



Article

Quantum-Inspired Applications for Classification Problems

Cesarino Bertini ^{1,†}  and Roberto Leporini ^{2,*,†} ¹ Department of Management, University of Bergamo, via dei Caniana 2, I-24127 Bergamo, Italy² Department of Economics, University of Bergamo, via dei Caniana 2, I-24127 Bergamo, Italy

* Correspondence: roberto.leporini@unibg.it

† These authors contributed equally to this work.

Abstract: In the context of quantum-inspired machine learning, quantum state discrimination is a useful tool for classification problems. We implement a local approach combining the k -nearest neighbors algorithm with some quantum-inspired classifiers. We compare the performance with respect to well-known classifiers applied to benchmark datasets.

Keywords: quantum-inspired machine learning; classification; local approach

1. Introduction

Quantum-inspired machine learning is a new branch of machine learning based on the application of the mathematical formalism of quantum mechanics to devise novel algorithms. It has revealed how such algorithms have the potential to provide benefits in spite of lacking the computational power of quantum computers with several qubits. Some of these binary classifiers have been analyzed from a geometric perspective [1]. In this work, we implement some algorithms, based on quantum state discrimination, within a local approach in the feature space by taking into account elements close to the element to be classified. In particular, we perform multi-class classification directly (without using binary classifiers) based on Helstrom discrimination following an approach suggested by Blanzieri and Melgani [2], where an unlabeled data instance is classified by finding its k nearest training elements before running a support vector machine (SVM) over the k training elements. This local approach improves the accuracy in classification and motivates the integration with the quantum-inspired Helstrom classifier since the latter can be interpreted as a SVM with linear kernel [3]. It has the potential to offer comparable performance using less complexity because it uses few training points per test point.

The quantum-inspired classifiers require the encoding of the feature vectors into density operators and methods for estimating the distinguishability of quantum states like the Helstrom state discrimination and the pretty-good measurement (PGM). Quantum-inspired machine learning has revealed how relevant benefits for machine learning problems can be obtained using the quantum information theory even without employing quantum computers [4]. Moreover, as we will show below, our PGM within our algorithms is more efficient than the one proposed by these authors in the case of multiple preparations in the same state because it removes duplicates and null values in encoding. Quantum-inspired methods are used in applications that solve industry-relevant problems related to finance, optimization and chemistry [5–9].

In the experimental part, we present a comparison of the performances of the local quantum-inspired classifiers against well-known classical algorithms in order to show that the local approach can be a valuable tool for increasing the performances of this kind of classifier.

In Section 2, we review the notion of quantum encoding of data vectors into density operators and quantum-inspired classification based on quantum state discrimination [10–13]. In Section 3, we use the k -nearest neighbors algorithm (kNN) as a procedure to restrict the training set to the nearest elements around the test elements enabling the local execution of the quantum-inspired classifiers. In Section 4, we present and discuss some empirical results



Citation: Bertini, C.; Leporini, R. Quantum-Inspired Applications for Classification Problems. *Entropy* **2023**, *25*, 404. <https://doi.org/10.3390/e25030404>

Academic Editor: Andreas Wichert

Received: 16 January 2023

Revised: 19 February 2023

Accepted: 22 February 2023

Published: 23 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

for evaluating the impact of locality in quantum-inspired classification comparing the performances of the proposed algorithms to classical methods over benchmark datasets. Furthermore, we compare quantum-inspired classifiers with SVMs within the local approach. In Section 5, there are the concluding remarks about the efficiency of local quantum-inspired classifiers.

2. Quantum-Inspired Classification

The first step of quantum-inspired classification is the *quantum encoding* that is any procedure to encode classical information into quantum states. In particular, we consider encoding of data vectors into density matrices on a Hilbert space H whose dimension depends on the dimension of the input space. *Density matrices* are positive semidefinite operators ρ such that $\text{tr} \rho = 1$ and are the mathematical objects used to describe the physical states of quantum systems. *Pure states* are all the density matrices of the form $\rho = |\psi\rangle\langle\psi|$, with $\|\psi\| = 1$, which are the rank-1 projectors that can be directly identified with unit vectors up to a phase factor. Let ρ be a density operator on a d -dimensional Hilbert space \mathbb{C}^d ; it can be written in the following form:

$$\rho = \frac{1}{d} \left(\mathbb{I}_d + \sqrt{\frac{d(d-1)}{2}} \sum_{j=1}^{d^2-1} b_j^{(\rho)} \sigma_j \right), \quad (1)$$

where $\{\sigma_j\}_{j=1, \dots, d^2-1}$ are the standard generators of the special unitary group $SU(d)$, also called *generalized Pauli matrices*, and \mathbb{I}_d is the $d \times d$ identity matrix. The vector $\mathbf{b}^{(\rho)} = (b_1^{(\rho)}, \dots, b_{d^2-1}^{(\rho)})$, with $b_j^{(\rho)} = \sqrt{\frac{d}{2(d-1)}} \text{tr}(\rho \sigma_j) \in \mathbb{R}$, is the *Bloch vector* associated with ρ which lies within the hypersphere of radius 1 in \mathbb{R}^{d^2-1} . For $d = 2$, the qubit case, the density matrices are in bijective correspondence to the points of the Bloch sphere in \mathbb{R}^3 , where the pure states are in one-to-one correspondence with the points of the spherical surface. For $d > 2$, the points contained in the unit hypersphere of \mathbb{R}^{d^2-1} are not in bijective correspondence with density matrices on \mathbb{C}^d , so the Bloch vectors do not form a ball but a complicated convex body. However, any vector within the sphere of radius $\frac{2}{d}$ gives rise to a density operator [14].

