QUANTUM TOPOLOGICAL FEATURE MAPS

JONAS KGOMO

Abstract. Visualising high-dimensional data has always been a difficult task, classical methods [1] in machine learning for extracting low-dimensional data sets like *principal component analysis(PCA)* have proven effective but not ideal for noisy or sparse data. *Topological Data Analysis (TDA)* uses tools from algebraic topology to understand the missing holes in data (shape of data). In this study, we look at how some properties of data remain unchanged under topological invariants using the *Vietoris-Rips* complex.
Firstly, in this article we use topological properties, namely *persistent homology* to extract features of a point cloud, X. We then replace X with *simplical complexes* that help us create persistent diagrams and filters on the point clouds. Secondly, we extract the features and embed the data points into a Hilbert space such that we can prepare the result as feature maps (*activation maps*) for quantum machine learning. This research proposes a framework that combines the feature results from topology to map them into quantum feature maps. Thus far research in TDA has been focused on algorithms and not specifically applying topological feature maps to quantum machine learning feature maps.

Key words and phrases. feature maps, simplical complexes, topology,Vietoris-Rips, TDA .

1. Introduction

There has been numerous accounts [9] of classification algorithms like t-SNE and for clustering data according to its properties. Traditional data analysis tools introduce artifacts that might cause biases and require algorithms to be fine-tuned to each case. In this research, we look at an alternative model of data analysis and computation.

Real high-dimensional data is typically sparse, and tends to have relevant low dimensional features. One task of TDA is to provide a precise characterization of this fact. Akin to Mak Kac's 1996 article: *"Can One Hear the Shape of a Drum?"* [2]

Topological invariants remains unchanged, that is, when the data shape is distored (stretched, twisted or bent) it still remains the same. We can apply the same theory to information and error mitigation in quantum data. The information is preserved under such transformations, in classical analysis, the data might be only invariant under rotation of symmetry, this approach helps us measure better.

Topological Data Analysis concerns itself with high-dimensional data sets that are robust under certain pertubations. High-dimensional data is necessary for building robust models, researchers at Alphabet [13] has been demonstrated that quantum advantage does not only rely on the complexity of the problem (task) but on the available data. TDA has been proposed for studying quantum algorithms and geometric analysis, however they have not been applied to enhance quantum data feature finding. We will be extracting topological featues from maps and embed the data features into a quantum machine learning model

Topological Quantum Computing. In 1997, Alexei Kitaev, introduced the model of topology for quantum computing, that is, using braids to form logical gates to create a stable computer. Moreover, in the quantum model, information under certain measurement changes the quantum state, under the topological regime, small changes cannot change the topological properties. There is also a general curiousity for mesoscopic quantum computing (topological regime) have been studied and recommended. Preskill [12], showed us that *decoherence* (unwanted interaction) arises so often, when we measure a *qubit* and under topological interaction , such properties could be realized without losing information, following [11] showed how this paradigm can introduce fault-tolerant quantum computation.

However, in our study, we will look at how tools in algebraic topology can help us create better tools for finding features in quantum data models.

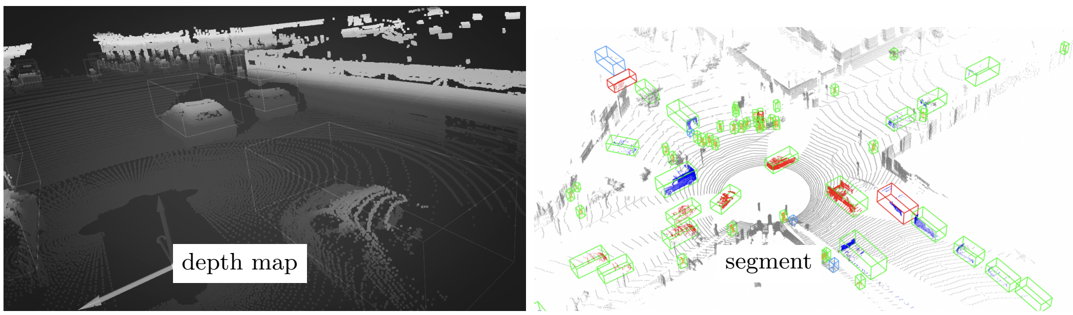
This would mean that most of the mechanism we will discuss would be native to such a infrastructure.

