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

Ziheng Cao (caozh003), Connie Hsueh (chsueh), Irene Zhang (ipzhang)

Introduction

- While quantum computing holds enticing promise, we are currently in the so-called "noisy intermediate-scale quantum" (NISQ) era, limited by number of physical qubits and poor fidelity
- An algorithm called "quantum SVM"(QSVM) [1] is investigated to explore the utility of machine learning on near-term quantum processors
- Such calculations are predicted to be classically intractable; here, we estimate the kernel via measurement of short-depth quantum circuits
- We compare the results of this quantum-enhanced classification algorithm to classical SVMs for data generated ad hoc and for data of an Ising model on a square lattice [2]

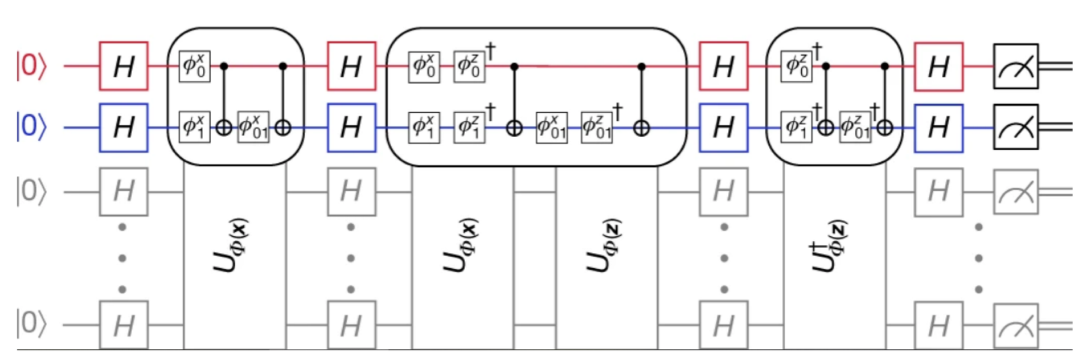

Fig. 1. Circuit diagram of the QSVM algorithm for input dimension of 2. The unitary feature map U is composed of phase gates and controlled phase gates applied to the two qubits. An extension to n qubits is shown in gray. Figure from [1].

Methods

- The algorithm of QSVM: The initial state of the qubits in the quantum processor is set to 0. An input data vector **x** is "written" to the qubits by applying a quantum circuit to the initial state, which adds relative phases to the qubits corresponding to the value of the input
- For each pair of inputs taken from the training data, the corresponding element of the kernel matrix is calculated by measuring the projection of the feature vectors onto each other.
- Measuring the final state many times and recording the frequency at which the initial state is output therefore gives an estimate for the elements of the kernel matrix
- For both the training and prediction phases, the kernel matrix is constructed via the quantum processor

Feature Selection

- The number of qubits needed is equal to the dimension of the input vector
- Due to the limited size of NISQ processors, the quantum SVM (QSVM) algorithm we used requires input vectors to be of length 3 or less.
- Therefore, we used PCA to reduce the dimensionality of datasets with greater dimensionality (i.e. the 2D Ising spins)
- 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

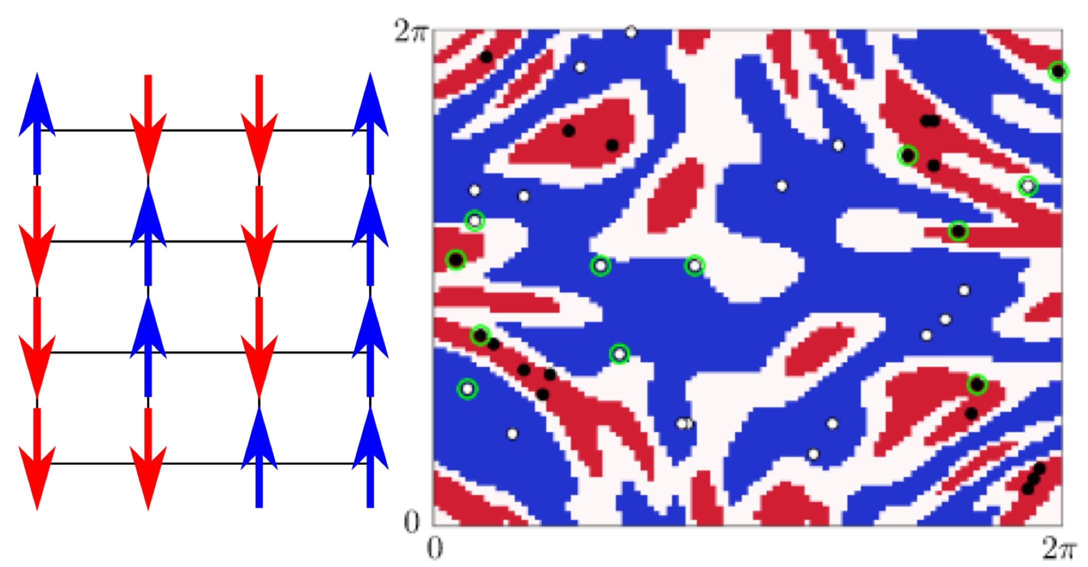

Fig. 2. (a) 2D Ising spin lattice of size 16. To generate our data we used standard Monte Carlo methods to generate lattices of size 256 at various temperatures. We label the generated configurations "ordered" for those below the ferromagnetic transition temperature and "disordered" for those above. (b) A quantum algorithm-generated dataset with d=2. This set is generated by applying Pauli Z gates to each qubit and then a random two-qubit unitary. (b) from Supplemental Information of [1].

