QUANTUM SUPPORT VECTOR REGRESSION FOR BIOPHYSICAL VARIABLE ESTIMATION IN REMOTE SENSING

Edoardo Pasetto[1,2], *Amer Delilbasic*[1,2], *Gabriele Cavallaro*[1,2], *Madita Willsch*[1,2], *Farid Melgani*[4], *Morris Riedel*[1,2,3], *Kristel Michielsen*[1,2,5]

[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
[2] AIDAS, 52425 Jülich, Germany
[3]School of Engineering and Natural Sciences, University of Iceland, Iceland
[4] Department of Information Engineering and Computer Science, University of Trento, Italy
[5] RWTH Aachen University, D52056 Aachen, Germany

ABSTRACT

Regression analysis has a crucial role in many Earth Observation (EO) applications. The increasing availability and recent development of new computing technologies motivate further research to expand the capabilities and enhance the performance of data analysis algorithms. In this paper, the biophysical variable estimation problem is addressed. A novel approach is proposed, which consists in a reformulated Support Vector Regression (SVR) and leverages Quantum Annealing (QA). In particular, the SVR optimization problem is reframed to a Quadratic Unconstrained Binary Optimization (QUBO) problem. The algorithm is then tested on the D-Wave Advantage quantum annealer. The experiments presented in this paper show good results, despite current hardware limitations, suggesting that this approach is viable and has great potential.

Index Terms— Support vector regression, quantum computing, quantum annealing, quantum machine learning, remote sensing

1. INTRODUCTION

Regression analysis [1] aims at finding the relationship between a set of independent variables $\{\mathbf{x_n}\}$ and a dependent variable y. Regression methods cover a wide range of applications, including economics, marketing, sociology, epidemiology and risk analysis [2]. Regression has enabled significant advancements in the context of Remote Sensing (RS). For example, biophysical parameters can be evaluated in a RS area by analyzing the complex relations between measured quantities, instead of relying on fixed and simplified assumptions [3]. Moreover, soft classification methods relying on logistic regression have been successfully applied to RS multispectral images [4].

In the Quantum Computing (QC) framework, an interdisciplinary field called Quantum Machine Learning (QML) focuses on the enhancement of classical Machine Learning (ML) algorithms by outsourcing computationally demanding steps to a quantum computer, which has the potential to outperform classical computation. Both theoretical and practical results have shown major improvements in several ML tasks that are expected to be exploited in the near future, e.g., for real applications with big data [5, 6]. Several quantum algorithms performing regression tasks have been proposed, i.e., for linear regression [7] and ridge regression [8].

This work focuses on a class of quantum algorithms, namely Quantum Annealing (QA). It is closely related to Adiabatic Quantum Computation (AQC) [9] and exploits the time evolution of a quantum mechanical system for solving combinatorial optimization problems. It has proven itself as a valuable asset in ML [10].

The objective of this paper is to define a novel Support Vector Regression (SVR) method based on QA, referred to as Quantum Support Vector Regression (QSVR). The main contribution consists in the redefinition of the SVR optimization problem [11] so that the training phase can be performed using a quantum annealer. The implementation proposed in this work is tested on a RS dataset and the hardware used for the annealing process is the D-Wave Advantage quantum processor, released by D-Wave Systems in 2020. The purpose is to show how RS can benefit from existing quantum technologies and which results can be obtained in regression tasks. More details, including documented codes and results related to Quantum Support Vector Regression (QSVR), will be available on the repository of this work[1].

The authors gratefully acknowledge the Jülich Supercomputing Centre for funding this project by providing computing time on the D-Wave Advantage system through the Jülich UNified Infrastructure for Quantum computing (JUNIQ). M.W. acknowledges support from the project JUNIQ that has received funding from the German Federal Ministry of Education and Research (BMBF) and the Ministry of Culture and Science of the State of North Rhine-Westphalia.

