A Mathematical Introduction to Data Science	**Apr. 8, 2025**

Homework 8. Persistent Homology

Instructor: Yuan Yao	*Due: 2 weeks later*

The problem below marked by * is optional with bonus credits.

1. *Persistent Homology:* Download the Javaplex (latest version 4.3.4) from the following site

   http://appliedtopology.github.io/javaplex/

   Follow the tutorial at

   https://github.com/appliedtopology/javaplex/wiki/Tutorial

   to get it work under Matlab environment. For example, extract the zip file and open Matlab, change Matlab's "Current Folder" to the directory where the load_javaplex.m file is located (src/matlab/for_distribution/ in version 4.3.4), run the file:

   ```
   >> load_javaplex
   ```

   Also in the Matlab command window, type the following command.

   ```
   >> import edu.stanford.math.plex4.*;
   ```

   Installation is complete. Confirm that Javaplex is working properly with the following command.

   ```
   >> api.Plex4.createExplicitSimplexStream()
   ans = edu.stanford.math.plex4.streams.impl.ExplicitSimplexStream@16966ef
   ```

   Your output should be the same except for the last several characters. Each time upon starting a new Matlab session, you will need to run load_javaplex.m.

   Now conduct the following numerical experiment with the example shown in class:

   (a) Construct a filtration (stream) of the simplicial complex in the Figure 1 (a). This could be done by

   ```
   >> stream = api.Plex4.createExplicitSimplexStream();
   >> stream.addVertex(0,0);
   >> stream.addVertex(1,1);
   >> stream.addVertex(2,2);
   >> stream.addVertex(3,3);
   >> stream.addElement([0,2],4);
   ```
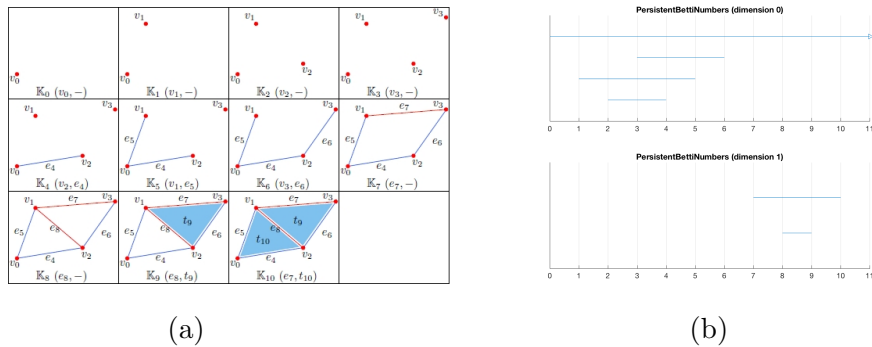
Figure 1: (a) A filtration of simplices; (b) barcodes of Persistent Betti Numbers.

```
>> stream.addElement([0,1],5);
>> stream.addElement([2,3],6);
>> stream.addElement([1,3],7);
>> stream.addElement([1,2],8);
>> stream.addElement([1,2,3],9);
>> stream.addElement([0,1,2],10);
>> stream.finalizeStream();
```

where you can check the number of simplices in the filtration (stream) is 11

```
>> num_simplices = stream.getSize()
```

(b) Compute the persistent homology for the filtration and plot the barcode as Figure 1 (b).

```
>> % Compute the Z/2Z persistence homology of dimension less than 3:
>> persistence = api.Plex4.getModularSimplicialAlgorithm(3, 2);
>> intervals = persistence.computeIntervals(stream);
>> options.filename = 'Persistent-Betti-Numbers';
>> options.max_filtration_value = 11;
>> % Plot the barcode of persistent Betti numbers:
>> plot_barcodes(intervals, options);
```

2. *Torus example:* In `javaplex`, construct a torus, the Vietoris-Rips complex, persistent homology, and plot the barcodes of persistent Betti numbers.

(a) Load a pre-stored torus embeded in 4-dimensional space with 400 samples

```
>> load tutorial_examples/pointsTorusGrid.mat
>> scatter3(pointsTorusGrid(:,1), pointsTorusGrid(:,2), pointsTorusGrid(:,3))
```

(b) Construct a Vietoris-Rips complex

```
>> max_dimension = 3;
>> max_filtration_value = 0.9;
>> num_divisions = 1000;
>> stream = api.Plex4.createVietorisRipsStream(pointsTorusGrid, ...
    max_dimension, max_filtration_value, num_divisions);
>> num_simplices = stream.getSize()
```

(c) Compute the persistent homology for the Vietoris-Rips complex

```
>> persistence = api.Plex4.getModularSimplicialAlgorithm(max_dimension, 2);
>> intervals = persistence.computeIntervals(stream);
```

(d) Plot the barcode for the persistent Betti numbers, like Fig. 2

```
>> options.filename = 'ripsTorus4.png';
>> options.max_filtration_value = max_filtration_value;
>> options.max_dimension = max_dimension - 1;
>> options.side_by_side = true;
>> plot_barcodes(intervals, options);
```

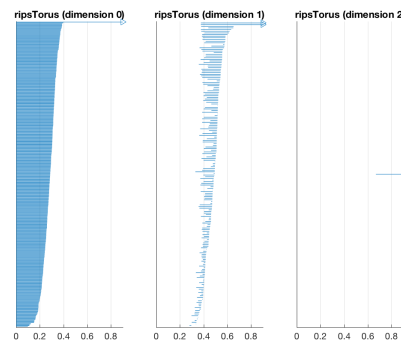(e) Explain the picture you find.

