

# Empirical Analysis of a Quantum Classifier Implemented on IBM's 5Q Quantum Computer

Wei Hu

Department of Computer Science, Houghton College, Houghton, USA

Email: wei.hu@houghton.edu

**How to cite this paper:** Hu, W. (2018) Empirical Analysis of a Quantum Classifier Implemented on IBM's 5Q Quantum Computer. *Journal of Quantum Information Science*, 8, 1-11.  
<https://doi.org/10.4236/jqis.2018.81001>

**Received:** December 26, 2017

**Accepted:** March 4, 2018

**Published:** March 7, 2018

Copyright © 2018 by author and Scientific Research Publishing Inc.  
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

The development of artificial intelligence today is marked with increased computational power, new algorithms, and big data. One such milestone impressive achievement in this area is Google's AlphaGo. However, this advancement is beginning to face increasing challenges and the major bottleneck of AI today is the lack of adequate computing power in the processing of big data. Quantum computing offers a new and viable solution to deal with these challenges. A recent work designed a quantum classifier that runs on IBM's five qubit quantum computer and tested its performance on the Iris data set as well as a circles data set. As quantum machine learning is still an emerging discipline, it may be enlightening to conduct an empirical analysis of this quantum classifier on some artificial datasets to help learn its unique features and potentials. Our work on the quantum classifier can be summarized in three parts. The first is to run its original version as a binary classifier on some artificial datasets using visualization to reveal the quantum nature of this algorithm, and the second is to analyze the swap operation utilized in its original circuit due to the hardware constraint and investigate its impact on the performance of the classifier. The last part is to extend the original circuit for binary classification to a circuit for multiclass classification and test its performance. Our findings shed new light on how this quantum classifier works.

## Keywords

Quantum Computation, Quantum Machine Learning, Quantum Circuit

---

## 1. Introduction

Quantum computation is a new computing approach based on the laws of quantum mechanics, which is a very hot topic at the frontiers of computing world today. By carefully exploiting the unique features of quantum states,

quantum computers can efficiently solve some problems that are believed to be hard for classical computers.

There are two well-known quantum algorithms that demonstrate the advantage of quantum computation over classical approach. Shor's algorithm for finding the prime factors of a number runs in polynomial time while any classical similar algorithms run in exponential times. Grover devised a quantum algorithm for the problem of unstructured search which achieves a quadratic speed up over the classical algorithms. Not available in classical computing, superposition, entanglement and interference of quantum states employed in quantum computing are generally considered as resources for this speed up. Furthermore, quantum computers can perform simulations of another quantum system such as protein folding that is beyond the capability of any classical computers.

IBM recently released the Quantum Experience to allow users to access a five qubit quantum processor, with services including circuit design, simulation, testing and actual computation on a physical device [1]. It offers researchers great opportunities to test their quantum theories and algorithms on actual quantum hardware. In today's data rich society, the growth rate of data generated by different devices goes up way faster than that of the computer processors' speed. As such quantum computation provides a new computing paradigm to process the enormous data and information we are facing today.

It is true that a quantum computer using quantum laws can process many quantum states simultaneously via quantum parallelism, say it can compute  $f(x)$  through evaluating the function for multiple values of  $x$  simultaneously. But when we need to measure the results of the computation, we can only get one result and sometimes this result is even nondeterministic. Therefore, to get a classical result, we have to run a quantum algorithm multiple times in some cases. In this sense, to produce the same classical result, an algorithm that requires less number of runs the better. In particular, for a quantum classifier, we are not only interested in the classification accuracy, but also less number of runs (or shots in IBM's term) of the algorithm.

Machine learning is generally categorized into three classes: supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, there are classification and regression problems. Quantum advantage has long been known for problems such as factoring, search, and simulation. Only recently have quantum algorithms begun to emerge that demonstrate their quantum speedups in machine learning. These quantum machine learning algorithms have to address the basic techniques such as how to compute the distance among the data points and inner product of two vectors. These computing tasks are easy for classical computers because they can measure the computed terms anytime during the whole process, but quantum computers usually can only take direct measurements at the end of the process. It is no easy task for a quantum algorithm to be creative in exploiting measurement in a non-trivial way during the computing process.

The work in [2] was to design a quantum circuit for a distance based binary classifier runnable on IBM's 5Q quantum computer. The quantum advantage of this classifier is the ability to compute the distance of the test data point to all the training data points simultaneously. In contrast, a major drawback of classical distance based classifier is that it can be computationally expensive. Other quantum machine learning algorithms can be seen in [3]-[13].

This classifier was tested on iris dataset and a circles data set. The numerical results were summarized in two tables [2]. We thought it might shed more light on the nature of this quantum classifier if some visualization of its working can be seen. More specifically how the distance of the data points influences the prediction probabilities caused by the stochastic nature of quantum computing. The second thought is how the swap operation in the circuit (Figure 1) has a toll on the prediction accuracy as this operation is necessary in quantum circuit design to meet certain hardware constraints. Our final thought is how to extend the quantum binary classifier to a multiclass classifier, a further step to illustrate the potential of quantum machine learning. The purposes of our study entail the use of IBM's quantum simulator instead of its real quantum device as we need to use more than five qubits.

