scientific reports

OPEN

Quantum algorithm for MMNG-based DBSCAN

Xuming Xie[1,3], Longzhen Duan[1,3], Taorong Qiu[1]✉ & Junru Li[2]

DBSCAN is a famous density-based clustering algorithm that can discover clusters with arbitrary shapes without the minimal requirements of domain knowledge to determine the input parameters. However, DBSCAN is not suitable for databases with different local-density clusters and is also a very time-consuming clustering algorithm. In this paper, we present a quantum mutual *MinPts*-nearest neighbor graph (MMNG)-based DBSCAN algorithm. The proposed algorithm performs better on databases with different local-density clusters. Furthermore, the proposed algorithm has a dramatic increase in speed compared to its classic counterpart.

Clustering, an important branch of unsupervised machine learning, is the process of partitioning a dataset into subsets of points called clusters, such that similar points are grouped in the same cluster and dissimilar points are put in different clusters. This procedure is widely used in many scientific fields, including bioinformatics[1–4], image processing[5–8], and social networks[9,10].

DBSCAN[11], a density-based clustering algorithm, is one of the most famous clustering algorithms. The distinguishing advantage of the DBSCAN algorithm is that it can be used to discover arbitrarily shaped clusters. Furthermore, it does not need the minimal requirements of domain knowledge to determine the input parameters, and can also exclude outliers from the clusters. However, DBSCAN has two dire drawbacks. First, DBSCAN has a low efficiency on databases with different local-density clusters; second, the algorithm is very time-consuming.

Quantum computing has attracted tremendous attention due to its parallel capability. In 1982, Feynman pointed out that quantum computers might achieve significant increase in speed over classical computers on certain specific problems[12]. Shor's algorithm[13] and Grover's algorithm[14] are two of the most popular quantum algorithms. Shor's algorithm has an exponential increase in speed, and Grover's algorithm has a quadratic increase in speed over their classical counterparts. With the rise of quantum computing, many researchers have also designed various quantum machine learning algorithms and quantum data mining algorithms, such as quantum linear regression[15–17], quantum support vector machine[18,19], quantum k-nearest neighbors classification[20], quantum deep learning[21], and quantum association rules mining[22]. Recently, tremendous advances have been made in constructing quantum computers. Krantz et al.[23,24] introduced the central concepts and challenges of superconducting quantum circuits. Huang et al.[25] provided experimental efforts toward large-scale superconducting quantum computers. Bruzewicz et al.[26] concluded the basics of trapped-ion quantum computing and explored the outlook for trapped-ion quantum computing.

Inspired by quantum computing, we propose a quantum mutual *MinPts*-nearest neighbor graph (MMNG)-based DBSCAN algorithm. First, we design a quantum mutual *MinPts*-nearest neighbor graph algorithm that is devoted to dividing a database into subsets. After that, we quantize the original DBSCAN algorithm to cluster each subset.

Preliminaries

In this section, we provide the necessary background knowledge for this paper. First, we briefly introduce the basic definitions of the classical DBSCAN algorithm. Then, we review the fundamental concepts of Grover's algorithm.

DBSCAN.

The DBSCAN algorithm offers a new notion of "cluster" and "noise" in a database D of N points of some k-dimensional space S. The whole set of definitions is given as follows.

Definition 1 (*Eps-neighborhood of a point*) The *Eps*-neighborhood of a point p, denoted by $N_{Eps}(p)$, is defined by $N_{Eps}(p) = \{q \in D | Dist(p, q) \leq Eps\}$.

[1]School of Information Engineering, Nanchang University, Nanchang 330031, People's Republic of China. [2]School of Software and Internet of Things Engineering, Jiangxi University of Finance and Economics, Nanchang 330013, People's Republic of China. [3]These authors contributed equally: Xuming Xie and Longzhen Duan. ✉email: qiutaorong@ncu.edu.cn

The *Eps*-neighborhood, the fundamental definition of the algorithm, can be used to distinguish core points and noncore points. *Eps* is the distance threshold. Core points are the points inside of any cluster, and noncore points are the points on the border of any cluster or the points belonging to none of the clusters. Let p be a point in a database D, where $|N_{Eps}(p)|$ denotes the number of points within the *Eps*-neighborhood of p. Let *MinPts* be the threshold of the number of points; if $|N_{Eps}(p)| \geq MinPts$, then p is a core point; otherwise, p is a noncore point.

