, f_2, f_3, \dots \\ f_0, f_1, f_2, \dots \end{bmatrix}?$$

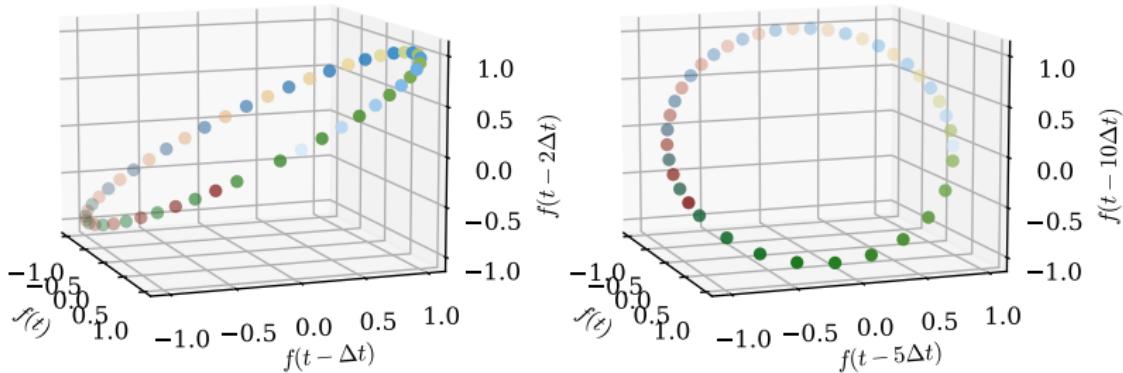
Takens embedding: easy case



Takens embedding: easy case

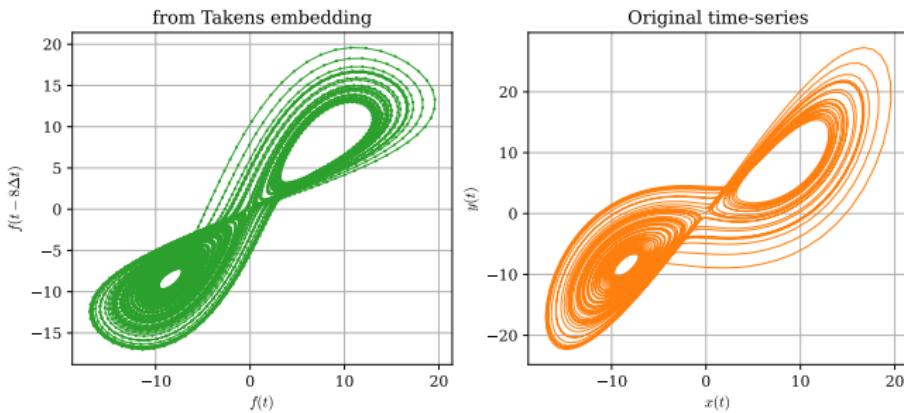
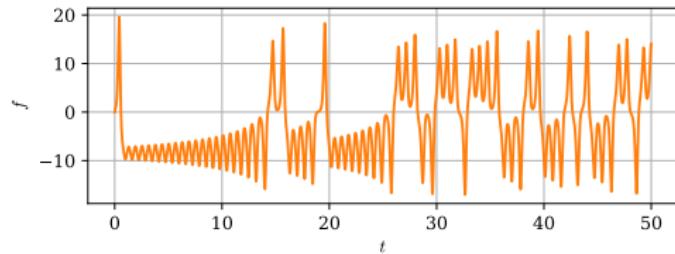


Takens embedding: easy case



- ▶ seems to be fundamentally a 2d structure
 - embedding in 3d still 'looks' 2d (more fundamental reason for this)
 - has a 2d 'hole' (**Betti number** $\beta_1 = 1$, H_1 homology later)

Takens embedding: more interesting case



Takens's theorem

Takens (1981) proved that (paraphrasing a bit here):

- ▶ the delay transformation for finite dimensional manifolds is an **embedding**
 - think ‘manifold’ as that traced out by solution trajectory
 - embedding = no information loss
 - **upper bound** in dimension of delay embedding
- ▶ manifold structures are persevered up to **diffeomorphisms**
 - diffeomorphism \Rightarrow **topological equivalence**



Figure: Floris Takens (1940-2010),
Dutch mathematician + flautist.

Takens's theorem

Takens (1981) proved that (paraphrasing a bit here):

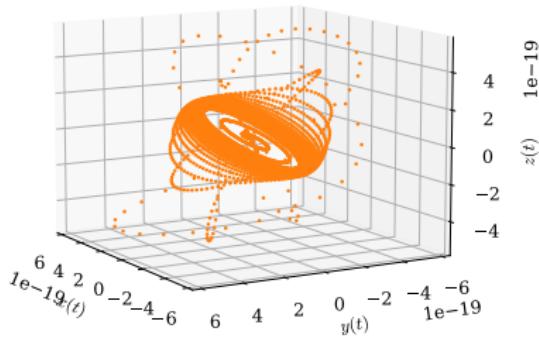
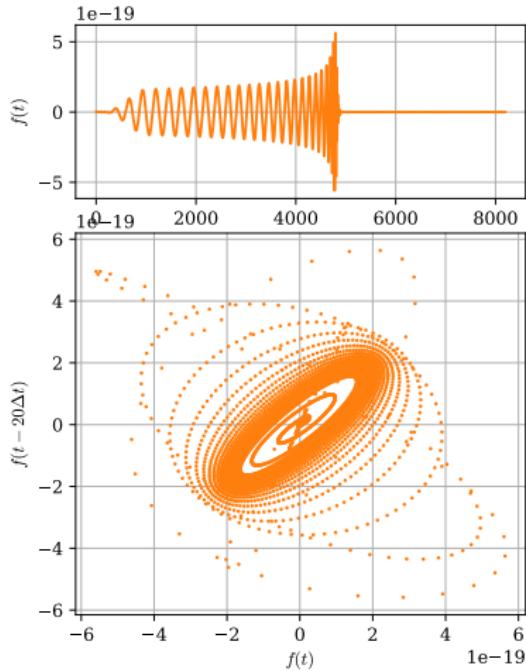
- ▶ the delay transformation for finite dimensional manifolds is an **embedding**
 - think ‘manifold’ as that traced out by solution trajectory
 - embedding = no information loss
 - **upper bound** in dimension of delay embedding
- ▶ manifold structures are persevered up to **diffeomorphisms**
 - diffeomorphism \Rightarrow **topological equivalence**



