Feb 9

CS-6170 Computational Topology Lecture 10

$\boxed{\text{Last class}}$ Persistent homology of Complex networks

→ Input data was networks → abstract / discrete graphs

→ Construct Simplicial Complex as:

ⓐ Neighborhood Complex $N(G)$ ⓑ Clique Complex $C(G)$

→ Defining filtration: SC's indexed by rank in filtration sequence i.e. filtration parameter ~ index (discrete)

→ Compute homology groups for each SC in filtration

→ Persistence of features through filtration ⇒ Persistent homology

→ Visualization in form of barcodes ⇒ length of bar = persistence

---

\# In case of point cloud data (PCD): points sampled from continuous topological space

→ Video on persistent homology shows how we can construct an SC to represent the underlying space: Rips Complex

→ Consider union of open balls of diameter "d": Underlying Space

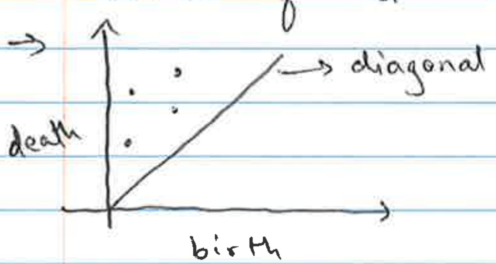
→ Each point in PCD is a vertex. Join two vertices if balls intersect } Rips Complex

3-way intersection: triangle, 4-way: tetrahedron etc.

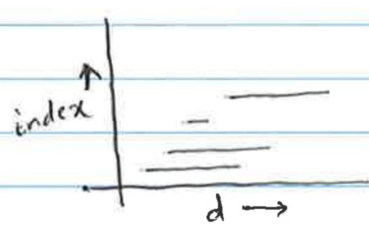
→ filtration: Sequence of SC's as "d" increases each SC in sequence corresponds to a specific "d" we have continuous filtration parameter.

→ Persistent homology: track value of "d" at which a homological feature appears (birth) and the value of "d" at which it disappears (death)

→



persistence diagram



barcode

each bar in barcode corresponds to a point in persistence diagram both are equivalent representations.

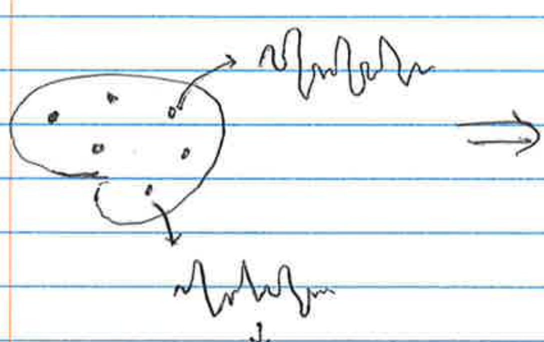
Kernel Partial Least Squares Regression for relating Functional Brain Network Topology to Clinical Measures of Behavior.

(#) Clinical Measures of behavior: data collected for autistic and control subjects. Diagnosis of autism is based on ADOS: Autism Diagnostic Observation Schedule

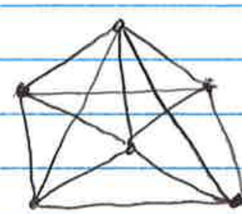
→ Behavior of subject scored from observation: Subjective
→ We want to find out how it relates to brain function.

(#) Brain Network: extracted from resting state fMRI
→ 264 regions in brain (Power regions)
→ time series of activity for each region - BOLD signal

→ We construct a network with 264 nodes (one for each region) functional association bet^n two regions is computed as Pearson correlation bet^n the corresponding time series
→ edge weight in the network = correlation bet^n nodes.



time series at node represents activity in the region



edge weights = correlation bet^n time series corresponding to the two regions.

→ Edge weights (correlations) carry important information we want to use this information

→ map the network to metric space using $d(x,y) = \sqrt{1 - \operatorname{cor}(x,y)}$
→ gives us point cloud with a distance matrix containing information of pairwise distances bet^n points.

$$\text{rs-fMRI} \xrightarrow[\text{lots of preprocessing}]{} \text{Brain Network} \xrightarrow[d = \sqrt{1-\text{corr}}]{} \text{PCD} \xrightarrow[\text{filtration } d = 0 \text{ to } \sqrt{2}]{} \text{persistent diagram}$$

→ 87 subjects give us 87 persistent diagrams

We restric ourselves to dim 0 (connected components) and dim 1 (loops) persistent homology.