1.1. **Sparcity.** Let M denote matrix, and let $\gamma = (x_1, x_2 \cdots x_n)$ denote elements of M and η denote elements with $x_i = 0$. We denote the sparsity as, $S = \dfrac{|\eta|}{|\gamma|}$, and say that the matrix M is *sparse* if $S < \epsilon$, in two dimensions $\epsilon = 1/2$. We can also collect the set of such sparsities as : $\mathcal{S} = \{\eta_i : \eta_i \in \gamma\}$

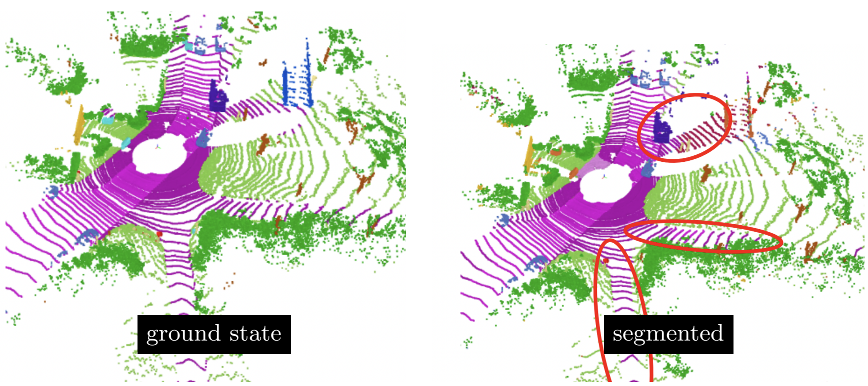
Methods like graph representation have been recommended (Jiaxuan You, 2020) [5] for imputation of missing data , however this does not account for sparse data. Laurens introduced `t-SNE` for handling high-dimensional sparse data, showing that the dimensionality of γ can be reduced to visualise the data in lower dimensions. PCA [3](Hotelling, 1933) and multidimensional scaling [4] (Torgeson, 1953) are techniques that are concerned with low-dimensional datapoints, however these techniques are limited by the linear mapping approach, and they lack general dimensionality reduction ($d > 3$).

1.2. **Point clouds.** We will consider discrete surfaces (point clouds), a point cloud $\mathcal{P} := \{p_1, p_2 \dots p_n \in \mathbb{R}\}$ will be considered. Organising principles for data sets are often classical and algorithmic. However the complexity of data can be structural and our hypothesis is that the shape of data can help us cluster the data. Raw data in a point cloud is not ideal for machine learning algorithms, we need to find a feature matrix F, that has a single point cloud, and its features.

Autonomous driving point clouds automatically labelled and rendered using Waymo's Open Data Set [6].



This is constructed from the range (depth) and colour map of a LiDAR dataset, see below for[0] a live example.