[1] https://gitlab.jsc.fz-juelich.de/sdlrs/quantum-support-vector-regression

2. QUANTUM SUPPORT VECTOR REGRESSION

2.1. Classical SVR Formulation

Given a training set $T = \{(\mathbf{x}_n, y_n) : n = 0, \ldots, N-1\}$, where $\mathbf{x}_n \in \mathbb{R}^d$ are the feature vectors and $y_n \in \mathbb{R}$ are the target values, the conventional dual formulation of a SVR can be summarized as the following Quadratic Programming (QP) problem:

$$L(\boldsymbol{\alpha}, \hat{\boldsymbol{\alpha}}) = \frac{1}{2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} (\alpha_n - \hat{\alpha}_n)(\alpha_m - \hat{\alpha}_m) k(\mathbf{x}_n, \mathbf{x}_m) + \\ - \epsilon \sum_{n=0}^{N-1} (\alpha_n + \hat{\alpha}_n) + \sum_{n=0}^{N-1} (\alpha_n - \hat{\alpha}_n) t_n \tag{1}$$

subject to the constraints:

$$\sum_{n=1}^{N} (\alpha_n - \hat{\alpha}_n) = 0 \tag{2a}$$

$$0 \leq \alpha_n \leq C \tag{2b}$$

$$0 \leq \hat{\alpha}_n \leq C \tag{2c}$$

The variables $\boldsymbol{\alpha} = \{\alpha_n : n = 0, \ldots, N-1\}$ and $\hat{\boldsymbol{\alpha}} = \{\hat{\alpha}_n : n = 0, \ldots, N-1\}$ are the variables of the dual problem, whereas $k(\mathbf{x}_n, \mathbf{x}_m)$ is the kernel function, C is a regularization parameter and ϵ is the error sensitivity.

2.2. QUBO Problem Definition

In order to execute the QSVR algorithm on a D-Wave quantum annealer, the optimization problem must be reformulated as a Quadratic Unconstrained Binary Optimization (QUBO) problem[2]. QUBO consists in the minimization of a quadratic energy function E:

$$E = \sum_{i \leq j} a_i Q_{ij} a_j, \tag{3}$$

where $a_i \in \{0, 1\}$ are the binary variables of the optimization problem and Q is an upper-triangular real matrix called QUBO weight matrix.

The required steps for the formulation of a QUBO problem are:

- Encoding of the solution space to a set of binary variables;
- Definition of penalty terms to account for the constraints;
- Computation of the QUBO weight matrix Q.

[2] https://docs.dwavesys.com/docs/latest/c_gs_3.html

Since the solutions of Eqs. (1)-(2) are real values, i.e., $\alpha_n, \hat{\alpha}_n \in \mathbb{R}$, the following binary encoding is used:

$$\alpha_n = \sum_{k=0}^{K-1} B^{k-P} a_{Kn+k}, \tag{4}$$

$$\hat{\alpha}_n = \sum_{k=0}^{K-1} B^{k-P} a_{K(N+n)+k}, \tag{5}$$

where $a_{Kn+k}, a_{K(N+n)+k} \in \{0, 1\}$ are binary variables, K is the number of binary variables to encode α_n and $\hat{\alpha}_n$, B is the base used for the encoding, and $P \geq 0$ is a parameter that allows for negative exponents.

Then, a squared penalty term multiplied by ξ is defined, which includes the constraint of Eq. 2a. Another penalty term controlled by the hyperparameter β is defined to enforce that, for each n, at least one between α_n or $\hat{\alpha}_n$ is equal to 0. In the experiments the values used for ξ and β were 1 and 20, respectively. The constraints in Eqs. (2b)-(2c) are automatically included through the encoding in Eqs. (4)-(5), since the maximum for α_n and $\hat{\alpha}_n$ is given by

$$C = \sum_{k=0}^{K-1} B^k. \tag{6}$$

The last step is rewriting Eqs. (1)-(2) in the form of an energy function, as in Eq. (3), by substituting the variables α_n and $\hat{\alpha}_n$ with their encoding and adding the penalty terms. Therefore, the final formulation is the following:

$$\sum_{n,m=0}^{N-1} \sum_{i,j=0}^{K-1} \sum_{s,t=0}^{1} a_{K(sN+n)+i} \\ \tilde{Q}_{K(sN+n)+i, K(tN+m)+j} a_{K(tN+m)+j} \tag{7}$$

where $\tilde{Q}$ is a $2KN \times 2KN$ matrix whose elements are given by:

$$\tilde{Q}_{K(sN+n)+i, K(tN+m)+j} = (-1)^{(1-\delta_{st})} B^{i+j-2P} \\ (\frac{1}{2} k(\mathbf{x}_n, \mathbf{x}_m) + \xi - (1-s) t \delta_{nm} \beta) + \\ + \delta_{nm} \delta_{ij} B^{i-P} \delta_{st} (\epsilon + (-1)^{(1-s)(1-t)} t_n) \tag{8}$$

with $n, m \in \{0, \ldots, N-1\}$, $i, j \in \{0, \ldots, K-1\}$ and $s, t \in \{0, 1\}$. Since $\tilde{Q}$ is symmetric, the QUBO weight matrix Q, which is upper-triangular, can be computed as $Q_{ij} = \widetilde{Q}_{ij} + \widetilde{Q}_{ji}$ for $i < j$ and $Q_{ii} = \widetilde{Q}_{ii}$.

