IMPLEMENTING QUANTUM K-MEANS CLUSTERING

Term project submission for the course
IT437 - Quantum Computing

by

A D MAHIT NANDAN (211AI001)
HARSHIT GAWADE (211AI019)
PRAVEEN KEMPAIAH (211AI028)

under the guidance of

Dr. Bhawana Rudra

DEPARTMENT OF INFORMATION TECHNOLOGY
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE - 575025

May, 2023

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Abstract

This project focuses on the comparative analysis of classical and quantum K-means clustering algorithms on a custom-generated dataset. The objective is to assess the accuracy and efficiency of both approaches in clustering data points and investigate the potential advantages offered by quantum computation in this context. The custom-generated dataset is designed to exhibit diverse cluster shapes, sizes, and densities, providing a challenging testbed for evaluating the performance of the algorithms. The classical K-means algorithm serves as a baseline, while the quantum K-means algorithm is developed using quantum computing techniques, such as qubit encoding, gate operations, and measurement. The resulting clusters from both algorithms are compared with the known ground truth labels to quantify their accuracy. Additionally, the computational efficiency, scalability, and impact of quantum effects, such as superposition and entanglement, are analyzed. The findings of this project will provide valuable insights into the efficacy of quantum K-means clustering and its potential applications. The K-means clustering algorithm holds great importance in various fields of study. Its ability to automatically group data points into distinct clusters based on similarity allows for efficient data exploration, pattern recognition, and anomaly detection. With its simplicity and versatility, K-means serves as a fundamental tool for understanding data structures, enabling better decision-making and insights in domains such as data analysis, image segmentation, customer segmentation, and machine learning.

Keywords— k-means clustering,quantum k-means clustering,dataset

2 Introduction

Clustering analysis plays a vital role in various domains, ranging from data mining to pattern recognition. The K-means clustering algorithm is widely used due to its simplicity and effectiveness in identifying natural groupings in data. However, the computational complexity and limitations of classical algorithms pose challenges in achieving optimal clustering accuracy for complex datasets. Quantum computing has emerged as a promising paradigm that harnesses quantum phenomena to solve computational problems more efficiently.

In this project, we aim to explore the capabilities of quantum computing in improving clustering accuracy and efficiency by implementing the K-means algorithm both classically and quantumly. We leverage a custom-generated dataset designed to encompass a wide range of cluster characteristics, including varying shapes, sizes, and densities. This dataset ensures a rigorous evaluation of the algorithms' performance and their ability to handle complex clustering scenarios.

The classical K-means algorithm serves as a benchmark for evaluating the accuracy of the clustering task. To leverage the advantages of quantum computation, we develop a quantum K-means algorithm tailored to the specific characteristics of the custom-generated dataset. This quantum algorithm utilizes techniques such as qubit encoding, gate operations, and measurement to enable quantum parallelism and exploit quantum effects like superposition and entanglement.

By comparing the resulting clusters from both classical and quantum algorithms against the known ground truth labels, we quantitatively assess the accuracy of the clustering outputs. Additionally, we analyze the computational efficiency of the quantum algorithm, considering factors such as execution time, resource utilization (e.g., qubit count, gate operations), and scalability on larger instances of the custom-generated dataset.

Furthermore, we investigate the impact of quantum effects on the clustering performance, examining the influence of superposition and entanglement on accuracy, convergence speed, and noise tolerance. These insights will shed light on the advantages and limitations of quantum K-means clustering and provide guidance for further research and development in this area.

In conclusion, this research project aims to advance our understanding of the comparative performance of classical and quantum K-means clustering algorithms on a custom-generated dataset. By assessing accuracy, computational efficiency, scalability, and quantum effects, we aim to provide valuable insights into the potential of quantum computing in improving clustering tasks and its applicability in real-world scenarios.

