QUANTUM SUPPORT VECTOR MACHINE ALGORITHMS FOR REMOTE SENSING DATA CLASSIFICATION

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

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

ABSTRACT

Recent developments in Quantum Computing (QC) have paved the way for an enhancement of computing capabilities. Quantum Machine Learning (QML) aims at developing Machine Learning (ML) models specifically designed for quantum computers. The availability of the first quantum processors enabled further research, in particular the exploration of possible practical applications of QML algorithms. In this work, quantum formulations of the Support Vector Machine (SVM) are presented. Then, their implementation using existing quantum technologies is discussed and Remote Sensing (RS) image classification is considered for evaluation.

Index Terms— Quantum computing, quantum circuit model, quantum annealing, quantum machine learning, support vector machine, classification, remote sensing

1. INTRODUCTION

The Support Vector Machine (SVM) is a valuable and widely used algorithm for supervised classification. It is a discriminative supervised learning model, able to handle classification problems with relatively small number of training samples. This feature makes the SVM particularly suitable for Remote Sensing (RS) [1], and its effectiveness has been proven in several applications, e.g., multispectral and hyperspectral image classification [2].

In the last years, the interest towards Quantum Computing (QC) has visibly increased, also due to the development of the first prototypes of quantum computers. The basic unit of quantum computers is a quantum system called *qubit* [3], represented by the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle$ and $|1\rangle$ are the two basis states of the system and α, β the coefficients of their linear combination, i.e., their *quantum superposition*. By definition, the possible states are infinite and subject to $\alpha, \beta \in \mathbb{C}$, $|\alpha|^2 + |\beta|^2 = 1$. However, the state of a qubit cannot be directly measured. In addition, a measurement process applied to a qubit also changes the state of the qubit itself. It can become either $|0\rangle$ or $|1\rangle$ with probability $|\alpha|^2$ and $|\beta|^2$. This state is also called *post-measurement state*.

The principles of QC are the cornerstone of *quantum supremacy*, i.e., the ability to solve problems that are considered intractable in classical computation. Recently, quantum supremacy has been experimentally demonstrated for a highly theoretical model [4].

Quantum algorithms are based on specific computational models. In this work we will focus on the *circuit model* and the *adiabatic model*. The quantum circuit model is a transposition of the classical circuit model. In fact, quantum gates, also represented in matrix form, are designed to process qubits, so that any of the infinite possible inputs can be handled. On the other hand, Adiabatic Quantum Computation (AQC) aims at solving optimization problems by exploiting the time evolution of a quantum mechanical system satisfying the requirements of the adiabatic theorem and ruled by the Schrödinger's Equation:

$$i\hbar \frac{\partial |\psi(t)\rangle}{\partial t} = H(t)\,|\psi(t)\rangle \tag{1}$$

where $\hbar$ is the reduced Planck constant, $|\psi(t)\rangle$ is the quantum state of the system varying in time and $H(t)$ is the Hamiltonian describing the system [5][6]. $H(t = t_{max})$ (i.e., the final Hamiltonian) encodes the objective function, while the ground state of the system (i.e., the minimum energy state) encodes the optimal solution. Quantum Annealing (QA) is closely related to AQC and describes a heuristic search approach for optimization [7].

Quantum Machine Learning (QML) has established itself as a successful interdisciplinary field which aims at enhancing

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. Part of this work was performed in the CoE RAISE project receiving funding from the European Union's Horizon 2020 Research and Innovation Framework Programme H2020-INFRAEDI-2019-1 under grant agreement no. 951733.

classical Machine Learning (ML) algorithms by outsourcing complex computations to a quantum computer, outperforming classical computation. Both theoretical and practical results have shown major improvements in ML tasks that are expected to be exploited in the near future, e.g., for big data applications [8].

The Quantum Support Vector Machine (QSVM) is the class of algorithms that is considered in this work. In [9], we have already trained a QA-based QSVM algorithm on the D-Wave 2000Q quantum annealer and tested it on RS multispectral images classification problems. In this paper, we run the experiments on the D-Wave's next-generation quantum processor Advantage. It can process larger datasets than the previous system, since it has about 2.5 times more qubits and a higher connectivity (up to 15 connections per qubit, instead of up to 6 connections per qubit). Furthermore, in this work we also run classification experiments with a QSVM implementation based on the quantum circuit model. The purpose is to show how RS can benefit from existing quantum technologies and which results can be obtained. The documented codes and a comprehensive bibliography of the articles related to QSVM are available on the repository of this work[1].

