Quantum Spectral Clustering of Mixed Graphs

Daniel Volya and Prabhat Mishra
University of Florida, Gainesville, Florida, USA

***Abstract*—Spectral graph partitioning is a well known technique to estimate clusters in undirected graphs. Recent approaches explored efficient spectral algorithms for directed and mixed graphs utilizing various matrix representations. Despite its success in clustering tasks, classical spectral algorithms suffer from a cubic growth in runtime. In this paper, we propose a quantum spectral clustering algorithm for discovering clusters and properties of mixed graphs. Our experimental results based on numerical simulations demonstrate that our quantum spectral clustering outperforms classical spectral clustering techniques. Specifically, our approach leads to a linear growth in complexity, while state-of-the-art classical counterpart leads to cubic growth. In a case study, we apply our proposed algorithm to preform unsupervised machine learning using both real and simulated quantum computers. This work opens an avenue for efficient implementation of machine learning algorithms on directed as well as mixed graphs by making use of the inherent potential quantum speedup.**

***Index Terms*—Quantum computing, spectral graph clustering, eigenvalue computation, eigenvector projection**

I. Introduction

Quantum computing is powerful because of the inherent handling of an exponential state space along with quantum mechanical effects such as entanglement and superposition. Despite such promise, there are only a few quantum algorithms known to give impressive speedup over classical computers. While this is presently true, a remarkable number of applications stem from these few primitives. One popular class of quantum speedup is derived from quantum simulation. These quantum algorithms provide an exponential speedup over the best known classical algorithms for simulating quantum dynamics. Other examples of these primitives include the influential Shor's algorithm and Grover's search as well as quantum walks [1] and quantum linear systems algorithms. While these primitives may be viewed as independent, all of them rely on ideas based in signal processing and spectral transformation.

These few primitives, and the core concept of spectral transformation, have opened many research areas in quantum algorithms. For example, eigenvalue computation, singular value decomposition, and matrix inversion are central to numerous algorithms in optimization and machine learning. Several works extend the primitives to develop quantum algorithms that provide speedup of matrix computations. This paper focuses on adapting these quantum algorithms to enhance spectral analysis for clustering directed graphs.

Spectral clustering utilizes eigenvalues and eigenvectors from a matrix representation of a graph to discover important substructures, and then solely focus on these substructures to obtain clusters. A distinct advantage of spectral clustering is the familiar vector-space mathematical properties which places guarantees on the solution. In general, graph clustering problems are NP-hard, hence heuristic and machine learning algorithms serve important roles in solving real-world applications. However, it is often unclear whether these algorithms will provide a reasonable solution based on the given graph structure. Although spectral clustering is still a heuristic approach, the results and implications are fully understood by linear algebra.

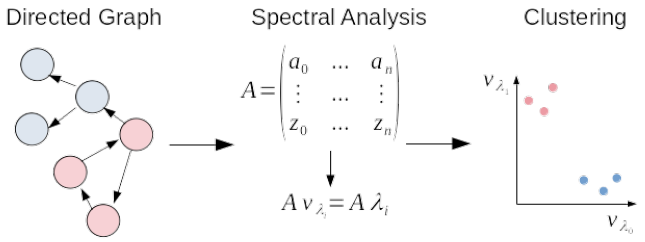

Directed Graph
Spectral Analysis
Clustering
$A = \begin{pmatrix} a_0 & \dots & a_n \\ \vdots & \dots & \vdots \\ z_0 & \dots & z_n \end{pmatrix}$
$A v_{\lambda_i} = A \lambda_i$
v_{λ_1}
v_{λ_0}

Fig. 1: An overview of a spectral clustering algorithm to discover clusters in a directed graph. Quantum algorithms are used to enhance (1) spectral analysis and (2) clustering.

Analyzing the structure of graphs brings insight to many real world phenomena. In particular, graph clustering and partitioning is a commonly used method to address problems in different scientific disciplines including VLSI layout, parallel job scheduling, social network analysis, and image segmentation. Although the potential applications are vast, in this paper we apply the constructed quantum spectral clustering algorithm to small unsupervised problems in machine learning, such as on XOR problem and on a non-convex dataset. This paper makes the following major contributions.

