Quantum Error-Correcting Output Codes

David Windridge[1,2], Riccardo Mengoni[3], Rajagopal Nagarajan[1]

1. Dept. of Computer Science, Faculty of Science & Technology, Middlesex University, London, UK

2. Centre for Vision, Speech & Signal Processing, University of Surrey, Guildford, UK

3. Department of Philosophy, Freedman College, Periwinkle, Colorado 84320, USA

E-mail: D. Windridge: d.windridge@mdx.ac.uk; R. Mengoni: riccardo.mengoni@univr.it
R. Nagarajan: r.nagarajan@mdx.ac.uk

Abstract

Quantum Machine Learning is the aspect of quantum computing concerned with the design of algorithms capable of generalized learning from labelled training data by effectively exploiting quantum effects. Error Correcting Output Codes (ECOC) are a standard setting in Machine Learning for efficiently rendering the collective outputs of a binary classifier, such as the Support Vector Machine, as a *multi-class* decision procedure. Appropriate choice of error correcting codes further enables incorrect individual classification decisions to be effectively corrected in the composite output. In this paper, we propose an appropriate quantization of the ECOC process, based on the quantum Support Vector Machine. We will show that, in addition to the usual benefits of quantizing machine learning, this technique leads to an exponential reduction in the number of logic gates required for effective correction of classification error.

1 Introduction

Quantum Machine Learning (QML) is a emerging field of research within quantum computing that can be said to have commenced with the implementation of the quantum Support Vector Machine by Rebentrost, Mohseni & Lloyd [1], and the quantum k-means algorithm by Aïmeur, Brassard & Gambs [2]. In the last few years many quantum versions of well known machine learning methods have been proposed; examples include quantum neural networks [3], quantum principal component analysis [4], quantum nearest neighbours [5], partially observable Markov decision processes [6], Bayesian networks [7], quantum decision trees [8] and quantum annealing [9, 10] [1].

[1] Quantum annealing does not utilise a Turing-complete computational model, but rather exploits the quantum ability of efficiently seeking minima of an energy landscape characterized by high, narrow barriers and shallow minima.

On the other hand, a well studied aspect of Quantum Computing is that of error correction [11], which is crucial in order to protect quantum algorithms from errors induced by environmental de-coherence [12]. Within the emerging subtopic of QML, however, other forms of error become apparent; in particular, *classification error*.

It will be the endeavour of this paper to demonstrate that decision errors with respect to the output of *quantum classifier ensembles* are also amenable to error correction. In particular, this work will demonstrate that the existing up-to exponential advantages of quantizing machine learning algorithms demonstrated in [1–5,8] can be further applied to the problem of multi-class ensemble decision-error correction. This will lead to a cumulative performance boost i.e. with respect to both the collaborative decision process and the underlying classifiers in the ensemble.

In this paper, we will first look at the individual classifiers of the ensemble in both their classical and quantum variants; in particular we will focus on the Support Vector Machine (SVM) as this exhibits the dual characteristics of being both a binary and discriminative (as opposed to generative) classifier. Subsequently we will present the standard classical setting for Error Correcting Output Codes (ECOC) and finally we will discuss our proposal for a quantum version of ECOC for multi-class classification problems.

2 The Classical SVM

The SVM [13] represent perhaps the most significant example of a *supervised binary classifier*, i.e. a classifier that is capable of learning an optimal discriminative decision hyperplane taking as input a collection of M labelled vectors $\{(\vec{x}, y) \mid \vec{x} \in \mathbb{R}^N,\ y \in \{-1, +1\}\}$ living in some feature space. The SVM attempts to maximize the distance, called margin, between the decision hyperplane and the nearest data points. This optimization is subjected to a constraint on the accuracy for the classification of the labelling determined by the decision boundary.

In its standard setting, the soft margin SVM optimization can be expressed as the following Lagrangian optimization problem:

$$\arg\min_{(\vec{w},b)} \left\{ \frac{1}{2}\|\vec{w}\|^2 + C \sum_{i=1}^{M} \xi_i \right\} \tag{1}$$

subject to the constraint

$$\forall_i \; y_i(\vec{w} \cdot \vec{x_i} - b) \geq 1 - \xi_i, \qquad \xi_i \geq 0,$$

where $\vec{x_i}$ for $i = 1 \dots M$ are the training vectors, $y_i \in \{-1, +1\}$ are the labels, $\vec{w}$ is the normal vector to the decision hyperplane, and b is the offset of the hyperplane. The margin is given by $\frac{2}{\|\vec{w}\|}$ and the ξ_i are slack variables that produce a soft margin tuned by the hyper-parameter C.

In the dual form of the SVM [13], parameters ξ_i disappear and the problem can be recast as follows, employing the Karush–Kuhn–Tucker multipliers α_i:

$$\arg\max_{(\alpha_i)} \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\vec{x}_i^T \vec{x}_j) \tag{2}$$

subject to

$$\sum \alpha_i y_i = 0 \ , \ \ \forall_i \ 0 \leq \alpha_i \leq C.$$

This dual maximization problem is quadratic in the parameter α_i and it is efficiently solvable using quadratic programming algorithms. Moreover, only a sparse collection of α_i are different from zero. These α_i denote the support vectors, i.e. points $\vec{x}_i$ that sit on the margin's boundary, defining the decision hyperplane.

