



AIDOS LAB
AI FOR DATA-ORIENTED SCIENCE



Lecture 7: (More) persistent homology

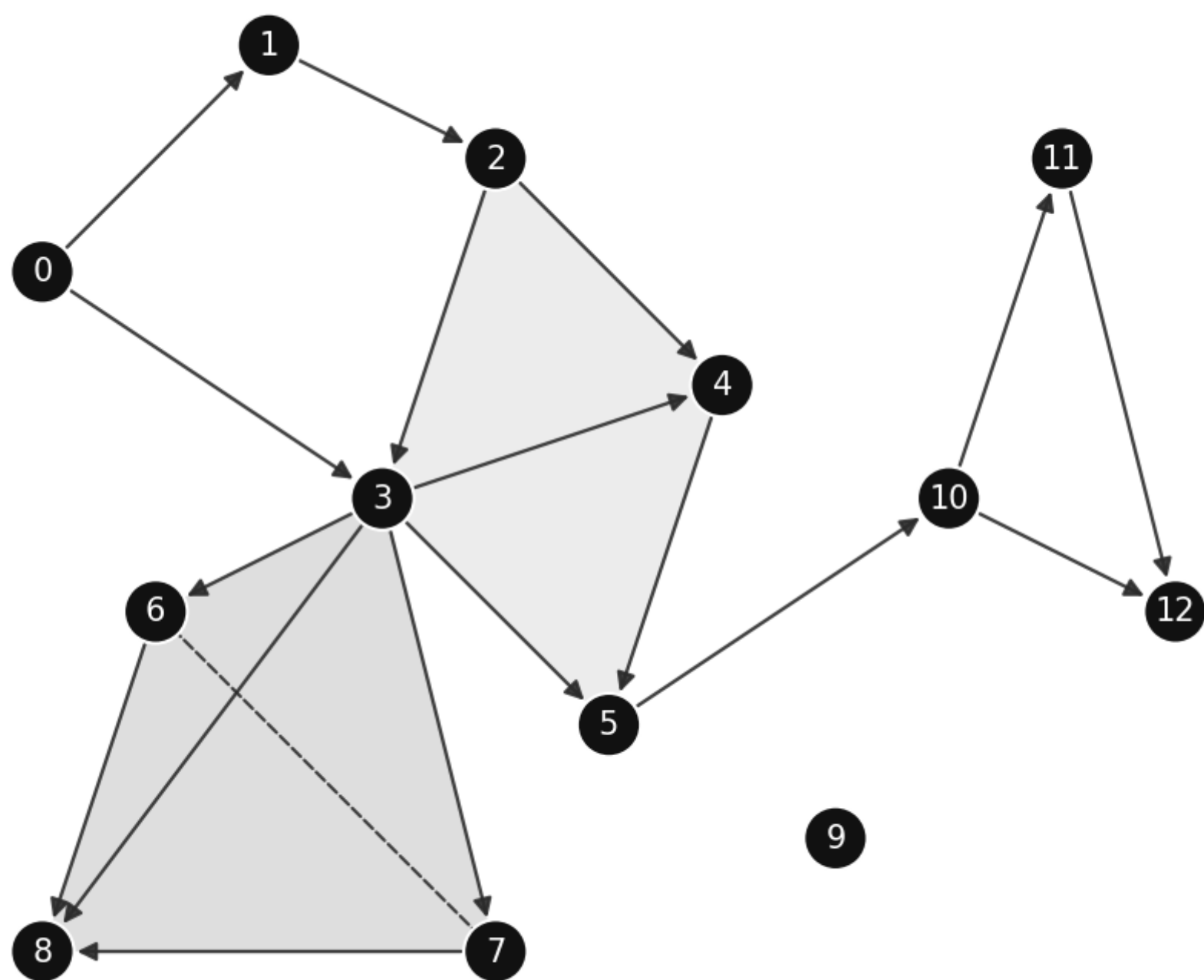
Geometry and Topology in Machine Learning

Recap

Lecture 7 — (More) persistent homology

We now know how to count holes

Chains and boundaries



- **Chain space:** let $k \geq 0$, the k -chain space $C_k(X)$ is the *vector space* spanned by the *oriented simplices*

$$C_k(X) = \text{span}\{(i_0, \dots, i_k) : i_0 < \dots < i_k\} \cong \mathbb{R}^{|X_k|}$$

$$c = \sum_{\sigma \in X_k} a_\sigma \cdot \sigma, \quad a_\sigma \in \mathbb{R}$$

- **Boundary operator:** *linear map* connecting chain spaces of consecutive dimension $\partial_k : C_k(X) \rightarrow C_{k-1}(X)$ defined on the basis elements as

$$\partial_k(i_0, \dots, i_k) = \sum_{j=0}^k (-1)^j (i_0, \dots, \widehat{i_j}, \dots, i_k)$$

Example:

$$\partial_3(3,6,7,8) =$$

We now know how to count holes

Betti numbers

Def. Homology group

$$H_k(X) = \frac{\ker \partial_k}{\operatorname{im} \partial_{k+1}}$$

Cycles

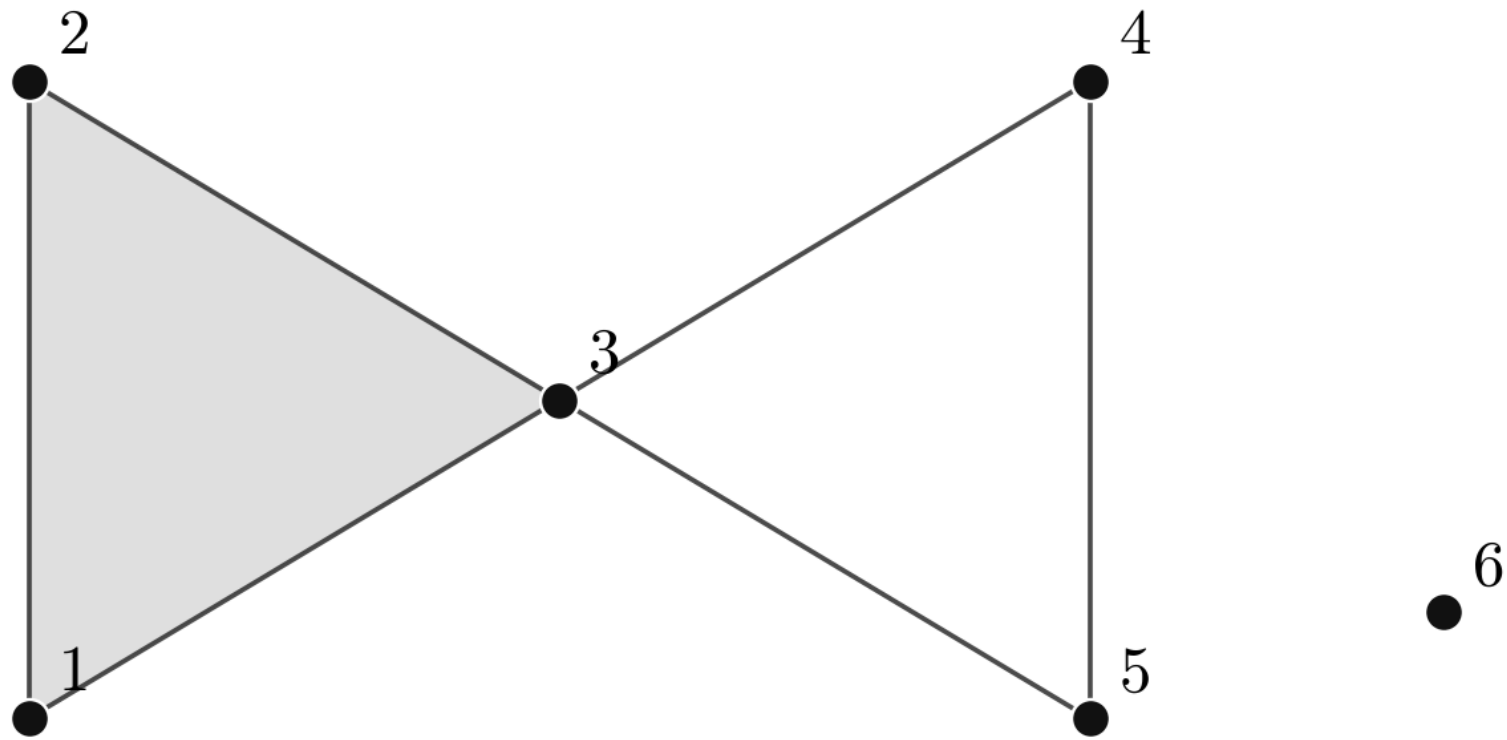
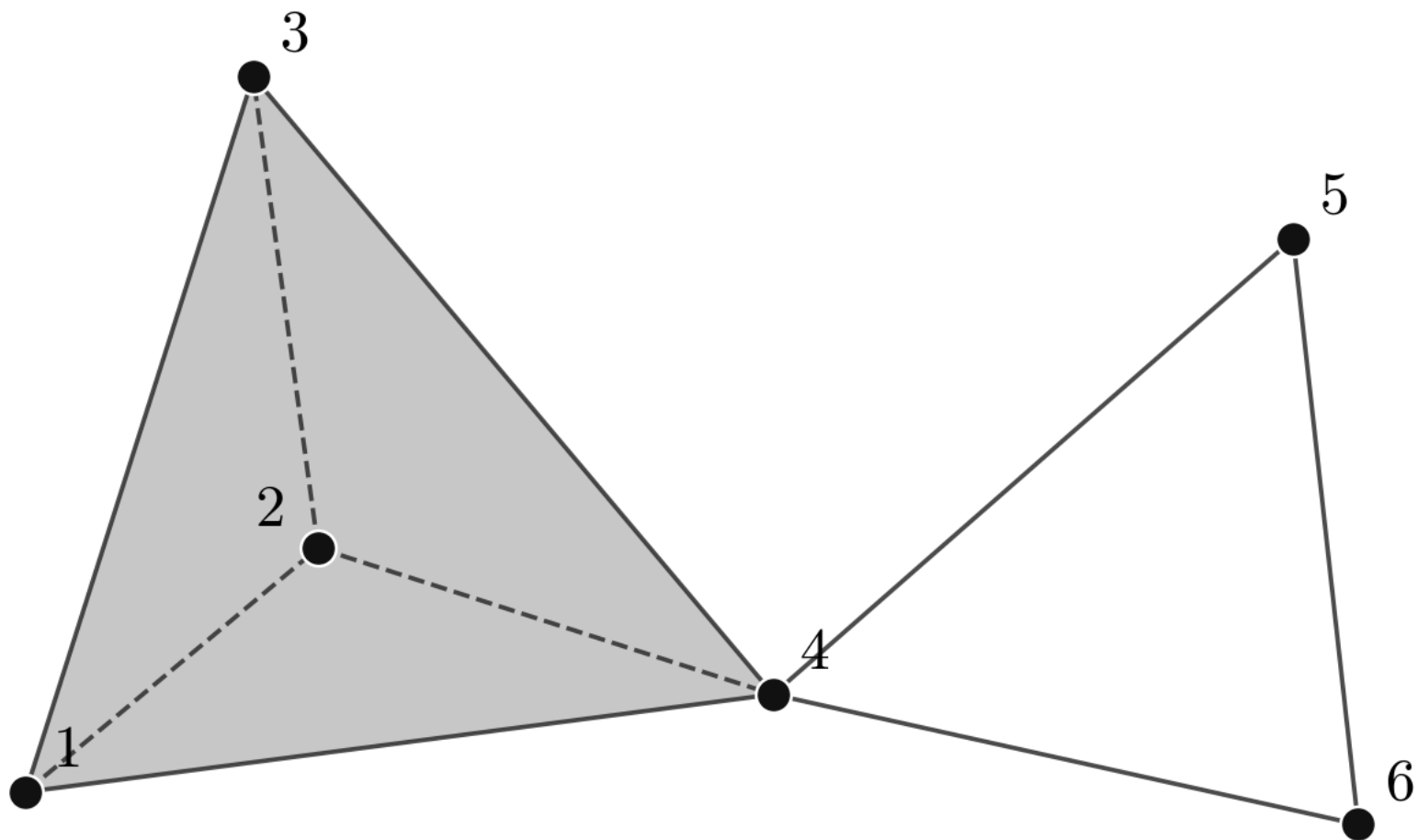
Boundaries

Def. Betti number

$$\beta_k(X) = \dim H_k(X)$$

"Number of k -dimensional holes in X "

Try it: What are the Betti numbers?



