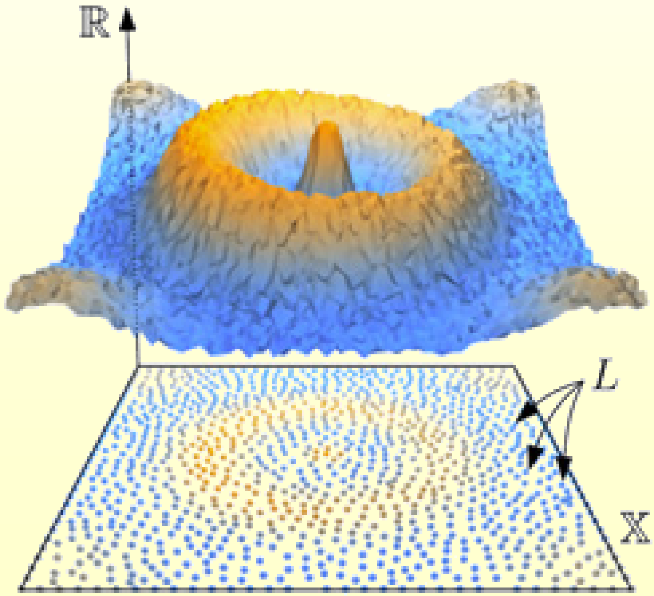
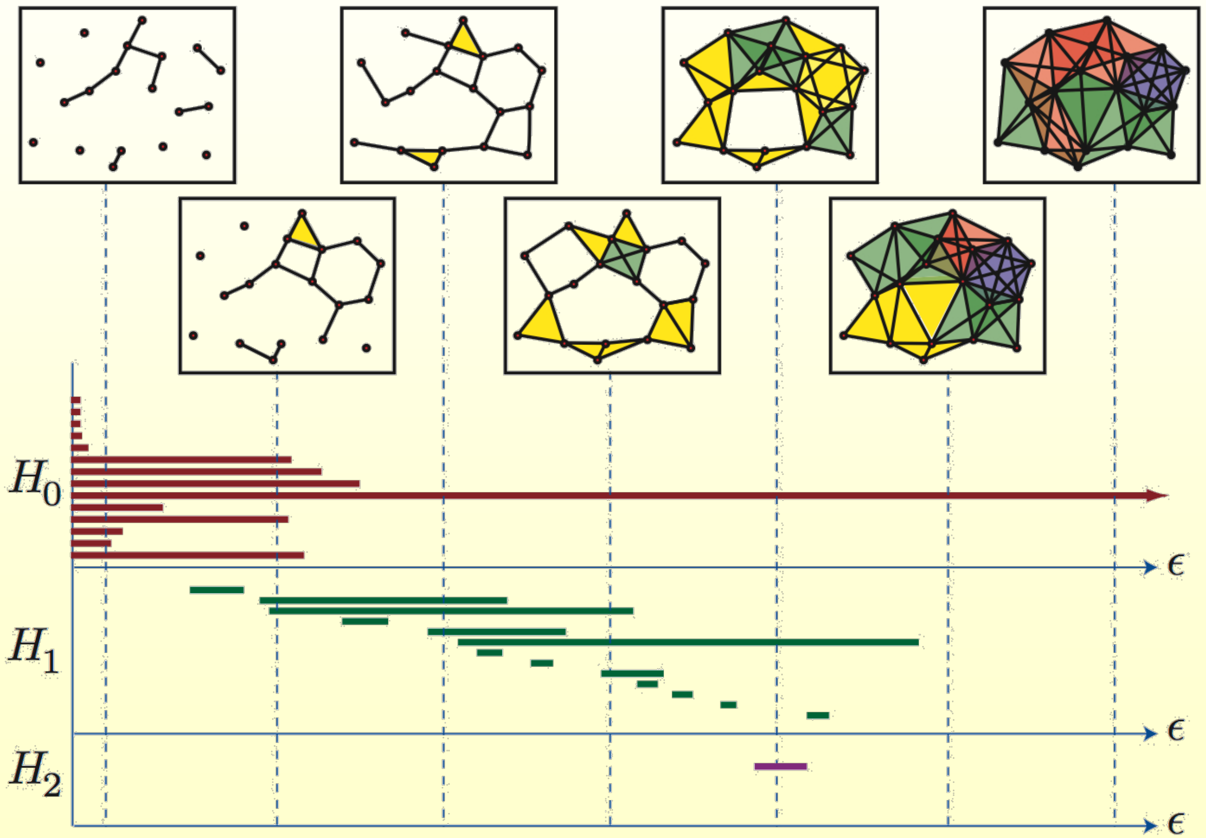
CS268: Computational Topology and Topological Data Analysis, II

Leonidas J. Guibas



Persistent Homology



Slides ack: Afra Zomorodian, Ryan Lewis, Fred Chazal, Robert Ghrist

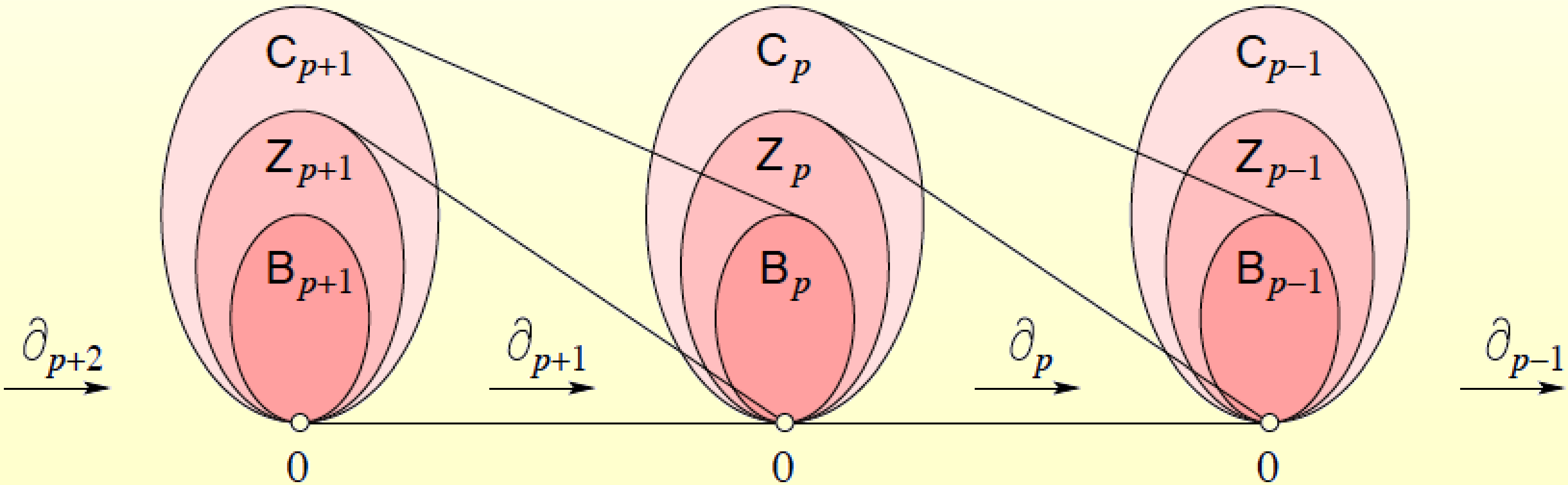
Homology

Homology

- The kth homology group is

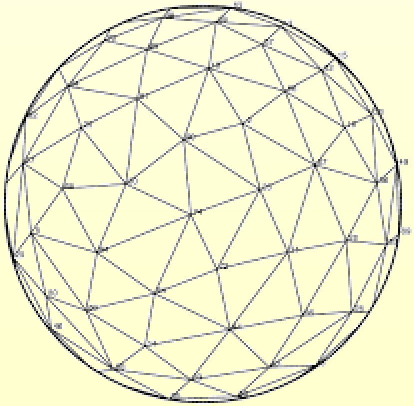
$$\mathsf{H}_k = \mathsf{Z}_k / \mathsf{B}_k = \ker \partial_k / \operatorname{im} \partial_{k+1}.$$

- Compute a basis for $\ker \partial_k$
- Compute a basis for $\operatorname{im} \partial_{k+1}$



Homology of 2-Manifolds

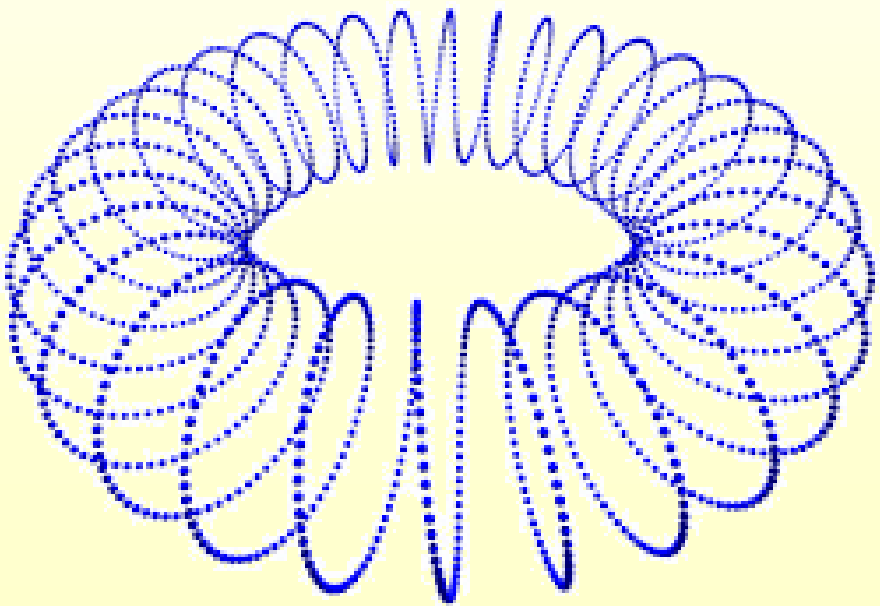
2-manifold	H_0	H_1	H_2
sphere	$\mathbb{Z}$	$\{0\}$	$\mathbb{Z}$
torus	$\mathbb{Z}$	$\mathbb{Z} \times \mathbb{Z}$	$\mathbb{Z}$
projective plane	$\mathbb{Z}$	$\mathbb{Z}_2$	$\{0\}$
Klein bottle	$\mathbb{Z}$	$\mathbb{Z} \times \mathbb{Z}_2$	$\{0\}$



Computing Homology via Bases

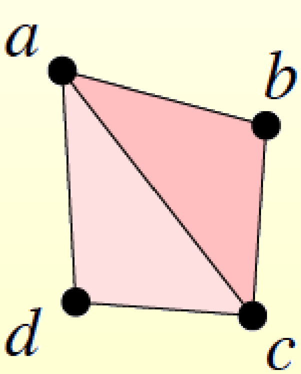
Computational Topology Software

- JavaPlex (Henry Adams) -- has very nice tutorial
- Dionysus (Dmitriy Morozov)
- PHAT (Michael Kerber)



Matrix Representation of ∂

- Boundary homomorphism is linear, so it has a matrix
- $\partial_k : \mathsf{C}_k \rightarrow \mathsf{C}_{k-1}$
- Use oriented simplices as bases for domain and codomain!
- M_k is the standard matrix representation for ∂_k



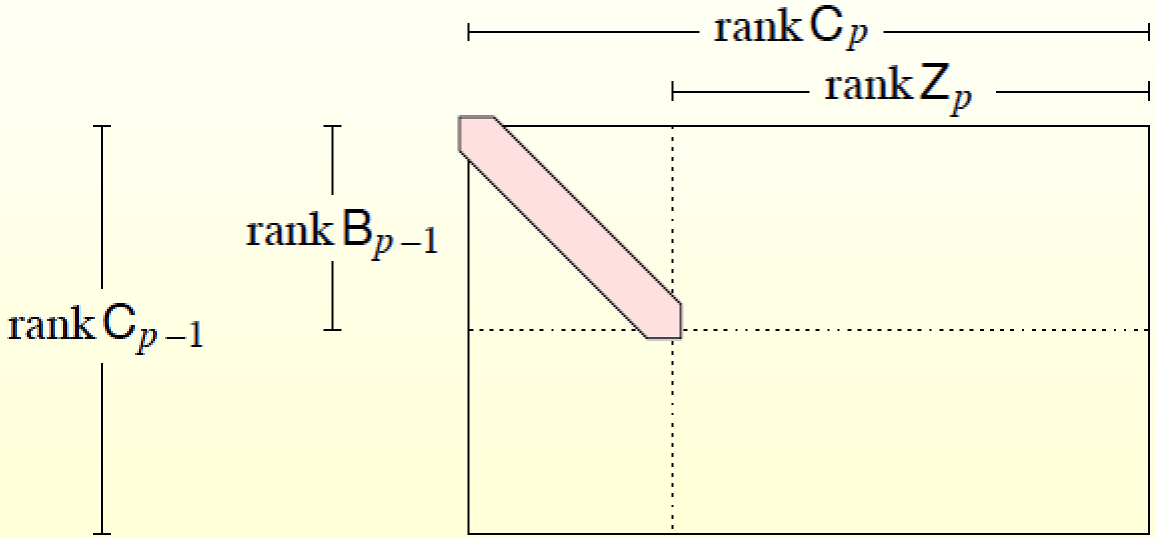
$$M_1 = \begin{array}{cc} & \begin{array}{ccccc} ab & bc & cd & ad & ac \end{array} \\ \begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} -1 & 0 & 0 & -1 & -1 \\ 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

[Two glued triangles, not the tetrahedron …]

Elementary Matrix Operations

- The elementary row operations on M_k are
  1. exchange row i and row j,
  2. multiply row i by -1,
  3. replace row i by (row i) $+ q$(row j), where q is an integer and $j \neq i$.
- Similar elementary column operations on columns
- Effect: change of bases

Smith Normal Form



Introduce columns from let to right
Keep doing Gaussian elimination steps …
For a complex with m simplices, this can take $O(m^3)$ operations

Reduction Algorithm

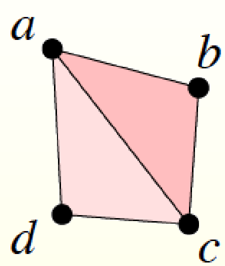
- Like Gaussian elimination, we keep changing the basis to get to the (Smith) normal form:

$$\tilde{M}_k = \left[\begin{array}{ccc|c} b_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & b_{l_k} & \\ \hline & 0 & & 0 \end{array}\right]$$

- $l_k = \operatorname{rank} M_k = \operatorname{rank} \tilde{M}_k, b^i \geq 1$
- $b_i | b_{i+1}$ for all $1 \leq i < l_k$

$b_i = 1 \quad \forall i$, if no torsion

Reduction Example



$$\tilde{M}_1 = \left[\begin{array}{c|ccccc} & cd & bc & ab & z_1 & z_2 \\ \hline d-c & 1 & 0 & 0 & 0 & 0 \\ c-b & 0 & 1 & 0 & 0 & 0 \\ b-a & 0 & 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

- $z_1 = ad - bc - cd - ab$ and $z_2 = ac - bc - ab$ form a basis for Z_1
- $\{d - c, c - b, b - a\}$ is a basis for B_0

Reduction Example

$$M_2 \quad = \quad \begin{array}{c|c} & \begin{array}{cc} abc & acd \end{array} \\ \hline \begin{array}{c} ac \\ ad \\ cd \\ bc \\ ab \end{array} & \begin{bmatrix} -1 & 1 \\ 0 & -1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \end{array}$$

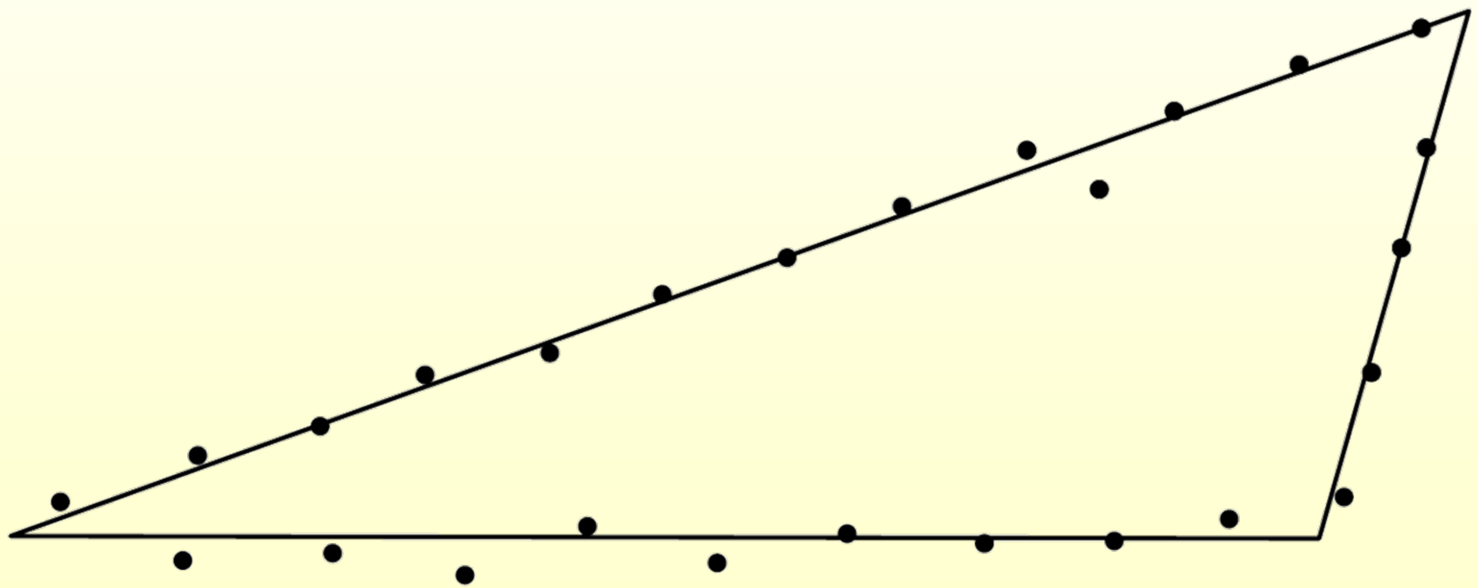
$$\tilde{M}_2 \quad = \quad \begin{array}{c|c} & \begin{array}{cc} -abc & -acd + abc \end{array} \\ \hline \begin{array}{c} ac - bc - ab \\ ad - cd - bc - ab \\ cd \\ bc \\ ab \end{array} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \end{array}$$

Persistent Homology

Filtrations

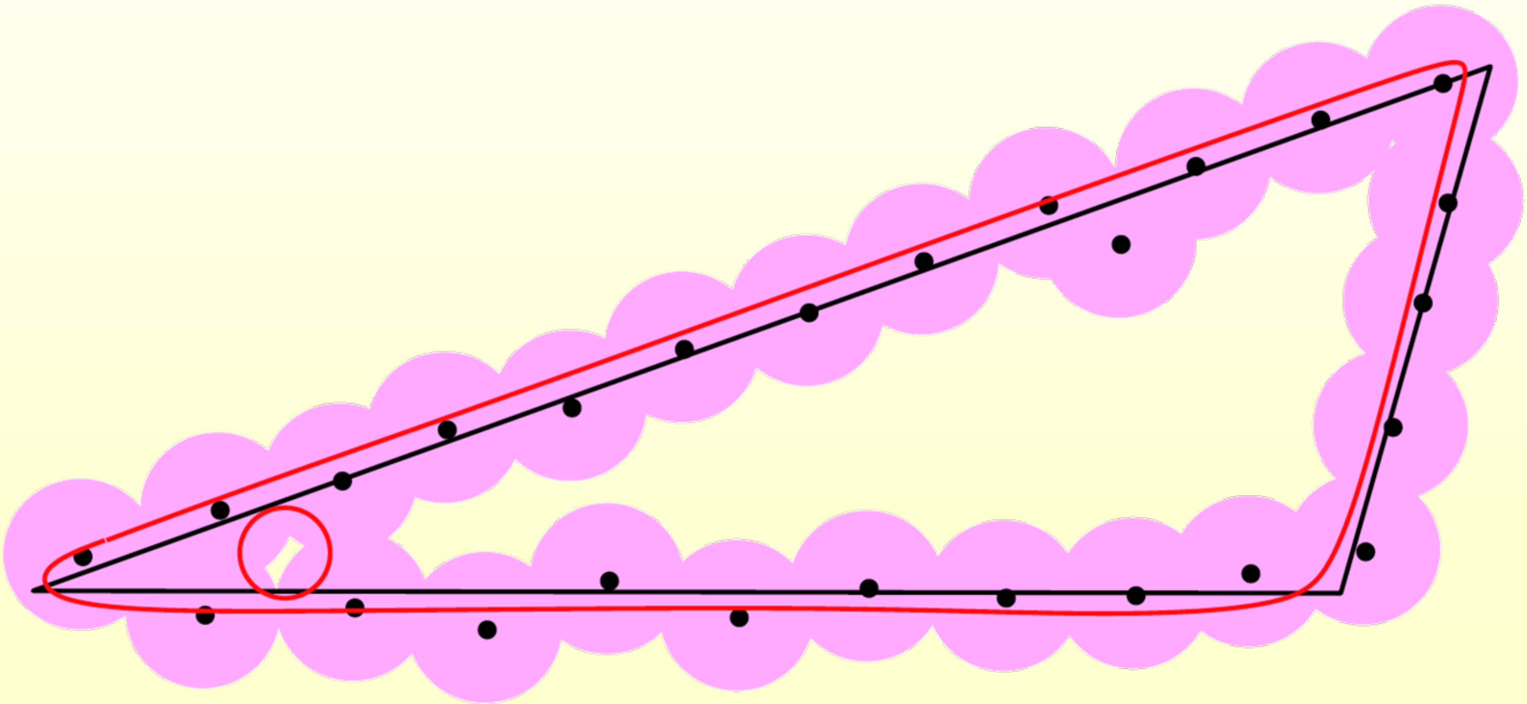
How to Choose ϵ?

- How to determine the topology of the underlying space from a point cloud approximation?



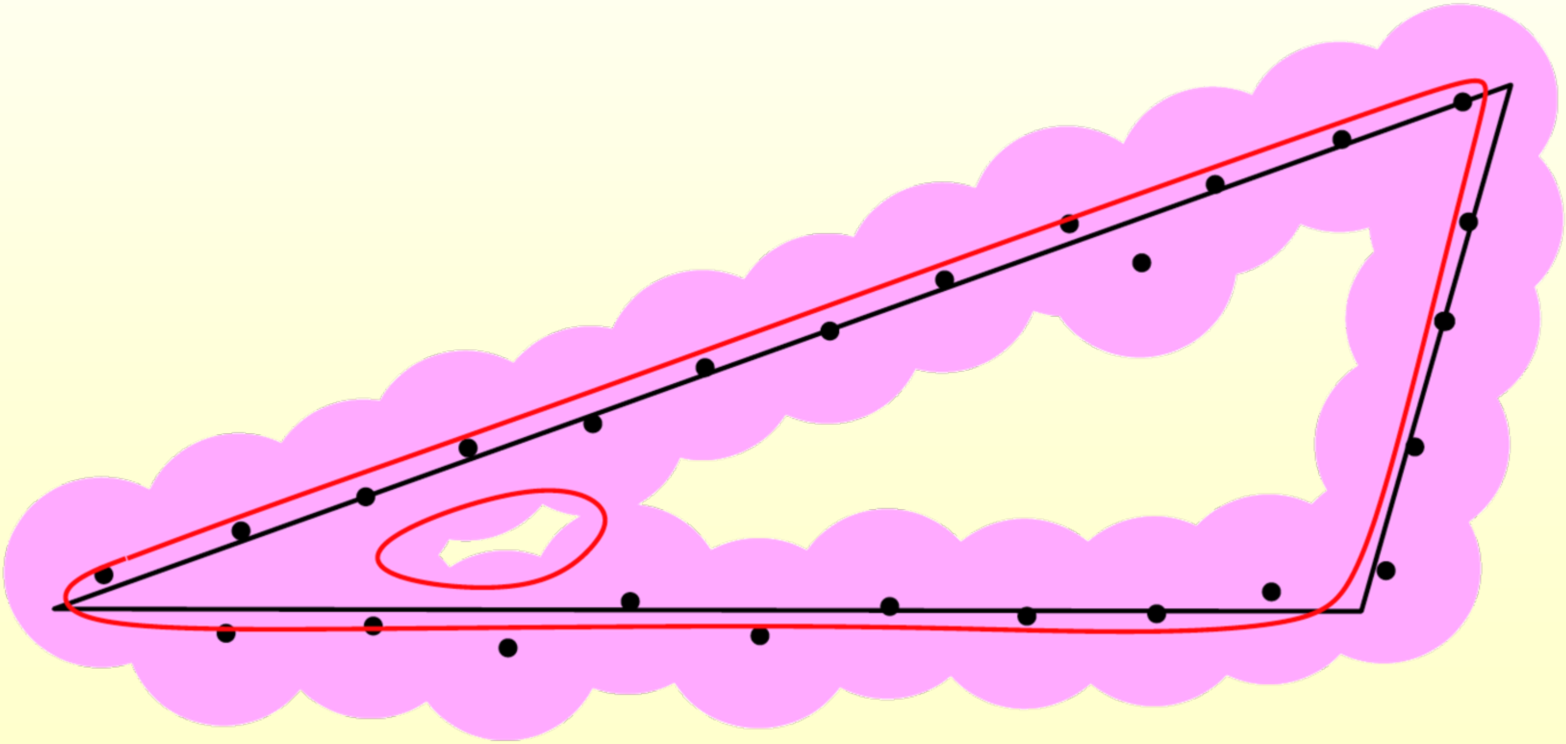
How to Choose ϵ?

- How to determine the topology of the underlying space from a point cloud approximation?



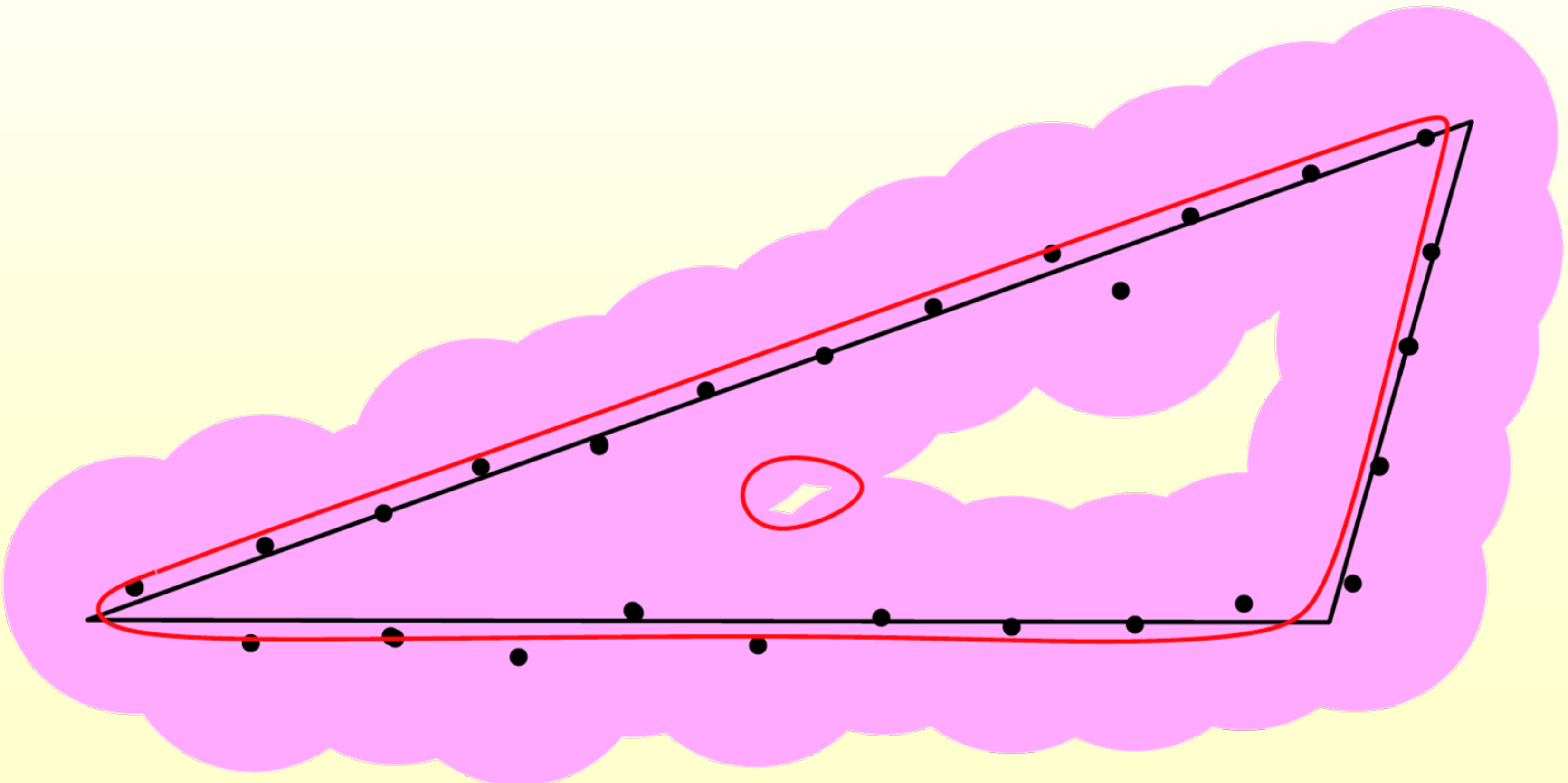
How to Choose ϵ?

- How to determine the topology of the underlying space from a point cloud approximation?

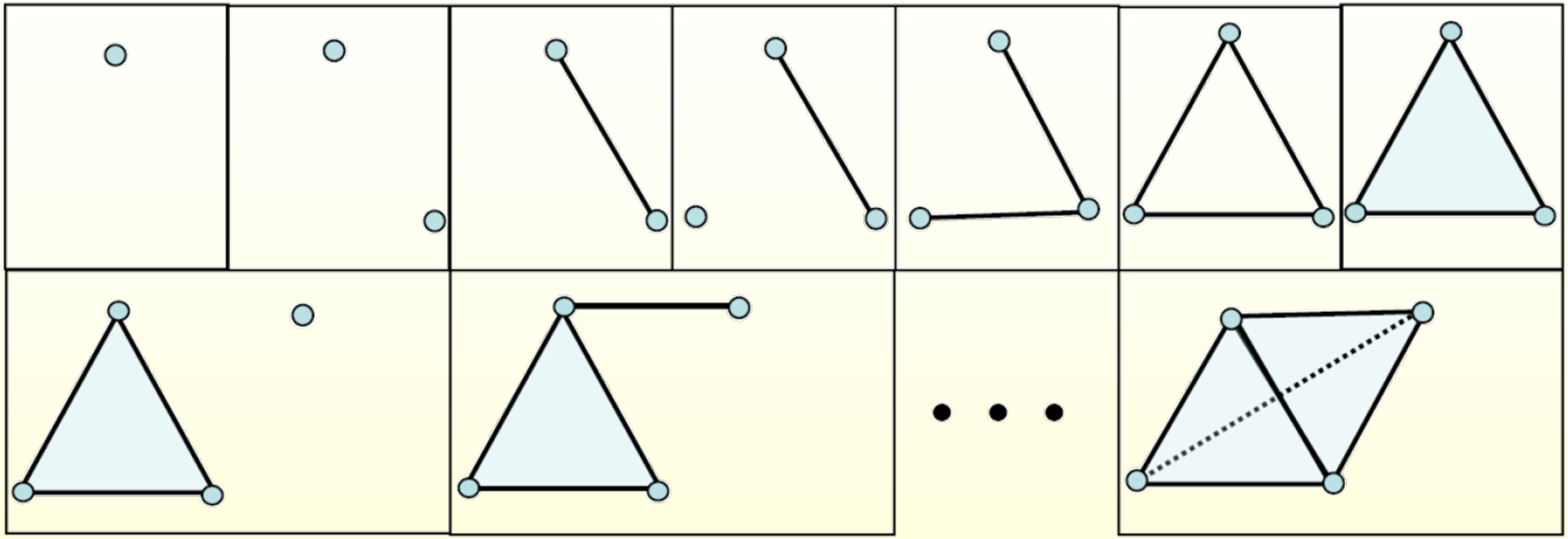


How to Choose ϵ?

- How to determine the topology of the underlying space from a point cloud approximation?



Filtrations



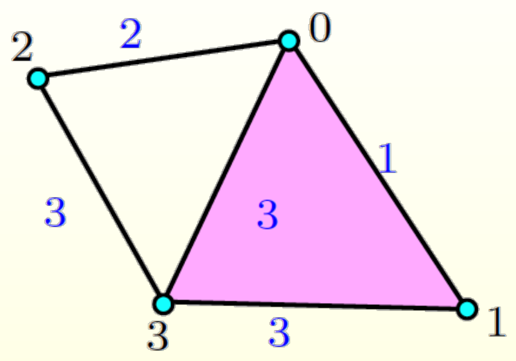
A filtration of a (finite) simplicial complex K is a sequence of subcomplexes such that

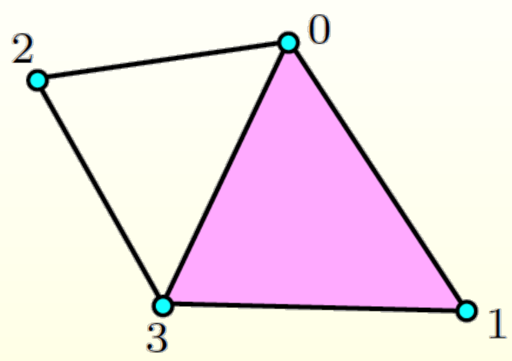
i) $\emptyset = K^0 \subset K^1 \subset \cdots \subset K^m = K$,

ii) $K^{i+1} = K^i \cup \sigma^{i+1}$ where σ^{i+1} is a simplex of K.

Sub-simplices of a simplex must be added before the simplex!

The Sub-Level Set Filtration

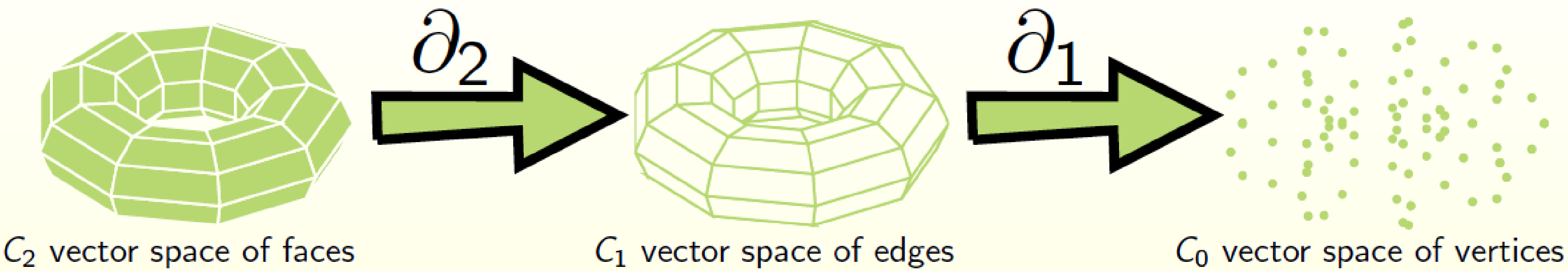




- f a real valued function defined on the vertices of K
- For $\sigma = [v_0, \cdots, v_k] \in K$, $f(\sigma) = \max_{i=0,\cdots,k} f(v_i)$
- The simplices of K are ordered according increasing f values (and dimension in case of equal values on different simplices).

Persistent Homology: Do not choose an ϵ!

Standard Homology



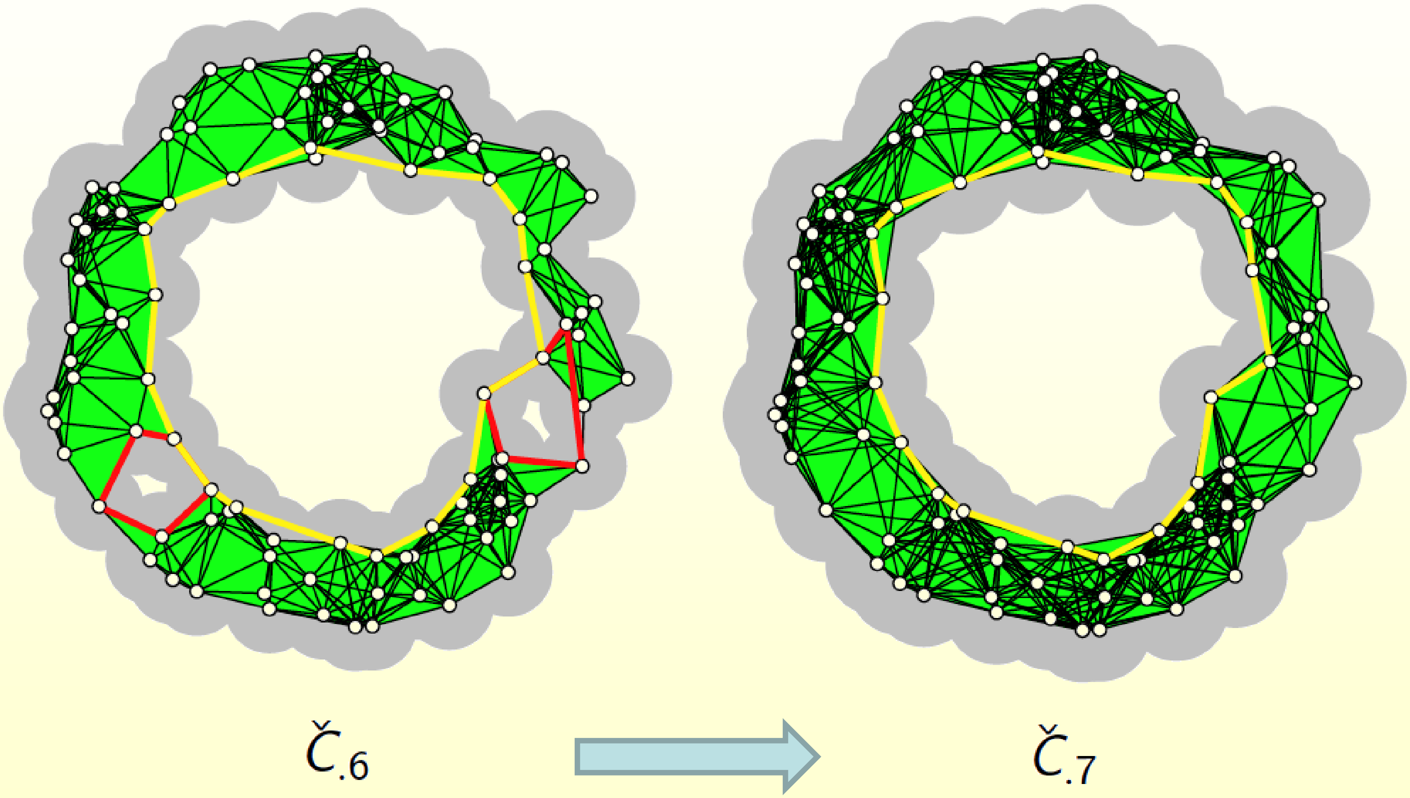
Take the linear extension of the boundary operator:

$$\partial_d([v_0, \dots v_d]) = \sum_{i=0}^{d} (-1)^i [v_0 \dots, \hat{v}_i, \dots v_d]$$

Fact: $\partial_{d-1} \circ \partial_d \equiv 0 \Rightarrow \operatorname{Im} \partial_d \subseteq \ker \partial_{d-1}$

Definition: $H_d(K) = \ker \partial_d / \operatorname{Im} \partial_{d+1}$

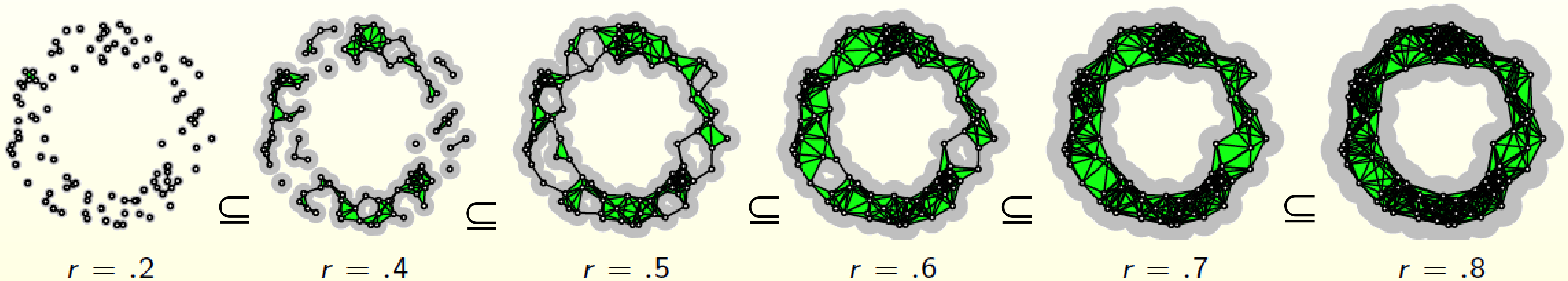
We Can Track Topological Features in a Filtration



Functorality allows us to systematically track holes over time!

