SCIENCE CHINA
Information Sciences

October 2022, Vol. 65 200504:1–200504:10
https://doi.org/10.1007/s11432-022-3492-x

• RESEARCH PAPER •
Special Focus on Quantum Information

Quantum spectral clustering algorithm for unsupervised learning

Qingyu LI, Yuhan HUANG, Shan JIN, Xiaokai HOU & Xiaoting WANG*

Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu 610051*, China*

Received 5 March 2022/Revised 12 April 2022/Accepted 26 April 2022/Published online 15 July 2022

Abstract Clustering is one of the most crucial problems in unsupervised learning, and the well-known k-means algorithm can be implemented on a quantum computer with a significant speedup. However, for the clustering problems that cannot be solved using the k-means algorithm, a powerful method called spectral clustering is used. In this study, we propose a circuit design to implement spectral clustering on a quantum processor with substantial speedup by initializing the processor into a maximally entangled state and encoding the data information into an efficiently simulatable Hamiltonian. Compared to the established quantum k-means algorithms, our method does not require a quantum random access memory or a quantum adiabatic process. It relies on an appropriate embedding of quantum phase estimation into Grover's search to gain the quantum speedup. Simulations demonstrate that our method effectively solves clustering problems and is an important supplement to quantum k-means algorithm for unsupervised learning.

Keywords quantum algorithm, machine learning, spectral clustering, quantum phase estimation, Grover's search, Hamiltonian simulation

Citation Li Q Y, Huang Y H, Jin S, et al. Quantum spectral clustering algorithm for unsupervised learning. Sci China Inf Sci, 2022, 65(10): 200504, https://doi.org/10.1007/s11432-022-3492-x

1 Introduction

Quantum machine learning (QML) is an interdisciplinary subject connecting quantum computing and machine learning (ML), focusing on solving ML problems in quantum processors and obtaining potential quantum speedup or other advantages [1]. Although the exploration of QML is still preliminary, existing results have shown that multiple QML algorithms substantially reduce the computational complexity compared to their classical counterparts. Examples of such advantages include quantum data-fitting [2], quantum support vector machine (QSVM) [3, 4], quantum principal component analysis [5], quantum Boltzmann machine [6], and quantum reinforcement learning [7]. Besides supervised and reinforcement learning, another important subfield in ML is unsupervised learning, wherein algorithms are designed to find hidden patterns in a set of unlabeled data. Typical unsupervised learning problems include anomaly detection [8], dimensionality reduction [9], and clustering [10]. Clustering aims to group a set of data points into different subgroups based on their similarities, and one of the most popular clustering algorithms is the k-means algorithm. The k-means algorithm can be implemented on a quantum computer by either converting it into a search problem [11] or utilizing a quantum random access memory (QRAM) [12] and adiabatic quantum computing [13,14]. Depending on the specific design, the quantum k-means algorithm can achieve a quadratic or exponential speedup [11, 14]. Nevertheless, not all clustering problems can be solved by k-means algorithm because it fails in some important clustering cases (Figure 1). In such cases, another method called spectral clustering is used to solve clustering problems. Spectral clustering is more flexible and adaptable to different data distributions than k-means clustering [15, 16]. In graph theory, spectral clustering is equivalent to a graph cut problem, which can be solved by calculating the first k smallest eigenvalues and the corresponding eigenvectors of the graph Laplacian matrix. Hence, the

* Corresponding author (email: xiaoting@uestc.edu.cn)

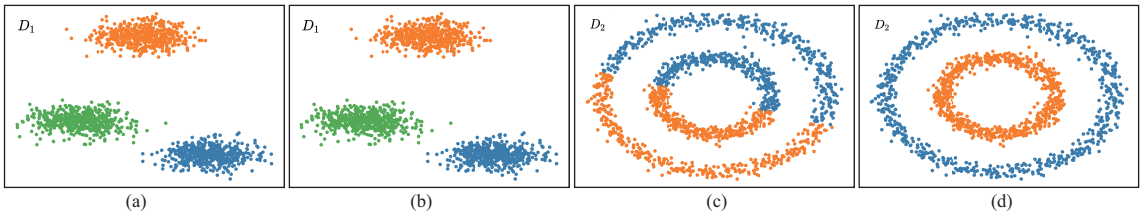

Figure 1 (Color online) Comparison of clustering results for data sets D_1 and D_2 under two different methods, k-means and spectral clustering: (a) and (c) under k-means, and (b) and (d) under spectral clustering. Both methods give good clustering for D_1, but only spectral clustering give a reasonable clustering for D_2.

essence of a quantum spectral clustering algorithm is to solve an eigenvalue problem on a quantum computer. The originally established quantum algorithm for spectral clustering was based on biased phase estimation [17], but its success cannot be guaranteed. Another effort is made with a crucial assumption on the availability of a QRAM [18]; in reality, physically building a QRAM is still a challenge. One major challenge of designing such a quantum algorithm is to create an initial state that overlaps with every eigenvector of the Laplacian matrix with an equal probability amplitude.

To address this problem, we propose an alternative quantum algorithm for spectral clustering based on a bipartite maximally entangled initial state. Compared with existing quantum k-means algorithm, our proposal does not require a QRAM or a quantum adiabatic process. We encode the Laplacian matrix of the clustering problem into a d-sparse Hamiltonian which can be efficiently simulated. Our designed clustering circuit combines Grover's search [19], quantum phase estimation [20], and d-sparse Hamiltonian simulation [21] in a way that the entire circuit complexity has a speedup over the well-known classical eigensolvers. After passing through the clustering circuit, the quantum state undergoes a quantum measurement of a pre-chosen observable. Finally, the measurement outcome is optimized over the observable choice, and the desired clustering result is achieved. Before getting into the detail of our quantum proposal, we first briefly review the classical spectral clustering algorithm.

2 Classical spectral clustering algorithm

Given a dataset $D = \{\boldsymbol{v}_i\}_{i=0}^{N-1} \subset \mathbb{R}^M$, a clustering task is to group the points in D into k subgroups based on their similarities, and the clustering outcome is described by a partition $\{P_j\}_{j=0}^{k-1}$ with $D = \bigcup_{j=0}^{k-1} P_j$ and $P_i \cap P_j = \varnothing$, for $i \neq j$. The value of k is given as an input of the clustering problem.

