APPROACHING REMOTE SENSING IMAGE CLASSIFICATION WITH ENSEMBLES OF SUPPORT VECTOR MACHINES ON THE D-WAVE QUANTUM ANNEALER

Gabriele Cavallaro[1], *Dennis Willsch*[1,2], *Madita Willsch*[1,2], *Kristel Michielsen*[1,2] *and Morris Riedel*[1,3]

[1] Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
[2] RWTH Aachen University, D-52056 Aachen, Germany
[3] School of Engineering and Natural Sciences, University of Iceland, Iceland

ABSTRACT

Support Vector Machine (SVM) is a popular supervised Machine Learning (ML) method that is widely used for classification and regression problems. Recently, a method to train SVMs on a D-Wave 2000Q Quantum Annealer (QA) was proposed for binary classification of some biological data. First, ensembles of weak quantum SVMs are generated by training each classifier on a disjoint training subset that can be fit into the QA. Then, the computed weak solutions are fused for making predictions on unseen data. In this work, the classification of Remote Sensing (RS) multispectral images with SVMs trained on a QA is discussed. Furthermore, an open code repository is released to facilitate an early entry into the practical application of QA, a new disruptive compute technology.

Index Terms— Quantum computation, quantum annealing, support vector machine, classification, multispectral image, remote sensing

1. INTRODUCTION

Support Vector Machines (SVMs) are non-parametric statistical approaches that can be adopted to analyze data and recognize patterns for supervised classification and regression problems [1]. They rely on the margin maximization principle to be less sensitive to overfitting [2] and they adopt the kernel trick [3] to generate nonlinear decision functions. In contrast to Deep Learning (DL) models [4], which require large amounts of training data, SVMs are generally used when only small sets of training samples are available. They have been intensively used in conjunction with handcrafted features for solving RS classification problems [5]. Furthermore, SVMs can be effectively combined with DL classifiers: they can be either placed on top of deep neural networks to classify the extracted features [6] or used for the computation of the filter weights of Convolutional Neural Networks (CNNs) [7].

The advances of ML and Quantum Computing (QC) open possibilities to address previously untenable problems. Their intersection attracted the attention of researchers to study a combination of both fields, termed quantum ML [8]. Concepts such as superposition and entanglement are considered to make quantum computers much faster than conventional computers for certain computational tasks [9]. The two main paradigms of QC are the gate-based quantum computer and the Quantum Annealer (QA). Various quantum computers and annealers with different degrees of technological maturity are available today and to some of them access is provided by e.g., IBM, Rigetti Computing, Google, IonQ, and D-Wave Systems. Among those, the D-Wave 2000Q (DW2000Q) QA has reached an adequate maturity level of QC technology (i.e., QTRL8 [1]) for studying prototype practical applications.

In [10], a method is proposed to train SVMs classifiers on a DW2000Q QA [11]. The SVM optimization problem is expressed as a Quadratic Unconstrained Binary Optimization (QUBO) problem. First, ensembles of quantum weak SVMs are generated by training each classifier on a small disjoint training subset. Splitting the whole training set is necessary because of the limited connectivity of the Chimera graph architecture of the DW2000Q quantum processor [11]. Next, the trained ensemble is used to classify an arbitrary number of unseen samples.

The contribution of this work is twofold: on the one hand, the classification of RS images is approached with the quantum ensemble of SVMs. To do so, experiments are conducted on two labeled multispectral datasets that are available to the public. On the other hand, an open repository that includes the Python code and the processing pipeline documented in a Jupyter notebook is released [2]. Since everyone can make a free account on the DW2000Q QA [3], this enables reproducing the classification results and facilitates an early entry into the practical application of QA, a new disruptive compute technology, in general.

The authors would like to thank Prof. Dr. Kristel Michielsen for providing computing time on the D-Wave 2000Q quantum annealer, in the context of the project JUNIQ (Jülich UNified Infrastructure for Quantum computing).

[1] https://www.fz-juelich.de/ias/jsc/EN/Research/ModellingSimulation/QIP/QTRL/_node.html

[2] https://gitlab.version.fz-juelich.de/cavallaro1/svm_quantum-annealer.git

[3] https://www.dwavesys.com/take-leap

2. SUPPORT VECTOR MACHINES ON A QUANTUM ANNEALER

2.1. Background

A SVM learns its parameters from a set of annotated training samples $D = \{(\mathbf{x}_n, y_n) : n = 0, \ldots, N-1\}$, with $\mathbf{x}_n \in \mathbb{R}^d$ being a feature vector and y_n its label. A SVM separates the samples of different classes in their feature space by tracing maximum margin hyperplanes. The training consists of solving a Quadratic Programming (QP) problem [12]:

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

subject to

$$0 \leq \alpha_n \leq C \;\; ; \;\; \sum_n \alpha_n y_n = 0. \quad (2)$$

For N coefficients $\alpha_n \in \mathbb{R}$, where C is a regularization parameter and $k(.,.)$ is a kernel function that enables a SVM to compute non-linear decision functions (by means of the kernel trick [3]). The type of kernel function which is most commonly used is the Radial Basis Function (RBF) [3]: $\mathrm{rbf}\left(\mathbf{x}_n, \mathbf{x}_m\right) = e^{-\overline{\gamma}\|\mathbf{x}_n - \mathbf{x}_m\|^2}$. The SVM decision boundary is based on the samples corresponding to $\alpha_n \neq 0$ (i.e., support vectors). The prediction for an arbitrary sample $\mathbf{x} \in \mathbb{R}^d$ can be made by evaluating the decision function (i.e., signed distance between the sample $\mathbf{x}$ and the decision boundary)

$$f(\mathbf{x}) = \sum_n \alpha_n y_n k\left(\mathbf{x}_n, \mathbf{x}\right) + b, \quad (3)$$

where the bias b can be computed by [12]

$$b = \frac{\sum_n \alpha_n \left(C - \alpha_n\right)\left[y_n - \sum_m \alpha_m y_m k\left(\mathbf{x}_m, \mathbf{x}_n\right)\right]}{\sum_n \alpha_n \left(C - \alpha_n\right)}. \quad (4)$$

The class label for $\mathbf{x}$ predicted is $\widetilde{y} = \mathrm{sign}(f(\mathbf{x}))$.

2.2. Quantum SVM

The DW2000Q QA requires the SVM training to be formulated as a QUBO problem [4] [10], which is defined as the minimization of the energy function

$$E = \sum_{i \leq j} a_i Q_{ij} a_j, \quad (5)$$

with $a_i \in \{0, 1\}$ the binary variables of the optimization problem, and Q the QUBO weight matrix (i.e., an upper-triangular matrix of real numbers). Since the solution of Eqs. (1)-(2) consists of real numbers $\alpha_n \in \mathbb{R}$ and Eq. (5) can only compute discrete solutions, the following encoding is used:

$$\alpha_n = \sum_{k=0}^{K-1} B^k a_{Kn+k}, \quad (6)$$

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

where $a_{Kn+k} \in \{0, 1\}$ are binary variables, K is the number of binary variables to encode α_n, and B is the base used for the encoding [10].

The formulation of the QP of Eqs. (1)-(2) as QUBO is obtained through the encoding of Eq. (6) and the introduction of a multiplier ξ to include the first constraint of Eq. (2) as a squared penalty term:

$$\begin{aligned} E = &\frac{1}{2}\sum_{nmkj} a_{Kn+k} a_{Km+j} B^{k+j} y_n y_m k\left(\mathbf{x}_n, \mathbf{x}_m\right) \\ &- \sum_{nk} B^k a_{Kn+k} + \xi \left(\sum_{nk} B^k a_{Kn+k} y_n\right)^2 \quad (7) \\ = &\sum_{n,m=0}^{N-1} \sum_{k,j=0}^{K-1} a_{Kn+k} \widetilde{Q}_{Kn+k,Km+j} a_{Km+j}, \quad (8) \end{aligned}$$

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

$$\begin{aligned} \widetilde{Q}_{Kn+k,Km+j} = &\frac{1}{2} B^{k+j} y_n y_m \left(k\left(\mathbf{x}_n, \mathbf{x}_m\right) + \xi\right) \\ &- \delta_{nm} \delta_{kj} B^k. \end{aligned} \quad (9)$$

Since $\widetilde{Q}$ is symmetric, the upper-triangular QUBO matrix Q is defined by $Q_{ij} = \widetilde{Q}_{ij} + \widetilde{Q}_{ji}$ for $i < j$ and $Q_{ii} = \widetilde{Q}_{ii}$. The second constraint of Eq. (2) is automatically included in Eq. (8) through the encoding given in Eq. (6), since the maximum for α_n is given by

$$C = \sum_{k=1}^{K} B^k. \quad (10)$$

The last step required to run the optimization on the DW2000Q QA is the embedding procedure [13]. This is necessary because the QUBO problem given in Eq. (5) includes some couplers $Q_{i,j} \neq 0$ between qubit i and qubit j for which no physical connection exists on the chip (i.e., constraint of the Chimera topology of the DW2000Q quantum processor) [10]. The embedding increases the number of logical connections between the qubits. When no embedding can be found, the number of nonzero couplers n_{cpl} is the parameter that can be reduced until an embedding is found. More detailed information about the quantum SVM is given in [10].

