Supervised Learning with Quantum Measurements

Fabio A. González
MindLab Research Group, Departamento de Ingeniería de Sistemas e Industrial
Universidad Nacional de Colombia, Bogotá, Colombia

Vladimir Vargas-Calderón and Herbert Vinck-Posada
Grupo de Superconductividad y Nanotecnología, Departamento de Física
Universidad Nacional de Colombia, Bogotá, Colombia

(Dated: April 6, 2020)

This letter reports a novel method for supervised machine learning based on the mathematical formalism that supports quantum mechanics. The method uses projective quantum measurement as a way of building a prediction function. Specifically, the correlation between input and output variables is represented as the state of a bipartite quantum system. The state is estimated from training samples through an averaging process that produces a density matrix. Prediction of the label for a new sample is made by performing a projective measurement on the bipartite system with an operator, prepared from the new input sample, and applying a partial trace to obtain the state of the subsystem representing the outputs. The method can be seen as a generalization of Bayesian inference classification and as a type of kernel-based learning method. One remarkable characteristic of the method is that it does not require learning any parameters through optimization. We illustrate the method with different 2-D classification benchmark problems and different quantum information encodings.

I. INTRODUCTION

In recent years, the interest in the combination of quantum information processing and machine learning has been growing fueled by the increasing popularity and advances in both fields [1]. The field product of the intersection of these research areas is commonly denoted as quantum machine learning [2]. The new field has produced a considerable amount of research work that explores different interactions between the two areas [3].

The different approaches to quantum machine learning can be broadly classified into four categories depending on whether a classic or quantum system generates the data and whether the processing device is a classic computer or a quantum computer [4]. In the category of classical-data/quantum-processing, a large amount of work has been devoted to the development of quantum versions of different classical machine learning algorithms with an emphasis on showing an advantage, at least theoretically, of the quantum version in terms of speedup [3].

The classical-data/classical-processing category refers to the use of tools of quantum information research to formulate machine learning methods that take advantage of the quantum conceptual machinery. This category has been less explored than the former one. The latter category is the primary motivation of the work discussed in this letter.

This letter presents a method for supervised machine learning based on the mathematical formalism that supports quantum mechanics. The method can be implemented both as an algorithm for a classical computer and as a hybrid classical/quantum algorithm. The main idea of the method is to represent the joint probability of input and output variables, \( P(x, y) \), as the state of a bipartite quantum system. Training corresponds to calculating this state from training samples. Prediction corresponds to performing a projective measurement with an operator, prepared from the new input sample to be classified, and subsequently calculating a partial trace to obtain the state of the output subsystem.

The representation of \( P(x, y) \) as the state of a quantum system, more specifically as a density matrix, generalizes the classical probabilistic representation and enriches it with the additional representation capabilities of the quantum formalism. We show that the method generalizes Bayesian inference and can also be seen as a type of kernel learning method. Another remarkable characteristic of the proposed framework is that learning does not require any kind of optimization to find parameters. Learning corresponds to averaging quantum states representing training samples.

Different works have addressed the implementation of supervised learning models based on formalism from quantum mechanics or quantum information processing. In [5] the authors present a quantum algorithm for supervised cluster assignment based on calculating the minimum distance from a sample to the centroids representing clusters. Analogous quantum algorithms based on nearest-neighbor classification have been explored in [6–8] among others. Different quantum versions of classical machine learning algorithms have been studied: support vector machines [9–11], decision trees [12], classifier ensembles [13], neural networks [14–17], among others. Another line of work is the application of methods traditionally used for modeling quantum systems to supervised machine learning. Tensor networks, a tool for...
efficient modeling and simulation of many-body quantum systems, are the most representative of these methods [18, 19] and have been applied to different classification problems including image classification [20], language analysis [21] and probabilistic modeling [22]. With the exception of nearest-neighbor-based methods, all the other methods rely on optimization to learn the parameters of the model. The method presented in this letter does not make use of optimization since averaging quantum states accomplish learning. The method can be seen as a form of kernel-based learning, but in contrast with typical kernel methods and nearest-neighbor learning, there is no need for storing any individual training sample to be used during prediction.