Data

- We ran the QSVM algorithm on a quantum algorithm-generated test set, the Breast Cancer dataset, and a 2D ferromagnetic Ising spin lattice dataset.
- We ran a classical SVM on the quantum dataset and 2D Ising dataset.
- These datasets were chosen to compare performance on a dataset generated with knowledge of the intended entanglement, a standard ML dataset, and a data generated from a quantum mechanical process.

Results

Dataset	Pre-processing	Feature dim	Kernel (backend)	Training / test set size	Test set error
Ad Hoc	-	3	rbf	3x440 / 3x40	0.125, 0.05, 0.75
	-	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

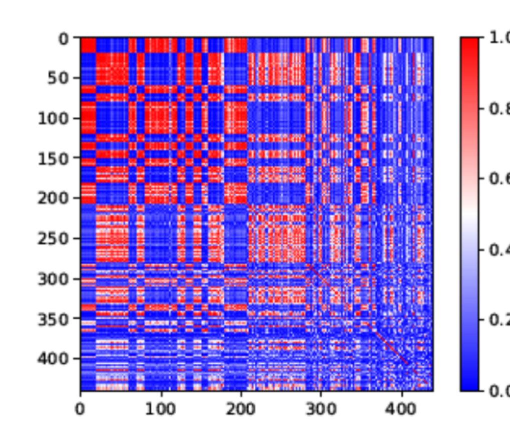

Fig. 3. Estimated training kernel matrix for Ising dataset with full entanglement.

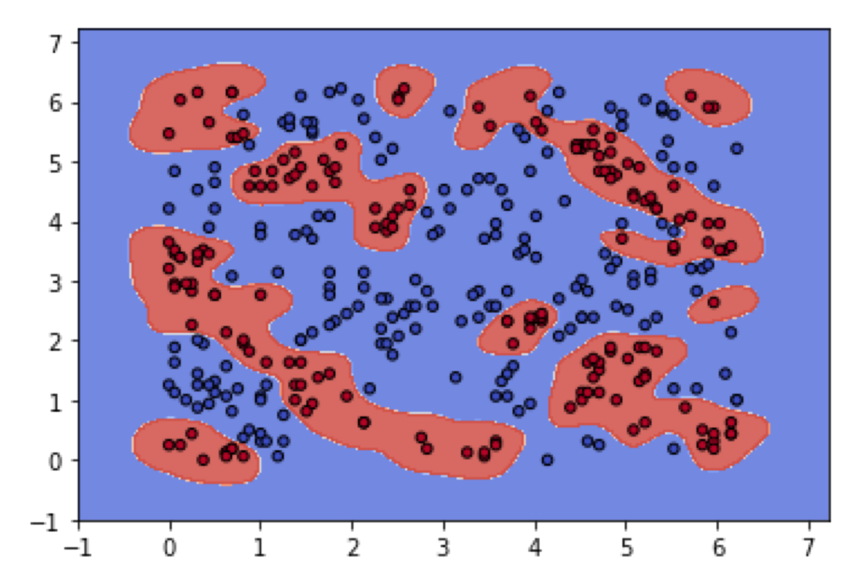

Fig. 4. Training result of a classical SVM with an rbf kernel on the quantum algorithm-generated dataset shown in Fig. 2.

Future Work

- Extending this algorithm to more qubits
- Further exploration of QPUs versus quantum simulators
- investigation of different entangling gates and physically-motivated choice of feature maps

Discussion

Algorithm:

- We tested a quantum-enhanced feature map on a custom dataset, a standard dataset, and a quantum-motivated dataset and found its comparison to classical kernels middling
- We estimated the kernel using IBM's Qiskit software framework, the IBM Q quantum simulator, and the IBM Q 16-qubit QPU
- We had expected the quantum SVM to perform comparably or better than classical SVMs on the quantum-motivated dataset because of the mapping onto a quantum state
- However, the performance of the QSVM algorithm is sensitive to the type of data and the choice of entangling gates
- Therefore, since our feature maps were not tailored to our data, we may not have achieved the best possible performance

Implementation:

- Current publicly available quantum processors suffer from circuit-number limitations and high noise.
- Queue times and classical interfacing also extend computation time
- Training the QSVM with just 40 training and test points requires thousands of circuits and is computationally expensive for these processors

Our work demonstrates that there are quantum algorithms that can solve classically-intractable problems, but their practical application is unclear

References

[1] V. Havlíček et al., "Supervised learning with quantum-enhanced feature spaces," Nature, vol. 567, no. 7747, pp. 209–212, Mar. 2019.

[2] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," Nat. Phys., vol. 13, no. 5, pp. 431–434, May 2017.