Quantum Algorithm for Ensemble Learning

Antonio Macaluso, Stefano Lodi, and Claudio Sartori

Dept. of Computer Science and Engineering, University of Bologna
{antonio.macaluso2, stefano.lodi, claudio.sartori}@unibo.it

Abstract. The idea of ensemble learning is to build a prediction model by combining the strengths of a collection of simpler base models. Although they are extensively used, ensemble methods have high requirements in terms of memory and computational time.
In this work, we propose a quantum algorithm that allows reproducing ensemble classification using bagging strategy. The algorithm generates many sub-samples in superposition, in such a way that only a single execution of a quantum classifier is required. In particular, the entanglement between a quantum register and different training sub-samples in superposition allows obtaining a sum of individual results which gives rise to the ensemble prediction. When considering the overall temporal cost of the algorithm, the single base classifier impacts additively rather than multiplicatively, as it usually happens in ensemble framework. Furthermore, given that the number of base models scales exponentially with the number of qubits of the control register, our algorithm opens up the possibility of exponential speed-up for quantum ensemble.

Keywords: Quantum Computing · Machine Learning · Ensemble Methods.

1 Background

Thanks to the quantum mechanical principles of superposition and entanglement, Quantum Computing (QC) can achieve vast amounts of parallelism [8] without the need for the multiple replications of hardware that are usually required in a classical computer. One of the most important fields in which QC promises to make an impact in the future is machine learning (ML). However, being an entirely new field, Quantum Machine Learning (QML) poses many open challenges [1].

The idea of a quantum ensemble based on Bayesian Model Averaging (BMA) is investigated in [9], but BMA approach is not very used in ML because of its limited performance in real-world applications [7].

In general, the idea of ensemble learning is to build a prediction model by combining the strengths of a collection of simpler base models to reduce the Expected Prediction Error [6]. One of the most popular schemas to build an ensemble is Bagging, in which a committee of independent weak classifiers cast a vote for the predicted class. It is the basis of the well-known method Random Forest [2] and constructs a homogeneous ensemble by applying the same learning algorithm under different training conditions. In practice, we compute $f_1(x), \ldots, f_B(x)$ using B separate training sets and average them to obtain a single low-variance model:

$$f_{bag}(x) = \frac{1}{B} \sum_{b=1}^{B} f_b(x).$$

It turns out that Bagging produces a combined model that outperforms the single model built from all training data, and is never substantially worse [4].

2 Contribution

In this work, we provide a quantum algorithm to perform ensemble classification using bagging. The idea is to generate different sub-samples of the training set in superposition, each entangled with a quantum state of a control register. Thus, a quantum classifier C is applied to obtain a large number of classifications in superposition. The proposed algorithm limits the number of state preparation routines and makes the evaluation of large ensemble feasible with small circuits.

3 Quantum Algorithm for Bagging Ensemble

The quantum algorithm for ensemble involves 5 quantum registers:

$$|\Phi_0\rangle = \underset{control}{|0\rangle} \otimes \underset{training}{|0\rangle} \otimes \underset{temp}{|0\rangle} \otimes \underset{test}{|0\rangle} \otimes \underset{target}{|0\rangle} \tag{1}$$

where the size of the *control* register determines the size of the ensemble (i.e. the number of base models), the size of *training* and *test* registers depend on the amount of data. The *temp* register determines the size of the sub-sample used as input in a single base model. Finally, the number of qubits in the *target* register depends on the nature of the target variable. Starting from these five registers, the algorithm involves four steps:

(Step 1) The *state preparation* consists in encoding the training and test set into their respective registers. Also, the *control* register is initialised into uniform superposition:

$$\begin{aligned} |\Phi_1\rangle &= \left(H^{\otimes d} \otimes S_{(x,y)} \otimes \mathbb{1} \otimes S_{(\tilde{x})} \otimes \mathbb{1}\right) |\Phi_0\rangle = \\ &= \left(\frac{1}{\sqrt{2^d}} \sum_{i=1}^{2^d} |i\rangle \right) \otimes |x, y\rangle \otimes |0\rangle \otimes |\tilde{x}\rangle \otimes |0\rangle \,, \end{aligned} \tag{2}$$

where $S_{(x,y)}$ and $S_{(\tilde{x})}$ are the unitaries that encode data in quantum states whose form depends on the encoding strategy chosen for data, and $H^{\otimes d}$ is the Walsh-Hadamard gate.

(Step 2) The second step is *sampling in superposition* and consists in generating many altered transformation of the original training data, (x, y), in superposition. We consider a quantum oracle V which entangles several sub-sample of data with the control register:

$$|\Phi_2\rangle = \big(V \otimes \mathbb{1} \otimes \mathbb{1}\big) |\Phi_1\rangle = \left(\frac{1}{\sqrt{2^d}} \sum_{i=1}^{2^d} |i\rangle |x, y\rangle |x_i, y_i\rangle \right) \otimes |\tilde{x}\rangle \otimes |0\rangle \, , \tag{3}$$

where $|x_i, y_i\rangle$ is a random sub-sample of $|x, y\rangle$. After this step, the training register can be in any state, it will not be used further in the computation.