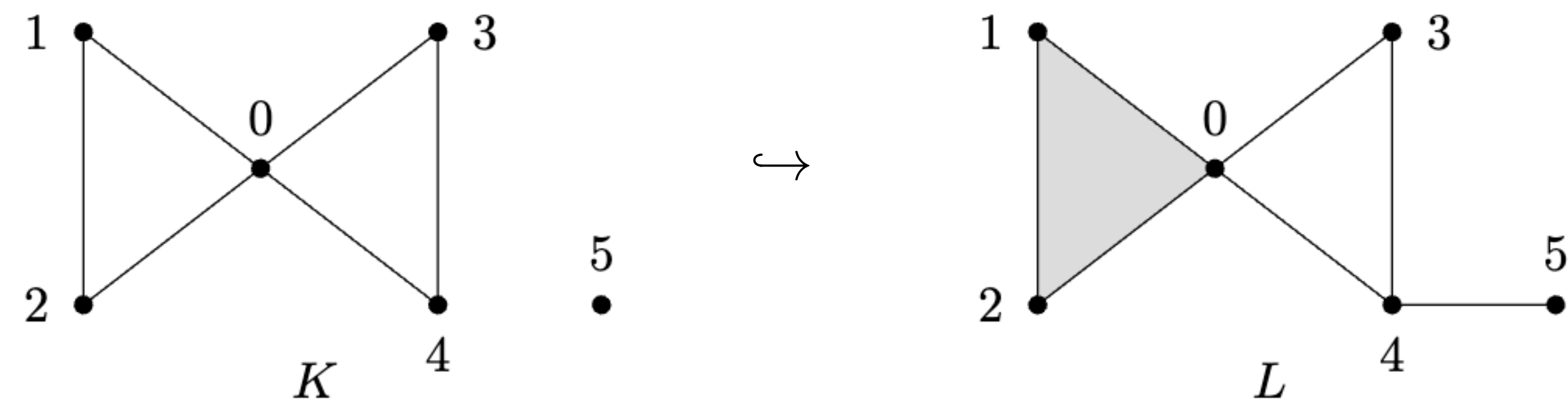
Induced maps

Def. Induced map at the chain level

A simplicial map $f : X \rightarrow Y$ induces a linear map $f_{\#} : C_k(X) \rightarrow C_k(Y)$ for all $k \geq 0$:

$$\alpha = \sum_{\sigma \in X_k} a_{\sigma} \cdot \sigma \mapsto f_{\#}(\alpha) = \sum_{\sigma \in X_k : f(\sigma) \in Y_k} a_{\sigma} \cdot f(\sigma)$$

Try it: What are the matrices of the induced maps I_0, I_1, I_2 ?



Induced maps

Def. Induced map at the homology level

A simplicial map $f : X \rightarrow Y$ induces a linear map $f_* : H_k(X) \rightarrow H_k(Y)$ for all $k \geq 0$:

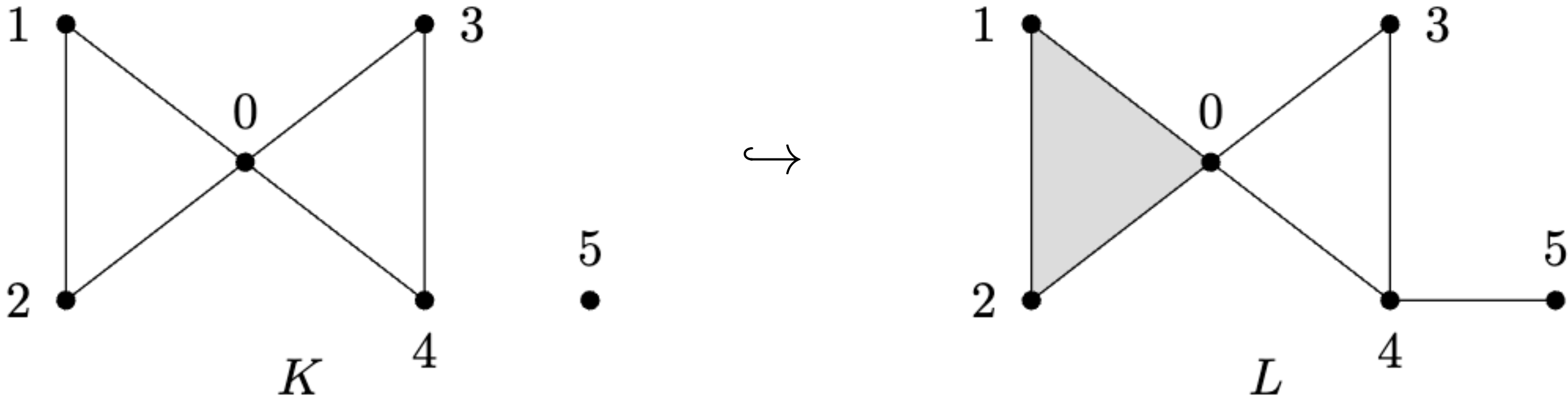
$$[\alpha] = \{\alpha + \partial_{k+1}\beta : \beta \in C_{k+1}(X)\} \mapsto f_*([\alpha]) = [f_{\#}(\alpha)] = \{f_{\#}(\alpha) + \partial_k^Y \beta : \beta \in C_{k+1}(Y)\}$$

Try it: What are the matrices of the induced maps in homology with bases

	K	L
H_0	$\{[0]_K, [5]_K\}$	$\{[0]_L\}$
H_1	$\{[a]_K, [b]_K\}$	$\{[b]_L\}$.

$$a = (0, 1) - (0, 2) + (1, 2)$$

$$b = (0, 3) - (0, 4) + (3, 4)$$



Induced maps

Two important properties

Properties of the induced maps

The induced homology map $f_* : H_k(X) \rightarrow H_k(Y)$ for all $k \geq 0$ satisfies the following two properties:

- If $\text{id} : X \rightarrow X$ is the identity map, sending each vertex in X to itself, then $\text{id}_* : H_k(X) \rightarrow H_k(X)$ is the identity map of vector spaces, which leaves every vector unchanged
- If $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ then

$$(g \circ f)_* = g_* \circ f_*$$

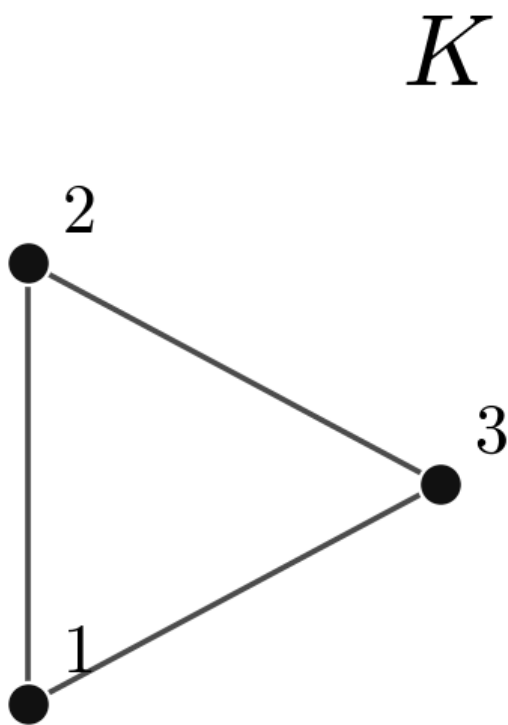
Induced maps

Two important properties

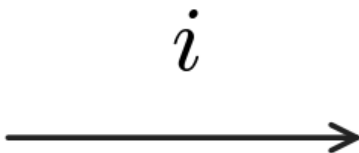
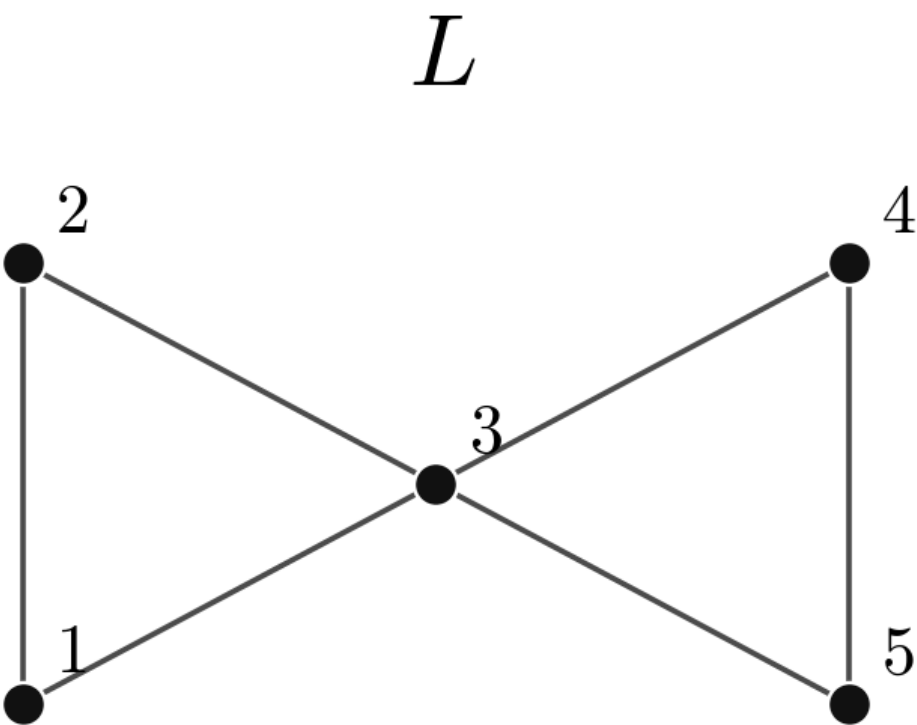
Properties of the induced maps

Induces maps in homology satisfy:

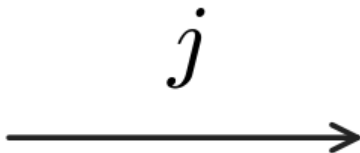
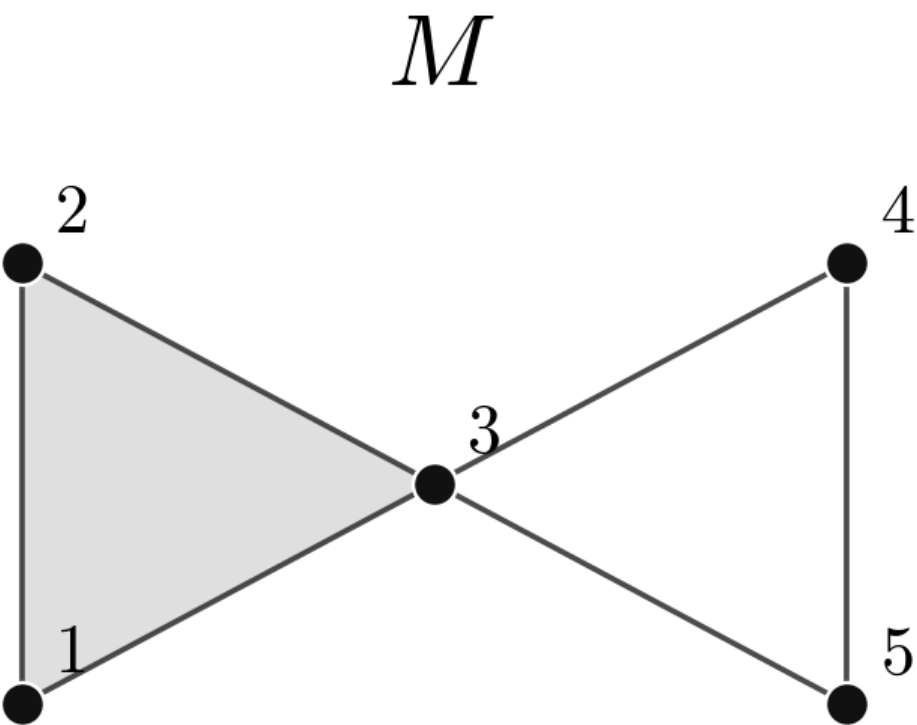
- $\text{id}_* = \text{id}_{H_k(X)}$
- $(g \circ f)_* = g_* \circ f_*$



