



Article

# Quantum Density Peak Clustering Algorithm

Zhihao Wu <sup>1</sup>, Tingting Song <sup>2,3,\*</sup> and Yanbing Zhang <sup>2</sup>

<sup>1</sup> College of Cyber Security, Jinan University, Guangzhou 510632, China; zhihaowu@stu2019.jnu.edu.cn

<sup>2</sup> College of Information Science and Technology, Jinan University, Guangzhou 510632, China; yanbing@stu2019.jnu.edu.cn

<sup>3</sup> Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China

\* Correspondence: tingtingsong@jnu.edu.cn

**Abstract:** A widely used clustering algorithm, density peak clustering (DPC), assigns different attribute values to data points through the distance between data points, and then determines the number and range of clustering by attribute values. However, DPC is inefficient when dealing with scenes with a large amount of data, and the range of parameters is not easy to determine. To fix these problems, we propose a quantum DPC (QDPC) algorithm based on a quantum *DistCalc* circuit and a Grover circuit. The time complexity is reduced to  $O(\log(N^2) + 6N + \sqrt{N})$ , whereas that of the traditional algorithm is  $O(N^2)$ . The space complexity is also decreased from  $O(N \cdot \lceil \log N \rceil)$  to  $O(\lceil \log N \rceil)$ .

**Keywords:** quantum information; quantum computation; quantum algorithm

## 1. Introduction

Cluster analysis originated from taxonomy, as an ancient skill mastered by human beings. In the past, people used to classify goods based on their experience and professional knowledge. With the development of modern society, people have higher and higher requirements for classification [1,2]. Classification based only on experience and professional knowledge has been gradually eliminated, and computer technology is now used for cluster analysis, using algorithms to address huge and complex cluster tasks [3,4]. Therefore, clustering algorithms have been proposed for applications in various settings [5,6]. Moreover, the world of massive data that we live in also makes the clustering process indispensable. Many research fields are faced with the problem of a large amount of data [7,8]. If there is no preprocessing such as clustering or data dimension reduction, it is difficult to carry out subsequent analysis [9–11]. For example, in the area of machine learning, the original entry of almost all important algorithms is a large amount of large-scale data. It is difficult to use these data without clustering or a dimensionality reduction [12–14]. In the field of quantum communication, quantum communication equipment is only supplied to few large companies. Many parties in quantum communication may be classical. Clustering algorithms can help communication parties to deal with the transmitted information more conveniently [15–17]. In the area of data dimension reduction, we are familiar with the principal component analysis algorithm (PCA) [18], multidimensional scaling (MDS), linear discrimination (LDA), locally linear embedding (LLE), and so on [19–22]. However, the dimension reduction algorithm will inevitably reduce the attribute value of the data. If the operation is improper, the data will lose accuracy and the results will have deviated. A clustering algorithm can be used to avoid such problems. Nowadays, clustering algorithms can be divided in the following way.

Partition-based clustering algorithms include K-means [23], K-medians [24], and kernel K-means algorithms [25]. Hierarchy-based clustering algorithms include BIRCH, CURE, and the CHAMELEON algorithm [26]. Density-based clustering algorithms include DBSCAN, mean-shift (MS) [27], and the density peak clustering algorithm (DPC) [28]. Each



**Citation:** Wu, Z.; Song, T.; Zhang, Y. Quantum Density Peak Clustering Algorithm. *Entropy* **2022**, *24*, 237. <https://doi.org/10.3390/e24020237>

Academic Editors: Xiang-Bin Wang, Cong Jiang and Leong Chuan Kwek

Received: 17 December 2021

Accepted: 1 February 2022

Published: 3 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

clustering algorithm has its own advantages and disadvantages, and each algorithm has its own suitable scenarios [29]. The advantage of the DPC algorithm is that there is no need to define the number of clusters, as in the K-means algorithm. Secondly, it can detect non-spherical data, which has high application value in computer image processing. In addition, it can automatically identify abnormal points, which is also a prominent advantage of many clustering algorithms.

In 2014, Rodriguez and Laio proposed a DPC algorithm, which can automatically find the cluster center and achieve efficient clustering of arbitrarily shaped data sets [30]. DPC is a clustering algorithm based on density, and its input parameters are less than those of the K-means algorithm [31,32] and the K-medians algorithm [33,34]. The process of DPC clustering does not need to map data to vector space, which reduces the computational complexity of the algorithm.