It is important to mention that, in order to submit a QUBO problem to a quantum annealer, an additional step is required, called *minor embedding* [12]. Each binary variable a_i is mapped to a chain of qubits, i.e., physically connected qubits, and each element of the QUBO matrix Q_{ij} (related to the product $a_i a_j$) is mapped to a physical connection between chains of qubits. The existence of an embedding for a given QUBO weight matrix is a necessary condition for the problem to be solvable by the quantum annealer.

Table 1: Values of MSE obtained on the test set for different combination methods and runs.

Run	SVR	QSVR 1	QSVR 2	QSVR 3	QSVR 4	QSVR 5	QSVR 6
1	7.961	10.9786	11.0244	11.0016	11.0811	10.2775	11.0849
2	12.1908	12.7688	12.7452	12.7919	12.9086	12.1013	12.9172
3	6.6338	10.5555	9.1293	11.2271	13.6113	10.9359	13.9183
4	9.6695	10.2681	10.2782	10.2851	10.3545	9.9055	10.3585
5	7.0943	11.7815	11.7834	11.7825	11.7844	10.9887	11.7845
6	12.4384	9.6309	9.6935	9.6967	9.7999	10.9737	9.8057
7	5.7134	9.7858	9.7859	9.7859	9.7859	9.1881	9.7859
8	6.8256	11.3143	11.4065	11.353	11.4646	11.1698	11.4684
9	5.9804	9.5261	9.5289	9.5272	9.5315	8.9554	9.5317
10	9.4644	13.3755	13.3756	13.3764	13.3795	12.1185	13.3798
Average	8.3972	10.9985	10.8751	11.0827	11.3701	10.6614	11.4035

3. EXPERIMENTAL RESULTS

3.1. Dataset

For the experimental validation, the SeaBAM dataset for regression has been used [13]. The dataset is used to estimate the concentration of chlorophyll in water and each example consists of a feature vector containing the data regarding 5 measures at different wavelengths and the associated level of chlorophyll concentration. For the training phase the values of the feature vectors and the target values have been converted to the logarithmic domain, as in [14]. The original values of the chlorophyll concentration ranges between 0.019 and 32.787 mg/m^3. The training set consists of 25 examples and the test set 668 examples. In each experimental run a different training and test datsets were used by varying the random seed used for their generation. The hyperparameters related to the classical and the quantum SVR are determined empirically on a validation set.

3.2. Experimental Setup

In the experimental phase, D-Wave Advantage is considered, which is the latest quantum annealer devised by D-Wave Systems. It is a quantum system composed of more than 5000 qubits and more than 35000 couplers, i.e., physical interconnections between qubits, based on the so called Pegasus qubit architecture. Access to D-Wave quantum and hybrid solvers is granted through the *D-Wave Leap*[3] cloud platform. Computing time on the D-Wave Advantage system is provided through the Jülich UNified Infrastructure for Quantum computing (JUNIQ)[4]. The QSVR algorithm is then compared with a classical SVR implementation, available in the Scikit-learn Python library.

[3] https://cloud.dwavesys.com/leap/

[4] https://www.fz-juelich.de/ias/jsc/EN/Expertise/JUNIQ/_node.html

3.3. Evaluation

The results of the comparison between the classical and quantum SVR are depicted in Tab. 1. The metric used for the evaluation is the Mean Squared Error (MSE). The MSE achieved by the SVR and QSVR for each run as well as the average across all the experiments are reported. For each run, the hyperparameters γ and C, used for both the classical and the quantum implementation of the SVR, have been determined empirically. The parameter values used for each QSVR run are $B = 2.5$, $K = 3$ and $P = 2$. The annealer provides as output a set of solutions whose number is selected by the user; in this work 40 solutions for each run were considered. The solutions of the annealer have been combined together through a weighted average in order to produce the final solutions. In this work, 6 different methods for combining the solutions are considered. The basic principle behind each of them is using the same dataset for the training phase to compute the coefficients of a weighted average that are later used to generate the final solution. The examples of the training set are evaluated using the solutions of the annealer and then to each solution a coefficient is assigned, depending on the values of a cost function between the actual values and the predicted ones. The value of the weight coefficient for a given solution is given by the multiplicative inverse of the corresponding value of the cost function. The coefficients of a weighted average must be non-negative and their sum must be equal to 1: the former property is enforced by the choice of the cost functions that are by definition non negative, whereas the latter is achieved by a normalization procedure on the coefficients. Each solution differs by the cost function and the normalization procedure used to determine the coefficients. More precisely:

- QSVR 1: it uses MSE as loss function and each normalized coefficient is obtained by dividing itself by the sum of all the others;

- QSVR 2: it uses MSE as cost function and the softmax function to obtain the coefficients of the weighted average;
- QSVR 3: it uses log-cosh as cost function and calculates the weights by dividing each score by the sum of all the others;
- QSVR 4: it uses log-cosh as cost function and then the softmax function to obtain the coefficients;
- QSVR 5: only the best solution in terms of MSE cost function is considered, i.e., the coefficient associated to the best solution is set to 1 whereas all the remaining ones are set to 0;
- QSVR 6: to each solution the same weight is assigned and is equal to $\frac{1}{M}$, where M is the number of solutions.

For a deeper description of the combination methods, the reader can refer to the repository related to this project.

In the experiments, the classical SVR performed better on average in terms of MSE with respect of its quantum counterpart. This is also due to the fact that, in this preliminary study, a low number of training example is considered, as the quantum annealer presents a tight memory limitation. As expected, the simple average of the solutions, performed by QSVR 6, returned the highest MSE. Among the different combination methods, QSVR 5 showed slightly better results. However, it can be noticed that the difference in MSE between the methods is quite low when compared to the classical SVR. It is also important to point out that, to the best of our knowledge, this is the first attempt to apply such implementation to real-world data, and further analysis is needed, considering a bigger dataset to assess the full potential of QSVR.

4. CONCLUSIONS

In this work, the applicability of QA to the implementation of a SVR using real-world data for training is verified. A QA-based implementation of the SVR has been tested and compared with the classical SVR on a remote sensing dataset for regression. Even though the classical SVR, on average, performed better than the QSVR, this constitutes an important result and opens the path to further research in this field. The next step is focusing on implementing different formulations of the QSVR algorithm, i.e., for the embedding and the energy function definition. Then, an increase in the number of training examples considered at the same time would significantly improve the MSE, nonetheless taking into account the current limit in the number of available qubits, which will eventually improve in the future.

5. REFERENCES

[1] N. R. Draper and H. Smith, *Applied Regression Analysis*. John Wiley & Sons, 1998, vol. 326.

[2] L. Fahrmeir, T. Kneib, S. Lang, and B. Marx, *Regression: Models, Methods and Applications*. Springer-Verlag Berlin Heidelberg, 9 2013, vol. 9783642343339.

[3] D. Tuia *et al.*, "Multioutput Support Vector Regression for Remote Sensing Biophysical Parameter Estimation," *IEEE Geoscience and Remote Sensing Letters*, vol. 8, pp. 804–808, 7 2011.

[4] Q. Cheng, P. K. Varshney, and M. K. Arora, "Logistic Regression for Feature Selection and Soft Classification of Remote Sensing Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, pp. 491–494, 10 2006.

[5] J. Biamonte *et al.*, "Quantum Machine Learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[6] A. Sebastianelli *et al.*, "On circuit-based hybrid quantum neural networks for remote sensing imagery classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 565–580, 2022.

[7] M. Schuld, I. Sinayskiy, and F. Petruccione, "Prediction by linear regression on a quantum computer," *Physical Review A*, vol. 94, no. 2, Aug 2016.

[8] C. H. Yu, F. Gao, and Q. Y. Wen, "An Improved Quantum Algorithm for Ridge Regression," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, pp. 858–866, 3 2021.

[9] C. C. McGeoch, *Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice*. Morgan & Claypool Publishers, 2014.

[10] W. Guan *et al.*, "Quantum Machine Learning in High Energy Physics," *Machine Learning: Science and Technology*, vol. 2, p. 011003, 3 2021.

[11] H. Drucker *et al.*, "Support vector regression machines," in *NIPS*, 1996.

[12] J. Cai, W. G. Macready, and A. Roy, "A Practical Heuristic for Finding Graph Minors," 2014.

[13] J. E. O'Reilly and S. Maritorena, "SeaBAM evaluation data set," *Proc. SeaWiFS Bio-Opt. Algorithm Mini Workshop (SeaBAM)*, 1997.

[14] L. Pasolli, F. Melgani, and E. Blanzieri, "Gaussian Process Regression for Estimating Chlorophyll Concentration in Subsurface Waters From Remote Sensing Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 3, pp. 464–468, 2010.