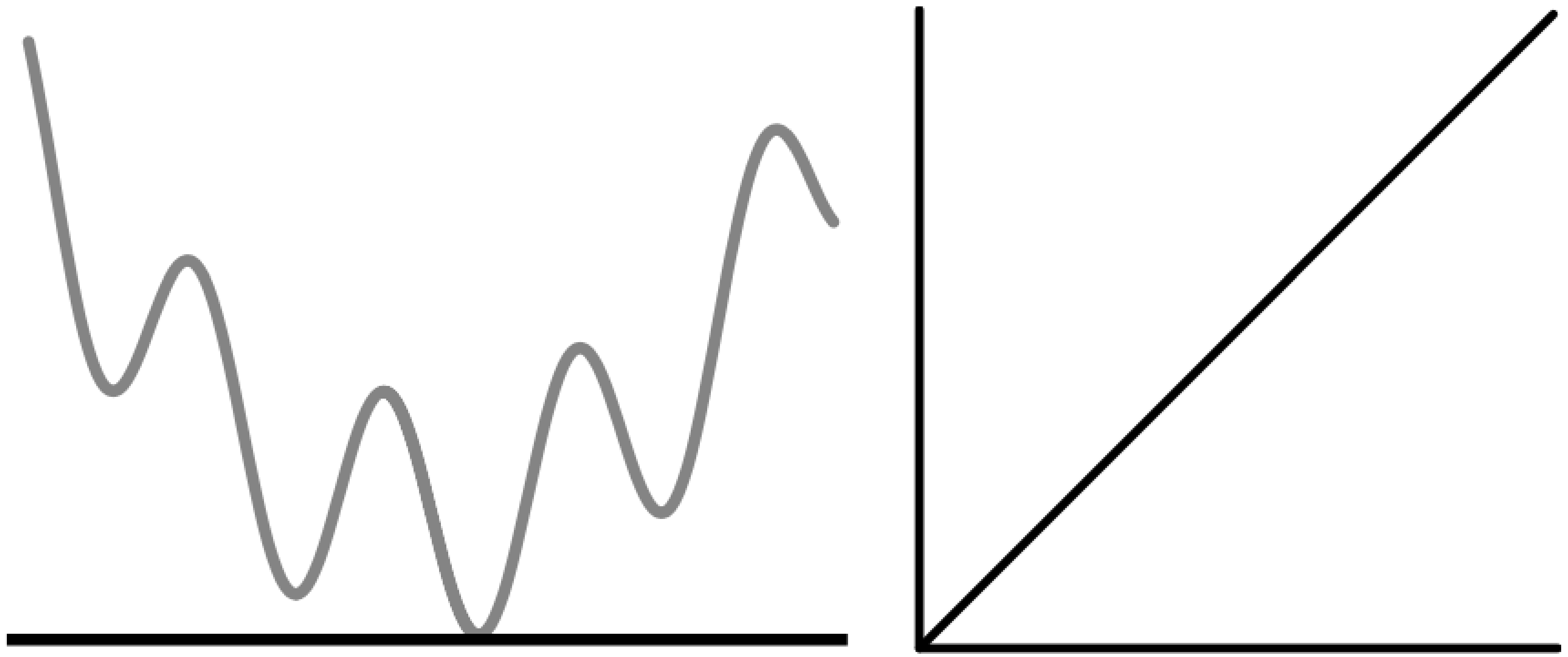
An introduction to persistent homology

MUSTAFA HAJIJ

Part I : Scalar Functions

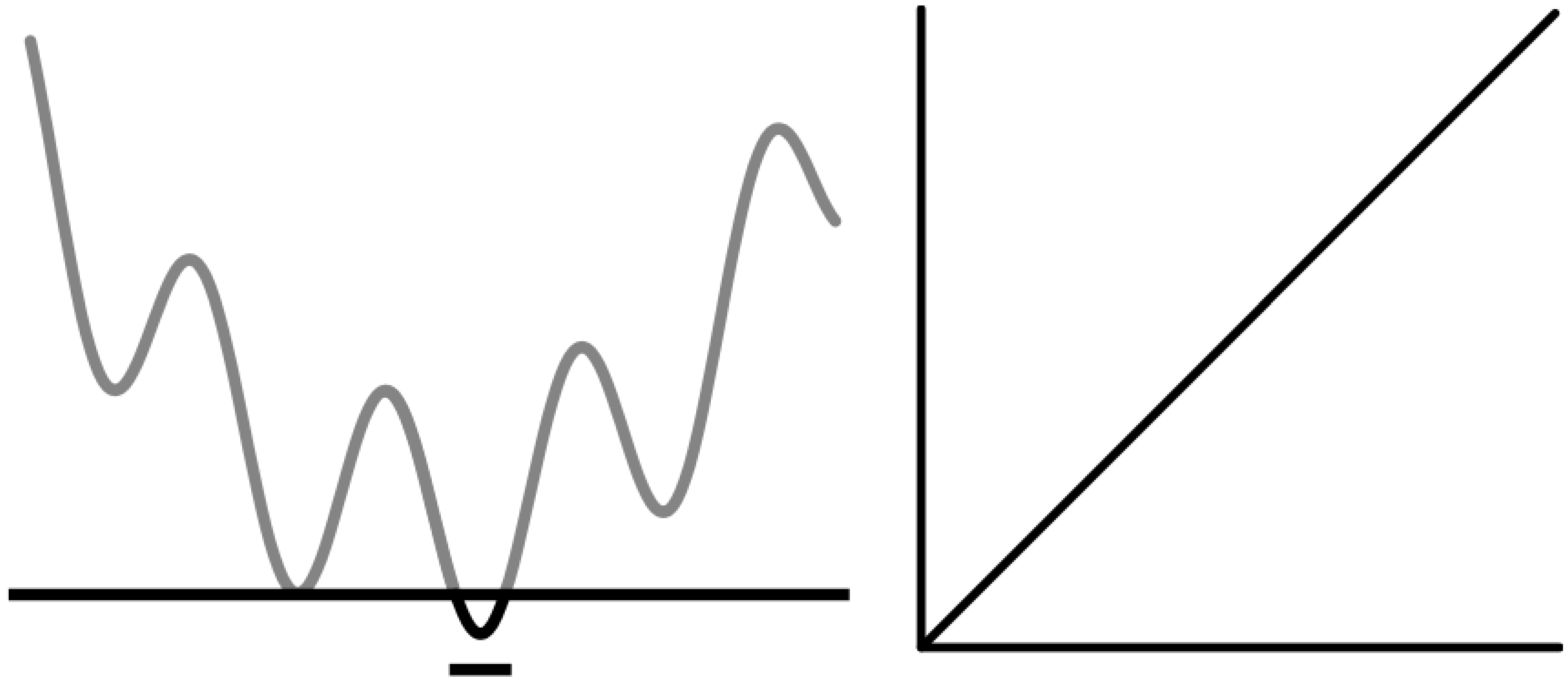
Persistence Diagram of a scalar function

- Track the evolution of the topology of sub-level sets as the threshold increases.
- Pair thresholds that create components with those that destroy them.



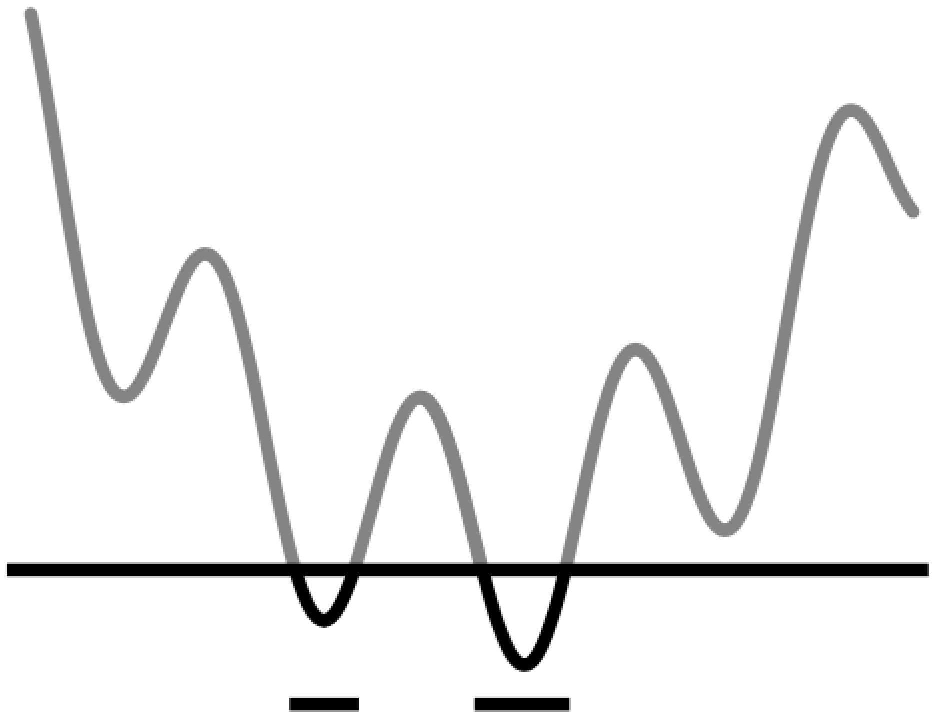
Persistence Diagram of a scalar function

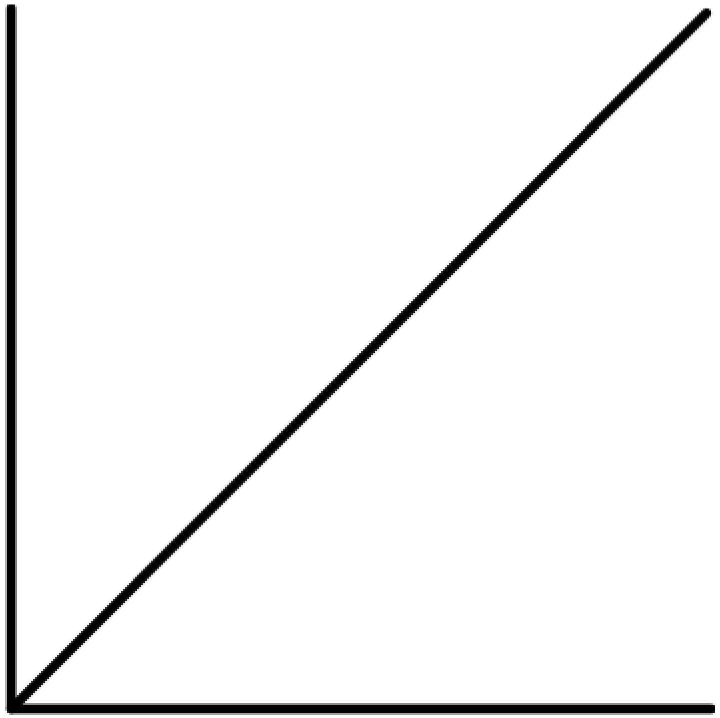
- Track the evolution of the topology of sub-level sets as the threshold increases.
- Pair thresholds that create components with those that destroy them.



