CS 665 – Final Project Report

Zhe Wang

1 Introduction

The purpose of this project is to evaluate the Stochastic Multiresolution Persistent Homology Kernel (SMURPH) [**?**] This kernel tries to capture the persistence homology information of point cloud data.

In order to evaluation SMURPH kernel, I used it to calculate kernel PCA. If the kernel is capables of what the paper claims, we should see from the 2D kernel PCA that point clouds with similar persistence homology should form a cluster.

I conducted experiments on three different datasets. The first dataset is the Kitchen Utensil Dataset, which is the dataset used in the original paper. The main reasion to use this dataset is to validate my implementation. If my implementation is correct, then the kernel PCA result should be at least similar to the result presented in the paper. The other two datasets are synthetic datasets with different features. One has different number of holes and the other has different scale of holes. Using synthetic datasets makes it easier to analyze and explain the results.

For each dataset, I also compared the kernel with two other kernels. The first is a simple linear kernel. This kernel serves as a base line. Since a linear kernel shouldn't make any sense considering persistence homology of a point cloud, we should expect the kernel PCA result don't form any clusters. When comparing with it, we should be able to see if a kernel is capturing the persistence homology features. The second kernel is proposed by my self, trying to come up with a simple but meaning kernel. Basically this kernel calculate a histogram of distances (HOD) between each pair of points. Then this histogram is used as a feature vector to calculate the inner product.

Detailed descriptions about these kernel will be presented in the following section.

2 Tested Kernels

2.1 SMURPH Kernel

The detailed algorithm for calculating SMURPH kernel can be found in Algorithm 1 in [**?**]. Here I just summerize the main idea of it. Given a point cloud, SMURPH kernel generate multiple samples at different scale: $[s_0, s_1, s_2, ..., s_n]$. Then for each s_i, SMURPH build a Vietoris-Rips filtration on it and calculate the persistence diagram. Next, the persistence diagram is converted to persistence landscape (PL) function, which becomes the representation r_i for the sample s_i. At last, each point cloud is represented by an array of PL functions: $[r_0, r_1, r_2..., r_n]$. The inner product of two different point clouds becomes the inner product of two array of PL functions, which could be calculated by taking integrals.

2.2 Linear Kernel

A simple linear kernel is used as baseline for comparison. The linear kernel generate same-sized samples from given point clouds. Then the inner product is defined as the sum of inner products of points from each sample.

2.3 Histogram of Distances Kernel

In order to have a meaningful yet still simple kernel for comparison, I proposed a new kernel: Histogram of Distances (HOD). The kernel generate a histogram of distances between each pair of points in a point cloud. Then a histogram of the distances is calculated. I use the normalized histogram as the vector representation of the point cloud. So the inner product of two point clouds becomes the inner product of two vectors. The intuition of this kernel is that point clouds don't have any holes tend to have a smooth and dense histogram of distances, while point clouds have many holes don't.

3 Datasets

3.1 Kitchen Utensil Dataset

This dataset [**?**] consists of 41 point clouds, generated by 3D scanning of kitchen utensils. Figure **??** shows two samples of this dataset.

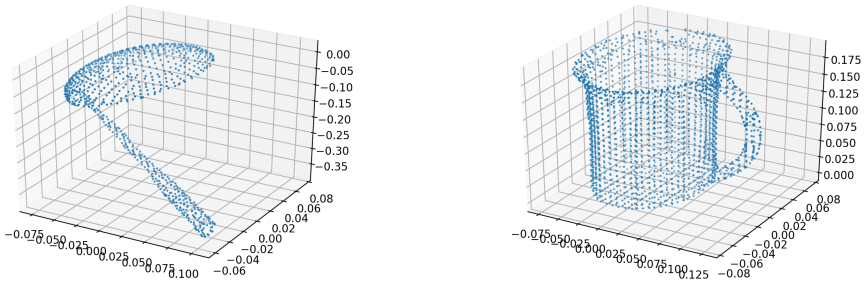


Figure 1: Samples from Kitchen Utensil Dataset

3.2 Synthetic Dataset – Multiple Holes

For synthetic dataset, I only generate 2D point clouds so that it's simple and fast to calculate, and also easy to understand.

The first synthetic dataset contains 2D point clouds with different number of holes. Figure **??** shows some samples from this dataset. Basically, this dataset starts from a disk-shaped point mesh. The distance between two nearest points is 1. Then I used different number of small holes (also have different size) to erode the disk. The number of holes are $0, 1, 2, 3$. The radius of holes are $2, 3, 4, 5$. There are totally 16 different point clouds in this dataset.

