FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Thi Bach Duong Bui

IMPLEMENTING QUANTUM K-MEANS CLUSTERING ALGORITHM

Bachelor's Thesis
Degree Programme in Computer Science and Engineering
May 2024

Bui T. (2024) Implementing Quantum K-Means Clustering Algorithm. University of Oulu, Degree Programme in Computer Science and Engineering, 31 p.

ABSTRACT

In this thesis, an advanced quantum k-means clustering algorithm is introduced, aiming to enhance both efficiency and accuracy compared to previous studies. Experiments are conducted by executing the algorithm and the corresponding quantum circuit on real quantum systems, noisy simulators, and ideal simulators. The results reveal that the quantum implementation achieves high accuracy across all quantum platforms. Furthermore, the quantum k-means clustering algorithm demonstrates competitive performance compared to its classical counterparts, with only a slight reduction in accuracy.

Keywords: k-means, quantum, algorithm, quantum machine learning

TABLE OF CONTENTS

FOREWORD

First of all, I would like to thank Business Finland for their generous funding support under grant 8436/31/2022 for the Towards Reliable Quantum Software Development: Approaches and Use Cases (TORQS) project. Secondly, I would like to express my gratitude to my supervisor, Professor Kimmo Halunen, who provided support for my thesis work and suggested the research direction of qubit-utilization quantum circuits. Thirdly, I would like to thank IBM for granting me access to the IBM quantum systems and cloud simulators.

Oulu, May 2nd, 2024

Thi Bach Duong Bui

LIST OF ABBREVIATIONS AND SYMBOLS

AI	Artificial intelligence
NISQ	Noisy Intermediate Scale Quantum
H	Hadamard gate
CX	Controlled NOT gate
RY	Rotation Y gate
θ	Angle
$\otimes$	Tensor product
arctan	Inverse tangent function

1. INTRODUCTION

Nowadays, machine learning algorithms have achieved remarkable successes with the explosion of AI models, including ChatGPT [1], which answers questions in human-like text, and Sora, which creates realistic and imaginative scenes from text instructions [2]. With these achievements, AI models are generally believed to help accomplish many tasks more efficiently. However, this revolution is beginning to face increasing challenges [3]. As datasets constantly grow larger and Moore's Law comes to an end [4], conventional computational tools will not be sufficient to solve such large problems [3]. Therefore, in order to further advance machine learning development, it is essential to seek new computational tools that allow solving larger and more complex machine learning problems.

One of the highly potential candidates is quantum computing, which is believed to be able to efficiently solve complex large-scale problems across various fields. Unlike conventional computing, which operates based on classical logic, quantum computing harnesses the principles of quantum physics [5]. A quantum bit (qubit) can represent multiple states by combining 0 and 1, whereas a classical bit can only be in the state 0 or 1. This means that with the same number of bits, quantum computers can handle exponentially more states than classical computers. Therefore, they can outperform classical computing when solving some complex and large-scale problems.

Inspired by classical machine learning algorithms, many quantum machine learning algorithms have been proposed, such as the quantum k-means clustering algorithm [6], quantum Boltzmann machine [7], quantum support vector machine [8], quantum convolutional neural network [9], quantum reinforcement learning [10], and quantum natural language processing [11]. This thesis work will delve into quantum approaches for the k-means clustering algorithm.

The k-means clustering algorithm is a popular unsupervised machine learning algorithm used to assign points in a dataset into k clusters. It aims at gaining meaningful insights from the data [12]. This algorithm is applied in various fields, including image segmentation [13], data processing [14], risk evaluation [15], and medical diagnosis [16] [17].

The k-means clustering algorithm is suitable to be conducted with the current stage of quantum computing for several reasons. Firstly, it is an unsupervised machine learning algorithm, which means it is not as computationally complex as supervised machine learning algorithms, making it easier to implement quantumly. Secondly, the k-means clustering algorithm can correct errors itself by recomputing the centroids after cluster assignment and redoing the assignments if the centroids change. This makes it suitable for current-stage quantum computers, often referred to as Noisy Intermediate Scale Quantum (NISQ) quantum computers. NISQ computers are the current and near-term quantum computers, which do not possess sufficient fault tolerance [18].

In this thesis, a quantum algorithm capable of calculating multiple distances simultaneously using quantum computers will be introduced. The work builds upon the study by Khan, Awan, and Vall-Llosera in 2019 [6], where they proposed an optimized quantum circuit for calculating the Euclidean distance between two points. From their study, this thesis will introduce an alternative formula for calculating the

Euclidean distance more accurately and an algorithm for using the quantum circuit more efficiently.

The structure of the thesis is as follows. Chapter 2 introduces all concepts essential for understanding the thesis. Chapter 3 thoroughly describes the new method for k-means clustering. Chapter 4 describes the results of the experiments of running the new quantum k-means clustering algorithm and quantum circuit for multiple distance calculations on different quantum platforms. Chapter 5 discusses the practicality of the quantum k-means clustering algorithm, including its applications and potential advantages. Finally, Chapter 6 summarizes what has been done in the thesis and the conclusions drawn based on that work.

2. RELATED WORK

This chapter introduces all concepts essential for understanding the thesis. It provides techniques and equations for calculating the quantum state after applying the quantum gates. Additionally, it introduces mathematical concepts that can simplify the quantum state. Specifically, it reviews the classical k-means clustering algorithm and previous studies on the quantum k-means clustering algorithm, providing the motivation for the work presented in this thesis.

2.1. Background

This section covers all the fundamental concepts needed to understand the operations of quantum computing, including mathematical and quantum concepts. These concepts are essential in quantum computing to comprehend the results of applying quantum gates efficiently and straightforwardly.

2.1.1. Mathematical Concept

1. Identity matrix [19]:
$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2. Matrix multiplication [19]:
$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \cdot \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} = \begin{bmatrix} a_1b_1 + a_2b_3 & a_1b_2 + a_2b_4 \\ a_3b_1 + a_4b_3 & a_3b_2 + a_4b_4 \end{bmatrix}$$

3. Tensor product [20]:
$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \otimes \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} = \begin{bmatrix} a_1 \cdot \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} & a_2 \cdot \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} \\ a_3 \cdot \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} & a_4 \cdot \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} \end{bmatrix}$$
$$= \begin{bmatrix} a_1b_1 & a_1b_2 & a_2b_1 & a_2b_2 \\ a_1b_3 & a_1b_4 & a_2b_3 & a_2b_4 \\ a_3b_1 & a_3b_2 & a_4b_1 & a_4b_2 \\ a_3b_3 & a_3b_4 & a_4b_3 & a_4b_4 \end{bmatrix}$$

4. Cartesian coordinate to polar coordinate and vice versa [21]

   In a plane, the positions of points can be represented with Cartesian coordinates or polar coordinates. Cartesian coordinates of a point are in the form (x, y), where x and y are the values where the line drawn through that point intersects the horizontal and vertical axes perpendicularly, respectively [21]. Polar coordinates describe the location of points in a plane in the form (r, θ), where r is the distance from the origin to the point and θ is the angle that the line drawn from the origin to the point makes with the horizontal axis[21]. The

formulas for converting Cartesian coordinates to polar coordinates and vice versa are presented below.