- To the best of our knowledge, our proposed approach is the first attempt at efficient clustering of directed as well as mixed graphs using quantum algorithms.
- We develop core concepts to project eigenvalues and eigenvectors of a graph onto important subspaces purely on a quantum computer.
- We demonstrate the quantum speedup in spectral clustering of directed graphs derived from both real and synthetic datasets.

This paper is organized as follows. Section II provides relevant background on quantum computing and survey related approaches. Section III outlines our problem formulation. Section IV describes our proposed framework for quantum spectral analysis of mixed graphs. Section V presents the experimental results to demonstrate the effectiveness of our proposed approach. Finally, Section VI concludes the paper.

II. Background and Related Work

This section provides a brief introduction to quantum computing followed by an overview of related efforts on graph clustering techniques.

A. Quantum Computing

We briefly describe an overview of quantum computing. The state $|\phi\rangle$ of a qubit lives in a 2-dimensional complex Hilbert space and can be described using the computational basis: $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ where $|\alpha|^2 + |\beta|^2 = 1$. Measuring a qubit will give 0 with a probability of $|\alpha|^2$ and 1 with a probability of $|\beta|^2$. A quantum register $|\psi\rangle$ consists of one or more qubits, with the dimension of the state space growing in order of 2^n where n is the number of qubits. A quantum gate, and in general, a quantum circuit U acts on a quantum register to transform it to another state: $U|\psi\rangle$. Fundamentally, a quantum algorithm will take quantum registers from an initial state $|\psi\rangle = |0\rangle^{\otimes n}$ and transform them to a useful state $|\psi'\rangle$ which encodes a solution to a given problem. The primary challenge is to develop the unitary operators U which define the desired transformation.

B. Graph Clustering

In general, clustering refers to the process of dividing a set into subsets, where elements in each subset are considered to be related. For example, in the area of machine learning and data mining, the task of clustering is generally referred to as "unsupervised learning" where the aim is to feasibly group together similar objects without any prior knowledge about their grouping [2]. Due to the NP-Hard nature of optimal clustering, there has been extensive research efforts to devise clustering mechanism for various problem formulations. Specific to clustering on graphs, mechanisms include approximating minimum cuts, preforming spectral analysis, and finding quasi-cliques [3]. Spectral analysis is a unique method which is solely based on utilizing the eigen-spectrum of a given matrix representation of graph. A number of theoretical work has connected the eigen-spectrum to properties of graphs, such as random walk [4]–[6] and clustering [7], [8]. Recent efforts have explored utilization of quantum computing to reduce time complexity of these approaches with some specific application to graph problems [9], [10]. Most of these efforts focus on undirected graphs. While directed graphs are used in many real-world applications, there is limited effort in applying clustering on directed graphs. As discussed in Section III, there are several fundamental challenges in developing a quantum spectral analysis algorithm targeted at directed graphs.

III. Problem Formulation

Along with an unfavorable run-time, clustering of directed graphs is a more challenging task than clustering of undirected graphs. While there has been significant research effort in clustering undirected graphs, those solutions cannot be directly applied to directed graphs. The purpose of this section is to formally outline the problems seen in directed graphs as well as to define terminology used in later sections.

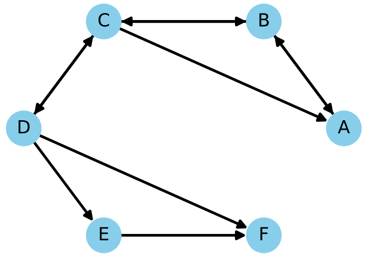


Fig. 2: An example mixed graph with six nodes and 4 directed edges and 3 undirected edges.

A. Directed Graphs

A directed graph is an ordered pair $G = (V, E)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of the graph's vertices, and $E \subseteq V^2$ is the set of its edges. Assigning a real number to each of the edges yields a weighted graph. In case of an undirected graph, any edge will also have a symmetric edge in the opposite direction.