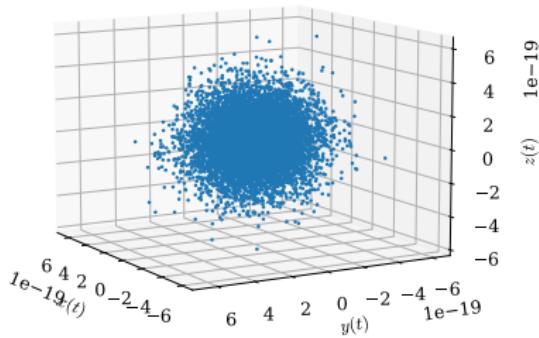
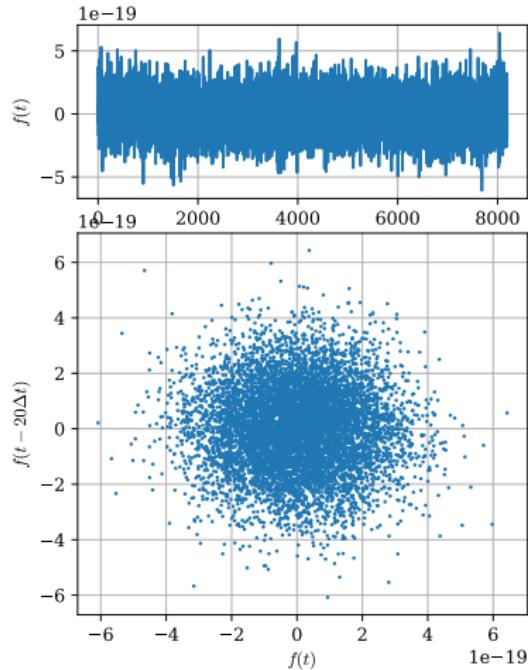
Figure: Floris Takens (1940-2010),
Dutch mathematician + flautist.

(above examples are solutions to 2d and 3d dynamical systems)

Takens embedding: gw case



Takens embedding: gw case



Some topology: equivalences

Great, but how to measure the shape of data?

Some topology: equivalences

Great, but how to measure the shape of data? Do something “coarser” (but more robust / global): how to measure **connectedness**?



- ▶ **topology**, concerned with a space and **continuous** maps on it
 - two spaces are X and Y are **equivalent** if there is some map between them with appropriate properties

Figure: Two homotopy equivalent manifolds.

Some topology: equivalences

Great, but how to measure the shape of data? Do something “coarser” (but more robust / global): how to measure **connectedness**?



- ▶ **topology**, concerned with a space and **continuous** maps on it
 - two spaces are X and Y are **equivalent** if there is some map between them with appropriate properties
- ▶ a **homotopy** between f_1 and f_2 on X is some continuous map that takes f_1 into f_2
 - a map can be continuously deformed into another map (e.g., a path)

Figure: Two homotopy equivalent manifolds.

Some topology: equivalences

Great, but how to measure the shape of data? Do something “coarser” (but more robust / global): how to measure **connectedness**?



- ▶ **topology**, concerned with a space and **continuous** maps on it
 - two spaces are X and Y are **equivalent** if there is some map between them with appropriate properties
- ▶ a **homotopy** between f_1 and f_2 on X is some continuous map that takes f_1 into f_2
 - a map can be continuously deformed into another map (e.g., a path)
- ▶ X and Y are **homotopy equivalent** if for $f : X \rightarrow Y$ and $g : Y \rightarrow X$, with $g \circ f$ and $f \circ g$ are homotopic to the identity maps

Figure: Two **homotopy equivalent** manifolds.

Some topology: equivalences

Great, but how to measure the shape of data? Do something “coarser” (but more robust / global): how to measure **connectedness**?



Figure: Two homotopy equivalent manifolds.

- ▶ **topology**, concerned with a space and **continuous** maps on it
 - two spaces are X and Y are **equivalent** if there is some map between them with appropriate properties
- ▶ a **homotopy** between f_1 and f_2 on X is some continuous map that takes f_1 into f_2
 - a map can be continuously deformed into another map (e.g., a path)
- ▶ X and Y are **homotopy equivalent** if for $f : X \rightarrow Y$ and $g : Y \rightarrow X$, with $g \circ f$ and $f \circ g$ are homotopic to the identity maps
 - cf. X and Y are **homeomorphic**, where $f \circ g$ and $g \circ f$ is the identity map

Some topology: homology and Betti numbers

- ▶ measure a topology by measuring its defects or **holes**
 - cf. classify a dynamical system by its singularities
 - implications if an inner product is also present (i.e., a **geometry**, e.g. **Gauss–Bonnet** linking curvature with Euler characteristic)

Some topology: homology and Betti numbers

- ▶ measure a topology by measuring its defects or **holes**
 - cf. classify a dynamical system by its singularities
 - implications if an inner product is also present (i.e., a **geometry**, e.g. **Gauss–Bonnet** linking curvature with Euler characteristic)
- ▶ homotopic equivalence \Rightarrow **invariants**, e.g. **Betti numbers** β_k :
 - β_0 = connected components
 - β_1 = 1d holes
 - β_2 = 2d holes etc.