Given a Cartesian coordinate of a point (x, y), its Polar coordinate can be calculated as follows:

$$R = \sqrt{x^2 + y^2}, \theta = \arctan \frac{y}{x}$$

Given a polar coordinate of a point (R, α), its Cartesian coordinate can be calculated as follows:

$$x = R\cos(\alpha), y = R\sin(\alpha) \tag{1}$$

5. Trigonometric formula [21]:

   To illustrate the evolution of quantum states after applying quantum gates, this thesis employs matrix multiplication. Trigonometric formulas are utilized to simplify the resulting quantum states of the matrix multiplication. The following trigonometric formulas are utilized in this thesis:

$$\cos(-\theta) = \cos\theta \tag{2}$$

$$\sin(-\theta) = -\sin\theta \tag{3}$$

$$(\sin\theta)^2 + (\cos\theta)^2 = 1 \tag{4}$$

$$(\cos\theta)^2 - (\sin\theta)^2 = \cos 2\theta \tag{5}$$

$$2\sin\theta\cos\theta = \sin 2\theta \tag{6}$$

$$\cos(\alpha_1 - \alpha_2) = (\cos(\alpha_1))(\cos(\alpha_2)) + (\sin(\alpha_1))(\sin(\alpha_2)) \tag{7}$$

6. Binomial square formula [21]:

   Aside from trigonometric formulas, this thesis also utilizes the following binomial square formulas to simplify the quantum states.

$$(a+b)^2 = a^2 + 2ab + b^2 \tag{8}$$

$$(a-b)^2 = a^2 - 2ab + b^2 \tag{9}$$

2.1.2. Quantum Concept

1. Quantum state:

   Quantum state, denoted as $|\psi\rangle$, indicates the probabilities of a particle being measured at locations $\{x_1, x_2, \ldots, x_n\}$ before measurement [22]. In this thesis, the quantum state is represented in the following form:

$$|\psi\rangle = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix},$$

where $a_0, a_1, \ldots, a_n$ are probability amplitudes of being in states $\{|x_1\rangle, |x_2\rangle, \ldots, |x_n\rangle\}$ [22].

2. Quantum logic gates

   Quantum gates are a means of qubit manipulation, which, when applied to a quantum state, will produce another quantum state [22]. Quantum gates, except measurement gates, must be reversible, and the squared norm of the probability amplitudes must sum to one after gate application [22]. With these conditions, the quantum gates can be represented by unitary matrices [22]. The matrix representation of the quantum gates used in this thesis is shown below.

   (a) Hadamard gate:

   The Hadamard gate is a gate that puts a qubit initially in a definite $|0\rangle$ or $|1\rangle$ state into a superposition of $|0\rangle$ and $|1\rangle$ states [23]. This gate is represented by the following symbol:

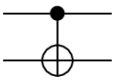
   The operation of the Hadamard gate can be represented by the following unitary matrix [23].

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

   (b) Controlled NOT gate (CNOT gate):

   The CNOT gate is a multi-qubit gate used to entangle two qubits together. After the application of a CNOT gate, the control qubit remains unchanged, while the target qubit follows the following rule: if the control qubit is $|0\rangle$, the target qubit remains unchanged. Conversely, if the control qubit is $|1\rangle$, the target qubit flips to $|1\rangle$ if it is $|0\rangle$ or flips to $|0\rangle$ if it is $|1\rangle$ [23].

   The CNOT gate is represented by the following symbol:

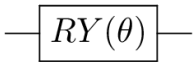


   The operation of the CNOT gate can be represented by the following unitary matrix [23].

$$CX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

   (c) Rotation Y (RY) gate:

   The RY gate is a gate that rotates a qubit by an angle θ around the y-axis [6]. This gate is represented by the following symbol:



The operation of the rotation Y gate can be represented by the following unitary matrix [6].

$$RY = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix}$$

3. Calculating quantum states when applying gate to single qubit

   In a multiple-qubit quantum circuit, when applying a quantum gate to a single qubit, the quantum state of the circuit after applying this gate can be calculated by taking the tensor product of the gate with the identity matrix on the qubits which do not have any gate applied at that stage. For example, in a 2-qubit quantum circuit, when applying a Hadamard gate to only qubit 1, the quantum state after this operation can be calculated as $|\psi_{i+1}\rangle = H \otimes I \cdot |\psi_i\rangle$, where I is the identity matrix and $|\psi_i\rangle$ is the quantum state before this operation [6].

4. Calculating probability of quantum state

   Given a quantum state $|\psi\rangle = [a_0, a_1, \ldots, a_n]^T$, the probability of each state is equal to the absolute value of the amplitude value of that state squared [6]. For example, the probability of state $|00\ldots0\rangle$ is $|a_0|^2$. The probabilities of all states need to satisfy the following condition:

$$|a_0|^2 + |a_1|^2 + \cdots + |a_n|^2 = 1$$

2.2. Related Algorithms

This section introduces the classical k-means clustering algorithm and reviews previous studies on quantum k-means clustering.

2.2.1. Classical K-Means Clustering Algorithm

The k-means clustering algorithm is used to partition a dataset with N objects, each having measurements on P variables, into K clusters $(C_1, C_2, \ldots, C_K)$, where C_k is the set of n_k objects in cluster k, and K is provided [24]. The classical k-means

clustering algorithm is as follows.

Algorithm 1. Classical k-means clustering algorithm [24]

Input : dataset, number of desired clusters k

Output: k clusters

1 Define k seeds as P-dimensional vectors $s_1^{(k)}, \ldots, s_P^{(k)}$, for $1 \leq k \leq K$

2 Calculate the squared Euclidean distances between objects in the dataset and k centroids using the following formula:

$$d^2(i,k) = \sum_{j=1}^{P} (x_{ij} - s_j^{(k)})^2, \tag{10}$$

where $d^2(i,k)$ is the squared Euclidean distance between the ith object and kth seed vector.

3 Allocate the objects to clusters where (10) is minimum

4 After initial object allocation, calculate centroids for each cluster by averaging the values on each variable over the objects within the clusters. The centroid vector for cluster C_k is given by:

$$\overline{x}^{(k)} = (\overline{x}_1^{(k)}, \overline{x}_2^{(k)}, \ldots, \overline{x}_P^{(k)})',$$

where

$$\overline{x}_j^{(k)} = \frac{1}{n_k} \sum_{i \in C_k} x_{ij}$$

Assign objects to the clusters based on the closest distance to the newly calculated centroids.

5 Calculate new centroids after all data points have been assigned to K clusters

6 Repeat steps 4 and 5 until no objects can be moved between clusters.

In the case of a two-dimensional dataset, the Euclidean distance mentioned in the algorithm above can be calculated using the following formula:

$$d(p,c) = \sqrt{(p_x - c_x)^2 + (p_y - c_y)^2}, \tag{11}$$

Here, (p_x, p_y) represents the coordinates of the data point, and (c_x, c_y) represents the coordinates of the centroid. Since this thesis will only consider two-dimensional datasets, formula (11) is utilized.

In step 5, algorithm 1 recomputes the centroids and compares them with the old ones. This helps the k-means clustering algorithm identify errors in the cluster allocations. If an object is inaccurately allocated to a cluster, it will be significantly different from other objects in the clusters. This results in newly calculated centroids that differ from the old centroids, providing the algorithm with another opportunity to assign objects to the correct clusters represented by the new centroids. This characteristic makes the k-means clustering algorithm suitable for current-stage quantum computers, which produce noisy results.

On the other hand, inaccuracies in classifying objects into clusters introduce additional computational effort to the quantum computer, as centroids are recalculated every time there are points with significant differences from other points in the clusters, and the entire object allocation task is conducted again. Therefore, it is important

to take into account the accuracy of the distance calculation to avoid redundant recomputation of centroids.