Persistence Diagram of a scalar function

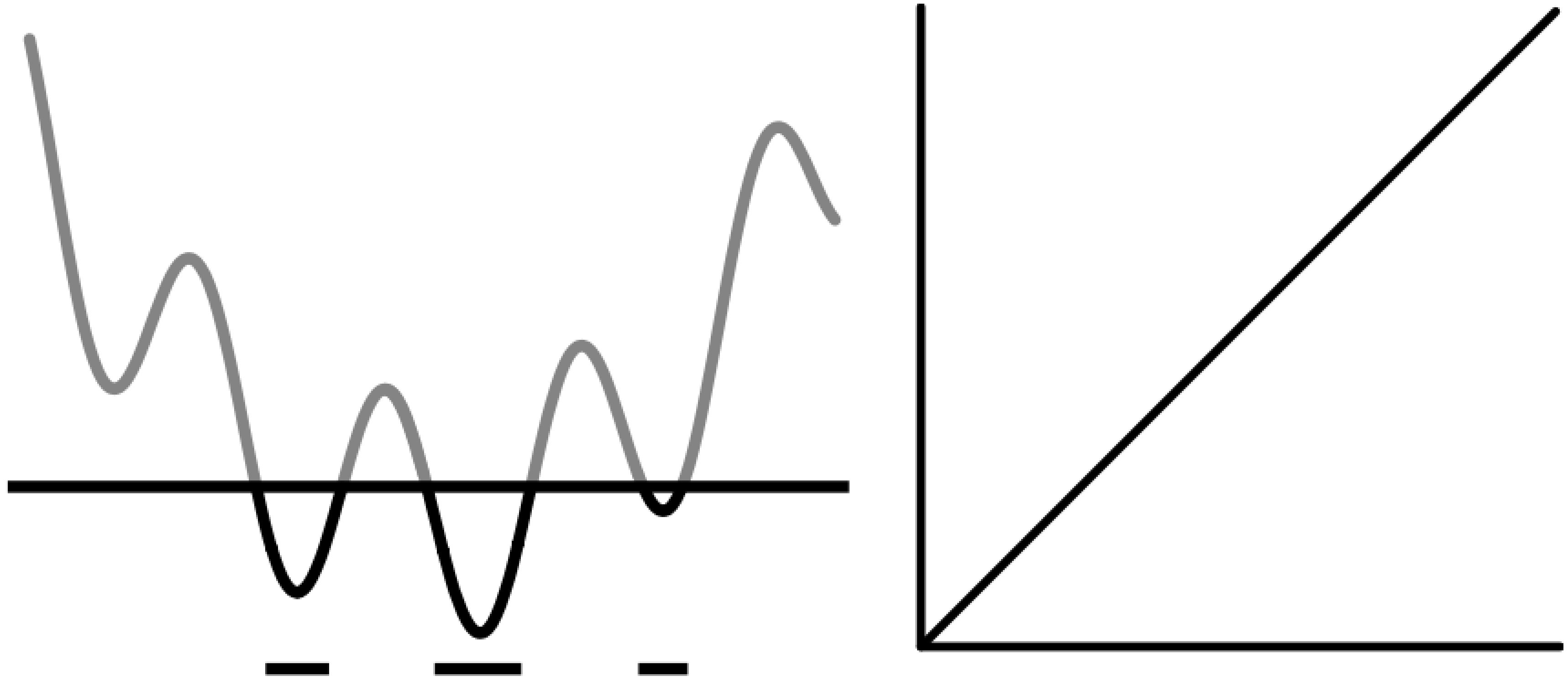
- Track the evolution of the topology of sub-level sets as the threshold increases.
- Pair thresholds that create components with those that destroy them.





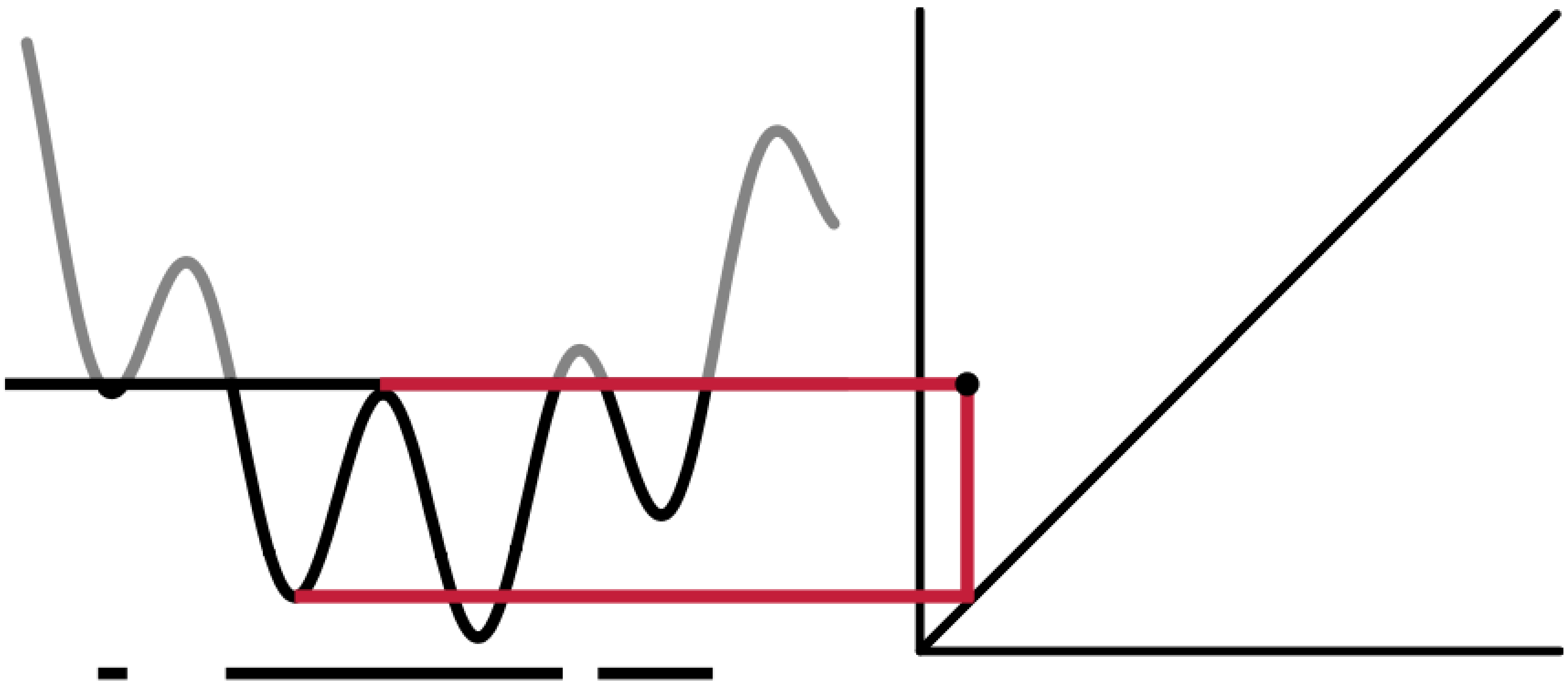
Persistence Diagram of a scalar function

- Track the evolution of the topology of sub-level sets as the threshold increases.
- Pair thresholds that create components with those that destroy them.



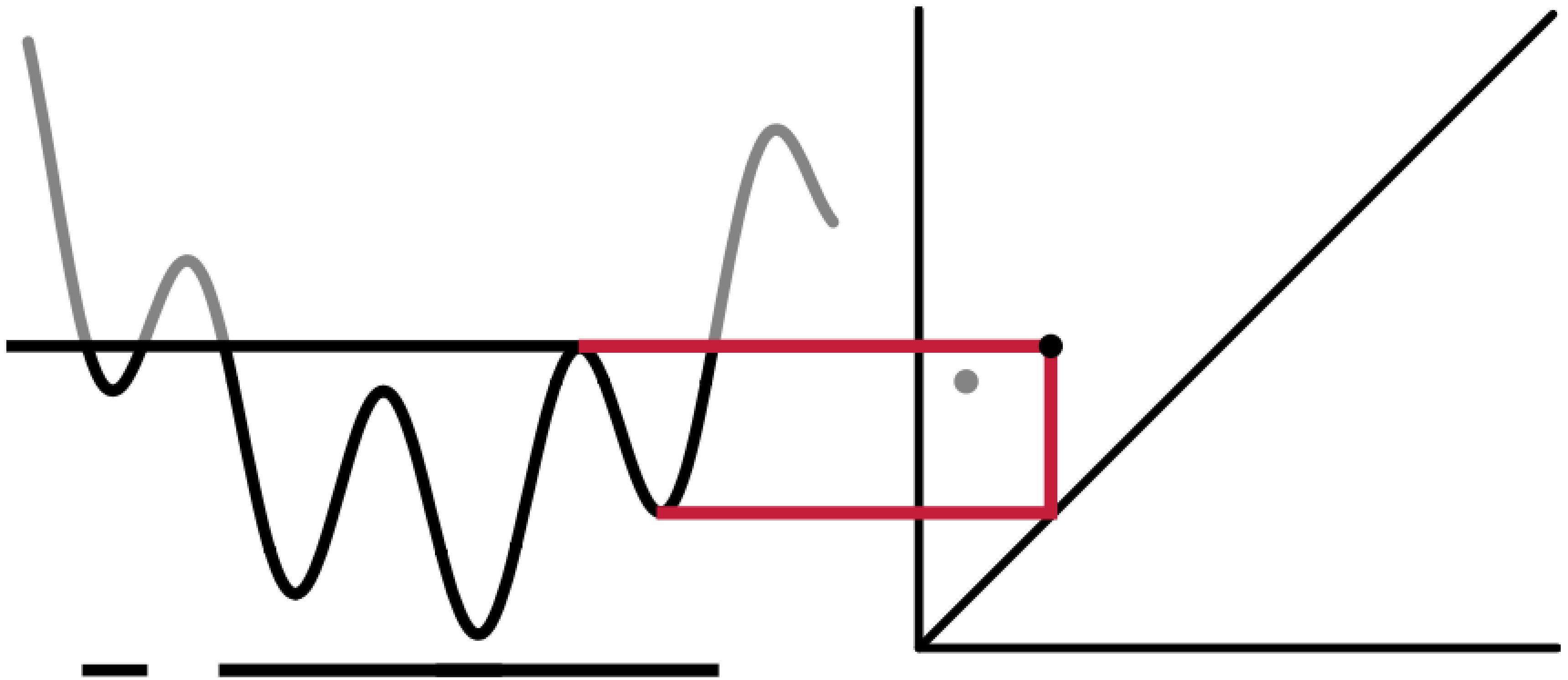
Persistence Diagram of a scalar function

