Feature Selection for Classification with QAOA

1st Gloria Turati
Politecnico di Milano
Milano, Italy
gloria.turati@polimi.it

2nd Maurizio Ferrari Dacrema
Politecnico di Milano
Milano, Italy
maurizio.ferrari@polimi.it

3rd Paolo Cremonesi
Politecnico di Milano
Milano, Italy
paolo.cremonesi@polimi.it

Abstract—Feature selection is of great importance in Machine Learning, where it can be used to reduce the dimensionality of classification, ranking and prediction problems. The removal of redundant and noisy features can improve both the accuracy and scalability of the trained models. However, feature selection is a computationally expensive task with a solution space that grows combinatorically. In this work, we consider in particular a quadratic feature selection problem that can be tackled with the Quantum Approximate Optimization Algorithm (QAOA), already employed in combinatorial optimization. First we represent the feature selection problem with the QUBO formulation, which is then mapped to an Ising spin Hamiltonian. Then we apply QAOA with the goal of finding the ground state of this Hamiltonian, which corresponds to the optimal selection of features. In our experiments, we consider seven different real–world datasets with dimensionality up to 21 and run QAOA on both a quantum simulator and, for small datasets, the 7–qubit IBM (ibm–perth) quantum computer. We use the set of selected features to train a classification model and evaluate its accuracy. Our analysis shows that it is possible to tackle the feature selection problem with QAOA and that currently available quantum devices can be used effectively. Future studies could test a wider range of classification models as well as improve the effectiveness of QAOA by exploring better performing optimizers for its classical step.

Index Terms—QAOA, Feature selection, QUBO, Classification

I. POSTER RELEVANCE

In this work we address the feature selection problem for classification using the Quantum Approximate Optimization Algorithm (QAOA). We observe that the final classifiers are comparable and sometimes outperform the classifiers obtained by performing feature selection with classical algorithms and Quantum Annealers (QA).

II. EXTENDED POSTER ABSTRACT

A. Introduction

In recent years, the spread of big data and Machine Learning has led to the need for methods that facilitate the understanding, analysis and processing of information. This is especially important when data are used for classification, ranking and prediction tasks, where the redundancy of information can lead to computationally expensive algorithms and cause overfitting problems. Feature selection is useful for this purpose, as the elimination of irrelevant features allows to speed up the training of the model and improve its accuracy. However, the process of deciding which features are the most relevant is computationally expensive: the direct evaluation of all possible subsets of features is an NP–hard problem and its complexity

increases factorially with the number of the features. The techniques traditionally used to perform feature selection can be summarized in three main categories: filter, wrapper and embedded methods, which have been empirically shown to be effective, especially when combined together, but still have a high computational cost [1] [2].

Some papers have already tackled feature selection with quantum computing with the goal of addressing its classical limitations. [3] has shown that QA allows to find features producing classification models with accuracy comparable to classical solvers, by testing three different feature selection approaches on 15 publicly available datasets. Another paper by [4] has focused on Mutual Information and shown similar results compared to [3]; the analysis includes the Variational Quantum Eigensolver (VQE) but the algorithm is only reported for a single synthetically generated dataset. In this paper we do a step forward and report preliminary results for QAOA on three feature selection approaches on seven real–world datasets.

B. Problem Formulation

We consider a classification problem with n features $f_1, \ldots, f_n \in F$ and target variable y: the goal is to select the most informative $k < n$ features with which to feed the classification algorithm. We treat the feature selection task as a Quadratic Unconstrained Binary Optimization (QUBO) [5] problem

$$\min_{x \in \{0,1\}^n} x^T Q x \tag{1}$$

where $x = (x_1, \ldots, x_n) \in \{0,1\}^n$ and Q is an upper triangular or symmetrical $n \times n$ matrix. The binary variables x represent the solution of the feature selection problem ($x_i = 1$ if and only if the i–th feature is selected) and Q is defined in such a way that the cost function represents the amount of information provided by the selected features. We consider three QUBO models characterized by different definitions of the objective function.

