Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 467 (2023) 108509

FUZZY sets and systems

www.elsevier.com/locate/fss

Multi-class classification based on quantum state discrimination

Roberto Giuntini [a], Andrés Camilo Granda Arango [a], Hector Freytes [a], Federico Hernan Holik [a,b], Giuseppe Sergioli [a,*]

[a] *University of Cagliari, Via Is Mirrions 1, Cagliari, 09123, Italy*
[b] *Universidad Nacional de La Plata, Instituto de Fisica (IFLP-CCT-CONICET), C.C. 727, La Plata, 1900, Argentina*

Received 17 September 2022; received in revised form 15 March 2023; accepted 23 March 2023
Available online 28 March 2023

Abstract

We present a general framework for the problem of multi-class classification using classification functions that can be interpreted as fuzzy sets. We specialize these functions in the domain of Quantum-inspired classifiers, which are based on quantum state discrimination techniques. In particular, we use unsharp observables (Positive Operator-Valued Measures) that are determined by the training set of a given dataset to construct these classification functions. We show that such classifiers can be tested on near-term quantum computers once these classification functions are "distilled" (on a classical platform) from the quantum encoding of a training dataset. We compare these experimental results with their theoretical counterparts and we pose some questions for future research.

Keywords: Quantum-inspired algorithms; Multi-class classification; Pretty Good Measurement

1. Introduction

Recent years have witnessed rapid development in quantum computers and other quantum information devices. Along with reports of quantum computers successfully performing tasks that are extremely challenging for classical computers [53,54,4,51,29], the emergence of the NISQ[1] era [35] has encompassed a diverse range of firmly established quantum algorithms [30,11].

In parallel to these developments, several mathematical and computational techniques that were originally designed to study quantum systems and quantum information theory problems have been adapted to solve different computa-

* Corresponding author.
E-mail address: giuseppe.sergioli@gmail.com (G. Sergioli).

[1] The acronym NISQ (Noisy Intermediate Scale Quantum) refers to quantum computers with a limited number of qubits (between a few tens and a hundred) and significant sensitivity to environmental factors during quantum operations. These devices are intermediate in scale with respect to the number of qubits and lack robust error correction protocols, limiting their ability to achieve the full potential of quantum computing. NISQ computers are seen as a stepping stone towards building large-scale (tens of thousands of qubits), fault-tolerant quantum computers, and are capable of performing certain quantum algorithms that are already challenging for classical computers. The term "NISQ" was first introduced by John Preskill [35].

https://doi.org/10.1016/j.fss.2023.03.012
0165-0114/

tional problems outside the domain of quantum physics [31,33,34,12,2,8,26,43,40,39,41,36,21,22,3,32,13,17,18,45,46,52]. The distinctive feature of the latter algorithms is that while relying on a mathematical formalism based on quantum theory, they can be implemented on fully classical hardware and still achieve good performance. Their analysis can also shed light on the differences between quantum and classical computing [3]. We refer to all these techniques as *quantum-inspired algorithms*. Thus, quantum-inspired algorithms can be added to the zoo of available technologies coexisting in the NISQ era.

In previous works, different quantum-inspired algorithms for the classification of classical data have been developed [43,40,39,41,36]. These algorithms are based on the idea of finding a representation of classical datasets in terms of quantum representatives. More specifically, each feature-vector of a dataset is encoded in a pure quantum state. This process is called *quantum encoding*. Then, using classical hardware, quantum state discrimination techniques (taken from quantum information theory) are applied to the quantum representatives to find a solution to the classification problem at issue. This is the case, for example, of the *Helstrom quantum classifier* (HQC), which turns out to be particularly beneficial in terms of accuracy for binary classification [42]. The problem, however, is that currently no method to extend HQC to multi-class classification is known.

To overcome this difficulty, in this work we take inspiration from the so-called *Pretty Good Measurement* [50] to define a quantum-inspired multi-class classifier. The proposed algorithm first assigns a (not necessarily projective) positive operator-valued measure (POVM)[2] (the Pretty Good Measurement) to a given training dataset. This POVM is then used to assign, via Born's rule, a probability-value to each element of the test dataset. Such a procedure allows one to define a function that *directly* classifies any object of the test dataset, without resorting to the (time consuming) strategies such as *One-vs-One* or *One-vs-All*) that are used for classifiers that do not natively support classification tasks with more than two classes.[3] We refer to this algorithm as the *PGM classifier*. Just like the Helstrom classifier, the PGM classifier, being quantum-inspired, brings a concrete computational advantage when running on fully classical hardware. The main goal of this work is to provide a precise mathematical description of the PGM classifier algorithm.

While the PGM classifier can be fully implemented on a classical computer, in this work we show how to test the PGM classifier using a quantum circuit, along with numerical simulations of its performance. The comparison with the classical-implemented version is useful to better understand the important differences between quantum and quantum-inspired algorithms (see also [49,32]).

The paper is organized as follows. In Section 2, we first review the differences between Quantum Machine Learning (QML) and Quantum-inspired Machine Learning (QiML), with a special focus on the problem of classification. Then, in Section 3, we describe the general settings of the quantum-inspired classification algorithms. In Section 4, we introduce the theoretical basis of the classification algorithms inspired by quantum state discrimination. More specifically, we first summarize the formal setting of the binary HQC, and then introduce the details of the PGM classifier. In Section 5, we describe how to test the PGM classifier using a quantum circuit. Finally, in Section 6, we draw some conclusions and we pose some relevant questions for future research.

2. Quantum-inspired and Quantum Machine Learning. Two different approaches

Machine learning is one of the most important subfields of artificial intelligence. It is based on the idea of programming an automated device to learn from previous computations and gradually improve its performance during the same computational process. In what follows, we restrict ourselves to the *supervised* approach, where a set of *a priori* known data (the *training dataset*) is used to make an accurate prediction about another set of data (the *test dataset*). Obviously, different strategies can produce results with different levels of accuracy. The use of quantum information techniques to improve classical machine learning algorithms has become a very promising area of research [37,10,38], and is included in the field of *Quantum Machine Learning* (QML). For an extended update on recent advances in this field, see [38].

However, QML is not the only approach to machine learning that makes use of quantum theory. An alternative approach that has emerged in the last years is known as *Quantum-inspired Machine Learning* (QiML) [17,18,45,46,52]. This approach is very different from QML in spirit, and also in the kind of benefits it brings. The QiML approach

[2] For a detailed analysis of the close connection between POVMs and fuzzy observables, see [1].

[3] A numerical comparison between these approaches is analyzed in a separate work [19].

is based on the fact that it is possible to tackle certain machine learning problems by using theoretical tools commonly used in quantum theory. The main difference between QML and QiML is that the latter does not necessarily require the use of actual quantum devices. The idea is that by reformulating a machine learning problem as a "quantum problem", it is possible to harness the power of the mathematical formalism of quantum theory to achieve remarkable benefits. Remarkably, these algorithms can be implemented on classical computers. The benefits do not necessarily give rise to a reduction in computational complexity. In some cases, improvements in other relevant elements of the computational process, such as accuracy, can be obtained.

In the rest of the paper we focus on a particular machine learning problem, namely the well-known *classification problem*. It has been widely studied in the standard literature and has a very wide range of applications in many different practical contexts [14]. In particular, we summarize the formal framework of a recent quantum-inspired model for classification.

2.1. Classification

Put simply, the general idea of classification is to design an algorithm that can evaluate the features of a class of objects and, given an unknown object, decide whether it belongs to a given class.

The choice of features considered relevant to characterize a given object depends on the application and the goals of the classification itself. Some common approaches to feature selection include correlation analysis, domain knowledge, wrapper methods, embedded methods and filter methods [15]. In many cases, a combination of methods is used.

Supervised classification refers to the idea that the whole process is supervised by an "expert" who first builds a preliminary set of correctly classified objects (*training dataset*). As a result of this "training", the algorithm is designed to classify new "unseen" objects (or objects from the test dataset) as accurately as possible.

Objects are described by a certain number d of *features*. These features are chosen as quantifiable aspects of objects that are relevant enough to describe them in the classification framework. For example, if we were using a database of different species of mammals, we could describe an individual by a vector formed by the values of its weight, width and length. Thus, we could associate a three-dimensional vector with the values of these quantities for each individual. Of course, other features could be used for different classification purposes.