- Track the evolution of the topology of sub-level sets as the threshold increases.
- Pair thresholds that create components with those that destroy them.



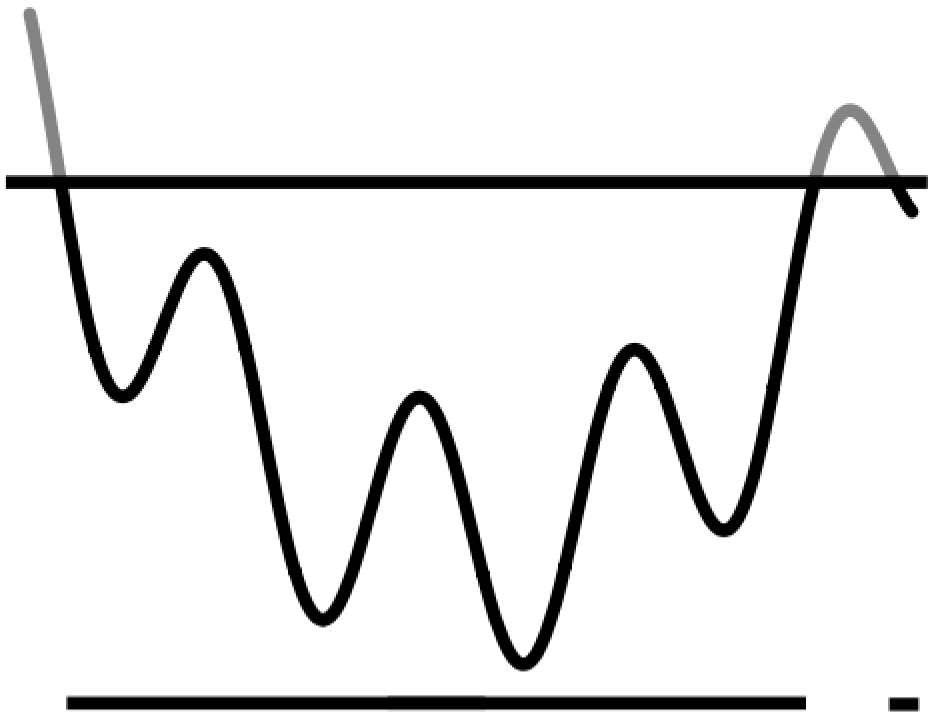
Persistence Diagram of a scalar function

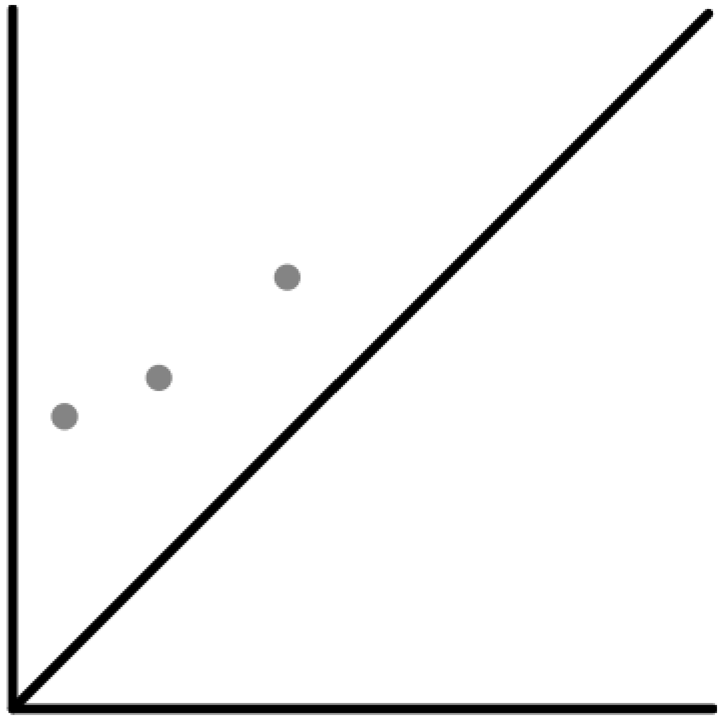
- Track the evolution of the topology of sub-level sets as the threshold increases.
- Pair thresholds that create components with those that destroy them.



Persistence Diagram of a scalar function

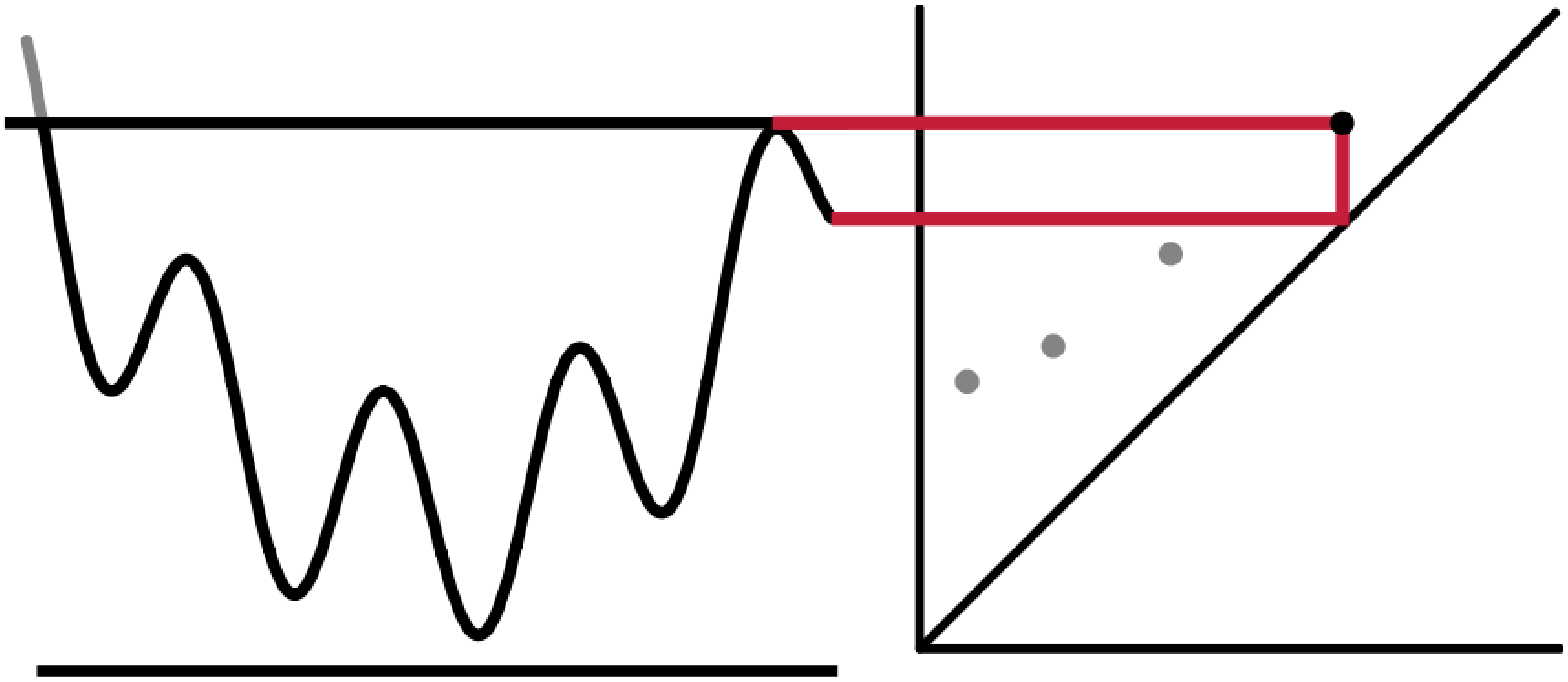
- Track the evolution of the topology of sub-level sets as the threshold increases.
- Pair thresholds that create components with those that destroy them.