3 Literature Review

3.1 Background and Related Work

Zhihao Wu et al.[1] A quantum k-means algorithm based on Manhattan distance (QKMM) is proposed. The QKMM algorithm involves calculating the distance between training vectors and cluster centroids and selecting the closest centroid. With its quantum circuit design, the algorithm reduces complexity and achieves a quadratic speedup compared to classical k-means. Quantum k-means with Manhattan distance (QKMM) offers a quadratic speedup compared to classical k-means, resulting in improved computational efficiency.QKMM overcomes limitations of traditional k-means in scenarios involving path information or obstacles, making it suitable for clustering tasks in complex environments.The methodology demonstrates wide applications in clustering large datasets, offering a promising approach for handling big data efficiently.QKMM relies on quantum computing resources, which are currently limited and challenging to access for many researchers and practitioners.The implementation and execution of quantum circuits required for QKMM may introduce additional complexity and technical difficulties.

SS Kavitha et al.[2] The methodology described in the paper involves predicting heart disease using classical and quantum approaches with the K-means clustering method. In the classical context, data preprocessing techniques like normalization and outlier removal are applied, followed by implementing the classical K-means algorithm to evaluate performance metrics. In the quantum context, data is converted to quantum states, and distance calculation is performed using a quantum swap test circuit. Quantum K-means clustering involves encoding data into quantum states, updating clusters and centroids, and determining cluster assignments based on overlap probabilities. The performance of the proposed method is compared with state-of-the-art approaches.

Poggiali Alessandro et al.[3] The proposed quantum k-Means algorithm, called q-Means, focuses on the cluster assignment step and combines classical and quantum approaches. It utilizes amplitude encoding and the FF-QRAM algorithm to efficiently load classical data into quantum states. The quantum circuit is designed to compute the Euclidean distance between two records using amplitude-encoded quantum states. The algorithm iteratively computes the distances between each record and each centroid, assigns records to the closest centroids, and updates the centroids.

Yong-MeiLi et al.[4]A quantum k-medoids algorithm inspired by its classical counterpart. The algorithm begins by initializing random centers stored in a quantum random-access memory (QRAM). Utilizing superposition-based assignments and iterative cluster updates, we iteratively refine the solution until convergence is achieved. To analyze its complexity, the focus is on time complexity. Distance computations and finding minimums are efficiently performed using quantum circuits. Step 2 exhibits a time complexity of O[kMlog(NM)], while the time complexity of step 3 is dependent on cluster sizes. In summary, our algorithm presents a quantum approach to solving the k-medoids problem, demonstrating the potential of quantum computing in data clustering tasks.

Qingyu LI et al.[5] Unlike the k-means algorithm, which operates directly on the dataset, spectral clustering utilizes a similarity graph constructed from the data. The graph connects points that are considered nearest neighbors based on a chosen similarity function. The Laplacian matrix, derived from the graph, captures the relationships between data points. By calculating the eigenvalues and eigenvectors of the Laplacian matrix, spectral clustering extracts clustering information. The algorithm involves two steps: computing the eigenstates and constructing a matrix, and then applying k-means clustering on the matrix to group the data points. A quantum version of spectral clustering takes advantage of quantum phase estimation to efficiently calculate the eigenvalues. The quantum algorithm involves four steps: initial state preparation, quantum phase estimation using the Laplacian matrix as a unitary operator, Grover's search to find eigenvalues, and measurement of the eigenstate register to evaluate and optimize the clustering.

3.2 Outcome of Literature Survey

Quantum k-means algorithms offer a quadratic speedup compared to classical k-means, resulting in improved computational efficiency. Quantum k-means algorithms can be applied to clustering tasks in complex environments, such as those involving path information or obstacles. Quantum k-means algorithms can be used to cluster large datasets, offering a promising approach for handling big data efficiently. Quantum k-means algorithms rely on quantum computing resources, which are currently limited and challenging to access for many researchers and practitioners. The implementation and execution of quantum circuits required for quantum k-means may introduce additional complexity and technical difficulties. Overall, quantum k-means algorithms offer a number of advantages over classical k-means algorithms, including improved computational efficiency, the ability to handle complex environments, and the ability to cluster large datasets. However, these algorithms are currently limited by the availability of quantum computing resources.