Some topology: homology and Betti numbers

- ▶ measure a topology by measuring its defects or **holes**
 - cf. classify a dynamical system by its singularities
 - implications if an inner product is also present (i.e., a **geometry**, e.g. **Gauss–Bonnet** linking curvature with Euler characteristic)
- ▶ homotopic equivalence \Rightarrow **invariants**, e.g. **Betti numbers** β_k :
 - β_0 = connected components
 - β_1 = 1d holes
 - β_2 = 2d holes etc.

(more formally, Betti numbers are the ranks of the **homology group** H_k with

$$H_k = \text{cycle group/boundary group} = \text{Ker } \partial_k / \text{Im } \partial_{k+1},$$

where ∂_k are **boundary maps**, and quotient groups are meant here)

Some topology: homology and Betti numbers

1-manifolds


$$\begin{array}{ll} \beta = 1 & \beta_0 = 1 \\ \beta_1 = 0 & \beta_1 = 1 \end{array}$$

2-manifolds

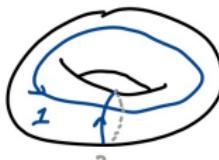
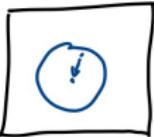

$$\begin{array}{ll} \beta_0 = 1 & \beta_0 = 1 \\ \beta_1 = 0 & \beta_1 = 1 \\ & \beta_1 = 0 \\ & \beta_0 = 1 \\ & \beta_1 = 2 \end{array}$$

Figure: Some example manifolds and corresponding β_0 and β_1 values.

Some topology: simplicial complexes

- ▶ above is still by inspection, more algorithmically?
→ build up a manifold with **simplices** (cf. a basis of sorts)

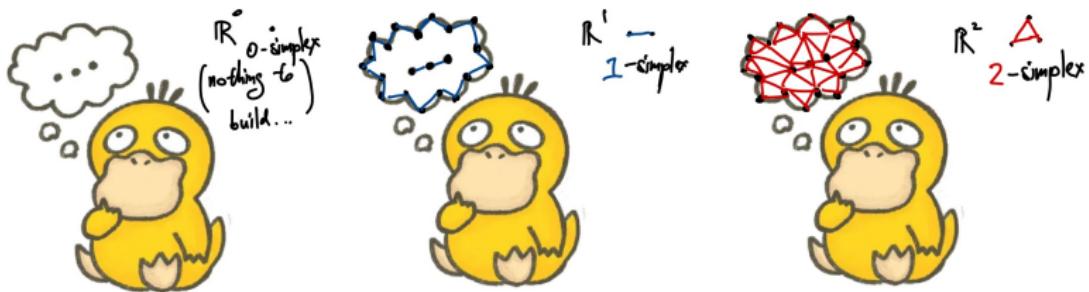


Figure: Thinking hard about topology in terms of simplices.

Some topology: simplicial complexes

- ▶ above is still by inspection, more algorithmically?
→ build up a manifold with **simplices** (cf. a basis of sorts)

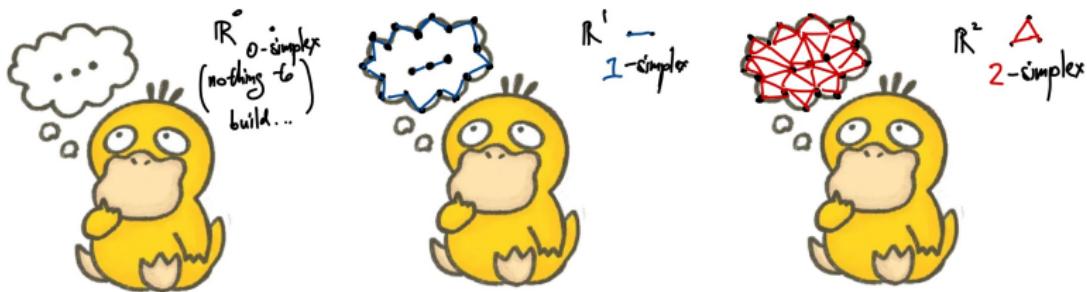


Figure: Thinking hard about topology in terms of simplices.

- ▶ consider the labelling the vertices $i, j, k \dots$ and encoding a simplex as **chains** (e.g., $\{i, j\}$ and $\{i, j, k\}$ would be a 1 and 2-simplex respectively)

Some topology: simplicial complexes

- ▶ above is still by inspection, more algorithmically?
→ build up a manifold with **simplices** (cf. a basis of sorts)

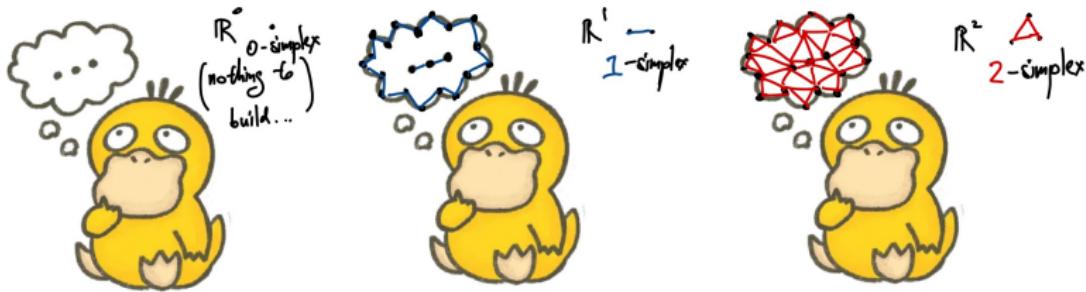


Figure: Thinking hard about topology in terms of simplices.