$H_1(K) = \langle [a] \rangle$



$H_1(L) = \langle [a], [b] \rangle$

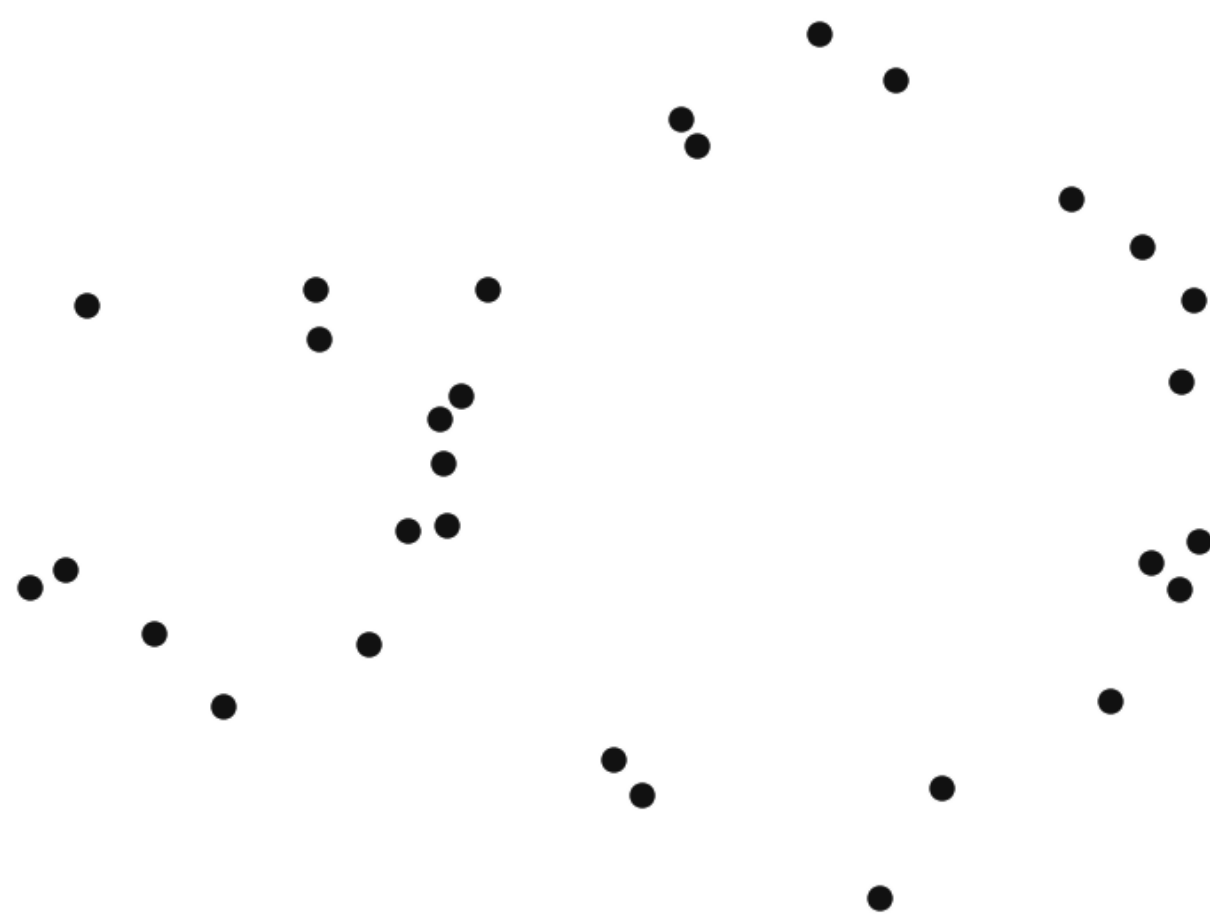


$H_1(M) = \langle [b] \rangle$

Motivation

What we want to solve today

Suppose you are given the following point cloud, how can you design a test to conclude that this sample comes from a distribution over two circles?



The nerve theorem

Lecture 7 — (More) persistent homology

Recap: Čech complex

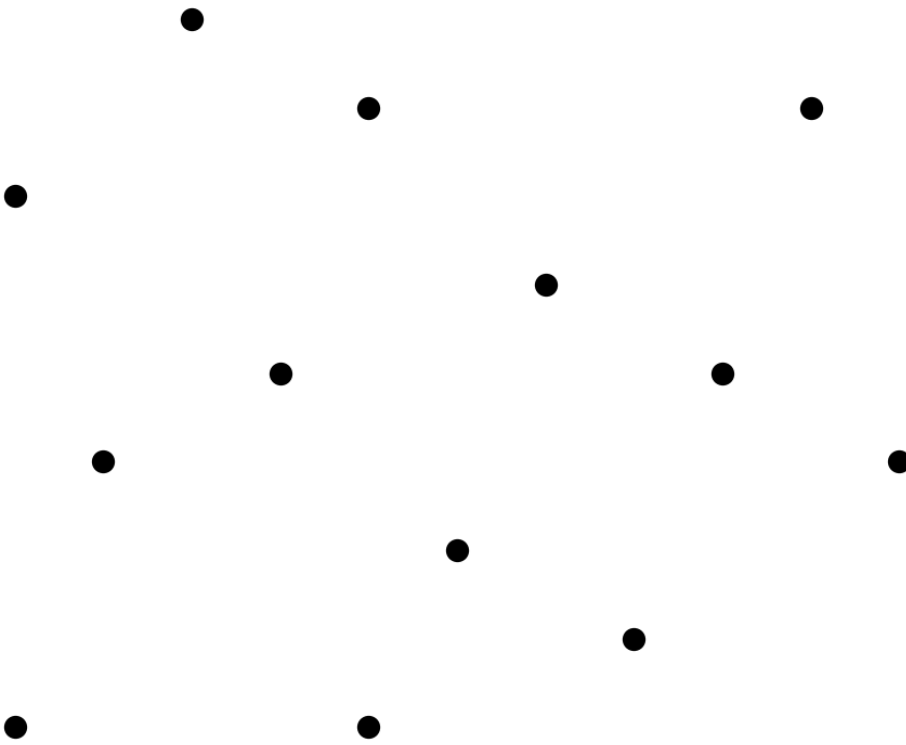
Finite metric space

Given (S, d) and $r \geq 0$, the **Čech complex** (or Nerve complex) at scale r is:

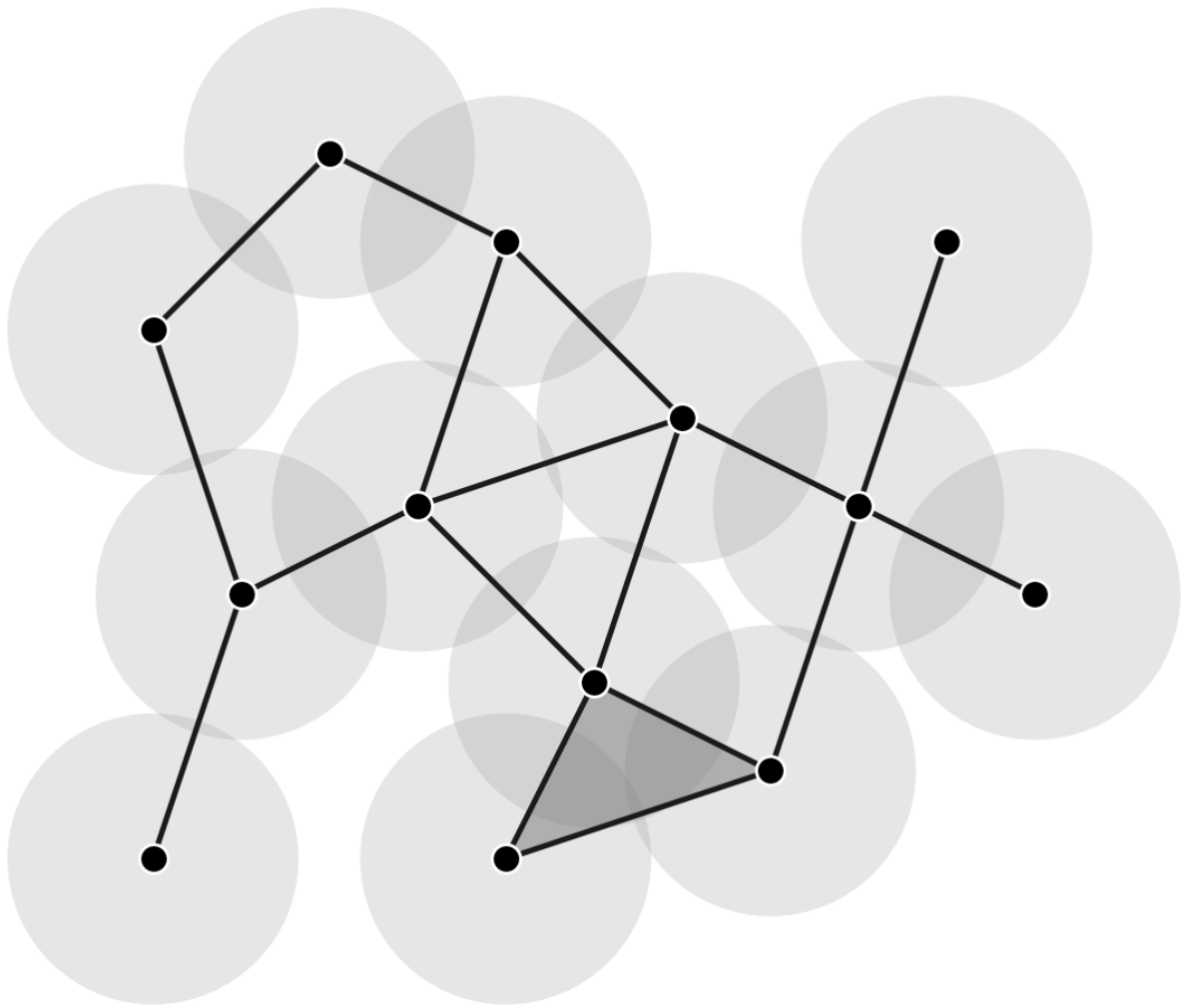
$$C_r(S, d) = \left\{ \{s_{i_0}, \dots, s_{i_k}\} : \bigcap_{j=0}^k B(s_{i_j}, r) \neq \emptyset \right\}$$

