Peer-reviewed version available at Entropy 2017, 19, 659; doi:10.3390/e19120659

Article

Quantum Minimum Distance Classifier

Enrica Santucci

[1] University of Cagliari, Piazza D'Armi snc - 09123 Cagliari (Italy); enrica.santucci@gmail.com

Abstract: We propose a quantum version of the well known minimum distance classification model called *Nearest Mean Classifier* (NMC). In this regard, we presented our first results in two previous works. In [34] a quantum counterpart of the NMC for two-dimensional problems was introduced, named *Quantum Nearest Mean Classifier* (QNMC), together with a possible generalization to arbitrary dimensions. In [33] we studied the n-dimensional problem into detail and we showed a new encoding for arbitrary n-feature vectors into density operators. In the present paper, another promising encoding of n-dimensional patterns into density operators is considered, suggested by recent debates on quantum machine learning. Further, we observe a significant property concerning the non-invariance by feature rescaling of our quantum classifier. This fact, which represents a meaningful difference between the NMC and the respective quantum version, allows to introduce a free parameter whose variation provides, in some cases, better classification results for the QNMC. The experimental section is devoted to: *i)* compare the NMC and QNMC performance on different datasets; *ii)* study the effects of the non-invariance under uniform rescaling for the QNMC.

Keywords: quantum formalism applications; minimum distance classification; rescaling parameter

1. Introduction

In the last few years, many efforts to apply the quantum formalism to non-microscopic contexts have been made [1,24,26,32,37,41]. The idea is that the powerful predictive properties of quantum mechanics, used for describing the behavior of microscopic phenomena, turn out to be particularly beneficial also in non-microscopic domains. Indeed, the real power of quantum computing consists in exploiting the strength of particular quantum properties in order to implement algorithms which are much more efficient and faster than the respective classical counterpart. At this purpose, several non standard applications involving the quantum mechanical formalism have been proposed, in research fields such as game theory [8,27], economics [11], cognitive sciences [2,40], signal processing [9], and so on. Further, particular applications, interesting for the specific topics of the present paper, concern the areas of machine learning and pattern recognition.

Quantum machine learning is an emerging research field which can use the advantages of quantum computation in order to find new solutions to pattern recognition and image understanding problems. About this, some attempts which connect quantum information to pattern recognition can be found in [31], while an exhaustive survey and bibliography of the developments regarding the use of quantum computing techniques in artificial intelligence are provided in [23,45].

In this context, there exist different approaches involving the use of quantum formalism in pattern recognition and machine learning. We can find for instance procedures which exploit quantum properties without presupposing the help of a quantum computer [13,19,38] or techniques supposing the existence of a quantum computer in order to perform in an inherently parallel way all the required operations, taking advantage of quantum mechanical effects and providing high performance in terms of computational efficiency [5,28,44].

One of the main aspects of pattern recognition is focused on the application of quantum information processing methods [20] to solve classification and clustering problems [5,12,39].

The use of quantum states for representing patterns has a twofold motivation: as already discussed, first of all it gives the possibility of exploiting quantum algorithms to boost the

computational efficiency of the classification process. Secondly, it is possible to use quantum-inspired models in order to reach some benefits with respect to classical problems.

Even if the state-of-art approaches suggest possible computational advantages of this sort [3,21,22], the main problem to find a *more convenient* encoding from classical to quantum objects is nowaday an open and interesting matter of debate [23,31]. Here, our contribution consists in constructing a quantum version of a minimum distance classifier in order to reach some convenience, in terms of the error in pattern classification, with respect to the corresponding classical model. We have already proposed this kind of approach in two previous works [33,34], where a "quantum counterpart" of the well known *Nearest Mean Classifier* (NMC) has been presented.

In both cases, the model is based on the introduction of two main ingredients: first, an appropriate encoding of arbitrary patterns into density operators; second, a distance measure between density operators, representing the quantum counterpart of the Euclidean distance in the "classical" NMC. The main difference between the two previous works is the following one: *i)* in the first case [34], we tested our quantum classifier on two-dimensional datasets and we proposed a generalization to arbitrary dimension from a theoretical point of view only; *ii)* in the second case [33], a new encoding for arbitrary n-dimensional patterns into quantum states has been proposed, and it was tested on different real-world and artificial two-class datasets. Anyway, in both cases we observed a significant improvement of the accuracy in the classification process. In addition, we found that, by using the encoding proposed in [33] and for two-dimensional problems only, the classification accuracy of our quantum classifier can be further improved by performing a uniform rescaling of the original dataset.

In this work we propose a new encoding of arbitrary n-dimensional patterns into quantum objects, extending both the thoretical model and the experimental results to multi-class problems, which preserves information about the norm of the original pattern. This idea has been inspired by recent debates on quantum machine learning [31], according to which it is crucial to avoid loss of information when a particular encoding of real vectors into quantum states is considered. Such an approach turns out to be very promising in terms of classification performance with respect to the classical version of the NMC. Further, differently from the NMC, our quantum classifier is invariant under uniform rescaling. More precisely, the accuracy of the quantum classifier changes by rescaling (of an arbitrary real number) the coordinates of the dataset. Consequently, we observe that, for several datasets, the new encoding exhibits a further advantage that can be gained by exploiting the non-invariance under rescaling, also for n-dimensional problems (conversely to the previous works). At this purpose, some experimental results have been presented.

The paper is organized as follows: in Section 2 we briefly describe the classification process and, in particular, the formal structure of the NMC for multi-class problems. Section 3 is devoted to the definition of a new encoding of real patterns into quantum states. In Section 4 we introduce the quantum version of the NMC, called *Quantum Nearest Mean Classifier* (QNMC), based on the new encoding previously described. In Section 5 we compare the NMC and the QNMC on different datasets showing that, in general, the QNMC exhibits a better performance (in terms of accuracy and other significant statistical quantities) with respect to the NMC. Further, starting from the fact that, differently from the NMC, the QNMC is not invariant under rescaling, we also show that for some datasets it is possible to provide a benefit from this non-invariance property. Some conclusions and possible further developments are proposed at the end of the paper. [1]

[1] The present work is an extended version of the paper presented at the conference *Quantum and Beyond 2016*, Vaxjo, 13-16 June 2016 [30], significantly enlarged in theoretical discussion, experimental section and bibliography.

2. Minimum distance classification

Pattern recognition [7,43] is the scientific discipline which deals with theories and methodologies for designing algorithms and machines capable of automatically recognizing "objects" (*i.e. patterns*) in noisy environments.

Here, we deal with *supervised learning*, *i.e.* learning from a training set of correctly labeled objects. In other words, this is the case in which examples of input-output relations are given to a computer and it has to infer a mapping from there. The most important task is *pattern classification*, whose goal is to assign input objects to different classes.