2.2.2. Quantum K-Means Clustering Algorithms

Quantum k-means clustering on NISQ computers

In 2019, Khan, Awan, and Vall-Llosera proposed three optimized implementations of the quantum k-means algorithm, which included a distance-based classifier circuit, a negative rotation circuit to find the nearest centroid to the data point, and a destructive interference circuit for calculating Euclidean distance quantumly [6]. This thesis utilizes the destructive interference quantum circuit for computing Euclidean distance. Consequently, this section provides a comprehensive description of the circuit.

The circuit designed by Khan, Awan, and Vall-Llosera for calculating Euclidean distance is as follows:

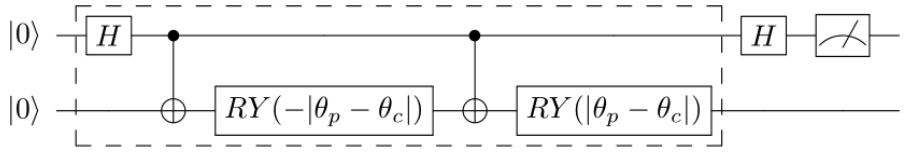


Figure 1. Khan, Awan, and Vall-Llosera proposed a quantum circuit for calculating Euclidean distance [6]. The section enclosed by dashed lines is dedicated to preparing the desired state, while the remaining section is for interference and measurement.

The primary concept behind this circuit is to generate a normalized quantum state

$$|\psi\rangle = \begin{bmatrix} p'_x & p'_y & c'_x & c'_y \end{bmatrix}^T, \tag{12}$$

where

$$Norm = \sqrt{p_x^2 + p_y^2 + c_x^2 + c_y^2} \tag{13}$$

$$p'_x = \frac{p_x}{Norm}, p'_y = \frac{p_y}{Norm}, c'_x = \frac{c_x}{Norm}, c'_y = \frac{c_y}{Norm}, \tag{14}$$

Then, by applying a Hadamard gate to the first qubit, the following state will be obtained:

$$[H \otimes I]\,|\psi\rangle = \frac{1}{\sqrt{2}} \left[\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right] \begin{bmatrix} p'_x \\ p'_y \\ c'_x \\ c'_y \end{bmatrix}$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} p'_x \\ p'_y \\ c'_x \\ c'_y \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} p'_x + c'_x \\ p'_y + c'_y \\ p'_x - c'_x \\ p'_y - c'_y \end{bmatrix} \begin{matrix} |00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle \end{matrix}$$

From this state, probability of measuring $|1\rangle$ in the first qubit can be calculated as:

$$P(|1\rangle) = \left[\frac{1}{\sqrt{2}}(p'_x - c'_x)\right]^2 + \left[\frac{1}{\sqrt{2}}(p'_y - c'_y)\right]^2 = \frac{1}{2}[(p'_x - c'_x)^2 + (p'_y - c'_y)^2]$$
$$= \frac{1}{2}\left[\left(\frac{p_x}{Norm} - \frac{c_x}{Norm}\right)^2 + \left(\frac{p_y}{Norm} - \frac{c_y}{Norm}\right)^2\right]$$
$$= \frac{1}{2}\frac{1}{Norm^2}[(p_x - c_x)^2 + (p_y - c_y)^2]$$

This is explained physically as the amplitude values, where the first qubit is in the state $|0\rangle$, interfere with the amplitude values, where the first qubit is in state $|1\rangle$ [6]. This results in quantum states with constructive interference on state $|0\rangle$ of the first qubit and destructive interference on state $|1\rangle$ [6]. By exploiting this destructive interference on state $|1\rangle$, Khan, Awan, and Vall-Llosera proposed a formula to calculate Euclidean distance (Eq.(11)) from the probability of the first qubit in state $|1\rangle$:

$$d(p, c) = \sqrt{(p_x - c_x)^2 + (p_y - c_y)^2} = Norm \times \sqrt{2}\sqrt{P(|1\rangle)} \tag{15}$$

Therefore, if the quantum state $|\psi\rangle$ (Eq.(12)) can be prepared, by applying a Hadamard gate to the first qubit, the Euclidean distance is obtained. Thus, the main problem that needs to be addressed is how to prepare the quantum state $|\psi\rangle$ (Eq.(12)). According to Khan, Awan, and Vall-Llosera [6], this problem can be solved with the following circuit:

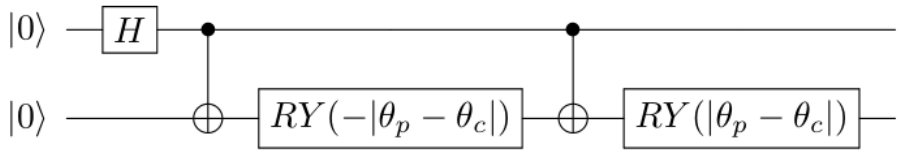


Figure 2. Quantum circuit for state preparation [6]

The idea of the state preparation circuit is to load the polar coordinates of the data point vector θ_p and centroid vector θ_c into the quantum circuit. However, because the focus is solely on the angular difference between p and c, loading the polar coordinates of p and c separately is redundant. Instead, the circuit depicted in Figure 2 keeps the data point vector p lying on the x-axis while rotating the centroid vector c by an angle $|\theta_p - \theta_c|$, where θ_p is the polar coordinate of p and θ_c is the polar coordinate of c [6]. This approach yields quantum states where the data point vector p and centroid vector c are located at the same angular difference as when the vectors are loaded separately, but with a significantly shorter circuit depth. This not only facilitates faster execution but also reduces errors [6].

Khan, Awan, and Vall-Llosera's method can only be applied in cases where the polar radius of the points has no difference, as it assumes that the polar radius of all points is equal to 1. However, in cases where the polar radii of data points differ significantly, this method might produce inaccurate results when comparing the distance between the data point and centroids.

For instance, consider a scenario where the Cartesian coordinate of the data point is $(\frac{1}{2}, \frac{\sqrt{3}}{2})$, centroid 1 is $(2, 2)$, and centroid 2 is $(\frac{\sqrt{3}}{2}, \frac{1}{2})$. The polar angle of the data point

is $60°$, centroid 1 is $45°$, and centroid 2 is $30°$. If only angular difference is considered, then centroid 1 appears nearer to the data point than centroid 2. This is because the angular difference between the data point and centroid 1 is $60° - 45° = 15°$, which is smaller than the angular difference between the data point and centroid 2, which is $60° - 30° = 30°$. However, upon comparing their Euclidean distances, it becomes evident that centroid 2 is closer to the data point than centroid 1.

$$d(p, c_1) = \sqrt{\left(\frac{1}{2} - 2\right)^2 + \left(\frac{\sqrt{3}}{2} - 2\right)^2} \approx 1{,}88$$

$$d(p, c_2) = \sqrt{\left(\frac{1}{2} - \frac{\sqrt{3}}{2}\right)^2 + \left(\frac{\sqrt{3}}{2} - \frac{1}{2}\right)^2} \approx 0{,}52$$

Therefore, in cases where the polar radii of points are significantly different, another method is needed to compute Euclidean distance. This thesis will propose a method to address such cases.