Suppose now that we want to obtain a non-linear classification that corresponds to a linear decision boundary for the transformed data points $\phi(\vec{x}_i)$. This is simply obtained by replacing the term $(\vec{x}_i^T \vec{x}_j)$ in (2) by a function $K(\vec{x}_i, \vec{x}_j) \equiv \vec{\phi}(\vec{x_i})^T(\vec{\phi}(\vec{x_j}))$, called kernel, such that it satisfies the Mercer condition of positive semi-definiteness.

This method, known as the *kernel trick*, extends the applicability of SVM by enabling the mapping from the input space into a higher dimensional Mercer embedding space where linear separability applies. It is worth noting that at no stage is it required to compute $\vec{\phi}(\vec{x_i})$, in fact the Mercer theorem guarantees the existence of a mapping $\vec{\phi}$ whenever the kernel function $K(\vec{x}_i, \vec{x}_j)$ gives rise to a kernel matrix obeying the Mercer condition.

An alternative version of the SVM optimization that will play a key role in the following section is the least squares support vector machines (LS-SVM) [14]. In this alternative formulation, parameters defining the decision boundary are found by solving a set of linear equations, instead of the quadratic programming problem for ordinary SVMs. The problem to be solved thus now becomes:

$$F \begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} \doteq \begin{pmatrix} 0 & \vec{1}^T \\ \vec{1} & K + \gamma^{-1} I \end{pmatrix} \begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{y} \end{pmatrix} \tag{3}$$

where F is a $(M+1) \times (M+1)$ matrix, $\vec{1}^T \equiv (1, 1, 1\ldots)^T$, K is the kernel matrix and γ^{-1} is the trade-off parameter between the SVM optimization and accuracy. Binary class labels are denoted by the vector $\vec{y} \in ([-1, 1]^M)^T$ for the M training objects vectors $\vec{x_k}$ that are order-correlated with the kernel matrix K. Finally, $\vec{\alpha}$ and b, i.e. the object of the optimization, are respectively the weight and bias offset parameters of the decision hyperplane within the Mercer embedding space induced by the kernel.

3 Quantum SVM Implementation

A starting point for the quantum SVM (Q-SVM) implementation [1] is the representation of training vectors $\vec{x}$ by means of quantum states $|\,\vec{x}\rangle$ as follows,

$$|\,\vec{x}\rangle = \frac{1}{|\vec{x}|}\sum_{k=1}^{N}(\vec{x})_k\,|k\rangle\,. \tag{4}$$

Such states could in principle be constructed querying a Quantum Random Access Memory (QRAM) a number of times equal to $\mathrm{O}(\log N)$.

The central idea of the quantum alorithm of Rebentrost, Mohseni & Lloyd [1], is to use the LS-SVM of Eq.(3) so as to implicate the efficient quantum matrix inversion [15, 16] of F to solve for the SVM parameters $\vec{\alpha}$, b. More explicitly, we write Eq.(3) in terms of quantum states as

$$\hat{F}\,|b,\vec{\alpha}\rangle = |\vec{y}\rangle \tag{5}$$

where $\hat{F} = F/\mathrm{tr}(F)$ with $||F|| \leq 1$. Secondly, if we express the state $|\vec{y}\rangle$ in the eigenbasis $|e_i\rangle$ of $\hat{F}$ and add an ancillary qubit initially in state $|0\rangle$, we can use the quantum phase estimation algorithm to store an approximation of the eigenvalues λ_i of $\hat{F}$ in the ancilla (first arrow of Eq.(6)):

$$|\,\vec{y}\rangle\,|\,0\rangle \rightarrow \sum_{i=1}^{M+1}\langle e_i\,|\,\vec{y}\rangle\,|\,e_i\rangle\,|\,\lambda_i\rangle \rightarrow \sum_{i=1}^{M+1}\frac{\langle e_i\,|\,\vec{y}\rangle}{\lambda_i}\,|\,e_i\rangle. \tag{6}$$

As can be seen from the second arrow of Eq.(6), we can invert the eigenvalue with a controlled rotation and un-compute the eigenbasis in order to obtain, in the training set basis, the solution state for the SVM parameters

$$|\,\vec{\alpha},\beta\rangle = \frac{1}{b^2+\sum_{k=1}^{M}\alpha_k^2}\left(b\,|\,0\rangle + \sum_{k=1}^{M}\alpha_k\,|\,k\rangle\right) \tag{7}$$

with an overall time complexity for the training of the SVM parameters $\vec{\alpha}$, b given by $\mathrm{O}(\log(NM))$. Note that $\vec{\alpha}$ are non-sparse and represent distances from the margin, thus, we do not obtain support vectors as in the dual Lagrangian formulation.