More precisely, each object can be characterized by its features; hence, a d-feature object can be naturally represented by a d-dimensional real vector, i.e. $\vec{x} = [x^{(1)}, \ldots, x^{(d)}] \in \mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is generally a subset of the d-dimensional real space representing the *feature space*. Consequently, any arbitrary object is represented by a vector $\vec{x}$ associated to a given class of objects (but, in principle, we do not know which one). Let $\mathcal{Y} = \{1, \ldots, L\}$ be the class label set. A *pattern* is represented by a pair $(\vec{x}, y)$, where $\vec{x}$ is the *feature vector* representing an object and $y \in \mathcal{Y}$ is the *label* of the class which $\vec{x}$ is associated to. The aim of the classification process is to design a function (*classifier*) that attributes (in the most accurate way) to any unlabeled object the corresponding label (where the label attached to an object represents the class which the object belongs to), by learning about the set of objects whose class is known. The *training set* is given by $\mathcal{S}_{\text{tr}} = \{(\vec{x}_n, y_n)\}_{n=1}^{N}$, where $\vec{x}_n \in \mathcal{X}$, $y_n \in \mathcal{Y}$ (for $n = 1, \ldots, N$) and N is the number of patterns belonging to $\mathcal{S}_{\text{tr}}$. Finally, let N_l be the cardinality of the training set associated to the l-th class (for $l = 1, 2, \ldots, L$) such that $\sum_{l=1}^{L} N_l = N$.

We now introduce the well known *Nearest Mean Classifier* (NMC) [7], which is a particular kind of minimum distance classifier widely used in pattern recognition. The strategy consists in computing the distances between an object $\vec{x}$ (to classify) and other objects chosen as prototypes of each class (called *centroids*). Finally, the classifier associates to $\vec{x}$ the label of the closest centroid. So, we can resume the NMC algorithm as follows:

1. computation of the *centroid* (*i.e.* the sample mean [15]) associated to each class, whose corresponding feature vector is given by:

$$\vec{\mu}_l = \frac{1}{N_l} \sum_{n=1}^{N_l} \vec{x}_n, \quad l = 1, 2, \ldots, L, \tag{1}$$

   where l is the label of the class;
2. classification of the object $\vec{x}$, provided by:

$$argmin_{l=1,\ldots L} d_E(\vec{x}, \vec{\mu}_l), \quad \text{with} \quad d_E(\vec{x}, \vec{\mu}_l) = \|\vec{x} - \vec{\mu}_l\|_2, \tag{2}$$

   where d_E is the standard Euclidean distance.[2].

Depending on the particular distribution of the dataset patterns, it is possible that a pattern belonging to a given class is closest to the centroid of another class. In this case, if the algorithm would be applied to this pattern, it would fail. Hence, for an arbitrary object $\vec{x}$ whose class is *a priori* unknown, the output of the above classification process has the following four possibilities [10]: *i) True Positive* (TP): pattern belonging to the l-th class and correctly classified as l; *ii) True Negative* (TN): pattern belonging to a class different than l, and correctly classified as not l; *iii) False Positive* (FP): pattern belonging to a class different than l, and uncorrectly classified as l; *iv) False Negative* (FN): pattern belonging to the l-th class, and uncorrectly classified as not l.

---

[2] We remind that, given a function $f : X \to Y$, the *argmin* (*i.e.* the argument of the minimum) over some subset S of X is defined as: $argmin_{x \in S \subseteq X} f(x) = \{x | x \in S \wedge \forall y \in S : f(y) \geq f(x)\}$. In this framework, the argmin plays the role of the classifier, *i.e.* a function that associates to any unlabeled object the correspondent label.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 October 2017 doi:10.20944/preprints201710.0100.v1

Peer-reviewed version available at Entropy 2017, 19, 659; doi:10.3390/e19120659

In order to evaluate the performance of a certain classification algorithm, the standard procedure consists in dividing the original labeled dataset $\mathcal{S}$ of N' patterns, into a training set $\mathcal{S}_{\text{tr}}$ of N patterns and a set $\mathcal{S}_{\text{ts}}$ of $(N'-N)$ patterns (*i.e.* $\mathcal{S} = \mathcal{S}_{\text{tr}} \cup \mathcal{S}_{\text{ts}}$). This set $\mathcal{S}_{\text{ts}}$ of patterns is called *test set* [7] and it is defined as $\mathcal{S}_{\text{ts}} = \{(\vec{x}_n, y_n)\}_{n=N+1}^{N'}$.

As a consequence, by applying the NMC to the test set, it is possible to evaluate the classification algorithm performance by considering the following statistical measures associated to each class l depending on the quantities listed above:

- *True Positive Rate* (TPR): $\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$;
- *True Negative Rate* (TNR): $\text{TNR} = \frac{\text{TN}}{\text{TN+FP}}$;
- *False Positive Rate* (FPR): $\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} = 1 - \text{TPN}$;
- *False Negative Rate* (FNR): $\text{FNR} = \frac{\text{FN}}{\text{FN+TP}} = 1 - \text{TPR}$.

Further, other standard statistical coefficients [10] used to establish the reliability of a classification algorithm are:

- *Classification error* (E): $\text{E} = 1 - \frac{\text{TP}}{N'-N}$;
- *Precision* (P): $\text{P} = \frac{\text{TP}}{\text{TP+FP}}$;
- *Cohen's Kappa* (K): $\text{K} = \frac{\text{Pr(a)}-\text{Pr(e)}}{1-\text{Pr(e)}}$, where
  $\text{Pr(a)} = \frac{\text{TP+TN}}{N'-N}$, $\text{Pr(e)} = \frac{(\text{TP+FP})(\text{TP+FN})+(\text{FP+TN})(\text{TN+FN})}{(N'-N)^2}$.

In particular, the classification error represents the percentage of misclassified patterns, the precision is a measure of the statistical variability of the considered model and the Cohen's Kappa represents the degree of reliability and accuracy of a statistical classification and it can assume values ranging from -1 to $+1$ (K$= +1$ corresponds to a perfect classification procedure while K$= -1$ corresponds to a completely wrong classification). Let us note that these statistical coefficients have to be computed for each class. Then, the final value of each statistical coefficient related to the classification algorithm is the weighted sum of the statistical coefficients of each class.

3. Mapping real patterns into quantum states

As already discussed, quantum formalism turns out to be very useful in non-standard scenarios, in our case to solve for instance classification problems on datasets of classical objects. At this purpose, in order to provide our quantum classification model, the first ingredient we have to introduce is an appropriate encoding of real patterns into quantum states. Quoting Schuld et al. [31], "in order to use the strengths of quantum mechanics without being confined by classical ideas of data encoding, finding 'genuinely quantum' ways of representing and extracting information could become vital for the future of quantum machine learning."