Monotonicity

For $r \leq s$: $C_r(S) \subseteq C_s(S)$



Point cloud



Čech complex

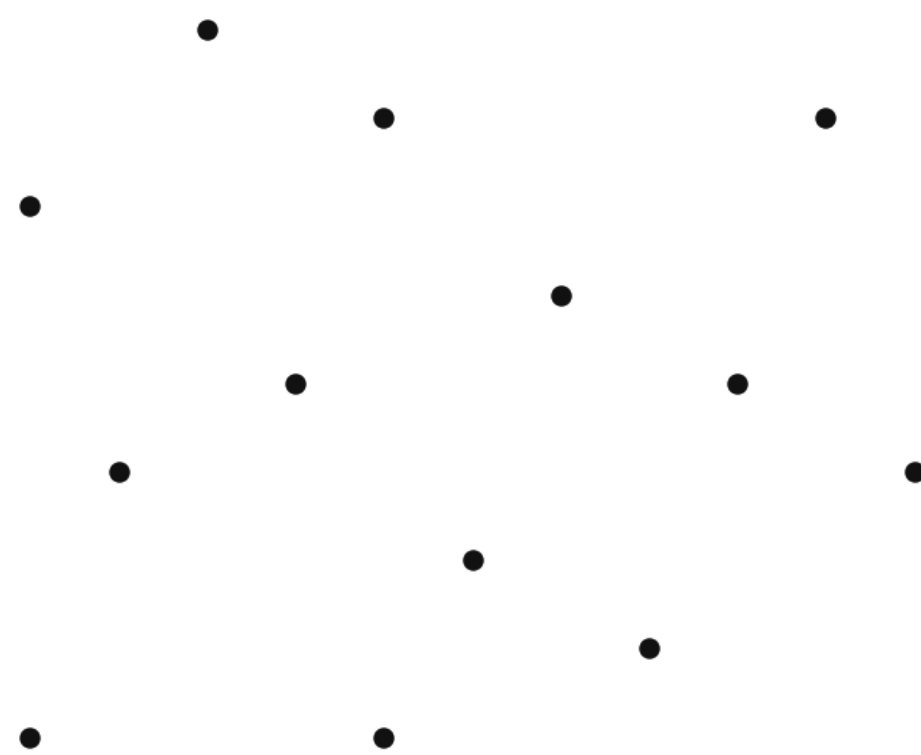
Recap: Vietoris—Rips complex

Given (S, d) and $r \geq 0$, the **Vietoris-Rips complex** at scale r is:

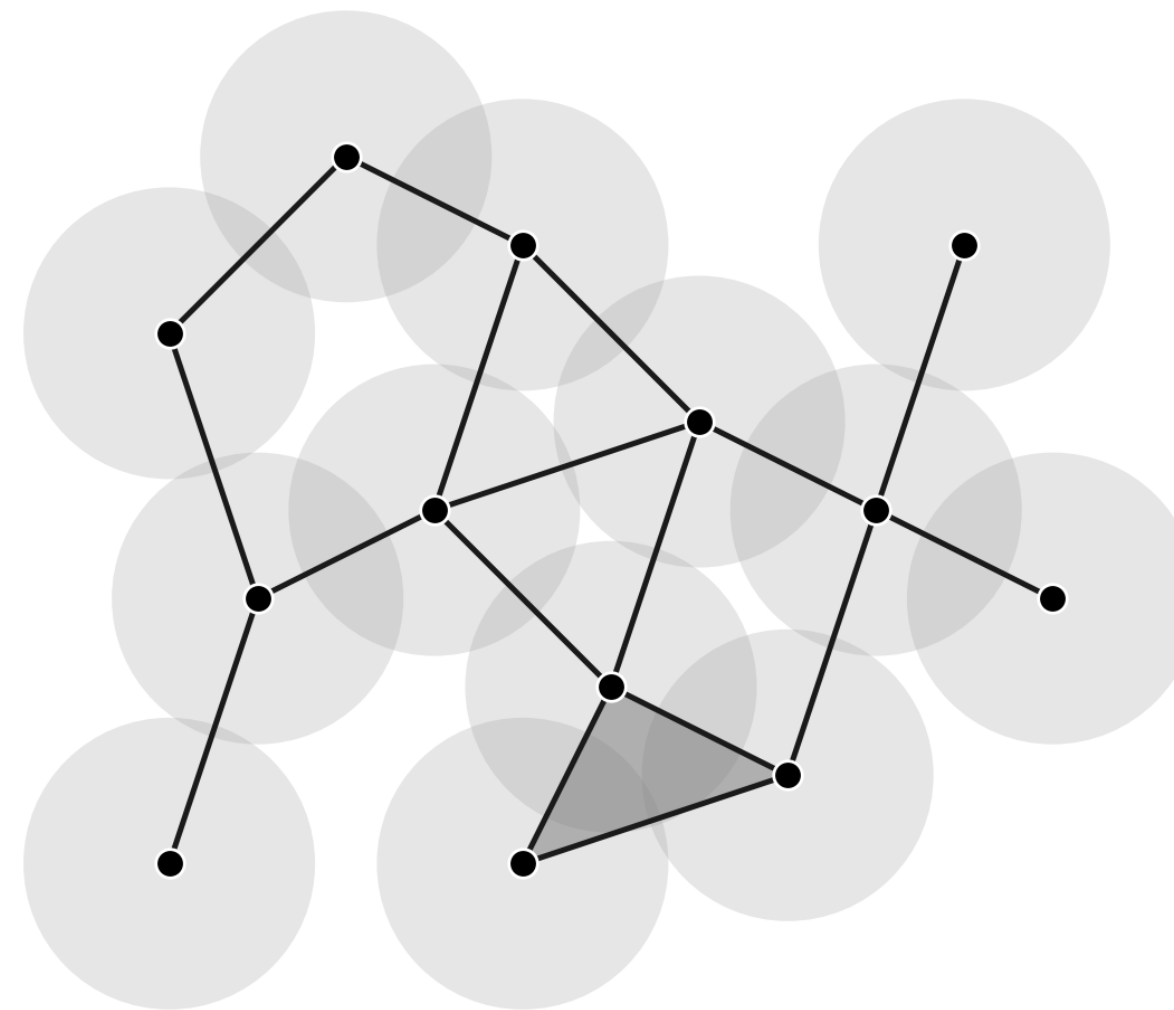
$$VR_r(S, d) = \left\{ \{s_{i_0}, \dots, s_{i_k}\} : d(s_i, s_j) \leq 2r, \forall i_0 \leq i < j \leq i_k \right\}$$

Monotonicity

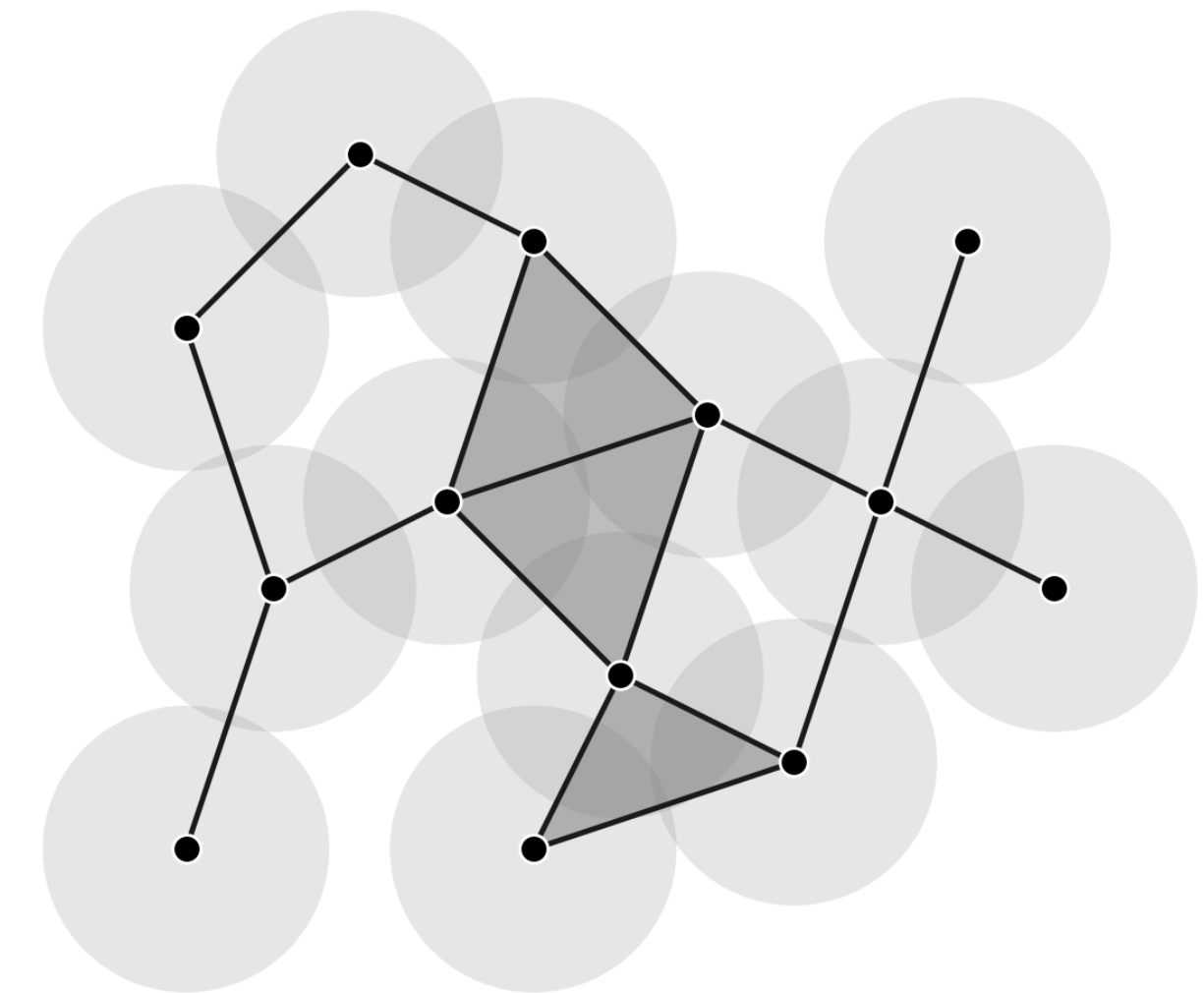
For $r \leq s$: $VR_r(S) \subseteq VR_s(S)$



Point cloud



Čech complex



VR complex

Nerve theorem

Def + thm. Nerve theorem (informal)

Let X be a topological space and $\mathcal{U} = \{B(x, r) : x \in S \subset X, r > 0\}$ a cover of this space, that is

