Quantum Kernel Estimation and Variational Classifiers applied to binary classification

Ario Altamura

Giulio Crognaletti

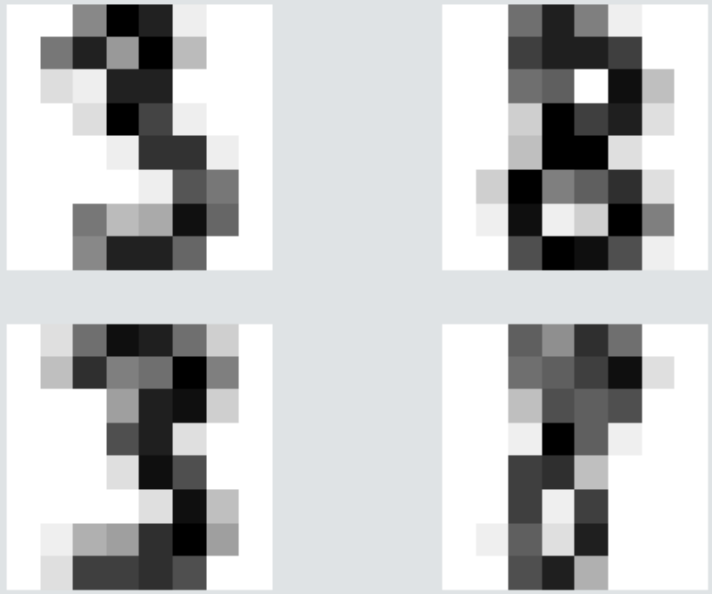
Antonio Mastropasqua

Datasets used

- Handwritten digits dataset, from scikit-learn library
  - Real world problem
  - Toy problem, almost linearly separable
- Ad hoc dataset, proposed by Havlíček, Córcoles, Temme *et al* [1]
  - Defined ad hoc to show quantum advantage
  - Hard to solve classically

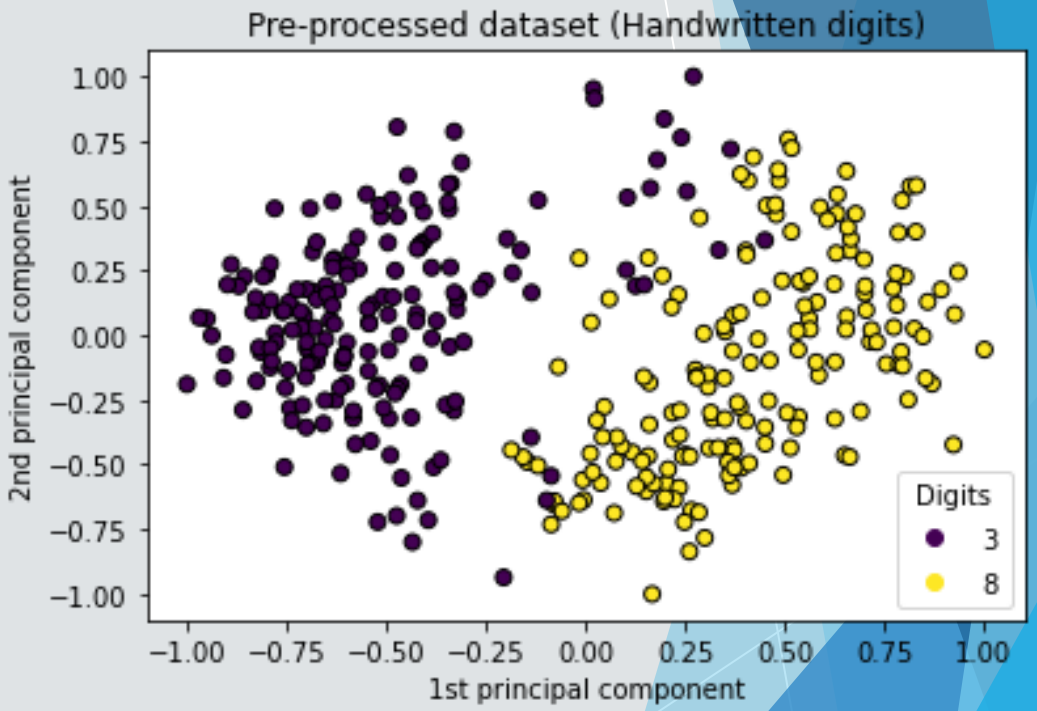
Data exploration - MNIST

Raw dataset (Handwritten digits)



Dimensionality Reduction

PCA



Explained Variance: 36.2 %

Algorithms used

- Support Vector Machines (SVM)
  - Classical Kernel Method

    Using Radial Basis Function Kernel
  - Quantum Kernel Estimation (QKE)

    Using Pauli Feature Maps
- Variational Quantum Classifier (VQC)

  Using Pauli Feature Map and TwoLocal ansatz

Classical Kernel Method

When using SVM, the classifier is able to train and make predictions using only inner products between datapoints.

To maximise performance, it is generally useful to first map each datapoint to a bigger "feature space" through a non-linear mapping F:

$$\langle \vec{x_i}, \vec{x_j} \rangle \rightarrow \langle F(\vec{x_i}), F(\vec{x_j}) \rangle$$

The idea of kernel methods is to evaluate this scalar product without actually computing the transformation.