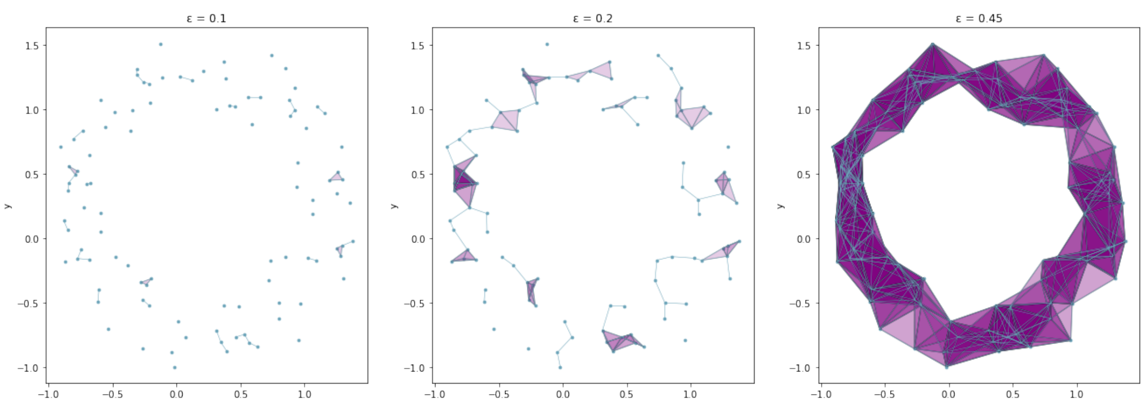
The accompanying diagram shows LiDAR 3D point clouds from the Semantic KITTI [8] dataset, they are segmented to identify certain objects in the data set. This method of using semantic segmentation was introduced at Hauwei [7] and multiple studies have also been done to survey [9] the landscape of deep learning for 3D point clouds. The research material in this area focuses on shape classification, object detection and tracking, and point cloud segmentation. The methods above are mainly motivated by research in 3D autonomous driving, however in higher dimensional data sets that have features and objects dissimilar to those found in driving training models, the research is yet to be developed. We dedicate this research to high-dimensional ($d > 3$) general data points that are.

[0]3D Point Cloud by Scale AI: `https://lidar-now.scale.com`

2. Persistant Homology

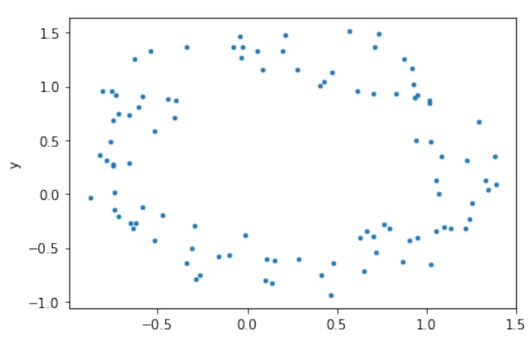
Definition 2.1. *Simplexes*
A simplex σ is a geometrical object that generalises a triangle in higher dimensions. A k-simplex is a geometric object that gives the convex hull of its $k+1$ vertices $v_0, v_1 \dots v_k \in \mathbb{R}^d$, denoted as $\sigma_k = [v_0, v_1, \cdots v_k]$. Notably, $k=0 \implies$ circle, $k=1 \implies$ line , $k=2 \implies$ triangle

The sketching algorithms and persistent diagrams
We use the method of the Persistent homology which uses a filtration method, instead of compressing the data , the latter method namely, *Mapper*, was developed by Singh et al. [14]

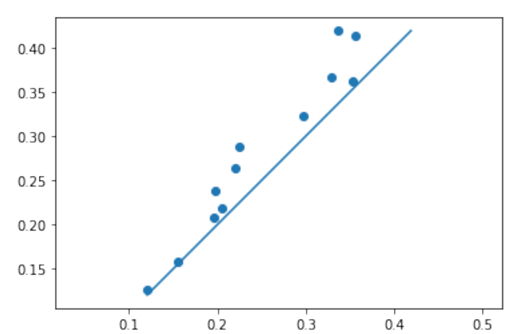


Simplex Connectivity $\epsilon_{1,2,3}$

Vietoris–Rips complex $\mathcal{R}_\epsilon(X)$ for a point cloud $\mathcal{P}$ at $\epsilon = 0.1, 0.2, 0.45$.
At each step we map circles *(1-simpleces)* that have radius $r = \epsilon/2$ around the point clouds. We choose ϵ that yields sufficient connectivity and gives rise to structural artifacts on the map. When ϵ grows, we begin to see more topological features develop, that is holes, gaps and voids (*simplices*). We can also assign temporal logic to this part, this could be generated by a dynamic temporal graph network [15]