In order to apply the spectral clustering algorithm, we further assume the similarity between points $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ is characterized by a similarity function S: $S(\boldsymbol{v}_i, \boldsymbol{v}_j) = S(\boldsymbol{v}_j, \boldsymbol{v}_i) \in [0, +\infty)$, satisfying $S(\boldsymbol{v}_i, \boldsymbol{v}_j)$ is large if the two points are from the same subgroup, and is small if they are from different subgroups. Compared with the k-means algorithm that deals with the dataset D directly, the spectral clustering algorithm deals with the similarity graph $G(V_D, E_{DS})$, whose nodes are points in D, and edges between every pair of points that are $(d-1)$-nearest neighbors to each other. In other words, two points $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ in G are connected by an edge e_{ij} with a weight $S(\boldsymbol{v}_i, \boldsymbol{v}_j)$, if they are $(d-1)$-nearest neighbors to each other. Here, d is a value chosen by the user to characterize the local connectivity of every point to its surrounding neighbors. In most applications, d is independent of N and $d \ll N$, which we will take as assumptions of the clustering problem. Then we can define W as the adjacent matrix of $G(V_D, E_{DS})$, with $w_{ij} = S(\boldsymbol{v}_i, \boldsymbol{v}_j)$ for each edge e_{ij} in G. Then the associated Laplacian matrix $L \in \mathbb{R}^{N \times N}$ for $G(V_D, E_{DS})$ is defined by

$$L_{ij} = \begin{cases} -w_{ij}, & i \neq j, \\ \sum_{l=0}^{N-1} w_{il}, & i = j. \end{cases} \tag{1}$$

L is symmetric and positive-semidefinite, with eigenvectors $\{\boldsymbol{u}_i\}$ and eigenvalues $\{\lambda_i\}$ satisfying $0 = \lambda_0 \leqslant \lambda_1 \leqslant \cdots \leqslant \lambda_{N-1}$. Notice that L has at least one zero eigenvalue, i.e., $\lambda_0 = 0$.

The idea of spectral clustering comes from the property of the Laplacian matrix L: the number of zero eigenvalues of L is equal to the number of connected components of the similarity graph. Hence, eigenvalues of L equal to or close to zero will provide useful clustering information. The classical spectral

clustering algorithm consists of two steps. In step 1, we calculate the first k smallest eigenvalues $\{\lambda_i\}$, $i = 0, \ldots, k-1$ as well as their corresponding eigenvectors u_i, and construct the matrix $A \equiv [\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{k-1}] \in \mathbb{R}^{N\times k}$. The rows of A are denoted as $\{\boldsymbol{y}_i\}$, $i = 0, \ldots, N-1$. In step 2, we apply the k-means algorithm to $\{\boldsymbol{y}_i\}$ and group them into k subgroups $\{C_j\}_{j=0}^{k-1}$. It turns out that the clustering $\{C_j\}$ on $\{\boldsymbol{y}_i\}$ leads to a good clustering $\{P_j\}$ on $D = \{\boldsymbol{v}_i\}$: if $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ belong to the same subgroup C_j, then $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ belong to the same subgroup P_j [15,16]. The entire process of spectral clustering is summarized in Algorithm 1. Since the time complexity of numerically calculating the first k smallest eigenvalues of L is $\mathcal{O}(kN^3)$ (e.g., using the inverse power method [22]), the time complexity of spectral clustering is at least $\mathcal{O}(kN^3)$.

Algorithm 1 Classical spectral clustering algorithm

Input: A dataset $D = \{\boldsymbol{v}_j\}$, a given value d, and the number of clusters k.
Output: Clusters $P_1, \ldots, P_k$.
1: Construct the similarity graph $G(V_D, E_{DS})$ upon D and $S(\boldsymbol{v}_i, \boldsymbol{v}_j)$.
2: Compute the graph Laplacian matrix L.
3: Calculate the first k eigenstates $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{k-1}$ of L.
4: Let $A \equiv [\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{k-1}] \in \mathbb{R}^{N\times k}$ and denote the i-th row of A as $\boldsymbol{y}_i \in \mathbb{R}^k$.
5: Cluster $\boldsymbol{y}_{i=0,\ldots,N-1} \in \mathbb{R}^k$ into k different clusters $C_0, \ldots, C_{k-1}$ with k-means algorithm.
6: Generate $P_0, \ldots, P_{k-1}$ by $P_j = \{\boldsymbol{v}_i | \boldsymbol{y}_i \in C_j\}$.

3 Quantum spectral clustering algorithm

As discussed above, the critical step of spectral clustering is to calculate the eigenvalues of the Laplacian matrix L for data set D. In this work, we assume that as the set size $N = |D|$ grows, all points in D are confined in the same compact region $D_c \subset \mathbb{R}^M$. In the classical case, there are many ways of defining the similarity function S, the similarity graph $G(V_D, E_{DS})$ and the corresponding L; in the quantum case, analogous to the classical case, for a given d, we can define the $(d-1)$-nearest neighbor graph $G(V_D, E_{DS})$, with its adjacent matrix $W = (w_{ij})$. For convenience, we choose all nonzero w_{ij} to be 1. Then the Laplacian matrix L can be defined by (1). Assuming $d \ll N$, we have L is Hermitian and d-sparse. We also assume $N = 2^n$, so that L can be encoded as a Hamiltonian on an n-qubit system. In addition, L can be further rescaled by a factor of $\frac{1}{2d}$ so that after rescaling all eigenvalues of L fall into the interval $[0, 1]$. Such treatment will not alter the corresponding eigenstates, nor the property of the Laplacian matrix, but will make it convenient to encode L into the quantum circuit. We denote the eigenvalues and eigenstates of L as $\{\lambda_j\}_{j=0}^{N-1}$ and $\{|u_j\rangle\}_{j=0}^{N-1}$, respectively. In addition, for every λ_j, its binary representation is $\lambda_j = 0.\lambda_{j1}\lambda_{j2}\lambda_{j3}\cdots$, where $\lambda_{jl} \in \{0, 1\}$. It turns out that quantum phase estimation (QPE) [20] embedded in Grover's search is a powerful method to calculate the eigenvalues of L. For $U = \mathrm{e}^{2\pi iL}$ and one eigenpair $(\mathrm{e}^{2\pi i\lambda_k}, |u_k\rangle)$ of U, the quantum phase estimation circuit U_{pe} can be constructed from the inverse quantum Fourier transform and a series of controlled-U^j gates, satisfying