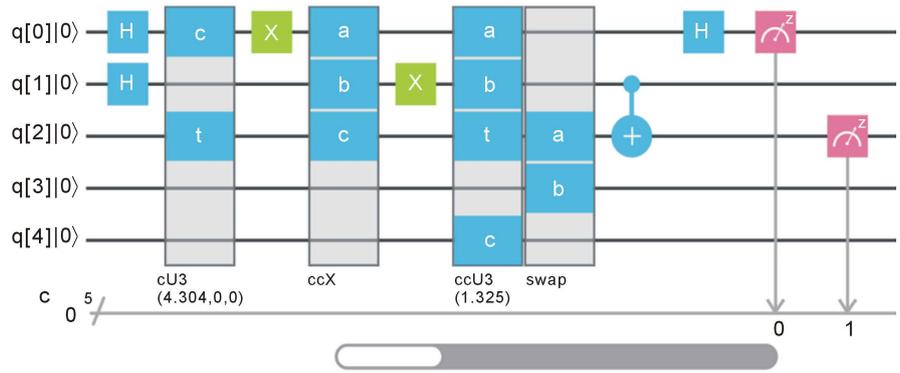
## 2. A Quantum Classifier

This paper reports our empirical analysis of a quantum classifier implemented on IBM's 5Q computer. We will briefly present the distance based quantum classifier created in [2]. Quantum computing requires the data be encoded in quantum states for storage, transformation, and processing. In the context of the analysis of classical data, one method is to encode the coordinates of a classical data point as amplitudes of the quantum state, adopted by [2]. Another straight forward encoding method is to encode one classical bit of information into a qubit. The advantage of amplitude encoding is its compactness to represent real numbers.

Let  $N = 2^n$  and  $X = (x_0, x_1 \dots, x_{N-1}) \in R^N$  is a unit vector. A dataset of size  $M$  can be denoted by  $D = \{(X^0, y^0), (X^1, y^1), (X^2, y^2), \dots, (X^{M-1}, y^{M-1})\}$ , where  $X^m \in R^N$  represents a data point and  $y^m$  represents its class label. For a binary classification,  $y^m \in \{0, 1\}$ . The amplitude encoding of  $X$  is  $|\psi_X\rangle = \sum_{i=0}^{N-1} x_i |i\rangle$ . The first step of the quantum classifier is, through some quantum operations (see the quantum circuit in Figure 1), to transform the dataset  $D$  together with a test point as:

$$|D\rangle = \frac{1}{Z_1} \sum_{m=0}^{M-1} |m\rangle (|0\rangle |\psi_{\tilde{X}}\rangle + |1\rangle |\psi_{X^m}\rangle) |y^m\rangle \quad (1)$$

where  $\tilde{X}$  is a test point,  $X^m$  are training points,  $m$  is an index for the data point  $X^m$ , and  $Z_1$  is a normalization constant to make the whole quantum state of length one. After applying the Hadamard gate to the data qubit that contains  $\tilde{X}$  and  $X^m$ , the state becomes



**Figure 1.** An implementation of the quantum circuit from [2] in IBM’s quantum composer. The first two H gates prepare the ancilla qubit  $q[0]$  and index qubit  $m$   $q[1]$  in superposition. The cU3 gate entangles a test data point with the ancilla qubit  $q[0]$ . The ccX gate entangles the first training data point with  $q[0]$  and  $q[1]$ . The ccU3 gate adds the second training data point to the circuit. Lastly the swap operation is used to add the class label to the circuit through a cnot gate due to the star topology of IBM’s 5Q quantum processor.

$$|D_1\rangle = \frac{1}{Z_1} \sum_{m=0}^{M-1} |m\rangle (|0\rangle |\psi_{\tilde{x}+X^m}\rangle + |1\rangle |\psi_{\tilde{x}-X^m}\rangle) |y^m\rangle \quad (2)$$

here  $\psi_{\tilde{x} \pm X^m} = |\psi_{\tilde{x}}\rangle \pm |\psi_{X^m}\rangle$ . The key observation is

$$\frac{1}{Z_2} \sum_{y^m=0} |\tilde{X} + X^m|^2 = 1 - \frac{1}{Z_2} \sum_{y^m=0} |\tilde{X} - X^m|^2$$

where  $Z_2$  is another normalization constant, which implies that the closer the test point  $\tilde{X}$  to the training point  $X^m$ , the larger the amplitudes in front of  $y^m$  and therefore the more likely to be observed at the last step of measurements. In another word, the probability of measuring the class qubit  $y^m$  in state  $|0\rangle$  is:

$$P(\tilde{y} = 0) = \frac{1}{Z_2} \sum_{y^m=0} |\tilde{X} + X^m|^2 = 1 - \frac{1}{Z_2} \sum_{y^m=0} |\tilde{X} - X^m|^2 \quad (3)$$

