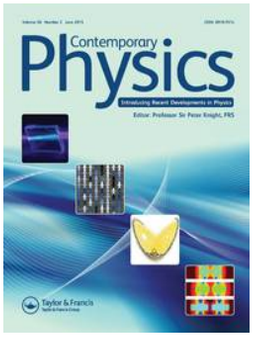

Contemporary Physics

ISSN: 0010-7514 (Print) 1366-5812 (Online) Journal homepage: https://www.tandfonline.com/loi/tcph20

An introduction to quantum machine learning

Maria Schuld, Ilya Sinayskiy & Francesco Petruccione

To cite this article: Maria Schuld, Ilya Sinayskiy & Francesco Petruccione (2015) An introduction to quantum machine learning, Contemporary Physics, 56:2, 172-185, DOI: 10.1080/00107514.2014.964942

To link to this article: https://doi.org/10.1080/00107514.2014.964942

Contemporary Physics, 2015
Vol. 56, No. 2, 172–185, http://dx.doi.org/10.1080/00107514.2014.964942

Taylor & Francis
Taylor & Francis Group

An introduction to quantum machine learning

Maria Schuld[a]*, Ilya Sinayskiy[a,b] and Francesco Petruccione[a,b]

[a] *Quantum Research Group, School of Chemistry and Physics, University of KwaZulu-Natal, Durban 4001, South Africa;* [b] *National Institute for Theoretical Physics (NIThePP), KwaZulu-Natal, South Africa*

(*Received 4 August 2014; accepted 10 September 2014*)

Machine learning algorithms learn a desired input-output relation from examples in order to interpret new inputs. This is important for tasks such as image and speech recognition or strategy optimisation, with growing applications in the IT industry. In the last couple of years, researchers investigated if quantum computing can help to improve classical machine learning algorithms. Ideas range from running computationally costly algorithms or their subroutines efficiently on a quantum computer to the translation of stochastic methods into the language of quantum theory. This contribution gives a systematic overview of the emerging field of quantum machine learning. It presents the approaches as well as technical details in an accessible way, and discusses the potential of a future theory of quantum learning.

Keywords: quantum machine learning; quantum computing; artificial intelligence; machine learning

1. Introduction

Machine learning refers to an area of computer science in which patterns are derived ('learned') from data with the goal to make sense of previously unknown inputs. As part of both artificial intelligence and statistics, machine learning algorithms process large amounts of information for tasks that come naturally to the human brain, such as image and speech recognition, pattern identification or strategy optimisation. These problems gain significant importance in our digital age, an illustrative example being Google's PageRank machine learning algorithm for search engines that was patented by Larry Page in 1997[1] and led to the rise of what is today one of the biggest IT companies in the world. Other important applications of machine learning are spam mail filters, iris recognition for security systems, the evaluation of consumer behaviour, assessing risks in the financial sector or developing strategies for computer games. In short, machine learning comes into play wherever we need computers to interpret data based on experience. This usually involves huge amounts of previously collected input-output data pairs, and machine learning algorithms have to be very efficient in order to deal with so called *big data*.

Since the volume of globally stored data is growing by around 20% every year (currently ranging in the order of several hundred exabytes [1]), the pressure to find innovative approaches to machine learning is rising. A promising idea that is currently investigated by academia as well as in the research labs of leading IT companies exploits the potential of quantum computing in order to optimise classical machine learning algorithms. In the last decades, physicists already demonstrated the impressive power of quantum systems for information processing. In contrast to conventional computers built on the physical implementation of the two states '0' and '1', quantum computers can make use of a qubit's superposition of two quantum states $|0\rangle$ and $|1\rangle$ (e.g. encoded in two distinct energy levels of an atom) in order to follow many different paths of computation at the same time. But the laws of quantum mechanics also restrict our access to information stored in quantum systems, and coming up with quantum algorithms that outperform their classical counterparts is very difficult. However, the toolbox of quantum algorithms is by now fairly established and contains a number of impressive examples that speed up the best known classical methods [2]. The technological implementation of quantum computing is emerging [3], and many believe that it is only a matter of time until the numerous theoretical proposals can be tested on real machines. On this background, the new research field of quantum machine learning might offer the potential to revolutionise future ways of intelligent data processing.

A number of recent academic contributions explore the idea of using the advantages of quantum computing in order to improve machine learning algorithms. For example, some effort has been put into the development of quantum versions [4–6] of artificial neural networks (which are widely

*Corresponding author. Email: schuld@ukzn.ac.za

used in machine learning), but they are often based on a more biological perspective and a major breakthrough has not been accomplished yet [7]. Some authors try to develop entire quantum algorithms that solve problems of pattern recognition [8–10]. Other proposals suggest to simply run subroutines of classical machine learning algorithms on a quantum computer, hoping to gain a speed up [11–13]. An interesting approach is adiabatic quantum machine learning, which seems especially fit for some classes of optimisation problems [14–16]. Stochastic models such as Bayesian decision theory or hidden Markov models find an elegant translation into the language of open quantum systems [17,18]. Despite this growing level of interest in the field, a comprehensive theory of quantum learning, or how quantum information can in principle be applied to intelligent forms of computing, is only in the very first stages of development.

This contribution gives a systematic overview of the emerging field of quantum machine learning, with a focus on methods for pattern classification. After a brief discussion of the concepts of classical and quantum learning in Section 2, the paper is divided into seven sections, each presenting a standard method of machine learning (namely k-nearest neighbour methods, support vector machines, k-means clustering, neural networks, decision trees, Bayesian theory and hidden Markov models) and the various approaches to relate each method to quantum physics. This structure mirrors the still rather fragmented field and allows the reader to select specific areas of interest. As summarised in Figure 1, for k-nearest neighbour methods, support vector machines and k-means clustering, authors are mainly concerned to find efficient calculations of classical distances on a potential quantum computer, while probabilistic methods such as Bayesian theory and hidden Markov models find an analogy in the formalism of open quantum systems. Neural networks and decision trees are still waiting for a convincing quantum version, although especially the former has been a relatively active field of research in the last decade. Finally, in Section 4, we briefly discuss the need for future works on quantum machine learning that concentrate on how the actual *learning* part of machine learning methods can be improved using the power of quantum information processing.

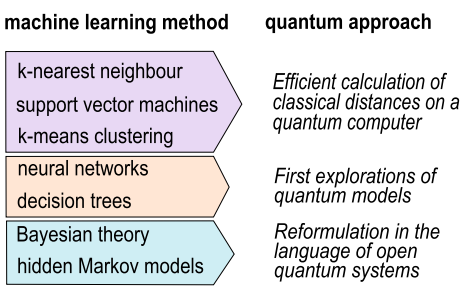

Figure 1. Overview of methods in machine learning and approaches from a quantum information perspective as presented in this paper.

2. Classical and quantum learning

2.1. *Classical machine learning*

The theory of machine learning is an important subdiscipline of both artificial intelligence and statistics, and its roots can be traced back to the beginnings of artificial neural network and artificial intelligence research in the 1950s [19, 20]. In 1959, Arthur Samuel gave his famous definition of machine learning as the 'field of study that gives computers the ability to learn without being explicitly programmed'[2]. This is in fact misleading, since the algorithm itself does not adapt in the learning process, but the function it encodes. In more formal language, this means that the input-output relation of a computer program is derived from a set of training data (which is often very big). Such methods gain importance as computers increasingly interact with humans and have to become more flexible to adapt to our specific needs. A prominent example is a spam mail filter that learns from user behaviour and external databases to classify new spam mails correctly. However, this is only one of many different cases where machine learning intersects with our every-day lives.

In the theory of machine learning, the term *learning* is usually divided into three types (see Figure 2), which help to illustrate the spectrum of the field: *supervised*, *unsupervised* and *reinforcement learning*. In supervised learning, a computer is given examples of correct input-output relations and has to infer a mapping there from. Probably the most important task is *pattern classification*, where vectors of input data have to be assigned to different classes. This might sound like a rather technical problem, but is in fact

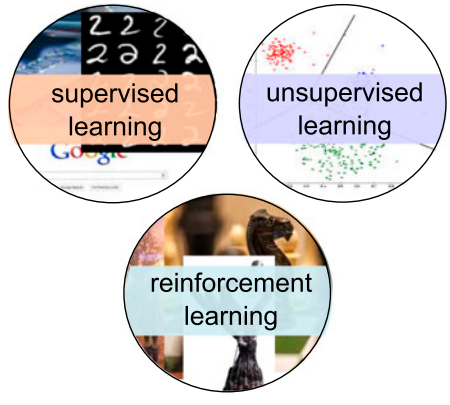

Figure 2. The three types of classical learning. Supervised learning derives patterns from training data and finds application in pattern recognition tasks. Unsupervised learning infers information from the structure of the input and is important for data clustering. Reinforcement learning optimises a strategy due to feedback by a reward function, and usually applies to intelligent agents and games.

something humans do continuously – for example, when we recognise a face from different angles and light conditions as belonging to one and the same person, or when we classify signals from our sensory organs as dangerous or not. We could even go so far and say that pattern classification is the abstract description of 'interpreting' input coming from our senses. It is no surprise that a big share of machine learning research tries to imitate this remarkable ability of human beings with computers, and there is an entire zoo of algorithms that generalise from large training data-sets how to classify new input.