3. IMPLEMENTATION

3.1. 2-Qubit Quantum Circuit for Calculating Distance

This thesis employs the quantum circuit proposed by Khan, Awan, and Vall-Llosera [6], as depicted in Figure 1. However, instead of using the formula (Eq. (15)) to calculate Euclidean distance from the probability of the first qubit in state $|1\rangle$, another formula will be utilized. Before introducing this alternative formula, let's first examine in detail how Khan, Awan, and Vall-Llosera's circuit (Figure 1) operates. Initially, the circuit is in the state $|00\rangle$, which is represented as:

$$|\psi_0\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}^T$$

1. Apply the H gate
   The first step in the circuit is applying the Hadamard gate to the first qubit. This step sets the data point vector p and the centroid vector c lying on the x-axis. This operation is indicated as follows:

$$|\psi_1\rangle = [H \otimes I]\,|\psi_0\rangle = \left[\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right] \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

2. Apply the CX gate

   After applying the Hadamard gate, a controlled-NOT gate is applied in the circuit, where the control qubit is the first qubit and the target qubit is the second qubit. This step entangles the data point vector p and the centroid vector c. As a result, vector c becomes oriented vertically at a 90° angle relative to vector p. This operation is indicated as follows:

$$|\psi_2\rangle = CX \cdot |\psi_1\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

3. Apply the $RY(-|\theta_p - \theta_c|)$ gate

Next, the circuit rotates the second qubit by $-|\theta_p - \theta_c|$ around the y-axis. Let $\theta = |\theta_p - \theta_c|$. This step rotates both vectors p and c by $(-\frac{\theta}{2})$ degrees. The operation is represented as follows:

$$|\psi_3\rangle = [I \otimes RY(-\theta)]\,|\psi_2\rangle = \left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \otimes \begin{bmatrix}\cos(-\frac{\theta}{2}) & -\sin(-\frac{\theta}{2})\\ \sin(-\frac{\theta}{2}) & \cos(-\frac{\theta}{2})\end{bmatrix}\right] \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix}1\\0\\0\\1\end{bmatrix}$$

$$= \begin{bmatrix}\cos(-\frac{\theta}{2}) & -\sin(-\frac{\theta}{2}) & 0 & 0\\ \sin(-\frac{\theta}{2}) & \cos(-\frac{\theta}{2}) & 0 & 0\\ 0 & 0 & \cos(-\frac{\theta}{2}) & -\sin(-\frac{\theta}{2})\\ 0 & 0 & \sin(-\frac{\theta}{2}) & \cos(-\frac{\theta}{2})\end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix}1\\0\\0\\1\end{bmatrix}$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix}\cos(-\frac{\theta}{2})\\ \sin(-\frac{\theta}{2})\\ -\sin(-\frac{\theta}{2})\\ \cos(-\frac{\theta}{2})\end{bmatrix}$$

By applying equations (2) and (3), state $|\psi_3\rangle$ can be written in a non-negative angle form:

$$|\psi_3\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix}\cos\frac{\theta}{2}\\ -\sin\frac{\theta}{2}\\ \sin\frac{\theta}{2}\\ \cos\frac{\theta}{2}\end{bmatrix}$$

4. Apply the CX gate

   At this step, the circuit applies a controlled-NOT gate, where the control qubit is the first qubit and the target qubit is the second qubit. This step entangles the data point vector p and the centroid vector c. As a result, vector c becomes oriented at an angle θ relative to vector p. This operation is indicated as follows:

$$|\psi_4\rangle = CX \cdot |\psi_3\rangle = \begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix}\cos\frac{\theta}{2}\\ -\sin\frac{\theta}{2}\\ \sin\frac{\theta}{2}\\ \cos\frac{\theta}{2}\end{bmatrix} = \frac{1}{\sqrt{2}} \cdot \begin{bmatrix}\cos\frac{\theta}{2}\\ -\sin\frac{\theta}{2}\\ \cos\frac{\theta}{2}\\ \sin\frac{\theta}{2}\end{bmatrix}$$

5. Apply the $RY(|\theta_p - \theta_c|)$ gate

   Next, the circuit rotates the second qubit by $|\theta_p - \theta_c|$ around the y-axis. Let $\theta = |\theta_p - \theta_c|$. This step rotates both data point vector p and centroid vector c by $\frac{\theta}{2}$ degrees. As a result, the data point vector p lies on the x-axis, while the

centroid vector c is at an angle θ relative to vector p. The operation is represented as follows:

$$|\psi_5\rangle = [I \otimes RY(\theta)]\,|\psi_4\rangle = \left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \otimes \begin{bmatrix}\cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right)\\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right)\end{bmatrix}\right] \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix}\cos\frac{\theta}{2}\\ -\sin\frac{\theta}{2}\\ \cos\frac{\theta}{2}\\ \sin\frac{\theta}{2}\end{bmatrix}$$

$$= \begin{bmatrix}\cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) & 0 & 0\\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) & 0 & 0\\ 0 & 0 & \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right)\\ 0 & 0 & \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right)\end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix}\cos\frac{\theta}{2}\\ -\sin\frac{\theta}{2}\\ \cos\frac{\theta}{2}\\ \sin\frac{\theta}{2}\end{bmatrix}$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix}(\cos\left(\frac{\theta}{2}\right))^2 + (\sin\left(\frac{\theta}{2}\right))^2\\ \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right)\\ (\cos\left(\frac{\theta}{2}\right))^2 - (\sin\left(\frac{\theta}{2}\right))^2\\ \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)\cos\left(\frac{\theta}{2}\right)\end{bmatrix}$$

By applying equations (4),(5), and (6), state $|\psi_5\rangle$ can be written in a non-negative angle form:

$$|\psi_5\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix}1\\ 0\\ \cos\theta\\ \sin\theta\end{bmatrix} \tag{16}$$

This is the final stage of the state preparation task in the circuit Figure 1. According to Khan, Awan, and Vall-Llosera, at this stage, the quantum state is as the equation (12), which can be represented with polar coordinates as follows.

$$|\psi\rangle = \begin{bmatrix}p'_x\\ p'_y\\ c'_x\\ c'_y\end{bmatrix} = \frac{1}{\sqrt{R_p^2 + R_c^2}} \cdot \begin{bmatrix}R_p\cos\theta_p\\ R_p\sin\theta_p\\ R_c\cos\theta_c\\ R_c\sin\theta_c\end{bmatrix} = \frac{1}{\sqrt{R_p^2 + R_c^2}} \cdot \begin{bmatrix}R_p\\ 0\\ R_c\cos\theta\\ R_c\sin\theta\end{bmatrix} \tag{17}$$

From comparing Equations (16), (17), it is clear that the polar radii R_p and R_c are ignored in Khan, Awan, and Vall-Llosera approach, where they assumed that $R_p = R_c = 1$. This condition is rarely met in reality. As mentioned in section 2.2.2, if the radii are also different, merely comparing the angular difference will yield inaccurate results. In such cases, the alternative formula proposed in this thesis will help calculate Euclidean distance accurately. Before introducing that formula, let's calculate the quantum state after performing interference tasks in the circuit.