The inclusion map among the complexes translates to a homomorphism between the homology groups

Persistent Homology is Functorial Homology

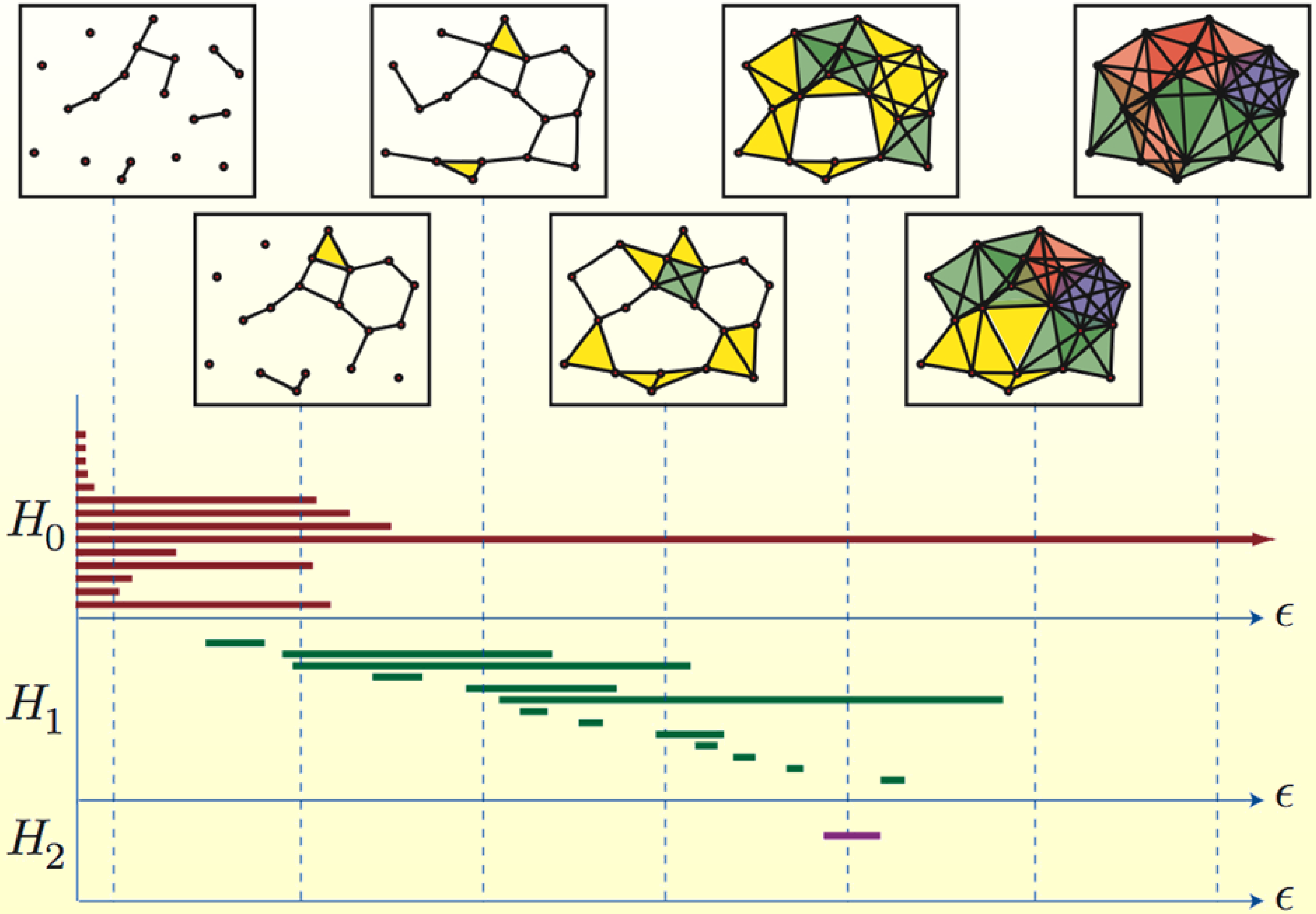


$$H_d(\check{C}_*) = \bigoplus_{\epsilon} H_d(\check{C}_\epsilon)$$

Homology of the entire filtration

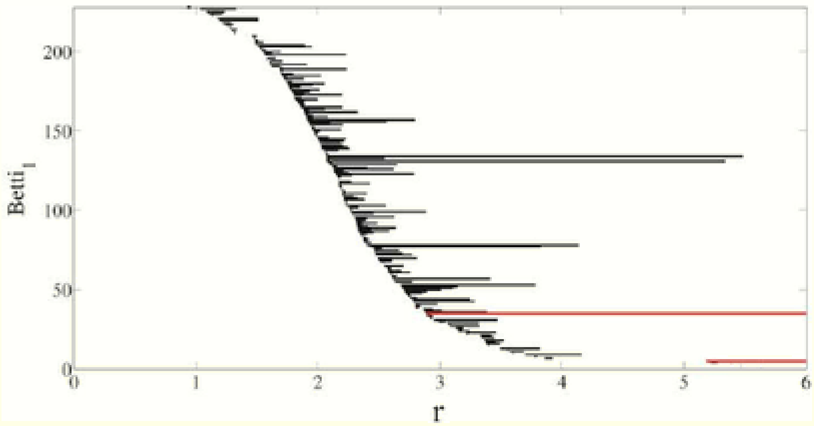
Homomorphisms at the homology level allow us
to track homology classes – i.e., topological features

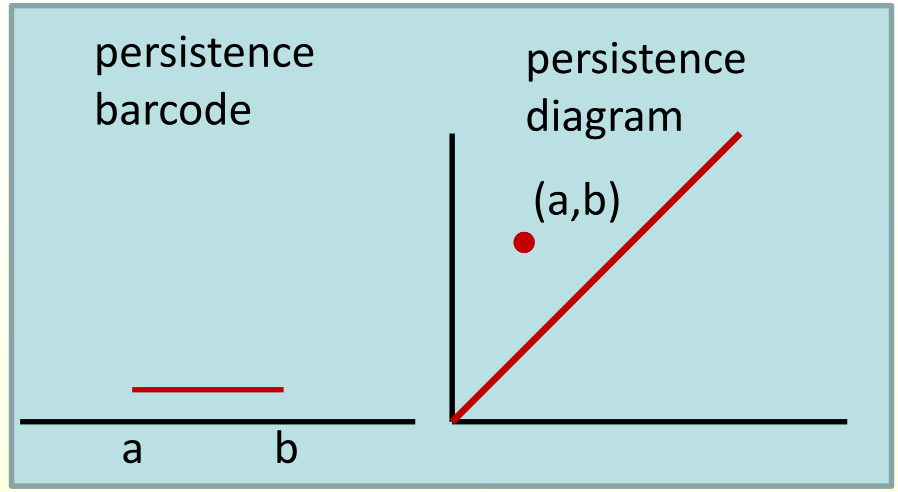
Barcodes are the Lifetimes of Topological Features



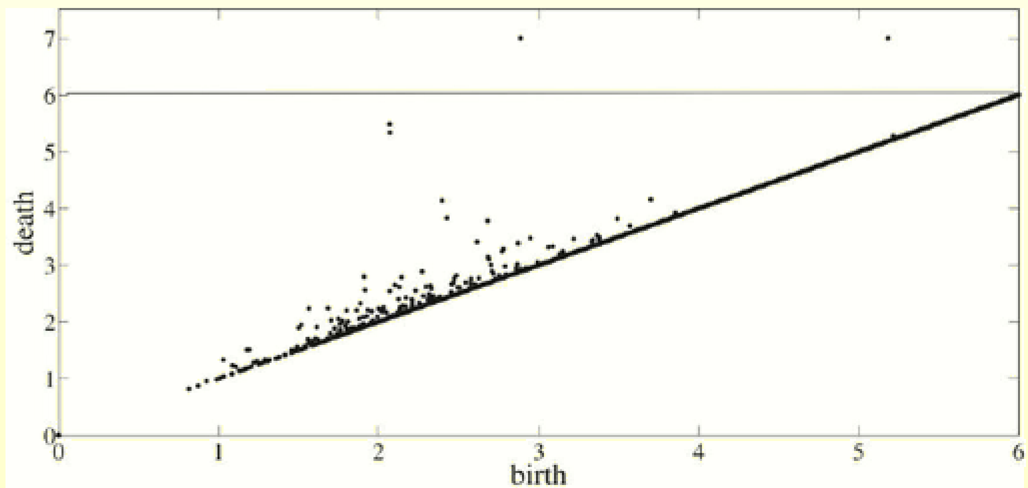
Barcodes are the output of persistent homology

Another View: Persistence Diagrams





long barcodes = points away from the diagonal = robust features

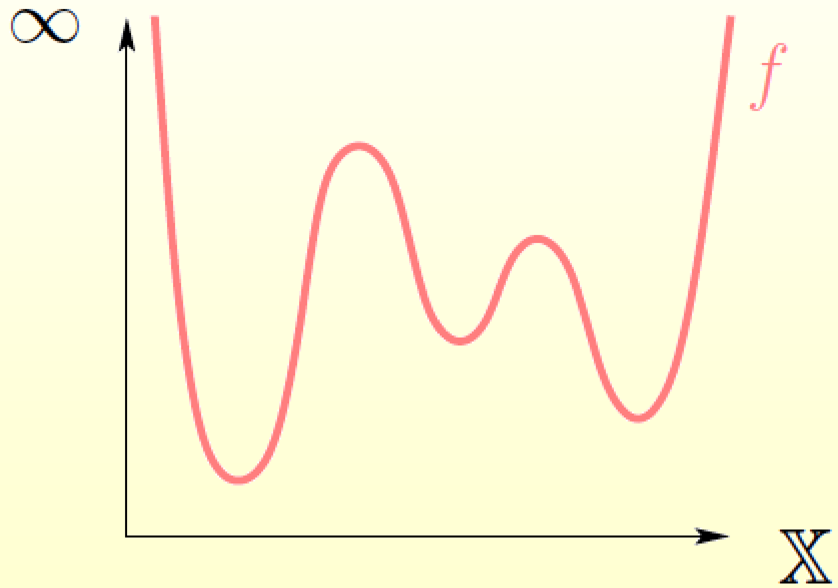


Short barcodes = points near the diagonal = noise

Map 1-D intervals to points in 2-D

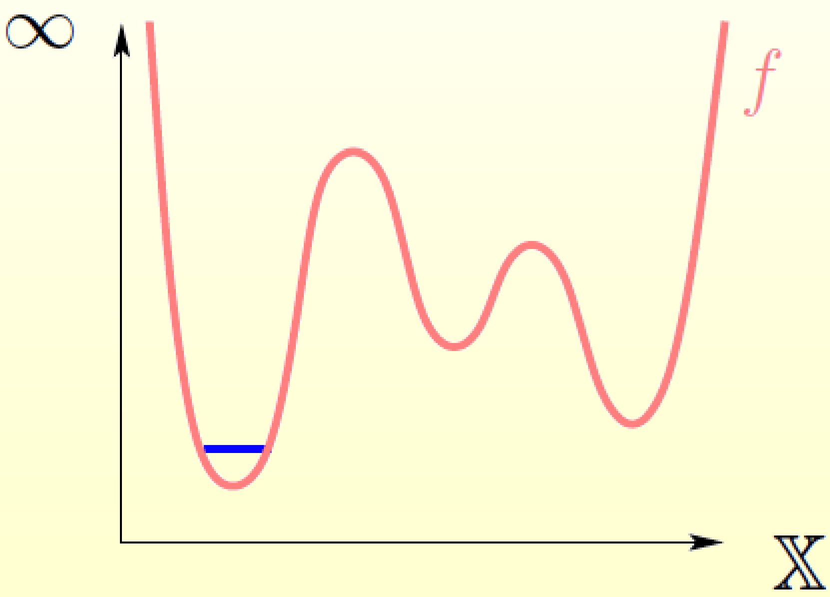
Persistence Provides a Pairing: Birth and Death of a Top Feature

- Sublevel sets of a function example



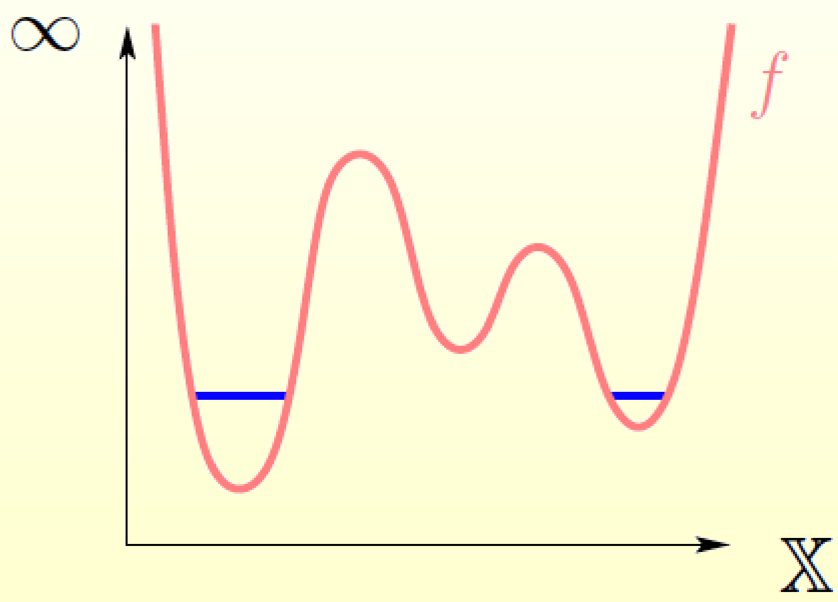
Persistence Provides a Pairing

- Sublevel sets of a function example



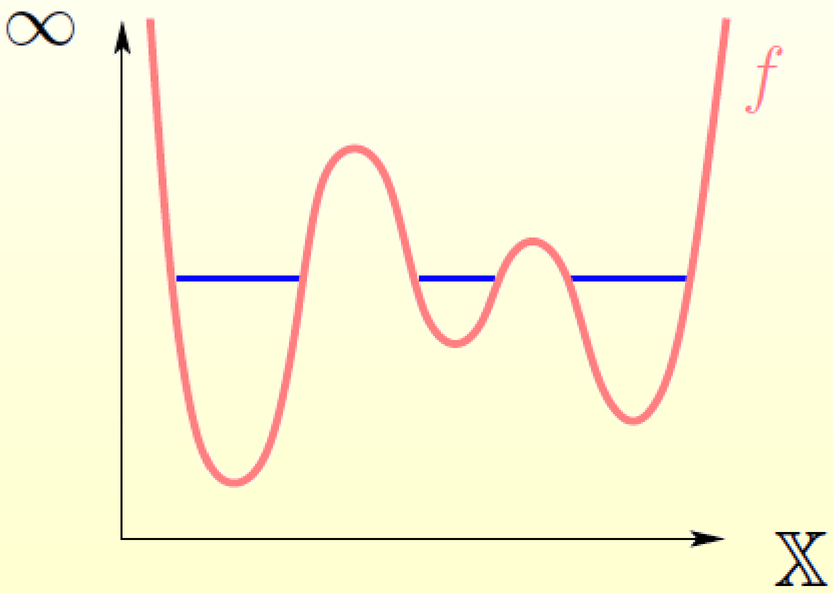
Persistence Provides a Pairing

- Sublevel sets of a function example



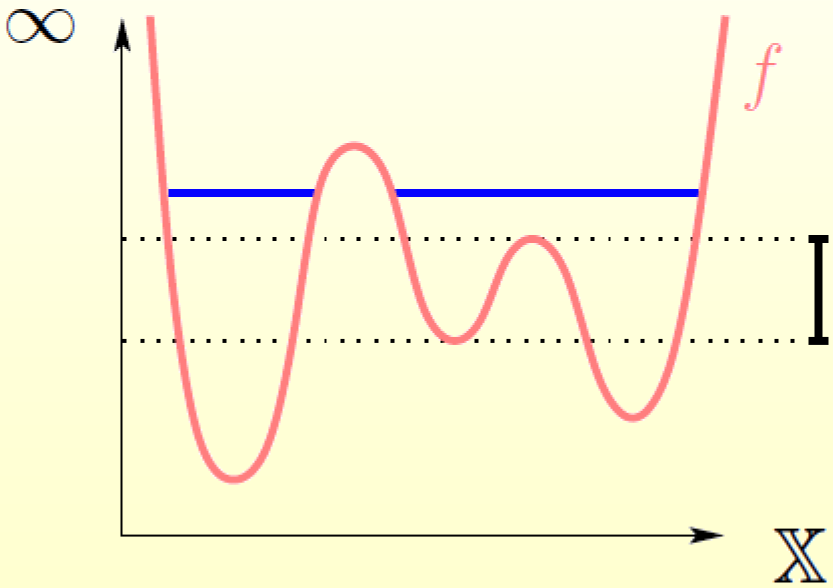
Persistence Provides a Pairing

- Sublevel sets of a function example



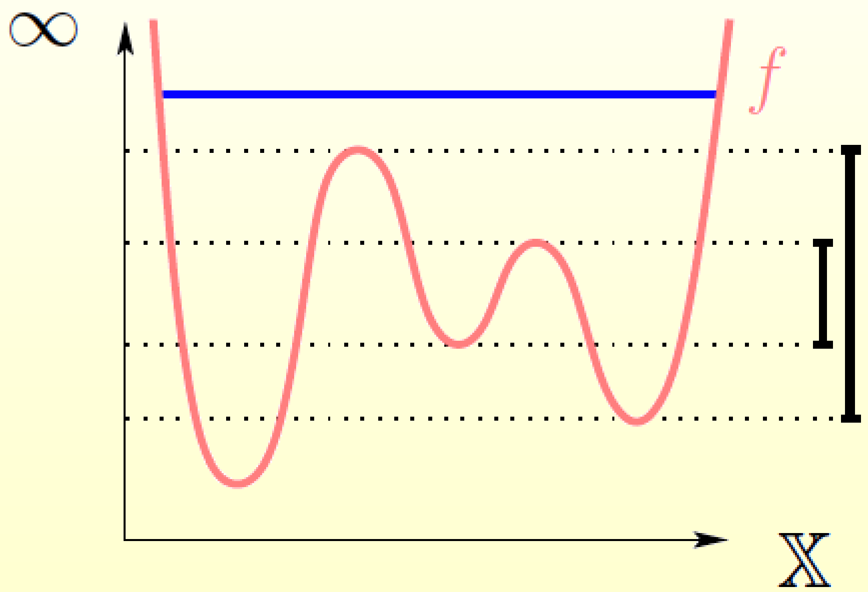
Persistence Provides a Pairing

- Sublevel sets of a function example



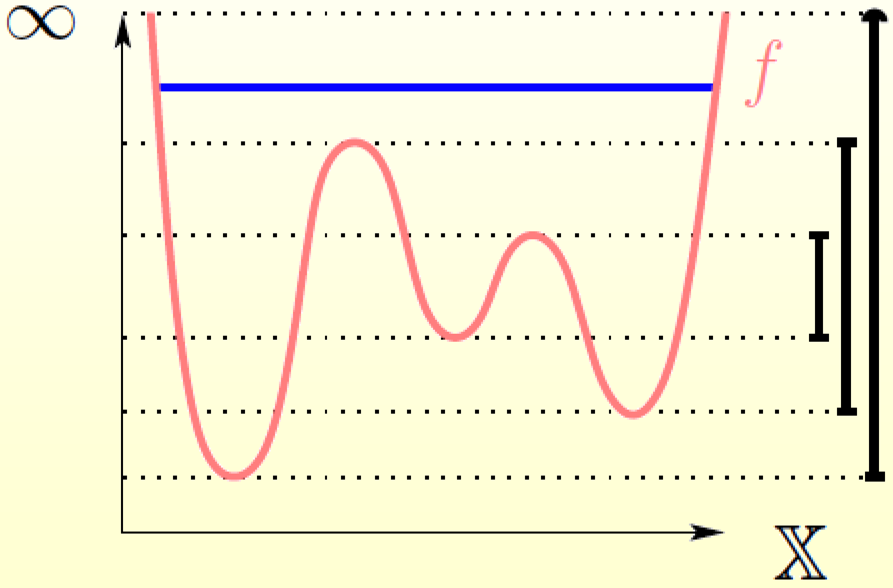
Persistence Provides a Pairing

- Sublevel sets of a function example



Persistence Provides a Pairing

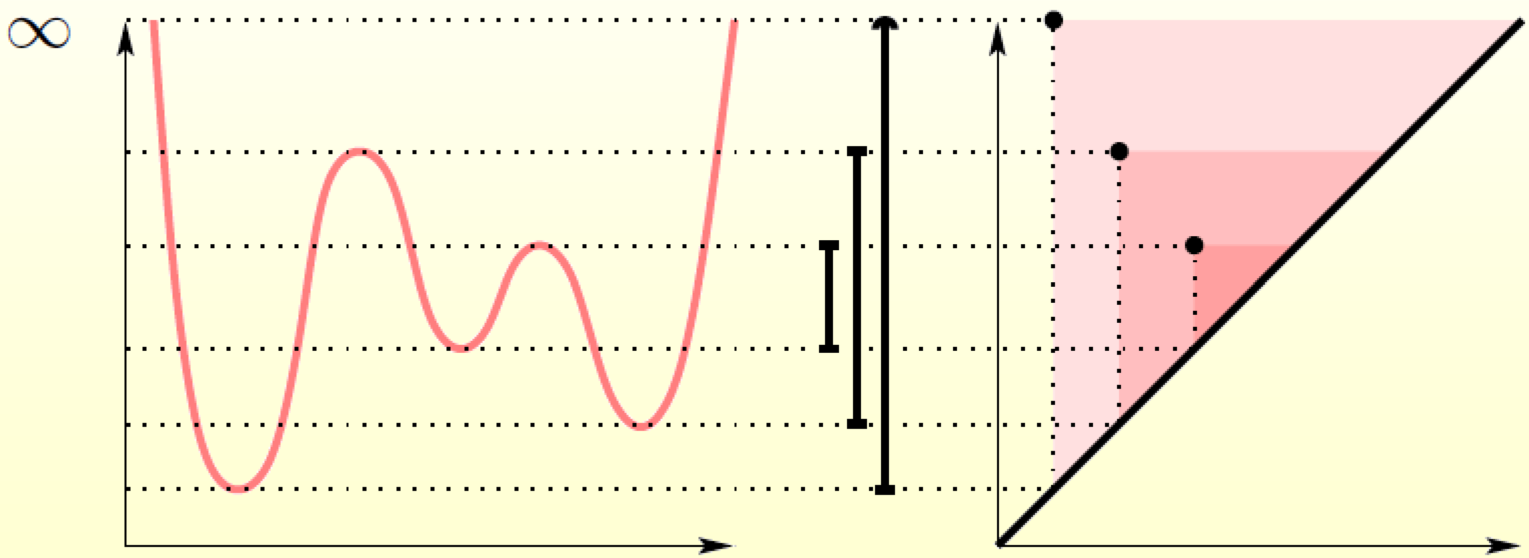
- Sublevel sets of a function example



- Pair thresholds that create components with those that destroy them

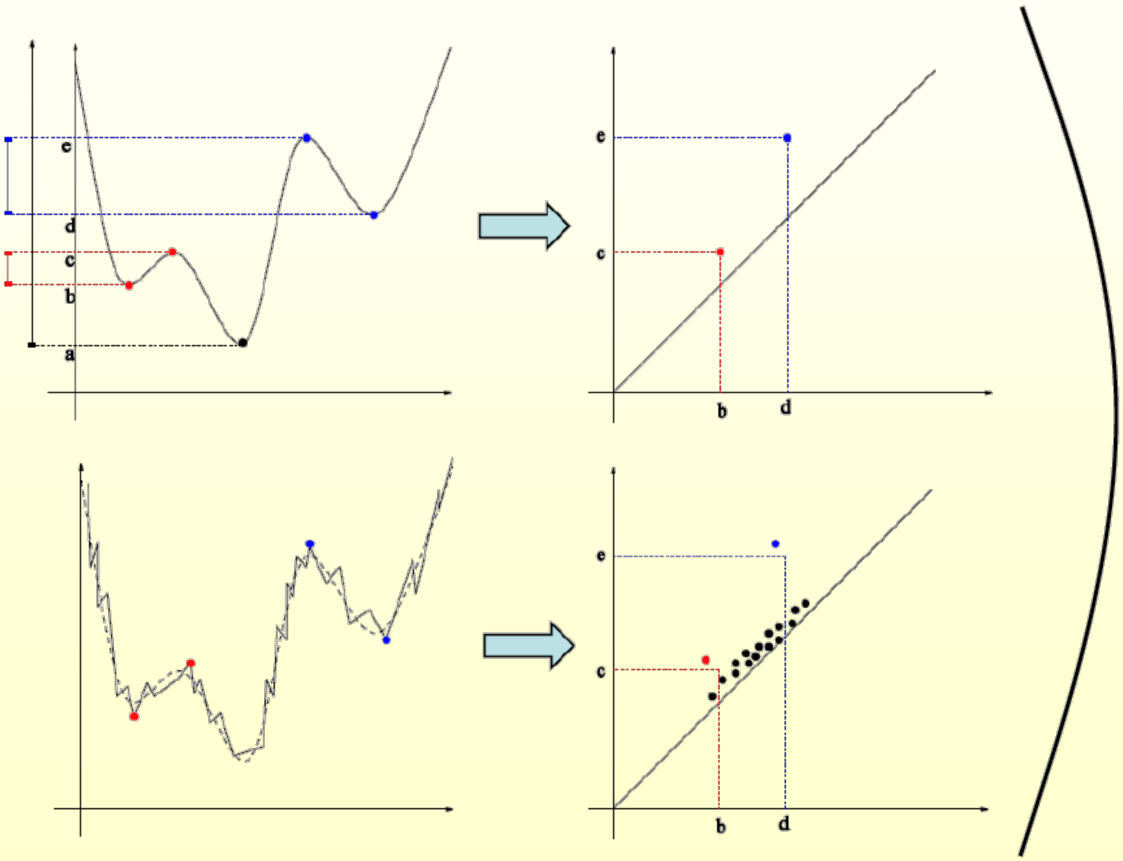
Persistence Provides a Pairing

- That pairing is the persistence diagram



- The diagonal is always included

Filtering Out Topological Noise

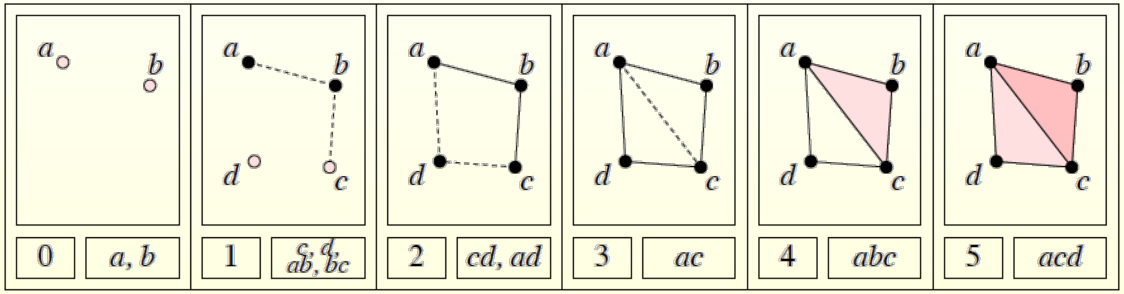


Persistence diagrams

Computing Persistent Homology

Simplicial Filtrations for Low D

- Use a simplicial filtration



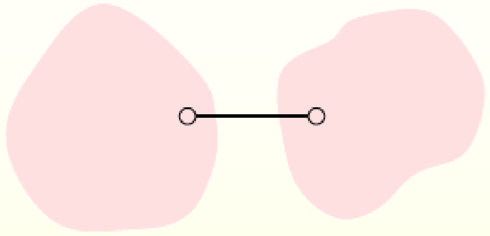
- A filtration of a complex K is $\emptyset = K^0 \subseteq K^1 \subseteq \ldots \subseteq K^m = K$.
- A filtration is a partial ordering
- Sort according to dimension
- Break other ties arbitrarily

Vertices

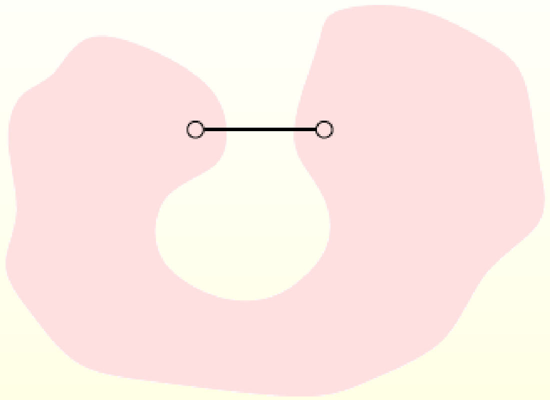
- Vertices always add a new component, so β_0++.
- Union-find data-structure:
  - MakeSet: initializes a set with an item
  - Find: finds the set an element belongs to
  - Union: forms the union of two sets
- Very simple to implement
- $O(n)$ space
- Amortized $\alpha(m)$ Find, Union
- MakeSet for each vertex

β_0 requires maintaining connected components

Edges



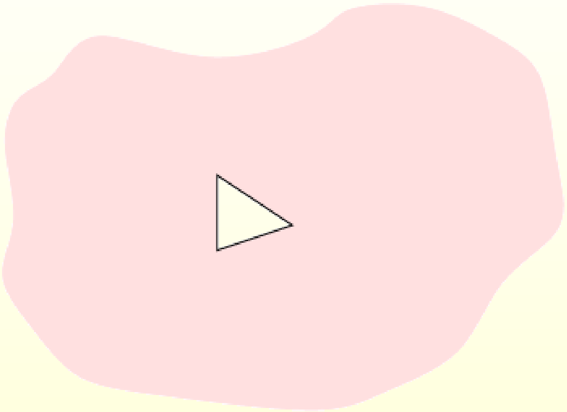
(a) β_0^{--}



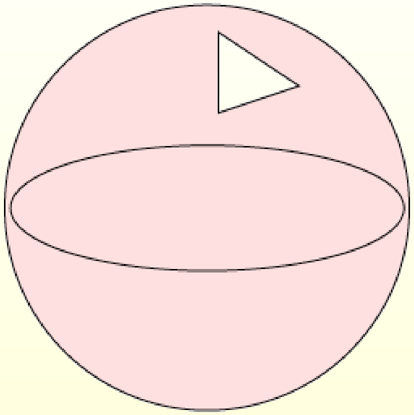
(b) β_1^{++}

- (a) Two FINDs, one UNION
- (b) Two FINDs

Triangles and Tetrahedra



(a) β_1^{--}

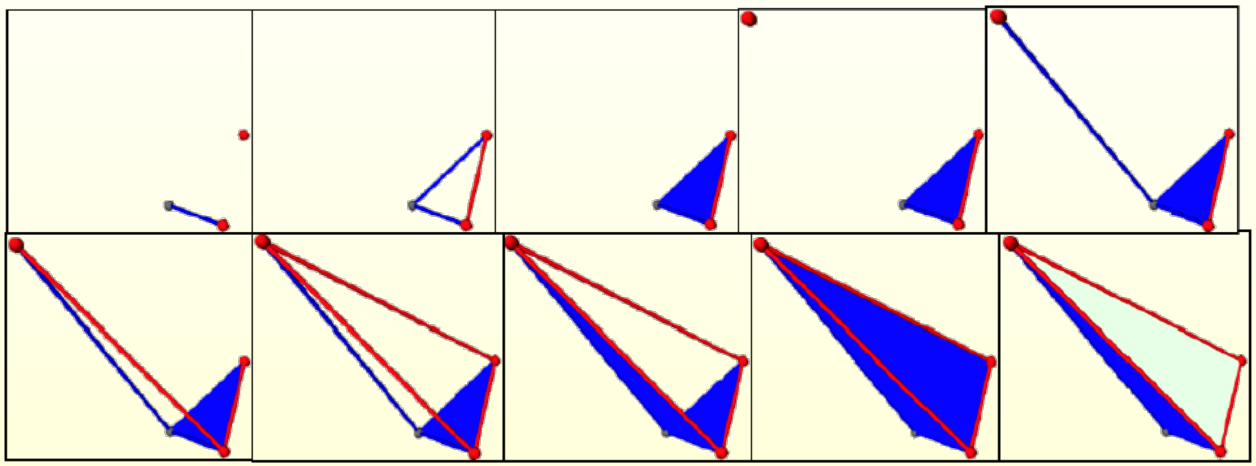


(b) β_2^{++}

- Tetrahedra always fill voids, so β_2^{--}

Positive and Negative Simplices

Let $\emptyset = K^0 \subset K^1 \subset \cdots \subset K^m = K$ be a filtration of a simplicial complex K s. t. $K^{i+1} = K^i \cup \sigma^{i+1}$ where σ^{i+1} is a simplex of K.



Definition: A $(k+1)$-simplex σ^i is positive if it is contained in a $(k+1)$-cycle in K^i. It is negative otherwise.

Create a new $(k+1)$-cycle in K^i

Destroy a k-cycle in K^i

$$\beta_k(K) = \sharp(\text{positive simplices}) \; - \; \sharp(\text{negative simplices})$$

Tracking Topological Features

Definition: A $(k+1)$-simplex σ^i is positive if it is contained in a $(k+1)$-cycle in K^i. It is negative otherwise.

Create a new $(k+1)$-cycle in K^i

Destroy a k-cycle in K^i

$$\beta_k(K) = \sharp(\text{positive simplices}) \; - \; \sharp(\text{negative simplices})$$

- How to keep track of the evolution of the topology all along the filtration?
- What are the created/destroyed cycles?
- What is the lifetime of a cycle?
- How to compute rank$(H_k(K^i) \to H_k(K^j))$?

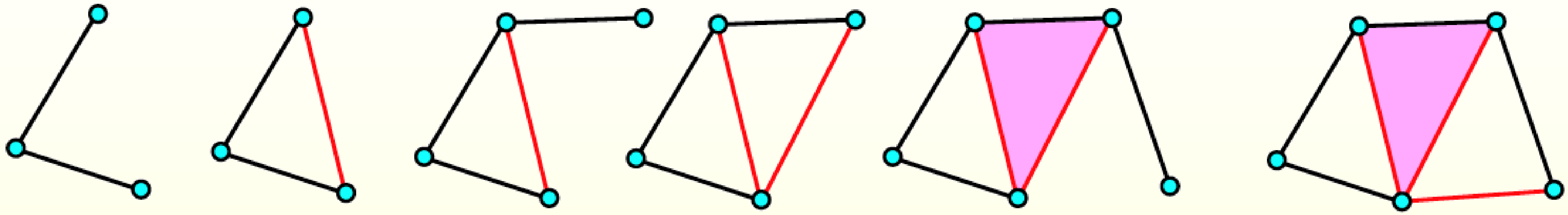
This is where topological persistence comes into play!

Notation

In the following:

- Let $\emptyset = K^0 \subset K^1 \subset \cdots \subset K^m = K$ be a filtration of a simplicial complex K s. t. $K^{i+1} = K^i \cup \sigma^{i+1}$ where σ^{i+1} is a simplex of K.
- Z_k^i = the k-cycles of K^i, B_k^i = the k-boundaries of K^i and H_k^i = the k^{th}-homology group of K^i.
- $Z_k^0 \subseteq Z_k^1 \subseteq \cdots \subseteq Z_k^i \subseteq \cdots \subseteq Z_k^m = Z_k(K)$
- $B_k^0 \subseteq B_k^1 \subseteq \cdots \subseteq B_k^i \subseteq \cdots \subseteq B_k^m = B_k(K)$

Cycle Associated to a Positive Simplex

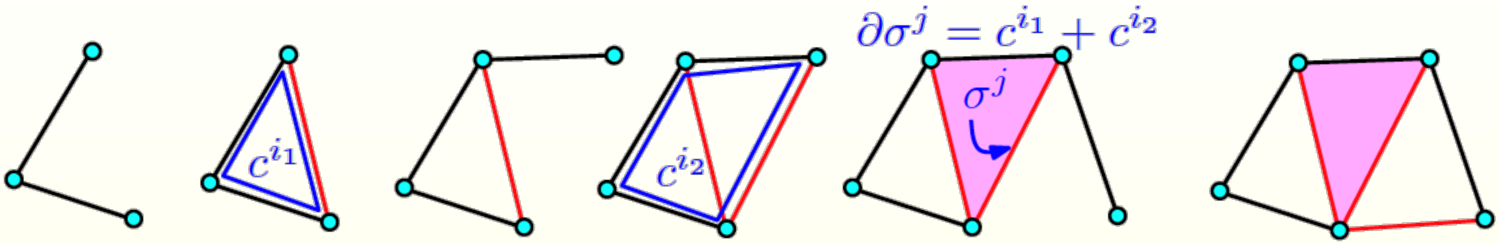


Lemma: If σ^i is a positive k-simplex, then there exists a k-cycle c_σ s.t.:
- c_σ is not a boundary in K^i,
- c_σ contains σ^i but no other positive k-simplex.