More formally, each object x is associated with a vector $\vec{x}$ (called *object-vector* or *feature-vector*) of a d-dimensional Hilbert space $\mathcal{H}^d$.[4] A *pattern* is defined as a pair $(\vec{x_j}, \lambda_j)$ where $\vec{x_j}$ is a feature-vector and λ_j is the class label which denotes the class which the object is supposed to belong to. For simplicity, we identify the set L of all class-labels with a finite sequence $(1, \ldots, \ell)$ of natural numbers that are in one-to-one correspondence with the ℓ classes which the objects belong to. Thus, a *training dataset* can be represented as a set

$$\mathcal{S}_{\text{tr}} := \{(\vec{x}_1, \lambda_1), \ldots (\vec{x}_m, \lambda_m)\},$$

where $\forall j \in \{1, \ldots, m\}$: $\lambda_j \in L$. Given any class label $i \in L$, we can define the set $\mathcal{S}^i_{\text{tr}}$ of all object-vectors whose associated class label is i:

$$\mathcal{S}^i_{\text{tr}} := \{\vec{x}_j \in \mathcal{S}_{\text{tr}} : \lambda_j = i\}. \tag{1}$$

The cardinality of $\mathcal{S}^i_{\text{tr}}$ is denoted by $|\mathcal{S}^i_{\text{tr}}|$. Clearly, $\sum_{i=1}^{\ell} |\mathcal{S}^i_{\text{tr}}| = m$.

The task of supervised classification is to infer a classifier-function (simply, a *classifier*) from the training dataset that assign, as accurately as possible, a class-label to any object-vector $\vec{x}$.

Formally, a classifier can be defined as a map

$$Cl : \mathbb{C}^d \to L.$$

Let $\mathcal{S}_{\text{tr}} = \{(\vec{x}_1, \lambda_1), \cdots, (\vec{x}_m, \lambda_m)\}$ be a training dataset. In order to define a classifier (based on $\mathcal{S}_{\text{tr}}$) one defines a map f (called "learning function") that associates to any feature-vector $\vec{x}$ a sequence of ℓ-numbers belonging to the unit real-interval $[0, 1]$:

[4] Unlike standard representations in machine learning, we do not exclude features that can be represented as complex numbers.

$$f : \mathbb{C}^d \to [0,1]^\ell.$$

The i^{th}-component of $f(\vec{x})$ will be denoted by $f(\vec{x})_i$.

The interpretation of the i-th component of $f(\vec{x})$ depends on the intended meaning of the function f itself. For example, if one adopts a "fuzzy perspective", the value $f(\vec{x})_i$ may represent the *degree of membership* of the object x (whose object-vector is $\vec{x}$) to the class labeled by λ_i. Thus, any label i determines a fuzzy set $f_i : \mathbb{R}^d \to [0,1]$ such that for any $\vec{x} \in \mathbb{R}^d$, $f_i(\vec{x}) = f(\vec{x})_i$. In a probabilistic framework, instead, $f(\vec{x})$ is assumed to be a *probability-vector* (i.e. $\sum_{i=1}^{\ell} f(\vec{x})_i = 1$) and the value $f(\vec{x})_i$ can be interpreted as the probability that the object x (with associated feature-vector $\vec{x}$) belongs to the class labeled by i.

The *classifier determined by* f (or simply, the f*-classifier*) is the map

$$Cl_f : \mathbb{C}^d \to L$$

that assigns to any feature-vector $\vec{x} \in \mathbb{C}^d$ the class-label that is associated to the greatest value of $f(\vec{x})_i$, with $1 \leq i \leq \ell$.

Since it may happen that f returns more than one class-label when there are matching $f(\vec{x})_i$ values, we pose by convention

$$Cl_f(\vec{x}) := \min\{i \in L : f(\vec{x})_i = \max\{f(\vec{x})_k : 1 \leq k \leq \ell\}\}. \tag{2}$$

An f-classifier induced by a learning function f shows clear analogies with the defuzzification process in fuzzy set theory [48], which involves mapping a fuzzy set to a crisp set.

A classifier Cl_f is called *probabilistic* iff for any $\vec{x} \in \mathbb{C}^d$ we have: $\sum_{i=1}^{\ell} f(\vec{x})_i = 1$. In other words, a classifier is probabilistic iff for any $\vec{x}$ the sequence $(f(\vec{x})_1, \ldots, f(\vec{x})_\ell)$ is a probability-vector.

3. General setting for Quantum-inspired classification

As introduced in the previous sections, the QiML approach is based on the idea of using quantum information theory to improve the learning process, even without the use of a quantum computer. In the following, we will describe the main ingredients of the QiML approach to supervised classification.

Given a training dataset, the construction of a quantum classifier is based on three basic steps: i) applying a *quantum feature map* (or *quantum encoding*) to encode the object-vectors of the training dataset into quantum objects [41]; ii) finding a suitable learning function f that determines the quantum classifier; iii) applying the quantum classifier to a quantum-encoded *object-vectors* to obtain the labels of the classes which the objects belong to.

Let us consider a training dataset $\mathcal{S}_{\text{tr}} = \{(\vec{x}_1, \lambda_1), \cdots, (\vec{x}_m, \lambda_m)\}$. A quantum encoding is a map that associates with any object-vector $\vec{x}$ of $\mathbb{C}^d$ a pure quantum state (called *object quantum-state*) $\rho_{\vec{x}}$ of a Hilbert space $\mathbb{C}^n$, whose dimension n depends on the number of the d-features. Given a quantum encoding $\vec{x} \mapsto \rho_{\vec{x}}$, a *quantum pattern* is any pair $(\rho_{\vec{x}_j}, \lambda_j)$. A *quantum training dataset* is defined as the set of all quantum patterns:

$$\mathcal{S}_{\text{Qtr}} := \left\{(\rho_{\vec{x}_1}, \lambda_1), \ldots, (\rho_{\vec{x}_m}, \lambda_m)\right\}.$$

Given any class label $i \in L$, we can also define the set $\mathcal{S}^i_{\text{Qtr}}$ as the set of all object quantum-states $\rho_{\vec{x}_j}$ that are associated to the set $\mathcal{S}^i_{\text{tr}}$ of all i-object-vectors:

$$\mathcal{S}^i_{\text{Qtr}} := \{\rho_{\vec{x}_j} : \vec{x}_j \in \mathcal{S}^i_{\text{tr}}\}. \tag{3}$$

Similarly to the case of the Nearest-Mean Classifier [14] where one associates to any class of object-vectors its centroid, in our QiML approach, one can define the crucial notion of *quantum centroid*. Intuitively, the quantum centroid associated to the class-label i is the density operator (in an appropriate Hilbert space) that represents the uniform sampling of all i-object quantum states.

Definition 3.1. Let $i \in L$ be a class-label. The quantum centroid associated to i (denoted by $\rho_{(i)}$) is:

$$\rho_{(i)} := \frac{1}{|\mathcal{S}^i_{\text{Qtr}}|} \sum_{\vec{x}_j \in \mathcal{S}^i_{\text{tr}}} \rho_{\vec{x}_j},$$

where $|\mathcal{S}^i_{\text{Qtr}}|$ is the cardinality of $\mathcal{S}^i_{\text{Qtr}}$. Clearly, $|\mathcal{S}^i_{\text{Qtr}}| = |\mathcal{S}^i_{\text{tr}}|$.

As one can easily realize, the ℓ class-labels are in one-to-one correspondence with the set $\{\rho_{(1)}, \ldots, \rho_{(\ell)}\}$ of all quantum centroids.

Despite they similarities, classical and quantum centroids behave in a quite different way as the following remark shows.

Remark 3.1. Let i be a class-label and let $\vec{C}_{(i)} = \frac{1}{|\mathcal{S}^i_{\text{tr}}|} \sum_{\vec{x}_j \in \mathcal{S}^i_{\text{tr}}} \vec{x}_j$ be the (classical) centroid associated to i. As one can easily realize, the quantum encoding $\rho_{\vec{C}_{(i)}}$ of $\vec{C}_{(i)}$ does not coincide with the quantum centroid $\rho_{(i)}$ associated to the class-label i (see Definition 3.1).