However, the DPC algorithm still has its drawbacks. When it deals with large amounts of data, the speed of the algorithm is significantly reduced. The algorithm computes the distance between the current data point and each data point in the set, so the complexity of this operation is  $O(N^2)$ , whereas  $N$  represents the number of data points [35]. At the same time, the process of the algorithm stores the distance between each data point and its remaining data points, which requires a large amount of storage space.

Some years ago, quantum technology was introduced to speed up the classical algorithms with large data volume, such as the Internet of Things industry and computer vision [36–38]. Typical quantum algorithms include the quantum K-means algorithm [32] and quantum principal component analysis [18]. They are not simple quantum version of classical algorithms. The running speeds of these quantum algorithms are greatly reduced. In this paper, we propose a QDPC algorithm, which applies a quantum *DistCalc* circuit to speed up the DPC algorithm.

In Section 2, the principle and flow of the classical DPC algorithm are introduced in detail. In Section 3, we propose the QDPC algorithm and its corresponding quantum circuit. In Section 4, the simulation experiments are discussed. An analysis of complexity and our conclusions are presented in Section 5.

## 2. Preliminary

### 2.1. Notation and Definitions

DPC is an algorithm that does not require iteration and can find the clustering center in one run. Distance information is the most important form of information that one must collect in the DPC algorithm. Based on the distance, one can compute the local density value.

The main ideas of DPC are based on the following assumptions:

- \* The clustering center has a relatively high local density value and is surrounded by data points with a low local density value.
- \* The clustering center is far away from any point with a higher local density value.

For each data point  $x_i$ , the algorithm computes two attribute values of the data point: its local density  $\rho_i$  and its distance  $\delta_i$  from the nearest higher density point. Both attribute values depend only on the distance  $d_{ij}$  between the current data point  $x_i$  and the rest of the data points  $x_j$ .

The local density  $\rho_i$  of data point  $x_i$  is defined as

$$\rho_i = \sum_j \chi(d_{ij} - d_c), \quad (1)$$

where  $d_{ij}$  is the distance between data point  $x_i$  and  $x_j$ ,  $d_c$  is the cutoff distance. The function  $\chi$  is defined as

$$\chi(x) = \begin{cases} 1 & x < 0 \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

which indicates the number of data points with distance from the data point  $x_i$  less than the cutoff distance.

The distance from the higher density point of data point  $x_i$  is defined as

$$\delta_i = \min_{j, \rho_j > \rho_i} (d_{ij}), \quad (3)$$

where  $\delta_i$  records the nearest distance from data point  $x_i$  to all data points with higher local density. If  $\delta_i$  is very small, there is a data point  $x_j$  with a higher local density around this data point  $x_i$ . As for the data point with the highest local density, it cannot find a data point with a higher local density, and its distance is  $\delta_i = \max_j(d_{ij})$  conventionally. It can be found that when the distance  $\delta_i$  is relatively large, this data point is the clustering center.

Therefore, after the two attribute values  $\rho_i$  and  $\delta_i$  of each data point are obtained, these data points are divided according to the rules.

- If the value of  $\rho_i$  and  $\delta_i$  are both anomalously large, it is the clustering center;
- If the value of  $\rho_i$  is relatively large and  $\delta_i$  is relatively small, it is the point in a cluster;
- If the value of  $\rho_i$  is relatively small and  $\delta_i$  is relatively large, it is an outlier.

According to the above rules, the algorithm can accurately find every clustering center of the cluster and cluster each data point.

## 2.2. The Workflow of the Classical DPC Algorithm

The main processes of the algorithm consist of calculating two attribute  $\rho_i$  and  $\delta_i$  values of each data point. Suppose we have a data set with large amounts of data points  $D = \{x_1, x_2, x_3, \dots, x_N\}$ , and the dimension of each data point is  $d$ . The steps of the DPC algorithm are as follows:

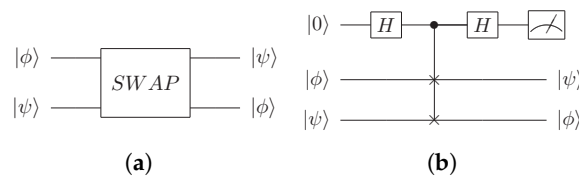
- a. Calculate the local density  $\rho_i$  of each data point  $x_i$ .
- b. For each data point  $x_i$ , the nearest distance of  $x_i$  is found in all data points with higher local density than  $x_i$ , and record this distance as  $\delta_i$ .
- c. According to  $\rho_i$  and  $\delta_i$  of each data point to determine the clustering center. If  $\rho_i$  and  $\delta_i$  of a data point are relatively large, it is the clustering center.
- d. Assign each data point to the nearest clustering center.