The second category, *unsupervised learning*, has not been part of machine learning for a long time, as it describes the process of finding patterns in data without prior experience or examples. A prominent task is data clustering, or forming subgroups out of a given data-set, in order to summarise large amounts of information by only a few stereotypes. This is, for example, an important problem in sociological studies and market research. Note that this task is closely related to classification, since clustering means effectively to assign a class to each vector of a given set, but without the goal of treating new inputs.

Finally, *reinforcement learning* is the closest to what we might associate with the expression 'learning'. Given a framework of rules and goals, an agent (usually a computer program that acts as a player in a game) gets rewarded or punished depending on which strategy it uses in order to win. Each reward reinforces the current strategy, while punishment leads to an adaptation of its policy [22,23]. Reinforcement learning is a central mechanism in the development and study of intelligent agents. However, it will not be in the focus of this paper, and it differs in many regards from the other two types of learning. Investigations into quantum games and quantum intelligent agents are diverse and numerous (see e.g. [24–28]), and shall be treated elsewhere.

Even within these categories, the expression 'learning' can relate to different procedures. For example, it may refer to a training phase in which optimal parameters of an algorithm (e.g. weights, initial states) are obtained. This is done by presenting examples of correct input-output-relations to a task, and adapting the parameters to reproduce these examples. The training set is then discarded [29]. An illustrative case close to human learning is the weight adjustment process in artificial neural networks through backpropagation or deep learning [30,31]. Training phases are often the most costly part of a machine learning algorithm, and efficient training methods become especially important when dealing with so-called *big data*. Besides learning as a parameter optimisation problem, there is a large number of machine learning algorithms that do not have an explicit learning phase. For example, if presented with an unclassified input vector, the k-nearest neighbour for pattern classification uses the training data to decide upon its classification. In this case, learning is not a parameter optimisation problem, but rather a decision function inferred from examples. In reinforcement learning, this decision function becomes a full *strategy*, and learning refers to the adaptation of the strategy to increase the chances of future reward.

Whatever type and procedure of learning is chosen, optimal machine learning algorithms run with minimum resources and have a minimum error rate related to the task (as indicated by misclassification of input, poor division into clusters, little reward of a strategy). Challenges lie in the problem of finding parameters and initial values that lead to an optimal solution, or to come up with schemes that reduce the complexity class of the algorithm.[3] This is where quantum computing promises to help.

2.2. *Quantum machine learning*

Quantum computing refers to the manipulation of quantum systems in order to process information. The ability of quantum states to be in a superposition can thereby lead to a substantial speed up of a computation in terms of complexity, since operations can be executed on many states at the same time. The basic unit of quantum computation is the qubit, $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ (with $\alpha, \beta \in \mathbb{C}$ and $|0\rangle, |1\rangle$ in the two-dimensional Hilbert space $\mathcal{H}^2$). The absolute squares of the amplitudes are the probability to measure the qubit in the 0 or the 1 state, and quantum dynamics always maintain the property of probability conservation given by $|\alpha|^2 + |\beta|^2 = 1$. In mathematical language, this means that transformations that map quantum states onto other quantum states (so called *quantum gates*) have to be unitary. Through single qubit quantum gates, we are able to manipulate the basis state, amplitude or phase of a qubit (for example, through the so called X-gate, the Z-gate and the Y-gate, respectively), or put a qubit with $\beta = 0$ ($\alpha = 0$) into an equal superposition $\alpha = \beta = {}^1/\sqrt{2}$ ($\alpha = {}^1/\sqrt{2}, \beta = -{}^1/\sqrt{2}$) (the Hadamard or H-gate). Multi-qubit gates are often based on controlled operations that execute a single qubit operation only if another (ancilla or control qubit) is in a certain state. One of the most important gates is the two qubit XOR-gate, which flips the basis state of the second qubit in case the first qubit is in state $|1\rangle$. A two-qubit gate that will be mentioned later is the SWAP-gate exchanging the state of two qubits with each other.

Quantum gates are usually expressed as unitary matrices (see also Figure 3). The matrices operate on 2^n-dimensional vectors that contain the amplitudes of the 2^n basis states of a n-dimensional quantum system. For example, the XOR-gate working on the quantum state $|\psi\rangle = {}^1/\sqrt{2}\,(|00\rangle + |11\rangle)$ would look like

$$\begin{pmatrix} 1&0&0&0\\ 0&1&0&0\\ 0&0&0&1\\ 0&0&1&0 \end{pmatrix} \cdot \frac{1}{\sqrt{2}} \begin{pmatrix} 1\\0\\0\\1 \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1\\0\\1\\0 \end{pmatrix},$$

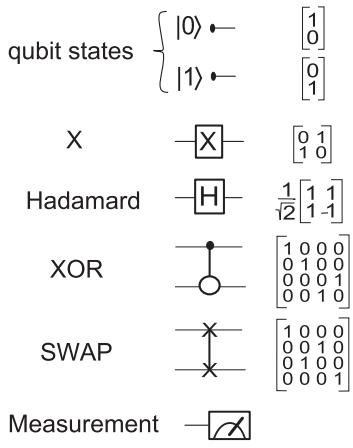

Figure 3. Representation of qubit states, unitary gates and measurements in the quantum circuit model and in the matrix formalism.

and produce $|\psi'\rangle = {}^1/\sqrt{2}\,(|00\rangle + |10\rangle)$. The art of developing algorithms for a potential quantum computer is to use such elementary gates in order to create a quantum state that has a relatively high amplitude for states that represent solutions for the given problem. A measurement in the computational basis then produces such a desired result with a relatively high probability. Quantum algorithms are usually repeated a number of times since the result is always probabilistic. For a comprehensive introduction into quantum computing, we refer to the standard textbook by Nielsen and Chuang [2] as well as Plenio and Vitelli's presentation of the concept of quantum information [32].

In quantum machine learning, quantum algorithms are developed to solve typical problems of machine learning using the efficiency of quantum computing. This is usually done by adapting classical algorithms or their expensive subroutines to run on a potential quantum computer. The expectation is that in the near future, such machines will be commonly available for applications and can help to process the growing amounts of global information. The emerging field also includes approaches vice versa, namely well-established methods of machine learning that can help to extend and improve quantum information theory.

As mentioned before, there is no comprehensive theory of quantum learning yet. Discussions of elements of such a theory can be found in [33–35]. Following the remarks above, a theory of quantum learning would refer to methods of quantum information processing that learn input-output relations from training input, either for the optimisation of system parameters (for example unitary operators, see [36]) or to find a 'quantum decision function' or 'quantum strategy'. There are many open questions of how an efficient quantum learning procedure could look like. For example, how can we efficiently implement an optimisation problem (that is usually solved by iterative and dissipative methods such as gradient descent) on a coherent and thus reversible quantum computer? How can we translate and process important structural information, such as distance metrics, using quantum states? How do we formulate a decision strategy in terms of quantum physics? And the overall question, is there a general way how quantum physics can in principle speed up certain problems of machine learning?

An underlying question is also the representation of classical data by quantum systems. The most common approach in quantum computing is to represent classical information as binary strings $(x_1, \ldots, x_n)$ with $x_i \in \{0, 1\}$ for $i = 1, \ldots, n$, that are directly translated into n-qubit quantum states $|x_1 \ldots x_n\rangle$ from a 2^n-dimensional Hilbert space with basis $\{|0\ldots.00\rangle, |0\ldots.01\rangle, \ldots, |1\ldots.11\rangle\}$, and to read information out through measurements. However, existing machine learning algorithms are often based on an internal structure of this data, for example, the Euclidean distance as a similarity measure between two examples of features. Alternative data representations have been proposed by Seth Lloyd and his co-workers, who encode classical information into the norm of a quantum state, $\langle x|\, x\rangle = |\vec{x}|^{-1}\vec{x}^2$, leading to the definition [11,12]

$$|x\rangle = |\vec{x}|^{-1/2}\vec{x}. \tag{1}$$

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.

