Quantum Hierarchical Agglomerative Clustering Based on One Dimension Discrete Quantum Walk with Single-Point Phase Defects

Gongde Guo[1], Kai Yu[1], Hui Wang[2], Song Lin[1, *], Yongzhen Xu[1] and Xiaofeng Chen[3]

Abstract: As an important branch of machine learning, clustering analysis is widely used in some fields, e.g., image pattern recognition, social network analysis, information security, and so on. In this paper, we consider the designing of clustering algorithm in quantum scenario, and propose a quantum hierarchical agglomerative clustering algorithm, which is based on one dimension discrete quantum walk with single-point phase defects. In the proposed algorithm, two nonclassical characters of this kind of quantum walk, localization and ballistic effects, are exploited. At first, each data point is viewed as a particle and performed this kind of quantum walk with a parameter, which is determined by its neighbors. After that, the particles are measured in a calculation basis. In terms of the measurement result, every attribute value of the corresponding data point is modified appropriately. In this way, each data point interacts with its neighbors and moves toward a certain center point. At last, this process is repeated several times until similar data points cluster together and form distinct classes. Simulation experiments on the synthetic and real world data demonstrate the effectiveness of the presented algorithm. Compared with some classical algorithms, the proposed algorithm achieves better clustering results. Moreover, combining quantum cluster assignment method, the presented algorithm can speed up the calculating velocity.

Keywords: Quantum machine learning, discrete quantum walk, hierarchical agglomerative clustering.

1 Introduction

As one of the most important fields in modern physics, quantum mechanics has not only changed the way we understand the physical world, but also provided a new method of solving some problems in the field of information technology [Liu, Xu, Yang et al.

[1] College of Mathematics and Informatics, Fujian Normal University, Fuzhou, 350007, China.

[2] School of Computing and Mathematics, University of Ulster, Northern Ireland, BT37 0QB, UK.

[3] School of Electronic Information Science, Fujian Jiangxia University, Fuzhou, 350108, China.

[*]Corresponding Author: Song Lin. Email: lins95@gmail.com.

Received: 06 May 2020; Accepted: 05 June 2020.

(2019); Jiang, Wang, Liang et al. (2020); Lin, Guo, Huang et al. (2016)]. After the pioneering works of Shor and Grover, various properties of quantum mechanics were utilized to design many subtle quantum algorithms [Montanaro (2016)], which can be exponentially faster than their classical counterparts. Machine learning enables computers to learn a certain hidden pattern from one data set and has a large variety of applications, such as image analysis, information retrieval, and bioinformatics, etc. However, if the number of the data set and/or the dimension of the data points are large, it is frequently required to cost a lot of times and huge computational resources. Especially, in the age of big data, this problem becomes more and more serious. Since quantum speed-up may be a good solution to this problem, quantum machine learning has recently been proposed and drawn a lot of attention [Biamonte, Wittek, Pancotti et al. (2017); Harrow, Hassidim and Lloyd (2009)].

In 2013, Lloyd et al. [Lloyd, Mohseni and Rebentrost (2013)] proposed two quantum machine learning algorithms. One is a supervised cluster assignment algorithm, the other is cluster finding algorithm that is used to obtain suitable seeds for quantum k-means algorithm. These two algorithms both offer an exponential speed-up over the corresponding classical counter-parts. Later on, Cai et al. [Cai, Wu, Su et al. (2015)] implemented them on a small-scale photonic quantum computer in experimental aspect. Besides that, some subtle quantum algorithms for machine learning have been put forward, e.g., quantum support vector machine [Rebentrost, Mohseni and Lloyd (2014); Li, Liu, Xu et al. (2015)], quantum decision tree [Lu and Braunstein (2014)], quantum nearest-neighbor algorithms [Wiebe, Kapoor and Svore (2015)], quantum principal component analysis [Lloyd, Mohseni and Rebentrost (2014); Yu, Gao, Lin et al. (2019)], quantum deep learning [Wiebe, Kapoor and Svore (2014)], quantum association rules mining [Yu, Gao, Wang et al. (2016)], quantum clustering [Aïmeur, Brassard and Gambs (2013); Li, He and Jiang (2011)], and so on.

Clustering analysis is an essential tool for knowledge discovery and becomes a major branch of machine learning [Chen, Xiong, Xu, et al. (2019); Zhou, Tan, Yu, et al. (2019); Xiang, Shen, Qin, et al. (2019)]. During the past decades, some subtle clustering algorithms were proposed from different points of view. In this paper, we consider hierarchical agglomerative clustering (called HAC), which is one main kind of clustering algorithm, in quantum scenario. Two features of quantum walks, localization and ballistic effect, are utilized to designed a quantum hierarchical agglomerative clustering algorithm. In this algorithm, each data point is represented by a particle that is firstly performed a one-dimensional discrete quantum walk with single-point phase defects. Here, the phase defect, which can govern the localization effect, is deter-mined by the local density of the corresponding data point. Then, one makes a measurement on this particle in a computational basis, and obtains a random outcome. Based on the outcome, every attribute of this data point is made an appropriate modification. After executing this process several times, the data points are clustered together and divided into several classes. Numerical simulations show the effectiveness and efficiency of the proposed quantum HAC algorithm.

