A quantum ensemble method for quantum binary classifiers

Emiliano Tolotti
emiliano.tolotti@unitn.it

Enrico Blanzieri
enrico.blanzieri@unitn.it

Davide Pastorello
davide.pastorello3@unibo.it

May 24, 2024

Abstract

Ensemble methods are well-known in machine learning and they aim to enhance prediction accuracy by combining the outputs of multiple models. One important principle of ensemble methods is to ensure diversity among the predictors, allowing each of them to capture different aspects of the data. As a result, the combined models provide a more comprehensive understanding and better prediction capability. In the context of homogeneous ensembles, where the internal models are of the same kind, diversity can be introduced by using different sets of data for each classifier. The principal methods include bagging [1], which considers subsets of training points sampled with replacement, and attribute bagging (random subspaces) that uses random subsets of features [2]. Weighted–average ensembles are a subclass of ensemble methods where certain models are considered to be more accurate than others. These superior models receive higher weights through a weight learning procedure, giving them greater influence on the final prediction. They can be considered as stacking ensembles [3], where the stacked classifier performs the weighted aggregation. By combining the predictions of individual models with varying weights, weighted–average ensembles leverage the strengths of each model to achieve superior performance.

We propose a method to achieve a weighted homogeneous quantum ensemble, based on quantum classifiers that rely on an indexing register for encoding data, considering the training instance vectors in superposition, such as the quantum distance classifier [4], the quantum cosine classifier [5], and a classification routine based on the swap test [6]. Our method enables subsampling of both features and training points for these instance-based quantum classifiers, by exploiting superposition and adding an additional control register that performs a controlled unitary on the data register. The idea is to have a quantum-parallel execution of the internal classifiers in superposition. To enable the weighting, the method requires two different execution modalities, one for training and one for testing, differing in the measured qubits.

The advantages of ensemble methods in the quantum setting have been presented in multiple works, where the proposed quantum-classical hybrid schemes shows classification performance improvements [7]–[10]. The idea of quantum ensembles has been introduced in [11], where the authors propose a Bayesian Model Averaging method with non-trained classifiers weighted by accuracy. The authors in [12] propose an ensemble strategy with the introduction of diversity by action of a controlled unitary execution acting on the data register. However, the authors of the previous work provide a single example with qubit encoding, where single data instances are encoded in different qubits, and the introduction of diversity by unitary application is performed entangling different training observations for each control state. In our proposal, we consider a more efficient data encoding strategy (superposition of data samples) that requires a different diversity introduction mechanism, in our case obtained by permutation of indexes and post-measurement state selection. Additionally, rather than relying solely on diverse classifiers, which require independence to be advantageous, we propose a weighted aggregation of the internal models to increase accuracy. The proposed procedure starts with the initial state of the classifier, encoding the features in the amplitudes of quantum states and includes indexing registers for the training points and features. To achieve this, we introduce an additional quantum register called control register, initialized in balanced superposition. Its size determines the number of diverse internal classifiers. Then we apply controlled unitary logic, controlled by the control register and targeting the indexing register of training points and features. The purpose of the controlled unitary is to introduce variability in the composition of the training set by entangling it with each basis state of the control register. This can be achieved through bit-flip logic, as an application of CNOT and CSWAP gates. After the bit flipping, a different permutation of the original training set is entangled with each basis state of the control register. The state preparation of the control register, classifier register and ancilla qubit, with consequent action of the "shuffling" controlled-unitary can be written as follows:

$$|0\rangle_{ctrl}^{\otimes N} \otimes |\psi\rangle_{class} \otimes |0\rangle_{aux} \xrightarrow{init} \sum_{c=1}^{N} \sqrt{w_c}\,|c\rangle \sum_{i,j=1}^{2^n,2^m} \alpha_{i,j}\,|i\rangle\,|j\rangle\,|\psi_{i,j}\rangle\,|0\rangle \xrightarrow{U_S} \sum_{c=1}^{N} \sqrt{w_c}\,|c\rangle \sum_{i,j=1}^{2^n,2^m} \alpha_{i,j}^{c}\,|i\rangle\,|j\rangle\,|\psi_{i,j}\rangle\,|0\rangle\,, \tag{1}$$

where $\sqrt{w_c}$ indicates the amplitude related to the weight w_c. In the training procedure the weights are uniform, and the control register can be prepared in balanced superposition by application of Hadamard gates. The amplitude $\alpha_{i,j}^{c}$ indicates a different feature amplitude depending on the control c for each index i, j. Then, a (multi-)controlled-NOT gate is applied with control on partial index and feature registers, while acting on an ancillary qubit. Assuming

control by 2 qubits, specifically the least significant in binary notation of the index and feature registers respectively:

$$\xrightarrow{CCX} \sum_{c=1}^{N} \sqrt{w_c} \left|c\right\rangle \sum_{i,j=1}^{2^{n-1},2^{m-1}} \alpha^c_{2i+1,2j} \left|2i+1\right\rangle \left|2j\right\rangle \left|\psi_{2i+1,2j}\right\rangle \left|0\right\rangle + \alpha^c_{2i,2j+1} \left|2i\right\rangle \left|2j+1\right\rangle \left|\psi_{2i,2j+1}\right\rangle \left|0\right\rangle \\ + \alpha^c_{2i,2j} \left|2i\right\rangle \left|2j\right\rangle \left|\psi_{2i,2j}\right\rangle \left|0\right\rangle + \alpha^c_{2i+1,2j+1} \left|2i+1\right\rangle \left|2j+1\right\rangle \left|\psi_{2i+1,2j+1}\right\rangle \left|1\right\rangle . \tag{2}$$

The selection of training indexes and features is obtained by measuring the ancilla and obtaining a post-measurement state corresponding to the related state of the control qubits. For example, measuring 1 selects the training indexes and features that have 1 in their binary representation in correspondence with the measured qubits, while measuring 0 selects the complementary set. The measurement probability is expected to be proportional to size of the subset, assuming the feature amplitudes being uniformly distributed. After measuring the ancilla, each basis state of the control register will have a different data composition, resulting in multiple diverse datasets in superposition. Supposing we want to select the set K where the ancilla is 0 ($p_0 \approx 0.75$), the data selection along with the consequent classifier execution is as follows:

$$\xrightarrow{measure\, ancilla\, 0} \frac{1}{\sqrt{p_0}} \sum_{c=1}^{N} \sqrt{w_c} \left|c\right\rangle \sum_{k \in K} \alpha^c_k \left|k\right\rangle \left|\psi_k\right\rangle \left|0\right\rangle \xrightarrow{classifier} \sum_{c=1}^{N} \sqrt{w_c} \left|c\right\rangle \left|\phi_c\right\rangle \left(\beta^c_0 \left|0\right\rangle + \beta^c_1 \left|1\right\rangle\right) \left|0\right\rangle , \tag{3}$$

with β^c representing the amplitude of the output qubit of the classifier with the related measurement probability being its output, and $\left|\phi_c\right\rangle$ indicating the rest of the classifier's state. During training, the circuit is executed on a validation set, and requires the measurement of the control register to collect the results of the individual internal classifiers. Then, the control register is measured along with the classifier output to obtain the single internal classifiers' predictions, and statistics are collected by executing the circuit multiple times. This procedure can be applied for predicting the training (or a different validation) set, collecting the internal classifiers' outputs for it.

Afterwards, the individual outputs are used to calculate weights for each internal classifier. These weights can be computed arbitrarily with classical or quantum procedures. A possibility is applying a weighting based on their performance on the training set, such as weighting by accuracy or by considering exponential weight decaying with errors. The chosen method involves optimization employing a logistic regression. The computed aggregation weights are then represented in the amplitudes of the control register, taking into account positive weights.

At test time, the classifiers are initialized with the original training set, and a control register is separately initialized with the computed weights in the amplitudes (Eq. 1). The shuffling logic is then executed (Eq. 2), and a measurement is taken to select the data in superposition (Eq. 3). Upon a successful measurement, the classifier is executed, and the output of the circuit statistics is determined as described in the following equation:

$$\mathbb{P}(0) = \frac{\sum_{c=1}^{N} w_c \beta_0^{c\,2}}{\sum_{c=1}^{N} w_c (\beta_0^{c\,2} + \beta_1^{c\,2})}. \tag{4}$$

Subsequently, the output can be post-processed classically, obtaining the weighted ensemble output. In our case, we computed a positive weight vector w, along with a bias b and a factor k for the logistic regression, minimizing the log-loss by bounded optimization using the L-BFGS-B algorithm. Then the prediction is obtained as follows:

$$y(x) = sign(\frac{1}{1 + e^{-(k\mathbb{P}(0)+b)}} - \frac{1}{2}). \tag{5}$$

We report some empirical results of the performance of the proposed weighted ensemble. We implemented the scheme in Qiskit [13], and tested it on 11 real-world datasets from the UCI Machine Learning Library [14], pre-processed to be suitable for a binary classification task. We employed the data with three different data standardization techniques. Specifically *none* refers to data as it is, *std* standardizes to mean 0 and standard deviation 1, and with *minmax* each feature belongs to $[0, 1]$. Regarding the execution, *statevector simulation* refers to exact results computed based on the statevector, while *local simulation* indicates circuit execution in a noiseless environment with 8192 repetitions using the Aer simulator. The same testing setup is extensively described in [10]. The weighted ensemble adopts the scheme described above, thus the selection is performed by measuring the ancilla in 0 after the action of the first index and feature qubits, and the aggregation is by means of logistic regression.