The cycle c^σ is unique.

Proof:
By induction on the order of appearence of the simplices in the filtration.

Updating the Homology Basis



- At the beginning: the basis of H_k^0 is empty.
- If a basis of H_k^{i-1} has been built and σ^i is a positive k-simplex then one adds the homology class of the cycle c^i associated to σ^i to the basis of $H_k^{i-1} \Rightarrow$ basis of H_k^i.
- If a basis of H_k^{j-1} has been built and σ^j is a negative $(k+1)$-simplex:
  - let $c^{i_1}, \cdots, c^{i_p}$ be the cycles associated to the positive simplices $\sigma^{i_1}, \cdots, \sigma^{i_p}$ that form a basis of H_k^{j-1}
  - $d = \partial\sigma^j = \sum_{k=1}^{p} \varepsilon_k c^{i_k}$
  - $l(j) = \max\{i_k : \varepsilon_k = 1\}$
  - Remove the homology class of $c^{l(j)}$ from the basis of $H_k^{j-1} \Rightarrow$ basis of H_k^j.

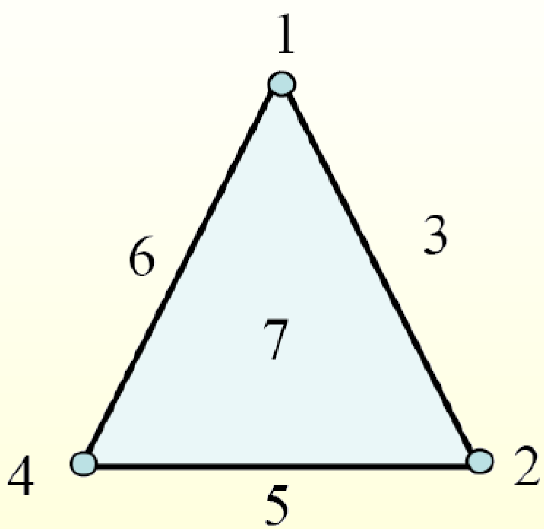
Pairing Simplices

- If a basis of H_k^{j-1} has been built and σ^j is a negative $(k+1)$-simplex:
  - let $c^{i_1}, \cdots, c^{i_p}$ be the cycles associated to the positive simplices $\sigma^{i_1}, \cdots, \sigma^{i_p}$ that form a basis of H_k^{j-1}
  - $d = \partial \sigma^j = \sum_{k=1}^{p} \varepsilon_k c^{i_k}$
  - $l(j) = \max\{i_k : \varepsilon_k = 1\}$
  - Remove the homology class of $c^{l(j)}$ from the basis of $H_k^{j-1} \Rightarrow$ basis of H_k^j.

The simplices $\sigma^{l(j)}$ and σ^j are paired to form a persistent pair $(\sigma^{l(j)}, \sigma^j)$.
$\rightarrow$ The homology class created by $\sigma^{l(j)}$ in $K^{l(j)}$ is killed by σ^j in K^j. The persistence (or life-time) of this cycle is : $j - l(j) - 1$.

The persistence pairing

Matrix of Boundary Operator



$$
\begin{array}{cc}
\begin{array}{ccccccc} & & & & j & & \end{array} & \\
\begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & \boxed{1} & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} & \begin{array}{c} \\ \\ \\ l(j) \\ \\ \\ \\ \end{array}
\end{array}
$$

- $M = (m_{ij})_{i,j=1,\cdots,m}$ with coefficient in $\mathbb{Z}/2$ defined by

$$m_{ij} = 1 \text{ if } \sigma^i \text{ is a face of } \sigma^j \text{ and } m_{ij} = 0 \text{ otherwise}$$

- For any column C_j, $l(j)$ is defined by

$$(i = l(j)) \Leftrightarrow (m_{ij} = 1 \text{ and } m_{i'j} = 0 \quad \forall i' > i)$$

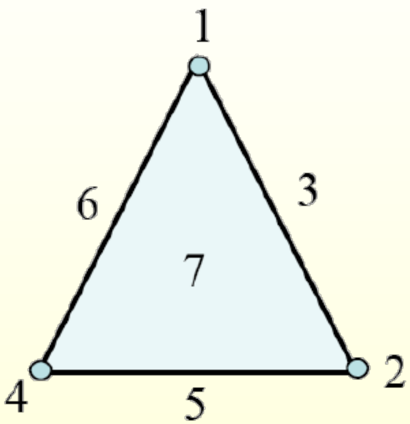
Persistence Algorithm, Version 2

Input: $\emptyset = K^0 \subset K^1 \subset \cdots \subset K^m = K$ a d-dimensional filtration of a simplicial complex K s. t. $K^{i+1} = K^i \cup \sigma^{i+1}$ where σ^{i+1} is a simplex of K.

For $j = 0$ to m
  While (there exists $j' < j$ such that $l(j') == l(j)$)
    $C_j = C_j + C_{j'}$ mod(2);
  End while
End for
Output the pairs $(l(j), j)$;

Remark: The worst case complexity of the algorithm is $O(m^3)$ but much lower in most practical cases.

Example



$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & \boxed{1} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \boxed{1} & \boxed{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

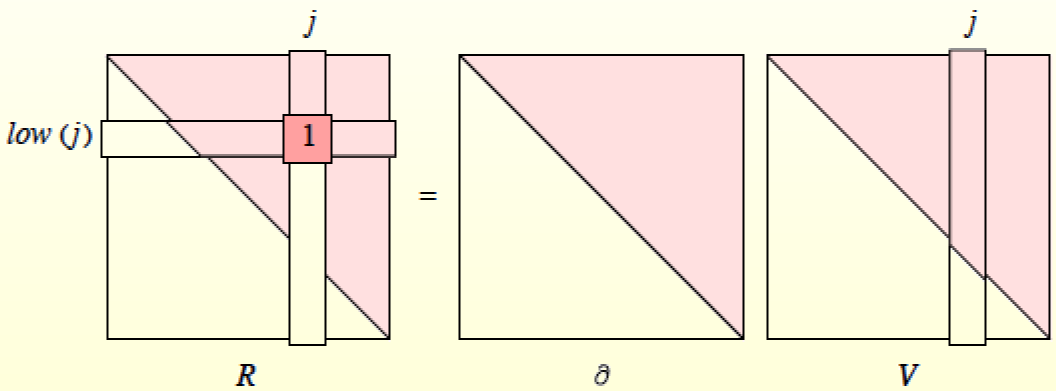
$$\begin{array}{ccccccc} & & & & & C_5 + C_6 & \\ & & & & & \downarrow & \end{array}$$
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & \boxed{1} & 0 & 1 & \boxed{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\begin{array}{ccccccc} & & & & & C_3 + C_6 & \\ & & & & & \downarrow & \end{array}$$
$$\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boxed{1} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Pairs: $(2,3)$ $(4,5)$ $(6,7)$

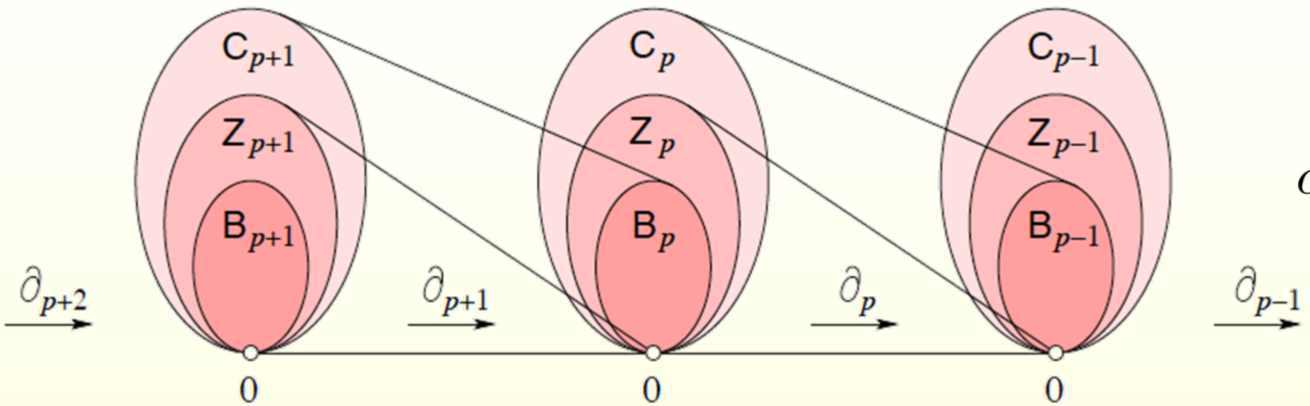
Persistence Algorithm Through Matrix Operations

- See the Edelsbrunner-Harer book



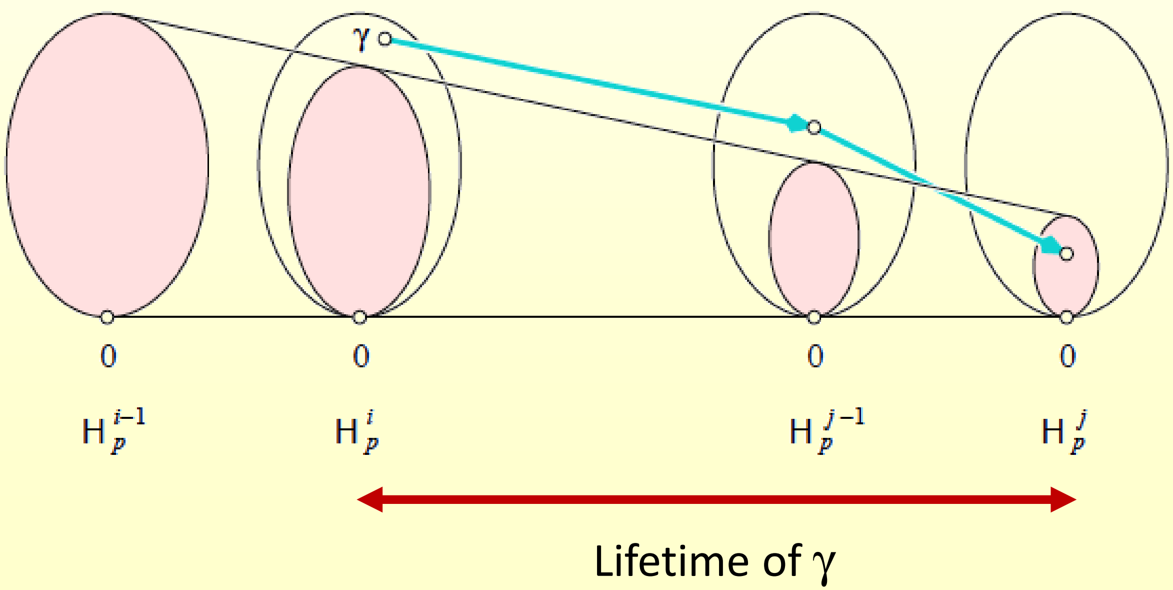
Topology Inference Pipeline

Persistence of Homology Classes



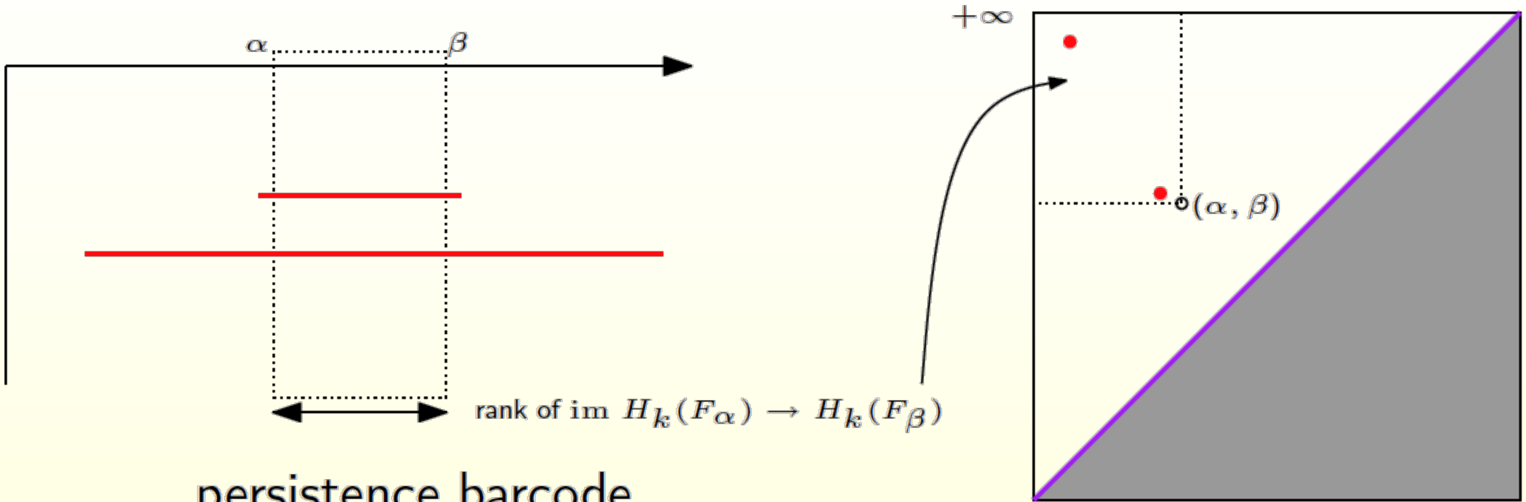
$$c_p = z_p + b_{p-1}$$

$$G/\ker(f) \equiv \mathrm{im}(f)$$



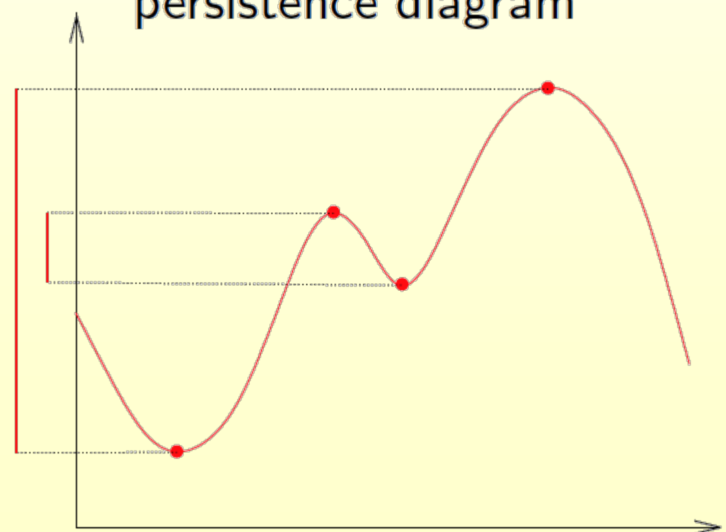
Barcodes and Persistence Diagrams, Stability

Barcodes vs Persistence Diagrams



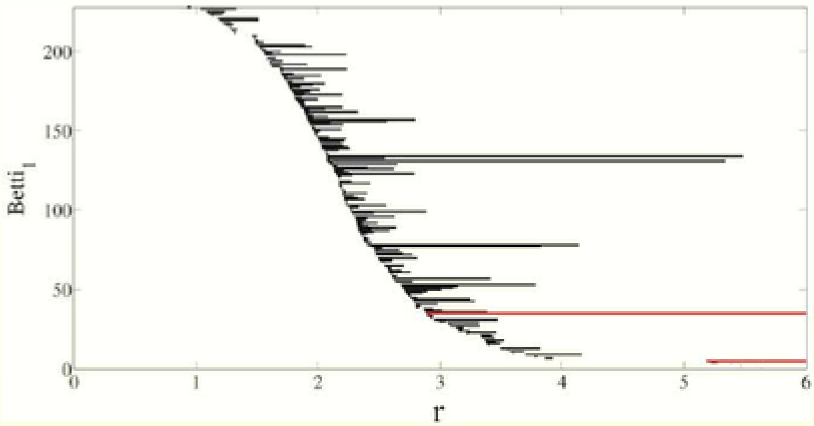
persistence barcode

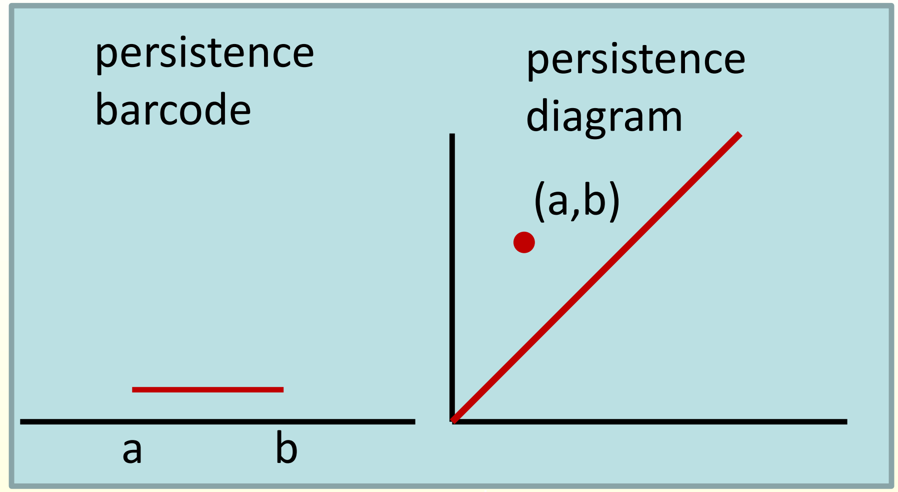
persistence diagram



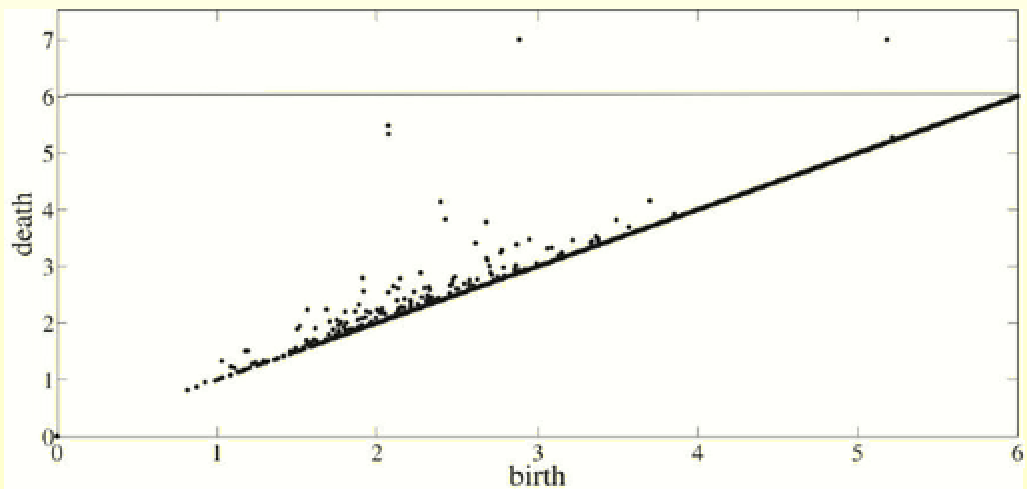
Structure Thm. [Carlsson, Zomorodian 04]
the kth persistent homology of $(\mathbb{X}, f)$ is fully described by a finite set of intervals, each of which represents the lifespan of an element in a basis that is compatible accross the filtration.

Barcodes vs Persistence Diagrams





long barcodes = points away from the diagonal = robust features

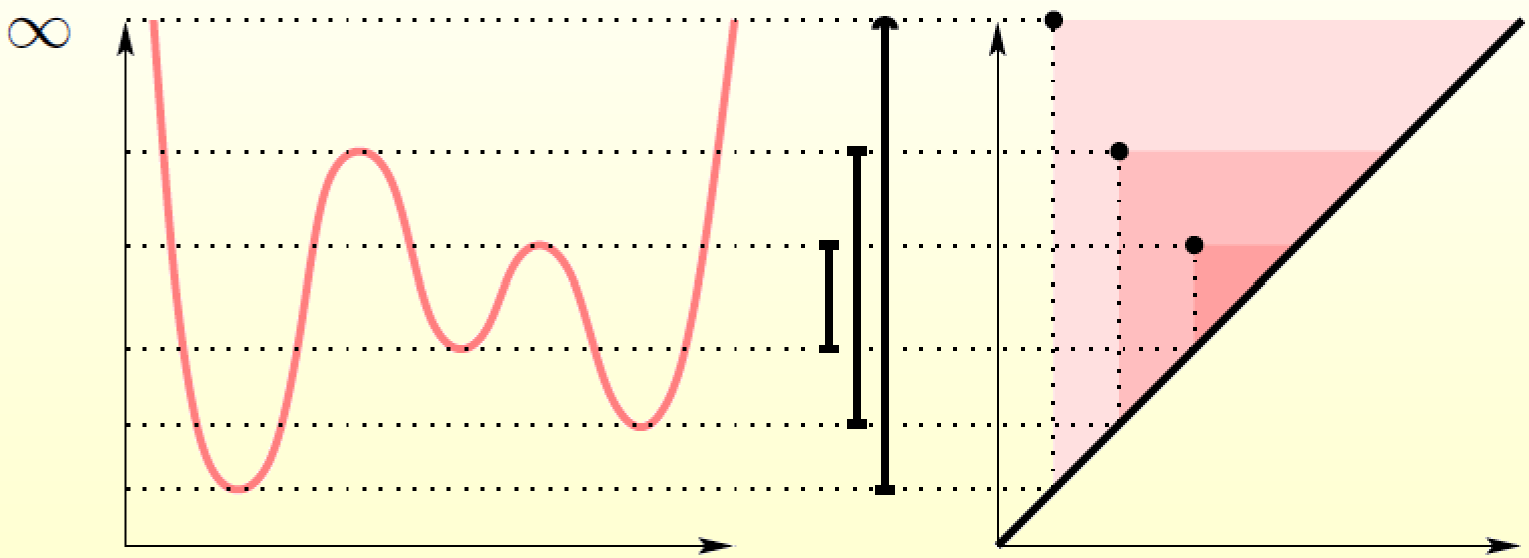


Short barcodes = points near the diagonal = noise

Map 1-D intervals to points in 2-D

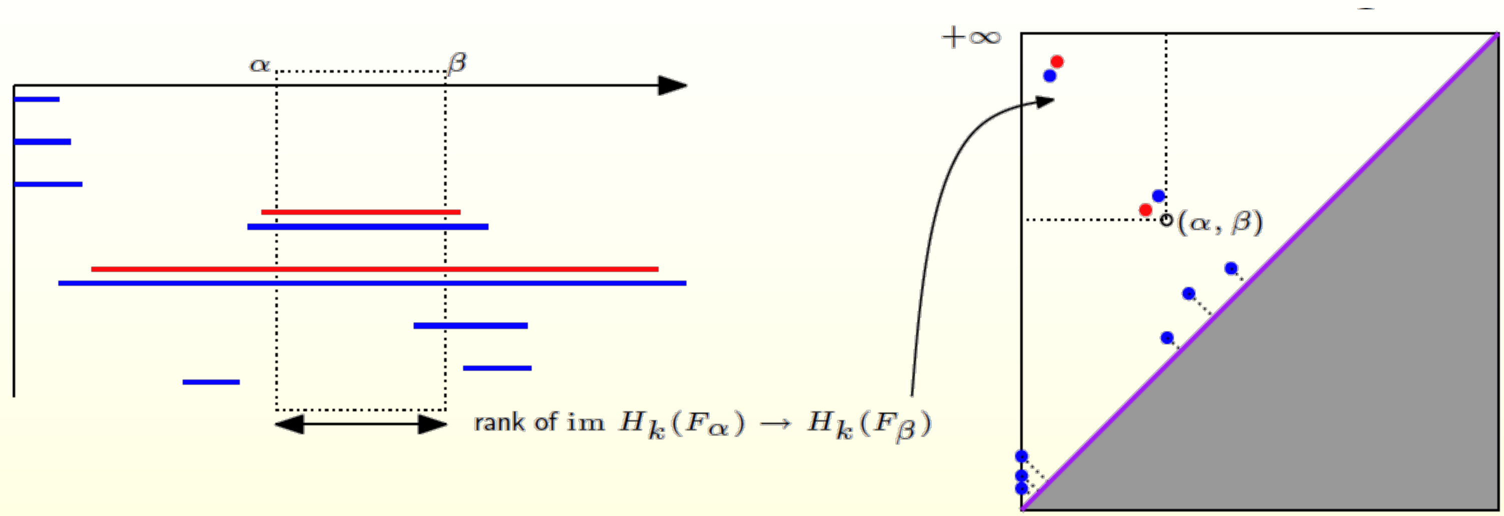
Persistence Provides a Pairing

- That pairing is the persistence diagram

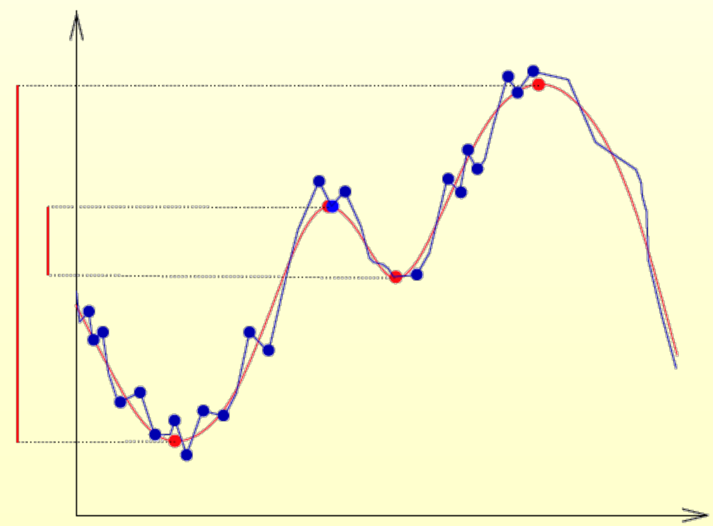


- The diagonal is always included

Filtering Out Topological Noise

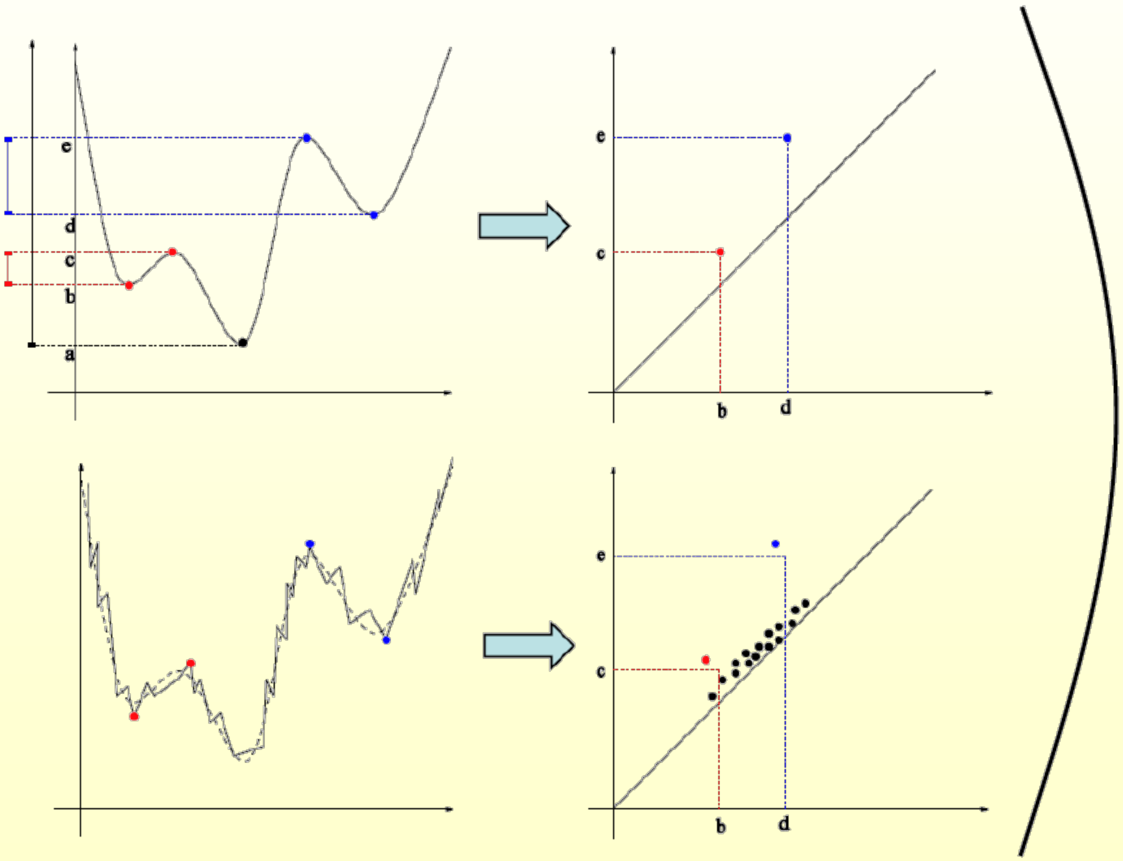


Stability: What if f is slightly perturbed?



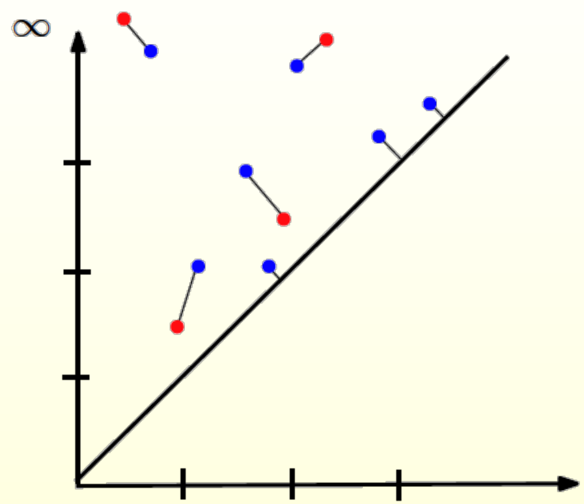
Structure Thm. [Carlsson, Zomorodian 04]
the kth persistent homology of $(\mathbb{X}, f)$ is fully described by a finite set of intervals, each of which represents the lifespan of an element in a basis that is compatible accross the filtration.

Filtering Out Topological Noise



Persistence diagrams

Bottleneck Distance Between Persistence Diagrams



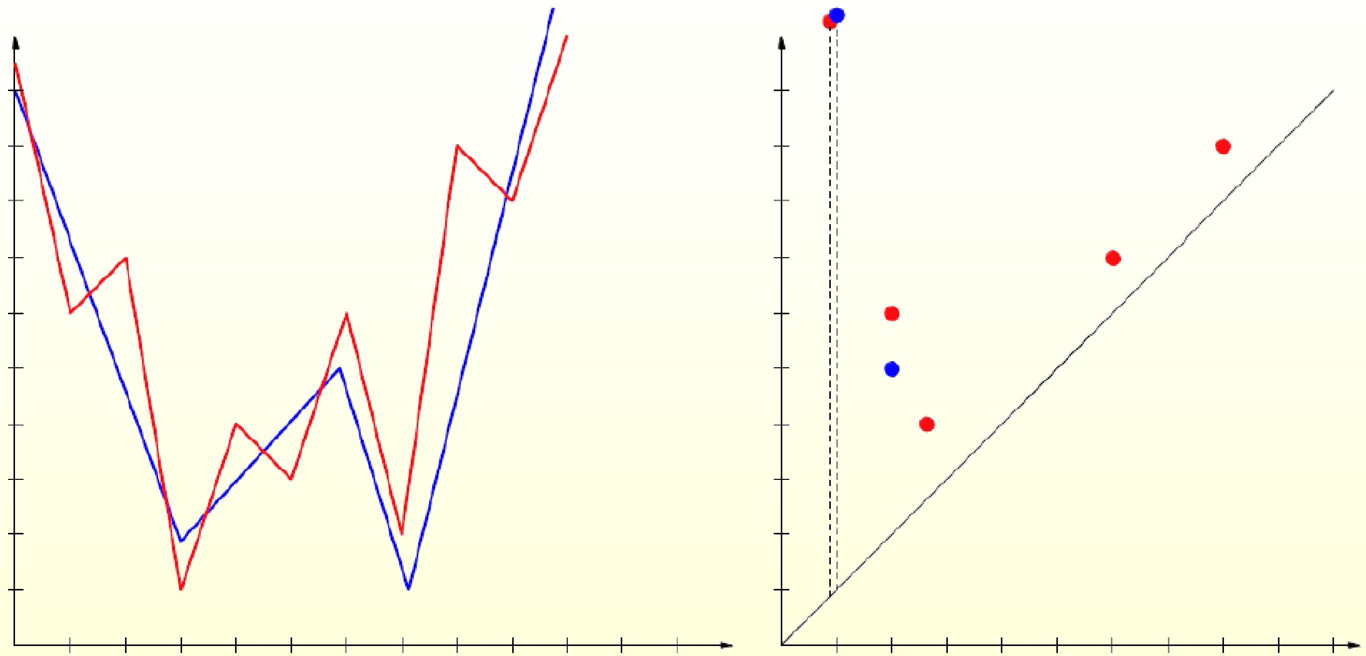
Let K be a simplicial complex and f, g two functions defined on the vertices of K. Let D_f and D_g be the persistence diagrams of f and g.

The bottleneck distance between D_f and D_g is

$$d_B(D_f, D_g) = \inf_{\gamma \in \Gamma} \sup_{p \in D_f} \|p - \gamma(p)\|_\infty$$

where Γ is the set of all the bijections between D_f and D_g and $\|p - q\|_\infty = \max(|x_p - x_q|, |y_p - y_q|)$.

Stability Theorems



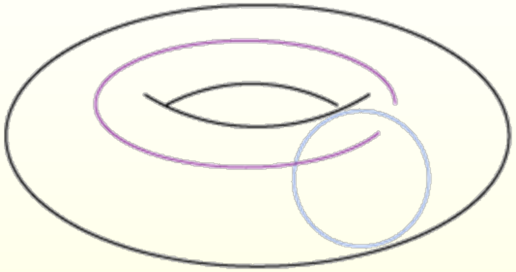
Theorem: Let K be a simplicial complex and let $f, g : K \to \mathbb{R}$.

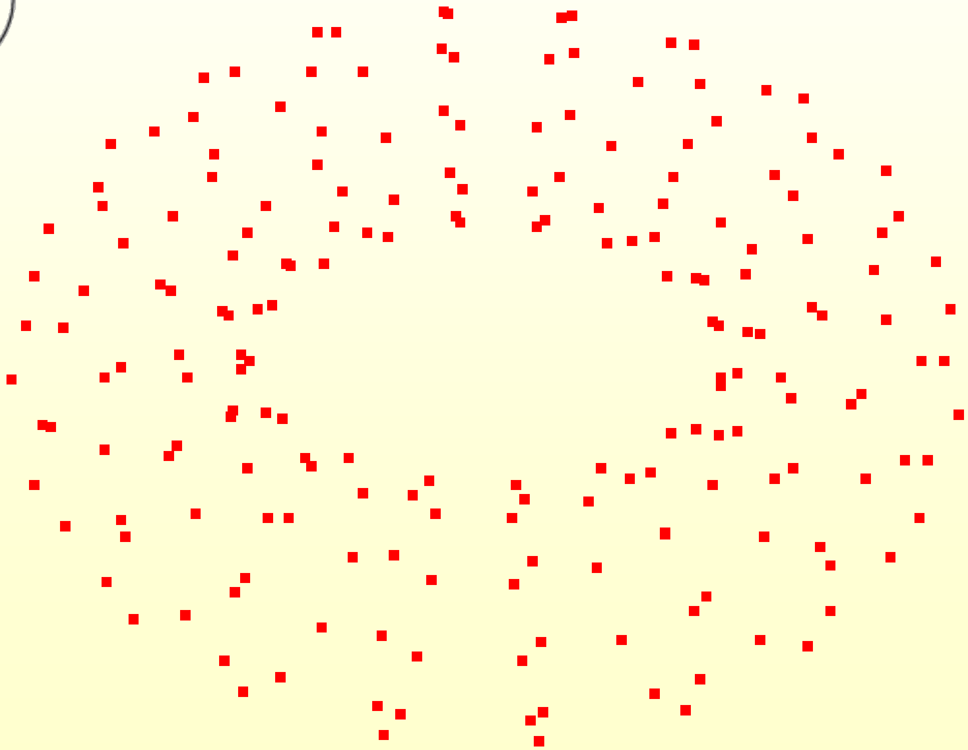
$$d_B(D_f, D_g) \leq \|f - g\|_\infty$$

where $\|f - g\|_\infty = \sup_{v \in vertices(K)} |f(v) - g(v)|$.

