A Quantum Binary Classifier based on Cosine Similarity

# A Quantum Binary Classifier based on Cosine Similarity

1st Davide Pastorello
*Dept. of Information Engineering and Computer Science*
*University of Trento, Via Sommarive 9, 38123 Povo (TN), Italy*
*TIFPA-INFN, Via Sommarive 14, 38123 Povo (Trento), Italy*
d.pastorello@unitn.it

2nd Enrico Blanzieri
*Dept. of Information Engineering and Computer Science*
*University of Trento, Via Sommarive 9, 38123 Povo (Trento), Italy*
*TIFPA-INFN, Via Sommarive 14, 38123 Povo (Trento), Italy*
enrico.blanzieri@unitn.it

***Abstract*—This proposal introduces the quantum implementation of a binary classifier based on cosine similarity between data vectors. The proposed quantum algorithm presents time complexity that is logarithmic in the product of the training set cardinality and the dimension of the vectors. It is based just on a suitable state preparation like the retrieval from a QRAM, a SWAP test circuit, and a measurement process on a single qubit. An implementation on an IBM quantum processor is presented.**

***Index Terms*—Quantum algorithms, quantum machine learning, binary classification, cosine similarity.**

QUANTUM machine learning is a rapidly emerging research area where quantum computing techniques are applied to machine learning tasks in order to pursue a computational speedup in performances. Within the supervised learning scenario, some interesting quantum classification algorithms have been proposed in recent years [1]–[3]. Here is a quantum algorithm that implements a model, based on cosine similarity, for binary classification of data vectors.

Let $X = \{\mathbf{x}_i, y_i\}_{i=0,\dots,N-1}$, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$ $\forall i \in \{0, ..., N-1\}$, be a training set of $N$ data instances with two-valued labels that are represented in a real feature space of dimension $d$. Let $\mathbf{x} \in \mathbb{R}^d$ be a new data instance to be classified as either 1 or $-1$. The considered classification model for the quantum implementation is defined as follows:

$$y(\mathbf{x}) := \operatorname{sgn}\left(\sum_{i=0}^{N-1} y_i \cos(\mathbf{x}_i, \mathbf{x})\right), \tag{1}$$

where *cosine similarity* is defined by:

$$\cos(\mathbf{x}, \mathbf{y}) := \frac{\mathbf{x} \cdot \mathbf{y}}{\parallel \mathbf{x} \parallel \parallel \mathbf{y} \parallel} \qquad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d. \tag{2}$$

A typical example where cosine similarity is adopted for classification and clustering is text analysis [4]–[6]. In the model (1), any training vector contributes to the prediction of the new label and such a contribution is weighted by the cosine similarity with the new instance. On the one hand, time complexity of the classical calculation of the new label is $O(Nd)$. On the other hand, assuming $d = 2^n$ for $n \in \mathbb{N}$ without loss of generality, the data vector $\mathbf{x}_i \in \mathbb{R}^d$ can be encoded in the amplitudes of a quantum state of $n$ qubits:

$$|\mathbf{x}_i\rangle = \frac{1}{\parallel \mathbf{x}_i \parallel} \sum_{j=0}^{d-1} x_{ij} |j\rangle \in \mathsf{H}_n, \tag{3}$$

where $|j\rangle$ is an orthonormal basis of the $n$-qubit Hilbert space $\mathsf{H}_n \simeq (\mathbb{C}^2)^{\otimes n}$ and $x_{ij}$ is the $j$th component of $\mathbf{x}_i$. Within the amplitude encoding, the correspondence $\cos(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x} | \mathbf{y} \rangle$ holds. One of the key feature of quantum machine learning is the efficient retrieval of data encoded into quantum states, e.g. a quantum random access memory (QRAM) allows to retrieve data in parallel. Assuming that the real components $\{x_{ij}\}_{j=0,\dots,d-1}$ of $\mathbf{x}_i$ are stored in an array of memory cells as floating point numbers and the norm $\parallel \mathbf{x}_i \parallel$ is given separately, the retrieval of the state (3) can be done in $O(\log d)$ steps according to the *bucket brigade architecture* [7]. Consider also a $\log N$-qubit register, with Hilbert space $\mathsf{H}_{index} \simeq (\mathbb{C}^2)^{\otimes \log N}$, to encode the indexes of training data vectors and construct the state:

$$|X\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle |\mathbf{x}_i\rangle |b_i\rangle \in \mathsf{H}_{index} \otimes \mathsf{H}_n \otimes \mathsf{H}_l, \tag{4}$$

