Quantum estimation of classically intractable kernels for highly-entangled feature maps

Ziheng Cao, Connie Hsueh, Irene Zhang

Introduction

A quantum computer encodes data in a “qubit,” which is a quantum state that can exist in a superposition of states. That is, in analogy to a classical bit that can take on the state of 0 or 1, a qubit exists as both 0 and 1. A calculation performed with n-qubits then is parallelized over 2^n states. As n gets large, this calculation in 2^n-space quickly surpasses the capabilities of classical computing.

Current quantum processor hardware is limited to a few dozens of physical qubits and additionally suffer from poor fidelity and limited connectivity, both of which pose large hurdles for scaling. As a point of reference, one estimate suggests it would require 20 million noisy qubits to break 2048-bit RSA encryption[1]. So presently, we are in the so-called "noisy intermediate-scale quantum" (NISQ) era within the roadmap toward societally-impactful quantum calculations, where the technological challenges to improve quantum hardware are paralleled by the search for low-depth quantum circuits which can perform meaningful calculations even within the limitations mentioned above.

The widespread interest in this pursuit was recently bolstered by Google's announcement of quantum supremacy (that is, a real example of a quantum speedup compared to the most powerful existing classical supercomputers), where they demonstrated million-fold sampling the probability distribution of an entangled quantum state[2]. This task, if attempted classically, becomes exponentially more difficult with increased number of qubits and gates, the authors estimate it would have taken about 10,000 years on a supercomputer.

Our project is aimed at exploring proposals for machine learning on near-term quantum computers. The successful application of these algorithms will be a balance of a number of practical and technical considerations, including the fidelity of quantum processors, the types of classically-difficult problems, and the choice of entanglement operations. We will focus on a quantum kernel estimation algorithm[3] which utilizes a quantum computer to construct an estimator for a kernel that is intractable to compute classically. This kernel can then be used for non-linear classification via support-vector machine.

Description of algorithm

The quantum SVM (QSVM) estimates a quantum-enhanced kernel matrix which is then used in a classical SVM to extract support vectors[3]. The QSVM quantum circuit operates on *d* qubits, where *d* is the dimension of the input vectors. Each qubit is initialized to the |0> state, so that the initial state of the system is |0^d>. The feature map adds relative phases to the qubits based on the elements of the input vector. The “entanglement” is accomplished by adding phases to qubits that are conditional on the states of the other qubits. The feature map is implemented by applying quantum logic gates which can be described as Hermitian operators being applied to the *d*-qubit state vector. If *U(x)* is the feature map circuit, the resulting feature vector is

$$|\phi(x)\rangle = U(x)|0^d\rangle$$

Each element of the kernel matrix is in principle the dot product of two feature vectors generated by this feature map. For each pair of inputs taken from the training data, the corresponding element of the kernel matrix is then

$$\langle \phi(x)|\phi(x)\rangle = \langle 0^d|U^\dagger(x)U(x)|0^d\rangle$$

The above quantity is the square root probability that the state vector will be in the |0^d> state after the operator *U(x)* and its adjoint are applied to the initial state. Therefore, to estimate each kernel matrix element one runs the kernel-estimation circuit $U^\dagger(x))U(x)$many times, recording the frequency at which the output state is |0^d>. Fig. 1 shows an example circuit diagram, with the individual gates composing *U(x)* explicitly drawn for *d = 2*. Due to qubit-number limitations in NISQ processors, the input dimension *d* for IBM Qiskit's quantum kernel-estimator is limited to 3 or below.

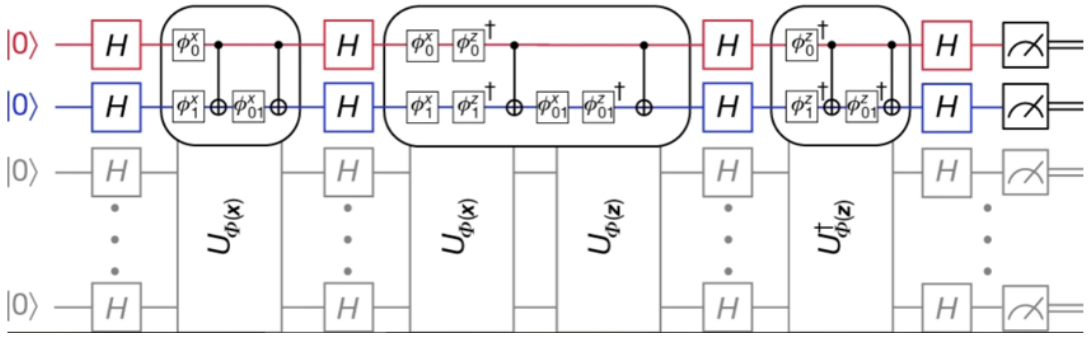

Fig. 1. Circuit diagram for estimating the kernel matrix given a quantum-enhanced feature map

A typical workflow using the QSVM algorithm is 1) reduce dimensions of data and scale if needed, 2) use quantum processor to estimate the training kernel, 3) classically compute the support vectors, 4) use the quantum processor to estimate the testing kernel.

