Quantum approximate optimization and k-means algorithms for data clustering

Jirawat Saiphet1,*, Sujin Suwanna1, Thiparat Chotibut3 and Areeya Chantasri1,2

1Optical and Quantum Physics Laboratory, Department of Physics, Faculty of Science, Mahidol University, Bangkok, 10400, Thailand
2Centre for Quantum Computation and Communication Technology (Australian Research Council), Centre for Quantum Dynamics, Griffith University, Brisbane, Queensland, 4111, Australia
3Department of Physics, Faculty of Science, Chulalongkorn University, Bangkok 10330, Thailand

*E-mail: jirawat.sai@outlook.com

Abstract. Noisy intermediate-scale quantum (NISQ) devices are cutting-edge technology expected to demonstrate potential and advantages of quantum computing over classical computing. Its low number of qubits and imperfection from noises restrict running full-scale quantum algorithms on such devices; however, quantum advantages can still be obtained. To achieve quantum advantages from NISQ devices, the hybrid quantum-classical algorithms were introduced. Quantum approximate optimization algorithm (QAOA) is a variational hybrid algorithm, which utilizes a NISQ device as a sub-unit for specific tasks and performs most calculations on a classical computer. QAOA provides an approximate solution, with arbitrary precision as the number of operations increases, for optimization problems. In this work we investigate the possibility of applying QAOA to a clustering problem and compare its performance with the classical k-means algorithm. It turns out that the weights in graph connectivity can degrade the algorithm operation and make it more difficult to approximate the solution. We also benchmark the QAOA by comparing the approximated solutions with the exact one obtained from a classical clustering algorithm.

1. Introduction
The development of quantum technology on noisy intermediate-scale quantum (NISQ) devices has shown great progress, with the realization of quantum supremacy demonstrated by Google’s experiment on a 53-qubit superconducting quantum processor [1]. These devices are in early stage of quantum devices that could perform calculations, which even the best available classical supercomputers cannot. The NISQ devices, even though not with thier full potential, are expected to be fast-approaching devices that have capabilities of solving classical-difficult tasks efficiently and outperforming a typical classical computer [2]. One advantage of the NISQ devices arises from exploiting variational hybrid quantum-classical algorithms to obtain approximate solutions for optimization problems in polynomial time. A promising algorithm we will focus on is the so-called “Quantum Approximate Optimization Algorithm (QAOA)”, which has been employed to solve several optimization problems [3,4]. QAOA can be used to find an approximate maximum cut of a graph, which is in an NP-hard complexity class known as a Max-Cut problem.
The algorithm was shown to beat a classical optimization algorithm when all edges of the graph have identical weights [5]. However, the most recent results have implied that QAOA exhibits lower performance on solving Max-Cut of weighted graphs [6].

In this work, we extend the investigation of the Max-Cut problem of weighted graphs to solve a clustering problem in machine learning [7, 8], where weights are assigned according to the distance between data points. We focus on complete graphs (i.e., graphs with all nodes connected) embedded in a two-dimensional space. The maximum cut of such graph can be used to find an optimal way to separate two distinct clusters, where its practicality at the lowest operation depth ($p = 1$) of QAOA for unsupervised machine learning was shown in Ref. [9]. This could be a significant application for a near-term usage of NISQ device. However, QAOA performance for clustering at higher $p$ has not been well studied, not to mention a rigorous comparison to existing classical algorithms for unsupervised clustering. The question of when and how QAOA outperforms classical clustering presents a remarkable challenge, both from physics and computer science aspects. To shed some light on these issues, we compare the accuracy of QAOA on solving the clustering problem with an unsupervised machine learning algorithm. We specifically use k-means clustering algorithm, and show the numerical results for randomly generated complete graphs.

2. Max-cut and clustering problems

In graph theory, a cut refers to a partition of vertices of a graph into two subsets. The size of a cut is defined as a summation of weighted edges that connect vertices from one subset to its complement. Hence, a maximum cut is the cut that maximizes its size, $C = \sum_{i \in S, j \in \overline{S}} w_{ij}$, where $S$ and $\overline{S}$ denote the two complementing subsets.