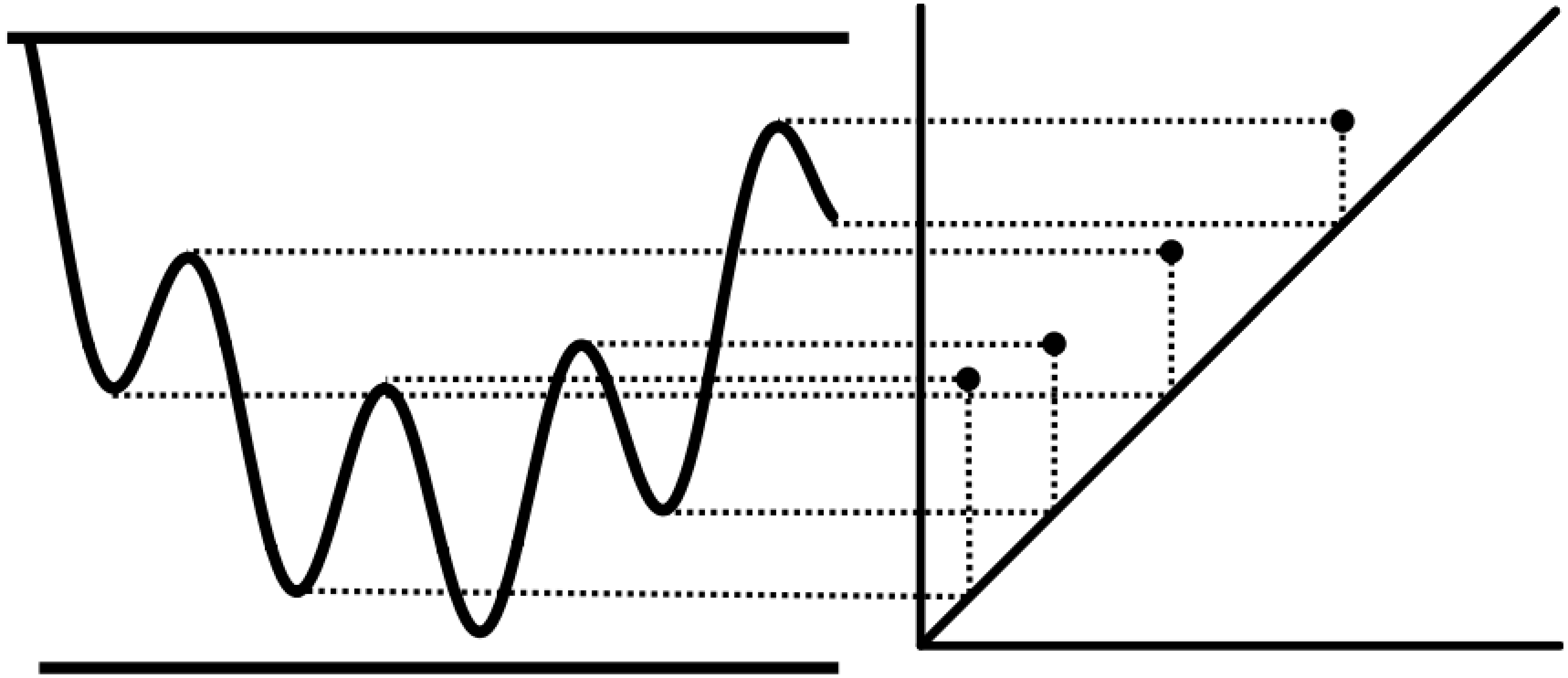
Persistence Diagram of a scalar function

- Track the evolution of the topology of sub-level sets as the threshold increases.
- Pair thresholds that create components with those that destroy them.

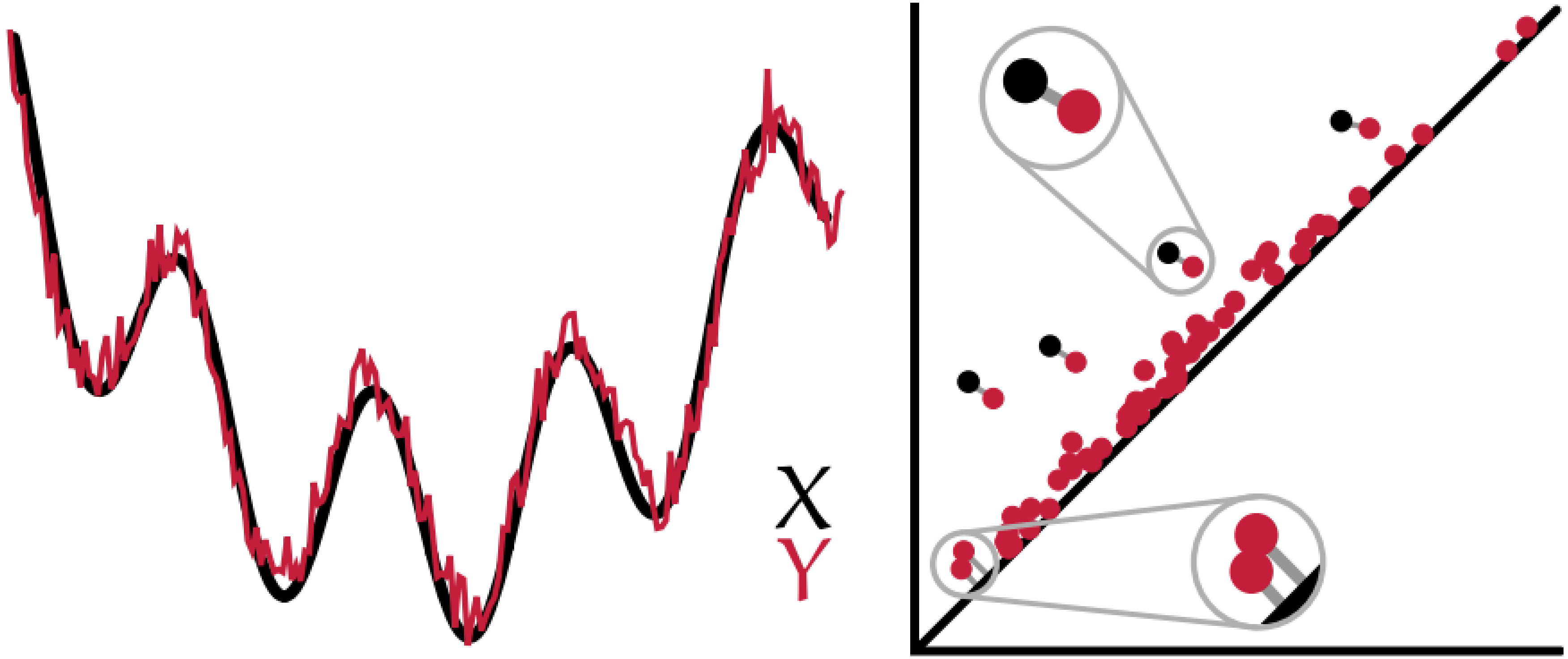


Persistence Diagram of a scalar function

- Track the evolution of the topology of sub-level sets as the threshold increases.
- Pair thresholds that create components with those that destroy them.



Another example



Persistence Diagram of a scalar function

Algorithm 3: Calculating discrete 0-dimensional persistent homology

Require: A discrete sample $\{(x_1, y_1), (x_2, y_2), \ldots\}$ of a function $f\colon \mathbb{D} \subseteq \mathbb{R} \to \mathbb{R}$

function PersistentHomology(f)
  $U \leftarrow \varnothing$ ▷ Initialize an empty union–find structure
  Sort the value tuples in ascending order, such that $y_1 \geq y_2 \geq \ldots$
  for Tuple (x_i, y_i) of f **do**
    if $y_{i-1} > y_i$ and $y_{i+1} > y_i$ **then** ▷ y_i is a local minimum
      U.add(i) ▷ Create a new connected component in U
    else if $y_{i-1} < y_i$ and $y_{i+1} < y_i$ **then** ▷ y_i is a local maximum
      $c \leftarrow$ U.get$(i-1)$ ▷ Get first connected component
      $d \leftarrow$ U.get$(i+1)$ ▷ Get second connected component
      U.merge(c, d) ▷ Merge the two connected components meeting at y_i
    else ▷ y_i is a regular point
      $c \leftarrow$ U.get$(i-1)$ ▷ Get connected component
      $U[c] \leftarrow U[c] \cup i$ ▷ Add y_i to the current connected component
    end if
  end for
  return U
end function

The pairing algorithm

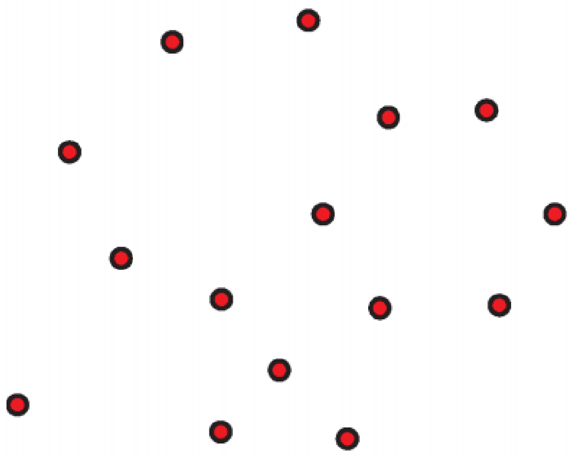
- Input : a discrete sample $P = \{p_1 = (x_1, y_1), \ldots, p_n = (x_n, y_n)\}$ representing a scalar function f.

- A collection of paired points.

