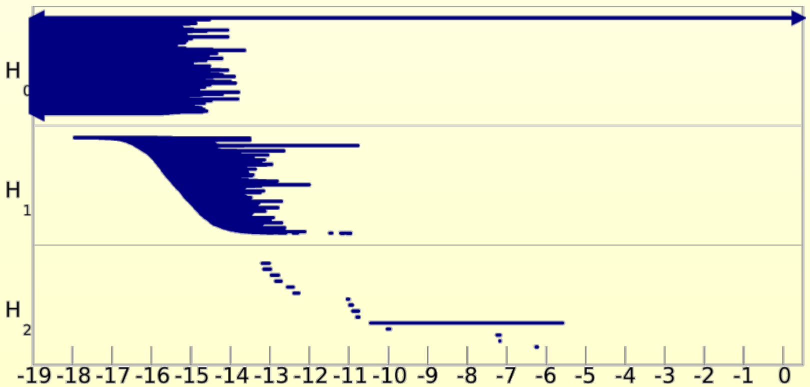
CS233: Geometric and Topological Data Analysis

Variations and Applications on Persistent Homology

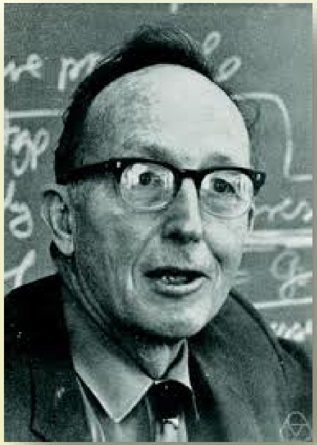
30 April 2018



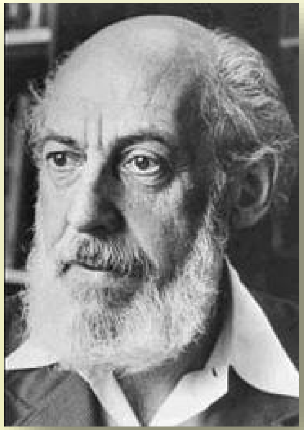
Slides ack: Afra Zomorodian, Maks Ovsjanikov, Steve Oudot

Homological Algebra: Functors and Categories

Herni Cartan



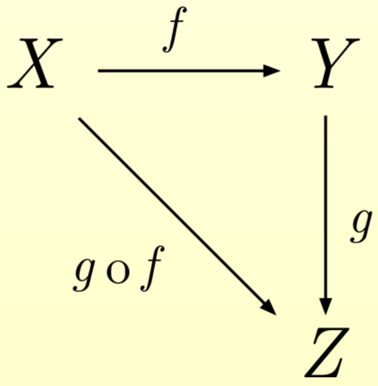
Saunders MacLane



Samuel Eilenberg

Categories

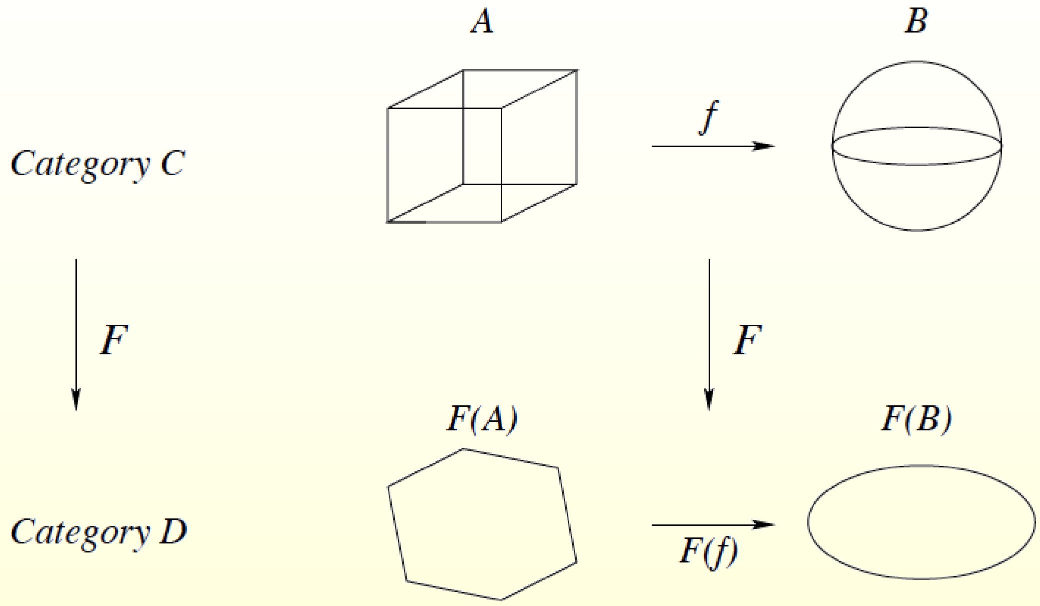
- A collection $\mathrm{Ob}(\mathcal{C})$ of objects
- Sets $\mathrm{Mor}(X,Y)$ of morphisms for each pair $X,Y \in \mathrm{Ob}(\mathcal{C})$
- An identity morphism $1 = 1_X \in \mathrm{Mor}(X,X)$ for each X.
- a composition of morphisms function $\circ : \mathrm{Mor}(X,Y) \times \mathrm{Mor}(Y,Z) \to \mathrm{Mor}(X,Z)$ for each triple $X,Y,Z \in \mathrm{Ob}(\mathcal{C})$, satisfying $f \circ 1 = 1 \circ f = f$, and $(f \circ g) \circ h = f \circ (g \circ h)$.
- A category $\mathcal{C}$



Example Categories

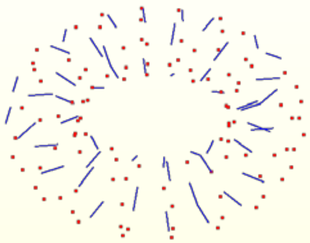
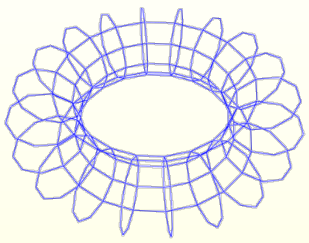
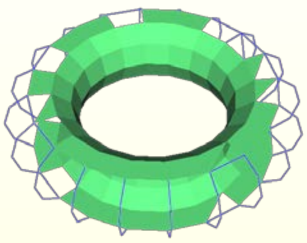
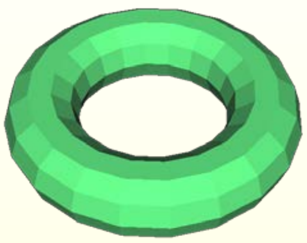
category	morphisms
sets	arbitrary functions
groups	homomorphisms
topological spaces	continuous maps
topological spaces	homotopy classes of maps

Functors



- $X \in \mathcal{C}, F(X) \in \mathcal{D}$,
- $f \in \text{Mor}(X, Y), F(f) \in \text{Mor}(F(X), F(Y))$
- $F(1) = 1$ and $F(f \circ g) = F(f) \circ F(g)$
- F is a (covariant) functor

Homology is a Functor


$\subseteq$

$\subseteq$

$\subseteq$


$K^{250} \longrightarrow K^{500} \longrightarrow K^{994} \longrightarrow K^{1452}$

$\subseteq$

Functor $\big\downarrow$

$H_k(K^{250}) \longrightarrow H_k(K^{500}) \longrightarrow H_k(K^{994}) \longrightarrow H_k(K^{1452})$

Functoriality

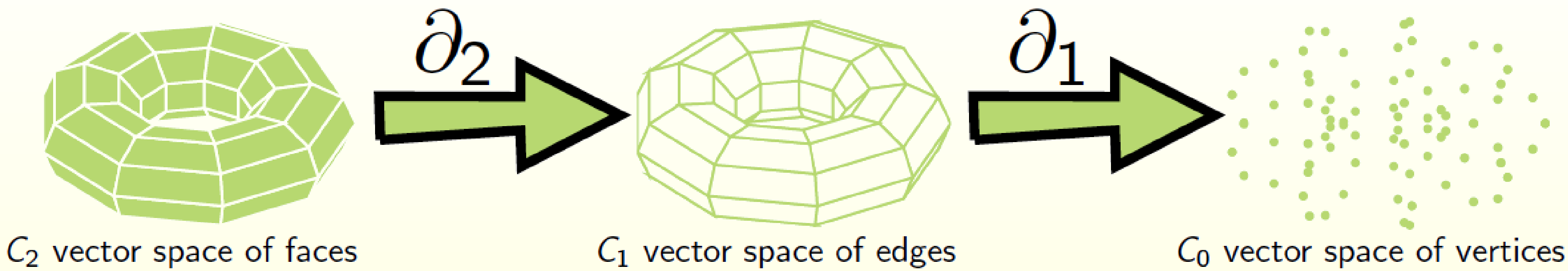
transformation of input $\Rightarrow$ transformation of output
Specifically, this is a commutative diagram:

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle H_*}\downarrow & & \downarrow{\scriptstyle H_*} \\ H_*(X) & \xrightarrow{H_*(f)} & H_*(Y) \end{array}$$

Moral: Invariants are not artifacts of arbitrary choices!

Last time:
Persistent Homology

Standard Homology



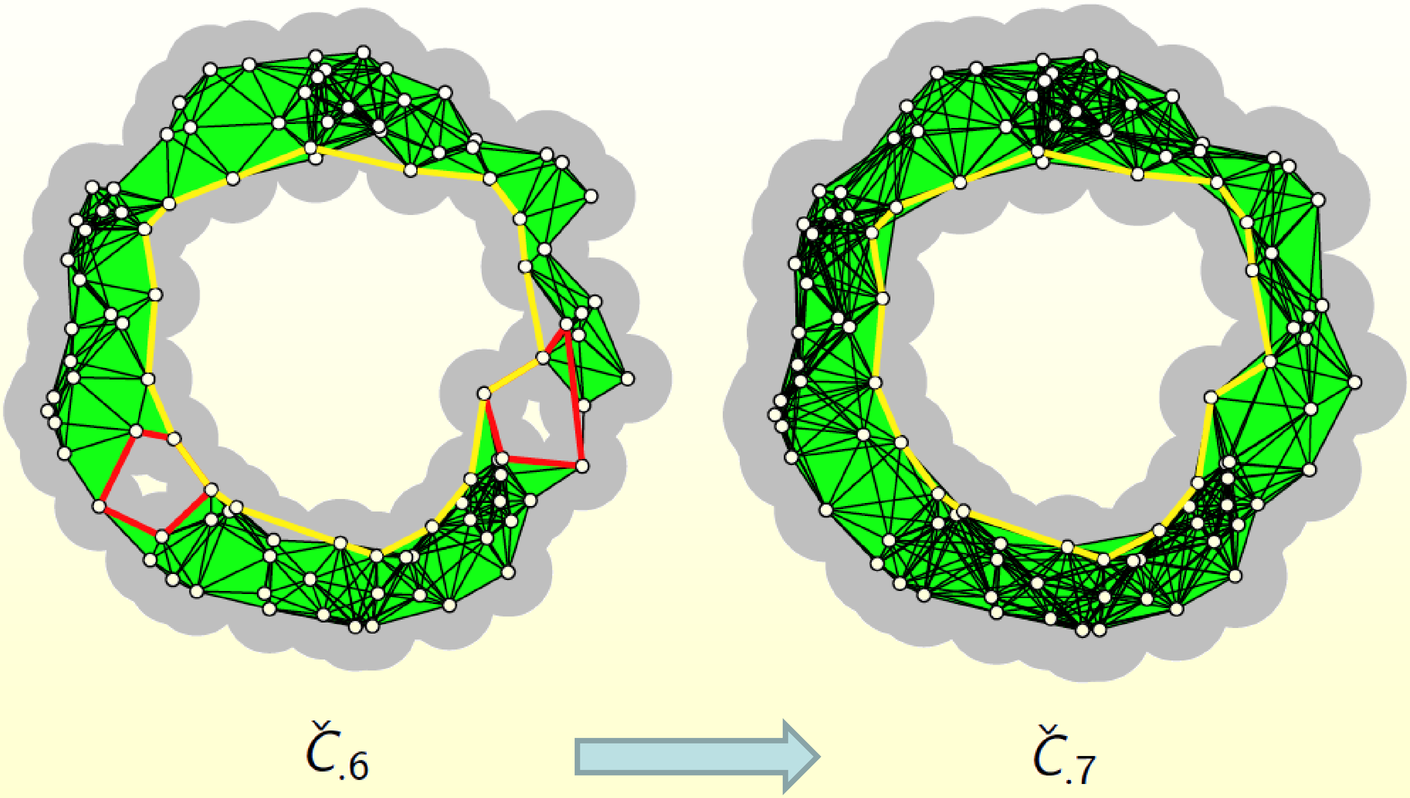
Take the linear extension of the boundary operator:

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

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

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

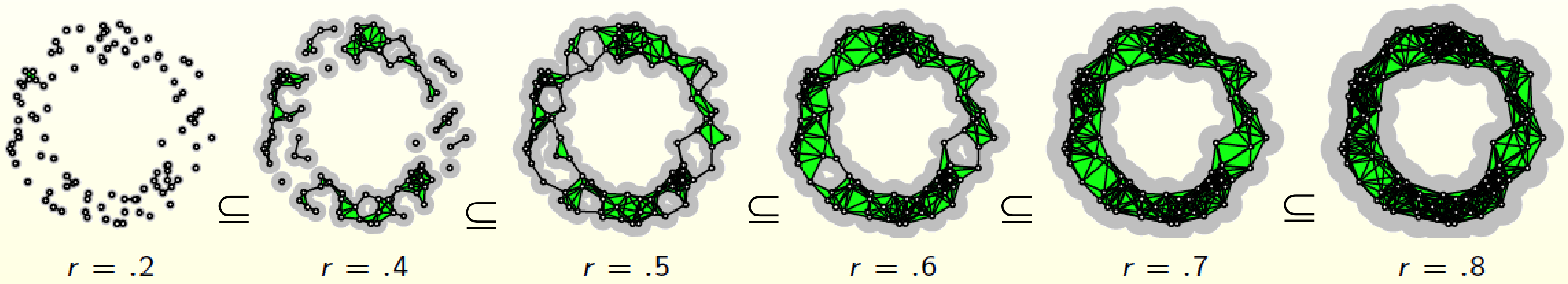
We Can Track Topological Features in a Filtration



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

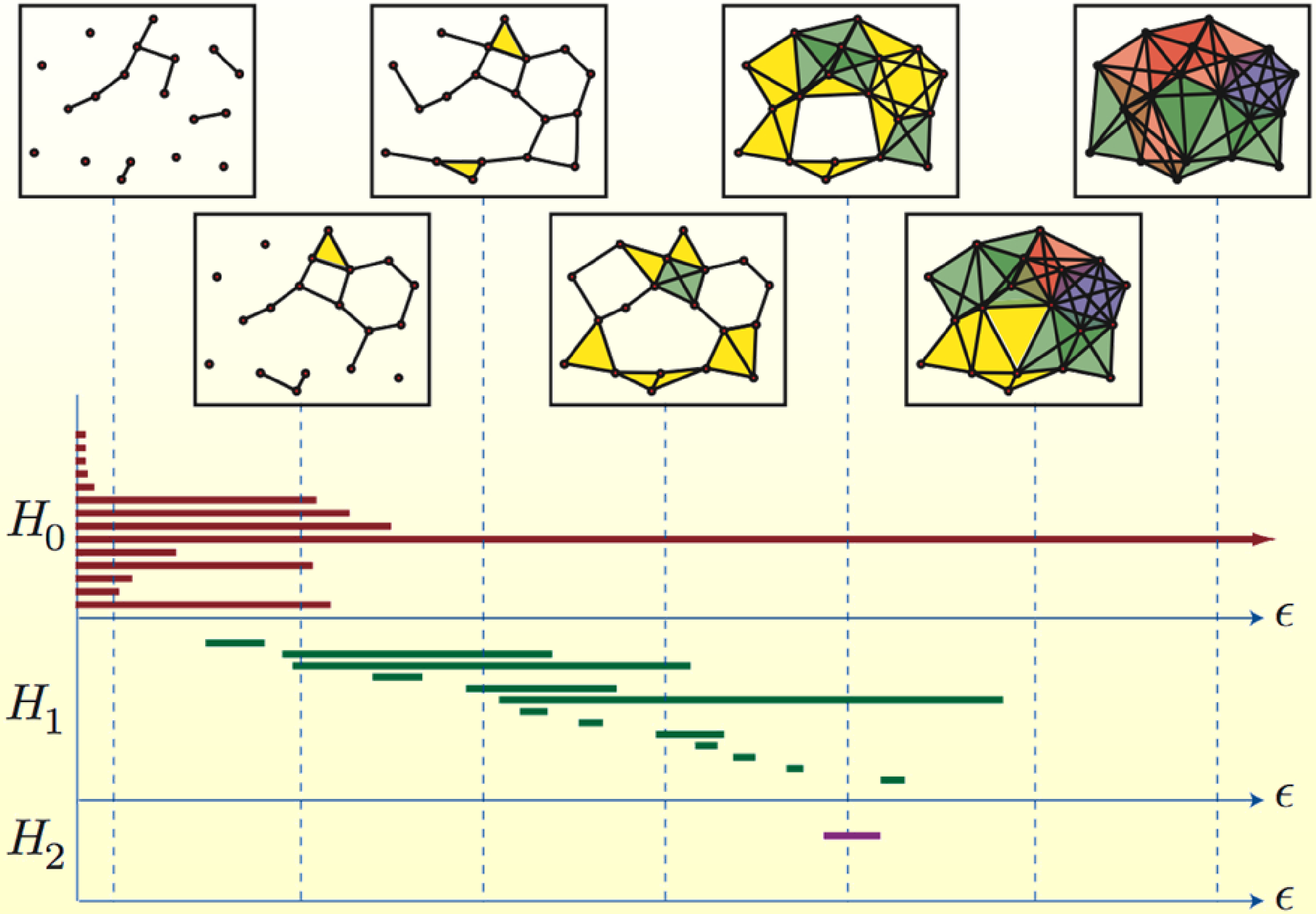


$$H_d(\check{C}_*) = \bigoplus_{\epsilon} H_d(\check{C}_\epsilon)$$

Homology of the entire filtration

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

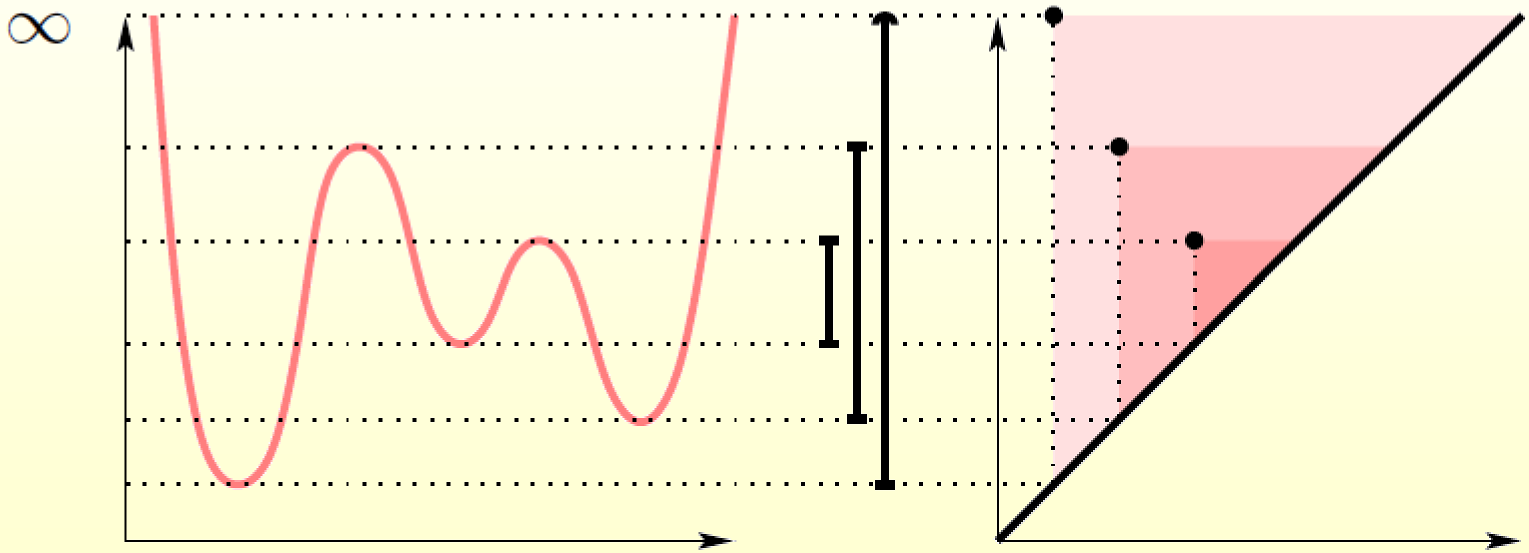
Barcodes are the Lifetimes of Topological Features



Barcodes are the output of persistent homology

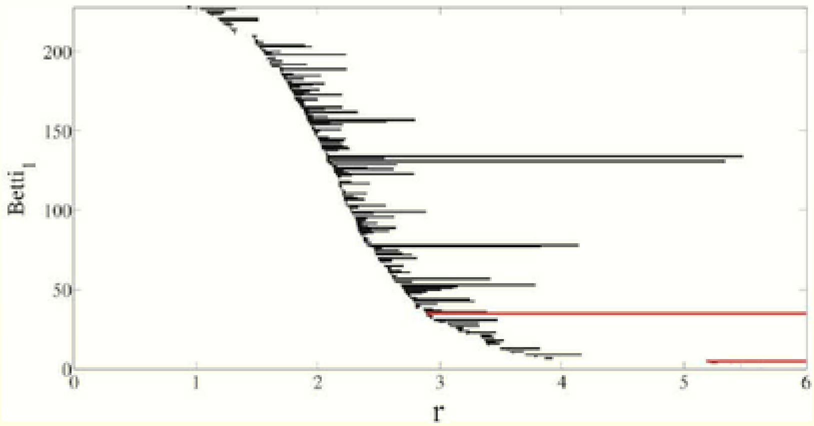
Persistence Provides a Pairing

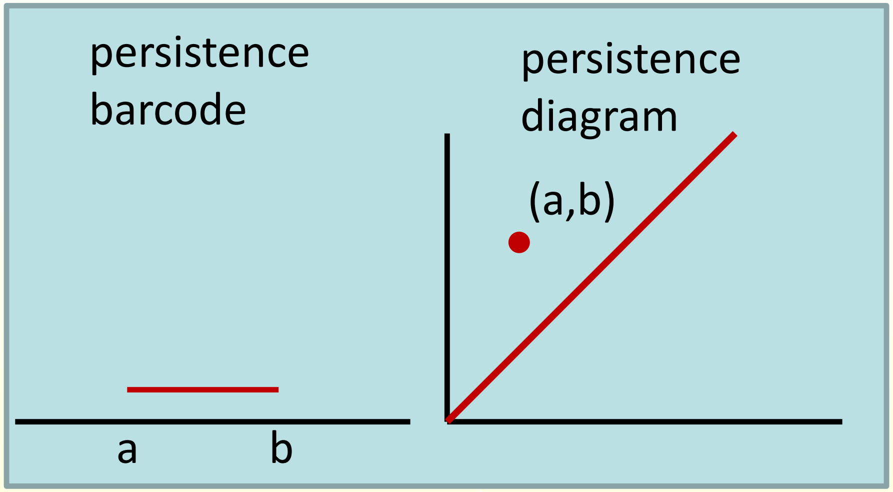
- That pairing is the persistence diagram



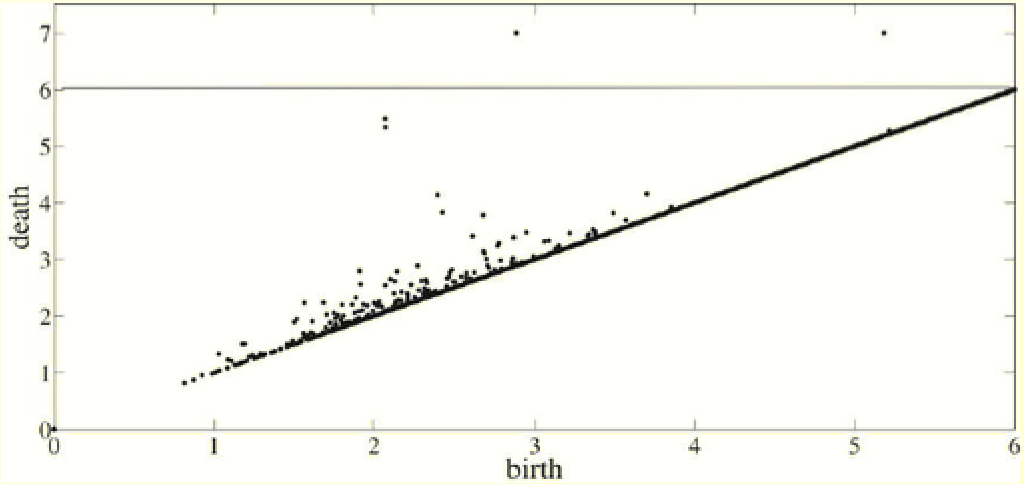
- The diagonal is always included

Multiple Views: Barcodes and Persistence Diagrams





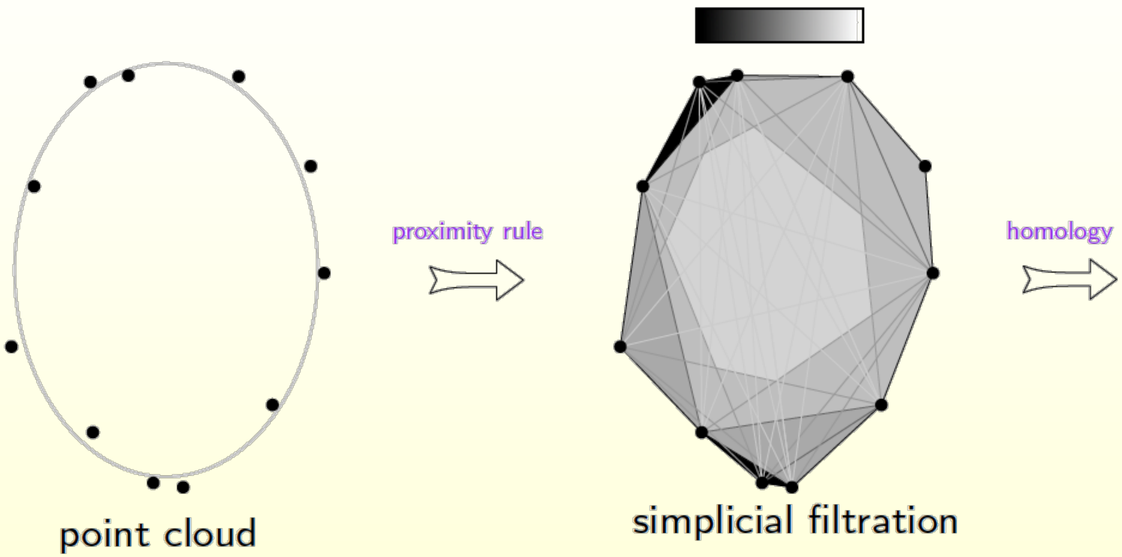
long barcodes = points away from the diagonal = robust features