II. CONCEPTS FROM MACHINE LEARNING

The problem of supervised learning is to induce a prediction function, \( f : \mathcal{X} \rightarrow \mathcal{Y} \), that maps an input space \( \mathcal{X} \) to an output space \( \mathcal{Y} \) from a set of training samples \( T = \{(x_i, y_i)\} \) with \( x_i \in \mathcal{X} \) and \( y_i \in \mathcal{Y} \) sampled from a, usually unknown, joint probability distribution \( P(x, y) \) [23]. This problem can be addressed using different approaches. The Bayesian inference approach assumes that the samples \( (x_i, y_i) \) are random variables and the problem of learning corresponds to estimating the conditional distribution \( P(y|x) \) from the samples. If \( \mathcal{Y} \) corresponds to a finite set, the prediction function can be defined as:

\[
 f(x) = \arg \max_{y_i} P(y = y_i|x). 
\]  

(1)

The conditional probability \( P(y|x) \) can be estimated using different strategies. These strategies can be broadly divided into two classes: generative modeling and discriminative modeling [24]. Generative models estimate the joint probability of inputs and outputs \( P(x, y) \), and from this, the conditional probability is calculated. Discriminative models directly learn the conditional probability \( P(y|x) \) from samples, usually assuming a particular functional form and estimating its parameters through optimization.

Another approach to supervised learning are kernel methods. In this approach input samples are implicitly mapped to a feature space \( \mathcal{F} \) using a kernel function \( k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \) (or \( k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{C} \) if \( \mathcal{F} \) is Hilbert space over \( \mathbb{C} \)) that calculates the dot product on the feature space:

\[
 k(x, x') = \langle \Phi(x), \Phi(x') \rangle_{\mathcal{F}}, 
\]  

(2)

where \( \Phi : \mathcal{X} \rightarrow \mathcal{F} \) is the mapping function implicitly defined by the kernel. Usually the prediction function corresponds to a linear function on the feature space that can be represented in terms of the kernel as:

\[
 f'(x) = \sum_{(x, y) \in S} \alpha_i k(x, x_i) y_i, 
\]  

(3)

where the \( \alpha_i \) are parameters learned from the training data and \( S \subseteq T \) is a subset of training samples. If \( y_i \in \{0, 1\} \), for instance, the prediction function may be defined as:

\[
 f(x) = \begin{cases} 
 1 & \text{if } f'(x) \geq 0, \\
 0 & \text{else}. 
\end{cases} 
\]  

(4)

III. QUANTUM MEASUREMENT CLASSIFICATION (QMC)

The proposed method is similar in principle to a generative Bayesian inference approach. During training, the joint probability of inputs and outputs is estimated from training samples and represented as a density matrix, \( \rho_{\text{train}} \), that corresponds to a quantum state of a bipartite system \( S_{XY} = S_X + S_Y \). \( S_X \) is the subsystem representing the inputs with associated Hilbert space \( \mathcal{H}_X \) and the subsystem \( S_Y \) represents the outputs in the Hilbert space \( \mathcal{H}_Y \). Consequently, the representation space of the system \( S_{XY} \) is \( \mathcal{H}_X \otimes \mathcal{H}_Y \). Prediction is made by performing a quantum measure over \( S_{XY} \) with an operator specifically prepared from a new input sample.

1. Quantum feature mapping. In this step each training sample is mapped to \( \mathcal{H}_X \otimes \mathcal{H}_Y \) using the following function:

\[
 \psi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{H}_X \otimes \mathcal{H}_Y \\
 (x, y) \mapsto |\psi_X(x)\rangle \otimes |\psi_Y(y)\rangle, 
\]  

(5)

where \( \psi_X : \mathcal{X} \rightarrow \mathcal{H}_X \) and \( \psi_Y : \mathcal{Y} \rightarrow \mathcal{H}_Y \) are functions that map inputs and outputs, respectively, to the corresponding quantum Hilbert spaces. As a short-hand notation, every data sample \( (x_i, y_i) \in T \) is mapped to the quantum feature space as \( \psi : (x_i, y_i) \mapsto |\psi_i\rangle := |\psi_X(x_i)\rangle \otimes |\psi_Y(y_i)\rangle \).