1) QUBO Correlation: This approach aims to select features highly correlated with the target label. Here the selection of features correlated to each other is discouraged to avoid redundancy. Thus we define

$$Q_{i,j} = \begin{cases} r(f_i, f_j) & \text{if } i \neq j \\ -r(f_i, y) & \text{if } i = j \end{cases} \tag{2}$$

where $r(\cdot, \cdot)$ is the Pearson correlation coefficient.

2) QUBO Mutual Information: Here the QUBO matrix is built as

$$Q_{i,j} = \begin{cases} -MI(f_i, y|f_j) & \text{if } i \neq j \\ -MI(f_i, y) & \text{if } i = j \end{cases} \quad (3)$$

where $MI(f_i, y)$ is the mutual information between the i–th feature and the target variable y and $MI(f_i, y|f_j)$ is the conditional mutual information between f_i and y, assuming that f_j is known.

3) QUBO Boosting: This metric makes use of a Support Vector Classifier (SVC) to estimate the level of information provided by a feature. It is built by first considering each of the $|F|$ features and training a support vector classifier (SVC) for each of them; each of the trained classifiers is then used to predict the outcome variable. If we denote by h_i the class predicted using the model trained considering only the i–th feature, we can define

$$Q_{i,j} = \begin{cases} r(h_i, h_j) & \text{if } i \neq j \\ \frac{S}{|F|^2} + \lambda - 2 \cdot r(h_i, y) & \text{if } i = j \end{cases} \quad (4)$$

where λ is a hyper–parameter, S is the number of samples in the dataset, $r(h_i, h_j)$ is the Pearson correlation coefficient between labels predicted with different models and $r(h_i, y)$ is the correlation between each predicted label and the correct one (see [6] for further details).

At this point, in order to control the number of selected features, we add a quadratic penalty term to the cost function so that its value increases when the number of actually selected features deviates from the target k. The resulting minimization problem is

$$\min_{x \in \{0,1\}^n} x^T Q x + \left(\sum_{i=1}^{n} x_i - k\right)^2 \quad (5)$$

Once our problem has been written in the QUBO formulation, we can convert it to the Ising formulation [7] and then apply QAOA.

C. Selecting Features with QAOA

Differently from other Variational Quantum Algorithms, QAOA has a specific circuit, which depends on the parameters $\gamma = (\gamma_1, \ldots, \gamma_p)$ and $\beta = (\beta_1, \ldots, \beta_p)$. It consists of p layers where gates implementing the cost operator $U_C(\gamma)$ and the mixer operator $U_M(\beta)$ are applied (see the original work by Farhi for further details on the structure of QAOA circuit [8]). QAOA algorithm involves a number of iterations, in each of which the parameters are fixed and the cost function is estimated by executing the QAOA circuit m times, measuring the final states and averaging the expectations of the Hamiltonian on the collapsed states. At the end of each iteration, a classical optimizer modifies the parameters with the goal of minimizing the cost function. The algorithm stops when the parameters meet certain tolerance criteria, specified in the construction of the optimizer (we will not analyze the stopping criterion, as the behaviour of the optimizer is outside the scope of our work). Once the optimal parameters have

been found, we run the final circuit m times again and pick the solution with the lowest cost, which represents the selection of the most relevant features.

D. Experiments and Results

We perform two experiments with the goal of:

- comparing QAOA with Quantum Annealers (QA) and classical solvers when used to select features for training classification models
- studying the dependence of the final classification accuracy on the depth p of the circuit.