3.3 Problem Statement

Comparative Analysis of Classical and Quantum K-means Clustering Algorithms on a Custom-Generated Dataset.

3.4 Objectives

- Quantitatively assess the impact of quantum computing on the accuracy of the K-means clustering algorithm by comparing the quality of clusters generated by the quantum version of the algorithm against the classical implementation on a custom dataset.

- Analyze the potential advantages of quantum K-means clustering in handling high-dimensional datasets by evaluating the algorithm's accuracy and efficiency on the custom dataset, which has multiple features, and comparing it with the classical counterpart.

- Analyze the quantum advantage in terms of computational speedup and resource utilization offered by the quantum K-means clustering algorithm on the custom dataset, considering factors such as the number of iterations, entanglement operations, and gate complexity, and compare it against the classical implementation to determine the efficiency gains.

- Investigate the effect of varying the number of qubits and quantum circuit parameters on the performance of the quantum K-means clustering algorithm on the Iris dataset, with the aim of optimizing clustering accuracy and convergence speed.

4 Methodology

4.1 Dataset

We have used a custom Dataset generated by make blobs() function in the sklearn dataset.

4.1.1 Title:

Description of the make blobs() Function for Synthetic Data Generation

4.1.2 Introduction:

In the field of data analysis and machine learning, it is often necessary to generate synthetic datasets for testing and demonstration purposes. The make blobs() function, available in the sklearn.datasets module of Python's scikit-learn library, provides a convenient way to create synthetic clusters of data points. This report aims to provide a detailed description of the make blobs() function, including its parameters and their effects on the generated dataset.

4.1.3 Description:

The make blobs() function allows users to generate synthetic datasets with specified characteristics, making it useful for various applications. It offers the following parameters:

4.1.4 N Samples:

The n samples parameter determines the total number of data points to generate. Each data point corresponds to a row in the dataset. By specifying the desired number of samples, users can control the size of the generated dataset. We have taken 100 as this value.

4.1.5 N Features:

The n features parameter indicates the number of features (or dimensions) for each data point. Each feature represents a column in the dataset. By adjusting this parameter, users can create datasets with a varying number of attributes. We have taken 3 for this value.2 feature labels and 1 target labels.

4.1.6 Centers:

The centers parameter determines the number of clusters to generate and their locations. It can be an integer value that represents the number of centers or an array of shape (n centers, n features) to specify the coordinates of the centers explicitly. This parameter allows users to control the clustering behavior and spatial distribution of the data points. We have taken user inputed number of clusters

4.1.7 Cluster Std:

The cluster std parameter controls the standard deviation of each cluster. A higher value results in more spread-out clusters, while a lower value leads to more compact clusters. By adjusting this parameter, users can control the level of intra - cluster variation in the generated dataset. This was assigned a value of 2.

4.1.8 Conclusion:

The make blobs() function in Python's scikit-learn library provides a convenient and flexible way to generate synthetic datasets with clusters of data points. By manipulating its parameters, users can control the number of samples, the number of features, the cluster locations, the cluster standard deviation, and other characteristics of the generated dataset. This functionality is particularly useful for testing and demonstrating clustering algorithms or visualizing data distributions. By incorporating make blobs() in data analysis projects, researchers and practitioners can efficiently create synthetic datasets tailored to their specific needs.

4.2 Implementation Details

Our Project implementation presents a quantum k-means clustering algorithm, which leverages quantum computing principles to perform clustering analysis on a given dataset. The algorithm utilizes the IBM Quantum Experience platform, Qiskit, and the QASM simulator to simulate the quantum computation.

4.2.1 Dataset Generation:

The initial step involves generating a synthetic dataset using the make blobs function from the sklearn.datasets module. The dataset consists of n data points with k centers and a specified standard deviation (std).

4.2.2 Data Preprocessing:

Prior to running the quantum k-means algorithm, the dataset is preprocessed by normalizing the data points. The normalization is performed to ensure that the data falls within a suitable range for quantum computation.