Remark 3.2. In the classical case, the Nearest-Centroid classification is invariant under scalar multiplication of object-vectors. On the contrary, it turns out that classification based on quantum centroids is no longer invariant under scalar multiplication. This characteristic is shown to be beneficial for classification tasks [39].

Let us consider a quantum training dataset $\mathcal{S}_{\text{tr}} = \{(\vec{x}_1, \lambda_1), \ldots (\vec{x}_m, \lambda_m)\}$. How to define a possible "quantum learning function" f in terms of $\mathcal{S}_{\text{Qtr}}$? As happens in the classical case, different answers are possible. In our approach we take the move to interpret the set of all class-labels as possible outcomes of a measurement. Let us briefly recall the notion of (quantum) measurement.

Let $\mathcal{B}(\mathcal{H})^+$ be the set of all positive semidefinite operators acting on a finite dimensional Hilbert space $\mathcal{H}$. A *measurement* is defined as a map $\mathcal{M}$ from a finite non-empty set $\mathcal{O}$ (representing a set of possible outcomes of a physical quantity) into $\mathcal{B}(\mathcal{H})^+$ such that $\sum_{i \in \mathcal{M}} \mathcal{O}(i) = \mathbb{I}$. A measurement is said to be a *von Neumann measurement* iff every $\mathcal{M}(i)$ is a projection, i.e., $\mathcal{M}(i)^* = \mathcal{M}(i)\mathcal{M}(i) = \mathcal{M}(i)$, where $\mathcal{M}(i)^*$ is the adjoint of $\mathcal{M}(i)$.

A *quantum classifier* is a classifier Cl_f, where the function f is determined by a measurement $\mathcal{M} : L \to \mathcal{B}(\mathcal{H})^+$ (see [50], Definition 2.34).

More precisely, the notion of quantum classifier is defined as follows.

Definition 3.2. A quantum classifier is a classifier Cl_f (see Equation (2)) such that the learning function $f : \mathbb{C}^d \to [0, 1]^\ell$ satisfies the following condition: there exists a measurement $\mathcal{M} : L \to \mathcal{B}(\mathcal{H})^+$ such that

$$\forall \vec{x} \in \mathbb{C}^d : f(\vec{x})_i = \text{tr}(\mathcal{M}(i)\rho_{\vec{x}}),$$

where tr is the trace of a matrix.

By definition of measurement, it turns out that $\sum_{i=1}^{\ell} \text{tr}(\mathcal{M}(i)\rho_{\vec{x}}) = 1$. Therefore, a quantum classifier is always probabilistic.

Intuitively, the i-th component $f(\vec{x})_i$ of the learning function f represents the probability that the object encoded by $\rho_{\vec{x}}$ belongs to the class i.

An interesting question is whether classification accuracy can be improved by increasing the dimension of the state space where the object quantum-states live.

As well known, one of the most powerful technique of (classical) machine learning is represented by *kernel method* [44], where the possibly non-linearly separable classes of a dataset become "more separable" by mapping the object-vectors into a higher dimensional feature space. The question naturally arises whether also in our approach one may hope to improve the performance of a quantum-inspired classifier by mapping the object quantum-states into a higher dimensional Hilbert space. Although computation in a larger feature space generally increases runtime, the expected improvement in prediction accuracy is crucial in certain machine learning applications, such as those specialized in medical diagnosis.

In our case, the dimensional increasing of the feature space is obtained by encoding any object-vector $\vec{x}$ as a tensor product $\rho_{\vec{x}} \underbrace{\otimes \ldots \otimes}_{\text{n-times}} \rho_{\vec{x}}$ of the object quantum-states $\rho_{\vec{x}}$. Accordingly, the set of all object quantum-states that are associated to the n-copies of all i-objects is defined by tensorizing the object quantum-states in $\mathcal{S}^i_{\text{Qtr}}$:

$$\mathcal{S}^{i^{(n)}}_{\mathrm{Qtr}} := \{\rho_{\vec{x}_j} \underbrace{\otimes \ldots \otimes}_{n\text{-times}} \rho_{\vec{x}_j} : \vec{x}_j \in \mathcal{S}^i_{\mathrm{tr}}\}.$$

Accordingly, the n-copy quantum centroid of $\mathcal{S}^{i^{(n)}}_{\mathrm{Qtr}}$ can be defined as follows:

$$\rho^{(n)}_{(i)} := \frac{1}{|\mathcal{S}^i_{\mathrm{Qtr}}|} \sum_{\vec{x}_j \in \mathcal{S}^i_{\mathrm{tr}}} \rho_{\vec{x}_j} \underbrace{\otimes \ldots \otimes}_{n\text{-times}} \rho_{\vec{x}_j}. \tag{4}$$

It should be noticed that, in general, $\rho^{(n)}_{(i)} \neq \rho_{(i)} \underbrace{\otimes \ldots \otimes \rho_{(i)}}_{\text{n-times}}$.

The $\otimes^n$-generalization of the quantum classifier introduced in Definition 3.2 can now be naturally defined as a function $f : \mathbb{C}^{d^n} \to [0,1]^\ell$ determined by a measurement $\mathcal{M} : L \to \mathcal{B}(\otimes^n \mathbb{C}^d)^+$. As we will show in the sequel, this procedure turns out to be advantageous in improving classification accuracy.

4. Classification inspired by quantum state discrimination

Since feature-vectors are encoded as quantum states and quantum centroids are quantum states it is natural to apply measurement techniques developed in the context of *quantum state discrimination* ([5,24,6,50]) to the classification problem. We can conceptualize the machine learning task as finding a quantum measurement that distinguishes quantum centroids in an optimal (or sub-optimal) way, which, as we will see, allows us to obtain highly performing classifiers.

The task of discriminating quantum states is a fundamental problem in quantum information theory, with deep implications in quantum cryptography and quantum error correction [9]. The problem of quantum state discrimination can be summarized as follows. Let us suppose that Alice wishes to send a message to Bob by using a quantum channel. To do this, Alice selects a state ρ_i with an a priori probability p_i from a given set of possible states in the Hilbert space $\mathbb{C}^d$. We indicate the ensemble R of these possible states with their respective a priori probability-values as follows:

$$R = \{(\mathrm{p}_1, \rho_1), \cdots, (\mathrm{p}_\ell, \rho_\ell)\},$$

where $(\mathrm{p}_i, \ldots, \mathrm{p}_\ell)$ is a probability-vector (a sequence of positive real numbers such that $\sum_{i=1}^{\ell} \mathrm{p}_i = 1$).

Bob knows a priori both the set of possible states and their associated a priori probabilities. His task is to determine, by means of a suitably chosen measurement, the state ρ_i he receives from Alice, and hence the intended message. However, the problem of finding an optimal strategy for discrimination among arbitrary states is still unresolved. Although error-free solutions can be found for some particular cases (for example when R contains only mutually orthogonal states), errors will necessarily occur in the discrimination process of arbitrary states (see, for example, [5]). This means that, given an ensemble $R = \{(\mathrm{p}_1, \rho_1), \cdots, (\mathrm{p}_\ell, \rho_\ell)\}$, in general there exists no measurement $\mathcal{M}$ such that for any $i, j \in \{1, \ldots \ell\}$ with $i \neq j$: $\mathrm{tr}(\mathcal{M}(i)\rho_j) = 0$. Thus, once Alice sends the i-th state to Bob, he can either conclude (erroneously) that he was given the state ρ_j or, conversely, he can conclude (correctly) that he was given the state ρ_i (*successful discrimination*). The average probability for Bob to perform a successful discrimination by means of a given measurement $\mathcal{M}$ is given by

$$\mathrm{p}^{\mathcal{M}}_{succ}(R) := \sum_{i=1}^{\ell} \mathrm{p}_i \mathrm{tr}(\mathcal{M}(i)\rho_i). \tag{5}$$

In order to minimize the error probability in the discrimination problem, it is necessary to find an optimal measurement $\mathcal{M}$ which maximizes Equation (5) or equivalently, that minimizes the *discrimination error probability* $1 - \mathrm{p}^{\mathcal{M}}_{succ}(R)$. One can prove (see [50]) that, for any ensemble R, there exists an *optimal measurement* $\mathcal{M}$, denoted by $Opt(R)$, such that

$$Opt(R) = \max\{\mathrm{p}^{\mathcal{N}}_{succ}(R) : \mathcal{N} \textit{ is a measurement}\}.$$

In other words, $Opt(R)$ is the measurement that allows Bob to achieve the maximum probability of successfully discriminating the states in R.

