DSC 214
Topological Data Analysis

Topic 9: Mapper

Instructor: Zhengchao Wan

- Persistent homology
  - One of the most important developments in computational topology in the last two decades
- Other topological structures for analyzing functions
  - Real valued functions, or more complex maps

Mapper

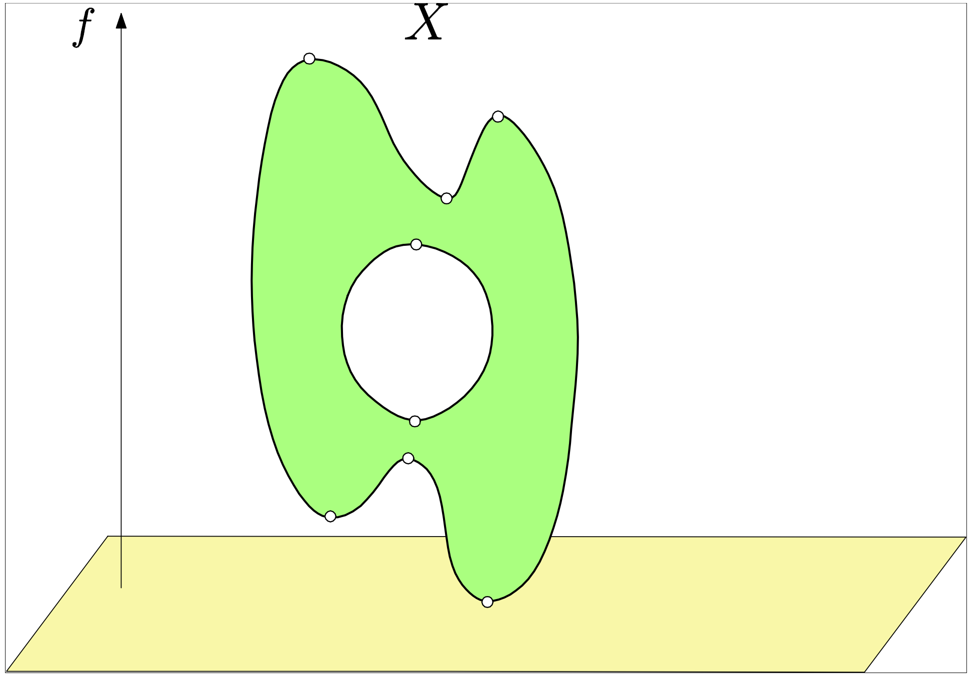
- [Singh, Mémoli, Carlsson, 2007]
- Dimension reduction through topological methods
- Data visualization

- Summarizing topological structure of a map $f : X \to Z$ into a graph

Section 0:
Reeb graph

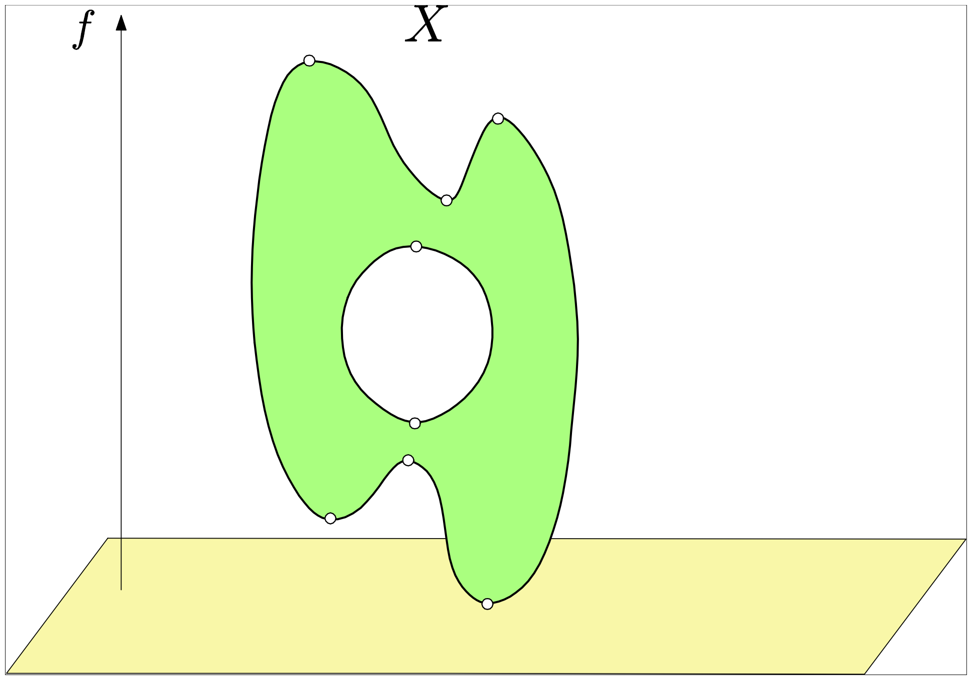
Introduction

- Given a topological space X and function $f: X \rightarrow R$



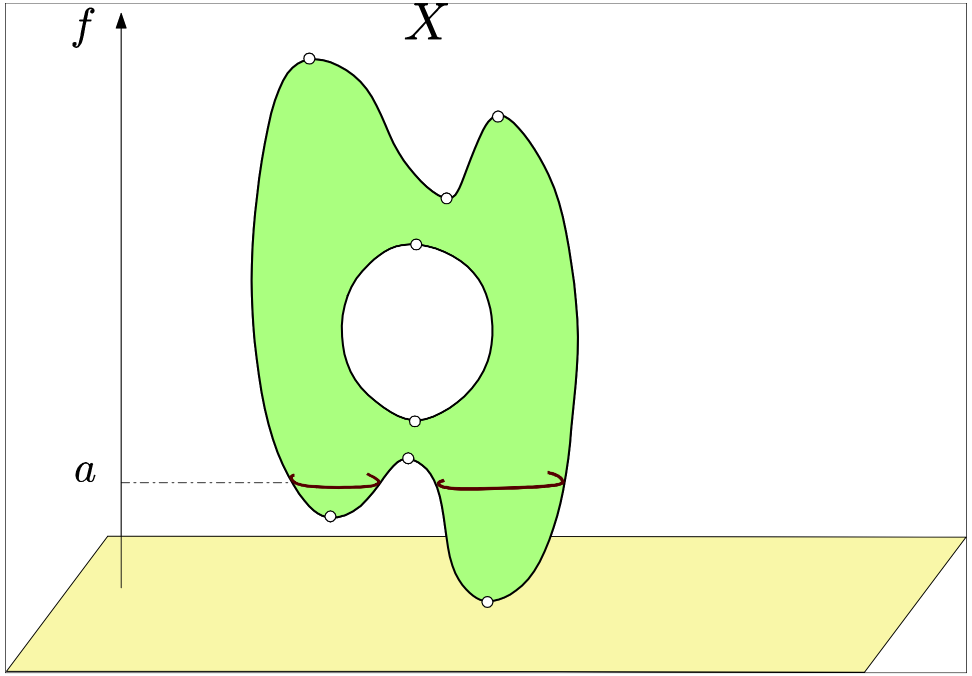
Introduction

- Given a topological space *X* and function $f: X \to R$
- *Level set* at value *a* :
  - $X_a := \{ x \in X \mid f(x) = a \}$



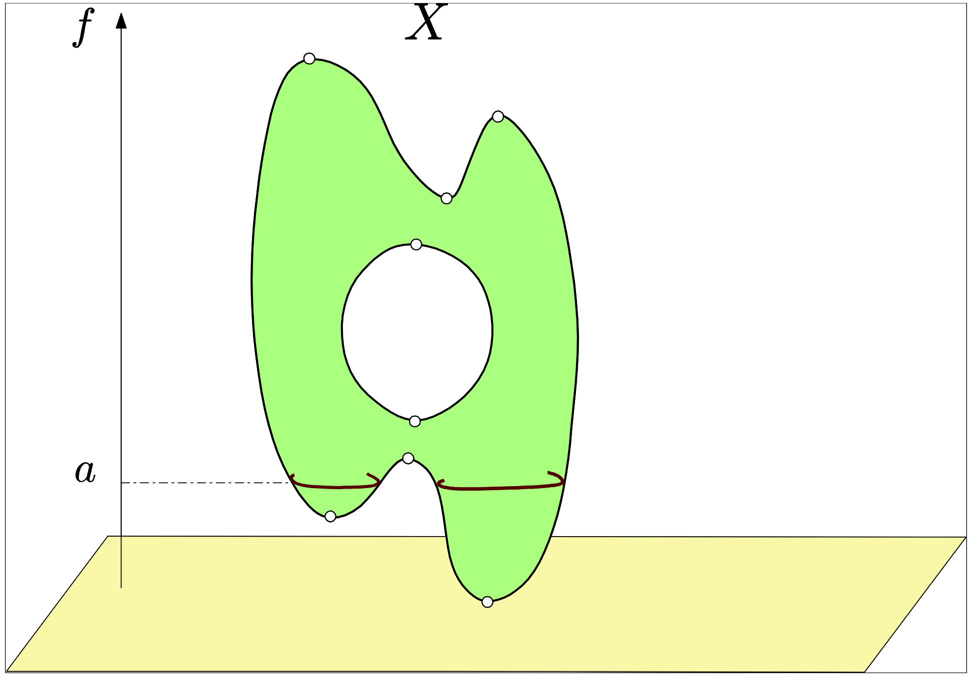
Introduction

- Given a topological space X and function $f: X \to R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a\}$



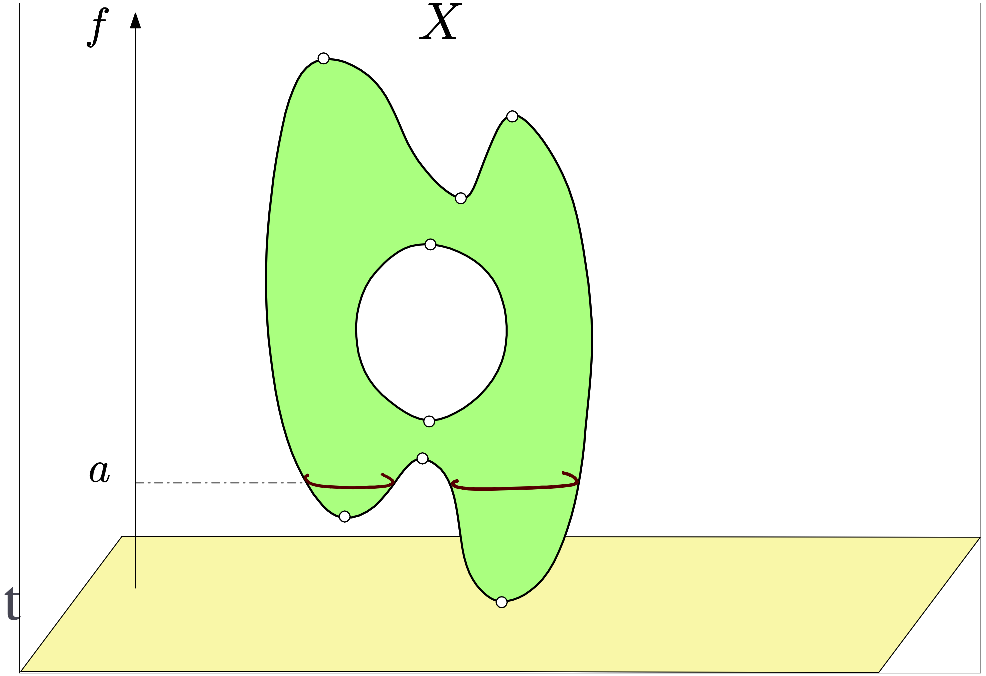
Introduction

- Given a topological space X and function $f: X \rightarrow R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a\}$
- A *contour* at value a :
  - a connected component of X_a



Introduction

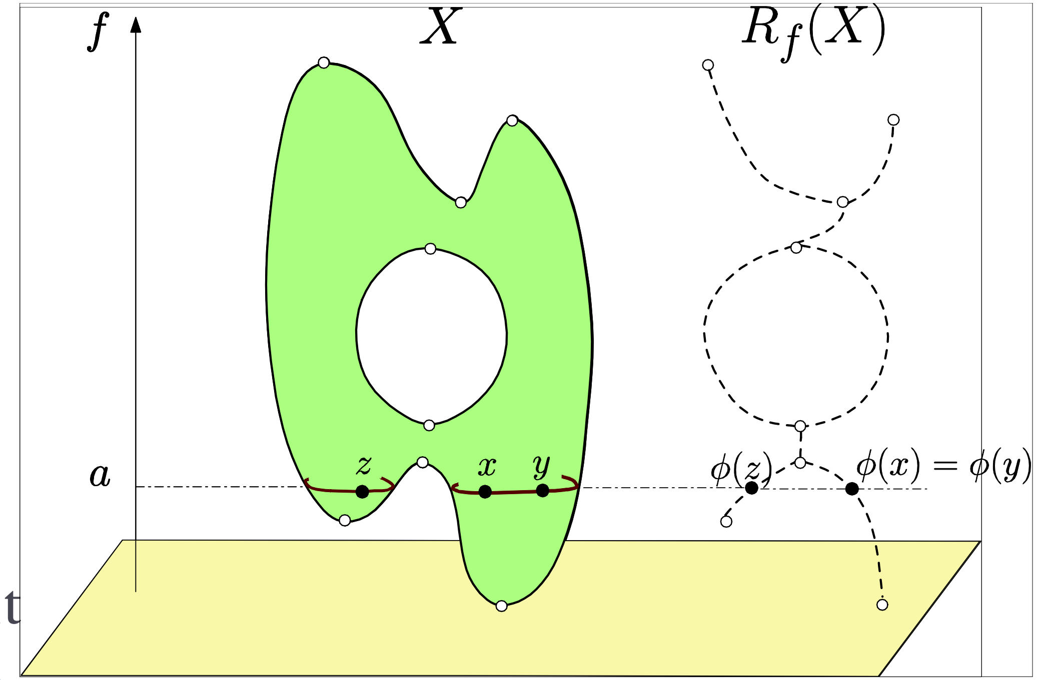
- Given a topological space X and function $f: X \to R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a



- *Reeb graph* $R_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point
  - A continuous surjection $\phi : X \to R_f(X)$ s.t, $\phi(x) = \phi(y)$ if and only if x and y is in the same contour

Introduction

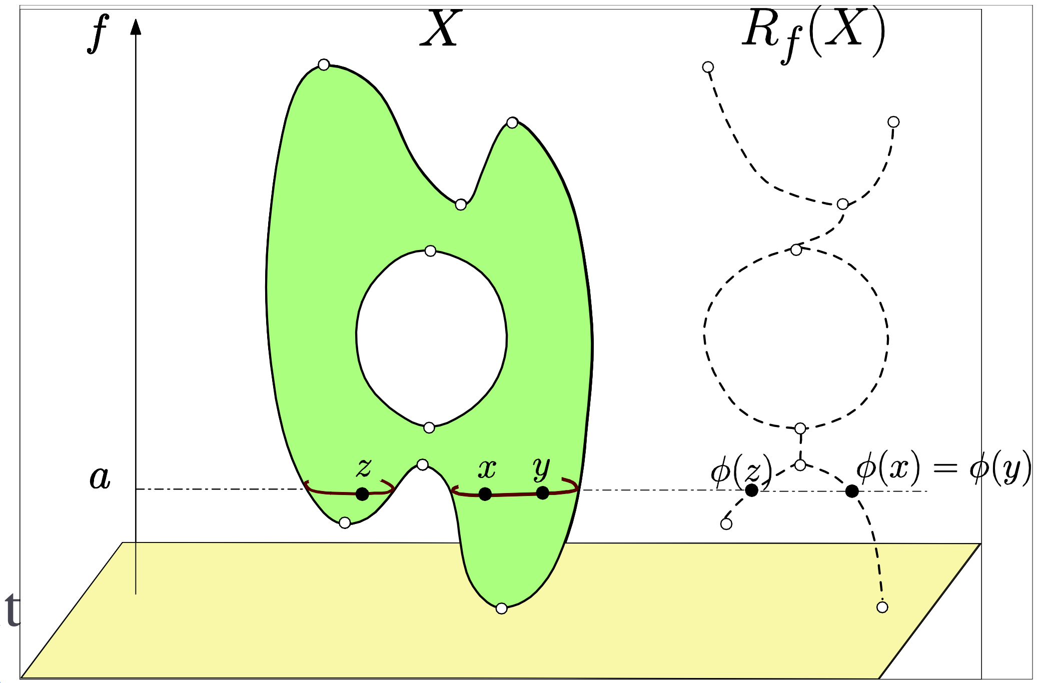
- Given a topological space X and function $f: X \to R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a



- *Reeb graph* $R_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point
  - A continuous surjection $\phi : X \to R_f(X)$ s.t, $\phi(x) = \phi(y)$ if and only if x and y is in the same contour

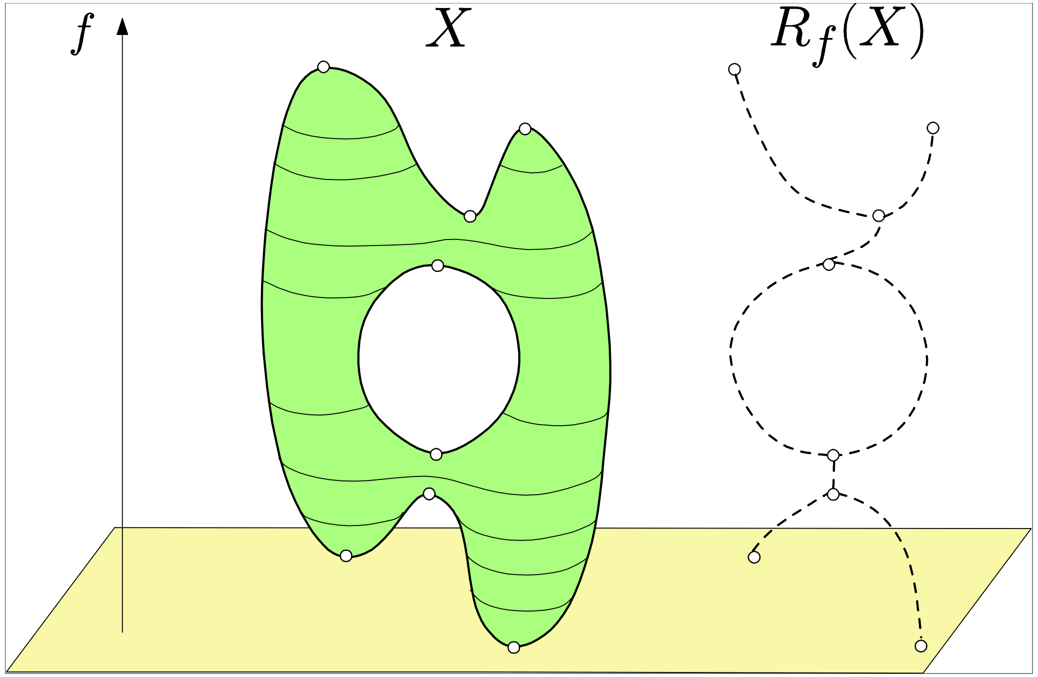
Introduction

- Given a topological space X and function $f: X \rightarrow R$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a\}$
- A *contour* at value a :
  - a connected component of X_a
- *Reeb graph* $R_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point
  - A continuous surjection $\phi : X \rightarrow R_f(X)$ s.t, $\phi(x) = \phi(y)$ if and only if x and y is in the same contour