$$U_{\mathrm{pe}}|0\rangle^{\otimes t}|u_k\rangle = |\tilde{\lambda}_k\rangle|u_k\rangle, \tag{2}$$

where $|\tilde{\lambda}_k\rangle = |\lambda_{k1}\lambda_{k2}\cdots\lambda_{kt}\rangle$ and its measurement outcome is a t-bits estimate of λ_k, up to an error precision $\mathcal{O}(2^{-t})$. For convenience, the first register is called the phase register, and the second called the eigenstate register. To implement quantum spectral clustering, it is sufficient to choose $U = \mathrm{e}^{2\pi iL}$ in U_{pe} and then to calculate the eigenstates of L. Notice that the value of d cannot be chosen too large; otherwise it would mistakenly connect two clusters that are generically distinctive. Hence, for our case of clustering on a compact set, d has an upper bound independent of N. Thus, $U = \mathrm{e}^{2\pi iL}$ can be efficiently simulated on a quantum circuit [21]. The entire procedure of our quantum algorithm for spectral clustering consists of four steps, as illustrated in Figure 2.

Step 1: Initial state preparation. If the input state of U_{pe} is $|0\rangle|\phi\rangle$, where $|\phi\rangle = \sum_{i=0}^{N-1}\alpha_i|u_i\rangle$, then we have

$$U_{\mathrm{pe}}|0\rangle|\phi\rangle = \sum_{i=0}^{N-1}\alpha_i U_{\mathrm{pe}}|0\rangle|u_i\rangle = \sum_{i=0}^{N-1}\alpha_i|\lambda_i\rangle|u_i\rangle. \tag{3}$$

Since $\{|u_i\rangle\}$ are unknown before the calculation, some of these amplitudes α_i could be significantly small for a poor choice of $|\phi\rangle$, which could lead to a problem in the subsequent steps. To address this problem, we propose to introduce an ancilla register and prepare a bipartite maximally entangled initial state. We need Lemma 1 for further analysis [23].

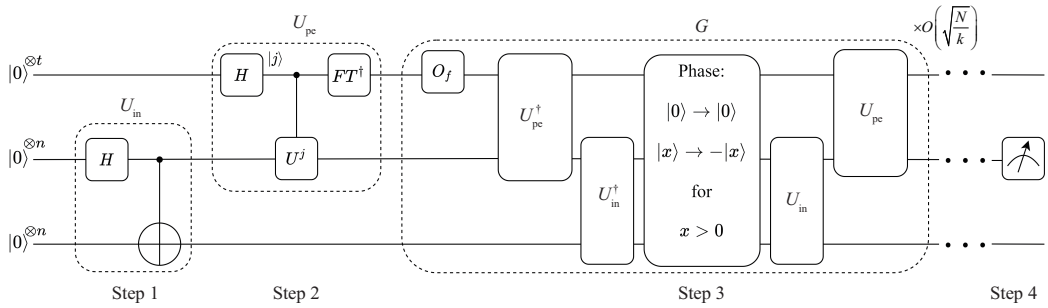

Figure 2 Quantum circuit illustrating our quantum spectral clustering algorithm. The quantum processor contains three registers, the phase register, the eigenstate register and an ancilla register, with t, n and n qubits respectively. The value of t determines the accuracy of phase estimation and $n = \log(N)$. The entire circuit contains four steps: initial state preparation (step 1), quantum phase estimation with $U = \mathrm{e}^{2\pi i L}$ (step 2), Grover iteration sequence to find the eigenvalues of L less than $\tilde{\lambda}$ (step 3), and quantum measurement on the eigenstate register to evaluate and optimize $\langle M\rangle = \mathrm{Tr}(\rho M)$, where $M = XX^{\mathrm{T}}$ and X is the clustering indicator matrix (step 4).

Lemma 1. For an N-dimensional quantum system, let $\{|i\rangle\}$ be the computational basis, and $\{|v_j\rangle\}$ be another orthonormal basis arbitrarily chosen, $j = 0, 1, \ldots, N-1$. Then we have

$$|\Phi\rangle \equiv \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle|i\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |v_j\rangle|v_j^*\rangle, \tag{4}$$

where $|v_j^*\rangle$ is the complex conjugate of $|v_j\rangle$.

Proof. Assuming $|v_j\rangle = \sum_i \alpha_{ji}|i\rangle$, since

$$|i\rangle = \sum_j |v_j\rangle\langle v_j|i\rangle = \sum_{j,k,m} \alpha_{jk}\alpha_{jm}^*|k\rangle\langle m|i\rangle = \sum_{j,k} \alpha_{jk}\alpha_{ji}^*|k\rangle,$$

we have

$$\sum_j |v_j\rangle|v_j^*\rangle = \sum_j \sum_k \alpha_{jk}|k\rangle \sum_i \alpha_{ji}^*|i\rangle = \sum_i \left(\sum_{j,k} \alpha_{jk}\alpha_{ji}^*|k\rangle \right) |i\rangle = \sum_i |i\rangle|i\rangle.$$

This lemma implies that the maximally entangled state $|\Phi\rangle$ can be expressed in any orthonormal basis, either known or unknown. In particular, we can choose $|v_i\rangle = |u_i\rangle$ in (4), where $\{|u_i\rangle\}_{i=0}^{N-1}$ is the eigenbasis of L and is unknown before the calculation. The significance of this lemma is, $|\Phi\rangle$ can be efficiently prepared using the known computational basis $\{|i\rangle\}$, and then it can be expressed in terms of the unknown basis $\{|u_i\rangle\}$, but with known coefficients. Notice that $|\Phi\rangle$ can be efficiently constructed using $U_{\rm in} \equiv \prod_{i=1}^{n} \mathrm{CNOT}_{i,i+n} H^{\otimes n}$: $|\Phi\rangle = U_{\rm in}|0\rangle^{\otimes 2n}$. Here, $U_{\rm in}$ consists of n Hadamard gates and n CNOT gates, where $H^{\otimes n}$ acts on the first n qubits and $\mathrm{CNOT}_{(i,i+n)}$ denotes the CNOT gate with the i-th qubit as the control and the $(i+n)$-th qubit as the target, as illustrated in Figure 2. Hence, in step 1, we prepare the three registers, the phase register, the eigenstate register, and the ancilla register into the initial state $|\psi_0\rangle \equiv |0\rangle^{\otimes t}|0\rangle^{\otimes n}|0\rangle^{\otimes n}$, and then apply $U_{\rm in}$ to get: $U_{\rm in}|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |0\rangle^{\otimes t}|i\rangle|i\rangle$.