6. Apply the H gate

   A Hadamard gate is applied to the first qubit to induce interference transformation, making the amplitude values where the first qubit is in state $|0\rangle$

interfere with the amplitude values where the first qubit is in state $|1\rangle$. This operation is indicated as follows:

$$|\psi_6\rangle = [H \otimes I]\,|\psi_5\rangle = \left[\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix} \otimes \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\right] \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix}1\\ 0\\ \cos\theta\\ \sin\theta\end{bmatrix}$$

$$= \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 1 & 0 & 1\\ 1 & 0 & -1 & 0\\ 0 & 1 & 0 & -1\end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \begin{bmatrix}1\\ 0\\ \cos\theta\\ \sin\theta\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1+\cos\theta\\ \sin\theta\\ 1-\cos\theta\\ -\sin\theta\end{bmatrix}$$

If measuring this circuit, the probability of the first qubit in state $|1\rangle$ is:

$$P(|1\rangle) = \frac{1}{2\cdot 2}[(1-\cos\theta)^2 + (-\sin\theta)^2] = \frac{1}{2\cdot 2}[1 - 2\cos\theta + (\cos\theta)^2 + (\sin\theta)^2]$$

Using the equation (4), $P(|1\rangle)$ can be written as:

$$P(|1\rangle) = \frac{1}{2\cdot 2}[1 - 2\cos\theta + 1] = \frac{1}{2\cdot 2}[2 - 2\cos\theta] = \frac{1}{2} - \frac{1}{2}\cos\theta \tag{18}$$

At this stage, instead of using Khan, Awan, and Vall-Llosera's equation (Equation (15)), the following equation should be used to calculate the Euclidean distance more accurately from the probability of the first qubit in state $|1\rangle$.

$$d(t,c) = \sqrt{R_p^2 + R_c^2 - 2R_pR_c(1 - 2P(|1\rangle))} \tag{19}$$

The correctness of this equation is proven using equations (4), (7), (9), (1) as follows:

$$R_p^2 + R_c^2 - 2R_pR_c(1 - 2P(|1\rangle))$$

$$= R_p^2[(\sin\theta_p)^2 + (\cos\theta_p)^2] + R_c^2[(\sin\theta_c)^2 + (\cos\theta_c)^2] - 2R_pR_c(1 - 2(\frac{1}{2} - \frac{1}{2}\cos\theta))$$

$$= R_p^2(\cos\theta_p)^2 + R_c^2(\cos\theta_c)^2 + R_p^2(\sin\theta_p)^2 + R_c^2(\sin\theta_c)^2$$
$$-2R_pR_c(1 - 1 + \cos(\theta_p - \theta_c))$$

$$= R_p^2(\cos\theta_p)^2 + R_c^2(\cos\theta_c)^2 + R_p^2(\sin\theta_p)^2 + R_c^2(\sin\theta_c)^2$$
$$-2R_pR_c(\cos(\theta_p)\cos(\theta_c) + \sin(\theta_p)\sin(\theta_c))$$

$$= [R_p^2(\cos\theta_p)^2 - 2R_p\cos(\theta_p)R_c\cos(\theta_c) + R_c^2(\cos\theta_c)^2]$$
$$+[R_p^2(\sin\theta_p)^2 - 2R_p\sin(\theta_p)R_c\sin(\theta_c) + R_c^2(\sin\theta_c)^2]$$

$$= (R_p\cos(\theta_p) - R_c\cos(\theta_c))^2 + (R_p\sin(\theta_p) - R_c\sin(\theta_c))^2$$

$$= (p_x - c_x)^2 + (p_y - c_y)^2$$

3.2. Qubit-Utilization Quantum Circuit

To calculate the distance between two points, two qubits are required. In the scenario where a quantum circuit with more than 2 qubits is available, it becomes feasible to calculate multiple distances simultaneously using the quantum circuit depicted below:

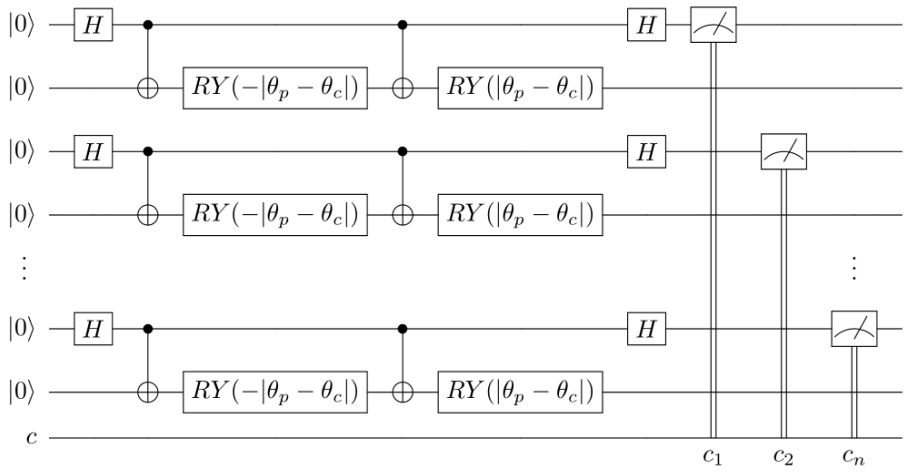


Figure 3. Quantum circuit for measuring multiple distances simultaneously.

The concept behind this circuit involves computing distances for various pairs of points within each group of 2 qubits and storing the measurement results of each pair in distinct classical bits. Subsequently, the probability corresponding to each classical bit being 1 is utilized to calculate the distance between each pair of points.

In the subsequent section, an algorithm for executing this quantum circuit will be introduced.

3.3. Qubit-Utilization Quantum K-Means Clustering Algorithm

Before describing the qubit-utilization quantum k-means clustering algorithm, let's go through two quantum subroutines of the algorithm. One subroutine constructs the quantum circuit, while the other subroutine calculates the distances between data points and centroids and chooses the nearest centroids for each data point.

Firstly, let's go through the subroutine for constructing the quantum circuit. This subroutine is used to cooperate with the "select_nearest_centroids" subroutine in the task of iteratively putting quantum gates to each group of 2 qubits. This is done by multiplying 2 with the number of iterations that have been done. For example, when there is no task in the quantum circuit, which means that the number of iterations is 0, then the qubits $2 \cdot 0 = 0$ and $2 \cdot 0 + 1 = 1$ are used to perform one iteration task. Now, the number of iterations is 1, and the qubits $2 \cdot 1 = 2$ and $2 \cdot 1 + 1 = 3$ are used to perform another iteration task. Repeating this, it is possible to automatically create a multiple-tasks

quantum circuit without the need for specifically assigning the qubits for each task.

Algorithm 2. Construct Quantum Circuit

Input : θ_p, θ_c, i, quantum_circuit
Output: quantum_circuit (after adding operations for calculating Euclidean distance)

$\theta = |\theta_p - \theta_c|$
Apply H gate to qubit $2i$
Apply CX gate, where qubit $2i$ is control qubit, and qubit $2i+1$ is target qubit
Apply $RY(-\theta)$ to qubit $2i+1$
Apply CX gate, where qubit $2i$ is control qubit, and qubit $2i+1$ is target qubit
Apply $RY(\theta)$ to qubit $2i+1$
Apply H gate to qubit $2i$
Measure qubit $2i$ and store the result in classical bit i

Now, let's move on to the next subroutine in the quantum k-means clustering algorithm: the "select_nearest_centroids" subroutine. This subroutine takes the role of calculating the distances between data points and centroids and choosing the nearest centroids for each data point. The subroutine uses a dictionary to store the information on the nearest centroid's coordinate and the smallest distance for each data point. With this information, there is no need to search for the minimum value in a list of distances from the data point to each centroid. Apart from this information, the dictionary also stores information about the centroids whose distance to the data point has not been calculated. This information is used for the task of calculating multiple distances simultaneously in the quantum circuit.

The idea of this subroutine is first to find, as much as the circuit can calculate at the same time, the pairs of data points and centroids whose distances have not been calculated, and calculate those distances using the quantum circuit constructed by the "construct_quantum_circuit" subroutine. Then, the subroutine sums up the probability of state $|1\rangle$ in the bit corresponds to the measurement results of that distance calculating task. Note that different quantum platforms might show the result in normal order or reverse order. This should be taken into account to extract the result from the correct bit.