1. Initiate an empty UnionFind U.
2. Sort P with respect the y values.
3. For every $p_i = (x_i, y_i)\ in\ P$:
    1. If $y_{i-1} > y_i$ and $y_{i+1} > y_i\ then$:
        1. U.add(i)
        2. Set the *birth* of i to y_i
    2. Else if $y_{i-1} < y_i$ and $y_{i+1} < y_i\ then$:
        1. c=U.get(i-1)
        2. d=U.get(i+1)
        3. U.merge(c,d)
        4. Pair i with c or d (choose the one that was born later)
    3. Else:
        1. c=U.get(i-1)
        2. U(c):=U(c) union i

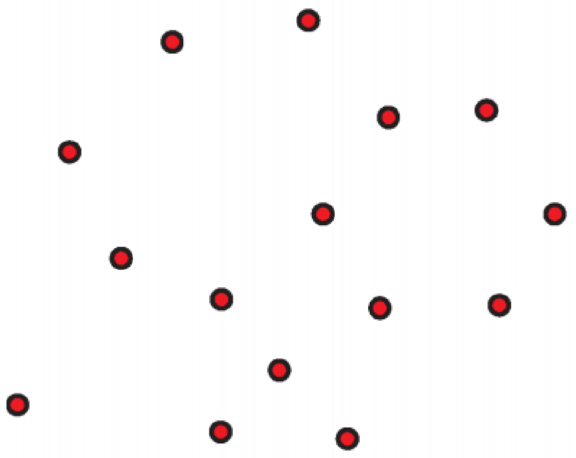
Part II : Point Clouds
Introduction to VR and Cech complexes

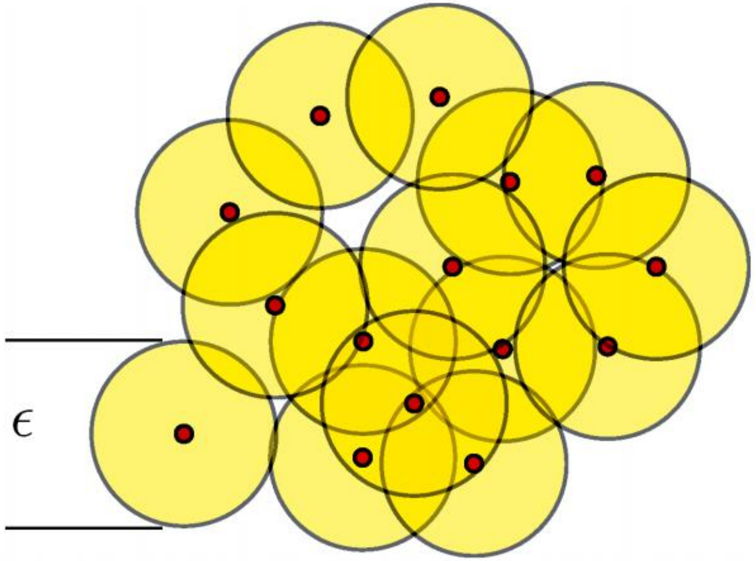
Nerve of a topological space



Given a set of points P sampled from a space X, how can we recover the topological features of the original space X from the point cloud P?

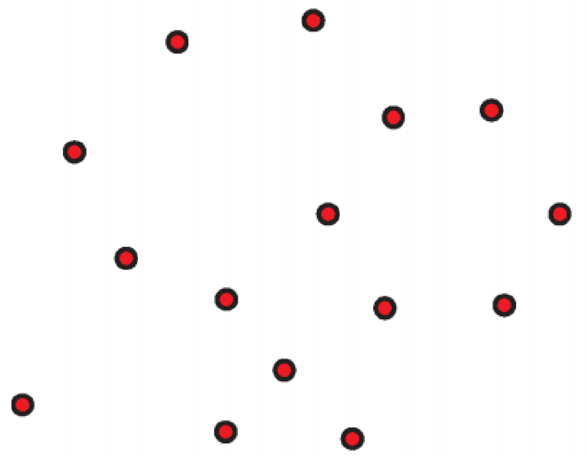
Nerve of a topological space

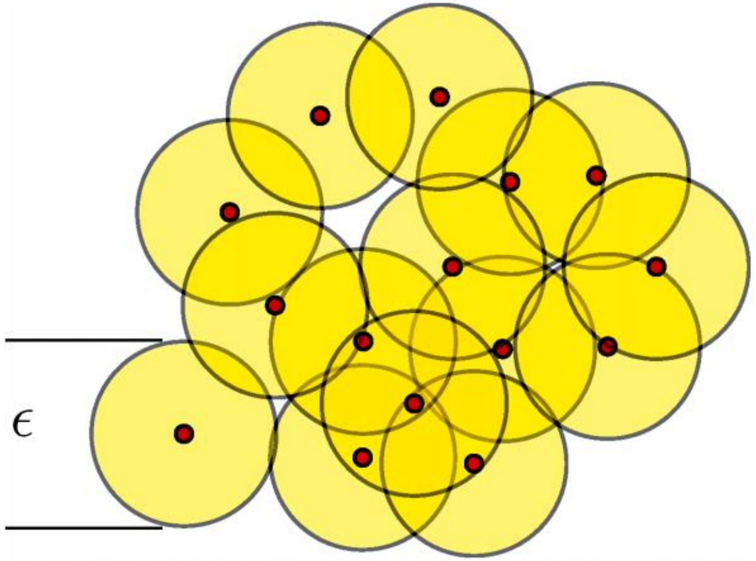


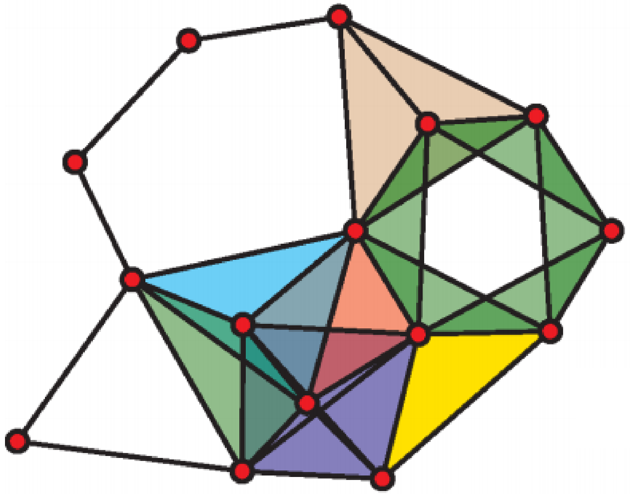


We want a discretized structure that capture the *shape* of the space and we want a reasonable way that is subtle enough to *measure* this shape.

Nerve of a topological space





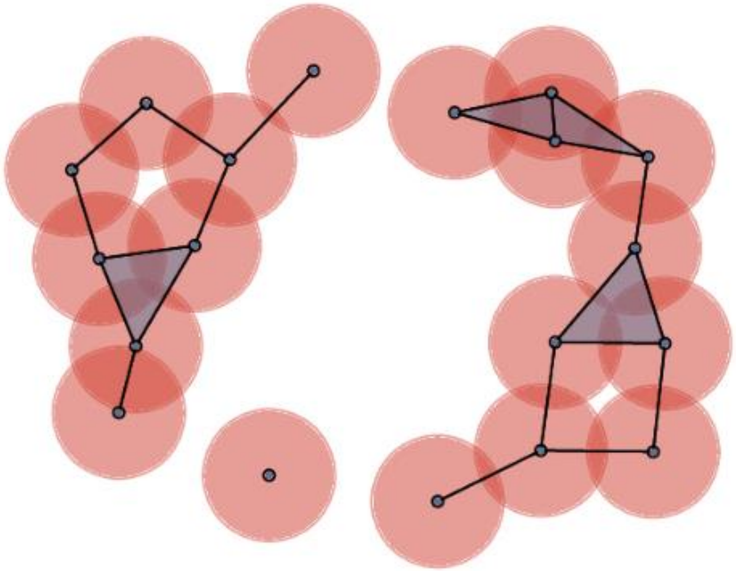


Čech complex

Given a point cloud X in some metric space and a number $\varepsilon > 0$, the Čech complex C_ε
is the simplicial complex whose simplices are constructed as follows :

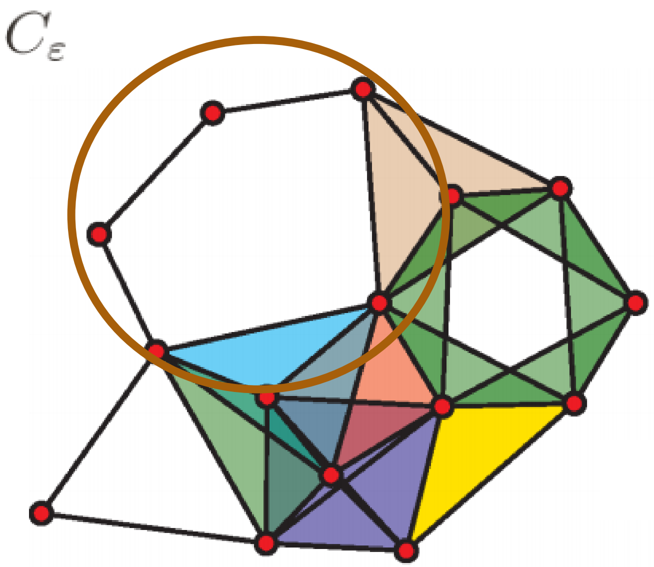
For each subset Y of X, form a $(\varepsilon/2)$ -ball around each point in Y, and include Y as a simplex

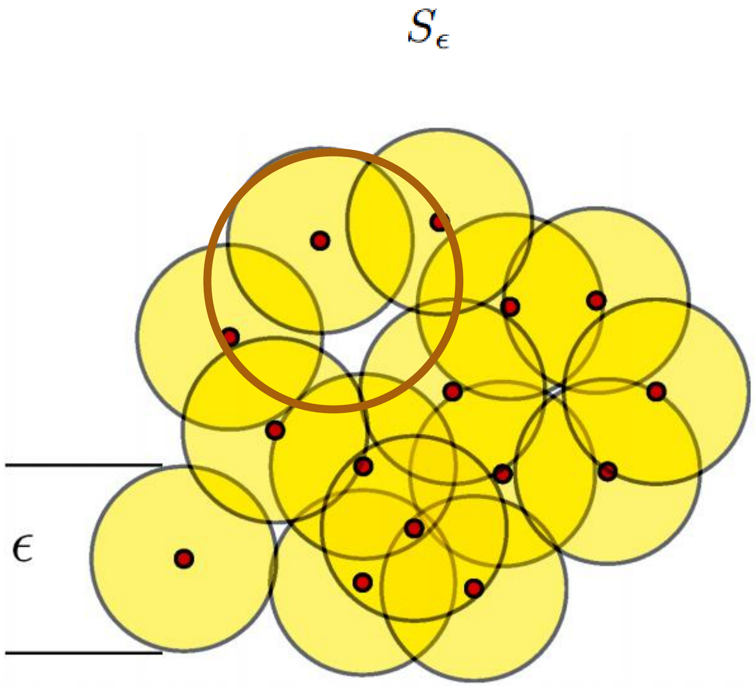
,of dimension |Y|, if there is a common point contained in all of the balls in Y.