Generally, given a d-dimensional feature vector, there exist different ways to encode it into a density operator [31]. In [34], the proposed encoding was based on the use of the stereographic projection [6]. In particular, it allows to unequivocally map any point $\vec{r} = (r_1, r_2, r_3)$ on the surface of a radius-one sphere $\mathbb{S}^2$ (except for the north pole) into an arbitrary point $\vec{x} = [x^{(1)}, x^{(2)}]$ in $\mathbb{R}^2$, *i.e.*

$$SP : (r_1, r_2, r_3) \mapsto \left(\frac{r_1}{1-r_3}, \frac{r_2}{1-r_3}\right) = [x^{(1)}, x^{(2)}]. \tag{3}$$

The inverse of the stereographic projection is given by:

$$SP^{-1} : [x^{(1)}, x^{(2)}] \mapsto \left[\frac{2x^{(1)}}{||\vec{x}||^2+1}, \frac{2x^{(2)}}{||\vec{x}||^2+1}, \frac{||\vec{x}||^2-1}{||\vec{x}||^2+1}\right] = (r_1, r_2, r_3), \tag{4}$$

where $||\vec{x}||^2 = [x^{(1)}]^2 + [x^{(2)}]^2$. By imposing that $r_1 = \frac{2x^{(1)}}{||\vec{x}||^2+1}$, $r_2 = \frac{2x^{(2)}}{||\vec{x}||^2+1}$, $r_3 = \frac{||\vec{x}||^2-1}{||\vec{x}||^2+1}$, if we consider r_1, r_2, r_3 as Pauli components[3] of a density operator $\rho_{\vec{x}} \in \Omega_2$[4], the density operator associated to the pattern $\vec{x} = [x^{(1)}, x^{(2)}]$ can be written as:

$$\rho_{\vec{x}} = \frac{1}{2}\begin{pmatrix} 1+r_3 & r_1 - ir_2 \\ r_1 + ir_2 & 1-r_3 \end{pmatrix} = \frac{1}{||\vec{x}||^2+1}\begin{pmatrix} ||\vec{x}||^2 & x^{(1)} - ix^{(2)} \\ x^{(1)} + ix^{(2)} & 1 \end{pmatrix}. \tag{5}$$

The advantage in using this encoding consists in the fact that it provides an easy visualization of an arbitrary two-feature vector on the Bloch sphere [34]. In the same work, we also introduced a generalization of our encoding to the d-dimensional case, allowing to express arbitrary d-feature vectors as points on the hypersphere S^d by writing a density operator ρ as a linear combination of the d-dimensional identity and $d^2 - 1$ $(d \times d)$-square matrices $\{\sigma_i\}$ (*i.e. generalized Pauli matrices* [4,17]).

At this purpose, we introduced the generalized stereographic projection [16], which maps any point $\vec{r} = (r_1, \ldots, r_{d+1}) \in \mathbb{S}^d$ into an arbitrary point $\vec{x} = [x^{(1)}, \ldots, x^{(d)}] \in \mathbb{R}^d$, *i.e.*:

$$SP : (r_1, \ldots, r_{d+1}) \mapsto \left(\frac{r_1}{1-r_{d+1}}, \frac{r_2}{1-r_{d+1}}, \ldots, \frac{r_d}{1-r_{d+1}}\right) = [x^{(1)}, \ldots, x^{(d)}]. \tag{6}$$

However, even if it is possible to map points on the d-hypersphere into d-feature patterns, such points do not generally represent density operators and the one-to-one correspondence between them and density matrices is guaranteed only on particular regions [14,17,18].

An alternative encoding of a d-feature vector $\vec{x}$ into a density operator was proposed in [33]. It is obtained: *i)* by mapping $\vec{x} \in \mathbb{R}^d$ into a $(d+1)$-dimensional vector $\vec{x}' \in \mathbb{R}^{d+1}$ according to the generalized version of Eq. (4), *i.e.*

$$SP^{-1} : [x^{(1)}, \ldots, x^{(d)}] \mapsto \frac{1}{||\vec{x}||^2+1}\left[2x^{(1)}, \ldots, 2x^{(d)}, ||\vec{x}||^2 - 1\right] = (r_1, \ldots, r_{d+1}), \tag{7}$$

where $||\vec{x}||^2 = \sum_{i=1}^{d}[x^{(i)}]^2$; *ii)* by considering the projector $\rho_{\vec{x}} = \vec{x}' \cdot (\vec{x}')^T$.

In this work we propose a different version of the QNMC based on a new encoding again and we show that this exhibits interesting improvements also by exploiting the non-invariance under rescaling of the features.

Accordingly with [21,28,31], when a real vector is encoded into a quantum state, in order to avoid a loss of information it is important that the quantum state keeps some information about the norm of the original real vector. In light of this fact, we introduce the following alternative encoding.

Let $\vec{x} = [x^{(1)}, \ldots, x^{(d)}] \in \mathbb{R}^d$ be an arbitrary d-feature vector.

1. We maps the vector $\vec{x} \in \mathbb{R}^d$ into a vector $\vec{x}' \in \mathbb{R}^{d+1}$, whose first d features are the components of the vector $\vec{x}$ and the $(d+1)$-th feature is the norm of $\vec{x}$. Formally:

$$\vec{x} = [x^{(1)}, \ldots, x^{(d)}] \;\mapsto\; \vec{x}' = [x^{(1)}, \ldots, x^{(d)}, ||\vec{x}||]. \tag{8}$$

2. We obtain the vector $\vec{x}''$ by dividing the first d components of the vector $\vec{x}'$ for $||\vec{x}||$:

$$\vec{x}' \;\mapsto\; \vec{x}'' = \left[\frac{x^{(1)}}{||\vec{x}||}, \ldots, \frac{x^{(d)}}{||\vec{x}||}, ||\vec{x}||\right]. \tag{9}$$

---

[3] We consider the representation of an arbitrary density operator as linear combination of Pauli matrices.
[4] The space Ω_d of density operators for d-dimensional systems consists of positive semidefinite matrices with unitary trace.

3. We consider the norm of the vector $\vec{x}''$, *i.e.* $||\vec{x}''|| = \sqrt{||\vec{x}||^2 + 1}$ and we map the vector $\vec{x}''$ into the normalized vector $\vec{x}'''$ as follows:

$$\vec{x}'' \mapsto \vec{x}''' = \frac{\vec{x}''}{||\vec{x}''||} = \Big[\frac{x^{(1)}}{||\vec{x}||\sqrt{||\vec{x}||^2+1}}, \ldots, \frac{x^{(d)}}{||\vec{x}||\sqrt{||\vec{x}||^2+1}}, \frac{||\vec{x}||}{\sqrt{||\vec{x}||^2+1}}\Big]. \tag{10}$$

Now, we provide the following definition.

Definition 1 (Density Pattern) Let $\vec{x} = [x^{(1)}, \ldots, x^{(d)}]$ be an arbitrary d-feature vector and $(\vec{x}, y)$ the corresponding pattern. Then, the *density pattern* associated to $(\vec{x}, y)$ is represented by the pair $(\rho_{\vec{x}}, y)$, where the matrix $\rho_{\vec{x}}$, corresponding to the feature vector $\vec{x}$, is defined as:

$$\rho_{\vec{x}} \doteq \vec{x}''' \cdot (\vec{x}''')^{\dagger}, \tag{11}$$

where the vector $\vec{x}'''$ is given by Eq. (10) and y is the label of the original pattern.

Hence, this encoding maps real d-dimensional vectors $\vec{x}$ into $(d+1)$-dimensional pure states $\rho_{\vec{x}}$. In this way, we obtain an encoding that takes into account the information about the initial real vector norm and, at the same time, allows to easily encode also arbitrary real d-dimensional vectors.

4. Density Pattern Classification

In this section we introduce a quantum counterpart of the NMC, named *Quantum Nearest Mean Classifier* (QNMC). It can be seen as a particular kind of minimum distance classifier between quantum objects (*i.e.* density patterns). The use of this new formalism could lead not only to achieve the well known advantages related to the quantum computation with respect to the classical one (mostly related to the speed up of the computational process), but also to make a full comparison between NMC and QNMC performance by using a classical computer only.