Definition 2 (*directly density-reachable*) A point p is *directly density-reachable* from a point q if

1. $p \in N_{Eps}(q)$ and
2. $|N_{Eps}(p)| \geq MinPts$

Directly density-reachable is not always symmetric. When p and q are both core points, the direct density reachability is symmetric; when one is a core point and the other is a border point, the direct density reachability is asymmetric.

Definition 3 (*density-reachable*) A point p is density-reachable from a point q if there is a chain of points $p_1, p_2, \ldots, p_N \in D$, $p_1 = q$, $p_N = p$ such that p_{i+1} is directly density-reachable from p_i.

Definition 4 (*density-connected*) A point p is density-connected to a point q if there is a point o such that p and q are density-reachable from o.

Definition 5 (*cluster*) Let D be a database of points. A cluster C is a nonempty subset of D satisfying the following conditions:

1. $\forall p, q$: if $p \in C$ and q is density-reachable from p then $q \in C$.
2. $\forall p, q \in C$: p is density-connected to q.

In the database D, not all the points belong to clusters. The points that do not belong to any cluster are defined as "noise" in the DBSCAN algorithm.

Definition 6 (*noise*) Let $C_1, \ldots, C_k$ be the clusters of the database D, $i = 1, \ldots, k$. Then, we define the noise as the set of points in the database D not belonging to any cluster C_i, i.e., noise$=\{p \in D | \forall i : p \notin C_i\}$.

Grover's algorithm.

Let us assume that we wish to search $M(1 \leq M \leq N)$ solutions from an unstructured search space of N elements. Rather than examining N elements one by one, Grover's algorithm checks the elements in parallel by assigning indexes to all of the elements and storing the indexes in a quantum register. With a series of unitary operations augmenting the success probability gradually, Grover's algorithm can obtain the indexes of the target elements with a high probability.

The proposed algorithm

In this section, we design a quantum mutual *MinPts*-nearest neighbor graph algorithm and a quantum DBSCAN algorithm and present a quantum MMNG-based DBSCAN algorithm.

Quantum MMNG algorithm.

Let D be a database, p and q be some objects in D, and *MinPts* be a positive integer. The relative concept will be introduced as follow.

Definition 7 (*Mutual MinPts-nearest neighbor (MMN)*): If p is in the *MinPts*-nearest neighborhood of q and q is in the *MinPts*-nearest neighborhood of p, then we call p a mutual *MinPts*-nearest neighbor of q; similarly, q is a mutual *MinPts*-nearest neighbor of p.

Definition 8 (*Mutual MinPts-nearest neighbor graph (MMNG)*): The mutual *MinPts*-nearest neighbor graph can be constructed by connecting each point to its mutual *MinPts*-nearest neighbors.

Note that MMNG is an algorithm with high complexity. To speed up the MMNG, we intend to quantize the MMNG algorithm. Dürr et al.[27] developed a *quant_find_smallest_values* algorithm for finding the c closest neighbors of a point with high probability within $O\sqrt{cn}$ time. Based on the *quant_find_smallest_values*, we propose a quantized MMNG algorithm, as shown in algorithm 1.

Algorithm 1 *quant_MMNG* (D , $MinPts$)

```
For each point do
Obtain the $MinPts$ -nearest neighborhood using quant_find_smallest_values
End for
Construct the mutual $MinPts$ -nearest neighbor graph;
marked[ $\forall x \in D$ ] = 0;
number = 1
For all $x \in D$ do
  If marked( $x$ )=0 then
     marked( $x$ )=1
 $Subset_{number} = Subset_{number} \cup MMN(x)$
    For all $y \in Subset_{number}$ do
        If marked( $y$ )=0 then
        Marked( $y$ )=1
 $Subset_{number} = Subset_{number} \cup MMN(y)$
        End if
    End for
    Number=number+1
  End if
End for
Return Subset
```

In this paper, algorithm 1 is used to obtain the subsets of database D. After obtaining the subsets, we apply the quantum DBSCAN algorithm on each subset to obtain the eventual clusters and the noise set.

