Training Support Vector Machines: a Quantum-Computing Perspective

Davide Anguita, Sandro Ridella, Fabio Rivieccio, and Rodolfo Zunino
DIBE – Dept. of Biophysical and Electronic Engineering - University of Genoa
Via Opera Pia 11a - 16145 Genova - Italy

Abstract – Recent advances in characterizing the generalization ability of Support Vector Machines (SVMs) exploit refined concepts, such as Rademacher estimates of model complexity and nonlinear criteria for weighting empirical classification errors. Those methods improve the SVM representation ability and tighten generalization bounds. On the other hand, Quadratic-Programming algorithms are no longer applicable, hence the SVM-training process cannot benefit from the notable efficiency featured by those specialized techniques. The paper considers the possibility of using Quantum Computing to solve the resulting problem of effective optimization, especially in the case of digital SVM implementations. The behavioral aspects of conventional and enhanced SVMs are compared, supported by experiments in both a synthetic and a real-world problem. Likewise, the related differences between Quadratic-Programming and Quantum-based optimization techniques are analyzed.

I. INTRODUCTION

The Support Vector Machine (SVM) [1] is a well-known and effective method for regression and pattern classification, and often leads to outstanding performances in real-world applications. The success of SVMs mainly derives from setting up the training process so as to optimize the run-time generalization performances of the resulting classifiers.

The key feature of Vapnik’s formulation [2] lies in posing the maximum-margin search process as a Quadratic-Programming (QP) optimization problem. In spite of the intricacies brought about by highly constrained (and often poorly conditioned) QP, effective tools are available for fast QP optimization [3]. This has ultimately boosted the practical impact of SVM classifiers.

At the same time, a vast literature in the area of Computational Learning Theory reports the search for newer, tighter bounds to the classifiers’ generalization errors. In this respect, sample-based methods that use maximal-discrepancy techniques to estimate model complexity [4,5] seem to represent a promising line of research. The notably tight generalization bounds attained in [5] result from combining two specialized approaches: a Rademacher estimate of model complexity and an advanced, nonlinear criterion for weighting empirical classification errors.

The research presented in this paper exploits these recent achievements as a single basic approach to SVM training. The paper first demonstrates the advantage of the error-weighting criterion for SVM training: the overall classifier is made robust to peculiar distributions that might divert the conventional error-weighting criterion.

On the other hand, a crucial issue raised by the nonlinear error-weighting approach is that the SVM training process cannot any longer be formulated as a conventional QP problem. Several optimization methods exist for the general case [6,7], yet the lack of an efficient algorithm such as QP can turn optimization into a problem with NP complexity.

This scenario leads one envision to exploit novel technologies for effective optimization. Quantum Computing (QC) [8] represents a promising paradigm, whose importance has increased very rapidly in the last decades, mainly for the recent definition of specialized algorithms to solve complex problems, such as large-number factorization and exhaustive search.

A basic feature that makes quantum approaches appealing to applied research is that QC involves a digital representation of processed information. This proves especially useful in training SVMs for two reasons: first, the overall problem is inherently digital in both quantum and classical computers; secondly, the optimization process has to scan exhaustively the set of possible bit configurations in the search space.

Therefore, the paper explores the possibility of using quantum-optimization algorithms for SVM training when conventional QP techniques are no longer applicable. The effectiveness of QC-based optimization is evaluated in both a synthetic and a real-world problem, and the performances are compared with those of a Montecarlo random-search method.

II. ERROR WEIGHTING FOR TRAINING SVM CLASSIFIERS

The reason that justifies the success of the SVM model lies in its structural approach. SVM training aims to find a function capable of incurring few errors on the training sample, while featuring a promising generalization ability.