The Cech complex approximates the topological space

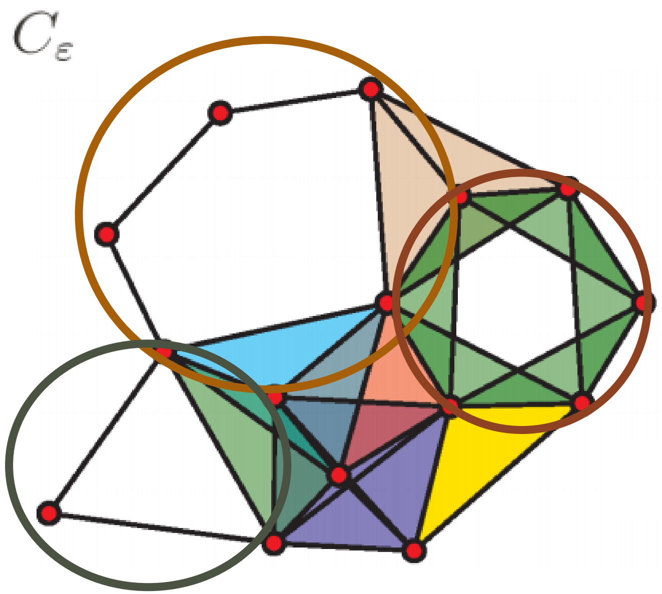
Theorem: The homotopy type of S_ϵ and C_ε are the same.

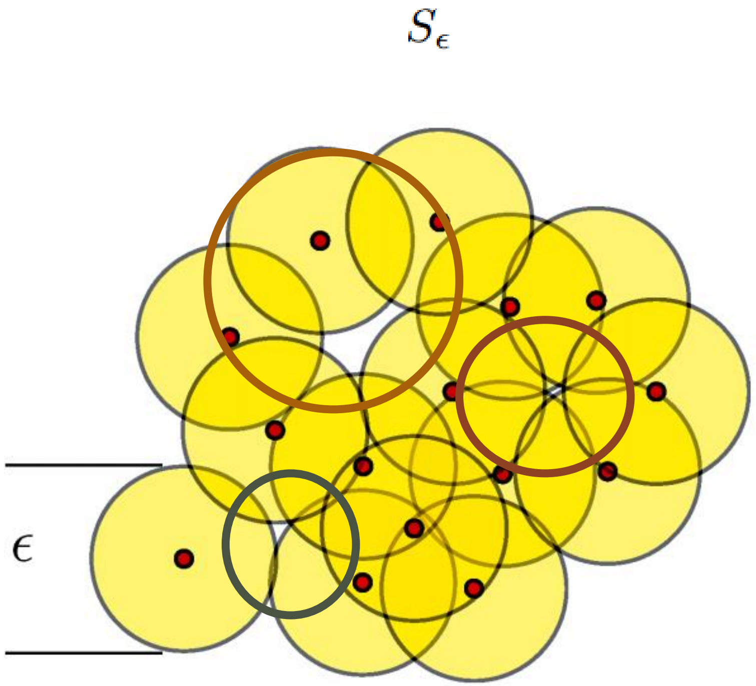




The Cech complex approximates the topological space

Theorem: The homotopy type of S_ϵ and C_ε are the same.





Čech complex size

For each subset Y of X, form a $(\varepsilon/2)$ -ball around each point in Y, and include Y as a simplex

,of dimension |Y|, if there is a common point contained in all of the balls in Y.

What is the computational problem in constructing a Čech complex?

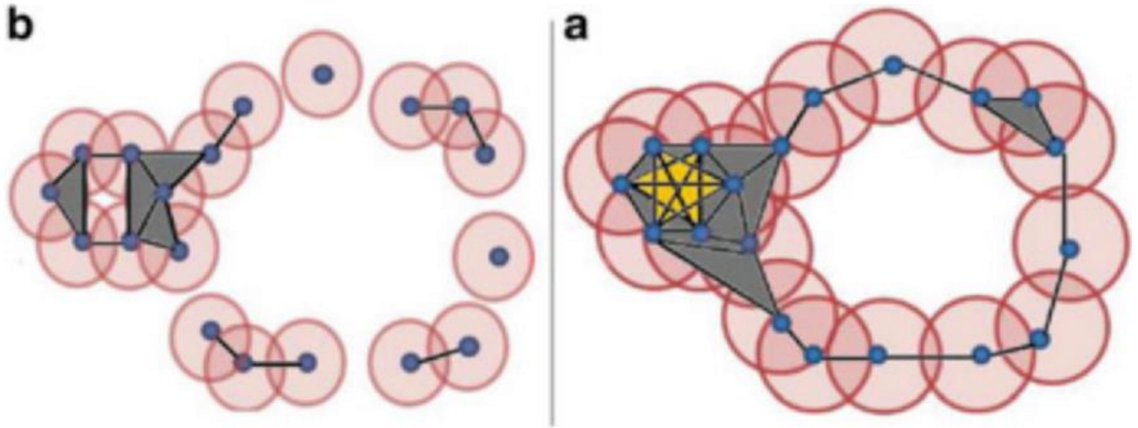
If we have a point cloud set X of size 40 then we have to check all subsets of X of size 40. This is 2^{40}. Very slow!

Vietoris–Rips complex

Let X is a subset of a metric space d and let $\epsilon > 0$. The Vietoris–Rips complex is constructed as follows :

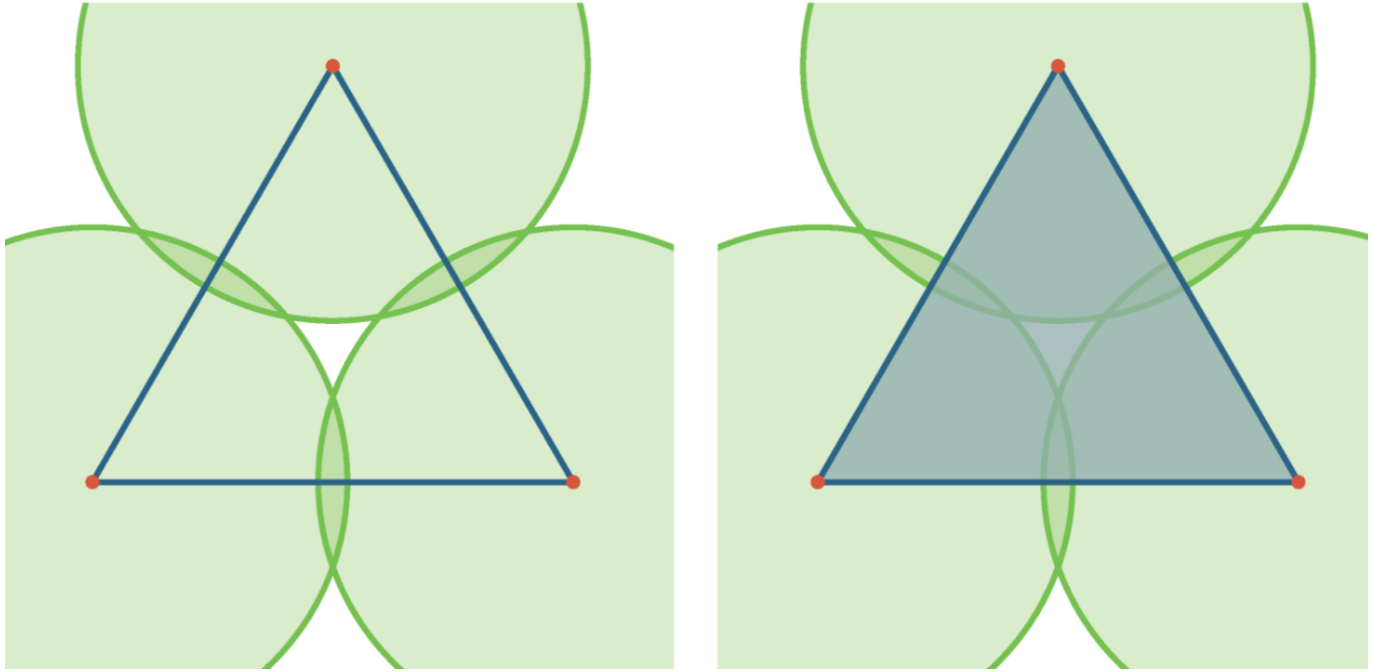
(1) For each point in X, make it as a 0-simplex.
(2) For each pair $x_1, x_2 \in X$, make a 1-simplex $([x_1, x_2])$ if $d(x_1, x_2) \leq \epsilon$.
(3) For $x_1, x_2, \cdots, x_n \in X$, make an $(n-1)$-simplex with vertices $x_1, x_2, \cdots, x_n$. Then, $d(xi, xj) \leq \epsilon$ for all $0 \leq i, j \leq n$; that is, if all the points are within a distance of ϵ from each other.

This complex is denoted by $VR(X, \epsilon)$



Čech complex and VR complex

Comparison between the two complexes :



Note that the VR complex does not necessarily have the same homotopy type of the space of the union of ball.

Čech complex and VR complex

What is the relation between the Čech complex and VR complex ?

Theorem: For all $\varepsilon > 0$, the following inclusions hold

$$C_\varepsilon \subset VR_\varepsilon \subset C_{2\varepsilon}$$

Čech complex and VR complex

What is the relation between the Čech complex and VR complex ?