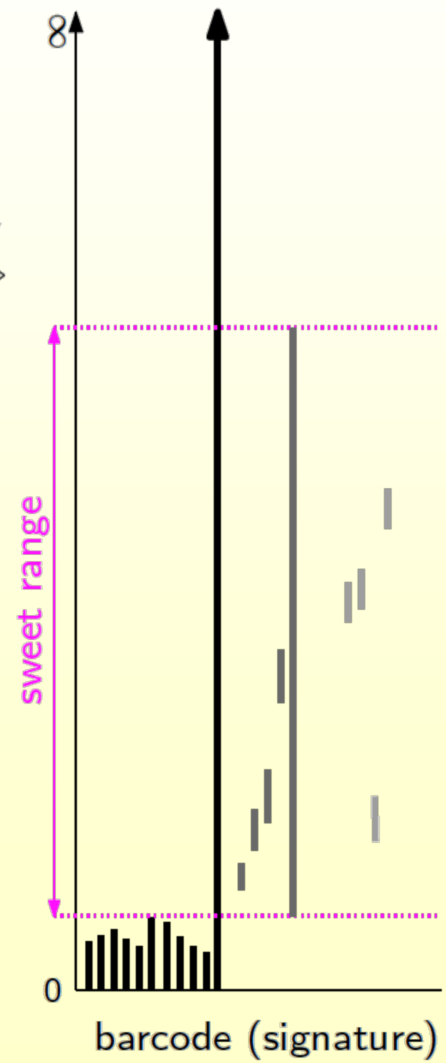


Short barcodes = points near the diagonal = noise

Map 1-D intervals to points in 2-D

Topology Inference Pipeline



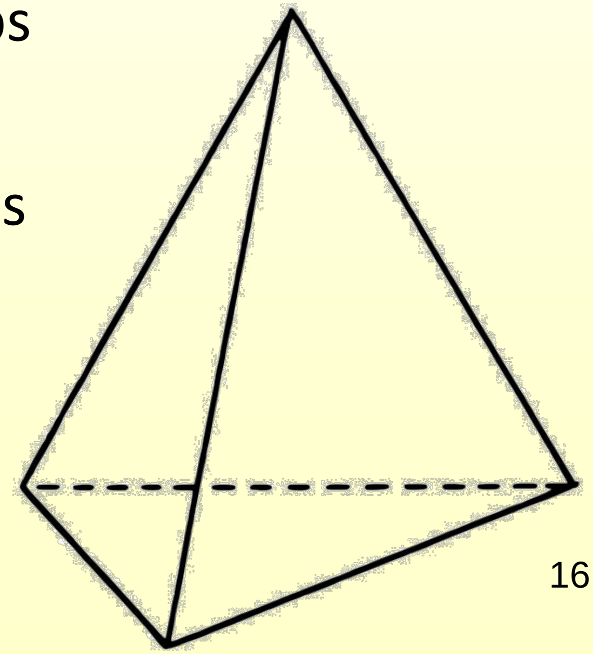


Challenges:

- scaling with input size and dimensionality
- theoretical guarantees (*sweet range*, SNR)

Recall: Betti Numbers β_i

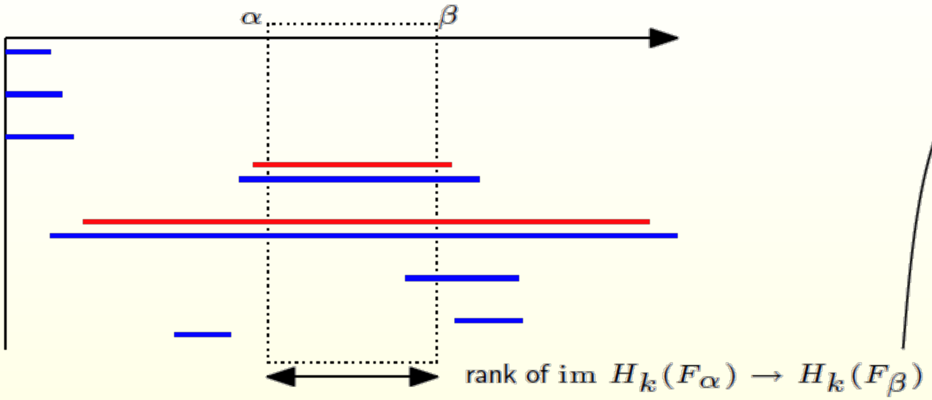
- Ranks of the free part of homology groups H_i
- β_0 counts the number of connected components
- β_1 counts the number of independent loops
- β_2 counts the number of independent voids
- …

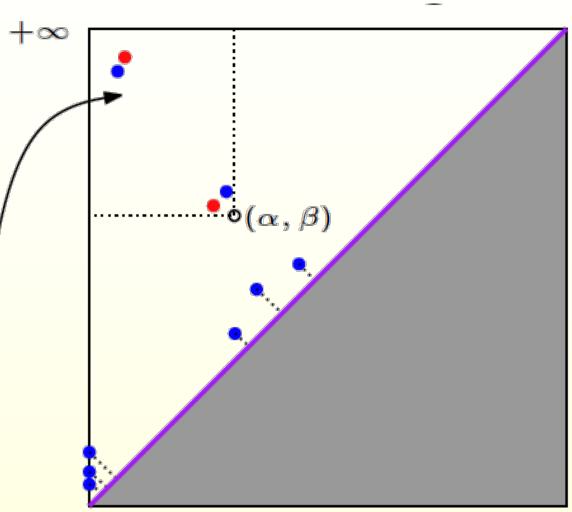


Topology is fundamentally a tool for classification

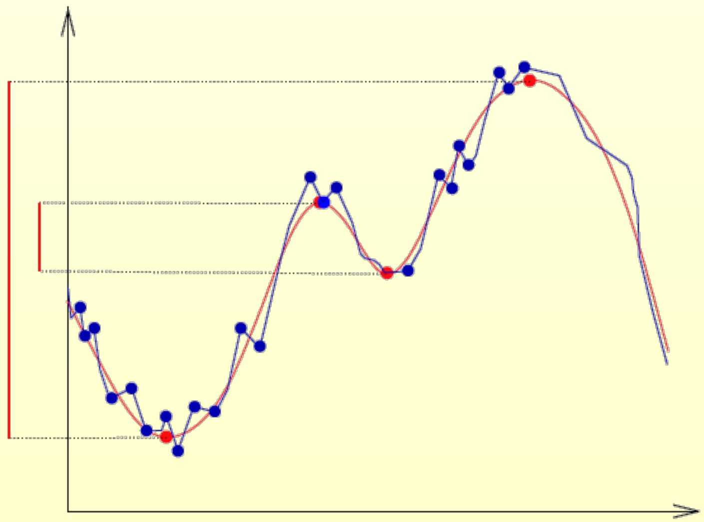
Stability of Barcodes and Persistence Diagrams

Filtering Out Topological Noise



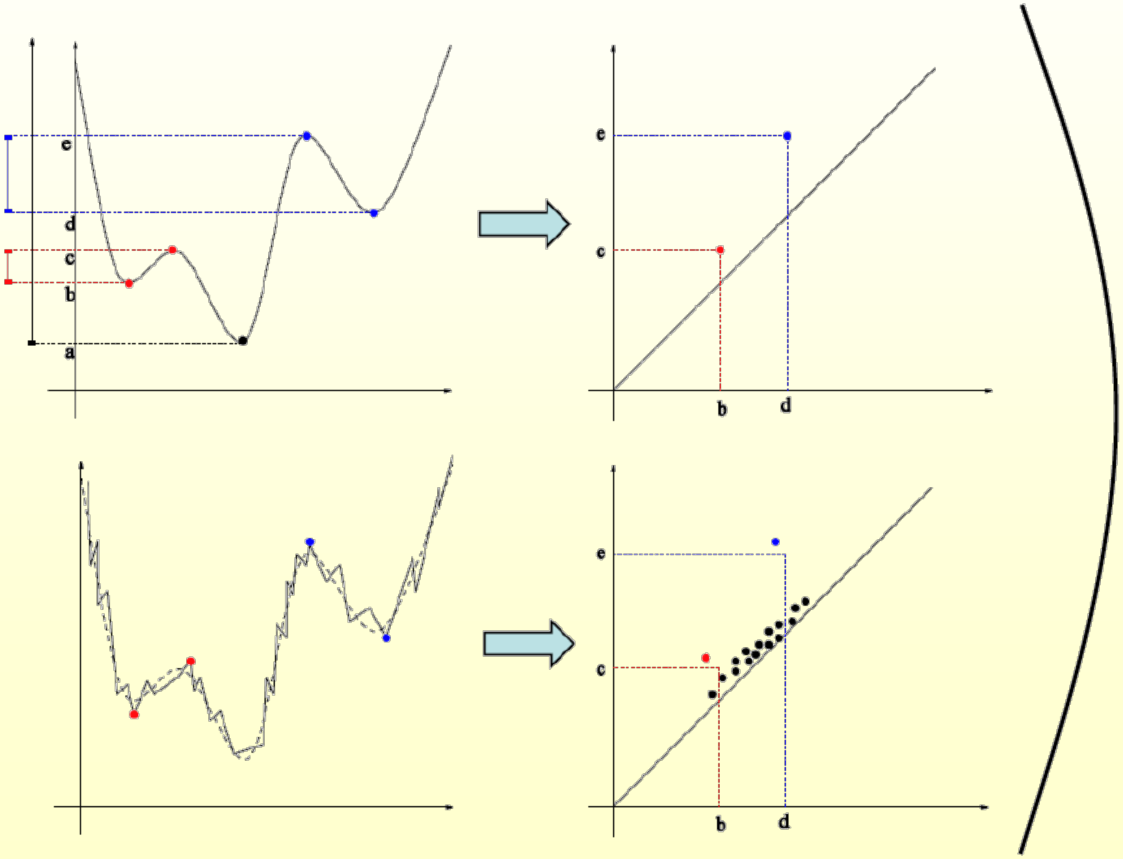


Stability: What if f is slightly perturbed?



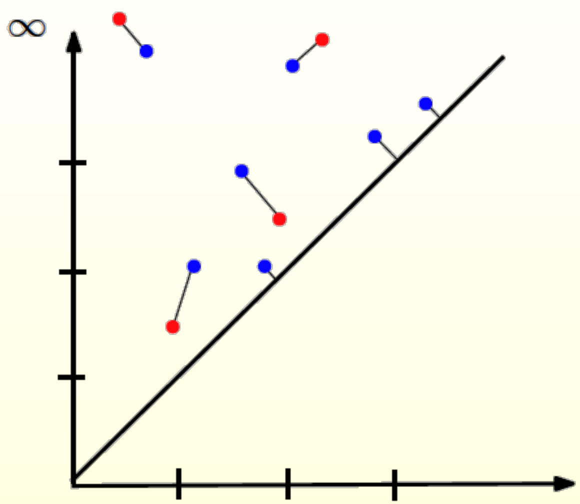
Structure Thm. [Carlsson, Zomorodian 04]
the kth persistent homology of $(\mathbb{X}, f)$ is fully described by a finite set of intervals, each of which represents the lifespan of an element in a basis that is compatible accross the filtration.

Filtering Out Topological Noise



Persistence diagrams

Bottleneck Distance Between Persistence Diagrams



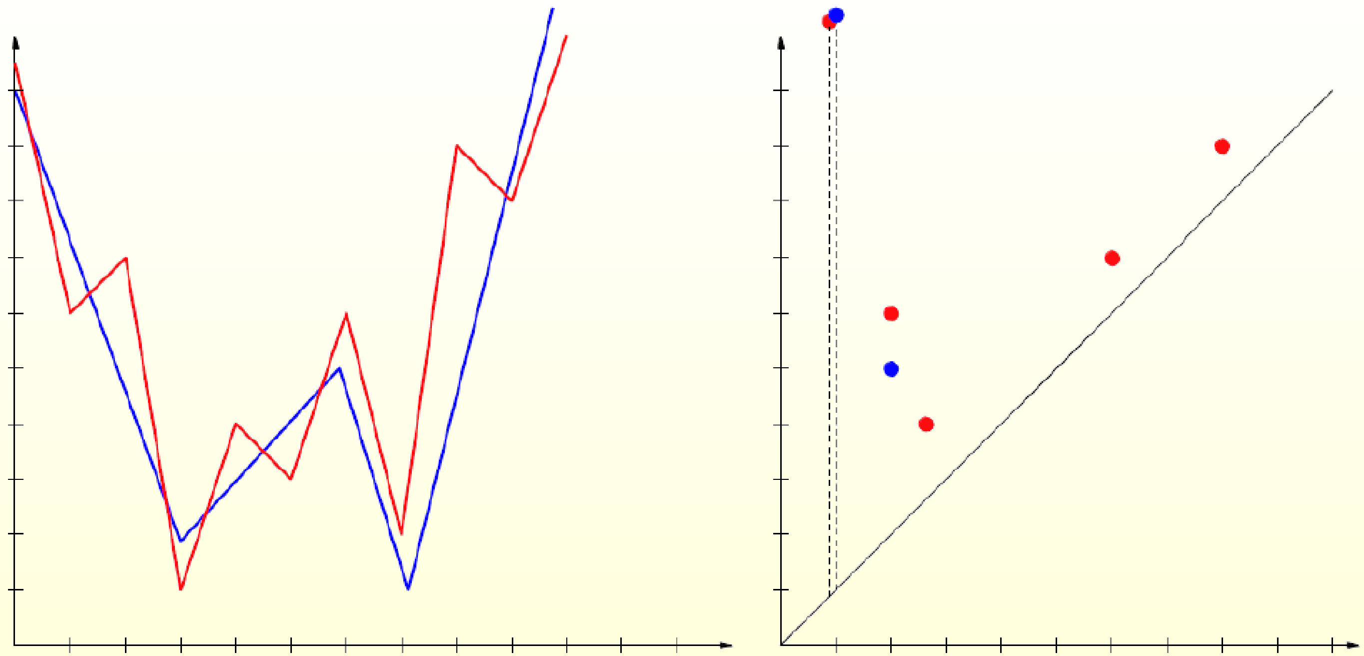
Let K be a simplicial complex and f, g two functions defined on the vertices of K. Let D_f and D_g be the persistence diagrams of f and g.

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



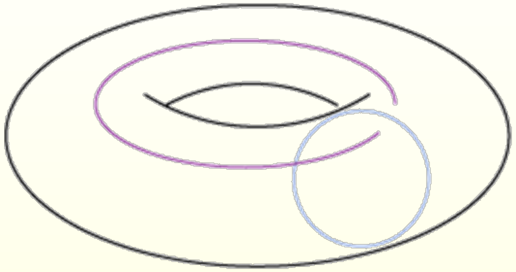
Theorem: Let K be a simplicial complex and let $f, g : K \to \mathbb{R}$.

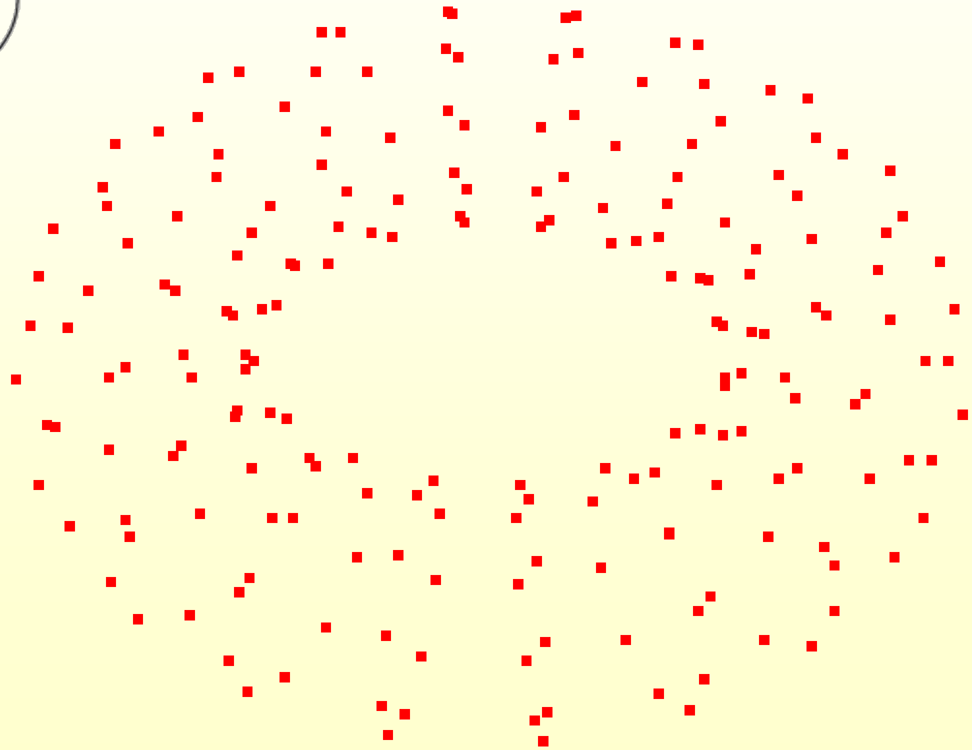
$$d_B(D_f, D_g) \leq \|f - g\|_\infty$$

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

Persistent Homology Examples

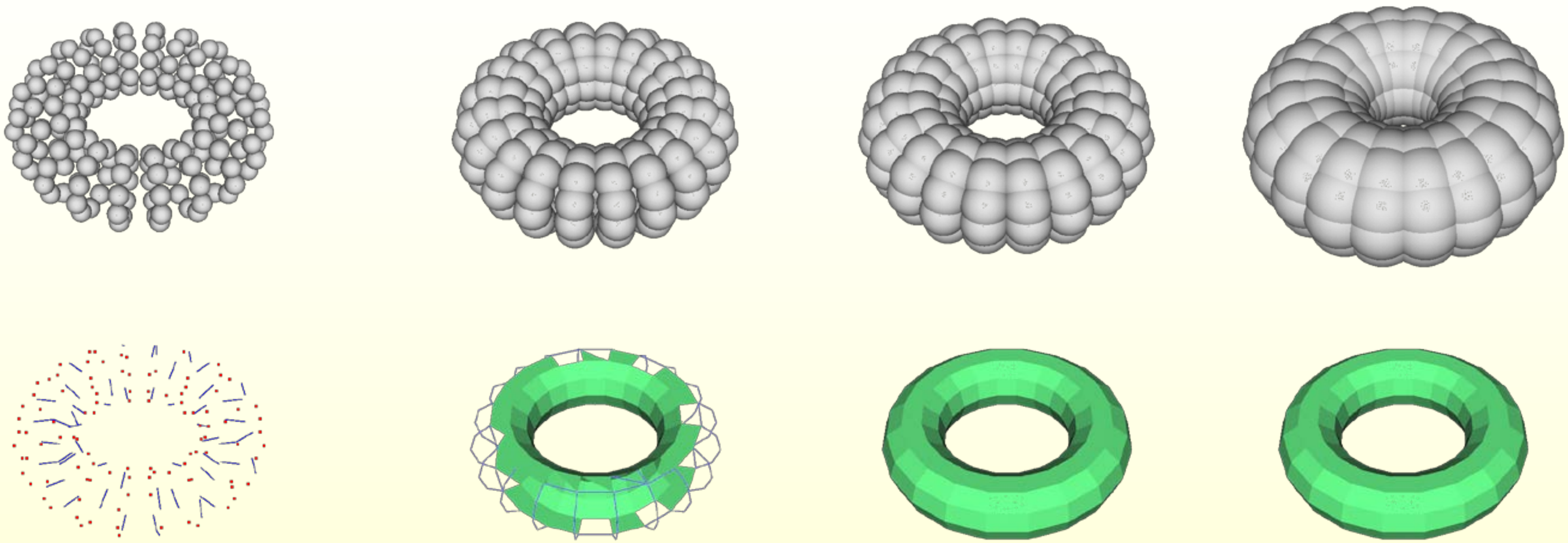
Detecting a Torus from Samples



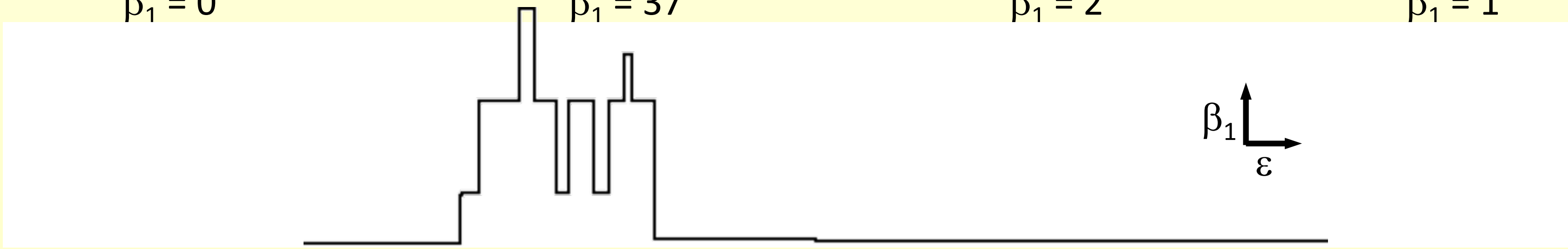


Point Cloud Data (PCD)

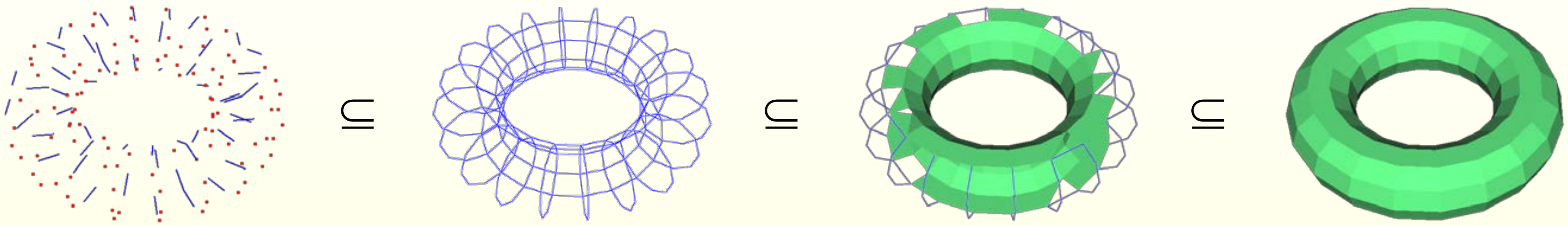
Question of Scale: A Rips Filtration



$\beta_0 = 150$	$\beta_0 = 1$	$\beta_0 = 1$	$\beta_0 = 1$
$\beta_1 = 0$	$\beta_1 = 37$	$\beta_1 = 2$	$\beta_1 = 1$



From Complex Inclusions to Homology Homomorphisms



$K^{250} \longrightarrow K^{500} \longrightarrow K^{994} \longrightarrow K^{1452}$

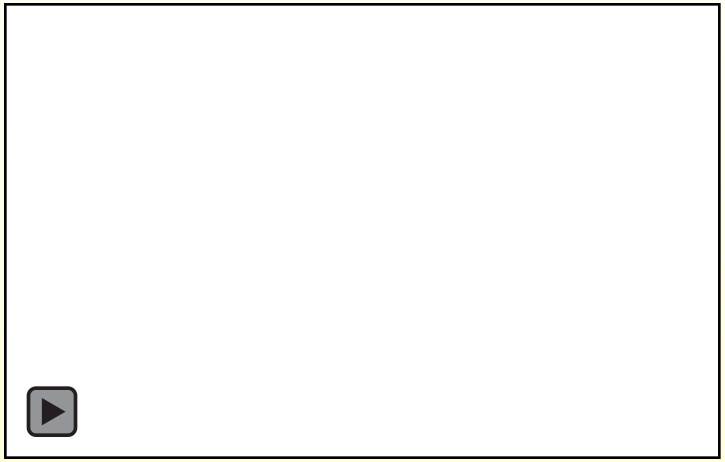
⊆

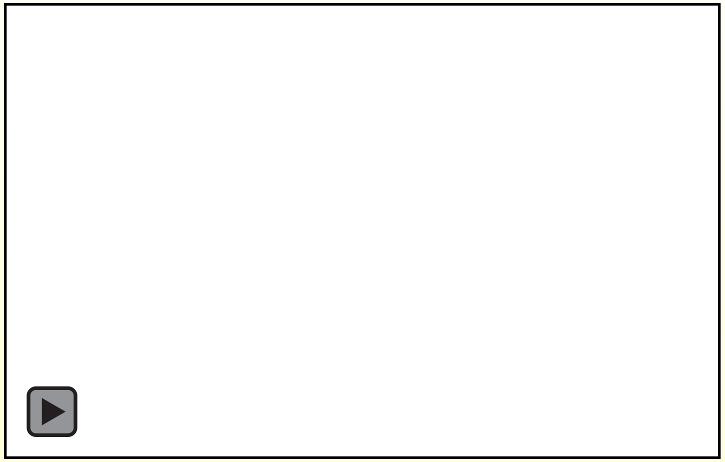
Functoriality ↓

$H_k(K^{250}) \longrightarrow H_k(K^{500}) \longrightarrow H_k(K^{994}) \longrightarrow H_k(K^{1452})$