In order to provide a quantum counterpart of the NMC, we need: *i)* an encoding from real patterns to quantum objects (already defined in the previous section); *ii)* a quantum counterpart of the classical centroid (*i.e.* a sort of quantum class prototype), that will be named *quantum centroid*; *iii)* a suitable definition of *quantum distance* between density patterns, that plays the same role as the Euclidean distance for the NMC. In this quantum framework, the quantum version $\mathcal{S}^q$ of the dataset $\mathcal{S}$ is given by:

$$\mathcal{S}^q = \mathcal{S}^q_{\text{tr}} \cup \mathcal{S}^q_{\text{ts}}, \quad \mathcal{S}^q_{\text{tr}} = \{(\rho_{\vec{x}_n}, y_n)\}_{n=1}^{N}, \quad \mathcal{S}^q_{\text{ts}} = \{(\rho_{\vec{x}_n}, y_n)\}_{n=N+1}^{N'},$$

where $(\rho_{\vec{x}_n}, y_n)$ is the density pattern associated to the pattern $(\vec{x}_n, y_n)$. Consequently, $\mathcal{S}^q_{\text{tr}}$ and $\mathcal{S}^q_{\text{ts}}$ represent the quantum versions of training and test set respectively, *i.e.* the sets of all the density patterns obtained by encoding all the elements of $\mathcal{S}_{\text{tr}}$ and $\mathcal{S}_{\text{ts}}$. Now, we naturally introduce the quantum version of the classical centroid $\vec{\mu}_l$, given in Eq. (1), as follows.

Definition 2 (Quantum Centroid) Let $\mathcal{S}^q$ be a labeled dataset of N' density patterns such that $\mathcal{S}^q_{\text{tr}} \subseteq \mathcal{S}^q$ is a training set composed of N density patterns. Further, let $\mathcal{Y} = \{1, 2, \ldots, L\}$ be the class label set. The *quantum centroid* of the l-th class is given by:

$$\rho_l = \frac{1}{N_l} \sum_{n=1}^{N_l} \rho_{\vec{x}_n}, \quad l = 1, \ldots, L \tag{12}$$

where N_l is the number of density patterns of the l-th class belonging to $\mathcal{S}^q_{\text{tr}}$, such that $\sum_{l=1}^{L} N_l = N$.

Notice that the quantum centroids are generally mixed states and they are not obtained by encoding the classical centroids $\vec{\mu}_l$, *i.e.*

$$\rho_l \neq \rho_{\vec{\mu}_l}, \ \forall l \in \{1, \ldots, L\}. \tag{13}$$

Accordingly, the definition of the quantum centroid leads to a new object that is no longer a pure state and does not have any classical counterpart. This is the main reason that establishes, even in a fundamental level, the difference between NMC and QNMC. In particular, it is easy to verify [34] that, unlike the classical case, the expression of the quantum centroid is sensitive to the dataset dispersion.

In order to consider a suitable definition of distance between density patterns, we recall the well known definition of trace distance between quantum states (see, *e.g.* [25]).

Definition 3 (Trace Distance) Let ρ and ρ' be two quantum density operators belonging to the same dimensional Hilbert space. The *trace distance* between ρ and ρ' is given by:

$$d_T(\rho, \rho') = \frac{1}{2} Tr|\rho - \rho'|, \tag{14}$$

where $|A| = \sqrt{A^\dagger A}$.

Notice that the trace distance is a true metric for density operators, that is, it satisfies: *i)* $d_T(\rho, \rho') \geq 0$ with equality iff $\rho = \rho'$ (*positivity*), *ii)* $d_T(\rho, \rho') = d_T(\rho', \rho)$ (*symmetry*) and *iii)* $d_T(\rho, \rho') + d_T(\rho', \rho'') \geq d_T(\rho, \rho'')$ (*triangle inequality*). The use of the trace distance in our quantum framework is naturally motivated by the fact that it is the simplest possible choice among other possible metrics in the density matrix space [36]. Consequently, it can be seen as the "authentic" quantum counterpart of the Euclidean distance, which represents the simplest choice in the starting space. However, the trace distance exhibits some limitations and downsides (in particular, it is monotone but not Riemannian [29]). On the other hand, the Euclidean distance in some pattern classification problems is not enough to fully capture for instance the dataset distribution. For this reason, other kinds of metrics in the classical space are adopted to avoid this limitation [7]. At this purpose, as a future development of the present work, it could be interesting to compare different distances in both quantum and classical framework, able to treat more complex situations (we will deepen this point in the conclusions).

We have introduced all the ingredients we need to describe the QNMC process, that, similarly to the classical case, consists in the following steps:

- constructing the quantum training and test sets $\mathcal{S}^q_{\text{tr}}$, $\mathcal{S}^q_{\text{ts}}$ by applying the encoding introduced in Definition 1 to each pattern of the classical training and test sets $\mathcal{S}_{\text{tr}}$, $\mathcal{S}_{\text{ts}}$;
- calculating the quantum centroids ρ_l ($\forall l \in \{1, \dots L\}$), by using the quantum training set $\mathcal{S}^q_{\text{tr}}$, according to Definition 2;
- classifying an arbitrary density pattern $\rho_{\vec{x}} \in S^q_{\text{ts}}$ accordingly with the following minimization problem:

$$argmin_{l=1,\dots,L} d_T(\rho_{\vec{x}}, \rho_l), \tag{15}$$

where d_T is the trace distance introduced in Definition 3.

5. Experimental results

This section is devoted to show a comparison between the NMC and the QNMC performances in terms of the statistical coefficients introduced in Section 2. We use both classifiers to analyze twenty-seven datasets, divided into two categories: artificial datasets (*Gaussian (I), Gaussian (II), Gaussian (III), Moon, Banana*) and the remaining ones which are real-world datasets, extracted both from the UCI and KEEL repositories[5]. Further, among them we can find also imbalanced datasets, whose main characteristic is that the number of patterns belonging to one class is significantly lower than those belonging to the other classes. Let us note that, in real situations, we usually deal with data whose distribution is unknown, then the most interesting case is the one in which we use real-world

[5] http://archive.ics.uci.edu/ml, http://sci2s.ugr.es/keel/datasets.php

datasets. However, the use of artificial datasets following known distribution, and in particular Gaussian distributions with specific parameters, can help to catch precious information.

5.1. Comparison between QNMC and NMC

In Table 1 we summarize the characteristics of the datasets involved in our experiments. In particular, for each dataset we list the total number of patterns, the number of patterns belonging to each class and the number of features. Let us note that, although we mostly confine our investigation to two-class datasets, our model can be easily extended to multi-class problems (as we show for the three-class datasets *Balance*, *Gaussian (III)*, *Hayes-Roth*, *Iris*).

In order to make our results statistically significant, we apply the standard procedure which consists in randomly splitting each dataset into two parts, the training set (representing the 80% of the original dataset) and the test set (representing the 20% of the original dataset). Finally, we perform ten experiments for each dataset, where the splitting is every time randomly taken.

Table 1. Characteristics of the datasets used in our experiments. The number of patterns in each class is shown between brackets.