Quantum DBSCAN algorithm. We consider a database $D_N = \{p_1, \ldots, p_N\}$, which is composed of n points, and each point p_i has k attributes. For each point p_i in $D_N = \{p_1, \ldots, p_N\}$, it is necessary to calculate $Dist(p_i, p_j)$ $n - 1$ times to determine the *Eps*-neighborhood of p_i. Determining the *Eps*-neighborhood is fairly time-consuming. To solve this problem, we intend to screen the points in the *Eps*-neighborhood of p_i with quantum search.

In our model, a quantum distance black box is proposed. The proposed black box can accept two types of inputs, as illustrated in Fig. 1. $|i\rangle$ is a one-state input and the index of point p_i; $|j\rangle$ is a superposition of inputs and includes the indexes of all the points. Evidently, this is feasible because one q-bit can be a pure state or a superposition of states. Furthermore, one query to this black box means asking for distances between the point p_i and all the points p_j s (when $i = j$, $Dist(p_i, p_j) = 0$). After obtaining the distances, the black box compares them with the *Eps* distance. Then, a selection function $f(i, j)$ assigns a value of 1 when $Dist(p_i, p_j)$ is smaller than or equal to the *Eps* distance and a value of 0 otherwise. The selection function is shown in Eq. (1).

$$f(i,j) = \begin{cases} 0, \text{ if } Dist(p_i, p_j) > Eps \\ 1, \text{ if } Dist(p_i, p_j) \le Eps \end{cases} \tag{1}$$

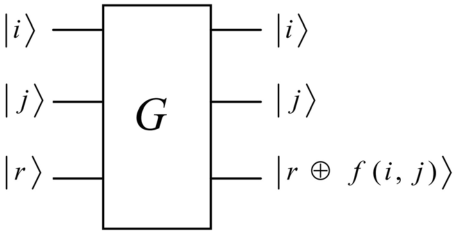
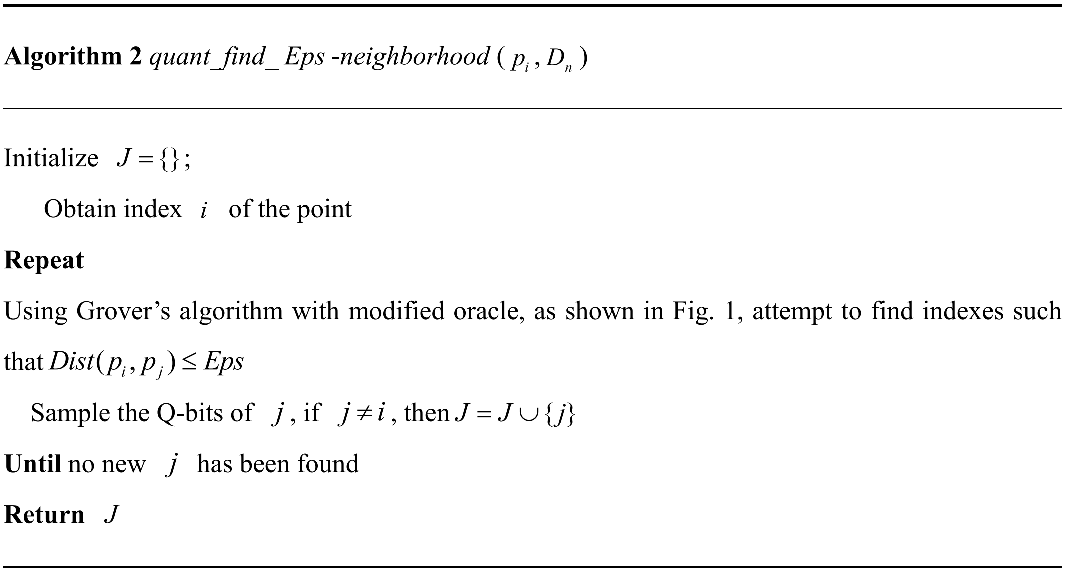


Figure 1. The oracle to compute the distance between p_i and all the possible p_js. $|r\rangle$ is an ancillary register.

