# Quantum Machine Learning with Quantum Topological Data Analysis

Team AnK - Ankit Khandelwal

February 2023

## 1 Introduction

Topological Data Analysis (TDA) is a powerful but computationally expensive data analysis tool. In this project, I use Quantum Topological Data Analysis (QTDA) to get the Betti numbers of data and use them for classification tasks. A hybrid quantum-classical machine learning model performed best in the experiments.

TDA is a robust method for extracting valuable data from high-dimensional and possibly noisy data. The data feature we consider is the $k^{th}$Betti number, which is the number of $k-$dimensional holes and voids in the data. QTDA algorithms can be used to get these Betti numbers, and machine learning tasks can be performed with these new features using both classical and quantum models.

## 2 Basics of TDA

The first step is to make a point cloud of the dataset ($\{x_i\}_{i=1}^{n}$) in an $m$ dimensional space where $m$ is the number of features in the dataset, and $n$ is the number of data points. This space has a distance function $d(x_i, x_j)$ defined on it, usually the Euclidean distance on $\mathbb{R}^m$. Using this distance function, we connect all points within $\epsilon$ (grouping scale, referred to as edge length in the notebooks) distance from each other with edges. We get a graph $G_\epsilon = (V, E_\epsilon)$, with vertices $V = \{v_1, v_2, \ldots, v_n\}$ and edges $E_\epsilon = \{(i, j) \mid d(x_i, x_j) \leq \epsilon\}$.

A $k$-simplex is a collection of $k+1$ vertices with $k(k+1)/2$ edges in $k$ dimensions. A number of these simplicies are present in the graph $G_\epsilon$, which are together called the simplicial complex $\mathcal{K}_\epsilon$.

Let $S_k^\epsilon$ be the set of $k$-simplicies in the complex $\mathcal{K}_\epsilon$ with individual simplicies denoted by $s_k^\epsilon \in S_k^\epsilon$ written as $[j_0, j_1, \ldots, j_k]$ where $j_i$ is the $i^{th}$ vertex of $s_k^\epsilon$. Note that the vertices are ordered in ascending fashion in the initial point cloud, and this order is kept throughout.

The restricted boundary operator $\partial_k^\epsilon$ is defined on the $k$-simplicies as:

$$\partial_k^\epsilon s_k^\epsilon = \sum_{t=0}^{k} (-1)^t [v_0, \ldots, v_{t-1}, v_{t+1}, \ldots, v_k] \tag{1}$$

$$= \sum_{t=0}^{k} (-1)^t s_{k-1}^\epsilon(t) \tag{2}$$

where $s_{k-1}^\epsilon(t)$ is defined as the lower simplex defined from $s_k^\epsilon$ by leaving out the vertex $v_t$.

From $\partial_k^\epsilon$ we get the $k$-homology group defined as the quotient space $\mathbb{H}_k^\epsilon$:

$$\mathbb{H}_k^\epsilon = \frac{\ker \partial_k^\epsilon}{\operatorname{Im} \partial_{k+1}^\epsilon} \tag{3}$$

where $\ker A$ and $\operatorname{Im} A$ are the kernel (null space, set of solutions of $A\vec{x} = \vec{0}$) and image (set of all outputs $A\vec{x}$) of $A$.

The $k^{th}$ Betti number $\beta_k^\epsilon$ is the dimension of $\mathbb{H}_k^\epsilon$:

$$\beta_k^\epsilon = \dim \mathbb{H}_k^\epsilon \tag{4}$$

where $\dim V$ is the cardinality of the basis of $V$.

Note that this Betti number depends on the choice of the grouping scale $\epsilon$.

Another way to calculate the Betti number $\beta_k^\epsilon$ is by defining the combinatorial laplacian $\Delta_k^\epsilon$:

$$\Delta_k^\epsilon = (\partial_k^\epsilon)^\dagger \partial_k^\epsilon + \partial_{k+1}^\epsilon (\partial_{k+1}^\epsilon)^\dagger \tag{5}$$

and getting $\beta_k^\epsilon$ as:

$$\beta_k^\epsilon = \dim \ker \Delta_k^\epsilon \tag{6}$$

Thus, $\beta_k^\epsilon$ is the number of zero eigenvalues of $\Delta_k^\epsilon$.

