**LETTER**

# Implementing a distance-based classifier with a quantum interference circuit

To cite this article: M. Schuld *et al* 2017 *EPL* **119** 60002

View the article online for updates and enhancements.

## Related content

## Recent citations

# Implementing a distance-based classifier with a quantum interference circuit

M. Schuld[1], M. Fingerhuth[1,2] and F. Petruccione[1,3]

[1] *Quantum Research Group, School of Chemistry and Physics, University of KwaZulu-Natal*
*Durban 4000, South Africa*
[2] *Maastricht Science Programme, University of Maastricht - 6200 MD Maastricht, The Netherlands*
[3] *National Institute for Theoretical Physics - KwaZulu-Natal, Durban 4000, South Africa*

**Abstract** – Lately, much attention has been given to quantum algorithms that solve pattern recognition tasks in machine learning. Many of these quantum machine learning algorithms try to implement classical models on large-scale universal quantum computers that have access to non-trivial subroutines such as Hamiltonian simulation, amplitude amplification and phase estimation. We approach the problem from the opposite direction and analyse a distance-based classifier that is realised by a simple quantum interference circuit. After state preparation, the circuit only consists of a Hadamard gate as well as two single-qubit measurements, and computes the distance between data points in quantum parallel. We demonstrate the proof of principle using the IBM Quantum Experience and analyse the performance of the classifier with numerical simulations.

Quantum machine learning, an emerging discipline combining quantum computing with intelligent data mining, witnesses a growing number of proposals for quantum algorithms that solve the problem of supervised pattern recognition [1–5]. In supervised learning, a dataset of labelled inputs or feature vectors is given, and the task is to predict the label of a new feature vector. For example, the inputs could be images of persons, while the label is the name of the person shown in the picture. Image recognition software is then supposed to recognise which person is shown in a previously unseen image. A central question of quantum machine learning is if and how a quantum computer could enhance methods known from machine learning [6]. A large share of the suggested quantum machine learning algorithms are based on the assumption of a large-scale universal quantum computer that can implement non-trivial circuits. This is specifically true for distance-based machine learning models: Quantum algorithms for $k$ nearest neighbour and clustering have been based on extensions of amplitude amplification [7,8] while quantum kernel methods such as support vector machines [1] and Gaussian processes [9] rely on the rather technical routines for quantum matrix inversion [10] or density matrix exponentiation [11]. Experimental implementations are

necessarily limited to demonstrations that concede a vast reduction in complexity of the quantum circuits [12–14]. Most importantly, quantum machine learning remains an enterprise to merely mimic methods from classical machine learning that have been tailor-made for classical computation.

The aim of this letter is to propose a change in perspective: We start with the simplest quantum circuit and show that it can be used as a —likewise simple— model of a classifier. Instead of choosing a textbook machine learning algorithm and asking how to run it on a quantum computer, we turn the question around and ask what classifier can be realised by a minimum quantum circuit. The basic idea is to use quantum interference to evaluate the distance measure of a distance-based classifier in quantum parallel. A similar idea has been investigated by some of the authors in ref. [15], but based on a less powerful information encoding strategy and a more complex circuit.

If an efficient state preparation routine is known, the algorithm explored here harvests the same logarithmic scaling in the dimension and number of the input data that has been claimed by other authors [1,9], but only requires a relatively simple setup that can easily be

implemented on small-scale quantum computing devices available today [16]. Evidently, by using only a single-qubit Hadamard gate this "speedup" is not necessarily based on quantum resources. However, besides the argument we want to make, we envision this to be interesting in situations where quantum states generated by quantum simulations —for example in quantum chemistry— have to be classified coherently. This case is sometimes referred to as "quantum data".

In order to demonstrate the circuit, a simplified supervised pattern recognition task based on the famous Iris flower dataset is solved with the 5-qubit quantum computer provided by the IBM Quantum Experience [17]. Since at the time of writing the interface only allowed an implementation of 80 quantum gates, numerical simulations show that the classifier performs well enough in simple benchmark tasks.