2. CIRCUIT-BASED QSVM

2.1. Quantum Matrix Inversion

A number of different circuit-based approaches to QSVM have been proposed [10][11][12]. The most celebrated algorithm, presented by Rebentrost *et al.* in [10], is a fully quantum algorithm with time complexity $O(\log(Nd))$, where N is the number of training samples and d their dimension. The Least Squares Support Vector Machine (LS-SVM) formulation is considered [13], and it is shown that the dual problem can be rewritten in matrix form. The algorithm consists in the adaptation of the HHL algorithm [14], an efficient quantum matrix inversion algorithm. The reader can refer to the original paper [10] for a complete description. A simplified version can be found in [11]. An implementation of the general case is unfeasible with current hardware.

2.2. Quantum-enhanced Feature Spaces

In a more recent work [12], quantum technology is used for an efficient calculation of the kernel matrix. A quantum feature map $\mathcal{U}_{\Phi(\mathbf{x})}$ is constructed, i.e., a variational circuit that encodes a vector $\mathbf{x}$ into the quantum state $|\Phi(\mathbf{x})\rangle$ [15]. The quantum feature map is applied to an initial state $|0\rangle^n$, with n number of qubits, equal to the dimension of the samples. The feature map can be chosen according to the desired complexity and accuracy. An advantage can be achieved when the feature map is difficult to compute classically. The structure of the feature maps are described in detail in [12]. Several implementations of the algorithm and the feature maps are available.

[1] https://gitlab.version.fz-juelich.de/cavallaro1/quantum_svm_algorithms

3. QA-BASED QSVM

The QSVM algorithm proposed by Willsch *et al.* in [16] is a first attempt to employ QA technologies in the training phase of a SVM. It is formulated as a Quadratic Unconstrained Binary Optimization (QUBO) problem[2], which is a necessary condition for an algorithm to be executed on a D-Wave quantum annealer.

3.1. Classical SVM

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 $y_n \in \{-1, 1\}$ are the binary labels, the conventional dual formulation of a SVM can be defined as the following Quadratic Programming (QP) problem [17]:

$$\min_{\boldsymbol{\alpha}} L(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{nm} \alpha_n \alpha_m y_n y_m k\left(\mathbf{x}_n, \mathbf{x}_m\right) - \sum_n \alpha_n, \quad (2)$$

subject to the constraints

$$0 \leq \alpha_n \leq C \quad ; \quad \sum_n \alpha_n y_n = 0, \quad (3)$$

where $\boldsymbol{\alpha} = \{\alpha_n : n = 0, ..., N-1\}$ are the Lagrange multipliers, i.e., the variables of the optimization problem, $k(\mathbf{x}_n, \mathbf{x}_m)$ is the kernel function and C is the regularization parameter.

3.2. QUBO Problem

QUBO is based on the minimization of a quadratic energy function defined as:

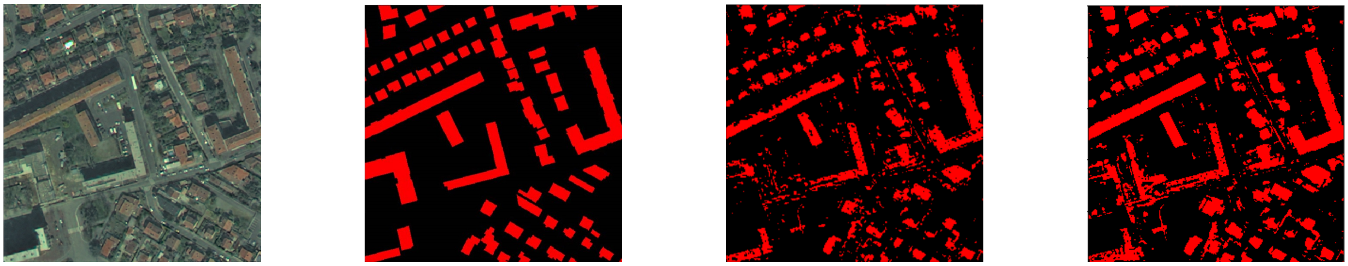
$$E = \sum_{i \leq j} a_i Q_{ij} a_j, \quad (4)$$

where $a_i \in \{0, 1\}$ are the binary variables of the optimization problem and Q the QUBO weight matrix, i.e., an upper-triangular matrix of real numbers. For the formulation, the required steps are (i) the encoding of the solution space to a vector of binary variables and (ii) the definition of the QUBO weight matrix Q. Since the solution of Eqs. (2)-(3) consists of real numbers $\alpha_n \in \mathbb{R}$, the following encoding is used:

$$\alpha_n = \sum_{k=0}^{K-1} B^{k-P} a_{Kn+k}, \quad (5)$$

where $a_{Kn+k} \in \{0, 1\}$ are binary variables, K is the number of binary variables to encode α_n, B is the base used for