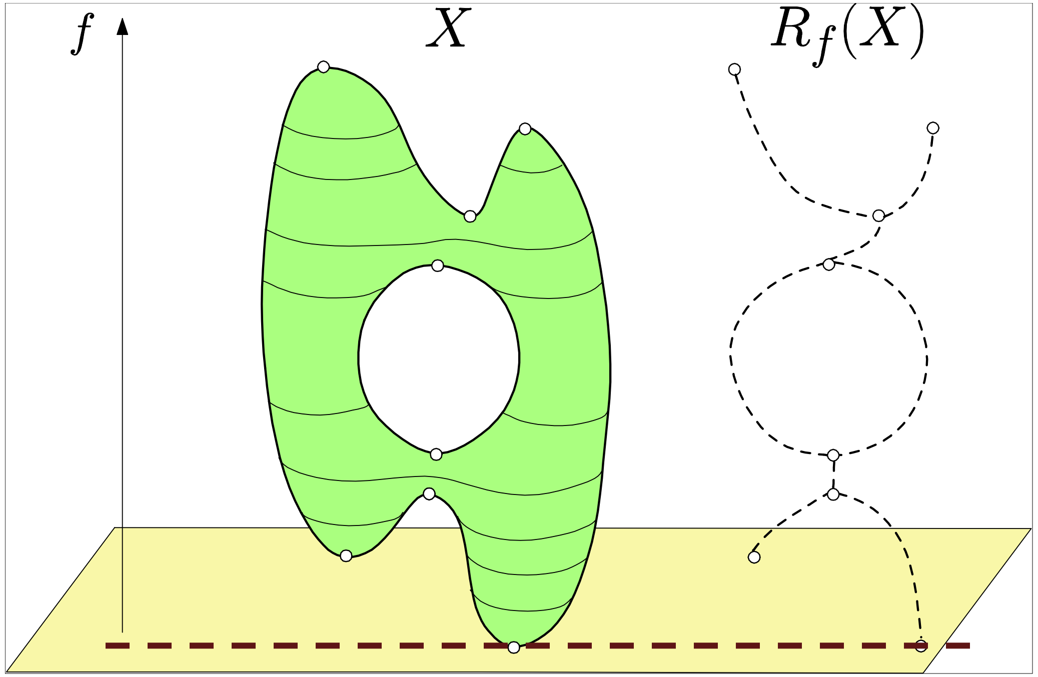
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



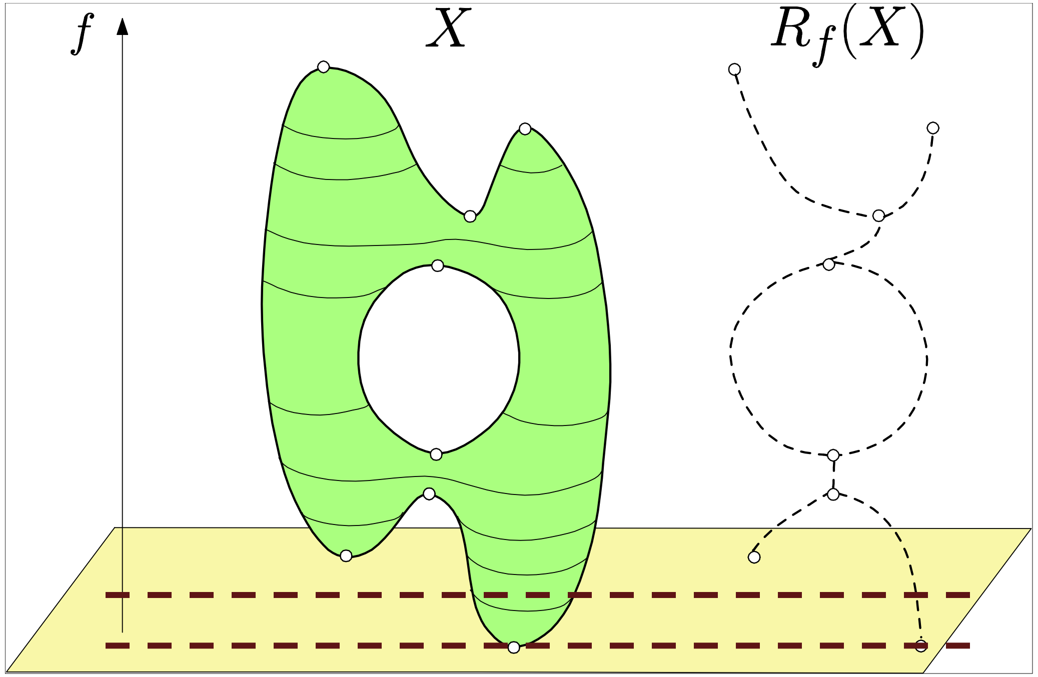
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



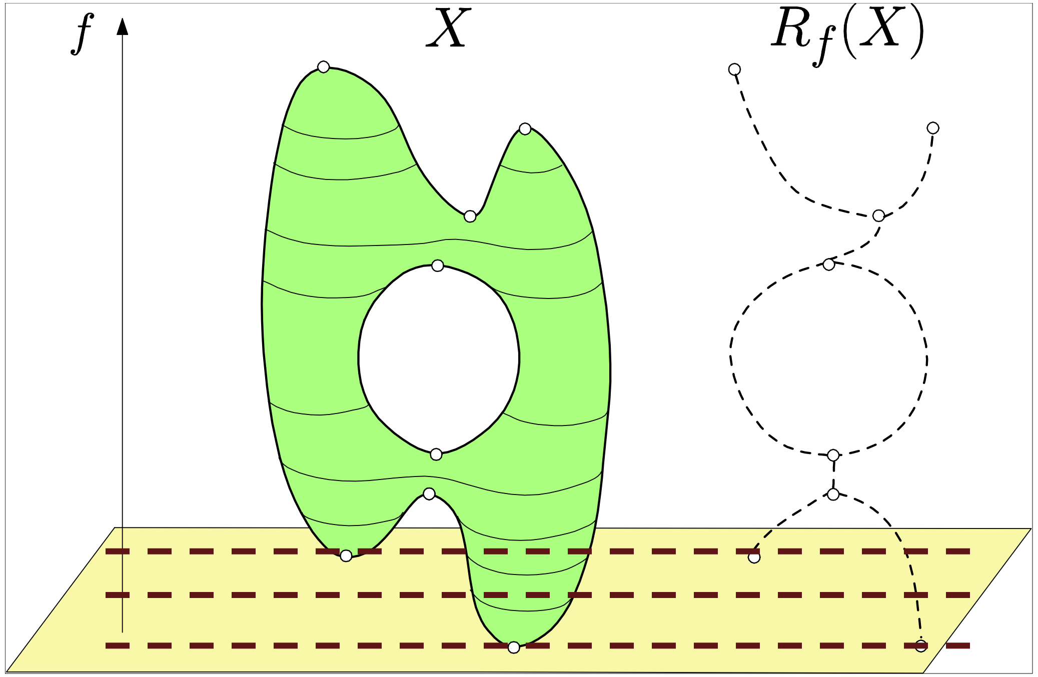
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



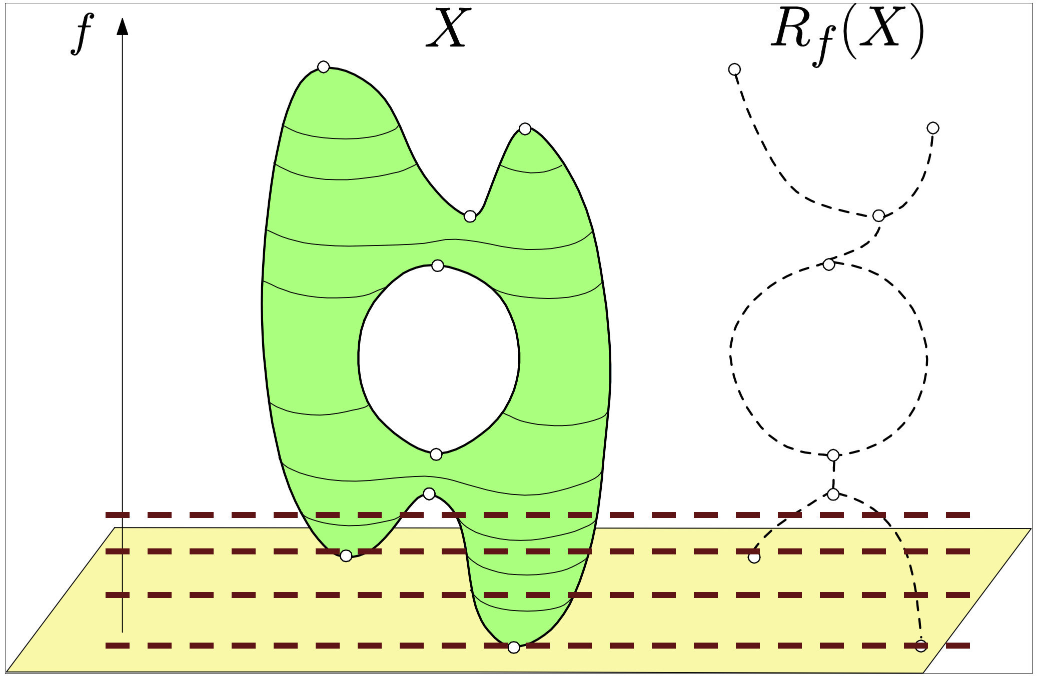
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



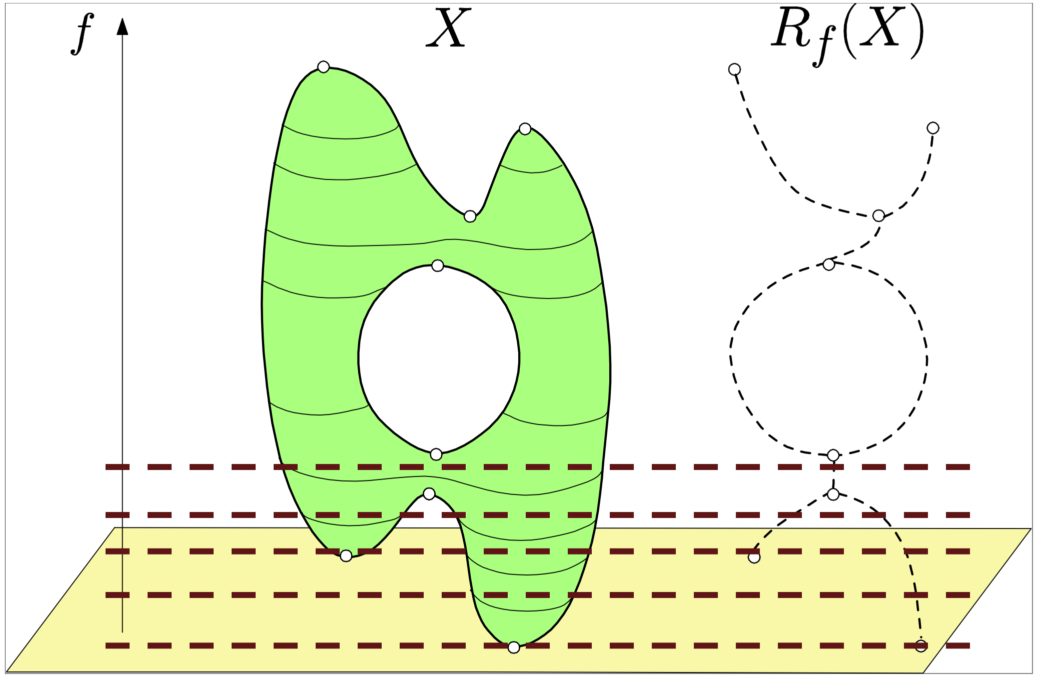
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



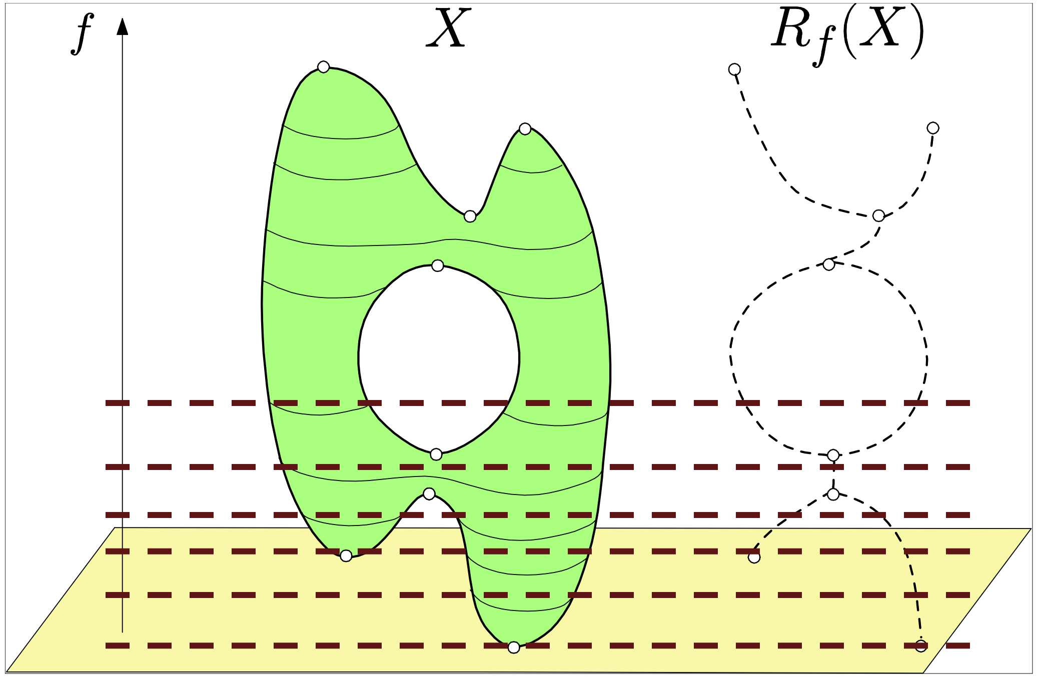
More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



More on Reeb Graph

- Imagine sweeping X in increasing order of f
  - Track the changes in 0-th homology of level sets
    - i.e, changes in contours
  - Node:
    - where changes happen
  - Arc:
    - evolution of a single contour



Reeb graph of Morse function

- Given an m-manifold *M* and *f: M → R*,
  - A point $p \in M$ is *critical* if gradient of *f* vanishes at *p*
- A critical point is non-degenerate
  - if it has non-degenerate Hessian
- For every non-degenerate critical point

MORSE LEMMA. Let u be a non-degenerate critical point of $f : \mathbb{M} \to \mathbb{R}$. There are local coordinates with $u = (0, 0, \ldots, 0)$ such that

$$f(x) = f(u) - x_1^2 - \ldots - x_p^2 + x_{p+1}^2 + \ldots + x_d^2$$

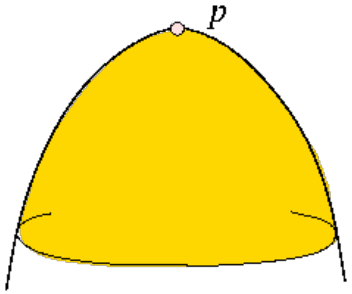
for every point $x = (x_1, x_2, \ldots, x_d)$ in a small neighborhood of u.

Critical Points cont.

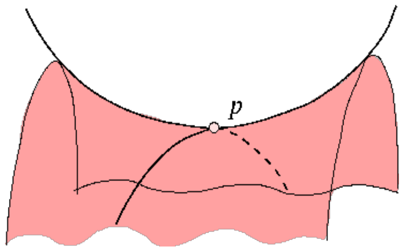
- For non-degenerate critical points
- Suppose M is 2-manifold

Critical Points cont.

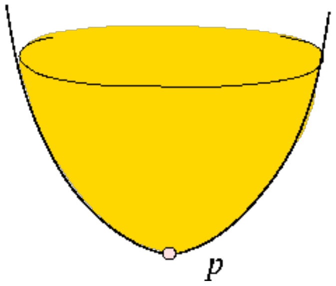
- For non-degenerate critical points
- Suppose M is 2-manifold



max

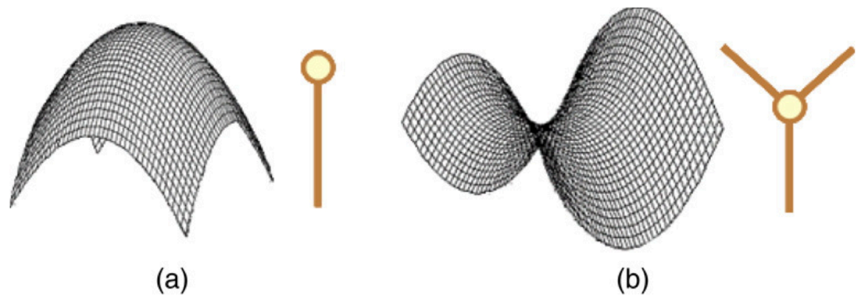


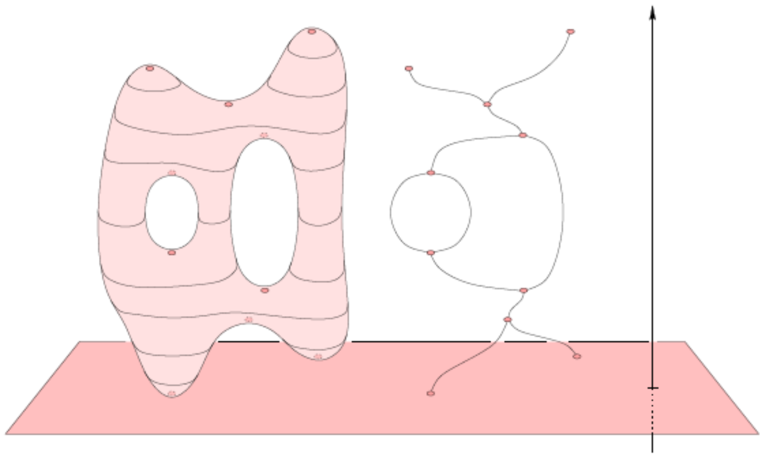
saddle



min

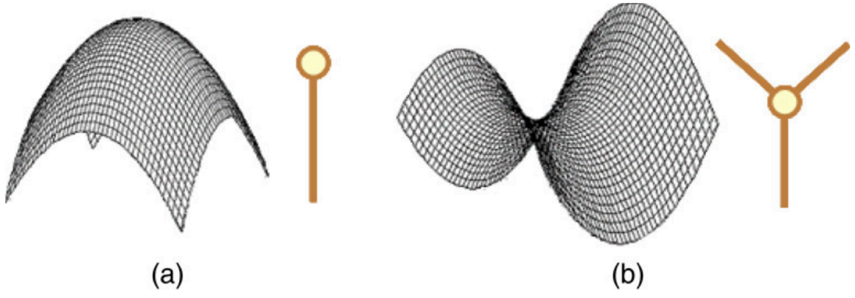
Bijection between critical points and tree nodes

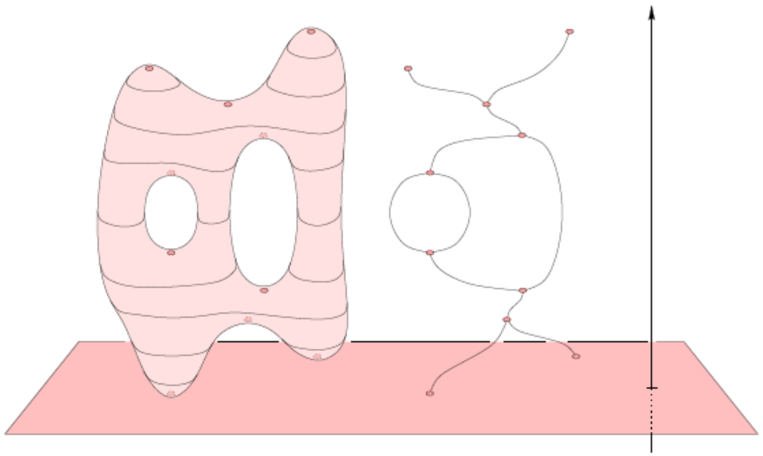




Bijection between critical points and tree nodes

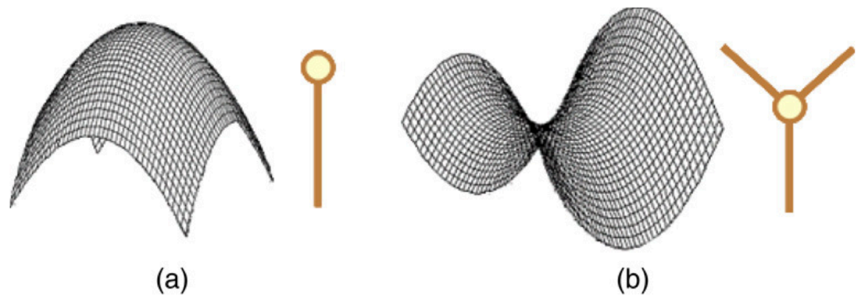
- If M is an d-manifold and $f: M \rightarrow R$ a Morse function