Complex vectors of dimension n can be encoded into density matrices of an $(n+1)$ -dimensional Hilbert space H in the following way:

$$\mathbb{C}^n \ni \mathbf{x} \mapsto |\mathbf{x}\rangle = \frac{1}{\sqrt{\|\mathbf{x}\|^2 + 1}} \left(\sum_{\alpha=0}^{n-1} x_\alpha |\alpha\rangle + |n\rangle \right) \in H, \quad (2)$$

where $\{|\alpha\rangle\}_{\alpha=0, \dots, n}$ is the computational basis of H , identified as the standard basis of \mathbb{C}^{n+1} . The map defined in (2), called *amplitude encoding*, encodes \mathbf{x} into the pure state $\rho_{\mathbf{x}} = |\mathbf{x}\rangle\langle\mathbf{x}|$ where the additional component of $|\mathbf{x}\rangle$ stores the norm of \mathbf{x} . Nevertheless the quantum encoding $\mathbf{x} \mapsto \rho_{\mathbf{x}}$ can be realized in terms of the Bloch vectors $\mathbf{x} \mapsto \mathbf{b}^{(\rho_{\mathbf{x}})}$ saving space resources. The improvement of memory occupation within the Bloch representation is evident when we take multiple tensor products $\rho \otimes \dots \otimes \rho$ of a density matrix ρ constructing a feature map to enlarge the dimension of the representation space [1].

Quantum-inspired classifiers are based on quantum encoding of data vectors into density matrices, calculations of centroids and various criteria of quantum state distinguishability such as: the Helstrom state discrimination, the pretty-good measurement [4,11] and the geometric construction of a minimum-error measurement [12]. Let us briefly recall the notion of quantum state discrimination. Given a set of arbitrary quantum states with respective a priori probabilities $R = \{(\rho_1, p_1), \dots, (\rho_N, p_N)\}$, in general there is no measurement process that discriminates the states without errors, i.e., a collection $E = \{E_i\}_{i=1, \dots, N}$ of positive semidefinite operators such that $\sum_{i=1}^N E_i = \mathbb{I}$, satisfying the following property: $\text{tr}(E_i \rho_j) = 0$ when $i \neq j$ for all $i, j = 1, \dots, N$. The probability of a successful state discrimination of the states in R performing the measurement E is:

$$\mathbb{P}_E(R) = \sum_{i=1}^N p_i \text{tr}(E_i \rho_i). \quad (3)$$

A complete characterization of the optimal measurement E_{opt} that maximizes the probability (3) for $R = \{(\rho_1, p_1), (\rho_2, p_2)\}$ is due to Helstrom [10]. Let $\Lambda := p_1 \rho_1 - p_2 \rho_2$ be the *Helstrom observable* whose positive and negative eigenvalues are, respectively, collected in the sets D_+ and D_- . Consider the two orthogonal projectors:

$$P_{\pm} := \sum_{\lambda \in D_{\pm}} P_{\lambda}, \quad (4)$$

where P_{λ} projects onto the eigenspace of λ . The measurement $E_{opt} := \{P_+, P_-\}$ maximizes the probability (3) that attains the *Helstrom bound*:

$$h_b(\rho_1, \rho_2) = p_1 \text{tr}(P_+ \rho_1) + p_2 \text{tr}(P_- \rho_2). \quad (5)$$

Helstrom quantum state discrimination can be used to implement a quantum-inspired binary classifier with promising performances. Let $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_M, y_M)\}$ be a training set with $\mathbf{x}_i \in \mathbb{C}^n$, $y_i \in \{1, 2\} \forall i = 1, \dots, M$. Assume that, to encode the data points into quantum states by means of $\mathbb{C}^n \ni \mathbf{x} \mapsto \rho_{\mathbf{x}} \in \mathfrak{S}(\mathcal{H})$, one can construct the quantum centroids ρ_1 and ρ_2 of the two classes $C_{1,2} = \{\mathbf{x}_i : y_i = 1, 2\}$:

$$\rho_{1,2} = \frac{1}{|C_{1,2}|} \sum_{\mathbf{x} \in C_{1,2}} \rho_{\mathbf{x}} \quad (6)$$

Let $\{P_+, P_-\}$ be the Helstrom measurement defined by the set $R = \{(\rho_1, p_1), (\rho_2, p_2)\}$, where the probabilities attached to the centroids are $p_{1,2} = \frac{|C_{1,2}|}{|C_1| + |C_2|}$. The *Helstrom classifier* applies the optimal measurement for the discrimination of the two quantum centroids to assign the label y to a new data instance \mathbf{x} , encoded into the state $\rho_{\mathbf{x}}$, as follows:

$$y(\mathbf{x}) = \begin{cases} 1 & \text{if } \text{tr}(P_+ \rho_{\mathbf{x}}) \geq \text{tr}(P_- \rho_{\mathbf{x}}) \\ 2 & \text{otherwise} \end{cases} \quad (7)$$

A strategy to increase the accuracy in classification is given by the construction of the tensor product of q copies of the quantum centroids $\rho_{1,2}^{\otimes q}$ enlarging the Hilbert space where data are encoded. The corresponding Helstrom measurement is $\{P_+^{\otimes q}, P_-^{\otimes q}\}$ and the Helstrom bound satisfies:

$$h_b(\rho_1^{\otimes q}, \rho_2^{\otimes q}) \leq h_b(\rho_1^{\otimes(q+1)}, \rho_2^{\otimes(q+1)}) \quad \forall q \in \mathbb{N}. \quad (8)$$

Increasing the dimension of the Hilbert space of the quantum encoding, one increases the Helstrom bound obtaining a more accurate classifier. The corresponding computational cost is evident; however, in the case of real input vectors, the space can be enlarged saving time and space by means of encoding into Bloch vectors.