B. Graph Representation using Adjacency Matrix

The adjacency matrix A of a graph $G = (V, E)$ is a $|V| \times |V|$ matrix such that:

$$A = \begin{cases} w_{ij} & \text{if } (i,j) \in E, \forall i, j \in 1, \ldots, |V| \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The definition above is generic for both directed and undirected graphs. However, one crucial property is that the matrix is symmetric, i.e., $A = A^T$ for an undirected graph. This property has led to powerful theoretical insight to undirected graphs. Namely, a symmetric matrix A can be diagonalized as $A = U\Lambda U^T$ where the orthogonal matrix U contains as columns the eigenvectors of A, and $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$ is a diagonal matrix with corresponding eigenvalues as its entries. The eigenvalues of an adjacency matrix defines the spectrum of a graph, and contains close connections with several important graph properties [6].

C. Representation of Mixed Graphs

Symmetric matrices have nice properties that lead to insights in undirected graphs. However, for directed or mixed graphs (e.g. figure. 2), the adjacency matrix is no longer symmetric. In this case, the eigenvalues may be complex-valued. A complex-valued spectrum no longer has the straight-forward connections with several graph properties. This problem limits the usage of non-symmetric matrix representation for directed or mixed graphs, specifically in the case of clustering. To overcome these shortcomings, we propose to transform a non-symmetric matrix into a symmetric matrix, and encode additional properties to maintain resemblance of a directed graph. After obtaining a symmetric matrix, the familiar methods of spectral analysis can be used.

D. Mixed Laplacian

The Laplacian provides a natural link between discrete representation of graphs, and continuous representation of vector spaces. Similar to adjacency matrices, the usual extension of a mixed graph Laplacian to undirected graphs leads to non-symmetric matrices. In a connected graph, where a directed path exists for each node pair, Chung [6] proposed a symmetric version of the Laplacian. In general, graphs are not fully connected. Hence, one strategy is to utilize the PageRank method [11] and assign small probabilities of transition for any given node pairs. This yields a fully connected graph, and the previous theoretical work for directed and mixed Laplacians applies [12].

E. Spectral Clustering

Spectral clustering reduces the space by removing irrelevant contributors. This is practically done by projecting data to the most significant eigenspaces that, in this case, are associated with the smallest eigenvalues. After this transformation, clustering, such as via K-means, can be preformed in the lower dimension. For a n-node graph, the classical algorithm to preform spectral clustering is decomposed as follows: creating the Laplacian matrix in $O(nm)$ where m is the number of graph edges, then eigenvalue and eigenvector extraction in $O(n^3)$, and finally a k-means clustering in $O(nk^2)$. The $O(n^3)$ dominant term can make spectral clustering impractical for large applications. In practice, approximation techniques based on random sampling or kernel prepossessing have been proposed. An important insight is that the spectral clustering algorithm returns two clusters where a random walker is likely to be trapped, with minimal chance of transitioning from one cluster to the other. As a result, the algorithm tends to preform well when the graph consists of high-density clusters.

F. Unsupervised Classification to Spectral Clustering

One of the primary objectives of machine learning is to find a labeling, or classification, for a given data set $X_1, \dots, X_n$, in an unsupervised manner. In the following we define a mapping between classification and clustering.

Let a similarity graph be a graph where data points X_i and X_j have an edge between them if they are considered similar. There are several ways to define similarity, but we use a generic approach by considering each pair of datasets and calculate the distance using the Gaussian similarity function

$$s(X_i, X_j) = \frac{-||X_i - X_j||^2}{2\sigma^2} \tag{2}$$

where σ is a parameter tuned for the particular dataset. Given some threshold ϵ, two data points, X_i and X_j, are considered similar if and only if $s(X_i, X_j) > \epsilon$. An adjacency matrix is then defined as:

$$A_{ij} = \begin{cases} 1 & s(X_i, X_j) > \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The degree matrix is defined as:

$$D_{ii} = \sum_{j}^{n} A_{ij} \tag{4}$$

Finally, the normalized Laplacian matrix can be viewed as:

$$L = I - D^{-1/2} A D^{-1/2} \tag{5}$$

Note, by construction, the normalized Laplacian matrix is symmetric. Also, the eigenvectors of L are related to a particular classification of the original dataset.