Step 2: Applying quantum phase estimation. After applying $U_{\rm in}$ to $|\psi_0\rangle$, we apply the phase estimation circuit $U_{\rm pe}$ to obtain:

$$|\psi\rangle_{\rm pe} \equiv U_{\rm pe}U_{\rm in}|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} U_{\rm pe}|0\rangle^{\otimes t}|i\rangle|i\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} U_{\rm pe}|0\rangle^{\otimes t}|u_i\rangle|u_i^*\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |\lambda_i\rangle|u_i\rangle|u_i^*\rangle, \tag{5}$$

where we have used $U_{\rm pe}|0\rangle^{\otimes t}|u_i\rangle = |\lambda_i\rangle|u_i\rangle$. From this, we can see the advantage of preparing $|\Phi\rangle$: each overlap α_i between $|\psi\rangle_{\rm pe}$ and every $|\lambda_i\rangle|u_i\rangle|u_i^*\rangle$ is known and equal to $\frac{1}{\sqrt{N}}$, even though $\{|u_i\rangle\}$ is unknown.

In order to efficiently construct $U_{\rm pe}$, we need to efficiently construct the following controlled-U gate, with $U = \mathrm{e}^{2\pi i L}$:

$$CU = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes \mathrm{e}^{2\pi i L} = \mathrm{e}^{2\pi i (|1\rangle\langle 1| \otimes L)}, \tag{6}$$

where $H \equiv |1\rangle\langle 1| \otimes L$ is also a d-sparse matrix. According to the well-known results in quantum simulation, since $H = |1\rangle\langle 1| \otimes L$ is d-sparse, CU and hence $U_{\rm pe}$ can be efficiently constructed, and the quantum complexity for $U_{\rm pe}$ is $\mathcal{O}(\mathrm{poly}(\log N)d^4/\epsilon)$, where ϵ represents the error of the estimated phase [21].

Step 3: Applying Grover's search to find the eigenvalues of L less than the threshold. In classical spectral clustering, the number of clusters k_0 is chosen as the input of the clustering problem, and a classical eigenvalue-solving algorithm will be applied to find the k_0 smallest eigenvalues of L. In comparison, in our quantum spectral clustering algorithm, we choose a threshold value $\tilde{\lambda} > 0$, and apply Grover's search [19] to $|\psi\rangle_{\rm pe}$ to find eigenstates of L smaller than $\tilde{\lambda}$. Specifically, given $\tilde{\lambda}$, the desired classical oracle f can be defined as follows:

$$f(x) = \begin{cases} 1, & \dfrac{x}{2^t} < \tilde{\lambda}, \\ 0, & \dfrac{x}{2^t} \geqslant \tilde{\lambda}, \end{cases} \tag{7}$$

where x is a t-bit Boolean variable. Then based on the reversible classical circuit that generates f, we can construct the corresponding quantum circuit that generates the quantum oracle O_f [24], satisfying

$$O_f|x\rangle = (-1)^{f(x)}|x\rangle. \tag{8}$$

Such O_f will add a phase -1 to all $|x\rangle$ satisfying $\frac{x}{2^t} < \tilde{\lambda}$. Then based on O_f, we can construct the Grover iteration G,

$$G \equiv U_{\rm inv}O_f = (2|\psi\rangle_{\rm pe}\langle\psi|_{\rm pe} - I)O_f, \tag{9}$$

where $U_{\rm inv} \equiv (2|\psi\rangle_{\rm pe}\langle\psi|_{\rm pe} - I)$ is the initial-state inversion with respect to $|\psi\rangle_{\rm pe}$. Hence, the constructions of f and O_f are completely determined by the value of $\tilde{\lambda}$, and does not require any prior information about the eigenvalues of L.

In addition, a well-known result in quantum computation is, if the circuit complexity to generate the classical oracle f is C, then the circuit complexity of the quantum oracle O_f is $\mathcal{O}(\mathrm{poly}(C))$ [24]. For our case, since division and comparison are performed on t bits to evaluate f, the classical circuit complexity to calculate f in (7) is $\mathrm{poly}(t)$, and hence the complexity of the above O_f is $\mathcal{O}(\mathrm{poly}(t))$, where $t = n + \lceil 2 + \log\frac{1}{2\epsilon_0}\rceil$, $n = \lceil \log N\rceil$ and $1 - \epsilon_0$ is the success probability of phase estimation. Hence, the circuit complexity to construct O_f is $\mathcal{O}(\mathrm{poly}(\log(N)))$, i.e., O_f and hence G can be efficiently constructed.

Since we are only interested in finding eigenvalues of L close to zero, $\tilde{\lambda}$ should be sufficiently small. As a rule of thumb, $\tilde{\lambda} = \frac{1}{2N}$ is a good first trial, and will give a reasonably good clustering. The value of $\tilde{\lambda}$ determines the number k of eigenvalues smaller than $\tilde{\lambda}$, and we can rewrite $|\psi\rangle_{\rm pe}$ as

$$|\psi\rangle_{\rm pe} = \frac{1}{\sqrt{N}}\sum_{\lambda_i<\tilde{\lambda}}|\lambda_i\rangle|u_i\rangle|u_i^*\rangle + \frac{1}{\sqrt{N}}\sum_{\lambda_i\geqslant\tilde{\lambda}}|\lambda_i\rangle|u_i\rangle|u_i^*\rangle. \tag{10}$$

After applying a standard Grover iteration sequence of length $r = \lceil\frac{\pi}{4}\sqrt{\frac{N}{k}}\rceil$ to the initial state $|\psi\rangle_{\rm pe}$, we obtain the output state:

$$|\psi\rangle_{\rm out} = \frac{1}{\sqrt{k}}\sum_{\lambda_i<\tilde{\lambda}}|\lambda_i\rangle|u_i\rangle|u_i^*\rangle \tag{11}$$

with high probability as long as $k \ll N$. Alternatively, we can apply a fixed-point Grover's search [25] to $|\psi\rangle_{\rm pe}$ with the final state converging to $|\psi\rangle_{\rm out}$ with an arbitrarily high precision. The total quantum complexity in step 3 in terms of N and k is $\mathcal{O}(\sqrt{\frac{N}{k}}\,\mathrm{poly}(\log N))$; in comparison, on a classical computer, the time complexity is $\mathcal{O}(kN^3)$ using the inverse power method [22].