We consider the task of supervised binary pattern classification which can be formalised as follows: Given a training dataset $\mathcal{D} = \{(\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^M, y^M)\}$ of inputs $\mathbf{x}^m \in \mathbb{R}^N$ with their respective target labels $y^m \in \{-1, 1\}$ for $m = 1, \ldots, M$, as well as a new unlabeled input $\tilde{\mathbf{x}} \in \mathbb{R}^N$, find the label $\tilde{y} \in \{-1, 1\}$ that corresponds to the new input. The classifier effectively implemented by the quantum interference circuit together with a thresholding function is given by

$$\tilde{y} = \text{sgn}\left( \sum_{m=1}^{M} y^m \Big[ 1 - \frac{1}{4M} |\tilde{\mathbf{x}} - \mathbf{x}^m|^2 \Big] \right). \quad (1)$$

The distance measure $\kappa(\mathbf{x}, \mathbf{x}') = 1 - \frac{1}{4M}|\mathbf{x} - \mathbf{x}'|^2$ can be interpreted as a kernel (and is in fact very similar to an Epanechnikov kernel [18]), see fig. 1. The model in eq. (1) can be interpreted as a nearest-neighbour classifier in which the labels of all training points contribute towards the prediction, but with an importance that decreases with the squared distance. Another perspective is to view the model as a kernelised binary classifier, $\tilde{y} = \text{sgn}(\sum_m w_m y^m \kappa(\tilde{\mathbf{x}}, \mathbf{x}^m))$ with uniform weights $w_m = 1$ [19]. Such a model can be derived from a perceptron in which the original weights are expressed by an expansion of the training data as motivated by the *representer theorem*, and inner products between inputs are replaced by another kernel function via the so called *kernel trick* [20]. We refer to the model in short as a "distance-based classifier".

The quantum machine learning algorithm that implements the classifier from eq. (1) is based on the idea of encoding the input features into the amplitudes of a quantum system and manipulate them through quantum gates —a strategy responsible for most claims of exponential speedups in quantum-enhanced machine learning. We will refer to this approach as "amplitude encoding" to distinguish it from the more common practice of encoding one bit of information into a qubit. Given a classical vector $\mathbf{x} \in \mathbb{R}^N$, where without loss of generality $N$ is assumed to be the $n$-th power of two, $N = 2^n$ (which can be achieved
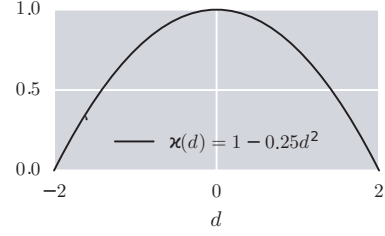


Fig. 1: Plot of the kernel function $\kappa(d) = 1 - \frac{1}{4M}d^2$, $d = |\mathbf{x} - \mathbf{x}'|$ on the interval $[-2, 2]$ and for $M = 1$. The kernel defines the distance measure of the binary classifier considered in this paper. For unit length feature vectors, the maximum distance between two vectors is 2, and the kernel only yields non-negative values.

by padding the vector with zero entries). Furthermore, assume that $\mathbf{x}$ is normalised to unit length, $\mathbf{x}^T\mathbf{x} = 1$. Amplitude encoding associates $\mathbf{x} = (x_1, \ldots, x_N)^T$ with the $2^n$ amplitudes describing the state of a $n$-qubit quantum system, $|\psi_\mathbf{x}\rangle = \sum_{i=0}^{N-1} x_i |i\rangle$. Here, $|i\rangle$ is an *index register* that flags the $i$-th entry of the classical vector with the $i$-th computational basis state.

If one can find an efficient quantum algorithm (*i.e.*, with resources growing polynomially in the number of qubits $n$), one manipulates the $2^n$ amplitudes "super-efficiently" (*i.e.*, with resources growing logarithmically in the dimension of the Hilbert space, $\mathcal{O}(\log N)$). A "super-efficient" algorithm can only maintain its speed if data encoding into a quantum state is also at most polynomial in the number of qubits. There are cases for which this is known to be possible [21,22]. A proposal frequently referred to is a Quantum Random Access Memory [1,9,23] that loads the bit strings representing $x_i$ in parallel into a qubit register and performs a conditional rotation and measurement of an ancilla to write the values into the amplitude. The chance of success of this postselective measurement is only high if the $x_i$ are uniformly close to one.