The classical preprocessing steps are shown in Fig. 1.

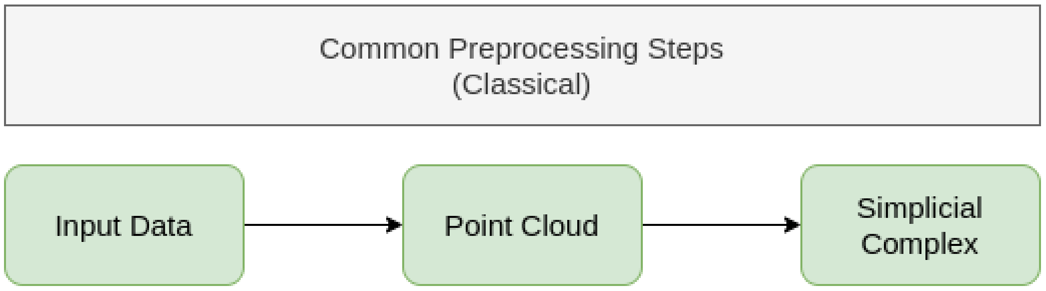


Figure 1: Common classical preprocessing steps to be performed on the data

## 3 Quantum Topological Data Analysis

In 2016, Lloyd et al. proposed the first algorithm to calculate Betti numbers using a quantum computer known as the LGZ (Lloyd, Garnerone and Zanardi) algorithm. Since then, various improvements and variations have been proposed. Here, we use the QPE algorithm to estimate the number of zero eigenvalues of the combinatorial laplacian.

Given the grouping scale $\epsilon$ and an integer $0 \leq k \leq n-1$, we start with the combinatorial laplacian $\Delta_k^\epsilon$. Due to its form, the combinatorial laplacian is a real symmetric matrix with dimension equal to the number of $k$-simplicies in $S_k^\epsilon = |S_k^\epsilon|$. If $\lambda_j$ is an eigenvalue of $\Delta_k^\epsilon$, then $e^{i\lambda_j}$ is an eigenvalue of the unitary $e^{i\Delta_k^\epsilon}$. The QPE algorithm estimates the eigenvalue of an eigenvector of a unitary operator, i.e. given a unitary matrix $U$ with a quantum state $|\psi\rangle$ such that $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, QPE estimates the value of $\theta$. Thus, $\theta = 0$ corresponds to $\lambda = 0$ for $\Delta_k^\epsilon$.

The unitary $U$'s shape needs to be $2^q \times 2^q$ to act on $q$ qubits. Thus, the matrix $\Delta_k^\epsilon$ also needs to be padded to get the dimension to the nearest power of 2. We pad the combinatorial laplacian $\Delta_k^\epsilon$ with an identity matrix with $\tilde{\lambda}_{max}/2$ in place of ones. Here, $\tilde{\lambda}_{max}$ is the estimate of the maximum eigenvalue of $\Delta_k^\epsilon$ using the Gershgorin circle theorem such that:

$$\tilde{\Delta_k^\epsilon} = \begin{bmatrix} \Delta_k^\epsilon & 0 \\ 0 & \frac{\tilde{\lambda}_{max}}{2} \cdot I_{2^q - |S_k^\epsilon|} \end{bmatrix}_{2^q \times 2^q} \tag{7}$$

where $\tilde{\Delta_k^\epsilon}$ is the padded combinatorial laplacian and $q = \lceil \log_2 |S_k^\epsilon| \rceil$ is the number of qubits this operator will act on.