3. Quantum versions of machine learning algorithms

Before proceeding to the discussion of classical machine learning algorithms and their quantum counterparts, we have to take a look on the actual problems these methods intend to solve, as well as introduce the formalism used throughout this article. Probably the most important application is the task of *pattern classification*, and there are many different classical algorithms tackling this problem. Based on a set of training examples consisting of feature vectors[4] and their respective class attributes, the computer has to correctly classify an unknown feature vector. For example, the feature vector could contain preprocessed information on patients and their correctly diagnosed disease. A machine learning algorithm then has to find the correct disease of a new patient. More precisely, given a training set $\mathcal{T} = \{\vec{v}^p, c^p\}_{p=1,\ldots,N}$ of N n-dimensional feature vectors $\vec{v}$ and their respective class c^p, as well as a new n-dimensional input vector $\vec{x}$, we have to find the class c^x of vector $\vec{x}$. Closely related to pattern classification are other tasks such as *pattern completion* (adding missing information to an incomplete input), *associative memory* (retrieving one of a number of stored memory vectors upon an input) or *pattern recognition* (including finding and examining the shape of patterns; this term is often used as a synonym to pattern classification).

The central problem of unsupervised learning is clustering data. Given a set of feature vectors $\{\vec{v}^p\}$, the goal is to assign each vector to one out of k different clusters so that similar inputs share the same assignment. Other problems of machine learning concern optimal strategies in terms of an unknown reward function, given a set of consecutive observations of choices and consequences. As stated above, we will not concentrate on the learning of strategies here.

3.1. *Quantum versions of k-nearest neighbour methods*

A very popular and simple standard textbook method for pattern classification is the k-nearest neighbour algorithm. Given a training set $\mathcal{T}$ of feature vectors with their respective classification as well as an unclassified input vector $\vec{x}$, the idea is to choose the class c^x for the new input that appears most often amongst its k-nearest neighbours (see Figure 4). This is based on the assumption that 'close' feature vectors encode similar examples, which is true for many applications. Common distance measures are thereby the inner product, the Euclidian or the Hamming distance[5]. Choosing k is not always easy and can influence the result significantly. If k is chosen too big, we loose the locality information and end up in a simple majority vote over the entire training set, while a very small k leads to noise-biased results. A variation of the algorithm suggests not to run it on the training set, but to calculate the means or centroid $1/N_c \sum_p \vec{v}^p$ of all N_c vectors belonging to one class c beforehand, and to select the class of the nearest centroid (we call this here the nearest-centroid algorithm). Another variation weights the influence of the neighbours by distance, gaining an independence of the parameter k (the weighted nearest neighbours algorithm [38]). Methods such as k-nearest neighbours are obviously based on a distance metric to evaluate the similarity of two feature vectors. Efforts to translate this algorithm into a quantum version, therefore, focus on the efficient evaluation of a classical distance through a quantum algorithm.

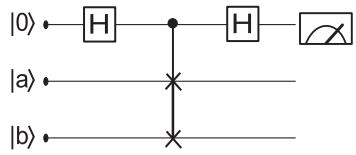

Figure 5. Quantum circuit representation of a swap test routine.

Aïmeur et al. [39] introduce the idea of using the overlap or fidelity $|\langle a|\ b\rangle|$ of two quantum states $|a\rangle$ and $|b\rangle$ as a 'similarity measure'. The fidelity can be obtained through a simple quantum routine sometimes referred to as a *swap test* [40] (see Figure 5). Given a quantum state $|a, b, 0_{\text{anc}}\rangle$ containing the two wavefunctions as well as an ancilla register initially set to 0, a Hadamard transformation sets the ancilla into a superposition $1/\sqrt{2}(|0\rangle + |1\rangle)$, followed by a controlled swap gate on a and b which swaps the two states under the condition that the ancilla is in state $|1\rangle$. A second Hadamard gate on the ancilla results in state $|\psi_{SW}\rangle = \frac{1}{2}|0\rangle\,(|a, b\rangle + |b, a\rangle) + \frac{1}{2}|1\rangle\,(|a, b\rangle - |b, a\rangle)$ for which the probability of measuring the ground state is given by

$$P(|0_{\text{anc}}\rangle) = \frac{1}{2} + \frac{1}{2}|\langle a|\ b\rangle|^2\,. \tag{2}$$

A probability of $1/2$ consequently shows that the two quantum states $|a\rangle$ and $|b\rangle$ do not overlap at all (in other words, they are orthogonal), while a probability of 1 indicates that they have maximum overlap.

Based on the swap test, Lloyd et al. [11] recently proposed a way to retrieve the distance between two real-valued n-dimensional vectors $\vec{a}$ and $\vec{b}$ through a quantum measurement. More precisely, the authors calculate the inner product of the ancilla of state $|\psi\rangle = \frac{1}{\sqrt{2}}(|0, a\rangle + |1, b\rangle)$ with the state $|\phi\rangle = \frac{1}{\sqrt{Z}}(|\vec{a}|\,|0\rangle - |\vec{b}|\,|1\rangle)$ (with $Z = |\vec{a}|^2 + |\vec{b}|^2$), evaluating $|\langle\phi|\ \psi\rangle|^2$ as part of a swap test. This looks complicated, but is first of all an inexpensive procedure since the states $|\phi\rangle$ and $|\psi\rangle$ can be efficiently prepared

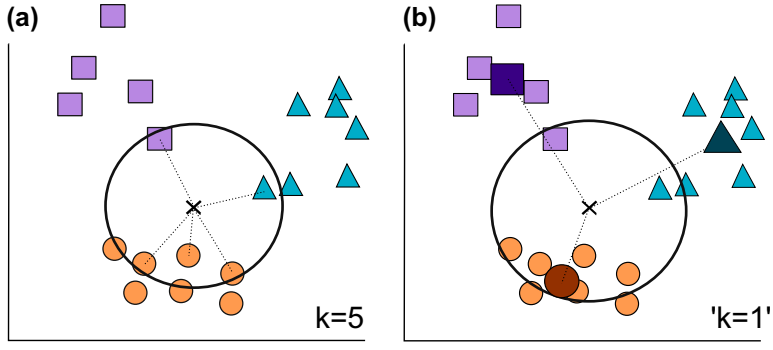

Figure 4. (*a*): Illustration of the kNN method of pattern classification. The new vector (black cross) gets assigned to the class that the majority of its k-closest neighbours have (in this case it would be the orange circle shape). (*b*): A variation is the nearest-centroid method in which the closest mean vector of a class of vectors defines the classification of a new input. This can be understood as a k-nearest neighbour method with preprocessed data and $k = 1$.

[11]. The trick lies in the clever definition of a quantum state given in Eq. (1), which encodes the classical length of a vector $\vec{x}$ into the scalar product of the quantum state with itself, $\langle x|\, x\rangle = |\vec{x}|^{-1}|\vec{x}|$. With this definition, the identity $|\vec{a}-\vec{b}|^2 = Z\,|\langle\phi|\,\psi\rangle|^2$ holds true. The classical distance between two vectors $\vec{a}$ and $\vec{b}$ can consequently be retrieved through a simple quantum swap test of carefully constructed states. Lloyd, Mohseni and Rebentrost use this procedure for a quantum version of the nearest-centroid algorithm. With $\vec{a} \equiv \vec{x}$ and $\vec{b} \equiv \frac{1}{N_c}\sum_p \vec{v}^p$, they propose to calculate the classical distance from the new input to a given centroid, $|\vec{x} - \frac{1}{N_c}\sum_p \vec{v}^p|$, through the above described procedure. The authors claim that even when considering the operations to construct the quantum states involved, this quantum method is more efficient than the polynomial runtime needed to calculate the same value on a classical computer.

Wiebe et al. [13] also use a swap test in order to calculate the inner product of two vectors, which is another distance measure between feature vectors. However, they use an alternative representation of classical information through quantum states. Given n-dimensional classical vectors $\vec{a}, \vec{b}$ with entries $a_j = |a_j|e^{i\alpha_j}, b_j = |b_j|e^{i\beta_j}$, $j = 1,\ldots,n$ as well as an upper bound $r_{\max}$ for the entries of the training vectors in $\mathcal{T}$ and an upper bound for the number of zeros in a vector d (the sparsity), the idea is to write the parameters into amplitudes of the quantum states $|A\rangle = \frac{1}{\sqrt{d}}\sum_j |j\rangle\,(\sqrt{1-\frac{|a_j|^2}{r_{\max}^2}}e^{-i\alpha_j}\,|0\rangle + \frac{a_j}{r_{\max}}\,|1\rangle)\,|1\rangle$ and $|B\rangle = \frac{1}{\sqrt{d}}\sum_j |j\rangle\,|1\rangle\,(\sqrt{1-\frac{|b_j|^2}{r_{\max}^2}}e^{-i\beta_j}\,|0\rangle + \frac{b_j}{r_{\max}}\,|1\rangle)$ and perform a swap test on $|A\rangle$ and $|B\rangle$. According to Eq. (2), the probability of measuring the swap test ancilla in the ground state is then $P(|0\rangle_{\text{anc}}) = \frac{1}{2} + \frac{1}{2}|\frac{1}{dr_{\max}^2}\sum_i a_i b_i|^2$ and the inner product of $\vec{a}, \vec{b}$ can consequently be evaluated by $|\sum_i a_i b_i|^2 = d^2 r_{\max}^4\,(2P(|0\rangle_{\text{anc}}) - 1)$, which is altogether independent of the dimension n of the vector. The authors in fact claim a quadratic speed-up compared with classical algorithms. In the same contribution, Wiebe, Kapoor and Svore also give a scheme for a (weighted) nearest-centroid algorithm based on the Euclidian distance evaluated by well-known algorithms from the toolbox of quantum information, the amplitude estimation algorithm [41] and Dürr and Høyer's *find_minimum* subroutine [42].

