HIGH SCHOOL EDITION
Journal of Student Research

Volume 11 Issue 2 (2022)

Novel Enhanced Unsupervised Quantum Clustering Algorithm

Diptanshu Sikdar[1], Joe Clapis[#], and Swetha Bhattacharya[#]

[1]BASIS Independent Silicon Valley, CA, USA
[#]Advisor

ABSTRACT

Every day, humans generate approximately 2.5 billion gigabytes of new data. While this data is crucial in applications such as drug development and cybersecurity, it is mostly unstructured and unlabeled which makes it difficult to process effectively. Businesses and governments alike solve this problem by using machine learning algorithms to characterize the data and make decisions with it. However, conventional machine learning processes consume excessive computational resources and time: one of the most prominent, K-means clustering, has an approximate time complexity of $O(n^2)$, where n is the input data size. This quadratic time complexity causes scalability problems with large datasets. In this project, I have created an improved, more scalable clustering algorithm by leveraging quantum computation. The approach (new Q-means) involves an Angle Embedding Feature Map, an d-dimensional SWAP Test, and a dynamic, threshold distance-based termination criteria. After implementing this algorithm using the IBM Qiskit SDK, the new Q-means algorithm was run several times on different dimensional datasets. On an average, the new Q-means performed 18% better than the K-means algorithm based on the Adjusted Rand Index. In terms of the simulation execution time, the new Q-means was about six times faster than the existing Q-means due to a multi-threaded implementation and faster convergence. The time complexity of the new Q-means without a Quantum Random Access Memory (QRAM) is $O(n^2)$. While the existing Q-means algorithm, which utilizes QRAM, has a time complexity of $O(n(\log n)^p)$, the time complexity of the new Q-means with QRAM is significantly better—$O(n)$.

Introduction

Over the past 20 years, there has been an exponential growth of new data. Every day, people use their electronic devices—smartphones, tablets, laptops—for tasks like shopping online, browsing the news, chatting on social media, trading stocks, etc. Furthermore, people continuously store and access their data using cloud services. Besides, sensors connected through the Internet of Things generate tons of new data every day. Not only do we generate data, but we also run algorithms to process and analyze this data, creating even more data. All this data—6,800 exabytes in 2020—sums up Big Data.

Learning from this data is extremely valuable because it drives effective and impactful decision-making. Informed decisions can translate into positive growth metrics for businesses. However, most of this data is unlabeled, unstructured, and chaotic. Since manual labeling is costly for most applications and labeled datasets are scarce, it is important to learn from unlabeled data. This is known as unsupervised learning.

There are three different types of unsupervised learning: clustering, dimension reduction, and association. Clustering has become a field of high interest because of its pertinence in the real world. Businesses often utilize clustering algorithms to make important and informed business decisions. “For companies, organization and categorization strategies are fundamental to success. Clustering information can make all the difference.” [4] Clustering can help businesses observe trends in sales (via product clustering) and perform customer segmentation (via client clustering). Clustering is also helpful in other applications like earthquake study, taxonomic classification, image segmentation, and anomaly detection. In addition, scientists are using clustering algorithms to identify cancerous cells from medical image data. Also, educators use clustering algorithms to monitor students' academic performance in various classes. Often, these applications have tons of features and data points. Therefore, it is crucial to run efficient algorithms on efficient computing systems for increasing the goodness of clustering and reducing overall execution time.

One of the most prominent clustering algorithms is K-means. It is an iterative process that utilizes centroids to cluster data. K-means has a time complexity of $O(tknd)$, where t is the number of iterations, k is the number of clusters, d is the number of dimensions, and n is the number of data points. $O(tknd)$ approximates to a quadratic complexity since the number of iterations is proportional to the number of data points, as per [7]. Such a time complexity shows that K-means cannot be scaled efficiently for real-world applications.

Quantum computing (QC) is an emerging computing paradigm that leverages the principles and properties of quantum mechanics. QC enables faster computation via extreme parallelism and often increased precision/accuracy for certain problems, resulting in increased efficiency. In recent years, researchers have considered different ways in which quantum technologies can aid machine learning. In fact, there have been several proposals, potentially promising significant speedups over classical algorithms.

Quantum Computing

The fundamental unit of quantum information is a quantum bit or a qubit. In classical computing, the bit is the smallest unit of information. It can only exist in a single state, either 0 or 1. A qubit's state is represented as a vector using Dirac's notation. Contrary to a bit, a qubit's state, $|\psi\rangle$, is a linear combination of the *zero-state* denoted by $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and the *one*-state denoted by $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$. At any one moment, n qubits can represent 2^n states, while n classical bits can represent only 1 state. This is due to the quantum phenomenon called superposition.

Equation 1: The general statevector of any qubit:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where α and β are complex numbers and the Born Rule ($\alpha^2 + \beta^2 = 1$) applies

α and β are the amplitudes, or coefficients, that govern the probability of the zero and one states. By the Born Rule, the square of the amplitude yields the probability.

Equation 2: The statevector of a qubit in equal superposition:

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

By squaring the amplitude ($\frac{1}{\sqrt{2}}$), we get ½ = 50%, implying that we would measure either $|0\rangle$ or $|1\rangle$ on a purely random basis.