Example (Radial Basis Function):

$$\langle F(\vec{x_i}), F(\vec{x_j}) \rangle := \langle \langle \vec{x_i}, \vec{x_j} \rangle \rangle_{RBF} = \exp(-\gamma ||\vec{x_i} - \vec{x_j}||^2)$$

Classical Kernel Estimation - Results

Optimal Parameters

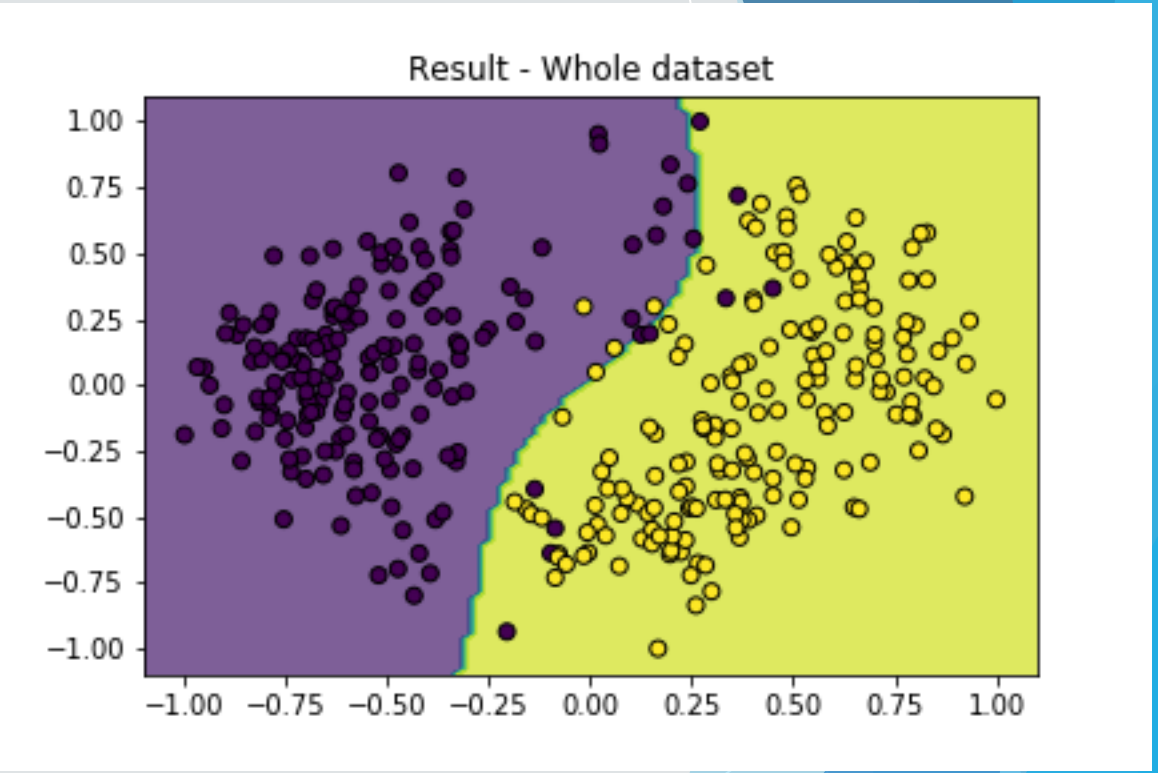
- Kernel type: Radial Basis Function
- Parameter: $\gamma = \frac{1}{N\mathrm{Var}[x]}$

Data used

- Training size: 50 datapoints
- Testing dataset size: 50 datapoints and 300 datapoints

Results

- Score (50 dp): **98.0 %**
- Score (300 dp): **96.0** %




```
classifier = SVC(kernel='rbf',probability = True,gamma='scale')
classifier.fit(data,label)
```

Quantum Kernel Estimation

The idea of Quantum Kernel Estimation is to approximate inner products as the expectation of quantum circuits:

$$\langle\langle \vec{x_i}, \vec{x_j} \rangle\rangle_{QKE} = \langle 0 | \mathcal{U}_{\Phi(\vec{x_i})} \mathcal{U}^{\dagger}_{\Phi(\vec{x_j})} | 0 \rangle$$

When using Pauli feature maps, we define $\mathcal{U}_{\Phi(\vec{x})} = \prod_{k=1}^{d} U_{\Phi(\vec{x})} H^{\otimes n}$ in terms of:

- Number of features *n*
- Depth *d*
- Choice of the unitaries $U_{\phi(x)}$

Quantum Kernel Estimation

The operators $U_{\phi(x)}$ also have definite structure:

$$U_{\Phi(\vec{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{i \in S} P_i\right)$$

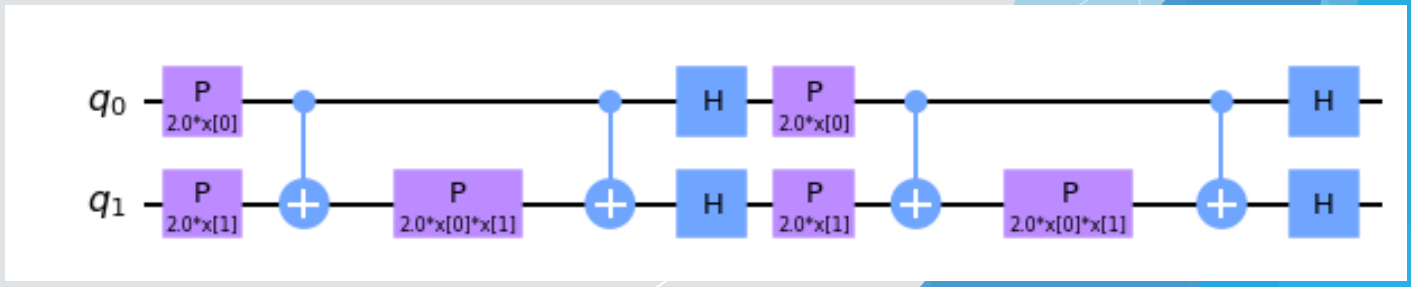
They are defined in term of:

- Data mapping function $\phi_S(x)$ and connectivity patterns S
- Pauli operators P_i

Example:

- $d = 2$, $P_i = X$,$\forall\ i$

$$\Phi(\vec{x}) = \prod_i x_i$$



Quantum Kernel Estimation - Results

Optimal parameters

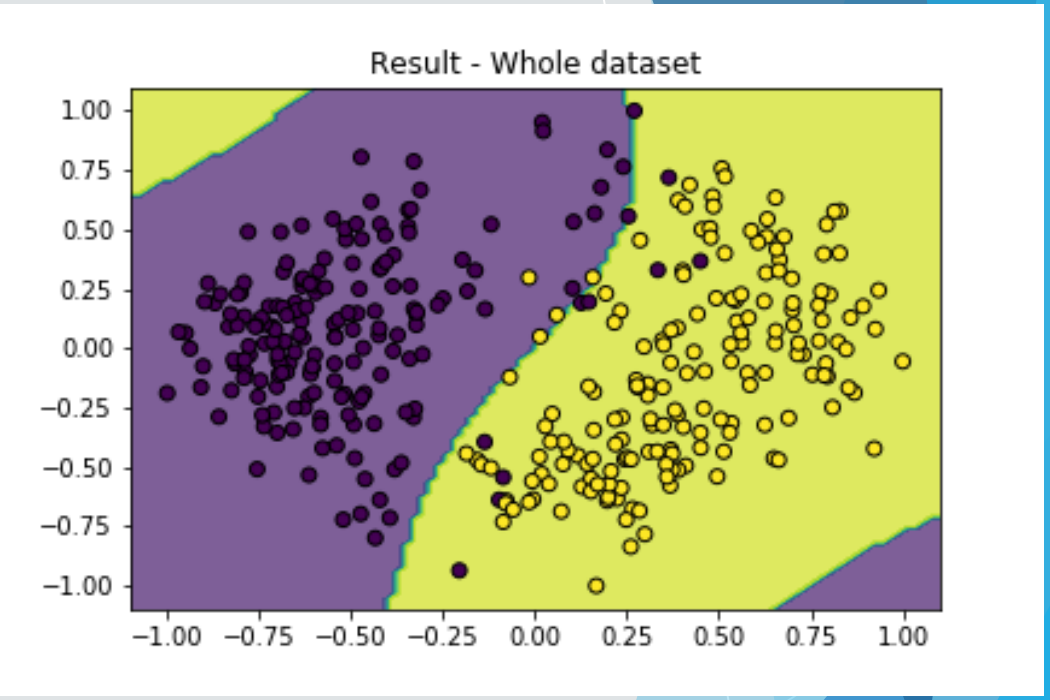
- Feature Map:
  - Depth: d=2
  - Pauli Operators: P_i = X ,$\forall$ i
  - Data mapping function: $\Phi(\vec{x}) = \prod_i x_i$

Data used

- Training size: 50 datapoints (real hw)
- Test size: 50 with real hw, 300 for simulation

Results

- Score (IBMQ Lima): **98.0 %**
- Score (Noisy Simulator): **95.7 %**



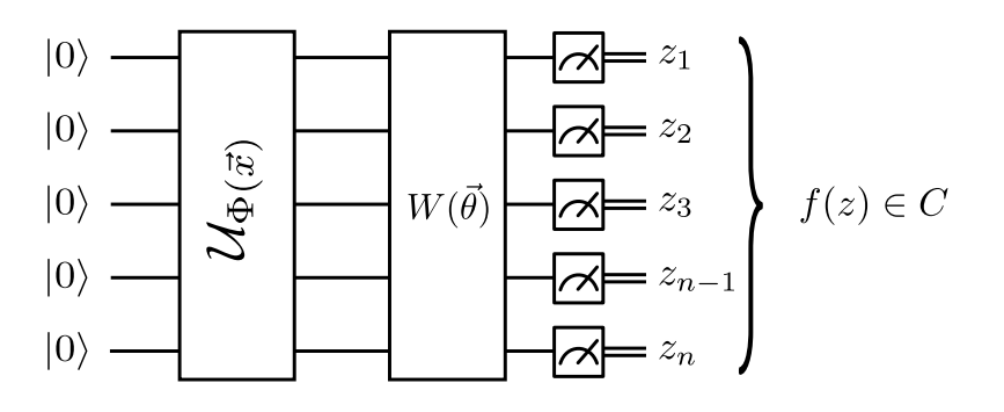

```
# Quantum Kernel definition
quantum_feature_map = PauliFeatureMap(paulis=['X','XX'],
                                      feature_dimension=2,
                                      reps=2)
quantum_kernel = QuantumKernel(feature_map=quantum_feature_map,
                               quantum_instance=ibqm_lima)

q_svc = SVC(kernel=quantum_kernel.evaluate)
q_svc.fit(data, label)
```

Variational Quantum Classifier

This classifier is based on the minimization of a cost function (i.e. Cross entropy loss function):

- In the training stage a set of labeled data points are encoded in the circuit by a feature map (Pauli feature map)
- W represents the variational form. At each step, classification is done using the weights $\theta_{current}$ for the circuit