3. EXPERIMENTAL RESULTS

3.1. Dataset

The experiments are carried out on the two multispectral RS datasets listened in Table 1. They are free and available at HyperLabelMe [14], which is a web platform that contains 43 image datasets and allows automatic benchmarking of different classifiers. For each dataset, training data pairs (spectra and land cover/use labels) and test data spectra are provided.

The test labels are not accessible since the predictions need to be uploaded to HyperLabelMe, which returns the accuracy, different scores and a ranked list of the best methods from different users.

Table 1: Datasets used for the classification experiments on the DW2000Q QA [14].

ID	Sensor	Data points	Train Samples	Classes
Im16	Landsat	200×200×7	500	2
Im40	Landsat	200×200×7	500	2

3.2. Setup

The experiments are conducted on the DW2000Q QA [13]. D-Wave's Ocean Software [5] is the Python library that is used for generating the embeddings and the executions on the system.

The DW2000Q QA computes a variety of close-to-optimal solutions (i.e., different coefficients $\{\alpha_n\}^{(i)}$ obtained from Eq. (6)). Many of these solutions may have a slightly higher energy than the global minimum $\{\alpha_n\}^*$ that can be found by the classical SVM. However, these solutions can still solve the classification problem for the training data. For each run on the DW2000Q QA, the 20 lowest energy samples from 10,000 reads are kept.

The classification pipeline includes three different phases: calibration, training, and testing [10]. For both datasets of Table 1, the training samples are used for both the calibration and training phase.

3.2.1. Calibration Phase

The SVM on the QA depends on four hyperparameters: the encoding base B, the number K of quantum bits (qubits) per coefficient α_n, the multiplier ξ, and the kernel parameter γ. The parameter n_{cpl} varies for each run and is not a parameter of the SVM itself. The hyperparameters are selected through a 10-fold cross-validation. Each training set includes only 30 samples (i.e., choice due to the limitations of the QA [10]). The validation includes the remaining samples that are used for the evaluation of the performance. For each dataset, the values are calibrated by evaluating the SVM for $B \in \{2, 3, 5, 10\}$, $K \in \{2, 3\}$, $\xi \in \{0, 1, 5\}$, and $\gamma \in \{-1, 0.125, 0.25, 0.5, 1, 2, 4, 8\}$.

3.2.2. Training Phase

To overcome the problem of the limited connectivity of the Chimera graph of the DW2000Q QA the whole training set is split into small disjoint subsets $D^{(train,l)}$ of 40 samples, with $l = 0, ..., int(N/40)$. The strategy is to build an ensemble of quantum weak SVMs (qeSVMs) where each classifier is trained on $D^{(train,l)}$. This is achieved in two steps. First, for each subset $D^{(train,l)}$ the twenty best solutions from the DW2000Q QA (i.e., qSVM$(B, K, \xi, \gamma)\#i$ for $i = 0, ..., 19$) are combined by averaging over the respective decision functions $f^{l,i}(\mathbf{x})$ (see Eq. (3)). Since the decision function is linear in the coefficients and the bias $b^{(l,i)}$ is computed from $\alpha_n^{(l,i)}$ via Eq. (4), this procedure effectively results in one classifier with an effective set of coefficients $\alpha_n^{(l)} = \sum_i \alpha_n^{(l,i)}/20$ and bias $b^l = \sum_i b^{(l,i)}/20$. Second, an average is made over the $int(N/40)$ subsets. Note, however, that the data points $\left(\mathbf{x}_n^{(l)}, y_n^{(l)}\right) \in D^{(\text{train },l)}$ are now different for each l. The full decision function is

$$F(\mathbf{x}) = \frac{1}{L} \sum_{nl} \alpha_n^{(l)} y_n^{(l)} k\left(\mathbf{x}_n^{(l)}, \mathbf{x}\right) + b, \tag{11}$$

where $b = \sum_l b^{(l)}/L$. As before, the decision for the class label of a point $\mathbf{x}$ is obtained through $\widetilde{t} = \text{sign}(F(\mathbf{x}))$.