By exploiting superposition, entanglement, and interference, this algorithm is able to evaluate the distance of the test point to all the training points at once. And more importantly, through clever mathematical manipulation, the probability of the test point belonging to class 0 or 1 can be determined at the end of the quantum circuit. However, when measuring the ancilla qubit (the qubit containing  $|0\rangle$  and  $|1\rangle$  in Equations (1) and (2)), we may see  $|1\rangle$  instead of  $|0\rangle$ . Therefore the final tally of the counts for  $P(0)$  or  $P(1)$  has to exclude the results of seeing  $|1\rangle$  in the ancilla qubit, which is way different from classical machine learning algorithms.

In this circuit in **Figure 1**, the first two Hadamard gates are employed to super position the qubits  $q[0]$  and  $q[1]$ . Entanglement of the qubits is seen among the controlled gates while the interference of the qubits is seen at the last Hadamard gate. According to [2],  $q[0]$  is an ancilla qubit,  $q[1]$  is index for each training da-

ta point,  $q[2]$  is data, and  $q[3]$  is for class label.  $q[4]$  is an temporary qubit that we use to build the ccU3 gate. Note that a swap operation is used at the end of this circuit to satisfy the star topology employed in IBM's 5Q quantum chip, see **Figure 1**.

A typical quantum circuit is composed of initial states, followed by several unitary operators, and terminated by a few measurements according to the task of the algorithm. During execution of the algorithm, the super positioned quantum states have to maintain their coherent superposition. As a result, the length of a quantum circuit depends directly on the amount of time that these quantum states can remain coherent [14].

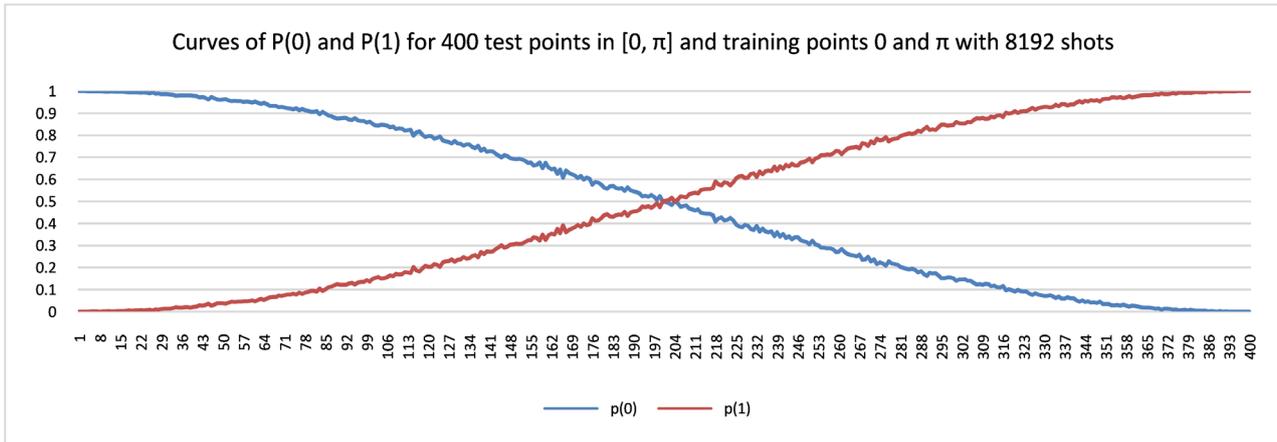
### 3. Results

The quantum classifier in [2] uses amplitude encoding technique, so the classical data points that we will use in our experiments are all of the form  $(x_1, x_2) = (\sin\theta, \cos\theta)$ , to make the notation easier to follow we simply use  $\theta$  to represent a point. Say, we can use  $\theta = \frac{\pi}{4}$  to represent the point of  $(\sin\theta, \cos\theta) = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$ . Since the classifier is distance based, we like to investigate experimentally how the distance of the data points affect its performance. The classifier is run several times with 1000 or 8192 shots, where 8192 is the maximal number of shots allowed on Quantum Experience. We use IBM's quantum simulator in this work rather than the real 5Q quantum computer because part of our goal is to extend the binary classifier in [2] to a multiclass classifier that uses more than five qubits. We summarize our findings in the following three subsections.