We can operate with more than one qubit in a quantum register. Multiple qubits are represented as tensor products. Two qubits can also be entangled, which means that measuring one qubit's state collapses the superposition of the other qubit. In simpler terms, the two qubits are quantum mechanically interrelated. By controlling one qubit, one can manipulate the other qubit as well. Quantum entanglement is a very significant property that enables extreme parallelism.

Qubits are housed in a quantum circuit. We can manipulate and transform the qubits' states by utilizing quantum gates, which are represented by unitary matrices. All quantum gates except the measurement gate are reversible in contrast to the classical gates; the only reversible gate in common in the NOT gate. The Hadamard gate enables qubits to be in superposition and is thus important and frequently used. The Controlled Not (CNOT) gate or any other controlled gate acts upon a certain qubit (target) if and only if the control qubit acted like a trigger. Once qubits are in superposition, the controlled gates can be used to entangle those qubits. For example, in the Bell State circuit (in the blue box):

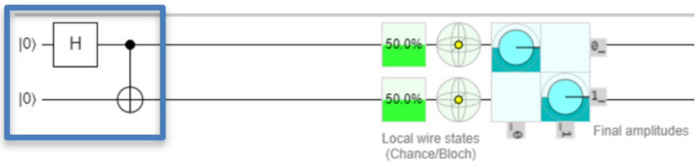


Figure 1. Bell State Quantum Circuit

Equation 3: First Bell Quantum Statevector

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

Notice that both qubits in the circuit shown in Figure 1 do not have any vector representing the state because they are entangled.

Clustering Algorithms

K-means

K-means is a classical partitional clustering algorithm, so it relies on detecting the proximity or distance between data points. K-means follows the four steps: initialization, assignment, update, and termination. First, the algorithm picks k data points as the initial centroids. Then, in the assignment stage, the algorithm calculates distances between each data point and each centroid. Based on the closest centroid distance, the algorithm assigns each data point to the corresponding cluster. For the update portion, the algorithm recalculates the location of each centroid via the mean of the newly assigned points. The assignment and update stages keep alternating until the termination criteria—that none of the cluster centroids move—is achieved.

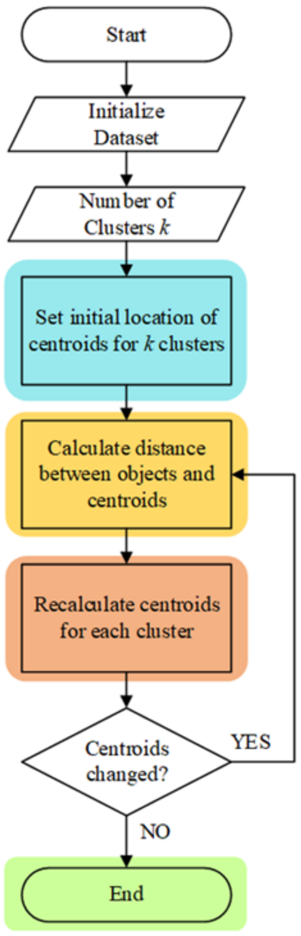

Start
Initialize Dataset
Number of Clusters k
Set initial location of centroids for k clusters
Calculate distance between objects and centroids
Recalculate centroids for each cluster
Centroids changed?
YES
NO
End

Figure 2. Flowchart for the K-means Clustering algorithm

There are four main disadvantages of K-means. The first is the Curse of Dimensionality. As the dimensionality of the data increases, the volume of the data space increases exponentially, and the data points become sparser and more dissimilar. As a result, it takes significantly more resources and time to determine the clusters. Besides, the assignment phase of the clustering algorithm is the slowest part because the process of calculating the distance between each data point and each centroid is done serially, not parallelly. Therefore, the time taken to execute the assignment phase is $O(kn)$. In contrast, the time taken to execute the centroid update phase is $O(k)$. As a result, the entire K-means has a large time complexity of $O(tknd) \approx O(kn^2d)$, which is another disadvantage of the algorithm. Third, K-means uses the Euclidean distance, which is not "meaningful" for higher dimensions. Instead, people prefer the Manhattan distance. The fourth issue is that one must specify the number of clusters ahead of time. However, one can maneuver this issue by using the elbow method detailed in [9].

Q-means

To counter the classical approach, Kerenidis et. al. proposed the Quantum K-means (AKA Q-means) algorithm. [2] Generally, there are three main aspects to any quantum partitional-clustering algorithm: input encoding, proximity determination, and measurement. Although there can be various methods of encoding the input, one common method is symmetric mapping. Their process of encoding first calculates the rotation angles classically based on the minimum, maximum, and current values in the dataset. Then, U3 gates use those rotation angles to encode the data. After the data is encoded, the SWAP Test is used to estimate the proximity between two vectors. The SWAP Test is relatively cheap to implement: first a Hadamard gate on an ancilla, then a Fredkin (Controlled SWAP) gate controlled by the ancilla, and then another Hadamard gate again on the ancilla.