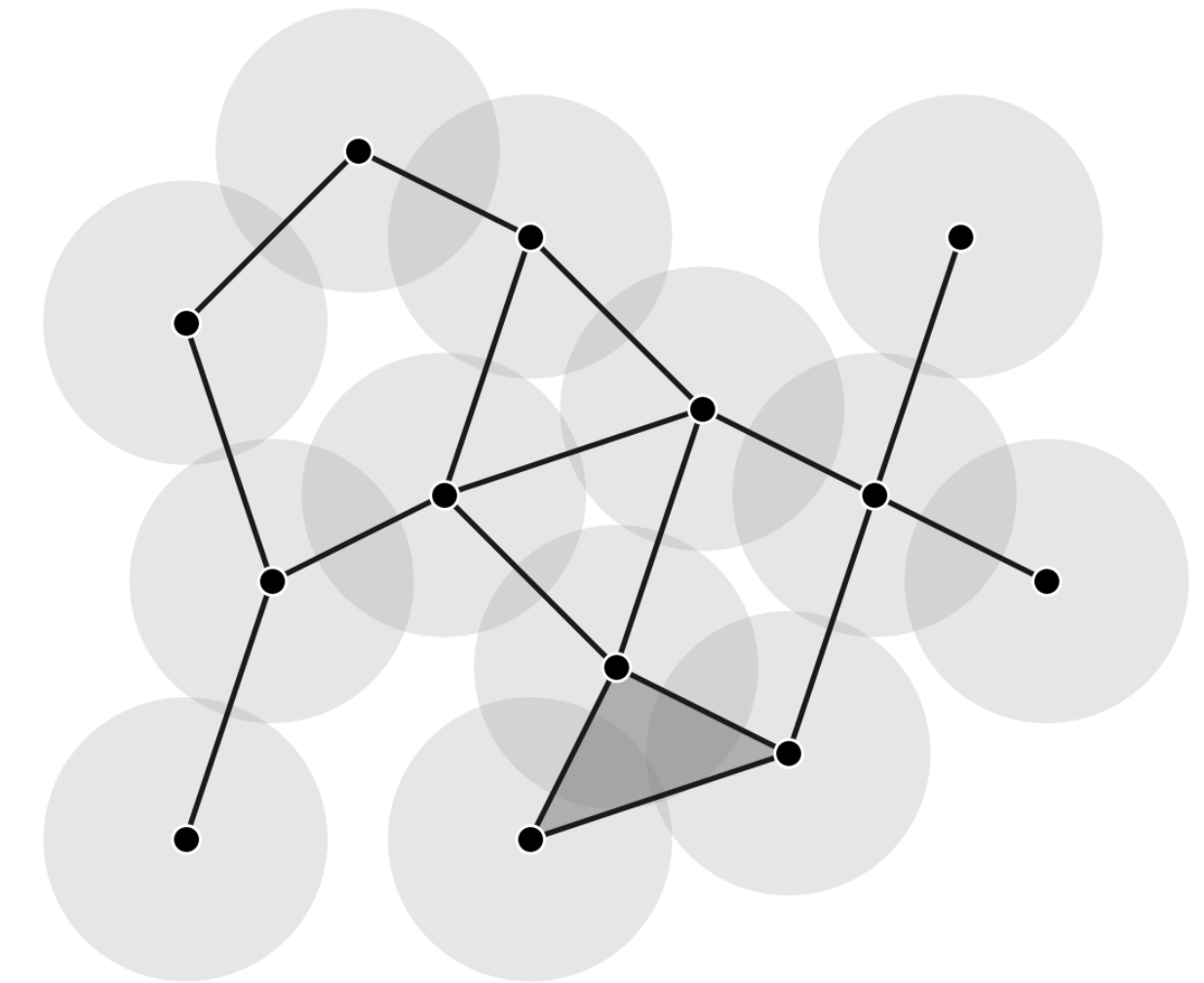
$$X \subset \bigcup_{x \in S} B(x, r)$$

The nerve of this cover is the following simplicial complex:

$$\text{Nerve}(\mathcal{U}) = \left\{ \sigma \subset S : \bigcap_{x \in \sigma} B(x, r) \neq \emptyset \right\}$$

With some *nice*ness assumptions on X and $\mathcal{U}$ we have:

$$\text{Nerve}(\mathcal{U}) \cong X \quad \textbf{(homotopy equivalent, same Betti numbers)}$$



The Čech complex is a nerve

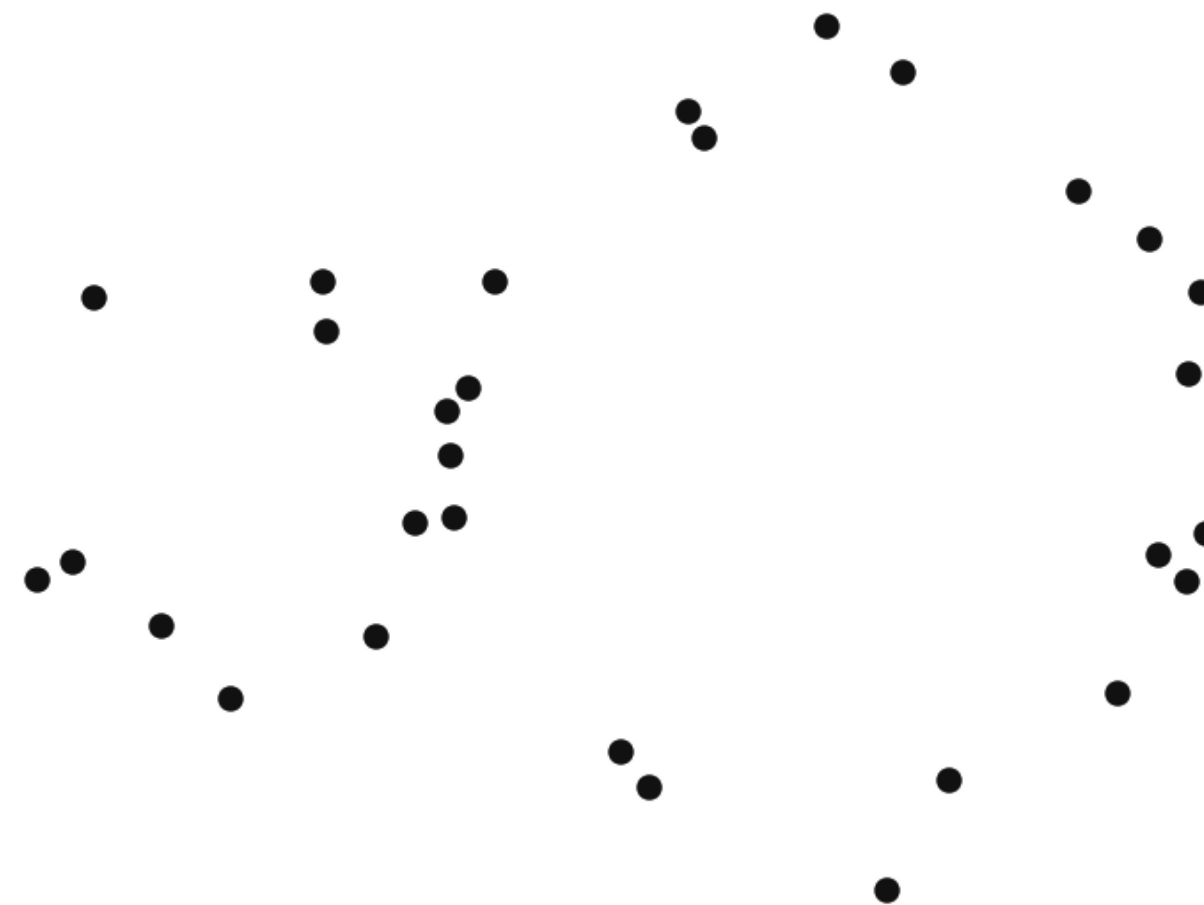
$$C_r(S, d) = \left\{ \{s_{i_0}, \dots, s_{i_k}\} : \bigcap_{j=0}^k B(s_{i_j}, r) \neq \emptyset \right\}$$

¹Ghrist, R., and Muhammad, A. “Coverage and Hole-Detection in Sensor Networks via Homology.” *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks*, 2005., April 2005, 254–60.

¹Bauer, U., Kerber, M., Roll, F., and Rolle, A. “A Unified View on the Functorial Nerve Theorem and Its Variations.” *Expositiones Mathematicae* 41, no. 4 (2023): 125503.

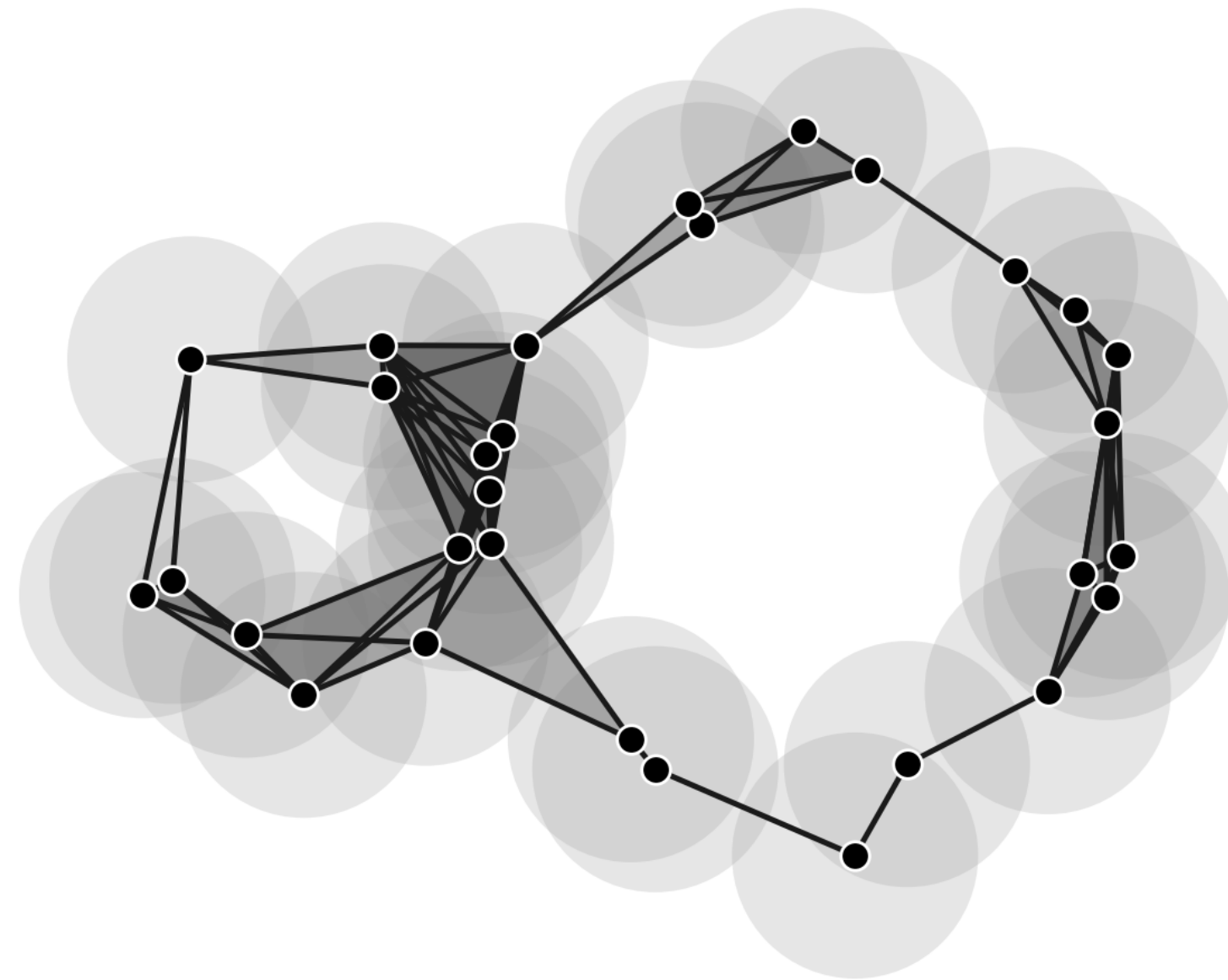
Nerve theorem: practical consequences

Suppose you are given the following point cloud, how can you design a test to conclude that this sample comes from a distribution over two circles?