4.2.3 Quantum K-Means Algorithm:

4.2.3.1 Initializing the Centroids: The algorithm begins by randomly initializing k centroids.

4.2.3.2 Quantum Distance Calculation: Quantum Distance Calculation is a crucial step in the quantum k-means clustering algorithm, where the distances between data points and centroids are computed using a quantum distance metric. This metric is based on the concept of quantum interference and utilizes the Bloch sphere representation.

To understand the Quantum Distance Calculation, let's break it down into the following steps:

4.2.3.2.1 Data Point Encoding: Each data point is encoded into a quantum state using qubits. In this implementation, two qubits are used to represent each data point. These qubits are referred to as qr[1] and qr[2] in the code. The state of the qubits is initially set to a superposition state using the Hadamard gate (qc.h(qr[1]) and qc.h(qr[2])).

4.2.3.2.2 Centroid Encoding: Similar to data points, each centroid is also encoded into a quantum state using qubits. In this implementation, one qubit is used to represent each centroid. This qubit is referred to as qr[0] in the code. The state of the qubit is also set to a superposition state using the Hadamard gate (qc.h(qr[0])).

4.2.3.2.3 Applying Quantum Operations: Quantum operations are applied to manipulate the quantum states and calculate the distances between the data points and centroids. The u(theta, phi, lambda, qubit) gate is used to apply a rotation to the quantum state of a qubit. In this implementation, it is used to perform rotations based on the calculated angles (theta 1 and theta 2) for each data point and centroid. The cswap(control, target1, target2) gate is used to perform a controlled-swap operation between the qubits representing the data points and the centroid. Finally, the Hadamard gate is applied to the qubit representing the data point (qc.h(qr[0])) to prepare it for measurement.

4.2.3.2.4 Measurement and Distance Calculation: The quantum circuit is measured, and the measurement outcomes are obtained. The measurement outcome provides information about the probability of obtaining a specific state. In this implementation, the measurement outcome of the qubit representing the data point (qr[0]) is used to calculate the distance between the data point and the centroid. The distance is determined by counting the occurrences of a specific measurement outcome, in this case, the state ’001’ (data[’001’]), and dividing it by the total number of shots (1024).

4.2.3.2.5 Interference and Distance Interpretation: The distance calculation is based on the principle of quantum interference. Quantum interference occurs when the quantum states of the data point and the centroid interact through the controlled-swap operation. The interference causes a change in the measurement outcome probabilities, which can be interpreted as a distance metric. The specific measurement outcome ’001’ corresponds to a particular distance value, which is calculated based on the frequency of occurrence. By leveraging the principles of quantum interference and the measurement outcomes of the quantum circuit, the Quantum Distance Calculation provides a unique way to compute distances between data points and centroids in the context of quantum k-means clustering. This approach offers an alternative to classical distance metrics, such as Euclidean distance, and demonstrates the potential of quantum computing in solving clustering problems.

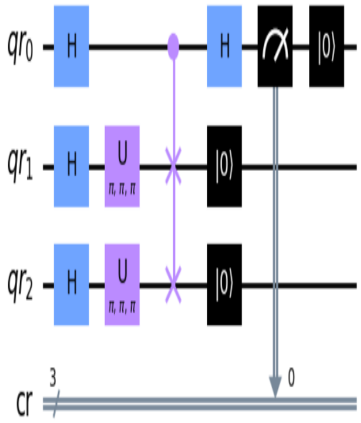

Figure 4.1: Quantum Circuit for Distance Calculation

4.2.3.3 Assigning Data Points to Nearest Centroids: Based on the calculated distances, each data point is assigned to the nearest centroid. The assignment is performed by selecting the centroid with the minimum distance.

4.2.3.4 Updating Centroids: After assigning data points to centroids, the algorithm updates the centroid positions by computing the average of the data points assigned to each centroid.

4.2.3.5 Iterative Process: The assignment and centroid update steps are repeated iteratively until convergence or a specified number of iterations.