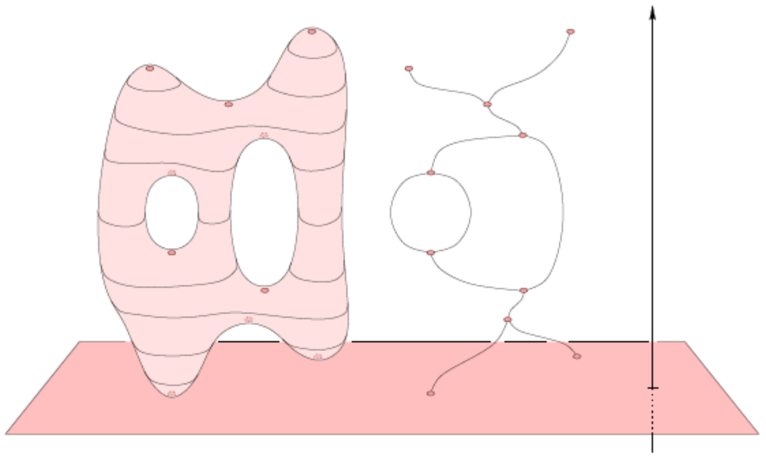




Bijection between critical points and tree nodes

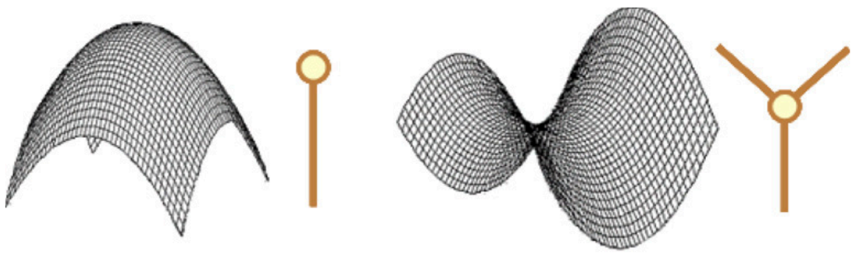
- If M is an d-manifold and $f: M \to R$ a Morse function
  - Degree 1 nodes:
    - Minimum or maximum



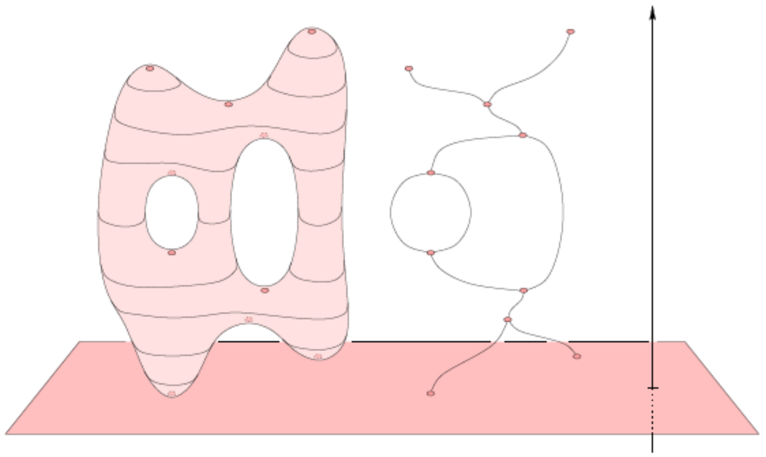


Bijection between critical points and tree nodes

- If M is an d-manifold and $f: M \to R$ a Morse function
  - Degree 1 nodes:
    - Minimum or maximum
  - Degree 3 nodes:
    - 1-saddles that merge two contours (merging forks)

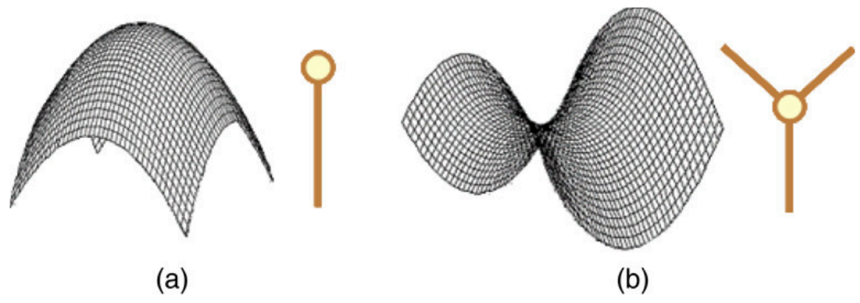


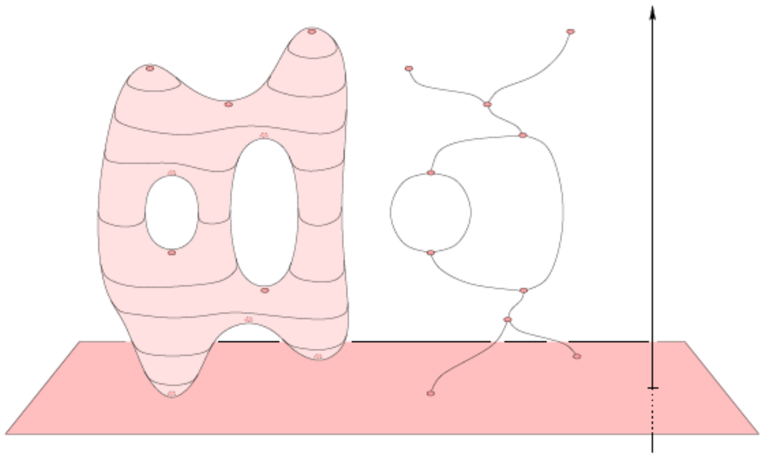
(a) (b)



Bijection between critical points and tree nodes

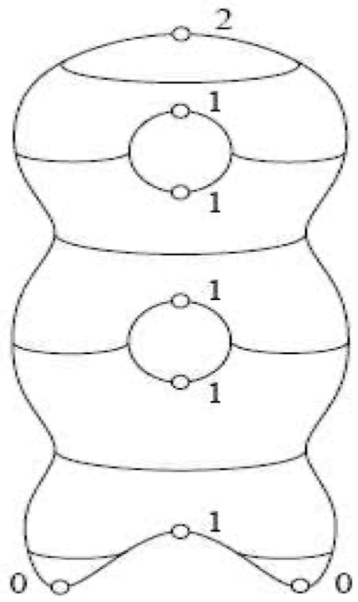
- If M is an d-manifold and $f: M \to R$ a Morse function
  - Degree 1 nodes:
    - Minimum or maximum
  - Degree 3 nodes:
    - 1-saddles that merge two contours (merging forks)
    - or (d-1)-saddles that split a contour into two (splitting forks)
  - Degree 2 nodes:
    - All other nodes

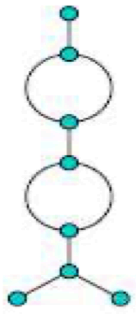




M is a 2-Manifold

The Reeb graph of a Morse function on a connected, orientable 2-manifold of genus g has g loops.





Homology Relations

- Reeb graph contains less topological information than its original space

Lemma

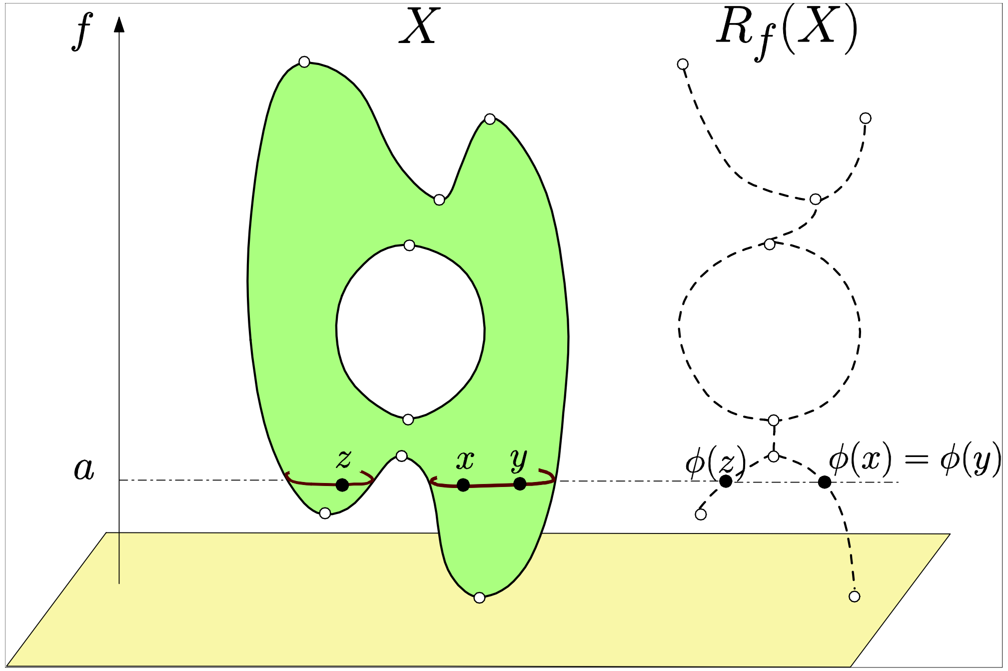
$$\beta_0(R_f(X)) = \beta_0(X)$$

$$\beta_1(R_f(X)) \leq \beta_1(X)$$

In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

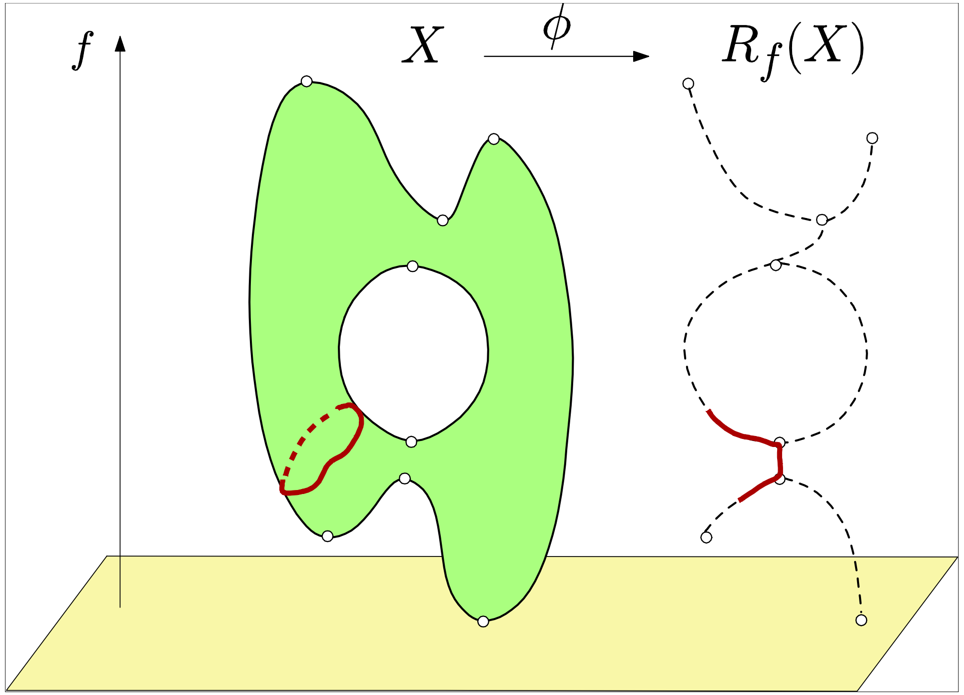


[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f : X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

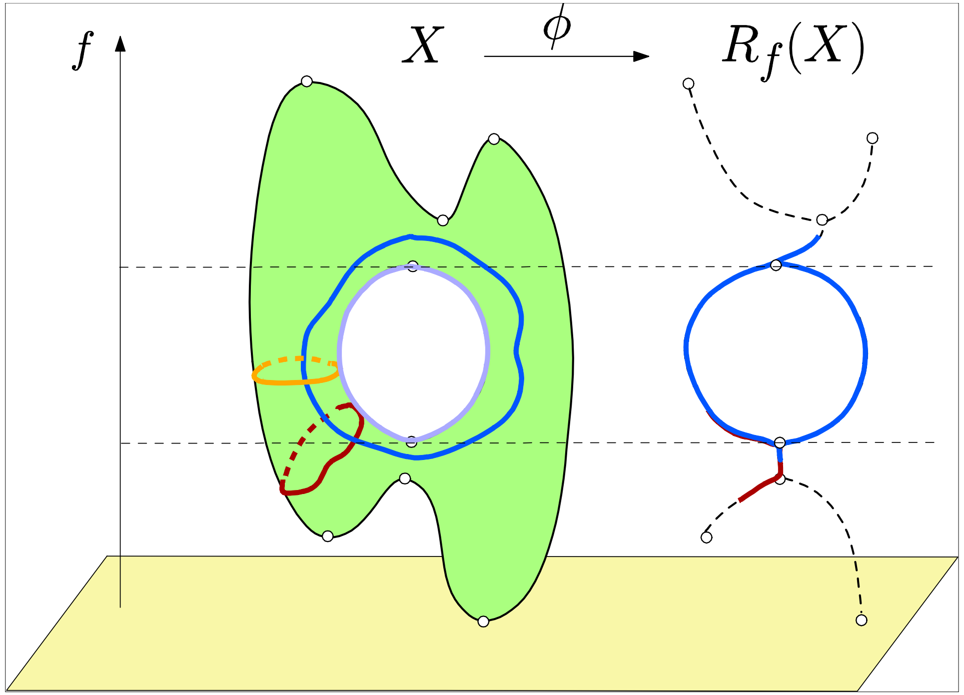


[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

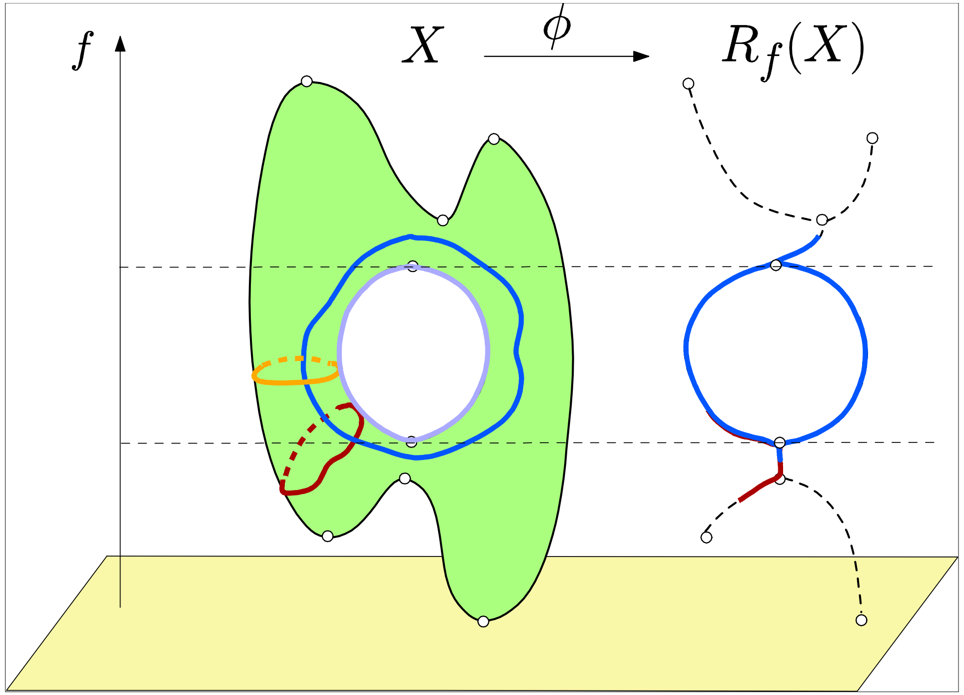


[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.

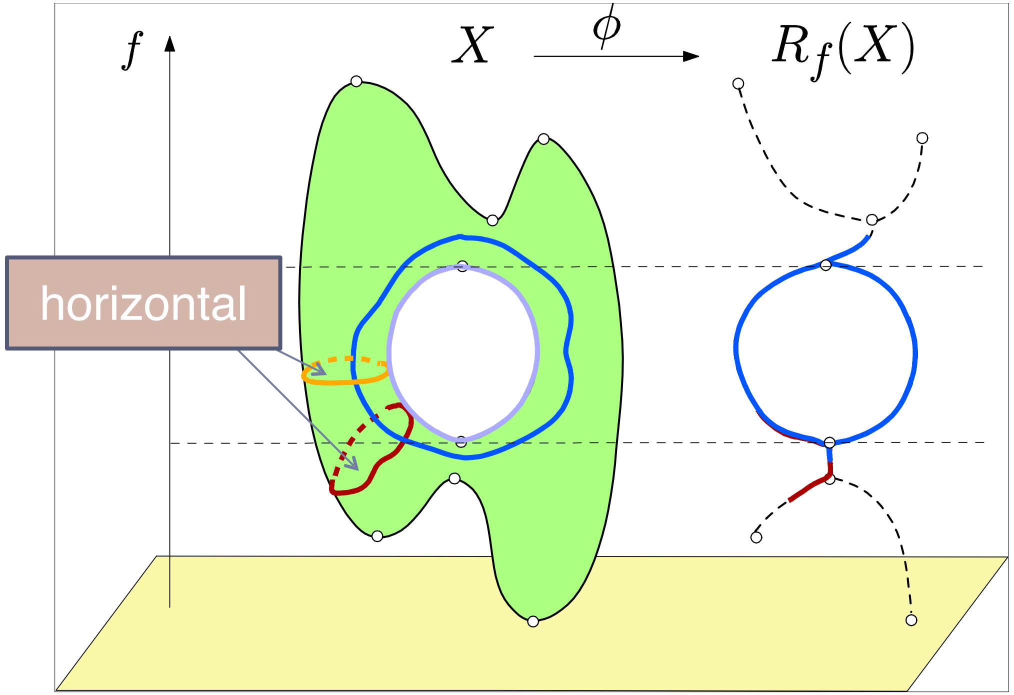
[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.



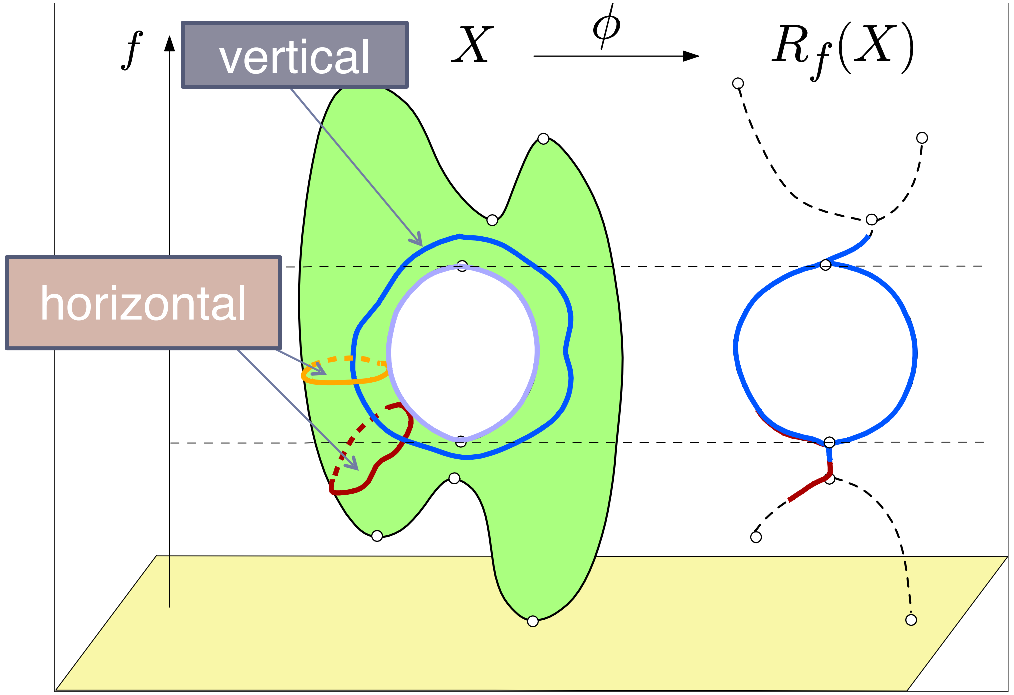
[Dey and Wang, DCG2012]

In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.