→ How to use this in regression? Space of persistence diagrams is not Euclidean.

→ Need to define inner product. For two PDs A, B

$$K(A,B) = \frac{1}{8\pi\sigma} \sum_{\substack{p \in A \\ q \in B}} e^{-\frac{\|p-q\|^2}{8\sigma}} - e^{-\frac{\|p-\bar{q}\|^2}{8\sigma}}$$

σ → bandwidth parameter. p, q are points in persistent diagrams. if $q = (x,y)$ $\bar{q} = (y,x)$ → reflect across diagonal.

→ We compute K_0^{TDA}, K_1^{TDA} for dim 0, dim 1 points separately.

→ linear combination of kernels is also a kernel.

→ Run regression using kernelized version of PLS algorithm. with different kernels:

K^{TDA} : $w_0 K_0^{TDA} + (1-w_0) K_1^{TDA}$

K^{cor} : linear kernel on vectorized correlation matrix

$K^{TDA+cor}$: ~~K^{TDA}~~ $w_0 K^{corr} + w_1 K_0^{TDA} + (1-w_0-w_1) K_1^{TDA}$

→ Permutation test on regression predictions shows that combining TDA features with correlations gives the best predictions.

Persistent Cohomology, Circular co-ordinates, Circular features in high dimensional data

→ Consider high dimensional point cloud data.

→ Dimensionality reduction algorithms attempt to find a low dimensional embedding that preserves the intrinsic structure of the data.

→ Underlying assumption is that the domain is convex i.e. there are no "holes" in the space from which data is sampled.

→ What to do when there are holes? eg. when data is sampled from a torus

→ A circle is essentially a 1-D object. Given center and radius, each point on circle can be represented by single value θ → angle with +ve x direction.

→ In standard co-ordinate systems, require 2 coordinates for points on circle.

\# Dimensionality reduction $\Rightarrow \phi : \mathbb{R}^m \rightarrow \mathbb{R}^n$, $m > n$

we can think of ϕ as a set of n real valued functions $f_1, f_2, \ldots f_n : \mathbb{R}^m \rightarrow R$ such that

for $x \in \mathbb{R}^m$: $\phi(x) = (f_1(x), f_2(x), \ldots f_n(x)) = y \in \mathbb{R}^n$

→ f_i are the coordinate functions. eg. for points on a circle, $(\cos\theta, \sin\theta)$ are the co-ordinate functions

\# Main idea is to extend this class of co-ordinate functions to circle valued functions

i.e. $\theta : M \rightarrow S^1$

→ These can be computed using <u>1-dimensional Cohomology group</u>

Circular Coordinates: Given a fixed radius 'r', θ can be thought of as a parameterization of the circle.

→ as θ goes from 0 to 2π, it traces the circle.

→ Co-homology tries to assign a parameterization to high dim. point cloud data s.t. as the parameter changes, it traces the tunnel boundary → Circle valued co-ordinate.

Homology and Cohomology are related concepts.

→ in homology we have chain groups. C_k which is the collection of ~~xxx~~ k-simplices with (addition modulo 2) operation.

→ in Cohomology we have Co-chain groups C^k which is a collection of ~~xxx~~ homomorphisms $\varphi: C_k \to \mathbb{Z}_2$

i.e. ~~k-Coha~~ k-cochain group C^k consists of functions that map k-chains to $\{0,1\}$. The functions are homomorphisms $\therefore$ under the group operation of (+ modulo 2) the set of such functions forms a group.

Let $c \in C_k$ be a k-chain, $\phi, \varphi \in C^k$ be two k-cochains

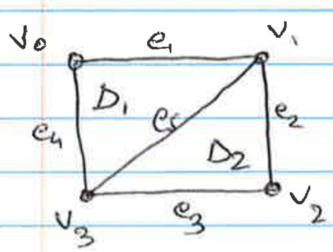
$$(\phi + \varphi)(c) = \phi(c) + \varphi(c) \in \{0,1\}$$

thus $(\phi+\varphi) \in C^k$ is also a k-cochain.

→ 0-cochains are functions that map 0-chains (vertices) to $\{0,1\}$

→ 1-Cochains are functions that map 1-chains (edges) to $\{0,1\}$

→ 2-Cochains are functions that map 2-chains (triangles) to $\{0,1\}$.