4.1. Helstrom measurement and binary classification

In 1969, C. W. Helstrom found an exact analytical description for the optimal measurement for ensembles of two quantum states in his seminal paper [24]. Let

$$R = \{(\mathrm{p}_1, \rho_1), (\mathrm{p}_2, \rho_2)\}$$

be an ensemble of two quantum states with a priori probabilities p_1 and $\mathrm{p}_2 = 1 - \mathrm{p}_1$. Let us define the *Helstrom observable* as

$$\Lambda = \mathrm{p}_1\rho_1 - \mathrm{p}_2\rho_2. \tag{6}$$

Let l_+ and l_- be the sets of all eigenvectors determined by the positive and negative eigenvalues of Λ, respectively. Let $P_+ := \sum_{\lambda \in l_+} P_\lambda$ and $P_- := \sum_{\lambda \in l_-} P_\lambda$, where P_λ denotes the projection associated to the eigenspace determined by the eigenvalue λ. Intuitively, P_+ and P_- represent the property of the measurement to correctly identify a state as being in state ρ_1 (ρ_2, respectively). The set $\{P_+, P_-\}$ determines a (von Neumann) measurement (called *Helstrom measurement*) since $P_+ + P_- = \mathbb{I}$. Helstrom proved that this measurement is optimal, meaning that it achieves the maximum probability of successfully discriminating between the two states in the ensemble. Such a maximum probability – called the *Helstrom bound of* R ($\mathtt{H_b}(R)$) – is given by

$$\mathtt{H_b}(R) := \mathrm{p}_1 \mathrm{tr}(P_+\rho_1) + \mathrm{p}_2 \mathrm{tr}(P_-\rho_2), \tag{7}$$

which is the sum of the probabilities of correctly identifying the two states weighted by their a priori probabilities. It turns out (see [50]) that

$$\mathtt{H_b}(R) = \frac{1}{2} + \frac{1}{2}\mathtt{T}\left(\mathrm{p}_1\rho_1, \mathrm{p}_2\rho_2\right)$$

where $\mathtt{T}$ is the trace distance induced by the trace norm $\|\cdot\|_1$:

$$\mathtt{T}\left(\mathrm{p}_1\rho_1, \mathrm{p}_2\rho_2\right) := \frac{1}{2}\left\|\mathrm{p}_1\rho_1 - \mathrm{p}_2\rho_2\right\|_1.$$

In the case when $\mathrm{p}_1 = \mathrm{p}_2 = \frac{1}{2}$, we will write $\mathtt{H_b}(\rho_1, \rho_2)$ instead of $\mathtt{H_b}(R)$. The Helstrom bound $\mathtt{H_b}(\rho_1, \rho_2)$ satisfies the following properties (see [50]):

i) $0 \leq \mathtt{H_b}(\rho_1, \rho_2) \leq \mathtt{1}$;
ii) $\mathtt{H_b}(\rho_1, \rho_2) = \frac{1}{2}$ iff $\rho_1 = \rho_2$;
iii) $\mathtt{H_b}(\rho_1, \rho_2) = \mathtt{1}$ iff $im(\rho_1)$ is orthogonal to $im(\rho_2)$, where $im(\rho)$ is the subspace spanned by the image of ρ.

Under Helstrom's formalism, let us now consider the particular case of binary classification, where $L := \{1, 2\}$. After quantum encoding, we obtain the quantum training datasets S^1_{Qtr} and S^2_{Qtr}, (defined according to Equation (3)), and ρ_1 and ρ_2 as their respective quantum centroids (see Definition 3.1). In this way, it is possible to define the *Helstrom observable* (as in Equation (6)) for the two quantum centroids ρ_1 and ρ_2, that we assume to have the same a priori probability-values, i.e., $p_1 = 1 - p_2 = \frac{1}{2}$.[5] The Helstrom measurement $\{P_+, P_-\}$ gives rise to a learning function f defined as follows:

$$\forall \vec{x} \in \mathbb{C}^d : f(\vec{x})_1 := \mathrm{tr}(P^+\rho_{\vec{x}}) \text{ and } f(\vec{x})_2 := \mathrm{tr}(P_-\rho_{\vec{x}}).$$

According to Definition 3.2, we can define the quantum classifier induced by f (called *Helstrom Quantum Classifier*) in the following way:

$$Cl_f(\vec{x}) := \begin{cases} 1, \text{ if} & f(\vec{x})_1 \geq f(\vec{x})_2; \\ 2, & \text{otherwise.} \end{cases}$$

[5] In same cases, the a priori probability-values p_1 and p_2 are assumed to be: $p_1 = \frac{|S^1_{Qtr}|}{|S_{Qtr}|}$ and $p_2 = \frac{|S^2_{Qtr}|}{|S_{Qtr}|}$.

In [42], an experiment is presented where the Helstrom Quantum Classifier (HQC) is applied to fourteen different datasets and compared with eleven different standard classifiers. The results show that HQC exhibits a high level of accuracy when compared to these standard classifiers. In that paper, the application of quantum state discrimination to the classification problem led the authors to the following notable conclusion: the higher the probability of successfully discriminating between quantum centroids, the better the accuracy performance of HQC. In other words, the increase of the Helstrom bound of the two quantum centroids seems to be strongly correlated with an increase in classification accuracy. The Helstrom bound, which is a measure of the distinguishability between the two quantum centroids, seems to set a fundamental limit on the accuracy that can be achieved by HQC. The experiment given in [43,42] provides empirical evidence of this claim.

As discussed at the end of Section 3, it is possible to perform the tensor product of each object quantum-state. This allows us to define the new quantum centroids $\rho_1^{(n)}$ and $\rho_2^{(n)}$ that give rise to the Helstrom measurement $\{P_+^{(n)}, P_-^{(n)}\}$ and to the following learning function $f^{(n)}$:

$$\forall \vec{x} \in \mathbb{C}^d : f^{(n)}(\vec{x})_1 := \mathrm{tr}(P_+^{(n)} \rho_{\vec{x}}^{(n)}) \text{ and } f^{(n)}(\vec{x})_2 := \mathrm{tr}(P_-^{(n)} \rho_{\vec{x}}^{(n)}).$$

Thus, the Helstrom Quantum Classifier takes the form

$$Cl_f^{(n)}(\vec{x}) := \begin{cases} 1, \text{ if} & f^{(n)}(\vec{x})_1 \geq f^{(n)}(\vec{x})_2; \\ 2, & \text{otherwise.} \end{cases}$$

The empirical results obtained in [42] strongly suggest that taking tensor products of multiple copies of the object quantum-states turns out to be beneficial for classification tasks, as the following Theorem seems to clearly indicate [19].

Theorem 1. *For any* $n \in \mathbb{N}^+$,

$$\mathrm{H_b}(\rho_{(1)}^{(n)}, \rho_{(2)}^{(n)}) \leq \mathrm{H_b}(\rho_{(1)}^{(n+1)}, \rho_{(2)}^{(n+1)}).$$

According to Theorem 1, increasing the number of copies of the object quantum states (and consequently, increasing the number of factors in the tensor product) leads to a higher Helstrom bound, resulting in increased accuracy performance of HQC (see the discussion in [19]).

Many questions remain open, and an extensive investigation into the relationship between the Helstrom bound and statistical scores in machine learning problems is ongoing. The results obtained so far motivate further research in this area. It is also relevant to note that, although increasing the number of copies in the tensor product can potentially improve the accuracy performance of the quantum classifier, it also incurs an additional computational cost (i.e., the runtime of the algorithm). Hence, depending on the size of the dataset and the machine being used, the choice of the maximum number of tensor copies should be based on practical considerations. This leads to another interesting question for future work, namely, whether it is possible to use quantum hardware to speed up the classification protocol.

4.2. Pretty Good Measurement and multi-class classification

A common strategy for performing multi-class classification is to decompose it into combinatorial compositions of binary classifications using strategies such as “One vs One” or “One vs Rest”. One disadvantage of these strategies is that, especially for datasets containing a large number of classes, a large number of binary classifications are required, resulting in increased complexity.

