Introduction to Zigzag Persistence

CSE 5339 Lecture 4

February 2, 2022

Persistence Modules

- Standard persistence module/vector space:

$$V_1 \longrightarrow V_2 \longrightarrow \ldots \longrightarrow V_{n-1} \longrightarrow V_n$$

Definition

A *zigzag* persistence module is one of the form:

$$V_1 \overset{f_1}{\longleftrightarrow} V_2 \overset{f_2}{\longleftrightarrow} \ldots \overset{f_{n-2}}{\longleftrightarrow} V_{n-1} \overset{f_{n-1}}{\longleftrightarrow} V_n$$

Persistence Modules

- Standard persistence module/vector space:

$$V_1 \longrightarrow V_2 \longrightarrow \dots \longrightarrow V_{n-1} \longrightarrow V_n$$

Definition

A *zigzag* persistence module is one of the form:

$$V_1 \overset{f_1}{\longleftrightarrow} V_2 \overset{f_2}{\longleftrightarrow} \dots \overset{f_{n-2}}{\longleftrightarrow} V_{n-1} \overset{f_{n-1}}{\longleftrightarrow} V_n$$

- The *type* of a zigzag persistence module, usually denoted τ, is the collection of arrow directions:
  - Ex:

$$M = V_1 \xleftarrow{f_1} V_2 \xrightarrow{f_2} V_3 \xrightarrow{f_3} V_4 \xleftarrow{f_4} V_5$$

$$\tau = (\leftarrow, \rightarrow, \rightarrow, \leftarrow)$$

Gabriel's Theorem

Theorem

Gabriel's Theorem implies a persistence module over the quiver:

$$\star \longleftrightarrow \star \longleftrightarrow \ldots \longleftrightarrow \star$$

decomposes into a direct sum of interval modules, regardless of directions of arrows!

Gabriel's Theorem

Theorem

Gabriel's Theorem implies a persistence module over the quiver:

$$\star \longleftrightarrow \star \longleftrightarrow \ldots \longleftrightarrow \star$$

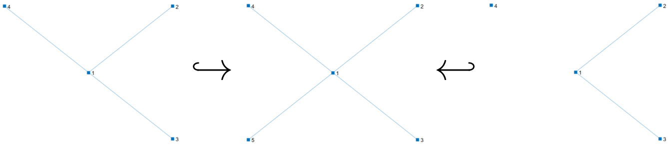
decomposes into a direct sum of interval modules, regardless of directions of arrows! Thus, for M a zigzag persistence module, we still have:

$$M = \bigoplus_{I \in \mathcal{I}} k^I$$

$\implies$ *we still have a persistence barcode/diagram for zigzag modules.*

Motivation for Zigzag Persistence

- In standard persistence, simplicial complexes grow over the filtration, with simplicial maps being inclusions.
- Zigzag filtrations allow maps in the opposite direction, so simplices can be deleted. We still want maps to be inclusions of simplicial complexes:



Motivation for Zigzag Persistence

Zigzag filtrations arise from natural, real-world data:

- Dynamic Graphs:
  - Connectivity in social networks
  - Interactions between regions of the brain

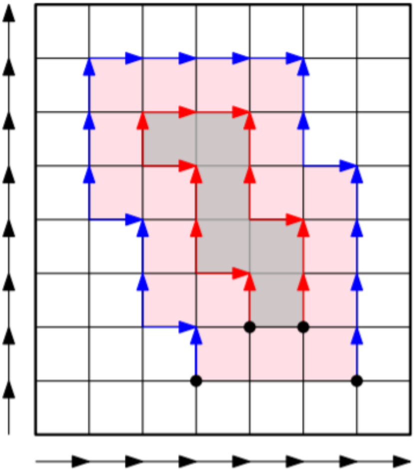
Motivation for Zigzag Persistence

Zigzag filtrations arise from natural, real-world data:

- Dynamic Graphs:
  - Connectivity in social networks
  - Interactions between regions of the brain
- Dynamic metric spaces:
  - Motion/flocking behavior of animals
- "Forgetting" older simplices: Next week!

Motivation for Zigzag Persistence

- Zigzag persistence modules can also be obtained by slicing multiparameter persistence modules!:



Theoretical Considerations

- Potential issue: data such as dynamic graphs can induce a sequence of simplicial complexes $K_1, K_2, \ldots, K_n$ without inclusions $K_i \hookrightarrow K_{i+1}$ or $K_i \hookleftarrow K_{i+1}$.
- This is dealt with by Carlsson and De Silva, [Carlsson and De Silva, 2010], via taking intersection or union sequences:

$$K_1 \longrightarrow K_1 \cup K_2 \longleftarrow K_2 \longrightarrow \ldots \longleftarrow K_{n-1} \longrightarrow K_{n-1} \cup K_n \longleftarrow K_n$$

$$K_1 \longleftarrow K_1 \cap K_2 \longrightarrow K_2 \longleftarrow \ldots \longrightarrow K_{n-1} \longleftarrow K_{n-1} \cap K_n \longrightarrow K_n$$

Theoretical Considerations

Theorem (Strong Diamond Principle, [Carlsson and De Silva, 2010])

There is a bijection between persistence diagrams found by taking homology of the intersection and union sequences. The bijection matches intervals with other intervals, differing by at most 1 in either the birth or death coordinate.

$$
\begin{tikzcd}
 & K_1 \cup K_2 & & K_2 \cup K_3 & & \dots \\
K_1 \arrow[ur] & & K_2 \arrow[ul] \arrow[ur] & & K_3 \arrow[ul] \arrow[ur] & \dots \\
 & K_1 \cap K_2 \arrow[ul] \arrow[ur] & & K_2 \cap K_3 \arrow[ul] \arrow[ur] & & \dots \arrow[ul]
\end{tikzcd}
$$

Dynamic Graphs

Zigzag persistence is frequently applied to dynamic graphs:

Definition

A *dynamic graph* consists of $G = \{G_t\}_{t \in T}$, where each $G_t = (V_t, E_t)$.