Figure 2: Barcodes of Persistent Betti Numbers of a Torus.

3. *Single Linkage Clustering and Persistent 0-Homology*: Hierarchical clustering algorithms performs the following algorithm to build up a cluster tree.

   (a) Start with a set of n data points as n initial clusters (leaves).

   (b) Repeatedly merge two "closest" clusters.

   There are a variety of "distance" (or "dissimilarity") notions between clusters:

   (1) **Single Linkage**: distances (dissimilarity) between two *closest* pair of points in two clusters;

   (2) **Complete Linkage**: distances (dissimilarity) between two *furthest* pair of points in two clusters;

   (3) **Complete Linkage**: average pairwise distances (dissimilarity) between two clusters.

   Show that

   (a) *0-dimensional Persistent Homology of n points in a metric space is equivalent to single linkage clustering without labeling, i.e. permutation invariant*;
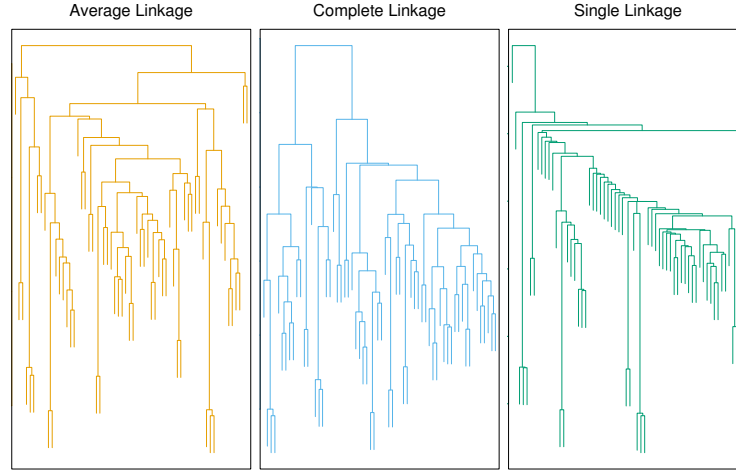
Figure 3: 10.12 (Introduction to Statistical Learning with Applications in R). Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.

(b) *For a continuous density function $p : \mathbb{R}^d \to \mathbb{R}_+$, let*

$$C_p(\lambda) := \{\,connected\ components\ of\ x \in \mathbb{R}^d : p(x) \geq \lambda\,\}$$

be the connected components of super-level sets of $p(x) \geq \lambda \geq 0$, then $C_p(\lambda) \subseteq C_p(\lambda')$ as $\lambda \geq \lambda'$ and $C_p(\lambda)$ induces a tree such that the vertex set at level λ consists of connected components and an edge connects connected components at adjacent levels if $\sigma \subseteq \tau$ for $\sigma \in C_p(\lambda)$ and $\tau \in C_p(\lambda')$.

(c) **Under what conditions, 0-dimensional Persistent Homology of n sample points in Euclidean space is consistent to recover connectivity of a probability density super-level set? (Hartigen, J. A. "Statistical theory in clustering", Journal of Classification, 2: 63-76, 1985)*

4. **Niyogi-Smale-Weinberger Consistency Theorem.* The following theorem establishes the homology consistency of Čech complex of discrete samples uniformly from a compact submanifold of $\mathbb{R}^d$.

Theorem. Let M be a compact submanifold of $\mathbb{R}^d$ with condition number τ and let $\{x_1, \ldots, x_n\}$ be a set of points drawn from M according to the volume measure. Fix $0 < \epsilon < \frac{\tau}{2}$. Then if

$$n > \beta_1 \left(\log \left(\beta_2 \right) + \log \left(\frac{1}{\delta} \right) \right),$$

there is a homotopy equivalence

$$\bigcup_{z \in \{x_1, \ldots, x_n\}} B_\epsilon(z) \simeq M$$

between the union of balls and M (and in particular the homology groups coincide) with probability $> 1 - \delta$. Here

$$\beta_1 = \frac{\mathrm{vol}(M)}{\cos^d(\theta_1)\,\mathrm{vol}\left(B_{\frac{\epsilon}{4}}^d\right)}$$

and

$$\beta_2 = \frac{\mathrm{vol}(M)}{\cos^d(\theta_2)\,\mathrm{vol}\left(B_{\frac{\epsilon}{8}}^d\right)},$$

where $\theta_1 = \arcsin\left(\frac{\epsilon}{8\tau}\right)$, $\theta_2 = \arcsin\left(\frac{\epsilon}{16\tau}\right)$, and $\mathrm{vol}\left(B_r^d\right)$ denotes the volume of the d-dimensional ball of radius r. *Can you read the paper and sketch the main idea for the proof? What about the noisy samples near the submanifold?*

Reference: Partha Niyogi, Stephen Smale, and Shmuel Weinberger. *Finding the homology of submanifolds with high confidence from random samples.* Discrete Comput. Geom. 39(1-3): 419-441, 2008.