A full quantum pattern recognition algorithm for binary features was presented by Trugenberger [9]. He expands his quantum associative memory circuit [43] for this purpose. At the centre is his subroutine to measure the Hamming distance between two binary quantum states. He constructs a quantum superposition containing all states of the quantum training set, and writes the Hamming distance to the binary input vector $|x\rangle = |x_1 \ldots x_n\rangle\,, x_i = \{0,1\}$ into the amplitude of each training vector state. This is done by the following useful routine based on elementary quantum operations. Given two binary strings $|a_1 \ldots a_n\rangle$ and $|b_1 \ldots b_n\rangle$ with entries $a_i, b_i \in \{0,1\}$, we construct the initial state $|\psi\rangle = |a_1 \ldots a_n, b_1 \ldots b_n\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, consisting of two registers for the qubits of a and b, respectively, as well as an extra 2-dimensional ancilla register in superposition. The inverse Hamming distance between each qubit of the first and second register,

$$\bar{d}_k = \begin{cases} 0, & \text{if } |a_k\rangle = |b_k\rangle\,, \\ 1, & \text{else,} \end{cases}$$

replaces the respective qubit in the second register. This is done by applying an $\text{XOR}_{a,b}$ gate which overwrites the second entry b_k with 0 if $a_k = b_k$ and else with 1, as well as a NOT gate. The result is the state

$$\left|\psi'\right\rangle = \left|a_1 \ldots a_n, \bar{d}_1 \ldots \bar{d}_n\right\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

To write the total Hamming distance $\bar{d}_H(\vec{a}, \vec{b})$ first into the phase and then into the amplitude, Trugenberger uses the unitary operator $U = \exp(-i\frac{\pi}{2n}H)$ with $H = 1 \otimes \sum_k (\frac{1}{2}(\sigma_z + 1))_{d_k} \otimes \sigma_z$ working on the three registers. Note that this adds a negative sign in case the ancilla qubit is in $|1\rangle$. A Hadamard transformation on the ancilla state, $H_{\text{anc}} = 1 \otimes 1 \otimes H$, consequently results in

$$\begin{aligned}\left|\psi''\right\rangle = &\cos\left[\frac{\pi}{2n}\bar{d}_H(\vec{a}, \vec{b})\right]\left|a_1 \ldots a_n, \bar{d}_1 \ldots \bar{d}_n, 0\right\rangle \\ &+ \sin\left[\frac{\pi}{2n}\bar{d}_H(\vec{a}, \vec{b})\right]\left|a_1 \ldots a_n, \bar{d}_1 \ldots \bar{d}_n, 1\right\rangle.\end{aligned}$$

Measuring the ancilla in $|0\rangle$ leads to a state in which the amplitude scales with the Hamming distance between $\vec{a}$ and $\vec{b}$. Of course, the power of this routine only becomes visible if it is applied to a large superposition of training states in the first register $|a_1, \ldots, a_n\rangle \to \sum_p |v^p\rangle$. A clever measurement then retrieves the states close to the input state with a high probability.

3.2. *Quantum computing for support vector machines*

A support vector machine is used for *linear discrimination*, which is a subcategory of pattern classification. The task in linear discrimination problems is to find a hyperplane that is the best discrimination between two class regions and serves as a decision boundary for future classification tasks. In a trivial example of one-dimensional data and only two classes, we would ask which point x lies exactly between the members of class 1 and 2, so that all values left of x belong to one class and all values right of x to the other. In higher dimensions, the boundary is given by a hyperplane (see Figure 6 for two dimensions). It seems like a severe restriction that methods of linear discrimination require the problem to be linearly separable, which means that there is a hyperplane that divides the datapoints so that all vectors of either class are on one side of the hyperplane (in other words, the regions of each class have to be disjunct). However, a nonseparable problem can be mapped onto a linearly separable problem by increasing the dimensions [22].

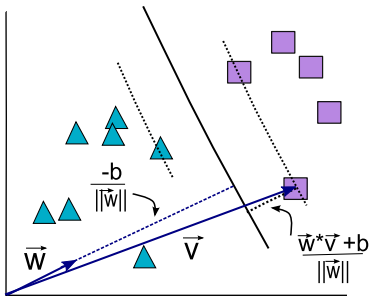

Figure 6. A support vector machine finds a hyperplane (here a line) with maximum margin to the closest vectors. This image illustrates the geometry of the optimisation problem based on [29].

A support vector machine tries to find the optimal separating hyperplane. The best discriminating hyperplane has a maximum distance to the closest datapoints, the so called support vectors. This is a mathematical optimisation problem of finding the maximum margin $|\vec{w}|^{-1}$ $(\vec{v}\vec{w} + b)$ between the hyperplane and the support vectors [29] (see Figure 6). In the two-dimensional case, the boundary conditions are

$$\begin{aligned} \vec{w}\vec{v}_i + b \geq 1, \quad & \text{when } c_i = 1, \\ \vec{w}\vec{v}_i + b \leq -1, \quad & \text{when } c_i = -1, \end{aligned} \tag{3}$$

for each support vector $\vec{v}_i$ from the training data-set and its classification $c_i \in \{-1, 1\}$. This means that while finding a maximum margin, the hyperplane must still separate the training vectors of the two classes correctly. This optimisation problem can be formulated using the Langrangian method [22] or in dual space [44].

Without going into the complex mathematical details of support vector machines, it is important to note that the mathematical formulation of the optimisation problem contains a kernel K, a matrix containing the inner product of the feature vectors $(K)_{pk} = \vec{v}_p \cdot \vec{v}_k$, $p, k = 1, \ldots, N$ (or the basis vectors they are composed of) as entries. Support vector machines are in fact part of a larger class of so-called kernel methods [29] (for more details see [22]) that suffer from the fact that calculating kernels can get very expensive in terms of computational resources. More precisely, quadratic programming problems of this form have a complexity of $O((Nn)^3)$ [29], where Nn is the number of variables involved, and computational resources, therefore, grow significantly with the size of the training data. It is, thus, crucial for support vector machines to find a method of evaluating an inner product efficiently. This is where quantum computing comes into play.

Rebentrost et al. [12] claim that, in general, the evaluation of an inner product can be done faster on a quantum computer. Given the quantum state[6] $|\chi\rangle = {}^1/\sqrt{N_\chi} \sum_{i=1}^{2^n} |\vec{x}_i|\,|i\rangle \left|x^i\right\rangle$, with $N_\chi = \sum_{i=1}^{2^n} |\vec{x}^i|^2$. The $\left|x^i\right\rangle$ are a 2^n-dimensional basis of the training vector space $\mathcal{T}$, so that every training vector $|v^p\rangle$ can be represented as a superposition $|v^p\rangle = \sum \alpha_i \left|x^i\right\rangle$. Similar to the same authors' distance measurement given in Eq. (1), the quantum evaluation of a classical inner product relies on the fact that the quantum states are normalised as

$$\left\langle x^j \middle| x^i \right\rangle = \frac{\vec{x}^i \cdot \vec{x}^j}{|\vec{x}^i||\vec{x}^j|}.$$

The kernel matrix of the inner products of the basis vectors, K with $(K)_{i,j} = \vec{x}^i \cdot \vec{x}^j$, can then be calculated by taking the partial trace of the corresponding density matrix $|\chi\rangle\langle\chi|$ over the states $\left|x^i\right\rangle$,

$$\mathrm{tr}_x[|\chi\rangle\langle\chi|] = \frac{1}{N_\chi} \sum_{i,j=1}^{2^n} \underbrace{\left\langle x^j \middle| x^i \right\rangle |\vec{x}^j||\vec{x}^i|}_{\vec{x}^j \cdot \vec{x}^i} |\,j\rangle\langle i| = \frac{\hat{K}}{\mathrm{tr}[K]}\,.$$

Rebentrost, Mohseni and Lloyd propose that the inner product evaluation can not only be used for the kernel matrix, but also when a pattern has to be classified, which invokes the evaluation of the inner product between the above parameter vector $\vec{w}$ and the new input (see Eq. 3).[7]

3.3. *Quantum algorithms for clustering*

Clustering describes the task of dividing a set of unclassified feature vectors into k subsets or clusters. It is the most prominent problem in unsupervised learning, which does not use training sets or 'prior examples' for generalisation, but rather extracts information on structural characteristics of a data set. Clustering is usually based on a distance measure such as the squared Euclidean distance $((\vec{a} - \vec{b})^2$ with $\vec{a}, \vec{b} \in \mathbb{R}^N)$.