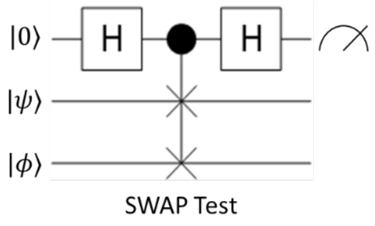


Figure 3. Quantum Circuit Diagram for the SWAP Test

The SWAP Test does not calculate the Euclidean distance. Instead, it finds fidelity of quantum states via the square of the inner product of two vectors. Thus, the SWAP Test is somewhat similar to the cosine distance as described by Figure 4 below.

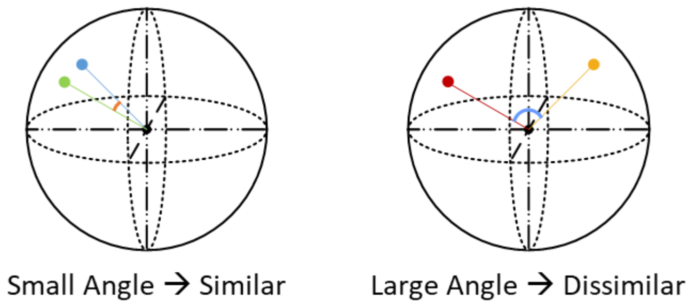


Figure 4. Demo Depiction of the Fidelity between Quantum Statevectors

The one drawback of the SWAP Test is that it needs many iterations to find the "probability that one state will pass a test to identify as the other" [10]. Implementation-wise, it could be slow to run the SWAP Test.

The Q-means clustering algorithm does provide a significant speedup, assuming there is a viable Quantum Random Access Memory (QRAM). However, because today we do not have a QRAM, implementing the Q-means does not provide many benefits.

Thus, the goal of this project is to leverage quantum computation for improving a classical clustering algorithm in order to increase the goodness of clustering and/or reduce the overall execution time. Additionally, the algorithm should have some advantage even when implemented on near-term quantum computers (i.e., without QRAM).

Methods

Any new algorithm should address at least one issue with the classical K-means clustering algorithm and provide a notable improvement in the accuracy or computational speed. In addition, the implementation of the enhanced Q-means algorithm must attempt to resolve the issues—reduced accuracy (in certain circumstances) and long iteration time—in the existing Q-means implementation. Lastly, the new Q-means algorithm should have *some* benefit—clustering accuracy or speedup—even when not using a Quantum Random Access Memory (QRAM). The following two clustering algorithms tackle the assignment and termination steps in the K-means.

An Oracle-based Approach to Calculate the Manhattan Distance

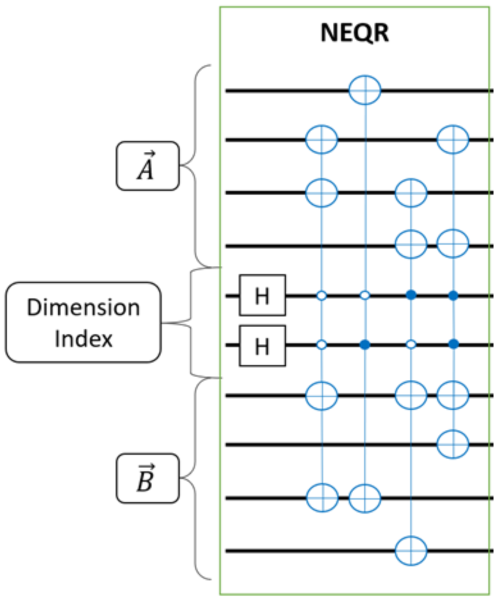
As mentioned above, the Euclidean Distance is not “meaningful” for high-dimensional (> 3 features) data. Instead, scientists often work with the Manhattan Distance, which is the sum of the absolute differences between two vectors. Also, mathematically, the formula of the Manhattan Distance is less computationally expensive than the Euclidean Distance, making it a prime target to use.

Like many algorithms, an oracle can be used to perform certain function—in this case, calculate the Manhattan Distance. An oracle is essentially a black box function. The inputs and outputs of the oracle are known, but the inside mechanism needs to be figured out. Another key feature of an oracle is that it performs one or more operations on a qubit independently. In other words, there can be no operation that depends on multiple qubits. An oracle is often repeated multiple times within a quantum circuit. The benefits of an oracle are often tied to a significant computational speedup, depending on how the oracle is built.

Encoding Classical Data into the Quantum Space

The first part of this algorithm is the encoding method. Each d-dimensional datapoint is encoded into d qubits via the Novel Enhanced Quantum Representation algorithm [12]. The NEQR is only advantageous if there is a QRAM at hand. Each Cartesian coordinate is represented by the basis states within the qubits, so the qubits are essentially being treated as bits. In the quantum circuit, the controlled-NOT gates are used to flip the initial $|0\rangle$ state, and the control qubits of these gates are linked to the respective dimension, which are also encoded in binary. The qubits labeled by “dimension index” are reused for both qubit registers $\vec{A}$ and $\vec{B}$.