Data set	Class Size	Features (d)
Appendicitis	106 (85+21)	7
Balance	625 (49+288+288)	4
Banana	5300 (2376+2924)	2
Bands	365 (135+230)	19
Breast Cancer (I)	683 (444+239)	10
Breast Cancer (II)	699 (458+241)	9
Bupa	345 (145+200)	6
Chess	3196 (1669+1527)	36
Gaussian (I)	400 (200+200)	30
Gaussian (II)	1000 (100+900)	8
Gaussian (III)	2050 (50+500+1500)	8
Hayes-Roth	132 (51+51+30)	5
Ilpd	583 (416+167)	9
Ionosphere	351 (225+126)	34
Iris	150 (50+50+50)	4
Iris0	150 (100+50)	4
Liver	578 (413+165)	10
Monk	432 (204+228)	6
Moon	200 (100+100)	2
Mutagenesis-Bond	3995 (1040+2955)	17
Page	5472 (4913+559)	10
Pima	768 (500+268)	8
Ring	7400 (3664+3736)	20
Segment	2308 (1979+329)	19
Thyroid (I)	215 (180+35)	5
Thyroid (II)	215 (35+180)	5
TicTac	958 (626+332)	9

In Table 2, we report the QNMC and NMC performance for each dataset, evaluated in terms of mean value and standard deviation (computed on ten runs) of the statistical coefficients, discussed in the previous section. For the sake of semplicity, we omit the values of FPR and FNR because they can be easily obtained by TPR and TNR values (*i.e.* FPR = 1 - TNR, FNR = 1 - TPR).

We observe, by comparing QNMC and NMC performances (see Table 2), that the first provides a significant improvement with respect to the standard NMC in terms of all the statistical parameters we have considered. In several cases, the difference between the classification error for both classifiers is very high, up to 22% (see *Mutagenesis-Bond*). Further, the new encoding, for two-feature datasets, provides better performance than the one considered in [34] (where the QNMC error with related

standard deviation was 0.174 ± 0.047 for *Moon* and 0.419 ± 0.015 for *Banana*) and it generally exhibits a quite similar performance with respect to the one in [33] for multi-dimension datasets or a classification improvement of about 5%, generally.

The artificial Gaussian datasets may deserve a brief comment. Let us discuss the way in which the three Gaussian datasets have been created. *Gaussian (I)* [35] is a perfectly balanced dataset (*i.e.* both classes have the same number of patterns), patterns have the same dispersion in both classes, and only some features are correlated [42]. *Gaussian (II)* is an unbalanced dataset (*i.e.* classes have a very different number of patterns), patterns do not exhibit the same dispersion in both classes and features are not correlated. *Gaussian (III)* is composed of three classes and it is an unbalanced dataset with different pattern dispersion in all the classes, where all the features are correlated.

For these Gaussian datasets, the NMC is not the best classifier [7] because of the particular characteristics of the class dispersion. Indeed, the NMC does not take into account data dispersion. Conversely, by looking at Table 2, the improvements of the QNMC seem to exhibit some kind of sensitivity of the classifier with respect to the data dispersion. A detailed description of this problem will be addressed in a future work.

Further, we can note that the QNMC performance is better also for imbalanced datasets (the most significant cases are *Balance*, *Ilpd*, *Segment*, *Page*, *Gaussian (III)*), which are usually difficult to deal with standard classification models. At this purpose, we can note that the QNMC exhibits a classification error much lower than the NMC, up to a difference of about 12%. Another interesting and surprising result concerns the *Iris0* dataset, which represents the imbalanced version of the *Iris* dataset: as we can observe looking at Table 2, our quantum classifier is able to perfectly classify all the test set patterns, conversely to the NMC.

As a remark, it is important to remind that, even if it is possible to establish whether a classifier is "good" or "bad" for a given dataset by the evaluation of some a priori data characteristics, generally it is no possible to establish an absolute superiority of a given classifier for any dataset, according to the well known *No Free Lunch Theorem* [7]. Anyway, the QNMC seems to be particularly convenient when the data distribution is difficult to treat with the standard NMC.

5.2. Non-invariance under rescaling

The final experimental results that we present in this paper regard a significant difference between NMC and QNMC. Let us suppose that all the components of the feature vectors $\vec{x}_n$ ($\forall n = 1, \ldots, N'$) belonging to the original dataset $\mathcal{S}$ are multiplied by the same parameter $\gamma \in \mathbb{R}$, *i.e.* $\vec{x}_n \mapsto \gamma \vec{x}_n$. Then, the whole dataset is subjected to an increasing dispersion (for $|\gamma| > 1$) or a decreasing dispersion (for $|\gamma| < 1$) and the classical centroids change according to $\vec{\mu}_l \mapsto \gamma \vec{\mu}_l$ ($\forall l = 1, \ldots, L$). Consequently, the classification problem for each pattern of the rescaled test set can be written as

$$argmin_{l=1,\ldots,L} d_E(\gamma \vec{x}_n, \gamma \vec{\mu}_l) = \gamma argmin_{l=1,\ldots,L} d_E(\vec{x}_n, \vec{\mu}_l), \ \forall n = N+1, \ldots, N'.$$

For any value of the parameter γ it can be proved [33] that, while the NMC is invariant under rescaling, for the QNMC this invariance fails. Interestingly enough, it is possible to consider the failure of the invariance under rescaling as a resource for the classification problem. In other words, by a suitable choice of the rescaling factor is possible, in principle, to get a decreasing of the classification error. At this purpose, we have studied the variation of the QNMC performance (in particular of the classification error) in terms of the *free* parameter γ and in Fig. 1 the results for the datasets *Appendicitis*, *Monk* and *Moon* are shown. In the figure, each point represents the mean value (with corresponding standard deviation represented by the vertical bar) over ten runs of the experiments. Finally, we have considered, as an example, three different ranges of the rescaling parameter γ for each dataset. We can observe that the resulting classification performance strongly depends on the γ range. Indeed, in all the three cases we consider, we obtain completely different classification results based on different choices of the γ values. As we can see, in some situations we observe an improvement of the QNMC

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 October 2017 doi:10.20944/preprints201710.0100.v1

