Adiabatic Quantum Kitchen Sinks for Learning Kernels Using Randomized Features

Moslem Noori
1QBit Information Technologies
Vancouver, BC, Canada
moslem.noori@1qbit.com

Seyed Shakib Vedaie*
1QB Information Technologies
Vancouver, BC, Canada
shakib.vedaie@1qbit.com

Inderpreet Singh
1QB Information Technologies
Vancouver, BC, Canada
inderpreet.singh@1qbit.com

Daniel Crawford
1QB Information Technologies
Vancouver, BC, Canada
daniel.crawford@1qbit.com

Jaspreet S. Oberoi†
1QB Information Technologies
Vancouver, BC, Canada
jaspreet.oberoi@1qbit.com

Barry C. Sanders
Institute for Quantum Science and Technology
University of Calgary
Calgary, AB, Canada
sandersb@ucalgary.ca

Ehsan Zahedinejad
1QB Information Technologies
Vancouver, BC, Canada
ehsan.zahedinejad@1qbit.com

Abstract

Hybrid quantum–classical approaches provide an immediate platform for exploring a possible quantum advantage for machine learning applications. One example of such a hybrid approach is "quantum kitchen sinks", which builds upon the classical algorithm known as "random kitchen sinks" to leverage a gate model quantum computer for machine learning applications. We propose an alternative approach called "adiabatic quantum kitchen sinks", which employs an adiabatic quantum device to transform data features into new features in a non-linear manner, which can then be used by classical machine learning algorithms. We present the effectiveness of our approach in performing binary classification on both synthetic and real-world datasets. In terms of classification accuracy, our approach significantly outperforms classical linear algorithms on the studied binary classification tasks and can potentially be implemented on a current adiabatic quantum device to solve practical problems.

1 Introduction

Quantum algorithms [1] are theoretically proven to solve certain computational problems faster than the best known classical algorithms [2, 3]. Despite the impressive progress that has been made toward

*Institute for Quantum Science and Technology, University of Calgary, Calgary, AB, Canada
†School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

building a universal quantum computer in the quest for quantum supremacy [4, 5], it remains an elusive goal due to the negative effects of noise present in quantum systems. Meanwhile, noisy, intermediate-scale quantum (NISQ) [6] devices readily provide a platform for demonstrating a potential quantum advantage for specific applications such as machine learning [7].

The rapid growth in the amount of available data requires increasingly faster computing devices to learn from big data, and quantum computers are a potential candidate. Several recent studies have shown the potential of NISQ technologies in machine learning [8–12]. Recently, an algorithm called "quantum kitchen sinks" (QKS) was proposed [13], which builds upon the idea of classical "random kitchen sinks" (RKS) [14–16]. The idea behind RKS is to map the input data samples into a randomized feature space, such that the overlap (the inner product) of the pair of data samples in the randomized space approximates a desired kernel [17] (a similarity measure between a pair of data samples). A linear machine learning algorithm can then act on the randomized samples generated from the original input data to execute the learning process. Despite its simplicity, the performance of RKS is comparable to state-of-the-art machine learning algorithms [18, 19].

We propose an alternative approach to QKS [14–16] we call "adiabatic quantum kitchen sinks" (AQKS). Our algorithm uses an adiabatic quantum device as an explicit feature map to transform the features of each data sample into new features called quantum randomized features. Given a data sample $\boldsymbol{x}$, we encode its input data features into the parameters of a quantum Hamiltonian. Evolving the quantum system and performing a measurement at the end of the evolution gives us a new data sample that represents $\boldsymbol{x}$ in the feature space $\mathcal{H}$. The kernel that results from such a transformation is non-linear due to the effect of the measurement process on the quantum system. Our simulations show that such a non-linear explicit feature map has a positive impact on learning kernel machines for classification problems.

We consider the following scenario to show the effectiveness of AQKS on a learning task. Given a dataset $\mathcal{D} \subset \mathbb{R}^p$, we first use AQKS to construct a new dataset $\mathcal{D}' \subset \mathcal{H}$. We then train two separate support vector machines with a linear kernel (LSVM), one on the original dataset $\mathcal{D}$ and the other on the transformed dataset $\mathcal{D}'$. We call the first model that is trained on $\mathcal{D}$ an LSVM, and the second trained on $\mathcal{D}'$ an AQKS+LSVM model. Keeping the learning algorithm in both models the same (i.e., an LSVM), we compare the performance (i.e., the classification accuracy) of the two models on two example datasets.