The remainder of this paper is organized as follow. In Section 2, we briefly review the essential preliminaries, i.e., classical hierarchical agglomerative clustering algorithm and discrete quantum walk with single-point phase defects. Then, a quantum hierarchical agglomerative clustering algorithm with HWSPPD is described in Section 3. Its numerical simulations and experimental evaluation are presented in Section 4. Finally, a short conclusion is provided in Section 5.

2 Preliminaries

2.1 Hierarchical agglomerative clustering

As compared to some supervised machine learning algorithms, clustering is unsupervised, which means that the training data points are unlabeled. In general, the goal of clustering analysis is to classify the data points into categories on the basis of their similarity, namely, to group these data points such that the intra-cluster similarity is maximized and the inter-cluster similarity is minimized. A general mathematical representation of the cluster analysis is depicted as follows. Given a set of data points $D = \{X_1, X_2, \cdots, X_m\}$. After executing the clustering algorithm, these elements are classified into l subsets, denoted by $\{M_1, M_2, \cdots, M_l\}$. And these subsets satisfy the following constraints,

$$M_1 \cup M_2 \cup \cdots \cup M_l = D,\ M_j \cap M_k = \emptyset\ (j \neq k), \tag{1}$$

where $\emptyset$ represents an empty set.

Hierarchical clustering is a traditional clustering algorithm that seeks to build a hierarchy of clusters. Generally, it is divided into two types: agglomerative and divisive. In a standard hierarchical agglomerative clustering (HAC), each data point is thought as a cluster initially. Afterward, the distances of arbitrary two clusters are calculated, and two closest clusters are merged as one. This process is executed repeatedly until all data points cluster to one class or a certain terminate condition is satisfied. Finally, a hierarchy of clusters is built. Since any valid measure of distance can be used in HAC, it has been extensively used in data mining and statistics. However, this clustering algorithm should cost a lot of times. Suppose that a data set has m data points and the dimension of each point is n. By simple calculation, we know its complexity is $O(m^3 n)$. This implies that HAC is too slow for large data sets. Therefore, the most restraint of HAC is its high computational complexity. In this paper, we try to overcome this obstacle by proposing a quantum counterpart, in which some quantum technologies are utilized to speed-up the calculation.

2.2 Discrete quantum walk with single-point phase defects

As a quantum mechanical analog of classical random walk, quantum walk [Venegas-Andraca (2012)] has attracted a great deal of interesting recently. It has exhibited some distinct features, which can be used to design some new algorithms for quantum computers, e.g. quantum search algorithms. In a standard model, a discrete quantum walk on an infinite line can be depicted by a Hilbert space H, which consists of two spaces, i.e., $H = H_p \otimes H_c$. One is a two-dimensional coin space H_c, the computational basis of which is $\{|0\rangle, |1\rangle\}$ corresponding to two possible directions of movement, rightward and leftward. The other space H_p is a position space that is spanned by the orthogonal position vectors $\{|j\rangle \mid j \in \mathbf{Z}\}$, where $\mathbf{Z}$ denotes the set of integers. So, an orthonormal basis of the whole quantum system H is $\{|j, c\rangle = |j\rangle \otimes |c\rangle \mid j \in \mathbf{Z}, c = 0,1\}$.

The movement of the walker at each step is determined by the result of a coin flip, which is implemented by a unitary operation $C \in SU(2)$. Afterward, a conditional position shift operation S_c is performed. So, the whole one-step evolution can be depicted as, $U = (\sum_{c=0,1} S_c \otimes |c\rangle\langle c|)(I \otimes C)$. In this paper, we will adopt a common quantum walk, Hadamard walk. In this walk, C is a Hadamard operator and has the following form,

$$C = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2}$$

And the conditional position shift operation is $S_c|j\rangle = |j + (-1)^c\rangle$. Given the walker starts from the origin, so the whole system is in a initial state, $|\psi(0)\rangle = |0\rangle \otimes (\alpha_0^0|0\rangle + \beta_0^0|1\rangle)$, where $|\alpha_0^0|^2 + |\beta_0^0|^2 = 1$. After η steps evolution, the form of the final state is, $|\psi(\eta)\rangle = U^\eta|\psi(0)\rangle = \sum_{j=-\eta}^{+\eta}(\alpha_\eta^j|j,0\rangle + \beta_\eta^j|j,1\rangle)$. Thus, the probability of finding the walker at the position j at time η is $p_\eta^j = |\alpha_\eta^j|^2 + |\beta_\eta^j|^2$, satisfying $\sum_{j=-\eta}^{+\eta} p_\eta^j = 1$. For example, when the particles are in the initial state $|\psi(0)\rangle = |\varphi\rangle = |0\rangle \otimes (\frac{1}{\sqrt{2}}|0\rangle + \frac{-i}{\sqrt{2}}|1\rangle)$, the position probability distribution after $\eta = 100$ steps is depicted in Fig. 1(a).