Peer-reviewed version available at Entropy 2017, 19, 659; doi:10.3390/e19120659

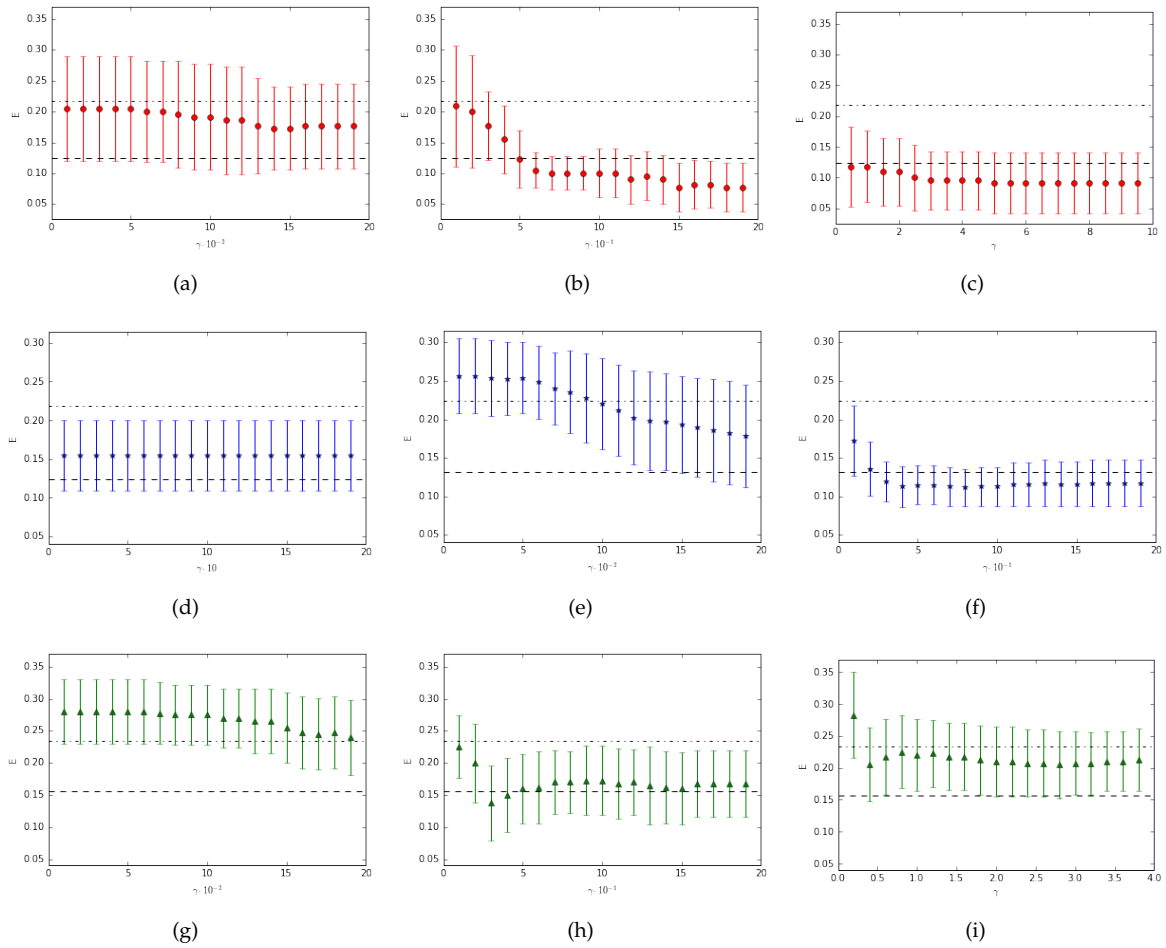

Figure 1. Comparison between NMC and QNMC performance in terms of the classification error for the datasets (a)-(c) *Appendicitis*, (d)-(f) *Monk*, (g)-(i) *Moon*. In all the subfigures, the simple dashed line represents the QNMC classification error without rescaling, the dashed line with points represents the NMC classification error (which does not depend on the rescaling parameter), points with related error bars (red for *Appendicitis*, blue for *Monk* and green for *Moon*) represent the QNMC classification error for increasing values of the parameter γ.

performance with respect to the unrescaled problem (subfigures (b), (c), (f), (h)), in other cases we get worse classification results (subfigures (a), (e), (g), (i)) and sometimes the rescaling parameter does not offer any variation of the classification error (subfigure (d)).

In conclusion, the range of the parameter γ for which the QNMC performance improves, is generally not unique and strongly depends on the considered dataset. As a consequence, we do not generally get an improvement in the classification process for any γ ranges. On the contrary, there exist some intervals of the parameter γ where the QNMC classification performance is worse than the case without rescaling. Then, each dataset has specific and unique characteristics (in completely accord to the No Free Lunch Theorem) and the incidence of the non-invariance under rescaling in the decreasing of the error, in general, should be determined by empirical evidences.

6. Conclusions and future work

In this work a quantum counterpart of the well known Nearest Mean Classifier has been proposed. We have introduced a quantum minimum distance classifier, called Quantum Nearest Mean Classifier,

obtained by defining a suitable encoding of real patterns, *i.e. density patterns*, and by recovering the trace distance between density operators.

A new encoding of real patterns into a quantum objects have been proposed, suggested by recent debates on quantum machine learning according to which, in order to avoid a loss of information caused by encoding a real vector into a quantum state, we need to normalize the vector mantaining some information about its norm. Secondly, we have defined the *quantum centroid*, *i.e.* the pattern chosen as the prototype of each class, which is not invariant under uniform rescaling of the original dataset (unlike the NMC) and seems to exhibit a kind of sensitivity to the data dispersion.

In the experiments, both classifiers have been compared in terms of significant statistical coefficients. In particular, we have considered twenty-seven different datasets having different nature (real-world and artificial). Further, the non-invariance under rescaling of the QNMC has suggested to study the variation of the classification error in terms of a free parameter γ, whose variation produces a modification of the data dispersion and, consequently, of the classifier performance. In particular we have showed as, in the most of cases, the QNMC exhibits a significant decreasing of the classification error (and of the other statistical coefficients) with respect to the NMC and, for some cases, the non-invariance under rescaling can provide a positive incidence in the classification process.

Let us remark that, even if there is not an absolute superiority of QNMC with respect to the NMC, the method we have introduced allows to get some relevant improvements of the classification when we have an *a priori* knowledge about the distribution of the dataset we have to deal with.

In light of such considerations, further developments of the present work will be focused on: *i)* finding out the encoding (from real vectors to density operators) that guarantees the *optimal* improvement (at least for a finite class of datasets) in terms of the classification process accuracy; *ii)* obtain a general method to find the suitable rescaling parameter range we can apply to a given dataset in order to get a further improvement of the accuracy; *iii)* understanding for which kind of distribution the QNMC performs better than the NMC. Further, as discussed in Section 4, in some situations the standard NMC is not very useful as classification model, especially when the dataset distribution is quite complex to deal with. In pattern recognition, in order to address such problems, other kinds of classification techniques are used instead of the NMC, for instance the well known *Linear Discriminant Analysis* (LDA) or *Quadratic Discriminant Analysis* (QDA) classifiers, where different distances between patterns are considered, taking into account more precisely the data distribution [7]. At this purpose, an interesting development of the present work could regard the comparison between the LDA or QDA models and the QNMC based on the computation of more suitable and convenient distances between density patterns [36].

References

1. D. Aerts, B. D'Hooghe, Classical logical versus quantum conceptual thought: examples in economics, decision theory and concept theory, *Quantum interaction*, *Lecture Notes in Computer Science*, **5494**:128–142. Springer, Berlin (2009)
2. D. Aerts, S. Sozzo, T. Veloz, Quantum structure of negation and conjunction in human thought, *Frontiers in Psychology*, **6**:1447 (2015)
3. E. Aïmeur, G. Brassard, S. Gambs. *Machine learning in a quantum world, Conference of the Canadian Society for Computational Studies of Intelligence*, Springer Berlin Heidelberg (2006)
4. R.A. Bertlmann, P. Krammer, Bloch vectors for qudits, *Journal of Physics A: Mathematical and Theoretical*, **41**(23):235303, 21 (2008)
5. S. Caraiman, V. Manta, *Image processing using quantum computing*, *System Theory, Control and Computing (ICSTCC), 2012 16th International Conference on*, 1–6, IEEE (2012)
6. H.S.M. Coxeter, *Introduction to geometry*, 2nd edn. (John Wiley & Sons, Inc., New York-London-Sydney, 1969)
7. R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, 2nd edn. (Wiley Interscience, 2000)
8. J. Eisert, M. Wilkens, M. Lewenstein, Quantum games and quantum strategies, *Physical Review Letters*, **83**(15):3077 (1999)