Clearly, defining a quantum encoding is equivalent to selecting a feature map to represent feature vectors into a space of higher dimension. In the case of the considered quantum amplitude encoding $\mathbb{R}^2 \ni (x_1, x_2) \mapsto \rho_{(x_1, x_2)} \in \mathfrak{S}(\mathbb{C}^3)$, the nonlinear explicit injective function $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^5$ to encode data into Bloch vectors can be defined as follows:

$$\varphi(x_1, x_2) := \frac{1}{x_1^2 + x_2^2 + 1} \left(2x_1x_2, 2x_1, 2x_2, x_1^2 - x_2^2, \frac{x_1^2 + x_2^2 - 2}{\sqrt{3}} \right). \quad (9)$$

The mapped feature vectors are points on the surface of a hyper-hemisphere, with centroids of the classes, calculated as the means of these feature vectors, inside the hypersphere and can be rescaled to a Bloch vector as shown below.

In order to make the classification more accurate, one can increase the dimension of the representation space providing k copies of the quantum states, in terms of a tensor product,

encoding data instances and centroids into density matrices $\rho^{\otimes q}$. Bloch encoding allows an efficient implementation of feature maps; by removing null and repeated entries from the Bloch vector we obtain the following injective function for data encoding. Therefore, the Bloch representation allows an efficient storing of redundant elements of density matrices $\rho^{\otimes q}$.

Let us consider a training set divided into the classes C_1, \dots, C_M ; assume we have any training point \mathbf{x} encoded into the Bloch vector $\mathbf{b}^{(\mathbf{x})}$ of a pure state on \mathbb{C}^d . The calculation of the centroid of the class C_i , within this quantum encoding, must take into account that the mean of the Bloch vectors $\mathbf{b}^{(i)} := \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{b}^{(\mathbf{x})}$ does not represent a density operator in general. In fact, for $d > 2$ the points contained in the unit hypersphere of \mathbb{R}^{d^2-1} are not in bijective correspondence with density matrices on \mathbb{C}^d . However, since any vector within the closed ball of radius $\frac{2}{d}$ gives rise to a density operator, a centroid can be defined in terms of a meaningful Bloch vector by a rescaling:

$$\hat{\mathbf{b}}^{(i)} := \frac{2}{d|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{b}^{(\mathbf{x})}. \quad (10)$$

A method of quantum state discrimination for distinguishing more than two states $\{(\rho_1, p_1), \dots, (\rho_N, p_N)\}$ is the square-root measurement, also known as the *pretty-good measurement*, defined by:

$$E_i = p_i \rho^{-\frac{1}{2}} \rho_i \rho^{-\frac{1}{2}}, \quad (11)$$

where $\rho = \sum_i p_i \rho_i$; PGM is the optimal minimum error when states satisfy certain symmetry properties [11]. Clearly, to distinguish between n centroids we need a measurement with at most n outcomes. It is sometimes optimal to avoid measurement and simply guess that the state is the a priori most likely state.

The optimal POVM $\{E_i\}_i$ for minimum-error state discrimination over

$$R = \{(\rho_1, p_1), \dots, (\rho_N, p_N)\}$$

satisfies the following necessary and sufficient Helstrom conditions [12]:

$$\Gamma - p_i \rho_i \geq 0 \quad \forall i = 1, \dots, N, \quad (12)$$

where the Hermitian operator, also known as the *Lagrange operator*, is defined by $\Gamma := \sum_i p_i \rho_i E_i$. It is also useful to consider the following properties which can be obtained from the above conditions:

$$E_j(p_j \rho_j - p_i \rho_i) E_i = 0 \quad \forall i, j. \quad (13)$$

For each i the operator $\Gamma - p_i \rho_i$ can have two, one or no zero eigenvalues, corresponding to the zero operator, a rank-one operator, and a positive-definite operator, respectively. In the first case, we use the measurement $\{E_i = \mathbb{I}, E_{i \neq j} = 0\}$ for some i where $p_i \geq p_j \forall j$, i.e., the state belongs to the a priori most likely class. In the second case, if $E_i \neq 0$, it is a weighted projector onto the corresponding eigenstate. In the latter case, it follows that $E_i = 0$ for every optimal measurement.

Given the following Bloch representations:

$$\Gamma = \frac{1}{d} \left(a \mathbb{I}_d + \sqrt{\frac{d(d-1)}{2}} \sum_{j=1}^{d^2-1} b_j \sigma_j \right), \quad \rho_i = \frac{1}{d} \left(\mathbb{I}_d + \sqrt{\frac{d(d-1)}{2}} \sum_{j=1}^{d^2-1} b_j^{(i)} \sigma_j \right), \quad (14)$$

in order to determine the Lagrange operator in \mathbb{C}^d we need d^2 independent linear constraints:

$$2p_i \left(a - \hat{\mathbf{b}}^{(i)} \cdot \mathbf{b} - \frac{p_i}{2} (1 - |\hat{\mathbf{b}}^{(i)}|^2) \right) = a^2 - |\mathbf{b}|^2. \quad (15)$$