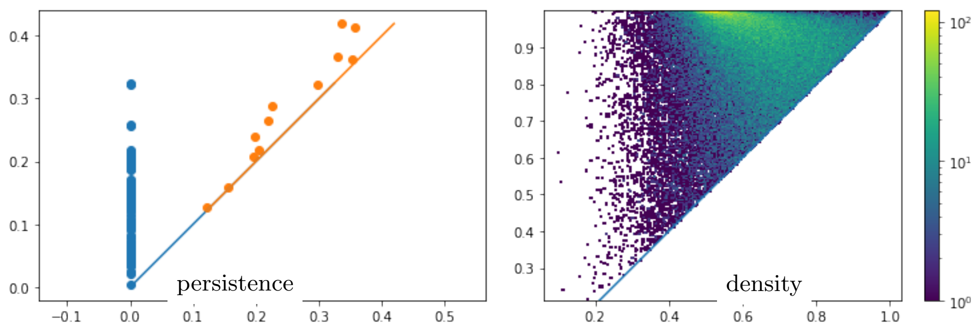


scatter plot

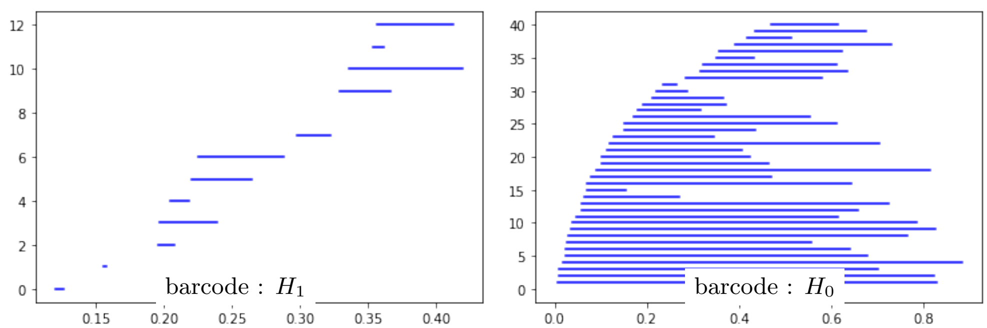


persistant diagram

Persistence diagrams show us the birth and dearth of topological features , the diagram on the right shows this plots with birth on the x axis and death on the y axis



A persistent diagram of births vs deaths of topological featuers , the density map shows the distribution of this points. We used *dionysus* python package and the notebook for [1]



This gives us the $\mathcal{R}$ Vietoris-Rips complex with Betti numbers at 0 and 1.

[1]`https://colab.research.google.com/drive/1BQQSIizbAxAY3XjQreHclBpHBPLxB9zv?usp=sharing`

3. PIPELINE WORKFLOW

data $\rightarrow$ (η) vectorize $\rightarrow$ persistence map $\rightarrow$ kernelize $\rightarrow$ quantum feature map

Given a data p_i , we use topological homology to create the persistent diagrams , the Vioteris-Rips method VR_ϵ allows us to create features maps on the diagram. The resultant maps are then mapped into a Hilbert space using kernels. This final result is representational of a quantum feature map.

3.1. **Topological Data Analysis.** TDA introduces a much more robust way to deal with high-dimensional data sets. In a manifold, we extend the idea of sparsity to *holes* in an n dimensional surface, the number of holes in a surface are called *genus*. Topological methods have also been recommended a representation map for partial clustering of data sets [10].

4. EMBEDDING

4.1. **State Preparation.** Using a $qRAM$ we can assume a vector $\vec{V_N}$ and map into a quantum state $\|V\rangle \in \mathbb{C}^{\mathbb{N}}$ with an exponential compression of $O(\log(N))$.

A quantum algorithm like Grover's algorithm on a simulated quantum computer with a few hundred qubits is able to find the full persistent homology of a data set over n points in time $O(2^n)$. While a classical computer has a weaker time-space complexity of $O(2^{2n})$, shown by Seth Lloyd [16].