Persistent Homology Examples

Detecting a Torus from Samples

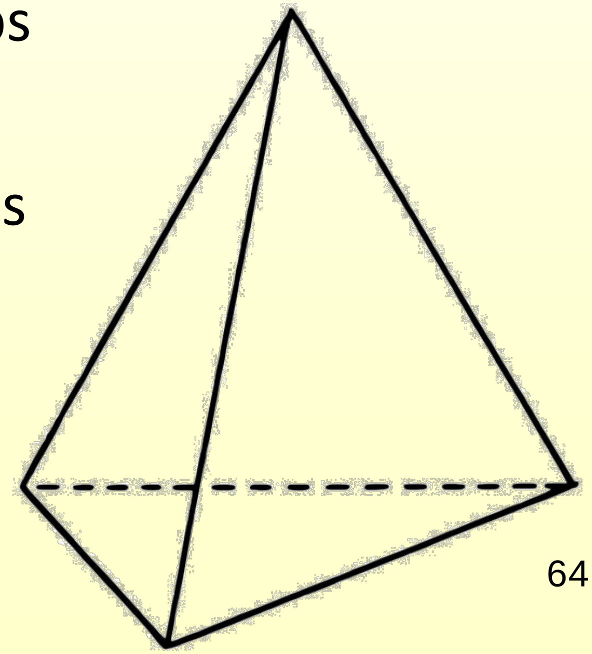




Point Cloud Data
(PCD)

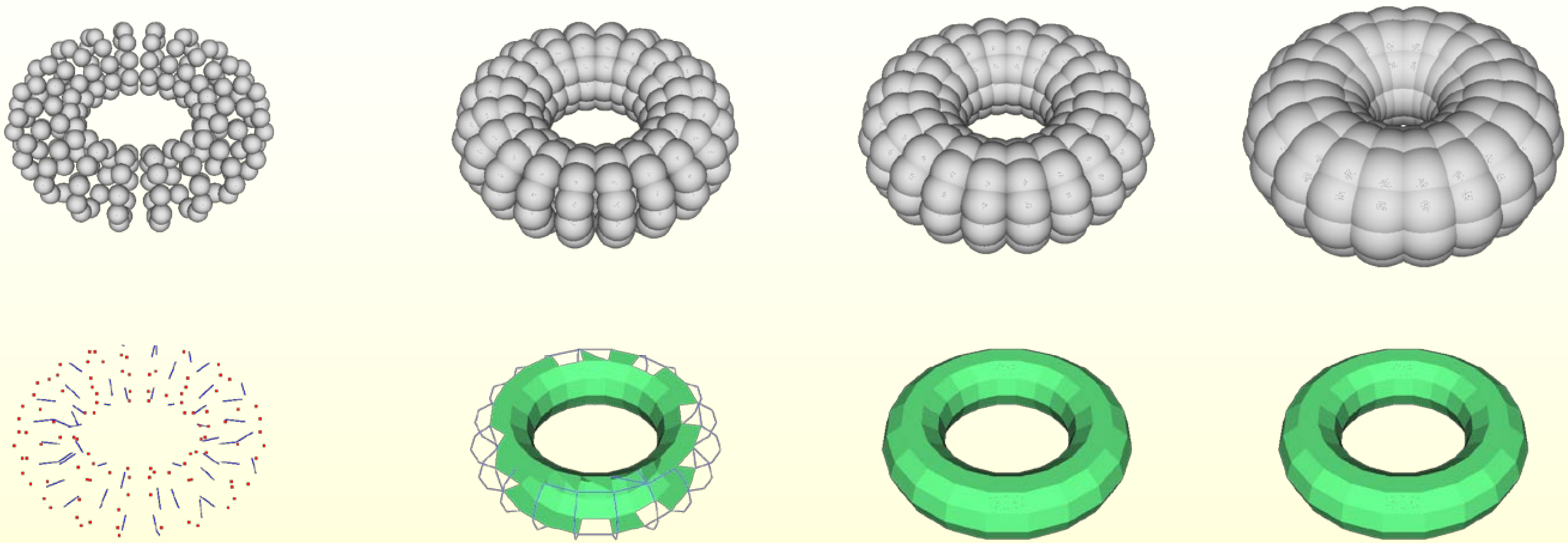
Recall: Betti Numbers β_i

- Ranks of the free part of homology groups H_i
- β_0 counts the number of connected components
- β_1 counts the number of independent loops
- β_2 counts the number of independent voids
- …



Topology is fundamentally a tool for classification

Question of Scale: A Rips Filtration

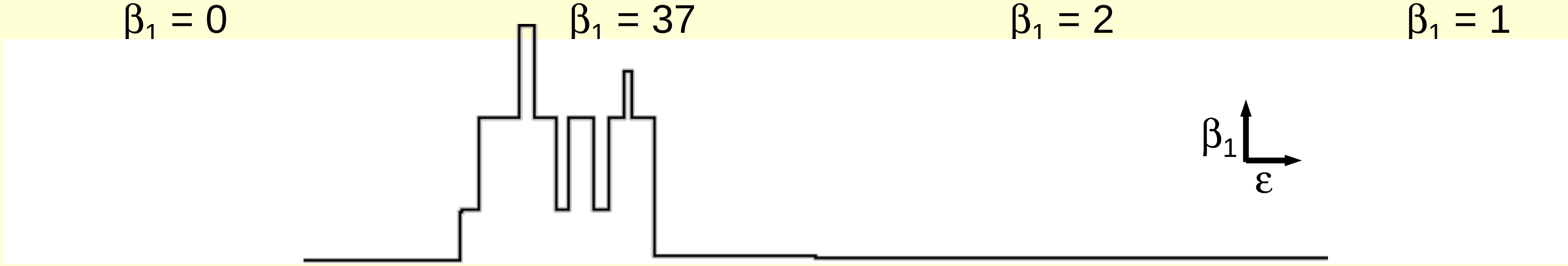


$\beta_0 = 150$
$\beta_1 = 0$

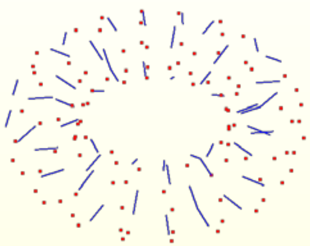
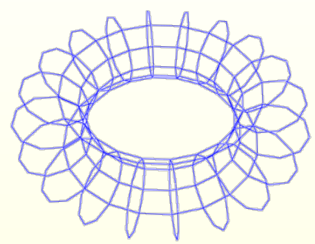
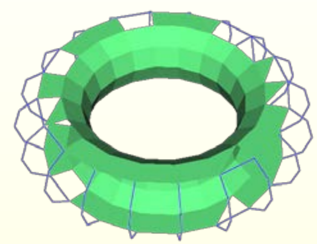
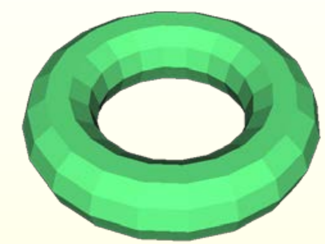
$\beta_0 = 1$
$\beta_1 = 37$

$\beta_0 = 1$
$\beta_1 = 2$

$\beta_0 = 1$
$\beta_1 = 1$



From Complex Inclusions to Homology Homomorphisms

 $\subseteq$  $\subseteq$  $\subseteq$ 

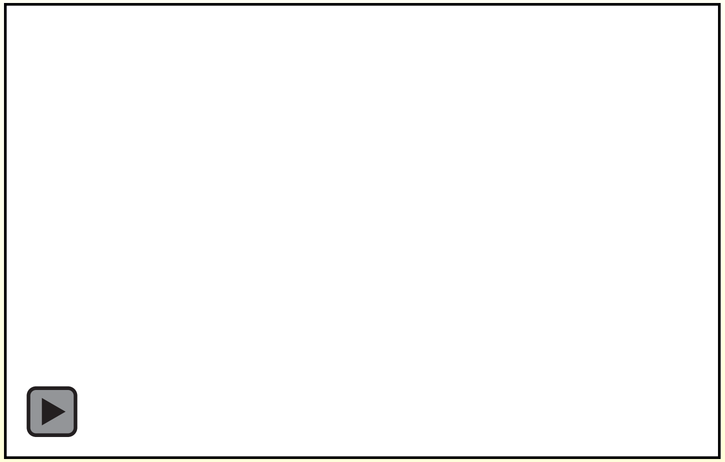
$$K^{250} \xrightarrow[i]{} K^{500} \longrightarrow K^{994} \longrightarrow K^{1452}$$

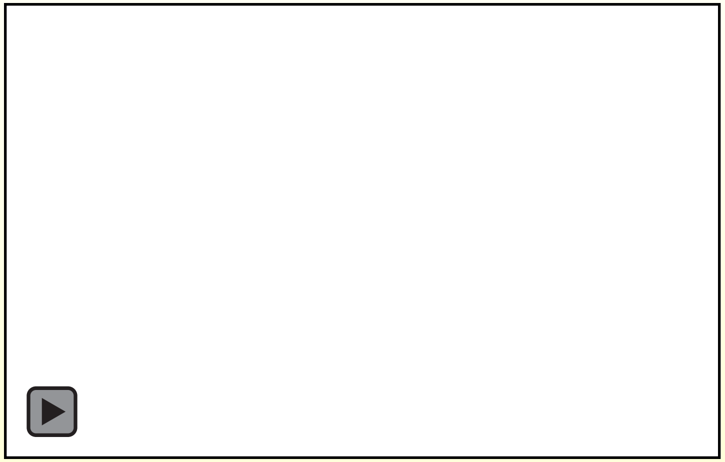
Functoriality $\big\downarrow$

$$H_k(K^{25}) \longrightarrow H_k(K^{50}) \longrightarrow H_k(K^{994}) \longrightarrow H_k(K^{1452})$$

Idea: Follow homology basis elements from birth to death while maintaining compatible bases

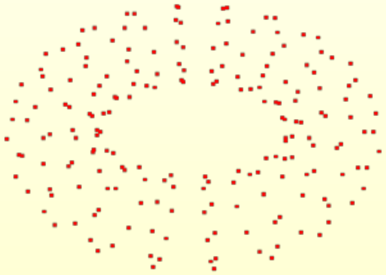
Consistent Bases Exist



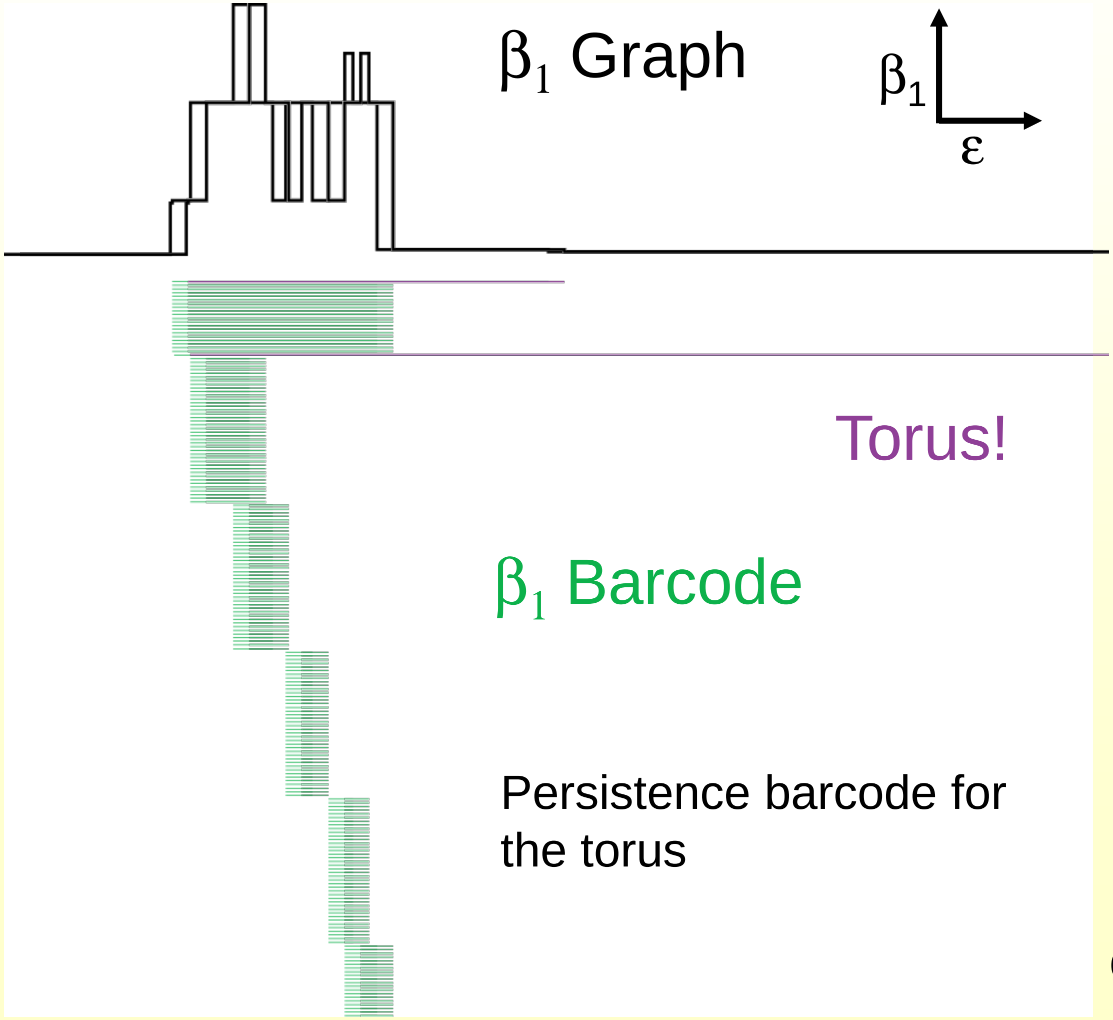


Basis elements for 1-homology

Deconstructing the Barcode

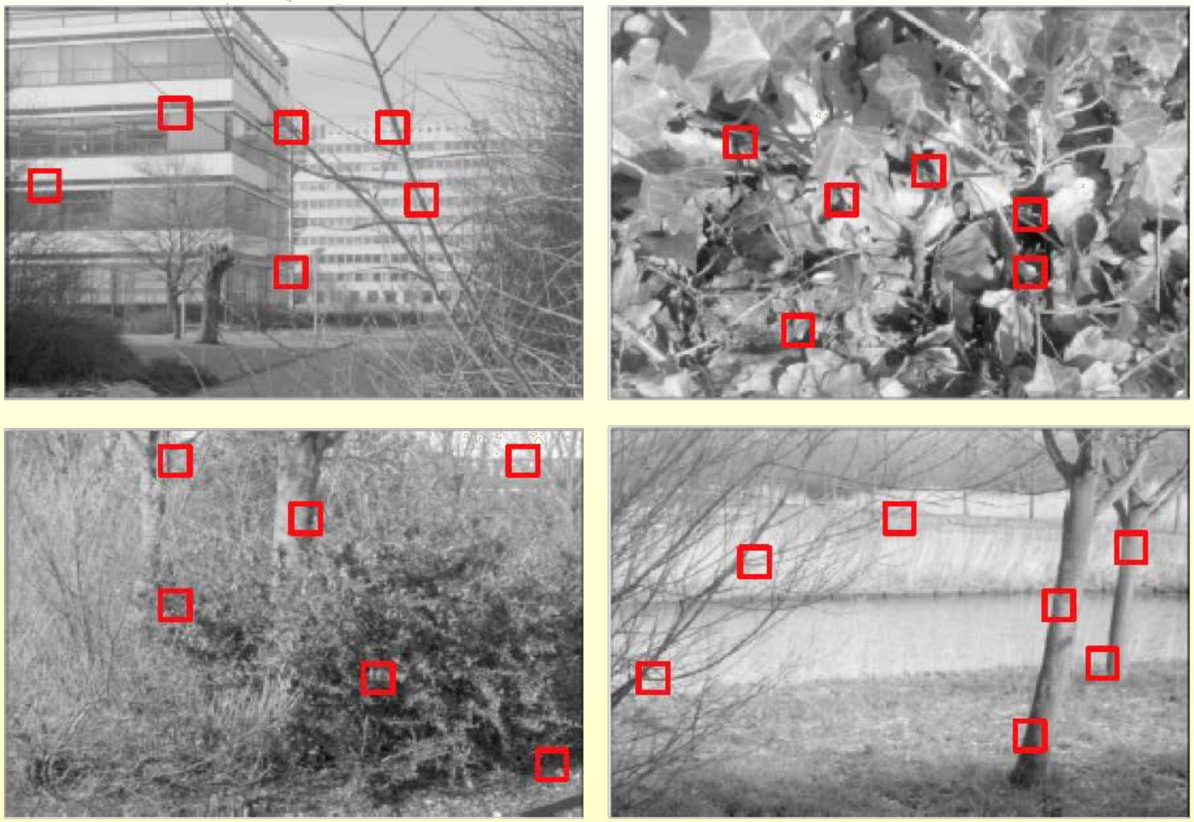


PCD



Back to the Natural Images Example

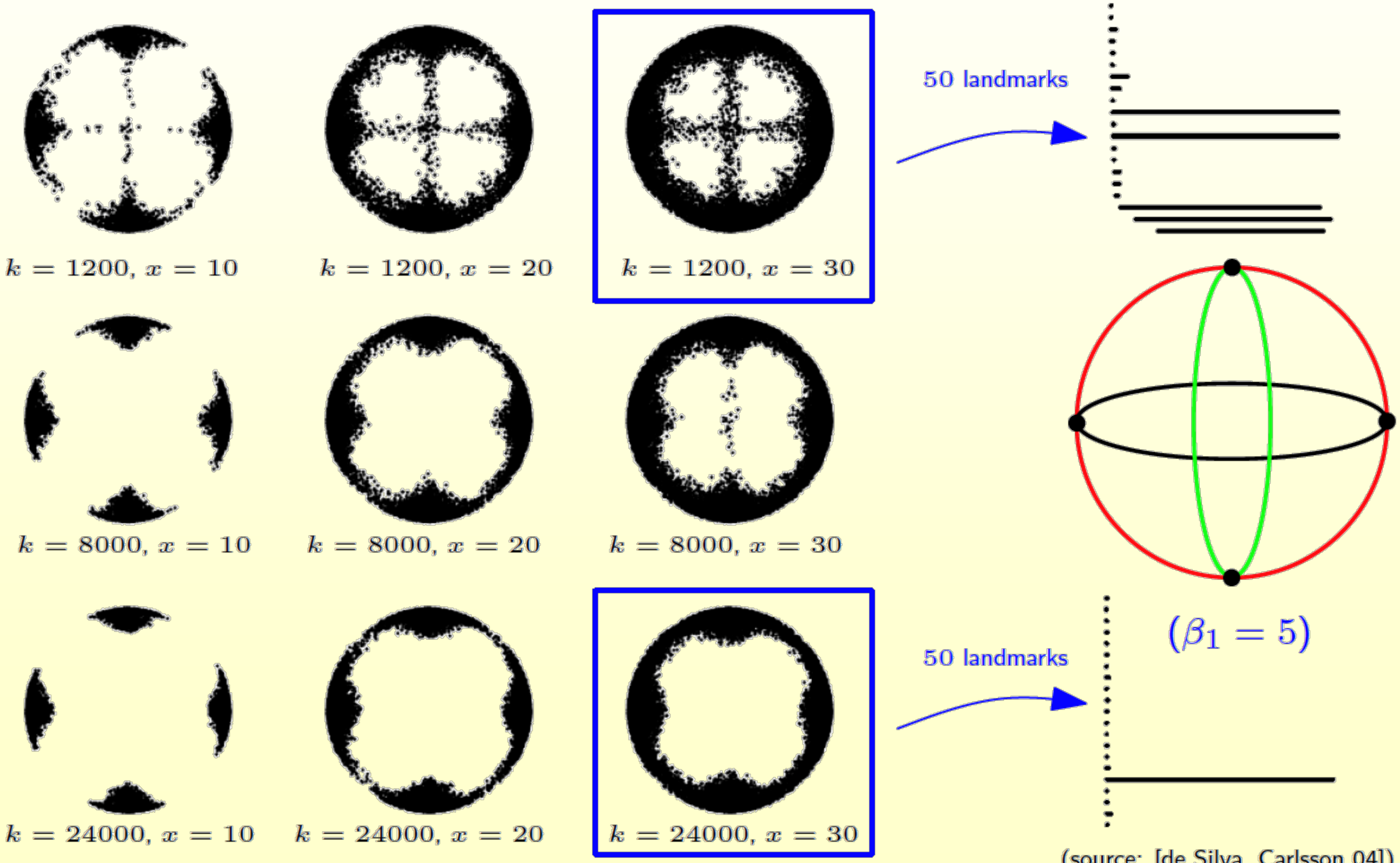
Input: 4 million data points on $\mathbb{S}^7$, coming from high-contrast 3×3 image patches



(source: [Lee, Pederson, Mumford 03])

Back to the Natural Images Example

Preprocessing: - select bottom $x\%$ of data points according to k-NN distance
- sample 5000 points uniformly at random from filtered point set

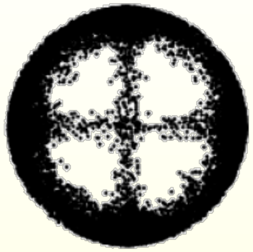


(source: [de Silva, Carlsson 04])

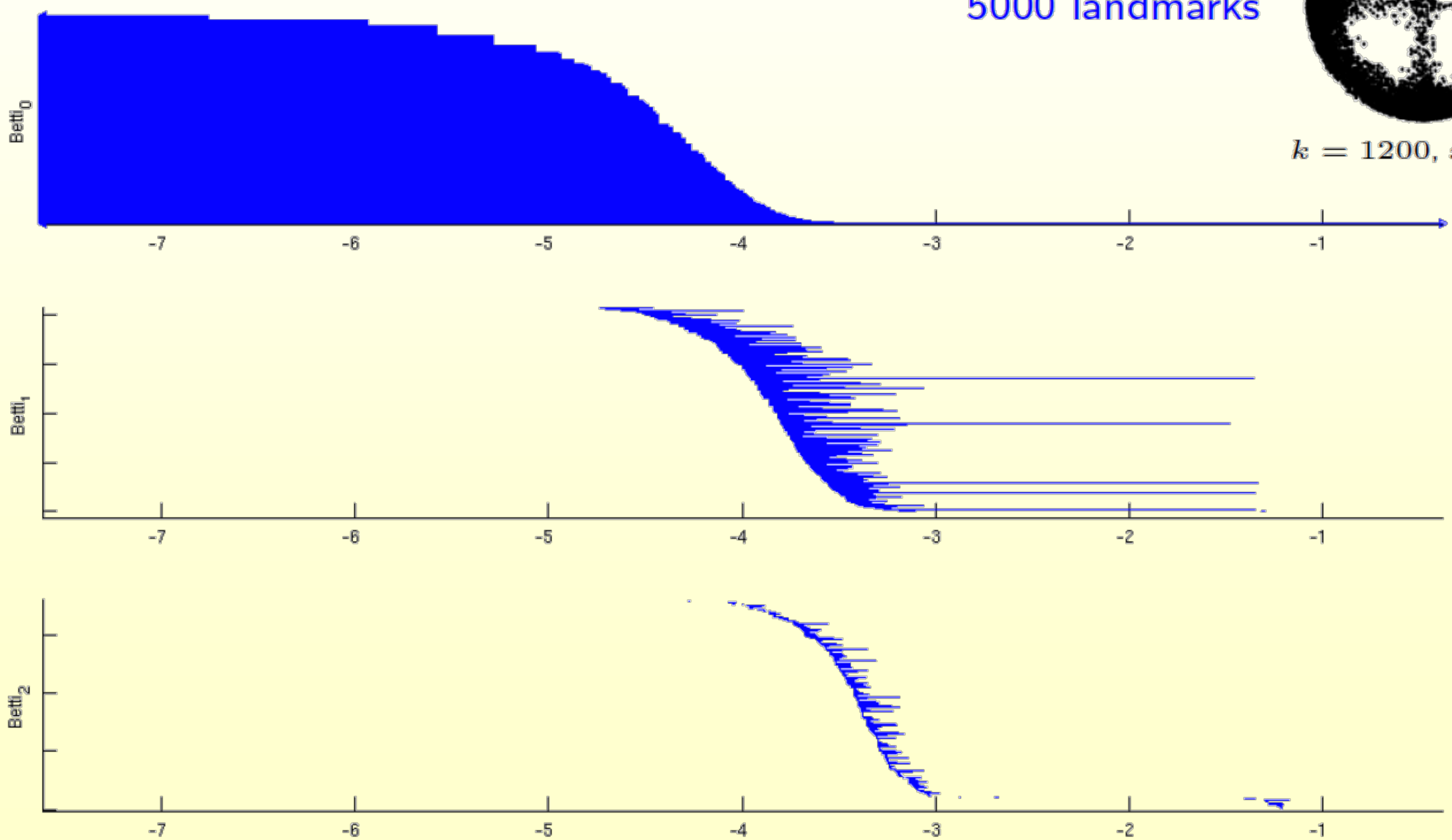
Back to the Natural Images Example

Preprocessing: - select bottom $x\%$ of data points according to k-NN distance

- sample 5000 points uniformly at random from filtered point set



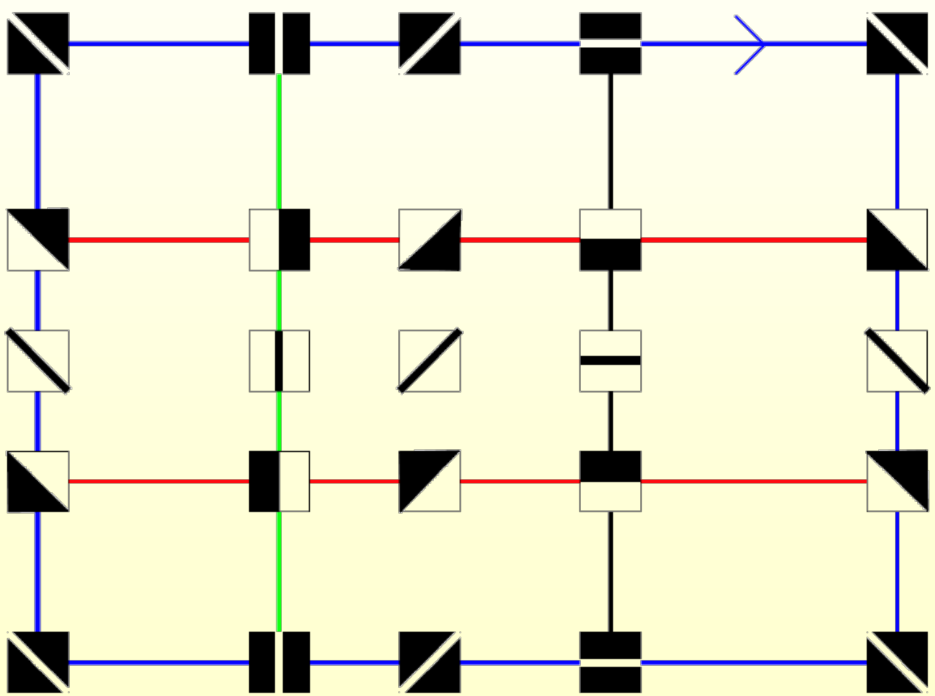
$k = 1200,\ x = 30$

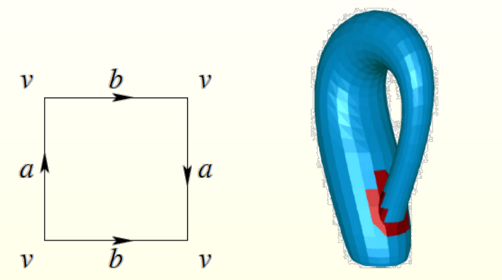


(source: [O., Sheehy 13])

Back to the Natural Images Example

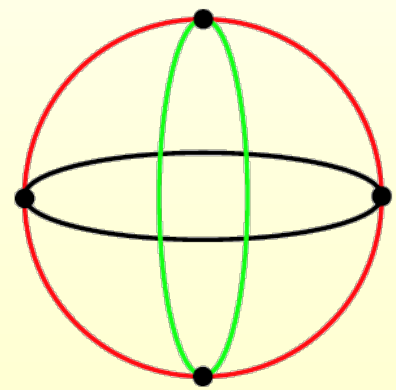
Preprocessing: - select bottom $x\%$ of data points according to k-NN distance
- sample 5000 points uniformly at random from filtered point set





(a) Diagram

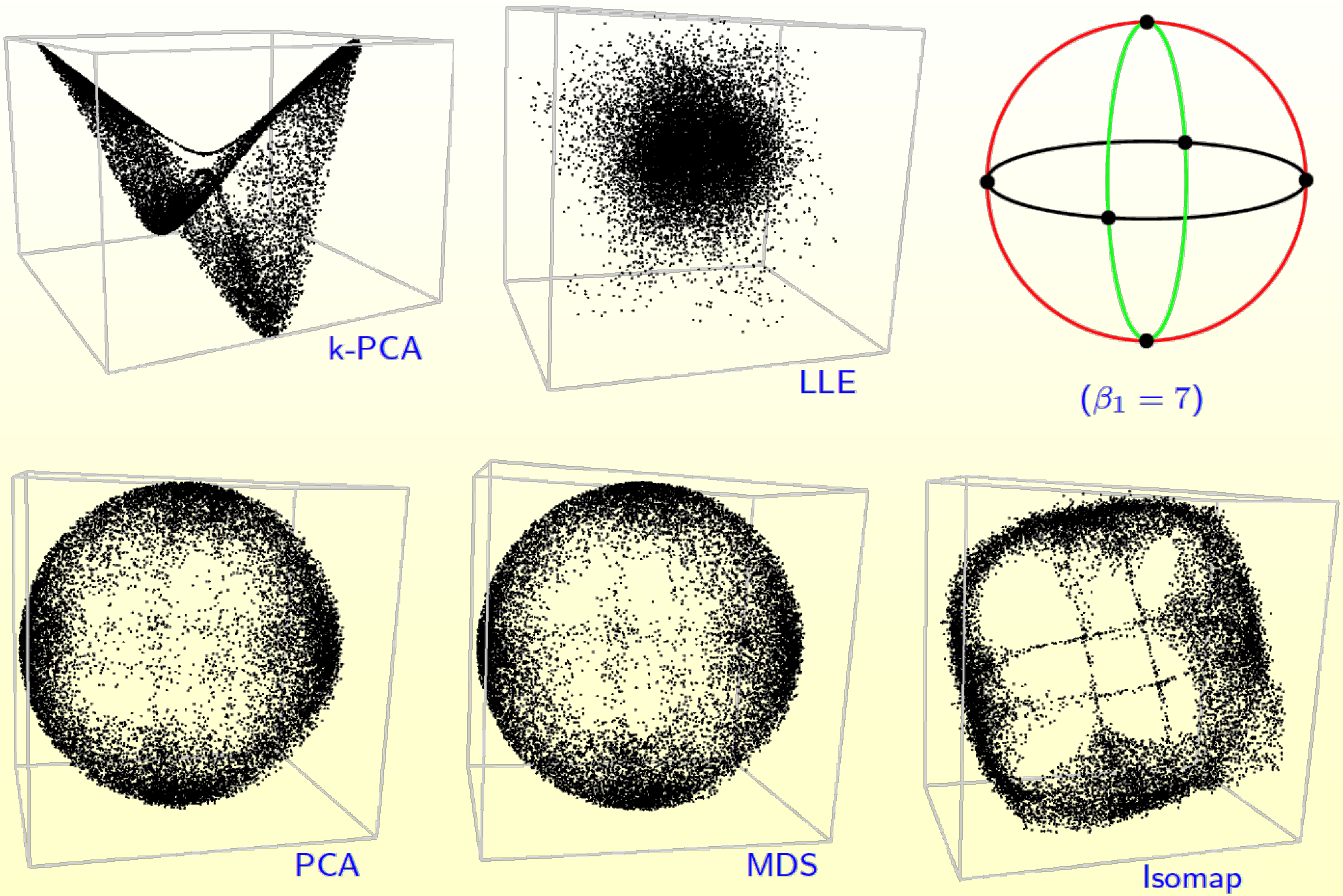
(b) An immersion



$(\beta_1 = 5)$

(source: [Carlsson, Ishkhanov, de Silva, Zomorodian 2008])

FYI, Other Methods



Getting More Out of Topology

Topology for Describing Shape: A Crude Descriptor

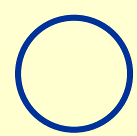
- Topology of the alphabet

F	A	B
$\beta_1 = 0$	$\beta_1 = 1$	$\beta_1 = 2$

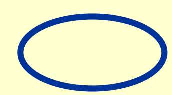
- Problem:
  - Cannot detect sharp features

U	V
$\beta_1 = 0$	$\beta_1 = 0$

  - Cannot detect soft features



$\beta_1 = 1$



$\beta_1 = 1$

Making Topology a Finer Tool

Geometry discriminating

Topology classifying

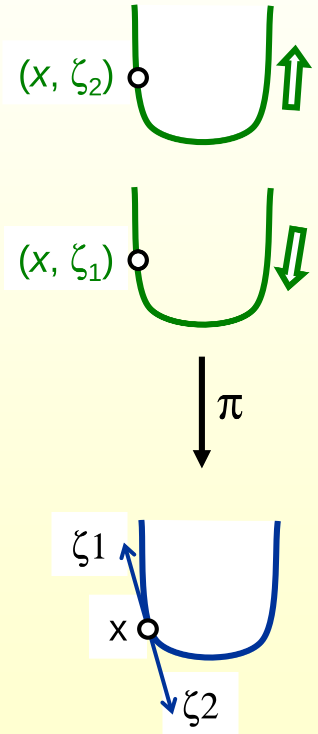
- Topology: connectivity of a space
- Key Idea: no reason to look at the original space only
  - Add geometry $\Rightarrow$ look at derived space(s)
  - Compute topology of derived space(s)

1. Find filtration
2. Compute persistence

via the tangent complex

Our recipe

2-D Curve Tangent Complex



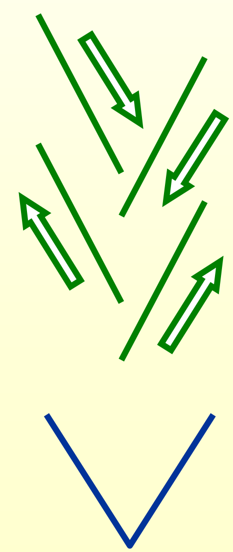
Covering space

T(X) has two components: $\beta_0(T(X)) = 2$

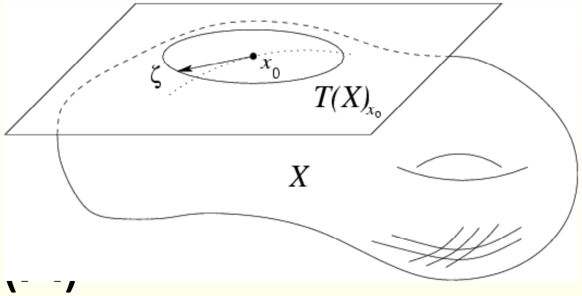
There are two points in its fiber $\pi^{-1}(x)$

Every point x on a smooth curve X has two tangent directions.

A corner point has four tangent directions: $\beta_0(T(X)) = 4$



3-D Curvature-Filtered Tangent Complex

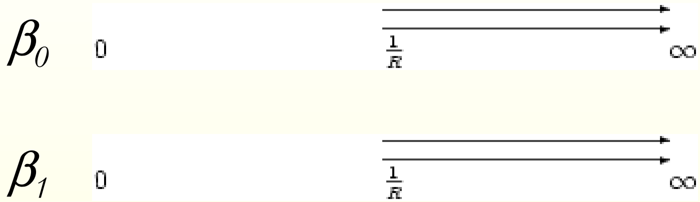


- Derived space
  - $T^0(X)$: space of (point, tangent)
  - Tangent complex $T(X)$: closure of $T^0(X)$
- Filtration by increasing curvature
  - Let $\rho(x, \zeta)$ be the radius of the circle of second order contact
  - $T_\delta^0(X)$: points of $T^0(X)$ with $1/\rho \leq \delta$.
  - $T_\delta(X)$: closure of $T_\delta^0(X)$
- Filtered tangent complex $T^{filt}(X)$ is the family

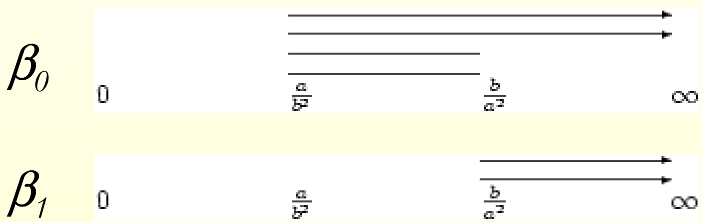
$$\{T_\delta(X)\}_{\delta \geq 0}$$

Persistence Barcodes: Circle vs. Ellipse

T^{filt}(circle of radius R) is simple: the entire complex (2 copies of circle) appears at once, at $\delta = 1/R$.

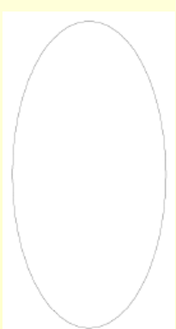


T^{filt}(ellipse) evolves through four stages: points at *lower* curvature appear earlier.