A measurement with more than d^2 outcomes can always be decomposed as a probabilistic mixture of measurements with at most d^2 outcomes. Therefore, if the number of classes is greater than or equal to d^2 and we get d^2 linearly independent equations, we construct the Lagrange operator and derive the optimal measurements. From the geometric point of view, we obtain the unit vectors corresponding to the rank-1 projectors $E_i = \frac{1}{d} \left(\mathbb{I}_d + \sqrt{\frac{d(d-1)}{2}} \sum_{j=1}^{d^2-1} n_j^{(i)} \sigma_j \right)$ where $\mathbf{n}^{(i)} = \frac{\hat{\mathbf{b}}^{(i)} - a\mathbf{b}}{|\hat{\mathbf{b}}^{(i)} - a\mathbf{b}|} \in \mathbb{R}^{d^2-1}$ giving the POVM of the measurement. It is also possible to further partition the classes in order to increase the number of centroids and of the corresponding equations. The classification is carried out in this way: an unlabeled point $\hat{\mathbf{x}}$ is associated with the first label y such that $\mathbf{b}^{(\hat{\mathbf{x}})} \cdot \mathbf{n}^{(y)} = \max_i \mathbf{b}^{(\hat{\mathbf{x}})} \cdot \mathbf{n}^{(i)}$, where $d = \lceil \sqrt{\text{length}(\mathbf{x}) + 2} \rceil$.

3. Local Quantum-Inspired Classifiers

In the implementation, we consider the execution of the classifiers described above after a selection of the k training elements that are closest to a considered unclassified instance.

The k -nearest neighbors algorithm (kNN) is a simple classification algorithm which consists of the following steps:

1. The computation of the chosen distance metric between the test element and the training elements;
2. The extraction of the k elements closest to the test instance;
3. The assignment of the class label through a majority voting based on the labels of the k nearest neighbors.

In the following, we apply the kNN for the extraction of the closest elements to the test element then the classification is performed by a quantum-inspired algorithm instead of majority voting. On the one hand, given a test element, the kNN can be executed over the data vectors in the input space, e.g., considering the Euclidean distance, then the k neighbors can be encoded into density matrices and used for a quantum-inspired classification. On the other hand, the entire dataset can be encoded into density matrices and the kNN selects the k neighbors evaluating an operator distance among quantum states. In the latter case, we consider the *Bures distance* that is a quantum generalization of the Fisher information and a distance derived by the super-fidelity. The Bures distance is defined by:

$$d_B(\rho_1, \rho_2) = \sqrt{2 \left(1 - \sqrt{\mathcal{F}(\rho_1, \rho_2)} \right)}, \quad (16)$$

where the fidelity between density operators is given by $\mathcal{F}(\rho_1, \rho_2) = (\text{tr} \sqrt{\sqrt{\rho_1} \rho_2 \sqrt{\rho_1}})^2$. Let us note that the fidelity reduces to $\mathcal{F}(\rho_1, \rho_2) = \langle \psi_1 | \rho_2 | \psi_1 \rangle$ when $\rho_1 = |\psi_1\rangle\langle\psi_1|$. Therefore the Bures distance between the pure state ρ_1 and the arbitrary state ρ_2 can be expressed in term of the Bloch representation as follows:

$$d_B(\rho_1, \rho_2) = \sqrt{2 \left(1 - \sqrt{\frac{1}{d} \left(1 + (d-1) \mathbf{b}^{(1)} \cdot \mathbf{b}^{(2)} \right)} \right)} \equiv D_B(\mathbf{b}^{(1)}, \mathbf{b}^{(2)}), \quad (17)$$

where $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ are the Bloch vectors of ρ_1 and ρ_2 , respectively, and d is the dimension of the Hilbert space of the quantum encoding. The special form (17) of the Bures distance, expressed in terms of Bloch vectors, is relevant for our purpose because data vectors can be encoded into pure states and in general quantum centroids are mixed states.

An alternative distance can be defined via super-fidelity [15]

$$d_G(\rho_1, \rho_2) = \sqrt{1 - \mathcal{G}(\rho_1, \rho_2)}, \quad (18)$$

where the super-fidelity between density operators is given by

$$\mathcal{G}(\rho_1, \rho_2) = \text{tr} \rho_1 \rho_2 + \sqrt{(1 - \text{tr} \rho_1^2)(1 - \text{tr} \rho_2^2)}.$$

Notice that the super-fidelity reduces to $\mathcal{G}(\rho_1, \rho_2) = \langle \psi_1 | \rho_2 | \psi_1 \rangle$ when $\rho_1 = |\psi_1\rangle\langle\psi_1|$. This distance can be expressed in term of the Bloch representation as follows:

$$D_G(\mathbf{b}^{(1)}, \mathbf{b}^{(2)}) = \sqrt{1 - \frac{1}{d} \left(1 + (d-1)(\mathbf{b}^{(1)} \cdot \mathbf{b}^{(2)}) + \sqrt{(1 - |\mathbf{b}^{(1)}|^2)(1 - |\mathbf{b}^{(2)}|^2)} \right)}, \quad (19)$$

where $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ are the Bloch vectors of ρ_1 and ρ_2 , respectively, and d is the dimension of the Hilbert space of the quantum encoding. The inner distance between the corresponding Bloch vectors represents the angle θ between the unit vectors $(\mathbf{b}^{(1)}, \sqrt{1 - |\mathbf{b}^{(1)}|^2})$ and $(\mathbf{b}^{(2)}, \sqrt{1 - |\mathbf{b}^{(2)}|^2})$, which is normalized to be 1:

$$\hat{D}_G(\mathbf{b}^{(1)}, \mathbf{b}^{(2)}) = \frac{\arccos \left(\mathbf{b}^{(1)} \cdot \mathbf{b}^{(2)} + \sqrt{(1 - |\mathbf{b}^{(1)}|^2)(1 - |\mathbf{b}^{(2)}|^2)} \right)}{\pi}. \quad (20)$$

For pure states the inner distance corresponds to the Fubini-Study distance.

