Quantum machine learning

Ievgeniia Oshurko
Quantum Information and Computation course, ENS Lyon

November 26, 2016

1 Introduction

Machine learning techniques are applied for solving a large variety of practical problems. The tasks attacked by machine learning algorithms include classification, regression, pattern recognition, etc. Traditionally, machine learning algorithms are divided into two groups depending on the nature of training data: supervised and unsupervised. Supervised machine learning algorithms take on the input labeled samples and learn how to predict labels for new data points. Unsupervised learning algorithms like the ones for clustering, pattern recognition, on the other hand, take on the input unlabeled data and try to discover the hidden structure in the data.

Algorithms of machine learning manipulate large amounts of data represented in a form of arrays of vectors of high dimension. These algorithms become practically intractable for classical computers as the amount of data grows rapidly [4], whereas quantum computing potentially provides an exponential speed up [11]. So far, a large number of quantum machine learning algorithm has been developed, for example, the algorithm for solving systems of linear equations HHL [10] is used for generic classification problems as perceptron or linear regression training, there were developed quantum algorithms for nearest centroid, k-nearest neighbours and support vector machines classsification [2].

In this review we will focus on the supervised classification quantum algorithm of *nearest centroid*, presented in [11]. Quantum computational model helps to overcome the main bottleneck of the algorithm: calculation of the distances between the vectors in highly dimensional space. Methods of computing distances similar to the one presented in the context of nearest centroid algorithm can be used in many other practical applications, and the first experimental results on implementation of this algorithm on a small-scale quantum photonic computer can be found in [4].

2 Quantum algorithms of nearest centroid classification

2.1 Classical algorithm for nearest centroid classification

Classification task lies in the assignment of label to the data sample based on the previously known and labeled data. Machine learning algorithms for classification usually try to infer a function which can discriminate between two labels in such a way that the prediction of the label would be the most accurate.

Having an input vector $\vec{x}$ and clusters V and W given by two sets of vectors $V = \{\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_{M-1}\}$ and $W = \{\vec{w}_0, \vec{w}_1, \ldots, \vec{w}_{M-1}\}$ (let us say of the same size M) *nearest centroid classification algorithm*

finds a representative vector – centroid – for each cluster, and assigns to the vector $\vec{x}$ a label corresponding to the set whose centroid is the closest to the input vector [12]. Classically, the complexity of the algorithm is $O(poly(MN))$. The algorithm is presented in the pseudo code below:

Algorithm 1: Cluster assignment procedure

Data: $\vec{x} \in \mathbb{R}^n$, set $V = \{\vec{v}_j\}_{1 \leq j \leq M}$ and $W = \{\vec{w}_k\}_{1 \leq k \leq M}$
Result: Cluster assignment of $\vec{x}$
$d(\vec{x}, V) \leftarrow |\vec{x} - \frac{1}{M}\sum_{j=1}^{M} \vec{v}_j|$;
$d(\vec{x}, W) \leftarrow |\vec{x} - \frac{1}{M}\sum_{k=1}^{M} \vec{w}_k|$;
if $d(\vec{x}, V) < d(\vec{x}, W)$ **then**
  $\vec{x} \in V$;
else
  $\vec{x} \in W$;

2.2 Data preparation

On the input the algorith receives classical data stored in arrays of vectors. We assume that data is stored in a form of complex vectors of dimension $N = 2^n$.

$$\vec{x} = (x_1, x_2, \ldots, x_N)$$

First stage of the algorithm consists of the preparation of the quantum state corresponding to each data vector $\vec{x}$. We would like to construct the state of the form:

$$|x\rangle = \frac{\vec{x}}{|\vec{x}|}$$

For the sake of convenience we put vector $\vec{v_0} = \vec{w_0} = \vec{x}$, then data preparation stage of our quantum algorithm consists of the following steps:

1. Construct the state $|\psi_0\rangle$ of the form:

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle + \frac{1}{\sqrt{M}}\sum_{j=1}^{M}|j\rangle\Big) \tag{1}$$

