Effectiveness of Adiabatic Quantum Computation on K-Means Clustering

Varin Sikand

The use of Quantum computation within the field of Machine Learning provides a new framework for learning techniques that can solve computational tasks classical supercomputers struggle with. Many of these tasks can be simulated using Quantum Circuits (QC), however these forms of quantum algorithms require multiple interactions using quantum gates and are therefore susceptible to noise and decoherence. Adiabatic Quantum Computing (AQC) instead describes both a problem Hamiltonian that encodes the given problem's solution in its ground state and an initial Hamiltonian with an arbitrary ground state. According to the Adiabatic Theorem, a slow changing Hamiltonian shifting from the initial to the problem Hamiltonian will keep the state $|\psi(t)\rangle$ in its ground state throughout the process, therefore ending in the solution state at time T. We also show how the rate of change in the Hamiltonian affects the final state's similarity to the solution state. This can be applied to a number of Machine Learning tasks, and we will analyze one such example defining an AQC method of simulating the classical k-means clustering algorithm with problem Hamiltonian $H_P = \sum_{c'j} |\vec{v}_j - \vec{v}_{c'}|^2 |c'\rangle\langle c'| \otimes |j\rangle\langle j|$ with j representing the label of vector v_j, c representing cluster assignments, and v_c representing the center of cluster c.[1] This Hamiltonian encodes the distance between each vector and the previously computed centroids using the coefficients $|\vec{v}_j - \vec{v}_{c'}|^2$, selecting the minimal distance assignments of each vector to their corresponding cluster centroid, with the ground state corresponding to the end result of the re-clustering step.

I. INTRODUCTION

Machine learning tasks in recent years handle extraordinarily large datasets on a regular basis regardless of the subject area. Examples can be seen with Language Models (LMs) such as BERT [2], T5 [3], or any of the GPT family of models [4–6]. In other domains, such as Graph Neural Networks (GNNs), issues of oversmoothing and subsequently limited model depth [7] plague the classical paradigm, placing a sort of ceiling on the performance of these Deep Neural Nets. In both the case of LLMs [8] and GNNs [9, 10], and indeed in the case of any model handling Big Data, the relatively new field of Quantum Machine Learning attracts the attention of many seeking to further optimize existing ML processes and even fit to previously intractable datasets due to the expressive potential of qubits and quantum computation. [11]

The intuition for this is clear - machine learning tasks manipulate data stored as vectors and tensor products in increasingly high-dimensional spaces due to the large number of features current models handle, and Quantum Computers are exceptionally good at manipulating these types of data both in terms of runtime and expressive capabilities. In fact, runtime analysis of quantum versions of ML algorithms already show massive improvements compared to their classical counterparts [12], with recent experiments also showing feasibility of several QML algorithms on currently existing quantum computers (albeit with restricted domains) [13–15]. One important distinction, however, is to acknowledge that there is more than one paradigm to conduct these QML algorithms. Quantum Circuits (QCs), defined by their usage of Quantum Gates to directly manipulate and apply unitary transformation on sets of qubits, offer one such setting to conduct Quantum Computing experiments. On the other hand, Adiabatic Quantum Computing [16] (AQC) offers solutions to a similar set of problems, but instead using the Adiabatic Theorem to reach a solution state through slow and careful procession of a time dependent Hamiltonian. The details of which is described in the Problem Statement section of this paper. Since the inception of both these paradigms, research has concluded that the algorithms themselves are polynomially equivalent [17, 18] - meaning that there exists a polynomial-time function that can map one problem to the other. Recent work even suggests an exact complexity equivalence between the two types of problems [19], a much stronger statement of equivalence.

The goal of this paper is to examine one such algorithm in-depth, choosing the adiabatic paradigm to conduct a K-Means clustering algorithm. We will calculate an explicit example of this algorithm, using simulated Hamiltonians and qudits to show the effects of the adiabatic algorithm along with necessary conditions for the algorithm to function correctly.