Using a suitable state preparation scheme, the quantum classification circuit takes a quantum system of $n$ qubits in state

$$|\mathcal{D}\rangle = \frac{1}{\sqrt{2MC}} \sum_{m=1}^{M} |m\rangle \Big( |0\rangle|\psi_{\tilde{\mathbf{x}}}\rangle + |1\rangle|\psi_{\mathbf{x}^m}\rangle \Big) |y^m\rangle. \quad (2)$$

Here, $|m\rangle$ is an index register running from $m = 1, \ldots, M$ and flagging the $m$-th training input. The second register is a single ancilla qubit whose excited state is entangled with the third register encoding the $m$-th training state, $|\psi_\mathbf{x}^m\rangle = \sum_{i=0}^{N-1} x_i^m |i\rangle$, while the ground state is entangled with the third register encoding the new input $|\psi_{\tilde{\mathbf{x}}}\rangle = \sum_{i=0}^{N-1} \tilde{x}_i |i\rangle$. The fourth register is a single qubit, which is zero if $y^m = -1$ and one if $y^m = 1$. Effectively, this creates an amplitude vector which contains the training inputs as well as $M$ copies of the new input. The normalisation constant $C$ depends on the preprocessing of the data. We will assume in the following that the feature vectors are normalised and hence $C = 1$.
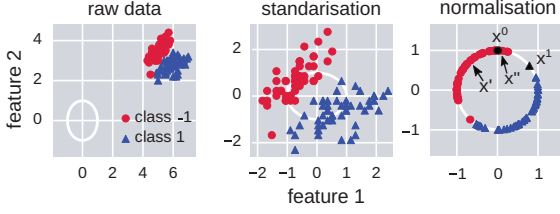
Fig. 2: (Colour online) Data processing illustrated with the example of the first two classes (here called −1 and 1) of the first two features of the Iris dataset. The raw data (left) gets standarised to zero mean and unit variance (center), after which each feature vector is normalised to unit length (right). The training points used in the experiment are marked in black, while the arrows point to the new feature vectors to classify.

After state preparation, the quantum circuit only consists of three operations. First, a Hadamard gate on the ancilla interferes the copies of the new input and the training inputs,

$$\frac{1}{2\sqrt{M}} \sum_{m=1}^{M} |m\rangle \Big( |0\rangle |\psi_{\tilde{\mathbf{x}}+\mathbf{x}^m}\rangle + |1\rangle |\psi_{\tilde{\mathbf{x}}-\mathbf{x}^m}\rangle \Big) |y^m\rangle,$$

where $|\psi_{\tilde{\mathbf{x}}\pm\mathbf{x}^m}\rangle = |\psi_{\tilde{\mathbf{x}}}\rangle \pm |\psi_{\mathbf{x}^m}\rangle$. The second operation is a conditional measurement selecting the branch with the ancilla in state $|0\rangle$. This postselection succeeds with probability $\mathrm{p}_{\mathrm{acc}} = \frac{1}{4M} \sum_m |\tilde{\mathbf{x}} + \mathbf{x}^m|^2$. It is more likely to succeed if the collective squared Euclidean distance of the training set to the new input is small. We will show below that if the data is standarised, postselection usually succeeds with a probability of around 0.5. If the conditional measurement is successful, the result is given by

$$\frac{1}{2\sqrt{M\mathrm{p}_{\mathrm{acc}}}} \sum_{m=1}^{M} \sum_{i=1}^{N} |m\rangle (\tilde{x}_i + x_i^m) |i\rangle |y^m\rangle.$$

The amplitudes weigh the class qubit $|y^m\rangle$ by the distance of the $m$-th data point to the new input. In this state, the probability of measuring the class qubit $|y^m\rangle$ in state 0,

$$\mathrm{p}(\tilde{y} = 0) = \frac{1}{4M\mathrm{p}_{\mathrm{acc}}} \sum_{m|y^m=0} |\tilde{\mathbf{x}} + \mathbf{x}^m|^2,$$

reflects the probability of predicting class −1 for the new input. The choice of normalised feature vectors ensures that $\frac{1}{4M\mathrm{p}_{\mathrm{acc}}} \sum_m |\tilde{\mathbf{x}}+\mathbf{x}^m|^2 = 1 - \frac{1}{4M\mathrm{p}_{\mathrm{acc}}} \sum_m |\tilde{\mathbf{x}}-\mathbf{x}^m|^2$, and choosing the class with the higher probability therefore implements the classifier from eq. (1). The Supplementary Material `Supplementarymaterial.pdf` (SM) shows that the number of measurements needed to estimate $\mathrm{p}(\tilde{y} = 0)$ to error $\epsilon$ with a reasonably high confidence interval grows with $\mathcal{O}(\epsilon^{-2})$.