In Algorithm 1, the locality is imposed by running the kNN on the input space finding the training vectors that are closest to the test element; then there is the quantum encoding into pure states and a quantum-inspired classifier (Helstrom, PGM or geometric Helstrom) is locally executed over the restricted training set. In Algorithm 2, the test element and all the training elements are encoded into Bloch vectors of pure states then a kNN is run w.r.t. the Bures distance to find the nearest neighbors in the space of the quantum representation; then a quantum-inspired classifier is executed with the training instances corresponding to the closest quantum states.

Algorithm 1 Local quantum-inspired classification based on kNN in the input space before the quantum encoding. The distance can be: Euclidean, Manhattan, Chessboard, Canberra or Bray–Curtis.

Require: Dataset X of labeled instances, unlabeled point \hat{x}

Ensure: Label of \hat{x}

find the k nearest neighbors x_1, \dots, x_k to \hat{x} in X w.r.t. the Euclidean distance

encode \hat{x} into a pure state $\rho_{\hat{x}}$

for $j = 1, \dots, k$ **do**

 encode x_j into a pure state ρ_{x_j}

end for

run the quantum-inspired classifier with training points encoded into $\{\rho_{x_j}\}_{j=1, \dots, k}$.

Algorithm 2 Local quantum-inspired classification based on kNN in the Bloch representation after quantum encoding. The distance can be: Bures, Super-Fidelity or Inner.

Require: Dataset X of labeled instances, unlabeled point \hat{x}

Ensure: Label of \hat{x}

encode \hat{x} into a Bloch vector $\mathbf{b}^{(\hat{x})}$ of a pure state

for $x \in X$ **do**

 encode x into a Bloch vector $\mathbf{b}^{(x)}$ of a pure state

end for

find the k nearest neighbors to $\mathbf{b}^{(\hat{x})}$ in $\{\mathbf{b}^{(x)}\}_{x \in X}$ w.r.t. the distance D_B

run the quantum-inspired classifier over the k nearest neighbors.

A local quantum-inspired classifier can be defined without quantum state discrimination but considering a *nearest mean classification* such as the following: after the quantum

encoding we perform a kNN selection and calculate the centroid of each class considering only the nearest neighbors to the test element, finally we assign the label according to the nearest centroid as schematized in Algorithm 3.

Algorithm 3 Local quantum-inspired nearest mean classifier.

Require: Training set X divided into n classes C_i , unlabeled point \hat{x}

Ensure: Label of \hat{x}

 encode \hat{x} into a Bloch vector $\mathbf{b}^{(\hat{x})}$ of a pure state

for $\mathbf{x} \in X$ **do**

 encode \mathbf{x} into a Bloch vector $\mathbf{b}^{(\mathbf{x})}$ of a pure state

end for

 find the neighborhood $K = \{\mathbf{b}^{(\mathbf{x}_1)}, \dots, \mathbf{b}^{(\mathbf{x}_k)}\}$ of $\mathbf{b}^{(\hat{x})}$ w.r.t. the distance D_B

for $i = 1, \dots, n$ **do**

 construct the centroid $\hat{\mathbf{b}}^{(i)} = \frac{2}{d|C_i^k|} \sum_{\mathbf{x} \in C_i^k} \mathbf{b}^{(\mathbf{x})}$ where $C_i^k := \{\mathbf{x} \in C_i : \mathbf{b}^{(\mathbf{x})} \in K\}$

end for

 find the closest centroid $\hat{\mathbf{b}}^{(l)}$ to $\frac{2}{d} \mathbf{b}^{(\hat{x})}$ w.r.t. the distance D_B

return label of the class C_l

4. Results and Discussion

In this section, we present some numerical results obtained by the implementation of the local quantum-inspired classifiers with several distances compared to well-known classical algorithms. In particular, we consider the SVM with different kernels: linear, radial basis function and sigmoid. Then, we run a random forest, a naive Bayes classifier and the logistic regression. In order to compare the results with previous papers, we take into account the following benchmark datasets from the PMLB public repository [16]: analcatdata_aids, analcatdata_asbestos, analcatdata_bankruptcy, analcatdata_boxing1, analcatdata_cyyoung9302, analcatdata_dmft, analcatdata_happiness, analcatdata_japansolvent, analcatdata_lawsuit, appendicitis, biomed, breast_cancer, iris, labor, new_thyroid, phoneme, prnn_fglass, prnn_synth, tae and wine_recognition. For each dataset we randomly select 80% of the data to create a training set and use the residual 20% for the evaluation. We repeated the same procedure 10 times and calculated the average accuracy in Table 1. Certainly, it is possible to compare the performances based on different statistic indices including Matthews correlation coefficient, F-measure and Cohen's parameter.