2. Training state estimation. In this step, we calculate a quantum state that summarizes the training
data set. This state is represented by a density matrix $\rho_{\text{train}}$. There are three alternatives to calculate $\rho_{\text{train}}$:

- **Pure state.** In this case the training state corresponds to a superposition of the states representing training samples. First we calculate the superposition state $|\psi_{\text{train}}\rangle$ as:

$$|\psi_{\text{train}}\rangle = \frac{1}{\|\sum_{i=1}^{n} |\psi_i\rangle\|} \sum_{i=1}^{n} |\psi_i\rangle,$$

and define

$$\rho_{\text{train}} = |\psi_{\text{train}}\rangle \langle \psi_{\text{train}}|.$$  \hspace{1cm} (6)

- **Mixed state.** Here, $\rho_{\text{train}}$ corresponds to a mixture of the states corresponding to the training samples:

$$\rho_{\text{train}} = \frac{1}{n} \sum_{i=1}^{n} |\psi_i\rangle \langle \psi_i|.$$  \hspace{1cm} (7)

- **Classical mixture.** In this case we extract the probabilities associated with the quantum state $|\psi_i\rangle$ and use them to build a quantum state, represented by a density matrix, that only has classical uncertainty:

$$\rho_{\text{train}} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} (|\psi_i\rangle \langle j|)^2 |j\rangle \langle j|,$$  \hspace{1cm} (8)

where $m$ is the dimension of $H_X \otimes H_Y$ and $|j\rangle$ are the elements of the canonical basis.

The three alternatives to calculate the training state in step 2 correspond to three different ways of combining quantum and classical uncertainty in a quantum state [24]. The pure state (eq. (6)) encodes the training data set using only quantum uncertainty, the classical mixture (eq. (7)) encodes the training samples using classical probabilities exclusively, and the mixed state (eq. (8)) uses a combination of quantum and classical uncertainty to encode the training samples in the training quantum state.

The prediction process is depicted in fig. 2. The process receives as input a new sample $x^*$ to be classified, and the training state $\rho_{\text{train}}$ from the training phase. The steps of the prediction process are described next.

1. **Quantum feature mapping.** $x^*$ is mapped to $|\psi_X(x^*)\rangle$.

2. **Prediction operator.** We define an operator that acts on $H_X \otimes H_Y$:

$$\pi(x^*) = |\psi_X(x^*)\rangle \langle \psi_X(x^*)| \otimes \text{Id}_{H_Y},$$  \hspace{1cm} (9)

where $\text{Id}_{H_Y}$ is the identity operator on $H_Y$.

3. **Quantum measurement.** The operator $\pi(x^*)$ is applied to $\rho_{\text{train}}$:

$$\rho' = \frac{\pi(x^*) \rho_{\text{train}} \pi(x^*)}{\text{Tr}[\pi(x^*) \rho_{\text{train}} \pi(x^*)]},$$  \hspace{1cm} (10)

4. **Partial trace.** To calculate the reduced state of $\rho'$ on subsystem $S_Y$ we calculate the partial trace with respect to subsystem $S_X$:

$$\rho_Y' = \text{Tr}_X[\rho']$$  \hspace{1cm} (11)

The density matrix $\rho_Y'$ contains information about the state of the subsystem $S_Y$ after the state of the subsystem $S_X$ is projected onto the state $|\psi_X(x^*)\rangle$. This density matrix gives information about the probability of predictions. For instance if $Y = \{y_k\}_{k=1,..,m}$ and $\psi_Y$ corresponds to a one-hot or a probability encoding (see Subsection 4VB), the diagonal element $\rho'_{y_i,i}$ can be interpreted as the probability of the value $y_i$, i.e., the probability that $x^*$ is labeled as $y_i$.

QMC not only resembles generative Bayesian inference, but it also generalizes it. The following proposition formally states this claim.