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

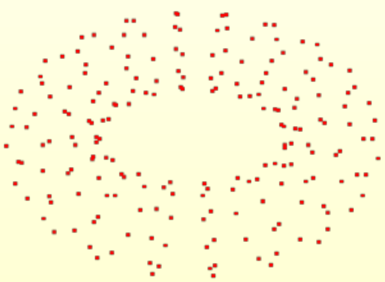
Consistent Bases Exist



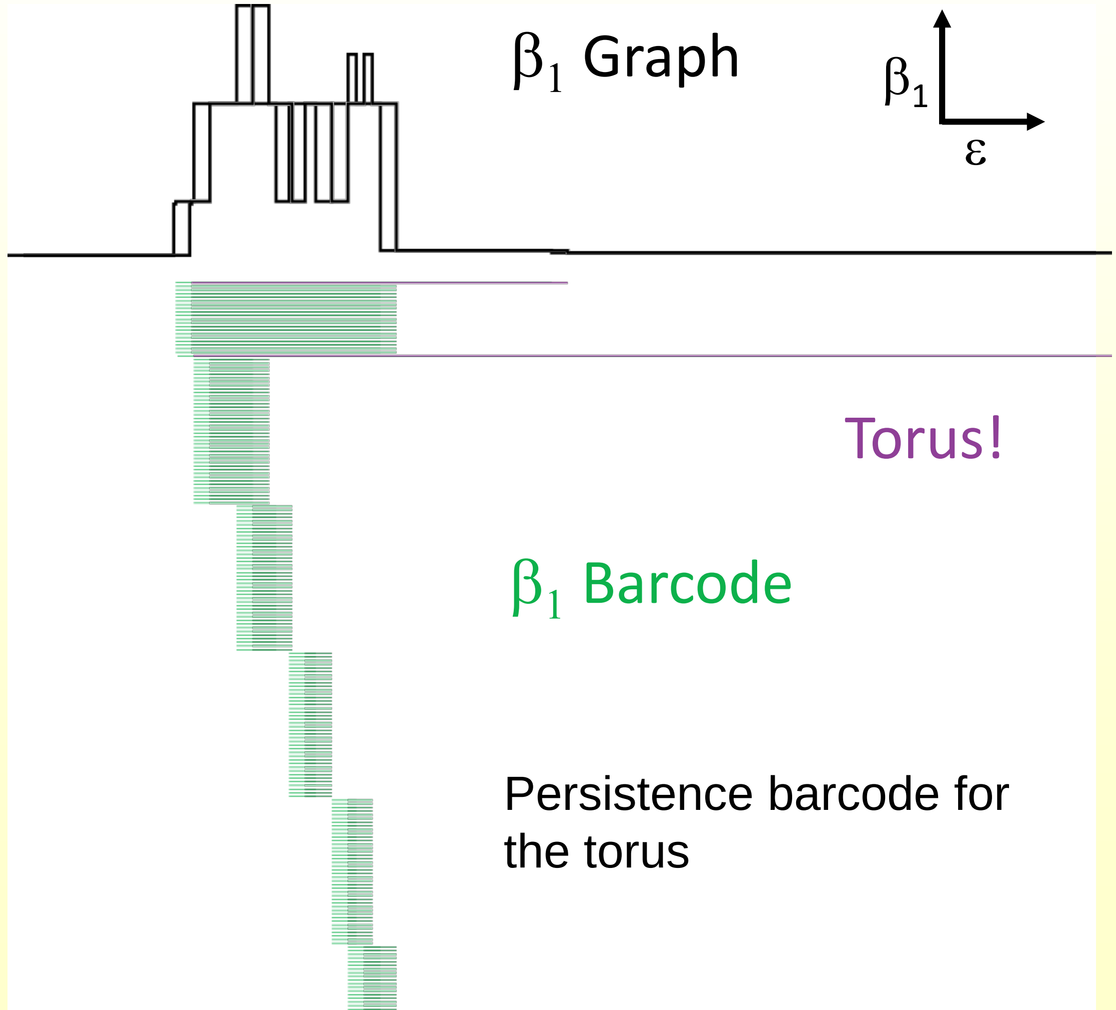


Basis elements for 1-homology

Deconstructing the Barcode

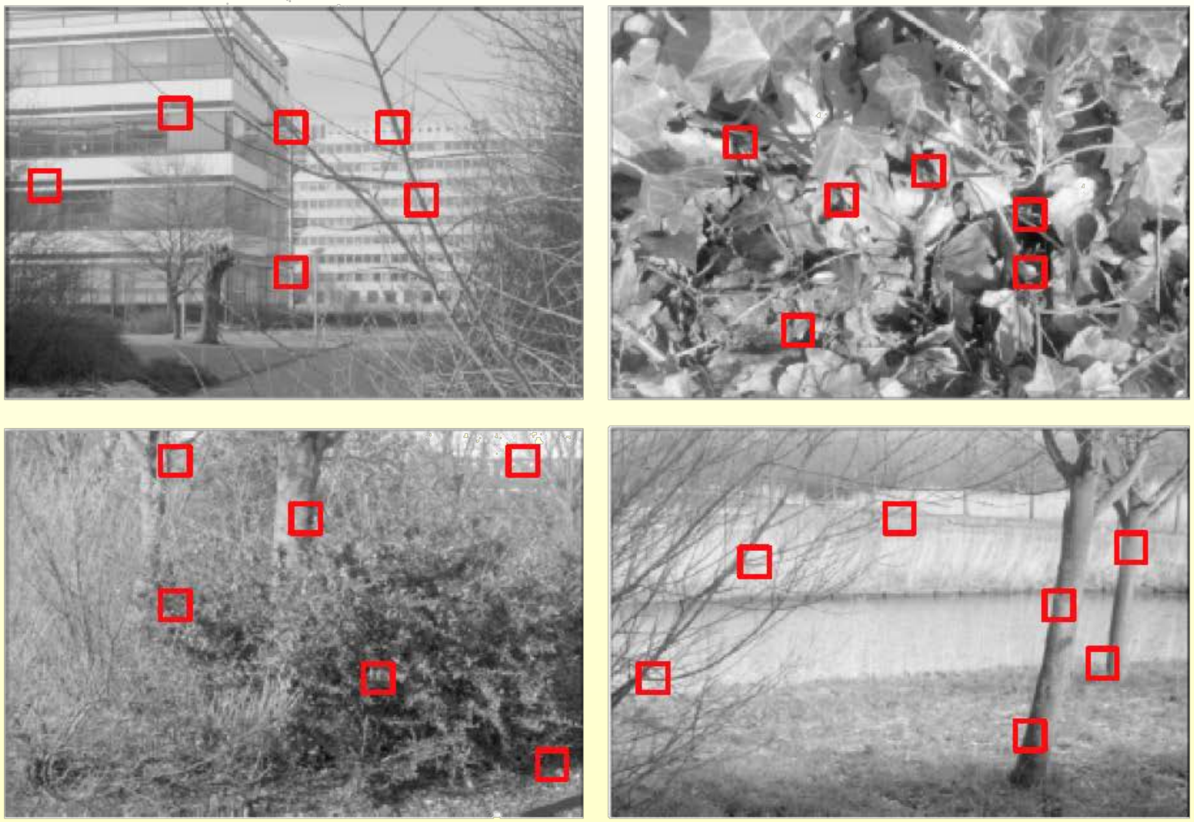


PCD



Back to the Natural Images Example

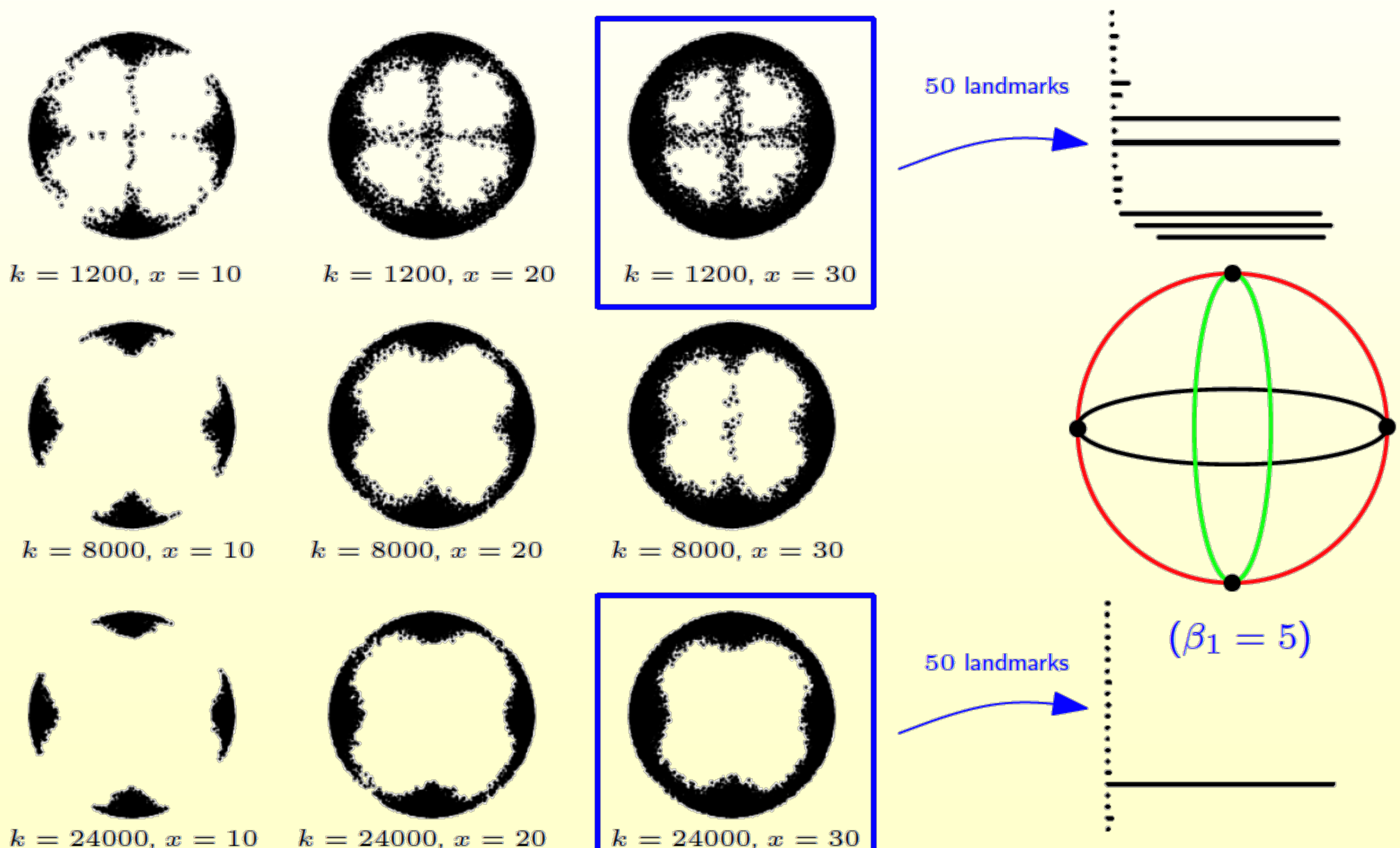
Input: 4 million data points on $\mathbb{S}^7$, coming from high-contrast 3×3 image patches



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

Back to the Natural Images Example

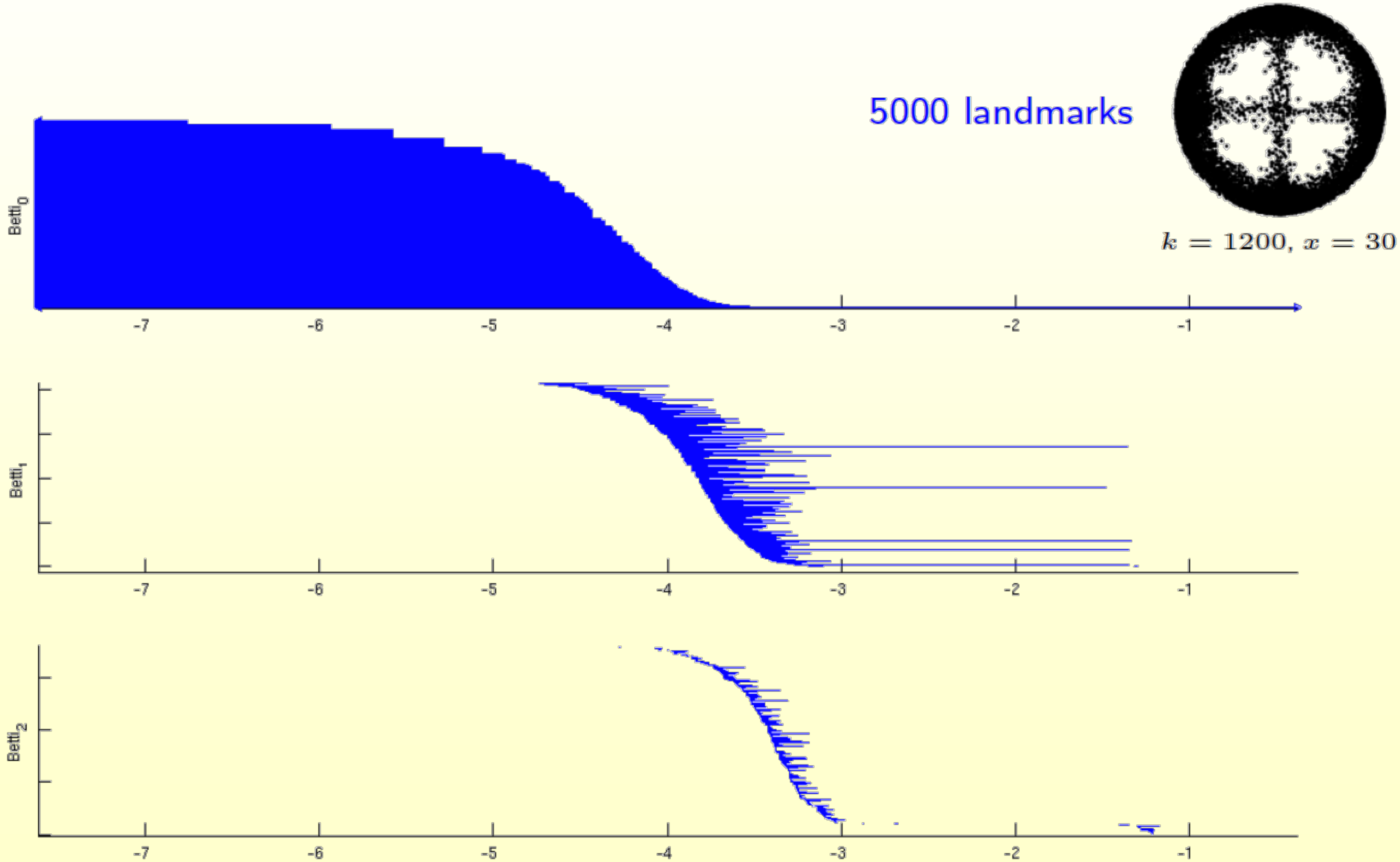
Preprocessing: - select bottom $x\%$ of data points according to k-NN distance
- sample 5000 points uniformly at random from filtered point set



(source: [de Silva, Carlsson 04])

Back to the Natural Images Example

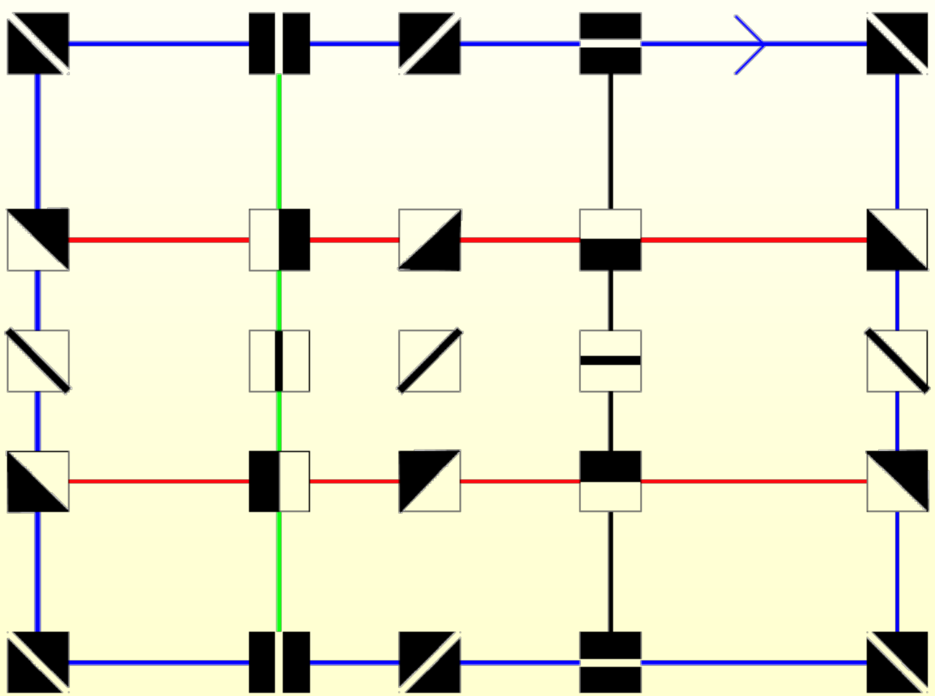
Preprocessing: - select bottom $x\%$ of data points according to k-NN distance
- sample 5000 points uniformly at random from filtered point set

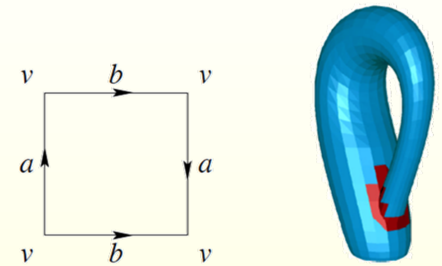


(source: [O., Sheehy 13])

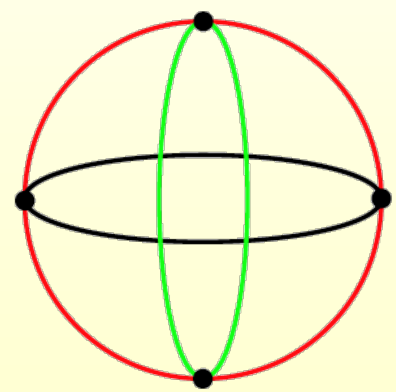
Back to the Natural Images Example

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





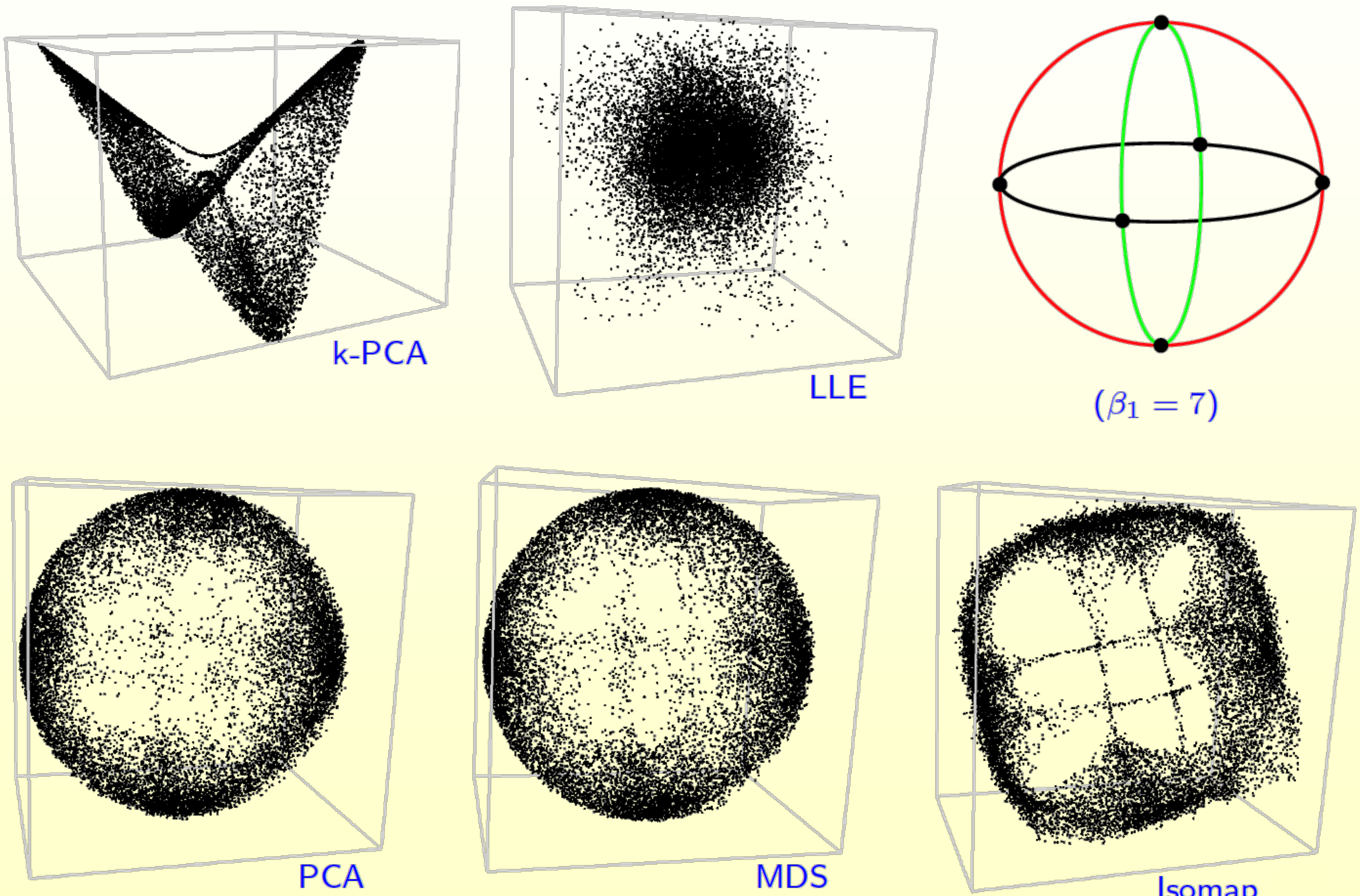
(a) Diagram (b) An immersion



$(\beta_1 = 5)$

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

FYI, Other Methods



Getting More Out of Topology

Topology for Describing Shape: A Crude Descriptor

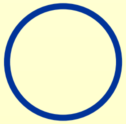
- Topology of the alphabet

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

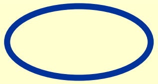
- Problem:
  - Cannot detect sharp features

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

  - Cannot detect soft features

    $\beta_1 = 1$

    $\beta_1 = 1$

Making Topology a Finer Tool

Geometry discriminating | Topology classifying

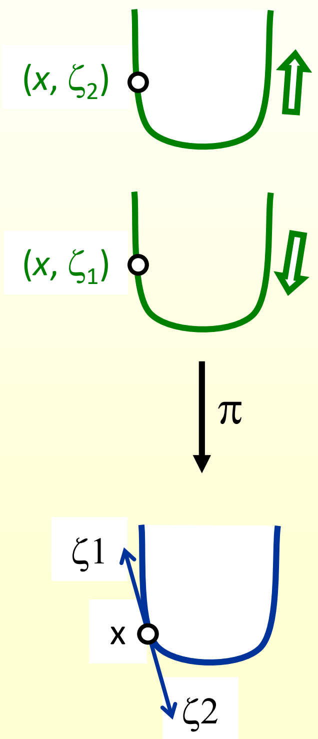
- Topology: connectivity of a space
- Key Idea: no reason to look at the original space only
  - Add geometry ⇒ look at derived space(s)
  - Compute topology of derived space(s)

1. Find filtration
2. Compute persistence

via the tangent complex

Our recipe

2-D Curve Tangent Complex

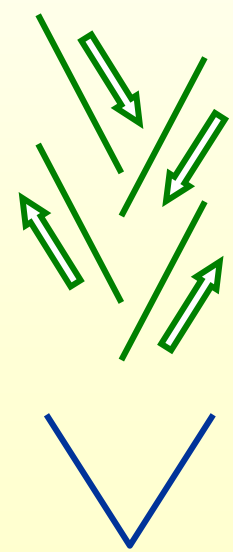


T(X) has two components:
$\beta_0(T(X)) = 2$

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

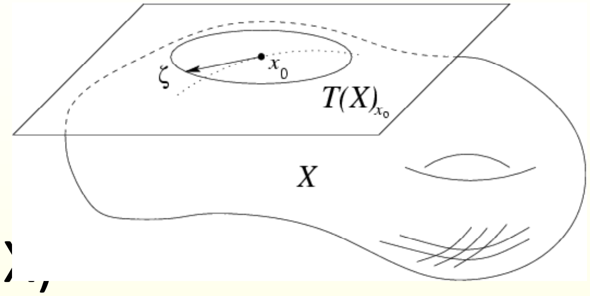
Every point x on a smooth curve X has two tangent directions.

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



Covering space

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 $1/\rho \le \delta$.
  - $T_\delta(X)$: closure of $T_\delta{}^0(X)$

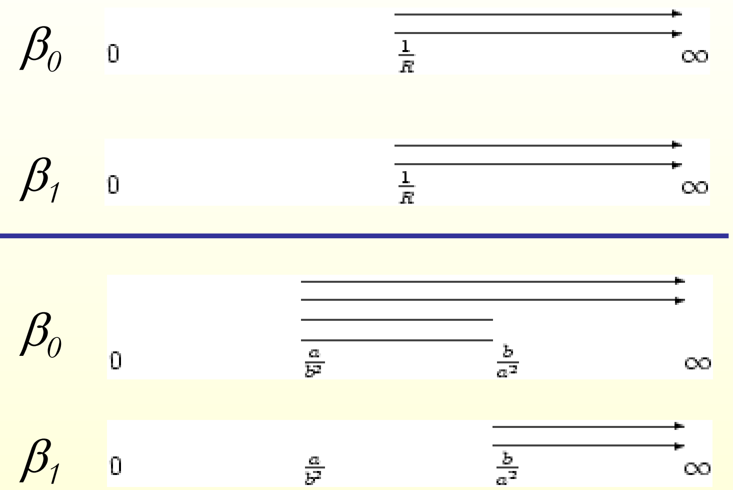
- Filtered tangent complex $T^{filt}(X)$ is the family

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

Persistence Barcodes: Circle vs. Ellipse

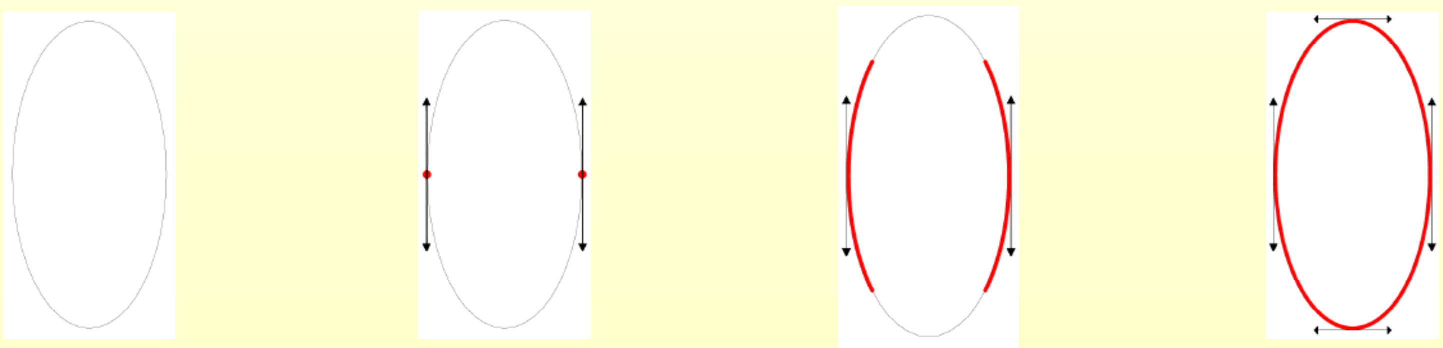
T^{filt}(circle of radius R) is simple:
the entire complex (2 copies of circle) appears at once, at $\delta = 1/R$.