Considering *statevector simulation*, hence the expected results, the weighted ensemble shows a noticeable average accuracy improvement for all the considered data normalization methods, and for both the quantum distance and swap test as internal classifiers. Moreover, the performance slightly improves increasing the control register size (d indicates the number of control qubits). These results are reported in Table 1. We considered a reduced-set testing with *local simulation*, to show the impact of circuit repetitions. We employed the quantum distance classifier and the 8 datasets with less then or equal to 100 samples, due to computational constraints. The single distance classifier shows little average accuracy lowering with shots, with fluctuations of mispredicted points near

the prediction threshold balancing out the correctly predicted points. On the other hand, the weighted ensemble is more affected by measurement repetitions, likely due to error propagation in the optimization part. Nevertheless, it still maintains an accuracy advantage over the single classifier. These results are displayed in Figure 1.

The proposed quantum weighted ensemble method demonstrates a performance improvement over single quantum classifiers, allowing for a theoretical exponential ensemble size in the control register dimension. Indeed, for linear classifiers, such as distance and cosine quantum classifiers, the weighted aggregation can be interpreted as a modified dataset with multiplicative factors on training points and features. This allows for an alternative execution of the weighted ensemble at test-time by using a single quantum classifier on the modified dataset, avoiding the data selection circuit logic but requiring a change in the data encoding procedure of the initial state.

For non-linear classifiers, such as those using the swap-test, the weighted ensemble resembles a two-layer neural network (NN) with some missing connections in the first layer and the second layer's weights trained by an optimization procedure. NNs based on the swap-test have been proposed in [15], by considering combination of quantum modules, while the proposed method exploiting a swap-test procedure is quantum. To assess the capabilities and correctness of our approach, we tested it on a synthetic 2-dimensional XOR dataset. As expected, the linear classifiers struggled, while the ensemble using the non-linear swap test achieved nearly perfect accuracy.

Additionally, a majority voting aggregation strategy can be implemented by sampling the circuit defined for the training procedure. This allows the individual outputs of the internal classifiers to be estimated and then combined in a bagging fashion via majority voting. Sampling with replacement-like data sampling may be advantageous and can be achieved by using a larger training dataset, employing multiple copies of the original training data, and allowing for training point repetitions in each subset up to the number of original training set copies.

Future directions include exploring different diversity introduction mechanisms not based on data subsetting, such as parametric controlled rotation operations. Overall, the proposed quantum weighted ensemble method offers a robust framework for improving classification accuracy in quantum machine learning.

References

[1] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[2] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[3] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[4] M. Schuld, M. Fingerhuth, and F. Petruccione, "Implementing a distance-based classifier with a quantum interference circuit," *EPL (Europhysics Letters)*, vol. 119, no. 6, p. 60 002, Sep. 2017.

[5] D. Pastorello and E. Blanzieri, "A Quantum Binary Classifier based on Cosine Similarity," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, IEEE, Oct. 2021.

[6] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, "Quantum Fingerprinting," *Phys. Rev. Lett.*, vol. 87, p. 167 902, 16 Sep. 2001.

[7] X.-Y. Zhang and M.-M. Wang, "An efficient combination strategy for hybrid quantum ensemble classifier," *International Journal of Quantum Information*, vol. 21, no. 06, p. 2 350 027, 2023.

[8] R. Qin, Z. Liang, J. Cheng, P. Kogge, and Y. Shi, *Improving quantum classifier performance in nisq computers by voting strategy from ensemble learning*, 2022.

[9] M. Incudini, M. Grossi, A. Ceschini, *et al.*, "Resource saving via ensemble techniques for quantum neural networks," *Quantum Machine Intelligence*, vol. 5, no. 39, 2023.

[10] E. Tolotti, E. Zardini, E. Blanzieri, and D. Pastorello, "Ensembles of quantum classifiers," *Quantum Information and Computation*, vol. 24, no. 3&4, 2024.

[11] M. Schuld and F. Petruccione, "Quantum ensembles of quantum classifiers," *Scientific Reports*, vol. 8, no. 2772, 2018.

[12] A. Macaluso, L. Clissa, S. Lodi, and C. Sartori, "An efficient quantum algorithm for ensemble classification using bagging," *IET Quantum Communication*, 2024.

[13] Q. contributors, *Qiskit: An Open-source Framework for Quantum Computing*, 2023.

[14] D. Dua and C. Graff, *UCI Machine Learning Repository*, `http://archive.ics.uci.edu/ml`, 2017.

[15] D. Pastorello and E. Blanzieri, "Scalable quantum neural networks by few quantum resources," *International Journal of Quantum Information*, 2024.