Figure 5. Quantum Circuit Diagram for the Novel Enhanced Quantum Representation applied to two datapoints (A and B) along with the dimension index qubits



Designing the Oracle

Suppose there are two points, $A = [A_0\ A_1\ A_2\ A_3]$, which is a centroid, and $B = [B_0\ B_1\ B_2\ B_3]$, which is a data point. The end goal is to find the Manhattan distance between points A and B ($|A_0 - B_0| + |A_1 - B_1| + |A_2 - B_2| + |A_3 - B_3|$).

Let I be the dimension index quantum register that tracks the values per dimension for both vectors. In this demonstration that assumes Little Endian notation, I is the following: $[00\ 10\ 01\ 11]_{bin} = [0\ 1\ 2\ 3]_{dec}$. By treating qubits like bits, and encoding the vectors using the Novel Enhanced Quantum Representation (NEQR) algorithm, the following equation would be the resultant quantum statevector, (the input to the oracle).

Equation 6: Full Quantum State of the Encoded Vector

$$|IAB\rangle = \alpha|00, A_0, B_0\rangle + \beta|10, A_1, B_1\rangle + \cdots$$

The output should yield us the Manhattan distance in some way, preferably by manipulating the qubit register shown above. Because the Manhattan distance is the sum of absolute value of the difference between two points in each dimension, the result should look like the following:

Equation 7: Full Quantum State of the Intended Vector

$$|IAB\rangle = \alpha|00, |A_0 - B_0|, -B_0\rangle + \beta|10, |A_1 - B_1|, -B_1\rangle + \cdots$$

The purpose of this oracle, as mentioned earlier, is to calculate the Manhattan distance (more specifically, going from the Encoded Vector to the Intended Vector).

Since the Manhattan distance is the sum of the absolute differences of two vectors, it is necessary to subtract point B from point A. Mathematically, $A - B = A + (-B)$. By taking the two's complement on point B, we can first revert its sign. Then, we can use a quantum adder circuit to add the two quantum registers containing points A and B.

Equation 8: New Quantum State after executing the two's complement on B

$$|IAB\rangle = \alpha|00, A_0, -B_0\rangle + \beta|10, A_1, -B_1\rangle + \gamma|01, A_2, -B_2\rangle + \delta|11, A_3, -B_3\rangle$$

Equation 9: New Quantum State after executing the quantum adder

$$|IAB\rangle = \alpha|00, A_0 - B_0, -B_0\rangle + \beta|10, A_1 - B_1, -B_1\rangle + \gamma|01, A_2 - B_2, -B_2\rangle + \delta|11, A_3 - B_3, -B_3\rangle$$

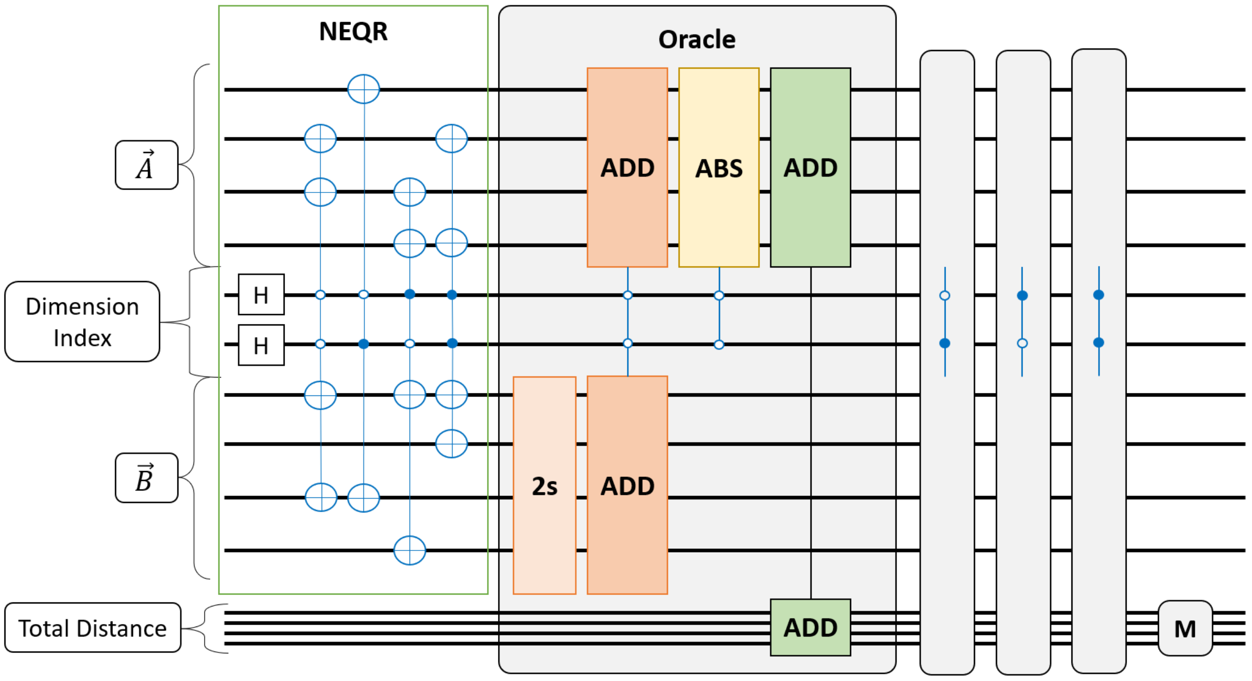
The final operation necessary to calculate Manhattan distance is the absolute value, which can be done via a quantum comparator circuit. Specifically, the circuit would apply a 2's complement on the outgoing result if and only if the value was less than 0.