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



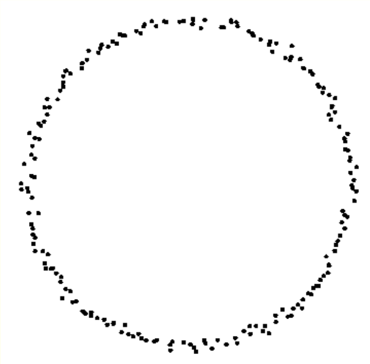
Persistence Barcodes

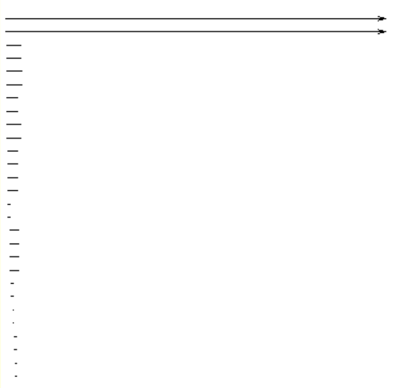
$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$



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

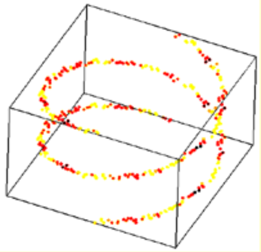
Applying Barcodes to 2D PCDs

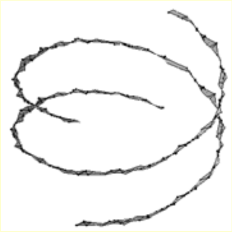




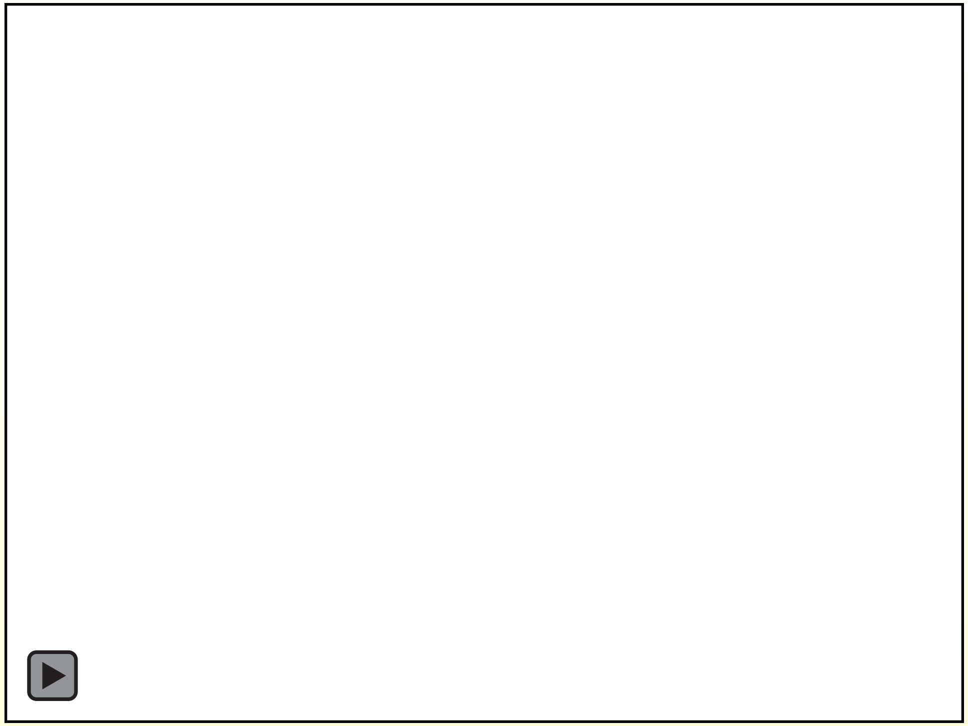
Input: Shape

Output: Descriptor



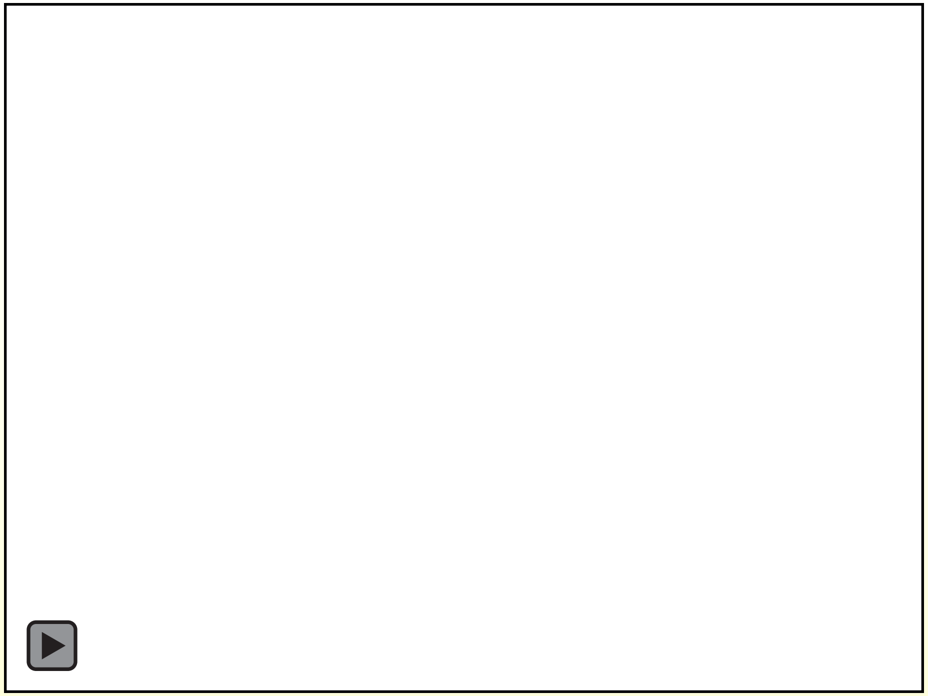


Fibers



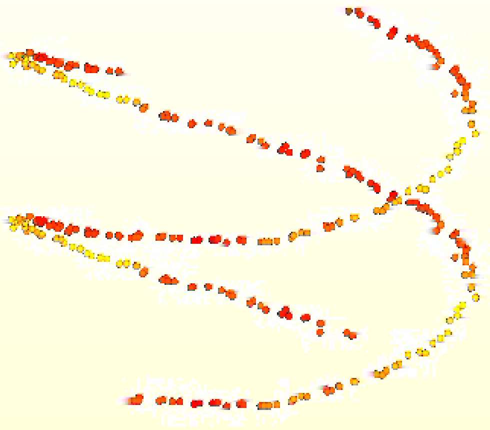
- PCD $P \subseteq X$, sampled from smooth closed 1-manifold
- We compute tangent fibers $\pi^{-1}(P)$ by normal estimation at each point

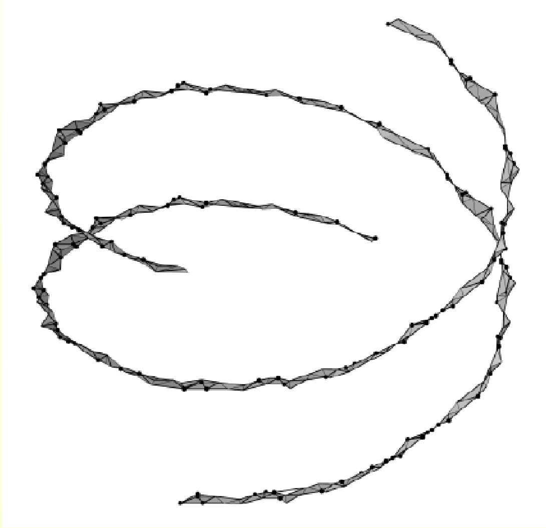
Filtering by Curvature



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

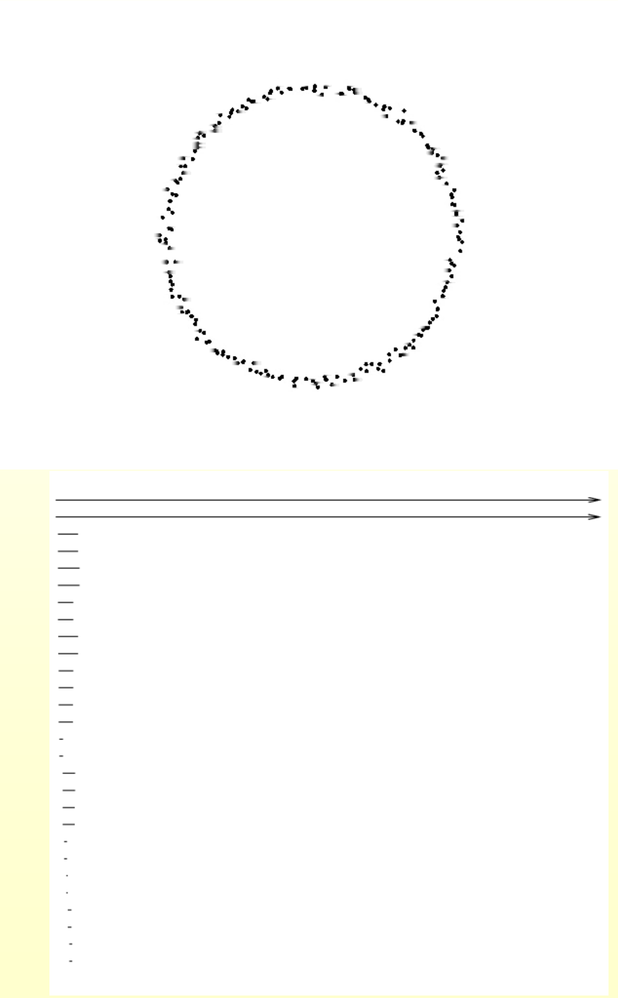
Approximating *T(X)*

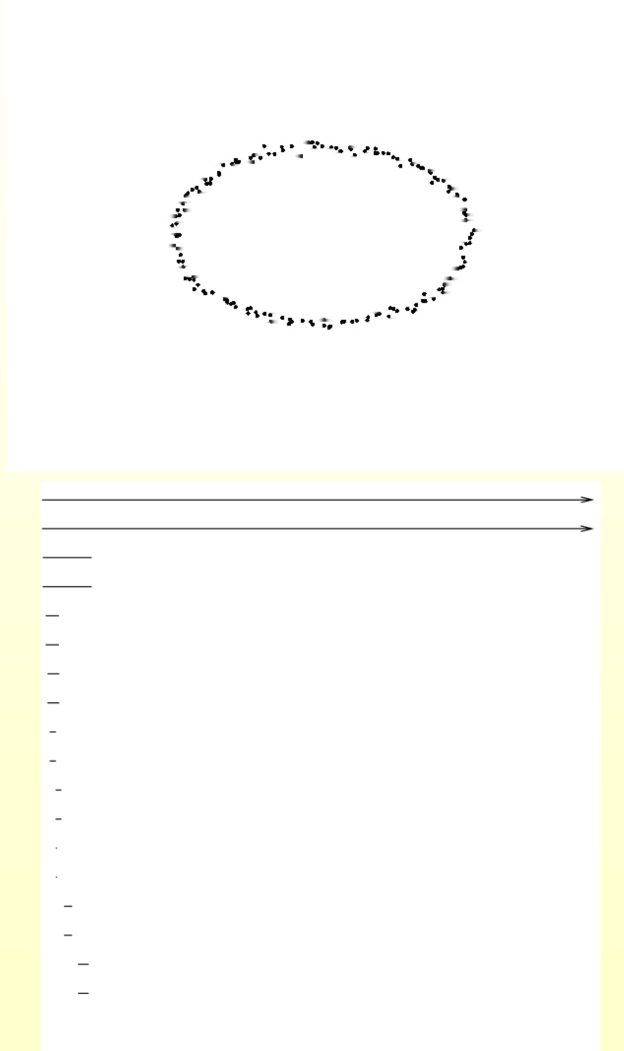


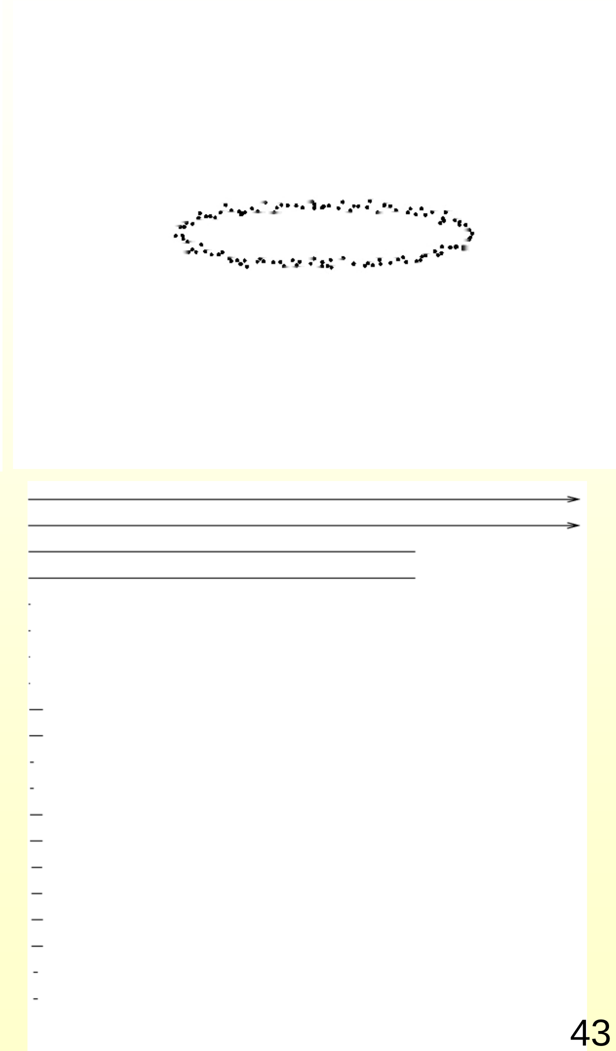


- $R^n \times S^{n-1}$ with $ds^2 = dx^2 + \omega^2 \, d\zeta^2$
- $T(X) \cong \bigcup_{p \in \pi^{-1}(P)} B_\varepsilon(p)$

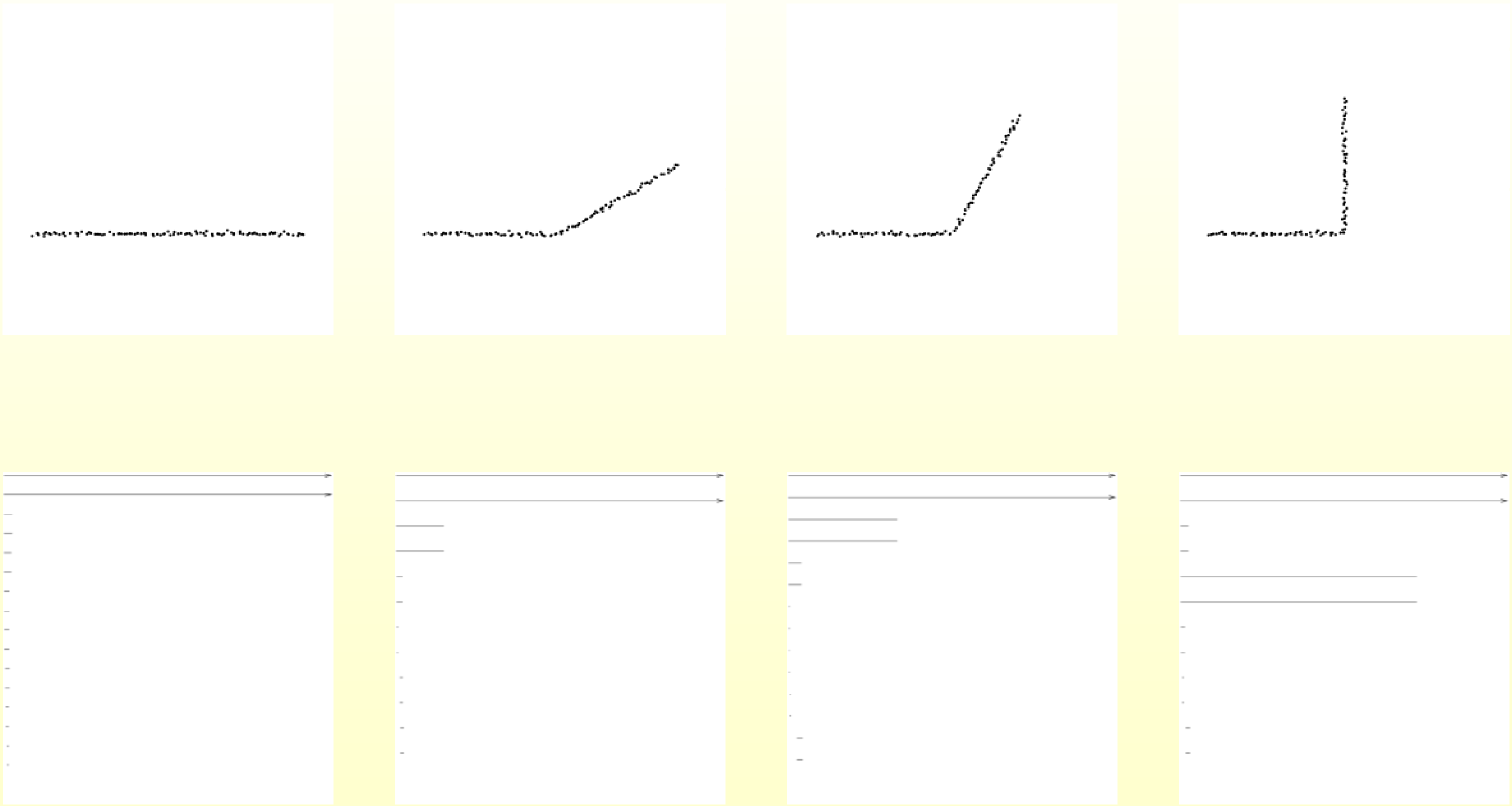
Family of Ellipses







Articulated Arm Parametrization



The Mapper Algorithm

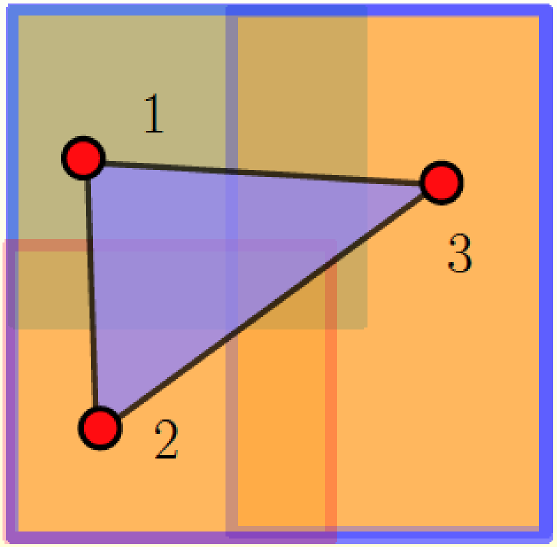
Review: Covers and Nerves

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: $A \supseteq \sigma \in N(\mathcal{U}) \Leftrightarrow \bigcap_{\alpha \in \sigma} U_\alpha \neq \emptyset$.



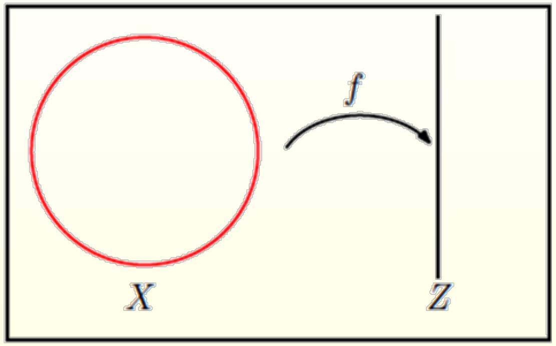
Pullback Covers and Their Nerves

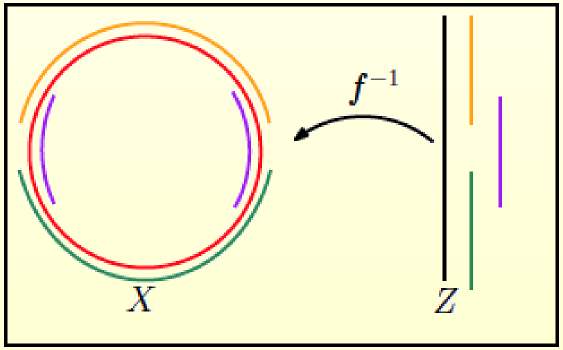
Studying data by looking at “lens” functions over the data

- 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:

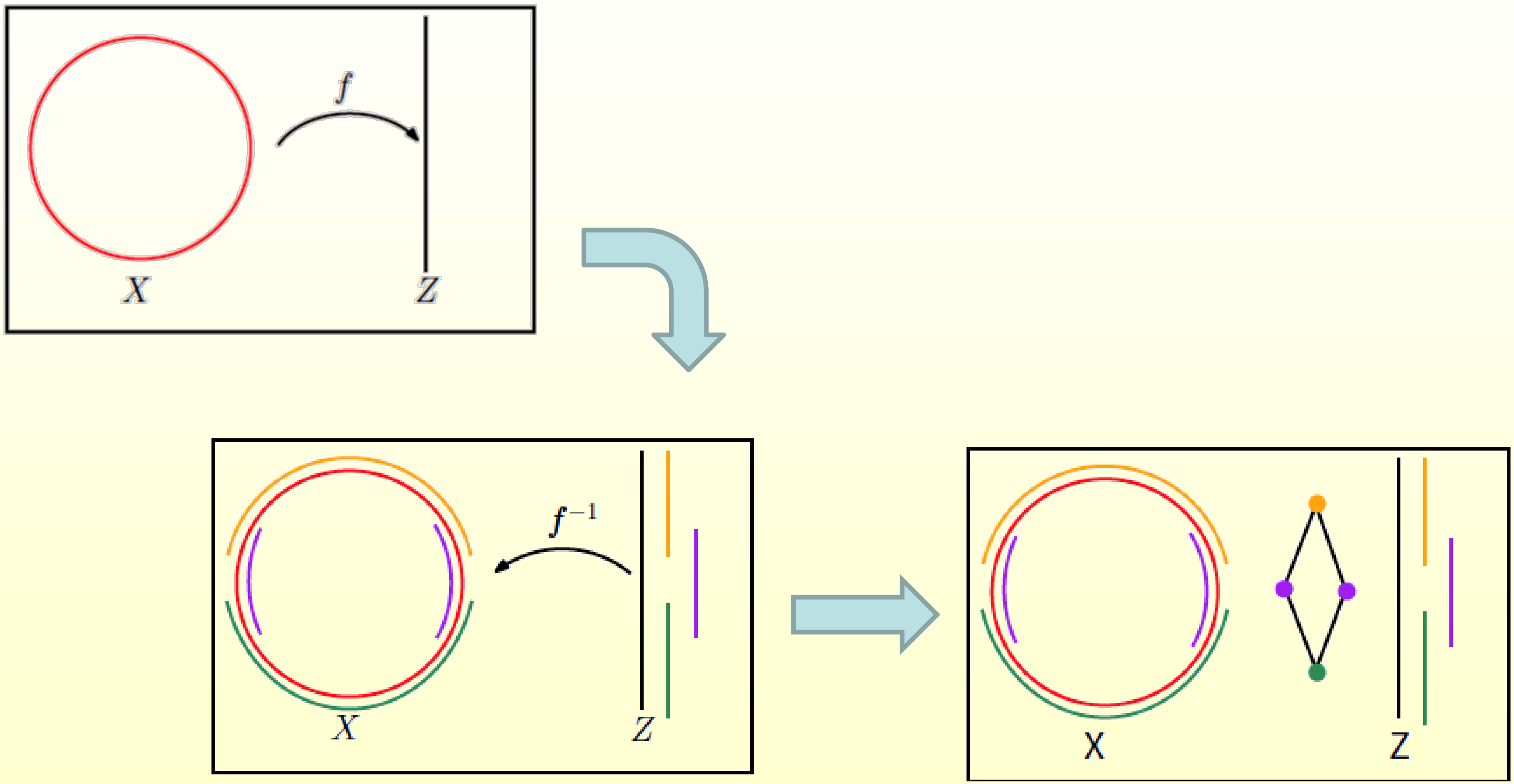
  $$f^*(\mathcal{U}) := \{V_{i,\alpha},\, 1 \leq i \leq j_\alpha,\, \alpha \in A\}.$$

  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.