To demonstrate the power of AQKS for machine learning, we evaluate the performance of our approach on a synthetic dataset as well as on the Modified National Institute of Standards and Technology (MNIST) dataset. Our simulations show that AQKS+LSVM significantly outperforms (in terms of the classification accuracy) the LSVM for classification tasks on the studied datasets. Specifically, our approach increases the classification accuracy on the synthetic dataset from 50% to 99.4%. On the MNIST dataset, our method reduces the classification error from 4.4% to 1.6%. It is important to mention that the AQKS algorithm can readily be applied to practical datasets with any number of features using a current quantum annealer.

2 Adiabatic Quantum Kitchen Sinks

To provide a more formal description of the AQKS algorithm, let us consider dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$, where each of the $\mathcal{X}$ (the input data) and $\mathcal{Y}$ (the target data) is a set comprising n data samples. We denote each element of $\mathcal{X}$ by $\boldsymbol{x}_i$ ($i \in \{1, 2, \ldots, n\}$), which is a p-dimensional vector defined over $\mathbb{R}^p$. The first step in transforming the given dataset from the original space (i.e., $\mathbb{R}^p$) to a new space (i.e., $\mathcal{H}$) is "encoding". To this end, we define $\mathbf{A}$ and $\boldsymbol{b}$, where $\mathbf{A}$ is a $q \times p$ random matrix with a classical probability distribution function $P(\mathbf{A})$ and $\boldsymbol{b}$ is a q-dimensional vector with a classical probability distribution function $P(\boldsymbol{b})$. Here, q refers to the number of qubits. We encode $\boldsymbol{x}_i$ into $\boldsymbol{j}_i$ by applying the linear transformation

$$\boldsymbol{j}_i = \mathbf{A}\boldsymbol{x}_i + \boldsymbol{b}. \tag{1}$$

The process of encoding $\boldsymbol{x}_i$ with different realizations of $\mathbf{A}$ and $\boldsymbol{b}$ is repeated multiple times. Each repetition is called an "episode" and e is used to denote the episode number. The corresponding encoded q-dimensional vector for an episode e is represented by $\boldsymbol{j}_i^e$, which is then mapped onto the

coefficients of the local σ^z terms of a q-body transverse-field Hamiltonian, $\mathbf{H}(t)^e_{\boldsymbol{x_i}}$, such that

$$\mathbf{H}(t)^e_{\boldsymbol{x}_i} = a(t)\sum_{v}^{q}\sigma_v^x + \sum_{\langle l,m\rangle} h_{lm}\sigma_l^z\sigma_m^z + \sum_{u}^{q} j_u^e\sigma_u^z. \tag{2}$$

In (2), j_u^e is the u-th element of the vector $\boldsymbol{j}_i^e$ and σ^α ($\alpha \in \{x, y, z\}$) denotes a Pauli operator. Each h_{lm} is a real number derived from a function of j_u^e (in our experiments, we consider $h_{l,m}^e = j_l^e j_m^e$) that could result in quantum entanglement, being the coefficient for the $\sigma_z\sigma_z$ interaction.

In order to generate a transformed data sample from each $\boldsymbol{x}_i$, we evolve (2) on an adiabatic quantum device from an initial time $t_\text{i} = 0$ to a final time $t_\text{f} = T$ and then perform a projective measurement along the z-axis at the end of the evolution. Stacking the outcomes of the measurements generated through a total of E episodes for $\boldsymbol{x}_i$ and normalizing the resultant vector by $\frac{1}{E}$ provides a $(q \times E)$-dimensional vector $\boldsymbol{u}_{\boldsymbol{x}_i}$, which represents the respective $\boldsymbol{x}_i$ in the feature space. We then train a linear machine learning model using this transformed dataset.

3 Simulation Setting

Here, we discuss the simulation settings used to evaluate the improved classification performance achieved using our algorithm, AQKS. Note that, throughout our experiments, we Trotterize [20] in order to simulate the quantum system's evolution using a classical computing device.