After obtaining the probability of state $|1\rangle$, the distance is calculated using Equation (19). This distance is then compared with the smallest distance which is already stored in the dictionary. If the distance is smaller, then this centroid is nearer than the previously visited centroids. Therefore, the dictionary updates the centroid as the nearest centroid and the recently calculated distance as the smallest distance. This process is repeated until all the distances from each data point to each centroid are calculated.

Algorithm 3. Select nearest centroids

Input : A dataset, a list of centroids (centroids), the number of available qubits (num_qubits)

Output: A dictionary contains the nearest centroid's coordinate for each data point (nearest_centroids_dict)

num_distances ← quotient of dividing num_qubits by 2
nearest_centroids_dict ← {{data_point_1, nearest_centroid_1, smallest_distance_1, unvisited_centroids_1}, . . . {data_point_n, nearest_centroid_n, smallest_distance_n, unvisited_centroids_n}}
while *distances from each data point to all centroids are not calculated* **do**
    calculated_distances ← Empty list
    $i \leftarrow 0$
    for *item in nearest_centroids_dict* **do**
        while *item's unvisited_centroids not empty and number of calculated_distances is less than num_distances* **do**
            data_point ← item's data_point
            $\theta_p \leftarrow$ cartesian_to_polar_angle(data_point)
            centroid ← item's unvisited centroids' centroid
            $\theta_c \leftarrow$ cartesian_to_polar_angle(centroid)
            quantum_circuit ← construct_quantum_circuit(θ_p, θ_c, i, quantum_circuit) (Algorithm 2)
            Append the pair (data_point, centroid) to calculated_distances
            Increase i by 1
        end
    end
    Run the prepared quantum circuit
    for *each pair (p, c) in calculated_distances* **do**
        $P(|1\rangle) \leftarrow$ probability of all states where the bit in the same index as pair (p, c) is 1
        $R_p \leftarrow$ cartesian_to_polar_radius(p)
        $R_c \leftarrow$ cartesian_to_polar_radius(c)
        distance $\leftarrow \sqrt{R_p^2 + R_c^2 - 2R_pR_c(1 - 2P(|1\rangle))}$
        for *item in nearest_centroids_dict* **do**
            if *item's data_point is p* **then**
                if *distance is smaller than item's smallest_distance* **then**
                    item's smallest_distance ← distance
                    item's nearest_centroid ← c
                end
                Remove c from item's unvisited_centroids
            end
        end
    end
end

The main algorithm, quantum k-means clustering, takes the role of assigning points into clusters based on the dictionary created by the "select_nearest_centroids"

algorithm. Afterwards, the quantum k-means clustering algorithm computes the mean of the points in the clusters and compares those means with the current centroids. If they are similar, which means the clusters reach convergence, then the algorithm finishes. If not, the algorithm redoes the aforementioned tasks with the means of the points in the clusters as the new centroids. The algorithm is as follows.

Algorithm 4. Quantum k-means clustering

Input : A dataset, the number of clusters (k), the number of available qubits (num_qubits)
Output: A dictionary contains k clusters ($clusters\_dict$)

```
Choose k centroids randomly from the dataset
clusters_dictionary ← {{centroid_1, points_1}, ..., {centroid_n, points_n}}
while not convergence do
    centroids ← clusters_dictionary's centroids
    nearest_centroids_dictionary ← select_nearest_centroids(dataset,
     centroids, number_qubits) (Algorithm 3)
    for item in nearest_centroids_dictionary do
        data_point ← item's data_point
        nearest_centroid ← item's nearest_centroid
        for cluster in clusters_dictionary do
            if cluster's centroid is nearest_centroid then
                Append data_point into cluster's points
            end
        end
    end
    if not enough k clusters then
        Choose another k random centroids
        Do the upper tasks again with the clusters_dictionary with newly
         chosen centroids.
    else
        Recompute centroids as means of the points in the clusters
        Check if convergence is reached or not
    end
end
```

4. EXPERIMENTS

In this thesis work, three experiments were conducted with different quantum platforms, including real quantum systems, noisy simulators, and ideal simulators. All the experiments were tested with the Iris Dataset [25], which contains 150 samples from 3 different species of Iris flower. Each sample has 4 dimensions, indicating 4 different features with real and positive values. However, these experiments only clustered based on 2 dimensions.

4.1. Real Quantum System: Qubit-Utilization Quantum Circuit

Initially, the qubit-utilization quantum circuit (see Figure 3) was tested using the IBM quantum system, "ibm_kyoto", which had 127 qubits, and an error per layered gate for a 100-qubit chain of $3{,}6\%$ [26]. The experiment was conducted with quantum circuits of different numbers of qubits, including a 10-qubit quantum circuit, a 20-qubit quantum circuit, and a 100-qubit quantum circuit. The quantum results of Euclidean distances calculation were close to the Euclidean distances calculated classically, with an average difference of approximately $0{,}065$. This result is as expected, given that the quantum circuit in Figure 3 has a constant and low circuit depth. Additionally, the real quantum system showed no significant difference in execution time when running different-sized quantum circuits. The time it took to run the 100-qubit quantum circuit was the same as the time it took to run the 10-qubit quantum circuit. In contrast, the quantum simulator took considerably longer to run larger-sized quantum circuits.

4.2. Noisy Quantum Simulator: Quantum K-Means Clustering

Apart from running on a real quantum system, an experiment was conducted to run the quantum k-means clustering (Algorithm 4) on a small-scale quantum simulator, which mimicked the behavior of a real quantum system and produced results with a noise factor. The noisy quantum simulator used in this experiment was the AerSimulator [27], which allowed simulation with noise similar to a specified real quantum system. In this experiment, the "ibm_kyoto" system was chosen as the model for this noisy simulator. The task tested in this experiment was classifying 30 randomly chosen 2-dimensional Iris data into 2 clusters using a 10-qubit quantum circuit. The quantum result was compared with the classical k-means clustering result and the provided correct labels of the Iris dataset.