- ▶ consider the labelling the vertices $i, j, k \dots$ and encoding a simplex as **chains** (e.g., $\{i, j\}$ and $\{i, j, k\}$ would be a 1 and 2-simplex respectively)
→ β_0 number of connected components (disjoint chains)
→ β_1 the number of **closed** 1-chains (i.e. 1-cycles) that are NOT part of 2-simplices
→ β_2 the number of 2-cycles that are NOT part of 3-simplices etc.

Some topology: Čech complex

- ▶ so far we have a manifold split into simplices, so how to use this for data that are point clouds to probe the underlying manifold?

Some topology: Čech complex

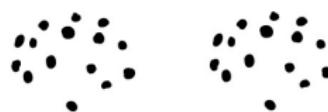
- ▶ so far we have a manifold split into simplicies, so how to use this for data that are point clouds to probe the underlying manifold?



Figure: Hermeowous Mora, Daedric prince of memory and knowledge, helping to demonstrate a [filtration](#) of a [Čech complex](#).

Some topology: Čech complex

- ▶ so far we have a manifold split into simplices, so how to use this for data that are point clouds to probe the underlying manifold?

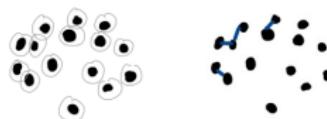


$$\begin{aligned}\beta_0 &= 15 \\ \beta_1 &= 0\end{aligned}$$

Figure: Hermeowous Mora, Daedric prince of memory and knowledge, helping to demonstrate a **filtration** of a **Čech complex**.

Some topology: Čech complex

- ▶ so far we have a manifold split into simplices, so how to use this for data that are point clouds to probe the underlying manifold?

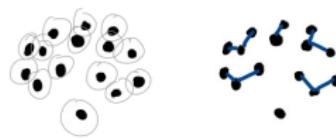


$$\begin{aligned}\beta_0 &= 11 \\ \beta_1 &= 0\end{aligned}$$

Figure: Hermeowous Mora, Daedric prince of memory and knowledge, helping to demonstrate a **filtration** of a **Čech complex**.

Some topology: Čech complex

- ▶ so far we have a manifold split into simplices, so how to use this for data that are point clouds to probe the underlying manifold?

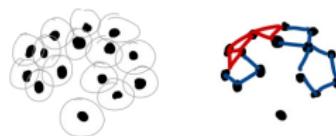


$$\begin{aligned}\beta_0 &= 6 \\ \beta_1 &= 0\end{aligned}$$

Figure: Hermeowous Mora, Daedric prince of memory and knowledge, helping to demonstrate a **filtration** of a **Čech complex**.

Some topology: Čech complex

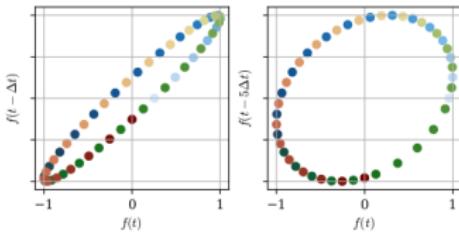
- ▶ so far we have a manifold split into simplices, so how to use this for data that are point clouds to probe the underlying manifold?



$$\frac{\beta_0}{\beta_1} = \frac{2}{3}$$

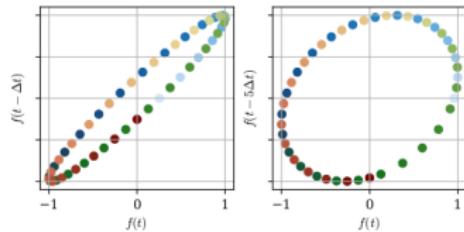
Figure: Hermeowous Mora, Daedric prince of memory and knowledge, helping to demonstrate a **filtration** of a **Čech complex**.

Some topology: persistent homology

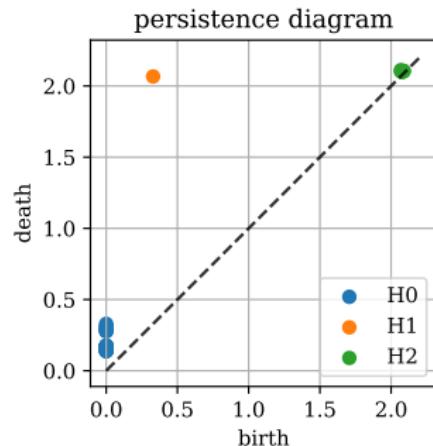
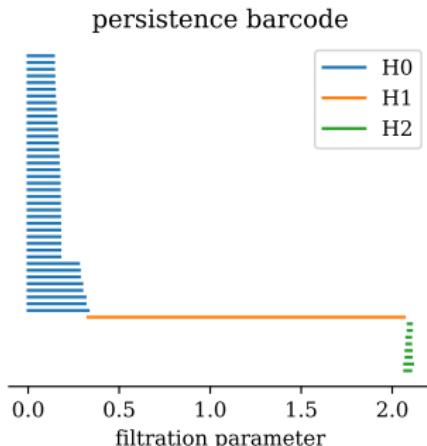