II. PROBLEM STATEMENT

A. K-Means Algorithm

Before delving into the quantum application of the K-Means clustering algorithm, we present a simplified view of the classical implementation for reference. K-Means is an algorithm designed for unsupervised classification, also known as "clustering", wherein a set of data represented by n-dimensional feature vectors (the process of obtaining these feature vectors is dependent on the specific task and such feature engineering will not be explored by this work) is partitioned iteratively - their classes determined by their distance to one of K cluster center (or "means")[20]. Beginning with initial seeds for cluster means, the process of selecting such seeds a separate topic of discussion discussed in the supplementary

material section of the Appendix, the K-Means clustering algorithm performs the following steps:

1. Assign each datapoint to one of k clusters according to distance from cluster center

2. Calculate new cluster centers using the mean of datapoints belonging to each newly updated cluster

These steps are repeated until convergence, which tends to happen quickly. Overall, the time for one classical step of K-Means is $O(kMN)$ where M denotes the number of data points and N denotes the dimension of each datapoint.

B. Adiabatic theorem

According to the Schrodinger equation, a quantum system evolves as follows:

$$i\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle \tag{1}$$

Where $|\psi(t)\rangle$ is a quantum state which varies with time, and $H(t)$ describes a time-dependent Hamiltonian being applied to this quantum state. Put succinctly, Equation 1 describes how an applied Hamiltonian will cause a quantum state to time-evolve. The adiabatic theorem tells us how this evolution proceeds when $H(t)$ is slowly varying. Namely that, when varying slowly enough, if $\psi(0)$ is an eigenstate of $H(0)$, then applying the slowly varying Hamiltonian will keep $\psi(t)$ in the same instantaneous eigenstate of $H(t)$. This is of interest due to the ability to set $\psi(0)$ to the ground state of a starting Hamiltonian and return the ground state of the slowly transformed Hamiltonian at the end. In general, we control the rate at which the Hamiltonian transforms by defining $\tilde{H}(s) = H(t/T)$ resulting in energy levels $E_l(s)$. Therefore, the adiabatic theorem explicitly states that if the gap between the two lowest levels is strictly greater for all $0 \leq s \leq 1$ and the initial state is set to the ground state: $|\psi(0)\rangle = |l = 0; s = 0\rangle$ then

$$\lim_{T\to\infty} |\langle l = 0; s = 1 \,|\, \psi(T)\rangle| = 1$$

and the final state is equivalent to the ground state of the final Hamiltonian. Let this minimum gap be defined by

$$g_{min} = \min_{0\leq s\leq 1} (E_1(s) - E_0(s))$$

. The value of T recommended for successful adiabatic transformation is $T \gg \frac{\epsilon}{g_{min}^2}$ where ϵ is a value of order of a typical eigenvalue of H and isn't too big. This means that the size of T is governed by g_{min}^{-2}

C. Problem & Initial Hamiltonian

For any optimization problem that aims to be solved using Adiabatic transformations, the motivation lies in the ground state of the problem's Hamiltonian. The general methodology is to encode the problem using a Hamiltonian such that its solution can be found within its ground state. This can be seen using the problem hamiltonian of the K-means clustering algorithm:

$$H_P = \sum_{c'\,j} |\vec{v}_j - \vec{v}_{c'}|^2 |c'\rangle\langle c'| \otimes |j\rangle\langle j|$$

with j representing the label of vector v_j, c representing cluster assignments, and v_c representing the center of cluster c. It's clear to see how this Hamiltonian encodes the distance from each point to its assigned cluster within its phase factor. Therefore, energy is minimized when these distances are minimized which defines the ground state of the problem hamiltonian as the solution to the clustering iteration. To be specific, the solution of a given iteration is

$$|\chi\rangle = 1/\sqrt{M} \sum_{c,j\in c} |c\rangle|j\rangle$$

with M being the number of vectors to cluster. This is a uniform superposition of all the vectors, with each assigned to its appropriate cluster can can be sampled to retrieve information about which states belong to which clusters.

Now that we know that our solution for a given iteration is the ground state of H_p, the issue remains in how we go about finding this state. This proves quite difficult actually, since directly constructing it would require knowledge of the solution in the first place, however we know the structure of the problem Hamiltonian and that its solution exists as its ground state. Pairing this with the Adiabatic theorem, it becomes clear that we must first construct an *Initial Hamiltonian* whose ground state is evident and easy to construct, and then slowly transform this initial Hamiltonian into the problem Hamiltonian. Formally, we define the overall Hamiltonian H as a combination of our initial Hamiltonian H_0 and our problem Hamiltonian as follows:

$$H_0 = \mathbb{1}_{k+M} - \frac{1}{k}\sum_{c,c'} |c\rangle\langle c'| \otimes \mathbb{1}_M \tag{2}$$

$$H(t) = (1 - t/T)H_0 + (t/T)H_P \tag{3}$$

where T controls the rate at which the Hamiltonian varies. In particular, an appropriate value of T is governed by The ground state of $H(0)$ is simply