As a demonstration we implement the interference circuit with the IBM Quantum Experience (IBMQE) [17] using the Iris dataset [24]. Data preprocessing consists of two steps (see fig. 2): We first standarise the dataset to have

zero mean and unit variance. This is common practice in machine learning to compensate scaling effects, and in our case it ensures that the data does not only populate a small subspace of the input space, which in higher dimensions leads to indistinguishably small distances between data points. Second, we need to normalise each feature vector to unit length. This strategy is popular in machine learning —for example with support vector machines— to only consider the angle between data points[1]. This preprocessing strategy allows us to fulfill the conditions of "super-efficient" preprocessing in refs. [22,23] The IBM Quantum Experience enables public use of a processor of five non-error-corrected superconducting qubits based on Josephson junctions located at the IBM Quantum Lab at the Thomas J. Watson Research Center in Yorktown Heights, New York. The current processor has limited connectivity between the five qubits and allows the implementation of 80 gates from a set of 12 single-qubit quantum logic gates as well as a CNOT gate (see SM for details). Due to these limitations, we will only use the first two features of two samples from the Iris dataset for the experimental implementation of the quantum algorithm. Consider the preprocessed training dataset $\mathcal{D}_1 = \{(\mathbf{x}^0, y^0), (\mathbf{x}^1, y^1)\}$ with the two training vectors $\mathbf{x}^0 = (0, 1)$, $y^0 = -1$ (Iris sample 33) and $\mathbf{x}^1 = (0.789, 0.615)$, $y^1 = 1$ (Iris sample 85). In two separate experiments we will consider the classification of two new input vectors of class −1 but with varying distances to the training points, $\tilde{\mathbf{x}}' = (-0.549, 0.836)$ (Iris sample 28) and $\tilde{\mathbf{x}}'' = (0.053, 0.999)$ (Iris sample 36) (see fig. 2).

Implementing this particular classification problem requires four qubits; one qubit for the index register $|m\rangle$ to represent two training vectors, one ancilla qubit, one qubit storing the class of each training instance and one qubit for the data register $|i\rangle$ to represent the two entries of each training and input vector as

$$
\begin{aligned}
|\psi_{\tilde{\mathbf{x}}'}\rangle &= -0.549 |0\rangle + 0.836 |1\rangle, \\
|\psi_{\tilde{\mathbf{x}}''}\rangle &= 0.053 |0\rangle + 0.999 |1\rangle, \\
|\psi_{\mathbf{x}^0}\rangle &= |1\rangle, \\
|\psi_{\mathbf{x}^1}\rangle &= 0.789 |0\rangle + 0.615 |1\rangle.
\end{aligned}
\tag{3}
$$

In this small-scale example, efficient state preparation does not require sophisticated routines as discussed above, but can be designed by hand (see fig. 3). The main idea is to use controlled rotation gates such that the input and training vectors become entangled with the corresponding states of the ancilla and index qubits. Two aspects have to be considered in the quantum circuit design. Firstly, the single and double controlled rotation gates (step B and D in fig. 3) as well as the Toffoli gate (see step C in fig. 3) required for the entanglement of the ancilla and index qubit with the training vectors $\mathbf{x}^0$ and $\mathbf{x}^1$ are not part