In QPE, as $2\pi\theta$ increases beyond $2\pi$, the eigenvalues will start repeating due to their periodic form. Thus, $\theta$ is restricted to $[0, 1)$. As $\lambda \to 2\pi\theta$ this

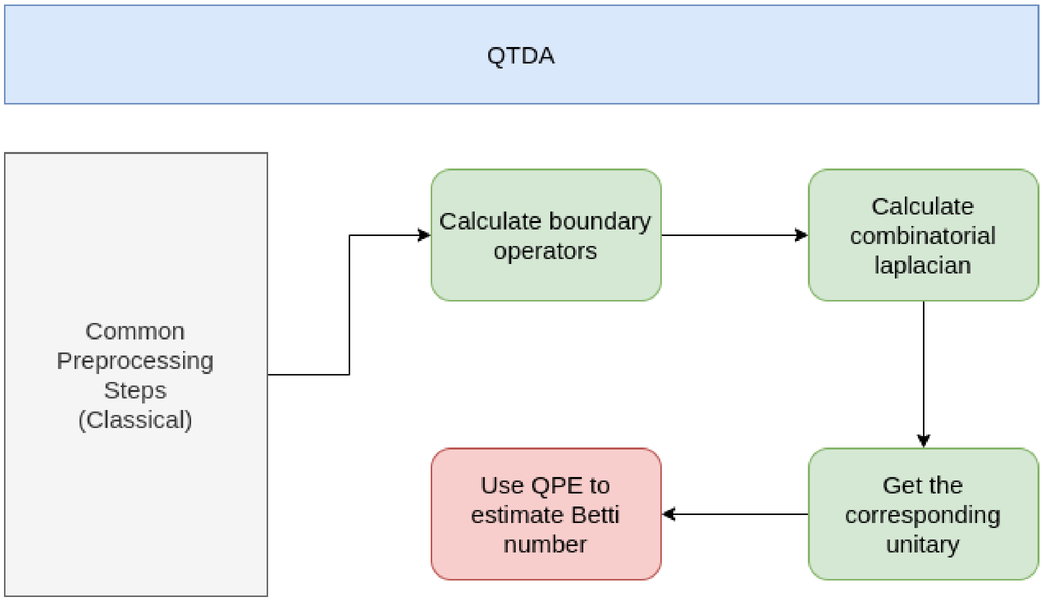

QTDA
Common Preprocessing Steps (Classical)
Calculate boundary operators
Calculate combinatorial laplacian
Get the corresponding unitary
Use QPE to estimate Betti number

Figure 2: The complete QTDA pipeline

means that $\lambda \in [0, 2\pi)$. Thus, we need to restrict the eigenvalues of the combinatorial laplacian to this range. This can be achieved by rescaling $\tilde{\Delta}_k^\epsilon$ by $\delta/\tilde{\lambda}_{max}$ where $\delta$ is slightly less than $2\pi$. Thus, the final unitary for the QPE algorithm is:

$$U^\epsilon = e^{iH^\epsilon} \tag{8}$$

$$H^\epsilon = \frac{\delta}{\tilde{\lambda}_{max}} \tilde{\Delta}_k^\epsilon \tag{9}$$

The initial state on which QPE acts is an eigenstate of the unitary, and the $\theta$ value is the corresponding phase of the eigenvalue. Suppose, instead, we use the maximally mixed state $I/2^q$ as the initial state and run the algorithm for $\alpha$ times. In that case, the probability of getting zero in the eigenvalue register is given by:

$$p(0) = \frac{|\{i \mid \tilde{\theta}_i = 0\}|}{\alpha} = \frac{\tilde{\beta}_k^\epsilon}{2^q} \tag{10}$$

$$\implies \tilde{\beta}_k^\epsilon \quad = 2^q \cdot p(0) \tag{11}$$

where $\tilde{\theta}_i$ is the $i^{th}$ estimate of $\theta$ from QPE and $\tilde{\beta}_k^\epsilon$ is our estimate of the Betti number $\beta_k^\epsilon$.