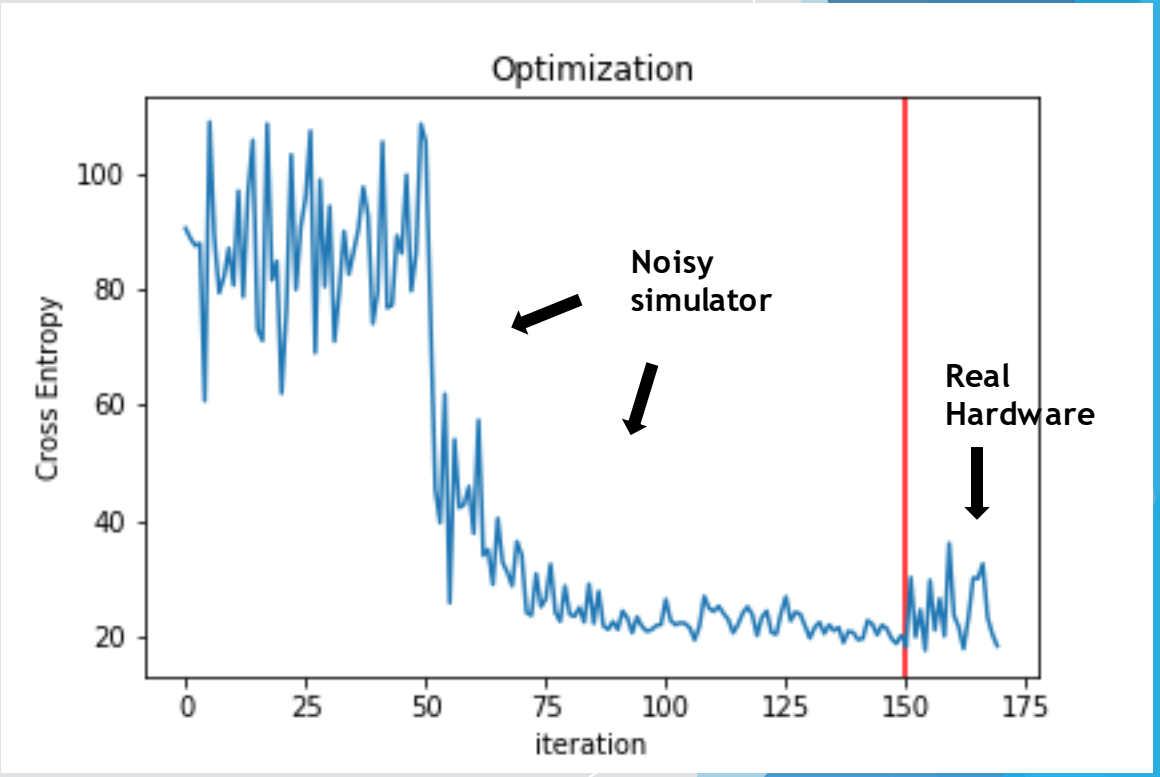


- An optimization routine is used to update the weights until the convergence in the cost function is reached

Variational Quantum Classifier

Optimization strategy

- using a random initial point is too demanding for the real backend, givent the typical run-times
- it is better to use a **warm start** provided by the optimal parameters obtained by a simulator that reproduce the same noise of the real hardware
- Optimizer used: SPSA (Simultaneous Perturbation Stochastic Approximation)



Variational Quantum Classifier - Results

Optimal Parameters

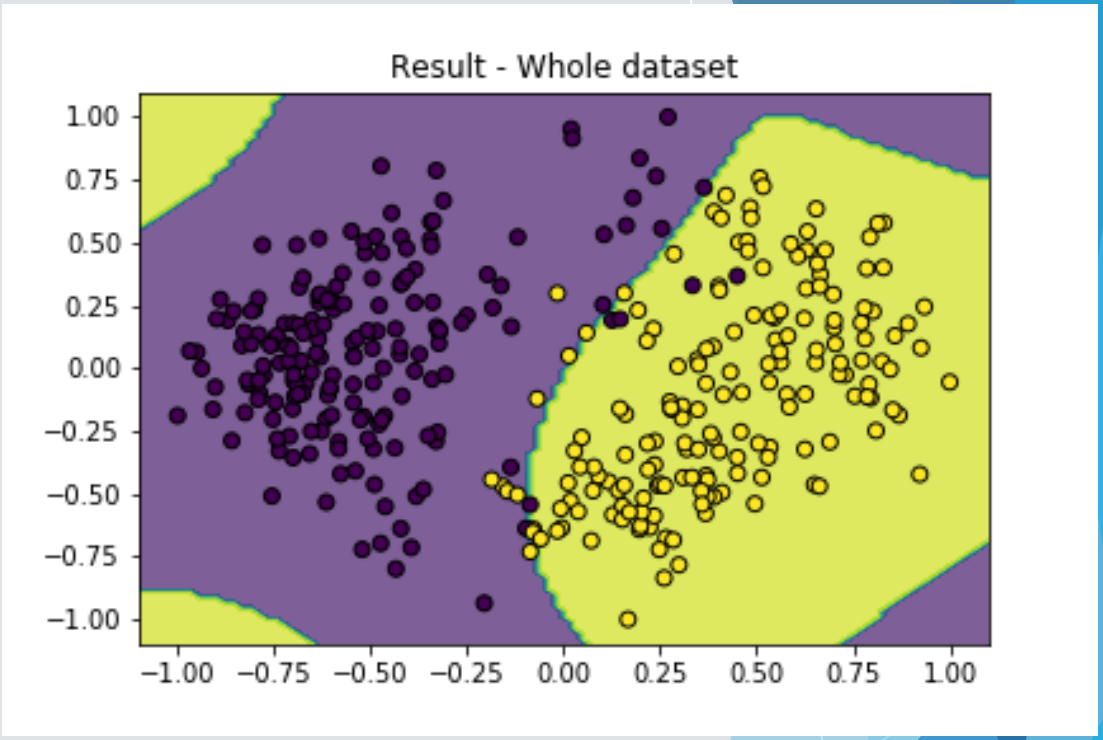
- Ansatz: Two Local Circuit
  - Single qubit rotations: Rx, Ry
  - Entangling gate: CZ
  - Entanglement map: linear
  - Depth: 3
- Feature map:
  - As in the Quantum Kernel section