Let \( \Phi \) be a mapping function from the input space \( X \) into a higher-dimensional space; then the general form for a hyperplane in the mapped space is:

\[
 f(x) = \sum_{i=1}^{np} \alpha_i y_i \Phi(x_i) \cdot \Phi(x) + b \tag{1}
\]

where the function is expressed as a weighted sum of the input samples, \( \{x_i, i=1, \ldots, np\} \), and the \( \alpha_i \) are positive bounded quantities. If we regard (1) as a classification surface...
and label by \( y \), the class associated with each input point \( x \), the general statement of the problem sketched above is:

\[
\begin{align*}
\min_{w, \xi \in \mathbb{R}^n} & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{np} \xi_i \\
\text{subject to} : & \\
& y_i (w \cdot \Phi_i + b) = 1 + \mu_i - \xi_i \quad \forall i = 1 \ldots np \\
& \mu_i, \xi_i \geq 0
\end{align*}
\]

where

\[
w = \sum_{i=1}^{np} \alpha_i y_i \Phi_i
\]

\( C \) is a constant, \( \mu \) are used to balance the equation in the case of a correct classification, and \( \xi \) is an analog measure of the error on each data-point. One of the main results of Statistical Learning Theory [2] is that the first term considered in (2) is proportional to the VC-dimension, hence its minimization enhances the generalization ability of the hyperplane in (1).

It is possible to demonstrate [7] that (2) has the same solution as a constrained QP optimization problem (the dual one) with respect to the \( \alpha_i \); in such a problem these variables are bounded by \( C \) and linearly constrained by the following relation:

\[
\sum_{i=1}^{np} \alpha_i y_i = 0
\]

In order to find the solution of the dual problem it is crucial to meet the Karush-Kuhn-Tucker (KKT) conditions:

\[
\begin{align*}
\alpha_i \mu_i &= 0 \\
(C - \alpha_i) \xi_i &= 0 \quad \forall i = 1 \ldots np
\end{align*}
\]

The formulation of the dual only involves the computation of the inner product of the \( \Phi_i \). The functions for which

\[
k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)
\]

are called Kernel Functions. These functions, together with the \( \alpha_i \), define the SVM expansion in (1), which becomes

\[
f(x) = \sum_{i=1}^{np} \alpha_i y_i k(x_i, x) + b
\]

thus allowing a non-linear class separation.

Among the several available methods to estimate the classification error, the ones with the tightest bounds seem to be those performing a sample-based complexity estimation [5]. This estimate involves the computation of the Rademacher Complexity, as follows:

\[
R_{np}(F) = \mathbb{E}_{P(x)} \left[ \mathbb{E}_{\sigma \sim \mathcal{D}^n} \left[ \sup_{f \in F} \frac{1}{np} \sum_{i=1}^{np} \sigma_i f(x_i) \right] \right]
\]

where \( F \) is a class of functions mapping the domain of the input samples into \( \mathbb{R} \), \( \mathbb{E}_{P(x)} \) is the expectation with respect to the probability distribution of the input data, and \( \mathbb{E}_\sigma \) is the expectation with respect to \( \sigma \), which are independent uniform random variables taking the values \{+1,-1\}.

One can regard \( R_{np}(F) \) as a measure of the ability of the class to which (7) belongs to classify the input samples if associated with a class at random: as the fitting ability of the function increases, so does its complexity.

Equation (8) can be used to devise the following bound on the generalization error of a classifier [5]; this bound holds with probability \((1-\delta)\):

\[
P(y \cdot f(x) \leq 0) = \hat{E}_{np} h(y \cdot f(x)) + 2L \cdot R_{np}(F) + \sqrt{\frac{\ln(\frac{2}{\delta})}{2np}}
\]

where \( \hat{E}_{np} \) is the error on the input data measured through a loss function \( h(\cdot) \) having the Lipschitz constant \( L \), that is:

\[
\hat{E}_{np} h(y \cdot f(x)) = \frac{1}{np} \sum_{i=1}^{np} h(y_i \cdot f(x_i))
\]

In the classical SVM formulation by Vapnik, the form of the \( h(\cdot) \) function is:

\[
h_{\psi}(y \cdot f(x)) = \begin{cases} 
0 & \text{if } y \cdot f(x) \geq 1 \\
1 - y \cdot f(x) & \text{if } y \cdot f(x) \leq 1
\end{cases}
\]

Bartlett and Mendelson [5] suggested the following better function to account for classification errors:

\[
h_{BM}(y \cdot f(x)) = \begin{cases} 
0 & \text{if } y \cdot f(x) \geq 1 \\
1 - y \cdot f(x) & \text{if } 0 \leq y \cdot f(x) \leq 1 \\
1 & \text{if } y \cdot f(x) \leq 0
\end{cases}
\]

which has \( L=1 \) and saturates to one for any misclassified pattern. Obviously, \( h_{\psi}(u) \geq h_{BM}(u) \) \( \forall u \). As an important consequence, the formulation of the loss function as per (12) inhibits the use of well-known linearly constrained Quadratic Programming algorithms.