---

[1] As an intuition, if we want to classify flowers, some items may have grown bigger than others due to better local conditions, but it is the proportion of the sepal and petal length that is important for the class distinction.
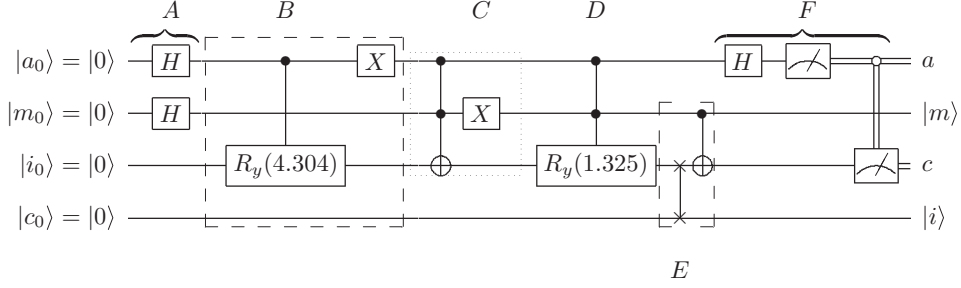
Fig. 3: Quantum circuit implementing the distance-based classifier using the two training vectors $\mathbf{x}^0$ and $\mathbf{x}^1$ and the input vector $\tilde{\mathbf{x}}'$ from the rescaled and normalised Iris flower dataset. First the ancilla and index qubits are put into uniform superposition (step A) and the input vector $\tilde{\mathbf{x}}'$ is entangled with the ground state of the ancilla (step B). Then the training vector $\mathbf{x}^0$ is entangled with the excited state of the ancilla and the ground state of the index qubit (step C) followed by entangling training vector $\mathbf{x}^1$ with the excited state of the ancilla and the index qubit (step D). Next, the data and class qubits are swapped and the class qubit is flipped conditioned on the index qubit being $|1\rangle$ (step E) which completes the initial state preparation. In step F, the Hadamard gate interferes the copies of $\tilde{\mathbf{x}}'$ with the training vectors and the ancilla is measured followed by a measurement of the class qubit (due to prior swapping now at the position of the $|i\rangle$ qubit) when the ancilla was found to be in the $|0\rangle$ state.

Table 1: Classification results for the two-dimensional input vectors $\tilde{\mathbf{x}}'$ and $\tilde{\mathbf{x}}''$ from the Iris flower dataset. Experimental results are always shown on top of their corresponding simulation result (marked with triangle) and theoretical prediction (marked with asterisks).

| Input vector | $p_{\text{acc}}$ | $p(|c\rangle = |0\rangle)$ | $p(|c\rangle = |1\rangle)$ |
|---|---|---|---|
| | 0.455 | 0.516 | 0.484 |
| $\tilde{\mathbf{x}}'$ | 0.731▷ | 0.629▷ | 0.371▷ |
| | 0.729* | 0.629* | 0.371* |
| | 0.494 | 0.589 | 0.411 |
| $\tilde{\mathbf{x}}''$ | 0.911▷ | 0.548▷ | 0.452▷ |
| | 0.913* | 0.547* | 0.453* |

Table 2: Test error for the quantum classifier on different datasets for 1000 random separations into test and training set. Using feature maps leads to a zero classification error in the examples of classes 2 and 3 of the Iris dataset as well as the circles dataset.

| Dataset | Test error | Variance | $p_{\text{acc}}$ |
|---|---|---|---|
| Iris class 1&2 | 0.00 | 0.000 | 0.50 |
| Iris class 1&3 | 0.00 | 0.000 | 0.50 |
| Iris class 2&3 | 0.07 | 0.003 | 0.50 |
| Iris class 2&3, feat. map | 0.00 | 0.000 | 0.50 |
| Circles | 0.62 | 0.006 | 0.50 |
| Circles, feat. map | 0.00 | 0.000 | 0.55 |

of IBM's universal gate set. Therefore, the state preparation routine needs to be mapped to the available hardware by decomposing the controlled rotation, Toffoli and SWAP gates (see SM). Secondly, state preparation for this classification problem requires at least one CNOT operation between qubits that are not directly connected in the hardware. This problem can be solved by exchanging adjacent qubits with a SWAP gate such that the CNOT operation between previously unconnected qubits becomes feasible (see step E in fig. 3).

Using the IBMQE, the resulting quantum circuits were first simulated in an error-free environment and then executed on the non-error-corrected hardware for the maximum number of 8192 runs, and the results are summarised in table 1. As expected the quantum circuits yield simulation results that closely resemble the theoretical predictions, while the experimental results show substantial errors. The main reasons for these deviations are the lack of error correction, the ancilla qubit's short transversal coherence time causing it to deviate from the initial uniform superposition, as well as the class qubit's short longitudinal coherence time. This mostly affects the class qubit in

state $|1\rangle$ storing the class of training vector $\mathbf{x}^1$ which will unavoidably decohere to its ground state. The rapid decoherence of the class qubit is especially troublesome since it makes classification of input vectors that are expected to be classified as class $\tilde{y} = 1$ impossible and, thus, only input vectors of class $\tilde{y} = -1$ are presented in this letter. In those cases, despite the large deviation between theory and experiment, both input vectors $\tilde{\mathbf{x}}'$ and $\tilde{\mathbf{x}}''$ were correctly classified as class $\tilde{y} = -1$. This example demonstrates how mapping a quantum machine learning algorithm to the available hardware requires adaptations to fit the device architecture, and how state preparation, even in this simplified example, constitutes the main bottleneck in the execution of many quantum machine learning algorithms.