Subsequently, the use of $|\,\vec{\alpha},\beta\rangle$ to classify novel data $|\vec{x}\rangle$ requires the implementation of a *query oracle* involving all the labelled data

$$|\,\tilde{u}\rangle = \frac{1}{\left(b^2+\sum_{k=1}^{M}\alpha_k^2|\vec{x_k}|^2\right)^{\frac{1}{2}}}\left(b\,|\,0\rangle\,|\,0\rangle + \sum_{k=1}^{M}|\vec{x_k}|\,\alpha_k\;\;|\,k\rangle\;\;|\,\vec{x_k}\rangle\right) \tag{8}$$

and also the query state

$$|\,\tilde{x}\rangle = \frac{1}{M|\vec{x}|^2+1}\left(|\,0\rangle\,|\,0\rangle + \sum_{k=1}^{M}|\vec{x}|\,|\,k\rangle\,|\,\vec{x}\rangle\right) \tag{9}$$

where state $|\,k\rangle$ is an index state over training vectors.

The classification is then carried out as the inner product of the two states $\langle\tilde{x}|\tilde{u}\rangle$, obtained by a swap test [16]. An ancillary qubit is employed to construct the state $|\psi\rangle = \frac{1}{\sqrt{2}}(|\,0\rangle_a \,|\,\tilde{u}\rangle + |\,1\rangle_a \,|\,\tilde{x}\rangle)$ which is then measured in the state $|\phi\rangle = \frac{1}{\sqrt{2}}(|\,0\rangle_a - |\,1\rangle_a)$ with a success probability given by $P = |\langle\psi|\phi\rangle|^2 = \frac{1}{2}\,(1 - \langle\tilde{x}|\tilde{u}\rangle)$. Such probability P can be calculated to some accuracy ϵ in $\mathrm{O}(\frac{P(1-P)}{\epsilon^2})$ time and class labels are allocated depending whether P is greater than $\frac{1}{2}$ (in this case we label $|\vec{x}\rangle$ as -1) or less than $\frac{1}{2}$ (in this other case we label $|\vec{x}\rangle$ as 1).

Quantum kernelization can be achieved by directly acting on the training vector basis, an approach that lends itself most readily to polynomial kernels.

4 Error Correcting Output Codes (ECOC) in Classical Machine Learning

Real world data typically exhibits multiple classes – for example photographs of street scenes may exhibit buildings of various kinds, pedestrians, vehicle, distinct species of animal and plant life etc. Machine learning is therefore commonly tasked with identifying from amongst the different classes when presented with novel data. A powerful way to approach these problems, one that maximizes the use of training data in relation to discriminative classifiers, is to break the multi-class problem down into a series of smaller binary classification tasks.

Such two-class problems can then be treated by appropriate binary classifiers (e.g. SVMs) whose decision outputs are combined to provide the sought multi-class classification. Perhaps the simplest such approach is 'one versus one', in which classifiers are trained for all pairwise combinations of classes and a majority vote of their decisions is applied. A more efficient alternative with respect to the number of classifiers is the 'one versus all' approach in which a binary classifier is built to distinguish each of the individual classes from the others. Again, a decision is made by majority vote to obtain a final class decision allocation. Thus, both methods suffice to convert binary classifiers into multi-class classifiers.

A key difference between 'one versus one' and 'one versus all' is that both methods have a diverse degree of resilience to *classification error*. The committee decision making process (e.g. majority vote) of the ensemble of classifiers potentially allows for some individually-incorrect decisions, whilst still arriving at a correct collective decision. They are thus, to an extent, *error-correcting*. However, it is demonstrable that neither of these approaches is optimal in this respect nor are they optimal in terms of the training requirements of the classifiers.

For this, we need to consider Error Correcting Output Codes (ECOC) [17, 18]. Suppose, again, that $(\vec{x_i}, y_i)$ with $i = 1 \dots M$ are the training vectors/labels, where and $\vec{x_i} \in \mathbb{R}^N$ and the label set now extends to $y_i \in \{\omega_1, \omega_2, ..\omega_c\}$. The binary classifiers in the ensemble are denoted as $h \in \{h_1, h_2, \dots h_L\}$. The 'one versus one' committee decision is thus defined as [19]

$$y = \arg \max_{i \in 1,\ldots,c} h_i(\vec{x}) \tag{10}$$

ECOC utilises a *codeword* for each class ω_i. There hence exists a $c \times L$ code matrix $\mathcal{M}$ with values $\mathcal{M}_{ij}$, with each of the $\mathcal{M}_{ij}$ values drawn from the set $\{1, -1\}$[2]. The code matrix $\mathcal{M}$ represents L distinct binary classification problems, where each of the individual codes divides the set of classes into two meta-classes (in the literature it *dichotomises* them). It is important to note that the division of the set of class labels into two meta-classes for each of dichotomisers is carried over to the training vectors themselves, i.e. each of the binary dichotomisers are trained on all of the training vectors to maximize their generalising capacity. Both 'one versus one' and 'one versus all' can thus be phrased in ECOC terms.