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



(a) (b) (c) OA=0.876, F1=0.697 (d) OA=0.874, F1=0.734

Fig. 1: Dataset: (a) false color image and (b) ground truth. SVM classification maps: (c) classical and (d) QA-based.

the encoding, and $P \geq 0$ is a parameter that allows for negative exponents. Eqs. (2)-(3) can be now reformulated in the form of an energy function, as in Eq. (4), by substituting the variables α_i with their encoding, and introducing a squared penalty term multiplied by $\frac{\xi}{2}$ for the second contraint, obtaining the following result:

$$E = \sum_{n,m=0}^{N-1} \sum_{k,j=0}^{K-1} a_{Kn+k} \widetilde{Q}_{Kn+k,Km+j} a_{Km+j}, \tag{6}$$

where $\widetilde{Q}$ is a matrix of size $KN \times KN$ given by

$$\begin{aligned} \widetilde{Q}_{Kn+k,Km+j} = \frac{1}{2} B^{k+j-2P} y_n y_m \left(k\left(\mathbf{x}_n, \mathbf{x}_m\right) + \xi\right) \\ - \delta_{nm} \delta_{kj} B^{k-P}. \end{aligned} \tag{7}$$

Since the QUBO weight matrix Q is upper-triangular by definition and $\widetilde{Q}$ is symmetric, a last step is performed by computing $Q_{ij} = \widetilde{Q}_{ij} + \widetilde{Q}_{ji}$ for $i < j$ and $Q_{ii} = \widetilde{Q}_{ii}$.

3.3. Minor Embedding

The QUBO matrix is required for the final step of the problem, called *minor embedding* [18]. The quantum annealer is configured in order to obtain the desired results, i.e., the ground state of the system is an optimal solution of the problem. To each binary variable a_i is assigned either a qubit or a *chain* of qubits linked by physical connections called *couplers*. Each element of the QUBO matrix Q_{ij} represents a logical relation between the variables a_i and a_j, which implies also a physical connection between the respective qubit chains. The main hardware limitation is that not all the qubits are physically linked with each other, so maintaining the logical structure of the problem is an essential requirement for a correct embedding. More detailed information about the proposed QSVM algorithm is given in [16].

4. EXPERIMENTAL RESULTS

4.1. Dataset

For the experiments, SemCity Toulouse [19] is considered, i.e., a multispectral image dataset focused on the city of Toulouse. An urban area of $500 \cdot 500 = 250000$ pixels is chosen as the test set, 50 samples being selected for training. Each pixel is represented by a feature vector $\mathbf{x}_n$ of dimension $d = 8$, and to each pixel a label $y_n \in \{-1, 1\}$ is assigned, indicating the absence or the presence of a building.

4.2. Experimental Setup

At present, two of the main providers of cloud QC environments are IBM and D-Wave Systems. *IBM Quantum Experience*[3] is a platform where users can access and program IBM quantum processors and simulators. The considered machines are `ibmq_qasm_simulator`, a 32 qubits general quantum simulator, and `ibmq_16_melbourne`, a 15 qubits quantum system, which is the largest available in the free usage plan. An implementation of the QSVM algorithm is already included in the IBM Qiskit Aqua library following the formulation in [12]. *D-Wave Leap*[4] offers access to D-Wave quantum annealers. In this experiment, the latest machine devised by D-Wave Systems is considered, i.e., D-Wave Advantage, which has more than 5000 qubits and 35000 couplers and is based on the Pegasus qubit architecture. Computing time on the D-Wave Advantage system for this experiment is provided through the Jülich UNified Infrastructure for Quantum computing (JUNIQ)[5]. The results are compared to the classification map generated by the classical SVM available in the Scikit-learn library in Python, run on an Intel Core i5-7200U processor.

4.3. Evaluation

The obtained results are shown in Fig. 1 and are evaluated considering two metrics, i.e., overall accuracy (OA), computed as the ratio of the number of correct predictions and the total number of predictions, and F1 score (F1), computed as the harmonic mean between precision and recall.