3.1 Performance Measure

To investigate the performance of AQKS, we first create a baseline for the underlying learning task by solving the classification problem using an LSVM without quantum randomization (i.e., with the data residing in its original space). We then compare the performance of LSVM applied on the transformed dataset, called AQKS+LSVM, against the mentioned baseline. This provides a systematic way to fairly assess the power of an adiabatic quantum device as a non-linear feature map.

3.2 Datasets

We evaluate the performance of AQKS+LSVM on two datasets. The first is a two-dimensional synthetic dataset, consisting of two classes, and generated using the `sklearn.datasets` Python module (see Fig. 1a). This dataset is linearly inseparable in the two-dimensional space, and thus is a good candidate for studying non-linear effect of quantum randomization. The second dataset is the MNIST dataset, a practical dataset widely used for testing and benchmarking machine learning

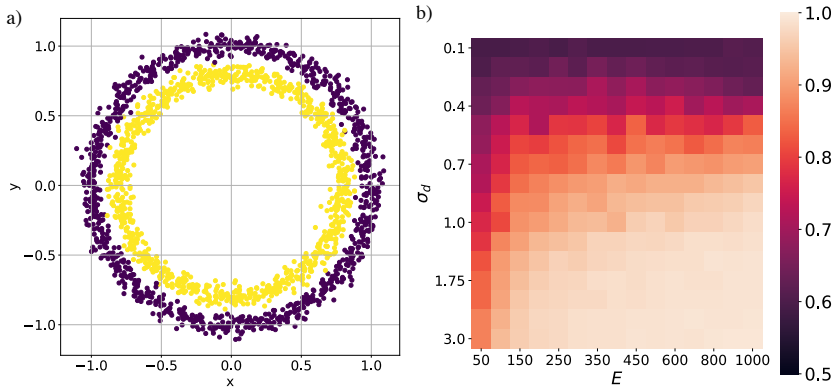

Figure 1: a) Representation of the "circles" dataset consisted of two classes of yellow and purple circles which consist of 1000 data samples in total. b) Results of the classification accuracy of AQKS+LSVM with a two-qubit quantum Hamiltonian for the synthetic circles dataset. Here, σ_d and E refer to the variance of the normal distribution and the total number of episodes for each setting, respectively. The highest average accuracy that AQKS attains over 10 trials for each setting is 99.4%.

algorithms. The dataset contains a large ensemble of handwritten digits, where each data sample is a 28-by-28-pixel greyscale image. Each image can be represented by a 784-dimensional vector $\boldsymbol{x}$ whose elements represent the shade, in grey, of the pixels, and ranges from 0 to 255. We evaluate the performance of the AQKS+LSVM method in classifying the handwritten digits "3" and "5". The dataset contains 7141 and 6313 instances of the digits 3 and 5, respectively.

4 Results and Discussion

Here, we report the results of the classification performance of AQKS+LSVM on both the synthetic dataset and the MNIST dataset.

4.1 The Synthetic Dataset

For the synthetic circles dataset (see Fig. 1a), Fig. 1b represents the results of the classification performance of AQKS+LSVM using a two-qubit quantum system. The annealing time and duration of the Trotterization time slots are $T = 5$ and $\tau = 1$, respectively. Elements of $\boldsymbol{b}$ are chosen randomly according to a uniform distribution over $[0, 2\pi)$. We use a zero-mean Gaussian distribution with a standard deviation of σ_d for generating the elements of $\mathbf{A}$.

We have deliberately selected a pattern where the two classes of data samples are not linearly separable. Specifically, for the two classes of concentric circles in Fig. 1a, the performance of a linear classifier in two dimensions (i.e., a straight line) will not exceed 50%.

For the best set of E and σ_d parameters, transforming the input features into a randomized feature space using an adiabatic quantum device improves the performance considerably, and AQKS+LSVM achieves an accuracy of 99.4% averaged over 10 classification trials. This provides a clear indication that the corresponding quantum explicit feature map has plausible non-linear properties. As both the encoding (1) and the learning algorithm (LSVM) are linear, it becomes apparent that the observed non-linearity of the quantum kernel is caused by the quantum feature map (i.e., the operation we perform using the adiabatic quantum device). As shown in Fig. 1b, the accuracy improves as the number of episodes is increased. In addition, we see that the choice of σ_d plays a significant role in the performance of the classification.