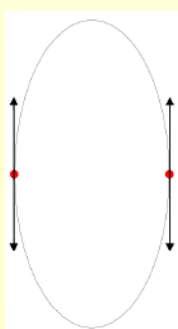


Persistence Barcodes

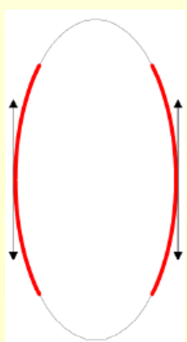
$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1$$



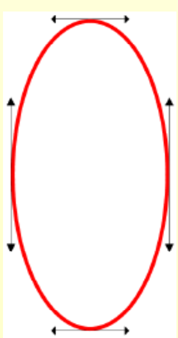
$0 \leq \delta < \frac{a}{b^2}$



$\delta = \frac{a}{b^2}$

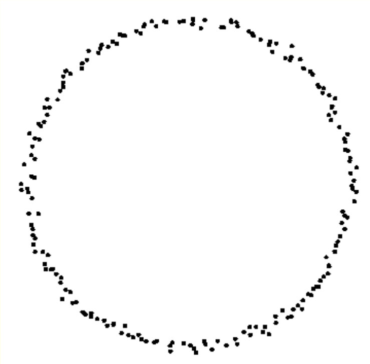


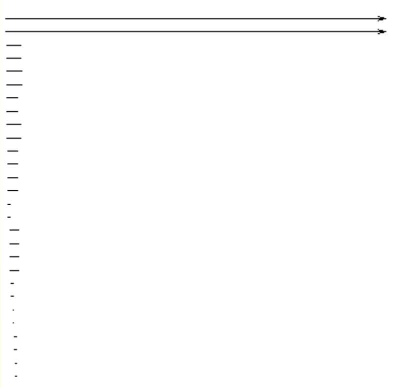
$\frac{a}{b^2} < \delta < \frac{b}{a^2}$



$\delta \geq \frac{b}{a^2}$

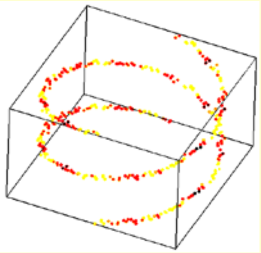
Applying Barcodes to 2D PCDs

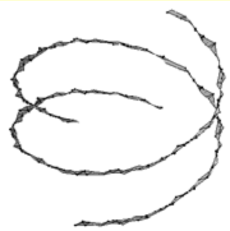




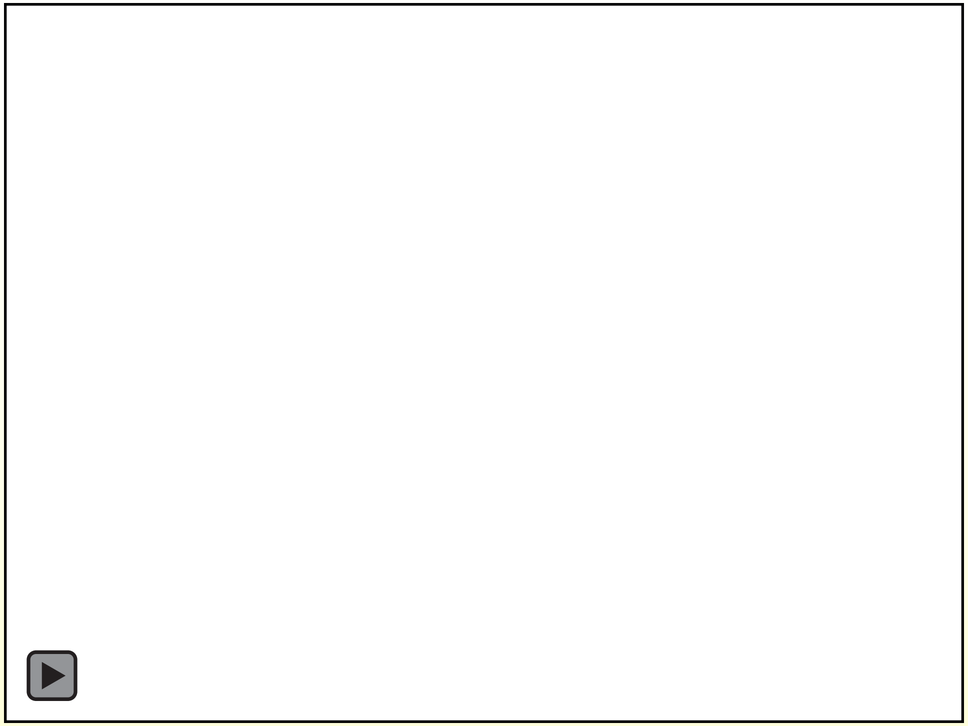
Input: Shape

Output: Descriptor



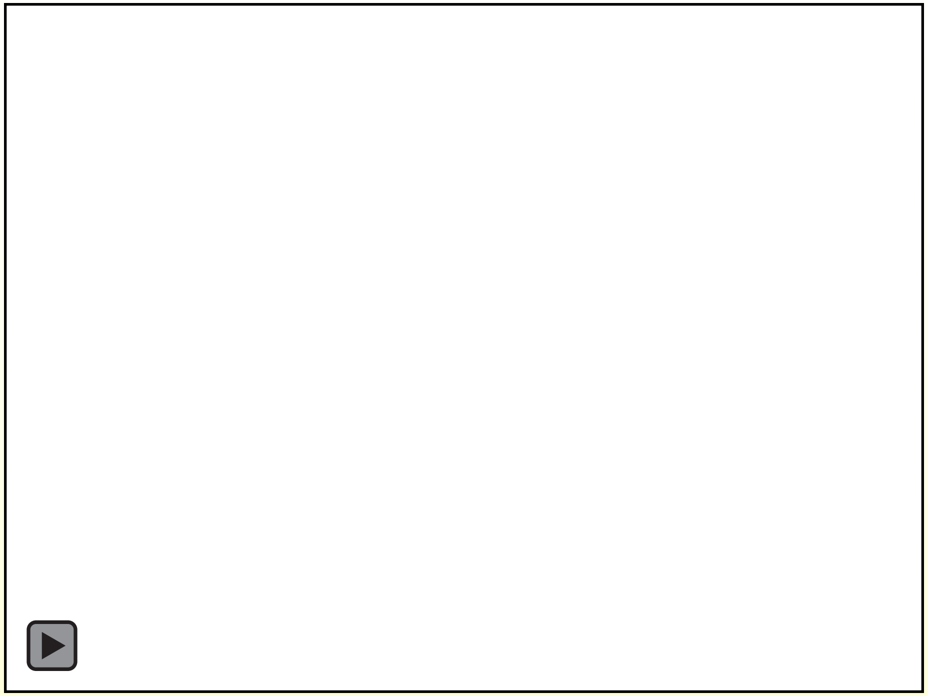


Fibers



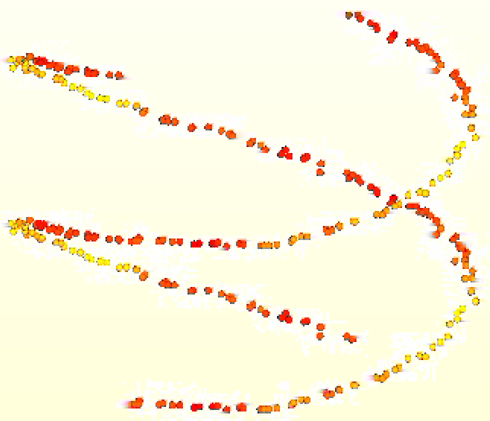
- PCD $P \subseteq X$, sampled from smooth closed 1-manifold
- We compute tangent fibers $\pi^{-1}(P)$ by normal estimation at each point

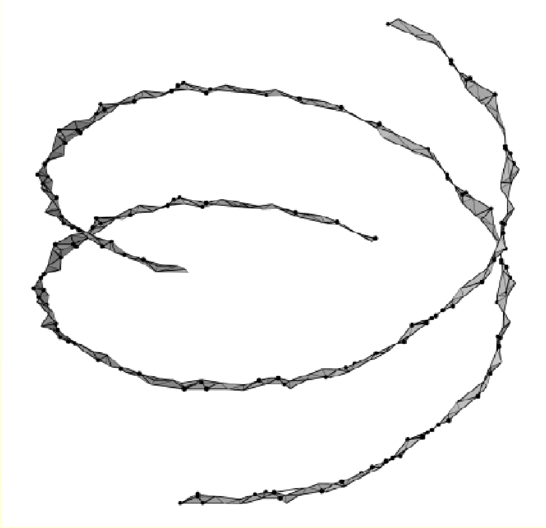
Filtering by Curvature



- Construct tangent complex incrementally
- Transform points to coordinate frame provided by tangent computation
- Fit osculating parabola to estimate curvature (more robust integral methods possible)

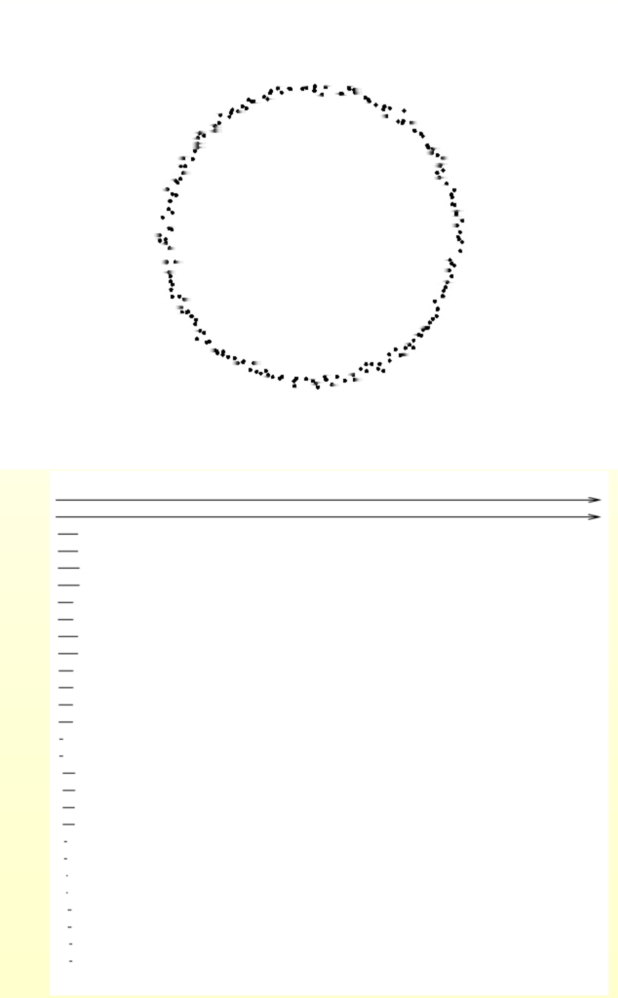
Approximating *T(X)*

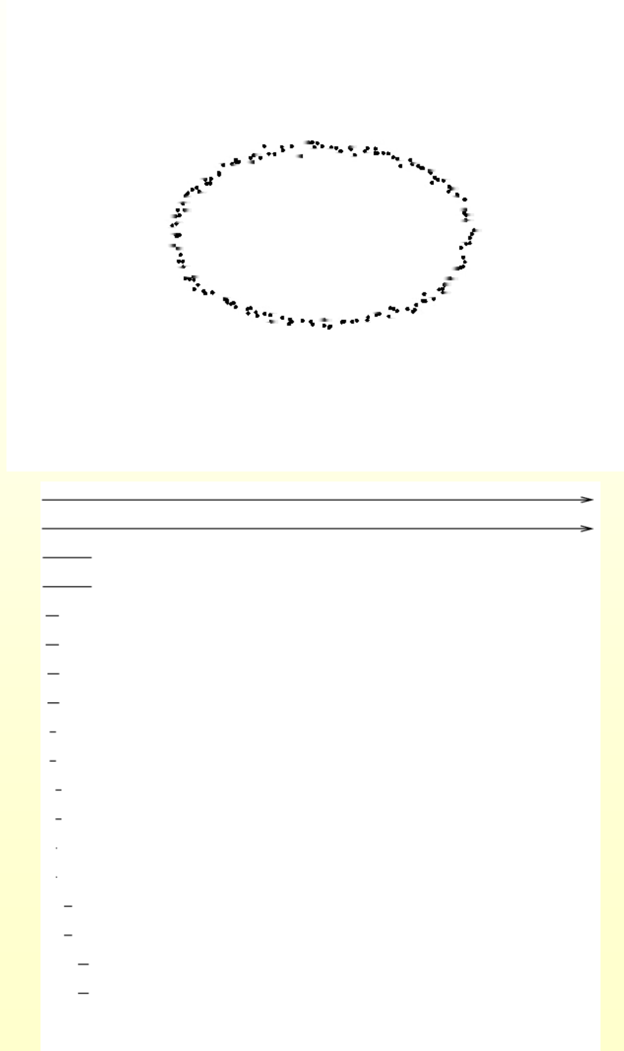


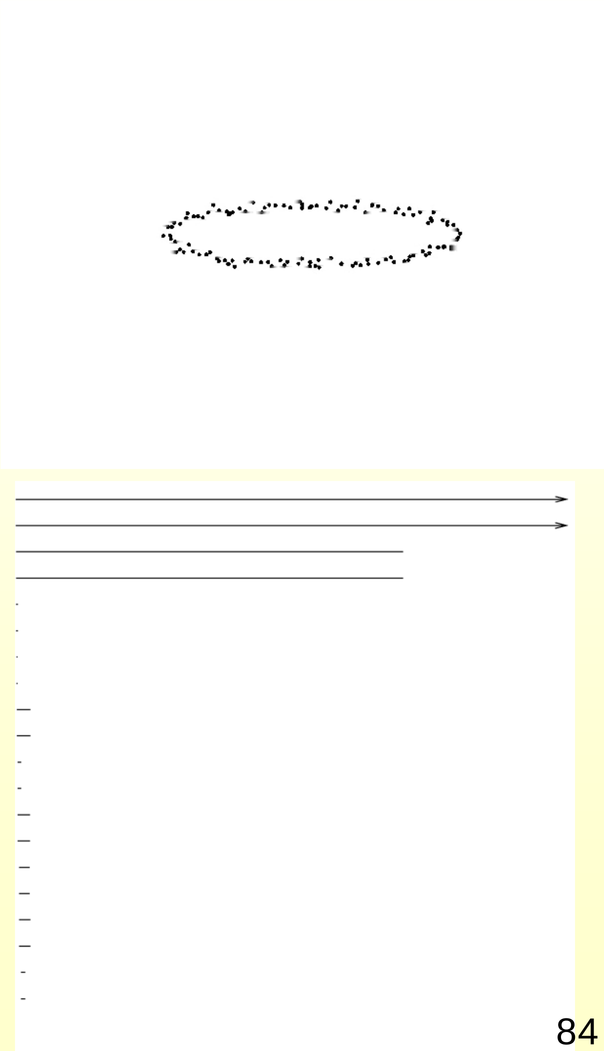


- $\mathbb{R}^n$ § $\mathbb{S}^{n-1}$ with $ds^2 = dx^2 + \omega^2\, d\zeta^2$
- $T(X) \cong \bigcup_{p \in \pi^{-1}(P)} B_\varepsilon(p)$

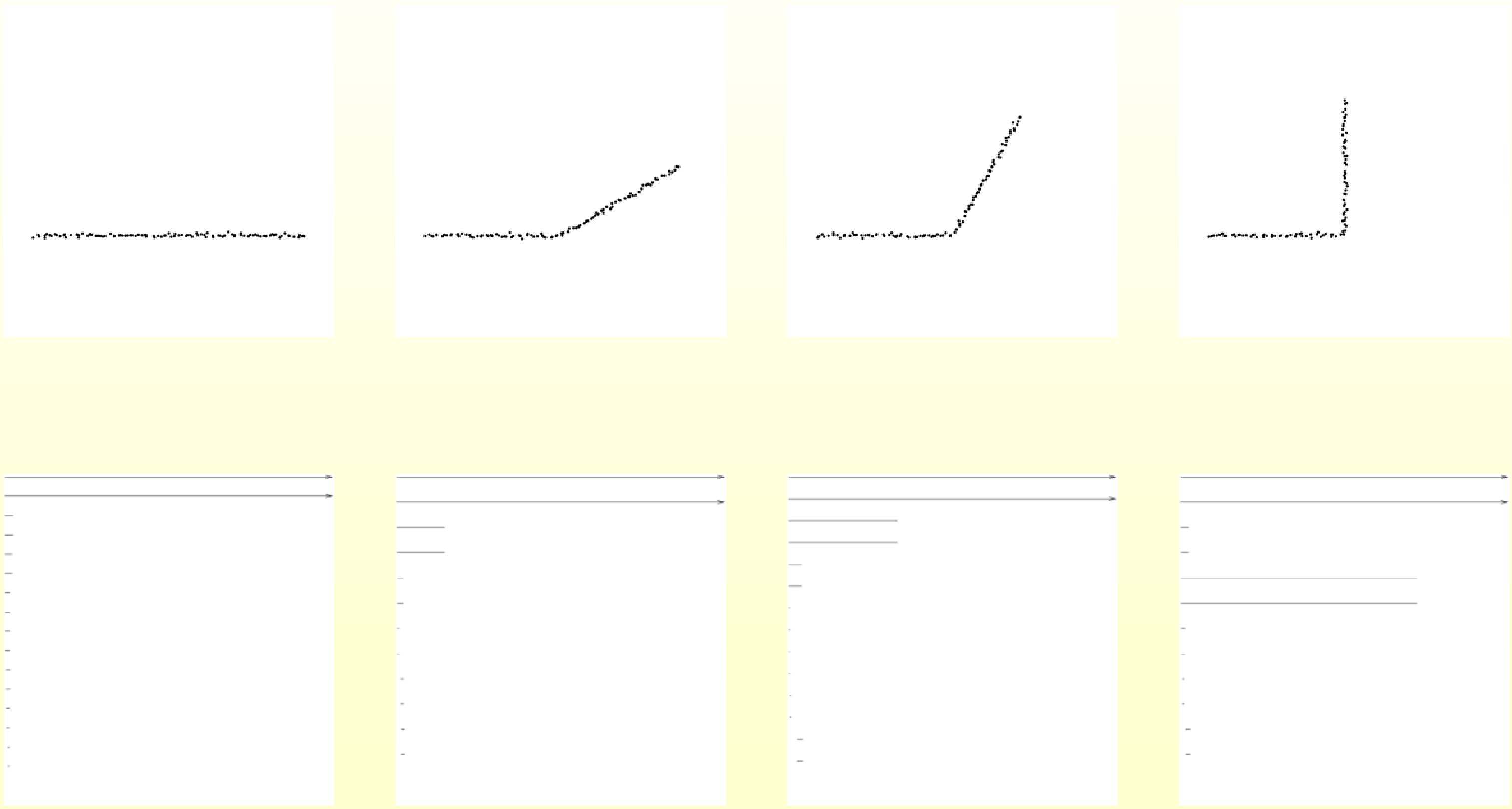
Family of Ellipses







Articulated Arm Parametrization



The Mapper Algorithm

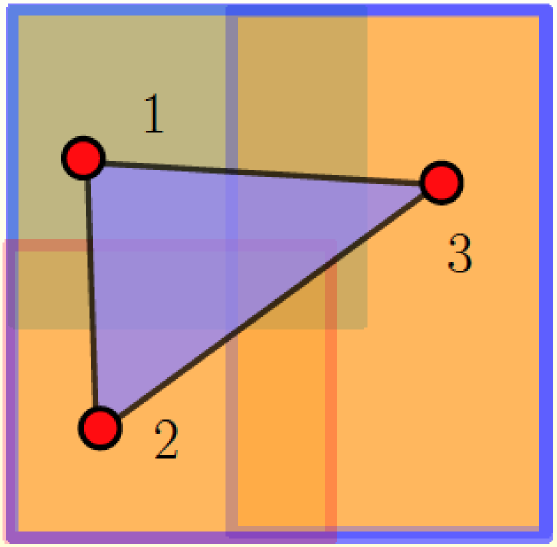
Review: Covers and Nerves

Finite cover of a topological space X

- $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ for a finite index set A.
- each $U_\alpha \subseteq X$ is open and $X = \bigcup_{\alpha \in A} U_\alpha$

Nerve of a cover

- Simplicial complex: $N(\mathcal{U})$ with vertex set A.
- simplices: $A \supseteq \sigma \in N(\mathcal{U}) \Leftrightarrow \bigcap_{\alpha \in \sigma} U_\alpha \neq \emptyset$.



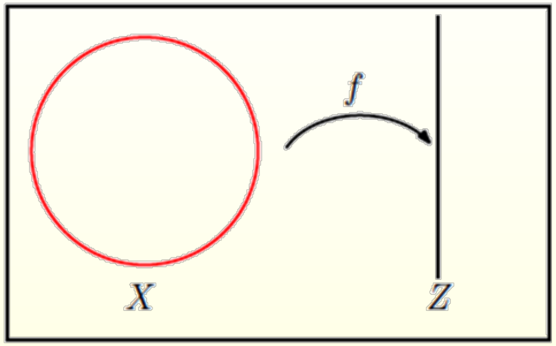
Pullback Covers and Their Nerves

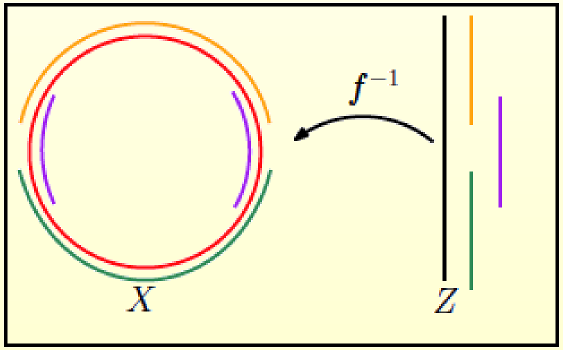
Studying data by looking at "lens" functions over the data

- Assume you have $f : X \to Z$ *well behaved* continuous function and $\mathcal{U} = \{U_\alpha\}_{\alpha \in A}$ finite open cover of Z.
- For each $\alpha \in A$ consider the connected components of $f^{-1}(U_\alpha) = \{V_{i,\alpha},\, 1 \leq i \leq j_\alpha\}$.
- Let $f^*(\mathcal{U})$ be the (finite) open cover of X thus induced:

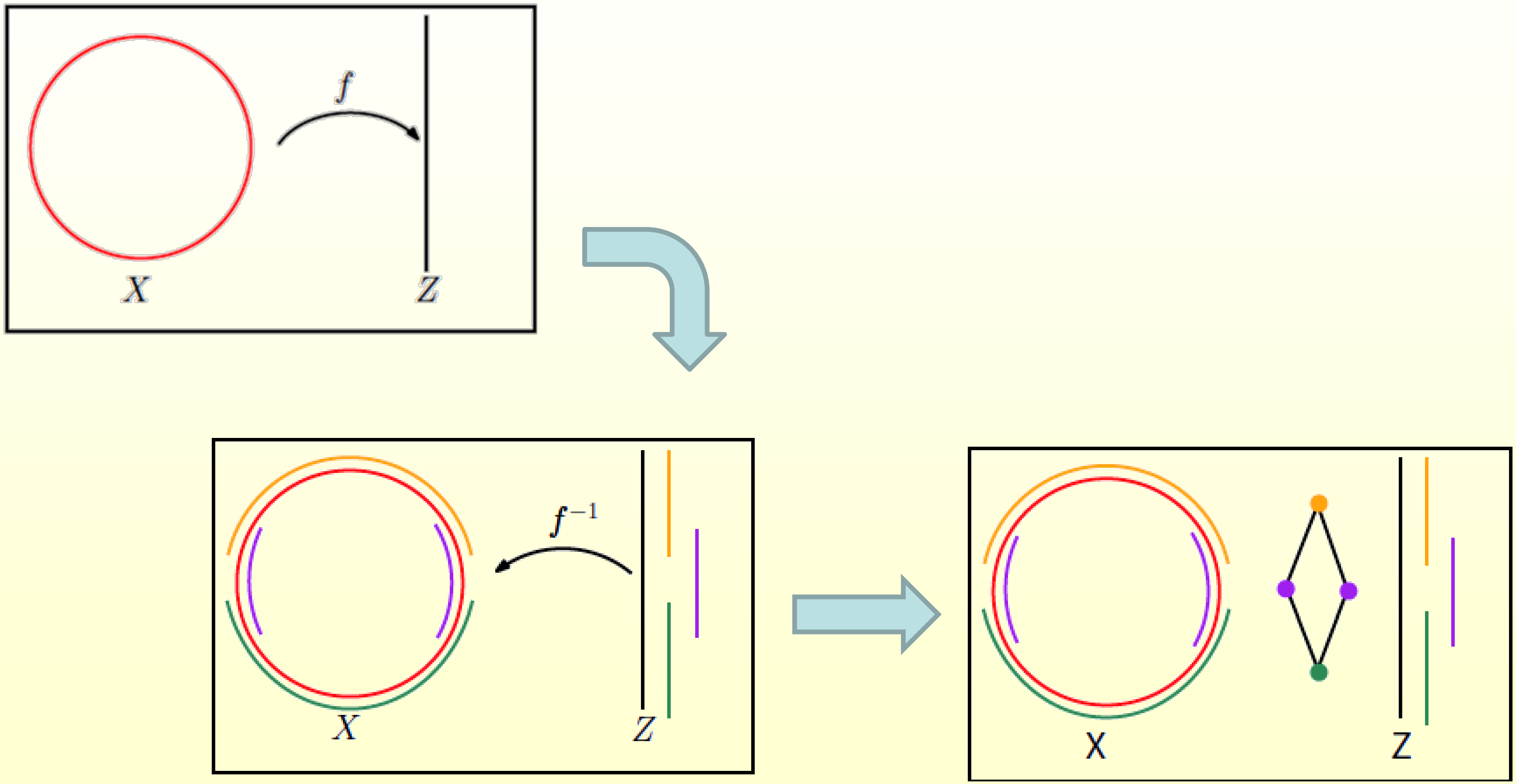
$$f^*(\mathcal{U}) := \{V_{i,\alpha},\, 1 \leq i \leq j_\alpha,\, \alpha \in A\}.$$

  This is the pullback of $\mathcal{U}$ via f.
- Now consider the nerve of the pullback: $N(f^*(\mathcal{U}))$. This complex often retains structural information about underlying space X.

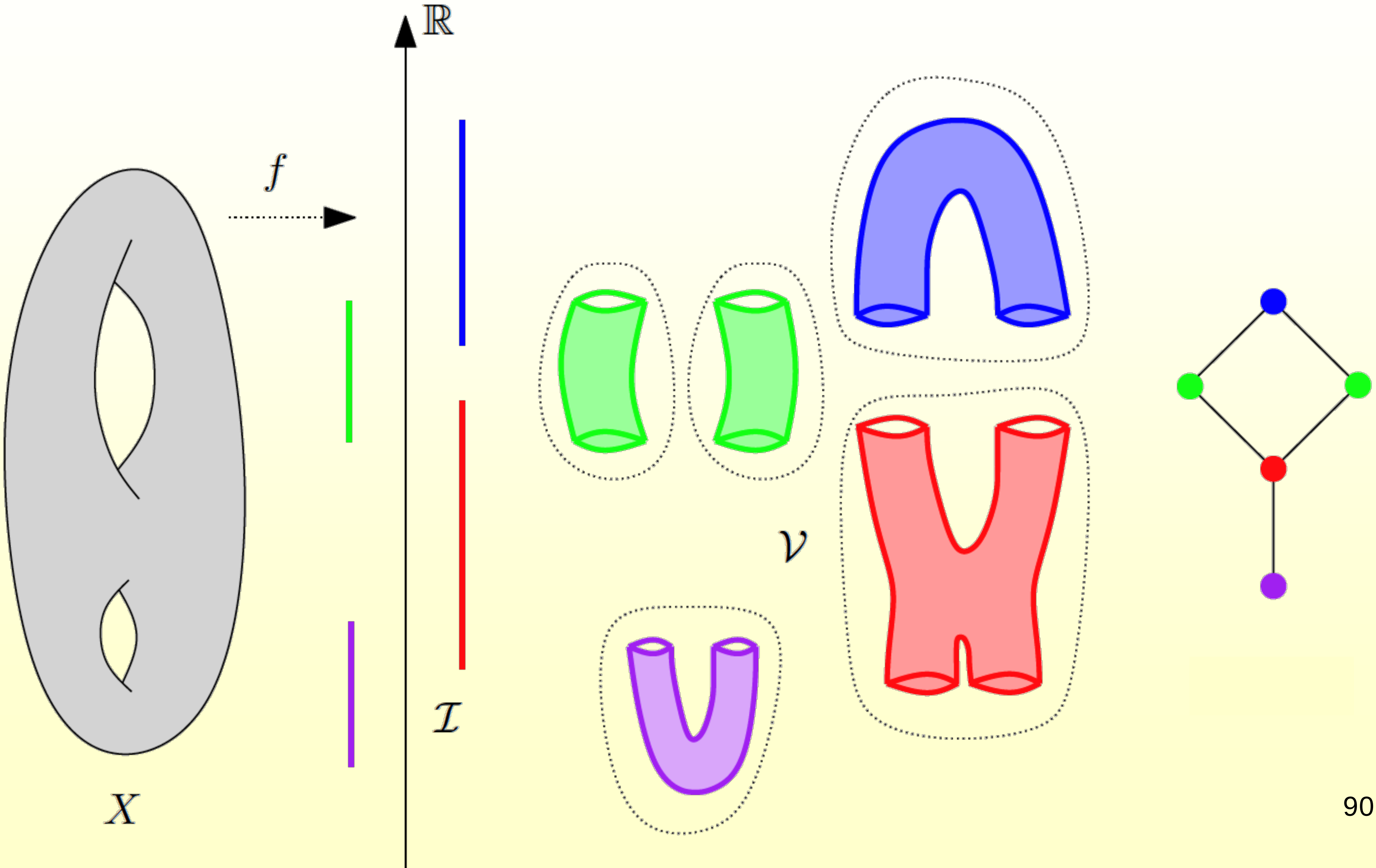




Pullback Covers and Their Nerves



Another Example

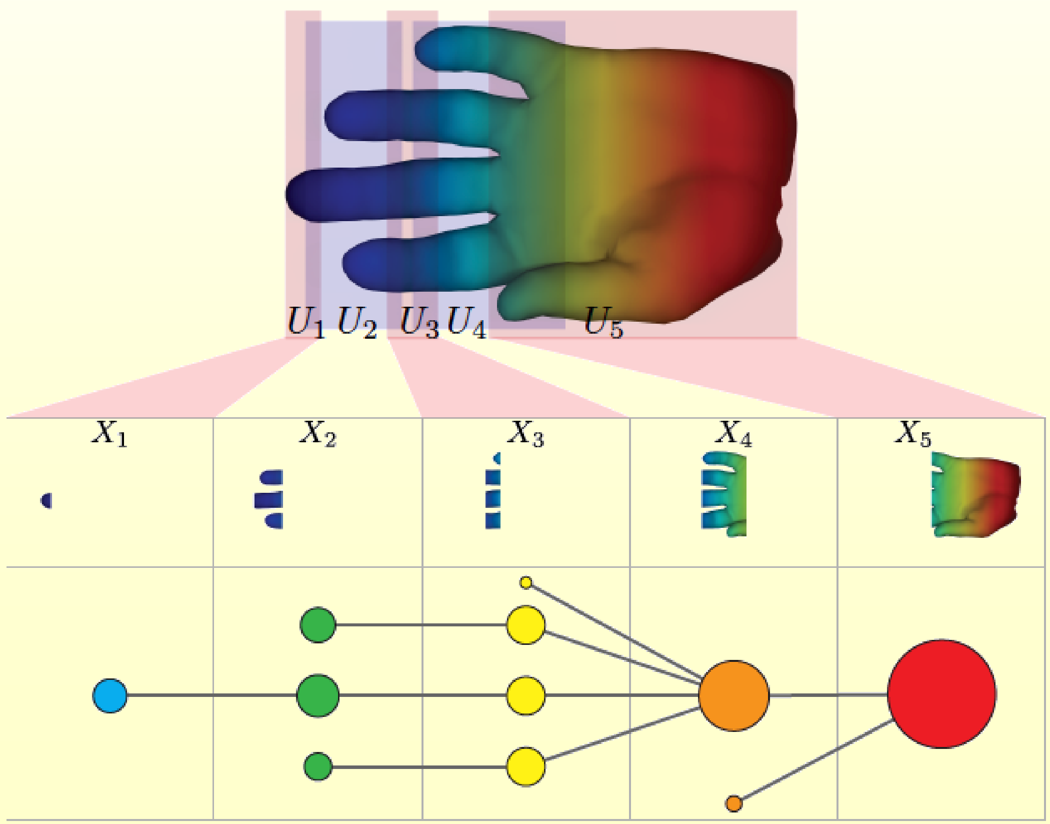


The Mapper Algorithm

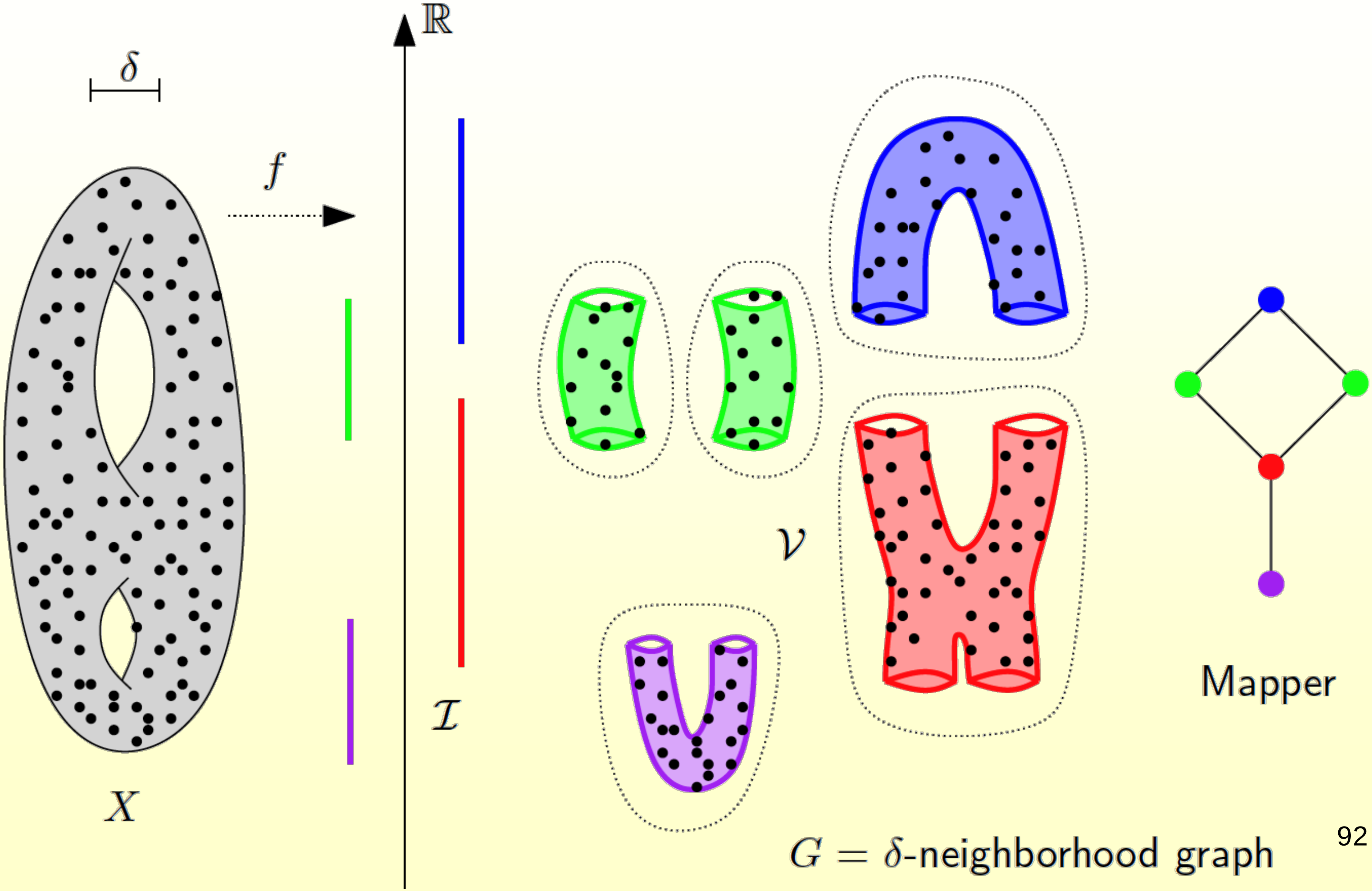
(Carlsson, Mémoli, Singh 2007)

Let $f : X \to Z$ be well behaved and continuous and $\mathcal{U}$ be finite open cover of Z, then the Mapper output corresponding to $\mathcal{U}$ and f is

$$\mathrm{M}(\mathcal{U}, f) := N(f^*(\mathcal{U})).$$

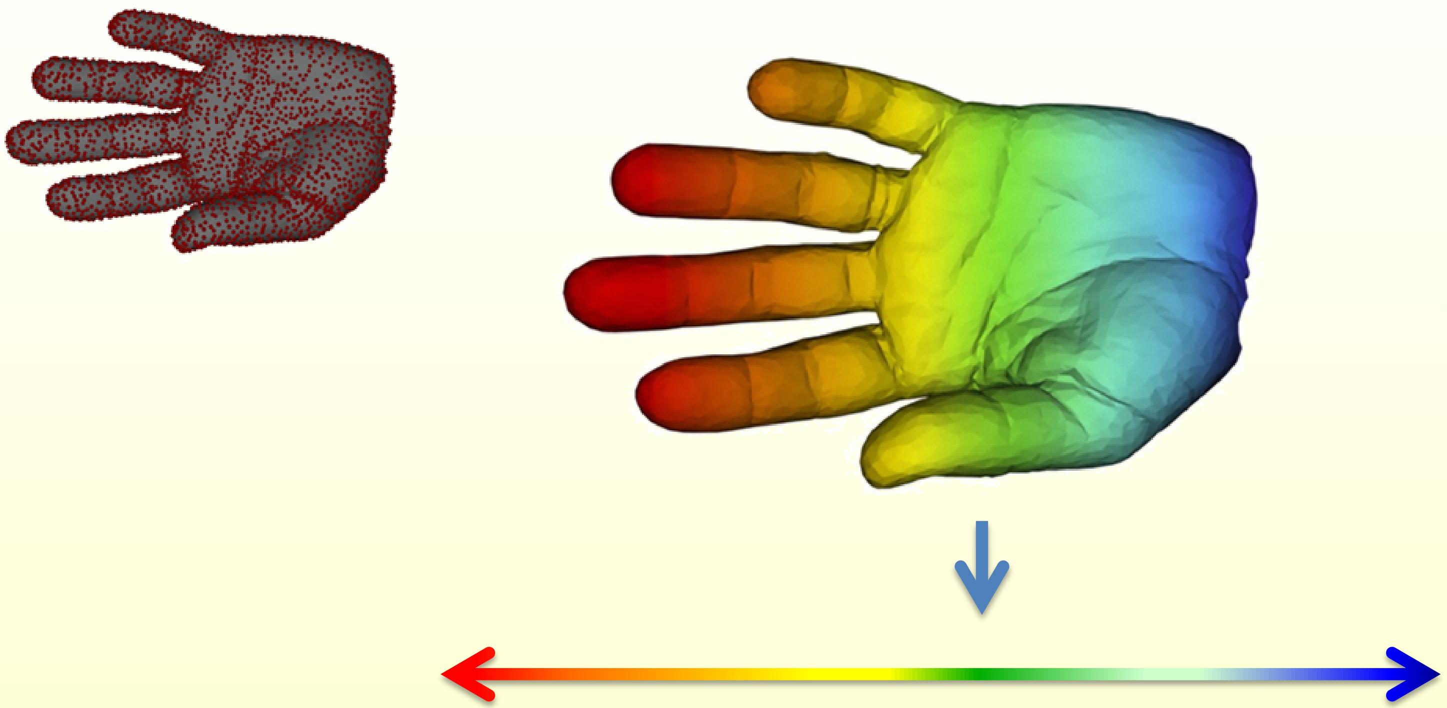


In Practice



$\mathbb{R}$
δ
f
X
$\mathcal{I}$
$\mathcal{V}$
Mapper
$G = \delta$-neighborhood graph

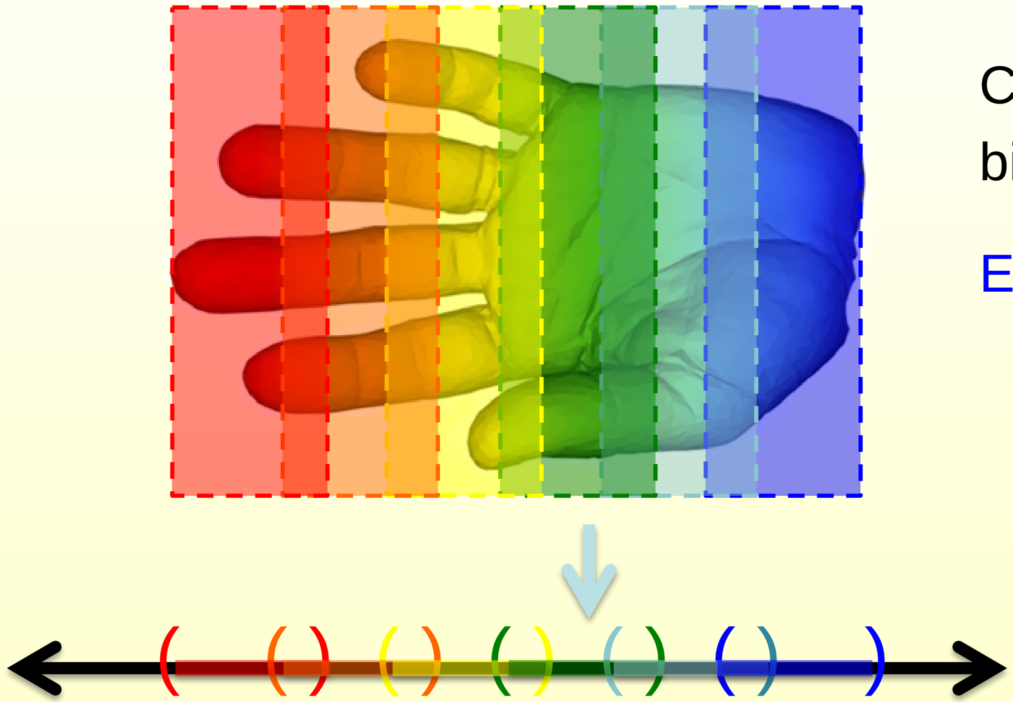
Step 1: Choose a Lens / Filter Function



Function f : Data Set → **R**

Ex 1: x-coordinate f : (x, y, z) → x

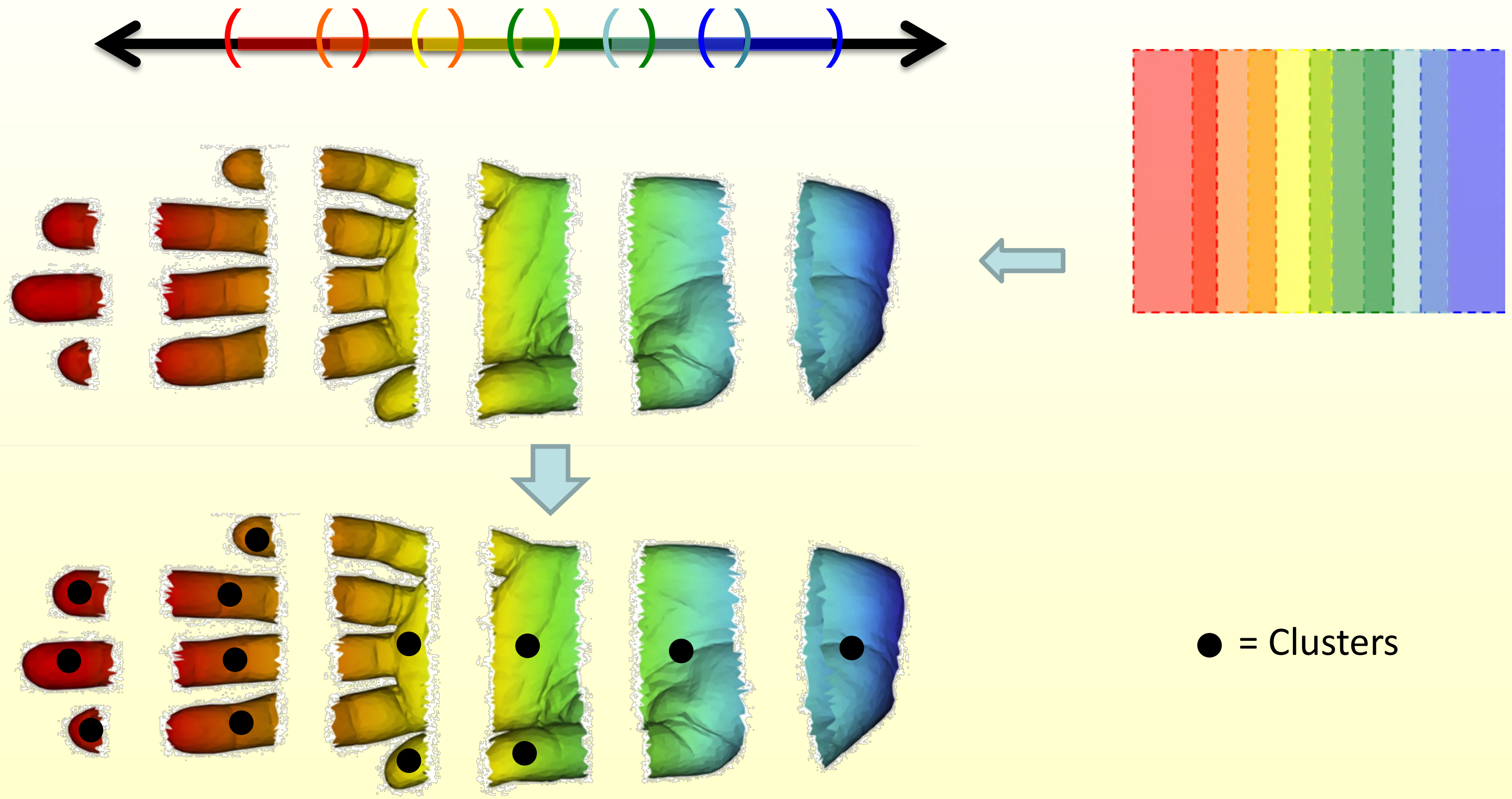
Step 2: Partition into Overlapping Bins



Cover data via overlapping bins.