First, the classical SVM algorithm has been trained. A calibration has been performed varying the kernel (linear and RBF), the regularization parameter C and the coefficient γ for

[3] `https://quantum-computing.ibm.com/`

[4] `https://cloud.dwavesys.com/leap/`

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

the RBF kernel. The best results are achieved with a linear kernel and $C = 11$, obtaining $OA = 0.876$ and $F1 = 0.697$.

Several experiments have been launched on the IBM machines. However, a number of limitations have arisen. The `ibmq_16_melbourne` quantum system presented a much longer queue time w.r.t. the quantum simulator. In both cases, no result could be achieved in a reasonable amount of time for the complete test set of 250000 samples. This problem was solved by selecting a smaller area of $15 \cdot 15 = 225$ pixels for testing, bearing in mind that this choice negatively affects statistical significance. By choosing a second-order Pauli-Z evolution circuit with $\phi(x) = x$ and $\phi(x, y) = xy$, `ibmq_qasm_simulator` generated a prediction map with $OA = 0.609$ and $F1 = 0.569$.

The experiments of the QA SVM were also preceded by a calibration of the algorithm. The setup $B = 3$, $K = 2$, linear kernel, $\xi = 5$ and $P = 0$ leads to $OA = 0.874$ and $F1 = 0.734$, comparable to the results obtained with the classical SVM. No splitting is needed, as the memory of D-Wave Advantage is sufficient for the usage of the whole considered training set.

5. CONCLUSIONS

In this work, the potential of QC for RS has been shown. The current state of circuit-based algorithms and hardware lacks the capability to handle a sufficient amount of data for SVM classification in RS. The QA-based SVM implementation, on the other hand, has proven to produce results comparable to the classic version of the SVM, which makes QA more suitable for near-term applications in RS classification.

6. REFERENCES

[1] G. Mountrakis, J. Im, and C. Ogole, “Support Vector Machines in Remote Sensing: A Review,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 3, pp. 247–259, 2011.

[2] F. Melgani and L. Bruzzone, “Classification of Hyperspectral Remote Sensing Images with Support Vector Machines,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[3] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011.

[4] F. Arute *et al.*, “Quantum Supremacy using a Programmable Superconducting Processor,” *Nature*, 2019.

[5] T. Kadowaki and H. Nishimori, “Quantum Annealing in the Transverse Ising Model,” *Physical Review E*, vol. 58, no. 5, p. 5355–5363, 1998.

[6] A. Finnila *et al.*, “Quantum Annealing: A New Method for Minimizing Multidimensional Functions,” *Chemical Physics Letters*, vol. 219, no. 5, pp. 343–348, 1994.

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

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

[9] G. Cavallaro *et al.*, “Approaching Remote Sensing Image Classification with Ensembles of Support Vector Machines on the D-Wave Quantum Annealer,” in *Proceedings of the IEEE IGARSS*, 2020, pp. 1973–1976.

[10] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum Support Vector Machine for Big Data Classification,” *Physical Review Letters*, vol. 113, no. 13, 2014.

[11] Z. Li, X. Liu, N. Xu, and J. Du, “Experimental Realization of a Quantum Support Vector Machine,” *Phys. Rev. Lett.*, vol. 114, p. 140504, 2015.

[12] V. Havlíček *et al.*, “Supervised Learning with Quantum-Enhanced Feature Spaces,” *Nature*, vol. 567, no. 7747, p. 209–212, 2019.

[13] J. A. K. Suykens and J. Vandewalle, “Least Squares Support Vector Machine Classifiers,” *Neural Process. Lett.*, vol. 9, no. 3, p. 293–300, 1999.

[14] A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum Algorithm for Linear Systems of Equations,” *Phys. Rev. Lett.*, vol. 103, p. 150502, 2009.

[15] M. Schuld and N. Killoran, “Quantum Machine Learning in Feature Hilbert Spaces,” *Physical Review Letters*, vol. 122, no. 4, Feb 2019.

[16] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen, “Support Vector Machines on the D-Wave Quantum Annealer,” *Computer Physics Communications*, vol. 248, p. 107006, 2020.

[17] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 2007.

[18] J. Cai, W. G. Macready, and A. Roy, “A Practical Heuristic for Finding Graph Minors,” 2014.

[19] R. Roscher *et al.*, “SemCity Toulouse: A Benchmark for Building Instance Segmentation in Satellite Images,” *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-5-2020, pp. 109–116, 2020.