In order to analyse the performance of the classifier in more depth, we finally present some numerical simulations. The Iris dataset was randomly divided into a training and test set (ratio 80/20). Table 2 shows the error, or the proportion of misclassified test instances to all test instances, for 1000 repetitions of the random division. The variance of the error is very low in all cases, and the acceptance probability of selecting the correct branch is around 0.5 due to the standarisation of the data. The
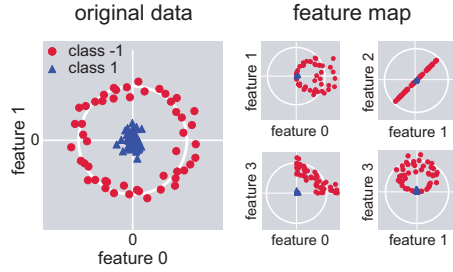
Fig. 4: (Colour online) The "circles" dataset is not suitable for a distanced-based classifier. However, by using two copies of the quantum states that represent feature vectors one can implement polynomial feature maps that increase the power of the classifier. The four plots on the right display the dataset after the feature map, standarisation and normalisation.

results show that running the quantum classifier circuit on the entire dataset for classes 1 and 2 as well as for 1 and 3 of Iris species results in a 100% success rate while distinguishing classes 2 and 3 only yields a 93% average success rate. This is due to the fact that classes 2 and 3 are not linearly separable and cannot be distinguished easily by distance-based methods —especially with a broad kernel that considers almost the entire input space. However, working with normalised feature vectors makes the classifier amenable for the "kernel trick", in which replacing the inner product in the distance by another kernel effectively implements a map on the input space which can vastly improve the power of the classifier [20]. This has also been proposed for quantum machine learning [1,25]. Implementing a polynomial feature map [1] requires to prepare copies of the quantum states which represent the feature vectors, $|\psi_{\mathbf{x}}\rangle \otimes \cdots \otimes |\psi_{\mathbf{x}}\rangle$. Using only two copies of each state allows the classifier to reach a 100% success rate on classes 2 and 3. This trick furthermore allows us to consider datasets which are not based on angles, such as the concentric circles in fig. 4, where the feature map again leads to perfect classification.

The results from experiment and simulations suggest that the distance-based classifier realised by the interference circuit presents an interesting toy model with various potential extensions. Such a model can help to analyse quantum machine learning on real quantum processors, and create models for pattern recognition that are inspired by the strengths of quantum computing. Future work could aim at amending the circuit to realise different kernel functions that allow for more localised measures in order to increase the power and flexibility of the classifier, as well as considering circuits that make more use of quantum resources such as entanglement. Furthermore, while we chose a digital or gate-based framework, one can extend the perspective established in this paper fruitfully to more general quantum devices and ask if their evolution can solve a learning problem (see also refs. [26,27]). We expect this question to become a much more prominent future research topic of quantum machine learning.

* * *

REFERENCES

[1] REBENTROST PATRICK, MOHSENI MASOUD and LLOYD SETH, *Phys. Rev. Lett.*, **113** (2014) 130503.

[2] KAPOOR ASHISH, WIEBE NATHAN and SVORE KRYSTA, *Quantum Perceptron Models*, in *Advances In Neural Information Processing Systems (NIPS 2016), Barcelona, Spain*, Vol. **29** (NIPs Foundation) 2016, pp. 3999–4007.

[3] BENEDETTI MARCELLO, REALPE-GÓMEZ JOHN, BISWAS RUPAK and PERDOMO-ORTIZ ALEJANDRO, *Quantum-assisted learning of graphical models with arbitrary connectivity*, arXiv preprint, arXiv:1609.02542 (2016).