- ▶ for the periodic signal example, expect H_1 hole to be **persistent**

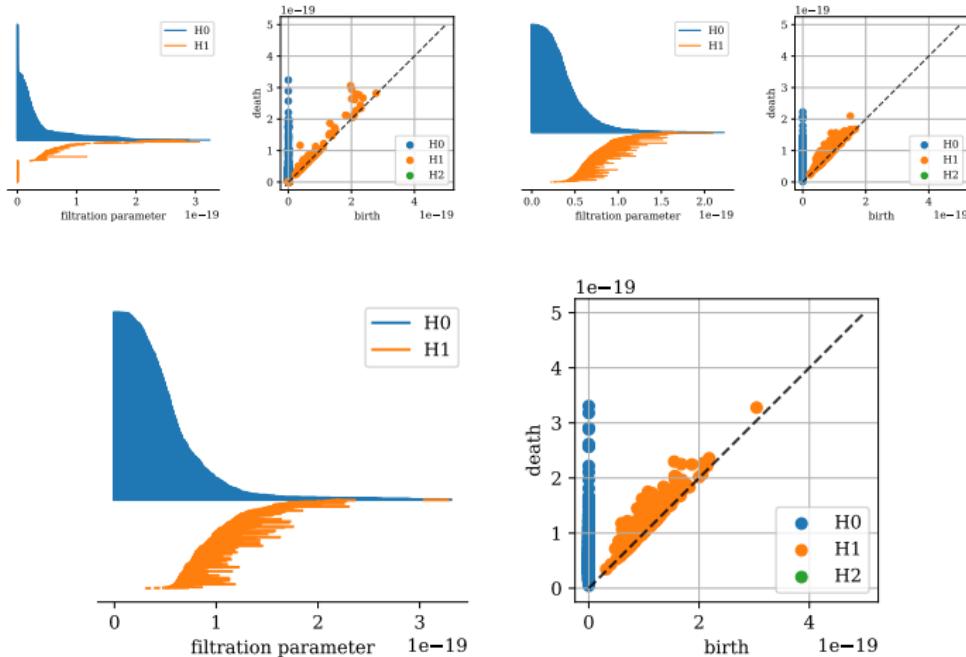
Some topology: persistent homology



- ▶ for the periodic signal example, expect H_1 hole to be **persistent**
→ codify in a **persistence barcode** and **diagram**



Back to the GW case ($R = 0.65$)

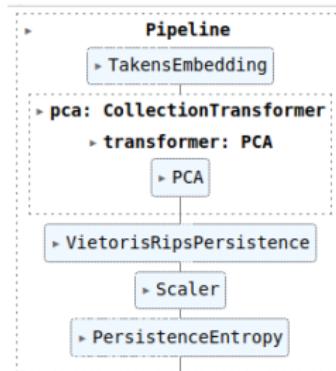


Machine Learning pipeline

- ▶ lower R (high noise) case feature less obvious visually...

Machine Learning pipeline

- ▶ lower R (high noise) case feature less obvious visually...
- ▶ ...but maybe who cares? Could throw that in a Machine Learning algorithm to train a classifier, e.g.,



Machine Learning pipeline

- ▶ simple demonstration with **logistic regression**
 - 1000 randomly generated samples, just topological features here

- ROC = Receiver operating characteristic, true positive rate against the false positive rate at each threshold setting
- AUC = Area under the ROC Curve, score for calculating, measure of skill for binary classifier (0.5 is crap, 0.9 onwards is pretty good, 1 is perfect)

In [49]: `print_scores(model)`

```
Accuracy on train: 0.719
ROC AUC on train: 0.776
Accuracy on valid: 0.69
ROC AUC on valid: 0.769
```

In [50]: `# 1 = 1 - 0, so if model predicts signal when there isn't one (false positive)
0 = 1 - 1 or 0 - 0, when it did it right
-1 = 0 - 1, so if model fails to predict a signal when it is there (false negative)`

```
dummy = model.predict(features) - labels
print(f" correct predictions {np.sum(dummy == 0)} / {len(dummy)}")
print(f" false positives {np.sum(dummy == 1)} / {len(dummy)}")
print(f" false negatives {np.sum(dummy == -1)} / {len(dummy)}")
```

```
correct predictions 716 / 1000
false positives 84 / 1000
false negatives 200 / 1000
```

Machine Learning pipeline

- ▶ simple demonstration with **logistic regression**
 - 1000 randomly generated samples, just topological features here

- ROC = Receiver operating characteristic, true positive rate against the false positive rate at each threshold setting
- AUC = Area under the ROC Curve, score for calculating, measure of skill for binary classifier (0.5 is crap, 0.9 onwards is pretty good, 1 is perfect)

In [49]: `print_scores(model)`

```
Accuracy on train: 0.719
ROC AUC on train: 0.776
Accuracy on valid: 0.69
ROC AUC on valid: 0.769
```

In [50]: `# 1 = 1 - 0, so if model predicts signal when there isn't one (false positive)
0 = 1 - 1 or 0 - 0, when it did it right
-1 = 0 - 1, so if model fails to predict a signal when it is there (false negative)`

```
dummy = model.predict(features) - labels
print(f" correct predictions {np.sum(dummy == 0)} / {len(dummy)}")
print(f" false positives {np.sum(dummy == 1)} / {len(dummy)}")
print(f" false negatives {np.sum(dummy == -1)} / {len(dummy)}")
```

```
correct predictions 716 / 1000
false positives 84 / 1000
false negatives 200 / 1000
```

- ▶ not complete garbage...? (certainly better than just doing it raw with the signal)
 - Brester & Jung do it with a CNN, larger ensemble of samples, include signal and topological features; works fairly well

Summary

- ▶ some concepts in TDA
 - convert signal to point cloud via **Takens embedding**
 - **filtrations** to generate a hierarchy of nested **simplicial complexes**
 - compute **persistent homology**, compute their ranks, and track their birth/death as a function of filtration
 - encode in **persistence barcodes/diagrams**
- ▶ topology fairly robust with noise
 - **stability** theorems of persistent features (not covered here)
 - add **topological features** into ML pipeline

Things not talked about + other ideas

- ▶ choice of complexes and/or persistence?
 - showed Čech, but Vietoris–Rips actually computed
 - other choices possible, depends on problem and nature of data

Things not talked about + other ideas

- ▶ choice of complexes and/or persistence?
 - showed Čech, but Vietoris–Rips actually computed
 - other choices possible, depends on problem and nature of data
- ▶ choice of features
 - barcodes (lines) and diagrams (dots) here
 - landscapes, Betti curves, images... (even better for ML features?)
 - talk about distances, statistics, (dis)similarities between features?