In general, even to find an approximate maximum cut of any graph is a non-trivial problem, and classical algorithms are not guaranteed to find the solution in polynomial time [10–12]. Additionally, several other problems can be formulated as Max-Cut problems. For example, the unsupervised clustering problem in machine learning, where a large data set is to be categorized into two different clusters. Maximizing clusters separation can then be formulated as the problem of finding the maximum cut of the dataset-associated graph. To solve the clustering problem, we consider a complete graph with $N$ vertices (NC), which is a graph where all vertices are connected to each other. The complete graph has two subcategories: unweighted graphs and weighted graphs. For the unweighted graphs (uNC), the weights of all edges are equal to 1, and the maximum cut is trivial to solve. The graph that corresponds to data in a two-dimensional plane is a weighed-complete graphs (wNC), where the weights are related to the distance between these vertices. Therefore, solving data clustering is considered as the Max-Cut problem on the wNC graph.

3. QAOA for data clustering

Let us consider a set of $N$ data points each specified by a coordinate $(x_i, y_i)$ for $i = 1, 2, ..., N$. The Euclidean distances for all data pairs can be represented as matrix elements of an adjacency matrix $W$. We will use the Euclidean distance as the distance between data points $i$ and $j$,

$$w_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (1)$$

The adjacency matrix is a symmetric matrix with 0 for all of its diagonal elements. For a max-cut problem, we define a cost function as

$$H_C = - \sum_{i,j}^N \frac{w_{ij}}{2} (1 - \sigma_i^z \sigma_j^z). \quad (2)$$
which should be minimized if the indices \(i\) and \(j\) represent two set of vertices in two clusters with maximum distance from each other. In order to implement the QAOA for the cost in equation (2), we first denote each vertex as a qubit, and \(\hat{\sigma}^z_i, \hat{\sigma}^z_j\) as the z-Pauli matrices of qubits \(i\) and \(j\), respectively. In this representation, the clusters are denoted by eigenstates of \(\hat{\sigma}^z\), 1 and -1.

Data points \(i\) and \(j\) from the same (different) cluster will have the same (different) eigenstates. Following the QAOA protocol, all qubits are prepared in the product state \(|+\>_N\) which is a ground state of \(\hat{B} = \sum_j \hat{\sigma}^z_j\). By defining two sets of parameters \(\vec{\beta} = (\beta_1, \beta_2, ..., \beta_p)\) and \(\vec{\gamma} = (\gamma_1, \gamma_2, ..., \gamma_p)\), a quantum device applies an alternation of evolutions to obtain the parametrized state

\[
|\psi(\vec{\beta}, \vec{\gamma})\rangle = \prod_{\alpha=1}^p e^{-i\beta_\alpha \hat{B}_\alpha} e^{-i\gamma_\alpha \hat{H}_C} |+\>_N,
\]

where \(p\) is the repetition time, and \(\hat{B}\) is also called a mixing Hamiltonian. One may think of QAOA as an approximation of the quantum annealing problem that roughly anneals the ground state of \(\hat{B}\) to a ground state of \(\hat{H}_C\). The state in equation (3) is then measured over multiple shots in the computational basis to get a series of bit strings as results. After that, the resulting bit strings are sent back to a classical computer to calculate an expectation value of \(\hat{H}_C\). At this stage, QAOA employs classical optimization algorithms to find an optimal set of new parameters \((\vec{\beta}^*, \vec{\gamma}^*)\), that maximizes the cost function. The process can be repeated multiple times in order to find converging solutions. Once the process is finished, we get the state \(|\psi(\vec{\beta}^*, \vec{\gamma}^*)\rangle\), which gives an approximated solution to data clustering. To quantify the precision of the algorithm, we use the approximation ratio

\[
r = \frac{F_p(\vec{\beta}, \vec{\gamma})}{F_p(\vec{\beta}^*, \vec{\gamma}^*)},
\]

where \(F_p(\vec{\beta}, \vec{\gamma}) := \langle \psi(\vec{\beta}, \vec{\gamma}) | \hat{H}_C | \psi(\vec{\beta}, \vec{\gamma}) \rangle\) or the expectation value of the cost function to be minimized. This ratio is proportional to \(p\) and should converge to one, when \(p \to \infty\) [3, 13].