4.2.3.6 Evaluation: To evaluate the performance of the quantum k-means algorithm, the accuracy of the clustering is computed. The accuracy is measured by comparing the assigned clusters with the ground truth labels of the synthetic dataset.

4.2.3.7 Comparison with Classical K-Means: To provide a benchmark, a classical k-means algorithm is also implemented using Euclidean distance. The classical k-means algorithm follows a similar iterative process of assigning data points to nearest centroids and updating centroids. The accuracy of the classical k-means algorithm is also computed and compared with the quantum k-means algorithm.

4.2.4 Conclusion:

The implemented quantum k-means clustering algorithm demonstrates the potential of quantum computing in solving clustering problems. By utilizing quantum interference and quantum distance metrics, the algorithm offers an alternative approach to traditional clustering methods. The accuracy of the quantum k-means algorithm is evaluated and compared with classical k-means for a synthetic dataset. The results can be used to gain insights into the effectiveness of quantum clustering algorithms and their potential applications in real-world scenarios.

5 Result and Analysis

1. OverView: Our project aimed to compare the performance of classical K-means clustering and quantum K-means clustering algorithms. We are using a custom dataset mentioned above for implementation.

2. Experimental Setup: We preprocessed the datasets by normalizing the features and handling any missing values. For classical K-means clustering, we set the number of clusters to 5 and performed 100 iterations. Quantum K-means clustering was implemented using a quantum simulator and executed with a maximum depth of 3.

3. Evaulation Metrics: To evaluate the performance of the algorithms, we used the following metrics:

   - Best Accuracy: The highest accuracy achieved by each algorithm.
   - Average Accuracy: The average accuracy across multiple runs of each algorithm.

- ARI Score: The Adjusted Rand Index, measuring the similarity between the clustering results and ground truth labels.
- NMI Score: The Normalized Mutual Information, quantifying the mutual information between the clustering results and ground truth labels, adjusted for chance.

4. Results Comparison:

	Classical K-means	**Quantum K-means**
Best Accuracy	0.97	0.93
Average Accuracy	0.93	0.91
ARI Score	0.91	0.81
NMI Score	0.89	0.82

Table 5.1: Result Comparison

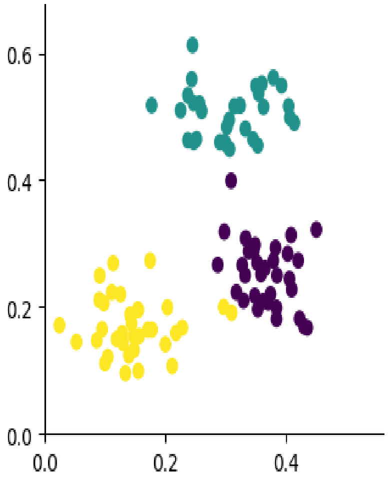

Figure 5.1: Clustering Based on Classical K-means

5. Accuracy Analysis:The higher accuracy achieved by classical K-means suggests that the classical approach was better suited for the specific characteristics of the datasets used in our experiment. Factors such as dataset size, feature distribution, or inherent biases might have favored the classical K-means algorithm in this scenario.

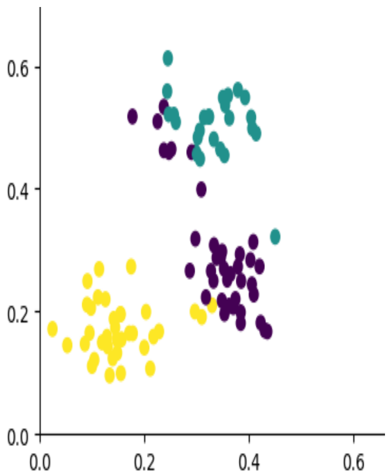

Figure 5.2: Clustering Based on Quantum K-means

6. ARI and NMI Analysis: While quantum K-means achieved respectable ARI and NMI scores, the higher scores obtained by classical K-means indicate a stronger agreement between the clustering results and ground truth labels for our datasets.