Meanwhile, a flipping operation is carried out in the black box. As depicted in Eq. (2), if $f(i,j) = 1$, the ancillary register $|r\rangle$ is flipped; if $f(i,j) = 0$, the ancillary register $|r\rangle$ remained unaltered. The symbol $\oplus$ denotes module 2, also known as an exclusive-or.

$$|i\rangle|j\rangle|r\rangle \rightarrow |i\rangle|j\rangle|r \oplus f(i,j)\rangle \tag{2}$$

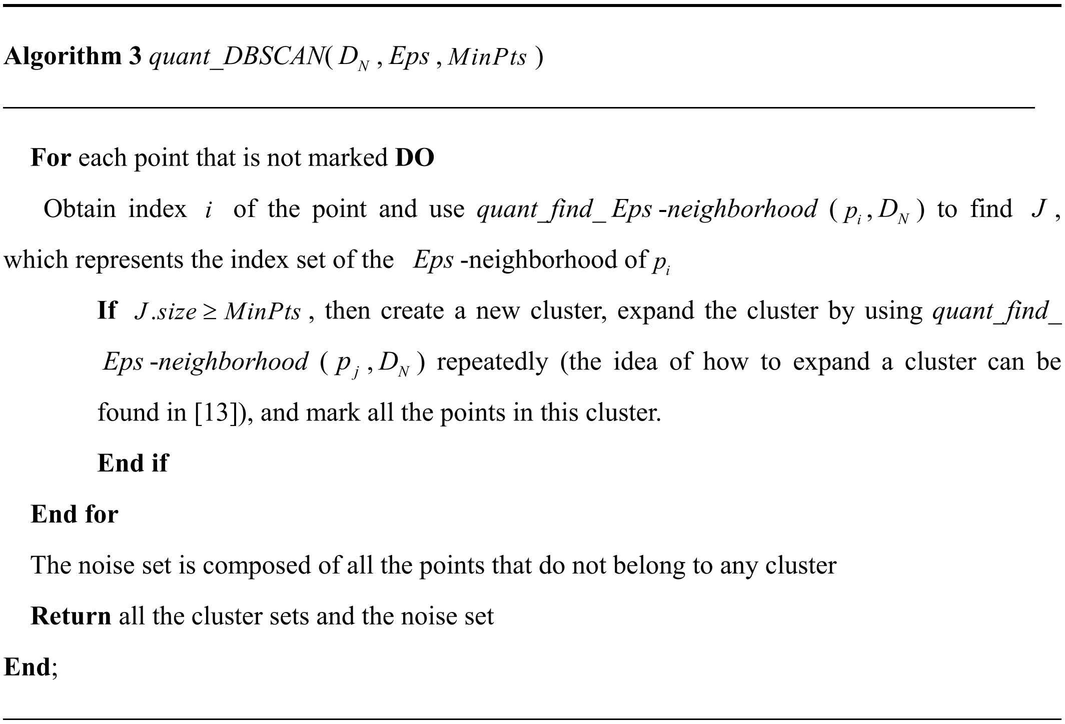
Based on the aforementioned black box, we designed algorithm 2 (*quant_find_Eps-neighborhood* as described below) as a subroutine of the quantum-based DBSCAN algorithm. Given a specific point p_i, algorithm 2 is able to fix its *Eps*-neighborhood.



```
Algorithm 2 quant_find_ Eps -neighborhood ( $p_i$ , $D_n$ )

Initialize $J = \{\}$;
   Obtain index $i$ of the point
Repeat
Using Grover's algorithm with modified oracle, as shown in Fig. 1, attempt to find indexes such that $Dist(p_i, p_j) \le Eps$
  Sample the Q-bits of $j$, if $j \neq i$, then $J = J \cup \{j\}$
Until no new $j$ has been found
Return $J$
```