The standard textbook example for clustering is the k-means algorithm, in which alternately each feature vector or datapoint is assigned to its closest current centroid vector to form a cluster for each centroid, and the centroid vectors get calculated from the clusters of the previous step (see Figure 7). Of course, the first iteration requires initial choices for the centroid vectors, and a free parameter is the number k of clusters to be formed. The procedure eventually converges to stable centroid positions. However, these may represent local minima, as only the position of the initial centroids defines whether a global minima can be reached [47]. Other problems of k-means clustering are how to choose the parameter k without prior knowledge of the data, and how to deal with clusters that are visibly not grouped according to distance measures (such as concentric circles). Still, k-means works well for many simple applications of reducing many datapoints into only a few groups, for example, in data compression tasks. A variation of the k-means algorithm is the k-median clustering, in which the role of the centroid is taken over by the datapoint of a cluster that has the smallest total distance to all other points.

Besides versions of quantum clustering that are merely *inspired* by quantum mechanics [48] or use the quantum mechanical fidelity $\mathrm{Fid}(|\psi\rangle, |\phi\rangle) = |\langle\psi|\,\phi\rangle|^2$ as a distance

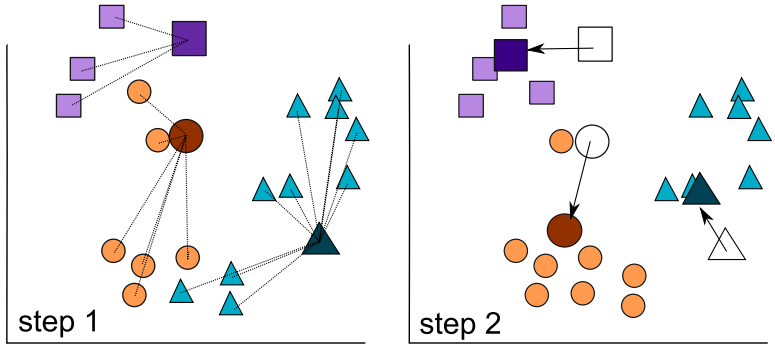

Figure 7. The alternating steps of a k-means algorithm. *Step 1*: The clusters (different shapes and colours) are defined by attributing each vector to the closest centroid vector (larger and darker shapes). *Step 2*: The centroids of each cluster defined in the previous cycle are recalculated and define a new clustering.

measure for an otherwise classical algorithm [39], several full quantum routines for clustering have been proposed. For example, Aïmeur et al. [49] use two subroutines for a quantum k-median algorithm. First, with the help of an oracle that calculates the distance between two quantum states, the total distance of each state to all other states of one cluster is calculated. Based on the *find_minimum* subroutine in [42], the authors then describe a routine to find the smallest value of this distance function and select the according quantum state as the new median for the cluster. Unfortunately, the oracle is not described in detail, and their quantum machine learning proposal largely depends on how and with what resources it can be implemented.

In their contribution discussed earlier, Lloyd et al. [11] present an unsupervised quantum learning algorithm for k-means clustering that is based on adiabatic quantum computing. Adiabatic quantum computing is an alternative to the above introduced method of implementing unitary gates, and tries to continuously adjust the quantum system's parameters in an adiabatic process in order to transfer a ground state which is easy to prepare into a ground state which encodes the result of the computation. Although not in focus here, quantum adiabatic computing seems to be an interesting candidate for quantum machine learning methods [15]. This is why we want to sketch the idea of how to use adiabatic quantum computing for k-means clustering.

In [11], the goal of each clustering step is to have an output quantum superposition $|\chi\rangle = {}^1/\sqrt{N_c} \sum_{c,\, p\in c} |c\rangle\, |\vec{v}^{\,p}\rangle$, where as usual $\{|v^p\rangle\}_{p=1,\ldots,N}$ is the set of N feature vectors or datapoints expressed as quantum states, and $|c\rangle$ is the cluster the subset $\{|v^j\rangle\}_{j=1,\ldots,N_c}$ is assigned to after the clustering step. The authors essentially propose to adiabatically transform an initial Hamiltonian $H_0 = 1 - \frac{1}{k} \sum_{c,c'} |c\rangle\langle c'|$ into a Hamiltonian

$$H_1 = \sum_{c',j} |\vec{v}^{\,p} - \vec{\bar{v}}_{c'}|^2 |c'\rangle\langle c'| \otimes |j\rangle\langle j|,$$

encoding the distance between vector $\vec{v}^{\,p}$ to the centroid of the closest cluster $\vec{\bar{v}}_c$. They give a more refined version and also mention that the adiabatic method can be applied to solve the optimisation problem of finding good initial or 'seed' centroid vectors.

3.4. *Searching for a quantum neural network model*

An artificial neural network is a n-dimensional graph where the nodes x_m are called neurons and their connections are weighted by parameters w_{ml} representing synaptic strengths between neurons $(m, l = 1, \ldots, n)$. An activation function defines the value of a neuron depending on the current value of all other neurons weighted by the parameters w_{ml}, and the dynamics of the neural network is given by successively updating the value of neurons through the activation function. An artificial neural network can, thus, be understood as a computational device, the input being the initial values of the neurons and the output either a stable state of the entire network or the state of a specific subset of neurons. 'Programming' a neural network can be done by selecting weight parameters w_{ml} and an activation function encoding a certain input-output relation. The power of artificial neural networks lies in the fact that they can learn their weights from training data, a fact that neuroscientists believe is the basic principle of how our brain processes information [50].

For pattern classification, we usually consider so called feed-forward neural networks in which neurons are arranged in layers, and each layer feeds its values into the next layer. An input is presented to a feed-forward neural network by initialising the input layer, and after each layer successively updates its nodes, the output (for example encoding the classification of the input) can be read out in the last layer (see Figure 8).

Feed-forward neural networks often use sigmoid activation functions

$$x_l = \mathrm{sgm}\left(\sum_{m=1}^{N} w_{ml} x_m; \kappa\right),$$

defined by $\mathrm{sgm}(a;\kappa) = (1 + \mathrm{e}^{-\kappa a})^{-1}$. If an appropriate set of weight parameters is given, feed-forward neural

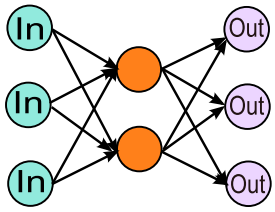

Figure 8. Illustration of a feed-forward neural network with a sigmoid activation function for each neuron.

networks are able to classify input patterns extremely well. To evoke the desired generalisation, the network is initialised with training vectors, the output is compared with the correct output and the weights adjusted through gradient descent in order to minimise the classification error. The procedure is called backpropagation [51]. A challenge for pattern classification with neural networks is the computational cost for the backpropagation algorithm, even when we consider improved training methods such as deep learning [30].

There are a number of proposals for quantum versions of neural networks. However, most of them consider another class, so called Hopfield networks, which are powerful for the related task of associative memory that is derived from neuroscience rather than machine learning. A large share of the literature on quantum neural networks tries to find specific quantum circuits that integrate the mechanisms of neural networks in some way [6,52–54], trying to use the power of neural computing for quantum computation. A practical implementation is given by Elizabeth Behrman [55–57] who uses interacting quantum dots to simulate neural networks with quantum systems. An interesting approach is also to use fuzzy feed-forward neural networks inspired by quantum mechanics [58] to allow for multi-state neurons. Also worth mentioning is the pattern recognition scheme implemented through adiabatic computing with liquid-state nuclear magnetic resonance [16]. Despite this rich body of ideas, there is no quantum neural network proposal that delivers a fully functioning efficient quantum pattern classification method that the authors know of. However, it is an interesting open challenge to translate the nonlinear activation function into a meaningful quantum mechanical framework [7], or to find learning schemes based on quantum superposition and parallelism.