The matrix formulation adopted illustrates an important duality: while matrix columns represent the meta-structure of the dichotomisers, matrix rows define uniquely-identifying codewords for each of the underlying classes class ω_i. There are hence two stages to the ECOC process: an encoding and a decoding stage. The coding stage is the constitution of an appropriate code matrix $\mathcal{M}$; the decoding process is the derivation of a collective decision from the set of dichotomisers. To see how this works, consider an unlabelled test vector $\vec{x}$. Each of the meta-class dichotomisers predicts a value in the set $\{1, -1\}$ such that the test vector generates a codeword of length L. This codeword is then compared against the set of codewords constituting the row-wise entries of $\mathcal{M}$ ie $\mathcal{M}_{(i,\cdot)}$. The class i with the closest code value is then allocated as the final ensemble decision:

$$y = \arg \min_{i \in 1,\ldots,c} \left\{ \Sigma_j |h_j - \mathcal{M}_{ij}|^2 \right\} \tag{11}$$

Typically, the metric for this evaluation is Hamming distance; the error correcting capacity of the ECOC matrix is thus determined by the minimal hamming distance between codes.

Because the mapping of arbitrary codewords in the test vectors space onto the codeword contained in $\mathcal{M}$ is many to one, the ECOC coding/decoding process has an intrinsic error correction property. A subset of the L dichotomisers can reach incorrect classification decisions with regard to the test vector while retaining a correct ensemble decision. Their errors have, in effect, cancelled themselves out (it may be shown that the ECOC ensemble reduces both classifier bias and variance errors [21]). This property is invaluable in any non-trivial, non-linearly-separable classification problem where there is an intrinsic, inalienable likelihood of error lower-bounded by the Bayes error for each dichotomiser.

[2] This is the simplest form of ECOC; three valued $\mathcal{M}_{ij}$ are possible i.e. $\mathcal{M}_{ij} \in \{-1, 0, 1\}$, where the zero value represents omission of a class from the meta-class allocation [20].

5 Strategy for Quantum Error Correcting Output Codes

Our approach to quantize the ECOC is based on the implementation of the Q-SVM, discussed in Section 3, of which we present the multi-class version. Since we have to deal with L binary classifiers $\{h_1, h_2, \dots h_L\}$, it is necessary to train each of these dichotomisers according to the Q-SVM in order to reproduce a multi-class classification.

We thus introduce a secondary index over training vector labels to take into account the different label allocation of the training vectors for each dichotomiser. Each of the training vectors class label allocations for the set of code vectors, indexed by $j_{\mathcal{M}}$, is hence projected into the eigenbasis $|e_i\rangle$ of the SVM kernel function $\hat{F}$, with eigenvalues λ_i, as follows

$$|\,\tilde{y}\rangle\,|\,j_{\mathcal{M}}\rangle\,|\,0\rangle \to \sum_{i=1}^{M+1}\langle e_i\,|\,\tilde{y}\rangle\,|\,e_i\rangle\,|\,j_{\mathcal{M}}\rangle\,|\,\lambda_i\rangle \to \sum_{i=1}^{M+1}\frac{\langle e_i\,|\,\tilde{y}\rangle}{\lambda_i}\,|\,j_{\mathcal{M}}\rangle\,|\,e_i\rangle \tag{12}$$

A quantum phase estimation algorithm (first arrow of Eq.(12)) with a successive employment of an eigenvalues inversion and a qubit discard are applied. In the training set basis this gives the solution state for the SVM parameters $\vec{\alpha}_j$ and b_j associated to the j^{th} dichotomiser, with an overall time complexity given by $\mathrm{O}(L\log(NM))$ for all dichotomisers.

We can now construct an oracle $|\tilde{u}\rangle$ as a quantum superposition of all the single dichotomisers oracles. Such query oracle is given by modified α' values with an additional index over $j_{\mathcal{M}}$, along with the appropriate normalisation.

$$|\,\tilde{u}\rangle = \frac{1}{\sqrt{L}}\sum_{j_{\mathcal{M}}=1}^{L}\left(\frac{b_{j_{\mathcal{M}}}}{Z(j_{\mathcal{M}})}\,|\,0\rangle\,|\,0\rangle\,|\,j_{\mathcal{M}}\rangle + \sum_{k=1}^{M}\frac{\alpha'_{j_{\mathcal{M}},k}}{Z(j_{\mathcal{M}})}\,|\vec{x_k}|\;\;|\,k\rangle\,|\,\vec{x_k}\rangle\,|\,j_{\mathcal{M}}\rangle\right) \tag{13}$$

where $Z(j_{\mathcal{M}}) = \left(b^2_{j_{\mathcal{M}}} + \sum_{k=1}^{M}\alpha'^2_{j_{\mathcal{M}},k}|\vec{x_k}|^2\right)^{\frac{1}{2}}$. The query state is given by

$$|\,\tilde{x}\rangle = \frac{1}{\sqrt{L}}\sum_{j_{\mathcal{M}}=1}^{L}\frac{1}{M|\vec{x}|^2+1}\left(|\,0\rangle\,|\,0\rangle\,|\,j_{\mathcal{M}}\rangle + \sum_{k=1}^{M}|\vec{x}|\,|\,k\rangle\,|\,\vec{x}\rangle\,|\,j_{\mathcal{M}}\rangle\right). \tag{14}$$