4. Numerical investigation

4.1. Quantum circuits for QAOA

We simulate the QAOA on Qiskit, a python-based framework for quantum computing developed by IBM [14]. The cost function that contains information about distances among data points are encoded to the quantum circuit via gate operations. At the first step, the program prepares the initial state \(|+\>_N\) by applying the Hadamard gate

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

(5)
to every single qubit. This initial state is a uniform superposition over all computational basis states. The next step is to apply gates that result in the unitary evolution associated with \(\hat{H}_C\), scaled with a parameter \(\gamma\). The gate operation that is applied to any two qubits \(i, j\) is written as

\[
\exp(-i\gamma \hat{H}_C,ij) = U_{ij}(\gamma) \prod_{a \neq j}^N U_{ia}(\gamma) \prod_{b \neq i}^N U_{bj}(\gamma).
\]

(6)

Each \(U_{ij}(\gamma)\) is a two-qubit gate given by

\[
U_{kl}(\gamma) = \exp \left( \frac{i\gamma}{2} (1 - \hat{\sigma}^z_k \hat{\sigma}^z_l) \right) = C_{u1}(-2\gamma)_{kl} u1(\gamma)_{k} u1(\gamma)_{l},
\]

(7)
where the general phase-rotation gate $u_1$ and its controlled-gate $C_{u_1}$ are

$$u_1(\gamma) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\gamma} \end{bmatrix}, \quad C_{u_1}(\gamma) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\gamma} \end{bmatrix}, \quad (8)$$

respectively. Finally, the program applies the $X$-rotation parameterized by $\beta$ to every single qubit $k$,

$$X(2\beta) = \exp(-i\beta \sigma_x) = \begin{bmatrix} \cos(\beta) & -i \sin(\beta) \\ -i \sin(\beta) & \cos(\beta) \end{bmatrix}, \quad (9)$$

which represents the evolution from the mixing Hamiltonian $\hat{B}$.

4.2. Approximation of the cost function

After completing $p$ layers of the above operations, the program measures the qubits in the computational basis, that is, measuring the qubits projectively in the $\sigma_z$ basis for $i = 1, 2, ..., N$. The measurement record for $N$ qubits is a bit string $x \in \{0, 1\}^N$. By repeating multiple shots of the measurement, we can construct an estimate of the state $|\psi(\vec{\beta}, \vec{\gamma})\rangle$ in the computational basis.

The cost function evaluation can be calculated from $\langle \psi(\vec{\beta}, \vec{\gamma}) | \hat{H}_C | \psi(\vec{\beta}, \vec{\gamma}) \rangle$, which represents the total weights of edges between vertices from different clusters. Multiple shots of measurement are required in order to acquire a good approximation of the maximum cut. In this work, we have implemented $2^N + 2$ shots to guarantee the full access to the $N$-qubit Hilbert space.

4.3. Classical k-means algorithm for data clustering

We here introduce the concept of the classical k-means clustering algorithm, which is used in this work to compare the performance of clustering data with the QAOA. There are many advanced classical techniques for data clustering, but we find that the simple k-means algorithm [15] is a good start for our propose. K-means is a centroid-based clustering, where clusters are represented by central vectors. The algorithm starts by appointing trial centers for $k$ clusters randomly on the data. Then, the centers are iteratively relocated, by using average of data positions, until their positions are stable. However, a solution that the k-means produces is heavily dependent on initial conditions, and the algorithm can easily be trapped at local optima. Therefore, intensive re-initializing and repeating the calculations are needed, but still a global optimum is not guaranteed. Moreover, the k-means algorithm needs the number of clusters to be given in advance; therefore, the technique is appropriate for our problem, as the max-cut is restricted to two clusters.

5. Results and discussion

We numerically perform data clustering using the QAOA and the k-means algorithms, and investigate their performance. The two-dimensional data (instances) are generated from the cluster generator of scikit-learn [16], where the random points are generated around two definite centers. Distances among points are measured and converted to the matrix elements of the adjacent matrix $W$. The algorithm depth $p$ is predetermined, and the initial set of variational parameters $(\vec{\beta}, \vec{\gamma})$ are randomly picked from a range of $\left(\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]^p, \ left[-\pi, \pi]^p \right)$ [6]. The classical optimization part of QAOA is performed using Powell’s algorithm, for the intensive search for an optimal value. The precision ratio is then obtained from equation (4).