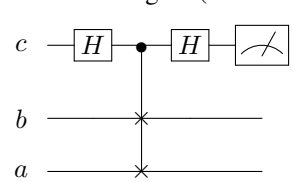
where $\mathsf{H}_l$ is the Hilbert space of a single qubit used for encoding the values of the labels with $b_i = \frac{1-y_i}{2} \in \{0, 1\}$, then $|b_i\rangle$ is an eigenstate of the Pauli matrix $\sigma_z$ with eigenvalue $y_i$. Moreover, in the same registers construct the state:

$$|\psi_{\mathbf{x}}\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle |\mathbf{x}\rangle |-\rangle \in \mathsf{H}_{index} \otimes \mathsf{H}_n \otimes \mathsf{H}_l, \tag{5}$$

where the label qubit is in the state $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, so the new data vector $\mathbf{x}$ is represented in a quantum superposition of the two possible classes. Now consider an ancillary qubit, called qubit $a$, and prepare the state:

$$\frac{1}{\sqrt{2}} \left(|X\rangle |0\rangle + |\psi_{\mathbf{x}}\rangle |1\rangle\right) \in \mathsf{H}_{index} \otimes \mathsf{H}_n \otimes \mathsf{H}_l \otimes \mathsf{H}_a, \tag{6}$$

that can be retrieved from the QRAM in time $O(\log(Nd))$. Now perform the SWAP test [8] between a second ancillary qubit, called qubit $b$, prepared in $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and the qubit $a$, moreover consider another qubit, say $c$, prepared in $|0\rangle$ to control the Fredkin gate (i.e. the controlled swap gate):




(7)

A straightforward calculation shows that after the action of (7) the probability to obtain the outcome 1 measuring the qubit $c$ is $\mathbb{P}(1) = \frac{1}{4}(1 - \langle X|\psi_{\mathbf{x}}\rangle)$ that is directly related to (1), in fact $\langle X|\psi_{\mathbf{x}}\rangle = \frac{1}{N\sqrt{2}} \sum_{i=0}^{N-1} y_i \cos(\mathbf{x}_i, \mathbf{x})$. Therefore the probability $\mathbb{P}(1)$ is related to the prediction of the label of $\mathbf{x}$, according to the model (1), by means of:

$$y(\mathbf{x}) = \operatorname{sgn}\left[1 - 4\,\mathbb{P}(1)\right]. \tag{8}$$

The procedure *preparation+test* must be repeated several times for sampling the qubit $c$ to estimate $\mathbb{P}(1)$ as the success probability of a Bernoulli trial. An estimation within an error $\epsilon$ requires a number of repetitions growing as $O(\epsilon^{-2})$ as provided by the binomial proportion confidence interval, so the overall time complexity of Algorithm 1 is $O(\epsilon^{-2} \log(Nd))$.

**Input:** training set $X = \{\mathbf{x}_i, y_i\}_{i=0,\dots,N-1}$, unclassified instance $\mathbf{x}$.
**Result:** label $y$ of $\mathbf{x}$.
1 **repeat**
2 initialize the register $\mathsf{H}_{index} \otimes \mathsf{H}_n \otimes \mathsf{H}_l$ and an ancillary qubit $a$ in the state (6);
3 initialize a qubit $b$ in the state $|-\rangle$;
4 perform the SWAP test on $a$ and $b$ with control qubit $c$ prepared in $|0\rangle$; *% circuit (7)*
5 measure qubit $c$;
6 **until** *desired accuracy on the estimation of* $\mathbb{P}(1)$;
7 Estimate $\mathbb{P}(1)$ as the relative frequency $\hat{\mathbb{P}}$ of outcome 1;
8 **if** $\hat{\mathbb{P}} > 0.25$ **then**
9 **return** $y = -1$
10 **else**
11 **return** $y = 1$
12 **end**

**Algorithm 1:** *Quantum implementation of the model (1).*

*Example for $N = 2$ and $d = 2$*: Consider a training set of two-dimensional data instances given by $X = \{(\mathbf{x}_0, y_0), (\mathbf{x}_1, y_1)\}$ where $\mathbf{x}_0 = (1, 0)$, $y_0 = 1$ and $\mathbf{x}_1 = (0.718, 0.696)$, $y_1 = -1$. Let $\mathbf{x} = (0.884, 0.468)$ be the unlabelled data instance to classify. The model (1) predicts the label of $\mathbf{x}$ as a *nearest neighbor* then it returns $y = -1$.