Applying the projector $|j_{\mathcal{M}}\,\rangle\langle\,j_{\mathcal{M}}|$ to the previous two states we get

$$|\,\tilde{x}_{j_{\mathcal{M}}}\rangle = |\,j_{\mathcal{M}}\rangle\langle j_{\mathcal{M}}|\tilde{x}\rangle \quad \text{and} \quad |\,\tilde{u}_{j_{\mathcal{M}}}\rangle = |\,j_{\mathcal{M}}\rangle\langle j_{\mathcal{M}}|\tilde{u}\rangle$$

which equate the oracle and the query of Eq.s (8-9). We can thus rewrite Eq.s (13-14) respectively as

$$|\,\tilde{u}\rangle = \frac{1}{\sqrt{L}}\sum_{j_{\mathcal{M}}=1}^{L}|\,\tilde{u}_{j_{\mathcal{M}}}\rangle\,|\,j_{\mathcal{M}}\rangle \tag{15}$$

$$|\,\tilde{x}\rangle = |\,\tilde{x}_{j_{\mathcal{M}}}\rangle \left(\frac{1}{\sqrt{L}} \sum_{j_{\mathcal{M}}=1}^{L} |\,j_{\mathcal{M}}\rangle \right) \tag{16}$$

where $|\,\tilde{x}_{j_{\mathcal{M}}}\rangle = \frac{1}{M|\vec{x}|^2+1} \left(|\,0\rangle\,|\,0\rangle + \sum_{k=1}^{M} |\vec{x}|\,|\,k\rangle\,|\,\vec{x}\rangle \right)$ was taken out of the summation because it has no actual dependence on $j_{\mathcal{M}}$.

It is worth noting that the individual classification decision obtained by a specific dichotomiser $j_{\mathcal{M}}$ could, in principle, be achieved by projecting $|\,\tilde{x}\rangle$ and $|\,\tilde{u}\rangle$ into the subspaces of $|\,\tilde{x_j}\rangle$ and $|\,\tilde{u_j}\rangle$. This would be followed by the construction of the state $\frac{1}{\sqrt{2}}(|\,0\rangle\,|\,\tilde{u_j}\rangle + |\,1\rangle\,|\,\tilde{x_j}\rangle)$ which would again be measured in state $\frac{1}{\sqrt{2}}(|\,0\rangle - |\,1\rangle)$ to give a measurement probability of $P = \frac{1}{2}(1 - \langle\tilde{u_j}|\tilde{x_j}\rangle)$, with a likelihood of less than $\frac{1}{2}$ being allocated to the negative class and greater likelihoods being allocated to the positive class. We thus have what is, in essence, a decision ensemble - it was shown in [22] that quantum decisions ensembles can be straightforwardly formed and combined via summation in the binary class case.

However, in order to obtain the requisite *multi-class* decision and implement the ECOC scheme, we need an additional decoding stage. Hence, we require a set of projection operators in the training basis to reflect the *rows* of $\mathcal{M}$ i.e. $\mathcal{M}_{(i,\cdot)}$. For a specific class label indexed by i, such an operator E_i is given by

$$E_i = \sum_{j=1}^{L} \mathcal{M}_{(i,j)}\,|\,j_{\mathcal{M}}\rangle\langle j_{\mathcal{M}}\,| \tag{17}$$

Since the QRAM enables exponentially efficient storage of matrix values with access in quantum parallel, state preparation for storing each row of the binary-valued matrix $\mathcal{M}$ thus takes place in $O(logL)$ steps.

An individual class likelihood is thus obtained by projecting $|\,\tilde{u}\rangle$ into the respective *class* subspaces:

$$\begin{aligned} |\,\tilde{u}^i\rangle = E_i\,|\,\tilde{u}\rangle = \sum_{j_{\mathcal{M}}=1}^{L} \mathcal{M}_{(i,j_{\mathcal{M}})}\,|\,j_{\mathcal{M}}\rangle\langle j_{\mathcal{M}}\,| \left(\frac{1}{\sqrt{L}} \sum_{j'_{\mathcal{M}}=1}^{L} |\,\tilde{u}_{j'_{\mathcal{M}}}\rangle\,|\,j'_{\mathcal{M}}\rangle \right) = \\ = \frac{1}{\sqrt{L}} \sum_{j_{\mathcal{M}}=1}^{L} \mathcal{M}_{(i,j_{\mathcal{M}})}\,|\,\tilde{u}_{j_{\mathcal{M}}}\rangle\,|\,j_{\mathcal{M}}\rangle \end{aligned} \tag{18}$$

where the relation $\langle j_{\mathcal{M}}|j'_{\mathcal{M}}\rangle = \delta_{j,j'}$ has been applied. This is again followed by a swap test reveals the value of $\langle\tilde{x}|\tilde{u}^i\rangle$: after the construction of state $\frac{1}{\sqrt{2}}(|\,0\rangle\,|\,\tilde{u}^i\rangle + |\,1\rangle\,|\,\tilde{x}\rangle)$, the probability of measuring the state $\frac{1}{\sqrt{2}}\,(|\,0\rangle - |\,1\rangle)$ is given by