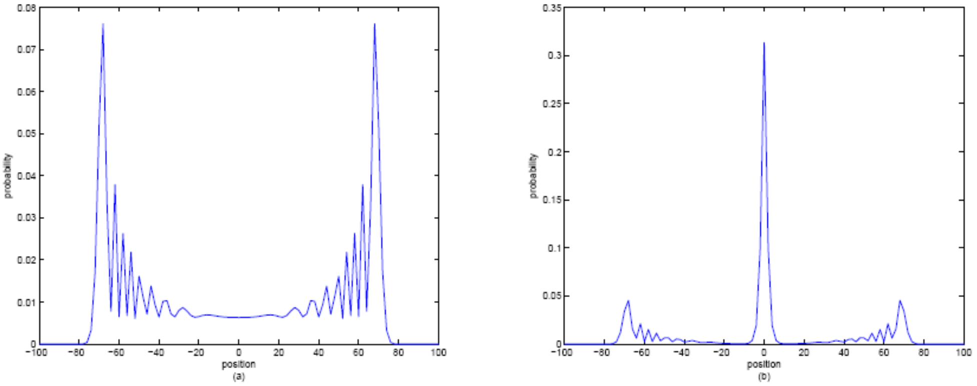


Figure 1: (a) Probability distribution after 100 steps of a Hadamard walk with the initial state $|\varphi\rangle$; (b) Probability distribution after 100 steps of a HWSPPD with $\theta = \pi/4$ and $|\psi(0)\rangle = |\varphi\rangle$

From this figure, it is shown that the movement of the quantum walker is ballistic. The phenomenon cannot exist in classical random walk, the distribution of which is a Gaussian centered at the origin.

Besides the ballistic diffusion, quantum walk has another nonclassical feature, localization, which has been found in some quantum walks [Schreiber, Cassemiro, Potocek et al. (2011)]. In 2012, Wojcik et al. [Wójcik, Łuczak, Kurzyński et al. (2012)] showed that localization effect can be obtained by changing a phase at a single point for discrete quantum walks without localization effect. In a discrete quantum walk with single-point phase defects, the phase of the particle is modified when it passes through a designed position e.g., $j = 0$, each time. Later, this issue has been studied deeply and made a great progress in both theory and experimental aspects [Zhang, Xue and Twamley (2014); Xue, Qin and Tang (2015)]. Considering Hadamard walk depicted above, a single-point phase shift $\theta \in (0,2\pi]$ is applied at the original point. That is, the operator S_c

is replaced by S_c^θ,

$$S_c^\theta = \exp\left(i\theta\delta_{j,0}\left|j+(-1)^c\right\rangle\right), \text{where } \delta_{j,0} = \begin{cases} 1, & \text{if } j=0 \\ 0, & \text{otherwise} \end{cases}. \quad (3)$$

In this way, a new quantum walk model is obtained (called HWSPPD). The localization effect of this quantum walk can be observed in Fig. 1(b).

3 Quantum HAC algorithm with HWSPPD

At first, a simple quantum hierarchical agglomerative clustering algorithm (Algorithm 1) is given, which can be derived directly from the conclusion of Ref. [Lloyd, Mohseni and Rebentrost (2013)]. In Lloyd et al. [Lloyd, Mohseni and Rebentrost (2013)], Lloyd et al. designed a quantum cluster assignment (called QCA) algorithm that accomplishes assigning a new data point to one of two sets by calculating the Euclidean distances between a new data point and two sets. Moreover, since this calculation only costs time $O(log(mn))$ on a quantum computer, this quantum algorithm provides an exponential speed-up over classical algorithms that should take time $O(mn)$. Drawing ideas from QCA algorithm, the calculating process of the Euclidean distance between two clusters can been obtained.

In this process, two clusters, $\{u_i \mid i = 1,2,\cdots,p\}$ and $\{v_j \mid j = 1,2,\cdots,q\}$ are given. First, qRAM is utilized to construct the state $|\phi\rangle_{12} = \frac{1}{\sqrt{2}}(|0\rangle|u\rangle + (\frac{1}{\sqrt{q}})\sum_{j=1}^{q}|j\rangle|v_j\rangle)_{12}$, where $|u\rangle = \frac{1}{\sqrt{p}}\sum_{i=1}^{p}|u_i\rangle$. Then, a projective measurement is performed on the first particle. Finally, according to the probability of the case, in which the measurement result is $|u||0\rangle - (\frac{1}{\sqrt{q}})\sum_{j=1}^{q}|v_j||j\rangle$, the distance between these two clusters can be calculated with time $O(log(mn))$. Hence, the complexity of Algorithm 1 will be reduced to $O(m^2log(mn))$. Additionally, we assume the termination condition is that the minimum of the distances is larger than a threshold ϵ. The detailed algorithm is described as follows.