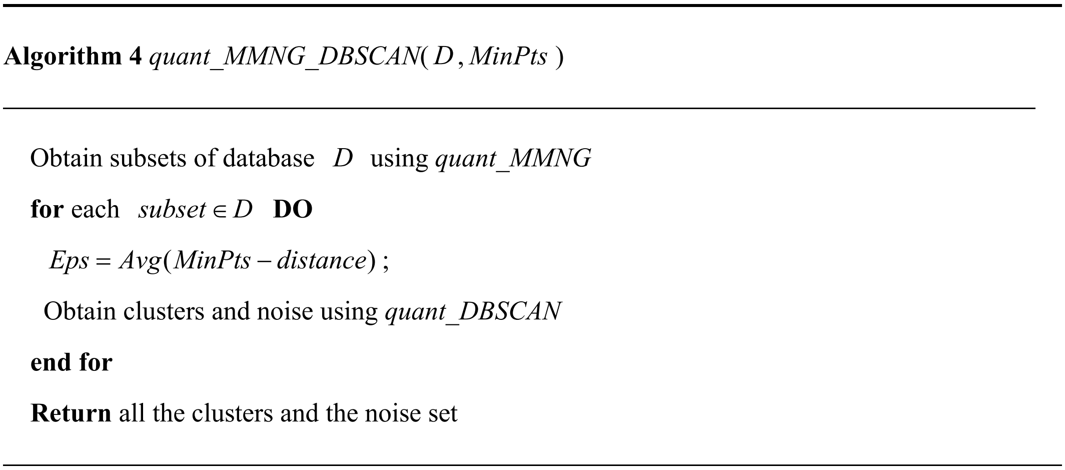
Once the *Eps*-neighborhoods are fixable quantum-mechanically, core points and noncore points become discernable according to the basic notions of the classic DBSCAN. If a point is a noncore point, we keep looking for a core point because there is no need to create a new cluster for a noncore point; however, if a point is a core point, we set up a new cluster and expand it. With the expanding methodology offered in the original DBSCAN algorithm, the quantum-based DBSCAN algorithm *quant_DBSCAN*(D_N, *Eps*, *MinPts*) is presented hereafter, as shown in algorithm 3.



Algorithm 3 *quant_DBSCAN*(D_N , Eps , $MinPts$)

```
For each point that is not marked DO
  Obtain index $i$ of the point and use quant_find_ Eps -neighborhood ( $p_i, D_N$ ) to find $J$ , which represents the index set of the $Eps$ -neighborhood of $p_i$
      If $J.size \geq MinPts$ , then create a new cluster, expand the cluster by using quant_find_ Eps -neighborhood ( $p_j, D_N$ ) repeatedly (the idea of how to expand a cluster can be found in [13]), and mark all the points in this cluster.
      End if
  End for
  The noise set is composed of all the points that do not belong to any cluster
  Return all the cluster sets and the noise set
End;
```

Quantum algorithm for MMNG-based DBSCAN. The proposed algorithm divides the database into subsets first and then applies the quantum DBSCAN algorithm to each subset. Note that different subsets have a different *Eps* in our algorithm. For a specific subset, we select the average *MinPts* distance as the *Eps* of the subset.



Algorithm 4 *quant_MMNG_DBSCAN*(D , $MinPts$)

```
  Obtain subsets of database $D$ using quant_MMNG
  for each $subset \in D$ DO
    $Eps = Avg(MinPts - distance)$ ;
    Obtain clusters and noise using quant_DBSCAN
  end for
  Return all the clusters and the noise set
```

The algorithm analysis

In this section, we briefly analyze the complexity of our algorithm first and then present the success probability of our algorithm.