We observe that the performances of the local quantum-inspired classifiers turn out to be definitely more accurate, where the hyperparameter k is set equal to the number of classes in the dataset. This value is reasonable to construct the centroids of the classes. In particular, Algorithm 1 with the Euclidean distance is the most accurate classifier for the datasets analcatdata_boxing1, analcatdata_happiness, biomed, prnn_fglass and wine_recognition, while the Manhattan distance is best for analcatdata_aids, analcatdata_japansolvent, breast_cancer, iris and tae, the Chessboard distance is best for analcatdata_cyyoung9302 and analcatdata_lawsuit, and the Bray–Curtis distance is best for analcatdata_bankruptcy and appendicitis. Algorithm 2 with the Bures distance outperforms Algorithm 1 and 3 for analcatdata_dmft and produces the same accuracy for labor. Algorithm 3 with the Bures distance is the most accurate classifier for analcatdata_asbestos, new_thyroid, phoneme and prnn_synth. Algorithm 1 uses a k-d tree in the training set, while the other two use a k-d tree in the corresponding Bloch vector space. The time complexity to construct the k-d tree is usually $O(dn \log n)$, where n is the cardinality of the training set and d the length of each vector, while the space complexity is $O(dn)$. The query to find the k nearest neighbors takes $O(k \log n)$. The time complexity of PGM is $O(cd^3)$ and is $O(dm)$ for the classification of the m elements of the test set in c classes. Our algorithm is more efficient than the one presented in [4] in the presence of multiple copies because it remove nulls and duplicates. In particular, we consider only 20 values instead of 81 matrix elements of $\rho_{(x_1, x_2)} \otimes \rho_{(x_1, x_2)}$, 51 values instead of 729 for $\rho_{(x_1, x_2)} \otimes \rho_{(x_1, x_2)} \otimes \rho_{(x_1, x_2)}$ and so on. In a future paper, we will analyze in detail the complexity of such algorithms in the

average case and in the worst case. For instance, one can construct the ball tree for clustered data instead of the k-d tree and consider different search techniques.

In Table 2, we show the methods that provided the best accuracy, with the respective execution times, compared with the classical method. These experimental results are promising and show that the methods are efficient when run on classical computers. Algorithm 3 with the Bures distance is not efficient for phoneme, but Algorithm 1 with the Euclidean distance is: 1.951 s with an average accuracy of 0.897. We will study in a future work how to also apply the local methods in implementations on quantum computers.

Let us focus on multi-class datasets for the comparison with the kNNSVM method proposed by Blanzieri and Melgani [2]. This method requires the choice of the hyperparameter k , and as is well known from the standard kNN algorithm, there is no general strategy to choose k a priori. In Table 3, the results obtained for some k values of the kNNSVM are shown. For *analcattdata_dmft*, kNNSVM presents an average accuracy that is only 2% lower than Algorithm 2 but requires 17 elements per test element instead of 6. For *analcattdata_happiness*, kNNSVM yields an average accuracy that is 10% lower than Algorithm 1 and requires 14 elements per test element instead of 3. However, kNNSVM outperforms local quantum-inspired classifiers for *iris* and *tae*, but only for the latter requires fewer elements, while for *wine_recognition* they are comparable. For *new_thyroid* and *prnn_fglass*, the best results are obtained with the nearest neighbor method, but with lower accuracy than Algorithms 1 and 3, respectively.

Table 1. Classification comparison, in terms of test average accuracy for 10 runs, for benchmark datasets.

Dataset/Method	Euclidean	Manhattan	Chessboard	Canberra	Bray–Curtis	Bures	Nearest Mean
analcattdata_aids	0.6	0.62	0.53	0.55	0.51	0.46	0.46
analcattdata_asbestos	0.794	0.794	0.794	0.782	0.794	0.794	0.8
analcattdata_bankruptcy	0.87	0.87	0.83	0.85	0.88	0.87	0.87
analcattdata_boxing1	0.725	0.721	0.696	0.688	0.721	0.688	0.646
analcattdata_cyyoung9302	0.811	0.794	0.844	0.789	0.811	0.817	0.833
analcattdata_dmft	0.201	0.199	0.198	0.191	0.202	0.209	0.18
analcattdata_happiness	0.475	0.458	0.383	0.433	0.475	0.45	0.325
analcattdata_japansolvent	0.79	0.82	0.8	0.74	0.81	0.77	0.74
analcattdata_lawsuit	0.979	0.974	0.981	0.968	0.974	0.972	0.976
appendicitis	0.866	0.811	0.867	0.876	0.886	0.838	0.857
biomed	0.917	0.91	0.9	0.898	0.91	0.91	0.91
breast_cancer	0.698	0.707	0.696	0.674	0.695	0.691	0.704
iris	0.943	0.957	0.95	0.947	0.947	0.943	0.937
labor	0.927	0.927	0.836	0.909	0.936	0.936	0.936
new_thyroid	0.974	0.981	0.96	0.956	0.977	0.97	0.984
phoneme	0.897	0.897	0.895	0.893	0.897	0.898	0.903
prnn_fglass	0.746	0.693	0.7	0.685	0.671	0.72	0.724
prnn_synth	0.874	0.864	0.896	0.856	0.878	0.894	0.9
tae	0.58	0.583	0.573	0.543	0.577	0.58	0.56
wine_recognition	0.989	0.983	0.975	0.967	0.972	0.981	0.981
Dataset/Method	Linear	Radial Basis Function	Polynomial	Sigmoid	Random Forest	Naive Bayes	Logistic Regression
analcattdata_aids	0.45	0.45	0.49	0.45	0.42	0.49	0.56
analcattdata_asbestos	0.7	0.724	0.694	0.735	0.694	0.735	0.724
analcattdata_bankruptcy	0.79	0.79	0.81	0.7	0.78	0.77	0.81
analcattdata_boxing1	0.621	0.654	0.65	0.629	0.713	0.658	0.654
analcattdata_cyyoung9302	0.778	0.756	0.767	0.7	0.778	0.75	0.817
analcattdata_dmft	0.175	0.175	0.17	0.168	0.18	0.182	0.191
analcattdata_happiness	0.35	0.3	0.325	0.283	0.358	0.35	0.442
analcattdata_japansolvent	0.77	0.77	0.76	0.72	0.75	0.8	0.75
analcattdata_lawsuit	0.955	0.953	0.955	0.904	0.955	0.945	0.964
appendicitis	0.852	0.857	0.842	0.743	0.867	0.848	0.867
biomed	0.857	0.857	0.864	0.838	0.857	0.881	0.857
breast_cancer	0.656	0.656	0.668	0.651	0.675	0.672	0.66
iris	0.94	0.927	0.93	0.87	0.933	0.923	0.937
labor	0.782	0.854	0.827	0.636	0.827	0.855	0.845
new_thyroid	0.963	0.958	0.947	0.958	0.949	0.947	0.96
phoneme	0.75	0.821	0.829	0.746	0.883	0.766	0.745
prnn_fglass	0.541	0.51	0.559	0.483	0.666	0.629	0.568
prnn_synth	0.85	0.85	0.844	0.848	0.846	0.844	0.86
tae	0.446	0.403	0.43	0.273	0.45	0.477	0.457
wine_recognition	0.978	0.978	0.978	0.947	0.975	0.975	0.983