9. Y.C. Eldar and A.V. Oppenheim, Quantum signal processing, *Signal Processing Magazine*, IEEE, **19**(6):12–32 (2002)
10. T. Fawcett, An introduction of the ROC analysis, *Pattern Recognition Letters*, **27**(8):861–874 (2006)
11. E. Haven, A. Khrennikov, *Quantum Social Science*, Cambridge University Press (2013)
12. F. Holik, G. Sergioli, H. Freytes, A. Plastino, Pattern Recognition in Non-Kolmogorovian Structures, *Foundations of Science*, https://doi.org/10.1007/s10699-017-9520-4 (2017)
13. D. Horn and A. Gottlieb, Algorithm for data clustering in pattern recognition problems based on quantum mechanics, *Physical Review Letters*, **88**(1), 018702 (2001)
14. L. Jakóbczyk and M. Siennicki, Geometry of bloch vectors in two-qubit system, *Physics Letters A*, **286**(6):383–390 (2001)
15. R.A. Johnson, D.W. Wichern, *Applied Multivariate Statistical Analysis*, Pearson Prentice Hall (2007)
16. B. Karlığa, On the generalized stereographic projection, *Beiträge zur Algebra und Geometrie*, **37**(2):329–336 (1996)
17. G. Kimura, The Bloch vector for N-level systems, *Physics Letters A*, **314**(5–6):339–349 (2003)
18. G. Kimura and A. Kossakowski, The Bloch-vector space for N-level systems: the spherical-coordinate point of view, *Open Systems & Information Dynamics*, **12**(03):207–229 (2005)
19. D. Liu, X. Yang, M. Jiang, *A Novel Text Classifier Based on Quantum Computation*, *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics*, 484–488, Sofia, Bulgaria, August 4-9 (2013)
20. J.A. Miszczak, *High-level Structures for Quantum Computing*, *Synthesis Lectures on Quantum Computing* **6**, Morgan & Claypool Publishers (2012)
21. S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411 (2013)
22. S. Lloyd, M. Mohseni, P. Rebentrost, Quantum principal component analysis, *Nature Physics*, **10**(9):631–633 (2014)
23. A. Manju, M.J. Nigam, Applications of quantum inspired computational intelligence: a survey, *Artificial Intelligence Review*, **42**(1):79–156 (2014)
24. E. Nagel, Assumptions in economic theory, *The American Economic Review*, 211–219 (1963)
25. M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information - 10th Anniversary Edition*, Cambridge University Press (2010)
26. M. Ohya and I. Volovich, *Mathematical foundations of quantum information and computation and its applications to nano- and bio-systems*, Theoretical and Mathematical Physics, Springer, Dordrecht (2011)
27. E.W. Piotrowski, J. Sladkowski, An invitation to quantum game theory. *International Journal of Theoretical Physics*, 42(5):1089–1099 (2003)
28. P. Rebentrost, M. Mohseni, S. Lloyd, Quantum support vector machine for big feature and big data classification, *Physical Review Letters*, **113**:130503 (2014)
29. M.B. Ruskai, Beyond strong subadditivity? Improved bounds on the contraction of generalized relative entropy, *Reviews in Mathematical Physics*, **06**:1147–1161 (1994)
30. E. Santucci, G. Sergioli. *Classification problem in a quantum framework*, Quantum Foundations, Probability and Information, Advanced Methods in Interdisciplinary Mathematical Research, Springer, *in press* (2017)
31. M. Schuld, I. Sinayskiy, F. Petruccione, An introduction to quantum machine learning, *Contemporary Physics*, **56**(2):172–185 (2014)
32. J.M. Schwartz, H.P. Stapp, M. Beauregard, Quantum physics in neuroscience and psychology: a neurophysical model of mind-brain interaction, *Philosophical Transactions of the Royal Society B: Biological Sciences*, **360**(1458):1309–1327 (2005)
33. G. Sergioli, G.M. Bosyk, E. Santucci, R. Giuntini. A quantum-inspired version of the classification problem, *International Journal of Theoretical Physics*, **52**(9):1–9 (2017)
34. G. Sergioli, E. Santucci, L. Didaci, J.A. Miszczak, R. Giuntini. A quantum-inspired version of the Nearest Mean Classifier, *Soft Computing*, 10.1007/s00500-016-2478-2 (2016)
35. M. Skurichina, R.P.W. Duin, Bagging, Boosting and the Random Subspace Method for Linear Classifiers, *Pattern Analysis and Applications*, **5**(2):121–135 (2002)
36. H.J. Sommers, K. Zyczkowski, Bures volume of the set of mixed quantum states, *Journal of Physics A: Mathematical and General*, **36**(39):10083—10100 (2003)
37. H.P. Stapp, *Mind, matter, and quantum mechanics*, Springer-Verlag, Berlin (1993)

38. K. Tanaka, K. Tsuda, A quantum-statistical-mechanical extension of gaussian mixture model, *Journal of Physics: Conference Series*, **95**(1):012023 (2008)
39. C.A. Trugenberger, Quantum pattern recognition, *Quantum Information Processing*, **1**(6):471–493 (2002)
40. T. Veloz, S. Desjardins, Unitary Transformations in the Quantum Model for Conceptual Conjunctions and Its Application to Data Representation, *Frontiers in Psychology*, **6**:1734 (2015)
41. B. Wang, P. Zhang, J. Li, D. Song, Y. Hou, Z. Shang, Exploration of quantum interference in document relevance judgement discrepancy, *Entropy*, **18**(4), 144 (2016)
42. L. Wassermann, *All of Statistic: a Concise Course in Statistical Inference*, Springer Texts in Statistics, Springer (2004)
43. A.R. Webb, K.D. Copsey, *Statistical Pattern Recognition*, Wiley, 3rd edition (2011)
44. N. Wiebe, A. Kapoor, K.M. Svore, Quantum nearest-neighbor algorithms for machine learning, *Quantum Information and Computation*, **15**(34):0318–0358 (2015)
45. P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining*, Academic Press (2014)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 October 2017 doi:10.20944/preprints201710.0100.v1

Peer-reviewed version available at Entropy 2017, 19, 659; doi:10.3390/e19120659

Table 2. Comparison between QNMC and NMC performances.