   The state $|\psi_0\rangle$ can be constructed applying unitary V on the quantum state $|0^{\otimes \log_2(M+1)}\rangle$. Unitary V should be such that [12]:

$$\big|V_{j0}\big| = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } j = 0, \\ \frac{1}{\sqrt{2M}}, & \text{otherwise.} \end{cases} \tag{2}$$

   Unitary V can be efficiently constructed using Hadamard, T and CNOT gates, the proof of this fact can be found in [12]. As the result we get the following evolution:

$$|0^{\otimes \log_2(M+1)}\rangle \rightarrow |\psi_0\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle + \frac{1}{\sqrt{M}}\sum_{j=1}^{M}|j\rangle\Big) \tag{3}$$

2. On the second step of the algorithm we construct the state $|\psi\rangle$:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle |v_0\rangle + \frac{1}{\sqrt{M}} \sum_{j=1}^{M} |j\rangle |v_j\rangle \Big) \tag{4}$$

We assume that for each set of vectors $\{\vec{v}_j\}_{0\leq j\leq M}$ of dimension $N = 2^n$ belonging to the cluster V we are given an oracle O_V which is able to access the classical data and construct the corresponding quantum state as follows:

$$O_V |j\rangle |0\rangle = |j\rangle |v_j\rangle \tag{5}$$

So, now querying the oracle O_V we can costruct the required state $|\psi\rangle$:

$$O_V |\psi_0\rangle |0^{\otimes n}\rangle = |\psi\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle |v_0\rangle + \frac{1}{\sqrt{M}} \sum_{j=1}^{M} |j\rangle |v_j\rangle \Big) \tag{6}$$

Oracle O_V can be implemented using Quantum Random Access Memory [8] or the procedure described in [9] and shortly presented in appendix A.

We may perform an estimation of complexity of our algorithm by means of the number of queries to the oracle. But for the realistic analysis of the algorithm we need to consider the cost of state preparation performed by oracle as well. For it to be efficient multiple requirements on the data itself should be satisfied. The construction of the state may be performed in $O(\log_2 N)$ if the partial sums $n_k = \sum_{i=1}^{k} |v_i|$ for any vector $\vec{v}$ can be computed efficiently.

2.3 Cluster assignment

Let us consider cluster V and the vectors that belong to this cluster $\{\vec{v}_j\}_{1\leq j\leq M}$. The quantum algorithm of finding the distance between the input vector $\vec{x} = \vec{v}_0$ and the centroid of a cluster $\frac{1}{M}\sum_{i=1}^{M} \vec{v}_j$ consists of the following steps (the detailed description of the steps will be presented later):

1. Prepare the state $|\psi\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle |v_0\rangle + \frac{1}{\sqrt{M}} \sum_{j=1}^{M} |j\rangle |v_j\rangle \Big)$ from the input data as described in the previous section.
2. Construct the state $|\phi\rangle$:

$$|\phi\rangle = \frac{1}{\sqrt{Z}}\Big(|\vec{v_0}| |0\rangle - \frac{1}{\sqrt{M}} \sum_{j=1}^{M} |\vec{v}_j| |j\rangle \Big),$$

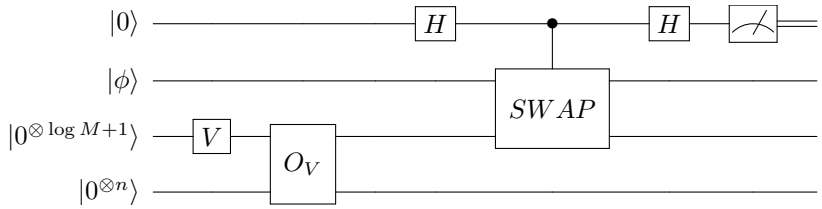
where $Z = |\vec{v_0}|^2 + \frac{1}{M}\sum_{j=1}^{M} |\vec{v}_j|^2$

3. Perform a projective measurement of ancilla variable of $|\psi\rangle$ onto the state $|\phi\rangle$ using *swap test*.
4. Repeat step 2 and 3 to determine the probability of success of swap test $P(0)$ up to an accuracy ϵ.