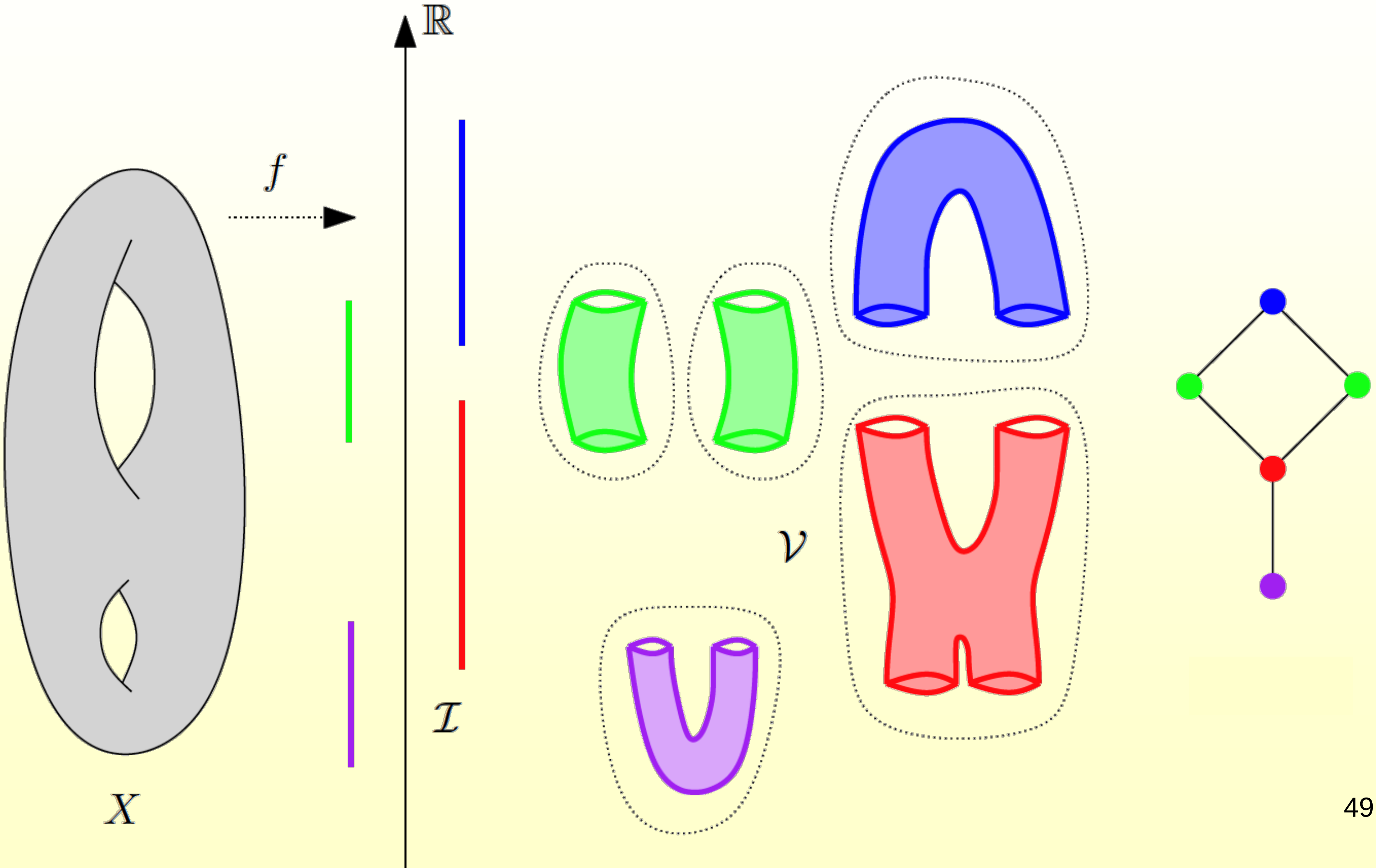




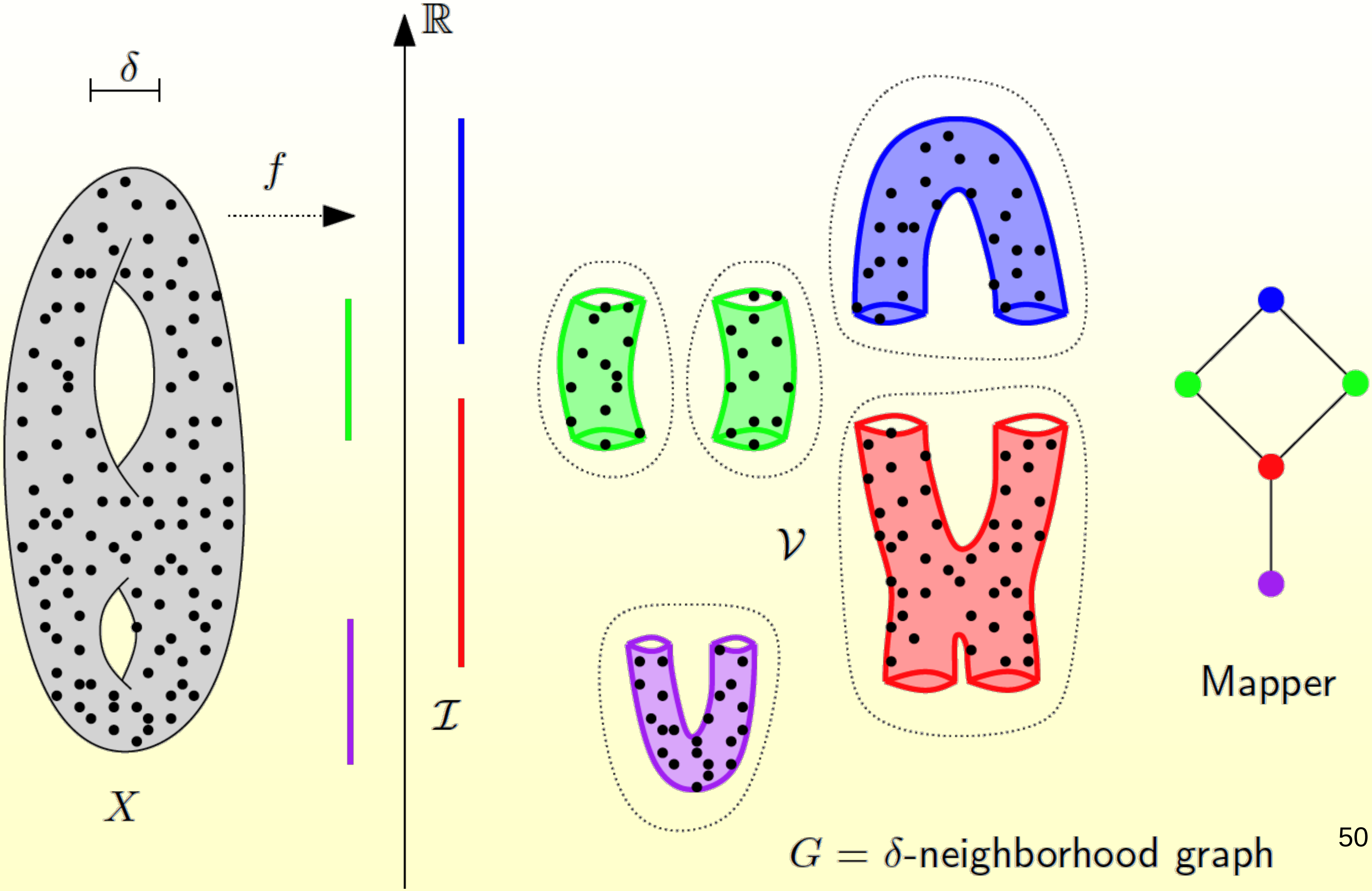
Pullback Covers and Their Nerves



Another Example



In Practice



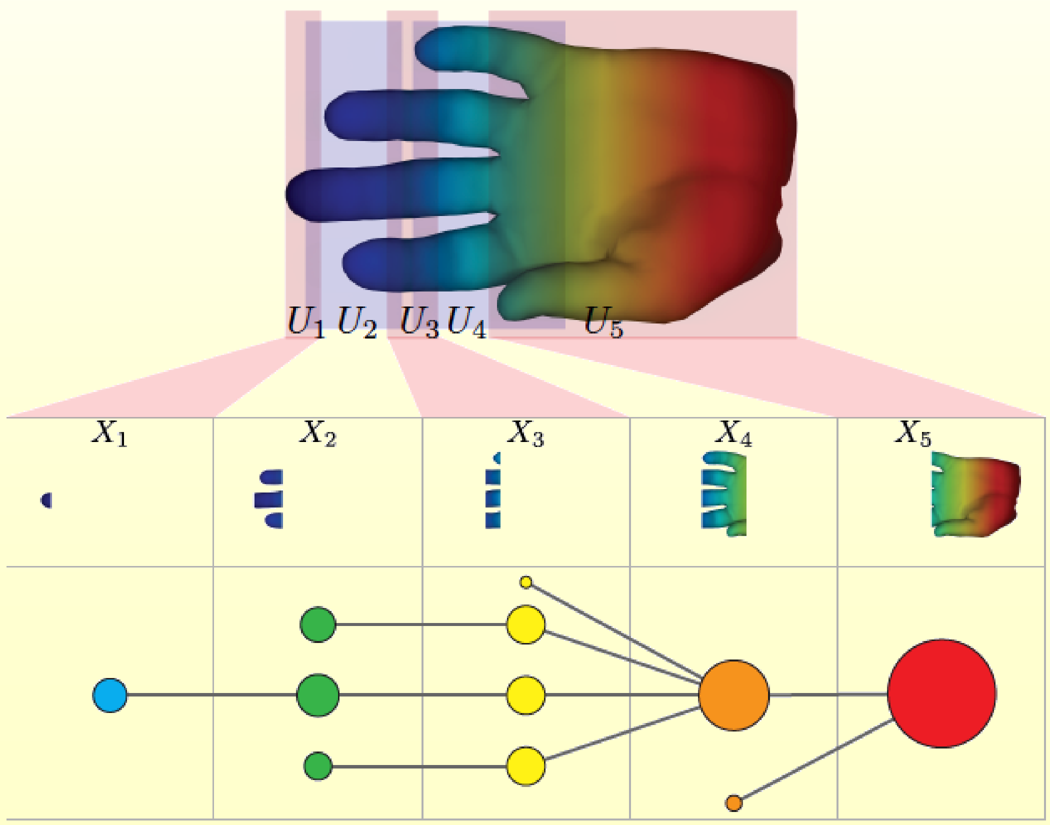
$G = \delta$-neighborhood graph

The Mapper Algorithm

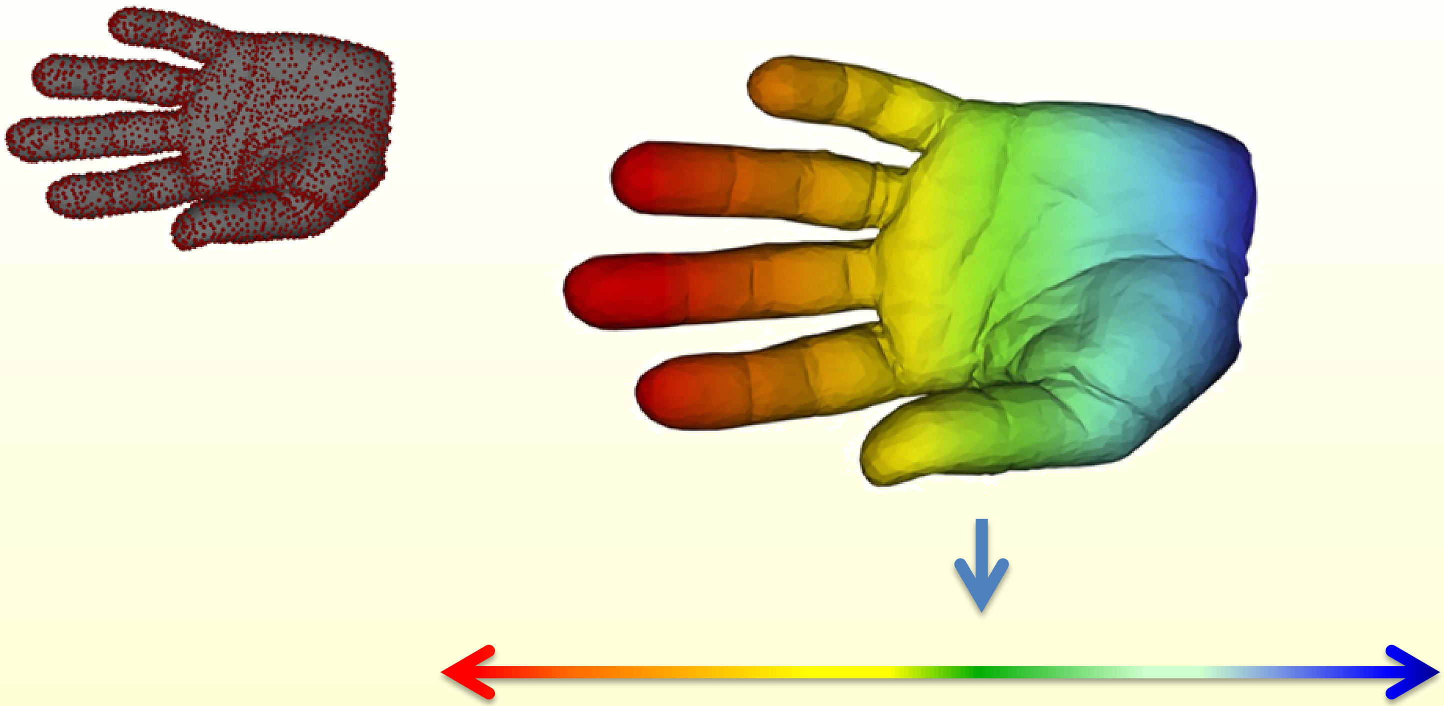
(Carlsson, Mémoli, Singh 2007)

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

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



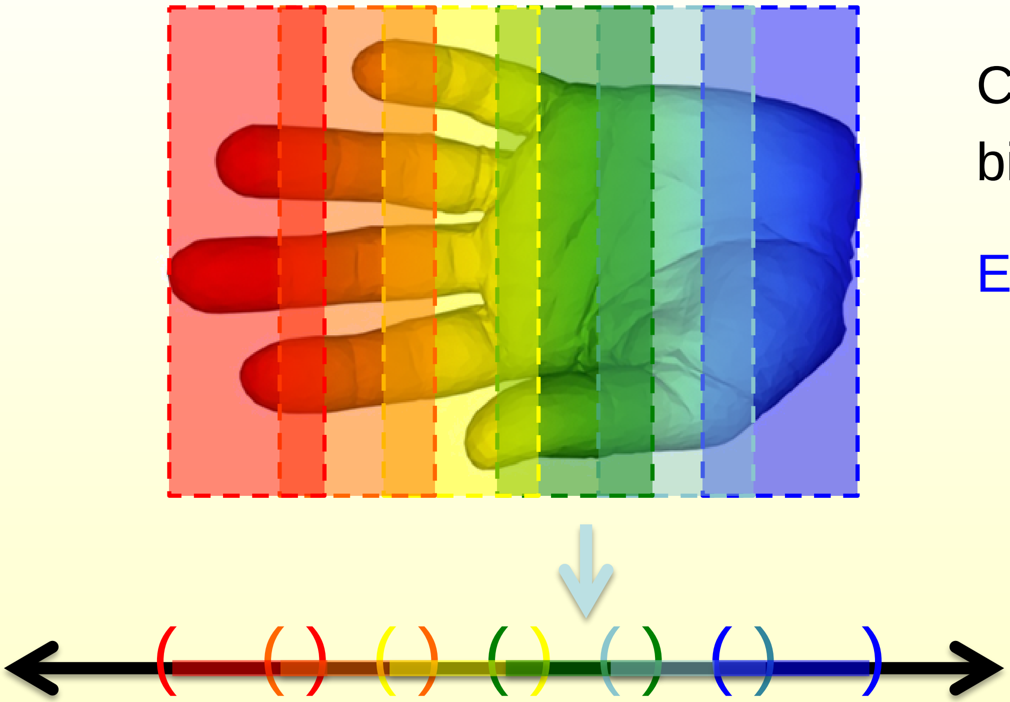
Step 1: Choose a Lens / Filter Function



Function f : Data Set → **R**

Ex 1: x-coordinate f : (x, y, z) → x

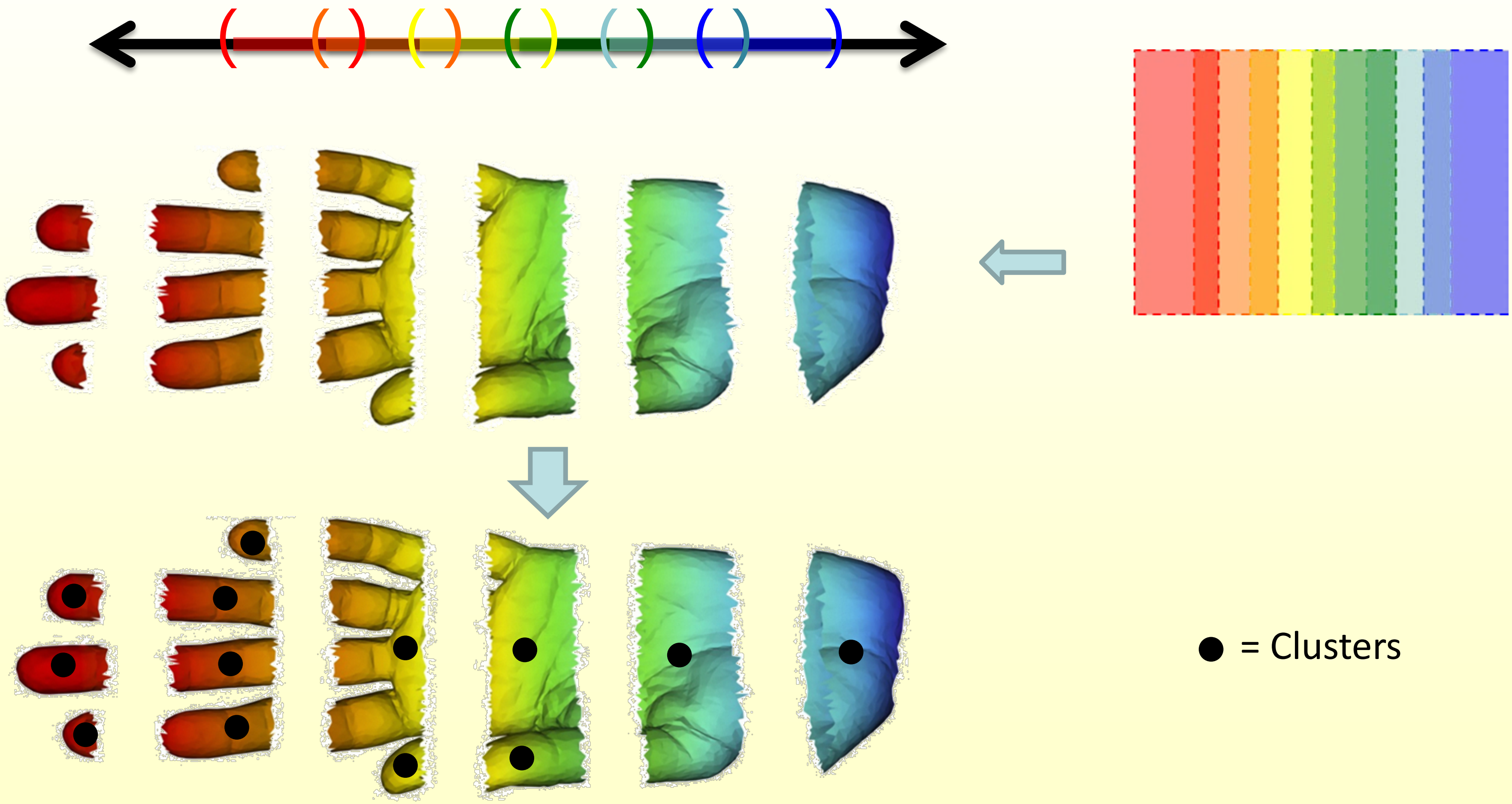
Step 2: Partition into Overlapping Bins



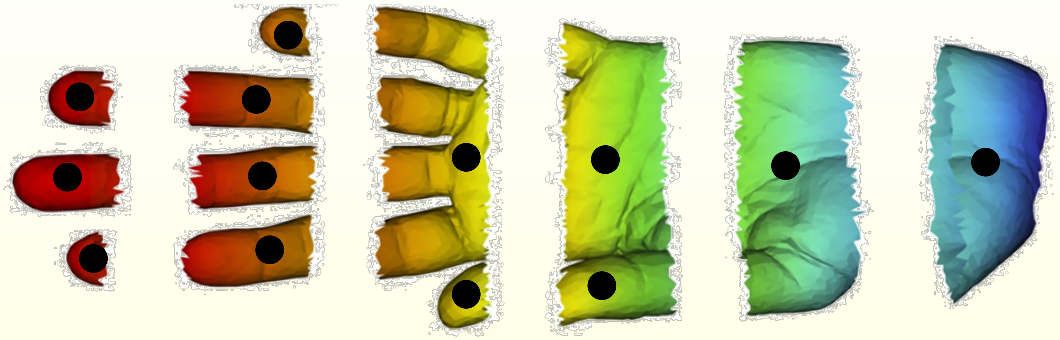
Cover data via overlapping bins.

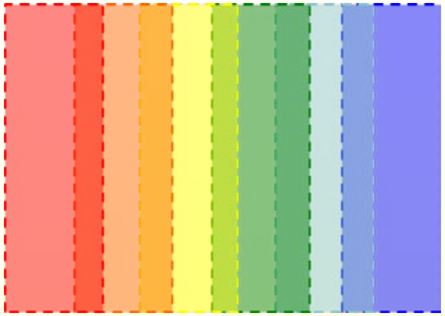
Example: $f^{-1}(a_i, b_i)$

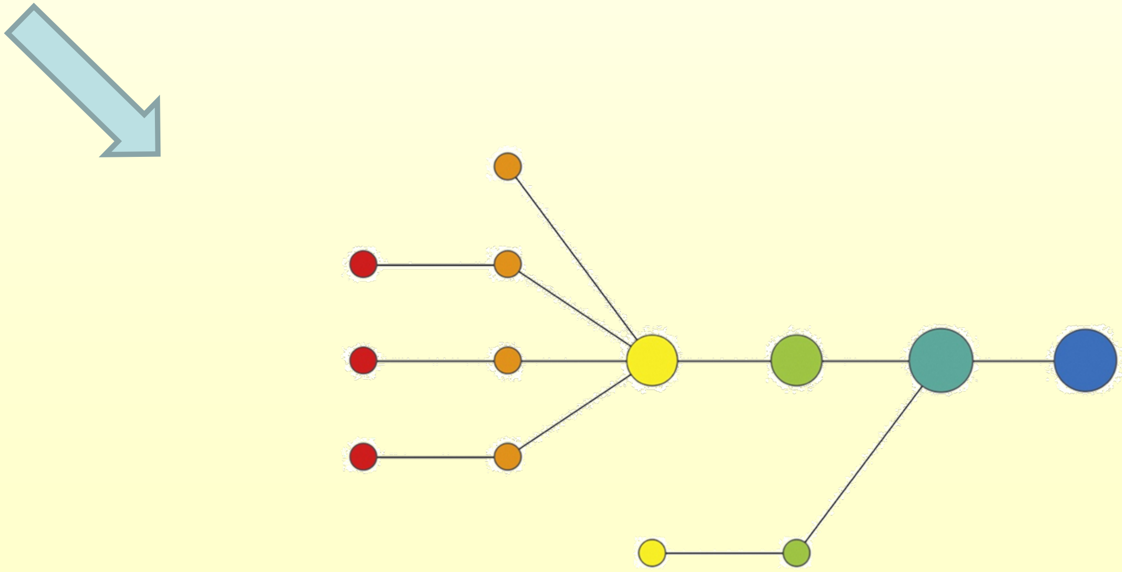
Step 3: Form Connected Components in the Bins



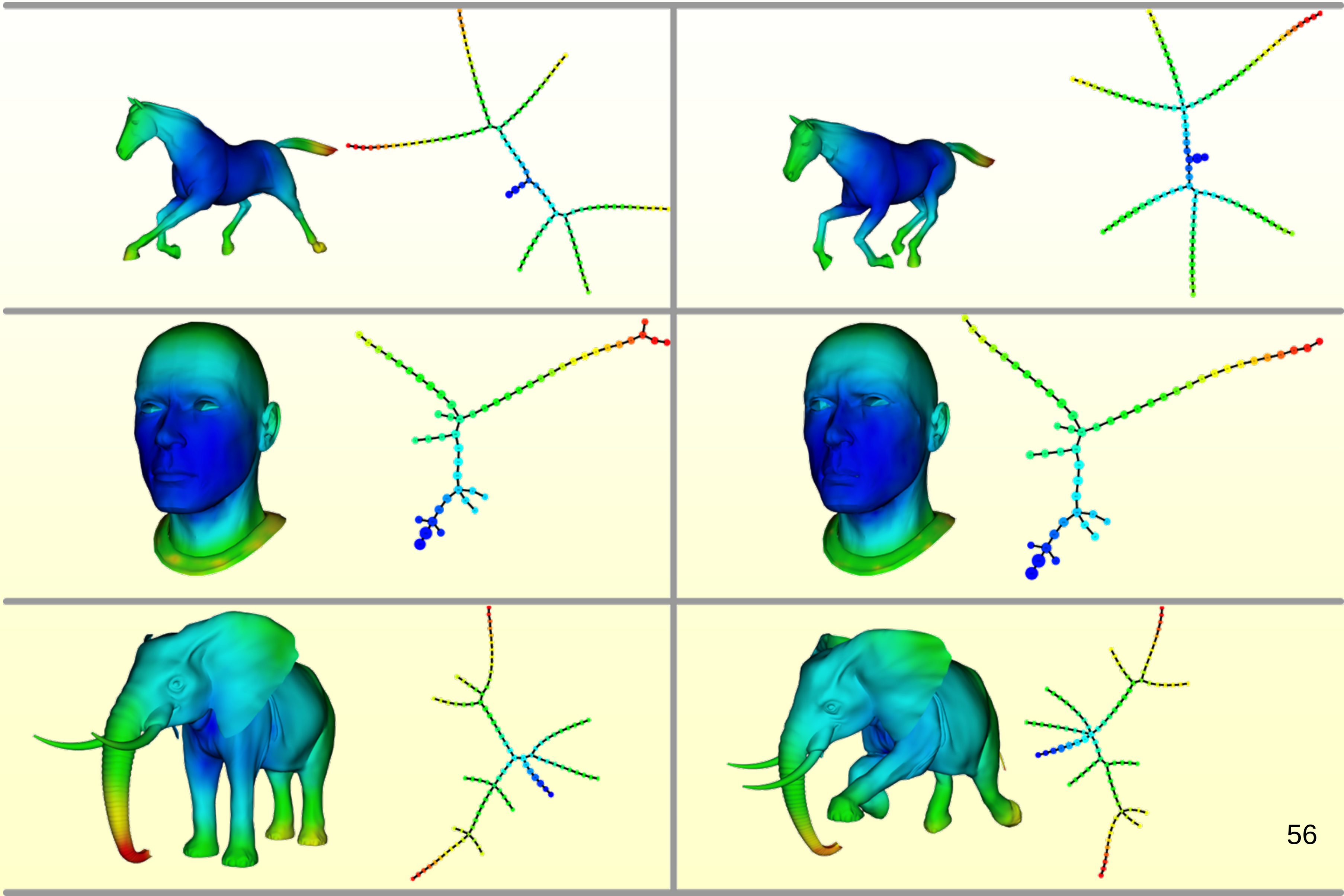
Step 4: Form a Network of Intersecting Clusters





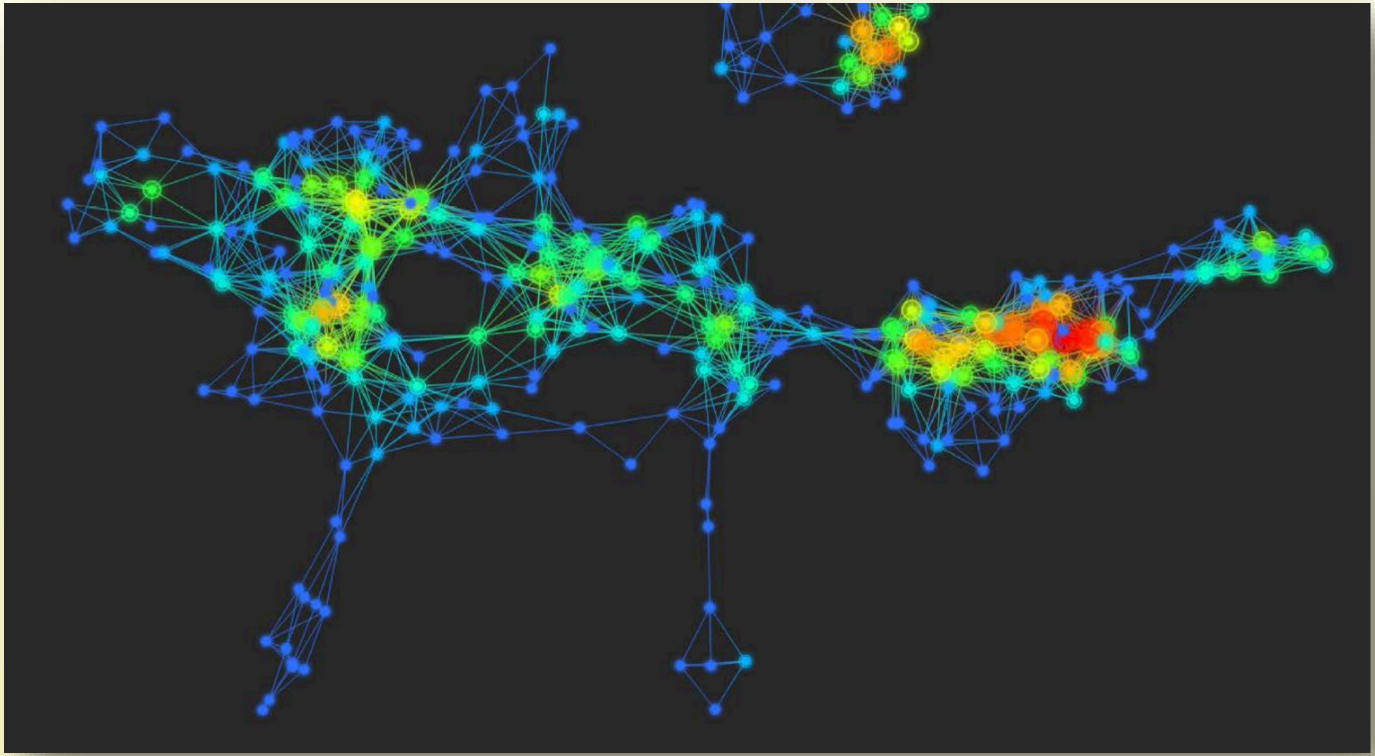


Centrality Filter Under Deformation



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."

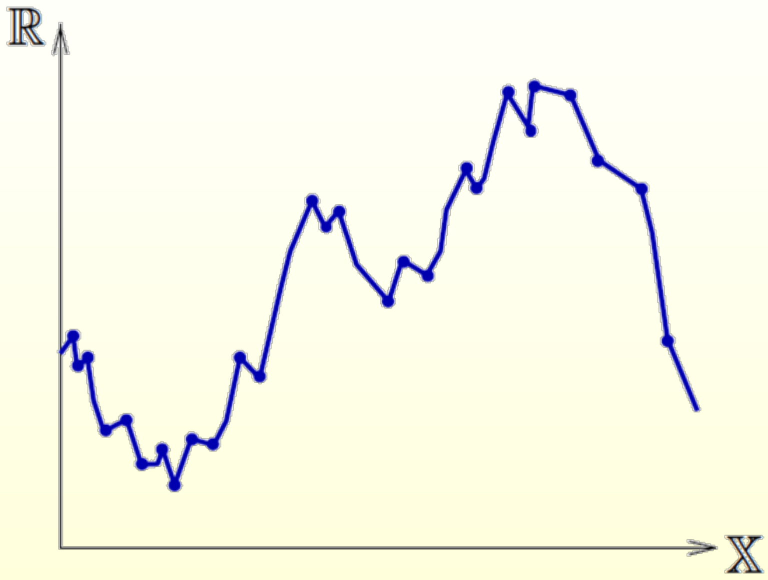


Persistence-Based Segmentation

3D Shape Segmentation

Partition a 3D model into meaningful components

Computing Segments

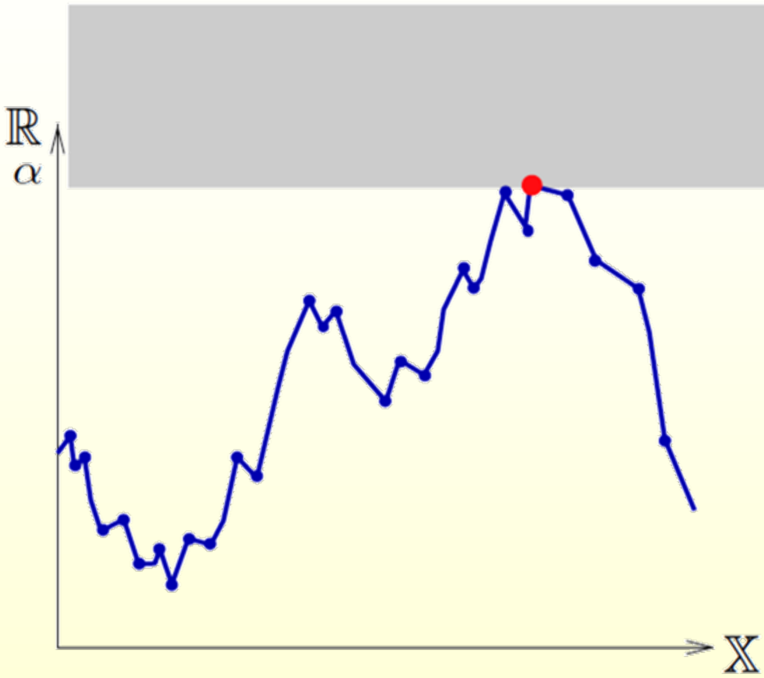


Use descriptor functions on shapes

How do we compute segments from a PD?

Do not merge segments with persistence greater than a threshold τ!