Table 1. Benchmark on quantum and classical regimes

Classical	**Topology**	**Quantum**
$\mathcal{O}(2^{2n})$	$\mathcal{O}(2^n)$	$\mathcal{O}(n^2)$
Distance	Metric	Coherence
Variant	Stable	Homology

5. Quantum Feature Maps

In the classical persistent homology algorithm, the data are simplices in a simplicial complex, which is the topological structure that results from the data being grouped together. The quantum pipeline to persistent homology begins by taking the quantum states that encode the original data.

There is a challenge in how we should pick the value ϵ, however we can calculate all possible states in a quantum state $\|\epsilon\rangle$ preparing take the TDA filter as a feature map

5.1. **Kernelization.** Given a point cloud set $\mathcal{X}$, a function $k : \mathcal{X} \times \mathcal{X}$ is a kernel, if there is a Hilbert space $\mathcal{H}$, and a feature map $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{H}$ so that $k(x, y) = \langle \mathcal{F}(x), \mathcal{F}(y)\rangle_{\mathcal{H}}$ for all $x, y \in \mathcal{X}$.

(inner product space with a complete metric space)

This kernel , k can be used to find objects that are not similar. The kernel of two persistent diagrams [19] we can also implement the Hausdorff distance d_H . Peter Bubenik [20] also recommended a statistical layer of persistence called the *Persistence Landscapes*.

5.2. **Filtration.** We use Grover's algorithm, by Lov Grover, 1996 [17] to find the simplical complex , S_ϵ. When collected together , this forms a *filtration.* Given a k-simplex s_k, with a basis vector of $|s_k\rangle \in \mathbb{C}^{\mathbb{N}}$, we will denote the Hilbert space of this simplices as $\mathcal{H}$ where $s_k \in S_k^\epsilon$.

The Grover algorithm below can be used to create the $k-$simplex state:

$$|\psi\rangle = \frac{1}{\sqrt{|S_k^\epsilon|}} \sum_{s_k} |s_k\rangle \tag{5.1}$$

S_k^ϵ represents the set of all $k-$simplices in the complex observed at scale ϵ, hence its cardinality gives us the number of $k-$simplices $|S_k^\epsilon| = \dim H_\epsilon^k$. $|\psi\rangle_k^\epsilon$ is the uniform superposition of the quantum states.

5.3. **Embedding.** Machine learning with kernels involves embedding a data point x as a vectors $\vec{x}$ into a Hilbert space. Persistence diagrams produce kernels in the Hilbert Space. Quantum computers also map data into a Hilbert space. This similarity is our motivation to using topological results to map into a quantum Hilbert space. The quantum analogue of the feature map is called the. *quantum feature map* due to Maria [21]

Using the notion we have about kernels in topological persistence, we will thus present a pipeline for embedding the kernel space into a Hilbert space.

5.4. **Classifiers RKHS.** Maria Schuld et al studied quantum classifiers and methods like squeezing as a feature map [21]. Here we have shown that the kernel space of a persistence diagrams can be used as a feature map. A kernel produces a reproducible kernel Hilbert space for our feature map using vector methods. The reproducing kernel $\mathcal{R}$ can be represented as $\mathcal{R} = f : \mathcal{X} \to \mathbb{C}$ for a given feature map.

Carlsson [18] showed us that we can use persistent homology as a layer for machine learning.

6. Topological Properties

Given persistence diagrams $\mathcal{D} = D_1, D_2 \cdots D_N$ we can produce a kernel matrix $\mathcal{K}$ as follows

$$\begin{bmatrix} k(D_1, D_1) & ... & k(D_1, D_1N) \\ . & ... & . \\ k(D_N, D_1) & ... & k(D_N, D_N) \end{bmatrix}$$

A unitary feature embedding circuit $U_\phi(x) : x \in 0, 1^n \to |i\rangle$, can be described for each kernel, $\|(D_i, D_j)$ as described by Aram, et al in quantum enhanced feature spaces [22].

Generally the distance between these diagrams are well studied under the Bottleneck distance and the Wasserstein distance, we confine to the general metric norm.