5. Find the distance classically from $P(0)$:

$$\left|\vec{v_0} - \frac{1}{M}\sum_{j=1}^{m}\vec{v_j}\right| = \sqrt{2Z(2P(0)-1)} \tag{7}$$

The following circuit summarises the steps of the algorithm:



Construction of the state $|\phi\rangle$. State $|\phi\rangle$ can be constructed using access to the previously defined oracle O_V and quantum simulation [7] to apply the transformation e^{iHt}, where Hamiltonian H is defined as follows:

$$H = \Big(\sum_{j=0}^{M} |\vec{v_j}|\,|j\rangle\,\langle j|\Big) \otimes \sigma_x \tag{8}$$

Here σ_x is the Pauli matrix. We put $|\phi(0)\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - \frac{1}{\sqrt{M}}\sum_{j=1}^{M}|j\rangle\big)\otimes|0\rangle$ and the measurement of ancilla of the evolution $e^{iHt}|\phi(0)\rangle$ results into the desired state $|\phi\rangle$. More details the efficient preparation of this state can be found in [11].

Swap test on the states $|\psi\rangle$ and $|\phi\rangle$ [3]. Swap test consists of the application of *controlled-SWAP* (cSWAP, also knowns as Fredkin gate) with the controlled qubit being in the state $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$ to the two states $|\psi\rangle$ and $|\phi\rangle$.

$$CSWAP\,|+\rangle\,|\psi\rangle\,|\phi\rangle = \frac{|0\rangle\,|\psi\rangle\,|\phi\rangle + |1\rangle\,|\phi\rangle\,|\psi\rangle}{\sqrt{2}} \tag{9}$$

Afterwards, the Haramard gate is applied to our control qubit alone, which results in the following evolution:

$$CSWAP\,|+\rangle\,|\psi\rangle\,|\phi\rangle \to \frac{|0\rangle\Big(|\psi\rangle\,|\phi\rangle + |\phi\rangle\,|\psi\rangle\Big) + |1\rangle\Big(|\psi\rangle\,|\phi\rangle - |\phi\rangle\,|\psi\rangle\Big)}{2} \tag{10}$$

After the measurement of the first register we get 0 with probability $P(0) = \frac{1+|\langle\psi|\phi\rangle|^2}{2}$. As we can see, if $|\psi\rangle = |\phi\rangle$ the probability to measure 0 equals to 1, on the other hand if they are orthogonal, this probability equals to $\frac{1}{2}$. The probability of passing the test (measuring 0), thus, depends on the overlap between $|\psi\rangle$ and $|\phi\rangle$, and this overlap gives a good estimate of how close two states are [6].

Analysis of success probability of swap test. So the probability of success of swap test is $P(0) = \frac{1+|\langle\psi|\phi\rangle|^2}{2}$. We want to see that from this probability deterministically we can find the desired distance. Remeber that we perform swap test only with ancilla of $|\psi\rangle$, so we have:

$$
\begin{aligned}
|\langle\phi|\psi\rangle|^2 &= \frac{1}{2Z}\Big|\Big(|\vec{v_0}|\langle 0| - \frac{1}{\sqrt{M}}\sum_{j=1}^{M}|\vec{v_j}|\langle j|\Big)\Big(|0\rangle|v_0\rangle + \frac{1}{\sqrt{M}}\sum_{j=1}^{M}|j\rangle|v_j\rangle\Big)\Big|^2 \\
&= \frac{1}{2Z}\Big||\vec{v_0}|\langle 0|0\rangle|v_0\rangle + \frac{1}{\sqrt{M}}\sum_{j=1}^{M}|\vec{v_0}|\langle 0|j\rangle|v_j\rangle \\
&- \frac{1}{\sqrt{M}}\sum_{j=1}^{M}|\vec{v_j}|\langle j|0\rangle|v_0\rangle - \frac{1}{M}\sum_{j=1}^{M}|\vec{v_j}|\langle j|j\rangle|v_j\rangle\Big|^2 \\
&= \frac{1}{2Z}\Big(\Big||\vec{v_0}||v_0\rangle - \frac{1}{M}\sum_{j=1}^{M}|\vec{v_j}||v_j\rangle\Big|^2\Big) \\
&= \frac{1}{2Z}\Big(|\vec{v_0}|^2 + \frac{1}{M^2}|\sum_{j=1}^{M}\vec{v_j}|^2 - 2\frac{1}{M}\sum_{j=1}^{M}\vec{v_0}\cdot\vec{v_j}\Big) \\
&= \frac{1}{2Z}\Big|\vec{v_0} - \frac{1}{M}\sum_{j=1}^{M}\vec{v_j}\Big|^2
\end{aligned}
$$