Algorithm 1 *M*=QHAC(*D*, ϵ)

```
Input: $D = \{X_1, X_2, \cdots, X_m\}$, $\epsilon$
Output: $M$
  for $k$=1; $k \leq m$; $k$++ do
    $M[k] \leftarrow k$
  end for;
  $count \leftarrow m$;
  $loopFlag \leftarrow$ True;
  while $loopFlag$ do
    Compute the distance between any two clusters by QCA;
    if the minimum of the distances < $\epsilon$ then
      Combine the corresponding cluster $j$ and cluster $k$, $M[j] \leftarrow M[k]$;
      $count \leftarrow count - 1$
    else
      $loopFlag \leftarrow$ False;
    end if;
```

```
        if count=1 then
            $loopFlag$ ← False;
        end if;
    end while;
    return $M$;
```

General speaking, in a data set, one data point is more related to nearby points than to points farther away. So, these date points are divided into three kinds: the center points, the border points, and the outlier points (or the noisy points). For example, in a two-dimensional data set DS1 shown in Fig. 2, there are two clusters, $\{X_1, X_2, \cdots, X_9\}$ and $\{X_{10}, X_{11}, \cdots, X_{18}\}$, and two outlier points, X_{19} and X_{20}. Consider three data points, X_{15}, X_{18}, and X_{19}, where X_{15} is the center point, who has many neighbors and is surrounded by them. In contrast, X_{19} is the outlier point, who has few neighbors and is isolated from the other points. For the point X_{18}, its neighborhood contains some points, and these neighbors are located towards the center points, thus X_{18} is named as the border point.

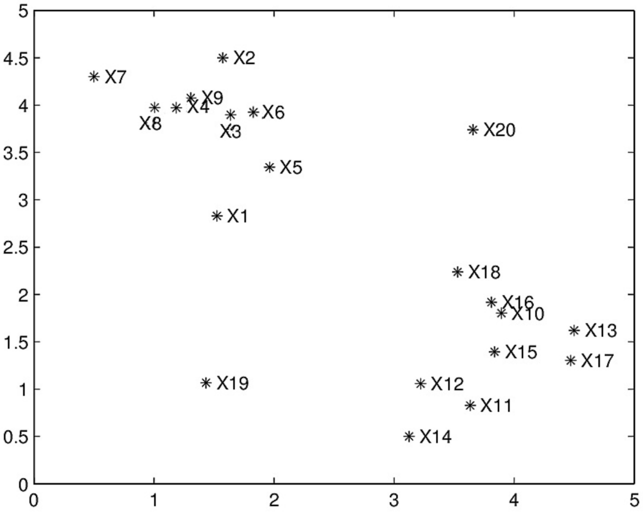


Figure 2: Point distribution

The second quantum hierarchical agglomerative clustering algorithm has its basis in this observation. In the algorithm, each data point is considered as a walker particle. According to the difference among these three kinds of data points, the corresponding particle performs a HWSPPD with different values of θ. It is determined by the density of its neighbors. Concretely, for the particles represented the center points or the outlier, the location effect of quantum walk is adopted to cause these particles move slowly or keep them motionless. While, for the border points, the ballistic effect is chosen to make the corresponding particles move towards the center points quickly. In this way, two nonclassical features of quantum walks are utilized to achieve clustering task.

Before presenting our clustering algorithm, we need to define some notions that are used in our algorithm. Suppose that there exists an unlabel data set with m data points denoted by $D = \{X_1, X_2, \cdots, X_m\}$, and each data point X_j has n attributes, $X_j = (x_j^1, x_j^2, \cdots, x_j^n)$. So, the

Euclidean distance between two data points, $X_j = (x_j^1, x_j^2, \cdots, x_j^n)$ and $X_k = (x_k^1, x_k^2, \cdots, x_k^n)$, is defined as $d(X_j, X_k) = \sqrt{\sum_{l=1}^{n}(x_j^l - x_k^l)^2}$. Based on this distance definition, the neighborhood of every data point can be directly obtained. Concretely, the ϵ −neighborhood of a data point X_j can be written as, $\Xi_j = \{X_k \in D \mid d(X_j, X_k) \leq \epsilon\}$. Then, a new quantity ξ_j is defined to represent the number of point X_j's neighbors, i.e., $\xi_j = |\Xi_j|$. This quantity was used and named as local density in Ref. [Rodriguez and Laio (2014)].