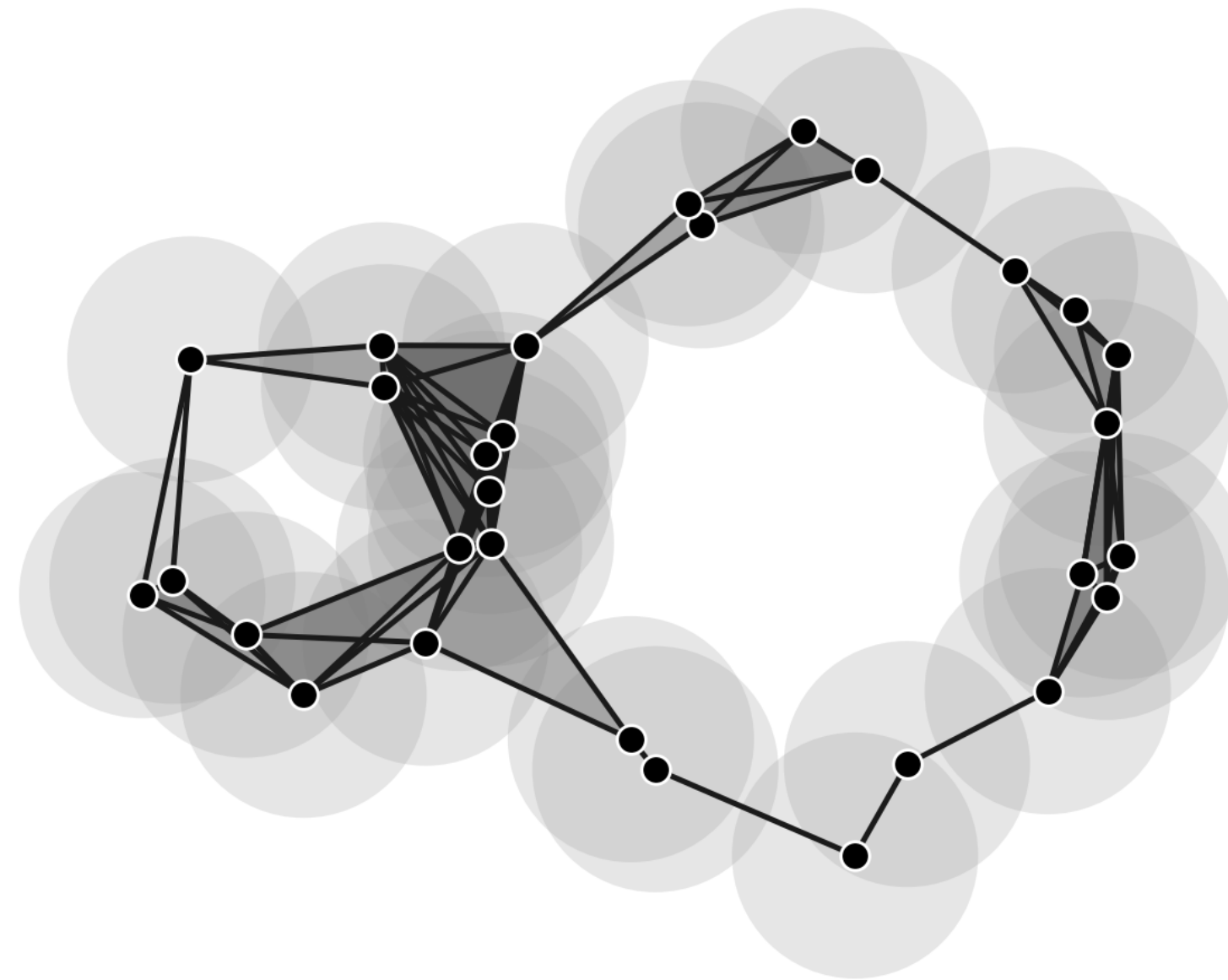
Nerve theorem: practical consequences

1. Approximate S with a Čech complex for a some radius that captures the holes:
 $C_r(S)$
2. Compute homology in degree 1: $H_1(C_r(S))$
3. If $\beta_1(C_r(S)) = 2$, then you are observing two holes



Nerve theorem: practical consequences

1. Approximate S with a Čech complex for a some radius that captures the holes:
 $C_r(S)$
2. Compute homology in degree 1: $H_1(C_r(S))$
3. If $\beta_1(C_r(S)) = 2$, then you are observing two holes



How do you find the radius r ?

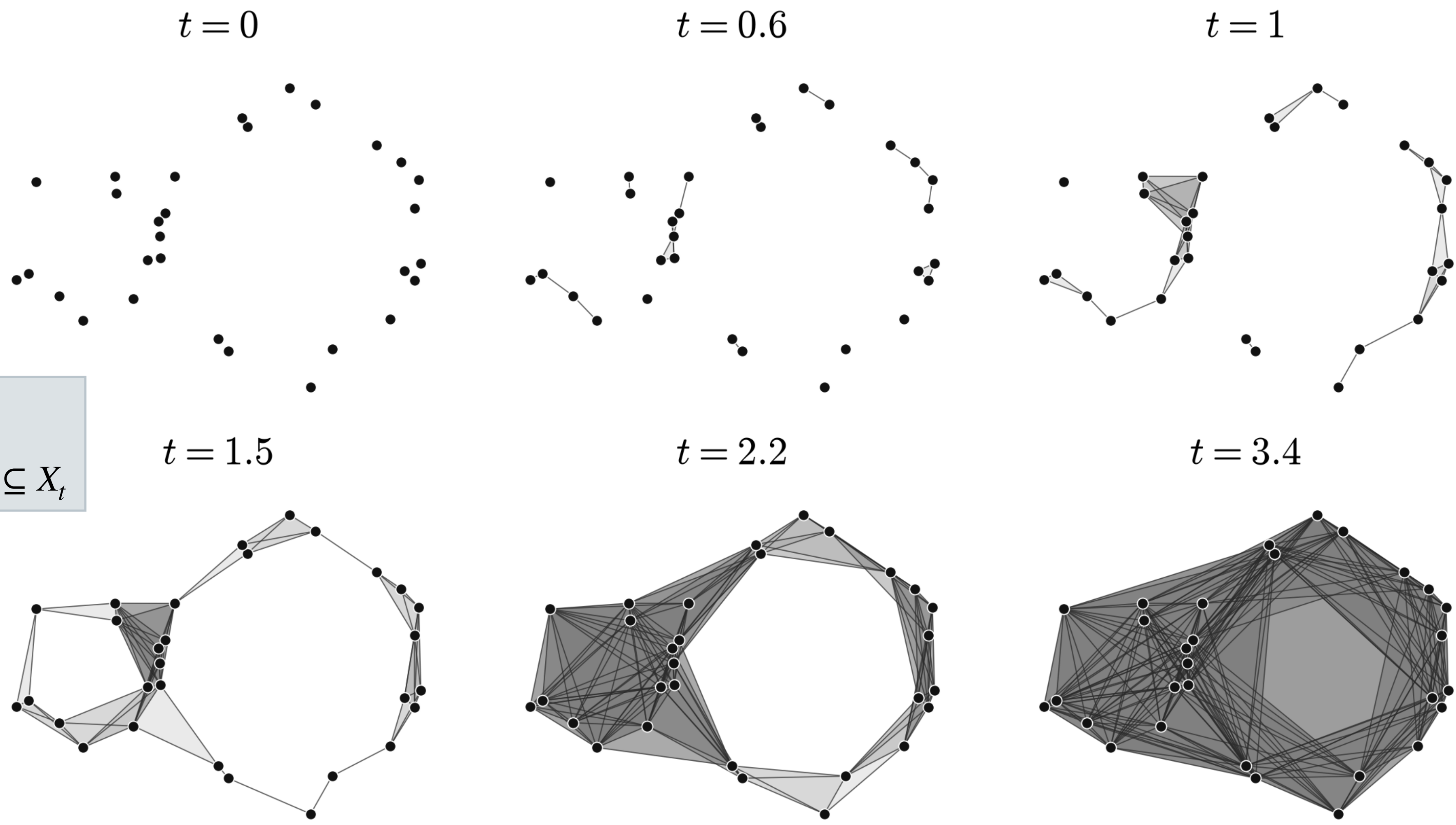
Persistent homology

Lecture 7 — (More) persistent homology

Filtrations

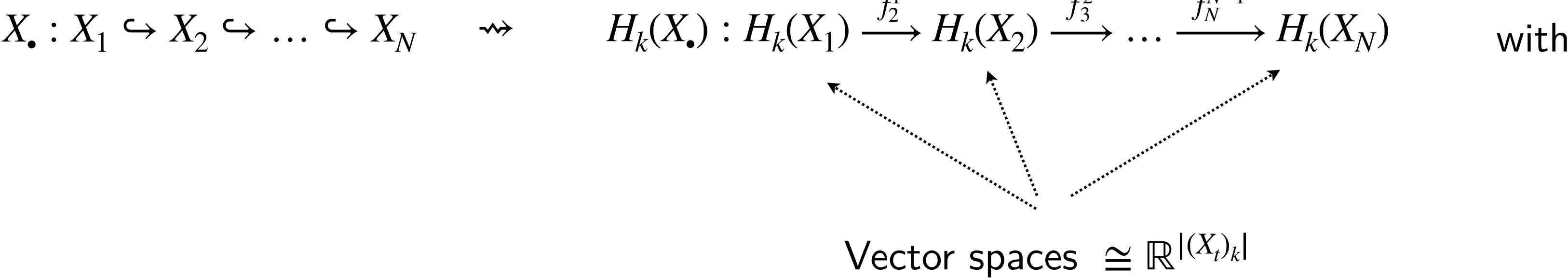
Consider all possible radii!

Def. Filtration
 $\{X_t : t \in \mathbb{R}\}$ such that for $s \leq t$, $X_s \subseteq X_t$



Persistent homology module

Persistent homology module:



From the property about compositions and induced maps!

$X_l \subseteq X_n \subseteq X_m$

$f_m^n \circ f_n^l = f_m^l$

Persistence module

Def. Persistence module (in full generality)

A family $\{V_n\}_{1 \leq n \leq N}$ of vector spaces and a family $\{\phi_m^n : V_n \rightarrow V_m\}_{n \leq m}$ of linear maps satisfying:

- $\phi_n^n = \text{id}_{V_n}$
- $\phi_n^m \circ \phi_m^l = \phi_n^l$ for all $l \leq m \leq n$

An example of persistence modules are interval modules:

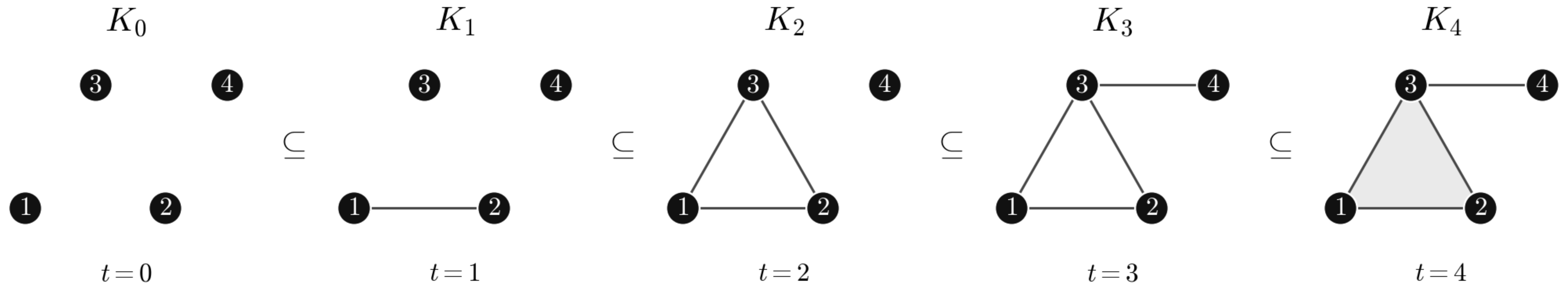
$$I[a, b]_n = \begin{cases} 0 & \text{if } n < a, \text{ or } n > b \\ \mathbb{R} & \text{if } a \leq n \leq b \end{cases} \qquad I[a, b]_m^n = \begin{cases} 0 & \text{if } n < a, \text{ or } m > b \\ \text{id}_{\mathbb{R}} & \text{if } a \leq n \leq m \leq b \end{cases}$$

Module: $0 \rightarrow 0 \rightarrow \cdots \rightarrow 0 \rightarrow \mathbb{R} \xrightarrow{\text{id}_{\mathbb{R}}} \mathbb{R} \rightarrow \cdots \rightarrow \mathbb{R} \xrightarrow{\text{id}_{\mathbb{R}}} \mathbb{R} \rightarrow 0 \rightarrow \cdots \rightarrow 0$

Parameters: $1 \quad 2 \qquad a-1 \quad a \quad a+1 \qquad b-1 \quad b \quad b+1 \qquad N$