[4] DENCHEV VASIL, DING NAN, NEVEN HARTMUT and VISHWANATHAN SVN, *Robust classification with adiabatic quantum optimization*, in *Proceedings of the 29th International Conference on Machine Learning, ICML, Edinburgh, Scotland, UK* (Omnipress) 2012, pp. 863–870.

[5] SCHULD MARIA, SINAYSKIY ILYA and PETRUCCIONE FRANCESCO, *Phys. Rev. A*, **94** (2016) 022342.

[6] DUNJKO VEDRAN, TAYLOR JACOB M. and BRIEGEL HANS J., *Phys. Rev. Lett.*, **117** (2016) 130501.

[7] WIEBE NATHAN, KAPOOR ASHISH and SVORE KRYSTA, *Quantum Inf. Comput.*, **15** (2015) 0318.

[8] AÏMEUR ESMA, BRASSARD GILLES and GAMBS SÉBASTIEN, *Quantum clustering algorithms*, in *Proceedings of the 24th International Conference on Machine Learning, Corvalis, Oregon, USA* (ACM) 2007, pp. 1–8.

[9] ZHAO ZHIKUAN, FITZSIMONS JACK K. and FITZSIMONS JOSEPH F., *Quantum assisted Gaussian process regression*, arXiv preprint, arXiv:1512.03929 (2015).

[10] HARROW ARAM W., HASSIDIM AVINATAN and LLOYD SETH, *Phys. Rev. Lett.*, **103** (2009) 150502.

[11] LLOYD SETH, MOHSENI MASOUD and REBENTROST PATRICK, *Nat. Phys.*, **10** (2014) 631.

[12] CAI X.-D., WU D., SU Z.-E., CHEN M.-C., WANG X.-L., LI LI, LIU N.-L., LU C.-Y. and PAN J.-W., *Phys. Rev. Lett.*, **114** (2015) 110504.

[13] LI ZHAOKAI, LIU XIAOMEI, XU NANYANG and DU JIANGFENG, *Phys. Rev. Lett.*, **114** (2015) 140504.

[14] LAMATA LUCAS, *Sci. Rep.*, **7** (2017) 1609.

[15] SCHULD MARIA, SINAYSKIY ILYA and PETRUCCIONE FRANCESCO, *Quantum computing for pattern classification*, in *Lect. Notes Comput. Sci.*, Vol. **8862** (Springer Cham, Switzerland) 2014, pp. 208–220.

[16] MOHSENI MASOUD, READ PETER, NEVEN HARTMUT, BOIXO SERGIO, DENCHEV VASIL, BABBUSH RYAN, FOWLER AUSTIN, SMELYANSKIY VADIM and MARTINIS JOHN, *Nature*, **543** (2017) 171.

[17] IBM Quantum Experience, `http://www.research.ibm.com/quantum/` (last accessed: 2017-03-21).

[18] Epanechnikov Vassiliy A., *Theory Probab. Appl.*, **14** (1969) 153.

[19] Aizerman A., Braverman Emmanuel M. and Rozoner L. I., *Autom. Remote Control*, **25** (1964) 821.

[20] Schölkopf Bernhard, *The kernel trick for distances*, in *Proceedings of the 13th International Conference on Neural Information Processing Systems* (MIT Press, Cambridge, Mass., USA) 2000, pp. 283–289.

[21] Grover Lov and Rudolph Terry, *Creating superpositions that correspond to efficiently integrable probability distributions*, arXiv preprint, arXiv:0208112v1 (2002).

[22] Soklakov Andrei N. and Schack Rüdiger, *Phys. Rev. A*, **73** (2006) 012307.

[23] Giovannetti Vittorio, Lloyd Seth and Maccone Lorenzo, *Phys. Rev. Lett.*, **100** (2008) 160501.

[24] Fisher Ronald A., *Ann. Eugen.*, **7** (1936) 179.

[25] Chatterjee Rupak and Yu Ting, *Quantum Inf. Commun.*, **17** (2017) 1292.

[26] Alvarez-Rodriguez Unai, Lamata Lucas, Escandell-Montero Pablo, Martín-Guerrero José D. and Solano Enrique, *Sci. Rep.*, **7** (2017) 13645.

[27] Amin Mohammad H., Andriyash Evgeny, Rolfe Jason, Kulchytskyy Bohdan and Melko Roger, *Quantum Boltzmann machine*, arXiv preprint, arXiv:1601.02036 (2016).