3.5. *Towards a quantum decision tree*

Decision trees are classifiers that are probably the most intuitive for humans. Depending on the answer to a question on the features, one follows a certain branch leading to the next question until the final class is found (see Figure 9). More precisely, a mathematical tree is an undirected graph in which any two nodes are connected by exactly one edge. Decision trees in particular have one starting node, the

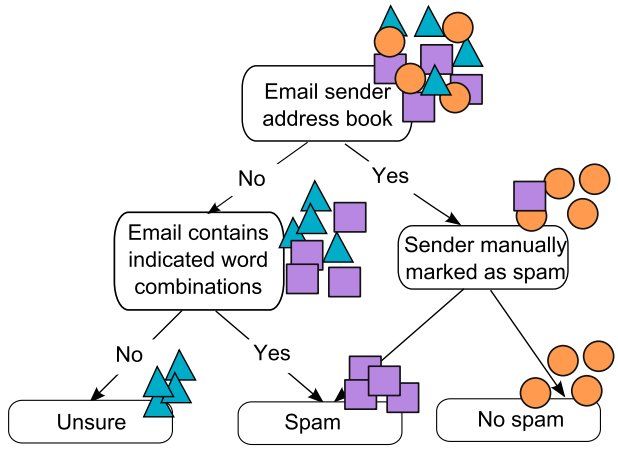

Figure 9. A simple example of a decision tree for the classification of emails. The geometric shapes symbolise feature vectors from different classes that are divided according to decision functions along the tree structure.

'root' (a node with outgoing but no incoming edges), and several end points or 'leaves' (nodes with incoming but no outgoing edges). Each node except from the leaves contains a decision function which decides which branch an input vector follows to the next layer, or in other words, which partition on a set of data is makes. The leaves then represent the final classification. As in the example in Figure 9, this procedure could be used to classify an email as 'spam', 'no spam' or 'unsure'.

Decision trees, as all classifiers in machine learning, are constructed using a training data-set of feature vectors. The art of decision tree design lies in the selection of the decision function in each node. The most popular method is to find the function that splits the given data-set into the 'most organised' sub-data-sets, and this can be measured in terms of Shannon's entropy [22]. Assume the decision function of a node splits a set of P feature vectors $\{\vec{v}^p\}$, $p = 1, \ldots, N$ into M subsets each containing $\{N_1, \ldots, N_M\}$ vectors, respectively, (and $\sum_{i=1}^{M} N_i = N$). Without further information, we calculate the probability of any vector $\vec{v}^p$ to be attributed to subset i, $i \in \{1, \ldots, M\}$ (in other words to proceed to the ith node of the next layer) as $\rho^i = \frac{N_i}{N}$, and the entropy caused by the decision function or partition is consequently $S = -\sum_{i=1}^{M} \rho^i \log(\rho^i)$. For example, in a binary tree where all nodes have two outgoing edges, the best partition would split the original set into two subsets of the same size. Obviously, this is only possible if one of the features allows for such a split. Depending on the application, an optimal decision tree would be small in the number of nodes, branches and/or levels.

Lu and Brainstein [59] propose a quantum version of the decision tree. Their classifying process follows the classical algorithm with the only difference that we use quantum feature states $|v\rangle^p = \left|v_1^p, \ldots, v_n^p\right\rangle$ encoding n features into the states of a quantum system. At each node of the tree,

the set of training quantum states is divided into subsets by a measurement (or as the authors call it, estimating attribute v_i, $i = 1, \ldots, n$). Lu and Brainstein do not give a clear account of how the division of the set at each node takes place and remain enigmatic in this essential part of the classifying algorithm. They contribute the interesting idea of using the von Neumann entropy to design the graph partition. Although the first step has been made, the potential of a quantum decision tree is still to be established.

3.6. *Quantum state classification with Bayesian methods*

Stochastic methods such as Bayesian decision theory play an important role in the discipline of machine learning. It can also be used for pattern classification. The idea is to analyse existing information (represented by the above training data set $\mathcal{T}$) in order to calculate the probability that a new input is of a certain class. An illustrative example is the risk class evaluation of a new customer to a bank. This is nothing else than a conditional probability and can be calculated using the famous Bayes formula

$$p(c|\vec{x}) = \frac{p(c)p(\vec{x}|c)}{p(\vec{x})}.$$

Here, $p(c)$, $p(\vec{x})$ are the probabilities of data being in class c and of getting input $\vec{x}$, respectively, while $p(c|\vec{x})$ is the conditional probability of assigning c upon getting $\vec{x}$ and $p(\vec{x}|c)$ is the class likelihood of getting $\vec{x}$ if we look in class c. Obviously, we assign the class with the highest conditional probability (or 'Bayes classifier') $p(c_l|\vec{x})$ to an input [22]. Values of interest, such as risk functions, can be calculated accordingly. Bayesian theory is an interesting candidate for the translation into quantum physics, since both approaches are probabilistic.

Opposed to above efforts to improve machine learning algorithms through quantum computing, Bayesian methods can be used for an important task in quantum information called quantum state classification. This problem stems from quantum information theory itself, and the goal is to use machine learning based on Bayesian theory in order to discriminate between two quantum states produced by an unknown or partly unknown source. This is again a classification problem, since we have to learn the discrimination function between two classes c_1, c_2 from examples. The two (unknown) quantum states are represented by density matrices ρ, σ. The basic idea is to use a positive operator-valued measurement (POVM) with binary outcome corresponding to the two classes as a Bayesian classifier, in other words, to learn (or calculate) the measurement on our quantum states that is able to discriminate them [60]. For this process, we have a training set consisting of examples of the two states and their respective classification, $\mathcal{T} = \{(\rho, c_1), (\sigma, c_2), (\rho, c_1), ...\}$ and the experimenter is allowed to perform any operation on the training set. Guţă and Kotłowski [60] find an optimal qubit classification strategy, while Sasaki and Carlini [61] are concerned with the related template matching problem[8] by solving an optimisation problem for the measurement operator. Sentis et al. [17] give a variation in which the training data can be stored as classical information. The proposals are so far of theoretical nature and await experimental verification of the usefulness of this scheme.

3.7. *Hidden quantum Markov models*

In the last couple of years, hidden Markov models were another important method of machine learning that has been investigated from the perspective of quantum information [18,62]. Hidden Markov models are Markov processes for which the states of the system are only accessible through observations (see Figure 10, for a very readable introduction see [63]). In a (first-order discrete and static) Markov model, a system has a countable set of states $\mathcal{S} = \{s_m\}_{m=1,\ldots,M}$ and the transition between these states are governed by a stochastic process in such a way that given a set of transition probabilities $\{a_{ml}\}_{m,l=1,\ldots,M}$, the system's state at time $t+1$ only depends on the previous state at time t. In a hidden model, the state of the system is only accessible through observations at time t $\{o_t\}$ that can take one of a set of symbols, and an observation again has a certain probability to be invoked by a specific state. Hidden Markov models are, thus, doubly embedded stochastic processes. To use a common application for pattern recognition as an example [29], consider a recorded speech. The speech is a realisation of a Markov process, a so-called Markov chain of successive words. The recording is the observation, and we shall for now imagine a way to translate the signal into discrete symbols. A Markov model is defined by the transition probabilities between words in a certain language, and the model can be learned from examples of speeches. A hidden Markov model also includes the conditional probabilities that given a certain signal observation, a certain word has been said. Goals of such models are to find the sequence of words that is the most likely for a recording, to predict the next word or, if only given the recording, to infer the optimal hidden Markov model that would encode it. Hidden Markov models play an important role in many other applications such as DNA analysis and online handwriting recognition [29].

Monras et al. [62] first introduced a hidden quantum Markov model in 2010. In contrast to a previous paper [64] in which the observations are represented by quantum basis states and the observation process is given by a von Neumann or projective measurement of an evolving quantum system, the authors consider the much more general formalism of open quantum systems (for an introduction to open quantum systems, see [65]). The state of a system is given by a density matrix ρ and transitions between states are governed by completely positive trace-nonincreasing superoperators $\mathcal{A}^i$ acting on these

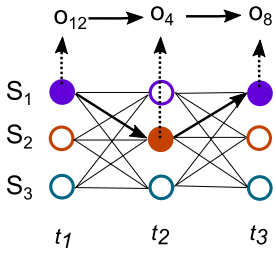

Figure 10. A hidden Markov model is a stochastic process of state transitions. In this sketch, the three states s_1, s_2, s_3 are connected with lines symbolising transition probabilities. A deterministic realisation is a sequence of states; here, the transition $s_1 \to s_2 \to s_1$ gives rise to observations $o_{12} \to o_4 \to o_8$. A task for hidden Markov models is to guess the most likely state sequence given an observation sequence.

matrices. These operations can always be represented by a set of Kraus operators [65] $\{\mathcal{K}_1^i, \ldots, \mathcal{K}_q^i\}$ fulfilling the probability conservation condition $\sum_q \mathcal{K}_q^{i\dagger}\mathcal{K}_q^i \leq 1$,

$$\rho' = \mathcal{A}^i\rho = \sum_k \mathcal{K}_k^i \rho \mathcal{K}_k^{i\dagger}.$$

The probability of obtaining state $\rho_s = P(\rho_s)^{-1}\mathcal{A}^s\rho$ is given by $P(\rho_s) = \mathrm{tr}[\mathcal{A}^s\rho]$ [62].