It should be noted that if  $\delta_i$  of a data point is large and  $\rho_i$  is small, then the point is an exception. It does not need to be assigned to any cluster.

## 3. QDPC Algorithm

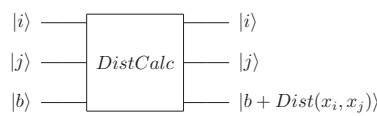
The classical algorithm takes the largest proportion of time to calculate the distance in the whole algorithm, so quantum circuits are used to optimize this part [39,40]. In quantum technology, fidelity is an important concept, which is similar to cosine similarity [41,42] in the classical framework. Fidelity can measure the similarity between two quantum states. If the value of fidelity is 1, the two quantum states are the same; if the value of fidelity is 0, the two quantum states are orthogonal. Therefore, the distance between data points can be calculated via fidelity only if the classical data are encoded into a quantum state.

The most commonly used quantum circuit to achieve fidelity is the *SwapTest*. This quantum circuit was proposed by Aïmeur et al in [43]. By taking the inner product of two quantum states  $|\phi\rangle$  and  $|\psi\rangle$ , the *SwapTest* circuit is used to calculate the fidelity of quantum states, as shown in Figure 1.



**Figure 1.** (a) Quantum *SwapTest* circuit to obtain the similarity between two quantum states  $|\phi\rangle$  and  $|\psi\rangle$ . (b) Details of quantum *SwapTest* circuit.

Based on the *SwapTest* circuit, the quantum *DistCalc* circuit [44] in Figure 2 can calculate the distance between data points  $x_i$  and  $x_j$ . The distance is stored in the third register.

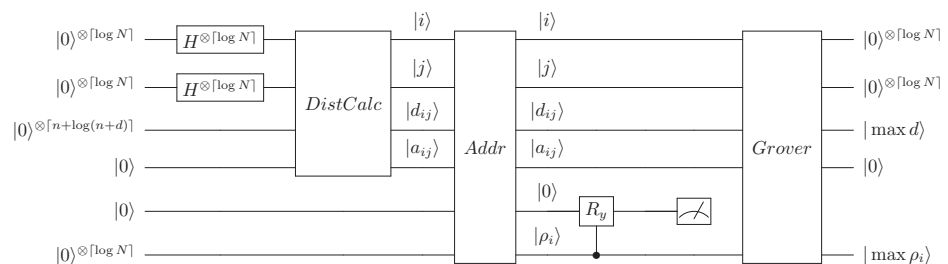


**Figure 2.** Quantum *DistCalc* circuit for calculating the distance between  $x_i$  and  $x_j$ .

*Procedure of the QDPC Algorithm*

Consider a set with  $N$  data points  $D = \{x_1, x_2, x_3, \dots, x_N\}$ . The dimension of each data point is  $d$ . Regardless of the number of clusters, QDPC will calculate two attribute values  $\rho_i$  and  $\delta_i$  for each data point. Then the clustering center is determined using these two attribute values.

An overview of the circuit for solving the QDPC is shown in Figure 3.



**Figure 3.** Overview of the quantum circuit for the QDPC algorithm, where  $|d_{ij}\rangle$  represents  $|d(x_i, x_j)\rangle$ .

The procedure used to cluster  $x_i$  includes the following seven steps:

- (i). Prepare six registers in  $|0\rangle^{\otimes \lceil \log N \rceil} \otimes |0\rangle^{\otimes \lceil \log N \rceil} \otimes |0\rangle^{\otimes \lceil n + \log(n+d) \rceil} \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle^{\otimes \lceil \log N \rceil}$ , and apply an  $H$  gate on each qubit in the first and second registers. The third register records the quantum state of the distance from two data points  $x_i$  and  $x_j$ . The fourth register stores the intermediate conversion value  $a_{ij}$ , which will be explained in more detail later. The fifth register is an ancillary qubit. The last register, the sixth register, records the attribute value  $\rho_i$ . By means of quantum *DistCalc*, the system state is

$$|\Psi_0\rangle = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \left( |i\rangle \otimes |j\rangle \otimes |d(x_i, x_j)\rangle \right) \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle^{\otimes \lceil \log N \rceil} + |G\rangle, \quad (4)$$

where  $|G\rangle$  is a garbage state.