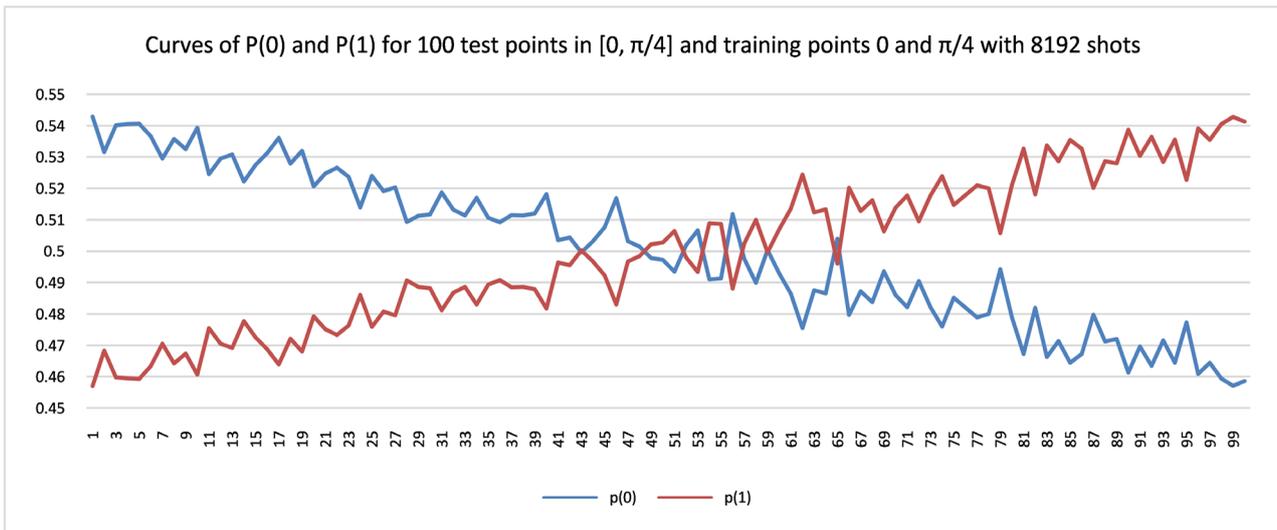
#### 3.1. Experiments on a Binary Classifier

Our first task is to test the binary classifier in [2], we set the two training data points to be  $\{0, \theta\}$  and label point 0 as class 0 and point  $\theta$  as class 1. Then select the evenly spaced points in the interval  $[0, \theta]$  as the test points.  $\theta$  can take values like  $\pi$ , or  $\pi/4$ . Since this is a binary classifier,  $P(0) + P(1) = 1$ . Our first experiment indicates that  $P(0)$  and  $P(1)$  can take values in the full range of  $[0, 1]$ , see **Figure 2**, where  $\theta = \pi$ . As the test point moves from the training point 0 (class label 0) to another training point  $\pi$  (class label 1),  $P(0)$  changes from 1 to 0 gradually. Similarly,  $P(1)$  changes from 0 to 1. The angular distance of the two training points 0 and  $\pi$  is  $\pi$ .

In the next two experiments, we choose  $\theta = \pi/4$  and  $\theta = \pi/16$ . As expected when the distance of the two training points becomes smaller, the binary classification task gets harder which can be seen from the values of  $P(0)$  and  $P(1)$  in **Figure 3** and **Figure 4**. The purpose of these experiments is to visualize the changes of  $P(0)$  and  $P(1)$  of the test points when the distances of the training points get smaller.



**Figure 2.** Curves of  $P(0)$  and  $P(1)$  for 400 test points in  $[0, \pi]$  and two training points  $0$  and  $\pi$  with 8192 shots. The values of  $P(0)$  and  $P(1)$  can take values from the full range of  $[0, 1]$ . The angular distance of  $0$  and  $\pi$  is  $\pi$ .



**Figure 3.** Curves of  $P(0)$  and  $P(1)$  for 100 test points in  $[0, \pi/4]$  and two training points  $0$  and  $\pi/4$  with 8192 shots. The angular distance of the two training points is  $\pi/4$  which is smaller than that in **Figure 2**, so the values of  $P(0)$  and  $P(1)$  cannot take the full range of  $[0, 1]$  but within the range of  $[0.46, 0.54]$  approximately.