III. A CASE STUDY ON THE EFFECTS OF OUTLIERS

We introduce a one-dimensional dataset for the purpose of illustrating the effects of the linear penalty \( \xi \) used in (2) to take into account possible classification errors.

The dataset is built as follows: the points belonging to one class, say the positive one, are concentrated in the origin; the negative-labeled ones are concentrated in \( x = \pm \lambda \) (Fig. 1). Supposing
the negative class to be composed of $N_1$ samples plus the outlier and the positive class to be composed of $N_2$ samples, the dataset can be described as follows (we keep the array notation for the sake of generality):

$$\begin{align*}
\mathbf{y}_n &= -1 \Rightarrow x_n = -1 \land x_n = +1, \quad n = 1, \ldots, N_1 + 1 \\
\mathbf{y}_p &= +1 \Rightarrow x_p = 0, \quad p = 1, \ldots, N_2
\end{align*}$$

(13)

Writing (2) for this specific setting and restricting our analysis to the monodimensional case of a linear kernel ($\Phi$ is the identity) gives:

$$\min_{w,b} \frac{w^2}{2} + C(N_1 \xi_1 + N_2 \xi_2 + \xi_3)$$

subject to:

$$\begin{align*}
w - b &= 1 + \mu_1 - \xi_1 \\
b &= 1 + \mu_2 - \xi_2 \\
-w \lambda - b &= 1 + \mu_3 - \xi_3
\end{align*}$$

(14)

It follows from the definition (3) that:

$$w = N_1 \alpha_1 - \lambda \alpha_3$$

(15)

The goal is now to find when the analog error caused by the outlier is large enough to draw the separation threshold beyond the positive class, thus causing its misclassification. The condition we are seeking can be stated as:

$$w = 0, \quad b = -1$$

(16)

which, if substituted into the constraints of (14), leads to:

$$\begin{align*}
-b &= 1 + \mu_1 - \xi_1 \\
b &= 1 + \mu_2 - \xi_2 \\
-b &= 1 + \mu_3 - \xi_3
\end{align*}$$

(17)

and, if substituted into (15), gives:

$$N_1 \alpha_1 = \alpha_3 \lambda$$

(18)

The positiveness of the constraints on $\mu_i$ and $\xi_i$ provides:

$$\begin{align*}
\mu_1 &= \xi_1 = \mu_3 = \xi_3 = 0 \\
\xi_2 &= 2
\end{align*}$$

(19)

and the KKT conditions require:

$$\alpha_2 = C = \frac{N_1 \alpha_1 + \alpha_3}{N_2}$$

(20)

where the last equality follows from (4).

Using relations (18) and (20), we rewrite the inequality constraints on $\alpha_i$ as:

$$\begin{align*}
\alpha_3 &= C = \frac{N_2}{1 + \lambda} \\
\alpha_1 &= C = \frac{\lambda \cdot N_2}{1 + \lambda} \quad N_1 \leq C
\end{align*}$$

(21)

thus the relations that allow the feasibility of the solution are:

$$\begin{align*}
N_2 &\leq 1 + \lambda \\
\lambda N_2 &\leq (1 + \lambda) N_1
\end{align*}$$

(22)

The present problem has been maliciously set in order to focus the reader’s attention on the difference between minimizing the (integer) number of errors and minimizing a bound on the number of errors. Indeed, under the conditions in (21), the classical SVM fails to find a reasonable solution. It is straightforward to prove that by using the loss function defined in (12) the optimal solution can be obtained with $w=2$ and $b=1$ when:

$$\begin{align*}
2 &\leq C(N_2 - 1) \\
2(1 - \lambda^2) &\leq C(N_1 - 1)
\end{align*}$$

(23)

The next section illustrates how to approach the np-hard problem of minimizing the number of errors by the computational paradigm of Quantum Computing.