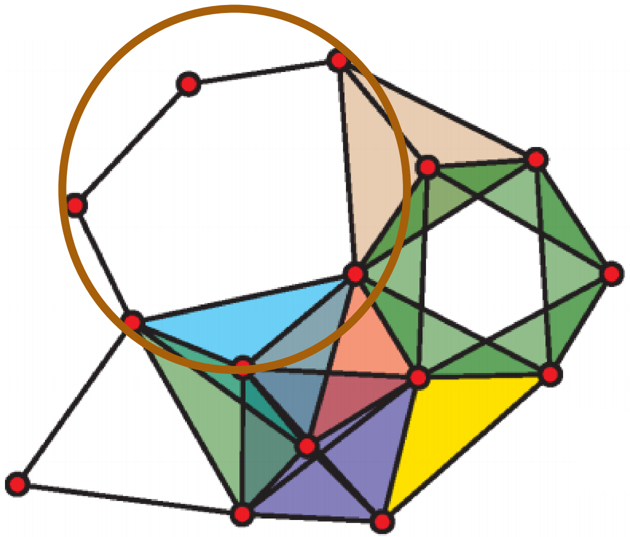
Theorem: For all $\varepsilon > 0$, the following inclusions hold

$$C_\varepsilon \subset VR_\varepsilon \subset C_{2\varepsilon}$$

So the VR complex forms a good approximation of the Čech complex.

Measuring the shape

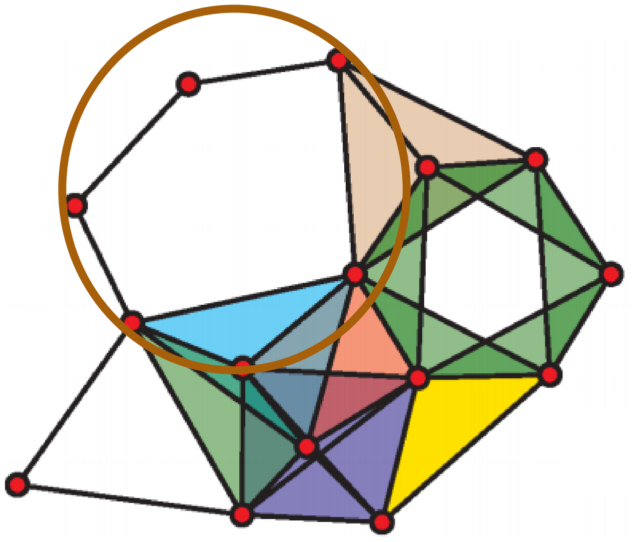
Now that we have a good representation of the space, how do we "measure" it?



Measuring the shape

Now that we have a good representation of the space, how do we "measure" it?

Answering this question can be done using a tool in topology called Homology.

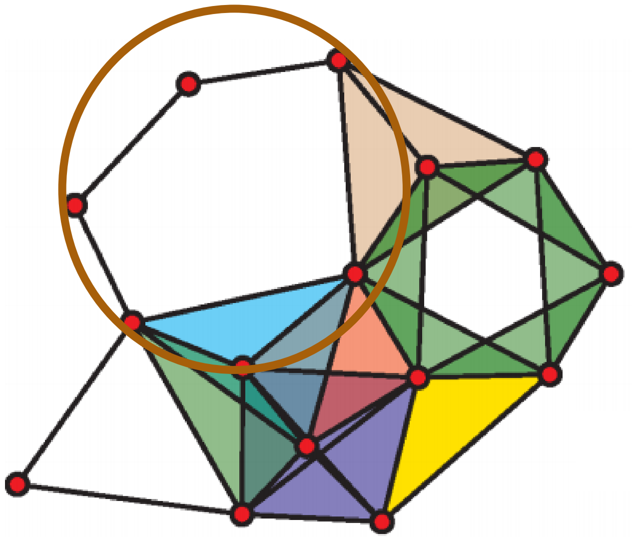


Measuring the shape

Now that we have a good representation of the space, how do we "measure" it?

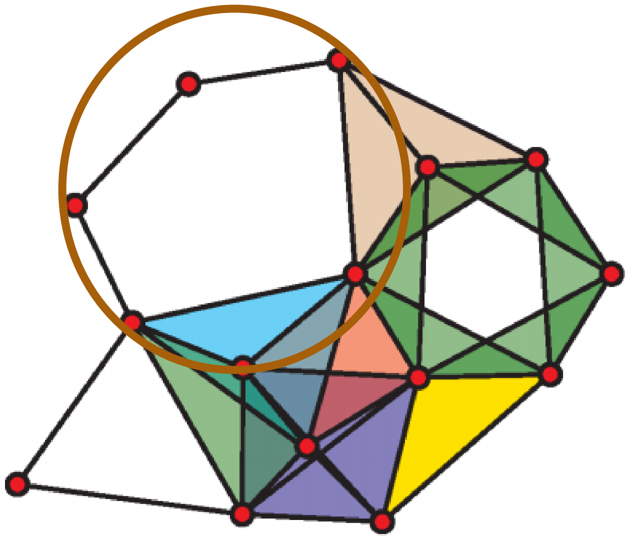
Answering this question can be done using a tool in topology called Homology.

Homology is computable via linear algebra



Measuring the shape

Now that we have a good representation of the space, how do we "measure" it?



Answering this question can be done using a tool in topology called Homology.

Homology is computable via linear algebra

Roughly speaking, homology counts :

- The number of connected components,
- The number of cycles
- The Number o voids in a space

Measuring the shape

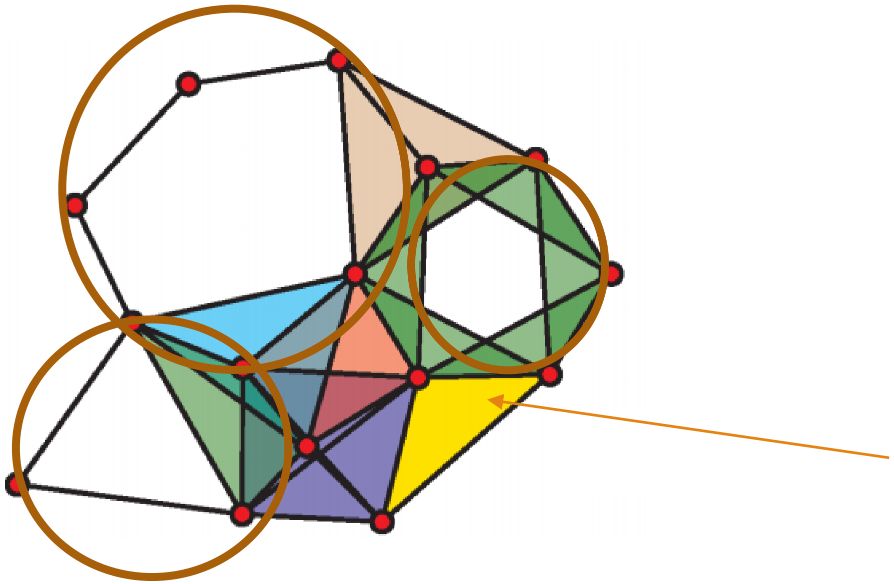
Now that we have a good representation of the space, how do we "measure" it?

Answering this question can be done using a tool in topology called Homology.

Homology is computable via linear algebra

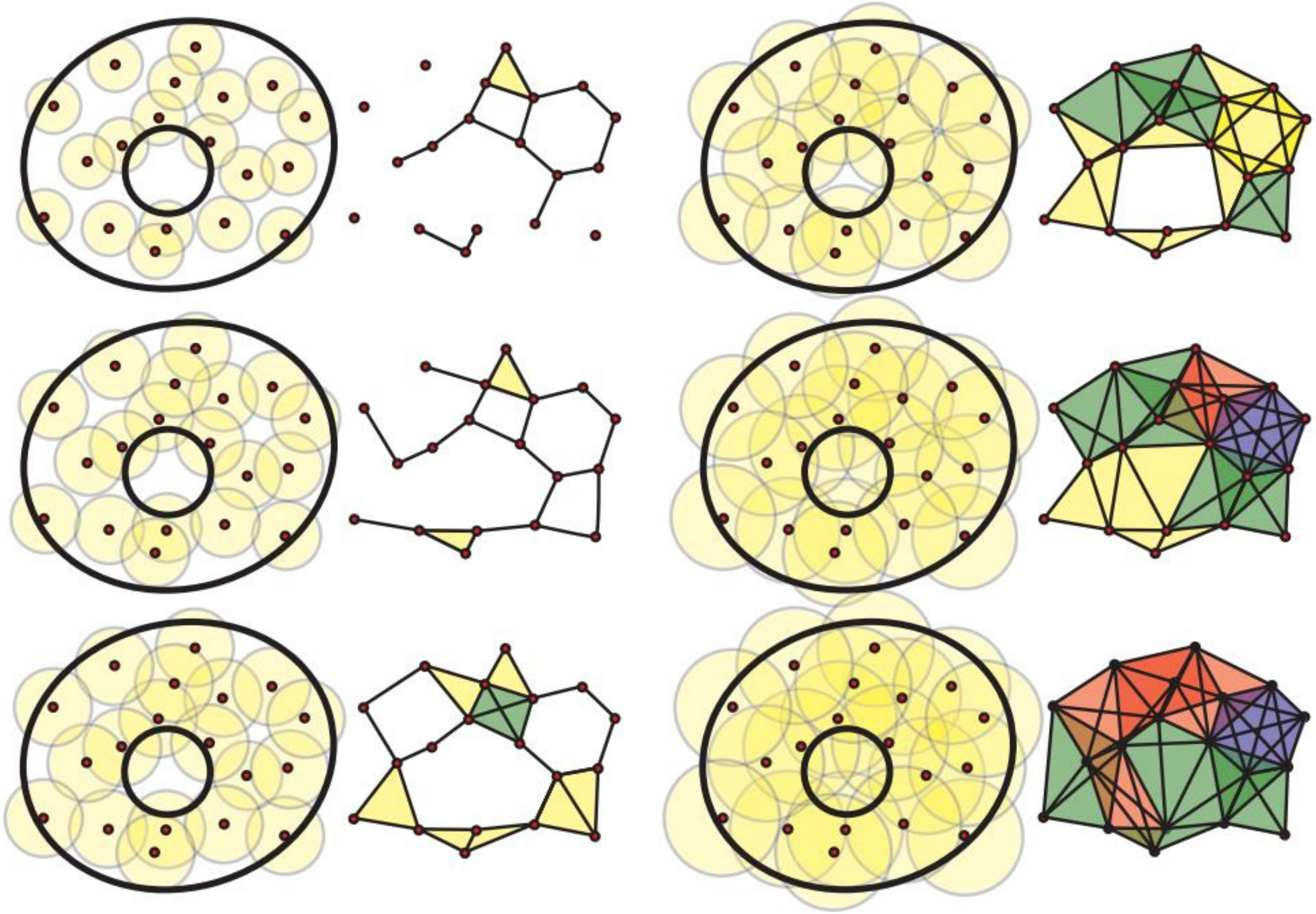
Roughly speaking, homology counts :

- The number of connected components,
- The number of cycles
- The Number o voids in a space



This space here has 1 connected component and 3 cycles

What size do we consider?