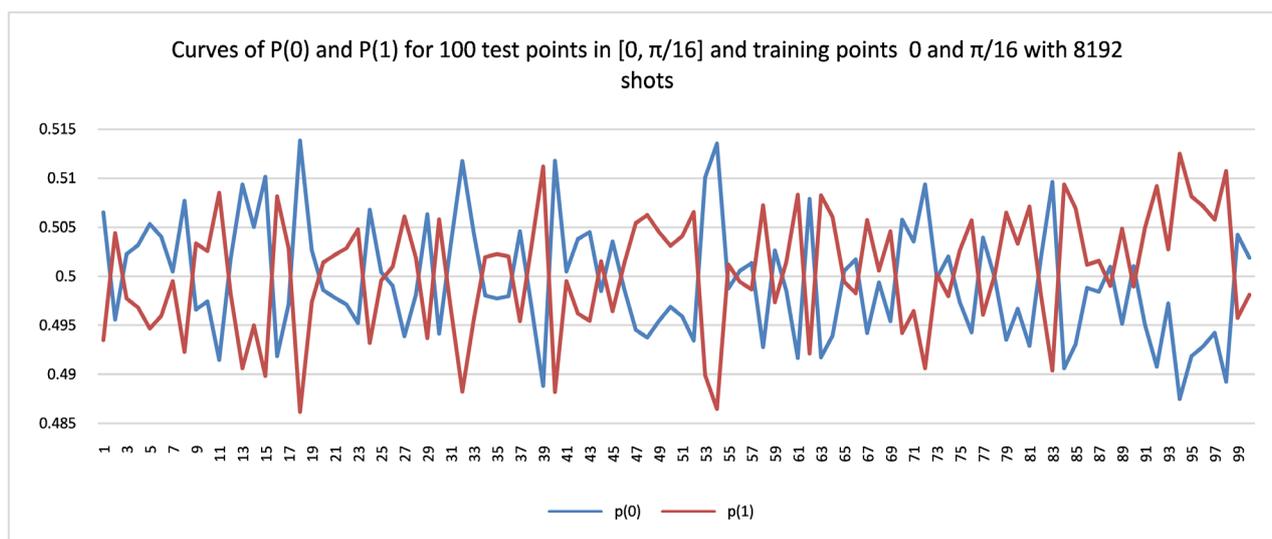
As the test point moves from  $0$  (class label  $0$ ) to  $\pi/4$  (class label  $1$ ),  $P(0)$  changes from about  $0.54$  to about  $0.46$  gradually. Similarly,  $P(1)$  changes from  $0.46$  to  $0.54$ . When the test point gets into the middle of the interval  $[0, \pi/4]$ , the classification problem becomes harder for a distance based classifier. Note that  $P(0)$  and  $P(1)$  do not take values near  $1$  and  $0$  as the distances between  $0$  and  $\pi/4$  is smaller than that of  $0$  and  $\pi$ , compared with **Figure 2**.

In this plot, the distance between  $0$  and  $\pi/16$  is even smaller than that of  $0$  and  $\pi/4$ , in comparison to **Figure 3**.  $P(0)$  and  $P(1)$  take values near  $0.5$  and far away from  $1$  and  $0$ , making this classification problem harder than that in **Figure 2** and **Figure 3**. The three experiments here clearly visualize the fact that the values of  $P(0)$  and  $P(1)$  depend on the distances between the test points and training points, and the training points themselves, and the stochastic nature of

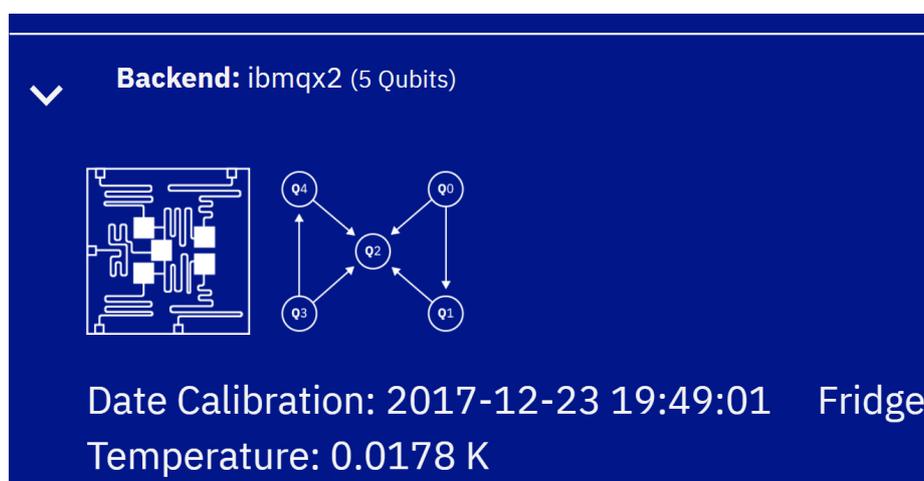
quantum computing caused by the genuinely probabilistic properties of quantum states.

### 3.2. Experiments on a Binary Classifier with or without Swap Operation

The IBM 5Q consists of five superconducting transmon qubits patterned on a silicon substrate. The qubits are labeled Q0, Q1, Q2, Q3, and Q4. For the IBM 5Q system, only 4 different types of CNOT gates are available: every CNOT must have Q2 as the target qubit and Q0, Q1, Q3, or Q4 as the control qubit, see **Figure 5**. The SWAP gate is routinely used to deal the hardware constraint that the CNOT gate cannot be applied to two arbitrary qubits.



**Figure 4.** Curves of  $P(0)$  and  $P(1)$  for 100 test points in  $[0, \pi/16]$  and two training points  $0$  and  $\pi/16$  with 8192 shots. This time the angular distance between the two training points is  $\pi/16$ , much smaller than  $\pi$ . As a result, the values of  $P(0)$  and  $P(1)$  are oscillating near 0.5, far away from 1 and 0.