IV. QUANTUM COMPUTING FOR SVM TRAINING

Research in Quantum Computing experienced an enormous growth in the last decades. Due to both the subject complexity and the huge amount of existing literature, the following, synthetic treatment of QC can only provide an informal hint at the current scenario.

A crucial concept that makes QC outperform classical computing paradigms is the possibility of superposed states [8]. In the quantum-based representation, a single digital quantity (“qbit”), $\psi$, is allowed to take on both state ‘0’ and ‘1’ at the same time. Each state is characterized by a complex number giving the probability amplitude of the state:
where \( \psi \) denotes the state direct product \([8]\); thus \( |\psi_0\rangle \) comprehends all possible states, which are equiprobable. In the second step, one feeds the cost-function algorithm with \( |\psi_0\rangle \), thus obtaining a superposition of all possible cost values.

By contrast, a classical computer would face an exponential computational overhead. The fact that a quantum computer can hold simultaneously and linearly the exponential number of states of a classical machine seems to hint at the fact that QC might tackle NP-problems by providing P-complex solutions.

Grover’s algorithm \([11]\) is one of the best-known QC techniques proposed so far: it tackles the (NP-complete) problem of searching an input string within an unsorted database. At start-up, Grover’s approach requires a single computation of the matching function on the superposition of all equiprobable input entries prepared as per (26). Then an iterative process makes the sought-for input entry emerge progressively from among other entries. The process uses a series of special transformations of the quantum-machine state that are repeated for a finite number of steps.

The repetitions involved in Grover’s algorithm proceed at the internal clock rate of the quantum machine, and cannot be compared with the conventional number of iterations of optimization procedures. For an input string including \( n \) bits and \( N=2^n \) possible states, the number of repetitions grows as \( O(\sqrt{N}) \).

Thus Grover’s method does not break the NP-completeness barrier, yet it has represented a popular basis for a large variety of algorithms. For the purposes of the research presented here, a quantum method for minimization is described in \([9]\). The number, \( R \), of repetitions to convergence for that algorithm is given by:

\[
R = 22.5 \cdot \sqrt{N} + 1.4 \cdot \lg^2 N \approx 22.5 \cdot \sqrt{N} \quad (27)
\]

Theory shows that a single run of the minimization algorithm \([9]\) finds out a valid solution with probability at least \( \frac{1}{2} \). Therefore, to increase the success probability one just applies the basic algorithm in a series of \( k \geq 1 \) different runs. With this approach, the total number of repetitions, i.e., the computational cost for the quantum machine, is

\[
R^{(k)} = k \cdot R \quad (28)
\]

and the associated probability of success becomes:

\[
P_q^{(k)} \geq 1 - \frac{1}{2^k} \quad (29)
\]

To sum up, to use Quantum Computing for SVM training first requires one to express the SVM model in a digital representation (including both the free parameters and the cost-function computation). The set of digital parameters to be optimized are stored in as many associated qbits, that are prepared in an initial, equiprobable superposition (26). Feeding the initial state to the cost-function supports an exhaustive scanning of the cost space. The resulting optimization problem is eventually solved by the minimization algorithm, whose quantum computational cost and success probability are given by (28) and (29), respectively.

Now one might want to analyze the specific advantages of the quantum approach. The comparison involves the quantum-minimization algorithm and a Montecarlo random-search process, which represents the ultimate resort for NP-complete problems in the lack of effective optimization techniques.