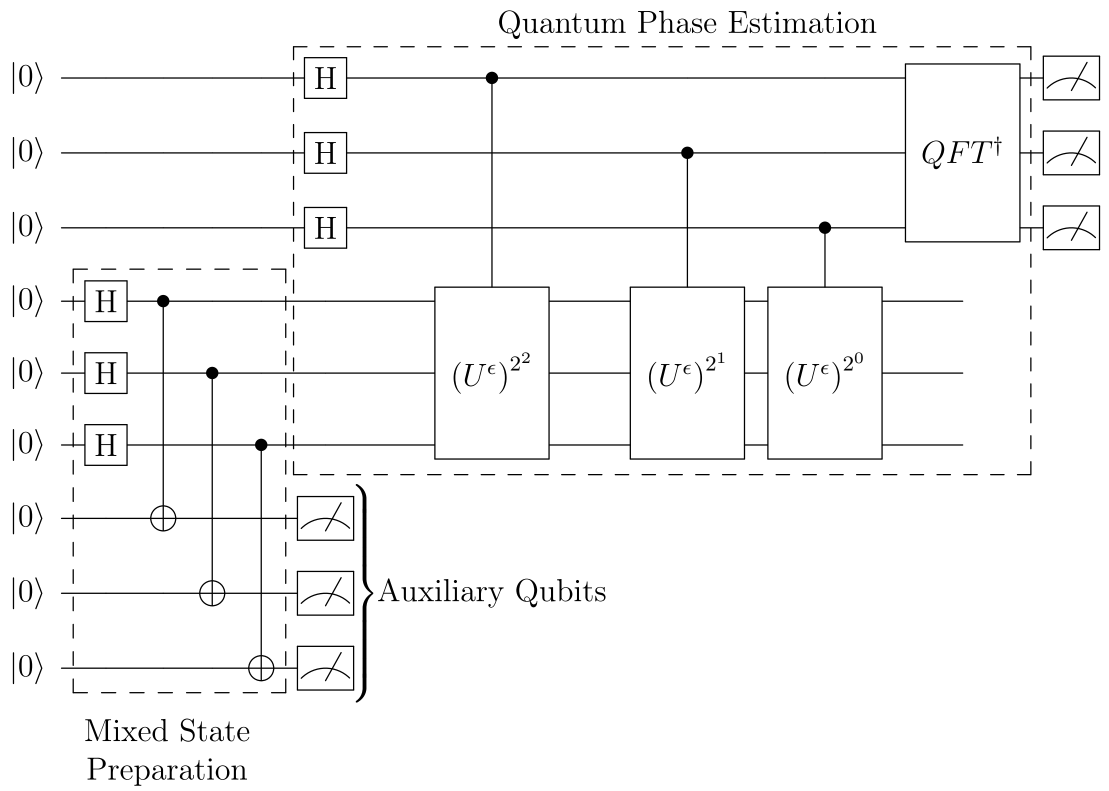


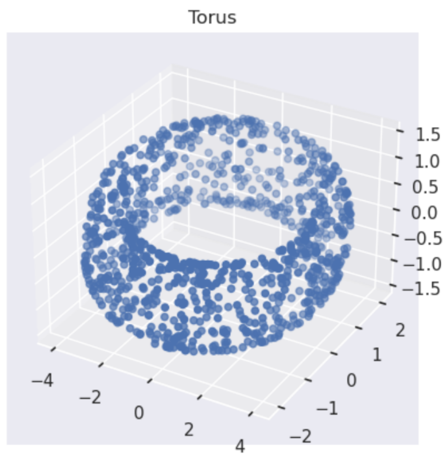
Figure 3: The quantum circuit for the QTDA algorithm (using 3 precision qubits and when the unitary is also simulated using 3 qubits).

The QTDA algorithm is captured in the pipeline shown in Fig. 2 and the circuit is shown in Fig. 3.