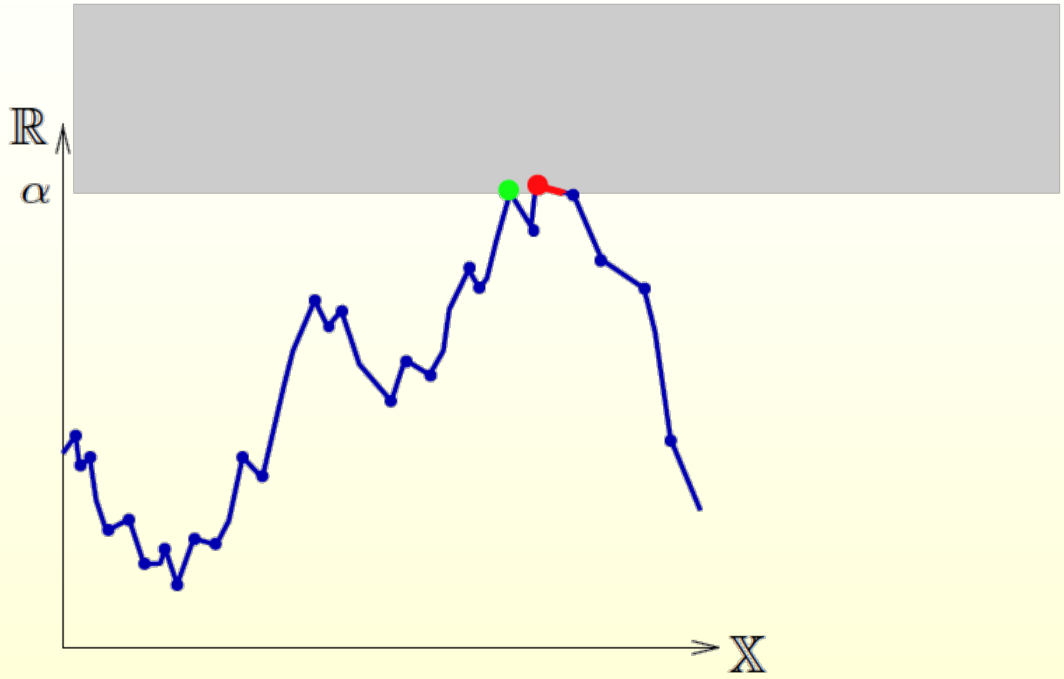
Computing Segments



How do we compute segments from a PD?

Do not merge segments with persistence greater than a threshold τ!

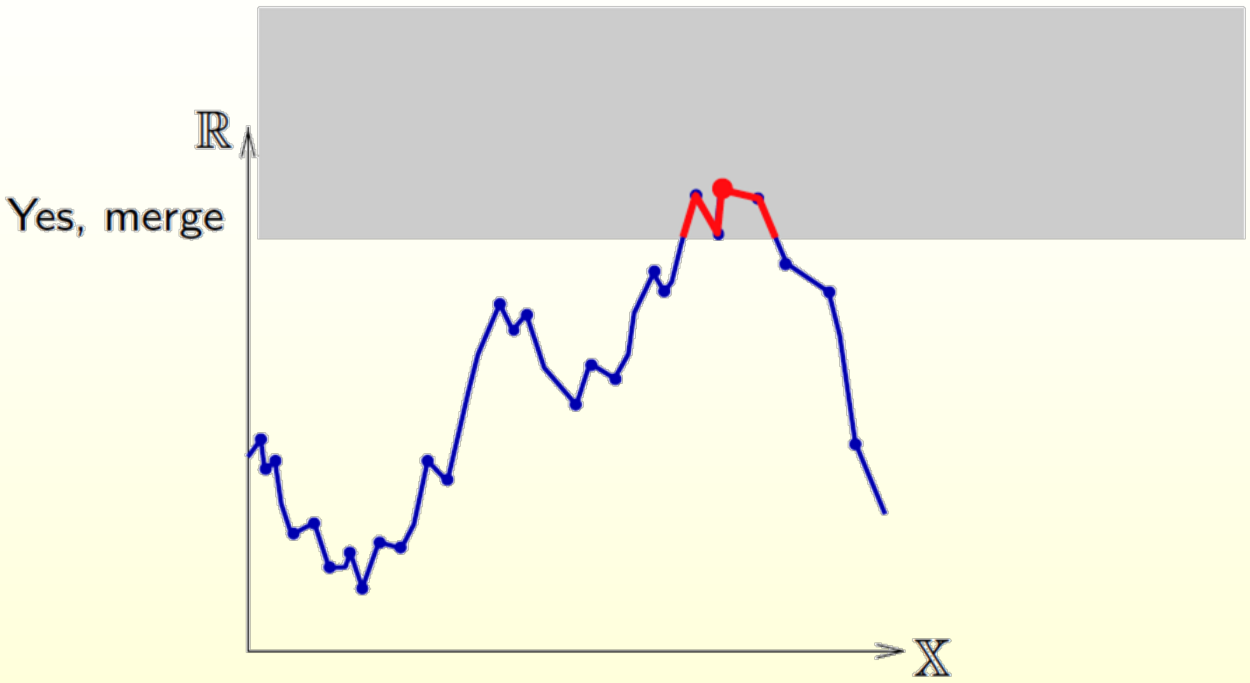
Computing Segments



How do we compute segments from a PD?

Do not merge segments with persistence more than a threshold τ!

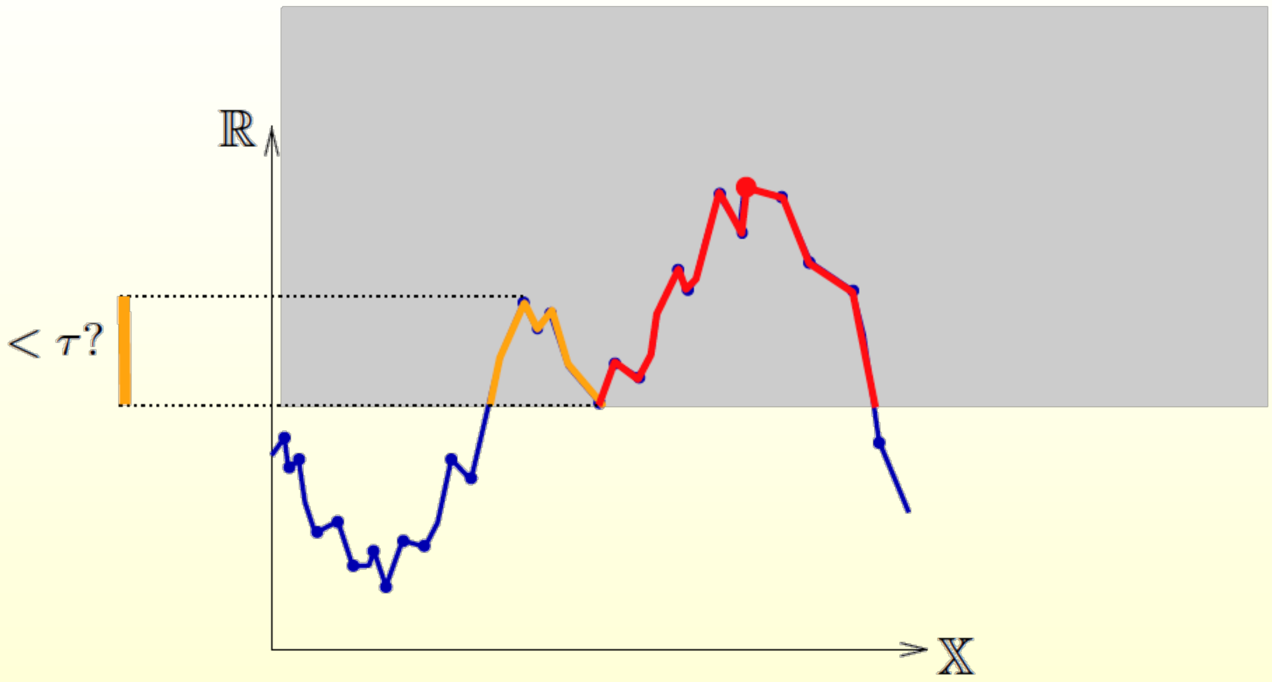
Computing Segments



How do we compute segments from a PD?

Do not merge segments with persistence greater than a threshold τ!

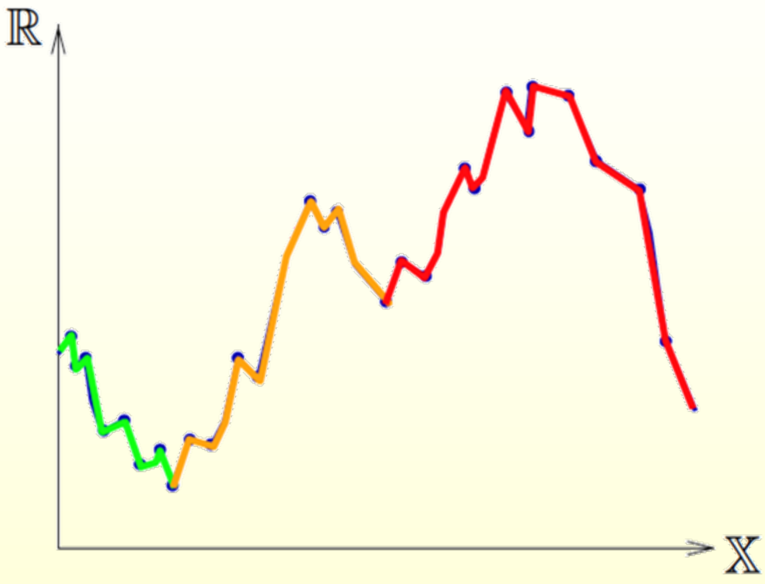
Computing Segments

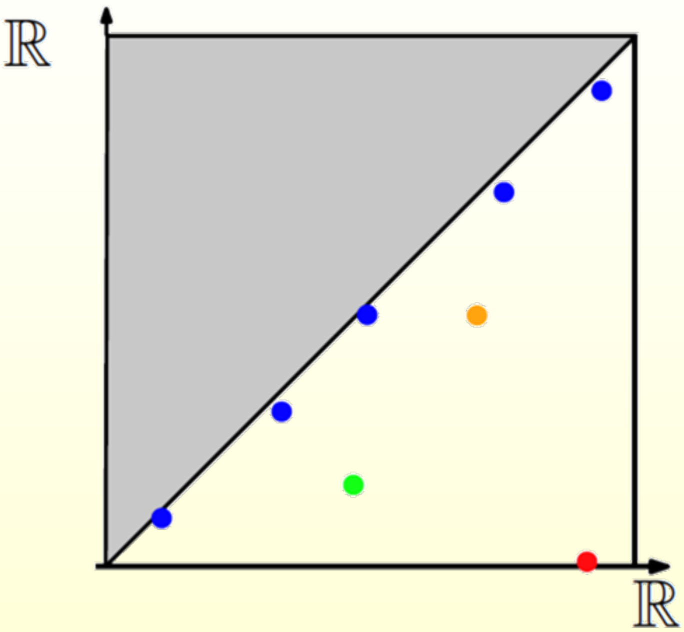


How do we compute segments from a PD?

Do not merge segments with persistence greater than a threshold τ!

Computing Segments





How do we compute segments from a PD?

Do not merge segments with persistence greater 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

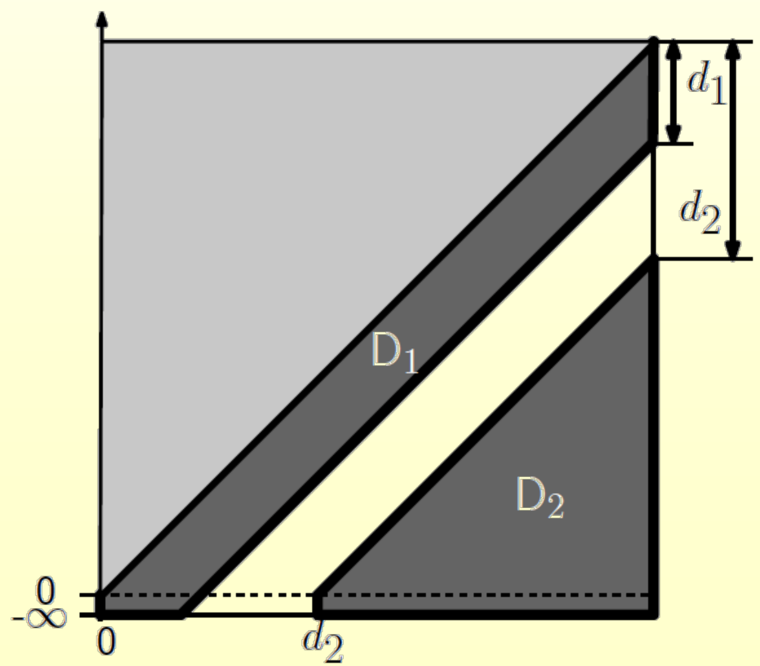
   if $|f(y) - f(x)| \leq \alpha$

   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

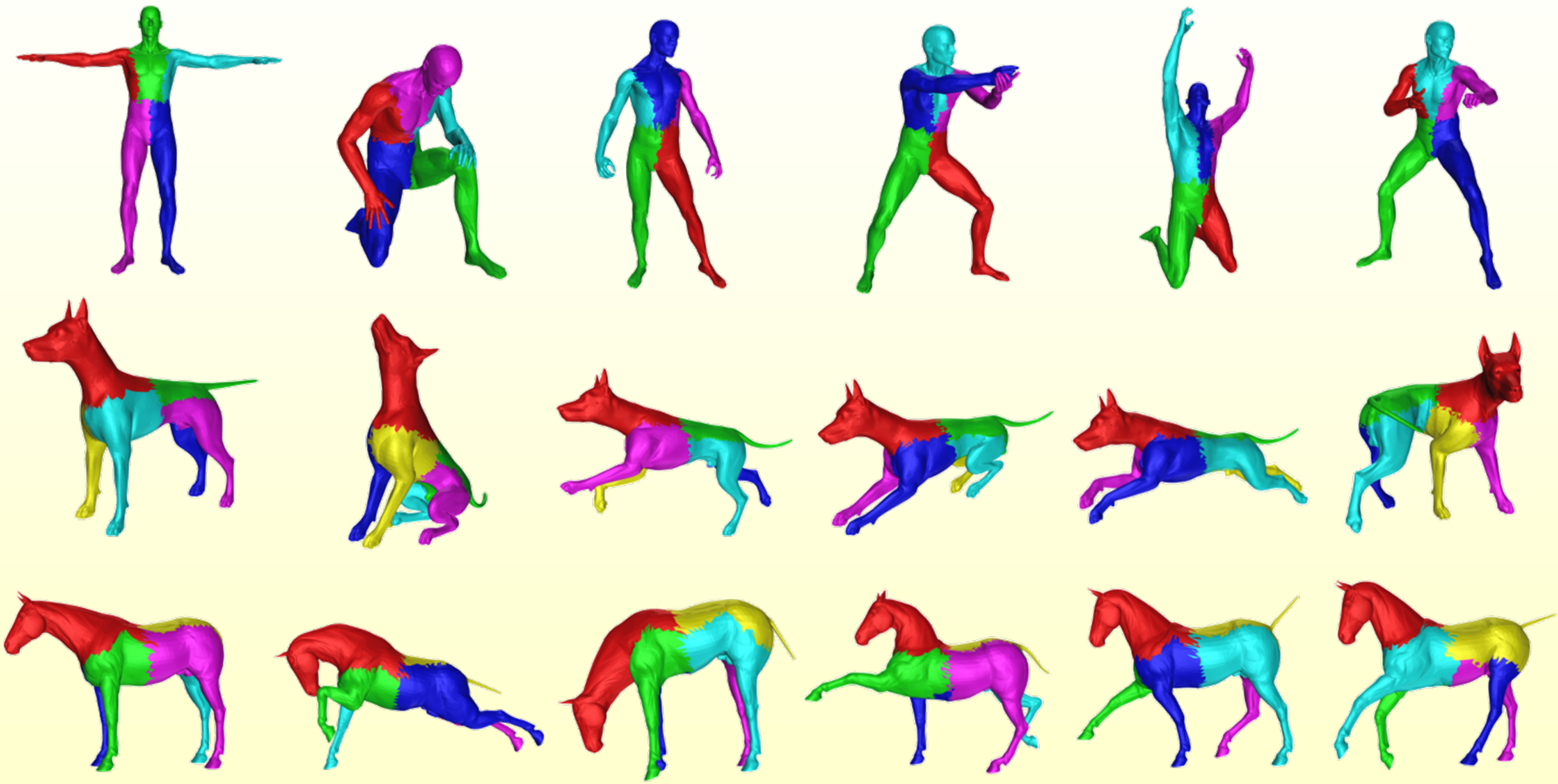


The PD itself can help us decide what the merging threshold should be

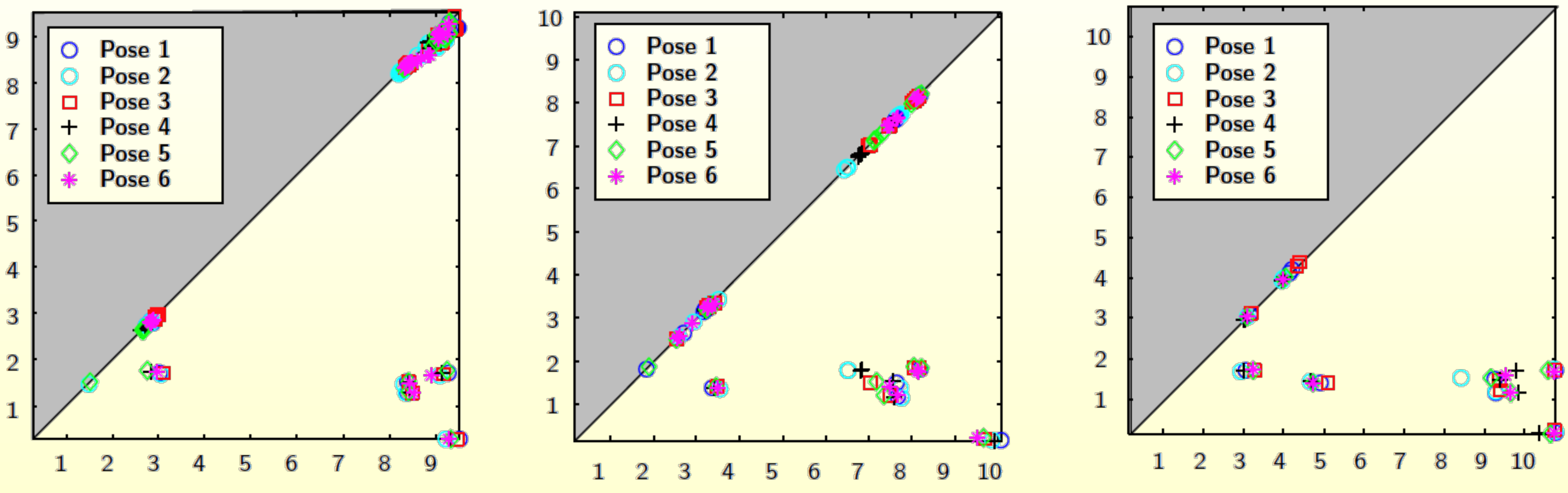
Choice of Filter Function is Crucial

- 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 under Isometric Deformations



Human Dog Horse

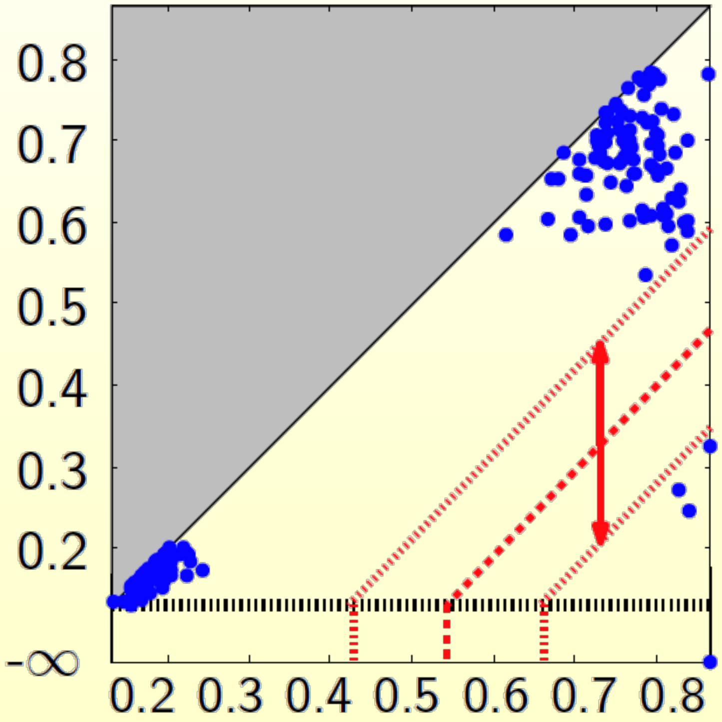
Caveats

- 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

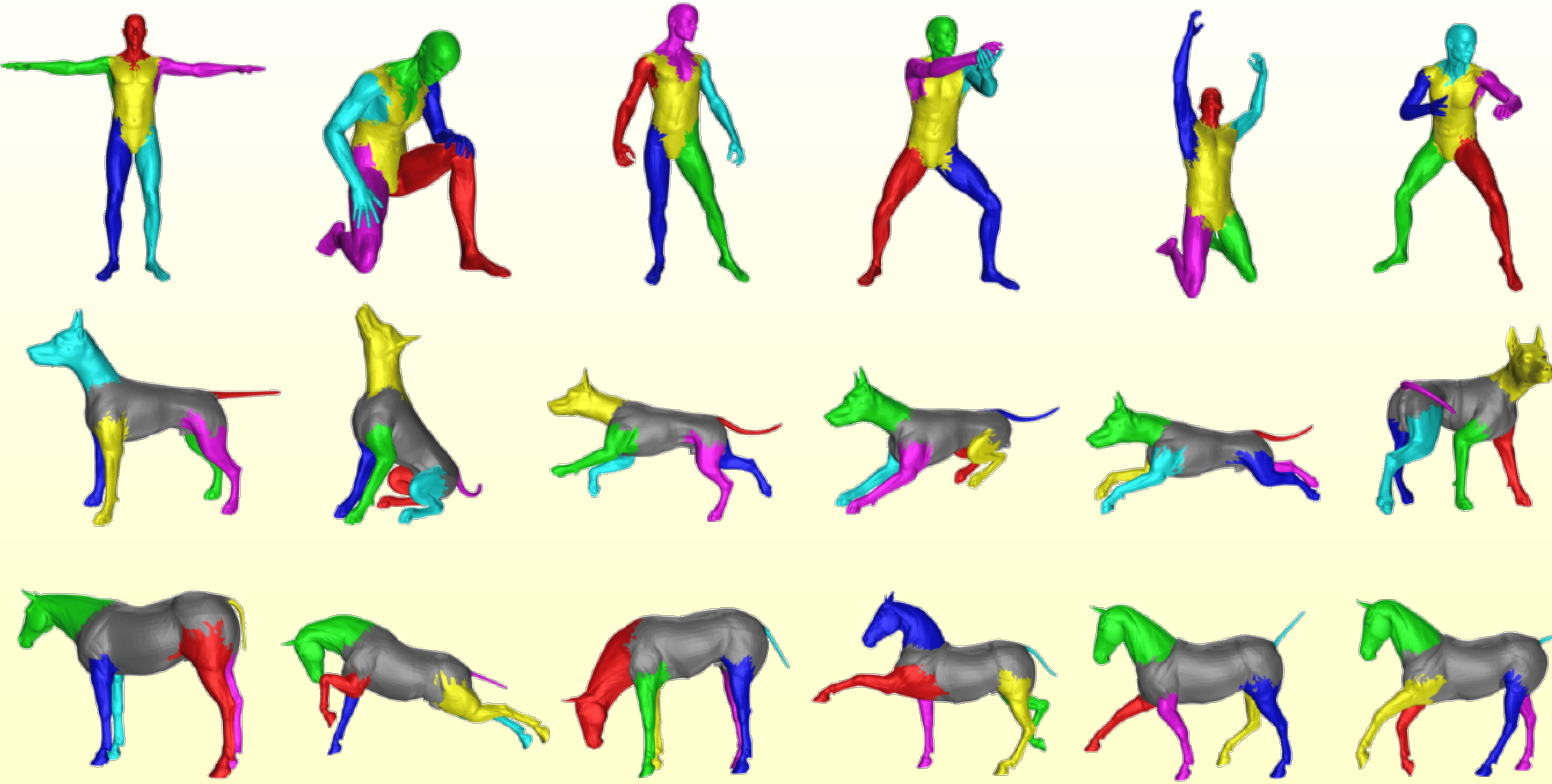


Extended Algorithm

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

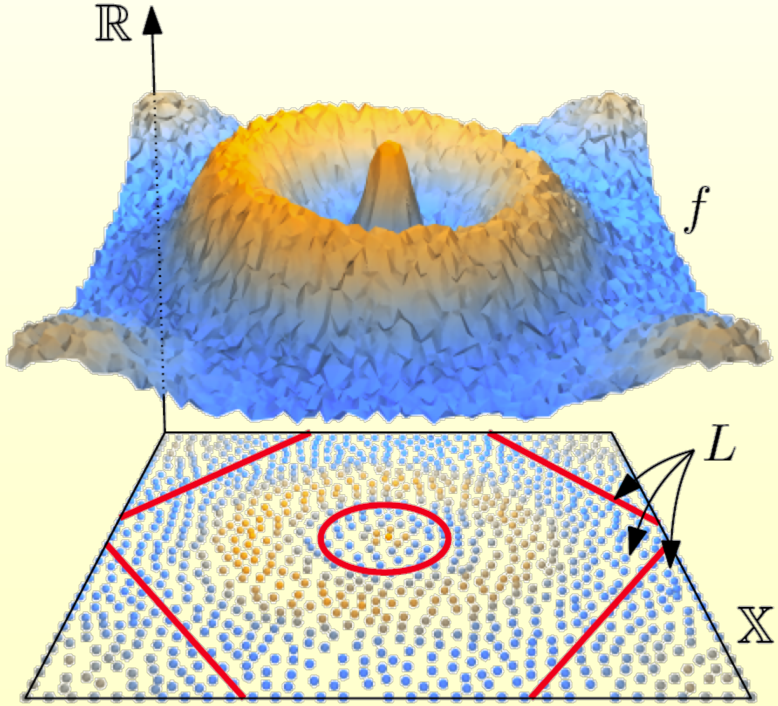
Setting: topological space $\mathbb{X}$, $f : \mathbb{X} \to \mathbb{R}$

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):

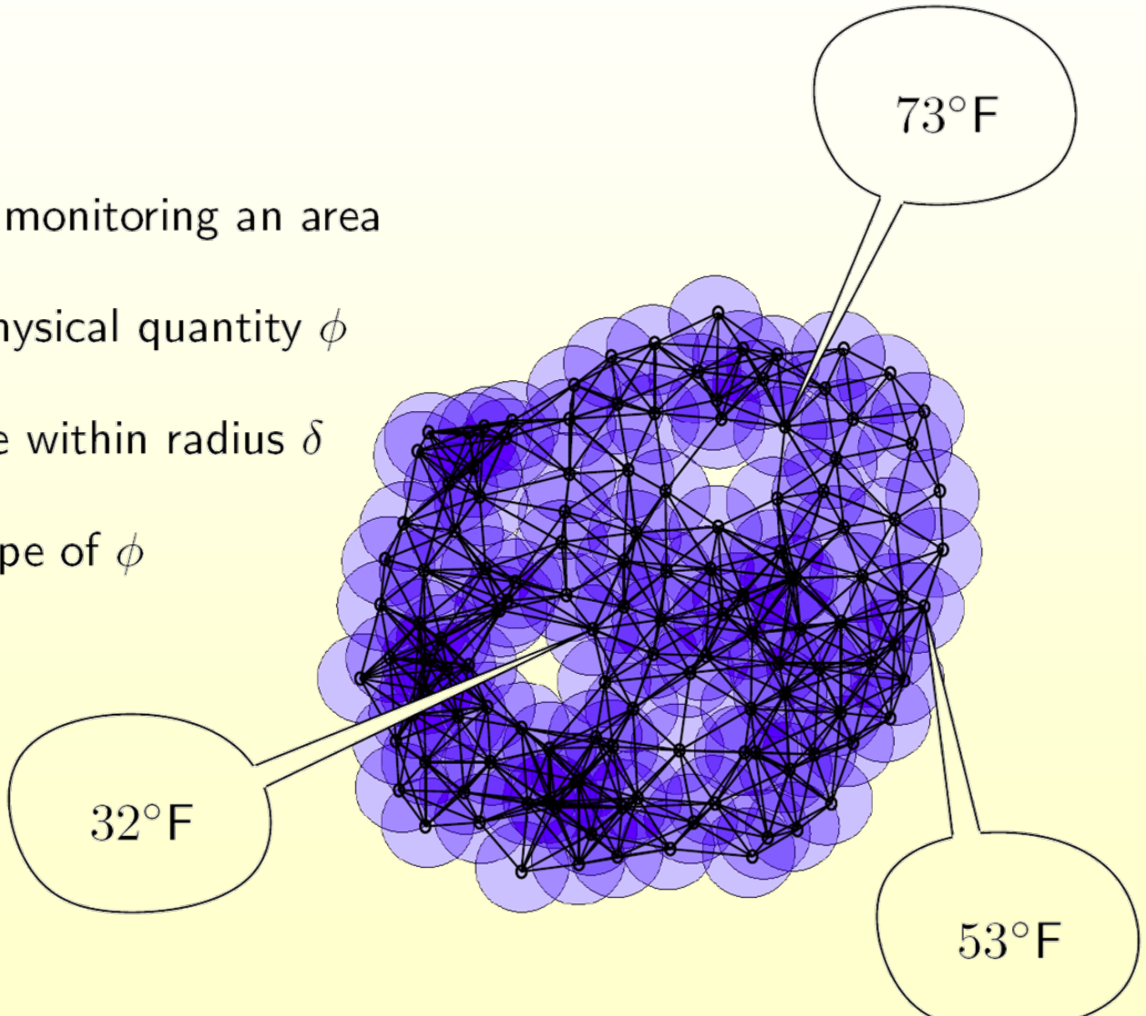
- prominent peaks/valleys
- basins of attraction
- in the presence of noise
- without explicit knowledge of the sample positions