**Proposition 1.** Let $T = \{(x_i,y_i)\}_{i=1,...,n}$ be a set of training samples, $x^*$ a sample to classify, with $x_i, x^* \in \{1,..,m\}$ and $y_i \in \{1,2\}$. Let $\rho_{\text{train}}$ be the state calculated using the mixed state, eq. (8) or equivalently the classical mixture eq. (9), and a one-hot encoding feature map for both $x_i$ and $y_i$. Then the diagonal elements of the density matrix $\rho_Y'$ calculated using eq. (12) correspond to an estimation, using Bayesian inference, of the conditional probabilities $P(y = i|x^*)$:

$$\rho'_{y_i,i} = P(y = i|x^*) = \frac{P(x^*|y = i)P(y = i)}{P(x^*)}.$$

where $P(x^*|y = i)$, $P(y = i)$ and $P(x^*)$ are estimated from $T$. 

![Figure 2. Prediction](image-url)
In other words, using a classic mixture (eq. 3) to estimate the training state is equivalent to make classical Bayesian inference. This is not surprising since the classical mixture corresponds to a conventional probabilistic encoding of the information in the training data set.

When using the more general mixed state (eq. 5) to estimate the training quantum state, the prediction process involves more complex interactions between states. The following proposition shows that, in this case, the resulting density matrix $\rho'_Y$ for the subsystem $S_Y$ corresponds to a linear combination of the density matrices representing the output variables of the training samples.

**Proposition 2.** Let $T = \{(x_i, y_i)\}$ be a set of training samples, $x^*$ a sample to classify, with $x_i, x^* \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. Let $\rho_{\text{train}}$ be the state calculated using a mixed state (eq. 5) and quantum feature maps $\psi_X$ and $\psi_Y$. Then the density matrix $\rho'_Y$, calculated with eq. 12, can be expressed as:

$$\rho'_Y = \mathcal{M} \sum_{i=1}^{N} |k(x^*, x_i)|^2 |\psi_Y(y_i)\rangle \langle \psi_Y(y_i)|,$$

where $k(x^*, x_i) = \langle \psi_X(x^*)|\psi_X(x_i)\rangle$ and $\mathcal{M}^{-1} = \text{Tr} [\pi(x^*)\rho_{\text{train}}(x^*)]$.

Equation (14) can be seen as an instance of eq. (3) where $\alpha_i := k(x^*, x_i)$ and $y_i$ is replaced by $|\psi_Y(y_i)\rangle \langle \psi_Y(y_i)|$. Notice that $k(x^*, x_i)$ corresponds to the dot product in the quantum Hilbert space $H_X$, so it is in fact a kernel function. This means that QMC can be seen as type of kernel-based learning method. However an important difference is that while conventional kernel methods require to learn, usually through optimization, the $\alpha_i$ parameters, in QMC there are not parameters to be learned.

It is worth emphasizing that QMC can, in principle, be implemented in quantum devices through the preparation of a pure training state of the form eq. 3 with well-known preparation protocols [20]. With the same protocol, the state of the new data sample $x^*$ can be built. Finally, the projective measurement can be achieved via a third ancillary state that allows a SWAP test [27,29].

### IV. QUANTUM FEATURE MAPS

In quantum machine learning literature, there are several approaches to represent data as quantum states. Schuld and Petruccione [3] propose different strategies such as basis encoding, encoding data directly as qubits, and amplitude encoding, which is encoding data in the amplitude of quantum states. Next, we describe several approaches that we use in the illustrative examples.

#### A. Softmax encoding

A common approach to represent real numbers in the interval $[0, 1]$ is to use the mapping $\phi : x \mapsto \sin (2\pi x) |0\rangle + \cos (2\pi x) |1\rangle$, encoding the number as a the superposed state of a qubit. The softmax quantum encoding extends this approach from two dimensions to multiple dimensions.

First we define a probability encoding for real values $P : [0, 1] \rightarrow \mathbb{R}^m$ where $m$ is the number of states:

$$P_i(x) = \left( \frac{\exp \{-\beta \|x - \alpha_i\|^2\}}{\sum_{i=1}^{m} \exp \{-\beta \|x - \alpha_i\|^2\}} \right) \quad ,$$

where $\alpha_i = \frac{i-1}{m-1}$. Using these probabilities we build a quantum state representing a real number

$$|\varphi(x)\rangle = \sum_{j=1}^{m} \sqrt{P_i(x)} |j\rangle .$$

The quantum state corresponding to a vector $x_i \in \mathbb{R}^n$ is defined as

$$|\psi_X(x_i)\rangle = |\varphi(x_{i,1})\rangle \otimes \cdots \otimes |\varphi(x_{i,n})\rangle .$$