4.2 The MNIST Dataset

For the MNIST dataset classification, we first perform a hyperparameter tuning on the standard deviation σ_d of the zero-mean Gaussian probability distribution $\mathbf{A}$. This is done by fixing the number of qubits to two, the number of episodes to 10,000, and $\boldsymbol{b}$ to be a zero vector. We run the classification exercise for 3000 images out of the total 13,454 data samples, each time using a different value for σ_d. Using 75% of the chosen 3000 images for training and the rest for testing, a value of $\sigma_d = 0.01$ yields the best performance. Similarly, a value of $\sigma_d = 0.01$ gives the best result for a four-qubit quantum system. After finding the optimal value for σ, we re-run AQKS+LSVM on all 13,454 samples of data, with 75% used for training and the rest for testing. The results reported below are for $E = 20,000$ and 10 classification trials. Full connectivity between the qubits is assumed, meaning that each qubit interacts with all other qubits in the system. Using SVMs with linear and RBF kernels, the accuracy of the models trained on this dataset is 95.6% and 99.0%, respectively. Compared to

		Two-qubit system	Four-qubit system
Method	LSVM	AQKS+LSVM	AQKS+LSVM
μ_c	0.951	0.977	0.984
σ_c	0.002	0.002	0.002

Table 1: Classification accuracy of an LSVM on the randomized features generated by AQKS using a two-qubit and a four-qubit quantum Hamiltonian. The values in the LSVM column show the performance of the LSVM over the original dataset without any randomization on the input features. Here, μ_c (the mean) and σ_c (the standard deviation) are taken over 10 trials, for different numbers of qubits.

the LSVM, the four-qubit AQKS+LSVM method results in greater accuracy: 98.4%. As shown in Table 1, quantum randomization improves the classification accuracy of the LSVM. The accuracy improves further as the number of qubits is increased. For instance, increasing the number of qubits from two to four reduces the classification error from 2.3% to 1.6%. This is another indication that the quantum kernel provides a non-linear property that boosts the performance of the classifier over the performance attained using a linear kernel.

One could still argue that AQKS+LSVM does not outperform or or perform as well as an SVM algorithm with a non-linear kernel (e.g., RBF). We wish to point out that performing extensive hyperparameter tuning for the parameters of our proposed method requires access to a quantum annealer. More specifically, whereas the simulation of quantum systems that have a few dozen qubits is feasible on a classical computing device, a total of E such simulations would be required for each data sample, making our approach computationally expensive to simulate classically.

5 Conclusion

In this work, we have introduced a hybrid quantum–classical machine learning algorithm that employs an adiabatic quantum device (a quantum annealer) as an explicit feature map to generate randomized features out of input data features. Our approach, called "adiabatic quantum kitchen sinks", combined with an LVSM, outperforms a linear learning method for both the synthetic dataset and the MNIST dataset. Even employing the limited-in-size quantum annealers of today [21], our approach can be applied to practical datasets.

In terms of future research, it is worth pointing out that throughout the experiments we performed in our study, we used the same probability distribution functions to generate all the elements of $\mathbf{A}$ and $\boldsymbol{b}$. We expect that using different types of probability distribution functions for each individual qubit can introduce more-complex forms of non-linearity into the quantum kernel [22, 23].

One of the advantage of AQKS is that, unlike [8, 24, 25], it does not require constructing a quantum system multiple times in a loop with a classical device. We can, however, modify AQKS to turn it into an adaptive algorithm where we update the parameters of AQKS with respect to the performance of the model, in an iterative fashion. To do so, conside (2), and assume that we encode data into the local field parameters (j) and that each interacting term (h^i_{mn}) represents the adaptive parameters which we intend to update iteratively. Then, for a given machine learning task on a given dataset, we use AQKS to train a model with a generalization error of F^i. Having access to F^i and each h^i_{mn}, we use a gradient-free optimization algorithm [26] to update each h^i_{mn} (i.e., each adaptive parameter) while reducing the error of the classification accuracy of the model. This iterative process continues until an F^i with a desired threshold has been met or the maximum number of iterations has been reached.

6 Acknowledgements

Barry C. Sanders acknowledges NSERC support. Partial funding for this work was provided by the Mitacs Accelerate program. We thank Marko Bucyk for reviewing and editing the manuscript.