The complexity. Dürr et al.[27] proved that the complexity of *quant_find_smallest_values* is $O\sqrt{cn}$. It is easy to see that the complexity of algorithm 1 is $O(N\sqrt{MinPts * n})$.

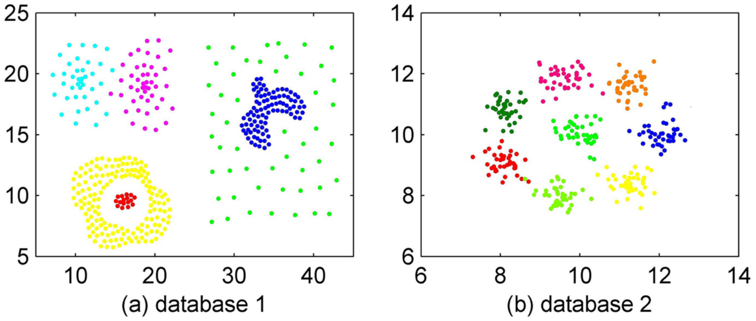


Figure 2. Sample databases.

For algorithm 2, there are $|N_{Eps}(p)|+1$ targets for each point. According to the original version of Grover's algorithm, algorithm 2 needs to interrogate the oracle approximately $\sqrt{\frac{N}{|N_{Eps}(p)|+1}}$ times. It can easily be perceived that the smaller $|N_{Eps}(p)|$ is, the more queries are needed. In the worst case scenario, when $|N_{Eps}(p)|=0$, the queries of the oracle are approximately $\sqrt{N}$ times. In other words, the complexity of algorithm 2 is $O(\sqrt{N})$.

For algorithm 3, we need to calculate the *Eps*-neighborhood of every point. This means that algorithm 3 needs to call algorithm 2 N times. Thus, we can ensure that the complexity of algorithm 3 is smaller than $O(N\sqrt{N})$, even though the $|N_{Eps}(p)|$s are different for different points.

In other words, the complexity of our proposed algorithm is approximately $O(N\sqrt{MinPts * n})$.

The success probability.

Dürr et al.[27] proved that *quant_find_smallest_values* is able to obtain c nearest neighbors with a high probability. It is easy to infer that algorithm 1 can obtain subsets with a high probability.

It is noteworthy that $|N_{Eps}(p)|$s are different from point to point, which means that there are different numbers of targets when algorithm 2 is dealing with different points. As a result, the success probabilities are different when calculating different *Eps*-neighborhoods. By referencing the former work, the success probability of algorithm 2 can be calculated via Eq. (3) after T iterations.

$$P=\sin^2((2T+1)\arcsin\sqrt{\frac{|N_{Eps}(p)|+1}{N}}) \tag{3}$$

We already know that $T\approx\sqrt{\frac{N}{|N_{Eps}(p)|+1}}$. Usually, *MinPts* is far less than N, and $|N_{Eps}(p)|$ are numbers close to *MinPts*. As a result, it can be inferred that the success probability of algorithm 3 is high.

Our proposed method is a combination of algorithm 1 and algorithm 3, thus success probability of the proposed method is high.

Performance evaluation

To show the effectiveness of the proposed algorithm, performance evaluation based on two databases is conducted. To compare our algorithm with the classic DBSCAN method and the NaNG method, we use the two synthetic sample databases depicted in Fig. 2.

The experimental results on database 1 are shown in Fig. 3. A total of 399 objects are included in database 1. In the figure, the black squares represent the points that are detected as outliers. The experimental result of DBSCAN on database 1 is undesirable, and the accuracy is approximately 74.6%. The experimental result of NaNG is better than that of DBSCAN, with an accuracy of 90.73%. Our proposed method has the best performance on database 1, with an accuracy of approximately 95.74%.

The experimental results on database 2 are shown in Fig. 4. As shown in Fig. 4a, database 2 includes 320 objects and 8 clusters. As shown in Fig. 4b, the result of DBSCAN is tolerable with an accuracy of 92.5%. From the result shown in Fig. 4c, we can see that NaNG mistakenly combines two clusters into one. The accuracy of NaNG is 87.5%. As shown in Fig. 4d, the performance of the proposed method is the same as DBSCAN with an accuracy of 92.5%.