Data Used

- As in the Quantum Kernel section

Results

- Score (IBMQ Lima): **98.0 %**
- Score (Noisy Simulator): **96.7 %**




```
#Variational Quantum Classifier definition
vqc= VQC(n_qubits=2,
         feature_map=quantum_feature_map,
         ansatz=TwoLocal(2,['rx','ry'],'cz','linear',reps=3),
         loss=CrossEntropyLoss(),
         optimizer=SPSA(maxiter=50),
         quantum_istance=ibqm_lima,
         initial_point=weight_parameters,
         callback=callback1)
```

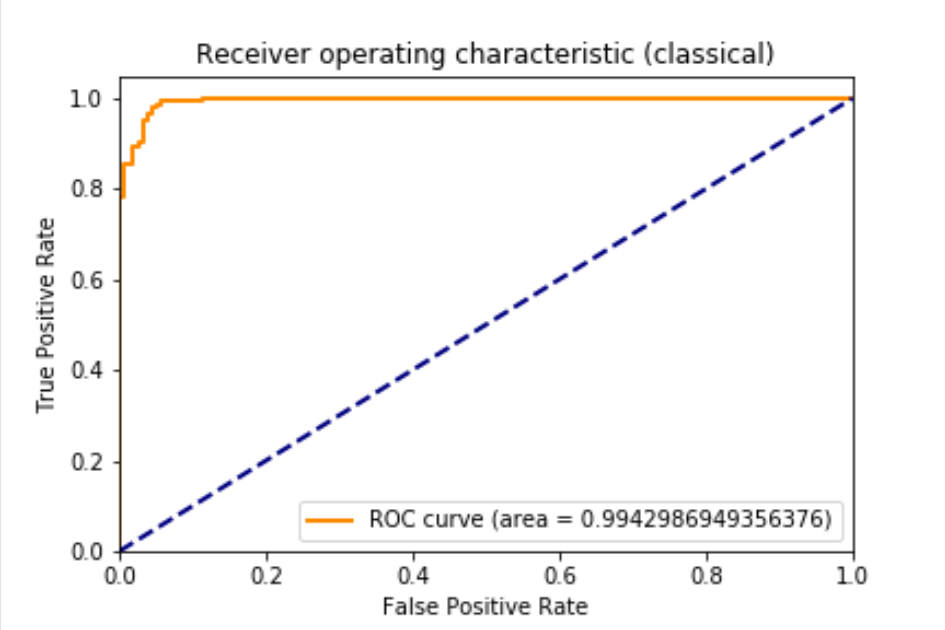
Performance metrics – Real Hardware

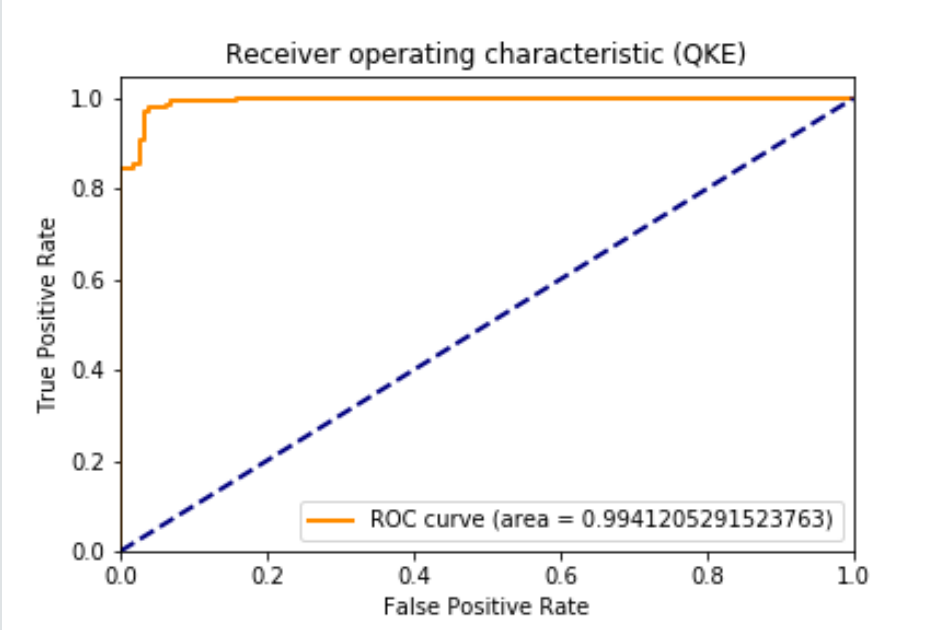
	Accuracy score	Balanced Accuracy	AUC
Classical Kernel SVM	**98.0 %**	**98.0 %**	**0.995**
Quantum Kernel Estimation	**98.0 %**	**98.0 %**	**0.999**
Variational Classifier	**98.0 %**	**98.0 %**	**0.999**