IV. Quantum Spectral Graph Clustering

Algorithm 1 shows an overview of our proposed spectral clustering algorithm that consists of both classical and quantum computation. Line 2 constructs a mixed Laplacian using PageRank. Line 3 derives a unitary matrix by a matrix exponential. Line 4 constructs a matrix B, which will preform *amplitude encoding* to efficiently encode data to a quantum state, resulting in $|b\rangle$. For a vector $x \in \mathcal{R}^d$, the encoded quantum state is given by $|x\rangle = \frac{1}{||x||} \sum_{i=0}^{d-1} x_i |i\rangle$. This means the i-th component from the vector x becomes the amplitude for $|i\rangle$. To maintain normalization, the state is divided by the norm of x. Line 5-15 preform the quantum component of the spectral graph clustering.

Algorithm 1: Quantum Spectral Graph Clustering

```
   Input: $G(V, E)$, $k$
   Output: $k$-clusters
 1 Classical:
 2     Build mixed Laplacian matrix $L$
 3     Obtain $U = e^{iL}$
 4     Construct $B$ for amplitude encoding $|b\rangle$, a
         superposition of all eigenvectors from $L$
 5 Quantum:
 6     Compute eigenvalues by preforming QPE on $|b\rangle$
 7     Use an ancilla qubit$_1$ to separate eigenstates $\leq \mu$
 8     if Measured 0 from ancilla qubit$_1$ then
 9         Preform conditional rotation with new
             ancilla qubit$_2$
10         if Measured 0 from ancilla qubit$_2$ then
11             Preform Quantum K-means Clustering
12         else
13             Repeat Quantum
14     else
15         Repeat Quantum
```

Figure 3 shows the two major steps of the quantum component in circuit representation. First, the classically-derived gate B (constructed in line 4 of Algorithm 1) is applied to create the input state in a superposition of eigenvectors of L. Next, *quantum phase estimation* (QPE) is preformed, where U is defined as successive control operations of $\exp\{iL\}^{2^k}$ where k denotes which control qubit to utilize spanning from