Things not talked about + other ideas

- ▶ choice of complexes and/or persistence?
 - showed Čech, but Vietoris–Rips actually computed
 - other choices possible, depends on problem and nature of data
- ▶ choice of features
 - barcodes (lines) and diagrams (dots) here
 - landscapes, Betti curves, images... (even better for ML features?)
 - talk about distances, statistics, (dis)similarities between features?
 - e.g. L^p distance of landscapes from US market indices and correlations to major financial market crashes (Gidea & Katz, 2018, Physica A)

Things not talked about + other ideas

- ▶ choice of complexes and/or persistence?
 - showed Čech, but Vietoris–Rips actually computed
 - other choices possible, depends on problem and nature of data
- ▶ choice of features
 - barcodes (lines) and diagrams (dots) here
 - landscapes, Betti curves, images... (even better for ML features?)
 - talk about distances, statistics, (dis)similarities between features?
 - e.g. L^p distance of landscapes from US market indices and correlations to major financial market crashes (Gidea & Katz, 2018, Physica A)
- ▶ computation of ranks of homology groups?
 - essentially matrix reduction (extra slides if wanted)

Things not talked about + other ideas

- ▶ choice of complexes and/or persistence?
 - showed Čech, but Vietoris–Rips actually computed
 - other choices possible, depends on problem and nature of data
- ▶ choice of features
 - barcodes (lines) and diagrams (dots) here
 - landscapes, Betti curves, images... (even better for ML features?)
 - talk about distances, statistics, (dis)similarities between features?
 - e.g. L^p distance of landscapes from US market indices and correlations to major financial market crashes (Gidea & Katz, 2018, Physica A)
- ▶ computation of ranks of homology groups?
 - essentially matrix reduction (extra slides if wanted)
- ▶ links with category theory?
 - persistent homology as a functor

Credit + questions

Description here to aid in developing intuition, likely have holes in rigour. See following for more + extensions:

- ▶ online lectures/tutorials by Henry Adams, Bastian Grossenbacher Rieck, Robert Ghrist + Vidit Nanda
- ▶ “A user’s guide to Topological Data Analysis” by Elizabeth Munch (short article)
- ▶ “Algebraic Topology for Data Scientists” by Michael Polstol (on arXiv)
- ▶ giotto-tda software package ([giotto-ai.github.io](https://github.com/giotto-ai/giotto-tda))



Figure: Questions and/or headaches?

Extras: definitions

(Assuming some background in algebra)

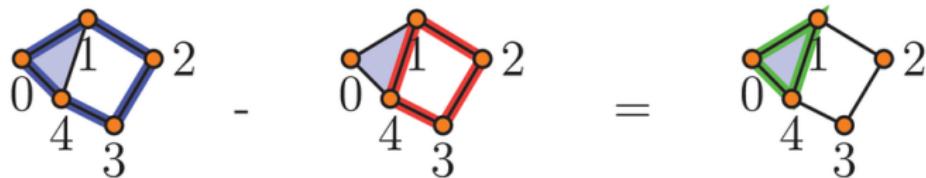


Figure: Example of homologous cycles. From Topaz, Ziegelmeier & Halverson (2014).

- ▶ σ a **k -simplex** with vertices $\{v_i\}_{i=0}^k$ with an ordering
 - assuming **equivalence** if ordering differ by an even permutation (i.e. element of S_k)
 - denote it $[v_0, v_1, \dots, v_k]$ (e.g., $[0, 1, 4]$ above is a 2-simplex)
 - a collection of these simplices would be a **simplicial complex** K

Extras: definitions

(Assuming some background in algebra)

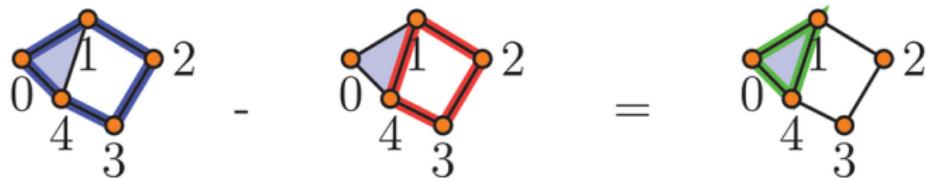


Figure: Example of homologous cycles. From Topaz, Ziegelmeier & Halverson (2014).

- ▶ σ a **k -simplex** with vertices $\{v_i\}_{i=0}^k$ with an ordering
 - assuming **equivalence** if ordering differ by an even permutation (i.e. element of S_k)
 - denote it $[v_0, v_1, \dots, v_k]$ (e.g., $[0, 1, 4]$ above is a 2-simplex)
 - a collection of these simplices would be a **simplicial complex** K
- ▶ a **k -chain** is some formal sum

$$\sum_i n_i \sigma_i, \quad n_i \in \mathbb{Z}, \quad \sigma_i \in K$$

(e.g. $2[0, 1] - 3[1, 2]$ above is a possible 1-chain)

Extras: definitions

- ▶ can add chains together and still get a chain, i.e. **closure** under a **binary operation**
 - can show associativity, existence of inverse and identity element, so forms the **chain group** $C_k(K)$
 - if $\dim K$ is the highest dimension of a simplex in K , then $C_k(K) = 0$ for $k > \dim K$ or $k < 0$