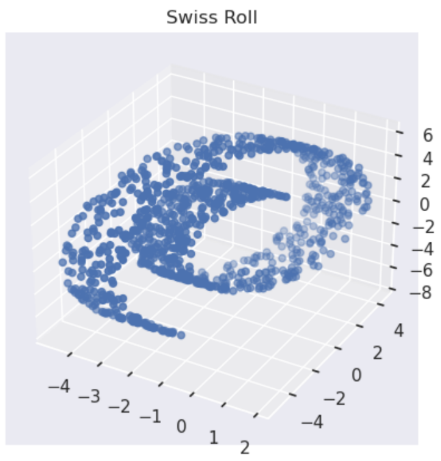
## 4 Classification using QTDA

The machine learning capabilities of the Betti numbers are demonstrated using two datasets. One is to classify the handwritten digits 0 and 1 from the MNIST dataset, and the other one is to classify between shapes of a torus and a swiss roll (see Fig. 4). The data consists of sampled points from the shape which are then embedded into a 10-dimensional space and rotated randomly.

We consider multiple approaches to accommodate Betti numbers in the machine learning algorithm, one of which is approach 3 as shown in Fig. 5.



(a) Torus



(b) Swiss Roll

Figure 4: Example shapes

This approach directly feeds the QTDA output into a variational quantum classifier (VQC) circuit. The probabilities after measurement of the VQC are then fed to a small classical neural network to get the final predictions. This method provided the best accuracy values on the testing data out of all the approaches tested.

We also performed experiments to show how the accuracy changes depending on the edge length and data noise.

## 5 Future work

There are multiple open ends to this problem. Some of the points under consideration are:

- How to reduce the resources used (qubits and circuit depth) in the QTDA algorithm?
- Use persistent Betti numbers instead. These do not depend on the edge lengths.
- A simple VQC was used here, what different qml models can be combined and what are the best methods?

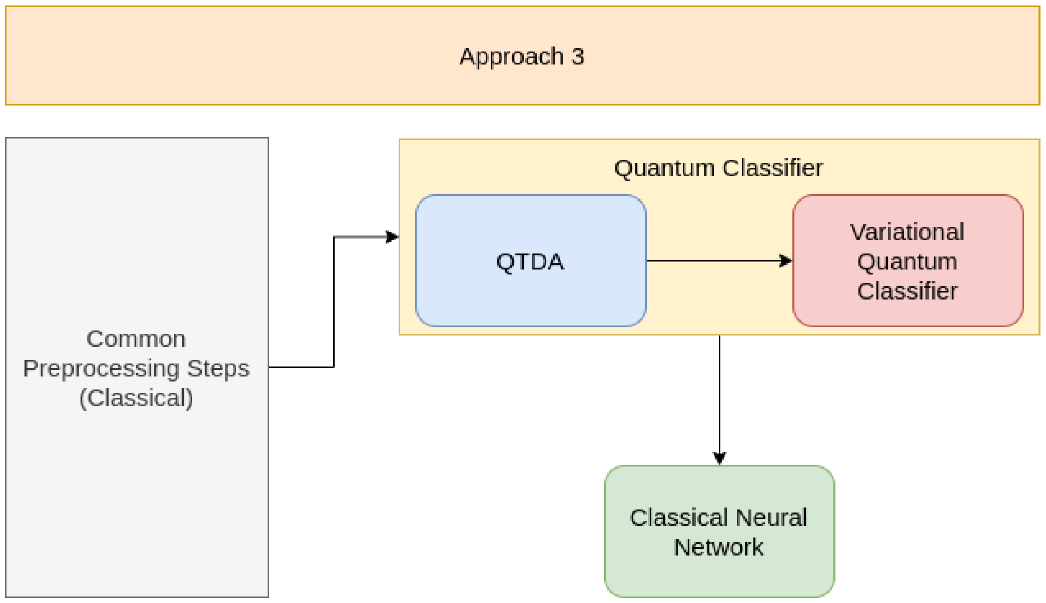

Approach 3
Common Preprocessing Steps (Classical)
Quantum Classifier
QTDA
Variational Quantum Classifier
Classical Neural Network

Figure 5: The hybrid quantum-classical machine learning pipeline.

- How will noise from the quantum circuit affect the results?
- How can this effect of noise be minimised?