Equation 10: New Quantum State after executing the quantum comparator (which is the intended vector)

$$|IAB\rangle = \alpha|00, |A_0 - B_0|, -B_0\rangle + \beta|10, |A_1 - B_1|, -B_1\rangle + \gamma|01, |A_2 - B_2|, -B_2\rangle + \delta|11, |A_3 - B_3|, -B_3\rangle$$

The oracle is repeated d times, and each time it is executed, the Manhattan distance keeps accumulating in the "Total Distance" quantum register. The last operation of the oracle is a second quantum adder circuit that adds the oracle's output to the total calculated distance. An overflow qubit should be assigned just in case the Manhattan distance exceeds the maximum representable number. Finally, a series of measurement gates should be applied on the "Total Distance" register to retrieve a bit string, which can then be easily converted into the decimal system.

Figure 6. Quantum Circuit Diagram for the complete algorithm, starting from the NEQR encoding, the oracle, and then the measurement of the total Manhattan distance



Time Complexity Analysis

This entire oracle-based approach is only a quantum subroutine in the entire Q-means algorithm. All other steps are done classically. Thus, the overall time complexity of this algorithm with QRAM is $O(tskd)$. Since the number of iterations t is directly proportional to the number of data points n, the time complexity simplifies to $O(n)$, which is linear.

Utilizing Angle Embedding and a Generalized SWAP Test

Angle Embedding

Because a hard-coded data encoding approach does not work well for many datasets, it is essential to adapt a dynamic encoding strategy that yields more consistent and better results. The feature map is known to give quantum machine learning models much more flexibility as the high-dimensional data can be mapped into the Hilbert Space easily. However, as noted by the previous approach, feature maps, like variational quantum algorithms, need to be trained, so there must be a dataset with corresponding labels. The problem is that there are no labels for unlabeled data. Instead, an Angle Embedding approach is taken to encode the classical data into the quantum space, mimicking the effect of a feature map. Each component of the data point is encoded into the qubit statevector. Because every qubit starts at $|0\rangle$, rotating the statevector in the Bloch Sphere by a datapoint-dependent angle can encode the data effectively.

Generalized SWAP Test

The SWAP Test was generalized for finding the proximity between two d-dimensional vectors and tailored to work with the angle embedding feature map. Because each dimension of each data point is encoded into a qubit, applying the standard SWAP Test was quite intuitive. The SWAP Test, as noted earlier, is a simple yet powerful means of estimating the proximity between two quantum states. The quantum subroutine is probabilistic and thus needs many shots to estimate the closeness. For every dimension, one SWAP Test was required, so for d dimensions, the algorithm requires d SWAP Tests. Therefore, the total number of qubits equals $3d$ and d classical bits to store the measurement results of the quantum circuit.

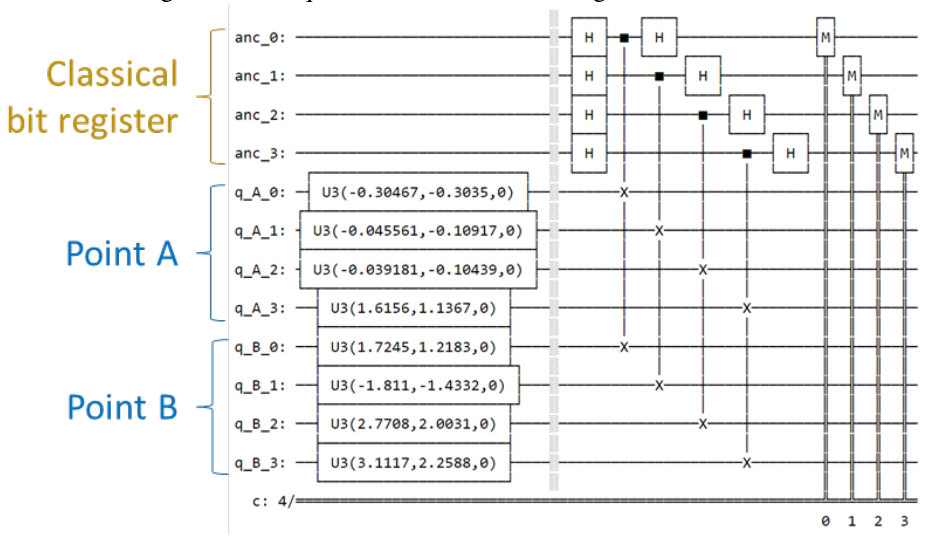
Dynamic Termination Criterion

The last feature of the algorithm is a dynamic, threshold distance-based termination criterion. This allows the Q-means algorithm to converge faster as it gives more leniency to the model. Essentially, the algorithm can terminate if the centroids of each cluster does not move beyond a certain threshold distance. As observed in the execution of the previous Q-means implementation, the number of iterations/epochs was too high without any noticeable benefit in goodness of clusters. Terminating earlier could help reduce the total execution time of the algorithm without sacrificing significant accuracy. Instead of taking about ten iterations for every dataset, the newer Q-means took about five iterations on an average.