In the second algorithm, each data point is considered as a particle. For convenience, we can assume that the corresponding particle of a data point X_j is P_j. This particle is prepared in the initial state $|\psi(0)\rangle$. At each step, particle P_j performs a HWSPPD with a parameter θ firstly. Generally speaking, one data point just interacts with its neighborhood points. Thus, in our algorithm, the parameter θ_j is determined by the neighborhood of the data point X_j. Concretely, θ_j is calculated as follows.

$$\theta_j = \frac{\Lambda - \lambda_j}{\Lambda} \times 0.7\pi, \tag{4}$$

where,

$$\Lambda = \max_{k=1,2,\cdots,m}(\lambda_k), \lambda_j = \begin{cases} 0, & \text{if } \xi_j = 0 \\ \left|\max_{X_k \in \Xi_j}(\xi_k) - \xi_j\right|, & \text{otherwise} \end{cases}. \tag{5}$$

After the walker P_j makes a HWSPPD with $\theta = \theta_j$, this particle is measured in the computational basis. Finally, according to the measurement result r_j, the attributes of X_j are changed. The detailed modification is described as, $x_j^l = x_j^l + |r_j| \times \tau_j^l$, where $\tau_j^l = \frac{1}{2n+1} \times (\frac{\sum_{X_k \in \Xi_j} x_k^l}{\xi_j} - x_j^l)$, and $\tau_j = (\tau_j^1, \tau_j^2, \cdots, \tau_j^n)$ represents the step length of point X_j.

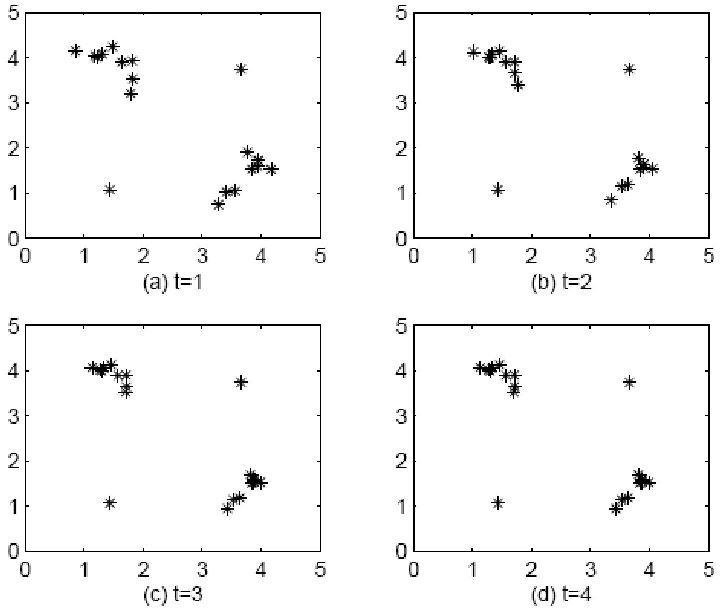
Here, the setups of these two quantities, θ_j and τ_j^l, are based on a general assumption. Namely, the cluster centers with higher local density are surrounded by neighbors with lower density. If the data point X_j is the center point, there generally exists another data point X_k in its neighborhood Ξ_j that is very close to the center point. Moreover, it is common that the local density of the data point X_k is equal or only slightly less than that of the center point i.e., $\xi_k \simeq \xi_j$. Thus, the quantity λ_j=$|\xi_k - \xi_j|$ approaches to 0, then $\theta_j \simeq 0.7\pi$. In this case, the localization effect takes action when particle P_j performs a HWSPPD with $\theta \simeq 0.7\pi$. On the other hand, the neighbors of point X_j are located around it symmetrically, i.e., $\frac{\sum_{X_k \in \Xi_j} x_k^l}{\xi_j} \simeq x_j^l$. It implies that the value of τ_j^l is also close to 0. Thus, the center point X_j is kept unchanged with high probability. The similar scenario occurs when point X_j is the outlier point. The reason is that this point has few neighbors, i.e., $\xi_j \simeq 0$ and $\tau_l^j \simeq 0$. Therefore, the outlier point is also steady in our algorithm.

However, it is going to be different when X_j is the border point. Generally, there may exist a data point $X_k \in \Xi_j$, who is the center point or near the center point. So, $\lambda_j \neq 0$, and the ballistic phenomenon will be found during the quantum walk of the corresponding particle. Moreover, since the neighbors of the border point X_j are located towards the center point, the corresponding τ_j^l of point X_j is larger than that of the center point. Under this condition, the border point X_j will move towards the center point. Further, considering two border points X_{j_1} and X_{j_2}, where point X_{j_1} is more close to the center point X_k than point X_{j_2}. This implies that the distance between points X_{j_2} and X_k is larger than that of X_{j_1} and X_k. In this case, the value of λ_{j_1} is less than that of λ_{j_2}, because it is common that point X_{j_1} has more neighbors than point X_{j_2}, i.e., $\xi_{j_1} \geq \xi_{j_2}$. Therefore, we can obtain $\theta_{j_1} \geq \theta_{j_2}$. Furthermore, it is evident that $|\tau_{j_1}^l| \leq |\tau_{j_2}^l|$. Hence, as compared with point X_{j_1}, point X_{j_2} moves more quick towards the center.

