2022 – TDA+Neuroscience course

tripodsx@lists.osu.edu

June 9, 2022

1 Pre-requisites

- **Linear Algebra:**
  - Vector spaces, basis of a vector space, span of a set of elements. Sub-vector space. Direct sum.
  - Linear maps between vector spaces.
  - Dimension of a vector space.
  - Images and Kernels of linear maps.
  - Matrices, and matrix manipulations.

  * See these Khan Academy lessons: https://www.khanacademy.org/math/linear-algebra
- **Coding:**
  - Basic familiarity with Matlab.
  - Foundational concepts such as if/else statements and for loops.

  * Here's a basic Matlab tutorial https://web.eecs.umich.edu/~aey/eecs451/matlab.pdf
- **Other concepts:**
  - Definition of a metric space. Basic examples, familiarity with these concepts.
  - Inner products and norms.

  * See this lecture: http://www-history.mcs.st-and.ac.uk/~john/MT4522/Lectures/L5.html

1.1 Other beneficial topics

- Normal forms and Gaussian elimination: See https://en.wikipedia.org/wiki/Smith_normal_form and https://en.wikipedia.org/wiki/Gaussian_elimination.
- Basics of graph theory (definition of a graph, connected components): See https://en.wikipedia.org/wiki/Graph_theory and https://en.wikipedia.org/wiki/Component_(graph_theory).
- Basic concepts about clustering: See https://en.wikipedia.org/wiki/Cluster_analysis.

2 Required software

We will be requiring Matlab. We are working to secure Matlab licenses for all participating students. JavaPlex must be installed and students should start running the tutorial before the first day of class. The main interface for JavaPlex is Matlab.[1]

- **JavaPlex Software**: `http://appliedtopology.github.io/javaplex/`
- **JavaPlex Tutorial:** `https://github.com/appliedtopology/javaplex/wiki/Tutorial` *Please start familiarizing yourself with the contents of the tutorial.*
- **Matlab:** `https://www.mathworks.com/products/matlab.html`
- **NeuMapper:** `https://braindynamicslab.github.io/neumapper/`

3 Course outline

- **Day/time:** TBD
- **First day of classes:** TBD
- **Duration:** 1hr.
- **Number of weeks:** TBD
- **Software component:** will be based on JavaPlex and Matlab.

Weekly Schedule

Week 1.

- Introduction Lecture: Overview of applied topology with motivation from neuroscience.
- Recitation: Linear algebra review. Must have the exercises completed by this point.

Week 2.

- Topology: Basics of topology. We will introduce graphs, metric spaces, and topological spaces as well as concepts such as continuous maps, homeomorphisms, and homotopy equivalences.
- Recitation: Metric space discussion, examples, exercises. Discuss topological connectivity.
- Project part 1: Single linkage hierarchical clustering.

Week 3.

- Simplicial Complexes: Simplicial complexes and simplicial maps. Rips and Čech filtrations. The nerve lemma.
- Recitation: Computational examples and non-example simplicial complexes. Algorithm for VR complex at a given scale parameter.

Week 4.

- Homology: Simplicial homology with $\mathbb{Z}/2\mathbb{Z}$ coefficients. Persistence modules.
- Recitation: Practice problems with simplicial homology, work through using boundary matrix reduction. Discuss impact of coefficients.
- Project part 1 due.

[1] See `https://github.com/appliedtopology/javaplex/wiki/Interoperability` for details about these other frontends.

Week 5.

- Persistent Homology: The persistence algorithm, barcodes/persistence diagrams, interpretations, and examples.
- Recitation: JavaPlex demo. Setup and demonstration. Work through parts of Javaplex tutorial.

Week 6.

- Preprocessing Data: Filtrations in arising in TDA and neuroscience. Subsampling and density thresholding.
- Invited Speaker: TBD
- Project part 2: Computing barcodes.

Week 7.

- Theory of Persistence: Interleavings, matchings, and the bottleneck stability theorem. Homology inference and the relationship between persistence diagrams obtained via Rips/Čech filtrations.
- Recitation: computational exercises comparing barcodes: bottleneck distance and Wasserstein distance.

Week 8.

- Extensions of Persistence: Multiparameter persistence, zigzag persistence, level set persistence and Reeb graphs.
- Recitation: Software Demo. RIVET and Dionysus 2.
- Project part 2 due.
- Project part 3: preprocessing data.

Week 9.

- Visualization: Multidimensional scaling and Mapper.
- Invited Speaker: TBD
- Project part 3 due.
- Project part 4: Interpreting barcodes and using Neu Mapper to visualize data.

Week 10.

- Neuroscience Background: Data collection, spike trains, place cells etc.
- Invited Speaker: TBD

Week 11.

- Geometric Structure of the Brain: Clique topology reveals intrinsic geometric structure in neural correlations. Cliques and cavities in the human connectome.
- Invited Speaker: TBD
- Project part 4 due.

Week 12.

- Memory: The importance of forgetting. Flexible memory networks.
- Invited Speaker: TBD

4 Project

We will use a variety of techniques from topological data analysis to analyze the Non-Rigid World TOSCA database of 3D shapes. This data set consists of numerous types of figures (e.g. cats, dogs, horses, etc.) in different poses. The goal of the project is to classify each figure using topological data analysis. Students will work in groups of 3–4.

Part 1

Students will code their own single linkage hierarchical clustering algorithms and apply them to the data sets.

Part 2

Students will use JavaPlex to compute persistence diagrams. We will provide preprocessed data for this step.

Part 3

Students will use a variety of preprocessing techniques to clean data sets. These techniques include subsampling and the use of different filtrations such as witness complexes.

Part 4

Students will use the persistence diagrams to classify the data sets. We will also use NeuMapper to visualize the data.

5 Other useful software links

- Ripser `https://github.com/Ripser/ripser`
- Ripser Matlab `https://github.com/ctralie/Math412S2017`
- Ripser Python `https://github.com/scikit-tda/ripser.py`
- Ripser Live `https://live.ripser.org/`
- Sklearn `https://github.com/MathieuCarriere/sklearn-tda`
- Dionysus `https://www.mrzv.org/software/dionysus/`
- Eirene `https://github.com/Eetion/Eirene.jl`

See `https://www.math.colostate.edu/~adams/advising/appliedTopologySoftware/` for a more comprehensive list.