#### B. One-hot encoding

This representation corresponds to a basis encoding for discrete variables with $m$ possible values, $\mathcal{X} = \{1, \ldots, m\}$. The encoding for $x = j$ is given by

$$\psi_X(j) = |j\rangle .$$

#### C. Squeezed states

Recently, Schuld and Killoran [30] proposed to encode data to the phase of a light squeezed state

$$|\langle r, \varphi \rangle\rangle = \frac{1}{\sqrt{\cosh(r)}} \sum_{n=0}^{\infty} \sqrt{(2n)!} (e^{i(\varphi+\pi)} \tanh(r))^n |2n\rangle ,$$

so that a vector $x_i \in [0, \pi]^n$ is mapped to

$$\psi_X(c, x_i) = |(c, x_{i,1})\rangle \otimes \cdots \otimes |(c, x_{i,n})\rangle .$$

#### D. Coherent states

Data can also be encoded into the average number of photons of a canonical coherent state [31]:

$$|\langle \alpha, \gamma \rangle\rangle = e^{-\frac{|\alpha|^2}{2}} \sum_{n=0}^{\infty} \frac{\alpha^n \gamma^{n/2}}{\sqrt{n!}} |n\rangle$$
where a scaling characterized by $\gamma$ has been introduced so that the dot product of the two states corresponds to a Gaussian kernel with $\gamma$ parameter. The mapping from a real data sample $x_j \in \mathbb{R}^n$ to the complex $\alpha$ is done as follows. An auxiliary variable $\theta_j \in [0, \pi]$, so that $x_j, b \mapsto x_j, e^{ib}\theta_j$. Therefore, a data point $x_j$ is mapped to the quantum feature space through

$$
\psi_X(x_j, \gamma) = \left| (x_j, t e^{i\theta_j}, \gamma) \right| \otimes \ldots \otimes \left| (x_j, n e^{i\theta_{j,n}}, \gamma) \right|
$$

(22)

which induces a kernel

$$
|k_\gamma(x_k, x_j)|^2 = |\langle \psi_X(c, x_k) | \psi_X(c, x_j) \rangle|^2
$$

(23)

$$
= \prod_{\ell=1}^n \exp(-\gamma |x_{k,\ell} e^{i\theta_{k,\ell}} - x_{j,\ell} e^{i\theta_{j,\ell}}|^2),
$$

(24)

where the argument of the exponential is explicitly $-\gamma (x_{k,\ell}^2 + x_{j,\ell}^2 - 2x_{k,\ell} x_{j,\ell} \cos(\theta_{k,\ell} - \theta_{j,\ell}))$, which imposes a higher distance penalty in the feature space for distant data points in the original space $\mathbb{R}^n$ than the usual Gaussian kernel.

### E. Random Fourier Features

As in quantum state representations, the feature space of kernel methods is a Hilbert space. This means that a quantum feature map implicitly defines a kernel. A natural question is whether the opposite conversion also works, i.e., given a particular kernel function, can we find a quantum feature map such that the inner product of the corresponding Hilbert space corresponds to the kernel. In general, the answer is no; however, it is possible to find an approximation for certain kernels. Random Fourier features (RFF) [32] provides a technique that finds an explicit Hilbert space such that the inner product in this space approximates a shift-invariant kernel. Specifically, for a given kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, RFF finds a map $z : \mathbb{R}^d \to \mathbb{R}^D$ such that $k(x, y) \approx \langle z(x) | z(y) \rangle$.

The quantum state corresponding to a vector $x_i \in \mathbb{R}^d$ is defined as

$$
|\psi_X(x_i)\rangle = \frac{1}{\|z(x_i)\|} \sum_{j=1}^D z_j(x_i) |j\rangle.
$$

(25)

### V. METHOD ILLUSTRATION

In this section, we illustrate the performance of QMC with binary classification toy problems for the different aforementioned feature maps. Figure 3 compares the decision boundary obtained from a mixed training state through the different feature maps, where the states are truncated up to the first 20 Fock states for each input feature. The color tells the probability that the output state belongs to the red or blue classes. For all the cases, the method achieves good discrimination in both classes.