$K = \{v_0, v_1, v_2, v_3, e_1, e_2, e_3, e_4, e_5, \Delta_1, \Delta_2\}$

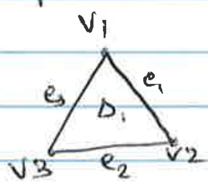
0-cochain : β eg. $\beta(v_0) = 1$ $\beta(v_i) = 0$ $i \neq 0$

1-cochain : α eg. $\alpha(e_1) = 1$, $\alpha(e_i) = 0$ $i \neq 0$

α, β are functions that map edges / vertices to 0 or 1.

\# Boundary operator in homology $\partial : C_k \to C_{k-1}$

maps a k-chain to (k-1)-chain (sum of k-simplices to the sum of their (k-1)-faces)

eg. 

$\partial(\Delta_1) = e_1 + e_2 + e_3$,

[Boundary of a boundary is always 0]

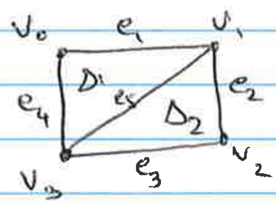
$\partial(\partial \Delta_1) = \partial(e_1 + e_2 + e_3)$
$= \partial e_1 + \partial e_2 + \partial e_3$
$0 = (v_1 + v_2) + (v_2 + v_3) + (v_1 + v_3)$

\# Co-boundary operator in Cohomology $\delta : C_k \to C_{k+1}$

→ maps a k-cochain (function on a k-chain) to a (k+1)-cochain which is a function on the (k+1)-co-faces of the k-chain

→ Co-face of a simplex (σ): all simplices that have σ as a face.

k-chain is a sum of k-simplices. (k+1)-coface is the sum of the (k+1)-co faces of the simplices in k-chain.

eg. 

Consider 1-chain $e_1 + e_2$
the 2-cofaces of $e_1 \Rightarrow \Delta_1$
—— " —— $e_2 \Rightarrow \Delta_2$

if α is a 1-cochain that maps $(e_1 + e_2)$ to 1
i.e. $\alpha(e_1 + e_2) = 1$, $\alpha(c) = 0$ for any other 1-chain
then $\delta\alpha$ is a function $\gamma \in C^2$ (i.e. a 2-cochain)

that evaluates the sum of 2-cofaces of $(e_1 + e_2)$ to 1
i.e. $\gamma(\Delta_1 + \Delta_2) = 1$ $\gamma(d) = 0$ for any other 2-cochain d.

→ Let β be a function on v_1 i.e. $\beta(v_1) = 1$, $\beta(v_i) = 0|_{i \neq 0}$

$\delta\beta$ evaluates 1-cofaces of v_1 to 1
i.e. $\delta\beta(e_1 + e_2 + e_5) = 1$, $\delta\beta(c) = 0$ for all other 1-chains.

$\delta(\delta\beta)$ would evaluate 2-cofaces of the 1-chain to 1, all other 2-chains to 0 i.e. $\delta(\delta\beta)(\Delta_1 + \Delta_2 + (\Delta_1 + \Delta_2)) = \delta(\delta\beta)(0)$

i.e., $\delta(\delta\beta)$ is a function that maps none of the 2-chains to 1 (it is a zero function)

$\Rightarrow$ Coboundary of a co-boundary is always a zero function.

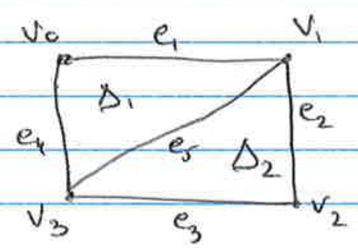
** We can now define co-cycle and co-boundary groups.

k-cocycle: Z^k: Set of all k-cochains (functions on k-chains) that are mapped to zero by coboundary operator

function that evaluates to 0 on all (k+1)-chains

k-Coboundary: B^k: a k-cochain is a k-boundary if there exists a (k-1)-cochain c s.t. $\delta(c)$ is the k-cochain.

e.g.



elementary 0-cochains: $v_0^*, v_1^*, v_2^*, v_3^*$

elementary 1-cochains: $e_1^*, e_2^*, e_3^*, e_4^*, e_5^*$

elementary 2-cochains: Δ_1^*, Δ_2^*

Think of these as indicator functions that only evaluate to 1 on the corresponding simplices

e.g. $v_0^*(v_0) = 1$, $v_0^*(v_i) = 0$ if $i \neq 0$.

all k-cochains can be represented as sum of elementary k-cochains.

$\Delta_1^* = \delta(e_1^*)$ $\therefore$ Δ_1^* is a 2-coboundary.

$\delta(e_5^*) = \Delta_1^* + \Delta_2^*$ $\therefore$ $\Delta_1^* + \Delta_2^*$ is also 2-coboundary.