**Figure 5.** A schematic demonstration of the five qubits in IBM's 5Q chip, an image taken from <https://quantumexperience.ng.bluemix.net/qx/editor>

To test the effects of the swap operation as shown in **Figure 1**, we create another circuit where the SWAP gate and CNOT gate in the original circuit are replaced by a Toffoli gate so they are logically equivalent but the latter involves no swap. We run 10 experiments for these two classifiers, each of which is made of 8192 shots for 200 test points and two training points, see **Figure 6**. From these experiments, the average error rate of the circuit with the swap is 20.55% and the one without swap is 22.05%. Our results imply no significant difference in the classification accuracy, given the stochastic nature of these two classifiers. We intentionally choose two close training points 0 and  $\pi/8$  to make it hard for the distance based classifiers to differentiate them. To further elucidate the observed results, we select one experiment for each classifier from the 10 experiments and plot the whole curves of  $P(0)$  and  $P(1)$  over all the 200 test points, see **Figure 7** and **Figure 8**. They show what the curves of  $P(0)$  and  $P(1)$  look like when the classification error rate is around 20%.

### 3.3. Experiments on a Multiclass Classifier

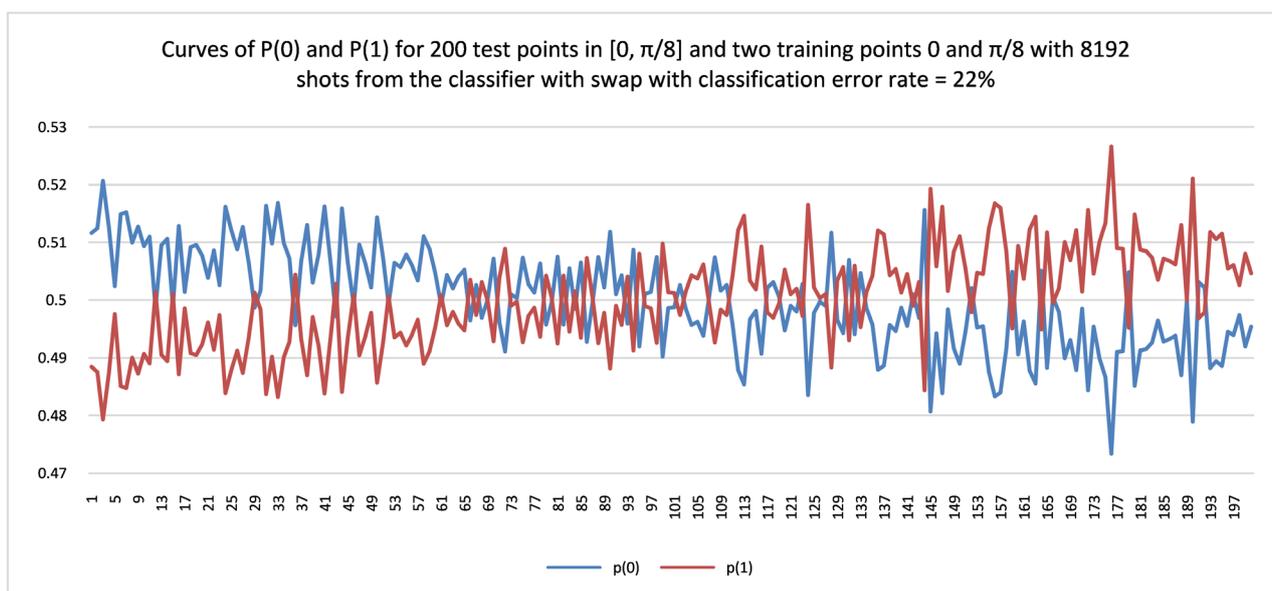
To make a multiclass classifier, four classes to be exact in this study, we use two qubits for index and two qubits for classes following the circuit patterns shown in **Figure 1**. In the following experiment, we set the four training data points to be  $\{0, \pi/2, \pi, 3\pi/2\}$  with class labels 0, 1, 2, and 3 respectively and then select the evenly spaced points in the interval  $[0, 2\pi]$  as the test points. Since we have four classes,  $P(0) + P(1) + P(2) + P(3) = 1$ . The idea is to create the test points so they can move from 0 to  $2\pi$ , as a result these four probabilities can display their four peaks in turn as shown in **Figure 9**.