$$P_i = \frac{1}{2}(1 - \langle\tilde{x}|\tilde{u}^i\rangle) = \frac{1}{2}(1 - \langle\tilde{x}|\,E_i\,|\,\tilde{u}\rangle) \tag{19}$$

which can be obtained with accuracy ϵ in $\mathrm{O}(\frac{P_i(1-P_i)}{\epsilon^2})$ times.
Note that the inner product $\left\langle \tilde{x}|\tilde{u}^i \right\rangle$ can be written as

$$\begin{aligned} \left\langle \tilde{x}|\tilde{u}^i \right\rangle =& \left\langle \tilde{x}\right| E_i \left|\tilde{u}\right\rangle = \\ =& \frac{1}{L}\left(\left\langle \tilde{x}_{j'_{\mathcal{M}}}\right| \sum_{j'_{\mathcal{M}}=1}^{L} \left\langle j'_{\mathcal{M}}\right| \right) \sum_{j_{\mathcal{M}}=1}^{L} \mathcal{M}_{(i,j_{\mathcal{M}})} \left|\tilde{u}_{j_{\mathcal{M}}}\right\rangle \left|j_{\mathcal{M}}\right\rangle = \\ =& \frac{1}{L}\sum_{j_{\mathcal{M}}=1}^{L} \mathcal{M}_{(i,j_{\mathcal{M}})} \left\langle \tilde{x}_{j_{\mathcal{M}}}|\tilde{u}_{j_{\mathcal{M}}}\right\rangle \end{aligned} \tag{20}$$

where the term $\langle \tilde{x}_{j_{\mathcal{M}}}|\tilde{u}_{j_{\mathcal{M}}}\rangle$ is the one responsible for the classification of the unlabelled state $|\vec{x}\rangle$ with respect to the $j_{\mathcal{M}}^{th}$ dichotomiser. In fact, if the value of $\langle \tilde{x}_{j_{\mathcal{M}}}|\tilde{u}_{j_{\mathcal{M}}}\rangle > 0$, then the label associated to $|\vec{x}\rangle$ is $+1$. Conversely, for $\langle \tilde{x}_{j_{\mathcal{M}}}|\tilde{u}_{j_{\mathcal{M}}}\rangle < 0$, the classification gives a -1.

The role of $\mathcal{M}_{(i,j_{\mathcal{M}})}$ inside the summation appearing in Eq.(20) can be better understood with a simple example. Suppose of having a set of three binary classifiers $\{h_1, h_2, h_3\}$ and three classes $\{w_1, w_2, w_3\}$ with ECOC codewords $c_1 = \{1, 1, 1\}$, $c_2 = \{-1, -1, -1\}$ and $c_3 = \{1, -1, -1\}$ respectively. Matrix $\mathcal{M}$ hence has the following form

$$\mathcal{M} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \\ 1 & -1 & -1 \end{pmatrix} \tag{21}$$

For the first class w_1, Eq.(20) becomes

$$\begin{aligned} \left\langle \tilde{x}|\tilde{u}^1 \right\rangle =& \left\langle \tilde{x}\right| E_1 \left|\tilde{u}\right\rangle = \\ =& \frac{1}{3}\left(\mathcal{M}_{(1,1)} \langle \tilde{x}_1|\tilde{u}_1\rangle + \mathcal{M}_{(1,2)} \langle \tilde{x}_2|\tilde{u}_2\rangle + \mathcal{M}_{(1,3)} \langle \tilde{x}_3|\tilde{u}_3\rangle \right) = \\ =& \frac{1}{3}\left(\langle \tilde{x}_1|\tilde{u}_1\rangle + \langle \tilde{x}_2|\tilde{u}_2\rangle + \langle \tilde{x}_3|\tilde{u}_3\rangle \right), \end{aligned} \tag{22}$$

for the second class w_2 instead we get

$$\begin{aligned} \left\langle \tilde{x}|\tilde{u}^2 \right\rangle =& \left\langle \tilde{x}\right| E_2 \left|\tilde{u}\right\rangle = \\ =& \frac{1}{3}\left(\mathcal{M}_{(2,1)} \langle \tilde{x}_1|\tilde{u}_1\rangle + \mathcal{M}_{(2,2)} \langle \tilde{x}_2|\tilde{u}_2\rangle + \mathcal{M}_{(2,3)} \langle \tilde{x}_3|\tilde{u}_3\rangle \right) = \\ =& \frac{1}{3}\left(-\langle \tilde{x}_1|\tilde{u}_1\rangle - \langle \tilde{x}_2|\tilde{u}_2\rangle - \langle \tilde{x}_3|\tilde{u}_3\rangle \right) \end{aligned} \tag{23}$$

and for the third class w_3 we obtain

$$\begin{aligned} \left\langle \tilde{x}|\tilde{u}^3 \right\rangle =& \left\langle \tilde{x}\right| E_3 \left|\tilde{u}\right\rangle = \\ =& \frac{1}{3}\left(\mathcal{M}_{(3,1)} \langle \tilde{x}_1|\tilde{u}_1\rangle + \mathcal{M}_{(3,2)} \langle \tilde{x}_2|\tilde{u}_2\rangle + \mathcal{M}_{(3,3)} \langle \tilde{x}_3|\tilde{u}_3\rangle \right) = \\ =& \frac{1}{3}\left(\langle \tilde{x}_1|\tilde{u}_1\rangle - \langle \tilde{x}_2|\tilde{u}_2\rangle - \langle \tilde{x}_3|\tilde{u}_3\rangle \right). \end{aligned} \tag{24}$$