Data selection and processing

We chose five datasets on which the test the quantum-enhanced feature map: the Breast Cancer dataset[4] from the UCI repository, the three artificially generated ad hoc datasets used in IBM paper[3], and a self-generated dataset of 2D square lattice Ising spin configurations.

The first choice of datasets is motivated by the question of whether one can expect the problems solvable by classical ML methods to easily map onto quantum ML methods. Although it is tempting to think of quantum computing as an enhancement of classical computing, they often are most useful for somewhat orthogonal groups of problems.

In the IBM paper[3], the QSVM is trained and tested on three ad hoc datasets that are artificially created to be completely separable by the QSVM. The data for each set were generated on a quantum processor using n=d=2 qubits by applying the feature map described above to randomly generated input vectors in $[0, 2\pi)^d$, then applying one and two-qubit unitary operators to the resulting state. Positive and negative labels are assigned according to the output of the final state with a boundary gap of 0.3. Because these datasets are generated using the quantum-enhanced feature map, we expect that the QSVM will do well on it. The main purpose of including this dataset is to see whether a classical kernel can perform as well.

Finally, because the quantum-enhanced feature map is implemented via a quantum mechanical process, we were interested in seeing its performance on a dataset generated by simulating a quantum mechanical process motivated by physics. The Ising model is a toy model

used to describe ferromagnetism, the phenomenon in which atomic spins in a material tend to align and form a permanent magnet. Each spin can be up (+1) or down (-1) and the spins sit on a 2D square lattice. The configuration of spins on an *LxL* lattice has an energy determined by how aligned each spin is with its neighbors. This configuration energy competes with thermal energy, which tends to disorder the spins. Therefore, at low temperatures, i.e. below the Curie temperature, spins tend to align with each other; and at high temperatures, the spins remain disordered. We generate spin configurations for various temperatures above and below the Curie temperature. The input vector has 256 components, one for each spin in a 16 x 16 grid, and its label is 0 if the configuration is ordered (below the Curie temperature) and 1 if it is disordered (above the Curie temperature). The spin configurations are generated via standard Monte Carlo methods with the Metropolis algorithm, as described in the attached Jupyter notebook.

To process the two non-ad hoc datasets, we scaled the input data and performed PCA using sci-kit learn's implementation to reduce the dimensionality of the input vectors to be within the constraints of the QSVM algorithm. For comparison with classical methods, we used the same preprocessing and an RBF kernel for the quantum data and degree-2 polynomial kernel for the 2D Ising data.

Results

To evaluate the performance of the quantum-enhanced feature map, we compared its classification accuracy to the RBF kernel SVM and the second-order polynomial kernel SVM. The quantum algorithm was run on IBM Q's quantum simulator[5] using the Qiskit software framework[6]. The simulator can be toggled between linear entanglement (more easily physically realizable) or full entanglement. Our results are tabulated in Table 1.

Table 1. Comparison of Test performance between QSVM and classical SVM

Dataset	Pre-processing	Feature dim	Kernel (backend)	Training set size	Test set size	Test set error
Ad Hoc	-	3	rbf	3x440	3x40	0.125, 0.05, 0.075
	-	3	QSVM (linear simulator)	3x440	3x40	0
Breast Cancer	Standardize, PCA, Min/Max to (-1, 1)	3	QSVM (linear simulator)	200	40	0.15
Ising	-	256	polynomial-2	440	440	0.050
	PCA, Standardize	3	polynomial-2	440	440	0.064

	Standardize, PCA, Min/Max to (-1, 1)	3	QSVM (linear simulator)	440	440	0.114
	Standardize, PCA, Min/Max to (-1, 1)	3	QSVM (full simulator)	440	440	0.143
	PCA, Min/Max to (-1, 1)	3	QSVM (linear simulator)	440	440	0.054