[Dey and Wang, DCG2012]

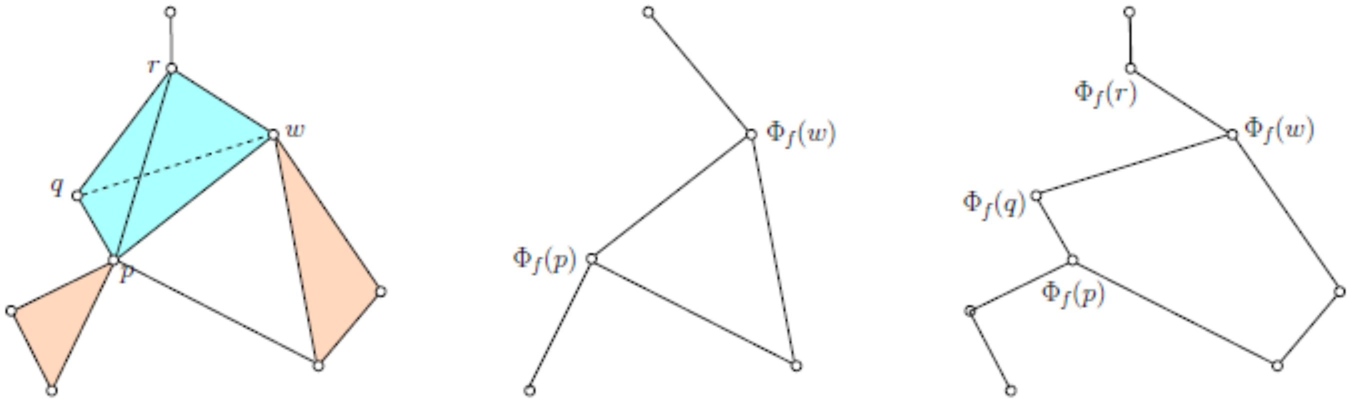
In general, the Reeb graph of a function $f: X \to R$ captures the so-called 1st *vertical homology* of X w.r.t. f.



[Dey and Wang, DCG2012]

PL Setting

- PL function f defined on simplicial complex K
  - f is decided by function values on the vertices V of K
  - only 2-skelenton (V,E,T) of K matters
  - Reeb graph $R_f(X)$ can be computed in $O(m \log n)$ time
    - m: number of vertices, edges, and triangles of X,
    - n: number of vertices

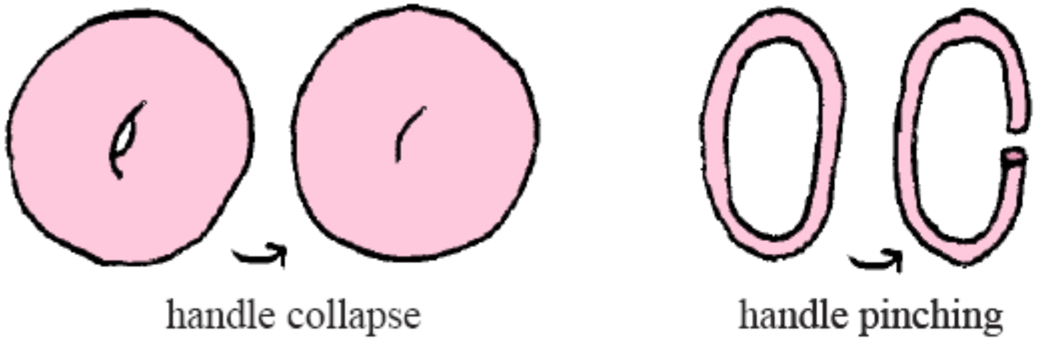


Applications

- Handle removal
- Skelentonize a shape
- Shape matching

Applications

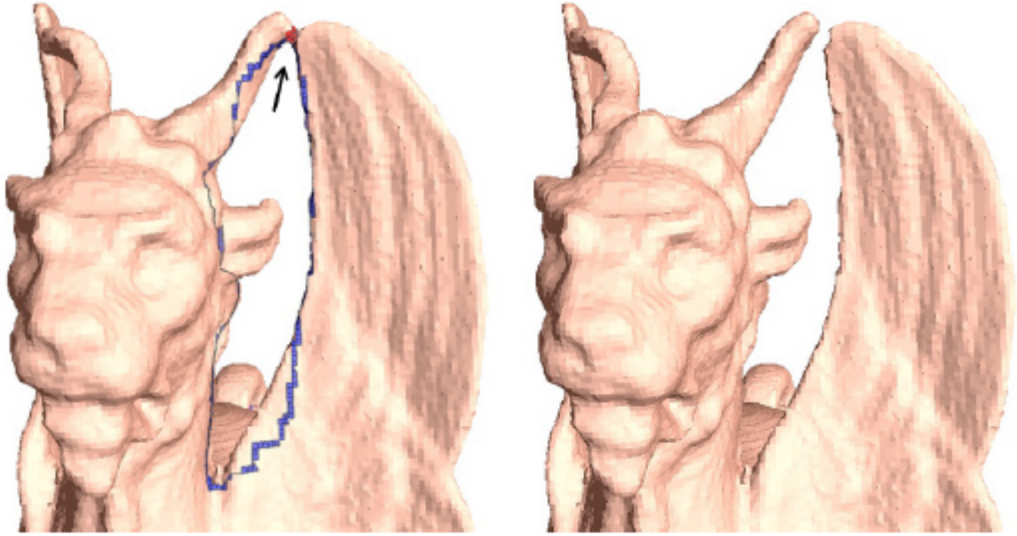
- Handle removal
- Skelentonize a shape
- Shape matching



Courtesy of Wood et al. 2002

Applications

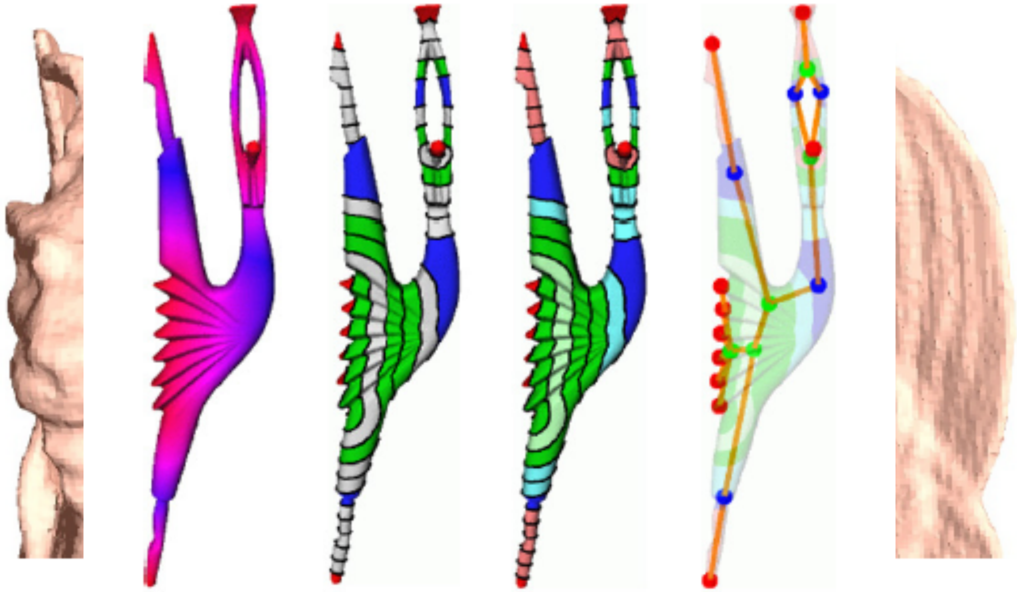
- Handle removal
- Skelentonize a shape
- Shape matching



Courtesy of Wood et al. 2002

Applications

- Handle removal
- Skelentonize a shape
- Shape matching



Courtesy of Biosotti et al. 2008

Applications

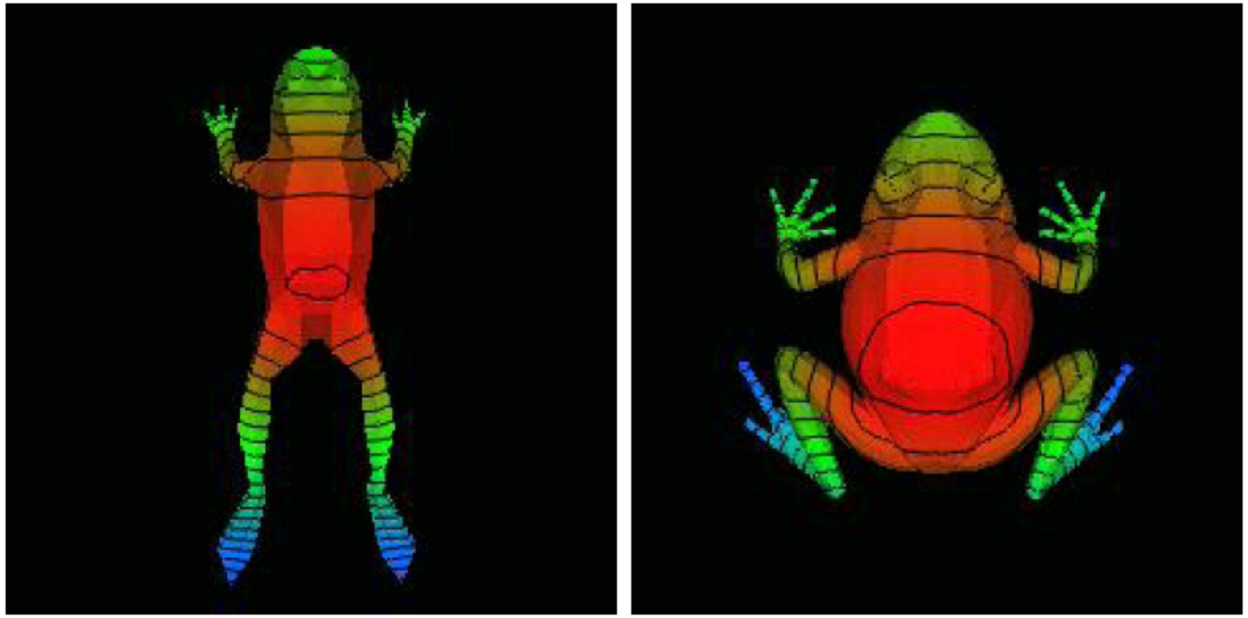
- Handle removal
- Skelentonize a shape
- Shape matching

(a) Our output (b) PGA [19] (c) LDPC [23]

Courtesy of Ge et al. 2011

Applications

- Handle removal
- Skelentonize a shape
- Shape matching



Courtesy of Hilaga et al. 2001

Reeb Space

- Given a topological space X and function $f : X \to Z$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a

- *Reeb space* $RS_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point
  - A continuous surjection $\phi : X \to R_f(X)$ s.t, $\phi(x) = \phi(y)$ if and only if x and y is in the same contour

Reeb Space

- Given a topological space X and function $f: X \to Z$
- *Level set* at value a :
  - $X_a := \{ x \in X \mid f(x) = a \}$
- A *contour* at value a :
  - a connected component of X_a
- *Reeb space* $RS_f(X)$ of X w.r.t. f:
  - continuous collapsing of each contour of f to a point

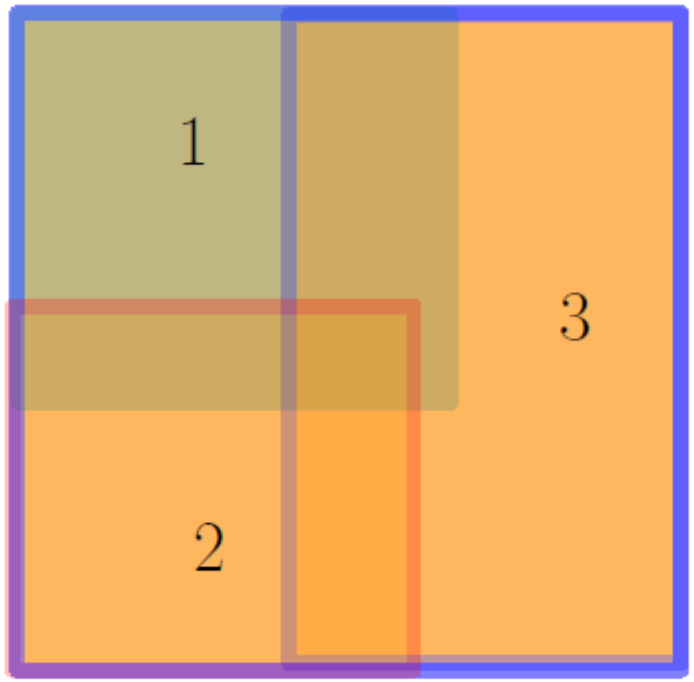
That is, Reeb space tracks connected components in the pre-image of any point in the co-domain Z.

- Study of Reeb space structure is intricate though. In general, it is not easy to compute the Reeb space.
  - [Edelsbrunner, Harer and Patel, 2008]
- The idea of viewing the structure of X from the lens of the map $f: X \to Z$ is interesting
  - In particular, often in practice, we may not know X but we can have several observations at points in X
- The Mapper construction!
  - [Singh, Mémoli and Carlsson, 2007]
  - Instead of considering pullbacks of all points in Z, and track their components, now considering pullbacks of elements in a cover of co-domain Z .
  - Tracking of components in such pullbacks is achieved via taking the nerve of these components.

Section 1:
Mapper: A topological summary of high dimensional data

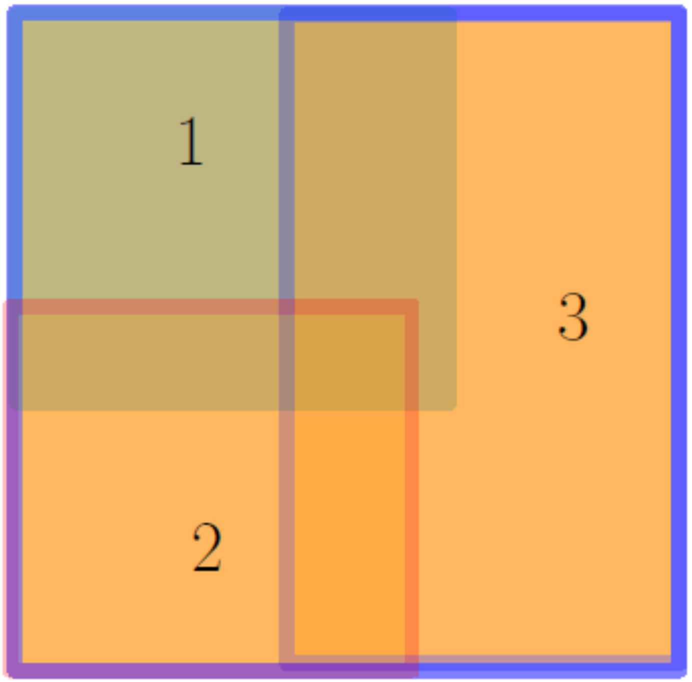
Covers and nerves

- A finite cover $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y



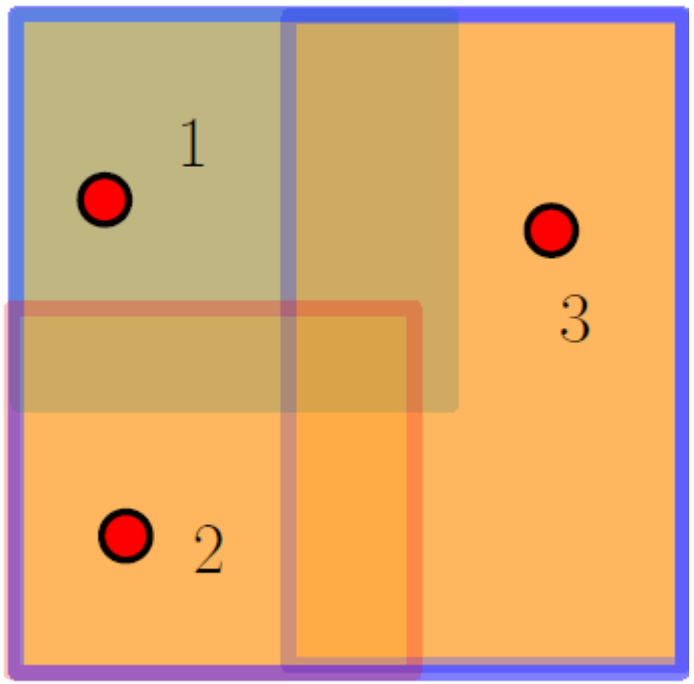
Covers and nerves

- A finite cover $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathscr{U}$: $\mathrm{N}(\mathscr{U})$
  - with vertex set A, and



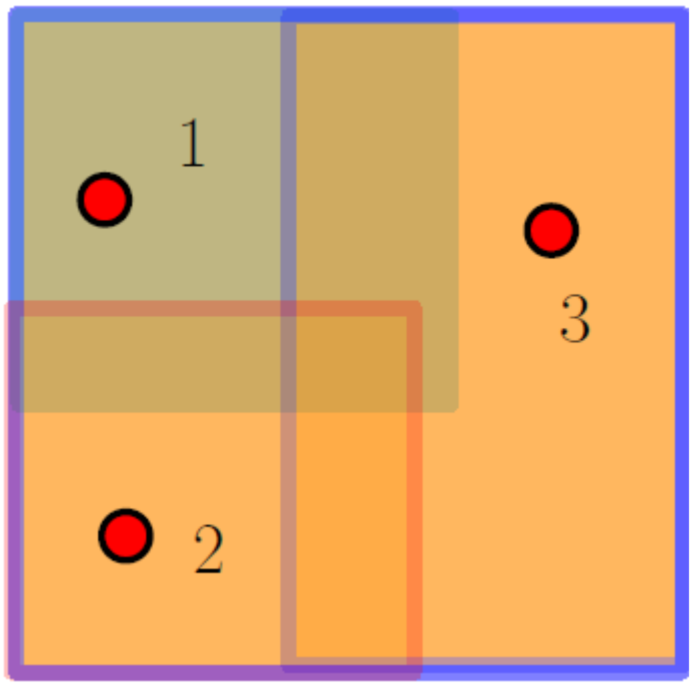
Covers and nerves

- A finite cover $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathscr{U}$: $\mathrm{N}(\mathscr{U})$
  - with vertex set A, and



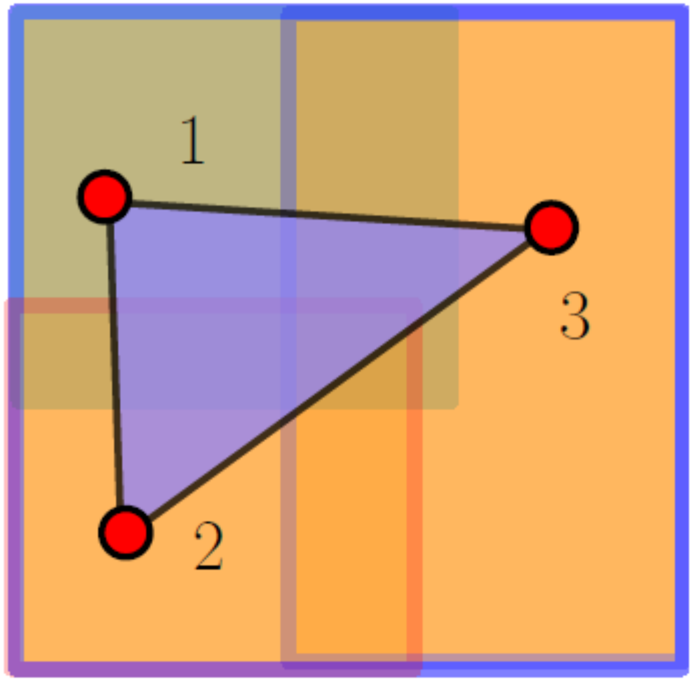
Covers and nerves

- A finite cover $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathscr{U}$: $\mathrm{N}(\mathscr{U})$
  - with vertex set A, and
  - simplex $(\alpha_0, \ldots, \alpha_k)$ iff $U_{\alpha_0} \cap U_{\alpha_1} \cap \cdots \cap U_{\alpha_k} \neq 0$