(Step 3) The *classification* step consists in the interaction via interference between the *temp* and *test* registers, to store the estimates of the target variable:

$$|\Phi_3\rangle = \big(\mathbb{1} \otimes \mathbb{1} \otimes C\big) |\Phi_2\rangle = \frac{1}{\sqrt{2^d}} \sum_{i=1}^{2^d} |i\rangle |x, y\rangle |x_i, y_i\rangle |\tilde{x}\rangle |f_i(\tilde{x})\rangle \, , \tag{4}$$

where $f_i(\tilde{x})$ is an estimate of the target variable for $\tilde{x}$ and it depends on the i-th sub-sample and the test set $\tilde{x}$. We refer to C as a quantum oracle that takes as input two sets of data encoded into two different registers, and provides an estimate of the target variable in an additional register.

(Step 4) Finally, the expectation *measurement* on the target qubit provides a sum of expectation values that corresponds to the ensemble prediction:

$$\begin{aligned} \langle M \rangle = & \big\langle \Phi_3 \big| \mathbb{1}^{\otimes d} \otimes \mathbb{1} \otimes \mathbb{1} \otimes \mathbb{1} \otimes M \big| \Phi_3 \big\rangle = \\ = & \frac{1}{2^d} \sum_{i=1}^{2^d} \langle f_i(\tilde{x}) | M | f_i(\tilde{x}) \rangle = \frac{1}{2^d} \sum_{i=0}^{2^d} \langle M_i \rangle = \frac{1}{B} \sum_{b=1}^{B} f_b = f_{bag}\big(\tilde{x} | (x, y)\big) \end{aligned} \tag{5}$$

where M is a measurement operator (e.g. $Pauli$ gate).

As we can see from Equation 5, measuring the *target* qubit and leaving untouched the other quantum registers, we obtain the average of different classifications based on different sub-samples. The separate computation of each base model is not required; indeed, it is only necessary to generate the quantum state expressed in Equation 3 and to execute the classifier C once. This implies, when considering the overall temporal cost of the algorithm, that the single classifier impacts additively rather than multiplicatively, as it usually happens in ensemble framework. In fact, in the classical ensemble, it is necessary to train the same algorithm under B different training conditions, and the overall temporal cost of the algorithm is, at least, the cost of the single classifier times B. In the case of quantum ensemble, the overall temporal cost of the algorithm depends on the generation of many sub-samples in superposition, plus one execution of the quantum classifier that, working via interference, allows propagating the use of the function f to all sub-samples. Furthermore, given a *control* register made up of d qubits, the ensemble size B is equal to 2^d. This, in turn, implies that the ensemble size B scales exponentially with the number of qubits of the control register, opening up the possibility to achieve exponential speed-up with respect to the classical ensemble methods.

4 Experiments

We provide small-scale experiments to show how quantum ensemble works. To the best of our knowledge, there is no quantum classifier which fulfils the requirements of Eq.(4); this problem will be investigated in future works. However, we show that given a unitary C that implements a generic function f, whose output is sensitive to the comparison between two inputs (*train* and *test*), it is possible to leverage the quantum algorithm described in Section 3 to obtain the average of multiple outputs of f, by implementing C only once.

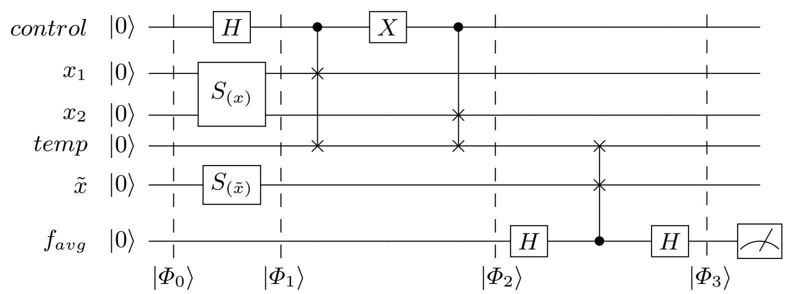

Fig. 1: Quantum circuit for the ensemble of two swap-tests. First, all the registers are initialising according to Equation 2 ($|\Phi_1\rangle$). In the second step, the two states of the *control* register are entangled through controlled-swap operations with the two qubits $|x_1\rangle$ and $|x_2\rangle$, by using the *temp* register ($|\Phi_2\rangle$). Finally, the swap-test is executed in order to obtain the average distance of $\tilde{x}$ from x_1 and x_2. All the details about the implementation are made available at the following GitHub project: https://github.com/amacaluso/Quantum-Algorithm-for-Ensemble-Learning.