$$|\psi_i\rangle = \frac{1}{\sqrt{Mk}} \sum_{c',j} |c'\rangle|j\rangle|\psi_1\rangle^{\otimes d}$$

where $|\psi_1\rangle$ corresponds to the solution for the previous iteration of K-means, and d is the number of copies that

can be made during the iteration. These copies are made to increase accuracy for the distance calculation. Showing that $|\psi_i\rangle$ is an eigenstate of H_0:

$$H_0|\psi_i\rangle = |\psi_i\rangle - \frac{1}{k}\left(\sum_c |c\rangle\right)\left(\sum_{c'}\langle c'|\right) \cdot \left(\frac{1}{\sqrt{Mk}}\sum_{c'}|c'\rangle\right)\sum_j |j\rangle|\psi_1\rangle$$

$$= |\psi_i\rangle - \frac{1}{\sqrt{Mk}}\sum_c |c\rangle\sum_j |j\rangle|\psi_1\rangle = |\psi_i\rangle - |\psi_i\rangle$$

Therefore the eigenvalue is 0 and $|\psi_i\rangle$ corresponds to the ground state of H_0. For the initial cluster, choose k vectors as initial seeds, denoted as $|i_c\rangle$, and instead use the initial state

$$|\psi_i\rangle = \frac{1}{\sqrt{Mk}}\sum_{c',j}|c'\rangle|j\rangle(\frac{1}{\sqrt{k}}\sum_c |c\rangle|i_c\rangle)$$

At the end of each iteration, the final state is thus $|\chi\rangle|\psi_1\rangle$ with the solution separable from the previous iteration's solution. Additionally, because the adiabatic part of the algorithm only acts on the c' cluster labels and the overlap squared between the initial and final state of each step is $O(1/k)$, the time per step required is no greater than $O(k \log kMN)$. This could be as small as $O(\log kMN)$ if the minimum gap g_{min} is of constant order. This is an enormous speedup compared to the classical K means algorithm, which has an asymptotic runtime of $O(kMN)$. Original derivations can be found in Lloyd *et al.* [1]

III. EXPERIMENT SETUP

Our experiment solves the K-Means clustering algorithm by simulating the quantum K-Means algorithm described above for 4 datapoints (see Figure 1), solving for the time-dependent state numerically. We obtain results for values of T ranging from 1 to 1,000 and report both the average energy at the final time $\langle\psi(T)|H(T)|\psi(T)\rangle$ along with the clusters reported by the algorithm. Usually, the value of T is chosen using the minimum gap, however the problem of solving for minimum gap in this algorithm is beyond the scope of this paper, therefore we choose a value of T large enough for the final energy to converge to its minimum. The first statistic shows how increasing the value of T continuously minimizes the average energy, displaying how the adiabatic theorem requires a large value of T to keep ψ in the ground state. The second statistic is gathered by randomly sampling the final state 1,000 times in the computational basis according to the probabilities given by the state. These samples are in the form $|c\rangle|j\rangle$, where the first $\log k$ bits correspond to the cluster assignment of vector j, obtained using the last $\log M$ bits.

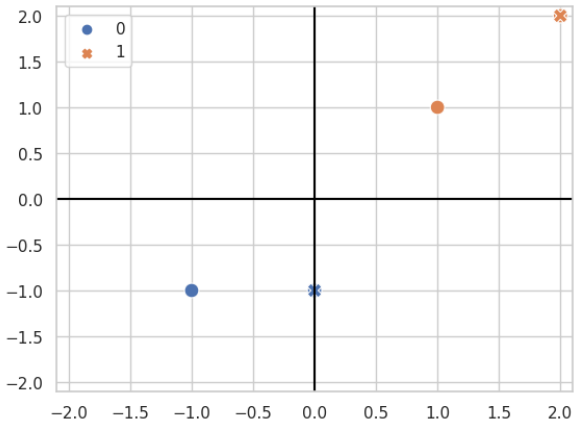

FIG. 1. 4 datapoints for clustering. Initial seeds represented by x marks and optimal first-iteration clustering shown by color

The above figure depicts the optimal assignment of clustering after the first iteration, where we randomly select two points to be initial seeds. Note that because the solution ground state must be a superposition of 4 states $|c\rangle|j\rangle$, along with the fact that not all of the vectors have a distance 0 from the cluster center, the ground state for the problem Hamiltonian will not have an energy of 0. Instead, the energy will converge to an average of the energies of the 4 lowest energy eigenstates of H_p such that no vector has two separate cluster assignments.