At this point, before we move on to step 4, we need to find the actual value of k, given $\tilde{\lambda}$. This can be achieved by applying the quantum counting circuit to $|\psi\rangle_{\rm pe}$ [26]. The value of k will then be used to construct the clustering indicator matrix X in step 4. Details of the quantum counting circuit will be discussed in the next section.

Step 4: Taking quantum measurement and optimizing over the clustering indicator matrix. At the end of step 3, after we obtain the value of k using quantum counting, we need to repeat the quantum circuit of $U_{\rm in}$, $U_{\rm pe}$ and the Grover's iterations to prepare the register into $|\psi\rangle_{\rm out}$ again. Then we take a quantum

measurement on the eigenstate register, whose density matrix can be derived by taking the partial trace of $|\psi\rangle_{\text{out}}$:

$$\rho = \text{Tr}_{1,3}\left(|\psi\rangle_{\text{out}}\langle\psi|_{\text{out}}\right) = \frac{1}{k}\sum_{i=0}^{k-1}|u_i\rangle\langle u_i| = \frac{1}{k}AA^{\text{T}}, \tag{12}$$

where $A \equiv [\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{k-1}] \in \mathbb{R}^{N\times k}$, with rows of A denoted as $\{\boldsymbol{y}_j\}$. As discussed in the classical spectral clustering algorithm, if we apply a clustering for $\{\boldsymbol{y}_j\}$, then the clustering outcome for $\{\boldsymbol{y}_j\}$ will correspond to a good clustering outcome for $\{\boldsymbol{v}_j\}$. In addition, the clustering for $\{\boldsymbol{y}_j\}$ can be conveniently realized in our proposal due to the relation $\rho = \frac{1}{k}AA^{\text{T}}$. Specifically, for any k-partition $\{C_i\}_{i=0}^{k-1}$ of $\{\boldsymbol{y}_i\}_{i=0}^{N-1}$, we define the clustering indicator matrix $X = (x_{ij}) = [\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-1}] \in \mathbb{R}^{N\times k}$ satisfying

$$x_{ij} = \begin{cases} \dfrac{1}{\sqrt{s_j}}, & \boldsymbol{y}_i \in C_j, \\ 0, & \boldsymbol{y}_i \notin C_j, \end{cases} \tag{13}$$

where $s_j = |C_j|$. Then the clustering of $\{\boldsymbol{y}_i\}_{i=0}^{N-1}$ can be formulated as an optimization problem with the following objective function [27]:

$$\max_X \frac{1}{k}\sum_{i=0}^{k-1}\text{Tr}(\boldsymbol{u}_i\boldsymbol{u}_i^{\text{T}}XX^{\text{T}}) = \max_X \text{Tr}(\rho XX^{\text{T}}) = \max_X \text{Tr}(\rho M) = \max_X \langle M\rangle, \tag{14}$$

subject to X defined by (13), where $M \equiv XX^{\text{T}}$ can be considered as a measurement observable. In our proposal, the above optimization problem is equivalent to finding an optimal observable $M = XX^{\text{T}}$ satisfying (13) to maximize $\langle M\rangle = \text{Tr}(\rho M)$. Strictly speaking, finding the exact optimal X is an NP-hard problem, but for most applications of clustering, a sub-optimal solution is sufficient. Many heuristic algorithms, including hill climbing [28], are good enough to find an acceptable sub-optimal X within a polynomial number $\mathcal{O}(kN)$ of iterations. We denote C_{eigen} as the quantum eigenvalue-solving circuit composed by U_{in}, U_{pe} and the Grover's iteration sequence, and it maps $|\psi_0\rangle$ to $|\psi\rangle_{\text{out}}$. In each iteration of the hill climbing algorithm, the quantum circuit C_{eigen} is repeatedly implemented for n_M times to get an estimation of $\langle M\rangle$. Since the complexity of C_{eigen} is $\mathcal{O}(\sqrt{N/k}\,\text{poly}(\log N))$, the total quantum circuit complexity for C_{eigen} in all these $\mathcal{O}(kN)$ iterations becomes $\mathcal{O}(\sqrt{k}N^{3/2}\,\text{poly}(\log N))$, where we have used the fact that n_M only depends on the estimation accuracy of $\langle M\rangle$, and does not increase as N increases.

4 Determine the number of eigenvalues below threshold

As mentioned above, at the end of step 3 in Figure 2, based on $|\psi\rangle_{\text{pe}}$, we need to calculate the value of k, i.e., the number of eigenvalues (with multiplicities) of L smaller than the given threshold $\tilde{\lambda}$. Specifically, from (10), we have

$$\begin{aligned} |\psi\rangle_{\text{pe}} &\equiv \sqrt{\frac{N-k}{N}}|\alpha\rangle + \sqrt{\frac{k}{N}}|\beta\rangle = \cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle, \\ |\alpha\rangle &\equiv \sqrt{\frac{1}{N-k}}\sum_{\lambda_i \geqslant \tilde{\lambda}}|\lambda_i\rangle|u_i\rangle|u_i^*\rangle, \\ |\beta\rangle &\equiv \sqrt{\frac{1}{k}}\sum_{\lambda_i < \tilde{\lambda}}|\lambda_i\rangle|u_i\rangle|u_i^*\rangle. \end{aligned}$$

Hence, $|\psi\rangle_{\text{pe}}$ is located on a 2-dimensional subspace, on which G is invariant and has the form:

$$G = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{15}$$

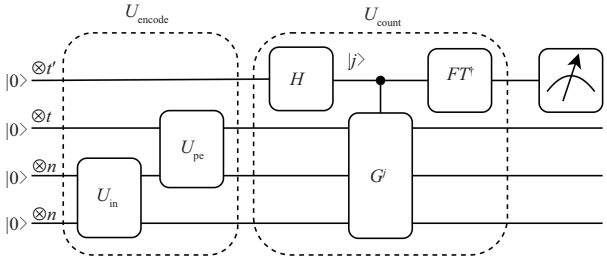

Figure 3 Schematic circuit for quantum counting to estimate calculate the number k of eigenvalues of L less than $\tilde{\lambda}$, where $U_{\text{encode}} \equiv U_{\text{pe}}U_{\text{in}}$, and $G \equiv U_{\text{inv}}O_f$.

