Milestone 3 Quantum

William Pelech

November 2024

1 Project Overview

The purpose of this document is to provide a brief overview of my code to understand the progress made between milestone 2 and 3. To start the functions made for this milestone are listed as follows:

- Encoding of Matrix for Quantum Computing
- Quantum Phase Estimation
- Cluster Algorithm

Some other unique features process of being worked on are implementing a Grover's search algorithm to be able to possibly order the eigenvalues of the matrix.This is optional and actually not an efficient solution so I will be thinking of ways to include Grover's Search in other ways.

1.1 Encoding the Matrix

To first understand the how this was done there was a sub-function used to normalize and encode each row of the matrix. This function was called encode row. Note that the there had to be a function to determine how many Qubits would be required for the encoding this is where the variable padded length comes into play. The encode matrix actually stores each row as a different quantum circuit.

1.2 Quantum Phase Estimation(QPE)

The QPE function has one sub function call which is to q inv a function that outputs the inverse quantum Fourier transform. The way QPE works is by "exposing" a quantum state by a unitary function by different lengths of time(the amount of times exposed to the unitary). The variable which is used in our code for this is r which gets incremented each time which will perform 2^n operations. Note in this part Qiskit barriers had to be used around the inverse function because the way Qiskit works is that it will optimize for each function which will not be necessarily beneficial when function calls are made.

1.2.1 Choice of Heap Data Structure

Not that once the eigenvalues were obtained they needed to be ordered for that the largest eigenvalues of most significance appear in Ascending order from the top left down to the bottom right. This was done using a heap. There can be arguments made on why to use Grover's search however I believe that min heap would have better performance have $O(log(n))$ runtime which is asymptotically better than $O(\sqrt(n))$ which is the runtime for Grover's Search. The reason this is because the problem really needs a data structure and not necessarily a search algorithm.

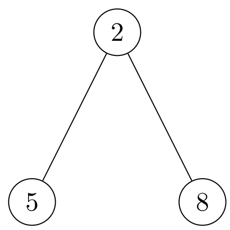

Diagram of heap data structure

1.3 Cluster Algorithm

To implement my point clustering algorithm I implemented a classical K-Means algorithm that would be used to group users off of varying statistics into different clusters. For the next milestone I am planning on implementing a different clustering algorithm known as a Gaussian Mixture model which relies on the assumption that characteristics of a user follow a normal distribution.

2 Next Steps

For the next steps of my project I would like to work on a couple added features and make different comparisons which will be included below.

- Comparing Cluster Algorithms
- Run all functions simultaneously
- Stress test decomposition and optimize data

My reach goal is to use Kaggle or Scrape from a website to get a working and comparable algorithm.