Eigenstate	E
000	**0.0**
001	5.0
010	13.0
011	**1.0**
100	13.0
101	**2.0**
110	**0.0**
111	18.0

TABLE I. Eigen-energy per Eigenstate

The energies corresponding to each eigenstate can be seen in Table I, and the eigenstates contained within the ground state assignment are bolded. According to these eigen-energies, the final energy of the correct ground state should equal $\frac{3}{4}$, taking the average of the 4 eigen-energies corresponding to the ground state. It can be observed that these eigenstates can be picked out using just Table I using the following method: For each datapoint j, compare the energies for $0 \otimes j$ and $1 \otimes j$, selecting the cluster assignment with the minimum energy. For example, for datapoint $j = 01$, the energy for $0 \otimes 01 = 001$ is 5, whereas the energy for $1 \otimes 01 = 101$ is 2, therefore datapoint 01 is assigned to cluster 1.

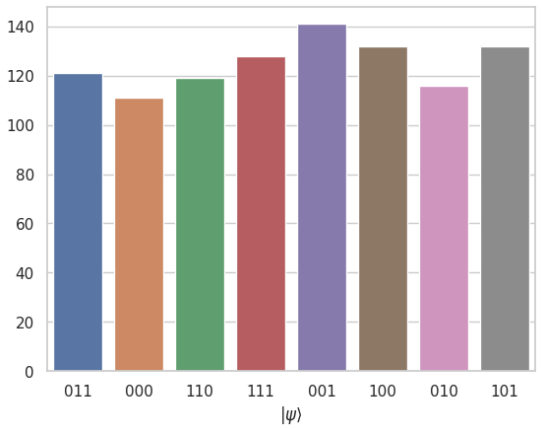

FIG. 2. Final sampled states with $T = 10$

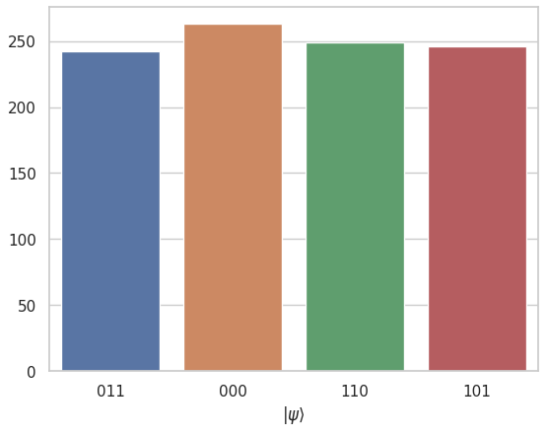

FIG. 3. Final sampled states with $T = 1,000$

IV. RESULTS

When given a sufficient value of T, the quantum K-Means clustering algorithm can find the correct solution for cluster assignment, as shown by Figure 3. However, the results also show the effects of an insufficient value of T, where $|\psi(T)\rangle$ is not close to the ground state of H_p. Instead, as shown by Figure 2, the final state is much closer to a uniform superposition of the computational basis in this case.

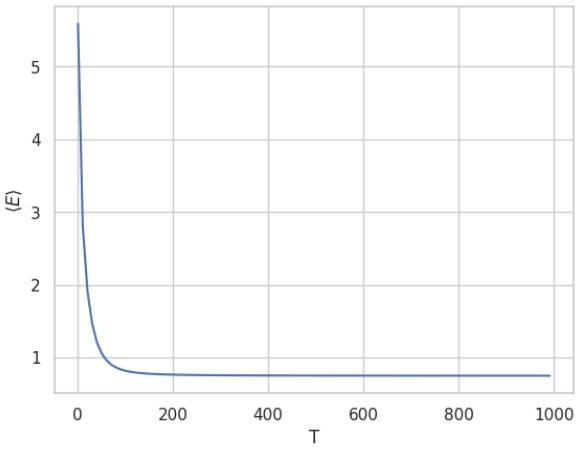

FIG. 4. Average energy of $|\psi(T)\rangle$ measured on H_p

As can be seen by Figure 4, the constraints of the Adiabatic Theorem are consistent with experimental results. and in fact the maximum value of T chosen is quite a bit larger than needed for convergence. Small values of T, indicating a fast-changing process, result in a final state energy much higher than expected. Additionally, the predicted energy of $\frac{3}{4}$ from the experimental setup is also consistent with experimental results as $\langle E\rangle$ approaches $\frac{3}{4}$ as $T \to \infty$ within the bounds of the experiment.