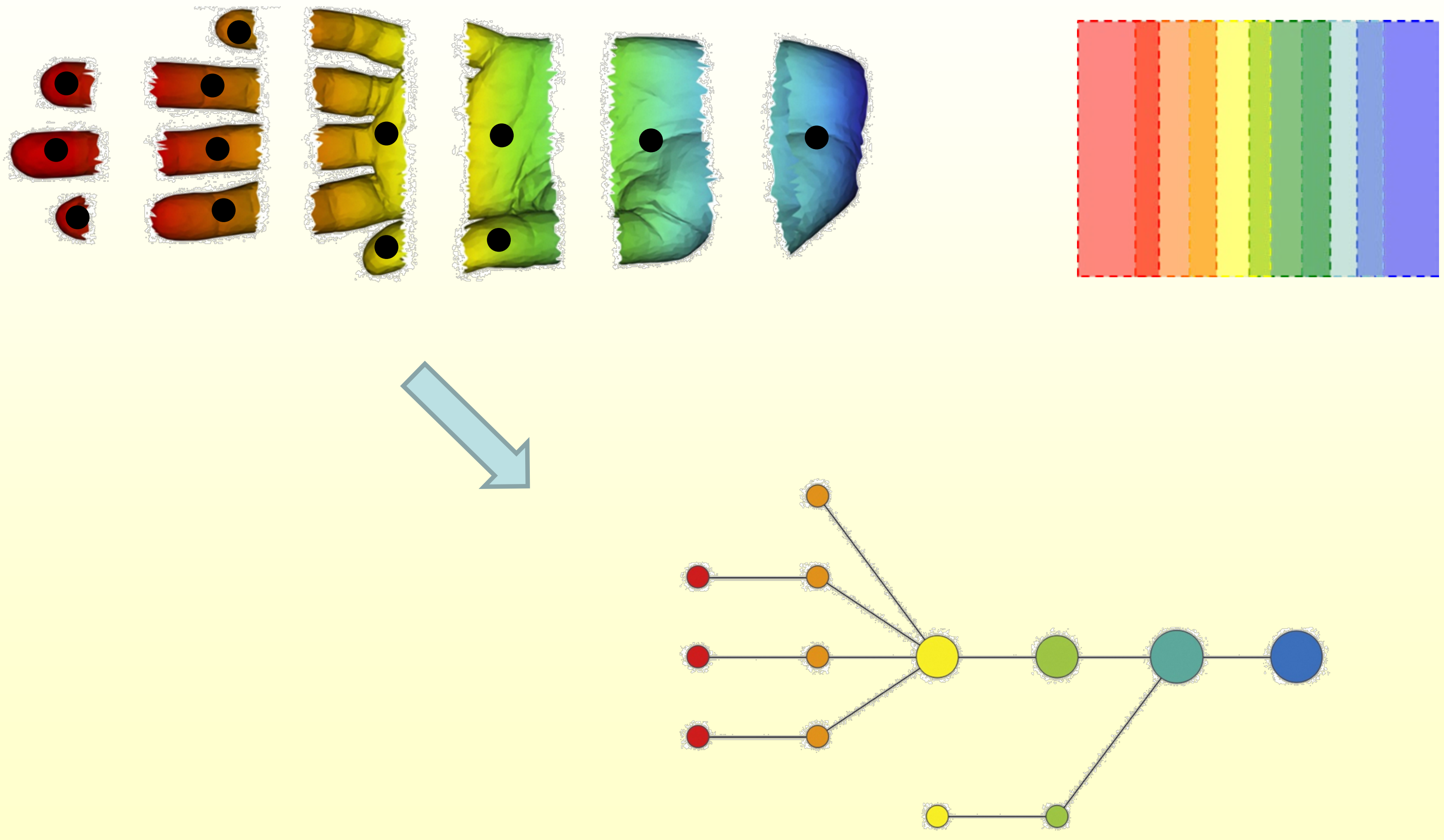
Example: $f^{-1}(a_i, b_i)$

Step 3: Form Connected Components in the Bins

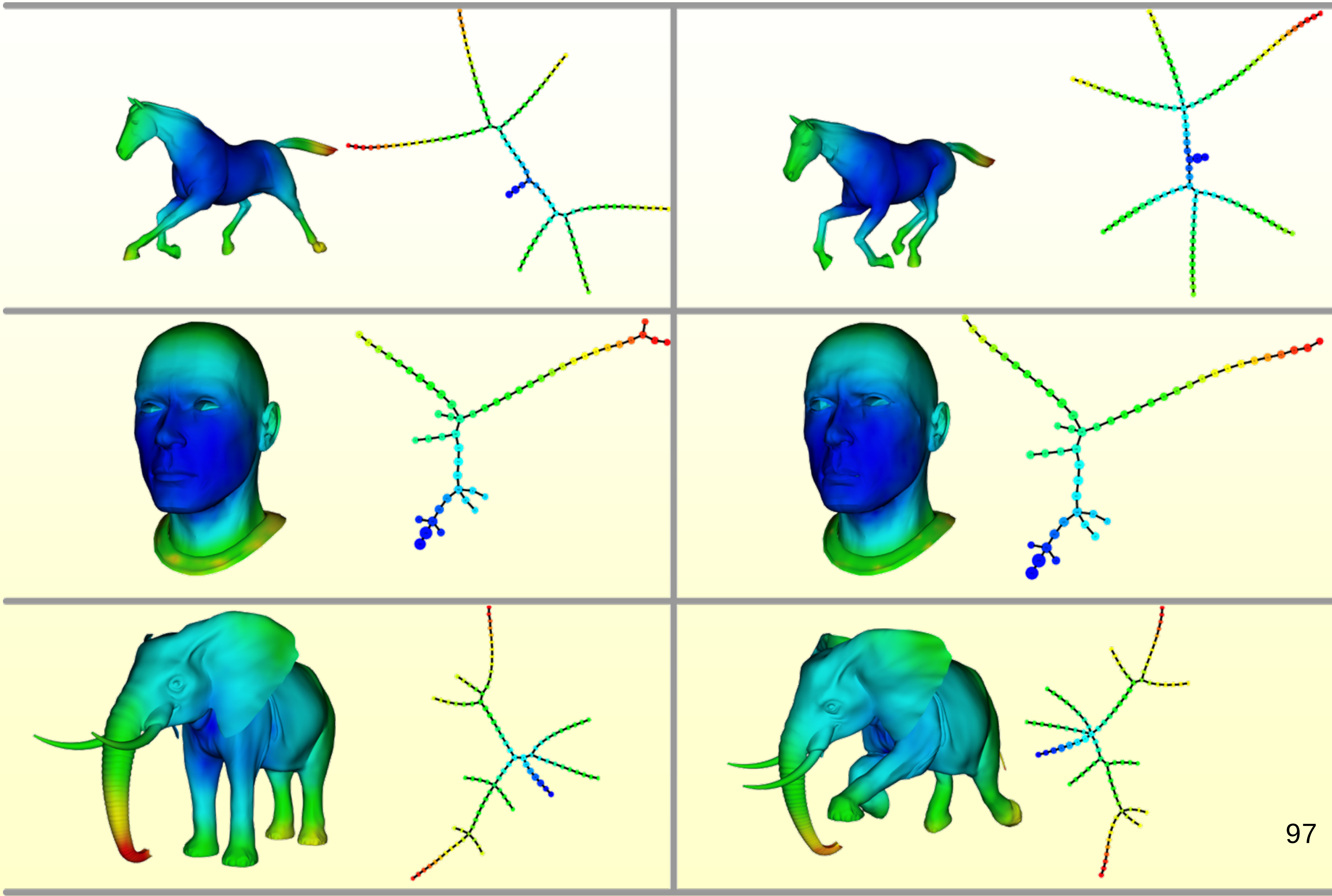


● = Clusters

Step 4: Form a Network of Intersecting Clusters

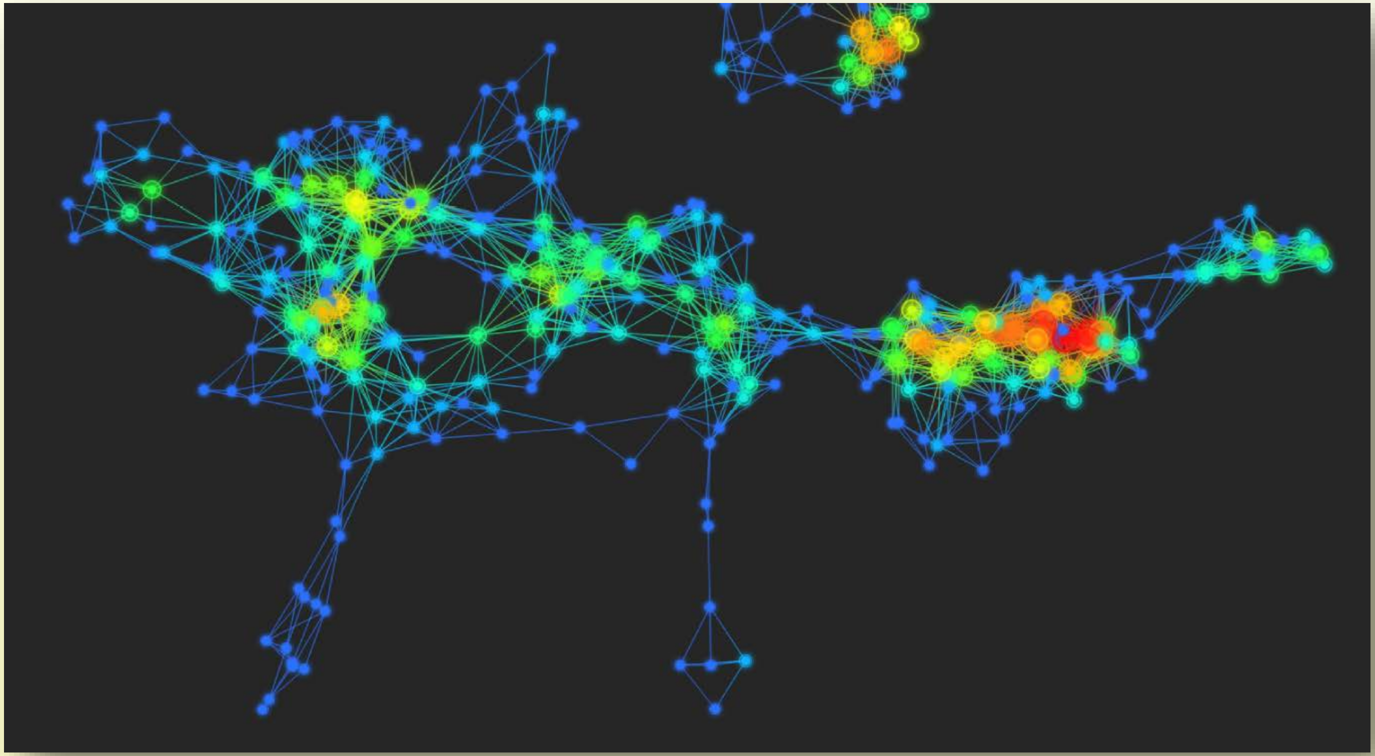


Centrality Filter Under Deformation



Many, Many Choices

"It is useful to think of Mapper as a camera, with lens adjustments and other settings. A different filter function may generate a network with a different shape, thus allowing one to explore the data from a different mathematical perspective."



Persistence-Based Segmentation

3D Shape Segmentation

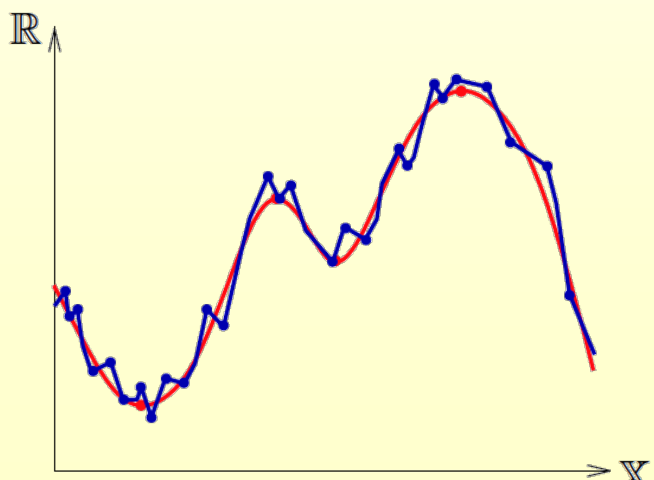
Partition a 3D model into meaningful components

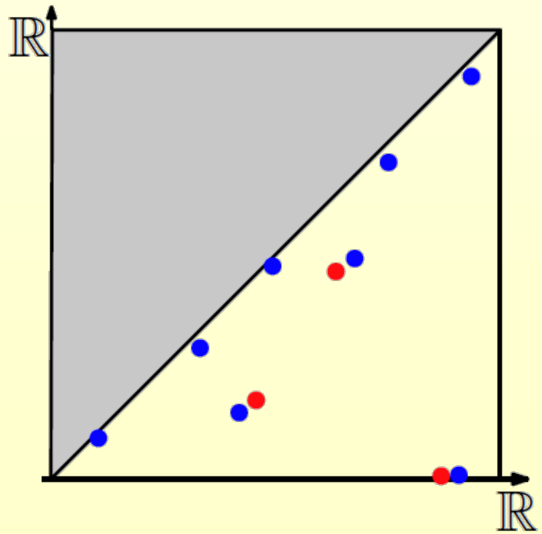
Key Segmentation Method Goals

- Robust to noise
- Intrinsic (invariant to isometric deformations)
- Efficiently computable
- Parametrizable

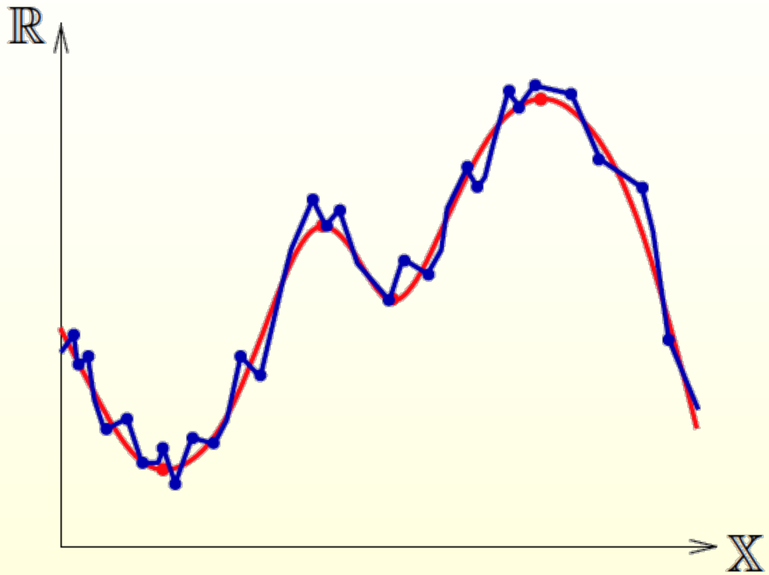
Approach: Use a Filter or Lens Function

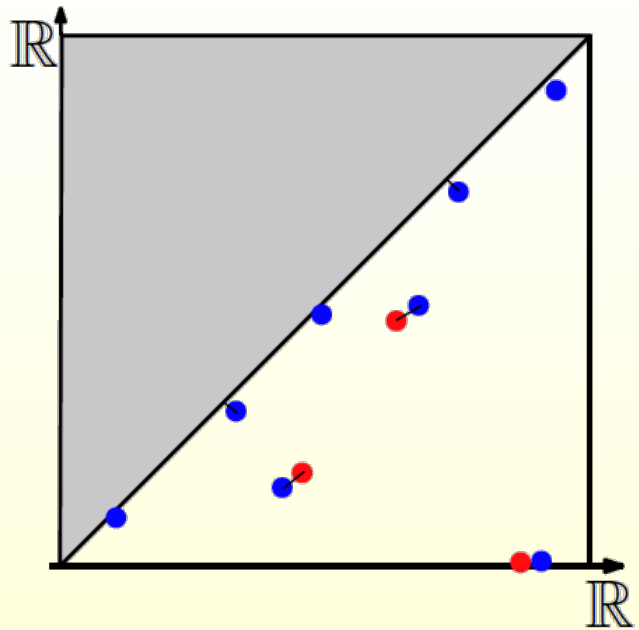
- Unlike Mapper, we want the data to guide us on how to aggregate function values
- Use the persistence diagram of the filter function to guide the segmentation process





Persistence Approximation



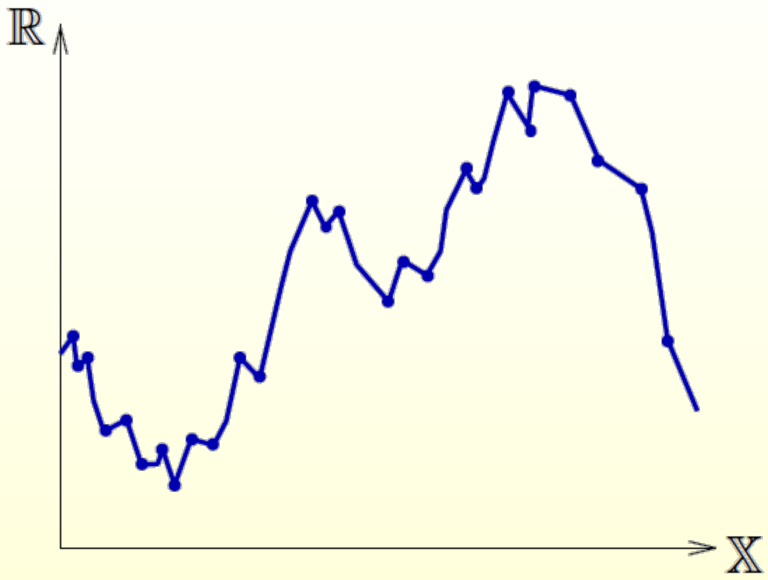


- PD represents the structure of the function
- Stable
  - noise in the function
  - noise in the domain

Bottleneck distance

$$d_B^\infty(D, D') = \inf_{\Phi: D \to D' \text{ multibijection}} \left(\sup_{p \in D} d^\infty(p, \Phi(p)) \right)$$

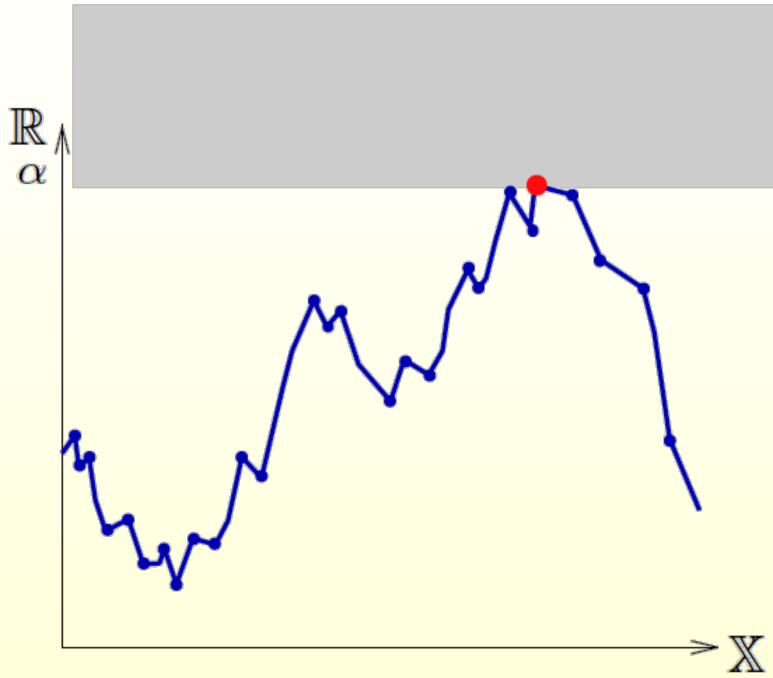
Computing Segments



How do we compute segments from a PD?

- Do not merge segments with persistence less than a threshold τ!

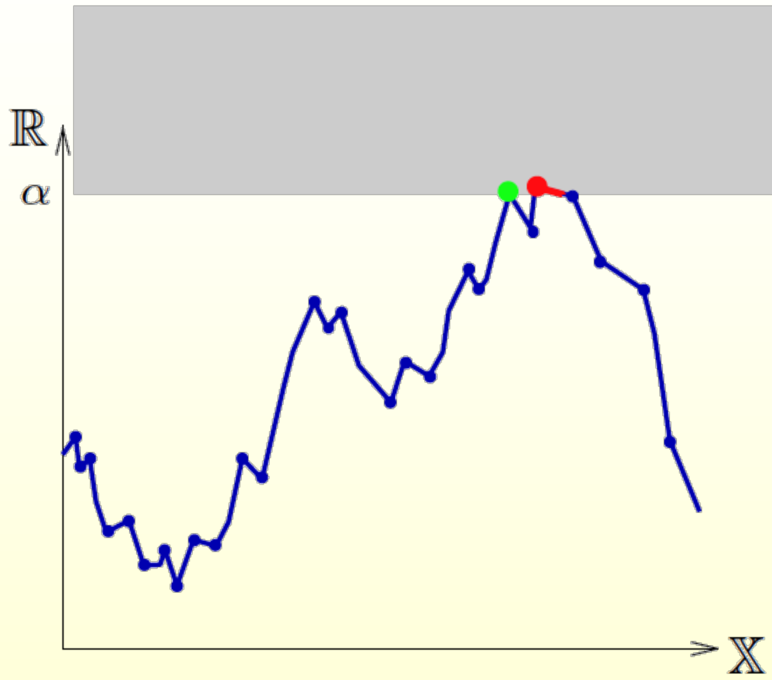
Computing Segments



How do we compute segments from a PD?

- Do not merge segments with persistence less than a threshold τ!

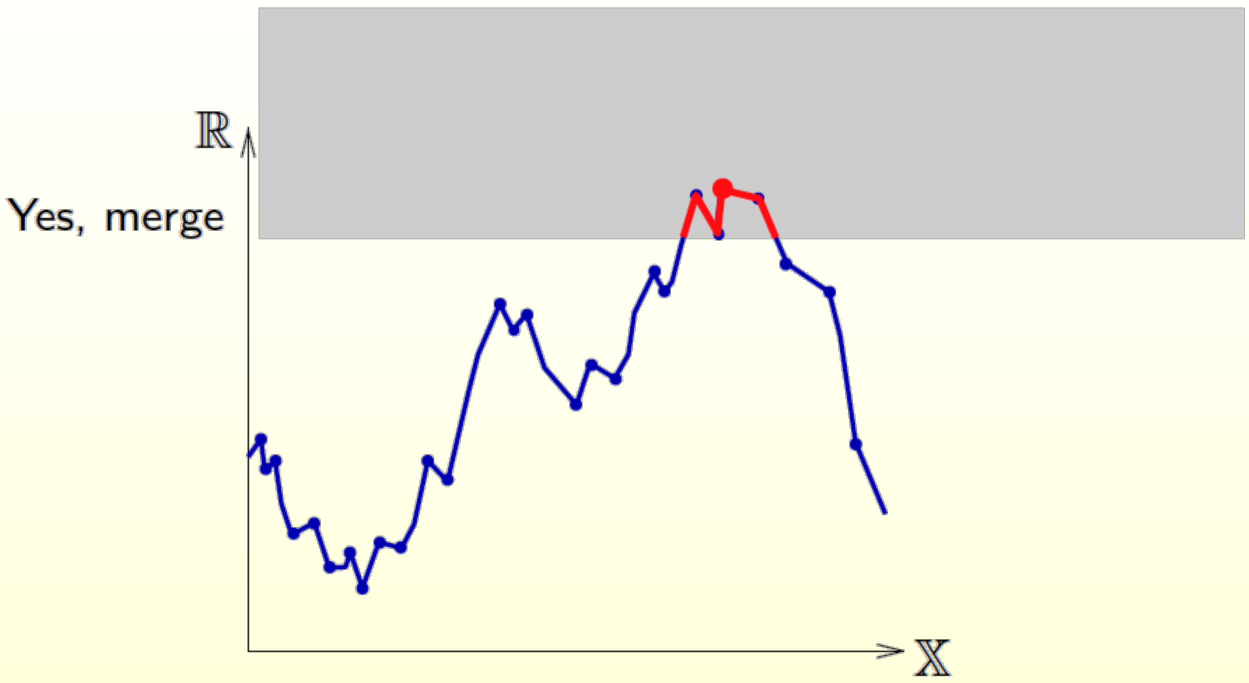
Computing Segments



How do we compute segments from a PD?

- Do not merge segments with persistence less than a threshold τ!

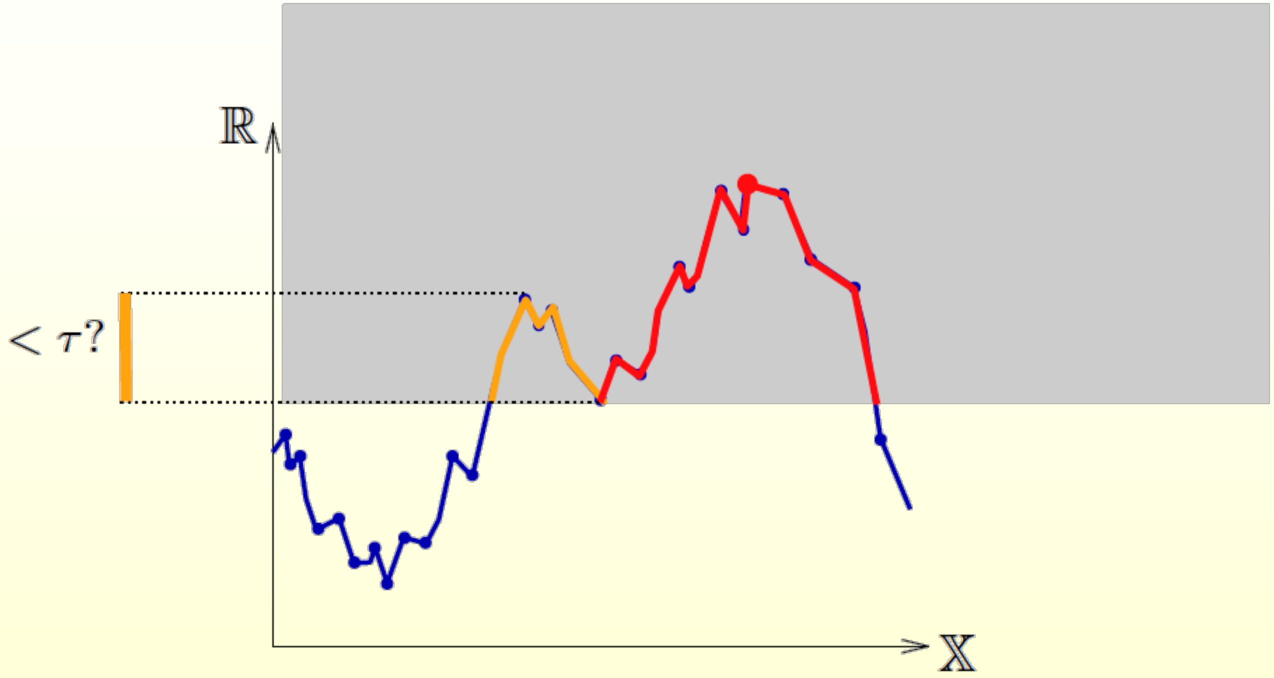
Computing Segments



How do we compute segments from a PD?

- Do not merge segments with persistence less than a threshold τ!

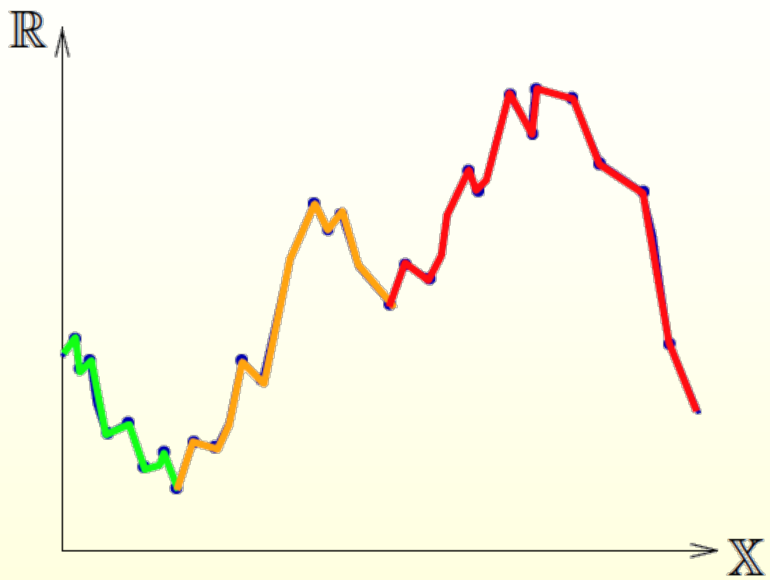
Computing Segments

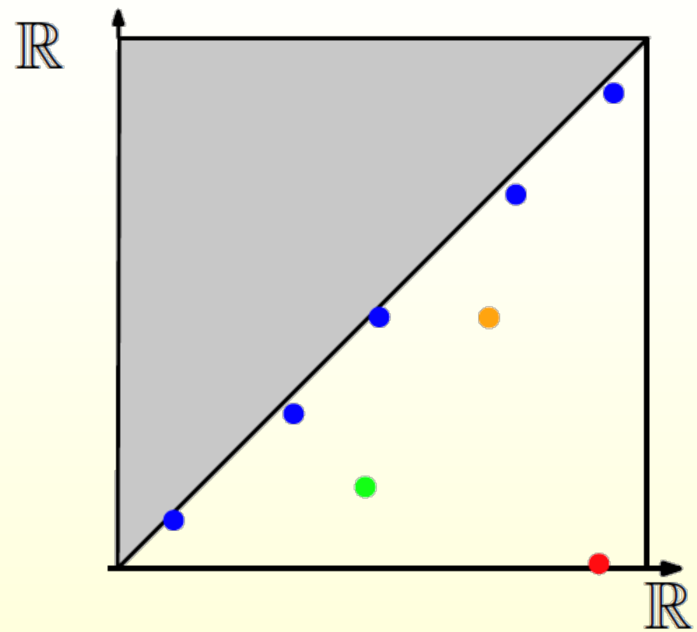


How do we compute segments from a PD?

- Do not merge segments with persistence less than a threshold τ!

Computing Segments





How do we compute segments from a PD?

- Do not merge segments with persistence less than a threshold τ!