In the following, we demonstrate how quantum state discrimination suggests an alternative method for multi-class classification that does not require any combinatorial decomposition into binary classification processes

Given an ensemble of possible states with their respective a priori probabilities

$$R = \{(\mathrm{p}_1, \rho_1), \cdots, (\mathrm{p}_\ell, \rho_\ell)\}, \tag{8}$$

it may be difficult to find an analytical description for the exact optimal measurement associated to R. One possible solution is to search for a sub-optimal measurement that can be expressed in an analytical form. This is known as the

so-called *Pretty Good Measurement* [50], which we will introduce in the following. Let us define the average state of R as

$$\sigma = \sum_{i=1}^{\ell} \mathrm{p}_i \rho_i.$$

For each $i : 1 \leq i \leq \ell$, let us define the following operator

$$E_i = (\sigma^{\urcorner})^{1/2} \mathrm{p}_i \rho_i (\sigma^{\urcorner})^{1/2},$$

where $\sigma^{\urcorner}$ is the pseudoinverse (or Moore-Penrose inverse) of σ. The operator E_i is well defined. Indeed, by Theorem 2 (iv) (see Appendix A), $\sigma^{\urcorner}$ is positive since σ is positive, and consequently, the square root of $\sigma^{\urcorner}$ exists and is unique. Since $\mathrm{p}_i \rho_i$ is positive semidefinite and it is enclosed on the left and on the right by the self-adjoint operator $(\sigma^{\urcorner})^{1/2}$, we can conclude that E_i is also positive semidefinite (see [25], p. 431). By Theorem 2 (vi) (see Appendix A), we have $\sigma\sigma^{\urcorner} = \sigma^{\urcorner}\sigma$. Consequently, by Theorem 6.6.4 (see [15], p. 223), $(\sigma^{\urcorner})^{1/2}$ commutes with σ. Thus, by Theorem 2 (vi), we obtain:

$$\begin{aligned}
\sum_{i=1}^{\ell} E_i &= \sum_{i=1}^{\ell} (\sigma^{\urcorner})^{1/2} \mathrm{p}_i \rho_i (\sigma^{\urcorner})^{1/2} \\
&= (\sigma^{\urcorner})^{1/2} \sigma (\sigma^{\urcorner})^{1/2} \\
&= \sigma (\sigma^{\urcorner})^{1/2} (\sigma^{\urcorner})^{1/2} \\
&= \sigma (\sigma^{\urcorner}) \\
&= P_{im(\sigma)}, \qquad (9)
\end{aligned}$$

where $P_{im(\sigma)}$ is the projection onto the subspace spanned by the image of σ.

Since any E_i is positive semidefinite, we have that $E_i \leq \sum_{i=1}^{\ell} E_i = P_{im(\sigma)}$. Hence, E_i is bounded by the identity operator $\mathbb{I}$ and therefore is an effect operator: $\mathbb{O} \leq E_i \leq \mathbb{I}$. However, the set $\{E_i\}_{i=1}^{\ell}$ does not determine a measurement since in general $\sum_{i=1}^{\ell} E_i < \mathbb{I}$. The set $\{E_i\}_{i=1}^{\ell}$ can be easily transformed into a set of ℓ effects that finally induce a measurement.

For each i with $1 \leq i \leq \ell$, let us define the following operators:

$$F_i := E_i + \frac{1}{\ell} P_{ker(\sigma)}, \qquad (10)$$

where $P_{ker(\sigma)}$ is the projection onto the subspace spanned by the kernel of σ. It turns out that the map $\mathcal{F} : \{1, 2, \cdots, \ell\} \to \mathcal{B}(\mathbb{C}^n)^+$ such that for any $i \in \{1, \ldots, \ell\}$:

$$\mathcal{F}(i) = F_i$$

is a measurement since, as we have seen above, $\sum_{i=1}^{\ell} E_i = P_{im(\sigma)}$ and thus

$$\begin{aligned}
\sum_{i=1}^{\ell} F_i &= \sum_{i=1}^{\ell} (E_i + \frac{1}{\ell} P_{ker(\sigma)}) \\
&= \sum_{i=1}^{\ell} E_i + \frac{1}{\ell} \sum_{i=1}^{\ell} P_{ker(\sigma)} \\
&= P_{im(\sigma)} + P_{ker(\sigma)} \qquad (11) \\
&= \mathbb{I}.
\end{aligned}$$

Unlike the Helstrom case, the measurement $\mathcal{F}$ (called *Pretty Good Measurement* [23], shortly PGM) is *unsharp* since the operators F_i are not projections. Therefore, the training dataset will determine a *fuzzy observable*. Since σ is invertible iff $P_{ker(\sigma)} = \mathbb{O}$, we can conclude, from Eqs. (9) and (11), that $E_i = F_i$ iff σ is invertible.

It turns out the Pretty Good Measurement $\mathcal{F}$ is *sub-optimal* [7] since

$$\mathrm{p}^{\mathcal{F}}_{succ}(R) \geq Opt(R)^2.$$

Let us now turn to the general problem of multi-class classification. We will follow a similar formalism to that of the Helstrom observable (for the binary classification case), but using a classifier based on the Pretty Good Measurement formalism. After the quantum encoding procedure, we consider the quantum training datasets S^i_{Qtr} as defined in Equation (3) and their respective quantum centroids $\rho_{(i)}$ as defined in Definition 3.1. Hence, it is possible to consider the ensemble R as defined in Equation (8) where, as in the Helstrom case, the a priori probability-values are assumed to be equal, i.e., for any $i \in \{1, \ldots, \ell\} : p_i = \frac{1}{\ell}$. Thus, we can associate to R a Pretty Good Measurement $\mathcal{F}$ according to Equation (10). In this case, the learning function f is defined as follows:

$$\forall \vec{x} \in \mathbb{C}^d, \forall i \in \{1, \ldots, \ell\} : \ f(\vec{x})_i := \mathrm{tr}(F_i \rho_{\vec{x}}).$$

According to Definition 3.2, the multi-class quantum classifier determined by f, called *Pretty Good Classifier* (*PGM classifier*), is defined as follows:

$$Cl_f(\vec{x}) := \min\{i \in \{1, \cdots, \ell\} : \mathrm{tr}(F_i \rho_{\vec{x}}) = \max\{\mathrm{tr}(F_k \rho_{\vec{x}}) : 1 \leq k \leq \ell\}\}. \tag{12}$$

Notice that if $P_{ker(\sigma)} = \mathbb{O}$, we can replace F_i by E_i in the above equation.

We can also generalize this framework by taking the tensor product of n-copies of states. Thus, the definition of the multi-class quantum classifier introduced in Equation (12), can be naturally extended as

$$Cl_f(\vec{x}) := \min\{i \in \{1, \cdots, m\} : \mathrm{tr}(F_i^{(n)} \rho_{\vec{x}}^{(n)}) = \max\{\mathrm{tr}(F_k^{(n)} \rho_{\vec{x}}^{(n)}) : 1 \leq k \leq \ell\}\}.$$

As in the Helstrom measurement case, it is possible to define a bound for the Pretty Good Measurement as follows:

$$\mathtt{PGM}_{\mathfrak{b}}(R) := \sum_{i=1}^{\ell} p_i \mathrm{tr}(F_i \rho_i).$$

It is currently unknown whether it is possible to obtain a general result regarding the relationship between the value of the PGM bound ($\mathtt{PGM}_{\mathfrak{b}}$) and the number of tensor products (as in Theorem 1) for the Pretty Good Measurement classifier. An initial insight and empirical evidence on this, suggesting an analogous result, is showed in [19].

As discussed at the beginning of the subsection, the PGM classifier avoids the need to decompose an n-ary classification into a combinatorial number of binary classifications, as required by the standard "One versus One" or "One versus Rest" procedures. A detailed experiment comparing the performance of PGM with other standard classifiers is presented in a separate work [20].