Conclusion

Inspired by the mutual neighbor method and quantum computing, in this work, we present a quantum MMNG-based DBSCAN. Compared to the original DBSCAN, the proposed method performs better on databases with different local-density clusters. Furthermore, the proposed method is dramatically faster than its classical counterpart.

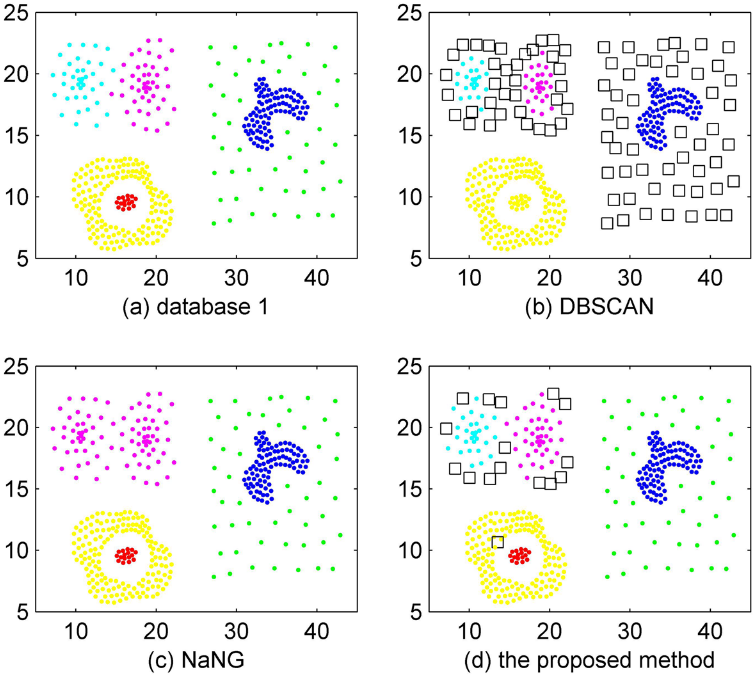


Figure 3. Clustering results of database 1.

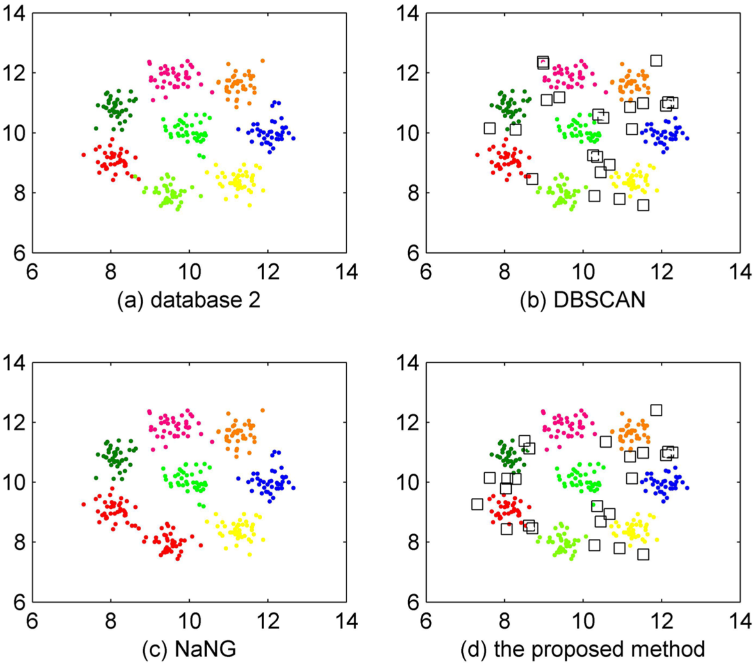


Figure 4. Clustering results of database 2.