Motivating Applications

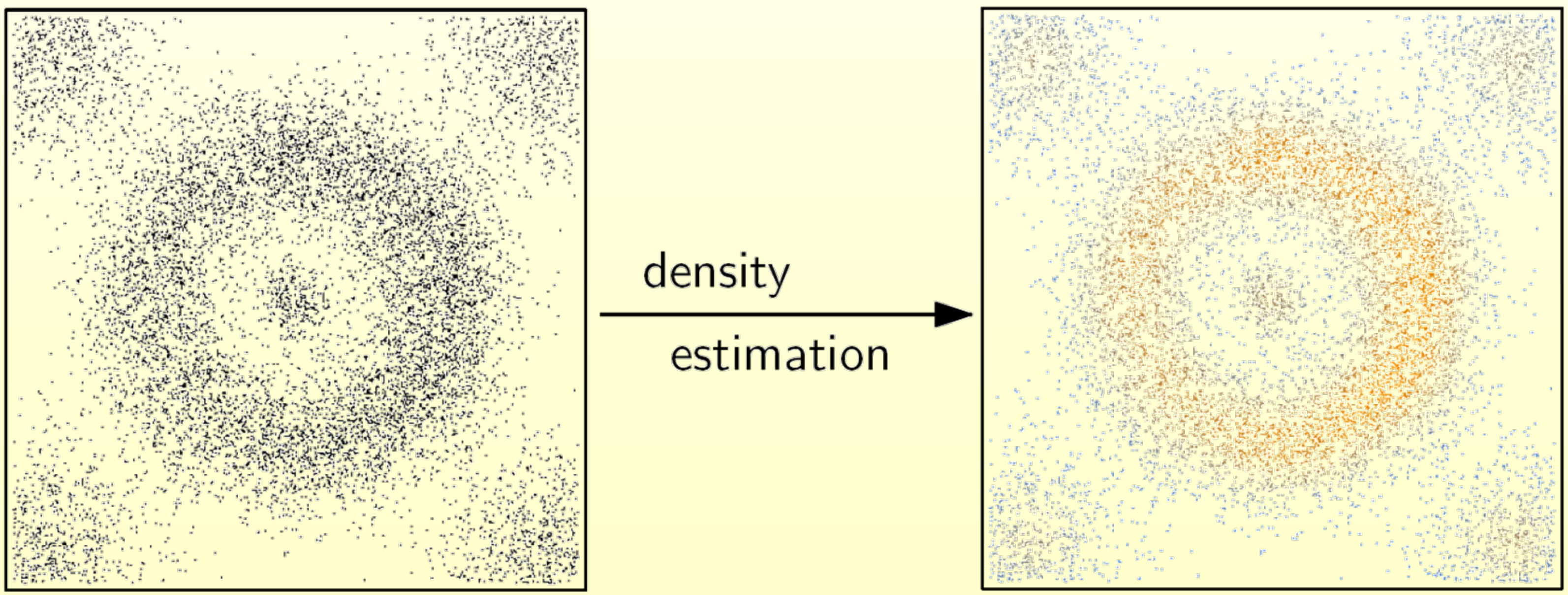
- sensor networks:
  - collection of sensors monitoring an area
  - sensors measure a physical quantity ϕ
  - sensors communicate within radius δ

Goal: analyze landscape of ϕ



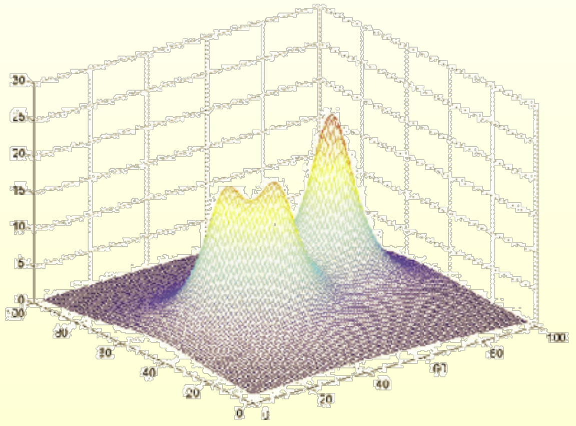
Motivating Applications

- 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

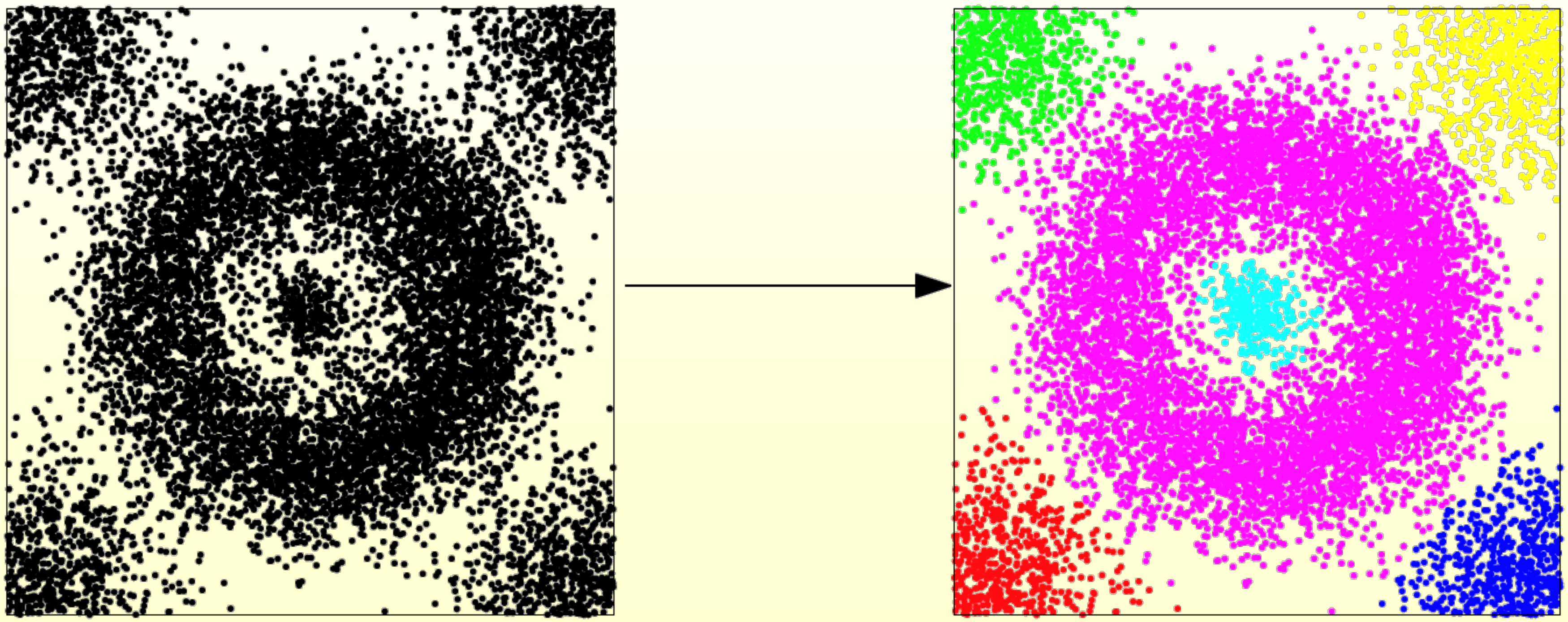


- Persistence-based: using a triangulation of $\mathbb{X}$ based on L, obtained from a parametrization or other means

Cluster Analysis

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

- distance / (dis-)similarity matrix

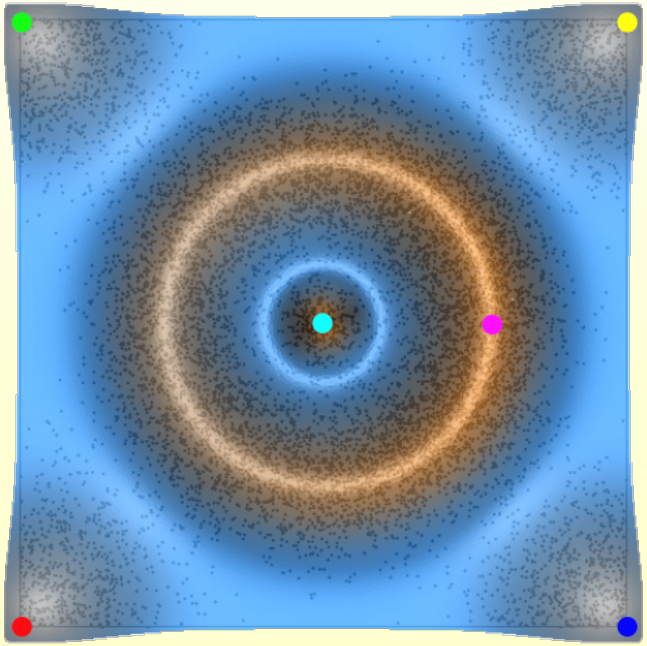


Task:

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

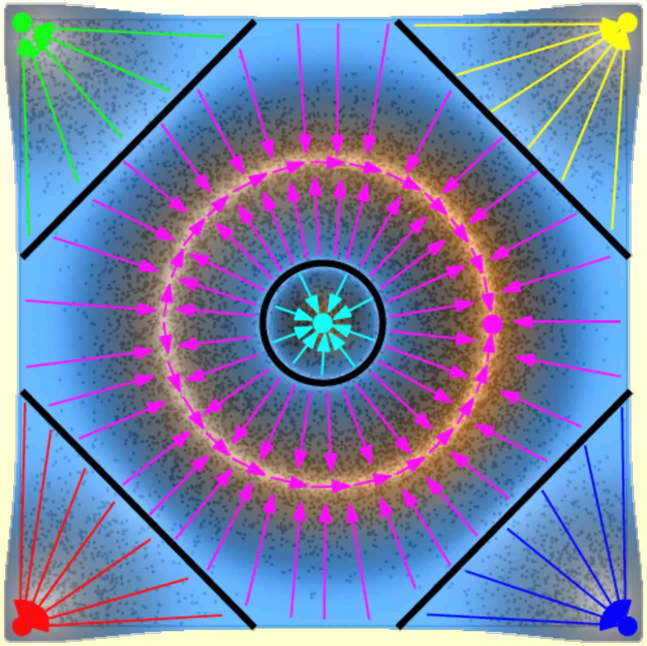
Mode-Seeking Paradigm

- 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



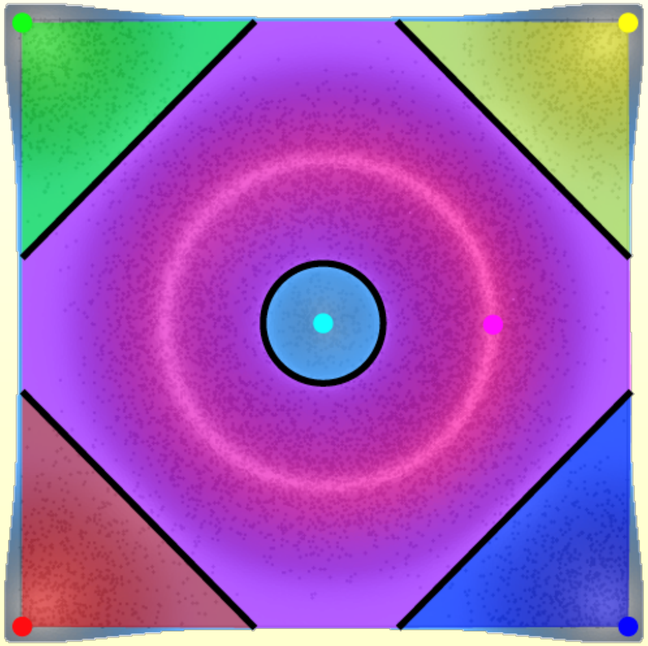
Mode-Seeking Paradigm

- 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

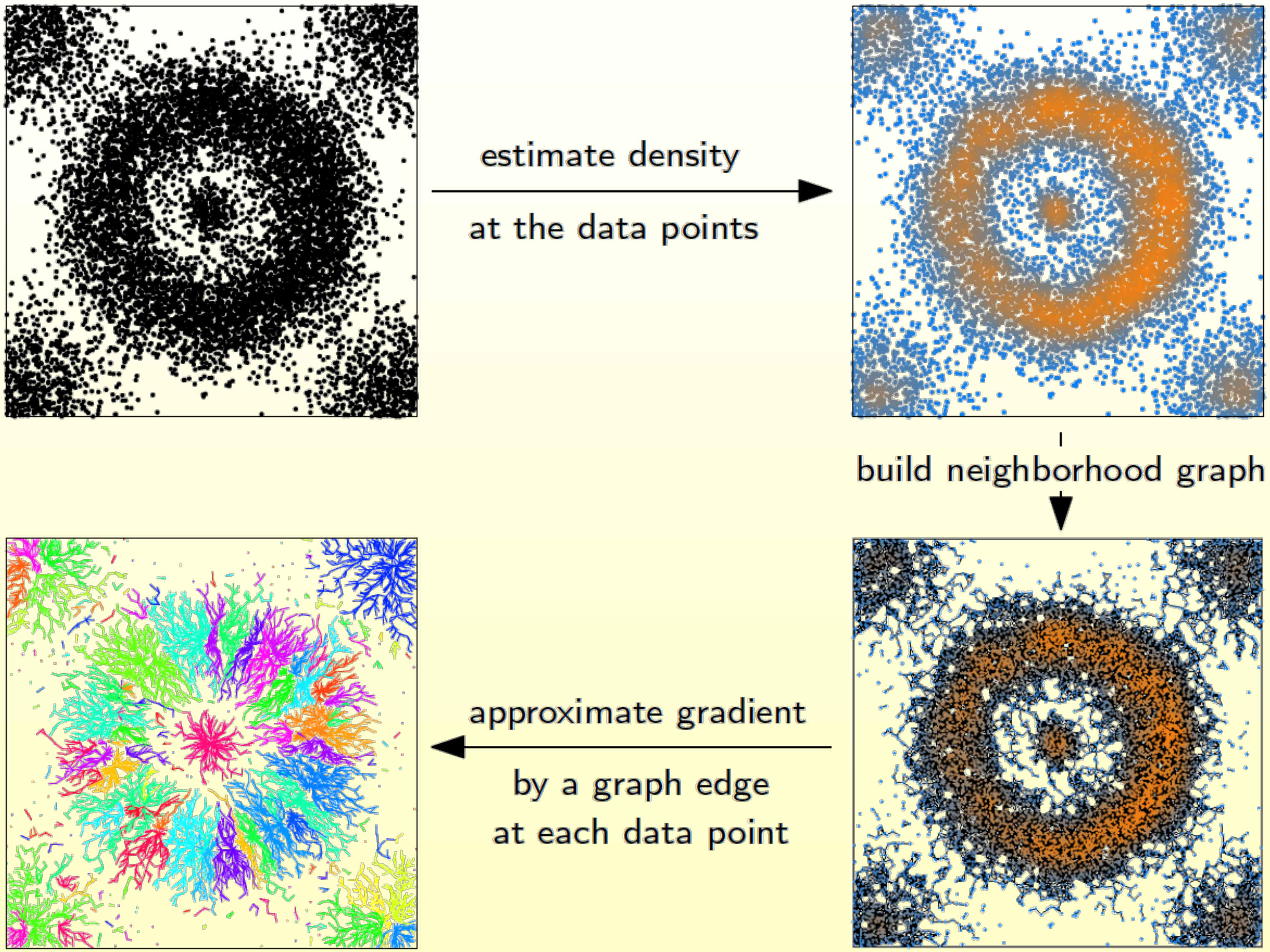


Mode-Seeking Paradigm

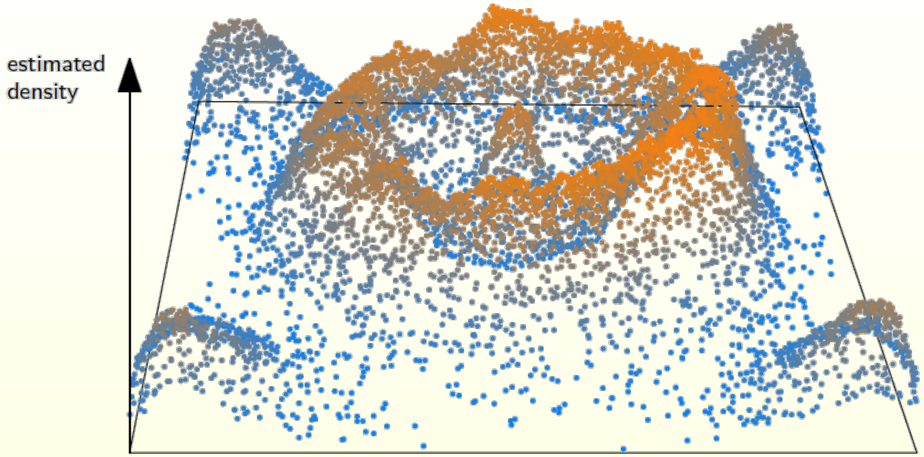
- Assume the data points are sampled from some unknown probability distribution
- Partition the data according to the basins of attraction of the peaks of the density

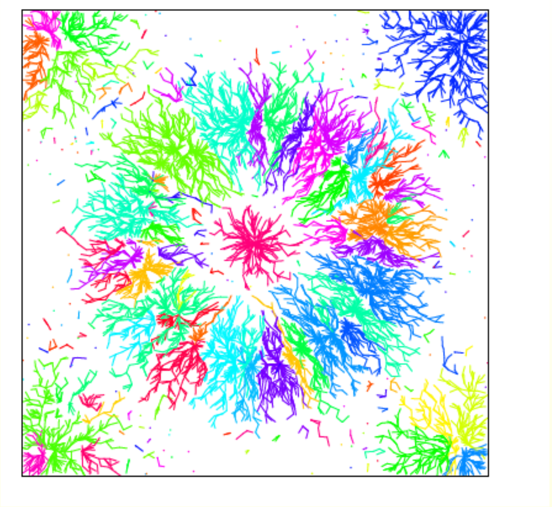


Mean-Shift and Variations

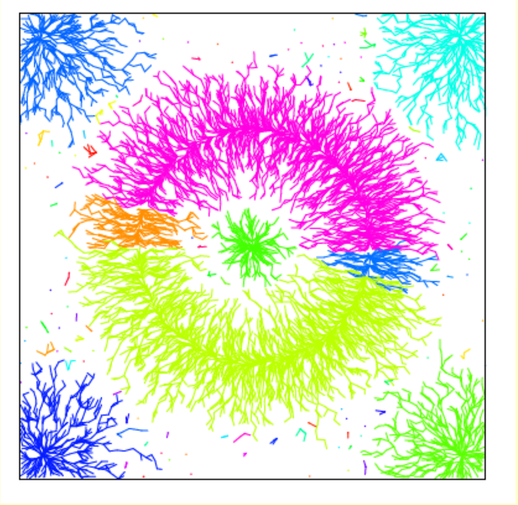


Things Can Go Wrong





Noisy density estimator



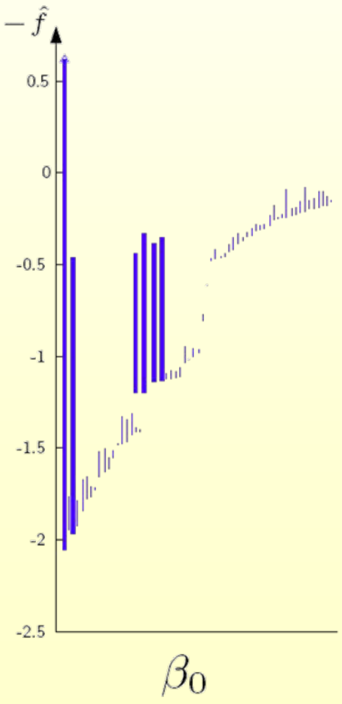
Bad proximity graph

Persistence-Based Approach

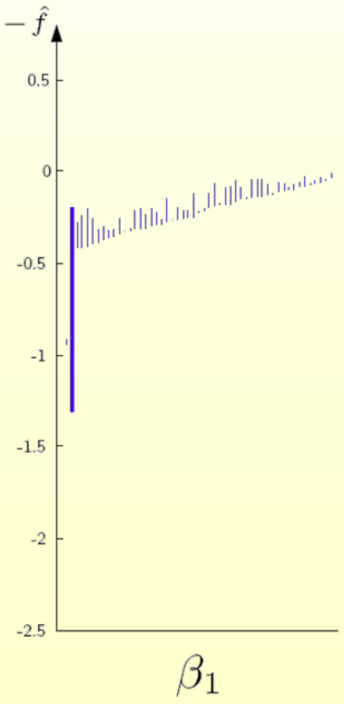
Assumptions: $\mathbb{X}$ triangulated space, $f : \mathbb{X} \to \mathbb{R}$ Lipschitz continuous

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

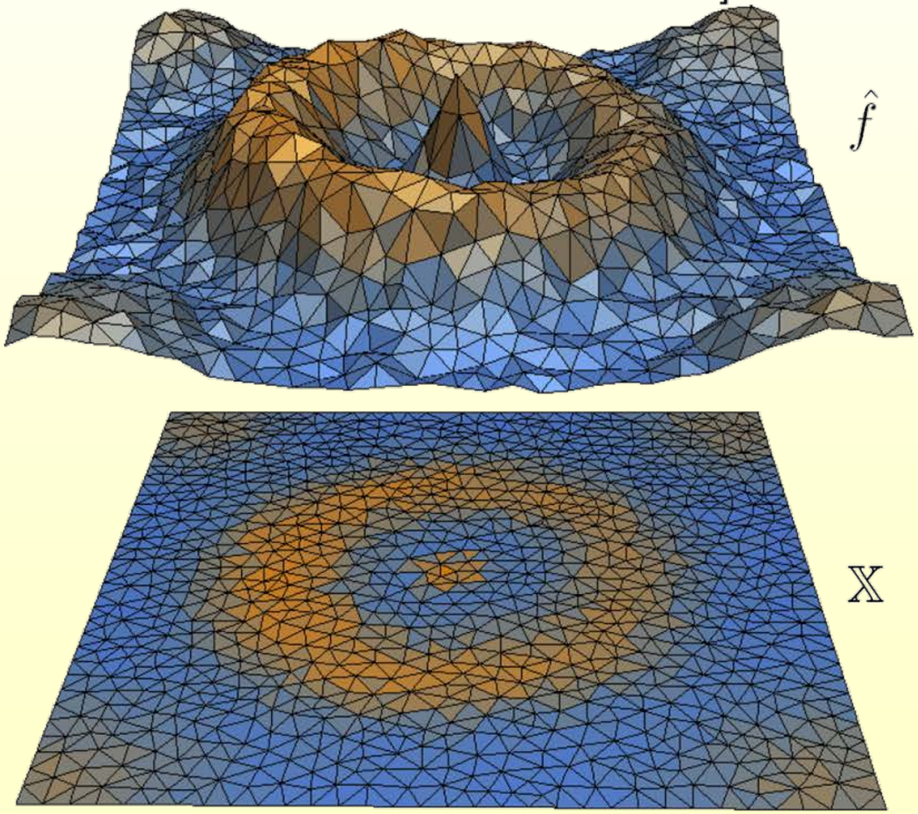
$\longrightarrow$ apply persistence algo. to $\pm\hat{f}$ [Edelsbrunner, Letscher, Zomorodian '00]



(6 prominent peaks)

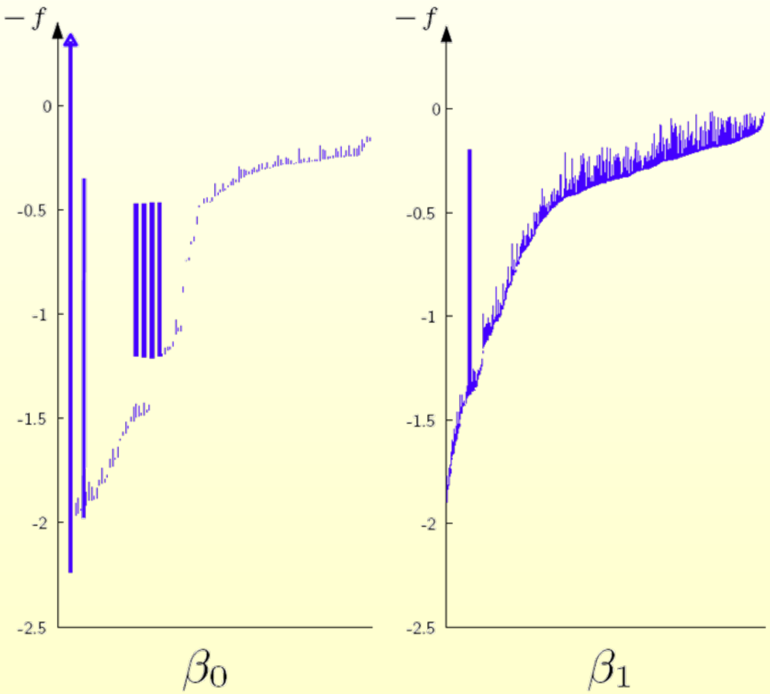


(ring-shaped basin of attraction)