In the above manner, all points expect the outlier point get together after executing this process several times. According to this basic idea and Algorithm 1, we can obtain the second quantum HAC algorithm (Algorithm 2), which is described as follows.

```
Algorithm 2 M=QHACQW(D, ϵ)
Input: $D = \{X_1, X_2, \cdots, X_m\}$, $\epsilon$, $m \times K$ particles $P_j$ $(j = 1,2,\cdots,m)$ are prepared in the initial state $|\varphi\rangle$
Output: M
  $loopFlag \leftarrow$ True;
  while $loopFlag$ do
    $loopFlag \leftarrow$ False;
    for all $X_j \in D$ do
      Compute its neighbor set $\Xi_j$;
      Obtain the local probability $\xi_j$;
      Compute the parameter $\theta_j$;
      if $\theta_j < 0.7\pi$ then
        $loopFlag \leftarrow$ True;
      end if;
    end for;
    if $loopFlag =$ False then
      $D^* \leftarrow D$;
      $M = QHAC(D^*, \epsilon)$;
      return M;
    end if;
    for j=1; $j \leq m$; j++ do
      if $\theta_j = 0.7\pi$ then
        continue;
      end if;
      Particle $P_j$ performs a HWSPPD with $\theta = \theta_j$;
      Measure particle $P_j$ and obtain the result $r_j$;
      Obtain new value of data point $X_j$;
    end for;
  end while;
```

Now, let us consider the data point set in Fig. 2. After the iterative process is executed one round, the border points, e.g. points X_{16} and X_{18}, move toward the center point X_{15}, whereas X_{15} does not move much and the outlier point X_{19} is steady [as shown in Fig. 3(a)]. Moreover, point X_{18} moves more quick than point X_{16}. After four iterations, from Fig. 3(d), it is shown that all data points cluster together except two outlier points X_{19} and X_{20}. Consequently, Algorithm 2 achieves the clustering task successfully.



(a) t=1

(b) t=2

(c) t=3

(d) t=4

Figure 3: The cluster result of the data set DS1 with different iteration round t

4 Numerical simulations and experimental evaluation

In this section, we implement the presented algorithms by numerical simulation on a classical computer. From the clustering results on synthetic and real-world data, the performance of **Algorithm 2** is evaluated. Because it facilitates the representation and manipulation of matrices, MATLAB is frequently used to simulate quantum states and operations. Therefore, the presented algorithms are programmed by MATLAB, and executed on a personal computer with Intel(R) Core (TM) i5-4590 CPU 3.30 GHz and 8.0 GB RAM.

Let us start with conducting experiments on two synthetic data sets DS2 and DS3, which are displayed in Fig. 4(a) and Fig. 4(b). The first data set consists of four arbitrarily shaped clusters, each of which has 100 data points. The second one is comprised of four clusters with different densities. When ϵ is 0.14 (0.16), all data points in the set DS2 (DS3) are clustered into four classes by executing **Algorithm 2**. The corresponding clustering results are shown in Figs. 4(c) and 4(d). This implies that the presented algorithm successfully detects all types of clusters without any errors.