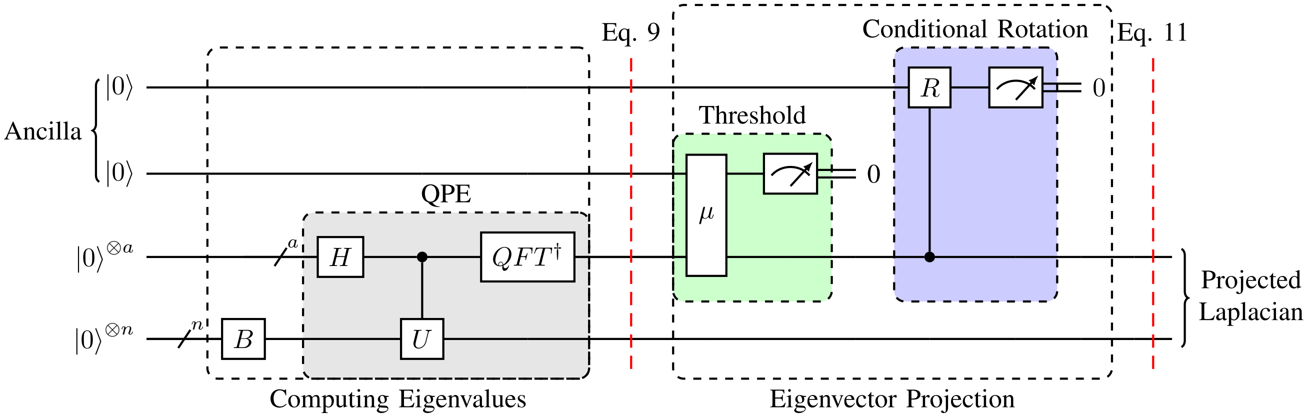


Fig. 3: Overview of the quantum algorithm in circuit representation. First, a classically-derived gate B is applied to create the input state in a superposition of eigenvectors of L. Next, QPE is preformed, where U is defined as successive control operations of $\exp\{iL\}^{2^k}$ where k denotes which control qubit to utilize spanning from 0 to $a-1$. Finally, thresholding and conditional rotation are applied to project the eigenvectors to a subspace with the smallest eigenvalues, hence producing a quantum state which represents a Laplacian L that has been projected to the subspace. a depends on the desired precision for eigenvalue estimates which follows from the analysis of QPE, and n is $\log_2$ of the number of nodes in the graph.

0 to $a-1$. Finally, *thresholding* and *conditional rotation* are applied to project the eigenvectors to a subspace with the smallest eigenvalues, hence producing a quantum state which represents a Laplacian L that has been projected to the subspace. The remainder of this section describes the two majors steps in Figure 3.

A. Computing Eigenvalues

Due to the mixed Laplacian matrix L being real and symmetric, it is therefore also a Hermitian matrix. Hence, a unitary matrix is derived by a matrix exponential:

$$U = e^{iL} \tag{6}$$

An important relationship between U and L is that they share the same eigenvectors $|\lambda_i\rangle$, and the original eigenvalues λ_i of L are encoded into the phase. Although we assume that U is given as part of the problem formulation, in practice computing the matrix exponential efficiently may not be possible. In the context of this problem, the general methods for computing the matrix exponential are coined as quantum Hamiltonian simulation. Various proposed methods target solving the matrix exponential using quantum algorithms (see for example [13]), however this topic remains largely as an open problem. Additionally, when taking the exponential of the matrix L, we have projected eigenvalues into a complex phase between 0 and 2π. This can cause some eigenvalues to be wrapped around 0 and 2π, hence removing the original ordering of eigenvalues, which will lead to incorrect thresholding of eigenvalues. A work-around is to rescale the original matrix, which will leave the overall properties unchanged, and will ensure the eigenvalues will cleanly map between 0 and 2π.

With a unitary that contains the important spectral properties of L, the *Quantum Phase Estimation* (QPE) algorithm [14] is used to create a superposition of eigenvectors with their amplitudes encoding the corresponding eigenvalues. Given a unitary matrix U with an eigen-pair relation $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$, QPE estimates the phase θ. The first step of QPE takes as input a superposition of eigenvectors with arbitrary normalized amplitudes and preforms following:

$$\sum_{j=0}^{2^n-1} \alpha_j |\lambda_j\rangle \mapsto \frac{1}{2^{\frac{n}{2}}} \sum_{k=0}^{2^n-1} \sum_{j=0}^{2^n-1} \alpha_j e^{2\pi i k\lambda_j} |\lambda_j\rangle \tag{7}$$

Of course, the eigenvectors are not known beforehand, hence there is no guarantee that input state will be a superposition of eigenvectors from L. A straightforward method is to conduct another experiment by generating random states, and investigating the output distribution of predicted eigenvalues using QPE.

The second step applies the inverse quantum Fourier transform to map the phase into the corresponding basis components, resulting in a state:

$$\frac{1}{2^n} \sum_{l=0}^{2^n-1} \sum_{j=0}^{2^n-1} \alpha_j \sum_{k=0}^{2^n-1} e^{\frac{2\pi i k}{2^n}(2^n\lambda_k - l)} |l\rangle |\lambda_j\rangle \tag{8}$$

For clarity, we rewrite the expression as:

$$\sum_{j=0}^{2^n-1} \alpha_j |\Lambda_j\rangle |\lambda_j\rangle + \text{other} \tag{9}$$