## 4. Conclusions

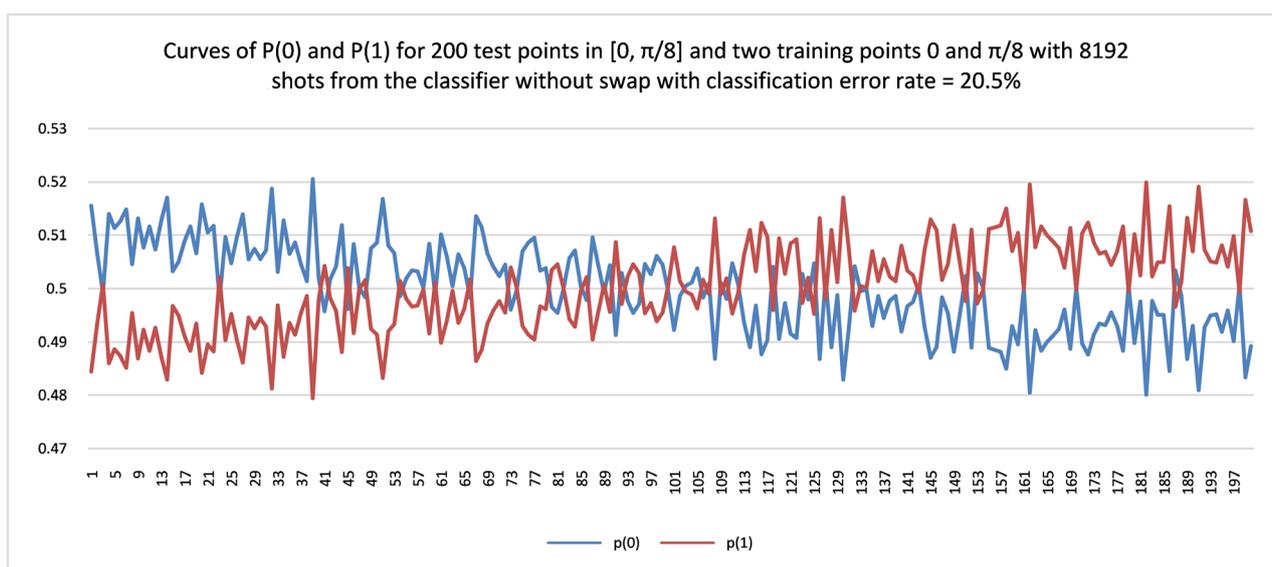
Quantum machine learning, though in its initial stage, has demonstrated its



**Figure 6.** Classification error rate from 10 experiments for two classifiers with and without swap operation, each of which is made of 200 test points in  $[0, \pi/8]$  and two training points 0 and  $\pi/8$  with 8192 shots. The average classification error rate for the classifier with swap is 20.55% and the one without is 22.05%.

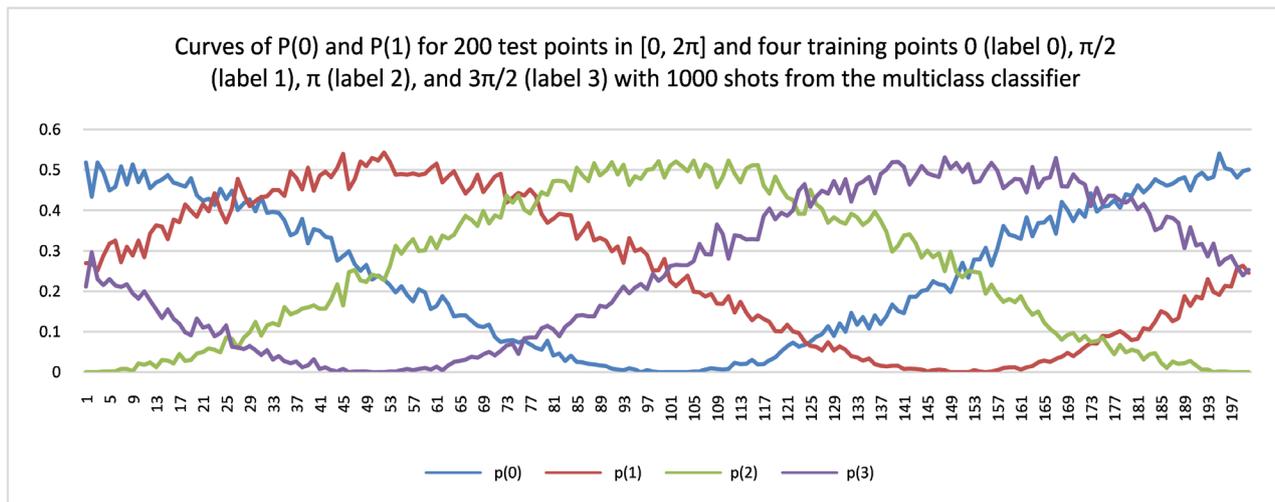


**Figure 7.** One selected experiment for the classifier with swap from the 10 experiments reported in **Figure 6**.



**Figure 8.** One selected experiment for the classifier without swap from the 10 experiments reported in **Figure 6**.

potential to speed up some of the costly machine learning calculations when compared to the existing classical approaches. This study presents an empirical analysis of the distance based quantum classifier created in [2]. Our work consists of three components. The first is to visualize the probabilities of observing a test data point being in class 0 or 1 in a binary classification as it moves among the training points. Although it is obvious the prediction depends on the distance of the data points implied by the nature of this kind of algorithms. Since it is a quantum classifier so it might be helpful if some experiments can be designed to reveal the quantum nature of this classifier. Our analysis shows that when the training points are close enough, the classifier has hard time to



**Figure 9.** Curves of  $P(0)$  and  $P(1)$  for 200 test points in  $[0, 2\pi]$  and four training points  $0$  (label 0),  $\pi/2$  (label 1),  $\pi$  (label 2), and  $3\pi/2$  (label 3) with 1000 shots from the multiclass classifier created in this study.

differentiate the training points. The values of  $P(0)$  and  $P(1)$  are very close to 0.5 and oscillating about 0.5. Because of the stochastic nature of quantum algorithms, this phenomena makes prediction more difficult than the corresponding classical algorithms.