At each point we can also adjust the scale at which we are analysing the persistence diagram, through the parameter ϵ and assess which topological features persist despite the changes in ϵ.

7. Conclusion

The results have been developed by looking at QML however we can discover similar outcomes by studying CML (Classical Machine Learning) The result of quantum embedding are foreign to a classical computer and can be done successfully on a quantum computer , as a result we are not able to assess the quantum advantage of this method.

The model used for inferring data and representing it into a Hilbert space can be further studied with a visual model, a way to insert an n dimensional vector . The RKHS method provided by our result is able to offer us tools for support vector machine, Principal component analysis, Change point analysis – however we have looked at this tool for embedding data into a Hilbert space.

References

[1] Laurens van der Maaten, Geoffrey Hinton, *Visualizing Data using t-SNE* Journal of ML Research 1 (2008) **1-48**

[2] Mark Kac *Can One Hear the Shape of a Drum?* , Rokeffeler

[3] Hotelling, H. (1933). *Analysis of a complex of statistical variables into principal components.* Journal of Educational Psychology, 24(6), 417 – 441

[4] Warren S. Torgerson , *Unconditional prime-representing functions*, Psychometrika, volume 17 **401-419** (1952) 639.

[5] Jiaxuan You, Xiaobai Ma, Daisy Yi Ding, Mykel Kochenderfer, Jure Leskovec *Handling Missing Data with Graph Representation Learning*, Department of Biomedical Data Science, Stanford University

[6] *Waymo's Open 3D Dataset*, Alphabet, `https://github.com/waymo-research/waymo-open-dataset`

[7] Ran Cheng, Ryan Razani, Yuan Ren and Liu Bingbing, *S3NET: 3D LiDAR sparse Semantic Segmentation Network*, Huawei Noah's Ark Lab, Toronto, Canada

[8] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, *Semantickitti: A dataset for semantic scene understanding of lidar sequences*, South Korea, IEEE, 2019 pp 9296

[9] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun *Deep Learning for 3D Point Clouds: A Survey*, IEEE

[10] Gurjeet Singh, Facundo Memoli and Gunnar Carlsson, *Topological Methods for the Analysis of High Dimensional Data*, Eurographics Symposium (2007) **53** (1947), 604

[11] A. Yu Kitaev *Fault-tolerant quantum computation by anyons*, L.D.Landau Institute for Theoretical Physics

[12] John Preskill, *Fault tolerant quantum computation*, California Institute of Technology, Pasadena, CA 91125, USA , 1997

[13] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo , Hartmut Neven & Jarrod R. McClean, *Power of data in quantum machine learning*, Google Research Blog

[14] Singh G, Memoli F, Carlsson GE *Topological methods for the analysis of high dimensional object recognition* In: SPBG, pp 91-100

[15] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca *Temporal Graph Networks*, Twitter Research

[16] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi, *Quantum algorithms for topological and geometric analysis of data*, Nature Communications, 2015.

[17] Lov Grover, *A fast quantum mechanical algorithm for database search* 1996

[18] Bradley J. Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J. Guibas, Gunnar Carlsson, *A Topology Layer for Machine Learning* Stanford University

[19] J. Reininghaus, S. Huber, U. Bauer and R. Kwitt, *A stable multi-scale kernel for topological ML*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

[20] P. Bubenik, *Statistical Topological Data Analysis Using Persistence Landscapes*, Journal of Machine Learning Research 16, 2015, pp. 77–102

[21] Maria Schuld and Nathan Killoran *Quantum machine learning in feature Hilbert spaces* Xanadu, 372 Richmond St W, Toronto, M5V 2L7, Canada

[22] Vojtech Havlicek, Antonio D. Corcoles, Kristan Temme, Aram W. Harrow *Supervised learning with quantum enhanced feature spaces* IBM T.J. Watson Research Center

Jonas Kgomo, Sussex School of Engineering and Informatics Sussex University, Brighton, Nagoya, Falmer, Brighton, BN1 9RH