- (ii). Set a desired threshold as  $d_{max}$  and set  $a_{ij} \in \{0, 1\}$  to indicate whether two data points  $x_i$  and  $x_j$  are close together. The value of  $a_{ij}$  is 1 if the distance of two data points

$d(x_i, x_j) \leq d_{max}$ , otherwise 0. Then we can easily store this value  $a_{ij}$  in the fourth register. The system state is

$$|\Psi_1\rangle = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \left( |i\rangle \otimes |j\rangle \otimes |d(x_i, x_j)\rangle \otimes |a_{ij}\rangle \right) \otimes |0\rangle \otimes |0\rangle^{\otimes \lceil \log N \rceil} + |G\rangle. \quad (5)$$

- (iii). Take a control-sum operation on the first, fourth, and sixth register. The first register  $|i\rangle$  is the control qubit, and the sixth register stores the sum of the values of  $a_{ij}$ , whereas the index  $i$  is fixed. Since the local density property of the data point  $x_i$  is  $\rho_i = \sum_{j=1}^N a_{ij}$ , the value stored in the sixth register is  $|\rho_i\rangle$ . Now the system is

$$|\Psi_2\rangle = \frac{1}{N} \sum_{i=1}^N \left[ |i\rangle \sum_{j=1}^N \left( |j\rangle \otimes |d(x_i, x_j)\rangle \otimes |a_{ij}\rangle \right) \otimes |0\rangle \otimes |\rho_i\rangle \right] + |G\rangle. \quad (6)$$

- (iv). Perform a control conditional rotation [45], where the first and the second registers are control qubits, and the fifth ancillary register is the target. Set the fifth register to  $|1\rangle$ , when  $\rho_j > \rho_i$ , otherwise set to  $|0\rangle$ . The whole system is divided into two parts, as shown below

$$|\Psi_3\rangle = \frac{1}{N} \sum_{i=1}^N \left[ |i\rangle \left( \sum_{j, \rho_j > \rho_i} |j\rangle |d(x_i, x_j)\rangle |a_{ij}\rangle |1\rangle + \sum_{j, \rho_j \leq \rho_i} |j\rangle |d(x_i, x_j)\rangle |a_{ij}\rangle |0\rangle \right) \otimes |\rho_i\rangle \right] + |G\rangle. \quad (7)$$

- (v). Apply a projection operation  $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$  on the fifth register, and keep the state when the measurement result is  $|1\rangle\langle 1|$ . The system is

$$|\Psi_4\rangle = \Gamma |\Psi_3\rangle = \alpha \sum_{i=1}^N \sum_{j, \rho_i < \rho_j} |i\rangle |j\rangle |d(x_i, x_j)\rangle |a_{ij}\rangle |1\rangle |\rho_i\rangle + |G\rangle, \quad (8)$$

where  $\alpha$  is the normalized parameter, and  $\sum |\alpha|^2 = 1$ . The third register and the last register store the attribute values  $\delta_i$  and  $\rho_i$  of each data point  $x_i$ , respectively.

- (vi). Perform a bit flip operation [46] on the third register, and the value is changed from  $d(x_i, x_j)$  to  $\overline{d(x_i, x_j)}$ . By doing this, the minimum value in the third register becomes the maximum value. In order to make the following Grover algorithm run under more convenient conditions, we change the target of the search to the maximum value of the two attributes. Data points that meet these two requirements are the center of clustering. Now the system is

$$|\Psi_5\rangle = \alpha \sum_{i=1}^N \sum_{j, \rho_i < \rho_j} |i\rangle |j\rangle |\overline{d(x_i, x_j)}\rangle |a_{ij}\rangle |1\rangle |\rho_i\rangle + |G\rangle. \quad (9)$$

- (vii). Apply the Grover algorithm [47] to find the index  $i$  of data point  $x_i$  with maximum  $\rho_i$  and the index  $j$  of the found data point  $x_i$  with maximum  $\delta_i$  with a full successful probability. The index  $i$  that meets the requirements is the center of a cluster.