1) Comparison of QAOA with QA and classical solvers: We consider the seven real–world datasets "iris", "cars1", "LED–display–domain–7digit", "breast–cancer", "wine", "vehicle", "thyroid–ann", taken from OpenML [9], and having $n = 4, 7, 7, 9, 13, 18, 21$ features respectively. Then we define the Q matrix according to the QUBO formulations described in II-B and choose $k = \lceil 50\% \, n \rceil$ as the target number of features to be selected. We run QAOA for all the possible combinations of datasets and methods using the IBM "QASM simulator" and consider circuits having depth p ranging from 1 to 10. The number of shots used for approximating the cost function and for the final sampling is fixed to 100. Finally, we choose the SciPy "COBYLA" as a classical optimizer, which is gradient–free and has been shown to have good performances when the environment noise is negligible [10]. In addition, for the three smallest datasets, we run QAOA using also the real 7–qubit "ibm–perth" QPU. We still perform 100 shots for each iteration and choose the "COBYLA" optimizer, but this time we only consider the circuit depths p ranging from 1 to 6, due to the long queue times to access the QPU.

The optimization of the QUBO functions and the selection of features with Quantum Annealers and classical solvers are performed as described in [3], using the publicly available code. In particular, experiments on Annealers are conducted on the D–Wave 5600–qubit Advantage QPU and with a hybrid quantum–classical approach consisting in decomposing the QUBO problem in smaller tasks which are then executed on the Annealer. The traditional solvers tested are Simulated Annealing (SA), Steepest Descent (SD) and Tabu Search (TS).

At this point, we use the selected features to train the classifier. We choose Random Forest as a classification model, since it has been shown to provide good accuracy even with limited tuning. The datasets used are split into training and test set with 70% and 30% of the samples respectively. Since some datasets have limited size, we do not create an additional subdivision for validation, but apply a 5–Fold CV on the training set and average the scores. For solvers making use of QAOA, the hyper–parameter p with the best score on the validation is chosen. Finally, we evaluate the model on the test set. Since Random Forest has an inherent stochasticity, we perform the evaluation 5 times and average the accuracy obtained. The results are summarized in Table I.

It can be observed that the accuracy of the model trained using features selected with QAOA is comparable to that obtained by selecting features with Quantum Annealers and

TABLE I

ACCURACY FOR ALL QUBO METHODS, DATASETS AND SOLVERS. THE BEST ACCURACY FOR EACH QUBO METHOD AND DATASET IS HIGHLIGHTED IN BOLD. EXPERIMENTS WITH THE IBM QPU ARE PERFORMED ONLY FOR DATASETS WITH NO MORE THAN SEVEN FEATURES

| Method | Dataset | n | QAOA | | QA | | Traditional Solver | | |
			Simulator	IBM QPU	D–Wave QPU	QPU Hybrid	SA	SD	TS
QUBO Correlation	iris	4	0.9022	0.9111	**0.9333**	0.9289	**0.9333**	**0.9333**	0.9289
	cars1	7	0.8424	0.8458	0.8373	0.8458	0.8458	**0.8508**	0.8458
	LED–display–domain–7digit	7	0.5467	0.5467	0.5333	0.5413	0.5427	0.5440	**0.6613**
	breast–cancer	9	0.6674	-	0.6884	0.6791	0.6953	0.6698	**0.7070**
	wine	13	0.9407	-	0.8852	0.8815	0.8778	0.8852	**0.9481**
	vehicle	18	0.7339	-	0.7118	0.7197	0.7063	0.7102	**0.7425**
	thyroid–ann	21	**0.9898**	-	0.9470	0.9473	0.9468	0.9475	0.9484
QUBO Mutual Information	iris	4	0.8756	0.8844	0.9111	0.9111	0.9111	0.9111	**0.9333**
	cars1	7	0.8441	0.8407	0.8424	0.8424	**0.8492**	0.8441	0.8390
	LED–display–domain–7digit	7	**0.6787**	0.6747	**0.6787**	0.6760	**0.6787**	0.6720	0.6773
	breast–cancer	9	**0.7302**	-	0.6791	0.6907	0.6860	0.6860	0.7233
	wine	13	0.9630	-	0.9556	0.9630	0.9630	0.9667	**0.9741**
	vehicle	18	0.7591	-	0.7283	0.7409	0.7386	0.7402	**0.7606**
	thyroid–ann	21	0.9890	-	0.9249	0.9249	0.9249	0.9249	**0.9929**
QUBO SVC Boosting	iris	4	**0.9333**	**0.9333**	**0.9333**	**0.9333**	**0.9333**	0.9244	**0.9333**
	cars1	7	**0.8271**	0.7695	0.7203	0.7220	0.7220	0.7237	0.7254
	LED–display–domain–7digit	7	**0.5480**	0.4373	0.4000	0.4000	0.4000	0.4000	0.3947
	breast–cancer	9	**0.7070**	-	**0.7070**	0.6860	0.6465	0.6651	0.6488
	wine	13	0.8778	-	0.9370	0.9370	0.9333	0.9370	**0.9556**
	vehicle	18	0.7134	-	**0.7386**	0.6866	0.6953	0.6913	0.7307
	thyroid–ann	21	**0.9811**	-	0.9661	0.9703	0.9666	0.9693	0.9684