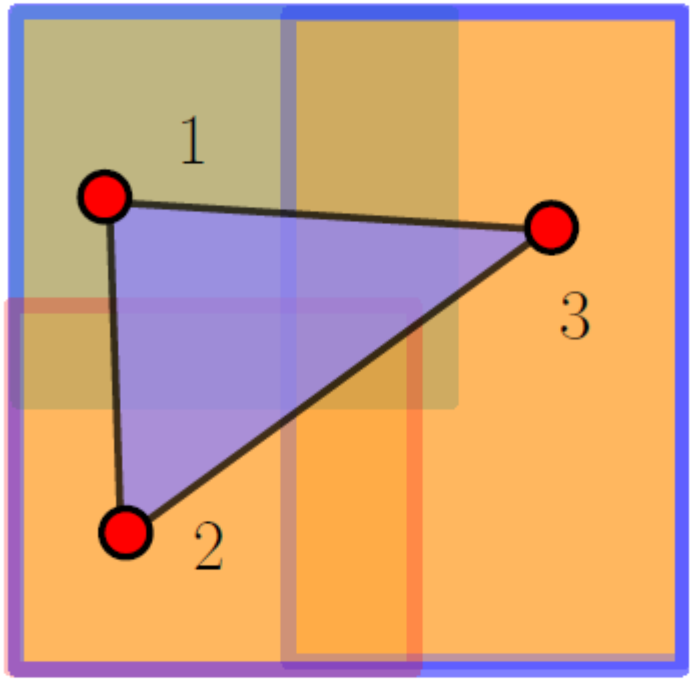
Covers and nerves

- A finite cover $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathscr{U}$: $\mathrm{N}(\mathscr{U})$
  - with vertex set A, and
  - simplex $(\alpha_0, \ldots, \alpha_k)$ iff $U_{\alpha_0} \cap U_{\alpha_1} \cap \cdots \cap U_{\alpha_k} \neq 0$



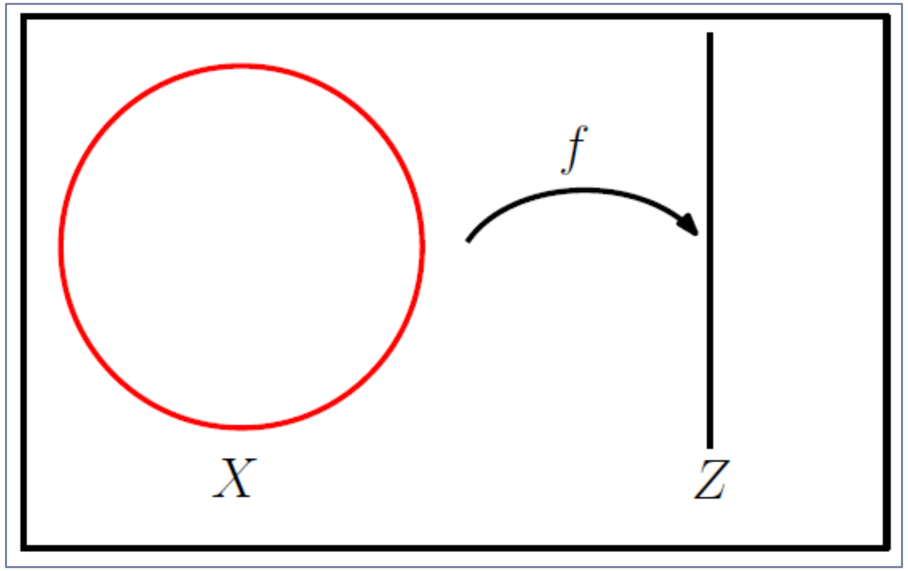
Covers and nerves

- A finite cover $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ of a space Y
- Nerve of $\mathscr{U}$: $\mathrm{N}(\mathscr{U})$
  - with vertex set A, and
  - simplex $(\alpha_0, \ldots, \alpha_k)$ iff $U_{\alpha_0} \cap U_{\alpha_1} \cap \cdots \cap U_{\alpha_k} \neq 0$
- One can view the nerve of a space as a discrete representation of the space via the cover
  - the cover provides the discretization



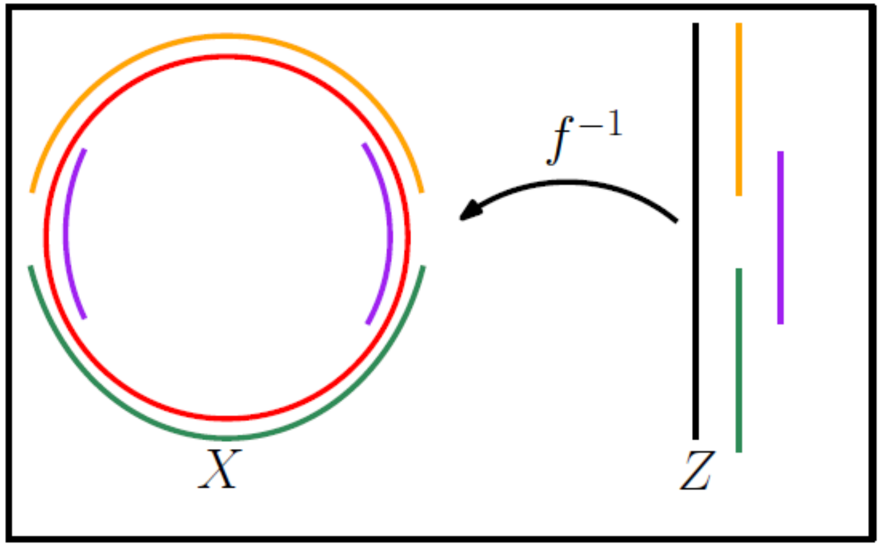
Pullback Cover

- Let $f\colon X \to Z$ be continuous and well-behaved, $\mathscr{U} = \{U_\alpha\}_{\alpha\in A}$ a finite cover of Z



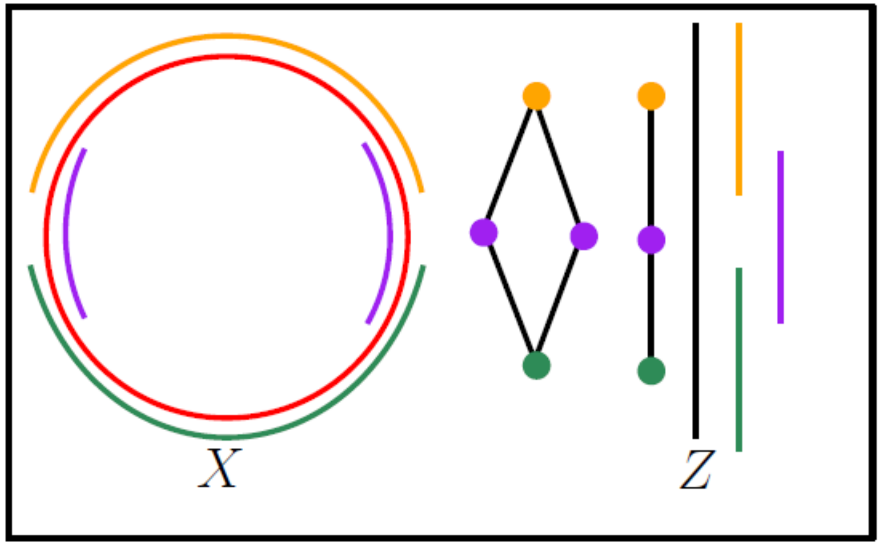
Pullback Cover

- Let $f: X \to Z$ be continuous and well-behaved, $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ a finite cover of Z
- Pullback cover via f and $\mathscr{U}$:
  - Connected components of $f^{-1}(U_\alpha) = \bigcup_{i=1}^{j_\alpha} V_{\alpha,i}$, for all $\alpha \in A$, form a cover $f^*(\mathscr{U})$ of X

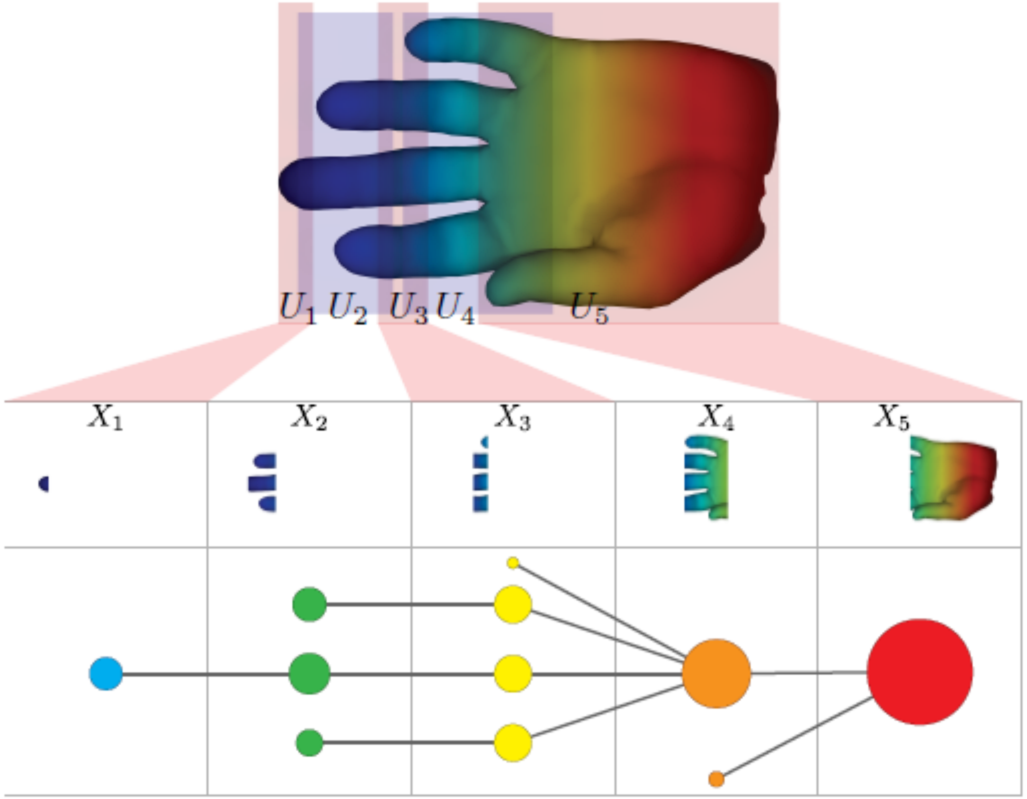


Pullback Cover

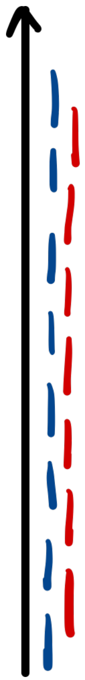
- Let $f: X \to Z$ be continuous and well-behaved, $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ a finite cover of Z
- Pullback cover via f and $\mathscr{U}$:
  - Connected components of $f^{-1}(U_\alpha) = \bigcup_{i=1}^{j_\alpha} V_{\alpha,i}$, for all $\alpha \in A$, form a cover $f^*(\mathscr{U})$ of X
- Mapper: $M(\mathscr{U}, f) := N(f^*(\mathscr{U}))$ the nerve of the pullback cover $f^*(\mathscr{U})$!

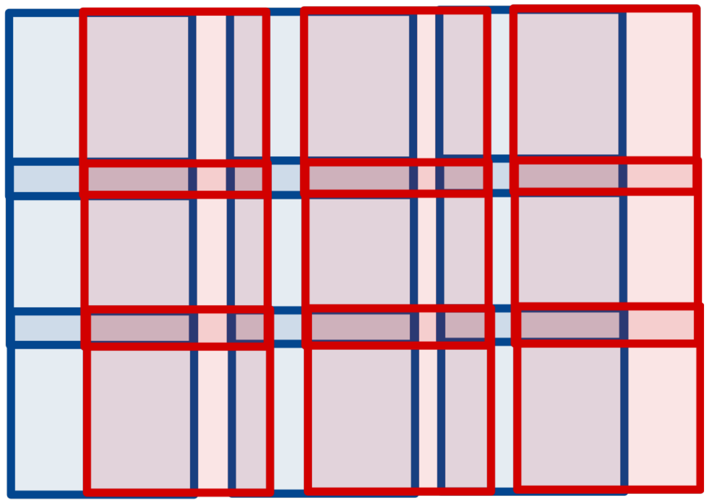


Another example



- Z is usually chosen to be $\mathbb{R}$ or $\mathbb{R}^2$
- Intervals or hypercube covers

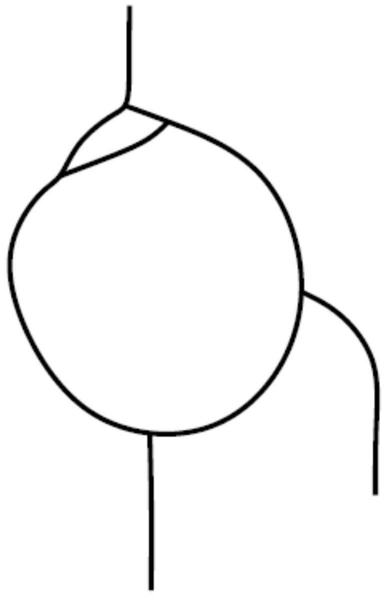




Mapper vs Reeb Space

- Consider a real-valued function $f: X \to R$ (i.e, $Z = R$)

$X = G$

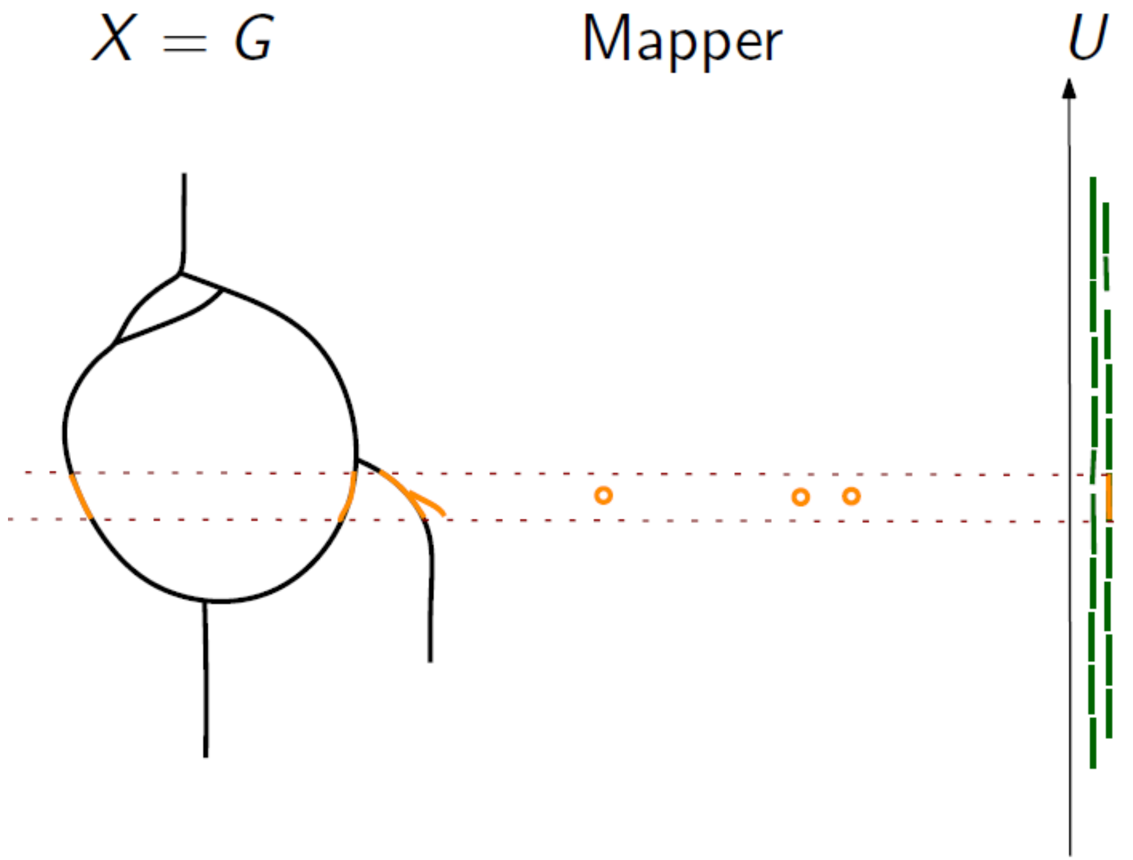


Mapper

U

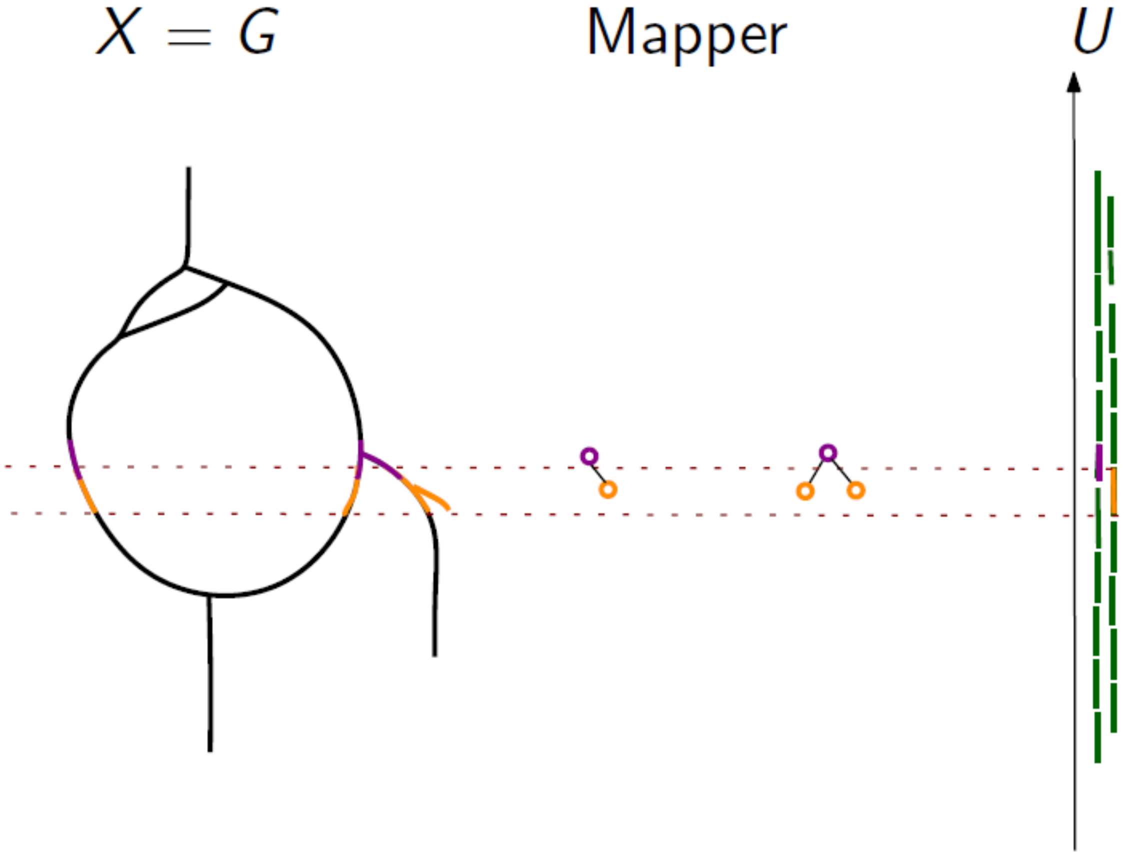
Mapper vs Reeb Space

- Consider a real-valued function $f: X \to R$ (i.e, $Z = R$)