Examples:

Dynamic Graphs

Zigzag persistence is frequently applied to dynamic graphs:

Definition

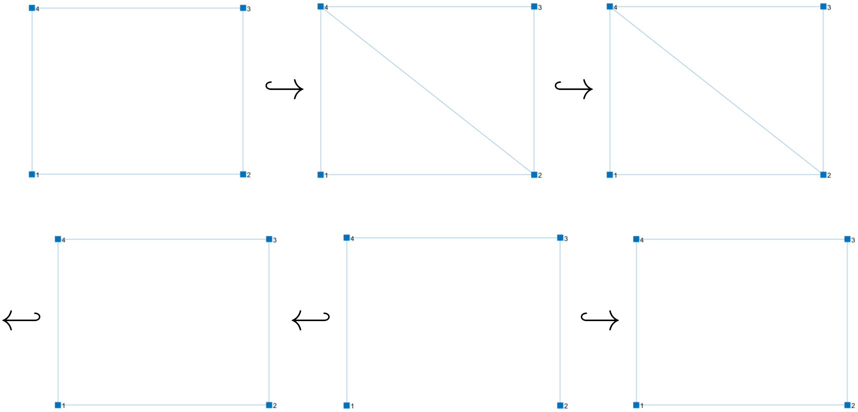
A *dynamic graph* consists of $G = \{G_t\}_{t \in T}$, where each $G_t = (V_t, E_t)$.

Examples:

- Social networks: people (vertices) come and go, relationships (edges) are formed and disappear.
- Motion of animals: animals (vertices) move closer together/further apart, creating/removing edges between them.
- Brain functionality: regions of brain (vertices) have interactions (edges).

Example

Let's compute 1-dimensional zigzag persistence for the following dynamic graph:



Software Implementations

- There are many freely available software packages for 1D persistence: `https://www.math.colostate.edu/~adams/advising/appliedTopologySoftware/`.
- Very few of these implement zigzag persistence.
- Recommend: Dionysus 2 by D. Morozov `https://mrzv.org/software/dionysus2/`.

Software Implementations

- There are many freely available software packages for 1D persistence: `https://www.math.colostate.edu/~adams/advising/appliedTopologySoftware/`.
- Very few of these implement zigzag persistence.
- Recommend: Dionysus 2 by D. Morozov `https://mrzv.org/software/dionysus2/`.
- Runs in C++, with bindings in Python.
- Natively supported on Linux, can be built on other OS.
- Is not GUI-based, requires some coding knowledge.

Dynamic Graph Example

- Data generated by simulated motion of birds using the "boids" (bird-oid) model. Model has parameters for separation, alignment, and cohesion. `https://en.wikipedia.org/wiki/Boids`.
- By varying parameters, 5 clearly distinct behaviors of motion were generated with 200 example dynamic point clouds for each behavior. Each dynamic point cloud lasts 1000 timesteps. Live examples:

Dynamic Graph Example

- Data generated by simulated motion of birds using the "boids" (bird-oid) model. Model has parameters for separation, alignment, and cohesion. `https://en.wikipedia.org/wiki/Boids`.
- By varying parameters, 5 clearly distinct behaviors of motion were generated with 200 example dynamic point clouds for each behavior. Each dynamic point cloud lasts 1000 timesteps. Live examples:
- Dynamic graphs are created by adding an edge between vertices at time t, if the vertices were within 10 units in distance at time t.
- Dionysus 2 used to compute zigzag persistent homology of resulting dynamic graphs, and then compare zigzag persistence modules via bottleneck distance:

Boids Example

Resulting clustering dendrogram:

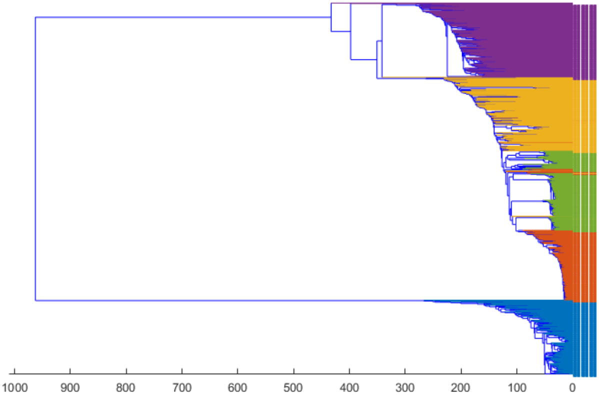


Figure: Clustering of Five Behaviors

References

Carlsson, G. and De Silva, V. (2010).
Zigzag persistence.
Foundations of computational mathematics, 10(4):367–405.