5. Representing PGM classifier in IBM-Q

As discussed in Section 2, the two approaches to machine learning, QML and QiML, are fundamentally different: while the former requires the use of a quantum device, the latter does not. Of course, this does not mean that QiML cannot in principle be implemented on quantum devices. Indeed, the aim of this section is to show that the testing part of the PGM classification can be realised on a "simulation backbone" on a near-term quantum computer,[6] such as IBM-Q. As a first step, we use Neumark's dilation theorem [16], which allows us to transform any POVM into the composition of unitary operators. In particular, the PGM can be represented as a suitable combination of quantum gates that can be implemented in a quantum circuit. Based on this intuition, the idea is to consider the PGM classifier process as divided into two distinct steps:

- First, the training part, which aims to define the PGM (and which is computed by a classical hardware).

[6] The term "near-term quantum computer" refers to a type of device that is expected to become available in the near future and to perform specific computations for specific applications, albeit with limited qubit numbers and coherence times. Examples of near-term quantum computers include NISQ devices such as quantum annealers and quantum simulators.

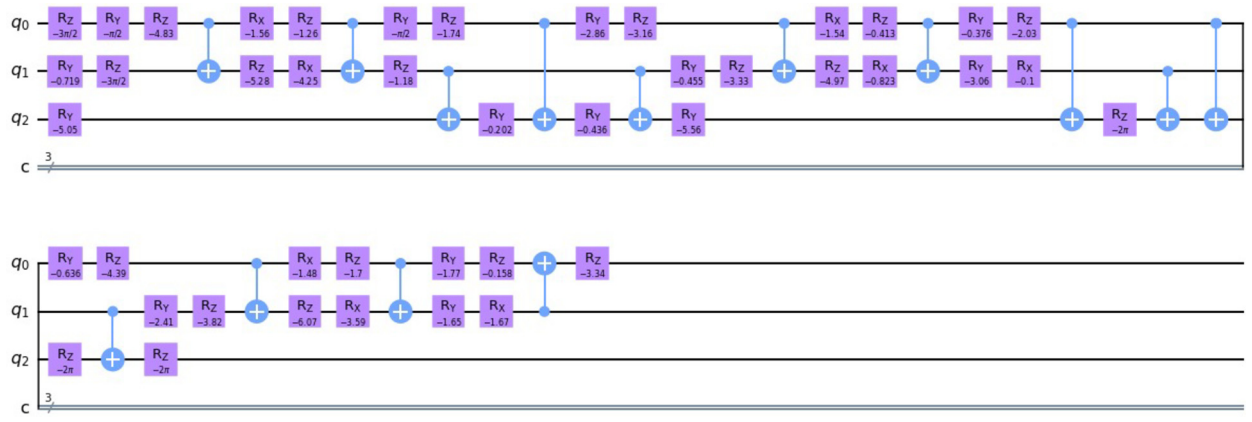

Fig. 1. The circuit that implements the PGM in IBM-Q simulator. This circuit only contains C-Not gates (indicated by the usual representation) and rotations $R_i(n)$ along a given direction i (x, y or z) of a given angle n.

- Second, the *test*, which is based on the application of the PGM to the test set. We show here that this second step can be instantiated in a quantum hardware by applying the quantum circuit implementing the PGM (obtained during training) to each of the quantum states representing the encoded objects of the test set (see also [49] for different implementations of quantum-like algorithms).

The following is a simple example using the following procedure.

- We start with a classical data set (*Lupus* data set [47], given by 87 two-feature vectors divided into two classes[7]), and we perform the quantum coding of the data. We show how to compute the PGM induced by the training set (using classical hardware).
- Next, we decompose the PGM in terms of unitary operators by appealing to Neumark's theorem. We do this using the QCompiler Wolfram package [28] (on classical hardware), which finds a representation in terms of the native gates of the quantum hardware (IBM-Q). The circuit representing the PGM for this example is shown in Fig. 1.
- Finally, we run the circuit in an IBM-Q quantum computer simulator using Qiskit [27]. We apply it to the different quantum states obtained by encoding the objects of the test set. For this example, we do not consider cross-validation, but we have fixed a random partition of the Lupus dataset into 80% elements for the training set and 20% elements for the test set.

The comparison between the expectation results computed in the classical version of the algorithm and the output of the IBM-Q simulation can be seen in Fig. 2 and in Table 1. More precisely, the expectation values of the classical version are obtained by computing the quantities[8] $p_k \mathrm{tr}(F_k \rho_{\vec{x}})$ that we have seen in Equation (12), for $k = 1$ and $k = 2$ (because the lupus data set consists of only two classes). The expectation values correspond to the probabilities of a given quantum object of the quantum test set being classified in the first or second class, respectively. Each quantum state of the test set is prepared as an actual quantum state in IBM-Q. In this way we run the circuit repeatedly to obtain the statistics associated with the application of the PGM circuit to each quantum test object. The output $|000\rangle$ corresponds to the classification in the first class, while the output $|100\rangle$ corresponds to the classification in the second class. We first consider a noiseless simulation with finite statistics (we use 1024 shots). Then we run the simulation with different noisy scenarios. The first one is implemented using the Qiskit simulator (backend *FakeVigo*). The second one is a kind of parameterised noise that is included in the Qiskit library as *Pauli error* (in our case we have fixed the value of the parameter p to $p = 0.02$). As expected, the noise produces different expectation values with respect to those obtained by the theoretical algorithm. However, as we can see from the mean squared error values in Table 1, this difference is quite small and has a marginal effect on the accuracy of the classification.

[7] For simplicity, we have considered a two-class data set. However, the algorithm can be applied, without loss of generality, to an arbitrary n-class dataset.

[8] In this example, the weights p_k are set equal to the uniform distribution.

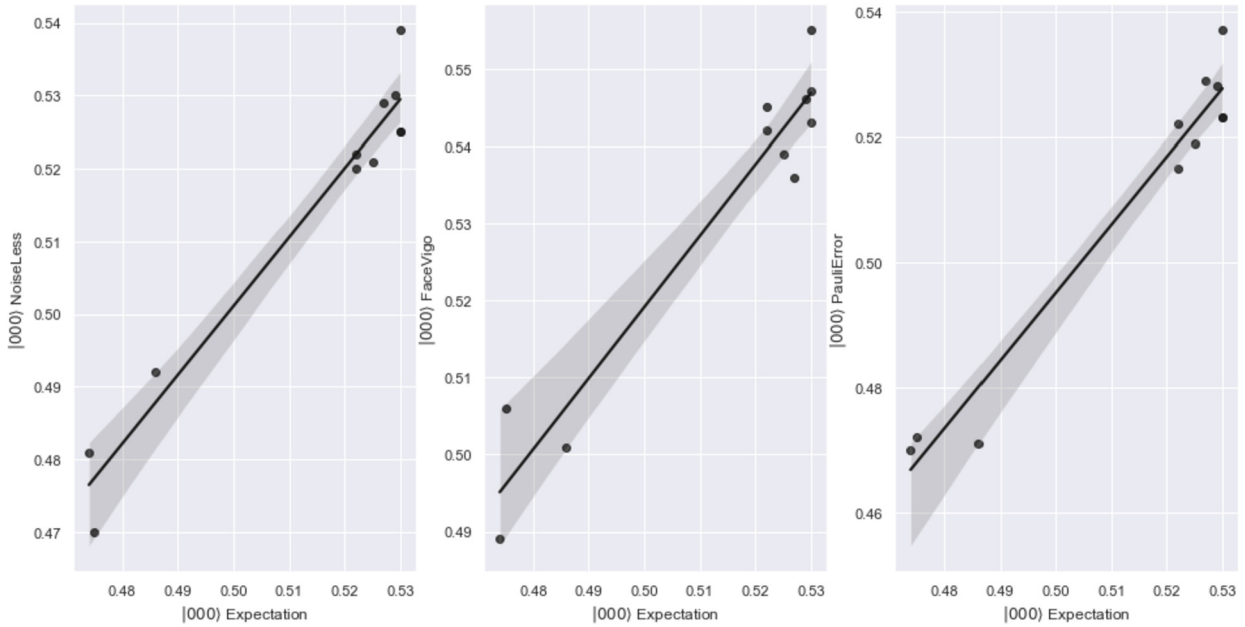

Fig. 2. Here the expectation values calculated on class 1 with the theoretical PGM classifier are plotted against those obtained via IBM-Q simulator.