G has two eigenvalues $\mu_1 = \mathrm{e}^{i\theta}$ and $\mu_2 = \mathrm{e}^{i(2\pi-\theta)}$, with two corresponding eigenvectors $|a_1\rangle$ and $|a_2\rangle$. For $k \ll N$, we have $0 < \theta = 2\arcsin\sqrt{\frac{k}{N}} < \frac{\pi}{2}$. We can apply the quantum counting circuit U_{count} in Figure 3 to find the value of θ. Specifically, for input state $|0\rangle^{\otimes t'}|\psi\rangle_{\text{pe}}$, the output state of U_{count} becomes:

$$U_{\text{count}}|0\rangle^{\otimes t'}|\psi\rangle_{\text{pe}} = \alpha_1|\theta\rangle|a_1\rangle + \alpha_2|2\pi-\theta\rangle|a_2\rangle. \tag{16}$$

Finally, through measuring the phase register, we obtain a measurement outcome of either θ or $2\pi-\theta$. We can tell which is which since the former is smaller than $\frac{\pi}{2}$ and the latter is larger than $\frac{3\pi}{2}$. Thus, we find the value of θ, and so as the value of k. By increasing the number of qubits in the phase register, one can reduce the uncertainty $|\Delta k|$ to a value less than 1, and then the value of k can be uniquely determined. The total circuit complexity of quantum counting is $\mathcal{O}(\sqrt{kN})$.

5 Complexity analysis

In classical spectral clustering, given L, the number of clusters k_0 is chosen by the user and can be considered as part of the clustering problem input. In comparison, in our quantum spectral clustering algorithm, before implementing the quantum circuit, a threshold $\tilde{\lambda}$ is chosen, and it completely determines the number k of eigenvalues of L smaller than $\tilde{\lambda}$, where k is equal to the number of clusters in the final clustering outcome. Hence, given L, in our proposal, the value of $\tilde{\lambda}$ completely determines the final clustering outcome, and hence it is taken as an input of the clustering problem. However, if someone is interested in using our algorithm to group the data set into exactly k_0 clusters, then we can do as follows: assuming the eigenvalues of L are sorted in a non-decreasing order, with $0 = \lambda_0 \leqslant \cdots \leqslant \lambda_{k_0-1} < \lambda_{k_0} \leqslant \cdots \leqslant \lambda_{N-1} \leqslant 1$. Here, we can reasonably assume $\lambda_{k_0} - \lambda_{k_0-1} = \delta > 0$, for if $\delta = 0$ then the data should be partitioned into $k+1$ rather than k clusters. Then we can apply the binary search algorithm to generate a sequence of $\{\tilde{\lambda}^{(i)}\}_{i=1}^{m}$ such that the final value $\tilde{\lambda}^{(m)} \in (\lambda_{k_0-1}, \lambda_{k_0}]$ with exactly $k = k_0$ eigenvalues of L smaller than $\tilde{\lambda}^{(m)}$. The total number of binary-search iterations is $m = \mathcal{O}(\log\frac{1}{\delta})$. In practice, if k_0 corresponds to a good clustering outcome, then δ must be far from zero, and m must be pretty small. Hence, due to the logarithm property of $m = \mathcal{O}(\log\frac{1}{\delta})$, we can efficiently find the value $\tilde{\lambda}$ satisfying $k = k_0$ through binary search.

In addition, similar to $\tilde{\lambda}$, the value of d is also taken as an input of the clustering problem. The choice of d and the generation of L belong to the data preprocessing part, and are taken as the preliminary information that will be used for the classical and the quantum spectral clustering algorithms. However, d does appear in the circuit complexity expression of simulating the unitary $U = \mathrm{e}^{2\pi iL}$. We will take d as a value chosen by the user for the clustering problem, while N and d are taken as independent. In most applications, d is chosen to be a constant much smaller than N. In the following, for given values of d and $\tilde{\lambda}$, we analyze the quantum circuit complexity of all steps in Figure 2, including quantum counting. The entire algorithm is summarized in Algorithm 2. First, quantum counting has a query complexity of $\mathcal{O}(\sqrt{kN})$ and each query has a circuit complexity of $O\big(\text{poly}(\log N)d^4/\epsilon\big)$, where ϵ denotes the error of the estimated phase in U_{pe}. Then we analyze the four steps of the circuit illustrated in Figure 2. The complexity of U_{in} is $\mathcal{O}(\log N)$, and the complexity of U_{pe} is $\mathcal{O}(\text{poly}(\log N)d^4/\epsilon)$. For step 3, the query complexity of the Grover iteration circuit is $\mathcal{O}(\sqrt{\frac{N}{k}})$, and each query has circuit complexity $O(\text{poly}(\log N)d^4/\epsilon)$. For step 4, the hill-climbing algorithm consists of $\mathcal{O}(kN)$ iterations, and each iteration repeatedly implements C_{eigen} for n_M times, with n_M independent of N. The total quantum circuit complexity for C_{eigen} in all

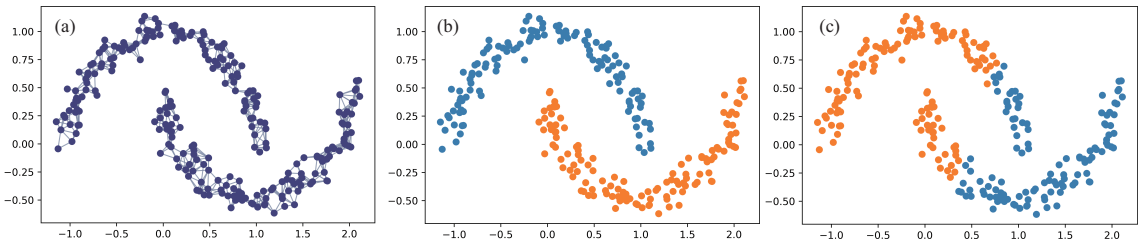

Figure 4 (Color online) (a) An illustrative image describing data set D_1 containing 256 points and the corresponding 8-nearest neighbor graph: two points are connected by a solid-line edge if they are an 8-nearest neighbor to each other; (b) for $\tilde{\lambda} = \frac{1}{2^9}$, the clustering results for D_1 using our quantum spectral clustering algorithm and (c) the clustering result using classical k-means with $k = 2$. The performance of our method for D_1 is much better than k-means.