Equations 7 to 9 correspond to three different alternatives to estimate the training state. The mixed and pure alternatives are expected to take advantage of the quantum correlations induced by the feature mapping and exploited in the projective measurement process. Figure 4 shows that this is, in fact, the case for the 2D spirals dataset. The three plots show prediction
regions for the three different estimation strategies using
the coherent state quantum feature mapping truncated
to the first 32 Fock states. The mixed state representa-
tion has the best performance, followed closely by the
pure state representation. Both are able to capture the
particular shape of both classes. The classical state rep-
resentation fails to do good discrimination. This is better
observed on the top-left plot. Where the classification pre-
cision is measured for a range of \( \gamma \) values, showing that
the worst classification scheme is the classical representa-
tion state, whereas the best one is the mixed representa-
tion state, closely followed by the pure representation
state. The same behavior is observed when the coer-
herent state is truncated to the first 20–64 Fock states (not
shown).

Regarding the squeezed state encoding, decision
boundaries for a circles dataset for pure and mixed train-
ing states are shown in fig. 5. Here, the mixed training
state outperforms the pure training state. The classical
training state is useless because the data is mapped to
the phase of the squeezed state, and the probabilities in
eq. \( \text{(9)} \) do not depend on this phase. Again, a signature
of the kernel induced by the squeezed state is seen in the
regions at the middle top, bottom, right, and left parts of the
decision heatmaps that are wrongly classified. These
regions emerge from the fact that the similarity induced
by the squeezed-based kernel between two points at a
fixed Euclidean distance is maximum if the horizontal or
vertical components of the two points are the same.

![Figure 5. Decision heatmap for a two-circles dataset of the
squeezed state feature map with pure and mixed training
states. Squeezed states were truncated to the first 10 con-
tributing Fock states, and a value of \( r = 2.5 \) was used, as in
\( \text{[30]} \). The white boxes are as in fig. \( \text{[3]} \).](image)

VI. CONCLUSIONS

This letter presented a method for supervised learn-
ing based on quantum measurement. The overall strat-
agy of the method is based on two mechanisms: first, to
represent the joint probability of inputs and outputs by
the state of a bipartite quantum system and, second, to
predict the outputs of new input samples performing a
quantum measurement.

Using this quantum measurement framework as a ba-
sis for function induction contributes a two-fold novel
perspective to supervised machine learning. On the one
hand, the training process does not require optimization
of parameters, since training corresponds to state aver-
ging. This is an essential departure from current ma-
chine learning models, both classical and quantum-based.
On the other hand, the classification model induced by
QMC can be understood as a generalization of Bayesian,-
inference classification and as a type of kernel classifi-
cation model. This connection is a consequence of the
harmonious combination of linear algebra and probabil-
ity provided by the quantum framework. Some works
connect kernel and probabilistic methods \( \text{[33–34]} \); how-
ever the quantum measurement framework constitutes a
novel unifying perspective.

The ability of QMC of inducing a classification model
without parameter optimization suggests the possibility
of an efficient classical implementation. This is the case
for the training process, whose time complexity is lin-
eral on the number of training samples. However, the
computational burden moves from the training process
to the prediction process and from time complexity to
space complexity. In particular the space required by
the training density matrix, \( \rho_{\text{train}} \), is \( O(m^2 \ell^2) \) where
\( m = |H_X| \) and \( \ell = |H_Y| \). Scaling QMC to large scale
learning problems requires dealing with this space com-
plexity. A promising research line to address this prob-
lem is to use tensor networks \( \text{[18]} \) to build a compact
representation of these density matrices employing ten-
sor factorizations. Another possibility of mitigating the
computational costs is the implementation of QMC as a
hybrid classical-quantum algorithm. Whether this could
exploit a quantum advantage is part of our future re-
search.

