CS665 Project Proposal

Zhe Wang

For the final project, I would like to implement the kernel described in this paper:

Zhu, Xiaojin, et al. "Stochastic Multiresolution Persistent Homology Kernel." International Joint Conference on Artificial Intelligence. 2016.

This paper introduced a kernel that captures the persistent homology of the data. It is very useful when the dataset is a set of point clouds and we care about the shape of different point clouds.

This project provide a novel way of using persistent homology, which is directly related to my current work: interactive topology data analysis. My work mainly focus on making persistent homology calculation fast enough for interactive data analysis. Besides providing an opportunity to explore what persistent homology can do in data analysis, this project could probably be integrated into a use case for my research. For example, if our dataset contains a categorical dimension, when user select a subset based on some other dimension, the system could calculate kernel PCA for different categories under current selection.

Expectations:

1. Implement the SMURPH kernel introduced in the paper

I intend to implement it in Python. It should take a set of point clouds as input and output a kernel.

2. Build a demo that use this kernel to calculate kernel PCA

This part will be a web based interactive visualization. User first select one of the provided datasets. Then the system will calculate the SMURPH kernel and use it to calculate the kernel PCA. The result of kernel PCA will be send back to client side. The web UI will plot the first two PCs.

Evaluation:

1. Calculate kernel PCA on synthetic dataset and labeled dataset

First, I'll generate a synthetic dataset. This dataset should contain several point clouds with different number of holes. If this kernel works well, we should see from the 2D PCA plot that the point clouds with the same amount of holes should form a cluster and point clouds with different number of holes should be far aways from each other.
I'll do the same test for the point cloud dataset used in the paper.

2. Compare with other kernels.

I'll use a naive linear kernel as baseline. The SMURPH kernel should perform better than the linear kernel.