B. Eigenvector Projection

The next key step is to project onto the k lowest eigenvectors. To do this we first separate eigenvalues lower than a threshold μ using an extra ancilla qubit. Hence the lowest eigenvalues are flagged with $|0\rangle$:

$$\sum_{\substack{j \\ \Lambda_i \leq \mu}} \alpha_j |\Lambda_j\rangle |\lambda_j\rangle |0\rangle + \sum_{\substack{j \\ \Lambda_i > \mu}} \alpha_j |\Lambda_j\rangle |\lambda_j\rangle |1\rangle \tag{10}$$

Here we use an ancilla to effectively create a non-unitary operation on the state. This projection can also be implemented deterministically in a few ways as discussed in literature [15], [16]. Using these methods, a quantum circuit can be designed for comparison by modifying a ripple-carry adder circuit. This of course requires additional ancilla qubits to maintain reversibility of the quantum circuit.

If the flag qubit is $|0\rangle$, a conditional rotation is preformed. For a quantum state $|\psi\rangle$, *conditional rotation* makes use of an ancillary qubit to preform $|\psi\rangle|0\rangle \mapsto |\psi\rangle(\psi|0\rangle + \sqrt{1-\psi}|1\rangle)$. Hence, this requires another ancilla qubit to produce:

$$\sum_{\substack{j\\ \Lambda_i \leq \mu}} \alpha_j |\Lambda_j\rangle|\lambda_j\rangle|0\rangle\left(\Lambda_j|0\rangle + \sqrt{1-\Lambda_j}|1\rangle\right) + \sum_{\substack{j\\ \Lambda_i > \mu}} \alpha_j |\Lambda_j\rangle|\lambda_j\rangle|1\rangle \quad (11)$$

If a result of 1 is measured for either of the ancilla qubits, the result is discarded and recomputed. Otherwise, the probability of measuring $|00\rangle$ is proportional to

$$p(00) \propto \sum_{\substack{j\\ \Lambda_i \leq \mu}} \alpha_j^2 \Lambda_j \quad (12)$$

The resulting states contain eigenvectors that are bounded by $\mu \geq k$. We therefore have found the important contributors and disregarded weak contributors in the spectrum of L. Hence, clustering can now be preformed on the remaining superposition of eigenvectors. This can be preformed in two ways: (i) classically by first measuring and obtaining probabilistic predictions of important eigenvectors, or (ii) the computation can be carried on post-ancilla-measurement and the clusters can be estimated using quantum K-means clustering [17] on the projected state.

V. Experiments

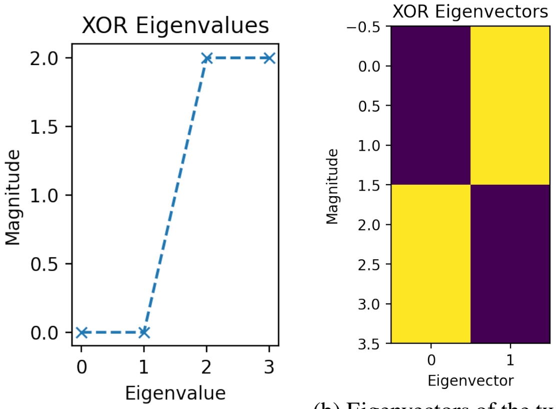
To demonstrate the effectiveness of our proposed quantum spectral graph clustering, we have used both real and synthetic benchmarks. While our approach is universally applicable (clustering any mixed graphs), for the real case, we focus on two fundamental machine learning problems that are small enough to be run on present-day quantum computers. The synthetic test is used to show the speed-up provided by the quantum algorithm.

A. Experimental Setup

We consider the following three types of datasets: exclusive-or, non-convex clusters and random graphs.

1) Exclusive OR: A classic problem typically targeted at multilayered neural networks is the exclusive OR (XOR). There are four points in the dataset: $(0, 1, 1), (1, 0, 1), (1, 1, -1), (0, 0, -1)$. Using a notion of similarity from Equation 2, we construct the symmetric Laplacian matrix as follows:

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \quad (13)$$



(a) Eigenvalue spectrum

(b) Eigenvectors of the two smallest eigenvalues

Fig. 4: The eigenvalues and eigenvectors for XOR Laplacian L in Equation 13.

The expected eigenvalues and the significant eigenvectors are shown in Figure 4.