We consider two 2-dimensional training vectors encoded in the amplitudes of two different qubits (*training* register) and a single qubit for each of the other registers (*control, temp, test, target*). As quantum gate C, we use the swap test [3] which is a procedure to check how much two quantum states differ. In particular, given two vectors x_i and x_j encoded in the amplitudes of two different qubits ($|x_i\rangle, |x_j\rangle$), the final state before measurement of the swap test is:

$$\frac{1}{2}|0\rangle\big(|x_i\rangle|x_j\rangle + |x_j\rangle|x_i\rangle\big) + \frac{1}{2}|1\rangle\big(|x_i\rangle|x_j\rangle - |x_j\rangle|x_i\rangle\big). \tag{6}$$

Measuring the first qubit of this state produces outcome $|0\rangle$ with probability $(1 + |\langle x_i|x_j\rangle|^2)/2$. This probability is 1 if $x_i = x_j$.

The quantum algorithm for the ensemble of swap tests (Figure 1) entangles the two training vectors with the two quantum states of the *control* qubit. Then, the *temp* qubit is given as input to the swap test, together with the *test* qubit. This procedure allows to obtain the average distance between the test vector and the two training vectors through a single execution of the swap test. To

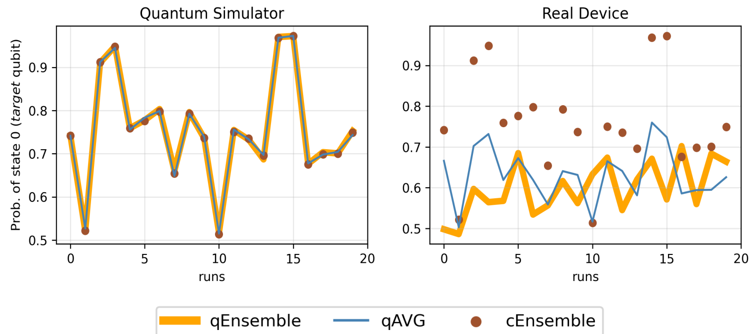

Fig. 2: The plots illustrate the comparison between the quantum ensemble of swap-test (*qEnsemble*), the average of two quantum swap-test executed separately (*qAVG*), and the same average computed classically (*cEnsemble*). Each run corresponds to the generation of a different dataset.

show that the idea of ensemble works, we generated 20 random datasets, each made up of three vectors (x_1, x_2, x_{test}). We implemented the circuit in Figure 1 and then measured the *target* qubit. Results are reported in Figure 2. The plot on the left shows the experiments considering the quantum simulator that assumes a fault-tolerant quantum computer. The agreement between the quantum ensemble (orange line) and the classical ensemble (blue line) is almost perfect; this confirms the possibility to perform quantum ensemble with the advantages described in Section 3. The plot on the right shows the results using a real device (*ibmq_16_melbourne*). In this case, we can see a significant deterioration. This may be due to the depth of the implemented circuit, which seems to be prohibitive considering the actual quantum devices.

5 Conclusion

In this work, we proposed a quantum algorithm for ensemble classification that uses bagging strategy. Besides the theoretical algorithm, we provided small-scale experiments to show how the algorithm works. In particular, we showed that it is possible to produce a quantum ensemble by executing the classification routine only once. The algorithm provides advantages in terms of temporal computational complexity, although it is not complete in its technical formulation.

Future works will be dedicated to design a proper quantum classifier C, which is able to provide different outputs based on different training sets. In fact, the ensemble outperforms the single model only if the outputs of the base models are accurate and diverse [5]. Also, a generalised quantum routine V to generate the sampling in superposition has to be designed.

Although some challenges still remain, we believe that this work may be the first step to show that QML can overcome the limitations of classical ML in the context of ensemble classification.

References

1. Aaronson, S.: Read the fine print. Nature Physics **11**(4), 291 (2015)
2. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001)
3. Buhrman, H., Cleve, R., Watrous, J., de Wolf, R.: Quantum fingerprinting. Phys. Rev. Lett. **87**, 167902 (Sep 2001). https://doi.org/10.1103/PhysRevLett.87.167902, https://link.aps.org/doi/10.1103/PhysRevLett.87.167902
4. Dietterich, T.G.: Ensemble methods in machine learning. In: International workshop on multiple classifier systems. pp. 1–15. Springer (2000)
5. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**(10), 993–1001 (Oct 1990). https://doi.org/10.1109/34.58871
6. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer **27**(2), 83–85 (2005)
7. Minka, T.P.: Bayesian model averaging is not model combination. Available electronically at http://www. stat. cmu. edu/minka/papers/bma. html pp. 1–2 (2000)
8. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2002)
9. Schuld, M., Petruccione, F.: Quantum ensembles of quantum classifiers. Scientific reports **8**(1), 2772 (2018)