As we can see from the caculations above desired distance is $\sqrt{2Z(2P(0)-1)}$. The success probability $P(0)$ can be determined up to accuracy ϵ repeating the steps 2 and 3 of the algorithm for $O(P(0)(1-P(0))/\epsilon^2)$ times [11]. Note that this number is independent of the dimensionality of vector as well as number of vectors in each cluster.

The main idea of the algorithm is to adjoint an ancilla to the states created from the input vector $\vec{v_0}$ and the cluster centroid $\frac{1}{M}\sum_{j=1}^{M}\vec{v_j}$ creating an entangled state. The bigger difference between them, the more entangled the resulting state is, and therefore we can use this entanglement to estimate the distance between the vectors [4]. From the resulting distance $D = \sqrt{2Z(2P(0)-1)}$ we can see that if vector $\vec{v_0}$ is identical to the centroid of the cluster V the state $|\psi\rangle$ is orthogonal to $|\phi\rangle$ and the success probability of projective measurement is $P(0) = 1/2$, therefore the distance $D = 0$.

2.4 Time complexity

Previously presented procedure for the preparation of an entangled state $|\psi\rangle$ from the input data takes time $O(\log_2 N)$ as long as:

1. the sub-norms of the vectors can be calculated efficiently if we use the state preparation procedure described in appendix A;

2. these sub-norms are given in quantum RAM.

We assume that one of this conditions is the case for our input.

For estimation of the probability of success of the projective measurement described above, the required number of iterations of the algorithm steps does not depend neither on N nor on M. It

depends on the desired error ϵ solely ($O(\epsilon^{-2}P(0)(1-P(0))$ iterations are required to estimate the probability of success up to an error ϵ, and we know that $D \propto \sqrt{P(0)}$).

Therefore, to find the distance between the centroid of a cluster and the input data sample $\vec{x}$ up to an error ϵ a quantum computer takes time $O(\epsilon^{-1}\log_2(MN))$. In addition, the fact that only $O(\log_2(MN))$ calls to the quantum database are required to perform an appropriate cluster assignment enhances the security of the database [11].

3 Discussion and Conclusions

Presented approach to finding distances between vectors can be used as a subroutine of various machine learning algorithms like k-nearest neighbour classification, support vector machines or k-means clustering algorithm. Quantum adiabatic algorithm which uses this procedure for k-mean clustering can be found in [11]. Together with HHL algorithm mentioned before, it provides a powerful quantum toolkit that can be used in various machine learning tasks, and that can potentially provide an exponential speed up over classical tools.

The algorithm presented in this review can be enhanced in various ways that would make it more suitable for the real-world applications. In [12] distances from the vector $\vec{x}$ and cluster centroids are normalized by the width of the cluster, which helps us in case if one cluster is significantly wider than another. They also implement nearest centroid classification in general case of $C \geq 2$ clusters applying the Durr-Hoyer algorithm [5] for finding the cluster whose centroid is the closest to the input sample.