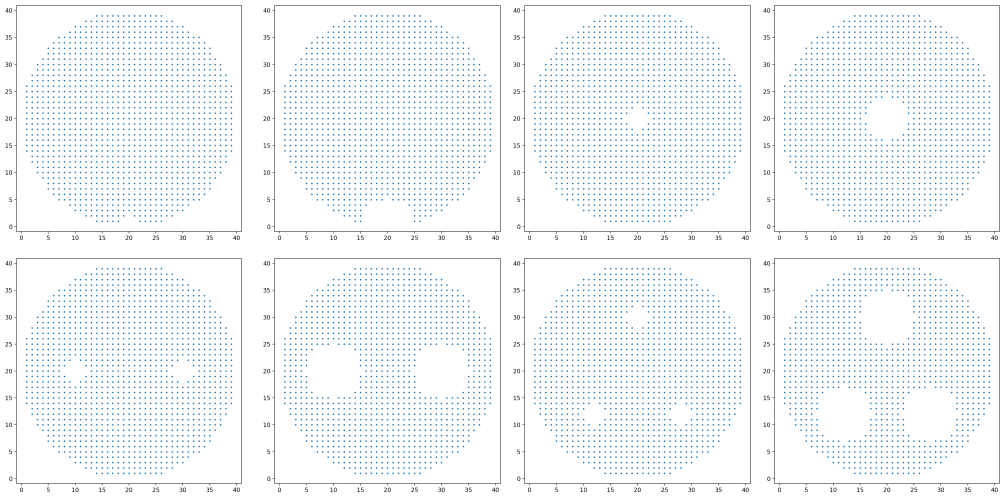


Figure 2: Synthetic Dataset: Different Number of Holes

3.3 Synthetic Dataset – Multiscale

The SMURPH kernel claims to be capable of doing multiresolution analysis. So I designed this dataset to evaluate this ability. Figure **??** shows a few samples. The dataset basically have two shapes of data: O-shaped and ∞-shaped. Each shape could be consist of solid ribbons, or ribbons with small holes. These four point clouds, as Figure **??** shows, form a set. There are 3 sets in this dataset with size 40×40, 25×25, and 8×8.

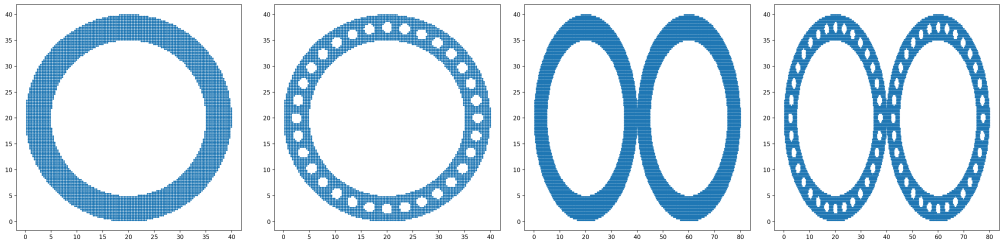


Figure 3: Synthetic Dataset: Different Scale of Holes

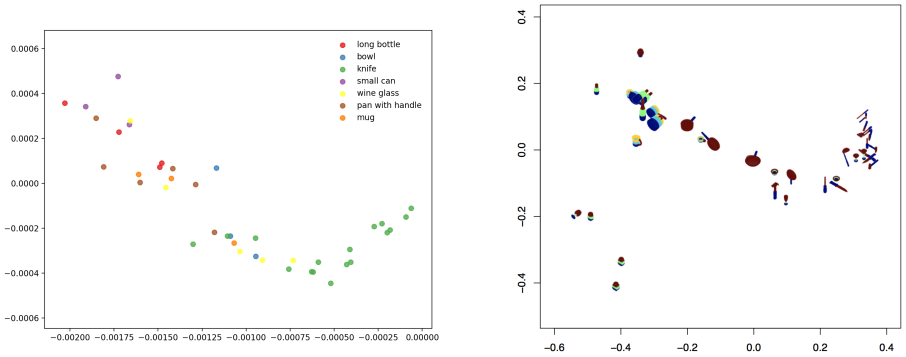
4 Implementation

The kernels are mainly written in Python. For SMURPH kernel, the persistence homology calculation is using Ripser (https://github.com/Ripser/ripser). Ripser computes the Vietoris-Rips persistence diagram, which is an important step for calculating SMURPH kernel. The reason to used Ripser is because it is currently the fastest library to compute VR persistence homology. Using Ripser greatly reduced the time to compute SMURPH kernel.