We expect the quality of the approximated solution to be improved as $p$ gets larger. Therefore, we investigate both the weighted and unweighted complete graphs, where $N = 6$ and 10, for
different values of $p = 1, 2, \ldots, 10$. For each $p$, we use 48 different realizations of the complete graphs. When the problem is an unweighted complete graph, $w_{ij} = 1$ for all $i \neq j$, the QAOA can approximate the cost function $\langle H_C \rangle$ with $r > 90\%$ on average as seen in figure 1. The approximated cost and its accuracy are improved as the depth of operations $p$ increases. The validity of the hybrid algorithm is demonstrated in this simple instances.

On the other hand, for the weighted complete graphs, we observe a significant drop in the QAOA’s performance on optimizing the cost function. The algorithm reveals insignificant improvement in $p$ of the cost function and the approximation ratio (as shown in figure 1) converges to some values of $r \sim 60 - 70\%$. The errors in $r$, from 48 different initializations, does not seem to be remarkably influenced by the operation layers $p$. This result may indicate the inefficiency of QAOA for solving a max-cut of weighted graphs, and modification of the algorithm may be needed in order to improve the performance of the algorithm.

The comparison of the performance between the QAOA and the classical k-means clustering algorithm is shown in figure 2 for $N = 6$ and $N = 10$. The data instances are repeatedly computed with varied cluster initial conditions. The result in figure 2 shows the histograms of the approximation ratio of the QAOA and the k-means algorithms, where the latter can achieve the optimal solution most of the times, and undoubtedly outperforms QAOA in data clustering.

6. Conclusion
We have explored the possibility of near-term utilization of a NISQ device by investigating a hybrid quantum-classical algorithm, in particular, the QAOA for machine learning tasks. We have applied QAOA for a data clustering task by mapping the problem to estimating the optimal max-cut of corresponding complete graphs. Ideally, we expected the QAOA to estimate a better max-cut as the layer of quantum operation $p$ grows. However, we find that, for a complete weighted graph, the approximation ratio stays almost unchanged at a lower value, compared to that for the unweighted graph. Increasing $p$ does not provide so much improvement in the solution quality. This might suggest the impracticality of the QAOA to approximate an optimal solution for a maximum cut of weighted graphs, which are used for data clustering. One open
and intriguing question is then how to improve the QAOA, e.g., finding a better way to pick the variational parameters or improving the classical optimization part. Otherwise, using QAOA for clustering might not be an example for a quantum algorithm that can prevail over existing classical machine learning algorithms.

Acknowledgments
All authors thank J. Tangpanitanon for fruitful discussions. J. Saiphet acknowledges Scholarship for Talented Students from the Faculty of Science, Mahidol University. A. Chantasri acknowledges the support of the Griffith University Postdoctoral Fellowship scheme and Australian Research Council Centre of Excellence Program CE170100012. This research was supported by the Program Management Unit for Human Resources & Institutional Development, Research and Innovation [grant number B05F630108] (Thailand).

References
[1] Arute F et al 2019 Nature 574 505–10
[2] Preskill J 2018 Quantum 2 79
[3] Farhi E, Goldstone J and Gutmann S 2014 e-Print arXiv:1411.4028
[4] Wang Z, Hadfield S, Jiang Z and Rieffel E G 2018 Phys. Rev. A 97(2) 022304
[5] Crooks G E 2018 e-Print arXiv:1811.08419
[6] Zhou L, Wang S T, Choi S, Pichler H and Lukin M D 2020 Phys. Rev. X 10(2) 021067
[7] Jain A K, Muty M N and Flynn P J 1999 ACM Comput. Surv. 31 264–323
[8] Cook W 1995 Combinatorial Optimization (Providence, R.I: American Mathematical Society)
[9] Otterbach J S et al 2017 e-Print arXiv:1712.05771
[10] Hästad J 2001 J. ACM 48 798–859
[11] Khot S, Kindler G, Mossel E and O’Donnell R 2007 SIAM J. Comput. 37 319–57
[12] Goemans M X and Williamson D P 2004 J. Comput. Syst. Sci. 68 442–70
[13] Farhi E, Goldstone J and Gutmann S 2014 e-Print arXiv:1412.6062
[14] Asfaw A et al 2020 Learn Quantum Computation Using Qiskit Online: http://community.qiskit.org/textbook
[15] VanderPlas J 2016 Python Data Science Handbook: Essential Tools for Working with Data (Beijing: OReilly)
[16] Pedregosa F et al 2011 J. Mach. Learn. Res. 12 2825–30