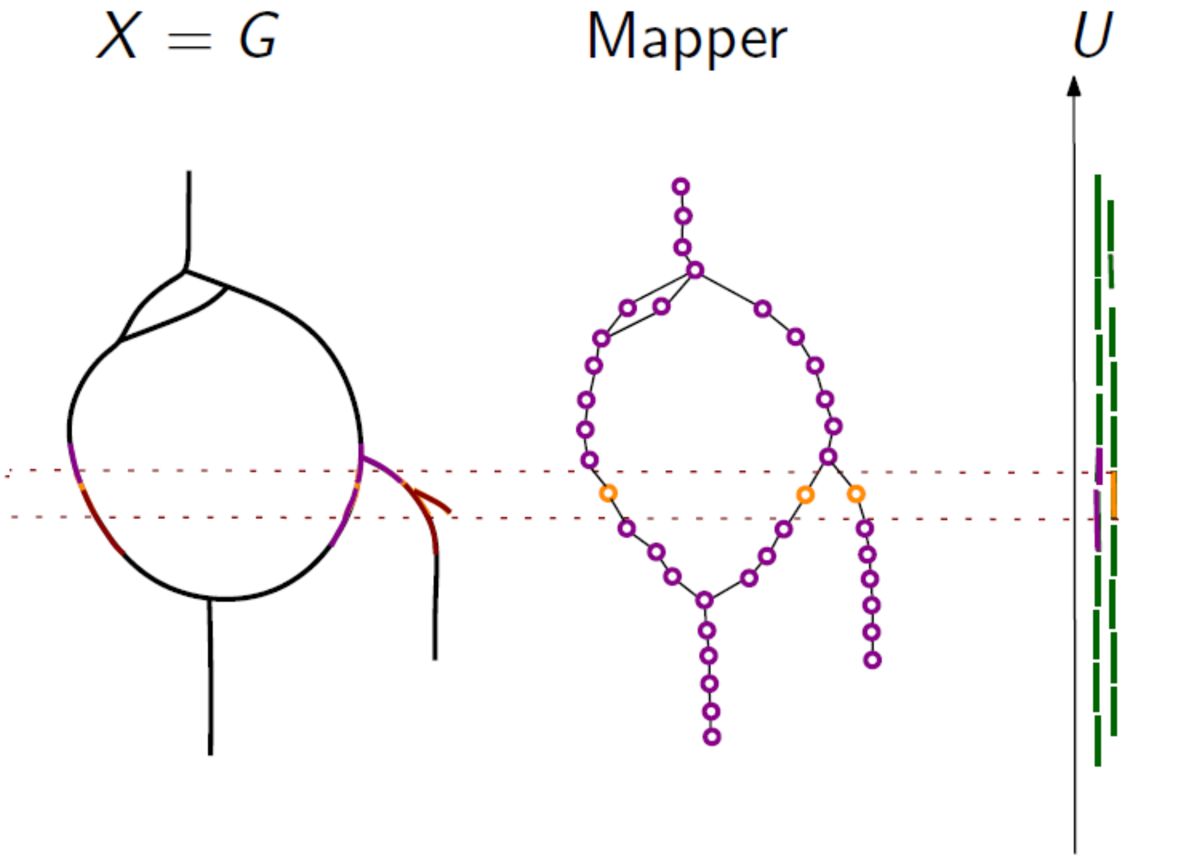
Mapper vs Reeb Space

- Consider a real-valued function $f: X \to R$ (i.e, $Z = R$)



Mapper vs Reeb Space

- Consider a real-valued function $f\colon X \to R$ (i.e, $Z = R$)



Mapper vs. Reeb Space

- In some sense, mapper structure can be considered as a coarsening of Reeb space via the coarsening of the co-domain (via a cover of it)
  - Certain convergences results known [Munch, B. Wang, 2016], [Dey, Mémoli, Wang, 2017]

Theorem 32. *Under the conditions above,*

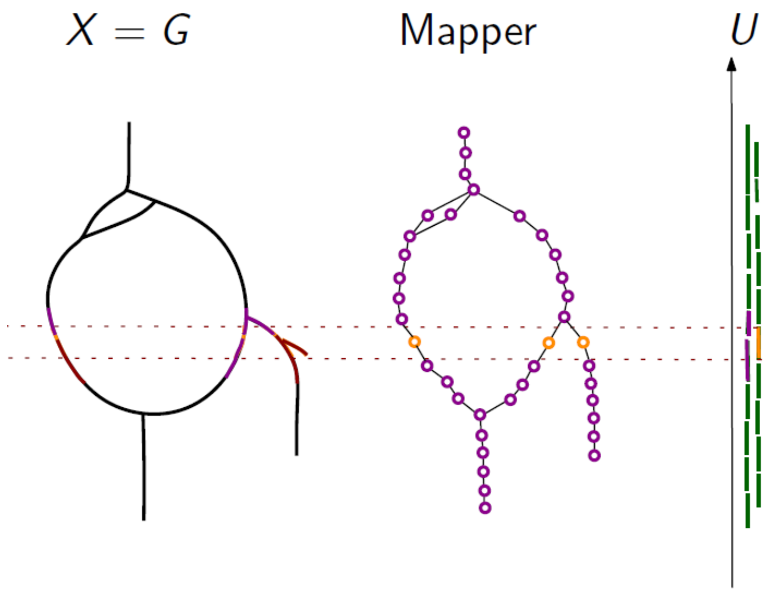
$$d_{GH}\big((R_f, \tilde{d}_f), (P_\delta, d_\delta)\big) \leq 5\delta.$$

Mapper vs. Reeb Space

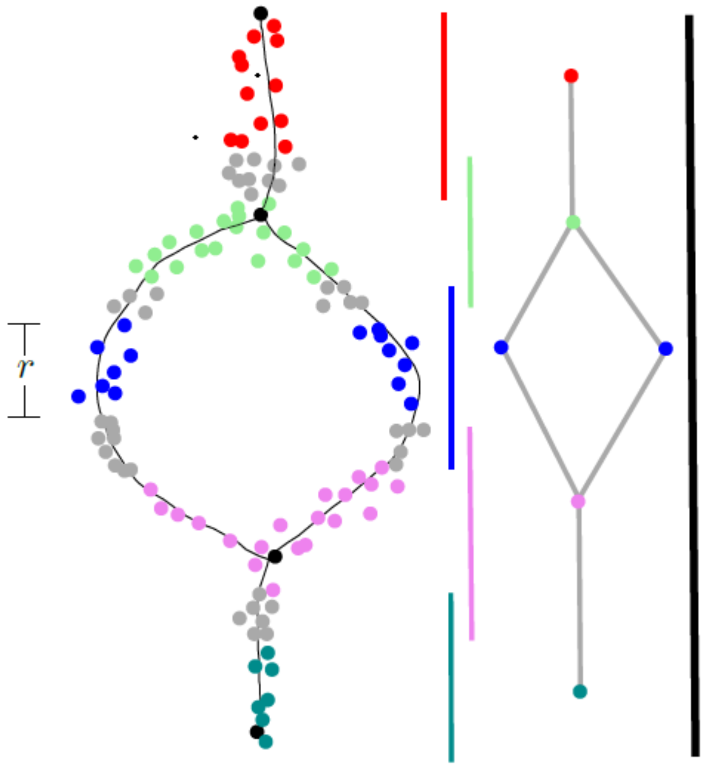
- In some sense, mapper structure can be considered as a coarsening of Reeb space via the coarsening of the co-domain (via a cover of it)
  - Certain convergences results known [Munch, B. Wang, 2016], [Dey, Mémoli, Wang, 2017]

Theorem 32. *Under the conditions above,*

$$d_{GH}\big((R_f, \tilde{d}_f), (P_\delta, d_\delta)\big) \leq 5\delta.$$

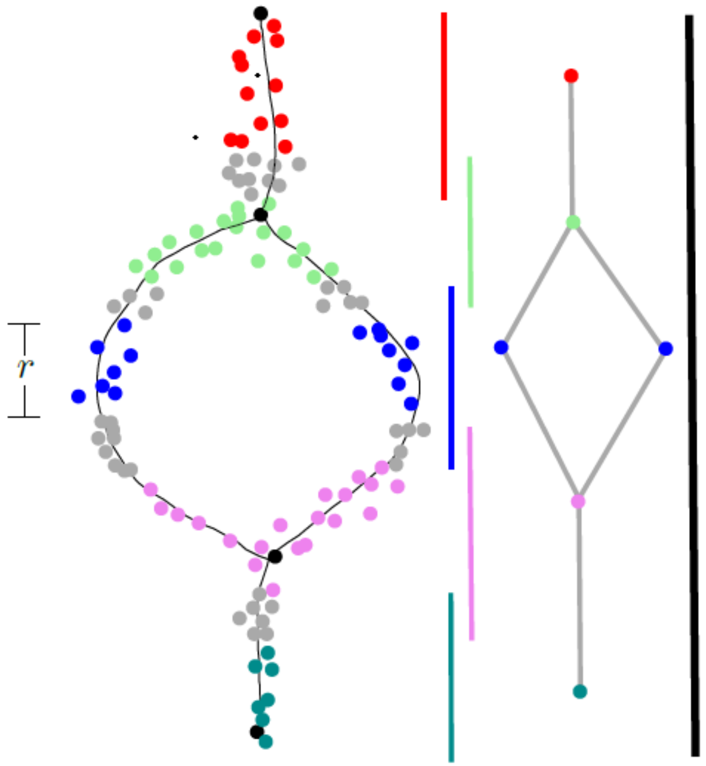


Mapper in Practice



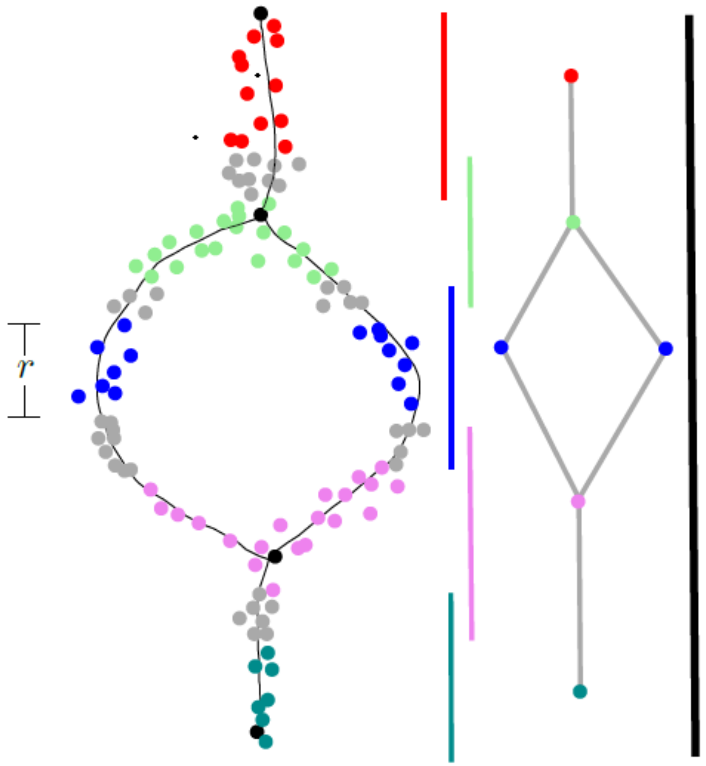
Mapper in Practice

- If X is discrete, then $f^{-1}U_\alpha$ is also discrete and is just a union of points. The nerve doesn't make sense now
- Use clustering algorithm to construct clusters from the set of points $f^{-1}U_\alpha$
  - DBSCAN
  - Single linkage clustering
  - …



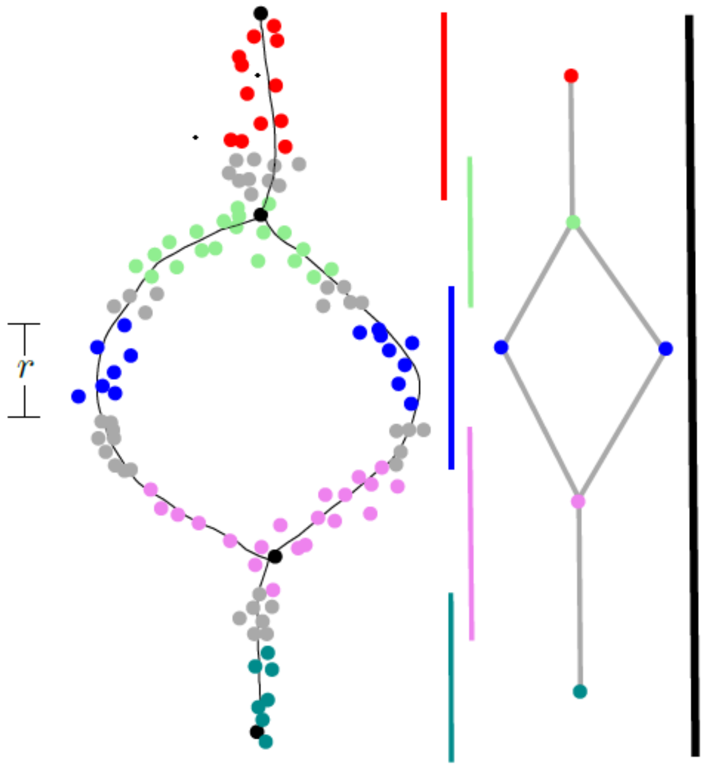
Mapper in Practice

- In practice, for point clouds data, a clustering algorithm can be applied to the pullback of each cover elements to identify the components



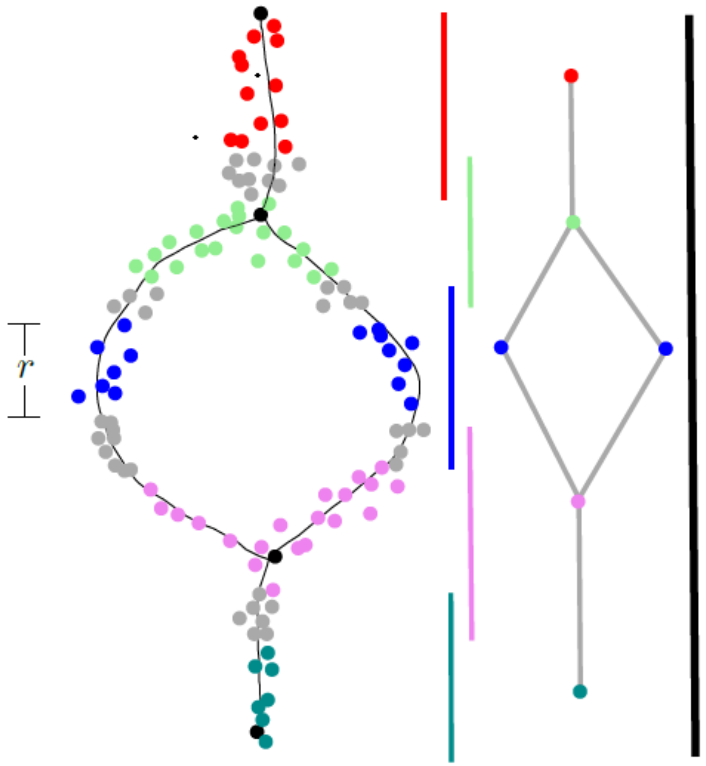
Mapper in Practice

- In practice, for point clouds data, a clustering algorithm can be applied to the pullback of each cover elements to identify the components
- 1-skeleton of the mapper structure is often used, as a platform for data exploration



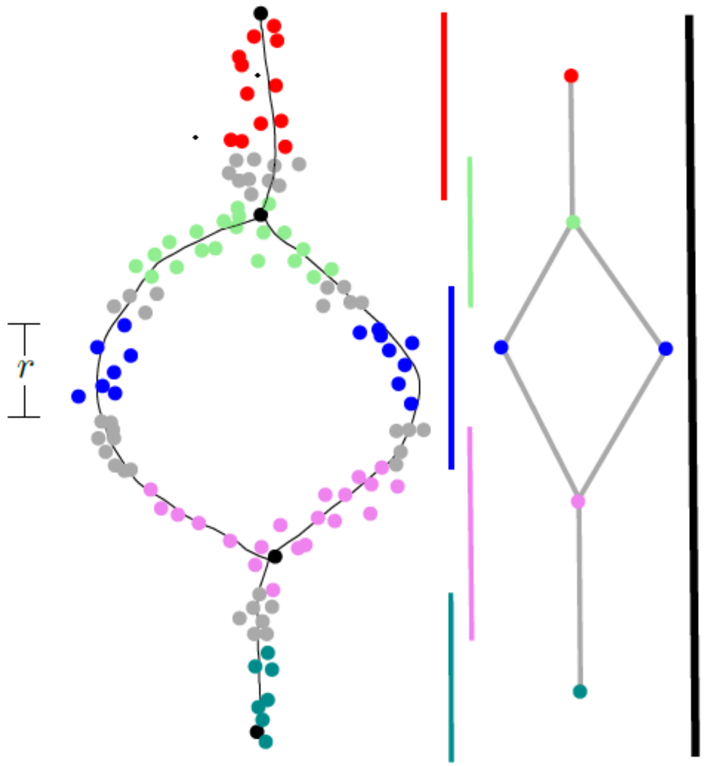
Mapper in Practice

- In practice, for point clouds data, a clustering algorithm can be applied to the pullback of each cover elements to identify the components
- 1-skeleton of the mapper structure is often used, as a platform for data exploration
- Mapper can be used as a replacement for dimensionality reduction
  - serving as a low-dimensional metaphor for the continuous space of high dimensional data

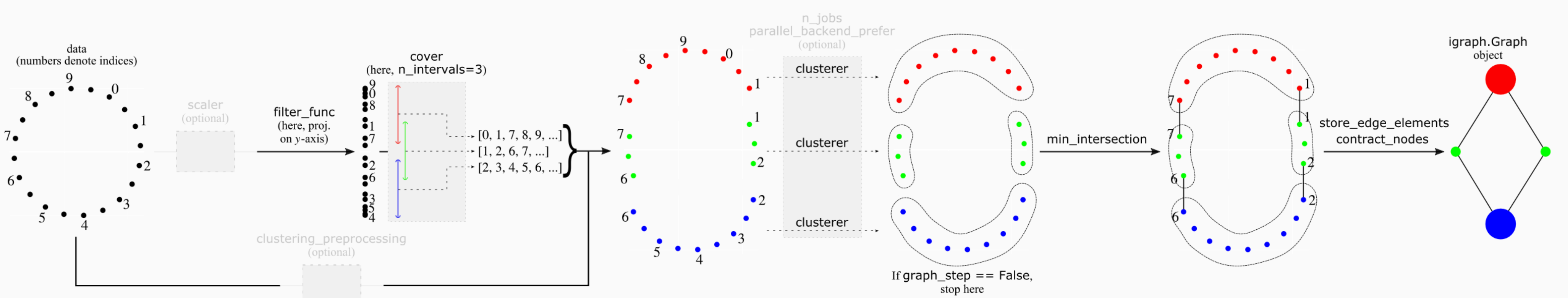


Mapper in Practice

- In practice, for point clouds data, a clustering algorithm can be applied to the pullback of each cover elements to identify the components
- 1-skeleton of the mapper structure is often used, as a platform for data exploration
- Mapper can be used as a replacement for dimensionality reduction
  - serving as a low-dimensional metaphor for the continuous space of high dimensional data
- Input data can be just point cloud data
  - helper functions (called filter functions) will be used to serve as $f: X \to R^d$