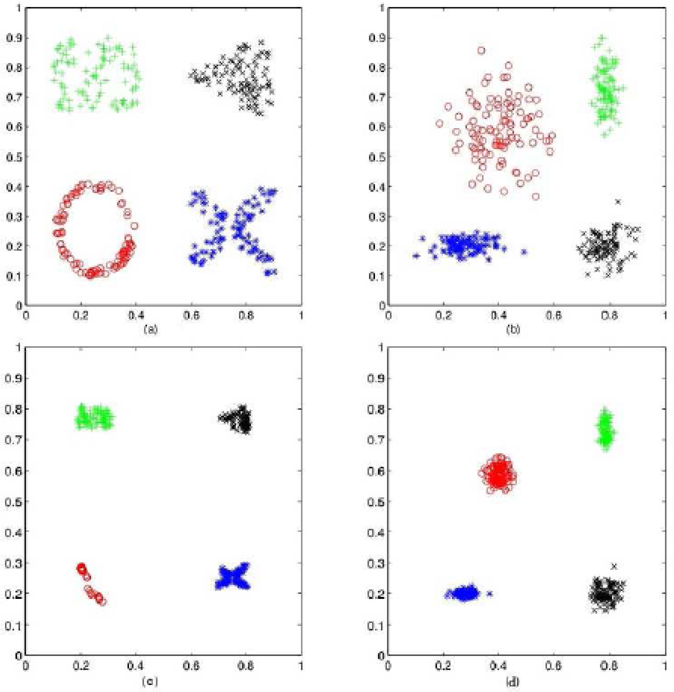


Figure 4: Clustering results of two synthetic data sets

In the following, we consider the other case, in which four real-world data publicly available at the UCI machine learning repository (http://archive.ics.uci.edu/ml), i.e., Wisconsin, Iris, Ecoli, and Wine, are clustered via **Algorithm 2**. To evaluate it better, the simulation experiment results of this algorithm are compared with that of two classical clustering algorithms, X-Means and MeanShift.

Furthermore, to provide an objective description of effectiveness, we use the normalized mutual information (NMI) as a measure for clustering quality. It is defined as, $\mathrm{NMI}(M,M')=\frac{I(M,M')}{\sqrt{H(M)H(M')}}$, where M and M' are two clustering results of a data set. $H(M)$ is the entropy associated with the clustering $M=\{M_1,M_2,\cdots,M_l\}$, i.e., $H(M)=-\sum_{j=1}^{l} prob_j \log(prob_j)$ where $prob_j=\frac{|M_j|}{m}$. $I(M,M')$ is the mutual information between these two clusters. The value range of $\mathrm{NMI}(M,M')$ is between 0 and 1. The higher the value, the better the clustering effect.

Derived from the study on breast cancer, the Wisconsin data set comprises of two classes. One is benign with 444 instances, the other is malignant with 239 instances. Each instance has 9 attributes. By Algorithm 2, this data set is clustered into two classes successfully. One cluster with 457 instances represents the class benign, among which 19 instances have been wrongly labeled. The other with 226 instances is malignant and has 6 wrong results. In total, there are only 25 instances wrongly clustered. It is better than the performance of the other two algorithms, which is shown in Tab. 1.

Besides, we have considered the other three real-world data sets, Iris (with 150 instances and 4 attributes), Ecoli (with 178 instances and 13 attributes) and Wine (with 336 instances and 7 attributes). The corresponding comparison results of three algorithms are listed in Tab. 2. Here, we execute the algorithms 50 times and obtain the corresponding average values. The comparison results verify the effectiveness of **Algorithm 2**.

Table 1: Comparison on real-world data sets with various clustering algorithms

NMI	**Algorithm 2**	X-means	MeanShift
Wisconsin	**0.755**	0.561	0.700
Iris	**0.761**	0.734	0.734
Ecoli	**0.706**	0.512	0.546
Wine	**0.732**	0.248	0.405

5 Conclusion

In summary, based on one-dimension discrete quantum walk with single-point phase defects, a new quantum hierarchical agglomerative clustering algorithm is introduced. Each data point is regarded as a particle that performs a HWSPPD with a parameter θ. Here, this parameter that can control the localization effect of this walk is determined by the local density of this data point. Then, this particle is measured. According to the measurement result, the corresponding data point makes an appropriate modification. In this way, each data point interacts with its neighbors. As time evolves, similar data points cluster together and form distinct classes. To illustrate the effectiveness of this algorithm, extensive simulation experiments on the synthetic and real world data are performed. Furthermore, in the presented algorithm, quantum cluster assignment method is utilized to speed up the calculating velocity. Hence, our approach is efficient.

In addition, there are two key technology problems in the implementation of the presented algorithm. One is the achievement of the quantum cluster assignment method. The experiment of this method has been accomplished by Cai et al. [Cai, Wu, Su et al. (2015)] on a small-scale photonic quantum computer. The other one is that of quantum walk with single-point phase defects. The corresponding experiment has also been achieved by Xue et al. [Xue, Qin and Tang (2015)] with optical interferometers. Therefore, the presented algorithm is experimentally feasible with current technology.

Funding Statement: This work was supported by National Natural Science Foundation o f China (Grants Nos. 61976053 and 61772134), Fujian Province Natural Science Foundat ion (Grant No. 2018J01776), Program for New Century Excellent Talents in Fujian Provi nce University, Probability and Statistics: Theory and Application (Grant No. IRTL1704), and the Program for Innovative Research Team in Science and Technology in Fujian Pro vince University.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Aïmeur, E.; Brassard, G.; Gambs, S. (2013): Quantum speed-up for unsupervised learning. *Machine Learning*, vol. 90, pp. 261-287.

Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N. et al. (2017): Quantum machine learning. *Nature*, vol. 549, pp. 195-202.

Cai, X. D.; Wu, D.; Su, Z. E.; Chen, M. C.; Wang, X. L. et al. (2015): Entanglement-based machine learning on a quantum computer. *Physical Review Letters*, vol. 114, no. 11, pp. 110504-110505.