The second is to see if the swap operation that is commonly utilized in quantum circuit design to meet the hardware constraints may have an impact on the performance of this classifier. A typical circuit of a quantum machine learning algorithm employs superposition, entanglement, and interference of qubits. Because of this, any swap of two qubits in the circuit may influence the final outcome of the algorithm. Our experiments suggest that the swap operation does not make a difference in the cases that we have tested.

Our final goal is to extend the binary classifier in [2] to a multiclass classifier to further demonstrate the capability of this classifier. The four-class classifier that we create in this work shows its ability to distinguish the four different training points with four peaks of the curves of  $P(0)$ ,  $P(1)$ ,  $P(2)$ , and  $P(3)$  occurring at the right test points.

It is known that some quantum machine learning algorithms can outperform their classical counterparts, but the full scope of their potential needs to be further investigated. Our work adds new knowledge to this field by helping learn the working of this quantum classifier, fulfilling part of the big goal of understanding the value of quantum computing to machine learning and AI.

## References

- [1] IBM. Quantum Experience. [www.research.ibm.com/quantum](http://www.research.ibm.com/quantum)
- [2] Schuld, M., Fingerhuth, M. and Petruccione, F. (2017) Implementing a Distance-Based Classifier with a Quantum Interference Circuit. *A Letters Journal of Exploring the Frontiers of Physics*, **119**, Article ID: 60002.
- [3] Schuld, M., Sinayskiy, I. and Petruccione, F. (2016) Prediction by Linear Regression

- on a Quantum Computer. *Physical Review A*, **94**, Article ID: 022342.  
<https://doi.org/10.1103/PhysRevA.94.022342>
- [4] Schuld, M., Sinayskiy, I. and Petruccione, F. (2015) Simulating a Perceptron on a Quantum Computer. *Physics Letters A*, **379**, 660-663.  
<https://doi.org/10.1016/j.physleta.2014.11.061>
- [5] Schuld, M., Sinayskiy, I. and Petruccione, F. (2014) Quantum Computing for Pattern Classification. *Pacific Rim International Conference on Artificial Intelligence*, Gold Coast, 1-5 December 2014, 208-220.
- [6] Schuld, M., Sinayskiy, I. and Petruccione, F. (2014) The Quest for a Quantum Neural Network. *Quantum Information Processing*, **13**, 2567-2586.  
<https://doi.org/10.1007/s11128-014-0809-8>
- [7] Schuld, M., Sinayskiy, I. and Petruccione, F. (2014) Quantum Walks on Graphs Representing the Firing Patterns of a Quantum Neural Network. *Physical Review A*, **89**, Article ID: 032333. <https://doi.org/10.1103/PhysRevA.89.032333>
- [8] Cai, X.-D., Wu, D., Su, Z.-E., Chen, M.-C., Wang, X.-L., Li, L., Liu, N.-L., Lu, C.-Y. and Pan, J.-W. (2015) Entanglement-Based Machine Learning on a Quantum Computer. *Physical Review Letters*, **114**, Article ID: 110504.  
<https://doi.org/10.1103/PhysRevLett.114.110504>
- [9] Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N. and Lloyd, S. (2017) Quantum Machine Learning. *Nature*, **549**, 195-202.  
<https://doi.org/10.1038/nature23474>
- [10] Dunjko, V., Taylor, J.M. and Briegel, H.J. (2016) Quantum-Enhanced Machine Learning. *Physical Review Letters*, **117**, Article ID: 130501.  
<https://doi.org/10.1103/PhysRevLett.117.130501>
- [11] Alsina, D. and Latorre, J.I. (2016) Experimental Test of Mermin Inequalities on a Five-Qubit Quantum Computer. *Physical Review A*, **94**, Article ID: 012314.  
<https://doi.org/10.1103/PhysRevA.94.012314>
- [12] Berta, M., Wehner, S. and Wilde, M.M. (2016) Entropic Uncertainty and Measurement Reversibility. *New Journal of Physics*, **18**, Article ID: 073004.  
<https://doi.org/10.1088/1367-2630/18/7/073004>
- [13] Hu, W. (2018) Empirical Analysis of Decision Making of an AI Agent on IBM's 5Q Quantum Computer. *Natural Science*, **10**, 45-58.  
<https://doi.org/10.4236/ns.2018.101004>
- [14] Nielsen, M.A. and Chuang, I.L. (2010) Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.  
<https://doi.org/10.1017/CBO9780511976667>