Algorithm

- Input: $f(x), \mathcal{M}, \alpha$

1. Sort x according to f
2. For $x \in L$

   2a. For neighbors of x in $\mathcal{M}$

   If no higher neighbors $\Rightarrow$ new cluster

   else assign x to ∇f

   2b. For adjacent clusters y to x

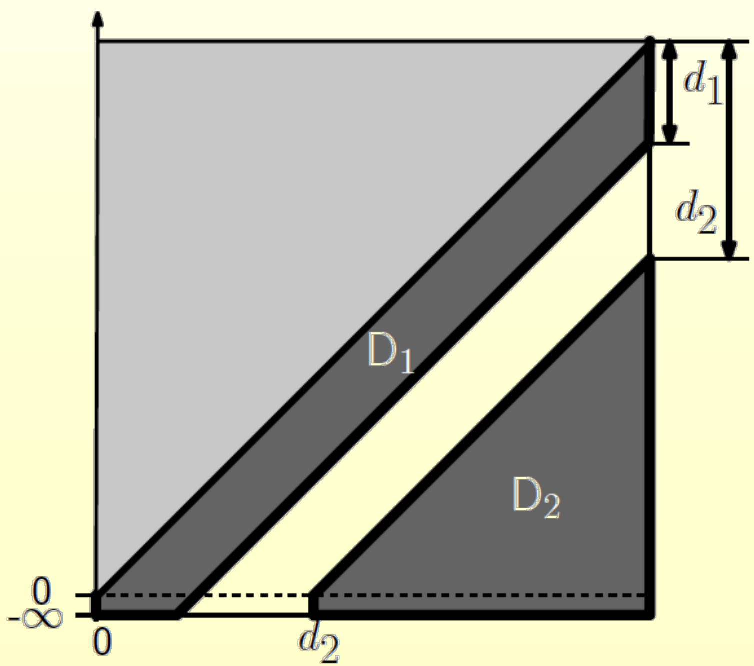
   if $|f(y) - f(x)| \leq \alpha$

   merge into oldest adjacent cluster

Union-find

Interpreting Persistence Diagrams

- If peaks are prominent enough, number of segments is stable
- Theoretically,
  - The number of segments is stable
  - The finer the mesh, the smaller the noise



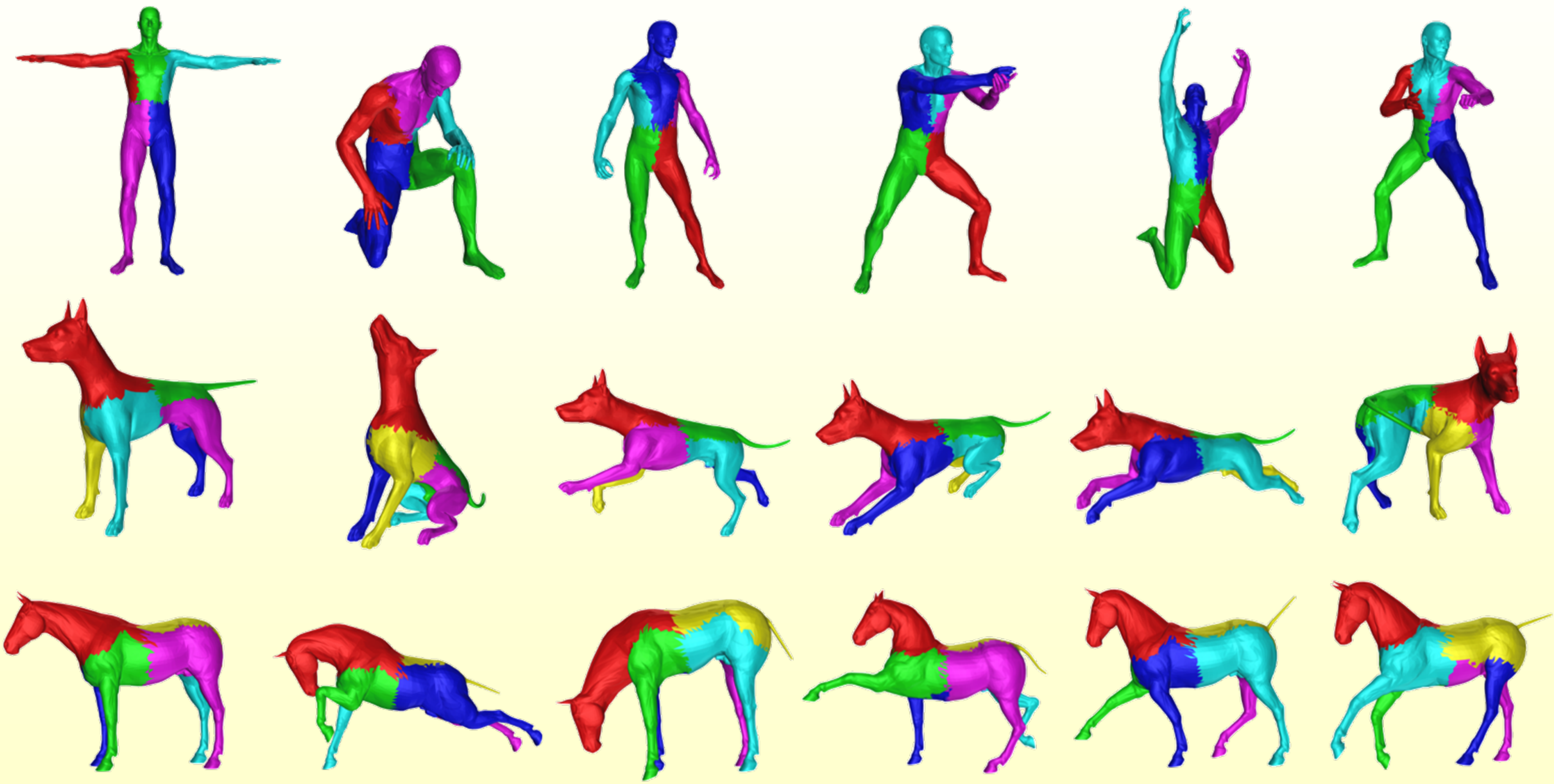
The PD itself can help us decide what the merging threshold should be

Choice of Filter Function is Crucial

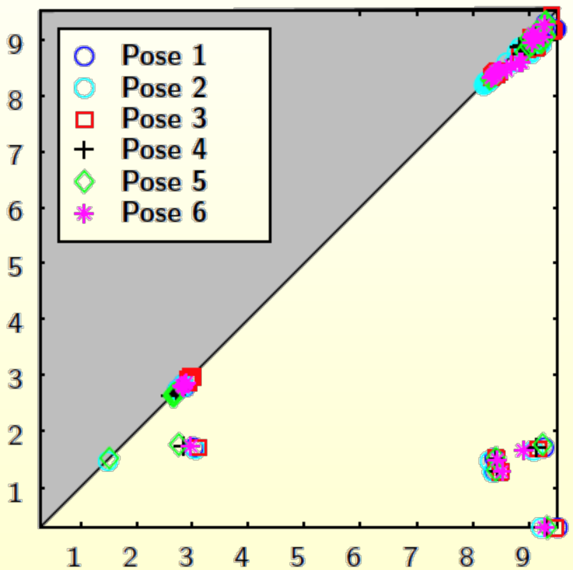
- Ideal function should be
  - Stable under perturbations
  - Invariant under rigid and isometric deformations
  - Informative: local maxima should correspond to segments
  - Efficiently computable

- Use heat kernel signature (HKS) or wave kernel signature
  - These are functions obtained from solving certain partial differential equations on the surface of a 3D shape
  - More later …

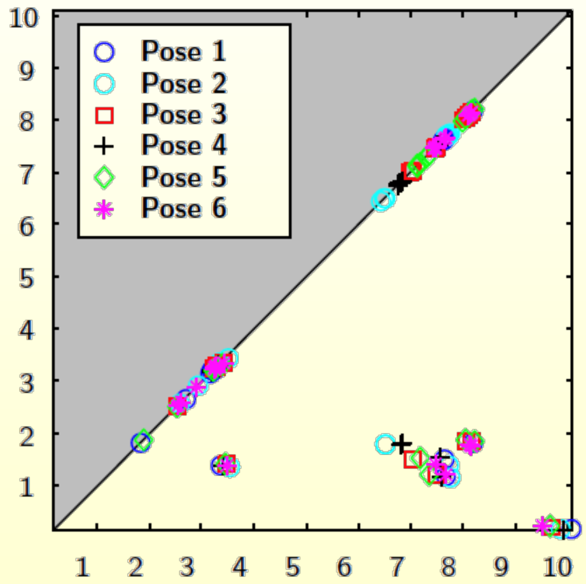
Segmentations



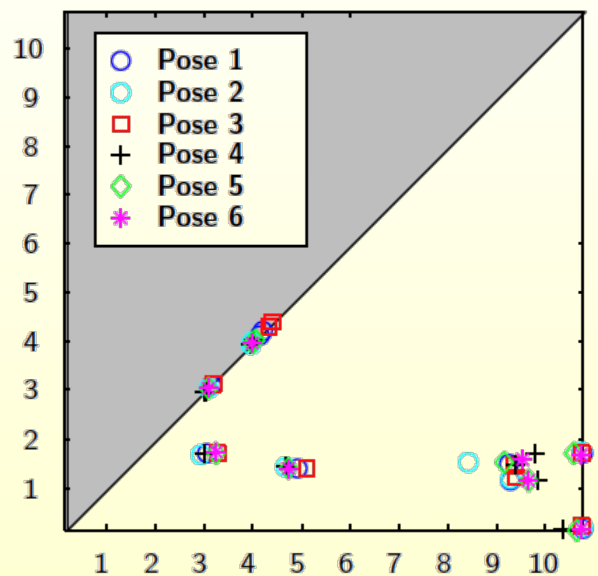
Stable Diagrams



Human



Dog



Horse

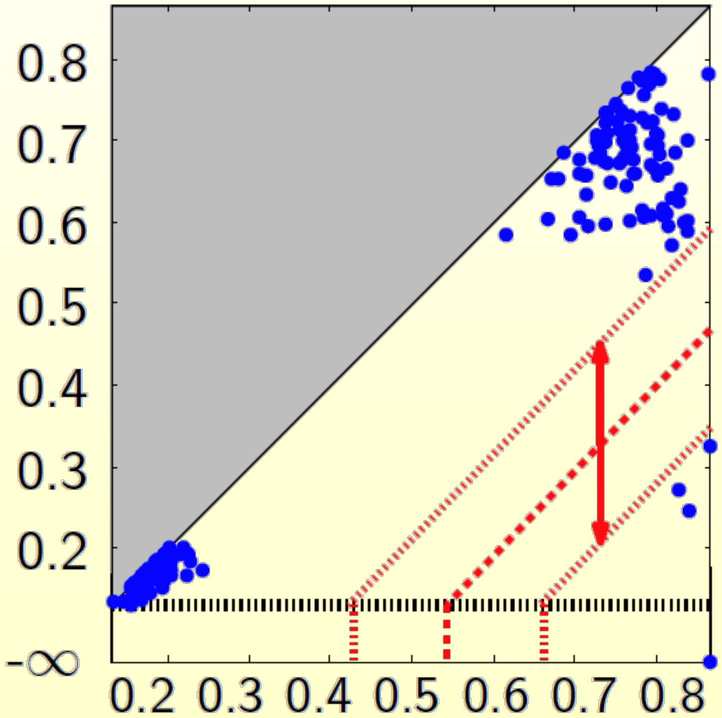
Caveats

- No single function is likely to be truly informative
- Regions is which a function is featureless create inherently unstable regions
  - Possible solution: perturb the mesh and look for stable regions
    - Identify segments stable under perturbations
    - Treat unstable regions separately

Extended Algorithm

1. Run the algorithm to obtain persistence diagram

2. Choose threshold and perturbation amount



Extended Algorithm

1. Run the algorithm to obtain persistence diagram

2. Choose threshold and perturbation amount

3. For $i = 1 \ldots N$

    a. Perturb function values

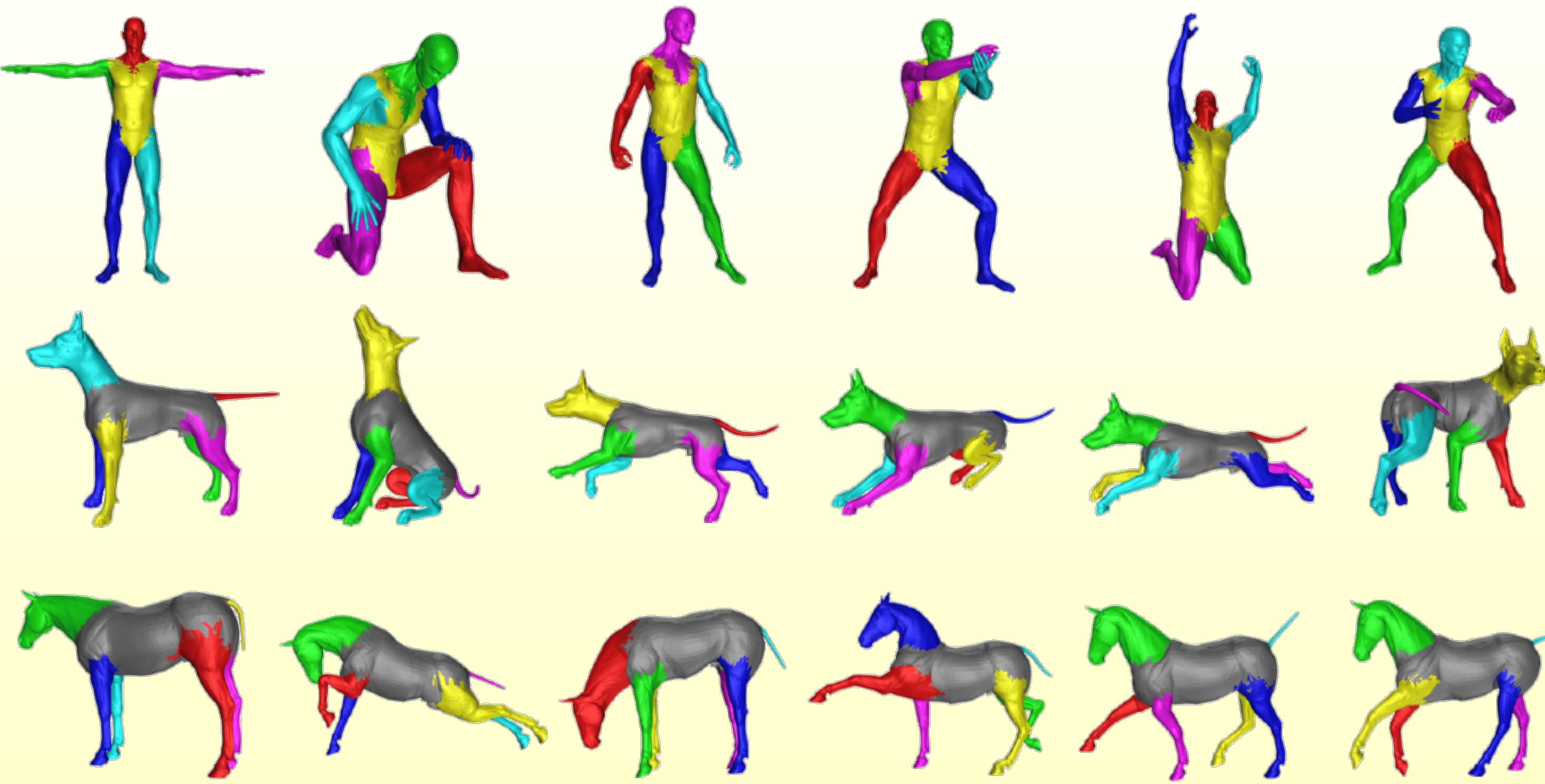
    b. Run clustering algorithm

    c. Find one-to-one correspondance between segments

4. Find stable and unstable parts

    Each point has a distribution over possible segments

Improved Results



Scalar Field Analysis

Scalar Field Analysis

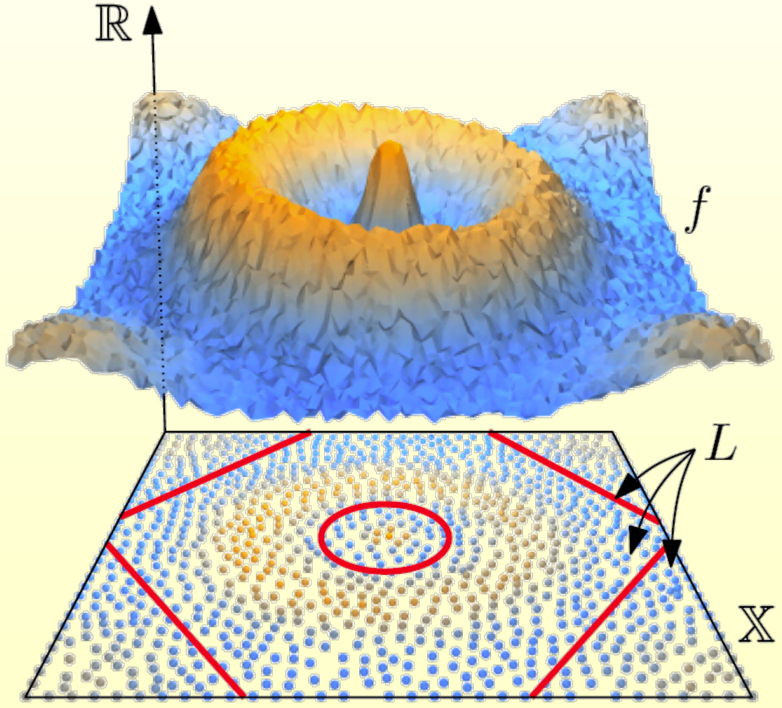
Setting: topological space $\mathbb{X}$, $f : \mathbb{X} \to \mathbb{R}$

Input: a finite sampling L of $\mathbb{X}$, the values of f at the sample points

- assuming f is smooth (Lipschitz condition)

Goal: Analyze landscape of graph(f):

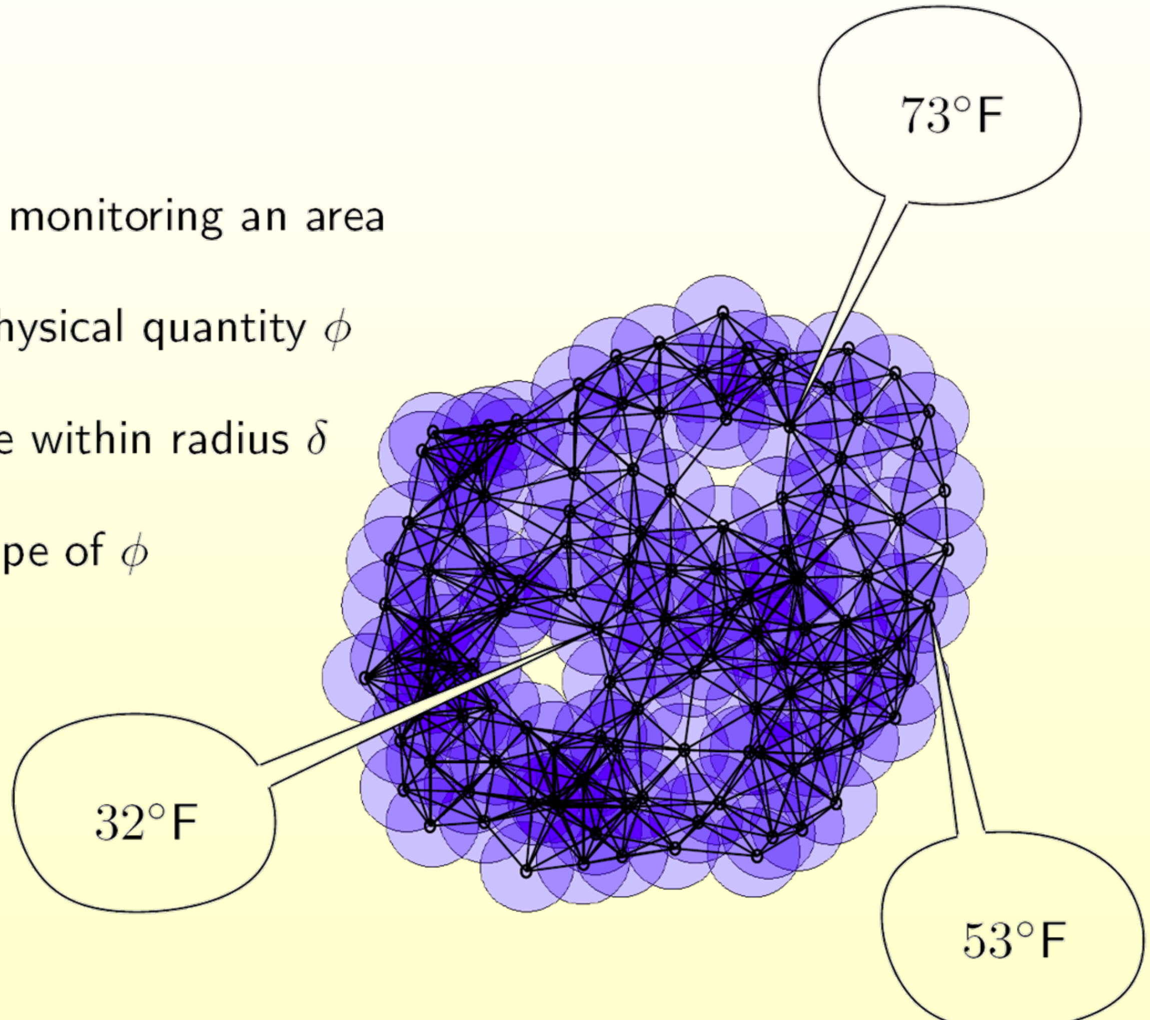
- prominent peaks/valleys
- basins of attraction
- in the presence of noise
- without explicit knowledge of the sample positions



Motivating Applications

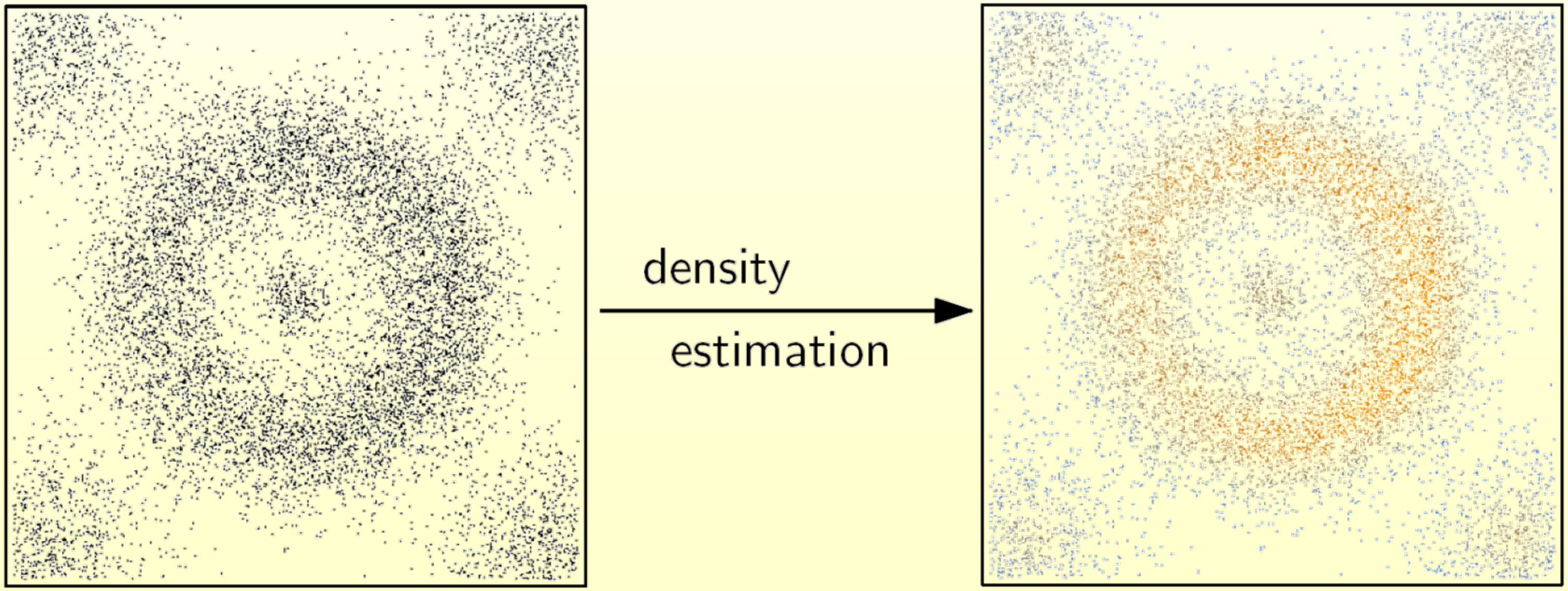
- sensor networks:
  - collection of sensors monitoring an area
  - sensors measure a physical quantity ϕ
  - sensors communicate within radius δ

Goal: analyze landscape of ϕ



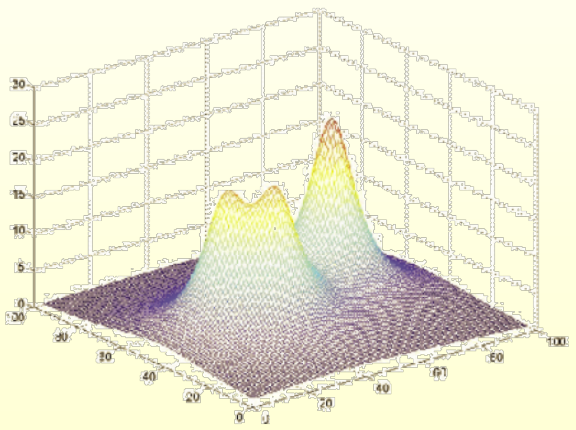
Motivating Applications

- unsupervised learning:
  - data points drawn at random from some unknown density distribution f
  - approximate f through some density estimator $\hat{f}$
  - cluster data points according to prominent basins of attraction of $\hat{f}$



Extant Approaches

- Classical: when a parametrization of $\mathbb{X}$ is available, this is a standard function interpolation or regression problem

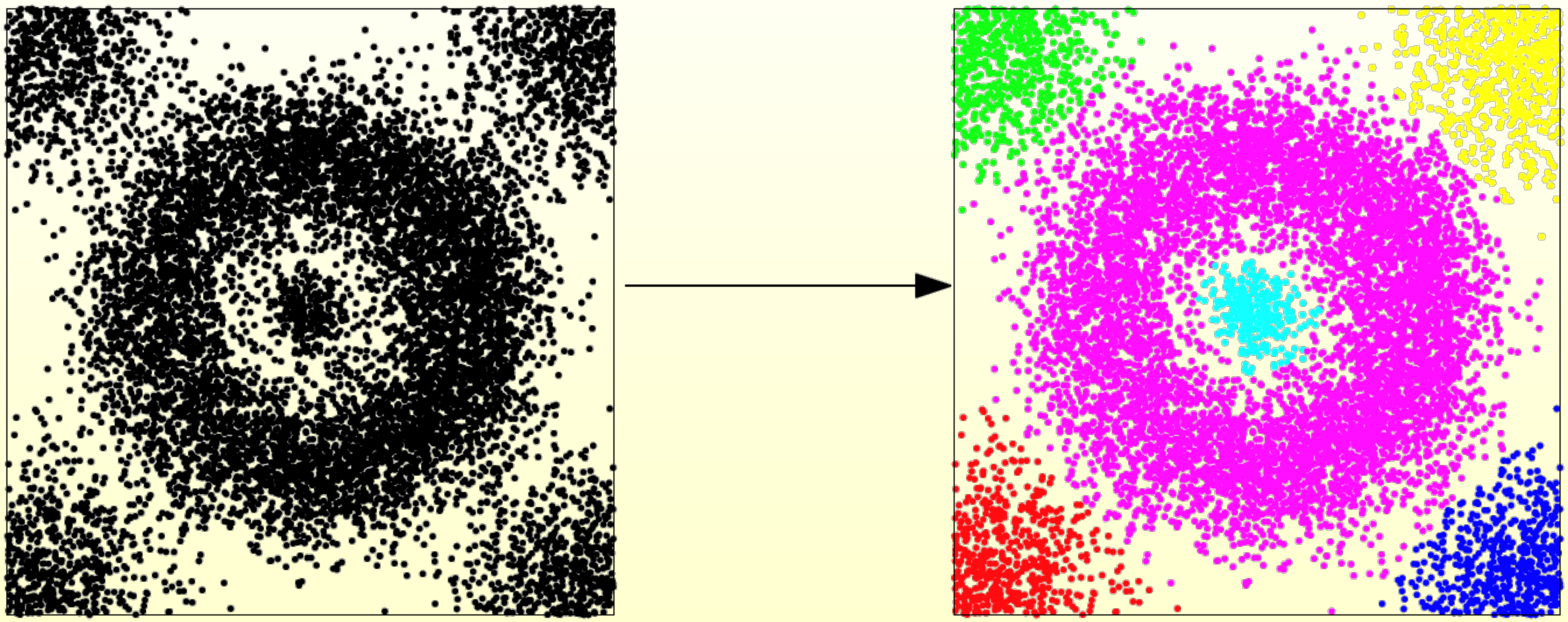


- Persistence-based: using a triangulation of $\mathbb{X}$ based on L, obtained from a parametrization or other means

Cluster Analysis

Input: a finite set of observations: - point cloud with coordinates

- distance / (dis-)similarity matrix

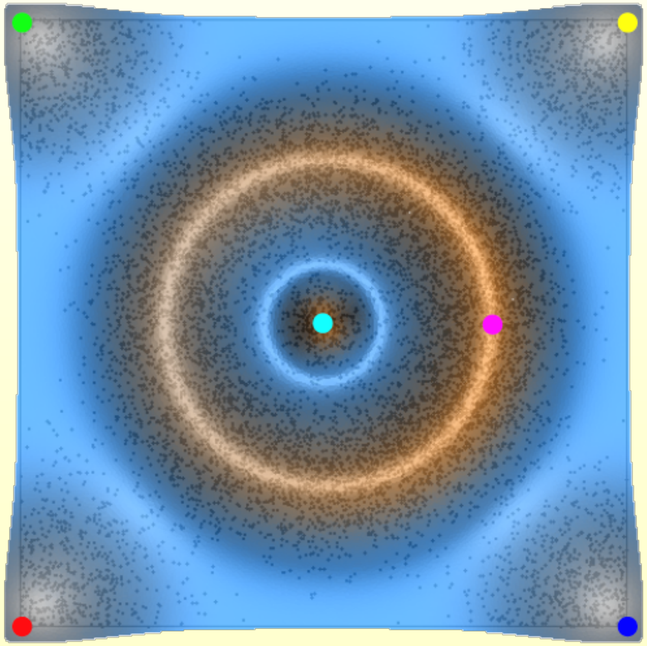


Task:

partition the data points into a collection of *relevant* subsets called clusters

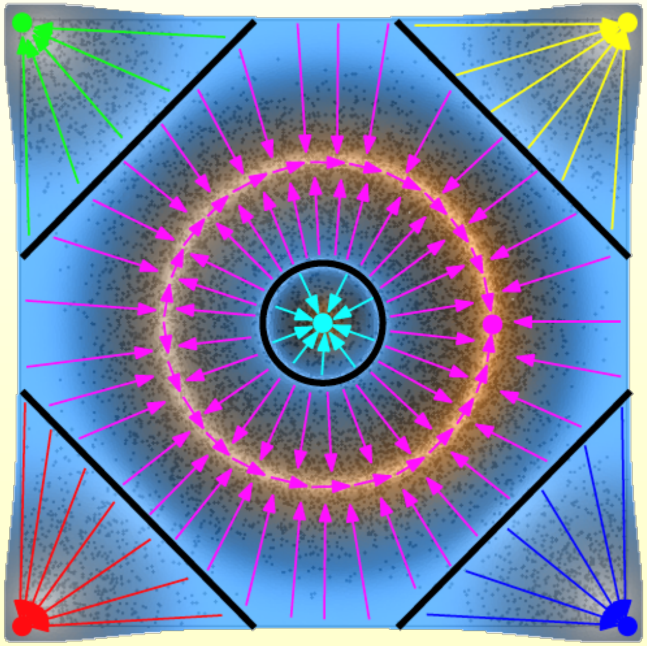
Mode-Seeking Paradigm

- Assume the data points are sampled from some unknown probability distribution
- Partition the data according to the basins of attraction of the peaks of the density



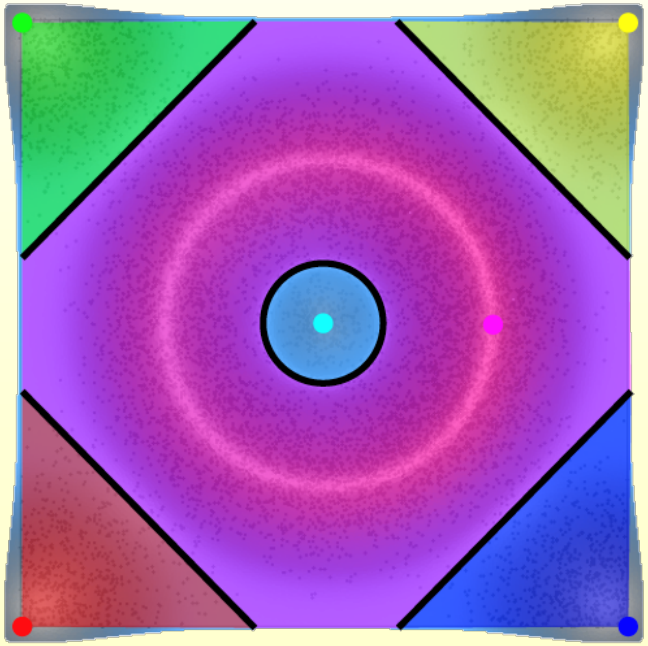
Mode-Seeking Paradigm

- Assume the data points are sampled from some unknown probability distribution
- Partition the data according to the basins of attraction of the peaks of the density

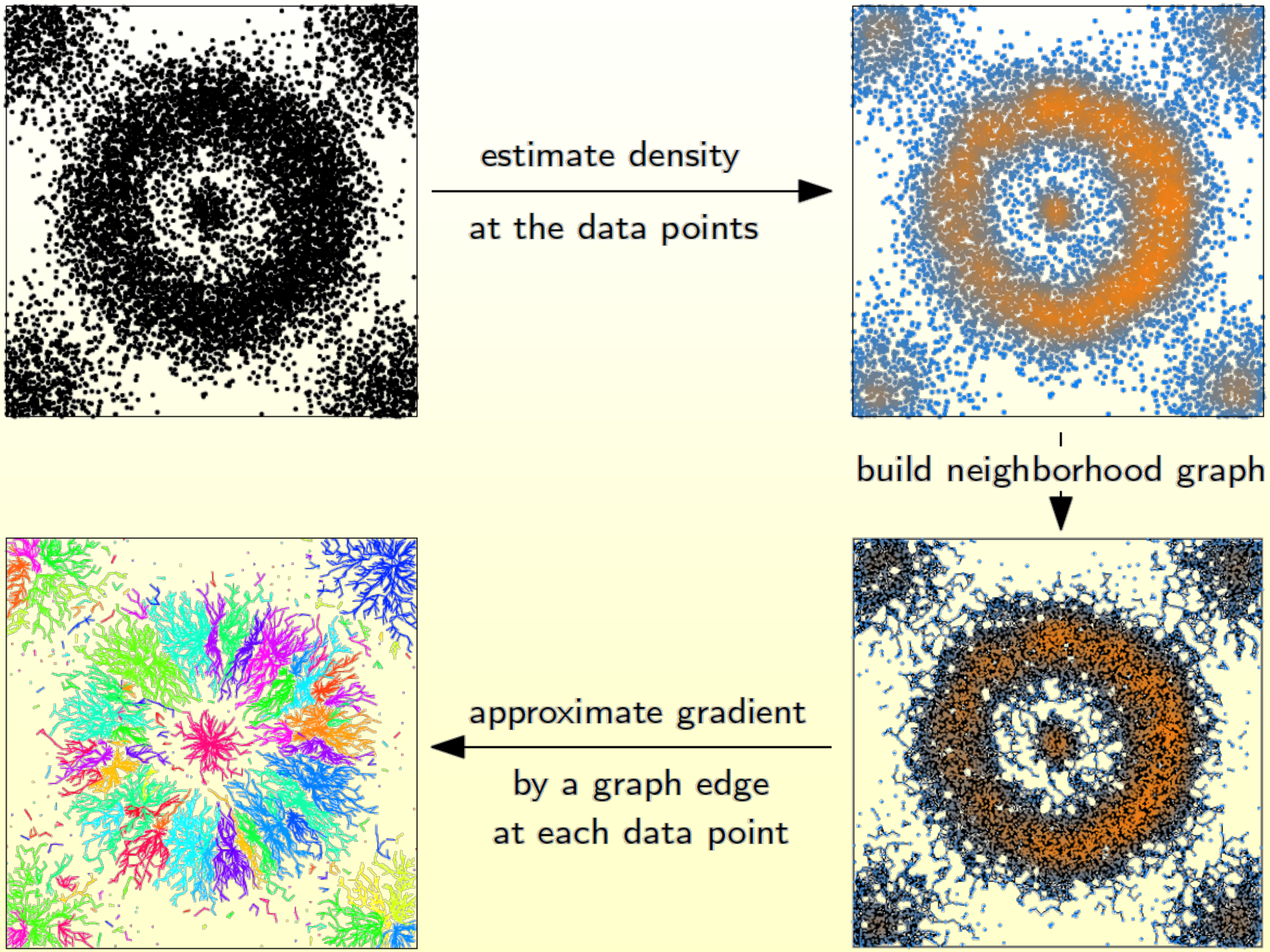


Mode-Seeking Paradigm

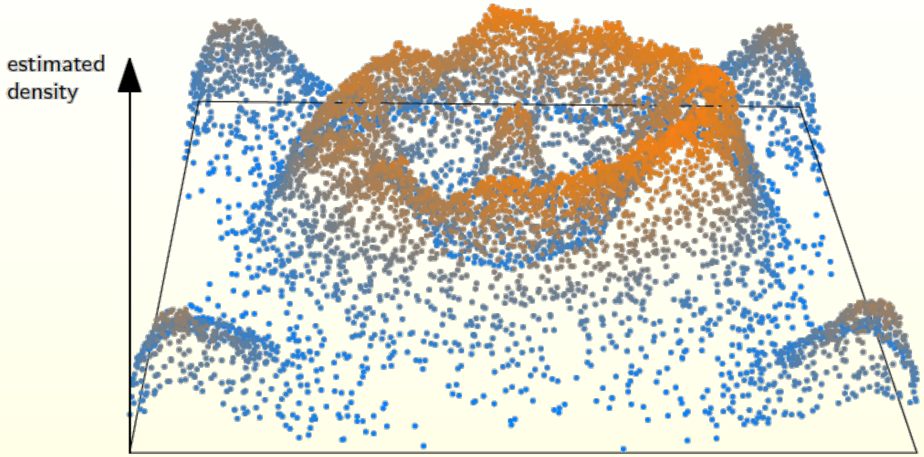
- Assume the data points are sampled from some unknown probability distribution
- Partition the data according to the basins of attraction of the peaks of the density

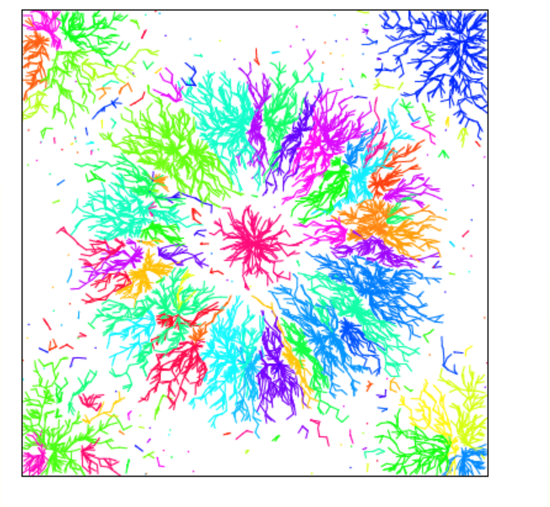


Mean-Shift and Variations



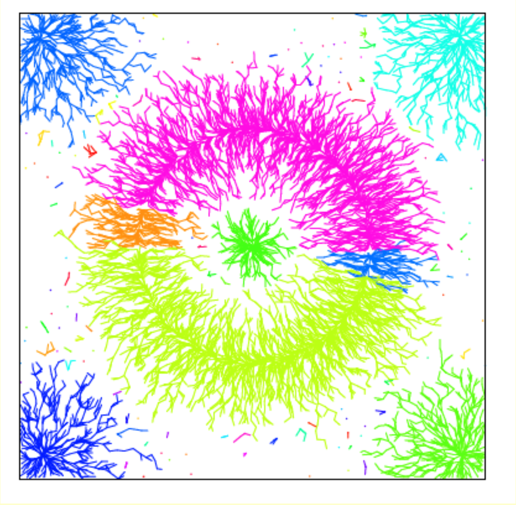
Things Can Go Wrong





Noisy density estimator

Bad proximity graph

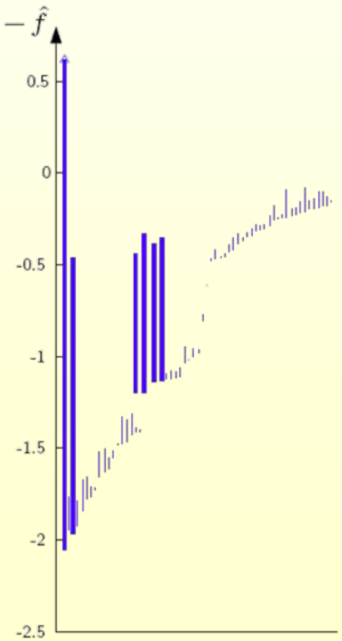


Persistence-Based Approach

Assumptions: $\mathbb{X}$ triangulated space, $f : \mathbb{X} \to \mathbb{R}$ Lipschitz continuous

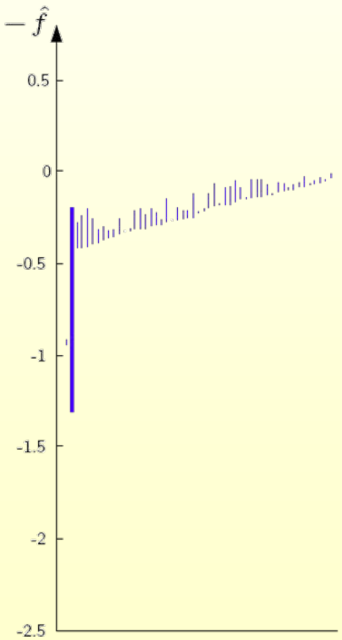
$\longrightarrow$ build PL approximation $\hat{f}$ of f