these $\mathcal{O}(kN)$ iterations adds up to $\mathcal{O}(\sqrt{k}N^{3/2}\,\mathrm{poly}(\log N)d^4/\epsilon)$. Hence, the total complexity of the entire algorithm becomes $\mathcal{O}((kN\sqrt{\frac{N}{k}} + \sqrt{kN})\,\mathrm{poly}(\log N)d^4/\epsilon) = O(\sqrt{k}N^{\frac{3}{2}}\,\mathrm{poly}(\log N)d^4/\epsilon)$, demonstrating a notable speedup compared to the complexity of the classical spectral clustering algorithm, e.g., $\mathcal{O}(kN^3)$ for inverse power method to find the k eigenvalues classically [22].

Algorithm 2 Quantum spectral clustering algorithm

Input: A data set $D = \{\boldsymbol{v}_i\}$, a given value d, and a threshold $\tilde{\lambda}$.
Output: Clusters $P_0, \ldots, P_{k-1}$.
1: Given d, construct the Laplacian matrix L.
2: Apply U_{in} to $|\psi_0\rangle$ to prepare the state $\frac{1}{\sqrt{N}}|0\rangle\sum_{i=0}^{N-1}|i\rangle|i\rangle = \frac{1}{\sqrt{N}}|0\rangle\sum_{i=0}^{N-1}|u_i\rangle|u_i^*\rangle$.
3: Apply the quantum phase estimation U_{pe} to generate $|\psi\rangle_{\mathrm{pe}} = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|\lambda_i\rangle|u_i\rangle|u_i^*\rangle$.
4: Apply quantum counting to $|\psi\rangle_{\mathrm{pe}}$ to find the value of k, i.e., the number of eigenvalues of L smaller than $\tilde{\lambda}$.
5: Repeat lines 2 and 3 to get $|\psi\rangle_{\mathrm{pe}}$, and apply Grover's search to obtain $\rho = \frac{1}{k}\sum_{i=0}^{k-1}|u_i\rangle\langle u_i|$ on the eigenstate register.
6: Construct an observable $M \equiv XX^{\mathrm{T}}$ with X satisfying (13), and calculate $\langle M\rangle = \mathrm{Tr}(M\rho)$ through measurement.
7: Repeat lines 5 and 6 and apply hill-climbing algorithm to optimize $\langle M\rangle$ over X. The sub-optimal $X = X^*$ gives the desired clustering outcome $P_0, \ldots, P_{k-1}$, with $P_j = \{\boldsymbol{v}_i | X_{ij}^* \neq 0\}$.

It is worthwhile to point it out that how to choose appropriate values of d and $\tilde{\lambda}$ for a given clustering problem is an interesting open question in spectral clustering research, but our work focuses on how to implement the spectral clustering through a quantum circuit, and hence both d and $\tilde{\lambda}$ are taken as given inputs of the clustering problem.

6 Numerical simulation

To demonstrate how to implement our quantum spectral clustering proposal in solving specific spectral clustering problems, we apply it to two typical problems that are often used to benchmark the performance of clustering algorithms. The first example is a data set $D_1 = \{\boldsymbol{v}^{(i)}\}_{i=0}^{255}$ defined on a 2-dimensional compact set, satisfying $\boldsymbol{v}^{(i)} = [v_1^{(i)}, v_2^{(i)}]^{\mathrm{T}}$ with $v_1^{(i)} \in [-1, 2]$ and $v_2^{(i)} \in [-0.5, 1]$, as illustrated in Figure 4(a). According to Algorithm 2, we first generate the 8-nearest neighbor graph of D, which will be used to generate the 9-sparse $L_1 \in \mathbb{R}^{256\times256}$. Here we choose $d = 8$ which is sufficient to give a good final clustering. Then we choose the threshold $\tilde{\lambda} = \frac{1}{2^9}$, prepare the initial state, and apply quantum counting to find the number of eigenvalues smaller than $\tilde{\lambda}$ to be $k = 2$. Next, we implement the quantum spectral clustering circuit in Figure 2 to obtain the final state ρ of the eigenstate register. Then we choose an initial guess of the clustering indicator matrix $X = X^{(0)}$, and use hill climbing algorithm to optimize $\langle M\rangle = \mathrm{Tr}(M\rho)$ over X, with $M \equiv XX^{\mathrm{T}}$. Through iterative optimization, we finally reach a sub-optimal solution $X = X^*$, which gives a clustering partition $\{P_0, P_1\}$, satisfying $P_j = \{\boldsymbol{v}_i | X_{ij}^* \neq 0\}$, for $i = 0, \ldots, 255$ and $j = 0, 1$. The clustering outcome is shown in Figure 4(b), with the two clusters well separated, demonstrating a good clustering result.

The second example is a dataset D_2 with 256 data points $\boldsymbol{v}^{(i)} = [v_1^{(i)}, v_2^{(i)}]^{\mathrm{T}}$ satisfying $v_1^{(i)} \in [-6, 8]$ and $v_2^{(i)} \in [-2, 6]$ (Figure 5(a)). Analogously to the first example, we choose $d = 8$ and construct the 9-sparse Laplacian matrix L. Then we choose $\tilde{\lambda} = \frac{1}{2^9}$, and run the quantum counting algorithm to find $k = 3$. After that we implement our spectral clustering circuit and hill-climbing algorithm, and through optimization we find a sub-optimal $X = X^*$, corresponding to a clustering partition $\{P_0, P_1, P_2\}$. The result is shown in Figure 5(b).