Persistent homology

An example

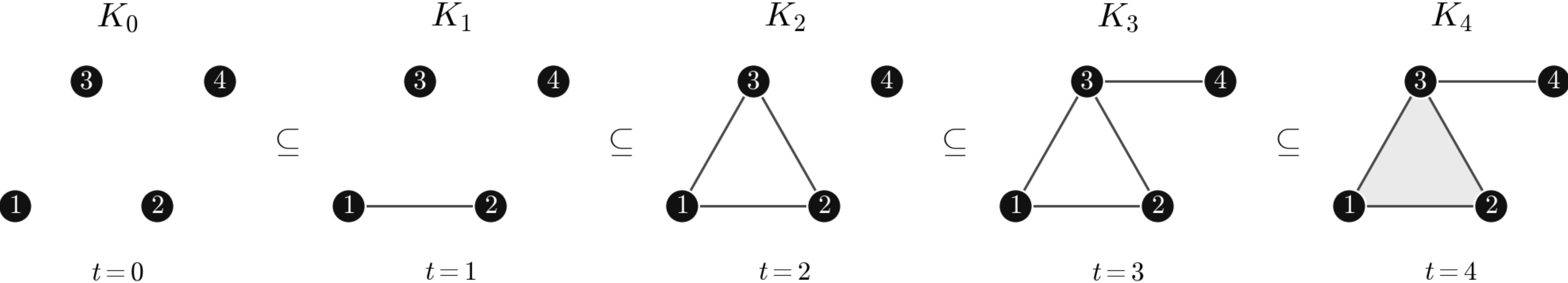


For 0 and 1-dimensional homology:

- Choose generators for the homology groups at each step in the filtration
- Connect the generators in subsequent steps in the filtration whenever the map induced in homology by the inclusion connects them
- No line needs to be drawn when the generator is directly mapped to 0 by the induced map

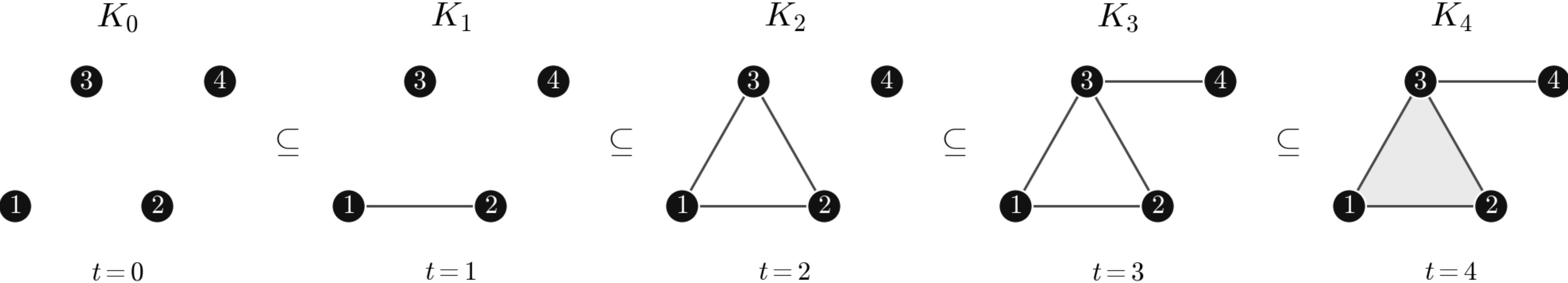
Persistent homology

$H_0(K.)$



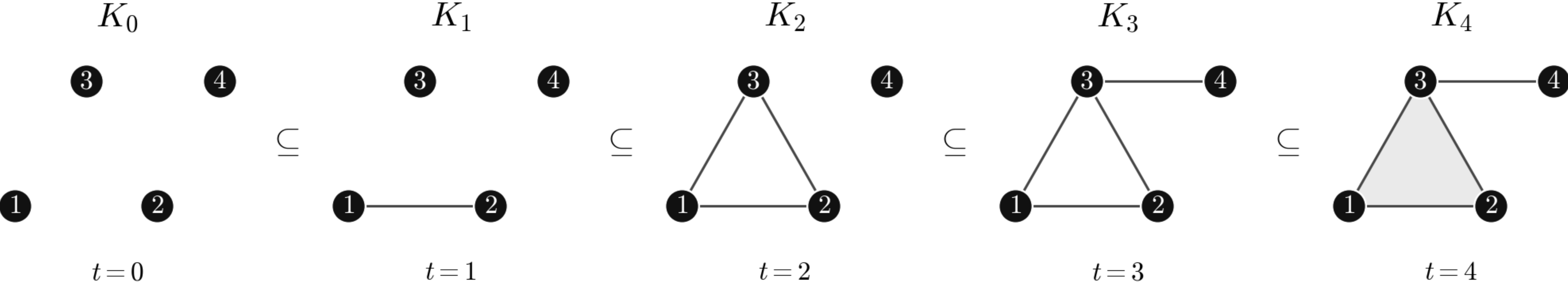
Persistent homology

$H_1(K.)$



Persistent homology

Choosing generators more wisely

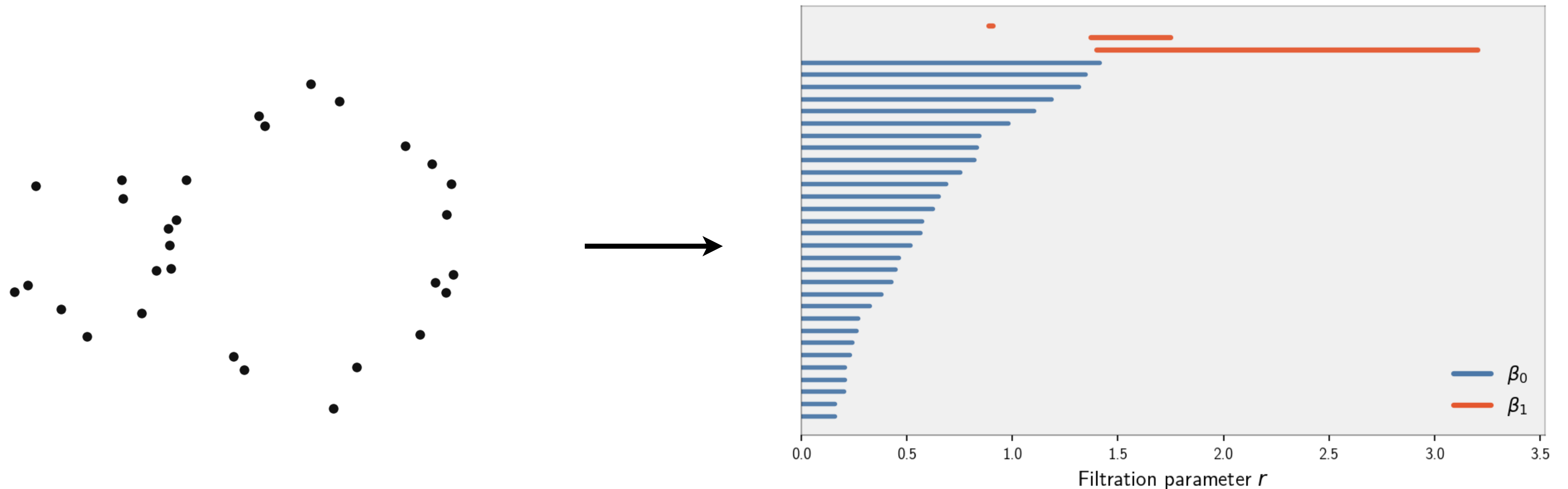


Persistence barcode

This is true in general by the **structure theorem**, whose statement and proof go beyond the scope of the course

It is enough that you know that for each filtration (and corresponding persistent homology module) we can compute a **barcode** that tracks how homology classes are created and destroyed as the filtration grows

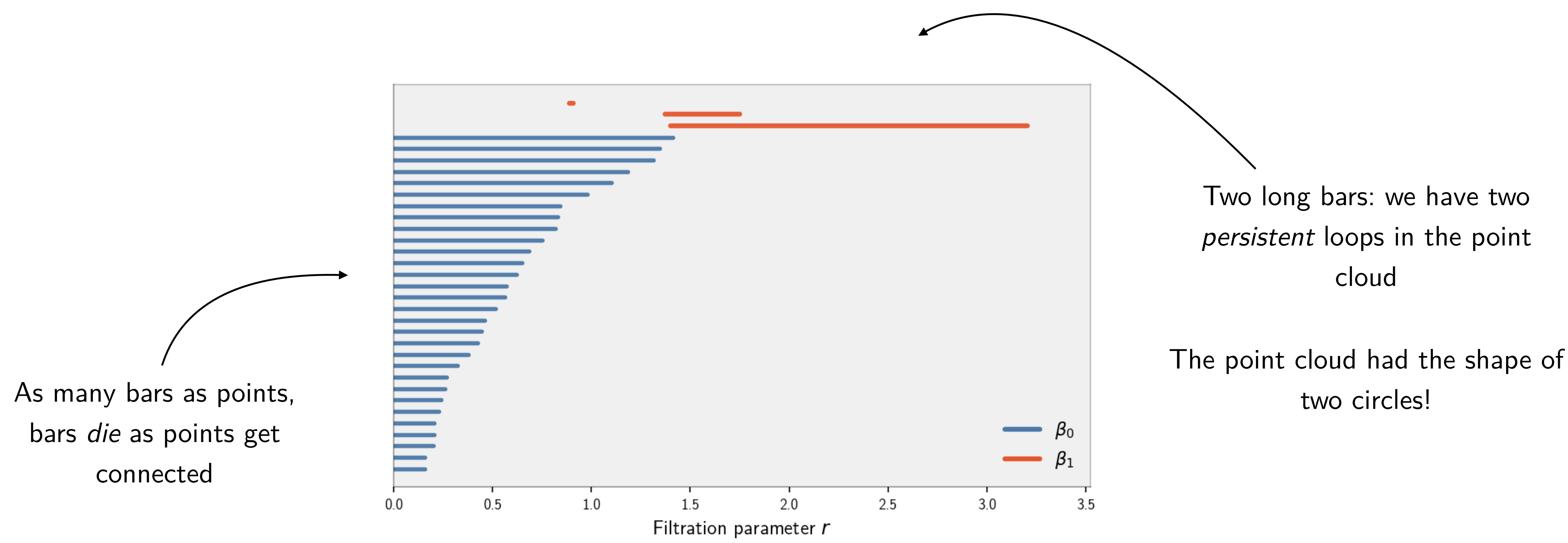
This is the same as tracking the evolution of the Betti numbers of the complexes in the filtration



Azumaya, Gorô. "Corrections and Supplementaries to My Paper Concerning Krull-Remak-Schmidt's Theorem." *Nagoya Mathematical Journal* 1 (June 1950): 117–24.