V. CONCLUSIONS

We have shown that for the quantum K-Means clustering algorithm the cluster reassignment step can be adiabatically solved, with a T value ≈ 80 being sufficient to obtain a final state approximately equivalent to the ground state of H_p. We have also presented the effects of varying the T value when adiabatically transforming the time-dependent Hamiltonian, showing the results of choosing an insufficient value of T. We have shown that the process for simulating a quantum adiabatic algorithm requires only the selection of a suitable problem Hamiltonian and a sufficient value of T (which can be obtained from an analysis of the minimum energy gap).

Therein lies the benefit of using quantum adiabatic algorithms for optimization and ML tasks. Using the results of the K-Means clustering experiment, it can be seen that such a problem Hamiltonian can be created intuitively based on the optimization task, and thus a quantum speedup can be realized as long as there is a limited minimum energy gap.

One drawback of this specific algorithm, however, is the lack of a suitable quantum computer to set up such a problem Hamiltonian. As of now, most adiabatic Quantum Computers (such as D-Wave [21]) are equipped to specifically find low energy states of Ising models [22]. This poses a problem of translating the K-Means clustering problem to that of finding minimum energy states of Ising models. Other works explore this, usually with some sort of restriction of the K-Means clustering problem to assist in the mapping to Ising models [23, 24]. This provides a future direction for the field of quantum adiabatic methods for Machine Learning algorithms.

[1] S. Lloyd, M. Mohseni, and P. Rebentrost, arXiv preprint arXiv:1307.0411 (2013).

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, edited by J. Burstein, C. Doran, and T. Solorio (Association for Computational Linguistics, Minneapolis, Minnesota, 2019) pp. 4171–4186.

[3] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, Journal of Machine Learning Research **21**, 1 (2020).

[4] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, OpenAI blog **1**, 9 (2019).

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, Advances in neural information processing systems **33**, 1877 (2020).

[6] OpenAI, "Gpt-4 technical report," (2023), arXiv:2303.08774 [cs.CL].

[7] D. Chen, Y. Lin, W. Li, P. Li, J. Zhou, and X. Sun, in *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34 (2020) pp. 3438–3445.

[8] Y. Chen, Y. Pan, and D. Dong, IEEE Transactions on Cybernetics (2021).

[9] G. Verdon, T. McCourt, E. Luzhnica, V. Singh, S. Leichenauer, and J. Hidary, arXiv preprint arXiv:1909.12264 (2019).

[10] J. Choi, S. Oh, and J. Kim, in *2021 International Conference on Information Networking (ICOIN)* (2021) pp. 46–49.

[11] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. Mcclean, Nature Communications **12** (2021), 10.1038/s41467-021-22539-9.

[12] M. Schuld, I. Sinayskiy, and F. Petruccione, Contemporary Physics **56**, 172 (2015).

[13] C. Tüysüz, F. Carminati, B. Demirköz, D. Dobos, F. Fracas, K. Novotny, K. Potamianos, S. Vallecorsa, and J.-R. Vlimant, arXiv preprint arXiv:2007.06868 (2020).

[14] S. DiAdamo, C. O'Meara, G. Cortiana, and J. Bernabé-Moreno, IEEE Transactions on Quantum Engineering **3**, 1 (2022).

[15] J. Li, P. Zhang, D. Song, and Y. Hou, Physica A: Statistical Mechanics and its Applications **456**, 51 (2016).

[16] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, arXiv preprint quant-ph/0001106 (2000).

[17] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, in *45th Annual IEEE Symposium on Foundations of Computer Science* (2004) pp. 42–51.

[18] A. Mizel, D. A. Lidar, and M. Mitchell, Physical review letters **99**, 070502 (2007).

[19] H. Yu, Y. Huang, and B. Wu, Chinese Physics Letters **35**, 110303 (2018).

[20] S. Lloyd, IEEE transactions on information theory **28**, 129 (1982).

[21] D-Wave press release: D-Wave announces D-Wave 2000Q quantum computer and first system order (2017).

[22] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, *et al.*, Nature **473**, 194 (2011).

[23] C. Bauckhage, C. Ojeda, R. Sifa, and S. Wrobel.

[24] D. Arthur and P. Date, Quantum Information Processing **20**, 1 (2021).