#### 4. Simulation Results

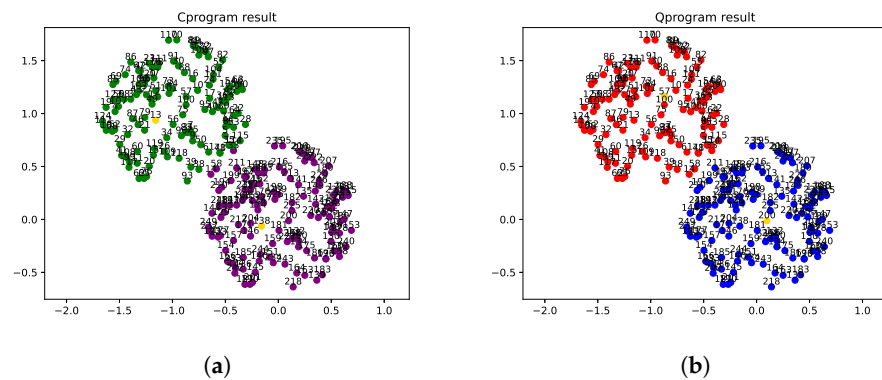
We clustered three differently distributed (horizontally, circularly, and discrete) data sets using our QDPC algorithm, implemented on Baidu’s quantum platform Paddle Quantum. Limited to the lack of QRAM devices, thread concurrency was used to read out all the data in a data set at one time. Data were generated by a random function with  $seed = 21$ , and the number of data  $N$  was fixed as 20, 40, 80, 250, 500, and 1000. Table 1 gives the common evaluation indicators purity, F-score and adjusted Rand index (ARI) of the two algorithms on the circularly distributed data. In the table, all the values lie between 0.95

and 1. When  $N = 20, 40, 80$ , the clustering results of the DPC algorithm are the same as those of the QDPC algorithm. For  $N = 20, 40, 80$ , the values of QDPC are greater than those of DPC, so the QDPC performs better than the DPC.

**Table 1.** Evaluation indicators of circularly distributed data.

Seed = 21	DPC			QDPC		
	Purity	F-Score	ARI	Purity	F-Score	ARI
$N = 20$	1.0	1.0	1.0	1.0	1.0	1.0
$N = 40$	1.0	1.0	1.0	1.0	1.0	1.0
$N = 80$	1.0	1.0	1.0	1.0	1.0	1.0
$N = 250$	0.988000	0.976104	0.952385	1.0	1.0	1.0
$N = 500$	0.992000	0.984066	0.968192	0.996000	0.992000	0.984032
$N = 1000$	0.990000	0.980164	0.960360	0.998000	0.996000	0.992008

We also depict the clustering performance of two algorithms when  $N$  is fixed at 250 in Figure 4. All the points are accompanied with their indexes. The points colored yellow are the centers of the clusters. Other points colored the same are clustering together, so both DPC and QDPC cluster the data into two groups. But DPC performs slightly worse than QDPC, since the points with indexes 14, 58 and 33 are colored with green, which should be purple.



**Figure 4.** Experimental results of two algorithms when the data are circularly distributed and the number  $N$  is fixed as 250. (a) Clustering performance of DPC algorithm; (b) clustering performance of QDPC algorithm.

The experiment was repeated 10 times and the average running times are recorded in Table 2. It can be seen that with increasing  $N$ , the running time of QDPC increases linearly, and that of the DPC increases exponentially. When  $N$  is fixed at 250, 500, or 1000, QDPC is faster than DPC, but when  $N$  is less than 80, DPC is faster than QDPC. The reason for this may be the fact that we simulated these results on a classical computer. If we ran the QDPC algorithm on a real quantum computer, the results may show an improvement.