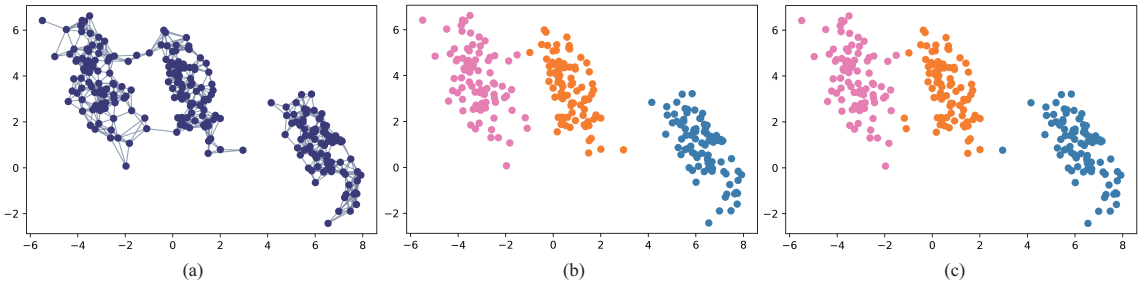

Figure 5 (Color online) (a) An illustrative image describing data set D_2 containing 256 points and the corresponding 8-nearest neighbor graph: two points are connected by a solid-line edge if they are an 8-nearest neighbor to each other; (b) for $\tilde{\lambda} = \frac{1}{2^9}$, the clustering results for D_1 using our quantum spectral clustering algorithm; (c) the clustering result using classical k-means with $k = 3$. The performance of k-means is reasonably good, except for a few points incorrectly clustered.

In comparison, we have also plotted the clustering results for D_1 and D_2 using the k-means algorithm, where we have chosen $k = 2$ and $k = 3$ respectively for the two examples, as shown in Figure 4(c) and Figure 5(c). From these figures, one can see that although the k-means clustering result for the second example is reasonably good, with only a few points incorrectly clustered, it is definitely not good for the first example. Hence, for problems like the first example, spectral clustering is necessary and advantageous to the k-means algorithm.

7 Conclusion

In this article, we explore the possibility of constructing a novel quantum spectral clustering algorithm using an appropriate combination of Grover's search, quantum phase estimation, and Hamiltonian simulation instead of QRAM. The essence of the method is to efficiently solve the eigenvalue problem on a quantum circuit and use the circuit outcome for the clustering task. One crucial trick of the proposed method is the initial preparation of the registers in a bipartite maximally entangled state. Furthermore, we use quantum counting to calculate the actual number of clusters. A suboptimal clustering outcome can be obtained by optimizing the measurement outcome using the hill-climbing algorithm. The overall quantum complexity of our method demonstrates a speedup compared to the classical counterpart. Our method successfully demonstrates the advantage of a quantum processor in solving machine learning problems, and these techniques can be applied to other interesting quantum computational problems.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2018YFA0306703).

References

1 Biamonte J, Wittek P, Pancotti N, et al. Quantum machine learning. Nature, 2017, 549: 195–202
2 Wiebe N, Braun D, Lloyd S. Quantum algorithm for data fitting. Phys Rev Lett, 2012, 109: 050505
3 Rebentrost P, Mohseni M, Lloyd S. Quantum support vector machine for big data classification. Phys Rev Lett, 2014, 113: 130503
4 Ye Z K, Li L Z, Situ H Z, et al. Quantum speedup of twin support vector machines. Sci China Inf Sci, 2020, 63: 189501
5 Lloyd S, Mohseni M, Rebentrost P. Quantum principal component analysis. Nat Phys, 2014, 10: 631–633
6 Amin M H, Andriyash E, Rolfe J, et al. Quantum Boltzmann machine. Phys Rev X, 2018, 8: 021050
7 Dong D Y, Chen C L, Li H X, et al. Quantum reinforcement learning. IEEE Trans Syst Man Cybern B, 2008, 38: 1207–1220
8 Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM Comput Surv, 2009, 41: 1–58
9 van der Maaten L, Postma E, van den Herik J, et al. Dimensionality reduction: a comparative. J Mach Learn Res, 2009, 10: 66–71
10 Jain A K, Murty M N, Flynn P J. Data clustering: a review. ACM Comput Surv, 1999, 31: 264–323
11 Aïmeur E, Brassard G, Gambs S. Quantum clustering algorithms. In: Proceedings of the 24th International Conference on Machine Learning, Corvalis, 2007. 1–8
12 Giovannetti V, Lloyd S, Maccone L. Quantum random access memory. Phys Rev Lett, 2008, 100: 160501
13 Farhi E, Goldstone J, Gutmann S, et al. Quantum computation by adiabatic evolution. 2000. ArXiv:quant-ph/0001106v1
14 Lloyd S, Mohseni M, Rebentrost P. Quantum algorithms for supervised and unsupervised machine learning. 2013. ArXiv:1307.0411
15 Ng A, Jordan M, Weiss Y. On spectral clustering: analysis and an algorithm. In: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, Vancouver, 2001. 849–856
16 von Luxburg U. A tutorial on spectral clustering. Stat Comput, 2007, 17: 395–416
17 Daskin A. Quantum spectral clustering through a biased phase estimation algorithm. Turk World Math Soc J Appl Eng Math, 2017, 10: 1
18 Kerenidis I, Landman J. Quantum spectral clustering. Phys Rev A, 2021, 103: 042415
19 Grover L K. A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, Philadelphia, 1996. 212–219

20 Kitaev A Y. Quantum measurements and the abelian stabilizer problem. 1995. ArXiv:quant-ph/9511026
21 Berry D W, Ahokas G, Cleve R, et al. Efficient quantum algorithms for simulating sparse hamiltonians. Commun Math Phys, 2007, 270: 359–371
22 Wilkinson J H. The Algebraic Eigenvalue Problem. Oxford: Oxford University Press, 1988. 570–646
23 Poulin D, Wocjan P. Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer. Phys Rev Lett, 2009, 103: 220502
24 Watrous J. Quantum computation: lecture notes. 2006. https://cs.uwaterloo.ca/~watrous/QC-notes/
25 Yoder T J, Low G H, Chuang I L. Fixed-point quantum search with an optimal number of queries. Phys Rev Lett, 2014, 113: 210501
26 Brassard G, Høyer P, Tapp A. Quantum counting. In: Automata, Languages and Programming. Berlin: Springer, 1998. 820–831
27 Zha H, He X, Ding C, et al. Spectral relaxation for k-means clustering. In: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, Vancouver, 2001. 1057–1064
28 Russell S J, Norvig P. Artificial Intelligence: A Modern Approach. Upper Saddle River: Prentice Hall, 2010. 131–146