Material to be presented

quantum distance classifier						
ensemble	d	norm	accuracy		F1 score	
			median	mean	median	mean
single	-	none	56.3	58.8	56.8	45.0
ensemble	1	none	92.9	83.0	92.3	83.5
ensemble	3	none	91.4	83.8	92.0	85.2
ensemble	5	none	92.9	85.2	92.6	86.9
single	-	std	92.9	84.4	93.3	80.4
ensemble	1	std	95.0	88.4	95.4	88.0
ensemble	3	std	95.0	89.1	95.8	88.9
ensemble	5	std	95.2	89.7	95.8	89.3
single	-	minmax	66.9	68.0	72.0	56.8
ensemble	1	minmax	86.4	84.3	85.7	83.5
ensemble	3	minmax	89.3	84.6	88.2	83.6
ensemble	5	minmax	89.0	85.9	89.6	84.7

quantum swap test classifier						
ensemble	d	norm	accuracy		F1 score	
			median	mean	median	mean
single	-	none	61.9	63.1	62.7	51.1
ensemble	1	none	92.9	82.7	92.3	83.4
ensemble	3	none	92.9	82.7	92.3	81.9
ensemble	5	none	92.9	84.1	92.6	85.5
single	-	std	50.0	54.0	42.7	37.6
ensemble	1	std	64.8	62.9	58.6	59.4
ensemble	3	std	65.8	63.7	57.1	60.4
ensemble	5	std	65.0	66.2	62.5	62.9
single	-	minmax	69.4	71.0	73.4	61.9
ensemble	1	minmax	87.5	85.3	87.6	84.1
ensemble	3	minmax	88.8	85.7	89.4	84.2
ensemble	5	minmax	88.8	86.1	87.2	84.0

Table 1: Performance with *statevector simulation* of single classifiers and weighted ensemble in terms of aggregated results over 10 Monte Carlo runs for 11 real-world datasets. The considered internal models are the quantum distance (left) and swap-test (right) classifiers. Multiple control register sizes (d) and normalization techniques are taken into account.

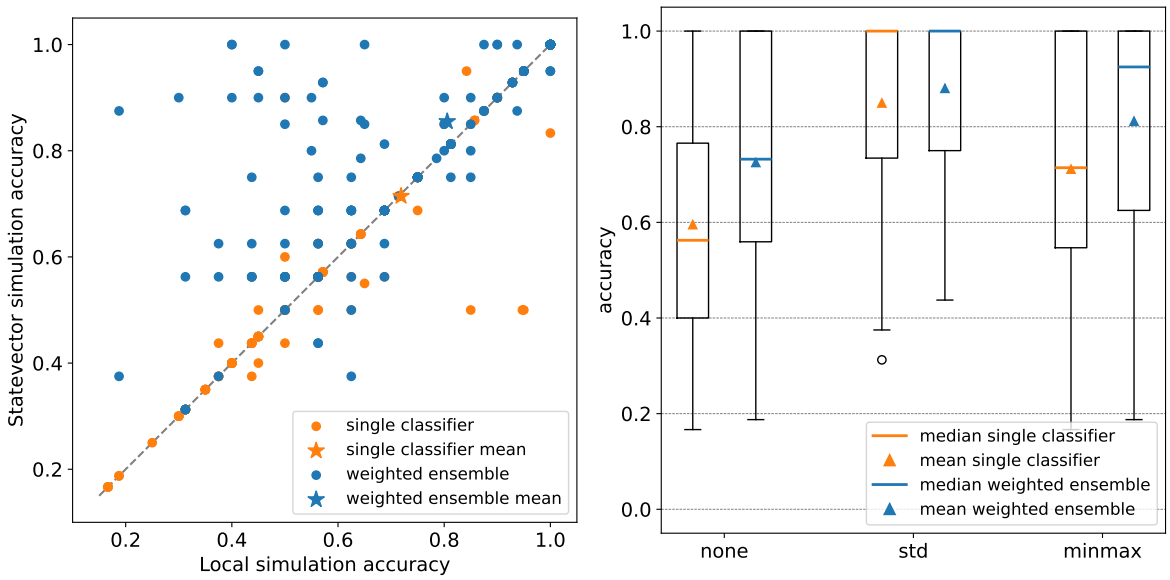

Figure 1: Accuracy performance of *local simulation* with 8192 shots, considering the quantum distance classifier on 8 datasets over 10 Monte Carlo runs. The left panel shows the accuracy comparison between *local simulation* and *statevector simulation*, where points above the bisector indicate accuracy degradation. The right panel presents the accuracy comparison of single and ensemble classifier ($d = 5$) with *local simulation* for the different normalization techniques.