Time Complexity Analysis

Without QRAM, the time complexity of this new approach is $O(tsknd)$, which is approximately quadratic. Note that s is the number of shots. Although it is the same time complexity as K-means, the new Q-means provides better clustering accuracy than K-means. If we now assume that we have a QRAM and an unlimited number of qubits, we can virtually run all the proximity checks for all the data points at the same time. The QRAM would store the pre-encoded data in the qubits' statevector, which could then be used easily by the SWAP tests. Then, by assuming that QRAM is available as an oracle with constant runtime, the time complexity would reduce to $O(tskd)$, which is linear. However, the total time complexity of the existing Q-means algorithm is $O(ts\alpha k^2 d(\log n)^p)$, where α and p are constants. This is more than the newly proposed Q-means with $O(tskd)$, which is linearly complex.

Figure 7. Quantum Circuit Diagram for the quantum distance-estimating subroutine within in the new Q-means

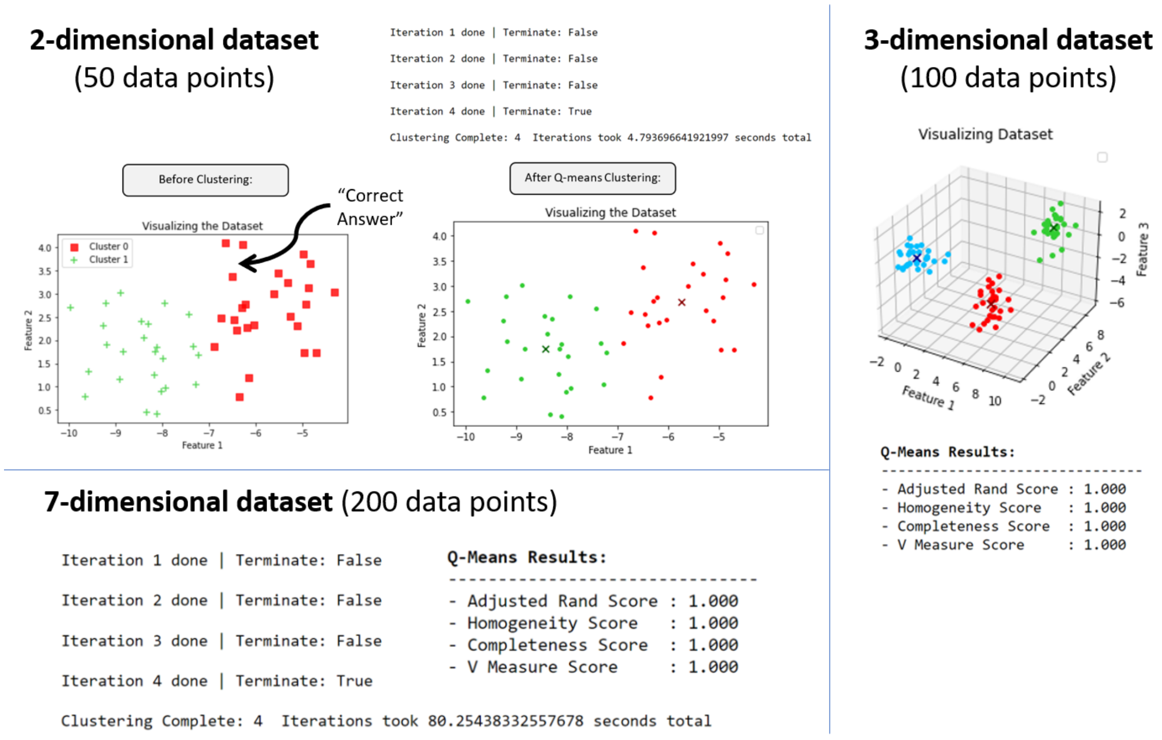


Results

After implementing this algorithm using the IBM Qiskit SDK, the new Q-means algorithm was run several times on different dimensional datasets. Goodness of clustering was measured via different metrics: Adjusted Rand Index (ARI), Silhouette Score, Homogeneity Score, Completeness Score, and V-Measure Score.

Here is a brief summary of the metrics. The Adjusted Rand Index or ARI is a measure of the similarity between two data clusters, and it is often loosely correlated with the clustering "accuracy." ARI values range from 0 to 1, and they are independent of the cluster labels. The Silhouette Score calculates the goodness of a clustering algorithm, and its value ranges from -1 to 1. The closer to 1, the more distinguishable and correct the clustering is. A high Homogeneity Score implies that all or most clusters contain data points belonging to only one class. On the other hand, a high Completeness Score implies that all or most data points belonging to a certain class are all elements of the same cluster. The V-Measure Score is simply the harmonic mean of the homogeneity and completeness scores.