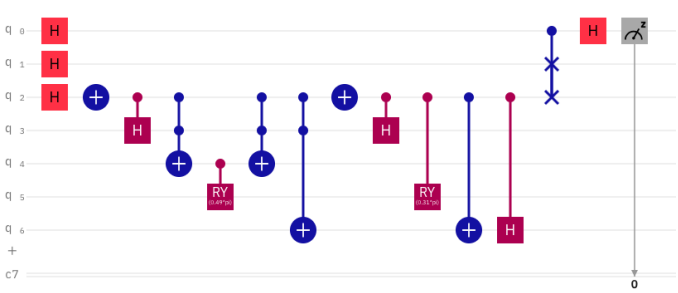



Fig. 1. Quantum circuit implementing lines 1-7 of Algorithm 1 in the *IBM Quantum Composer* where $N = 2$, $d = 2$, with training set $\{((1, 0), 1), ((0.718, 0.696), -1)\}$ and test $\mathbf{x} = (0.884, 0.468)$.

In Fig. 1, there is the circuit, represented in the *IBM Quantum Composer*, implementing Algorithm 1 for the example just introduced. The qubits $q_0, q_1, q_2$ are the ancillas used for the SWAP test, $q_3$ is the 1-qubit index register, $q_4$ is an additional ancillary qubit necessary to control the gate $RY_{(0.49\pi)}$, $q_5$ is the 1-qubit register for the amplitude encoding of data and $q_6$ is the qubit encoding the labels.

The run of the algorithm on the IBM quantum processor *ibmq_16_melbourne* provides 1024 shots for sampling the qubit $q_0$. The obtained estimation of $\mathbb{P}(1)$ is $\hat{\mathbb{P}} = 490/1024 \simeq 0.48$ then the label assigned to $\mathbf{x} = (0.884, 0.468)$ is $y = -1$ as expected. Despite the correct classification in this test, a comparison with the result of the simulator *ibm_qasm_simulator* suggests that the considered quantum machine is too noisy for a good classification by means of Algorithm 1. The output statistic of the simulator provides $\hat{\mathbb{P}} = 273/1024 \simeq 0.27$. This result is consistent with the fact that the unclassified data vector $\mathbf{x}$ is close to the intermediate point between the training vectors. Repeating the experiment with the same training points and the new unlabelled instance $\mathbf{x} = (0.951, 0.309)$ (whose correct classification is $y = 1$), the quantum machine fails. In fact ibmq_16_melbourne returns the relative frequency $\hat{\mathbb{P}} = 338/1024 \simeq 0.38$, so it classifies $\mathbf{x}$ as $y = -1$. On the same test, the simulator ibm_qasm_simulator returns $\hat{\mathbb{P}} = 244/1024 \simeq 0.24$ classifying correctly. The observed lack of accuracy in classification depends on the low *quantum volume*[1] (QV = 8) of the considered quantum processor. Matter for future work may be the test on larger and more reliable hardware (e.g. the IBM quantum machine *ibmq_montreal* with 27 qubits and QV=128).

The exponential speedup of the presented quantum classifier is due to the efficient preparation of quantum states in logarithmic time and to the classification itself performed in constant time (that depends on the desired accuracy). Indeed, the choice of the QRAM is motivated by an explicit estimation of the overall time complexity but alternative efficient initializations are allowed to run this quantum classifier.

## Acknowledgment

This work is supported by the Q@TN consortium. The authors are grateful to E. Zardini for useful comments.

## References

[1] M. Schuld, F. Petruccione *Supervised learning with quantum computers* Springer International Publishing 2018
[2] P. Rebentrost, M. Mohseni, S. Lloyd *Quantum support vector machine for big data classification* Phys. Rev. Lett. 113, 130503 (2014)
[3] S. Lloyd, Mohseni, M., Rebentrost, P. *Quantum algorithms for supervised and unsupervised machine learning*. Preprint at https://arxiv.org/abs/1307.0411 (2013)
[4] K. Hornik, I. Feinerer, M. Kober, C. Buchta *Spherial k-means clustering* Journal of Statistical Software vol. 50, n. 10 (2012)
[5] S. Al-Ananzi, H. AlMahmoud, I. Al-Turaiki *Finding similar documents using different clustering techniques* Procedia Computer Science 82 (2016)
[6] C. Focil-Arias, J. Ziiniga, G. Sidorov, I. Batyrshin, A. Gelbukh *A tweets classifier based on cosine similarity* CEUR Workshop Proceedings (2017)
[7] V. Giovannetti, S. Lloyd, L. Maccone *Quantum random access memory* Phys. Rev. Lett. 100, 160501 (2008)
[8] H. Buhrman, R. Cleve, J. Watrous, R. de Wolf *Quantum Fingerprinting* Phys. Rev. Lett. 87, 167902 (2001)
[9] N. Moll et al. *Quantum optimization using variational algorithms on near-term quantum devices* Quantum Sci. Technol. 3 030503 (2018)

[1]Quantum volume has been defined in [9] as a metric to measure the capability and robustness of a quantum computer.