We will come back to this question later.

Remarks on computing the Vietoris–Rips complex

Given a point cloud X we want to construct a filtration F using VR construction.

Remarks on computing the Vietoris–Rips complex

Given a point cloud X we want to construct a filtration F using VR construction.

We will utilize the following easy fact :

$$\text{Let } \varepsilon_1 \leq \varepsilon_2 \text{ then } VR(X, \varepsilon_1) \leq VR(X, \varepsilon_2)$$

Remarks on computing the Vietoris–Rips complex

Given a point cloud X we want to construct a filtration F using VR construction.

We will utilize the following easy fact :

$$\text{Let } \varepsilon_1 \leq \varepsilon_2 \text{ then } VR(X, \varepsilon_1) \leq VR(X, \varepsilon_2)$$

This allows us in practice to compute the VR complex for some maximum scale a ∈ R and then extract the complex at any lower scale b less than a.

Remarks on computing the Vietoris–Rips complex

Given a point cloud X we want to construct a filtration F using VR construction.

We will utilize the following easy fact :

$$\text{Let } \varepsilon_1 \leq \varepsilon_2 \text{ then } VR(X, \varepsilon_1) \leq VR(X, \varepsilon_2)$$

This allows us in practice to compute the VR complex for some maximum scale a ∈ R and then extract the complex at any lower scale b less than a.

Given a VR(X,a), suppose that we want to compute VR(X,b) for b less than a.
How do we compute determine the simplices from VR(X,a) that belongs to VR(X,b)?

Remarks on computing the Vietoris–Rips complex

Given a complex $VR(X,\varepsilon)$ define the weight function $w : VR(X,\varepsilon) \to \mathbb{R}$

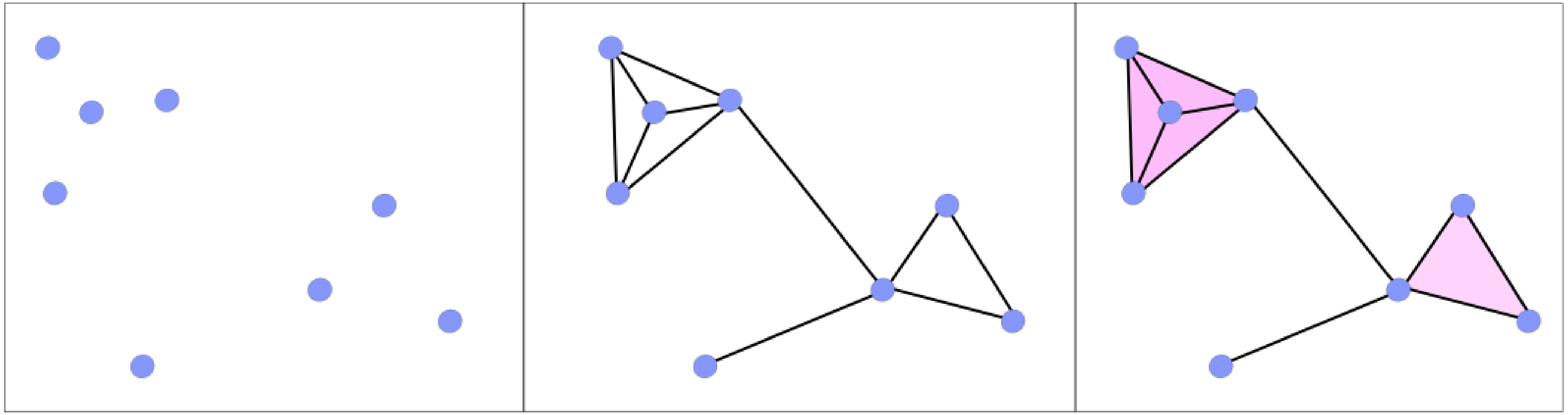
$$\omega(\sigma) = \begin{cases} 0, & \dim(\sigma) \leq 0, \\ \mathrm{d}(u,v), & \sigma = \{u,v\} \\ \max_{\tau \subset \sigma} \omega(\tau), & \text{otherwise.} \end{cases}$$

That is, the weight $\omega(\sigma)$ is equal to the maximum of the weights (lengths) of all its edges.

After defining the weight function on $VR(X,\varepsilon_2)$ we sort the simplices according to their weights, extracting the VR complex for any $\varepsilon_1 \leq \varepsilon_2$ as a prefix of this ordering.

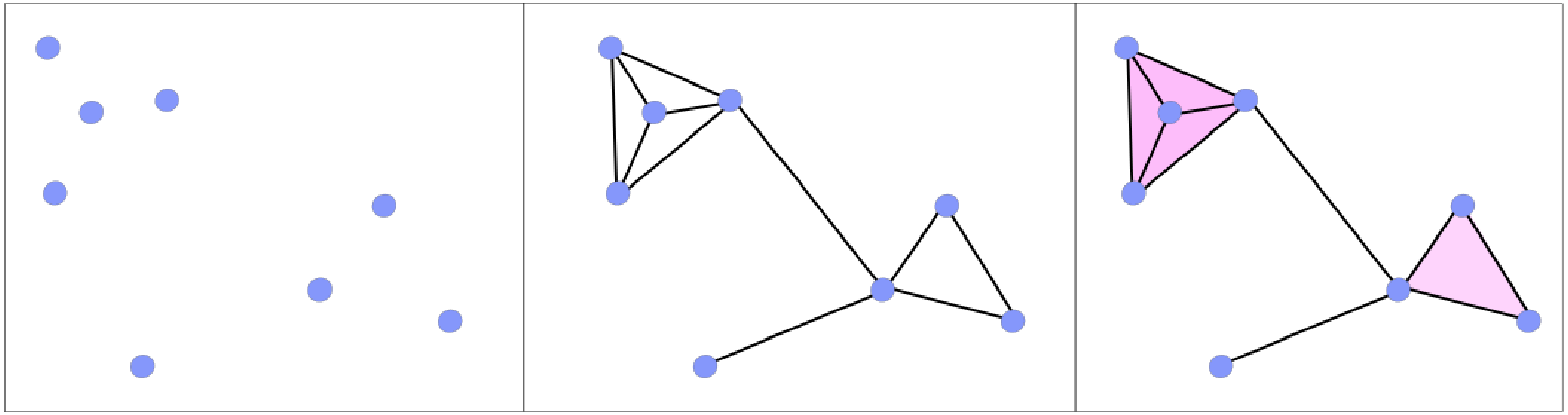
This gives a filtration of $VR(X,\varepsilon_2)$

The relation between neighborhood graph and the Vietoris–Rips complex



The data (left) has the **ϵ-**neighborhood graph (middle).
This is precisely the VR complex (right) at that same resolution.

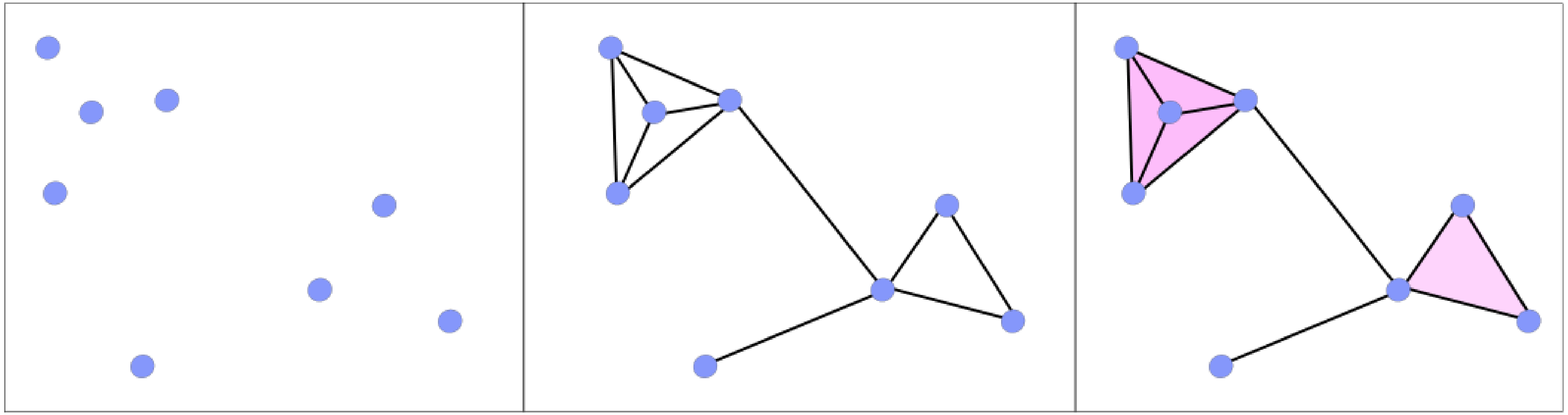
The relation between neighborhood graph and the Vietoris–Rips complex



The data (left) has the **ϵ**-neighborhood graph (middle).
This is precisely the VR complex (right) at that same resolution.

Question: given the **ϵ**-neighborhood (middle), how can we recover the VR complex from it (right) ?

The relation between neighborhood graph and the Vietoris–Rips complex



The data (left) has the **ϵ**-neighborhood graph (middle).
This is precisely the VR complex (right) at that same resolution.

Question: given the **ϵ**-neighborhood (middle), how can we recover the VR complex from it (right) ?

Answer: Higher dimensional simplices recovered from the *cliques* of the **ϵ**-neighborhood graph.