The QSVM classified all three sets of ad hoc data with 100% accuracy, compared to RBF kernel accuracies of 87.5%, 95%, and 92.5%. This demonstrates that for these specifically-generated datasets, mapping the data onto a quantum state indeed improves its separability. However, QSVM classifies data from the breast cancer standard dataset with just 85% accuracy, indicating that this highly-entangled feature map does not improve performance for all classification problems. This is unsurprising---a more complex feature map does not generically mean improved performance, especially if the real classification of the data does not depend on the more complex feature map. For the Ising data, the second-order polynomial SVM, without any preprocessing to the data, can classify the test set with 95% accuracy. Using PCA to reduce the data to three dimensions decreases this accuracy to 93.6%. Our initial attempts at classifying with the QSVM were not very promising---88.6% for linear entanglement and 85.7% for full entanglement. However, we realized that the standardization step may not be necessary or justified for discretized features as in the Ising model; removing this step increased our accuracy to 94.5%, outperforming the classical SVM. This demonstrated that the QSVM, even performed on just 3 qubits, can classify this (reduced) quantum dataset with at least as high accuracy as a classical SVM, and perhaps its extension to more qubits would improve it further.

Discussion

For the quantum motivated dataset ad hoc data, which is artificially generated by quantum processors, we get perfect test performance with our QSVM algorithm, outweighing classical RBF kernel method. Through this trial, we confirm that the quantum generated ad hoc data is completely separable by QSVM.

The performance of QSVM on breast cancer standard dataset is not ideal, which implies that the quantum-based, entangled feature map doesn't match the philosophy governing a normal real-world dataset quite well. This motivates us to turn to data generated by Ising model which we believe is inherently controlled by quantum mechanics. The results shown in Table1 indicates: (a) Increasing the entanglement or making the feature map more complicated doesn't necessarily increase the test performance of QSVM; (b) Method of pre-processing of data may be crucial to the test result. In this case, when we get rid of the standardization step, we are able to increase the test performance by ten percentage points; (c) Even within the limitation of number of qubits we have for QSVM, we can get comparable or even better performance on QSVM algorithm than classical SVM with the same feature dimension for problems related or generated from quantum mechanical systems.

In addition to running the QSVM algorithm using IBM Q's quantum simulator, we also implemented the QSVM on IBM Q's "Melbourne" 16-qubit and "Essex" 5-qubit QPUs. We found

that current publicly available quantum processors suffer from qubit-number limitations and high noise. Furthermore, queue times and classical interfacing also extend computation time. Compounding the aforementioned issues is the number of circuits that need to be implemented to estimate any given kernel matrix. As an example, training the QSVM with just 40 training and 40 test points requires thousands of circuits and is computationally expensive for these processors.

Conclusions/Future Work

Our work demonstrates that quantum computers can estimate kernels for highly-entangled feature maps thought to be classically-intractable. This quantum-enhanced mapping provides comparable or even better classification for some of the investigated datasets; however, the trade-off between its slight improvement in accuracy versus the increased overhead costs related to time (e.g. queue times), ease-of-implementation, and noise if ultimately implemented on a QPU, make its practical application doubtful in the near future.

But despite the unfavorable assessment for immediate application, this algorithm and other prospects for machine learning with quantum computers should continue to be investigated in the context of their long-term potential as the number and fidelity of qubits increase. Possibilities include: evaluating how performance changes with number of qubits, as that allows for increased feature size but also magnifies the number of circuit elements; running the algorithm on QPUs to investigate the effect of noisy qubits; choosing different entangling gates; and finding niche applications that may *a priori* physically motivate a choice of quantum-enhanced feature map.

Contributions:

Z.C. contributed to writing the report, C. H. contributed to implementing the QSVM on the quantum simulator, and I. Z. contributed to generating the Ising dataset and implementing the classical SVM and the QSVM on QPUs.

Link to project code:

https://drive.google.com/open?id=1-Q6kwuO9YsW7UA3ELxpAkoL3D55to0zW

References

(1) Gidney, C.; Ekerå, M. How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits. ArXiv Prepr. ArXiv190509749 2019.

(2) Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J. C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F. G.; Buell, D. A. Quantum Supremacy Using a Programmable Superconducting Processor. Nature 2019, 574 (7779), 505–510.

(3) Havlíček, V.; Córcoles, A. D.; Temme, K.; Harrow, A. W.; Kandala, A.; Chow, J. M.; Gambetta, J. M. Supervised Learning with Quantum-Enhanced Feature Spaces. Nature 2019, 567 (7747), 209.

(4) https://Archive.Ics.Uci.Edu/Ml/Datasets/Breast+Cancer.

(5) https://Www.Ibm.Com/Quantum-Computing/.

(6) https://Github.Com/Qiskit.