Assume that the minimization problem has \( M \) different solutions; in a search space with \( N \) possible configurations, the probability of success of a Montecarlo search after \( r \) test iterations is expressed as

\[
P_m^{(r)} = 1 - \left(1 - \frac{M}{N}\right)^r \quad (30)
\]

To compare the two optimization methods on a fair basis, one should try a Montecarlo search for \( R^{(k)} \) times; the quantum
approach exhibits an advantage whenever \(\frac{p_{m}^{(e_{i})}}{p_{q}^{(k)}}\), by using (29) and (30) one easily obtains:

\[
22.5 \cdot M < \sqrt{N} \ln 2 \tag{31}
\]

Interestingly, the expression (31) does not depend on the number, \(k\), of test runs of the quantum algorithm. Rather, the resulting condition exclusively depends on the specific problem complexity, involving the number of solutions and the search-space extension. In particular, one observes that condition (31) is most often fulfilled in common practice, as the measure of the number of optimized bits is usually much larger than that of problem minima.

As a consequence, the applicative interest in Quantum Computing paradigms also stems from the fact that their relative effectiveness increases with the difficulty of the specific problem at hand.

V. EXPERIMENTAL RESULTS

The different effects of using the loss functions (11) and (12) have been studied experimentally on the “Breast Cancer” testbed. This dataset includes 699 patterns with 9 features; each pattern is classified as benign or malignant. After excluding 16 samples featuring missing values, all features were normalized in the range [-1,1]. Previous results [10] showed that the information carried by the whole dataset is well represented by just two features, i.e., features 6 and 8. Therefore, the present analysis reduces the Breast Cancer dataset to a 2-D problem, which makes it possible to obtain useful information from a visual inspection of both the data and the classification surfaces.

A classical SVM (7) with a linear kernel and \(C=1000\) scored an empirical error rate of 5.124%; the separation surface is shown in Fig.2. This result was obtained by using the algorithm [3], and had to be compared with that obtained by using the loss function \(h_{BM}(u)\).

To that purpose, the analysis considered a digital SVM implementation, whose weights and bias were coded as 8-bit values. The related quantities varied in the range [-100,100] taking on \(2^{8}=256\) discrete values. The optimization process minimized, with respect to \(w\) and \(b\), the digital cost function, \(DCF\), defined as

\[
DCF = \frac{\|w\|^{2}}{2} + C \sum_{i=1}^{np} h_{BM}(y_{i}(w \cdot x_{i} + b)) \tag{32}
\]

Fig. 2 presents the class-separation boundary associated with the minimum of (32), giving an empirical misclassification error of 4.25%.

An histogram-based analysis of the results (Fig.3) shows that about 6% of the \(DCF\) values are less than twice the minimum of the \(DCF\) itself.

When implementing the digital optimization problem on a quantum machine, in the notation adopted in Section IV, one has \(N=16777216\), \(M=6\). The latter value was obtained by keeping valid solutions those which did not displace more than 1\% from the minimum of (32).

Therefore, the probability of finding a good solution is

\[
\frac{M}{N} = 3.6 \cdot 10^{-7} \tag{33}
\]

and the condition (31) is fulfilled.
VI. CONCLUSIONS

The power and effectiveness of SVMs in their original formulation as a general pattern-processing paradigm is not being questioned. However, computer based implementations that take into account the digital nature of represented quantities, and refined formulations that shrink generalization bounds can invalidate the applicability of efficient QP training algorithms.

Scanning a digital bit space without a gradient-based method can turn the optimization task into an NP-complete exhaustive-search problem. This ultimately shifts the interest toward novel and promising computational paradigms such as Quantum Computing.

The main pro for such an approach derives from the principle of quantum superposition of states, which enables an inherent parallelism in information processing that is not achievable by classical computing machinery. On the other hand, two cons seem to hold back an excessive enthusiasm for quantum approaches: first, quantum machinery is reportedly not a mature technology yet, hence one should not expect to have quantum optimization available for practical purposes in the near future. Secondly, no proof has been given so far that Quantum Computing can break the NP-completeness barrier in a real, interesting problem.

In view of these issues, a basic conclusion might anyway be drawn from the research presented in this paper: QC can yet prove effective for an important problem such as training SVMs for digital implementations. The reported analysis also showed that the computational benefits conveyed by quantum optimization increase when the problem complexity increases.

The presented simulations on a real-world problem open new vistas over the possibility of tuning SVM classifiers that are apt to direct and effective realizations in digital circuitry.

REFERENCES