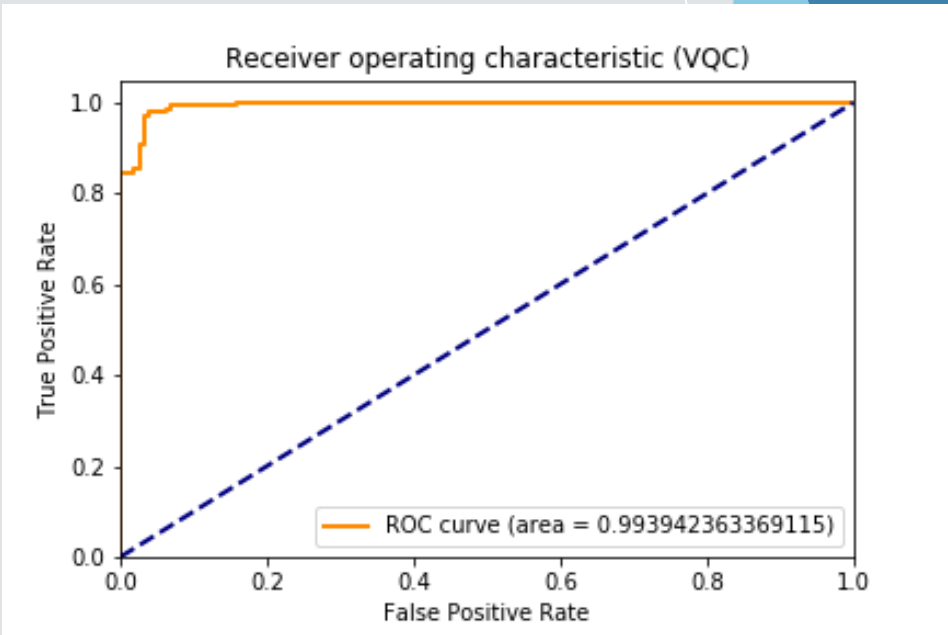
Performance metrics – Noisy simulator

	Accuracy score	Balanced Accuracy	AUC
Classical Kernel SVM	**96.0 %**	**96.0 %**	**0.994**
Quantum Kernel Estimation	**95.7 %**	**95.3 %**	**0.994**
Variational Classifier	**96.7 %**	**96.7 %**	**0.994**

More on AUC


Receiver operating characteristic (classical)
True Positive Rate
False Positive Rate
ROC curve (area = 0.9942986949356376)
0.0
0.2
0.4
0.6
0.8
1.0


Receiver operating characteristic (QKE)
True Positive Rate
False Positive Rate
ROC curve (area = 0.9941205291523763)
0.0
0.2
0.4
0.6
0.8
1.0

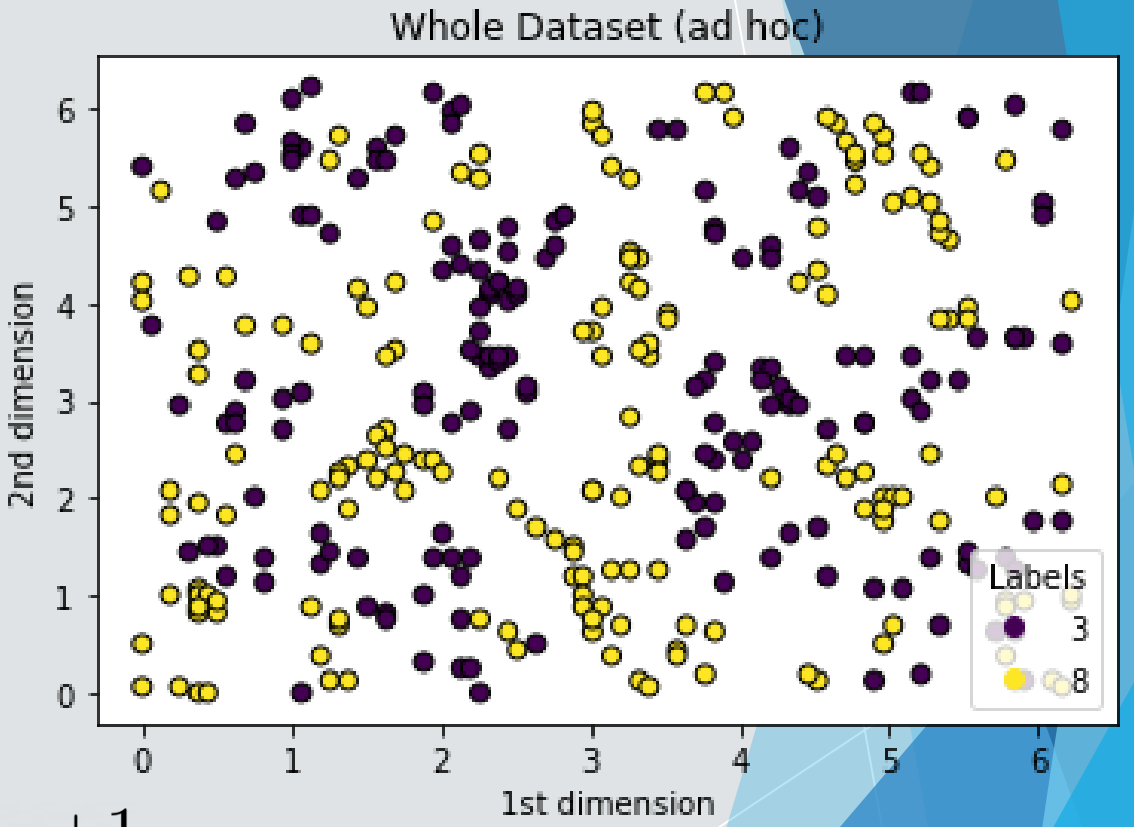

Receiver operating characteristic (VQC)
True Positive Rate
False Positive Rate
ROC curve (area = 0.993942363369115)
0.0
0.2
0.4
0.6
0.8
1.0

Ad hoc dataset – Data exploration

The datapoints are generated choosing a unitary $V \in SU(2^n)$ and using the rule below, given a gap parameter Δ and a number of features n