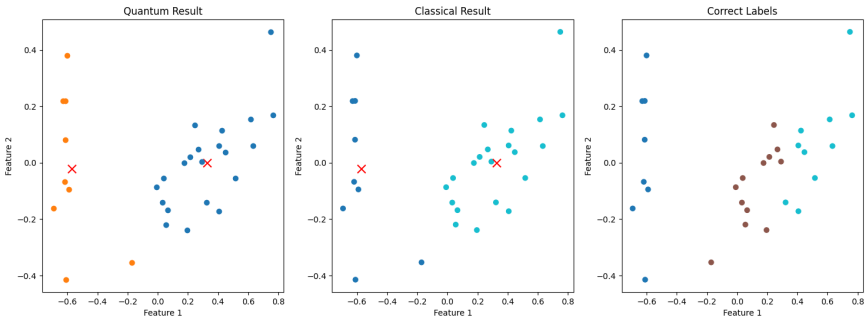


Figure 4. Experiment result of running Algorithm 4 in noisy simulator 1

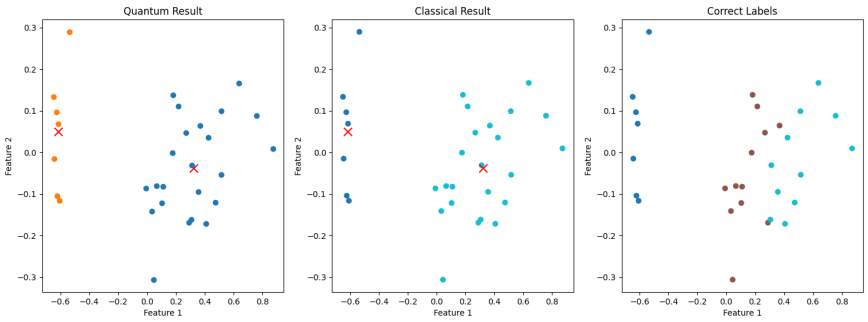


Figure 5. Experiment result of running Algorithm 4 in noisy simulator 2

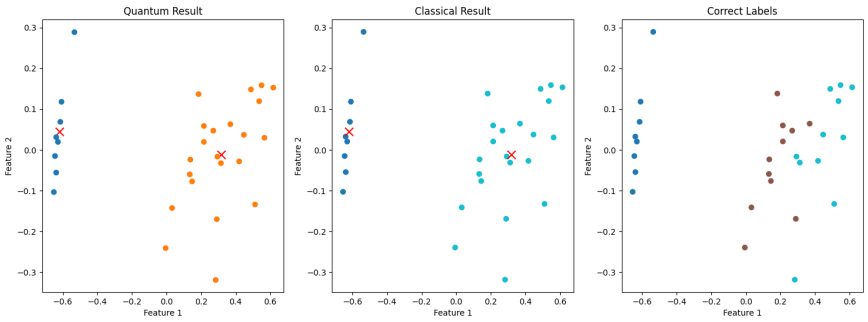


Figure 6. Experiment result of running Algorithm 4 in noisy simulator 3

The k-means clustering algorithm's task was to classify the dataset into 2 clusters, unlike the correct label, which classified the dataset into 3 clusters. Therefore, the experiment result was considered correct if all objects in the 3rd cluster could belong to either the 1st or 2nd cluster. For example, in experiments Figure 5 and Figure 6, the quantum and classical k-means clustering algorithms produced exact results, while in experiment Figure 4, the quantum and classical k-means clustering algorithms classified one object into the wrong cluster. However, this result did not indicate the inaccuracy of the quantum computer in calculating Euclidean distance, since the

wrongly allocated object was nearer to the wrong cluster's centroid than the correct cluster's centroid. Therefore, even when run on the noisy model, the quantum k-means clustering algorithm (Algorithm 4) still performed well with a $3{,}33\%$ error rate, and the quantum circuit in Figure 3 produced exact results.

4.3. Ideal Quantum Simulator: Quantum K-Means Clustering

In this experiment, an IBM cloud-based ideal simulator, "ibmq_qasm_simulator" [28], was used to perform large-scale experiments on running quantum k-means clustering (Algorithm 4) on the whole Iris dataset. Ideal simulators mean that they do not take into account the noise factor, which is unignorable in real quantum systems. Firstly, an experiment was conducted to run Algorithm 4 to cluster 150 data into 3 clusters using a 30-qubit quantum circuit. The result is shown in the figure below.

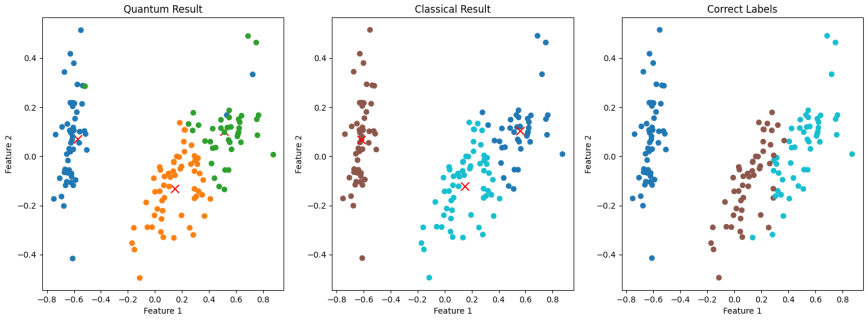


Figure 7. Experiment result of running Algorithm 4 on ideal simulator 1

The result showed that the quantum algorithm (Algorithm 4) produced an almost similar result to its classical counterpart, with a $1{,}33\%$ difference. However, both quantum and classical k-means clustering algorithms failed to correctly classify the objects in the two right-hand side clusters, which were located closely to each other.

Apart from this experiment, experiments were conducted to run Algorithm 4 to classify 100 data into 2 clusters using a 30-qubit quantum circuit. The result is shown in the figures below.

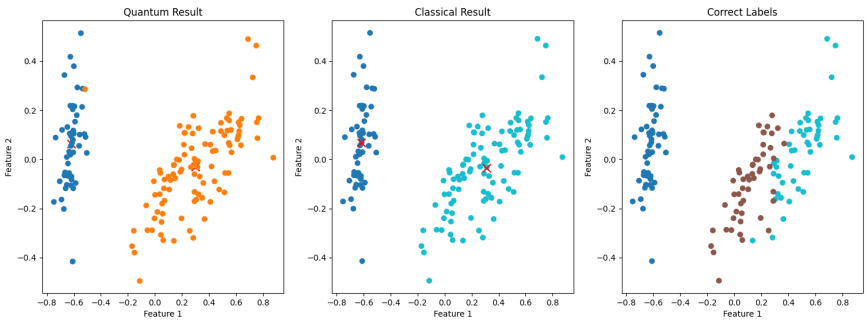


Figure 8. Experiment result of running Algorithm 4 on ideal simulator 2

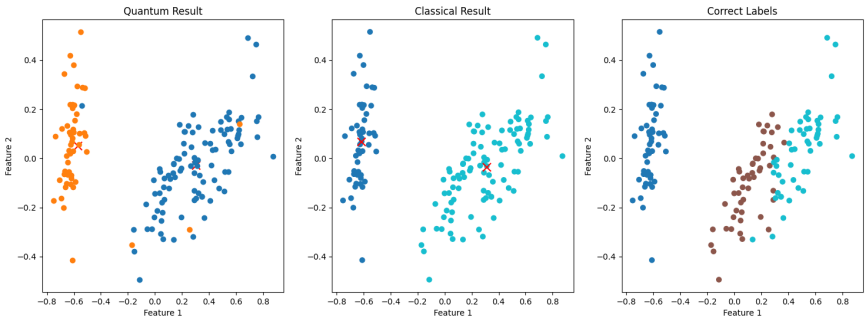


Figure 9. Experiment result of running Algorithm 4 on idea simulator 3

In both experiments, the quantum algorithm (Algorithm 4) produced lower accuracy clustering results than the classical algorithm, with error rates of $0{,}67\%$ and $2{,}67\%$

In general, the qubit-utilization quantum circuit (see Figure 3) produced nearly similar Euclidean distances to the results calculated classically, even in a real quantum system. Additionally, the quantum k-means clustering algorithm showed competitiveness compared to the classical k-means clustering algorithm when running on a noisy quantum simulator. However, when solving large-scale problems, quantum algorithm 4 produced lower accuracy results compared to the classical counterpart.

One potential reason for this is that when the size of the dataset is so large, the number of objects in each cluster is large. In this case, the small number of wrongly allocated objects with significant differences from other correctly allocated objects cannot cause a large change in the centroid. Therefore, the step of checking if convergence is reached cannot detect the allocation errors, and the algorithm stops with wrongly allocated objects. However, these low error rates might be complemented with computational efficiency and speedup of quantum computing in the future.