TABLE II

AVERAGE ACCURACY AND STANDARD DEVIATIONS OF THE CLASSIFICATION MODELS FOR THE DATASET "THYROID–ANN" OBTAINED WITH THE FEATURES SELECTED WITH ALL QUBO METHODS AND $p = 1, \ldots, 10$. THE ERROR BARS IN THE PLOT REPRESENT THE STANDARD DEVIATION

p	QUBO Correlation	QUBO Mutual Information	QUBO SVC Boosting
1	0.9766 ± 0.0093	0.9657 ± 0.0192	0.9722 ± 0.0076
2	0.9717 ± 0.0090	0.9323 ± 0.0075	0.9637 ± 0.0106
3	0.9422 ± 0.0236	0.9666 ± 0.0253	0.9623 ± 0.0084
4	0.9701 ± 0.0183	0.9567 ± 0.0261	0.9669 ± 0.0113
5	0.9639 ± 0.0262	0.9583 ± 0.0281	0.9609 ± 0.0138
6	0.9575 ± 0.0278	0.9620 ± 0.0200	0.9604 ± 0.0096
7	0.9468 ± 0.0280	0.9505 ± 0.0276	0.9685 ± 0.0120
8	0.9765 ± 0.0088	0.9653 ± 0.0296	0.9683 ± 0.0061
9	0.9535 ± 0.0304	0.9606 ± 0.0247	0.9583 ± 0.0265
10	0.9441 ± 0.0302	0.9639 ± 0.0250	0.9701 ± 0.0152
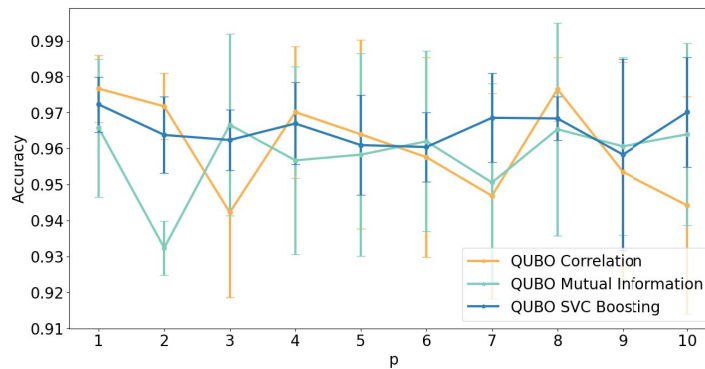
traditional solvers. In particular, there is no QUBO formulation that outperforms the others and the accuracy is strictly dependent on the dataset. Finally, the results obtained by running QAOA on the simulator are very similar to those

provided by the real QPU, an indication that also the quantum hardware is reliable in solving the feature selection problem for classification.

2) Dependence of the accuracy on p: In order to study the dependence of the accuracy on the depth of the circuit p, we consider the same datasets and QUBO formulations as the previous experiment and run QAOA on the IBM "QASM simulator" for all p in the range from 1 to 10. We still perform 100 executions of the circuit per iteration and choose "COBYLA" as a classical optimizer. Once we get the most relevant features, we train the Random Forest and compute the accuracy. We execute this pipeline 5 times and average the accuracy.