[5] `https://docs.ocean.dwavesys.com`

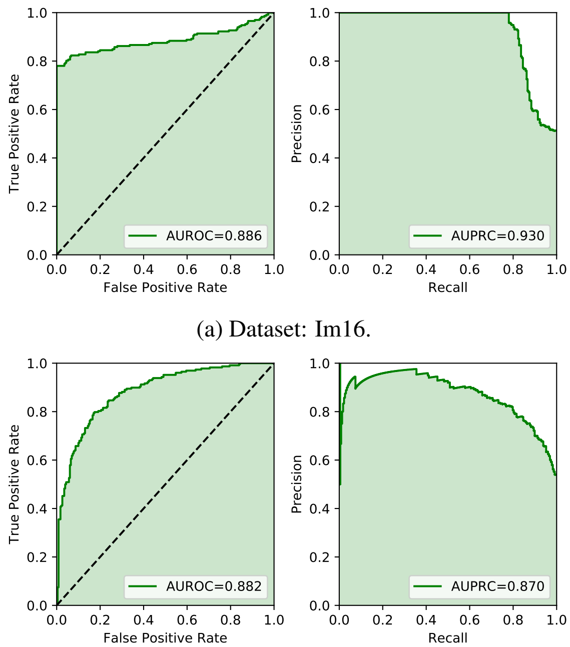
Table 2: Classification results obtained with the DW2000Q QA for the data sets listed in Table 1.

ID dataset	OA	Kappa	z-score
16	94.26	0.89	134.71
40	78.90	0.57	47.21

3.3. Testing Phase

The two selected multispectral RS datasets listed in Table 1 include 500 annotated training samples. This limited amount of annotated training samples make the data sets suitable for analysis by a DW2000Q QA. As explained in Sec. 3.2.2, the training data is split into small disjoint subsets of 40 samples (i.e., for both datasets, $int(N = 500/40) = 12$ subsets) to enable the D-Wave's Ocean SDK to find the embedding.

The performance of the qeSVMs is evaluated directly on HyperLabelMe [14]. Table 2 reports the computed metrics for both datasets. These are the common metrics Overall Accuracy (OA), Cohen's Kappa coefficient (Kappa) and standard score (z-score). Furthermore, the plots of Fig. 1 depict the Receiver Operating Characteristic (ROC) and Precision (PR) curves for a single trained SVM on the subset $D^{(train,l=1)}$. These are more robust metrics that are not based on a single evaluation of the classifier, but rather on the performance of the classifier as a function of the bias b in Eq. (4) (more details are given in [10]). For the training subset $D^{(train,l=1)}$, Fig. 1 shows that the qSVM can compute a robust classifier since it computes almost optical curves for both datasets.



(a) Dataset: Im16.

(b) Dataset: Im40.

Fig. 1: The ROC and PR curves are plotted for qSVM($B = 2, K = 3, \xi = 1, \gamma = 0.250$)#0.

4. CONCLUSIONS AND OUTLOOK

The classification of RS multispectral images with SVMs on a QA was studied. The setup of the procedure and the description of the framework are available on the repository of this work. It can be concluded that the QA version of the SVM is a valuable alternative to the classical SVM for RS classification problems where a limited number of training examples is available.

Future research will focus on conducting more systematic comparisons between quantum and classical SVMs. This will include the training of suboptimal ensembles of both classifiers on small datasets.

5. 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] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, 2nd ed. Springer-Verlag New York, 2002.

[3] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.

[4] L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, “Deep Learning in Remote Sensing Applications: A Meta-Analysis and Review,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 152, pp. 166–177, 2019.

[5] U. Maulik and D. Chakraborty, “Remote Sensing Image Classification: A survey of support-vector-machine-based advanced techniques,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 1, pp. 33–52, 2017.

[6] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, “Factors of Transferability for a Generic ConvNet Representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1790–1802, 2016.

[7] G. Cavallaro, Y. Bazi, F. Melgani, and M. Riedel, “Multi-Scale Convolutional SVM Networks for Multi-Class Classification Problems of Remote Sensing Images,” in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 875–878.

[8] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[9] F. Arute, K. Arya, and A. Et, “Quantum Supremacy Using a Programmable Superconducting Processor,” *Nature*, vol. 574, pp. 505–510, 2019.

[10] D. Willsch, M. Willsch, H. De Raedt, and K. Michielsen, “Support Vector Machines on the D-Wave Quantum Annealer,” 2019.

[11] D-Wave Systems Inc, “Technical Description of the D-Wave Quantum Processing Unit,” Burnaby, BC, Canada, 2019.

[12] 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.

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

[14] J. Munoz-Mari, E. Izquierdo-Verdiguier, M. Campos-Taberner, A. Perez-Suay, L. Gomez-Chova, G. Mateo-Garcia, A. B. Ruescas, V. Laparra, J. A. Padron, J. Amoros-Lopez, and G. Camps-Valls, “HyperLabelMe : A Web Platform for Benchmarking Remote-Sensing Image Classifiers,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 79–85, 2017.