7. Impact of Quantum K-means: In our experiment, classical K-means demonstrated its effectiveness in achieving higher accuracy and better alignment with ground truth labels compared to quantum K-means. These results highlight the limitations of current quantum computing capabilities or the need for further optimization of quantum K-means algorithms.

8. Discussion of Findings: The findings emphasize the importance of considering the specific characteristics of the datasets and the limitations of quantum computing when applying quantum K-means clustering. Further research is required to explore potential modifications to the quantum algorithm or investigate hybrid approaches to leverage the strengths of classical and quantum computing.

9. Conclusion: In conclusion, our study revealed that classical K-means outperformed quantum K-means in terms of accuracy and agreement with ground truth labels for the selected datasets. The results shed light on the challenges and limitations associated with quantum K-means clustering, highlighting the

need for continued research to refine and optimize quantum algorithms for clustering tasks.

6 Conclusion and Future Work

6.1 Conclusion

In this work, our main focus was on unsupervised methods for pattern clustering tasks. We began by introducing the classical K-means algorithm, which is a widely used clustering technique. However, as we delved deeper into exploring innovative approaches, we expanded our research to include quantum K-means.

Quantum K-means is a variant of the traditional K-means algorithm that leverages the principles and tools of quantum computing. By harnessing the power of quantum mechanics, this approach aims to overcome the limitations of classical computing in solving complex clustering problems. Quantum K-means takes advantage of quantum superposition and entanglement to process data in parallel and potentially discover hidden patterns and structures more efficiently.

In our study, we not only presented the concept of quantum K-means but also developed and implemented our own version of the algorithm. We strived to demonstrate how quantum computing techniques could enhance the performance and scalability of the K-means algorithm in large datasets.

The motivation behind exploring quantum K-means was to exploit the unique capabilities of quantum computers, such as their potential for exponential speedup compared to classical computers. By leveraging these advantages, we aimed to achieve significant improvements in the efficiency and effectiveness of clustering tasks.

By introducing quantum computing techniques to the field of clustering, we aimed to contribute to the ongoing advancements in unsupervised learning and pattern recognition. The successful outcomes of our proposed quantum K-means approach encourage further exploration and utilization of quantum algorithms in solving real-world clustering problems.

6.2 Future Work

Future work in quantum K-means clustering can focus on the following areas://

1. Hybrid Quantum-Classical Approaches: Investigate hybrid methods that combine classical and quantum computing to enhance performance and reliability.

2. Real-World Applications: Apply quantum K-means to various domains and evaluate its performance on real-world datasets.

3. Quantum K-means Optimization: Investigate optimization techniques specifically tailored for quantum K-means clustering, considering factors such as convergence speed and global optima identification.

4. Quantum K-means on Noisy Data: Explore the robustness of quantum K-means algorithms to noisy or incomplete data and develop techniques to handle noise in quantum clustering.

References

[1] Wu, Zhihao, Tingting Song, and Yanbing Zhang. "Quantum k-means algorithm based on Manhattan distance." Quantum Information Processing 21, no. 1 (2022): 19.

[2] Kavitha, S. S., and Narasimha Kaulgud. "Quantum K-means clustering method for detecting heart disease using quantum circuit approach." Soft Computing (2022): 1-14.

[3] Poggiali, Alessandro, Alessandro Berti, Anna Bernasconi, Gianna M. Del Corso, and R. Giudotti. "Clustering Classical Data with Quantum k-Means." In Proceedings of the 23rd Italian Conference on Theoretical Computer Science, Roma, Italy. 2022.

[4] Li, Yong-Mei, Hai-Ling Liu, Shi-Jie Pan, Su-Juan Qin, Fei Gao, Dong-Xu Sun, and Qiao-Yan Wen. "Quantum k-medoids algorithm using parallel amplitude estimation." Physical Review A 107, no. 2 (2023): 022421.

[5] Li, Qingyu, Yuhan Huang, Shan Jin, Xiaokai Hou, and Xiaoting Wang. "Quantum spectral clustering algorithm for unsupervised learning." Science China Information Sciences 65, no. 10 (2022): 200504.