Due to space reasons, we report in Table II only the results for the dataset "thyroid–ann", which is the largest we tested. The behaviour of p in the other datasets is similar. The table and the corresponding graph show the average accuracy and the standard deviation of the final classifier. It can be observed that the accuracy varies with p, but, due to the high standard deviation (with a mean of 0.0191), we can only conclude that p is a challenging hyper–parameter to tune.

E. Conclusions and Future Work

This work shows that it is possible to perform feature selection with QAOA and obtain classification accuracy comparable to what obtained by selecting features with classical approaches. However, further studies may be conducted to test a wider range of classification models and analyze the behaviour of the QPU on datasets with a larger number of features. Furthermore, the effectiveness of QAOA may be improved by exploring other optimizers which may be better suited for specific feature selection heuristics or to the noisy nature of the QPU.

REFERENCES

[1] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 16–28, 2014. [Online]. Available: https://doi.org/10.1016/j.compeleceng.2013.11.024

[2] A. Jovic, K. Brkic, and N. Bogunovic, "A review of feature selection methods with applications," in *38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2015, Opatija, Croatia, May 25-29, 2015*, P. Biljanovic, Z. Butkovic, K. Skala, B. Mikac, M. Cicin-Sain, V. Sruk, S. Ribaric, S. Gros, B. Vrdoljak, M. Mauher, and A. Sokolic, Eds. IEEE, 2015, pp. 1200–1205. [Online]. Available: https://doi.org/10.1109/MIPRO.2015.7160458

[3] M. Ferrari Dacrema, F. Moroni, R. Nembrini, N. Ferro, G. Faggioli, and P. Cremonesi, "Towards feature selection for ranking and classification exploiting quantum annealers," in *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, E. Amigó, P. Castells, J. Gonzalo, B. Carterette, J. S. Culpepper, and G. Kazai, Eds. ACM, 2022, pp. 2814–2824. [Online]. Available: https://doi.org/10.1145/3477495.3531755

[4] S. Mücke, R. Heese, S. Müller, M. Wolter, and N. Piatkowski, "Quantum feature selection," *CoRR*, vol. abs/2203.13261, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2203.13261

[5] F. W. Glover, G. A. Kochenberger, R. Hennig, and Y. Du, "Quantum bridge analytics I: a tutorial on formulating and using QUBO models," *Ann. Oper. Res.*, vol. 314, no. 1, pp. 141–183, 2022. [Online]. Available: https://doi.org/10.1007/s10479-022-04634-2

[6] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, "Qboost: Large scale classifier training with adiabatic quantum optimization," in *Proceedings of the 4th Asian Conference on Machine Learning, ACML 2012, Singapore, Singapore, November 4-6, 2012*, ser. JMLR Proceedings, S. C. H. Hoi and W. L. Buntine, Eds., vol. 25. JMLR.org, 2012, pp. 333–348. [Online]. Available: http://proceedings.mlr.press/v25/neven12.html

[7] A. Lucas, "Ising formulations of many NP problems," *CoRR*, vol. abs/1302.5843, 2013. [Online]. Available: http://arxiv.org/abs/1302.5843

[8] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014. [Online]. Available: https://arxiv.org/abs/1411.4028

[9] J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo, "Openml: networked science in machine learning," *SIGKDD Explor.*, vol. 15, no. 2, pp. 49–60, 2013. [Online]. Available: https://doi.org/10.1145/2641190.2641198

[10] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. de Jong, "Classical optimizers for noisy intermediate-scale quantum devices," in *IEEE International Conference on Quantum Computing and Engineering, QCE 2020, Denver, CO, USA, October 12-16, 2020*. IEEE, 2020, pp. 267–277. [Online]. Available: https://doi.org/10.1109/QCE49297.2020.00041