Table 1
Along the different rows we have the different elements of the test set, divided in the two classes. In the first column we have the expectation value for each element to be classified in the first or the second class, respectively. In the second column we have the noiseless simulation with finite statistics; in the other columns we have the statistics obtained for FakeVigo and Pauli error noises, respectively.

	State	Expectation Value		Noiseless		FakeVigo		Pauli Error	
		$\|000\rangle$	$\|100\rangle$	$\|000\rangle$	$\|100\rangle$	$\|000\rangle$	$\|100\rangle$	$\|000\rangle$	$\|100\rangle$
Class 1	**1**	0.529	0.471	0.530	0.470	0.546	0.454	0.528	0.472
	2	0.522	0.478	0.520	0.480	0.545	0.455	0.515	0.485
	3	0.474	0.526	0.481	0.519	0.489	0.511	0.470	0.530
	4	0.530	0.470	0.525	0.475	0.543	0.457	0.537	0.463
	5	0.527	0.473	0.529	0.471	0.536	0.464	0.529	0.471
	6	0.530	0.470	0.539	0.461	0.547	0.453	0.523	0.477
	7	0.522	0.478	0.522	0.478	0.542	0.458	0.522	0.478
	8	0.486	0.514	0.492	0.508	0.501	0.499	0.471	0.529
	9	0.525	0.475	0.521	0.479	0.539	0.461	0.519	0.481
	10	0.475	0.525	0.470	0.530	0.506	0.494	0.472	0.528
	11	0.530	0.470	0.525	0.475	0.555	0.455	0.523	0.477
Class 2	**1**	0.359	0.641	0.363	0.637	0.415	0.585	0.365	0.635
	2	0.068	0.932	0.071	0.929	0.202	0.798	0.104	0.896
	3	0.377	0.623	0.379	0.621	0.438	0.562	0.392	0.608
	4	0.395	0.605	0.392	0.608	0.435	0.565	0.412	0.588
	5	0.522	0.478	0.526	0.474	0.547	0.453	0.525	0.475
	6	0.522	0.478	0.512	0.488	0.537	0.463	0.521	0.479
Mean Squared Error				0.00002		0.00183		0.00013	
Accuracy		0.706		0.706		0.824		0.706	

6. Conclusion and open problems

In this work, we have addressed the problem of solving multi-class classification problems using quantum-inspired algorithms. We have presented the PGM algorithm, which is based on the PGM quantum state discrimination tech-

nique and can be fully implemented on classical hardware. It represents an improvement in terms of computational efficiency over other variants, such as the Helstrom classifier. The reason is that, unlike the latter, the algorithm presented here is based on an intrinsically multi-class classifier that does not need to rely on computationally expensive strategies such as the *One vs One* or *One vs Rest*. The training part is the most time consuming part of our algorithm. It involves not only the encoding of each piece of classical information, but also the computation of the PGM matrix once the quantum centroids of the classes have been determined. Once these are obtained, one can compare each element of the test set, using the trace as a kind of distance in the quantum state space. The classification function is based on a fuzzy observable associated with a POVM determined by the training set. It is shown that an advantage in classification accuracy can be obtained, especially when considering tensor copies of the quantum representatives.

Thus, the quantum-inspired approach introduced here leads to several potential benefits. First, as discussed in section 4, we obtain an increase (on average) in the accuracy of the classification compared to other standard classical methods. Secondly, by using the PGM classifier, we avoid resorting to the standard One vs One or One vs Rest strategies, which are time-consuming. Finally, we have shown a promising example that opens the door to the possibility of implementing our quantum-inspired classifier in a real quantum computer.

Many questions remain. Perhaps the most important one concerns the choice of the most appropriate quantum encoding for each data set. In [39,41], we have compared the application of some encodings to certain sets of artificial or real datasets, and we have found that the so-called amplitude encoding has a positive effect on the final accuracy of the quantum-inspired classifiers.

We have also explored how to implement the test part of the PGM algorithm in a quantum computer. This task requires a number of further steps. The quantum implementation allows us to highlight some important differences from its classical counterpart. First, it is important to emphasize that the encoding part is challenging. In principle, the quantum encoded version of a piece of classical information assumes that it is possible to prepare an arbitrary quantum state with high accuracy. As reported in the reference [49], this task can be very demanding for state-of-the-art quantum computers, since, in addition to the need to find the necessary gates using classical hardware, one must add the fact that an arbitrary state may involve the use of many elementary gates, creating a scenario in which decoherence plays an important role. A similar consideration applies to the implementation of the PGM circuit, whose unitary matrix (computed according to Neumark's theorem) is computed in a classical computer, as well as its decomposition in terms of native quantum hardware gates. Future research should therefore focus on the problem of finding quantum encoding functions that use fewer resources. This is a difficult task that raises non-trivial questions about which encodings actually lead to a performance advantage. In the quantum version, the averages are also calculated by performing repeated measurements on the same quantum state. Again, there is the problem of preparing a large enough number of copies to obtain a robust statistic for estimating these quantities.

Apart from the above challenges, one may wonder what happens when the number of features is large enough to create a quantum state of high dimension and reach a regime intractable by a classical computer. The same problem arises when tensor copies of the same state are used to improve performance (see the discussion in [19]). In this case, the algorithm could only be implemented in a quantum computer to overcome the large memory problem. Of course, this possibility would require that it is possible to find an encoding mechanism that allows to efficiently generate the quantum representatives of each piece of classical information and the PGM circuits. This is clearly an open problem that we will address in future work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This work is partially supported by the following projects: 1) MIUR, project PRIN 2017: "Theory and applications of resource sensitive logics" (code: 20173WKCM5). 2) MIUR project PRIN 2017: "Logic and Cognition: theory, experiments, applications" (code: 20173YP4N3). 3) FdS projects "Ubiquitous Quantum Reality (UQR): understanding the natural processes under the light of Quantum-like structures" (code: F73C22001360007).

Appendix A. Mathematical properties of the pseudoinverse

Let $\mathbb{C}^n$ be the finite complex Hilbert space of dimension n. $\mathcal{B}(\mathcal{H})^+$ will denote the set of all positive semidefinite bounded linear operator of $\mathbb{C}^n$. The set of all projections of $\mathcal{H}$ will be denoted by $P(\mathcal{H})$. Using $E(\mathcal{H})$ we will denote the set of all effects of $\mathcal{H}$, i.e. the set of all positive operator in $\mathcal{B}(\mathcal{H})^+$ that are bounded by the identity operator $\mathbb{I}$. Thus, an effect is an operator E of $\mathcal{H}$ such that $\mathbb{O} \leq E \leq \mathbb{I}$.

Definition 6.1. Let A be a linear operator of $\mathbb{C}^n$. The pseudoinverse (or Moore-Penrose inverse) of A is an operator X of $\mathbb{C}^n$ such that the following conditions are satisfied:

i) $AXA = A$;
ii) $XAX = X$;
iii) $(AX)^\dagger = AX$, where † is the adjoint operation;
iv) $(XA)^\dagger = XA$.

One can prove that the pseudoinverse of any operator exists and is unique. The pseudoinverse of an operator A will be denoted by $A^\neg$. It turns out that if A is invertible, then the inverse of A (i.e. A^{-1}) coincides with $A^\neg$.