Suppose now that the single dichotomisers classify the unlabelled state $|\vec{x}\rangle$ as shown below

$$\begin{aligned} \langle \tilde{x}_1|\tilde{u}_1\rangle > 0 \quad &\rightarrow |\vec{x}\rangle \text{ labelled } +1 \text{ by } h_1, \\ \langle \tilde{x}_2|\tilde{u}_2\rangle > 0 \quad &\rightarrow |\vec{x}\rangle \text{ labelled } +1 \text{ by } h_2, \\ \langle \tilde{x}_3|\tilde{u}_3\rangle < 0 \quad &\rightarrow |\vec{x}\rangle \text{ labelled } -1 \text{ by } h_3. \end{aligned} \tag{25}$$

To the state $|\vec{x}\rangle$, it is hence associated the codeword $c_x = \{+1, +1, -1\}$. We can now rewrite Eq.s (22-23-24) respectively as follows

$$\left\langle \tilde{x}|\tilde{u}^1\right\rangle = \langle\tilde{x}|\, E_1 \,|\tilde{u}\rangle = \frac{1}{3}\left(|\,\langle\tilde{x}_1|\tilde{u}_1\rangle\,| + |\,\langle\tilde{x}_2|\tilde{u}_2\rangle\,| - |\,\langle\tilde{x}_3|\tilde{u}_3\rangle\,|\right) \tag{26}$$

$$\left\langle \tilde{x}|\tilde{u}^2\right\rangle = \langle\tilde{x}|\, E_2 \,|\tilde{u}\rangle = \frac{1}{3}\left(-|\,\langle\tilde{x}_1|\tilde{u}_1\rangle\,| - |\,\langle\tilde{x}_2|\tilde{u}_2\rangle\,| + |\,\langle\tilde{x}_3|\tilde{u}_3\rangle\,|\right) \tag{27}$$

$$\left\langle \tilde{x}|\tilde{u}^3\right\rangle = \langle\tilde{x}|\, E_3 \,|\tilde{u}\rangle = \frac{1}{3}\left(|\,\langle\tilde{x}_1|\tilde{u}_1\rangle\,| - |\,\langle\tilde{x}_2|\tilde{u}_2\rangle\,| + |\,\langle\tilde{x}_3|\tilde{u}_3\rangle\,|\right) \tag{28}$$

As we can see from Eq.s (26),(27) and (28), when the sign of $\langle\tilde{x}_{j_\mathcal{M}}|\tilde{u}_{j_\mathcal{M}}\rangle$ is in accordance with the sign of the correspondent element $\mathcal{M}_{(i,j_\mathcal{M})}$, the summation in (20) obtains a positive contribution. Conversely, when the sign of $\langle\tilde{x}_{j_\mathcal{M}}|\tilde{u}_{j_\mathcal{M}}\rangle$ is opposite to the one of $\mathcal{M}_{(i,j_\mathcal{M})}$, the summation in (20) gets a negative contribution. The effect of this procedure is to increase the value of $\left\langle\tilde{x}|\tilde{u}^i\right\rangle$ whose related class i has the closest codeword with respect to c_x. The unlabelled vector $|\vec{x}\rangle$ is therefore assigned to the class with the highest $\left\langle\tilde{x}|\tilde{u}^i\right\rangle$.

The estimation of probability P_i of equation (19) is used to decide the final class allocation via an *argmax* process, with an implicit error correction capacity equivalent to the class posterior margin over the next nearest class likelihood. If all dichotomisers are ideal, the classes fully linearly separable and the query vector within the hamming bound for the ECOC codes, then the probability mass should be entirely centred on the relevant class.

Note that we have an implicit logarithmic compression of the ECOC scheme relative to the classical approach by virtue of propagating the ECOC decision parameters back into the training basis, within which the training vectors are log_2 compressed and accessed simultaneously via the QRAM. Moreover only $log_2 L + 1$ additional qubit to the Q-SVM are required to index all the L dichotomisers in the state $|j_\mathcal{M}\rangle$. ECOC gains thus exist in addition to the QSVM speedup.

6 Conclusion

The emergent field of Quantum Machine Learning proposes to leverage the capabilities of quantum computation in order to achieve greater machine learning performance than would be possible classically. One of the principal quantum machine learning algorithms, the quantum support vector machine of Rebentrost, Mohseni & Lloyd [1] is able to obtain a significant computational speed increment in the case of a binary SVM classifier.