**Table 2.** Comparison of the complexity of the two algorithms in simulation experiments.

Algorithms	$N = 20$	$N = 40$	$N = 80$	$N = 250$	$N = 500$	$N = 1000$
DPC(/s)	0.00500488	0.01701570	0.06606030	0.65659642	2.54531145	10.08315921
QDPC(/s)	0.02302074	0.03603339	0.06806207	0.19617867	0.39135551	0.77470422
Difference(/s)	-0.01801586	-0.0190176	-0.00200177	0.46041775	2.15395594	9.30845499

### 5. Discussion and Conclusions

We now analyze the complexity of the QDPC algorithm step by step. In (i), a quantum *DistCalc* circuit is applied to obtain the distance between two data points. The time complexity of this step depending on the distance definition is  $\log(N \cdot N)$  [44]. In (ii) and (iii), we convert the value  $d(x_i, x_j)$  into  $a_{ij}$  and add up  $a_{ij}$ . The time complexity of these two steps can be measured by the number of register accesses and the quantum addition circuit. Therefore, in general, the time complexity of this part is  $O(N + 5N)$  [48]. In (iv), (v), and (vi), we perform the conditional rotation operation, the projection operation, and the bit flip operation. The time complexity is relatively negligible compared with other steps. Finally, step (vii) requires the application of the Grover algorithm, which introduces a time complexity of  $O(\sqrt{N})$  [49,50]. Thus, the time complexity of the whole algorithm is  $O(\log(N^2) + 6N + \sqrt{N})$ . The space complexity is the space size of the quantum registers, i.e.,  $\lceil \log N \rceil + \lceil \log N \rceil + \lceil n + \log(n + d) \rceil + 1 + 1 + \lceil \log N \rceil = 3\lceil \log N \rceil + \lceil n + \log(n + d) \rceil + 2$ .

For the DPC algorithm, the most time-consuming step is to calculate the distance between data points. It can be seen that the total distances of  $\frac{1}{2}N(N - 1)$  times need to be calculated [51]. So the complexity of the classical DPC algorithm is  $O(N^2)$ . The space complexity of DPC depends on the space stored,  $\rho_i$  and  $\delta_i$ , for each point. The space required is  $N \cdot \lceil \log N \rceil + N \cdot \lceil n + \log(n + d) \rceil$  bits.

A corresponding comparison between classical and quantum algorithms is shown in Table 3. Based on Table 3, the QDPC algorithm costs less than the DPC algorithm in terms of both time and space complexities.

**Table 3.** Theoretical comparison of the complexity of the two algorithms.

Complexity	DPC	QDPC
time	$O(N^2)$	$O(\log(N^2) + 6N + \sqrt{N})$
space	$N \cdot \lceil \log N \rceil + N \cdot \lceil n + \log(n + d) \rceil$	$3\lceil \log N \rceil + \lceil n + \log(n + d) \rceil + 2$

In this paper, we have proposed a QDPC algorithm that is more efficient in both time and space than the classical algorithm. We applied it to two key circuits, a quantum *DistCalc* circuit and a Grover circuit. The quantum *DistCalc* circuit calculates the distance between data points in the data set, from which two important attribute values,  $\rho_i$  and  $\delta_i$ , required by the QDPC algorithm are obtained. Then, the Grover algorithm is used to search the index of clustering center points that meet the conditions from the data set. In the future, we will investigate some possible application scenarios of the QDPC algorithm and compare the efficiency of algorithms on different data set structures.