Extras: definitions

- ▶ can add chains together and still get a chain, i.e. **closure** under a **binary operation**
 - can show associativity, existence of inverse and identity element, so forms the **chain group** $C_k(K)$
 - if $\dim K$ is the highest dimension of a simplex in K , then $C_k(K) = 0$ for $k > \dim K$ or $k < 0$
- ▶ a **k -cycle** is a closed k -chain
 - cycles form a subgroup of the chain group

Extras: definitions

- ▶ can add chains together and still get a chain, i.e. **closure** under a **binary operation**
 - can show associativity, existence of inverse and identity element, so forms the **chain group** $C_k(K)$
 - if $\dim K$ is the highest dimension of a simplex in K , then $C_k(K) = 0$ for $k > \dim K$ or $k < 0$
- ▶ a **k -cycle** is a closed k -chain
 - cycles form a subgroup of the chain group
- ▶ a homomorphism known as the **boundary operator** ∂_k is such that

$$\partial_k : C_k(K) \rightarrow C_{k-1}(K), \quad [v_0, \dots, v_k] \mapsto \sum_{i=0}^k (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_k],$$

i.e. an alternating sum ignoring each vertex \hat{v}_i

(e.g. for a 2-simplex $[1, 2, 3]$, $\partial_2[1, 2, 3] = [2, 3] - [1, 3] + [1, 2]$, which is a 1-cycle and boundary of the original 2-simplex, i.e. ∂_k guts the innards)

→ boundary cycles also form a subgroup of the chain group

Extras: definitions

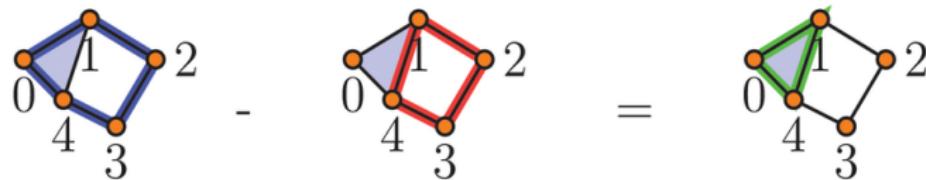


Figure: Example of homologous cycles. From Topaz, Ziegelmeier & Halverson (2014).

- ▶ want to count “holes”
 - “holes” being things that are not filled in

Extras: definitions

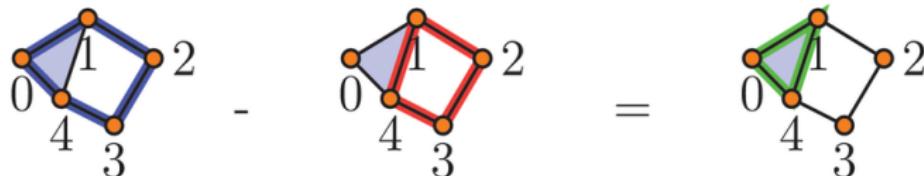


Figure: Example of homologous cycles. From Topaz, Ziegelmeier & Halverson (2014).

- ▶ want to count “holes”
 - “holes” being things that are not filled in
- ▶ argue here that the two cycles here encompass the same “hole”
 - cycles can be deformed into another via crossing the filled in 2-simplex (cf. homotopy)
 - two cycles here are **homologous**, count them as one under an equivalence relation

Extras: definitions

- ▶ want to count “holes”

→ want the number of k -cycles that are NOT boundaries of $(k + 1)$ -simplices, i.e. throw away homologous cycles, i.e.,

$$H_k = \frac{k\text{-cycles}}{k\text{-cycles that are boundaries of } (k+1)\text{-simplices}}$$

under the homology equivalence relation

- ▶ with the boundary map ∂_k :

→ k -cycles if no boundaries, i.e. $\text{Ker } \partial_k$

→ boundaries of $(k + 1)$ -simplices are all the results of ∂_{k+1} , i.e., $\text{Im } \partial_{k+1}$

$$\Rightarrow H_k = \text{Ker } \partial_k / \text{Im } \partial_{k+1} ,$$

and $\beta_k = \text{rank}(H_k)$

Extras: computation

- ▶ suppose we have a triangle with $K = \{[1], [2], [3], [1, 2], [1, 3], [2, 3]\}$
→ note this triangle is not filled in, since we don't have $[1, 2, 3]$

1) For H_0 ,

- ▶ since $\partial_0 : C_0(K) \rightarrow C_{-1}(K) = \{0\}$, $\text{Ker } \partial_0$ is full rank and spanned by $\{[1], [2], [3]\}$
- ▶ $\text{Im } \partial_1$ is spanned by $\{[2] - [1], [3] - [1], [3] - [2]\}$, but $[3] - [2] = ([2] - [1]) + ([3] - [1])$, so rank of $\text{Im } \partial_1 = 2$
- ▶ so $\beta_0 = 3 - 2 = 1$, as expected

Extras: computation

- ▶ suppose we have a triangle with $K = \{[1], [2], [3], [1, 2], [1, 3], [2, 3]\}$
→ note this triangle is not filled in, since we don't have $[1, 2, 3]$

1) For H_0 ,

- ▶ since $\partial_0 : C_0(K) \rightarrow C_{-1}(K) = \{0\}$, $\text{Ker } \partial_0$ is full rank and spanned by $\{[1], [2], [3]\}$
- ▶ $\text{Im } \partial_1$ is spanned by $\{[2] - [1], [3] - [1], [3] - [2]\}$, but $[3] - [2] = ([2] - [1]) + ([3] - [1])$, so rank of $\text{Im } \partial_1 = 2$
- ▶ so $\beta_0 = 3 - 2 = 1$, as expected