Figure 8. Several snapshots of Clustering by the new Q-means algorithm for different dimensional datasets.



```
Iteration 1 done | Terminate: False

Iteration 2 done | Terminate: False

Iteration 3 done | Terminate: False

Iteration 4 done | Terminate: True

Clustering Complete: 4  Iterations took 4.793696641921997 seconds total
```

```
Q-Means Results:
-------------------------------
- Adjusted Rand Score : 1.000
- Homogeneity Score   : 1.000
- Completeness Score  : 1.000
- V Measure Score     : 1.000
```

```
Iteration 1 done | Terminate: False

Iteration 2 done | Terminate: False

Iteration 3 done | Terminate: False

Iteration 4 done | Terminate: True

Clustering Complete: 4  Iterations took 80.25438332557678 seconds total
```

```
Q-Means Results:
-------------------------------
- Adjusted Rand Score : 1.000
- Homogeneity Score   : 1.000
- Completeness Score  : 1.000
- V Measure Score     : 1.000
```

Table 1. Clustering Performances of K-means and the new Q-means implementations based on various metrics

Clustering Metrics	K-Means	New Q-Means
Adjusted Rand Index	0.843	0.996
Silhouette Score	0.690	0.786
Homogeneity Score	0.888	0.995
Completeness Score	0.933	0.995
V Measure Score	0.908	0.995

As can be observed, the new Q-means algorithm is about 18% better than the K-means, considering the ARI metric. Also, we see that the V-measure score of the new Q-means is about 10% better than that of K-means, which implies that the clusters generated by the new Q-means were more distinguishable in general. Considering all the five metrics, the new Q-means is about 12% better than the K-means, in terms of goodness of clustering.

As far as the execution time, the new Q-means implementation was about six times faster than the existing Q-means implementation due to a multi-threaded implementation and faster convergence. However, the Q-means execution time was nowhere near the execution time of K-means. This is primarily due to running a quantum program on a simulation (on a classical computer). For every qubit, the simulation uses double the RAM, which would inherently slow down the total time taken. In addition, the K-means algorithm utilizes classical RAM, while the current Q-means implementation does not use QRAM. So, evaluating K-means with RAM and Q-means without QRAM is an unfair comparison in terms of execution time and time complexity. However, comparing the new and old Q-means implementations, which both run on the same quantum simulation, is a fair comparison.

Table 2. Clustering Performances of the old Q-means and new Q-means based on various metrics; one sample from several runs executed. The dataset used contained 100 data points, 3 features, and 4 clusters.

Clustering Metrics	**Old Q-Means**	**New Q-Means**
Adjusted Rand Index	0.481	0.973
Silhouette Score	0.191	0.641
Homogeneity Score	0.554	0.969
Completeness Score	0.628	0.970
V Measure Score	0.589	0.970
Number of Iterations	10	5
Time of execution	267.158	46.587
Time per iteration	26.716	9.317

As can be observed, the ARI of the new Q-means implementation is about 102% better than that of the old Q-means. The V-measure score also is about 65% better for the new Q-means than for the old Q-means. Looking at the total time of execution, the new Q-means was about 5.7 times faster. In addition, the number of iterations was half for the new Q-means, demonstrating a faster convergence. Thus, the average time per iteration for new Q-means was about 2.9 times faster than that of the old Q-means.

Discussion

Below are the key technical takeaways and major lessons learned from this project. First and foremost, the SWAP Test is very effective proximity measure subroutine, but it takes many iterations. It is also relatively cheap to implement in a quantum computer, which translates to reduced noise when executing the algorithm. The second note is about the data encoding. Angle Embedding is more efficient and feasible than the NEQR-based encoding as angle embedding does not require any QRAM. However, the caveat associated with angle embedding is the practical limit of the precision of the angle of a qubit. The more features, the smaller the qubit rotation would become. Even if qubits could be initialized at that state, it would be extremely difficult to retain that state and perform meaningful calculations without errors. Also, oracles can be used to calculate distances. As shown in first approach, the oracle was designed to calculate the Manhattan distance to work better with high-dimensional data. Lastly, the threshold distance-based termination criterion reduces number of iterations and helps Q-means algorithm converge faster.

It is also important to note that the second approach works without QRAM, making it ready to use in Noisy Intermediate Scale Quantum (NISQ) computers. This means that we can use the new Q-means for clustering high-dimensional with near term computers. With just a 127-qubit computer that already exists, we can cluster 42-dimensional data already. However, we need to see how noise in quantum computers affects the accuracy of the clustering.

Conclusion

A novel enhanced quantum clustering algorithm was developed so that it works more efficiently with high-dimensional data. The new Q-Means utilizes an Angle Embedding Feature Map, a generalized SWAP Test, and a dynamic threshold-based termination criterion. In essence, the angle embedding allows efficient encoding of classical data into the quantum space. The generalized SWAP Test is specifically tailored to the angle embedding method. Finally, the special termination criterion helps reduce the number of total iterations taken by the algorithm. The good thing about this second approach is that it does not need QRAM to function. However, adding it can be useful to reducing the time complexity. But, as can be observed from the results, the new Q-means, regardless of QRAM, is generally more accurate than K-means. Assuming QRAM is available and has a constant runtime, the new Q-Means is linearly complex, while K-means is quadratically complex. In addition, the implementation for the new Q-Means is about six times faster than that of the existing Q-means.