$\delta(e_1^* + e_3^* + e_5^*) = \Delta_1^* + \Delta_2^* + (\Delta_1^* + \Delta_2^*) = 0$ $\therefore$ $(e_1^* + e_3^* + e_5^*)$ is 1-cocycle.

Since there is no combination of elementary 0-cochains (v_i^*) such that $\delta(\sum g_i v_i^*) = (e_1^* + e_3^* + e_5^*)$

$(e_1^* + e_3^* + e_5^*)$ is not a co-boundary but it is a co-cycle.

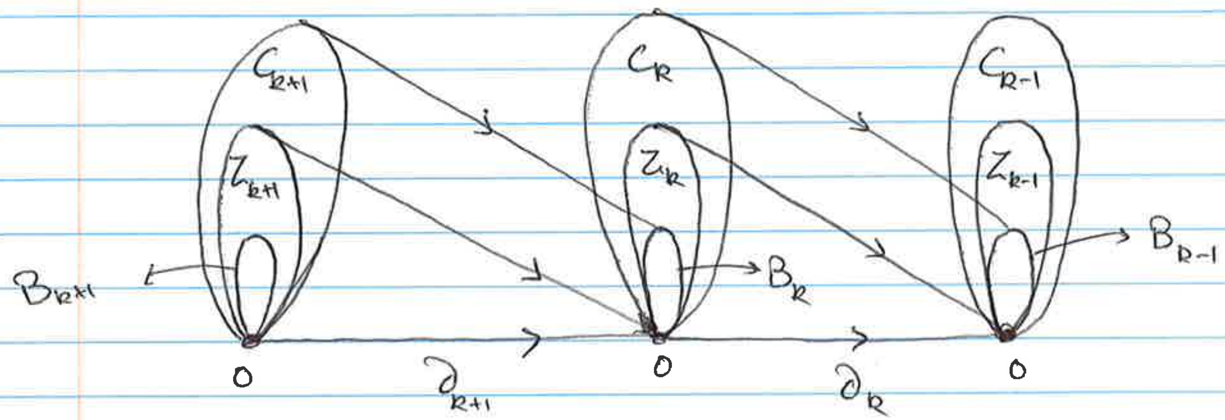
$$Z^k = \ker \delta^k : C^k \to C^{k+1}$$

$$B^k = \operatorname{im} \delta^{k-1} : C^{k-1} \to C^k$$

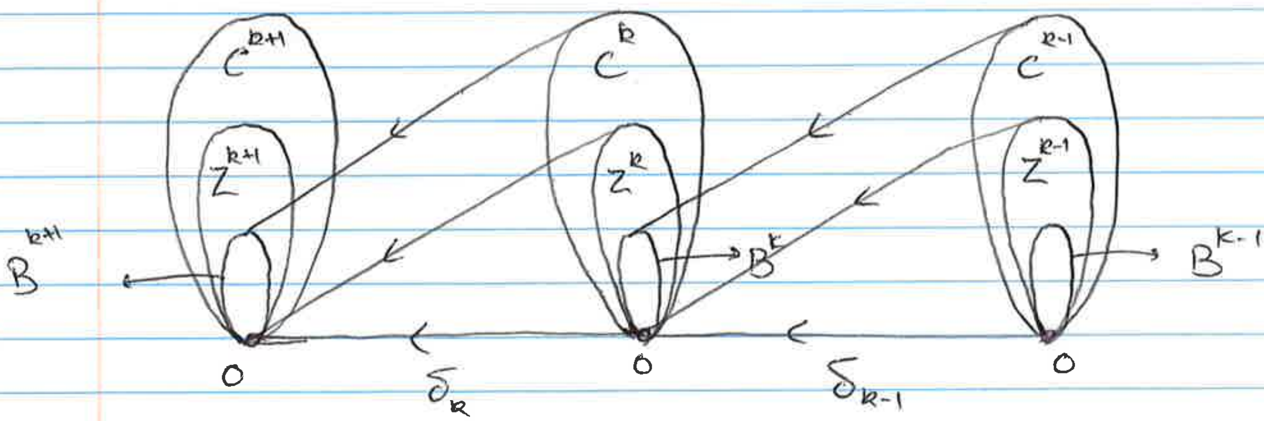
(*) The k-th cohomology group is the quotient of k-th cocycle group modulo the k-th coboundary group

$$H^k = Z^k / B^k \quad \text{for all } k.$$

→ Co-cycles that are not co-boundaries.



Homology



Cohomology