The advantage of hidden quantum Markov models is that they contain classical hidden Markov models and are, therefore, a generalisation offering richer dynamics than the original process [62]. In future, there might also be the possibility of 'calculating' the outcomes of classical models via quantum simulation. That would be especially interesting if the quantum setting could learn models from given examples, a problem which is nontrivial [63]. Clark et al. [18] add the notion that hidden quantum Markov models can be implemented using open quantum systems with instantaneous feedback, in which information obtained from the environment is used to influence the system. However, a rigorous treatment of this idea is still outstanding, and the power of hidden quantum Markov models to solve the problems for which classical models where developed is yet to be shown.

An interesting sibling of hidden quantum Markov models are quantum observable Markov decision processes [66] which use a very similar idea. Classical observable Markov decision processes can be understood as hidden Markov models in which before each step, an agent takes a decision for a certain action, leading to the next state of the system. The state of the system is again only accessible through observations that deliver probabilistic information. The goal is to find a strategy (defining what action to take upon what observation) that maximises the rewards given by a reward function. This is a problem of reinforcement learning by intelligent agents which is not the focus of this contribution. However, we also find the striking analogy to Kraus operations on open quantum systems representing the actions that manipulate the density matrix or stochastic description of the system.

4. Conclusion

This introduction into quantum machine learning gave an overview of existing ideas and approaches to quantum machine learning. Our focus was thereby on supervised and unsupervised methods for pattern classification and clustering tasks, and it is, therefore, by no means a complete review. In summary, there are two main approaches to quantum machine learning. Many authors try to find quantum algorithms that can take the place of classical machine learning algorithms to solve a problem, and show how an improvement in terms of complexity can be gained. This is dominantly true for nearest neighbour, kernel and clustering methods in which expensive distance calculations are sped up by quantum computation. Another approach is to use the probabilistic description of quantum theory in order to describe stochastic processes. In the case of hidden quantum Markov models, this served to generalise the model, while Bayesian theory was also used for genuinely quantum information tasks like quantum state discrimination. A great deal of contributions is still in a phase of exploring possibilities to combine formalisms from quantum theory and methods of machine learning, as seen in the area of quantum neural networks and quantum decision trees.

As previously remarked, a quantum theory of learning is yet outstanding. Although working on quantum machine learning algorithms, only very few contributions actually answer the question of how the strength and defining feature of machine learning, the *learning* process, can actually be simulated in quantum systems. Especially learning methods of parameter optimisation have not yet been accessed from a quantum perspective. Different approaches to quantum computing can be investigated for this purpose. In quantum computing based on unitary quantum gates, the challenge would be to parameterise and gradually adapt the unitary transformations that define the algorithm. Several ideas in that direction have been investigated already [36, 67,68], and important tools could be quantum feedback control [69] or quantum Hamiltonian learning [70]. As mentioned before, adiabatic quantum computing might lend itself to learning as an optimisation problem [15]. In summary, even though there is still a lot of work to do, quantum machine learning remains a very promising emerging field of research with many potential applications and a great theoretical variety.

Acknowledgements

This work was supported by the South African Research Chair Initiative of the Department of Science and Technology and the National Research Foundation.

Notes

1. See https://www.princeton.edu/achaney/tmve/wiki100k/docs/PageRank.html [Last accessed 24/6/2014]
2. It is interesting to note that although quoted in numerous introductions to machine learning, the original reference to the machine learning pioneer's most famous statement is very difficult to find. Authors either refer to other secondary publications, or falsely cite Samuel's seminal paper from 1959 [21].
3. The complexity of a problem tells us by what factor the computational resources needed to solve a gfproblem grow if we increase the input to the problem (e.g. the digits of a number) by one.
4. A feature vector has entries that refer to information on a specific case, in other words a datapoint.
5. The Hamming distance between two binary strings is the number of flips needed to turn one into the other [37].
6. The initial state can be constructed by using a Quantum Random Access Memory oracle described in [45], accessing a superposition of memory states in $O(\log(nM))$.
7. In the same paper, Rebentrost et al. [12] also present another quantum support vector machine that uses the reformulation of the optimisation as a least-squares problem, which appears to be a system of linear equations. Following [46], this can be solved by a quantum matrix inversion algorithm, which under some conditions (depending on the matrix and the output information required) can be more efficient than classical methods. The classification is then proposed to be done through a swap test.
8. Template matching is the task to assign the most similar training vector of a training set to an input vector.

Notes on contributors

Maria Schuld is a PhD candidate of the Quantum Research Group. In her previous MSc at the Technical University of Berlin as well as in her current doctoral research, she works on the question of how to merge methods of machine learning, specifically artificial neural networks, with quantum information theory. She won the study award of the Berlin Physical Society as well as the Technical University of Berlin for her Master's research.

Ilya Sinayskiy is a researcher of the South African National Institute for Theoretical Physics, and has a strong background in Quantum Information Processing, Open Quantum Systems and Quantum Biology. He received his PhD from Samara State University in Russia in 2007. Ilya is a NRF Y2-rated researcher and joined the Quantum Research Group in 2008.

Francesco Petruccione holds the South African Research Chair for Quantum Information Processing and Communication at the University of KwaZulu-Natal and head of the Quantum Research Group in Durban. He studied Physics at the University of Freiburg (Germany) and received his PhD and 'Habilitation' in 1988 and 1994 respectively. Francesco is the co-author of a monograph on 'The Theory of Open Quantum Systems' that was published in 2002.

References

[1] M. Hilbert and P. López, *The world's technological capacity to store, communicate, and compute information*, Science 332 (2011), pp. 60–65.

[2] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2010.

[3] I.M. Georgescu, S. Ashhab, and F. Nori, *Quantum simulation*, Rev. Modern Phys. 86 (2014), pp. 153–185.

[4] G.G. Rigatos and S.G. Tzafestas, *Neurodynamics and attractors in quantum associative memories*, Integr. Comput-Aided Eng. 14 (2007), pp. 225–242.

[5] E.C. Behrman and J.E. Steck, *A quantum neural network computes its own relative phase*, IEEE Symposium Series on Computational Intelligence 2013, Singapore, April 15–19, (2013).

[6] S. Gupta and R. Zia, *Quantum neural networks*, J. Comput. Syst. Sci. 63 (2001), pp. 355–383.

[7] M. Schuld, I. Sinayskiy, and F. Petruccione, *The quest for a quantum neural network*, Quant. Inform. Proc. (2014). doi: 10.1007/s11128-014-0809-8.

[8] D. Ventura and T. Martinez, *Quantum associative memory*, Inform. Sci. 124 (2000), pp. 273–296.

[9] C.A. Trugenberger, *Quantum pattern recognition*, Quant. Inform. Proc. 1 (2002), pp. 471–493.

[10] R. Schützhold, *Pattern recognition on a quantum computer*, Phys. Rev. A 67 (2003), pp. 062311.

[11] S. Lloyd, M. Mohseni, and P. Rebentrost, *Quantum algorithms for supervised and unsupervised machine learning*, preprint arXiv:1307.0411 (2013).

[12] P. Rebentrost, M. Mohseni, and S. Lloyd, *Quantum support vector machine for big data classification*, Phys. Rev. Let. 113 (2014), pp. 130503-1–130503-5.

[13] N. Wiebe, A. Kapoor, and K. Svore, *Quantum nearest-neighbor algorithms for machine learning*, preprint arXiv:1401.2142 (2014).

[14] H. Neven, V.S. Denchev, G. Rose, and W.G. Macready, *Training a large scale classifier with the quantum adiabatic algorithm*, preprint arXiv:0912.0779 (2009).

[15] K.L. Pudenz and D.A. Lidar, *Quantum adiabatic machine learning*, Quant. Inform. Proc. 12 (2013), pp. 2027–2070.

[16] R. Neigovzen, J.L. Neves, R. Sollacher, and S.J. Glaser, *Quantum pattern recognition with liquid-state nuclear magnetic resonance*, Phys. Rev. A 79 (2009), pp. 042321-1–042321-7.

[17] G. Sentís, J. Calsamiglia, R. Muñoz-Tapia, and E. Bagan, *Quantum learning without quantum memory*, Sci. Rep. 2 (2012), pp. 1–8.

[18] L.A. Clark, W. Huang, T.M. Barlow, and A. Beige, *Hidden quantum Markov models and open quantum systems with instantaneous feedback*, in *ISCS 2014: Interdisciplinary Symposium on Complex Systems 14*, A. Sanayei, O.E. Rössler and I. Zelinka, eds., Springer, 2015, pp. 143–151. Available at http://link.springer.com/book/10.1007%2F978-3-319-10759-2.

[19] S.J. Russell, P. Norvig, J.F. Canny, J.M. Malik, and D.D. Edwards, *Artificial Intelligence: A Modern Approach*, Vol. 3, Prentice Hall, Englewood Cliffs, 2010.