**Author Contributions:** Z.W.: Conceptualization, methodology, software, data curation, writing—original draft. T.S.: supervision, project administration, writing—review and editing. Y.Z.: methodology, investigation. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Guangxi Key Laboratory of Cryptography and Information Security (No. GCIS201922), and Fundamental Research Funds for the Central Universities (No. 21620433).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors would like to thank the reviewers for their valuable comments and suggestions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Aïmeur, E.; Brassard, G.; Gambs, S. Quantum clustering algorithms. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007; pp. 1–8.
2. Nielsen, M.A.; Chuang, I. Quantum computation and quantum information. *Am. J. Phys.* **2002**, *70*, 558. [[CrossRef](#)]
3. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum Algorithm for Linear Systems of Equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [[CrossRef](#)] [[PubMed](#)]
4. Yu, C.H.; Gao, F.; Wang, Q.L.; Wen, Q.Y. Quantum algorithm for association rules mining. *Phys. Rev. A* **2016**, *94*, 042311. [[CrossRef](#)]
5. Kerenidis, I.; Prakash, A. Quantum recommendation systems. *arXiv* **2016**, arXiv:1603.08675.
6. Greche, L.; Jazouli, M.; Es-Sbai, N.; Majda, A.; Zarghili, A. Comparison between Euclidean and Manhattan distance measure for facial expressions classification. In Proceedings of the 2017 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS), Fez, Morocco, 19–20 April 2017; pp. 1–4.
7. Wiebe, N.; Kapoor, A.; Svore, K.M. Quantum nearest-neighbor algorithms for machine learning. *Quantum Inf. Comput.* **2018**, *15*, 318–358.
8. Zidan, M.; Abdel-Aty, A.H.; El-shafei, M.; Feraig, M.; Al-Sbou, Y.; Eleuch, H.; Abdel-Aty, M. Quantum classification algorithm based on competitive learning neural network and entanglement measure. *Appl. Sci.* **2019**, *9*, 1277. [[CrossRef](#)]
9. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv* **2013**, arXiv:1307.0411.
10. Kerenidis, I.; Landman, J.; Luongo, A.; Prakash, A. q-means: A quantum algorithm for unsupervised machine learning. *arXiv* **2018**, arXiv:1812.03584.
11. Shaikh, T.A.; Ali, R. Quantum computing in big data analytics: A survey. In Proceedings of the 2016 IEEE International Conference on Computer and Information Technology (CIT), Nadi, Fiji, 8–10 December 2016; pp. 112–115.
12. Yang, N. KNN Algorithm Simulation Based on Quantum Information. In Proceedings of the Student-Faculty Research Day Conference, CSIS, New York, NY, USA, 3 May 2019.
13. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [[CrossRef](#)]
14. Nakahara, M. *Quantum Computing: From Linear Algebra to Physical Realizations*; CRC Press: Boca Raton, FL, USA, 2008.
15. Song, T.; Li, P.; Weng, J. Concise security bounds for sending-or-not-sending twin-field quantum key distribution with finite pulses. *Phys. Rev. A* **2021**, *103*, 042408. [[CrossRef](#)]
16. Song, T.; Tan, X.; Weng, J. Statistical fluctuation analysis of measurement-device-independent quantum random-number generation. *Phys. Rev. A* **2019**, *99*, 022333. [[CrossRef](#)]
17. Jiang, C.; Yu, Z.W.; Hu, X.L.; Wang, X.B. Unconditional security of sending or not sending twin-field quantum key distribution with finite pulses. *Phys. Rev. Appl.* **2019**, *12*, 024061. [[CrossRef](#)]
18. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **2014**, *10*, 631–633. [[CrossRef](#)]
19. Kopczyk, D. Quantum machine learning for data scientists. *arXiv* **2018**, arXiv:1804.10068.
20. Hussain Shah, S.; Javed Iqbal, M.; Bakhsh, M.; Iqbal, A. Analysis of Different Clustering Algorithms for Accurate Knowledge Extraction from Popular DataSets. *Inf. Sci. Lett.* **2020**, *9*, 4.
21. Lakshmi, S.A.; Mary, S.A.S.A. Group Mosquito Host Seeking Algorithm Based Self Organizing Technique for Genetic Algorithm. *Appl. Math. Inf. Sci.* **2019**, *13*, 231–238. [[CrossRef](#)]
22. Mustafa, W. Shrink: An Efficient Construction Algorithm for Minimum Vertex Cover Problem. *Inf. Sci. Lett.* **2021**, *10*, 9.
23. Arora, P.; Varshney, S. Analysis of k-means and k-medoids algorithm for big data. *Procedia Comput. Sci.* **2016**, *78*, 507–512. [[CrossRef](#)]
24. Whelan, C.; Harrell, G.; Wang, J. Understanding the k-medians problem. In *Proceedings of the International Conference on Scientific Computing (CSC)*; The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp): Athens, Greece, 2015; p. 219.
25. Likas, A.; Vlassis, N.; Verbeek, J.J. The global k-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451–461. [[CrossRef](#)]
26. Montanaro, A. Quantum algorithms: An overview. *NPJ Quantum Inf.* **2016**, *2*, 1–8. [[CrossRef](#)]
27. Comaniciu, D.; Meer, P. Mean shift analysis and applications. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Corfu, Greece, 20–25 September 1999; Volume 2, pp. 1197–1203.
28. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd* **1996**, *96*, 226–231.
29. He, X.; Cai, D.; Shao, Y.; Bao, H.; Han, J. Laplacian regularized gaussian mixture model for data clustering. *IEEE Trans. Knowl. Data Eng.* **2010**, *23*, 1406–1418. [[CrossRef](#)]
30. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [[CrossRef](#)] [[PubMed](#)]
31. Yu, S.S.; Chu, S.W.; Wang, C.M.; Chan, Y.K.; Chang, T.C. Two improved k-means algorithms. *Appl. Soft Comput.* **2018**, *68*, 747–755. [[CrossRef](#)]
32. LIU, X.j.; YUAN, J.b.; XU, J.; DUAN, B.j. Quantum k-means algorithm. *J. Jilin Univ.* **2018**, *2*, 539–544.
33. Zhang, G.; Zhang, C.; Zhang, H. Improved K-means algorithm based on density Canopy. *Knowl.-Based Syst.* **2018**, *145*, 289–297. [[CrossRef](#)]