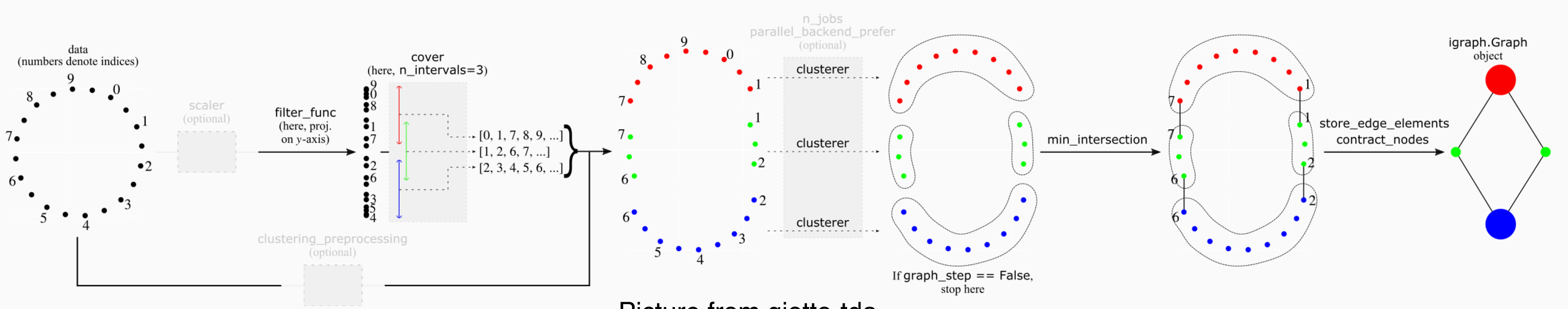
A standard Mapper pipeline in practice



Picture from giotto-tda

- Input: high dimensional PCD P
  - Step 1: Choose a few (d) filter functions $F: P \to R^d$, which could incorporate domain knowledge (or can be as simple as eigenfunctions from PCA)
  - Step 2: Create the Mapper structure (upto 2-skeleton, i.e, vertices, edges and triangles; often just 1-skeleton) w.r.t. F and some cover of R^d (i.e, just some ``rectangular"-tiling), where connected components in pullbacks are computed by some clustering algorithm
  - Step 3: Visualize the 1-skeleton (graph skeleton) of Mapper structure using a graph layout algorithm

A standard Mapper pipeline in practice



Picture from giotto-tda

- Parameters
  - Filter function $f : X \to \mathbb{R}$
  - Cover of $im(f)$ by open intervals
  - Clustering method and its parameters

Demo

Mapper for high-D data exploration

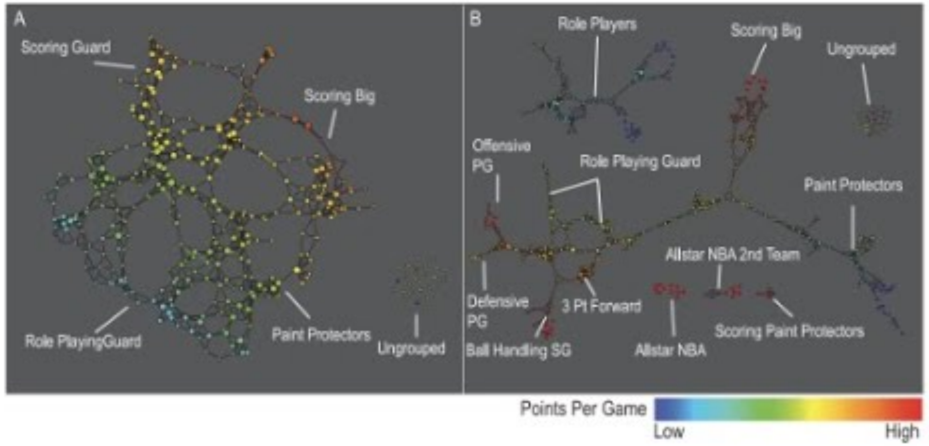
- Visualizing and exploring high dimensional data
  - Clustering is limited, ignoring the continuous space behind data
  - Dimensionality reduction can be misleading due to that the target dimension is much smaller than intrinsic dimension
- Mapper provides a low-D metaphor for the continuous space behind high dimensional data, via the lens of filter functions
  - bears similarity to the topological landscape via contour trees

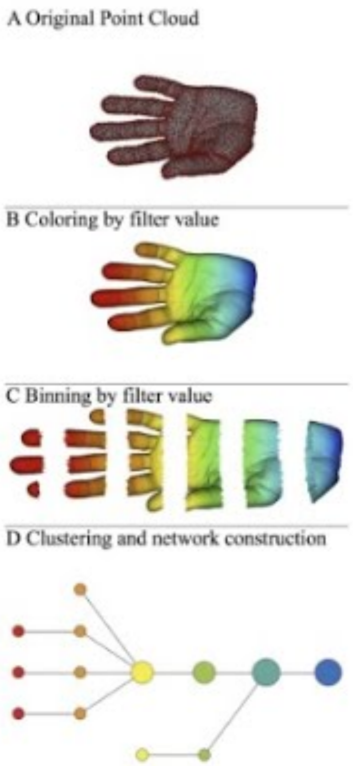
Mapper in Applications

- Extracting insights from the shape of complex data using topology, Lum et al., Nature, 2013
- Topological Data Analysis for Discovery in Preclinical Spinal Cord Injury and Traumatic Brain Injury, Nielson et al., Nature, 2015
- Using Topological Data Analysis for Diagnosis Pulmonary Embolism, Rucco et al., arXiv preprint, 2014
- Topological Methods for Exploring Low-density States in Biomolecular Folding Pathways, Yao et al., J . Chemical Physics, 2009
- CD8 T-cell reactivity to islet antigens is unique to type 1 while CD4 T-cell reactivity exists in both type 1 and type 2 diabetes, Sarikonda et al., J . Autoimmunity , 2013
- Innate and adaptive T cells in asthmatic patients: Relationship to severity and disease mechanisms, Hinks et al., J . Allergy Clinical Immunology , 2015

Using TDA to Define College Basketball Positions with Mapper

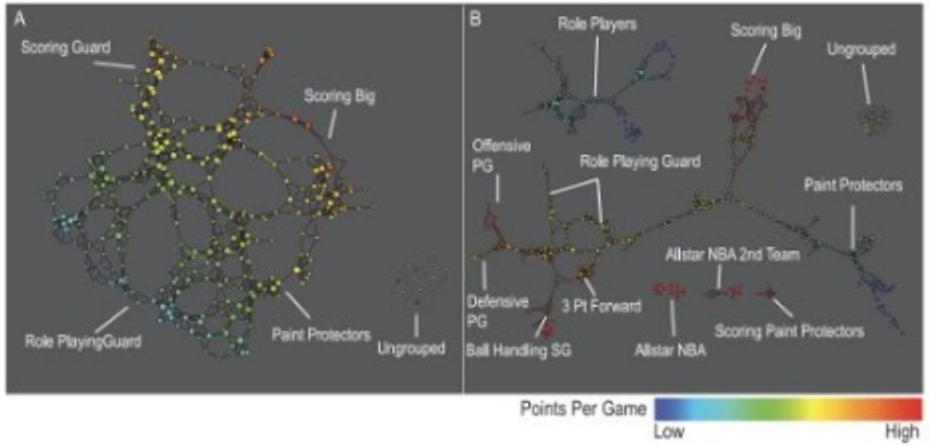
Mark Yukelis and Alan Suh

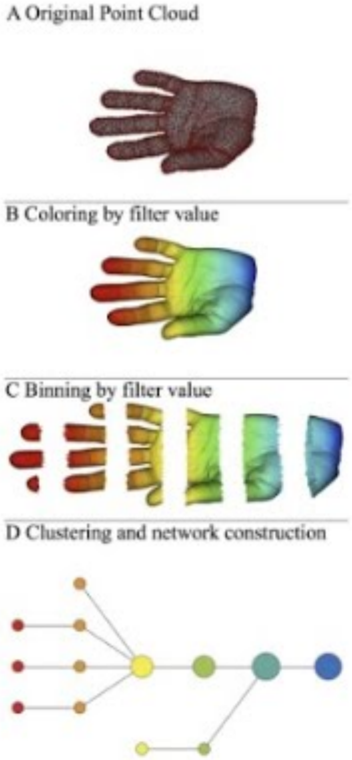




Using TDA to Define College Basketball Positions with Mapper

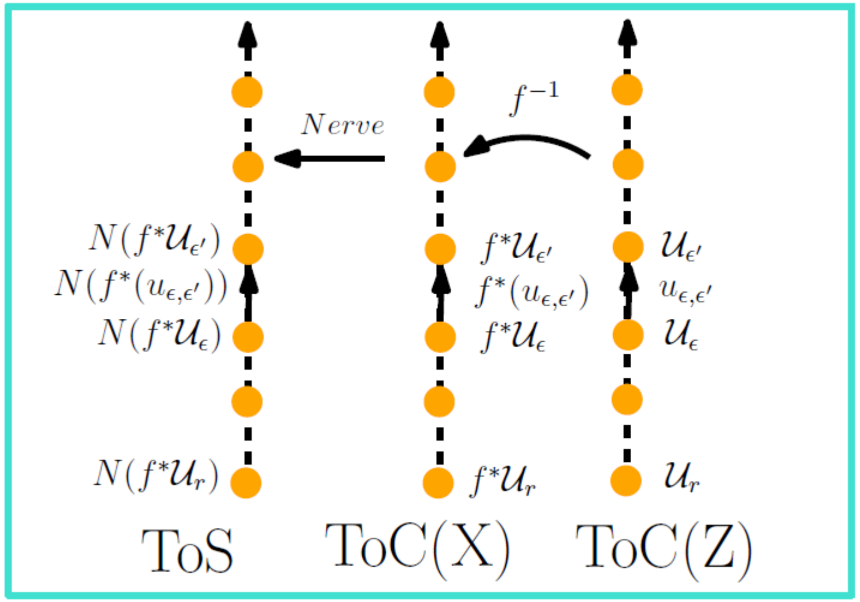
Mark Yukelis and Alan Suh





Section 2:
Multi-mapper: A multiscale representation of general maps

Main idea



- Consider a sequence of coarser and coarser covers (aka, look at the discretization at coarser and coarser resolution)
- Pull back, look at the sequence of coarser and coarser Mapper structures
- This gives rise to a sequence of simplicial complexes, called multiscale mapper, and we can compute its persistent homology.

Motivation

- Mapper is at a fixed scale
  - How to choose the scale? Why not look at all scales?

Motivation

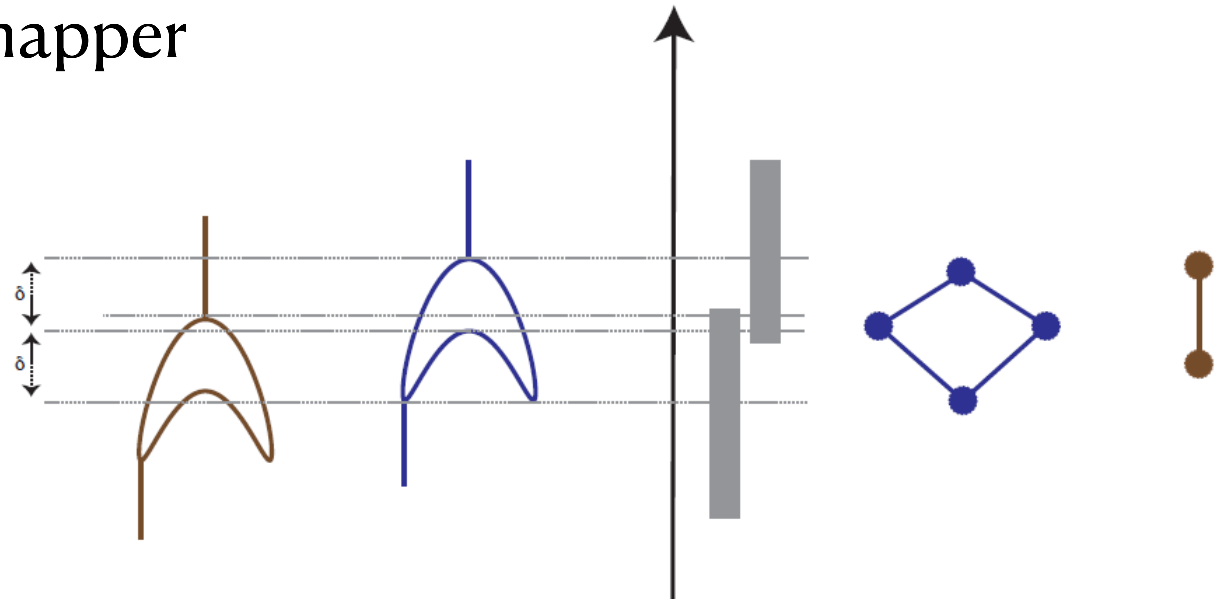
- Mapper is at a fixed scale
  - How to choose the scale? Why not look at all scales?
- Mapper is a structure, not an easy feature representation to compare
  - Can we obtain a persistence-like summary?

Motivation

- Mapper is at a fixed scale
  - How to choose the scale? Why not look at all scales?
- Mapper is a structure, not an easy feature representation to compare
  - Can we obtain a persistence-like summary?
- Un-stability of mapper

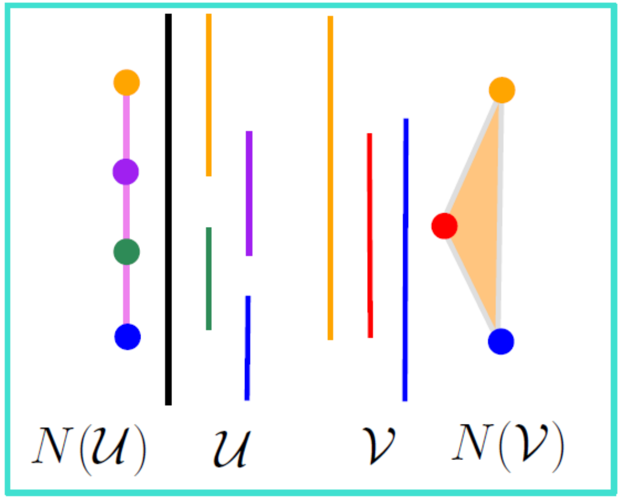
Motivation

- Mapper is at a fixed scale
  - How to choose the scale? Why not look at all scales?
- Mapper is a structure, not an easy feature representation to compare
  - Can we obtain a persistence-like summary?
- Un-stability of mapper

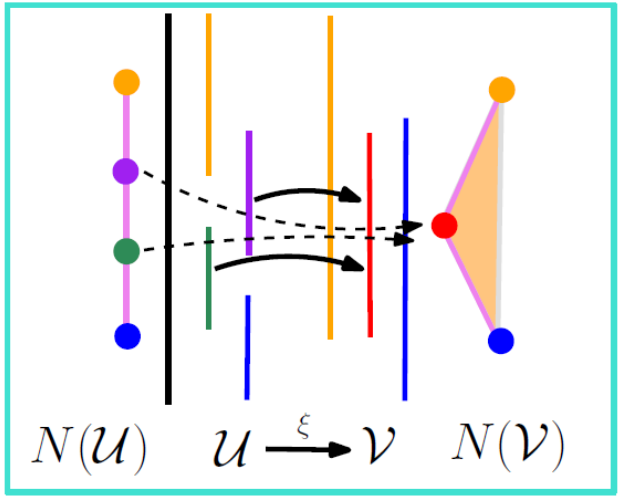


Maps between covers

Maps between covers

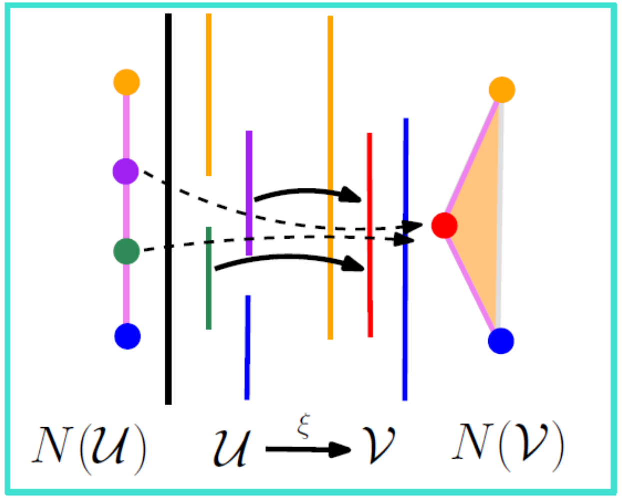


Maps between covers



Maps between covers

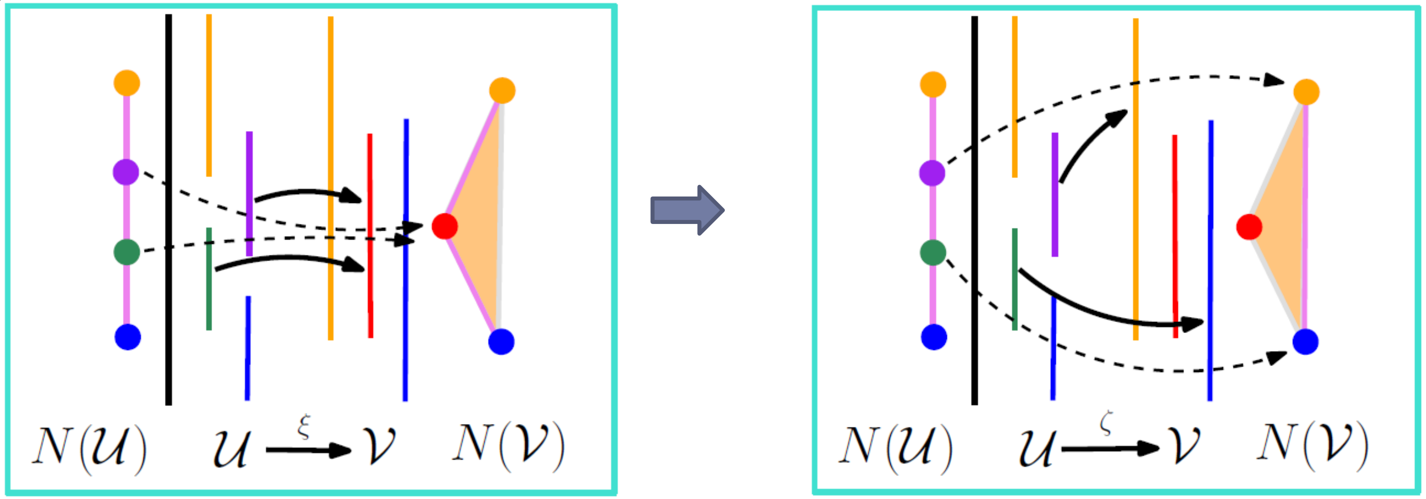
- Given two covers $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathscr{V} = \{V_\beta\}_{\beta \in B}$ of the same space Y,
  - a cover map $\xi : \mathscr{U} \to \mathscr{V}$ is any set map $\xi : A \to B$ such that $U_\alpha \subseteq V_{\xi(\alpha)}$ for all $\alpha \in A$
  - Intuitively, a cover map can connect covers at different resolutions (B is coarser than A)
- A cover map $\xi : A \to B$ induces a simplicial map in nerves: $N(\xi) : N(\mathscr{U}) \to N(\mathscr{V})$



Maps between covers

- Given two covers $\mathscr{U} = \{U_\alpha\}_{\alpha\in A}$ and $\mathscr{V} = \{V_\beta\}_{\beta\in B}$ of the same space Y,
  - a cover map $\xi\colon \mathscr{U} \to \mathscr{V}$ is any set map $\xi\colon A \to B$ such that $U_\alpha \subseteq V_{\xi(\alpha)}$ for all $\alpha \in A$
  - Intuitively, a cover map can connect covers at different resolutions (B is coarser than A)
- A cover map $\xi\colon A \to B$ induces a simplicial map in nerves:

$$N(\xi) : N(\mathscr{U}) \to N(\mathscr{V})$$



Maps between covers

- Given two covers $\mathscr{U} = \{U_\alpha\}_{\alpha \in A}$ and $\mathscr{V} = \left\{V_\beta\right\}_{\beta \in B}$ of the same space Y,
  - a cover map is any set map $\xi: A \to B$ such that $U_\alpha \subseteq V_{\xi(\alpha)}$ for all $\alpha \in A$
  - Intuitively, a cover map can connect covers at different resolutions (B is coarser than A)
- A cover map $\xi: A \to B$ induces a simplicial map in nerves:

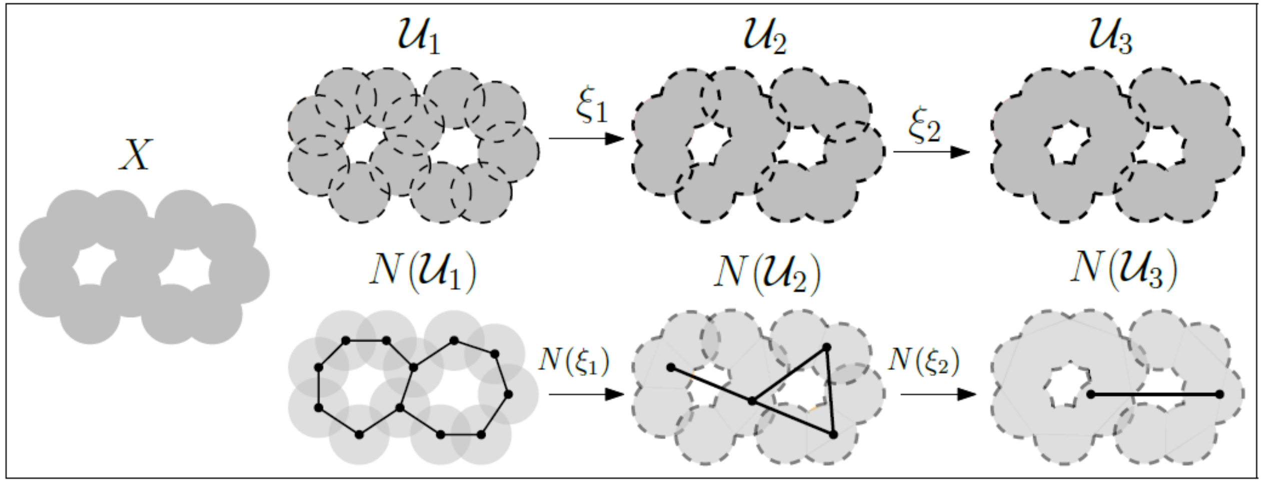
$N(\xi): N(\mathscr{U}) \to N(\mathscr{V})$ $\boxed{\text{if } \mathcal{U} \xrightarrow{\xi_1} \mathcal{V} \xrightarrow{\xi_2} \mathcal{W}, \text{ then } N(\xi_2 \circ \xi_1) = N(\xi_2) \circ N(\xi_1)}$

Maps between covers

- Given two covers $\mathscr{U} = \{U_\alpha\}_{\alpha\in A}$ and $\mathscr{V} = \left\{V_\beta\right\}_{\beta\in B}$ of the same space Y,
  - a cover map is any set map $\xi : A \to B$ such that $U_\alpha \subseteq V_{\xi(\alpha)}$ for all $\alpha \in A$
  - Intuitively, a cover map can connect covers at different resolutions (B is coarser than A)
- A cover map $\xi : A \to B$ induces a simplicial map in nerves:

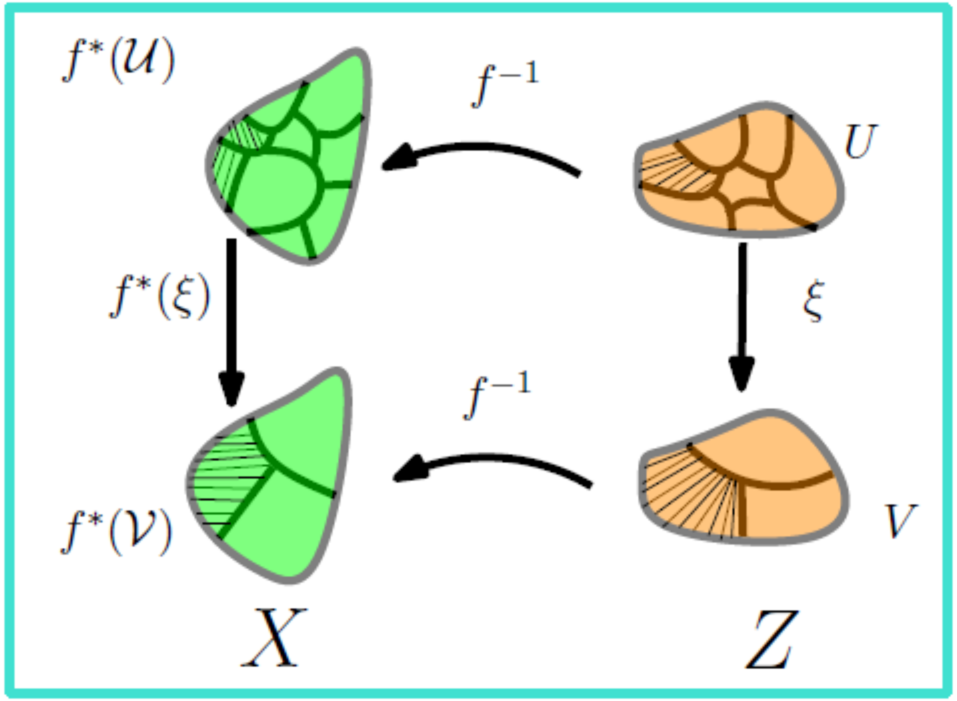
$$N(\xi) : N(\mathscr{U}) \to N(\mathscr{V})$$

$\boxed{\text{if } \mathcal{U} \xrightarrow{\xi_1} \mathcal{V} \xrightarrow{\xi_2} \mathcal{W}, \text{ then } N(\xi_2 \circ \xi_1) = N(\xi_2) \circ N(\xi_1)}$

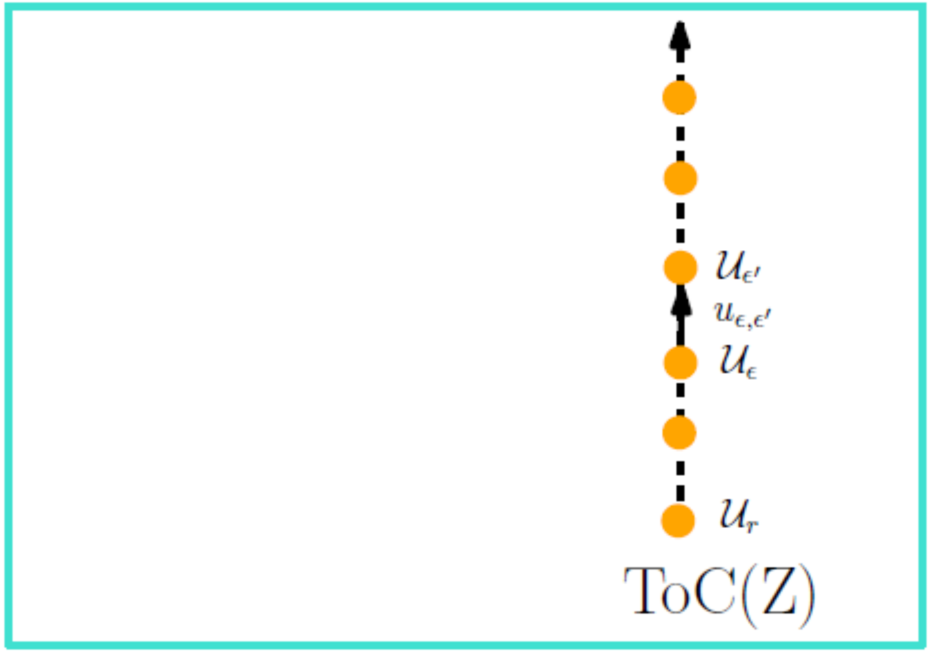


Pullback covers

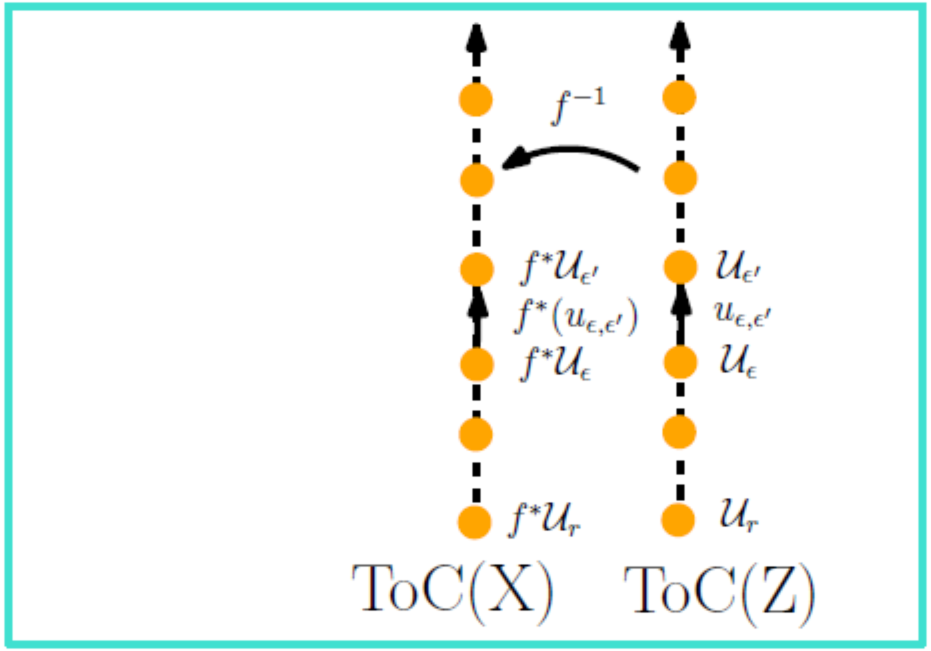
- $f: X \to Z$ continuous, and well-behaved
- A map $\xi: \mathscr{U} \to \mathscr{V}$ between covers of Z
- $\Rightarrow$ a cover map for pullback covers of X
  - $f^*(\xi): f^*(\mathscr{U}) \to f^*(\mathscr{V})$



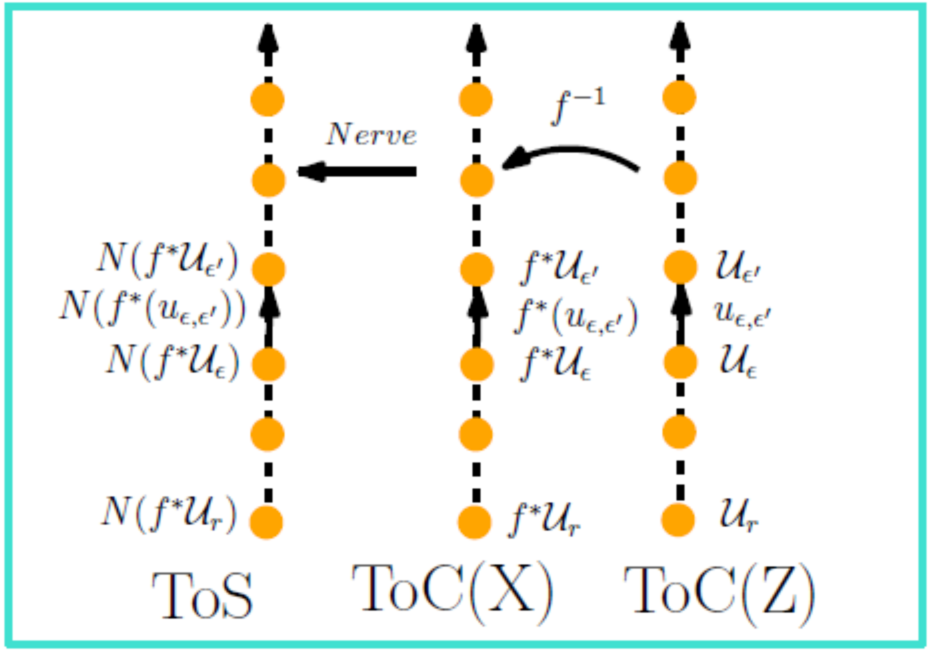
Multiscale Mapper



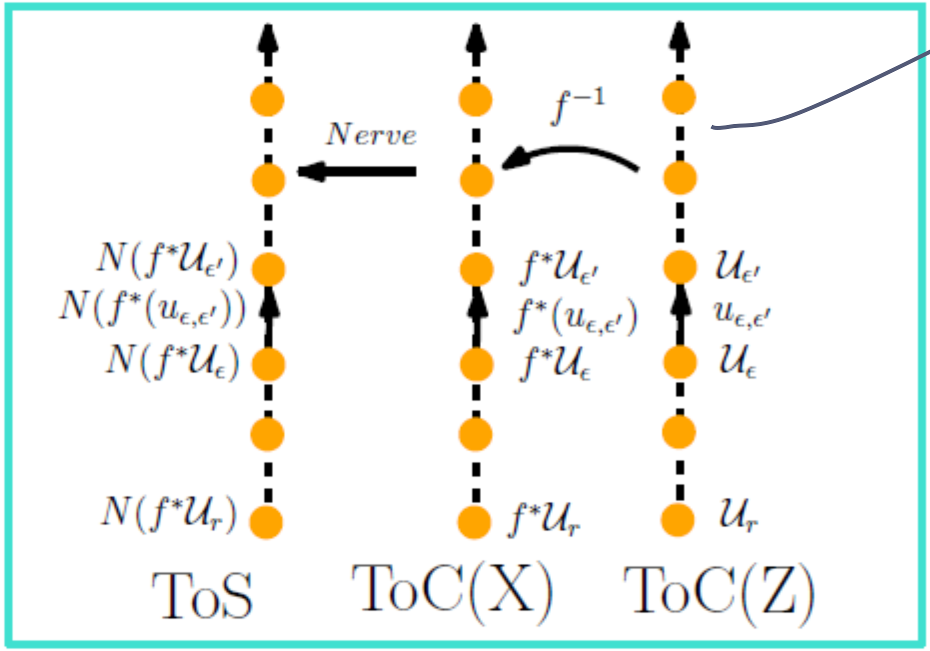
Multiscale Mapper



Multiscale Mapper

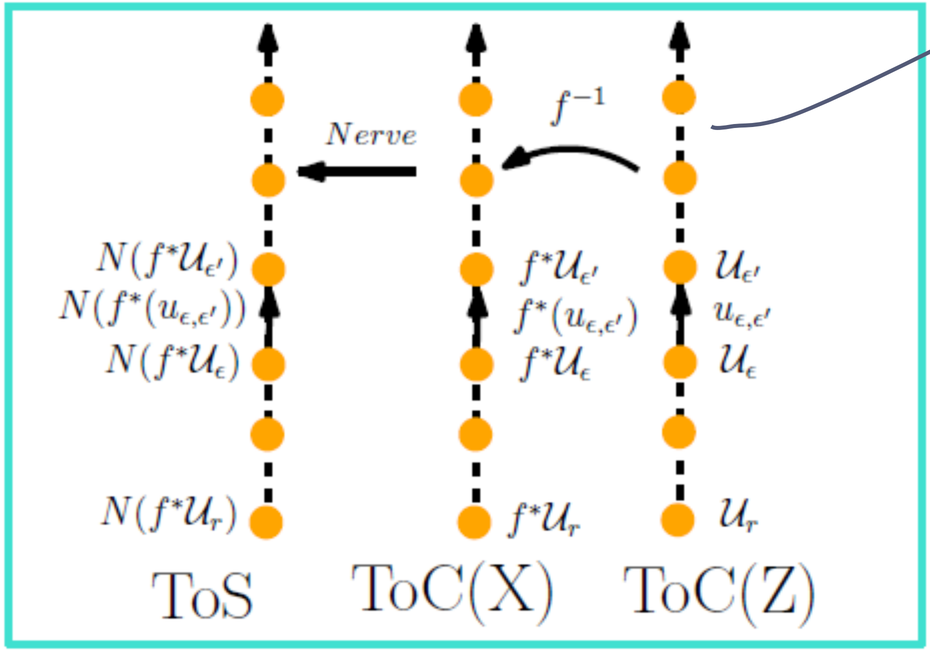


Multiscale Mapper



a tower of covers
$\mathfrak{U} = \{\mathscr{U}_\varepsilon\}_\varepsilon$

Multiscale Mapper

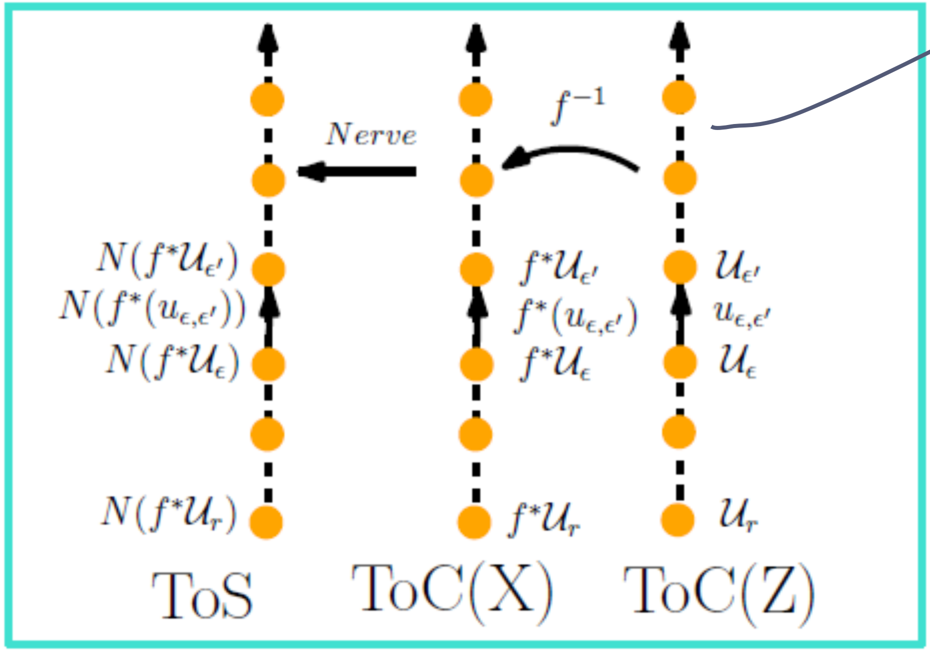


a tower of covers
$\mathfrak{A} = \{\mathscr{U}_\varepsilon\}_\varepsilon$

Multiscale Mapper:

$$MM(\mathfrak{A},\ f) := N(f^*(\mathfrak{A}))$$

Multiscale Mapper



a tower of covers
$\mathfrak{U} = \{\mathscr{U}_\varepsilon\}_\varepsilon$

Multiscale Mapper:

$$MM(\mathfrak{U}, f) := N(f^*(\mathfrak{U}))$$

$\mathrm{D_k MM}(\mathfrak{U}, f)$= persistence diagram of:

$$\mathrm{H}_k(N(f^*(\mathcal{U}_{\varepsilon_1}))) \to \mathrm{H}_k(N(f^*(\mathcal{U}_{\varepsilon_2}))) \to \cdots \to \mathrm{H}_k(N(f^*(\mathcal{U}_{\varepsilon_n})))$$

Remarks

- Intuitively, persistent homology of multiscale mapper captures important features of X through the lens of the map f and a tower of covers (across resolutions) of co-domain Z

Remarks

- Intuitively, persistent homology of multiscale mapper captures important features of X through the lens of the map f and a tower of covers (across resolutions) of co-domain Z

- For H_1, turns out that we cannot create new features, only kill existing features at the highest resolution gradually
  - time a feature is killed (ie., its persistence) corresponds to its importance

Remarks

- Intuitively, persistent homology of multiscale mapper captures important features of X through the lens of the map f and a tower of covers (across resolutions) of co-domain Z
- For H_1, turns out that we cannot create new features, only kill existing features at the highest resolution gradually
  - time a feature is killed (ie., its persistence) corresponds to its importance
- The multiscale mapper and its PH summaries have several stability properties w.r.t. perturbation of functions and tower of covers.

Remarks

- Intuitively, persistent homology of multiscale mapper captures important features of X through the lens of the map f and a tower of covers (across resolutions) of co-domain Z

- For H_1, turns out that we cannot create new features, only kill existing features at the highest resolution gradually
  - time a feature is killed (ie., its persistence) corresponds to its importance

- The multiscale mapper and its PH summaries have several stability properties w.r.t. perturbation of functions and tower of covers.

- Finally, there is an interleaving distance between multiscale mappers, much like the one for persistent homology.

FIN