Received: 3 March 2021; Accepted: 30 June 2021
Published online: 30 July 2021

References

1. Li, W. & Godzik, A. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**, 1658–1659 (2006).
2. King, A. D., Przlj, N. & Jurisica, I. Protein complex prediction via cost-based clustering. *Bioinformatics* **20**, 3013–3020 (2004).
3. Huang, D., Sherman, B. T. & Lempicki, R. A. Systematic and integrative analysis of large gene lists using david bioinformatics resources. *Nat. Protoc.* **4**, 44–57 (2009).

4. Rossant, C., Kadir, S. N. & Goodman, D. F. M. Spike sorting for large, dense electrode arrays. *Nat. Neurosci.* **19**, 634–641 (2016).
5. Moosmann, F., Nowak, E. & Jurie, F. Randomized clustering forests for image classification. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**, 1632–1646 (2008).
6. Ducournau, A. *et al.* A reductive approach to hypergraph clustering: An application to image segmentation. *Pattern Recognit.* **45**, 2788–2803 (2012).
7. Chaira, T. A novel intuitionistic fuzzy c means clustering algorithm and its application to medical images. *Appl. Soft Comput.* **11**, 1711–1717 (2011).
8. Zheng, M. *et al.* Graph regularized sparse coding for image representation. *IEEE Trans. Image Process.* **20**, 1327–1336 (2011).
9. Chakrabarti, S. Data mining for hypertext: A tutorial survey. *ACM SIGKDD Explor. Newsl.* **1**, 1–11 (2000).
10. Chang, M. *et al.* Exact algorithms for problems related to the densest k-set problem. *Inf. Process. Lett.* **114**, 510–513 (2014).
11. Martin, E., Hans-Peter, K., Jorg, S., *et al.* A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovering in Databases and Data Mining*, 122–128 (1996).
12. Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488 (1982).
13. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**, 1484–1509 (1997).
14. Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, 212–219 (1996).
15. Wiebe, N., Braun, D. & Lloyd, S. Quantum algorithm for data fitting. *Phys. Rev. Lett* **109**, 050505 (2012).
16. Schuld, M., Sinayskiy, I. & Petruccione, F. Prediction by linear regression on a quantum computer. *Phys. Rev. A* **94**, 022342 (2016).
17. Wang, G. Quantum algorithm for linear regression. *Phys. Rev. A* **96**, 012335 (2017).
18. Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett* **113**, 130503 (2014).
19. Li, Z., Liu, X. & Xu, N. Experimental realization of a quantum support vector machine. *Phys. Rev. Lett* **114**, 140504 (2015).
20. 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.* **56**, 3496–3507 (2017).
21. Cheng, S., Chen, J. & Wang, L. Quantum entanglement: from quantum states of matter to deep learning. *Physics* **46**, 416–423 (2017).
22. Yu, C. H., Gao, F., Wang, Q. L. & Wen, Q. Y. Quantum algorithm for association rules mining. *Phys. Rev. A* **94**, 042311 (2016).
23. Krantz, P. *et al.* A quantum engineer's guide to superconducting qubits. *Appl. Phys. Rev.* **6**, 021318 (2019).
24. Kjaergaard, M. *et al.* Superconducting qubits: current state of play. *Annu. Rev. Condens. Matter Phys.* **11**, 369–395 (2019).
25. Huang, H. L. *et al.* Superconducting quantum computing: a review. *Sci. China Inf. Sci.* **63**, 1–32 (2020).
26. Bruzewicz, C. D. *et al.* Trapped-ion quantum computing: Progress and challenges. *Appl. Phys. Rev.* **6**, 021314 (2019).
27. Dürr, C., Heiligman, M., Høyer, P., & Mhalla, M. Quantum query complexity of some graph problems. In *Proceedings of the 31st International Conference on Automata, Languages and Programming*, 481–493 (2004)

Author contributions

X.X. and L.D. wrote the main manuscript text. T.Q. and J.L. analysed the performance of the algorithm. All authors reviewed the manuscript.

Funding

Funding was provided by National Natural Science Foundation of China(62066027).

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to T.Q.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.