Appendix A: Proof of propositions 1 and 2

**Proposition 1.** Let \( T = \{(x_i, y_i)\}_{i=1,...,n} \) be a set of
training samples, \( x^* \) a sample to classify, with \( x_i, x^* \in \{1,...,n\} \) and \( y_i \in \{1,2\} \). Let \( \rho_{\text{train}} \) be the state cal-
culated using the mixed state, eq. \( \text{(8)} \) or equivalently the
classic mixture eq. \( \text{(9)} \), and a one-hot encoding feature
map for both \( x_i \) and \( y_i \). Then the diagonal elements of the
density matrix \( \rho_{y_i} \) calculated using eq. \( \text{(12)} \) correspond to
an estimation, using Bayes inference, of the conditional
probabilities \( P(y = i|x^*) \):

\[
\rho_{y_i} = P(y = i|x^*) = \frac{P(x^*|y = i)P(y = i)}{P(x^*)}, \quad \text{(A1)}
\]

where \( P(x^*|y = i) \), \( P(y = i) \) and \( P(x^*) \) are estimated
from \( T \).

**Proof.** Since both \( x_i \) and \( y_i \) are represented using a one-
hot encoding representation, then

\[
|\psi_i\rangle = |x_i\rangle \otimes |y_i\rangle = |x_i y_i\rangle. \quad \text{(A2)}
\]
Applying eq. (8):
\[
\rho_{\text{train}} = \frac{1}{n} \sum_{i=1}^{n} |x_i y_i \rangle \langle x_i y_i |
\]
(A3)
\[
= \sum_{j=1}^{m} \sum_{k=1}^{2} P(x = j, y = k) |j k \rangle \langle j k |,
\]
\[
\text{with } P(x = j, y = k) = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i,j} \delta_{y_i,k}. \]
Applying the prediction operator (eq. (10)) \(y\) can be expressed as:
\[
\psi = \pi(x^*) \langle x^* | \otimes \text{Id}_{H_y}
\]
(A4)
to eq. (A3) produces
\[
\rho' = \frac{\sum_{k=1}^{2} P(x = x^*, y = k) |x^* k \rangle \langle x^* k |}{\sum_{k=1}^{2} P(x = x^*, y = k)}
\]
\[
= \sum_{k=1}^{2} P(y = k|x = x^*) |x^* k \rangle \langle x^* k |.
\]
(A5)
Finally, we calculate the partial trace of eq. (A5) to obtain:
\[
\rho'_y = \text{Tr}_X[\rho']
\]
\[
= \sum_{k=1}^{2} P(y = k|x = x^*) |k \rangle \langle k |.
\]
(A6)
\[
\text{Proposition 2. Let } T = \{(x_i, y_i)\}_{i=1,\ldots,n} \text{ be a set of training samples, } x^* \text{ a sample to classify, with } x_i, x^* \in \mathcal{X} \text{ and } y_i \in \mathcal{Y}. \text{ Let } \rho_{\text{train}} \text{ be the state calculated using a mixed state (eq. (8)} \text{ and quantum feature maps } \psi_X \text{ and } \psi_Y. \text{ Then the density matrix } \rho'_y, \text{ calculated with eq. (12), can be expressed as:}
\]
\[
\rho'_y = \mathcal{M} \sum_{i=1}^{n} |k(x^*, x_i) \rangle \langle y(y_i) | \langle y(y_i)|,
\]
(A7)

where \(k(x^*, x_i) = \langle \psi_X(x^*) | \psi_X(x_i) \rangle \) and \(\mathcal{M}^{-1} = n \text{ Tr}[\pi(x^*) \rho_{\text{train}} \pi(x^*)].\)

\[
\text{Proof. Equation (8) can be expressed as:}
\]
\[
\rho_{\text{train}} = \frac{1}{n} \sum_{i=1}^{n} |\psi_i \rangle \langle \psi_i |
\]
\[
= \frac{1}{n} \sum_{i=1}^{n} |\psi_X(x_i) \rangle \langle \psi_X(x_i) | \otimes |\psi_Y(y_i) \rangle \langle \psi_Y(y_i) |
\]
\[
= \frac{1}{n} \sum_{i=1}^{n} \pi(x_i) \otimes \pi(y_i),
\]
(A8)
where \(\pi(x_i) = |\psi_X(x_i) \rangle \langle \psi_X(x_i) | \and \pi(y_i) = |\psi_Y(y_i) \rangle \langle \psi_Y(y_i) |. \text{ Applying eq. (11) to eq. (A8) we get:}
\]
\[
\rho' = \mathcal{M} \sum_{i=1}^{n} \sigma(x_i) \sigma(y_i),
\]
(A9)
\[
= \mathcal{M} \sum_{i=1}^{n} |k(x^*, x_i) \rangle \langle y(y_i) | \langle y(y_i)|.
\]
(A10)
Finally, we calculate the partial trace of eq. (A9) to obtain:
\[
\rho'_y = \text{Tr}_X[\rho']
\]
\[
= \mathcal{M} \sum_{i=1}^{n} |k(x^*, x_i) \rangle \langle y(y_i)|.
\]