Table 2. Classification comparison, in terms of test average accuracy for 10 runs, for benchmark datasets. For quantum-inspired methods we show the average execution time in seconds on an Intel i7-9750H CPU @ 2.60GHz 6-core with 32 GB RAM, and for classical methods the training time and evaluation time per test set element.

Dataset	Method	Accuracy	Execution time	Method	Accuracy	Training time	Evaluation Time
analcdata_aid	Manhattan	0.62	0.013	LogisticRegression	0.56	0.942	0.0018
analcdata_asbestos	NearestMean	0.8	0.018	Sigmoid	0.735	1.244	0.0015
analcdata_bankruptcy	BrayCurtis	0.88	0.019	Polynomial	0.81	0.893	0.0015
analcdata_boxing1	Euclidean	0.725	0.021	RandomForest	0.713	0.351	0.0054
analcdata_cyyoung9302	Chessboard	0.844	0.074	LogisticRegression	0.817	1.201	0.0014
analcdata_dmft	Bures	0.209	0.013	LogisticRegression	0.191	3.103	0.0018
analcdata_happiness	Euclidean	0.475	0.021	LogisticRegression	0.442	1.416	0.0024
analcdata_japansolvent	Manhattan	0.82	0.06	NaiveBayes	0.8	0.636	0.0045
analcdata_lawsuit	Chessboard	0.981	0.074	LogisticRegression	0.964	2.243	0.0022
appendicitis	BrayCurtis	0.886	0.094	RandomForest	0.867	0.657	0.0087
biomed	Euclidean	0.917	0.194	NaiveBayes	0.881	0.682	0.0045
breast_cancer	Manhattan	0.707	0.339	RandomForest	0.675	0.832	0.009
iris	Manhattan	0.957	0.029	Linear	0.94	0.904	0.0015
labor	BrayCurtis	0.936	0.013	NaiveBayes	0.855	0.318	0.0038
new_thyroid	NearestMean	0.984	0.099	Linear	0.963	1.452	0.0015
phoneme	NearestMean	0.903	39.204	RandomForest	0.883	2.157	0.0093
prnn_fglass	Euclidean	0.746	0.378	RandomForest	0.666	1.646	0.01
prnn_synth	NearestMean	0.9	0.12	LogisticRegression	0.86	2.323	0.0024
tae	Manhattan	0.583	0.148	NaiveBayes	0.477	3.801	0.0065
wine_recognition	Euclidean	0.989	0.394	LogisticRegression	0.983	2.426	0.003

Table 3. kNNSVM comparison, in terms of test average accuracy for 10 runs, for benchmark multi-class datasets.

Dataset/Method	1NNSVM	2NNSVM	3NNSVM	4NNSVM	5NNSVM	6NNSVM	7NNSVM	8NNSVM	9NNSVM
analcdata_dmft	0.186	0.198	0.198	0.15	0.177	0.187	0.187	0.187	0.196
analcdata_happiness	0.242	0.317	0.325	0.375	0.383	0.383	0.375	0.375	0.35
iris	0.937	0.943	0.943	0.95	0.943	0.957	0.937	0.95	0.96
new_thyroid	0.974	0.965	0.958	0.96	0.958	0.965	0.949	0.951	0.96
prnn_fglass	0.72	0.71	0.658	0.678	0.683	0.678	0.666	0.673	0.676
tae	0.57	0.593	0.557	0.55	0.553	0.543	0.53	0.53	0.493
wine_recognition	0.986	0.981	0.986	0.989	0.989	0.989	0.983	0.983	0.986
Dataset/Method	10NNSVM	11NNSVM	12NNSVM	13NNSVM	14NNSVM	15NNSVM	16NNSVM	17NNSVM	18NNSVM
analcdata_dmft	0.197	0.196	0.2	0.202	0.193	0.185	0.197	0.204	0.204
analcdata_happiness	0.358	0.4	0.317	0.408	0.433	0.425	0.35	0.375	0.4
iris	0.95	0.95	0.95	0.957	0.95	0.953	0.943	0.937	0.943
new_thyroid	0.951	0.937	0.963	0.944	0.944	0.949	0.96	0.953	0.955
prnn_fglass	0.676	0.641	0.685	0.659	0.663	0.654	0.693	0.678	0.685
tae	0.49	0.433	0.52	0.48	0.43	0.427	0.48	0.467	0.487
wine_recognition	0.983	0.972	0.983	0.972	0.964	0.967	0.989	0.969	0.975

5. Conclusions

The present paper focuses on the implementation of classification algorithms based on quantum state discrimination. A novel contribution is the local approach adopted to execute the classifier, not over the entire training set, but in a neighborhood of the test element. Once partitioned, for the training set the k nearest data elements are encoded into Bloch vectors and used to define the quantum centroid of each class.