5. DISCUSSION

Quantum k-means clustering algorithms can be used to develop a trading approach that helps investors construct diversified portfolios and analyze stocks with high correlations in returns, categorizing them into specific groups[29]. Additionally, it can be applied in anomaly detection, such as identifying fraudulent credit card transactions and financial documents [29]. Moreover, an experiment conducted by DiAdamo et al. [30] applying the quantum k-means clustering algorithm to real-world German electricity grid data showed positive results. This demonstrates the algorithm's potential for application in predictive maintenance in the energy operations sector.

Quantum k-means clustering algorithms can provide exponential speedup in state preparation. This is because the algorithm uses amplitude encoding, where an N-dimensional vector is loaded using $\log N$ qubits, whereas in classical computing, an N-dimensional vector is encoded using N bits [6]. However, for subsequent operations, quantum computers offer no significant advantages. The issue lies in the lack of an efficient method for encoding the exact value of the data point. With the current approach, only the object's polar angle is encoded. Consequently, classical calculations are still necessary to compute the Euclidean distance from quantum results, preventing quantum computers from offering significant advantages in k-means clustering tasks. In the future, if a method for efficiently encoding a precise estimation of the data point is developed, one that accurately reflects the differences in distances between data points and each centroid, the advantages of quantum computing will become more apparent.

6. SUMMARY

In summary, the thesis reviewed a previous quantum approach for k-means clustering, pointed out its limitation, and suggested a method to address this limitation. Additionally, the thesis proposed an algorithm to calculate multiple distances simultaneously to enhance the efficiency of the quantum k-means clustering program.

Through the thesis studies, the following conclusions are obtained. First, it is feasible to employ quantum computers for calculating Euclidean distances, which are essential tasks in the k-means clustering algorithm. The Euclidean distances computed by quantum computers are nearly asymptotic to those calculated by classical computers. This is because the quantum circuit for computing Euclidean distance has low and constant circuit depth, thus mitigating severe noise. Additionally, it is possible to calculate multiple distances simultaneously in one quantum circuit, thereby enhancing efficiency. The quantum k-means clustering algorithm, which utilizes this quantum circuit for Euclidean distance calculation, performs equally well with classical k-means clustering algorithms in small-scale problems. However, in large-scale problems, the quantum k-means clustering algorithm produces slightly less accurate results. The experiments in this thesis did not account for the execution time factor. Nevertheless, theoretically, quantum computers would exponentially speed up the state preparation process for calculating quantum distance, as it requires only logarithmic qubits to represent the dataset. However, since classical calculation is still needed to compute the distance from the results obtained from the quantum computer, the quantum speedup for the whole task is not significant. Therefore, to utilize the advantages of quantum computing, a better method for encoding the data into the quantum circuit should be used.

7. REFERENCES

[1] OpenAI (2022), ChatGPT. URL: https://openai.com/chatgpt.

[2] OpenAI (2024), Sora. URL: https://openai.com/sora?ref=aihub.cn.

[3] Ciliberto C., Herbster M., Ialongo A.D., Pontil M., Rocchetto A., Severini S. & Wossnig L. (2018) Quantum machine learning: a classical perspective. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 474, p. 20170551.

[4] Markov I.L. (2014) Limits on fundamental limits to computation. Nature 512, p. 147–154.

[5] Mikio N. & Yoshitaka S. (2013) Quantum Information And Quantum Computing - Proceedings Of Symposium. No. vol. 6 in Kinki University Series on Quantum Computing, World Scientific.

[6] Khan S.U., Awan A.J. & Vall-Llosera G. (2019), K-means clustering on noisy intermediate scale quantum computers.

[7] Amin M.H., Andriyash E., Rolfe J., Kulchytskyy B. & Melko R. (2018) Quantum boltzmann machine. Phys. Rev. X 8, p. 021050.

[8] Rebentrost P., Mohseni M. & Lloyd S. (2014) Quantum support vector machine for big data classification. Physical Review Letters 113.

[9] Cong I., Choi S. & Lukin M.D. (2019) Quantum convolutional neural networks. Nature Physics 15, p. 1273–1278.

[10] Dong D., Chen C., Li H. & Tarn T.J. (2008) Quantum reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38, pp. 1207–1220.

[11] Lorenz R., Pearson A., Meichanetzidis K., Kartsaklis D. & Coecke B. (2023) Qnlp in practice: Running compositional models of meaning on a quantum computer. Journal of Artificial Intelligence Research 76, p. 1305–1342.

[12] Dabbura I. (2018), K-means clustering: Algorithm, applications, evaluation methods, and drawbacks. URL: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.

[13] Nanda S.J., Ishank Gulati R.C., Modi R. & Dhaked U. (2019) A k-means-galactic swarm optimization-based clustering algorithm with otsu's entropy for brain tumor detection. Applied Artificial Intelligence 33, pp. 152–170.

[14] Nandapala E. & Jayasena K.P.N. (2020) The practical approach in customers segmentation by using the k-means algorithm. 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS) , pp. 344–349.

[15] Zhu Z. & Liu N. (2021) Early warning of financial risk based on k-means clustering algorithm. Complex. 2021, pp. 5571683:1–5571683:12.

[16] Singh A., Mehta J., Anand D., Nath P., Pandey B. & Khamparia A. (2020) An intelligent hybrid approach for hepatitis disease diagnosis: Combining enhanced k -means clustering and improved ensemble learning. Expert Systems 38.

[17] Ikotun A.M., Ezugwu A.E., Abualigah L., Abuhaija B. & Heming J. (2023) K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences 622, pp. 178–210.

[18] Preskill J. (2018) Quantum computing in the nisq era and beyond. Quantum 2, p. 79.

[19] Karim M. A. & Jan R. M. (2005) Matrix Algebra. No. Vol. 1 in Econometric Exercises, Cambridge University Press.

[20] Daizhan C., Hongsheng Q. & Yin Z. (2012) Introduction To Semi-tensor Product Of Matrices And Its Applications, An. World Scientific.

[21] Adams R.A. & Essex C. (2022) Calculus: A Complete Course. Pearson, North York, Ontario, 10th ed.

[22] Kopczyk D. (2018), Quantum machine learning for data scientists.

[23] Hughes C., Isaacson J., Perry A., Sun R.F. & Turner J. (2021) Quantum Computing for the Quantum Curious. Springer Cham, 1 ed., XV, 150 p.

[24] Steinley D.L. (2006) K-means clustering: a half-century synthesis. The British journal of mathematical and statistical psychology 59 Pt 1, pp. 1–34.

[25] Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M. & Duchesnay E. (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, pp. 2825–2830.

[26] IBM, ibm_kyoto. URL: `https://quantum.ibm.com/services/resources?tab=systems&system=ibm_kyoto`.

[27] Qiskit, Qiskit Aer. URL: `https://github.com/Qiskit/qiskit-aer/blob/stable/0.11/qiskit_aer/backends/aer_simulator.py`.

[28] IBM, Using IBM quantum cloud-based simulators. URL: `https://docs.quantum.ibm.com/verify/using-ibm-quantum-simulators`.

[29] Herman D., Googin C., Liu X., Galda A., Safro I., Sun Y., Pistoia M. & Alexeev Y. (2022), A survey of quantum computing for finance.

[30] DiAdamo S., O'Meara C., Cortiana G. & Bernabé-Moreno J. (2022) Practical quantum k-means clustering: Performance analysis and applications in energy grid classification. IEEE Transactions on Quantum Engineering 3, pp. 1–16.