5 Evaluation

5.1 Kitchen Utensil Dataset

First, in order to validate my implementation of SMURPH kernel, I compared the kernel PCA result with the result given in the original paper. One thing to notice is that the parameters I used in my implementation is slightly different. This is due to the computation limitation. Specifically, in the original paper, they used a radius of $r = 0.1$, $m = 20$ centers per point cloud, $s = 1$ samples per center, and a budget of $b = 350$ points per sample. In my experiment, I used a radius of $r = 0.1$, $m = 10$ centers per point cloud, $s = 1$ samples per center, and a budget of $b = 100$ points per sample. The comparison is shown in Figure **??**. As we can see, the overall distribution of my implementation is very close to the result from original paper. The only noticable difference is that the small cans don't form a cluster in my implementation. This probably because the sample size is only 100 points compared to the original 350 points. The smaller sample size could capture the local structure but failed to capture the overall topological structure. So small cans and large pans are all considered as cylinders though their overall shapes are different.

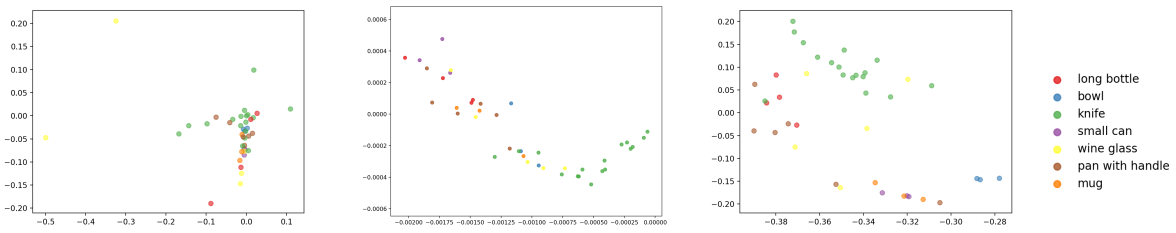


(a) My implementation

(b) Result from original paper

Figure 4: SMURPH kernel PCA results for Kitchen Utensil Dataset

I also calculated the linear kernel ($s = 100$ for sample size) and HOD kernel ($b = 10$ for number of bins in histogram) of this dataset. The kernel PCA using these three different kernels are shown in Figure **??** First of all, we can easily see that the linear kernel don't perform very well. We can hardly see any meaningful structure from figure. The HOD kernel performs pretty well comparing with the other two. There are clearly four clusters using HOD kernel: {knife}, {pan with handle, long bottle}, {small can, mug}, and {bowl}. It seems HOD kernel could also capture topological feature of a point cloud. However, since the four clusters not only varies in topology but also varies in shape, it's also possible HOD kernel is capturing the overall shape (e.g. long and thin vs. short and round).

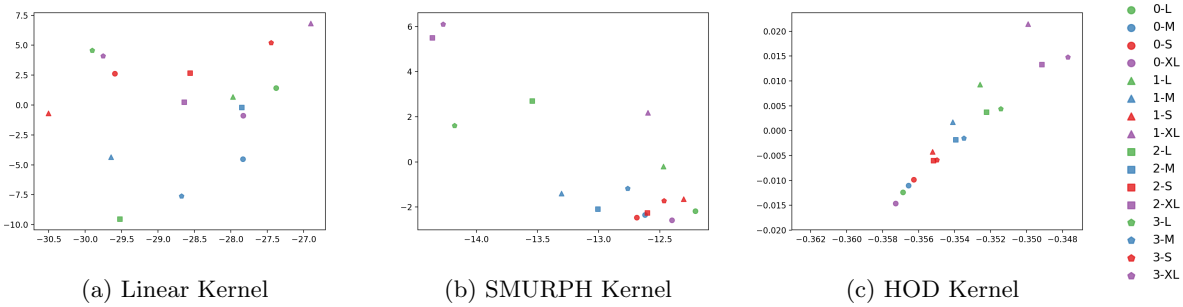


(a) Linear Kernel (b) SMURPH Kernel (c) HOD Kernel

Figure 5: Kernel PCA results using three different kernels for Kitchen Utensil Dataset

5.2 Synthetic Dataset – Multiple Holes

The SMURPH kernel, linearl kernel and HOD kernel are calculated for this dataset. The parameters for SMURPH kernel are: multiple radius of $r = [40, 20, 10]$, $m = 20$ centers per point cloud, $s = 1$ samples per center, and a budget of $b = 100$ points per sample. Linearl kernel samples $s = 100$ points from each point cloud. HOD kernel use a 10-bin histogram. The kernel PCA results are shown in Figure **??** First, there isn't any obvious structure in the result for linear kernel. So clearly linear kernel can't capture topological features of the data. For the result of SMURPH kernel, let's put the dots of the same color together as a group, which means we group together the point clouds with the same size of holes in it. So we have red set, which is {*-S}, blue set, which is {*-M}, green set, which is {*-L}, and purple set {*-XL}. We can see that although each set don't form a perfect seprated group, they both share the same tendency: from point clouds containing small sized holes to large sized holes, the 2D PCA becomes more and more sparse. This indicates SMURPH kernel is good at seperating point clouds with different number of holes. One thing to notice is that this doesn't mean SMURPH kernel is not good at seperating point clouds with different size of holes. Because in this dataset, the difference between holes of size S, M, L and XL is not so much. For HOD kernel, we can still analyze the PCA result in the same way. First, as we observed in the case of SMURPH kernel, the the group of point clouds with smaller holes forms a dense cluster. The group of point clouds with larger holes forms a sparse cluster. Besides, we can also see that point clouds of the same size forms a cluster. This means HOG kernel also captures the size of holes.