34. Pandit, S.; Gupta, S. A comparative study on distance measuring approaches for clustering. *Int. J. Res. Comput. Sci.* **2011**, *2*, 29–31. [[CrossRef](#)]
35. Gultom, S.; Sriadhi, S.; Martiano, M.; Simarmata, J. Comparison analysis of K-means and K-medoid with Ecludience distance algorithm, Chanberra distance, and Chebyshev distance for big data clustering. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *420*, 012092. [[CrossRef](#)]
36. Brassard, G.; Hoyer, P.; Mosca, M.; Tapp, A. Quantum amplitude amplification and estimation. *Contemp. Math.* **2002**, *305*, 53–74.
37. Giovannetti, V.; Lloyd, S.; Maccone, L. Architectures for a quantum random access memory. *Phys. Rev. A* **2008**, *78*, 52310. [[CrossRef](#)]
38. Zidan, M.; Aldulaimi, S.; Eleuch, H. Analysis of the Quantum Algorithm based on Entanglement Measure for Classifying Boolean Multivariate Function into Novel Hidden Classes: Revisited. *Appl. Math* **2021**, *15*, 643–647.
39. Kerenidis, I.; Landman, J. Quantum spectral clustering. *arXiv* **2020**, arXiv:2007.00280.
40. Kapil, S.; Chawla, M. Performance evaluation of K-means clustering algorithm with various distance metrics. In Proceedings of the 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, 4–6 July 2016; pp. 1–4.
41. Sahu, L.; Mohan, B.R. An improved K-means algorithm using modified cosine distance measure for document clustering using Mahout with Hadoop. In Proceedings of the 2014 9th International Conference on Industrial and Information Systems (ICIIS), Gwalior, India, 15–17 December 2014; pp. 1–5.
42. Lee Rodgers, J.; Nicewander, W.A. Thirteen ways to look at the correlation coefficient. *Am. Stat.* **1988**, *42*, 59–66. [[CrossRef](#)]
43. Aïmeur, E.; Brassard, G.; Gambs, S. Machine learning in a quantum world. In *Conference of the Canadian Society for Computational Studies of Intelligence*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 431–442.
44. Aïmeur, E.; Brassard, G.; Gambs, S. Quantum speed-up for unsupervised learning. *Mach. Learn.* **2013**, *90*, 261–287. [[CrossRef](#)]
45. Kaye, P. Reversible addition circuit using one ancillary bit with application to quantum computing. *arXiv* **2004**, arXiv:quant-ph/0408173.
46. Buhrman, H.; Cleve, R.; Watrous, J.; De Wolf, R. Quantum fingerprinting. *Phys. Rev. Lett.* **2001**, *87*, 167902. [[CrossRef](#)]
47. Durr, C.; Hoyer, P. A quantum algorithm for finding the minimum. *arXiv* **1996**, arXiv:quant-ph/9607014.
48. Ruan, Y.; Xue, X.; Liu, H.; Tan, J.; Li, X. Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance. *Int. J. Theor. Phys.* **2017**, *56*, 3496–3507. [[CrossRef](#)]
49. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996.
50. Boyer, M.; Brassard, G.; Høyer, P.; Tapp, A. Tight bounds on quantum searching. *Fortschritte Phys. Prog. Phys.* **1998**, *46*, 493–505. [[CrossRef](#)]
51. Singh, A.; Yadav, A.; Rana, A. K-means with Three different Distance Metrics. *Int. J. Comput. Appl.* **2013**, *67*, 13–17. [[CrossRef](#)]