References

[1] A. Montanaro, "Quantum algorithms: an overview," *npj Quantum Inf.*, vol. 2, p. 15023, 2016.

[2] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997.

[3] L. K. Grover, "A fast quantum mechanical algorithm for database search," *arXiv preprint quant-ph/9605043*, 1996.

[4] A. W. Harrow and A. Montanaro, "Quantum computational supremacy," *Nature*, vol. 549, p. 203, 2017.

[5] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. L. Heras, R. Babbush, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, E. Solano, H. Neven, and J. M. Martinis, "Digitized adiabatic quantum computing with a superconducting circuit," *Nature*, vol. 534, p. 222, 2016.

[6] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[7] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, p. 195, 2017.

[8] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.

[9] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, "Quantum Boltzmann machine," *Phys. Rev. X*, vol. 8, p. 021050, 2018.

[10] D. Crawford, A. Levit, N. Ghadermarzy, J. S. Oberoi, and P. Ronagh, "Reinforcement learning using quantum Boltzmann machines," *Quantum Inf. Comput.*, vol. 18, pp. 51–74, 2018.

[11] A. Levit, D. Crawford, N. Ghadermarzy, J. S. Oberoi, E. Zahedinejad, and P. Ronagh, "Free energy-based reinforcement learning using a quantum processor," *arXiv preprint arXiv:1706.00074*, 2017.

[12] J. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, B. Bloom, S. Caldwell, N. Didier, E. S. Fried, S. Hong, *et al.*, "Unsupervised machine learning on a hybrid quantum computer," *arXiv preprint arXiv:1712.05771*, 2017.

[13] C. Wilson, J. Otterbach, N. Tezak, R. Smith, G. Crooks, and M. da Silva, "Quantum kitchen sinks: An algorithm for machine learning on near-term quantum computers," *arXiv preprint arXiv:1806.08321*, 2018.

[14] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems 20* (J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, eds.), pp. 1177–1184, Curran Associates, Inc., 2008.

[15] A. Rahimi and B. Recht, "Uniform approximation of functions with random bases," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, pp. 555–561, Sep. 2008.

[16] A. Rahimi and B. Recht, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," in *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 1313–1320, Curran Associates, Inc., 2009.

[17] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *Ann. Stat.*, vol. 36, no. 3, pp. 1171–1220, 2008.

[18] A. Rahimi and B. Recht, "Reflections on random kitchen sinks," 2017.

[19] A. May, A. B. Garakani, Z. Lu, D. Guo, K. Liu, A. Bellet, L. Fan, M. Collins, D. Hsu, B. Kingsbury, *et al.*, "Kernel approximation methods for speech recognition.," *J. Mach. Learn. Res.*, vol. 20, no. 59, pp. 1–36, 2019.

[20] L. A. Wu, M. S. Byrd, and D. A. Lidar, "Polynomial-time simulation of pairing models on a quantum computer," *Phys. Rev. Lett.*, vol. 89, p. 057904, 2002.

[21] A. D. King, J. Carrasquilla, J. Raymond, I. Ozfidan, E. Andriyash, A. Berkley, M. Reis, T. Lanting, R. Harris, F. Altomare, *et al.*, "Observation of topological phenomena in a programmable lattice of 1,800 qubits," *Nature*, vol. 560, no. 7719, p. 456, 2018.

[22] C. Chesneau, H. S. Bakouch, and T. Hussain, "A new class of probability distributions via cosine and sine functions with applications," *Commun. Stat. - Simul. Comput.*, vol. 0, no. 0, pp. 1–14, 2018.

[23] D. Kumar, U. Singh, and S. K. Singh, "A new distribution using sine function–its application to bladder cancer patients data," *J. Stat. Appl. Prob.*, vol. 4, no. 3, p. 417, 2015.

[24] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Phys. Rev. A*, vol. 98, p. 032309, Sep 2018.

[25] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Phys. Rev. Lett.*, vol. 122, p. 040504, Feb 2019.

[26] K. M. Nakanishi, K. Fujii, and S. Todo, "Sequential minimal optimization for quantum-classical hybrid algorithms," *arXiv preprint arXiv:1903.12166*, 2019.