Chen, Y. T.; Xiong, J.; Xu, W. H.; Zuo, J. W. (2019): A novel online incremental and decremental learning algorithm based on variable support vector machine. *Cluster Computing*, vol. 22, no. 3, pp. 7435-7445.

Harrow, A. W.; Hassidim, A.; Lloyd, S. (2009): Quantum algorithm for linear systems of equations. *Physical Review Letters*, vol. 103, no. 15, pp. 150502-150503.

Jiang, D. H.; Wang, J.; Liang, X. Q.; Xu, G. B.; Qi, H. F. (2020): Quantum voting scheme based on locally indistinguishable orthogonal product states. *International Journal of Theoretical Physics*, vol. 59, pp. 436-444.

Li, Q.; He, Y.; Jiang, J. P. (2011): A hybrid classical-quantum clustering algorithm based on quantum walks. *Quantum Information Processing*, vol. 10, no. 1, pp. 13-26.

Li, Z. K.; Liu, X. M.; Xu, N. Y.; Du, J. F. (2015): Experimental realization of a quantum support vector machine. *Physical Review Letters*, vol. 114, no. 14, pp. 140504.1-140504.5.

Lin, S.; Guo, G. D.; Huang, F.; Liu, X. F. (2016): Quantum anonymous ranking based on the Chinese remainder theorem. *Physical Review A*, vol. 93, pp. 012318.

Liu, W. J.; Xu, Y.; Yang, J. C. N.; Yu, W. B.; Chi, L. H. (2019): Privacy-preserving quantum two-party geometric intersection. *Computers Materials & Continua*, vol. 58, no. 2, pp. 1237-1250.

Lloyd, S.; Mohseni, M.; Rebentrost, P. (2013): Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint quant-ph*, 1307.0411.

Lloyd, S.; Mohseni, M.; Rebentrost, P. (2014): Quantum principal component analysis. *Nature Physics*, vol. 10, pp. 631-633.

Lu, S.; Braunstein, S. L. (2014): Quantum decision tree classifier. *Quantum Information Processing*, vol. 13, pp. 757-770.

Montanaro, A. (2016): Quantum algorithms: an overview. *NPJ Quantum Information*, vol. 2, pp. 1-8.

Rebentrost, P.; Mohseni, M.; Lloyd, S. (2014): Quantum support vector machine for big data classification. *Physical Review Letters*, vol. 113, no. 13, pp. 130503-130507.

Rodriguez, A.; Laio, A. (2014): Clustering by fast search and find of density peaks. *Science*, vol. 344, no. 6191, pp. 1492-1496.

Schreiber, A.; Cassemiro, K. N.; Potocek, V.; Gabris, A.; Jex, I. et al. (2011): Decoherence and disorder in quantum walks: from ballistic spread to localization. *Physical Review Letters*, vol. 106, pp. 180403.1-180403.4.

Venegas-Andraca, S. E. (2012): Quantum walks: a comprehensive review. *Quantum Information Processing*, vol. 11, pp. 1015-1106.

Wiebe, N.; Kapoor, A.; Svore, K. M. (2014): Quantum deep learning. *arXiv preprint quant-ph*, 1412. 3489.

Wiebe, N.; Kapoor, A.; Svore, K. M. (2015): Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Information & Computation*, vol. 15, no. 3, pp. 0318-0358.

Wójcik, A.; Łuczak, T.; Kurzyński, P.; Grudka, A.; Gdala, T. et al. (2012): Trapping a particle of a quantum walk on the line. *Physical Review A*, vol. 85, no. 1, pp. 1-5.

Xiang, L. Y.; Shen, X. B.; Qin, J. H.; Hao, W. (2019): Discrete multi-graph hashing for large-scale visual search. *Neural Processing Letters*, vol. 49, no. 3, pp. 1055-1069.

Xue, P.; Qin, H.; Tang, B. (2015): Trapping photons on the line: controllable dynamics of a quantum walk. *Scientific Reports*, vol. 4, pp. 4825-4830.

Yu, C. H.; Gao, F.; Lin, S.; Wang, J. B. (2019): Quantum data compression by principal component analysis. *Quantum Information Processing*, vol. 18, no. 8, pp. 249-269.

Yu, C. H.; Gao, F.; Wang, Q. L.; Wen, Q. Y. (2016): Quantum algorithm for association rules mining. *Physical Review A*, vol. 94, no. 4, pp. 042311-042315.

Zhang, R.; Xue, P.; Twamley, J. (2014): One-dimensional quantum walks with single-point phase defects. *Physical Review A*, vol. 89, no. 4, pp. 10091-10096.

Zhou, L. L.; Tan, F.; Yu, F.; Liu, W. (2019): Cluster synchronization of two-layer nonlinearly coupled multiplex networks with multi-links and time-delays. *Neurocomputing*, vol. 359, pp. 264-275.