Theorem 2. *Let A be a linear operator of $\mathbb{C}^n$. The following properties hold:*

i) *A is invertible iff $A^{-1} = A^\neg$;*
ii) *$(A^\neg)^\neg = A$;*
iii) *$(A^\dagger)^\neg = (A^\neg)^\dagger$;*
iv) *if $A \in \mathcal{B}(\mathcal{H})^+$, then $A^\neg \in \mathcal{B}(\mathcal{H})^+$;*
v) *$AA^\neg$ and $A^\neg A$ are projections;*
vi) *if $A \in \mathcal{B}(\mathcal{H})^+$, then $AA^\neg = A^\neg A = P_{im(A)}$, where $P_{im(A)}$ is the projection that projects onto the image of A.*

References

[1] S.T. Ali, C. Carmeli, T. Heinosaari, et al., Commutative POVMs and fuzzy observables, Found. Phys. 39 (2009) 593–612.
[2] M.S. Alvarez Alvarado, F.E. Alban Chacón, E.A. Lamilla Rubio, C.D. Rodríguez Gallegos, W. Velásquez, Three novel quantum-inspired swarm optimization algorithms using different bounded potential fields, Sci. Rep. 11 (2021) 11655.
[3] J.M. Arrazola, A. Delgado, B.R. Bardhan, S. Lloyd, Quantum-inspired algorithms in practice, Quantum 4 (2020) 307.
[4] F. Arute, K. Arya, R. Babbush, et al., Quantum supremacy using a programmable superconducting processor, Nature 574 (2019) 505–510.
[5] J. Bae, L. Kwek, Quantum state discrimination and its applications, J. Phys. A 48 (2015) 083001.
[6] S.M. Barnett, S. Croke, Quantum state discrimination, Adv. Opt. Photonics 1 (2009) 238–278.
[7] H. Barnum, E. Knill, Reversing quantum dynamics with near-optimal quantum and classical fidelity, J. Math. Phys. 43 (2002) 2097–2106.
[8] B. Bauer, S. Bravyi, M. Motta, G.K.L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, Chem. Rev. 120 (2020) 12685–12717.
[9] C.H. Bennett, G. Brassard, Quantum cryptography: public key distribution and coin tossing, in: Theoretical Aspects of Quantum Cryptography, Celebrating 30 Years of BB84, Theor. Comput. Sci. 560 (2014) 7–11.
[10] J. Biamonte, P. Wittek, N. Pancotti, et al., Quantum machine learning, Nature 549 (2017) 195–202.
[11] F. Bova, A. Goldfarb, R.G. Melko, EPJ Quantum Technol. 8 (2021) 2.
[12] J. Chen, H.P. Cheng, J. Freericks, A quantum-inspired algorithm for the factorized form of unitary coupled cluster theory, https://arxiv.org/abs/2008.06637, 2020.
[13] N.H. Chia, H.H. Lin, C. Wang, Quantum-inspired sublinear classical algorithms for solving low-rank linear systems, https://arxiv.org/abs/1811.04852, 2018.
[14] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley Interscience, New York, 2000.

[15] A. Friedman, Foundations of Modern Analysis, Dover Books on Mathematics Series, Dover, 1982.
[16] I.M. Gelfand, M.A. Neumark, On the imbedding of normed rings into the ring of operators in Hilbert space, Mat. Sb. 54 (1943) 197–217.
[17] A. Gilyén, S. Lloyd, E. Tang, Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension, https://arxiv.org/abs/1811.04909, 2018.
[18] A. Gilyén, Z. Song, E. Tang, An improved quantum-inspired algorithm for linear regression, Quantum 6 (2022) 754, https://doi.org/10.22331/q-2022-06-30-754.
[19] R. Giuntini, H. Freytes, D.K. Park, C. Blank, F. Holik, K.L. Chow, G. Sergioli, Quantum state discrimination for supervised classification, Appl. Soft Comput. (2023) 134.
[20] R. Giuntini, F. Holik, D.K. Park, H. Freytes, C. Blank, A quantum-inspired algorithm for direct multi-class classification, Appl. Soft Comput. 134 (2023) 109956.
[21] F.A. González, V. Vargas Calderón, H. VinckPosada, Classification with quantum measurements, J. Phys. Soc. Jpn. 90 (2021) 044002.
[22] F.A. González, A. Gallego, S. Toledo Cortés, V. Vargas Calderón, Learning with density matrices and random features, Quantum Mach. Intell. 4 (2021), arXiv:2102.04394.
[23] P. Hausladen, W.K. Wootters, A pretty good measurement for distinguishing quantum states, J. Mod. Opt. 41 (1994) 2385–2390.
[24] C.W. Helstrom, Quantum detection and estimation theory, J. Stat. Phys. 1 (1969) 231–252.
[25] R.A. Horn, C.R. Johnson, Matrix Analysis, 2 ed., Cambridge University Press, 2012.
[26] I. Hull, O. Sattath, E. Diamanti, G. Wendin, Quantum technology for economists, https://doi.org/10.48550/arXiv.2012.04473, 2021.
[27] IBM quantum, https://quantum-computing.ibm.com/, 2021.
[28] R. Iten, O. Reardon Smith, E. Malvetti, L. Mondada, G. Pauvert, E. Redmond, R. Singh Kohli, R. Colbeck, Introduction to UniversalQCompiler, https://doi.org/10.48550/arXiv.1904.01072, 2021.
[29] L.S. Madsen, F. Laudenbach, M.F. Askarani, et al., Quantum computational advantage with a programmable photonic processor, Nature 606 (2022) 75–81.
[30] A. Montanaro, Quantum algorithms: an overview, npj Quantum Inf. 2 (2016) 15023.
[31] O. Montiel, Y. Rubio, C. Olvera, A. Rivera, Quantum-inspired acromyrmex evolutionary algorithm, Sci. Rep. 9 (2019) 12181.
[32] O.H. Montiel Ross, A review of quantum-inspired metaheuristics: going from classical computers to real quantum computers, IEEE Access 8 (2020) 814–838.
[33] S. Mugel, C. Kuchkovsky, E. Sanchez, et al., Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks, Phys. Rev. Res. 4 (2020) 013006.
[34] S. Mugel, E. Lizaso, R. Orus, Use cases of quantum optimization for finance, https://arxiv.org/abs/2010.01312, 2020.
[35] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum 2 (2018) 79.
[36] E. Santucci, G. Sergioli, Classification problem in a quantum framework, https://doi.org/10.1007/978-3-319-74971-6_16, 2017.
[37] M. Schuld, I. Sinayskiy, F. Petruccione, An introduction to quantum machine learning, Contemp. Phys. (2014) 172–185.
[38] M. Schuld, F. Petruccione, Machine Learning with Quantum Computers, Springer Nature ed., 2021.
[39] G. Sergioli, G.M. Bosyk, E. Santucci, R. Giuntini, A quantum-inspired version of the classification problem, Int. J. Theor. Phys. 56 (2017) 3880–3888.
[40] G. Sergioli, E. Santucci, L. Didaci, J.A. Miskczak, R. Giuntini, A quantum-inspired version of the nearest mean classifier, Soft Comput. 22 (2018) 691–705.
[41] G. Sergioli, G. Russo, E. Santucci, A. Stefano, S.E. Torrisi, S. Palmucci, C. Vancheri, R. Giuntini, Quantum-inspired minimum distance classification in a biomedical context, Int. J. Quantum Inf. 16 (2018) 1840011.
[42] G. Sergioli, R. Giuntini, H. Freytes, A new quantum approach to binary classification, PLoS ONE 14 (2019) e0216224.
[43] G. Sergioli, C. Militello, L. Rundo, L. Minafra, F. Torrisi, G. Russo, K. Loon Chow, R. Giuntini, A quantum-inspired classifier for clonogenic assay evaluations, Sci. Rep. 11 (2021) 2830.
[44] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2018.
[45] E. Tang, Quantum-inspired classical algorithms for principal component analysis and supervised clustering, https://arxiv.org/abs/1811.00414, 2018.
[46] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, 2019, pp. 217–228.
[47] UCI machine learning repository, https://archive.ics.uci.edu/ml/index.php.
[48] W. Van Leekwijck, E.E. Kerre, Defuzzification: criteria and classification, Fuzzy Sets Syst. 108 (1999) 159–178.
[49] V. Vargas Calderón, F.A. González, H. Vinck Posada, Optimisation-free density estimation and classification with quantum circuits, Quantum Mach. Intell. 4 (2022) 16.
[50] J. Watrous, The Theory of Quantum Information, Cambridge University Press, 2018.
[51] Y. Wu, W.S. Bao, S. Cao, et al., Strong quantum computational advantage using a superconducting quantum processor, Phys. Rev. Lett. 127 (2021) 180501.
[52] G. Zhang, Quantum-inspired evolutionary algorithms: a survey and empirical study, J. Heuristics 17 (2011) 303–351.
[53] H.S. Zhong, H. Wang, Y.H. Deng, et al., Quantum computational advantage using photons, Science 370 (2020) 1460–1463.
[54] H.S. Zhong, Y.H. Deng, J. Qin, et al., Phase-programmable Gaussian boson sampling using stimulated squeezed light, Phys. Rev. Lett. 127 (2021) 180502.