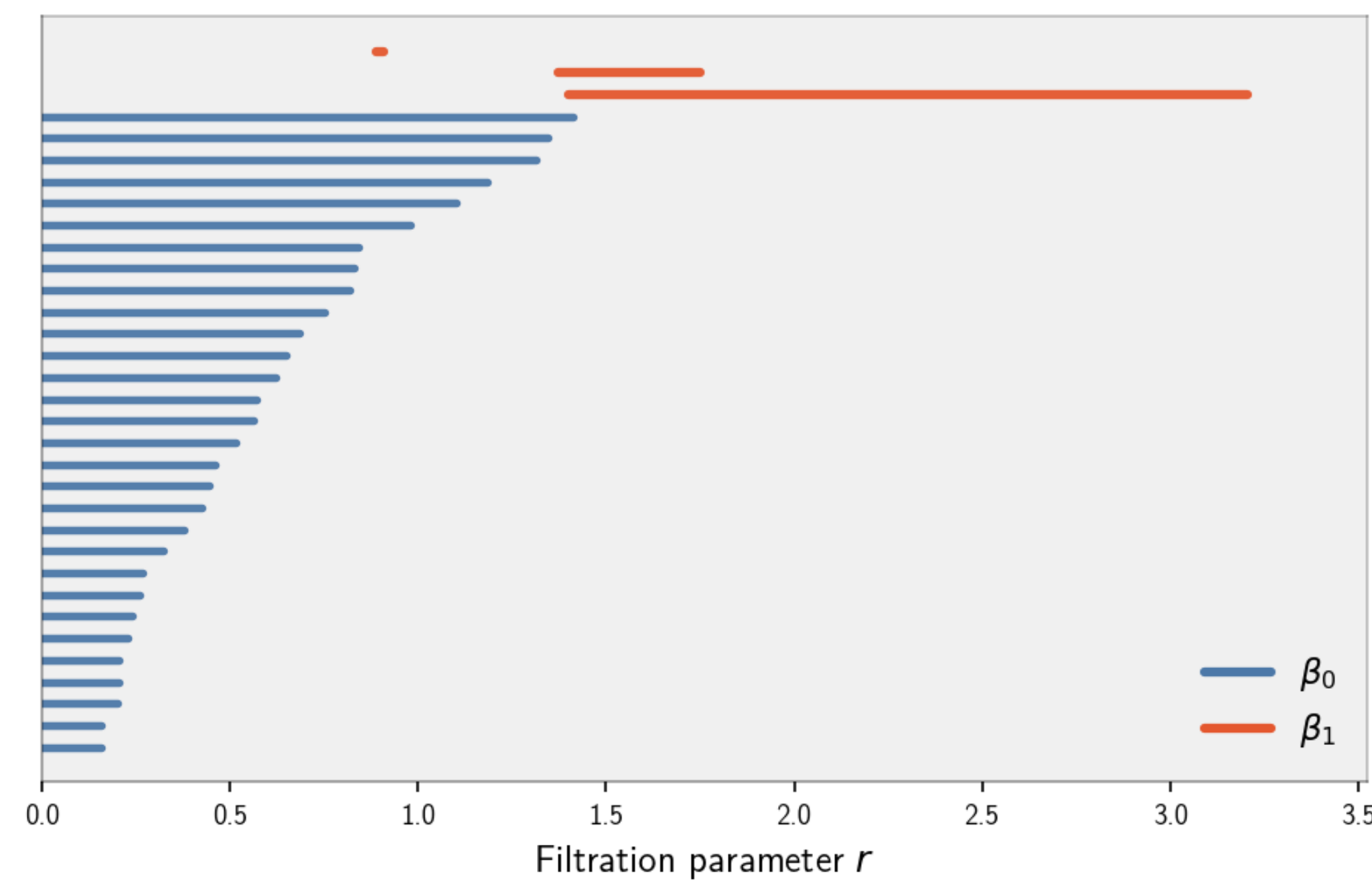
Crawley-Boevey, William. "Decomposition of Pointwise Finite-Dimensional Persistence Modules." *Journal of Algebra and Its Applications* 14, no. 05 (2015): 1550066.

Persistence barcode

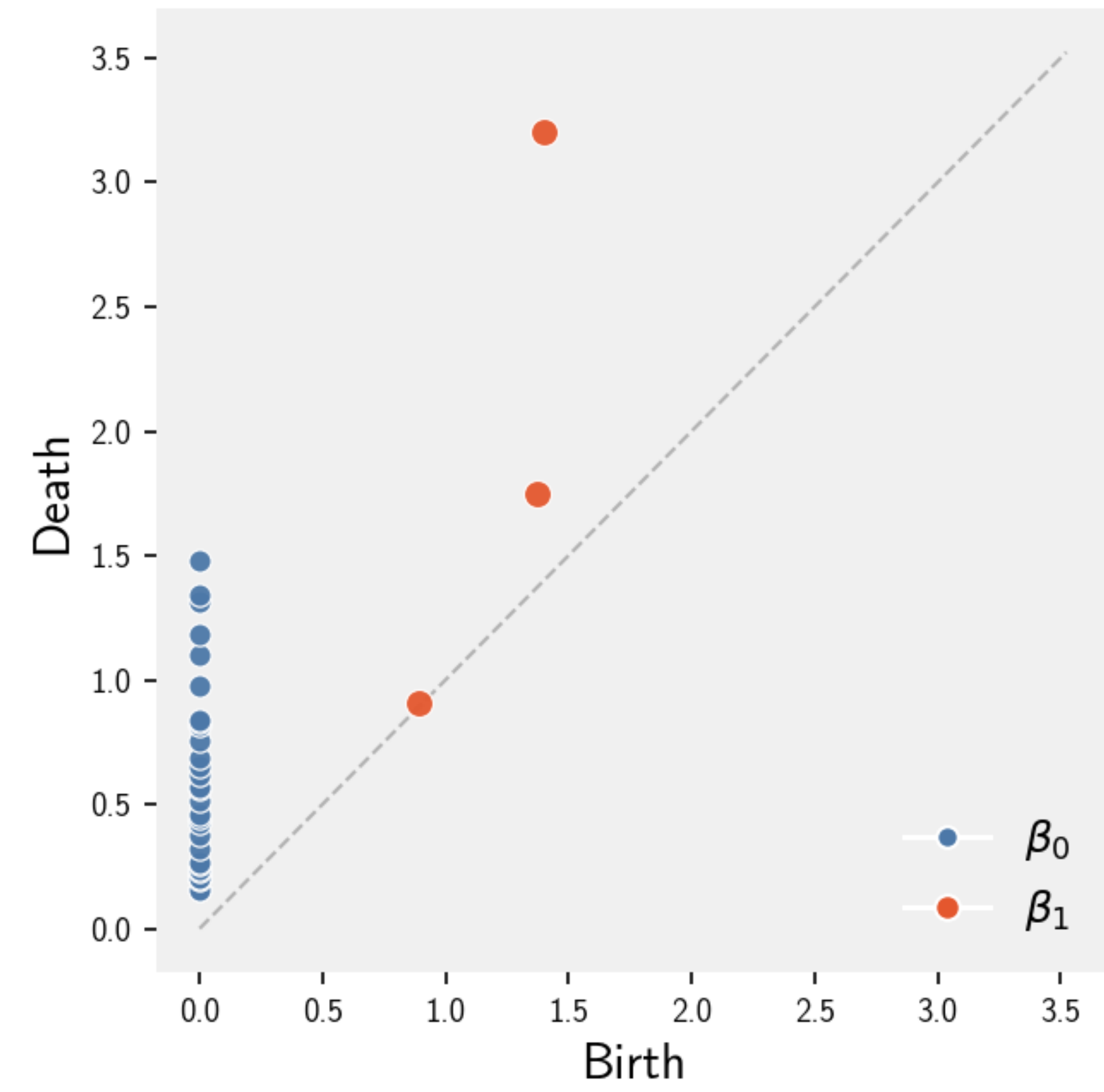


Persistence barcode

Persistence diagram



Persistence barcode



Persistence diagram

Computing barcodes

Lecture 7 — (More) persistent homology

Filtration boundary matrix

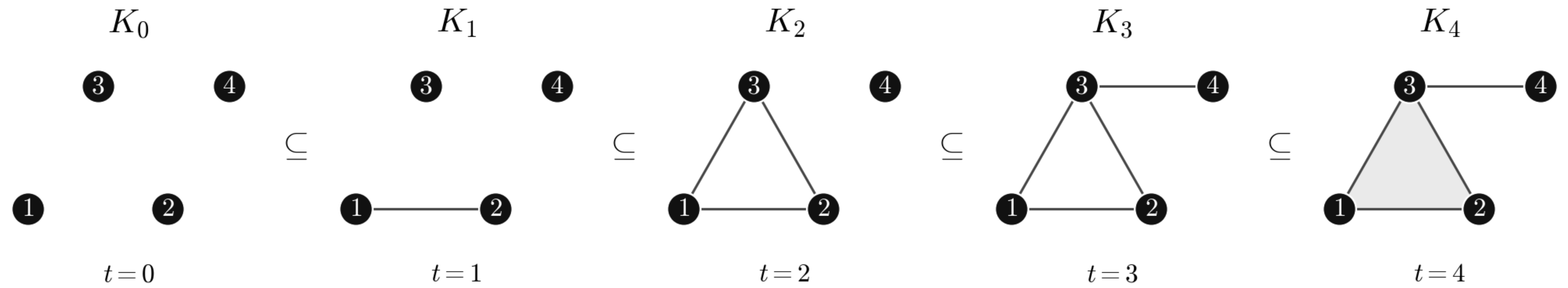
How to define it?

- Order all simplices in the simplicial complex at the last step of the filtration $K = K_N$ by increasing filtration value
- When several simplices enter the filtration at the same time, order them by dimension
- Break ties arbitrarily when two simplices of the same dimension enter the filtration at the same time
- Compute the boundary matrices D_k representing the boundary maps $\partial_k : C_k(K) \rightarrow C_{k-1}(K)$ of the simplicial complex at the last step of the filtration, where all simplices have been added
- Write the full boundary matrix as the block matrix given by all these matrices

$$D = \bigoplus_{k=0}^{\dim K} D_k$$

- D_{ij} = coefficient of σ_i in $\partial\sigma_j$ with indices ranging all the simplices in K

Reducing the boundary matrix



Reducing the boundary matrix

Reduction algorithm

Input: boundary matrix of a filtration D

Steps:

$$R \leftarrow D$$

$$n \leftarrow \# \text{ columns of } R$$

For $1 \leq j \leq n$:

 If $R_j \neq 0$:

$$\text{pivot}(R_j) \leftarrow \max\{i : R_{ij} \neq 0\}$$

 For $1 \leq k < j$:

 If $\text{pivot}(R_k) = \text{pivot}(R_j)$:

$$R_j \leftarrow R_j \pm R_k \text{ so that } \text{pivot}(R_j) \text{ increases}$$

Output: reduced matrix R

Example

$$D = \left(\begin{array}{c|cccc|cccc|c} & 1 & 2 & 3 & 4 & 12 & 13 & 23 & 34 & 123 \\ \hline 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 23 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 123 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Reading off the barcode

- If $\text{pivot}(R_j) = i$: σ_i creates a homology class and σ_j kills it. We have to add a finite bar

$$[f(\sigma_i), f(\sigma_j))$$

where $f(\sigma_i)$ = filtration value at which σ_i enters the filtration, in dimension $\dim(\sigma_i)$

- If $R_j = 0$, that means that this column represents a simplex that creates a cycle. There are two options for that bar:

- There exists a later column R_k , $j < k$ such that $\text{pivot}(R_k) = j$. Then the cycle created by σ_j is killed by σ_k and we have to add

$$[f(\sigma_j), f(\sigma_k))$$

in dimension $\dim(\sigma_j)$

- Otherwise, the cycle created by σ_j never gets killed, so we get an essential bar

$$[f(\sigma_j), +\infty)$$

in dimension $\dim(\sigma_j)$

The idea behind the algorithm

Notice that we are doing exactly the same thing as we did before:

- Starting from the basis of simplices in each dimension (basis of the chain space)
- We add or subtract simplices that have already appeared in the filtration to reduce the boundary, that is, we change the element of the basis over which we are computing the boundary, adding the vectors of the basis that we already had
- We do this until we obtain a boundary that is linearly independent with respect to the other boundaries already present, or until we arrive to the zero column (the combination of simplices is a cycle)
- This process gives us a basis from which we can read the barcode, just as before!

To sum up

Today we have learnt how to...

- How to define the persistence module
- How to represent a persistence module as a persistence barcode or a persistence diagram
- How to compute barcodes

Thank you for your attention!



AIDOS LAB
AI FOR DATA-ORIENTED SCIENCE