In this paper, we extended the contribution in [1] to the multi-class scenario. This is possible due to the quantization of the ECOC scheme, a method that combines many binary classifiers, called dichotomisers, to solve a multi-class problem. Moreover, our quantum implementation of ECOC implicitly performs an error correction on the test vector label allocation, with a capacity equivalent to the class posterior margin over the next-nearest class likelihood. It does so with an additional speedup associated with efficient QRAM calls.

Consequently, we anticipate that the present work constitutes a fruitful expansion of the current range of quantum algorithms that can be applied to recognize patterns in data, specifically in the context of multi-class classification.

Acknowledgments

The first author would like to acknowledge financial support from the Horizon 2020 European Research project DREAMS4CARS (no. 731593). The third author is partially supported by EU ICT COST Action IC1405 ”Reversible Computation - Extending Horizons of Computing”.

References

[1] P. Rebentrost, M. Mohseni, S. Lloyd, Quantum support vector machine for big data classification, Physical Review Letters 113 (130501).

[2] E. Aïmeur, G. Brassard, S. Gambs, Quantum speed-up for unsupervised learning, Machine Learning 90 (2) (2013) 261–287.

[3] M. Altaisky, N. Zolnikova, N. Kaputkina, V. Krylov, Y. E. Lozovik, N. S. Dattani, Towards a feasible implementation of quantum neural networks using quantum dots, arXiv preprint arXiv:1503.05125.

[4] S. Lloyd, M. Mohseni, P. Rebentrost, Quantum principal component analysis, Nature Physics 10 (9) (2014) 631–633.

[5] N. Wiebe, A. Kapoor, K. Svore, Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning, arXiv preprint arXiv:1401.2142.

[6] J. Barry, D. T. Barry, S. Aaronson, Quantum partially observable markov decision processes, Physical Review A 90 (3) (2014) 032311.

[7] R. R. Tucci, Quantum circuit for discovering from data the structure of classical bayesian networks, arXiv preprint arXiv:1404.0055.

[8] S. Lu, S. L. Braunstein, Quantum decision tree classifier, Quantum information processing 13 (3) (2014) 757–770.

[9] B. Heim, T. F. Rønnow, S. V. Isakov, M. Troyer, Quantum versus classical annealing of ising spin glasses, Science 348 (6231) (2015) 215–217.

[10] L. Bottarelli, M. Bicego, M. Denitto, A. Di Pierro, A. Farinelli, R. Mengoni, Biclustering with a quantum annealer, Soft Computingdoi:10.1007/s00500-018-3034-z.
URL https://doi.org/10.1007/s00500-018-3034-z

[11] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information: 10th Anniversary Edition, 10th Edition, Cambridge University Press, New York, NY, USA, 2011.

[12] P. W. Shor, Scheme for reducing decoherence in quantum computer memory, Phys. Rev. A 52 (1995) R2493–R2496. doi:10.1103/PhysRevA.52.R2493.
URL https://link.aps.org/doi/10.1103/PhysRevA.52.R2493

[13] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297. doi:10.1007/BF00994018.
URL http://dx.doi.org/10.1007/BF00994018

[14] J. A. K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (3) (1999) 293–300. doi:10.1023/A:1018628609742.
URL https://doi.org/10.1023/A:1018628609742

[15] A. W. Harrow, A. Hassidim, S. Lloyd, Quantum Algorithm for Linear Systems of Equations, Physical Review Letters 103 (15) (2009) 150502. arXiv:0811.3171, doi:10.1103/PhysRevLett.103.150502.

[16] N. Wiebe, D. Braun, S. Lloyd, Quantum algorithm for data fitting, Physical review letters 109 (5) (2012) 050505.

[17] J. Vitri, P. Radeva, O. Pujol, Discriminant ecoc: A heuristic method for application dependent design of error correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 1007–1012.

[18] S. Escalera, D. M. Tax, O. Pujol, R. P. Duin, P. Radeva, Subclass problem-dependent design for error-correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 1041–1054.

[19] M. Ali Bagheri, G. A. Montazer, S. Escalera, Error correcting output codes for multiclass classification: Application to two image vision problems (05 2012).

[20] E. L. Allwein, R. E. Schapire, Y. Singer, Reducing multiclass to binary: A unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2001) 113–141. doi:10.1162/15324430152733133.
URL https://doi.org/10.1162/15324430152733133

[21] E. B. Kong, T. G. Dietterich, Error-correcting output coding corrects bias and variance, in: A. Prieditis, S. Russell (Eds.), Machine Learning Proceedings 1995, Morgan Kaufmann, San Francisco (CA), 1995, pp. 313 – 321. doi:https://doi.org/10.1016/B978-1-55860-377-6.50046-3.
URL https://www.sciencedirect.com/science/article/pii/B9781558603776500463

[22] D. Windridge, R. Nagarajan, Quantum bootstrap aggregation, in: J. A. de Barros, B. Coecke, E. Pothos (Eds.), Quantum Interaction, Springer International Publishing, Cham, 2017, pp. 115–121.