2) Non-convex Clusters (NCC): Spectral clustering has been shown to be effective for non-convex clusters (NCC), when compared to naive unsupervised techniques. Consider two concentric circles as shown in Figure 5. A similarity graph is constructed which encodes the relationship between the inner and outer points of the circles.

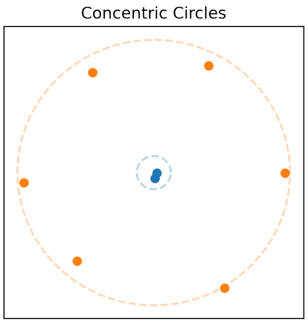


Fig. 5: Eight points used to generate concentric circles

3) Random Graphs: In order to investigate the running time of both the classical approach and the proposed quantum algorithm, we also created random directed graphs and observe the total number of operations. In this case, the quantum algorithm is numerically simulated on a classical computer. Although it would be interesting to preform the same experiments using a real quantum computer, such computers are not yet available. Simulation with large number of qubits is impractical, hence the random graphs are restricted in size to ensure simulation is feasible. Additionally, in order to mimic real quantum computers, a quantum compiler [18] is used to transform the quantum circuit to an equivalent circuit that is composed of gates taken from a universal gate set.

B. Results

We use the 5-qubit quantum computer, *ibmq_athens* [19], for XOR, and the 16-qubit quantum computer, *ibmq_16_melbourne* [20], for NCC provided by IBM Q. For the XOR problem, two qubits are used to represent eigenstates of the 4×4 Laplacian matrix, and another two qubits for QPE precision. The remaining qubit is used as an ancilla qubit for *conditional rotation*. To fit this problem on a 5-qubit computer, the initial state is prepared such that the thresholding step can be ignored. For NCC, three qubits are used for encoding the eigenstates, and four qubits for QPE precision. The circuits were then transpiled with maximum optimization for the respective devices.

A classical simulation results reveal 100% accuracy when observing the quantum state before measurement. This is expected as QPE will not preform any approximation to the eigenvalues since these problems are built with a simple Laplacian matrix. However, the accuracy on real devices are 37% and 13% for the XOR and NCC problems, respectively. The quantum devices have a respective quantum volume of 32 and 16 [21], which indicates the realistic size in qubit count and circuit depth that a quantum circuit can have before noise overpowers the results [22]. In both cases, the circuit depth exceeds the capabilities of the devices. This study shows promise of solving encoded graph-based problems by a quantum computer. Future work needs to incorporate variations of error-correction or control optimization to improve the overall fidelity of the quantum algorithms.

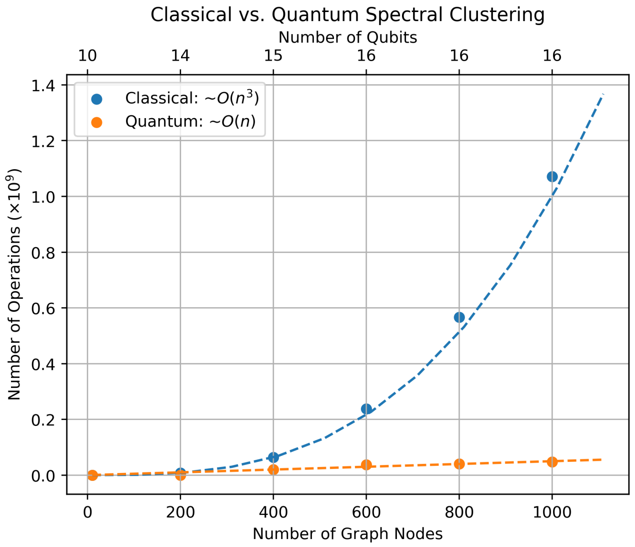


Fig. 6: Runtime complexity for quantum and classical spectral clustering preformed on random directed graphs.

Figure 6 compares the performance of our approach compared to classical spectral clustering using random graphs. In case of classical quantum clustering, the number of required operations significantly increases with the size of the graph. As expected, our approach leads to linear increase in the number of operations, since the number of qubits will remain constant for a range of n and consequently the number of operations in QPE will remain the same.