[20] F. Rosenblatt, *The perceptron: A probabilistic model for information storage and organization in the brain*, Psychol. Rev. 65 (1958), pp. 386–408.

[21] A.L. Samuel, *Some studies in machine learning using the game of checkers*, IBM J. Res. Dev 44 (2000), pp. 206–226.

[22] E. Alpaydin, *Introduction to Machine Learning*, MIT Press, Cambridge, 2004.

[23] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, John Wiley & Sons, New York, 2012.
[24] S.E. Landsburg, *Quantum game theory*, in *Wiley Encyclopedia of Operations Research and Management Science*, J.J. Cochran, L.A. Cox, P. Keskinocak, J.P. Kharoufeh, and J.C. Smith, eds., John Wiley & Sons, 2011.
[25] J. Eisert, M. Wilkens, and M. Lewenstein, *Quantum games and quantum strategies*, Phys. Rev. Lett. 83 (1999), pp. 3077–3080.
[26] H.J. Briegel and G. CuevasDe las, *Projective simulation for artificial intelligence*, Sci. Rep. 2 (2012), pp. 1–16.
[27] J. Du, H. Li, X. Xu, M. Shi, J. Wu, X. Zhou, and R. Han, *Experimental realization of quantum games on a quantum computer*, Phys. Rev. Lett. 88 (2002), pp. 137902-1–137902-4.
[28] E.W. Piotrowski and J. Sładkowski, *An invitation to quantum game theory*, Int. J. Theor. Phys. 42 (2003), pp. 1089–1099.
[29] C.M. Bishop, *Pattern Recognition and Machine Learning*, Vol. 1, Springer, New York, 2006.
[30] G. Hinton, S. Osindero, and Y.W. Teh, *A fast learning algorithm for deep belief nets*, Neural Comput. 18 (2006), pp. 1527–1554.
[31] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors*, Nature 323 (1986), pp. 533–536.
[32] M.B. Plenio and V. Vitelli, *The physics of forgetting: Landauer's erasure principle and information theory*, Contemp. Phys. 42 (2001), pp. 25–60.
[33] M. Sasaki and A. Carlini, *Quantum learning and universal quantum matching machine*, Phys. Rev. A 66 (2002), pp. 022303-1–022303-10.
[34] E. Aïmeur, G. Brassard, and S. Gambs, *Quantum speed-up for unsupervised learning*, Machine Learn. 90 (2013), pp. 261–287.
[35] M. Hunziker, D.A. Meyer, J. Park, J. Pommersheim, and M. Rothstein, *The geometry of quantum learning*, Quantum Inf Process. 9 (2010), pp. 321–341.
[36] A. Bisio, G. Chiribella, G.M. DAriano, S. Facchini, and P. Perinott, *Optimal quantum learning of a unitary transformation*, Phys. Rev. A 81 (2010), pp. 032324-1–032324-6.
[37] R.W. Hamming, *Error detecting and error correcting codes*, Bell Syst. Tech. J. 29 (1950), pp. 147–160.
[38] K. Hechenbichler and K. Schliep, *Weighted k-nearest-neighbor techniques and ordinal classification*, Sonderforschungsbereich 386, Paper 399 (2004). Available at http://epub.ub.uni-muenchen.de/.
[39] E. Aïmeur, G. Brassard, and S. Gambs, *Machine learning in a quantum world Advances in Artificial Intelligence*, *Lecture Notes in Computer Science Volume 4013*, Springer, 2006, pp. 431–442.
[40] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, *Quantum fingerprinting*, Phys. Rev. Lett. 87 (2001), pp. 167902.
[41] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, *Quantum amplitude amplification and estimation*, preprint arXiv: quant-ph/0005055 (2000).
[42] C. Dürr and P. Høyer, *A quantum algorithm for finding the minimum*, arXiv preprint quant-ph/9607014 (1996).
[43] C.A. Trugenberger, *Probabilistic quantum memories*, Phys. Rev. Lett. 87 (2001), pp. 067901.
[44] B.E. Boser, I.M. Guyon, and V.N. Vapnik, *A training algorithm for optimal margin classifiers*, Proceedings of the fifth annual workshop on Computational learning theory, New York, 1992, pp. 144–152.
[45] V. Giovannetti, S. Lloyd, and L. Maccone, *Quantum random access memory*, Phys. Rev. Lett. 100 (2008), pp. 160501-1–160501-4.
[46] A.W. Harrow, A. Hassidim, and S. Lloyd, *Quantum algorithm for linear systems of equations*, Phys. Rev. Lett. 103 (2009), pp. 150502-1–150502-4.
[47] S. Rogers and M. Girolami, *A First Course in Machine Learning*, CRC Press, Boca Raton, FL, 2012.
[48] D. Horn and A. Gottlieb, *Algorithm for data clustering in pattern recognition problems based on quantum mechanics*, Phys. Rev. Lett. 88 (2002), pp. 018702-1–018702-4.
[49] E. Aïmeur, G. Brassard and S. Gambs, *Quantum clustering algorithms*, Proceedings of the 24th international conference on machine learning, New York, 2007, pp. 1–8.
[50] P. Dayan and L.F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, Vol. 31, MIT Press, Cambridge, MA, 2001.
[51] J.A. Hertz and A.S. Krogh, *Introduction to the Theory of Neural Computation*, Vol. 1, Westview Press, Redwood City, CA, 1991.
[52] W. Oliveira, A.J. Silva, T.B. Ludermir, A. Leonel, W.R. Galindo, and J.C. Pereira, *Quantum logical neural networks*, 10th Brazilian Symposium on Neural Networks. 2008. SBRN'08, 2008. pp. 147–152.
[53] A.J. Silvada, W.R. Oliveirade, and T.B. Ludermir, *Classical and superposed learning for quantum weightless neural networks*, Neurocomputing 75 (2012), pp. 52–60.
[54] M. Panella and G. Martinelli, *Neural networks with quantum architecture and quantum learning*, Int. J. Circuit Theory Appl. 39 (2011), pp. 61–77.
[55] E.C. Behrman, J.E. Steck and S.R. Skinner, *A spatial quantum neural computer*, International Joint Conference on Neural Networks, 1999. IJCNN'99, Vol. 2, IEEE, Washington, DC, 1999, 874–877.
[56] G. Tóth, C.S. Lent, P.D. Tougaw, Y. Brazhnik, W. Weng, W. Porod, R.W. Liu, and Y.F. Huang, *Quantum cellular neural networks*, Superlattice. Microst. 20 (1996), pp. 473–478.
[57] J. Faber and G.A. Giraldi, *Quantum models for artificial neural networks*, 2002. Available at http://arquivosweb.lncc.br/pdfs/QNN-Review.pdf.
[58] G. Purushothaman and N. Karayiannis, *Quantum neural networks (QNNs): Inherently fuzzy feedforward neural networks*, IEEE Trans. Neural Netw. 8 (1997), pp. 679–693.
[59] S. Lu and S.L. Braunstein, *Quantum decision tree classifier*, Quant. Inform. Proc. 13 (2014), pp. 757–770.
[60] M. Guţă and W. Kotłowski, *Quantum learning: Asymptotically optimal classification of qubit states*, New J. Phys. 12 (2010), pp. 123032.
[61] M. Sasaki, A. Carlini, and R. Jozsa, *Quantum template matching*, Phys. Rev. A 64 (2001), pp. 022317-1–022317-11.
[62] A. Monras, A. Beige, and K. Wiesner, *Hidden quantum Markov models and non-adaptive read-out of many-body states*, Appl. Math. Comput. Sci. 3 (2010), pp. 93–122.
[63] L.R. Rabbiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, Proc. IEEE 77 (1989), pp. 257–286.
[64] K. Wiesner and J.P. Crutchfield, *Computation in finitary stochastic and quantum processes*, Phys. D: Nonlinear Phenom. 237 (2008), pp. 1173–1195.
[65] H.P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems*, Oxford University Press, New York, 2002.

[66] J. Barry, D.T. Barry, and S. Aaronson, *Quantum partially observable Markov decision processes*, Phys. Rev. A. 90 (2014), pp. 032311-1–032311-11.

[67] S. Gammelmark and K. Mølmer, *Quantum learning by measurement and feedback*, New J. Phys. 11 (2009), pp. 033017-1–033017-11.

[68] S. Gammelmark and K. Mølmer, *Bayesian parameter inference from continuously monitored quantum systems*, Phys. Rev. A 87 (2013), pp. 032115-1–032115-9.

[69] A. Hentschel and B.C. Sanders, *Machine learning for precise quantum measurement*, Phys. Rev. Lett. 104 (2010), pp. 063603-1–063603-4.

[70] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, *Quantum Hamiltonian learning using imperfect quantum resources*, Phys. Rev. A 89 (2014), pp. 042314-1–042314-16.