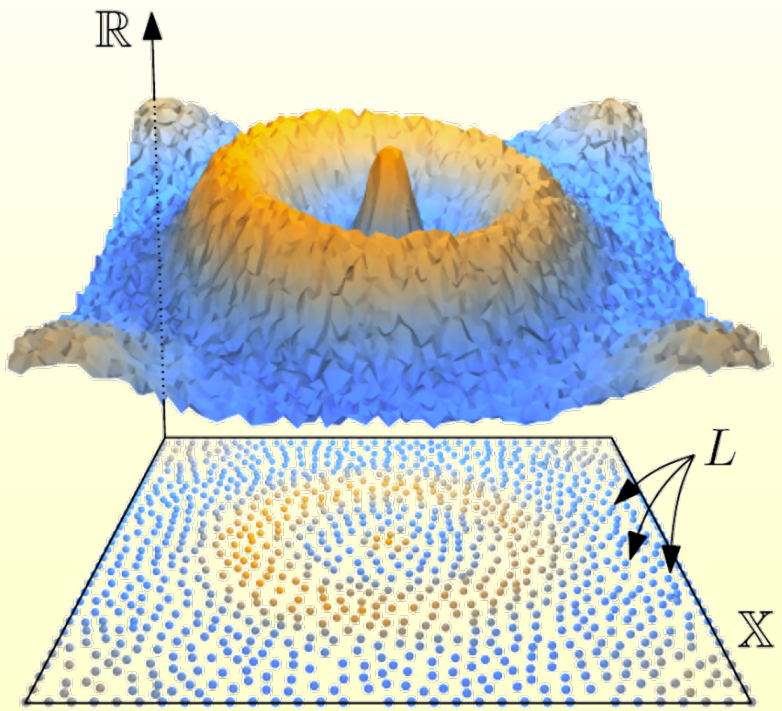


Clustering Example A

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



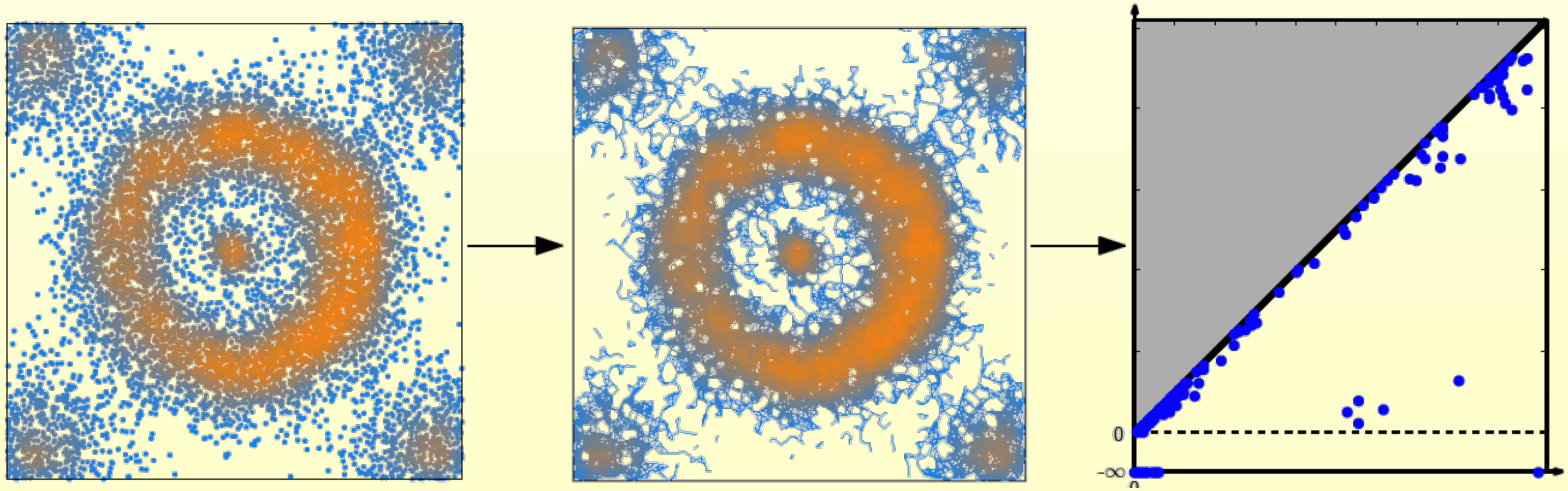
(6 prominent peaks) (ring-shaped basin of attraction)



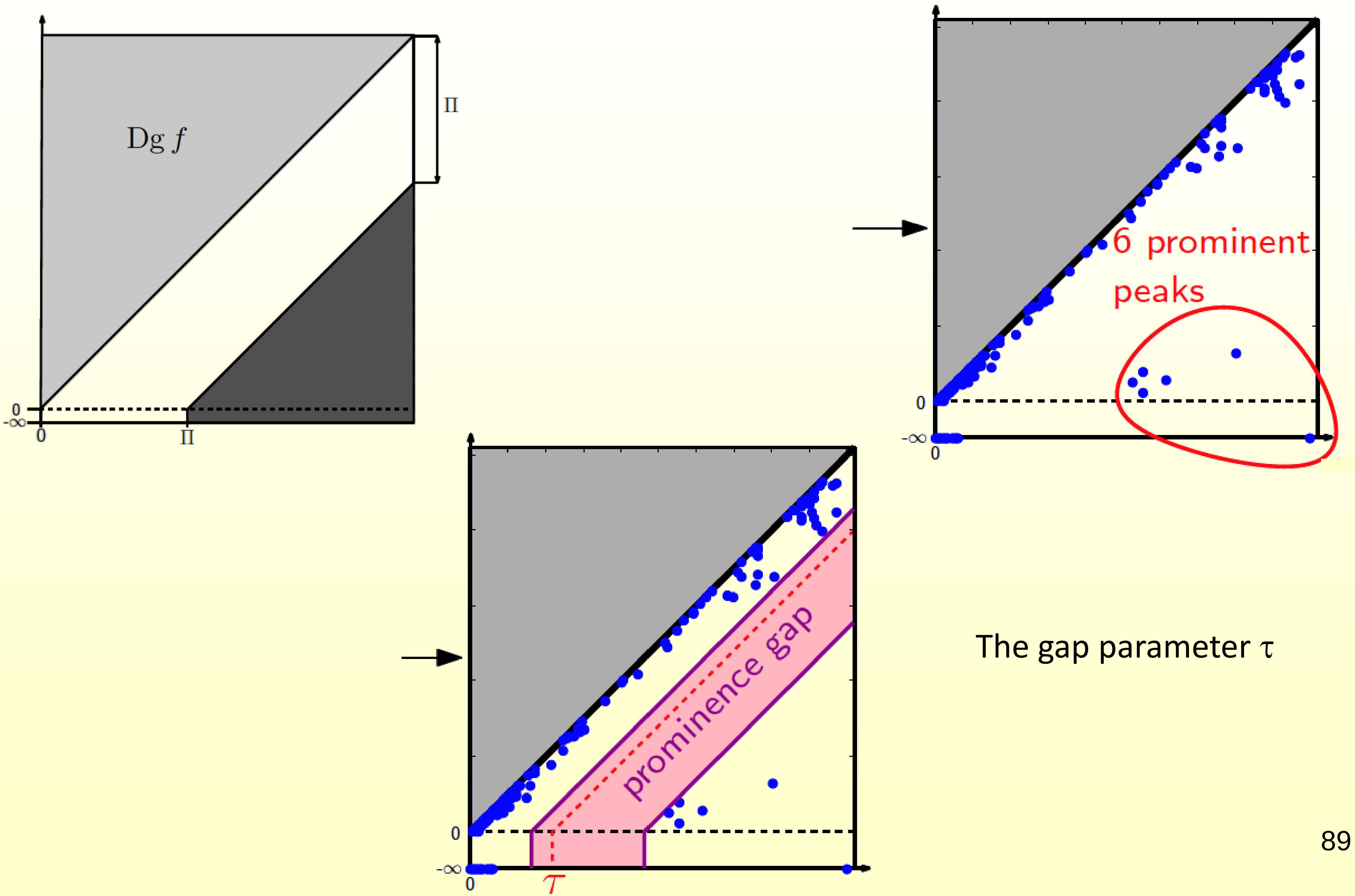
The Persistence Approach: ToMATo

Topological
Mode
Analysis
Tool

- 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 → *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 → union-find data structure)



Estimating the Prominent Clusters

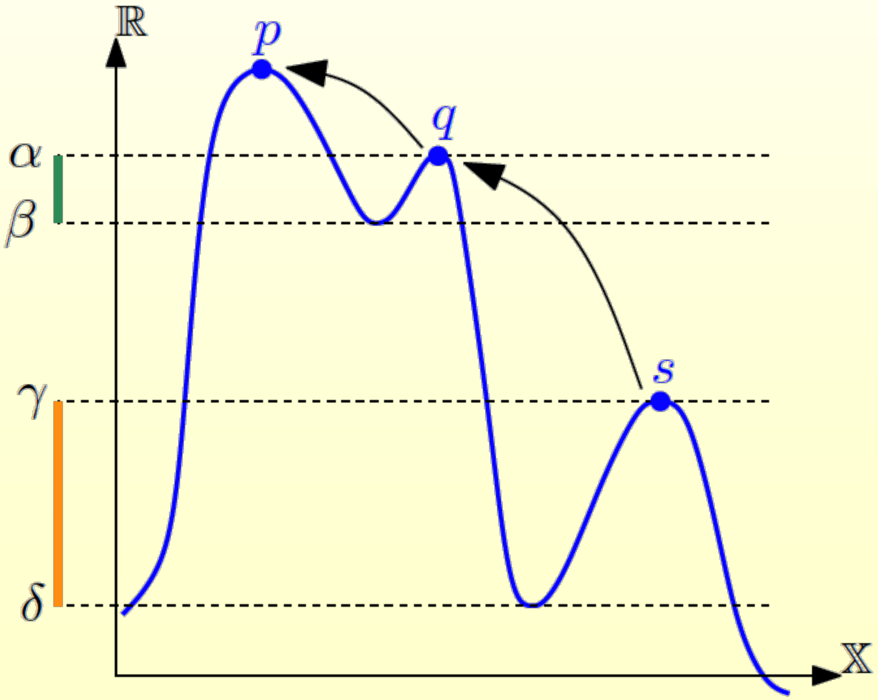


The gap parameter τ

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

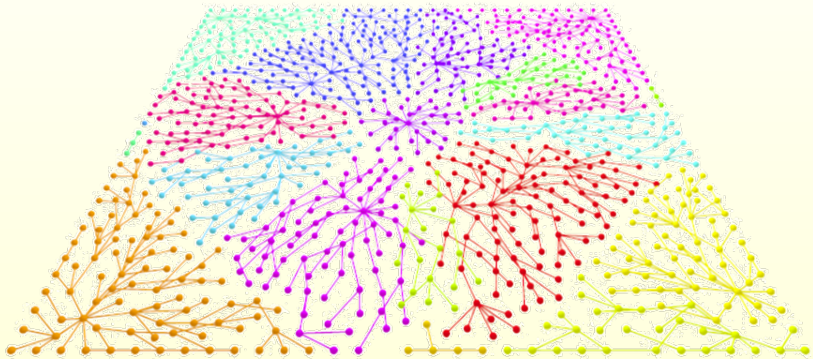
Goal: approximate basins of attraction of significant peaks of f

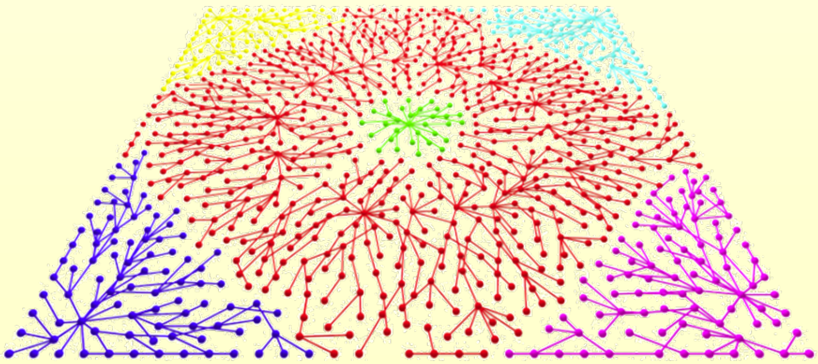
$\Rightarrow$ segmentation/clustering of point cloud L

Approach:

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

$\rightarrow$ union-find data structure

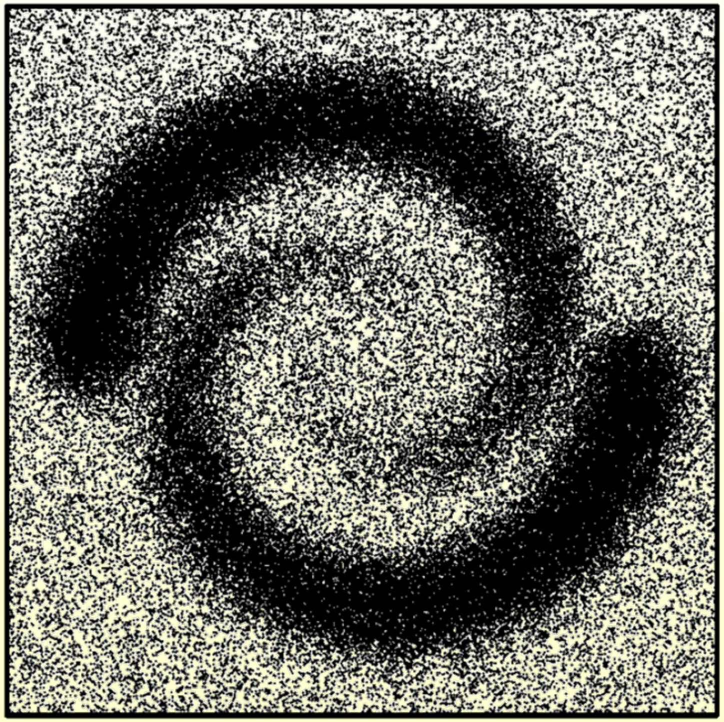


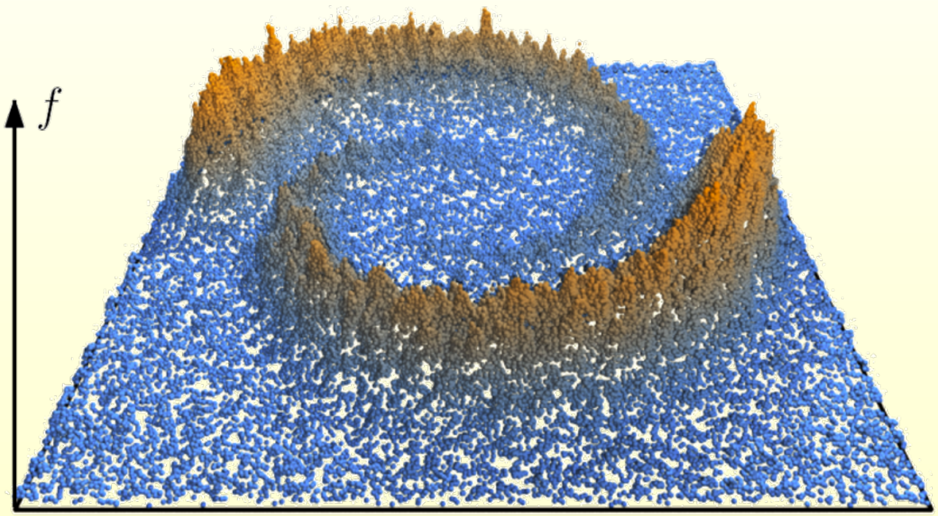


Clustering B – The Rips Parameter δ

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

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



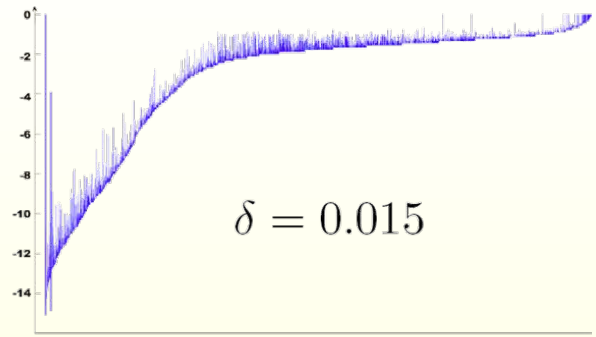


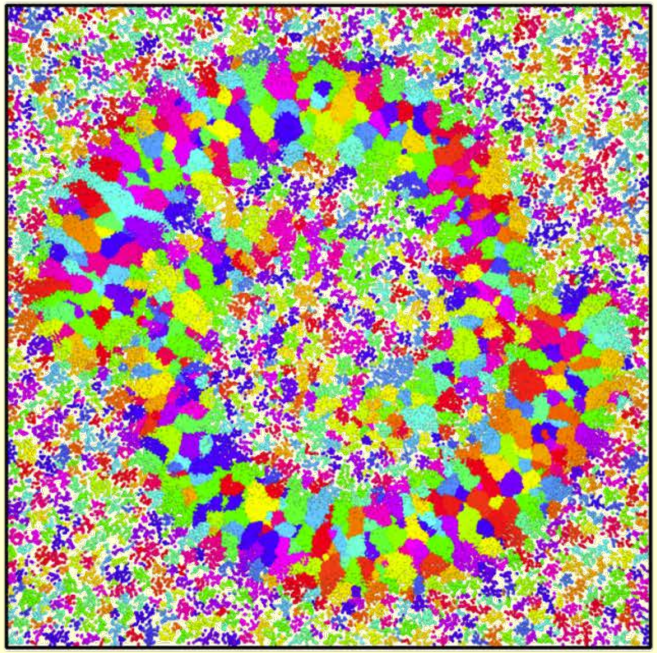
Clustering B

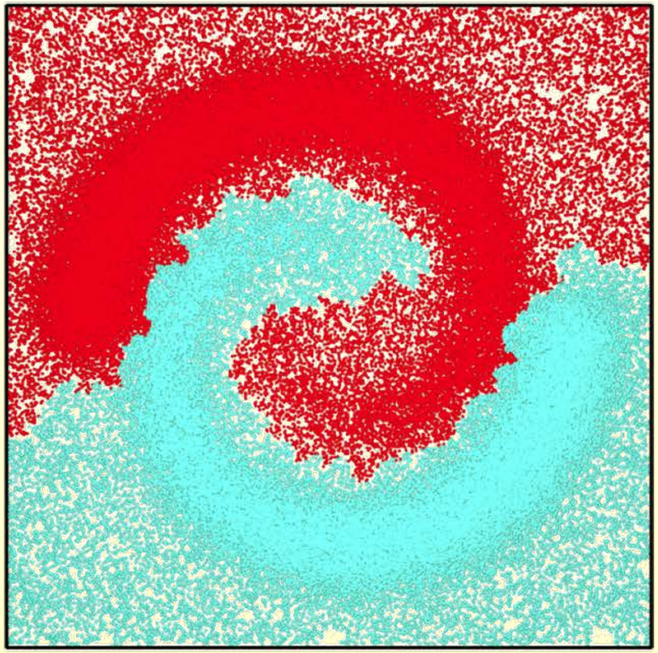
Clustering B

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

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





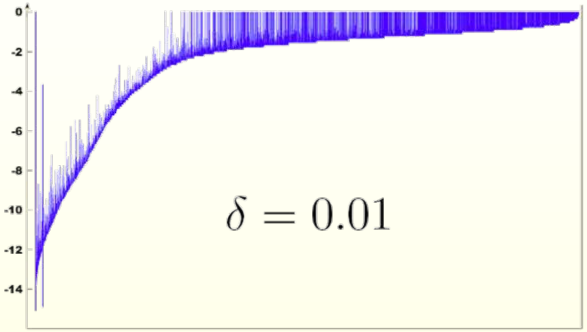


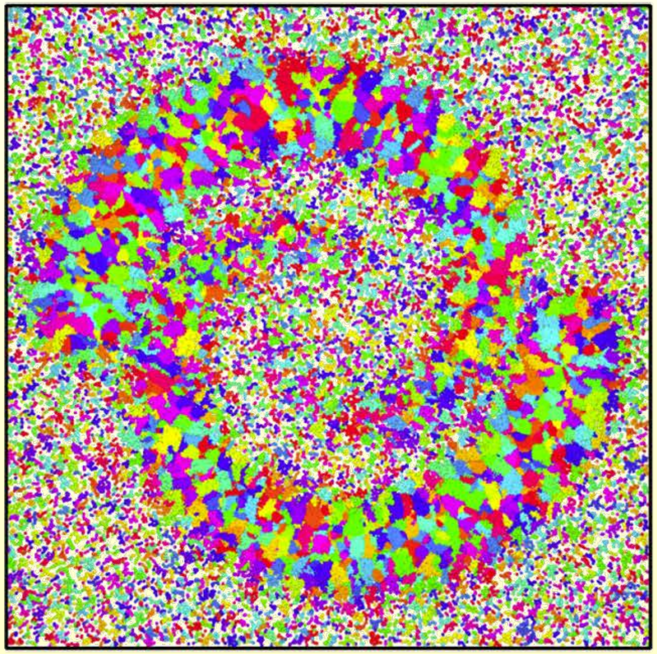
Clustering B

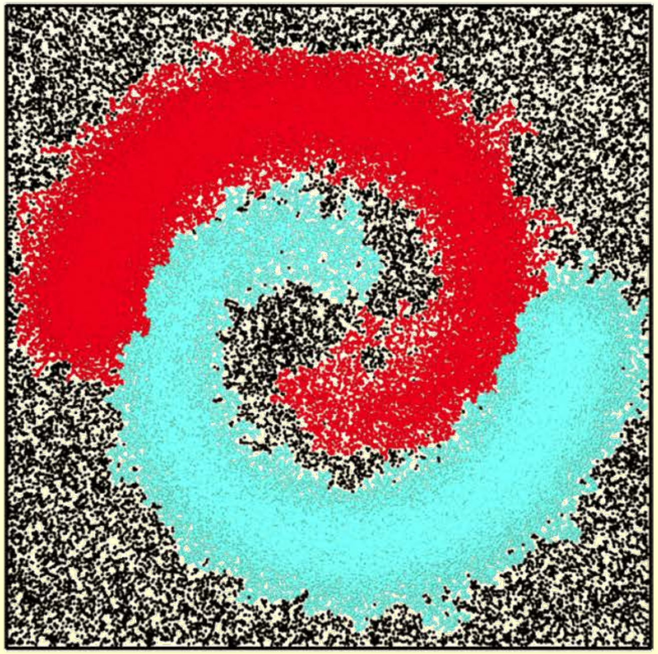
Clustering B

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

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

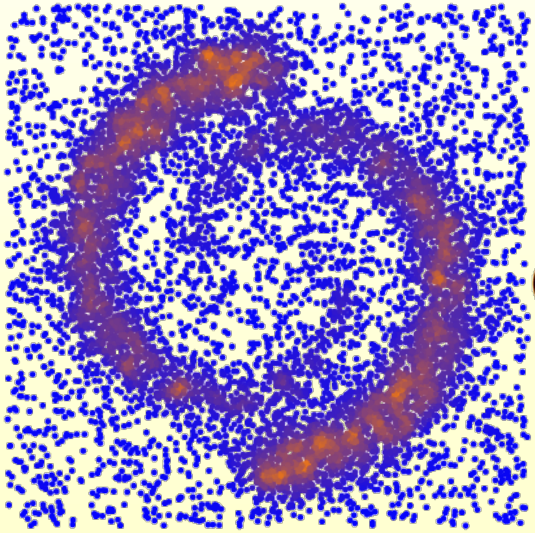


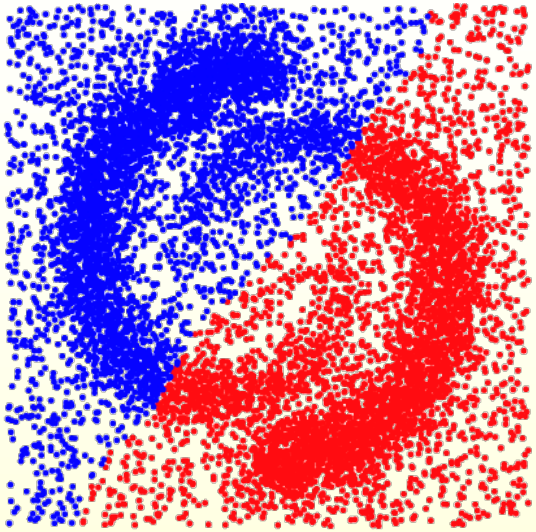




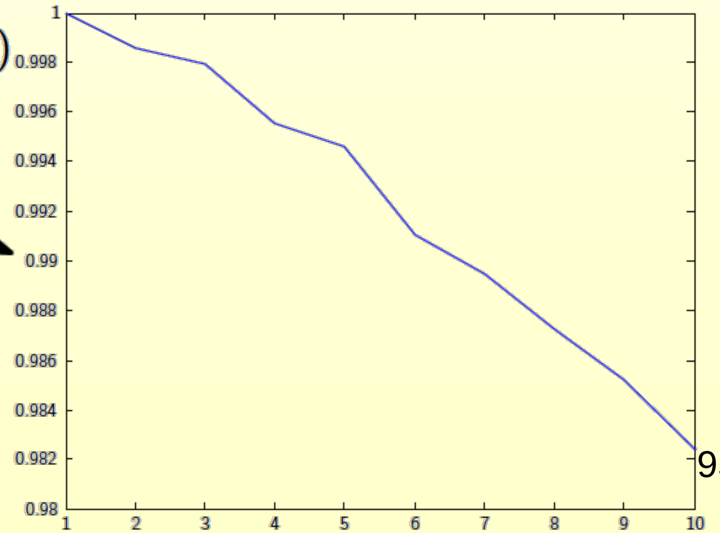
FYI, Spectral Clustering

Synthetic Data





Spectral clustering
(k-means in eigenspace)



Another Hard Example

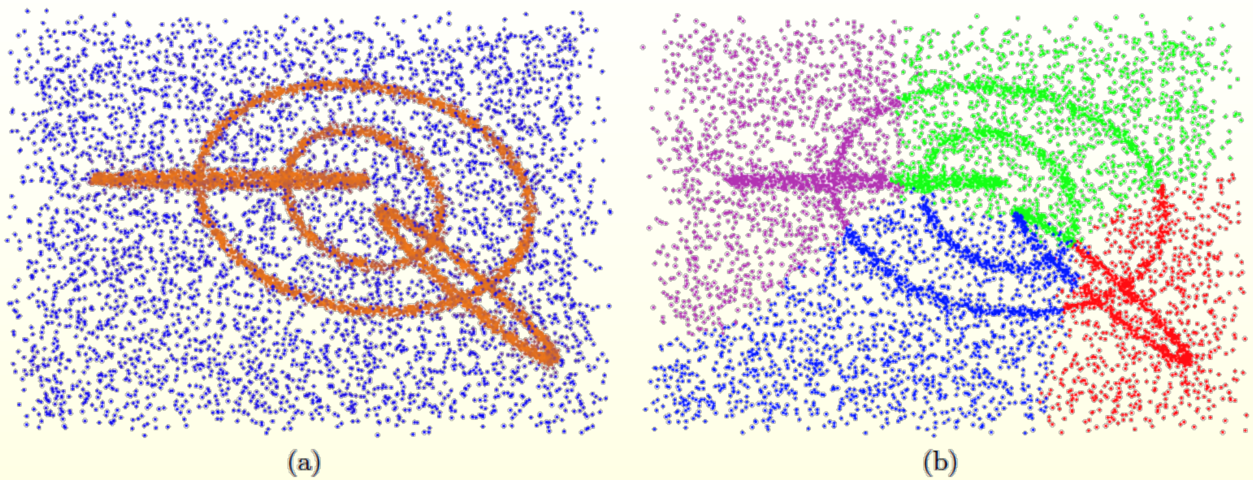


Figure 7: (a) The rings data set with the estimated density function. (b) The result obtained using spectral clustering.

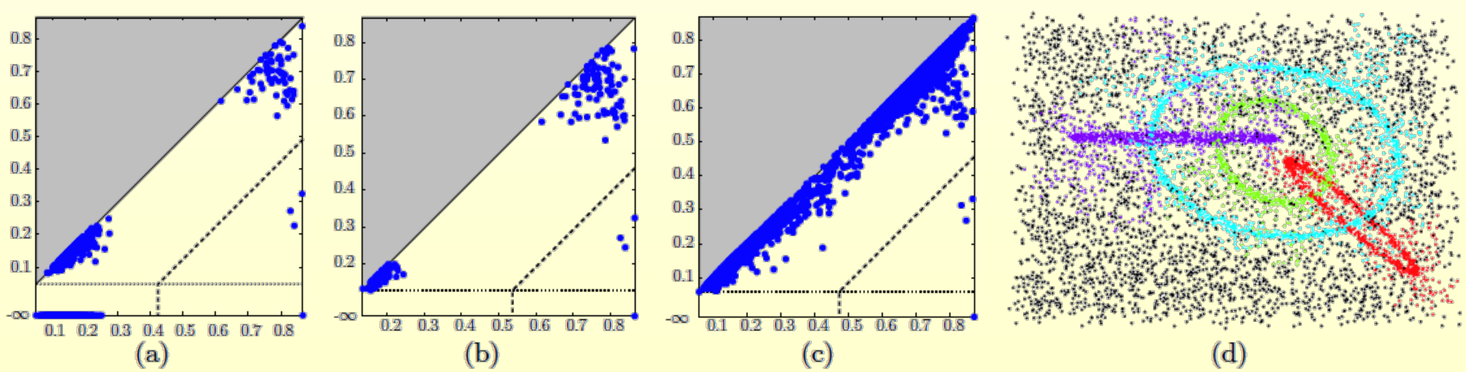


Figure 8: Outputs of ToMATo on the rings data set: the obtained PD with (a) δ-Rips graph, (b) k-nn graph, and (c) Delaunay graph. (d) Clustering obtained with the δ-Rips graph.

The End