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Nature 549, 195 (2017).
[2] M. Schuld, I. Sinayskiy, and F. Petruccione, Contemporary Physics 56, 172 (2015).
[3] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, Quantum Science and Technology 3, 035002 (2018).
[4] M. Schuld and F. Petruccione, Supervised Learning with Quantum Computers, Springer International Publishing, Cham (2018).
[5] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, Tech. Rep. (2013).
[6] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum Information and Computation 15, 318 (2015).
[7] G. Sergeioli, E. Santucci, L. Didaci, J. A. Miszczak, and R. Giuntini, Soft Computing 22, 691 (2018).
[8] M. Schuld, M. Fingerhuth, and F. Petruccione, EPL (Europhysics Letters) 119, 60002 (2017).
[9] D. Anguita, S. Ridella, F. Rivieccio, and R. Zunino, Neural Networks 16, 763 (2003).
[10] P. Rebentrost, M. Mohseni, and S. Lloyd, Physical Review Letters 113, 130503 (2014).
[11] Z. Li, X. Liu, N. Xu, and J. Du, Physical Review Letters 114, 140504 (2015).
[12] S. Lu and S. L. Braunstein, Quantum Information Processing 13, 575 (2014).
[13] M. Schuld and F. Petruccione, Scientific Reports 8, 2772 (2018).
[14] M. Schuld, I. Sinayskiy, and F. Petruccione, Physics Let-
[15] M. Schuld, I. Sinayskiy, and F. Petruccione, Quantum Information Processing 13, 2567 (2014).

[16] N. Wiebe, A. Kapoor, and K. M. Svore, in Advances in Neural Information Processing Systems (2016) pp. 4006–4014.

[17] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. S. Kim, npj Quantum Information 3, 36 (2017).

[18] E. M. Stoudenmire and D. J. Schwab, in Advances in Neural Information Processing Systems (2016) pp. 4806–4814.

[19] E. M. Stoudenmire, Quantum Science and Technology 3, 034003 (2018).

[20] S. Khus and P. Gelß, Algorithms 12, 240 (2019).

[21] L. Zhang, P. Zhang, X. Ma, S. Gu, Z. Su, and D. Song, in Proceedings of the AAAI Conference on Artificial Intelligence Vol. 33 (2019) pp. 7450–7458.

[22] I. Glasser, R. Sweke, N. Pancotti, J. E. A. i. N. . . ., and U. 2019, in Advances in Neural Information Processing Systems (2019) pp. 1496–1508.

[23] C. M. Bishop, Pattern recognition and machine learning (springer, 2006).

[24] A. Y. Ng and M. I. Jordan, in Advances in neural information processing systems (2002) pp. 841–848.

[25] K. Jacobs, Quantum measurement theory and its applications (Cambridge University Press, 2014).

[26] M. Plesch and i. c. v. Brukner, Phys. Rev. A 83, 032302 (2011).

[27] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Phys. Rev. Lett. 87, 167902 (2001).

[28] J. C. Garcia-Escartín and P. Chamorro-Posada, Phys. Rev. A 87, 052330 (2013).

[29] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, New Journal of Physics 20, 113022 (2018).

[30] M. Schuld and N. Killoran, Physical Review Letters 122 (2019), 10.1103/PhysRevLett.122.040504.

[31] R. Chatterjee and T. Yu, Quantum Information and Communication 17, 1292 (2017).

[32] A. Rahimi and B. Recht, in Advances in neural information processing systems (2008) pp. 1177–1184.

[33] T. Jaakkola and D. Haussler, in Advances in neural information processing systems (1999) pp. 487–493.

[34] K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf, et al., Foundations and Trends® in Machine Learning 10, 1 (2017).