2) For H_1 ,

- ▶ note that for $\text{Ker } \partial_1$ we want

$$\begin{aligned} 0 &= n_1([2] - [1]) + n_2([3] - [1]) + n_3([3] - [2]) \\ &= -(n_1 + n_2)[1] + (n_1 - n_3)[2] + (n_2 + n_3)[3], \end{aligned}$$

so if $n_3 = n_1$, we have $n_2 = -n_1$, so one degree of freedom, thus rank 1

- ▶ $C_2(K) = 0$ so $\text{Im } \partial_2$ is trivial
- ▶ so $\beta_1 = 1 - 0 = 1$, as expected

Extras: computation

- ▶ suppose triangle is filled in so $K = K + \{[1, 2, 3]\}$, then
- 1) For H_0 ,
 - ▶ as before, $\beta_0 = 1$, as expected
- 2) For H_1 ,
 - ▶ $\text{Ker } \partial_1$ is the same, but $\text{Im } \partial_2$ is now spanned by $\{[2, 3] - [1, 3] + [1, 2]\}$, so has rank 1 (we have a 1-cycle)
 - ▶ $\beta_1 = 1 - 1 = 0$, as expected, since there is no hole
- 3) For H_2 ,
 - ▶ kernel is trivial and rank 0 since only a zero coefficient will make above span to be zero (or because there are no 2-cycles)
 - ▶ image is trivial because there are no 3-simplices
 - ▶ $\beta_2 = 0 - 0 = 0$, also expected

Extras: computation

Not entirely elegant, but since we are talking about span, maybe we can leverage linear algebra to do some stuff

- ▶ there is a **vector space** structure, so the following is in fact justified
- ▶ for K the non-filled triangle, note that we have the **boundary matrices**

$$\partial_0(K) = \begin{pmatrix} [1] & [2] & [3] \\ 0 & 0 & 0 \end{pmatrix}, \quad \partial_1(K) = \begin{pmatrix} [1, 2] & [1, 3] & [2, 3] \\ -1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \begin{matrix} [1] \\ [2] \\ [3] \end{matrix}$$

$$\partial_2(K) = \begin{pmatrix} [1, 2, 3] \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{matrix} [1, 2] \\ [1, 3] \\ [2, 3] \end{matrix}$$

(e.g., $\partial_1([1, 2] + [2, 3] - [1, 3]) = \partial_1(1, -1, 1)^T = 0$, as it should be since it is a 1-cycle)

Extras: computation

- reduce down to a Smith normal form (cf. Gaussian elimination), e.g. for ∂_1 above,

$$\begin{pmatrix} -1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & -1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- can then just read stuff off
 - rank Ker ∂_k = number of zero columns of ∂_k
 - rank Im ∂_k = number of NON-zero columns of ∂_k

Extras: computation

- ▶ reduce down to a Smith normal form (cf. Gaussian elimination), e.g. for ∂_1 above,

$$\begin{pmatrix} -1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & -1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- ▶ can then just read stuff off
 - rank $\text{Ker } \partial_k$ = number of zero columns of ∂_k
 - rank $\text{Im } \partial_k$ = number of NON-zero columns of ∂_k
 - $\beta_0 = 3$ [zero columns of δ_0] - 2 [non-zero columns of δ_1] = 1
 - $\beta_1 = 1$ [zero column of δ_1] - 0 [non-zero columns of δ_2] = 1

Extras: computation

- ▶ reduce down to a Smith normal form (cf. Gaussian elimination), e.g. for ∂_1 above,

$$\begin{pmatrix} -1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & -1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

- ▶ can then just read stuff off
 - rank $\text{Ker } \partial_k$ = number of zero columns of ∂_k
 - rank $\text{Im } \partial_k$ = number of NON-zero columns of ∂_k
 - $\beta_0 = 3$ [zero columns of δ_0] - 2 [non-zero columns of δ_1] = 1
 - $\beta_1 = 1$ [zero column of δ_1] - 0 [non-zero columns of δ_2] = 1

- ▶ if K has filled in triangle, then $\partial_2 = (1, 0, 0)^T$ in the Smith normal form, so $\beta_1 = 1 - 1 = 0$, and $\beta_2 = 0 - 0 = 0$ as before

Extras: persistent homology

For a **filtration** of a complex K^i (indexed by i),

- ▶ **p -persistent k^{th} -homology group** is

$$H_k^{i,p} = \frac{k\text{-cycles}}{k\text{-boundaries} \cap k\text{-cycles}} = \frac{Z_k^i}{B_k^{i+p} \cap Z_k^i}$$

- ▶ the **persistent Betti numbers** $\beta_k^{i,p}$ associated with $H_k^{i,p}$ can be defined accordingly
 - the rank of the **free** part of $H_k^{i,p}$
- ▶ persistent homology is **stable** in a precise sense
 - persistence diagrams have bounded difference in the **bottleneck distance** to perturbations in input filtration (Cohen-Steiner, Edelsbrunner & Harer, 2006)

Extras: categories and functors

- ▶ a **category** $C = (\text{object}, \text{morphism})$
 - morphisms are maps between the objects in a category, e.g.
 - groups and homomorphisms
 - topological spaces and continuous maps
 - sets and functions
 - vector spaces and linear transformations

Extras: categories and functors

- ▶ a **category** $C = (\text{object}, \text{morphism})$
 - morphisms are maps between the objects in a category, e.g.
 - groups and homomorphisms
 - topological spaces and continuous maps
 - sets and functions
 - vector spaces and linear transformations
- ▶ **functors** map categories to each other
 - homology mapping category of topological spaces to category of **graded** groups
- ▶ idea (?)
 - results from category-theoretic studies apply generally
 - results in other fields can have analogs in other fields

(e.g. used a lot in computer science, applications in **categorical quantum mechanics** and quantum information)