VI. Conclusion

While graph clustering has been widely studied, classical spectral algorithms suffer from a cubic growth in runtime. We proposed a quantum spectral clustering algorithm for discovering clusters in directed as well as mixed graphs. Experimental results using synthetic and simulated benchmarks demonstrated that our approach is scalable and can obtain beneficial clusters in linear time. In contrast, state-of-the-art classical algorithms are not useful in practice since they require cubic time complexity. Moreover, we have shown that near-future quantum computers can be used to encode machine learning problems based on spectral clustering. This work can be used as a stepping stone to study other graph-based methods in machine learning and optimization problems that rely on spectral analysis.

References

[1] S. Venegas-Andraca, "Quantum walks: A comprehensive review," *Quantum Information Processing*, vol. 11, no. 5, 2012.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[3] C. Aggarwal and H. Wang, "A survey of clustering algorithms for graph data," in *Managing and Mining Graph Data*, 275–301, 2010.

[4] L. Lovász *et al.*, "Random walks on graphs: A survey," *Combinatorics, Paul erdos is eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[5] J. Dodziuk, "Difference equations, isoperimetric inequality and transience of certain random walks," *Transactions of the American Mathematical Society*, vol. 284, no. 2, pp. 787–794, 1984.

[6] F. Chung, "Laplacians and the Cheeger inequality for directed graphs," *Annals of Combinatorics*, vol. 9, no. 1, pp. 1–19, 2005.

[7] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local Higher-Order Graph Clustering," *International Conference on Knowledge Discovery & Data Mining (KDD)*, vol. 2017, pp. 555–564, Aug. 2017.

[8] F. D. Malliaros and M. Vazirgiannis, "Clustering and Community Detection in Directed Networks: A Survey," *Physics Reports*, vol. 533, no. 4, pp. 95–142, Dec. 2013.

[9] S. X. Cui, M. H. Freedman, O. Sattath, R. Stong, and G. Minton, "Quantum Max-flow/Min-cut," *Journal of Mathematical Physics*, vol. 57, no. 6, p. 062206, Jun. 2016.

[10] A. Daskin, "Quantum Spectral Clustering through a Biased Phase Estimation Algorithm," *arXiv:1703.05568 [quant-ph]*, Jul. 2017.

[11] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, 1998.

[12] R. Andersen, F. Chung, and K. Lang, "Local Partitioning for Directed Graphs Using PageRank," p. 17.

[13] G. H. Low and I. L. Chuang, "Hamiltonian simulation by qubitization," *Quantum*, vol. 3, p. 163, 2019.

[14] A. Y. Kitaev, "Quantum measurements and the Abelian Stabilizer Problem," *arXiv:quant-ph/9511026*, Nov. 1995.

[15] S. Cuccaro *et al.* A new quantum ripple-carry addition circuit. [Online]. Available: http://arxiv.org/abs/quant-ph/0410184

[16] H. Thapliyal and N. Ranganathan, "Design of Efficient Reversible Logic Based Binary and BCD Adder Circuits," *ACM Journal on Emerging Technologies in Computing Systems, 9(3)*, pp. 1–31.

[17] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," *arXiv:1307.0411 [quant-ph]*, Nov. 2013.

[18] H. Abraham *et al.*, "Qiskit: An open-source framework for quantum computing," 2019.

[19] "IBM q 5 athens backend specification v1.2.2," 2020. [Online]. Available: https://quantum-computing.ibm.com

[20] "IBM q 16 melbourne backend specification v2.3.2," 2020. [Online]. Available: https://quantum-computing.ibm.com

[21] A. Cross *et al.*, "Validating quantum computers using randomized model circuits," *Physical Review A*, vol. 100, no. 3.

[22] D. Volya and P. Mishra, "Special session: Impact of noise on quantum algorithms in noisy intermediate-scale quantum systems," in *IEEE International Conference on Computer Design (ICCD)*, 2020.