$\longrightarrow$ apply persistence algo. to $\pm\hat{f}$ [Edelsbrunner, Letscher, Zomorodian '00]



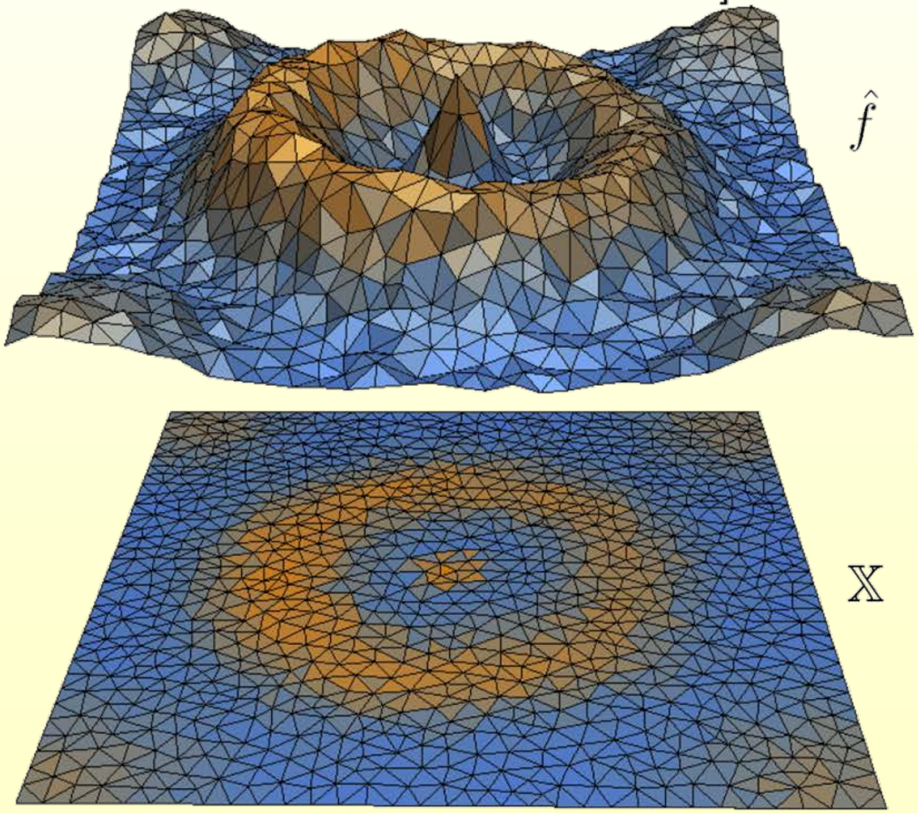
β_0

(6 prominent peaks)



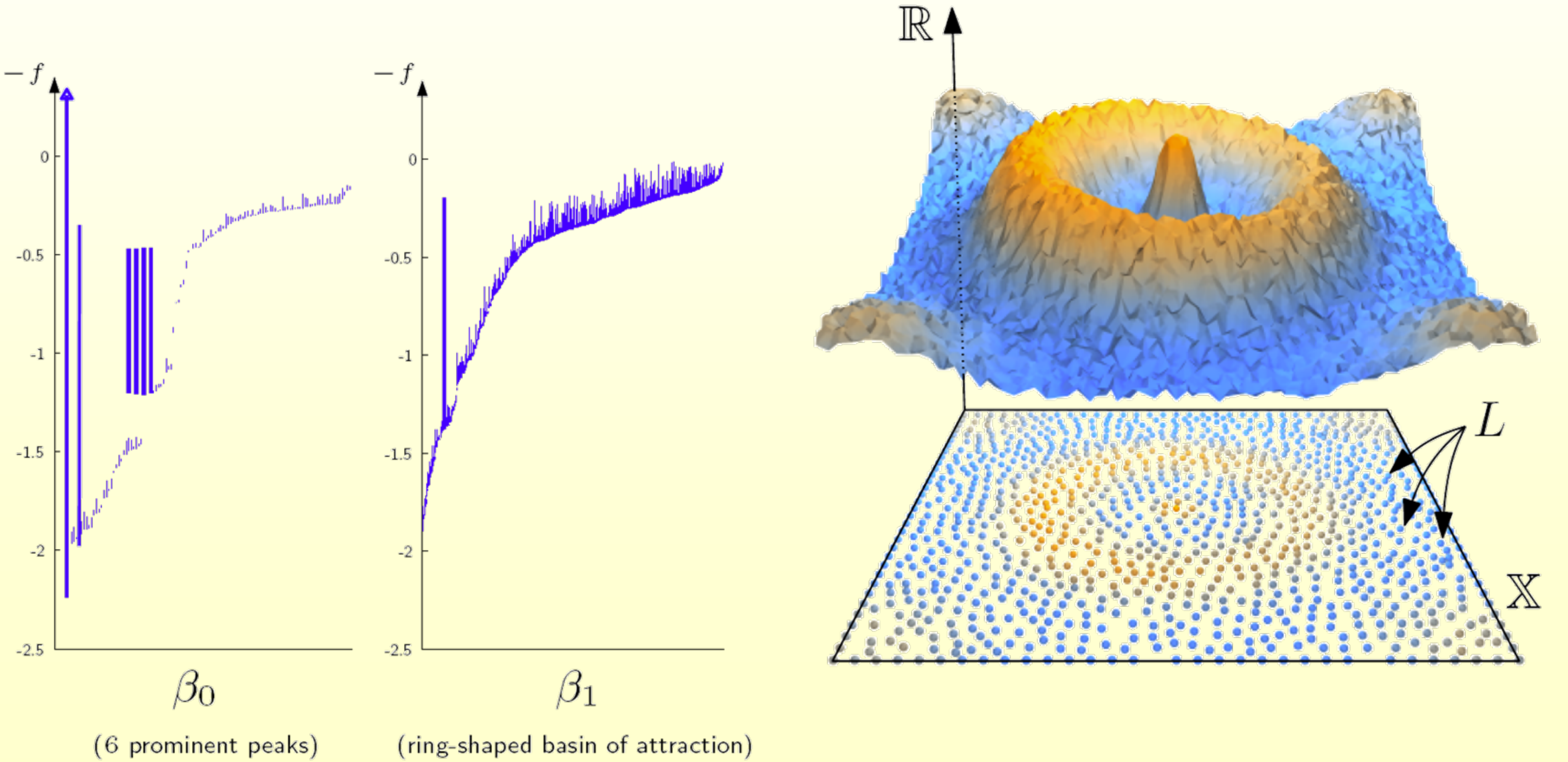
β_1

(ring-shaped basin of attraction)



Clustering Example A

Assumptions: $\mathbb{X}$ Riemannian manifold, $f : \mathbb{X} \to \mathbb{R}$ c-Lipschitz,
L geodesic ε-cover of $\mathbb{X}$, for some unknown $\varepsilon > 0$.



The Persistence Approach: ToMATo

- Density estimator $\hat{f}$ defines an order on the point cloud

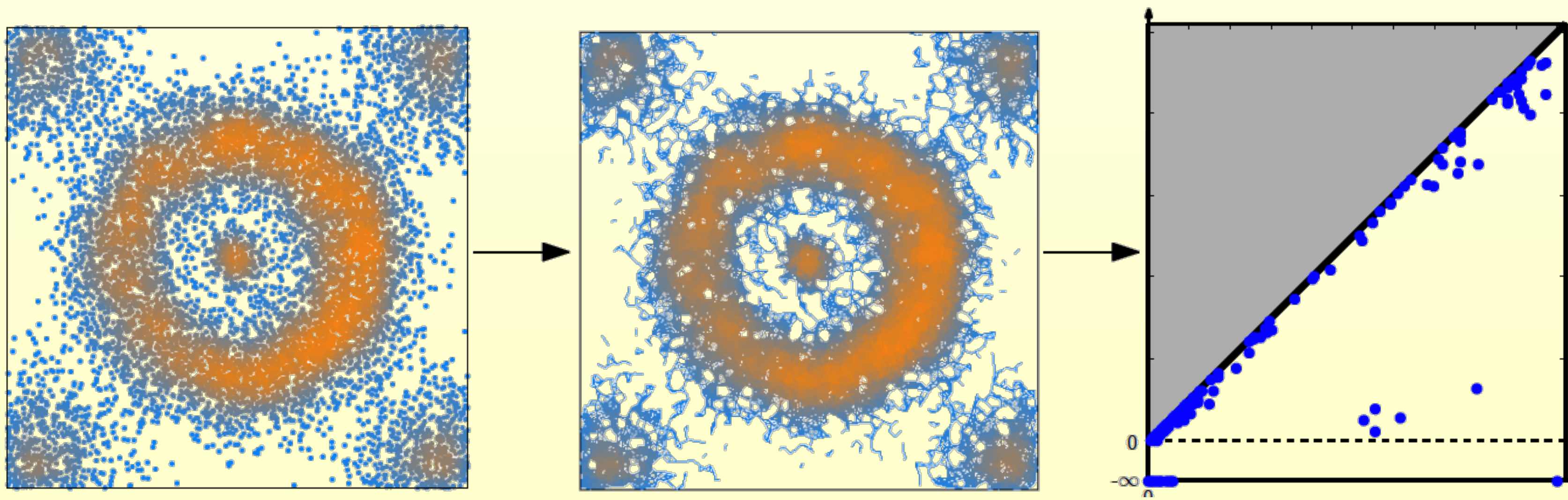
  (sort data points by **decreasing** estimated density values)

- Extend order to the graph edges $\rightarrow$ *upper-star filtration*

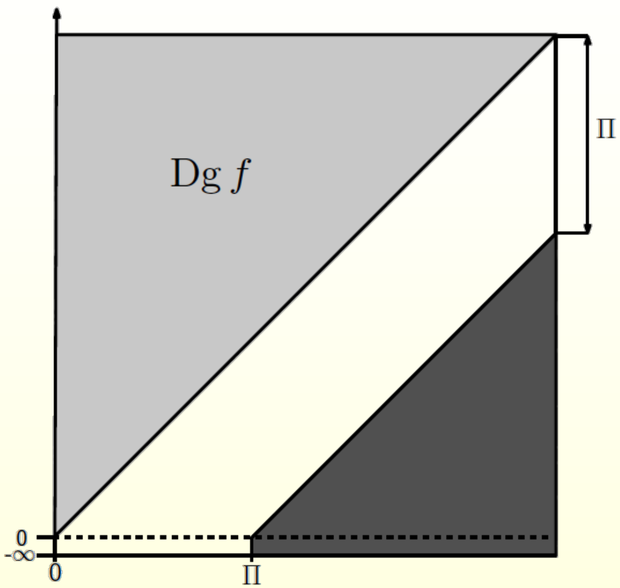
  $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$

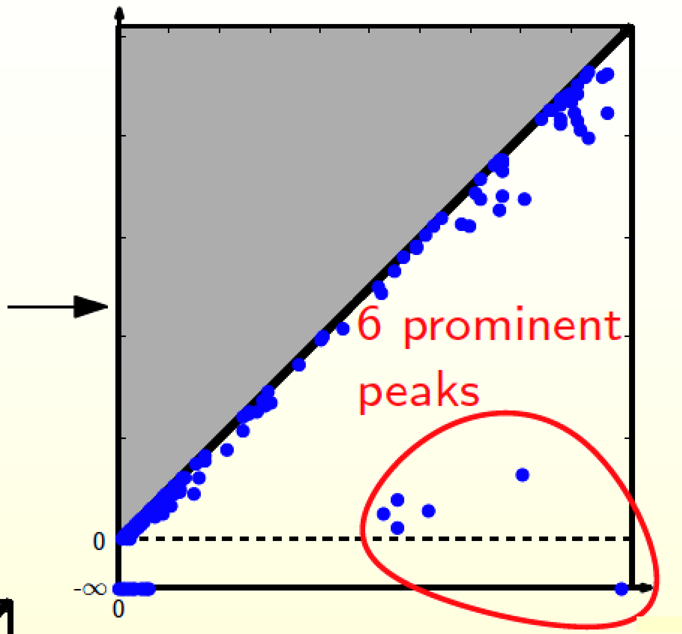
- Compute the 0-dimensional persistence diagram of this filtration

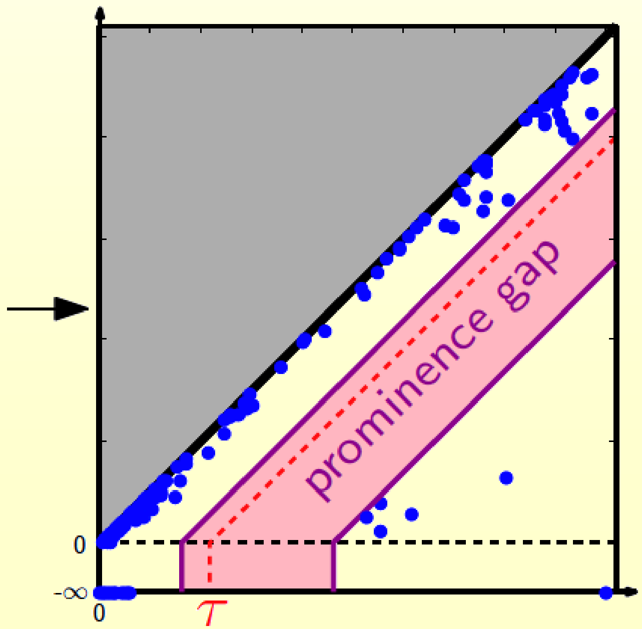
  (apply 0-dimensional persistence algorithm $\rightarrow$ union-find data structure)



Estimating the Prominent Clusters





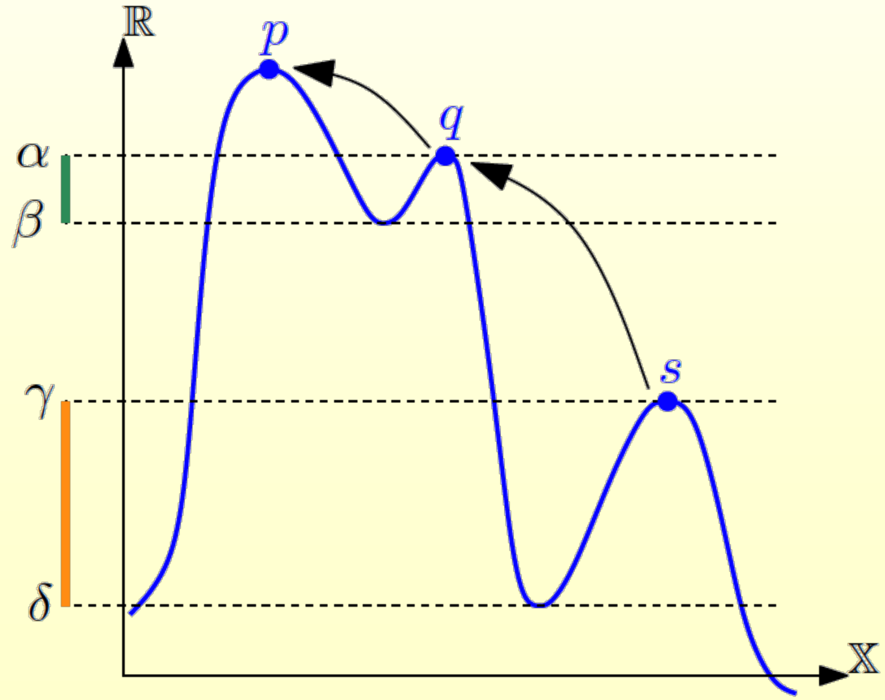


The gap parameter τ

Merging Clusters

- 0-dimensional persistence builds a hierarchy of the peaks of $\hat{f}$ (merge tree)
- merge clusters according to the hierarchy (merge each cluster into its parent)
- given a fixed threshold $\tau \geq 0$, only merge those clusters of prominence $< \tau$

$$\gamma - \delta < \tau \leq +\infty$$



Basins of Attraction for A

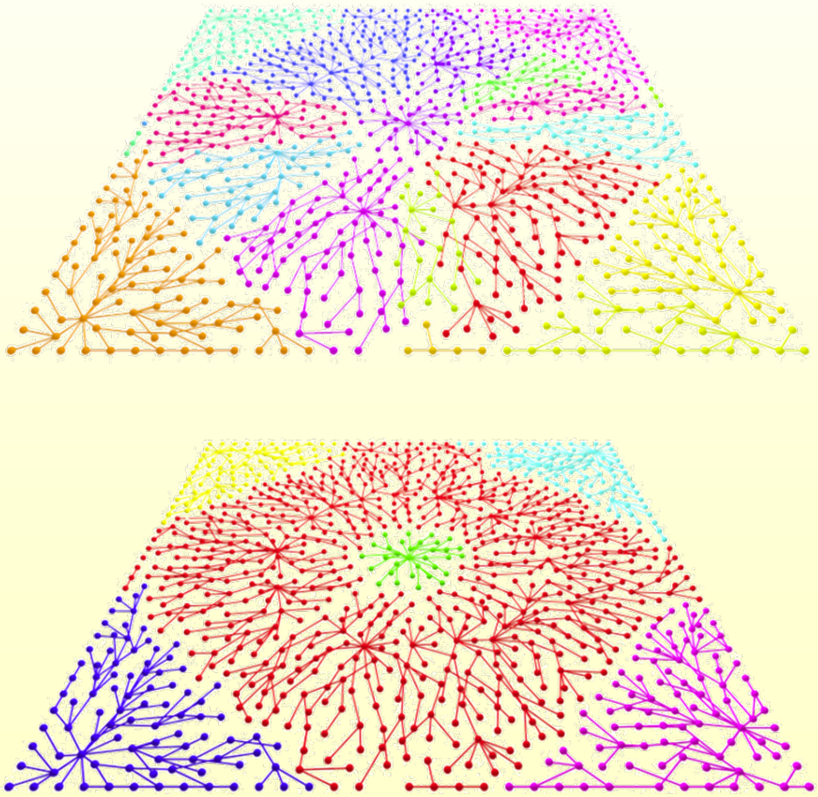
Goal: approximate basins of attraction of significant peaks of f

$\Rightarrow$ segmentation/clustering of point cloud L

Approach:

- rough approximation of gradient of f within Rips graph,
- merge clusters according to 0-dimensional barcode.

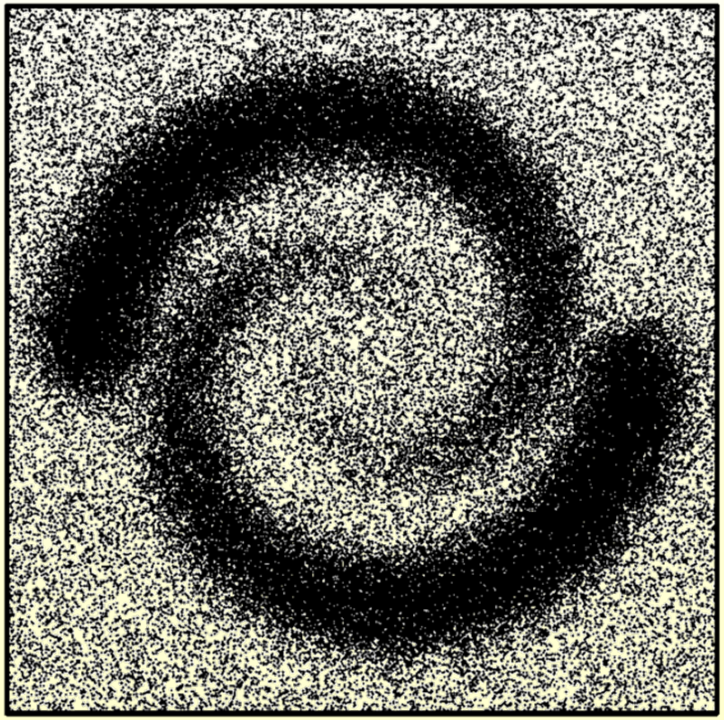
$\rightarrow$ union-find data structure

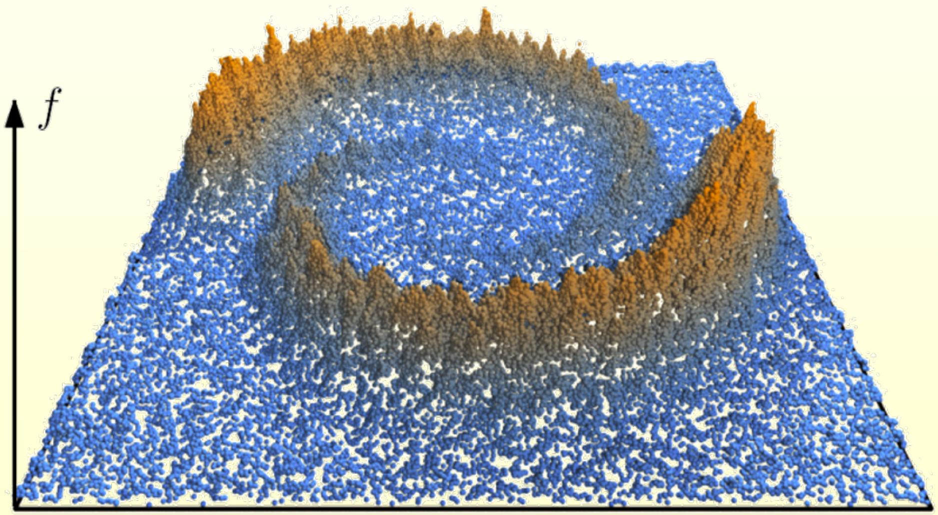


Clustering B – The Rips Parameter δ

Input: $\mathbb{X} = [0,1]^2$; $|L| = 100{,}000$;

$f = \#\ \{$ data pts in fixed-radius ball $\}$



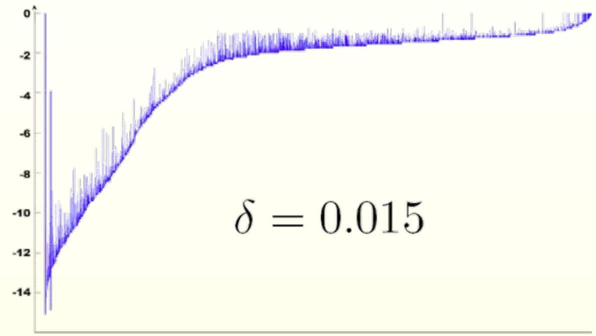


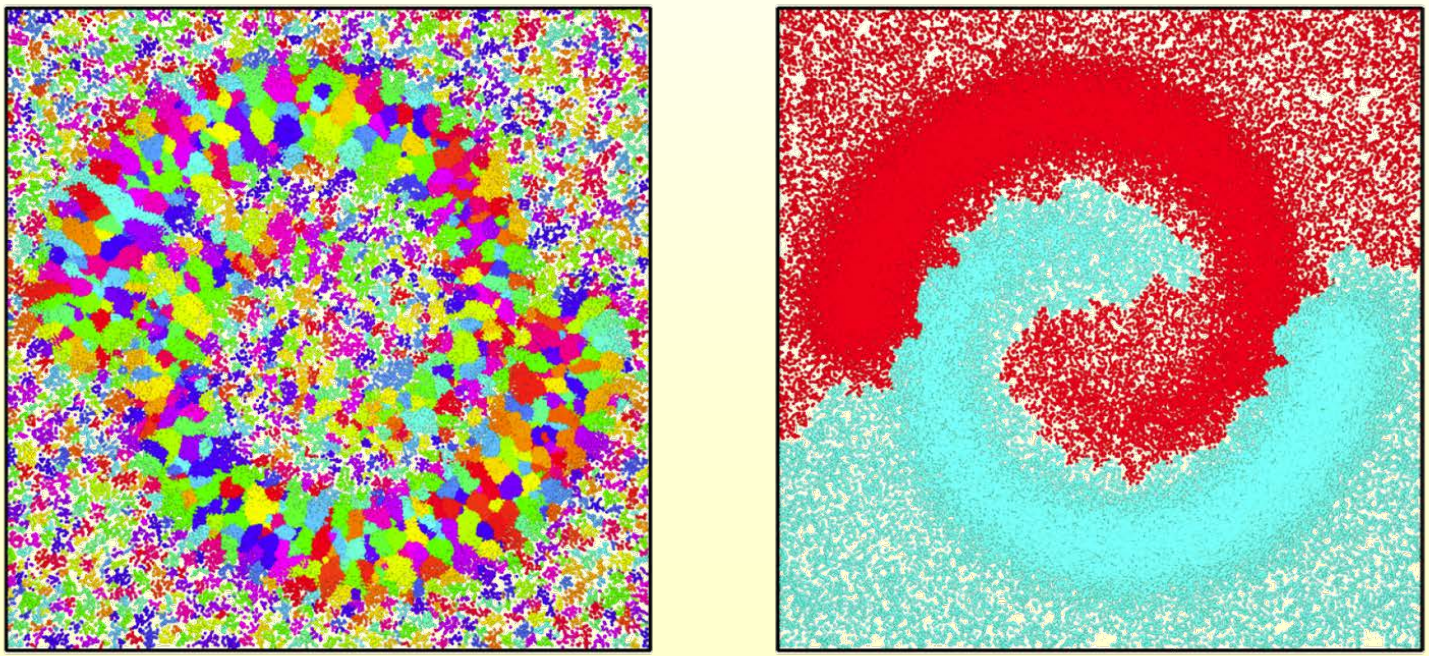
Clustering B

Clustering B

Input: $\mathbb{X} = [0,1]^2$; $|L| = 100,000$;

$f = \#\ \{$ data pts in fixed-radius ball $\}$



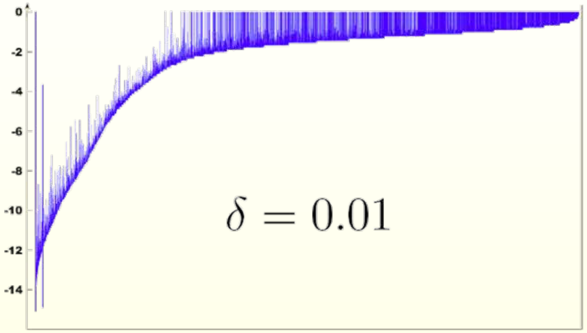


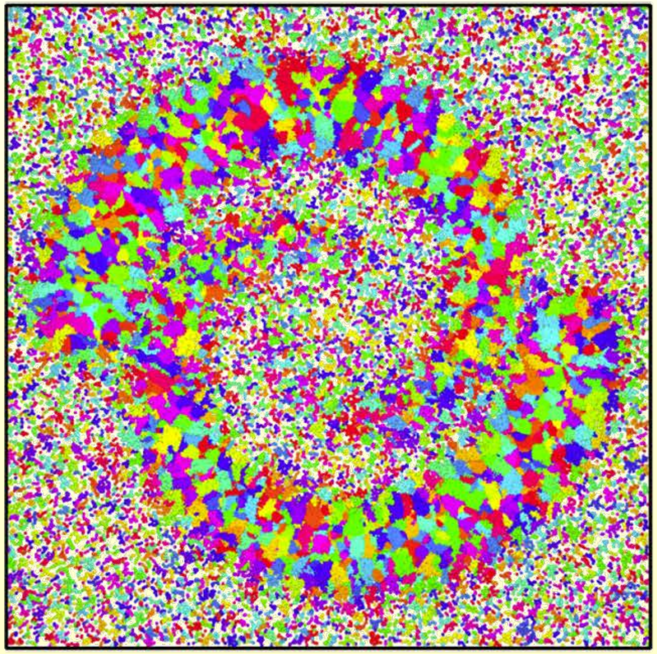
Clustering B

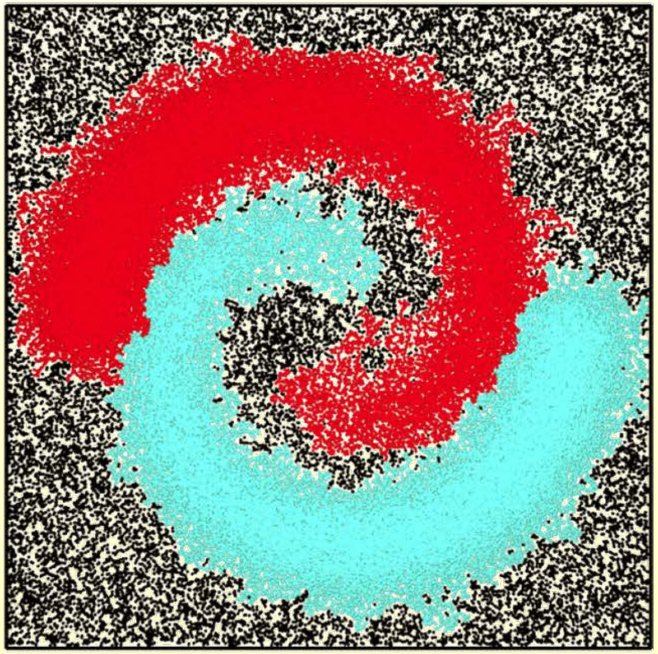
Clustering B

Input: $\mathbb{X} = [0,1]^2$; $|L| = 100,000$;

$f = \#$ { data pts in fixed-radius ball }

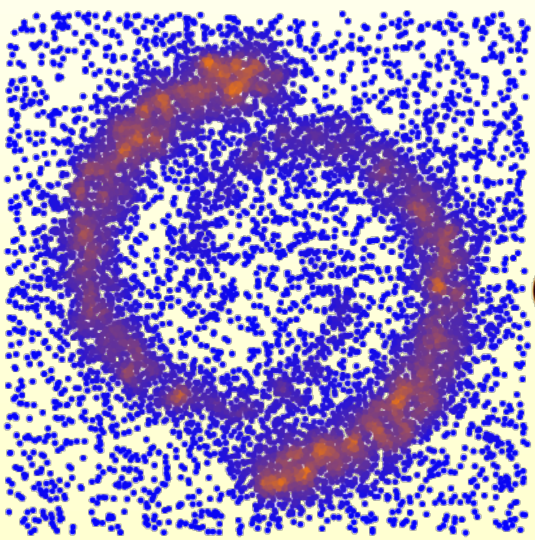




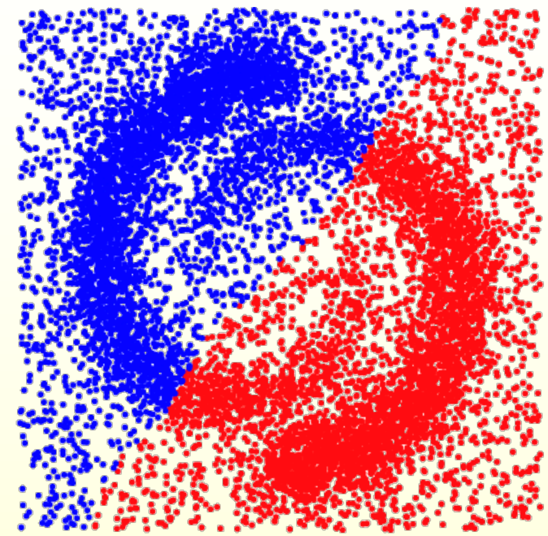


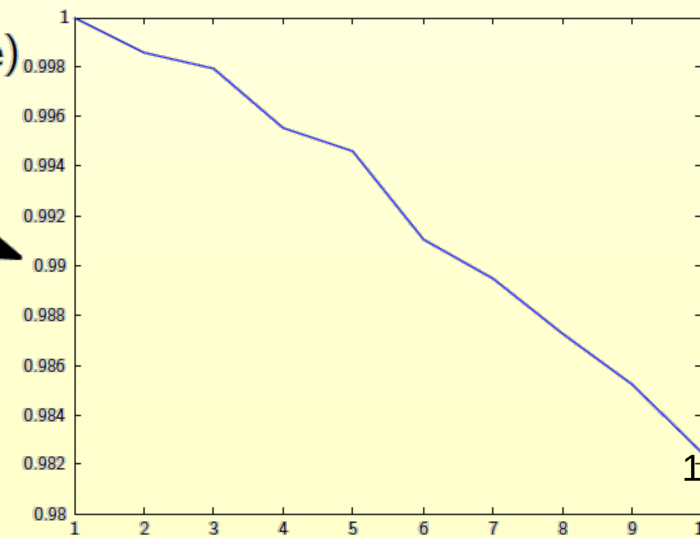
FYI, Spectral Clustering

Synthetic Data



Spectral clustering
(k-means in eigenspace)





Another Hard Example

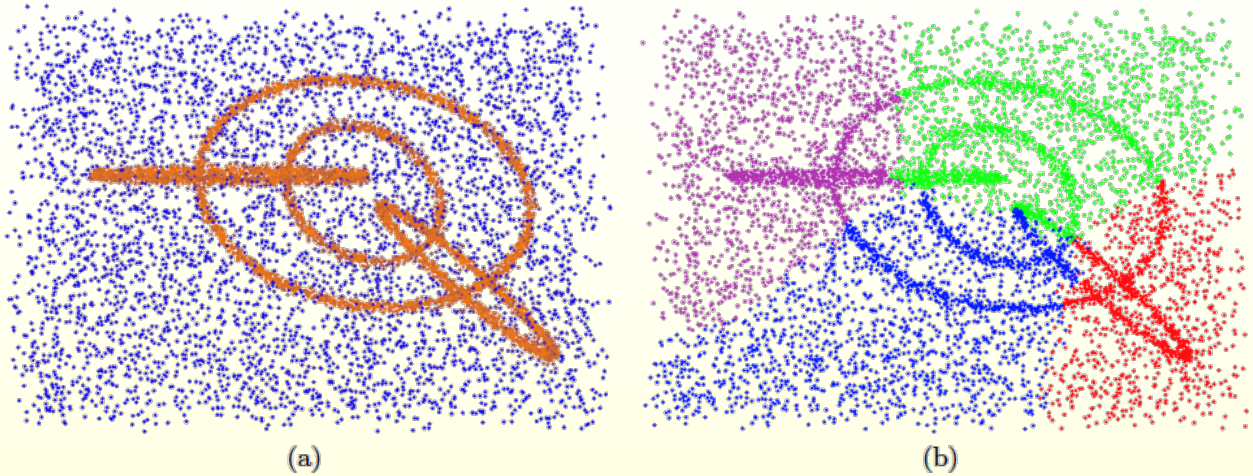


Figure 7: (a) The rings data set with the estimated density function. (b) The result obtained using spectral clustering.

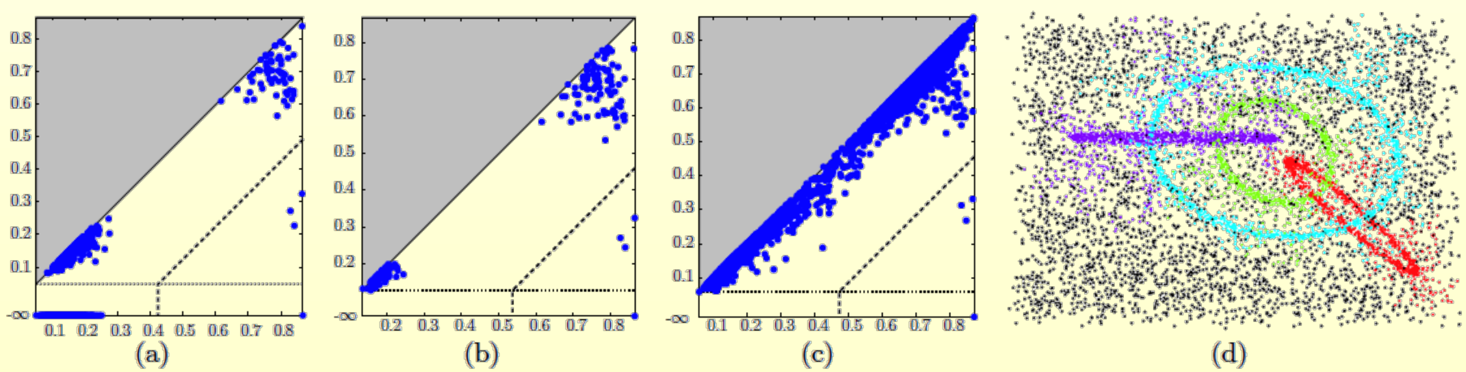
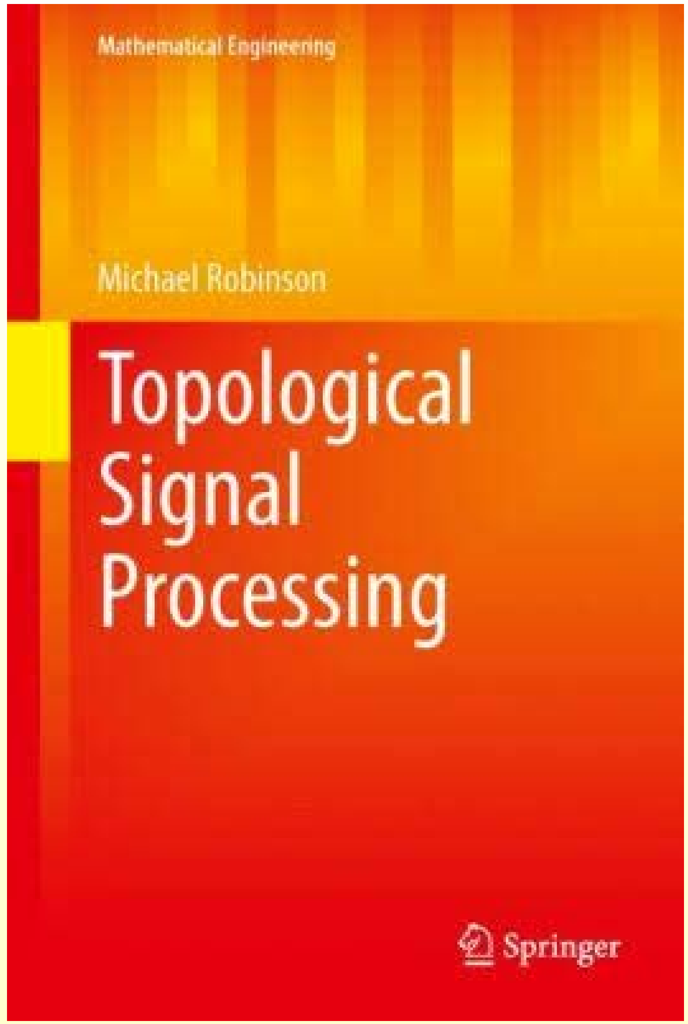


Figure 8: Outputs of ToMATo on the rings data set: the obtained PD with (a) δ-Rips graph, (b) k-nn graph, and (c) Delaunay graph. (d) Clustering obtained with the δ-Rips graph.

Topological Signal Processing



The End