			QNMC		
Dataset	E	TPR	TNR	P	K
Appendicitis	0.124 ± 0.058	0.876 ± 0.058	0.708 ± 0.219	0.886 ± 0.068	0.553 ± 0.223
Balance	0.148 ± 0.018	0.852 ± 0.018	0.915 ± 0.014	0.862 ± 0.022	0.767 ± 0.029
Banana	0.316 ± 0.017	0.684 ± 0.017	0.660 ± 0.017	0.684 ± 0.018	0.350 ± 0.034
Bands	0.394 ± 0.053	0.606 ± 0.053	0.528 ± 0.071	0.606 ± 0.058	0.133 ± 0.112
Breast Cancer (I)	0.386 ± 0.038	0.614 ± 0.038	0.444 ± 0.045	0.583 ± 0.044	0.062 ± 0.069
Breast Cancer (II)	0.040 ± 0.015	0.946 ± 0.023	0.986 ± 0.016	0.993 ± 0.009	0.912 ± 0.033
Bupa	0.389 ± 0.044	0.610 ± 0.044	0.641 ± 0.052	0.359 ± 0.052	0.066 ± 0.044
Chess	0.256 ± 0.017	0.744 ± 0.017	0.747 ± 0.016	0.748 ± 0.016	0.488 ± 0.033
Gaussian (I)	0.274 ± 0.051	0.726 ± 0.051	0.728 ± 0.049	0.745 ± 0.048	0.452 ± 0.099
Gaussian (II)	0.210 ± 0.025	0.790 ± 0.025	0.744 ± 0.061	0.900 ± 0.019	0.308 ± 0.058
Gaussian (III)	0.401 ± 0.036	0.599 ± 0.036	0.558 ± 0.026	0.654 ± 0.041	0152 ± 0.043
Hayes-Roth	0.413 ± 0.039	0.588 ± 0.039	0.780 ± 0.025	0.602 ± 0.063	0.339 ± 0.060
Ilpd	0.351 ± 0.037	0.649 ± 0.037	0.705 ± 0.056	0.734 ± 0.041	0.292 ± 0.073
Ionosphere	0.165 ± 0.049	0.835 ± 0.049	0.764 ± 0.059	0.842 ± 0.051	0.624 ± 0.105
Iris	0.047 ± 0.031	0.953 ± 0.031	0.977 ± 0.014	0.957 ± 0.028	0.929 ± 0.045
Iris0	0 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
Liver	0.342 ± 0.037	0.607 ± 0.057	0.783 ± 0.059	0.870 ± 0.039	0.318 ± 0.061
Monk	0.132 ± 0.034	0.869 ± 0.034	0.885 ± 0.030	0.891 ± 0.025	0.738 ± 0.065
Moon	0.156 ± 0.042	0.857 ± 0.063	0.831 ± 0.066	0.841 ± 0.066	0.683 ± 0.085
Mutagenesis-Bond	0.266 ± 0.021	0.734 ± 0.021	0.281 ± 0.017	0.662 ± 0.040	0.023 ± 0.021
Page	0.154 ± 0.009	0.846 ± 0.009	0.471 ± 0.039	0.869 ± 0.010	0.274 ± 0.035
Pima	0.304 ± 0.030	0.696 ± 0.030	0.690 ± 0.044	0.720 ± 0.030	0.365 ± 0.066
Ring	0.098 ± 0.006	0.902 ± 0.006	0.903 ± 0.006	0.905 ± 0.006	0.805 ± 0.012
Segment	0.194 ± 0.017	0.807 ± 0.017	0.718 ± 0.045	0.864 ± 0.015	0.401 ± 0.041
Thyroid (I)	0.078 ± 0.040	0.922 ± 0.040	0.747 ± 0.148	0.923 ± 0.043	0.695 ± 0.153
Thyroid (II)	0.081 ± 0.034	0.919 ± 0.034	0.754 ± 0.122	0.923 ± 0.035	0.684 ± 0.121
Tic Tac	0.410 ± 0.032	0.590 ± 0.032	0.597 ± 0.039	0.629 ± 0.036	0.172 ± 0.061
			NMC		
Dataset	E	TPR	TNR	P	K
Appendicitis	0.218 ± 0.086	0.782 ± 0.086	0.724 ± 0.167	0.835 ± 0.070	0.423 ± 0.201
Balance	0.267 ± 0.038	0.733 ± 0.038	0.969 ± 0.014	0.925 ± 0.025	0.686 ± 0.034
Banana	0.453 ± 0.019	0.548 ± 0.019	0.552 ± 0.020	0.556 ± 0.020	0.098 ± 0.038
Bands	0.435 ± 0.048	0.565 ± 0.048	0.582 ± 0.055	0.605 ± 0.054	0.135 ± 0.092
Breast Cancer (I)	0.442 ± 0.037	0.558 ± 0.037	0.464 ± 0.046	0.551 ± 0.039	0.022 ± 0.076
Breast Cancer (II)	0.042 ± 0.015	0.973 ± 0.015	0.931 ± 0.032	0.963 ± 0.017	0.908 ± 0.033
Bupa	0.530 ± 0.029	0.470 ± 0.029	0.625 ± 0.030	0.620 ± 0.036	0.066 ± 0.044
Chess	0.307 ± 0.018	0.693 ± 0.018	0.707 ± 0.016	0.714 ± 0.016	0.393 ± 0.033
Gaussian (I)	0.322 ± 0.042	0.679 ± 0.042	0.680 ± 0.043	0.685 ± 0.042	0.355 ± 0.085
Gaussian (II)	0.320 ± 0.032	0.680 ± 0.032	0.588 ± 0.102	0.860 ± 0.032	0.129 ± 0.055
Gaussian (III)	0.530 ± 0.029	0.470 ± 0.029	0.625 ± 0.030	0.620 ± 0.036	0.066 ± 0.044
Hayes-Roth	0.503 ± 0.066	0.497 ± 0.066	0.689 ± 0.063	0.514 ± 0.075	0.180 ± 0.121
Ilpd	0.470 ± 0.037	0.530 ± 0.037	0.757 ± 0.041	0.761 ± 0.037	0.193 ± 0.051
Ionosphere	0.323 ± 0.051	0.677 ± 0.051	0.676 ± 0.051	0.680 ± 0.051	0.351 ± 0.102
Iris	0.110 ± 0.052	0.890 ± 0.052	0.946 ± 0.033	0.904 ± 0.041	0.831 ± 0.087
Iris0	0.023 ± 0.021	0.977 ± 0.021	0.990 ± 0.009	0.980 ± 0.018	0.946 ± 0.050
Liver	0.472 ± 0.048	0.388 ± 0.057	0.891 ± 0.055	0.905 ± 0.045	0.193 ± 0.060
Monk	0.224 ± 0.022	0.776 ± 0.022	0.775 ± 0.022	0.779 ± 0.022	0.550 ± 0.043
Moon	0.234 ± 0.065	0.772 ± 0.089	0.762 ± 0.085	0.771 ± 0.091	0.528 ± 0.130
Mutagenesis-Bond	0.481 ± 0.013	0.519 ± 0.013	0.525 ± 0.029	0.630 ± 0.020	0.034 ± 0.029
Page	0.215 ± 0.013	0.785 ± 0.013	0.205 ± 0.028	0.809 ± 0.014	-0.010 ± 0.024
Pima	0.375 ± 0.033	0.625 ± 0.033	0.546 ± 0.045	0.622 ± 0.037	0.173 ± 0.075
Ring	0.238 ± 0.011	0.763 ± 0.011	0.761 ± 0.011	0.768 ± 0.011	0.524 ± 0.022
Segment	0.311 ± 0.022	0.689 ± 0.022	0.824 ± 0.041	0.870 ± 0.014	0.286 ± 0.038
Thyroid (I)	0.134 ± 0.042	0.867 ± 0.042	0.739 ± 0.150	0.887 ± 0.040	0.545 ± 0.139
Thyroid (II)	0.134 ± 0.048	0.866 ± 0.048	0.777 ± 0.159	0.897 ± 0.046	0.542 ± 0.157
Tic Tac	0.439 ± 0.031	0.561 ± 0.031	0.571 ± 0.042	0.606 ± 0.036	0.119 ± 0.063