The local quantum-inspired classifier considered, for reasonable values of the hyperparameters, was found to be a method with performance comparable to classical algorithms for multi-class classification. We performed some experiments using benchmark datasets and found that local quantum-inspired classifiers were even more accurate than SVM with different kernels, a random forest, a naive Bayes classifier and the logistic regression classification algorithm.

The present proposal offers a family of classifiers. In fact, several strategies to impose a notion of locality over a training set, and several procedures of quantum state discrimination, can be applied. Both the local approach to classification and the quantum-inspired data encoding/processing deserve further investigation to clarify the impact of these ideas on machine learning, but the results achieved clearly indicate that both approaches to machine learning are promising.

Author Contributions: Conceptualization, R.L.; methodology, C.B.; software, R.L.; validation, C.B.; formal analysis, R.L.; writing—original draft preparation, R.L.; writing—review and editing, R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code is also available at the following repository: <https://github.com/leporini/classification> (accessed on 21 February 2023). Penn Machine Learning Benchmarks <https://github.com/EpistasisLab/pmlb> (accessed on 21 February 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SVM	Support vector machine
PGM	Pretty good measurement
NN	Nearest neighbors

References

1. Leporini, R.; Pastorello, D. An efficient geometric approach to quantum-inspired classifications. *Sci. Rep.* **2022**, *12*, 8781. [CrossRef] [PubMed]
2. Blanzieri, E.; Melgani, F. An Adaptive SVM Nearest Neighbor Classifier for Remotely Sensed Imagery. In Proceedings of the IEEE International Conference on Geoscience and Remote Sensing Symposium, Denver, CO, USA, 31 July–4 August 2006; pp. 3931–3934.
3. Leporini, R.; Pastorello, D. Support Vector Machines with Quantum State Discrimination. *Quantum Rep.* **2021**, *3*, 482–499. [CrossRef]
4. Giuntini, R.; Holik, F.; Park, D.K.; Freytes, H.; Blank, C.; Sergioli, G. Quantum-inspired algorithm for direct multi-class classification. *Appl. Soft Comput.* **2023**, *134*, 109956. [CrossRef]
5. Montiel, O.; Rubio, Y.; Olvera, C.; Rivera, A. Quantum inspired acromyrmex evolutionary algorithm. *Sci. Rep.* **2019**, *9*, 12181. [CrossRef]
6. Mugel, S.; Kuchkovsky, C.; Sánchez, E.; Fernández-Lorenzo, S.; Luis-Hita, J.; Lizaso, E.; Orús, R. Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks. *Phys. Rev. Res.* **2022**, *4*, 013006. [CrossRef]
7. Mugel, Lizaso, E.; Orús, R. Use cases of quantum optimization for finance. In *Credibile Asset Allocation, Optimal Transport Methods, and Related Topics*; Sriboonchitta, S., Kreinovich, V., Yamaka, W., Eds.; Springer International Publishing: Cham, Switzerland, 2022; pp. 211–220.
8. Chen, J.; Cheng, H.; Freericks, J.K. Quantum-inspired algorithm for the factorized form of unitary coupled cluster theory. *J. Chem. Theory Comput.* **2021**, *17*, 841–847. [CrossRef] [PubMed]
9. Alvarez-Alvarado, M.S.; Alban-Chacón, F.E.; Lamilla-Rubio, E.A.; Rodríguez-Gallegos, C.D.; Velásquez, W. Three novel quantum-inspired swarm optimization algorithms using different bounded potential fields. *Sci. Rep.* **2021**, *11*, 11655. [CrossRef] [PubMed]
10. Helstrom, C.W. Quantum detection and estimation theory. *J. Stat. Phys.* **1969**, *1*, 231–252. [CrossRef]
11. Mochon, C. Family of generalized pretty good measurements and the minimal-error pure-state discrimination problems for which they are optimal. *Phys. Rev. A* **2006**, *73*, 032328. [CrossRef]
12. Bae, J. Structure of minimum-error quantum state discrimination. *New J. Phys.* **2013**, *15*, 073037. [CrossRef]
13. Bae, J.; Kwek, L.C. Quantum state discrimination and its applications. *J. Phys. A* **2015**, *48*, 083001. [CrossRef]
14. Kimura, G.; Kossakowski, A. The Bloch-Vector Space for N-Level Systems: The Spherical-Coordinate Point of View. *Open Syst. Inf. Dyn.* **2005**, *12*, 207–229. [CrossRef]
15. Miszczak, J.A.; Puchała, Z.; Horodecki, P.; Uhlmann, A.; Życzkowski, K. Sub- and super-fidelity as bounds for quantum fidelity. *Quantum Inf. Comput.* **2009**, *9*, 103–130. [CrossRef]
16. Romano, J.D.; Le, T.T.; La Cava, W.; Gregg, J.T.; Goldberg, D.J.; Chakraborty, P.; Ray, N.L.; Himmelstein, D.; Fu, W.; Moore, J.H. PMLB v1.0: An open source dataset collection for benchmarking machine learning methods. *arXiv* **2020**, arXiv:2012.00058.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.