Parameters used

- $\Delta = 0.3$
- $n = 2$



$$\langle \Phi(\vec{x}) | V^\dagger Z^{\otimes n} V | \Phi(\vec{x}) \rangle \geq \Delta \quad \Rightarrow \quad m(\vec{x}) = +1$$

$$\langle \Phi(\vec{x}) | V^\dagger Z^{\otimes n} V | \Phi(\vec{x}) \rangle \leq -\Delta \quad \Rightarrow \quad m(\vec{x}) = -1$$

Ad hoc dataset – Classical Kernel Estimation

Optimal Parameters

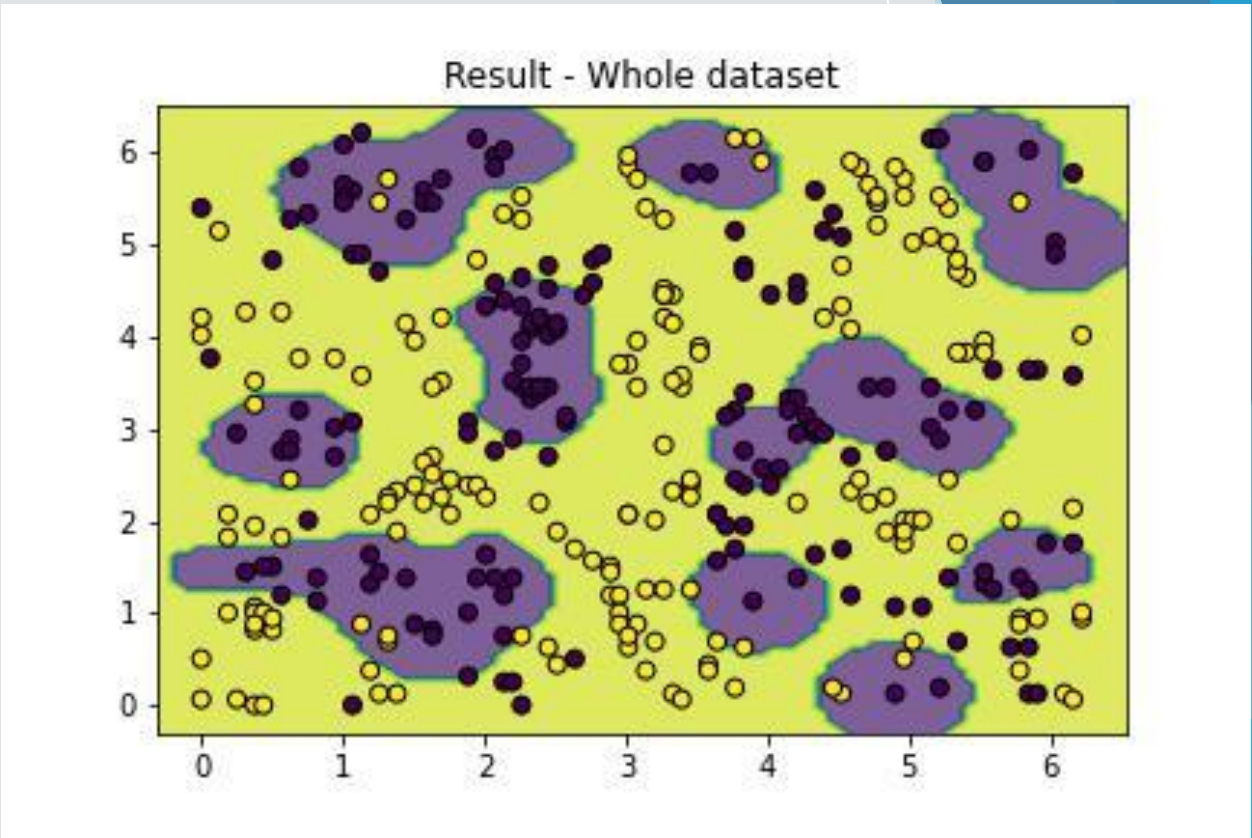
- Kernel type: Radial Basis Function
- Parameter: $\gamma = 10.0$

Data used:

- Training size: 50 datapoints
- Testing dataset size: 300 datapoints

Results:

- Score: **72.7 %**



Ad hoc dataset – Quantum Kernel Estimation

Optimal Parameters

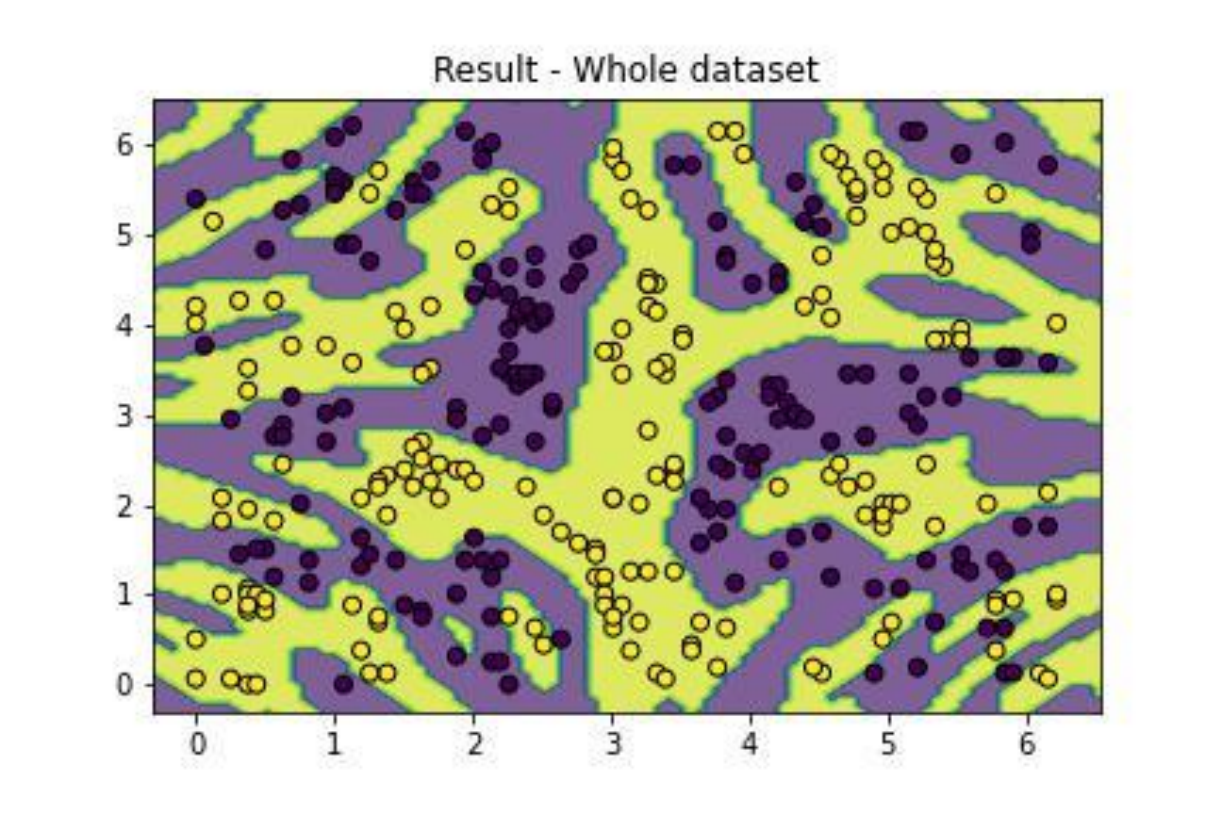
- Depth: d = 2
- Pauli operators: P_i = Z ,$\forall$ i
- Data mapping function: default

Data used:

- Training size: 50 datapoints
- Testing dataset size: 300 datapoints

Results:

- Score (Noisy simulator): **99.7 %**



Ad hoc dataset – Variational Quantum Classifier

Optimal parameters

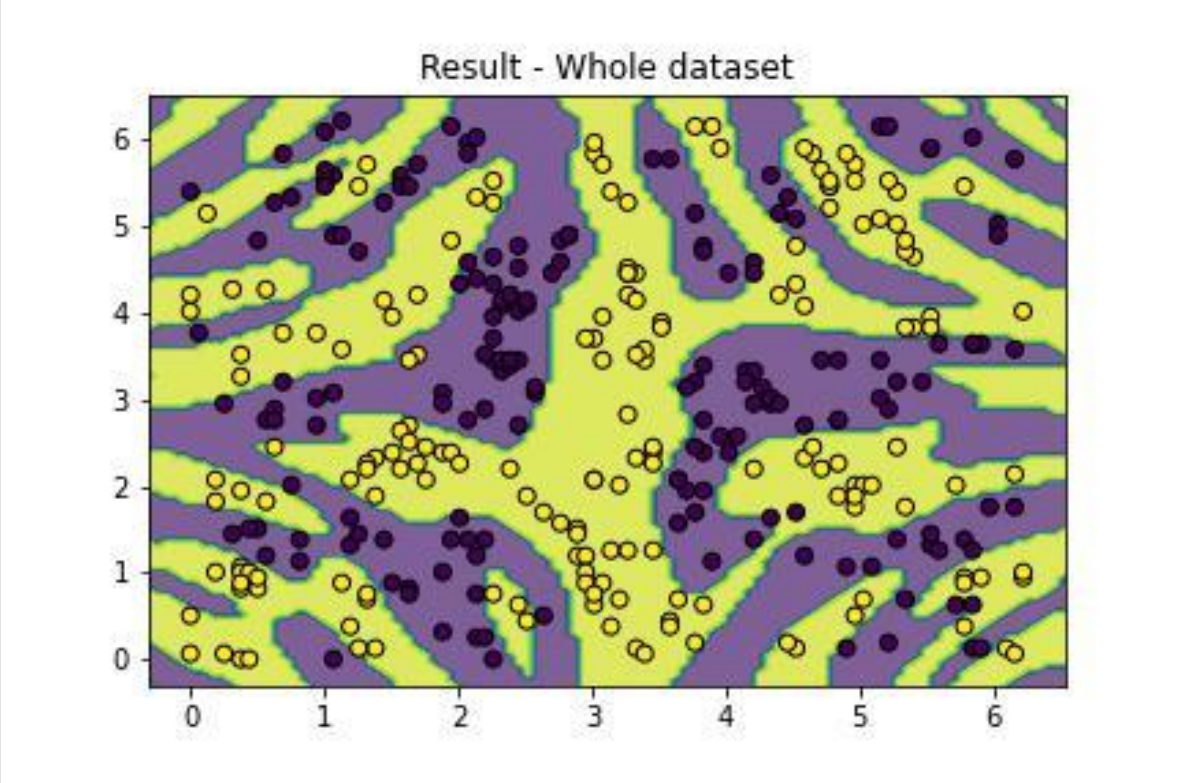
- Ansatz: Two Local Circuit
  - Single qubit rotations: Ry, Rz
  - Entangling gate: CZ
  - Entanglement map: linear
  - Depth: 3
- Feature map:
  - As in the Quantum Kernel section

Data Used

- As in the Quantum Kernel section

Results

- Score (Noisy simulator): **97.7 %**



Bibliography

[1] Havlíček, V., Córcoles, A.D., Temme, K. *et al*. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019). https://doi.org/10.1038/s41586-019-0980-2

Thanks for your attention