On a separate note, the first approach for the Q-means, which uses an oracle to calculate the Manhattan Distance, is efficient as well in that its time complexity is $O(n)$, which is better than the existing Q-means complexity of $O(n(\log n)^p)$. However, this approach needs a QRAM for the encoding to work.

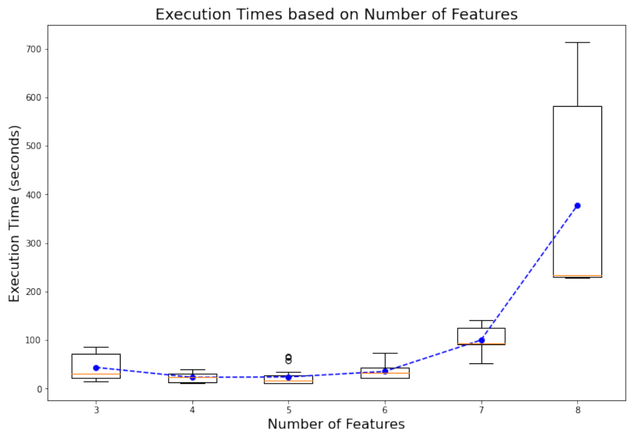
The new Q-means algorithm proposed in this paper has significant improvements in some aspects over existing classical and quantum clustering algorithms. Whether the algorithm is robust to noise in quantum computers and its performance on high-dimensional data with and without QRAM are topics of future research and development.

Limitations

I was not able to test the Q-means algorithm with QRAM to verify that it achieves a higher or lower clustering accuracy. I was also unable to implement the existing Q-means algorithm with QRAM. Also, I was unable to visually verify the clusters for high-dimensional data (beyond 3 dimensions). So, I only relied on the clustering metrics such as the Adjusted Rand Index, V-measure score, and Silhouette Score. Given the availability restrictions, I was not able to run my algorithm on any large server. Instead, I had to run all algorithms on my desktop computer, which only has an 8-core CPU, not a GPU.

Reconsidering the simulation execution time, an exponential trend (shown below) is observed. Due to the exponential RAM needed for simulation, as the number of features increases, the number of qubits increases linearly. This causes the amount of RAM to grow exponentially.

Figure 9: Simulation Execution Time vs. Number of Features in the Data



Acknowledgements

I would like to thank my external advisor, Mr. Clapis, for providing guidance and answering many questions, my internal advisor, Mrs. Bhattacharya, for reviewing my work and presentations, and my parents for their encouragement and support.

References

[1] Aimeur, E., Brassard, G., & Gambs, S. (2012, August 31). Quantum Speed-up for unsupervised learning - machine learning. SpringerLink. https://doi.org/10.1007/s10994-012-5316-5

[2] Kerenidis, I., Landman, J., Luongo, A., & Prakash, A. (2018, December 12). *Q-means: A quantum algorithm for unsupervised machine learning*. arXiv. https://doi.org/10.48550/arXiv.1812.03584

[3] Li, J., & Kais, S. (2021, June 13). Quantum cluster algorithm for Data Classification. arXiv. https://doi.org/10.48550/arXiv.2106.07078

[4] MJV Team. (2021, September 23). *Types of clustering: Why is it so important for business?* MJV Technology & Innovation. Retrieved February 29, 2022, from https://www.mjvinnovation.com/blog/types-of-clustering/

[5] Neukart, F., Dollen, D. V., & Seidel, C. (2018, June 14). Quantum-assisted cluster analysis on a quantum annealing device. Frontiers. https://doi.org/10.3389/fphy.2018.00055

[6] Nielsen, M. A., & Chuang, I. L. (2018). Quantum Computation and Quantum Information: 10th anniversary ed. Cambridge University Press.

[7] Pakhira, M. K. (2014). *A linear time-complexity K-means algorithm using cluster shifting*. IEEE Xplore. https://doi.org/10.1109/CICN.2014.220

[8] Preskill, J. (2021, June 25). Quantum Computing 40 years later. arXiv. https://doi.org/10.48550/arXiv.2106.10522

[9] Wikipedia Staff. (2022, April 26). *Elbow method (clustering)*. Wikipedia. Retrieved March 13, 2022, from https://en.wikipedia.org/wiki/Elbow_method_(clustering)

[10] Wikipedia Staff. (2022, February 10). *SWAP Test*. Wikipedia. Retrieved February 9, 2022, from https://en.wikipedia.org/wiki/Swap_test

[11] Xie, X., Duan, L., Qiu, T. et al. (2021, July 30). Quantum algorithm for MMNG-based DBSCAN. Nature. https://doi.org/10.1038/s41598-021-95156-7

[12] Zhang, Y., Lu, K., Gao, Y., & Wang, M. (2013, March 26). *NEQR: A novel enhanced quantum representation of digital images*. SpringerLink. https://doi.org/10.1007/s11128-013-0567-z