TDA+Neuroscience course Project: Shape Analysis and classification via Persistent Homology, Final Part

April 18, 2023

This part of the project will be due by 5:00pm Eastern time on Friday, April 28th. Note that we have been flexible with due dates previously, but due to this being the end of the semester this will be a hard deadline, so be sure to work ahead as possible. We will have our last one-on-one meetings of the semester the week of April 24th. You do not need to have the project done by these meetings, but you should have made substantial progress by this point.

1 Visualizations

At this stage of the project, you should have bottleneck distance matrices corresponding to the bottleneck distance between 0th-dimensional, 1st-dimensional, and 2nd-dimensional persistence diagrams of all the shapes. We refer to these distance matrices as dB0, dB1, and dB2, respectively.

We already visualized the distance matrices with a heat-map to give us an idea of how well persistent homology has been able to discriminate between shapes in the tosca database, but at this stage of the project we want to use more refined visualization tools. You will be required to use at least two visualization tools at this stage. The first is a dendrogram, which we discussed at the beginning of the semester. You may use the implementation for dendrograms you wrote earlier in the semester, but also feel free to use the built-in `linkage` and `dendrogram` functions in Matlab: `https://www.mathworks.com/help/stats/linkage.html`, `https://www.mathworks.com/help/stats/dendrogram.html` .

Compute and plot the dendrograms associated to dB0, dB1, and dB2. Be sure to include all nodes of the dendrograms in the plots, and also create labels based on what type of shape each node corresponds to. For details on how to do this, look at the documentation in the url's above. There are approaches to define the labels in an automated way using the `dir` command in Matlab `https://www.mathworks.com/help/matlab/ref/dir.html` to extract all the filenames of the data files, and then taking only the first n letters of each filename as the labels.

For submission: Include plots of the dendrograms corresponding to dB0, dB1, and dB2. With clustering more explicit now, include a more detailed description of clustering than before. Which matrix resulted in the best clustering overall? Which classes were properly clustered the most often? Were there classes properly clustered with one distance matrix, but not another?

There are other visualization techniques below. Pick at least one of the bullet points below, and include the plots for the method(s) applied to each distance matrix and an explanation of the results, including how the resulting plots differ from the classical dendrograms already computed.

Other visualization techniques:

- The dendrograms computed above were based on single-linkage hierarchical clustering. There are many other clustering methods that the `linkage` function accepts, see the documentation linked above. Choose *two* or more and plot the resulting dendrograms, comparing them with the previous and each other.
- Another option is multi-dimensional scaling `https://en.wikipedia.org/wiki/Multidimensional_scaling`. This is a dimension reduction technique that aims to plot a metric space in lower dimensions

while distorting the distances as little as possible. There are built-in Matlab functions for this, including `https://www.mathworks.com/help/stats/cmdscale.html`. Be sure to use labels in some form, either explicitly or via setting all shapes from the same class as a single color.

- Use mapper in some way, as explained in the recitation video. There is a wide array of freedom here, and you may use any mapper implementation you find online, as well as any parameters for the mapper algorithm. If you follow this route, you need to write a justification for the choices you made, and explain how the mapper graph can be used to detect the different shape classes.

- If there is some other visualization technique you wish to use, feel free to ask Nate for approval. You will need to justify why said technique should give useful visualizations, but the above list is not exhaustive and we leave it open to you if there are other visualization techniques you wish to try.

For submission: Your final submission should include at least two of the visualization techniques as discussed above, and you should write a paragraph re-summarizing the persistent homology pipeline you performed over the course of the semester from start-to-finish. What are some potential differences or difficulties you could see when trying to apply this pipeline as-is on other datasets stemming from real-world data?

2 Optional Continuations

There are many ways in which this project could be extended. None of the following is required to do for submission as it pertains to this project. That said, if you are interested in pursuing TDA further and/or going into data analytics they are all good things to consider. Nate is happy to discuss any of these options with you the remainder of the semester and beyond if you wish to pursue them. Also, if you are planning on attending GTDAML `https://gtdaml.wixsite.com/2023` and wish to present something there, performing one or more of these continuations could help strengthen a presentation:

- Classification tasks: analyze quantitatively how the bottleneck distance performs at classification tasks, such as the k-nearest neighbor classification task: `https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm`

- Using a different filtration aside from Vietoris-Rips. You do not need to use Matlab for any of these continuations, but if you do so you can use any number of filtrations in JavaPlex via manually defining a filtration as we saw early on in the semester. Possible filtrations include sublevel-set filtrations, a witness complex filtration, or a Vietoris-Rips filtration but with the distance matrices "re-weighted" according to some other factor such as eccentricity.