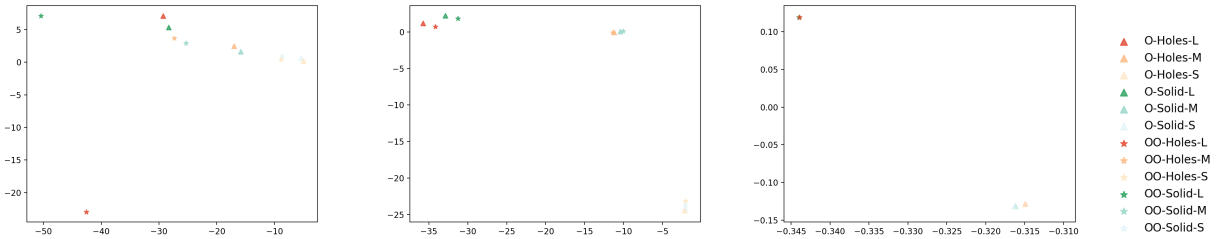


(a) Linear Kernel (b) SMURPH Kernel (c) HOD Kernel

Figure 6: Kernel PCA results for Synthetic Multiholes Dataset. The legend X-Y means the point cloud has X number of holes and the size of holes is Y (S=Small, M=Medium, L=Large, XL=Extra Large).

5.3 Synthetic Dataset – Multiscale

SMURPH kernel, linear kernel and HOD kernel are are evaluated for this dataset. The parameters for SMURPH kernel are: multiple radius of $r = [40, 10, 5]$, $m = 5$ centers per point cloud, $s = 1$ samples per center, and a budget of $b = 100$ points per sample. Linearl kernel samples $s = 300$ points from each point cloud. HOD kernel use a 10-bin histogram. The kernel PCA results are shown in Figure **??**. The result of linear kernel now has some structure in it. Small point clouds are on the right side of the figure and large point clouds are on the left side of the figure. The result of SUMRPH kernel also has this structure. It seperates point clouds with different sizes very well. How ever, it failes to seperate point clouds with obvious different topological features. On contrary, HOD kernel seems captured the topological features. O-shaped point clouds form a cluster and ∞-shaped point clouds form another. But clearly HOD kernel didn't capture the scale difference.



(a) Linear Kernel (b) SMURPH Kernel (c) HOD Kernel

Figure 7: Kernel PCA results for Synthetic Multiscale Dataset. The legend X-Y-Z means the point cloud is X-shaped, formed with ribbon Y, and size of the point cloud is Z (S=Small, M=Medium, L=Large, XL=Extra Large).

6 Discussion

The experiments show that SMURPH kernel can capture topological features of point clouds. However, it is not significantly better than other simple kernel. When used in practice, I doubt it will always improve the performance of an application. Although in the original paper, the authors have conducted other experiments to show the effectiveness of SMURPH kernel, they only compared it with simple linear kernel and RBF kernel, which is way too simple and don't really make sense in the context of point cloud data.

Another limiation is this kernel has too many free parameters to tune. User need to decide a radius scheme, which could be one value or a list of different values, number of centers, bootstrap sample size, and number of bootstraps. These parameters are essential to the performance of the kernel. So it will cost the user a lot of time for tuning. Also, with so many degree of freedom, when the kernel is used for regression or classification, it would be very suseptible to overfitting.

As a summary, the idea of SMURPH kernel is novel and have the potential to capture topological features of datasets. However, without further investigation and improvement, SMURPH kernel will not be suitable for using in practice.

References

[1] Xiaojin Zhu, Ara Vartanian, Manish Bansal, Duy Nguyen, and Luke Brandl. Stochastic multiresolution persistent homology kernel. In *International Joint Conference on Artificial Intelligence*, 2016.

[2] M. Neumann, P. Moreno, L. Antanas, R. Garnett, and K. Kersting. Graph Kernels for Object Category Prediction in Task-Dependent Robot Grasping. In *Proceedings of the Eleventh Workshop on Mining and Learning with Graphs (MLG–2013)*, Chicago, US, 2013.