There are issues to be considered when talking about the exponential speed up for both presented procedure and HHL algorithm. As was pointed out in [1], this speed up relies on the assumption that the quantum states corresponding to the input vectors can be prepared in $O(\log_2(N))$, where N is the dimensionality of the vectors. The preparation of these states can be achieved using quantum RAM, which would store classical values in the memory and allow to access them in quantum superposition. Another alternative to the usage of quantum RAM is a state preparation procedure described in the appendix A, and in this case we require that the partial sums $n_k = \sum_{i=1}^{k} |x_i|$ can be estimated efficiently. In addition to these requirements, the amplitudes of the quantum state should be relatively uniform, otherwise the preparation procedure takes time $O(poly(N))$ and the exponential speed up disappears. In [1] Scott Aaronson mentions that if the amplitudes are uniform then there is a classical random sampling algorithm which calculates the inner products between vectors in $O(\epsilon^{-2}\log(MN))$, which shows that our quantum algorithm achieves only quadratic speed up over this algorithm.

Nevertheless, apart from the speed up potentially offered by quantum machine learning algorithms, we can get new insights on the learning capacities of quantum systems for the applications to artificial intelligence. Quantum computing may help us not only with costly subroutines in classical algorithms (like the one presented here), but it also can reveal a new type of 'purely quantum' learning.

References

[1] Scott Aaronson. "Read the fine print". In: *Nature Physics* 11.4 (2015), pp. 291–293.

[2] Jeremy Adcock et al. "Advances in quantum machine learning". In: *arXiv preprint arXiv:1512.02900* (2015).

[3] Harry Buhrman et al. "Quantum fingerprinting". In: *Physical Review Letters* 87.16 (2001), p. 167902.

[4] X-D Cai et al. "Entanglement-based machine learning on a quantum computer". In: *Physical review letters* 114.11 (2015), p. 110504.

[5] Christoph Durr and Peter Hoyer. "A quantum algorithm for finding the minimum". In: *arXiv preprint quant-ph/9607014* (1996).

[6] Juan Carlos Garcia-Escartin and Pedro Chamorro-Posada. "swap test and Hong-Ou-Mandel effect are equivalent". In: *Physical Review A* 87.5 (2013), p. 052330.

[7] IM Georgescu, S Ashhab, and Franco Nori. "Quantum simulation". In: *Reviews of Modern Physics* 86.1 (2014), p. 153.

[8] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. "Quantum random access memory". In: *Physical review letters* 100.16 (2008), p. 160501.

[9] Lov Grover and Terry Rudolph. "Creating superpositions that correspond to efficiently integrable probability distributions". In: *arXiv preprint quant-ph:0208112* (2002).

[10] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations". In: *Physical review letters* 103.15 (2009), p. 150502.

[11] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum algorithms for supervised and unsupervised machine learning". In: *arXiv preprint arXiv:1307.0411* (2013).

[12] Nathan Wiebe, Ashish Kapoor, and Krysta Svore. "Quantum nearest-neighbor algorithms for machine learning". In: *arXiv preprint arXiv:1401.2142* (2014).

A Creating superpositions from probability distributions

In this appendix we present the method given in [9] which can be used for the preparation of the quantum states from the input vectors for our nearest centroid classification algorithm.

Our objective is, having the probability distribution $\{p_i\}_{1\leq i\leq n}$, to efficiently construct the corresponding quantum superposition:

$$|\psi(\{p_i\})\rangle = \sum_i \sqrt{p_i}\,|i\rangle \tag{11}$$

Assume we want to create the superposition of 2^n states. This procedure is iterative, and on each iteration we create the superposition of 2^{m+1} states from the superposition of 2^m (for $1 \leq m \leq n-1$) states until we reach required number of qubits. Assume we have the following state:

$$|\psi_m\rangle = \sum_{i=0}^{2^m-1} \sqrt{p_i^{(m)}}\,|i\rangle \tag{12}$$

We want to add one qubit to the state and achieve the following transformation:

$$\sqrt{p_i^{(m)}}\,|i\rangle \to \sqrt{\alpha_i}\,|i\rangle\,|0\rangle + \sqrt{\beta_i}\,|i\rangle\,|1\rangle\,, \tag{13}$$

where α_i (β_i) would be the probability of x to lie in the left (right) half of the region i. To show it is possible to perform such a transformation the function $f(i)$ is defined as follows:

$$f(i) = \frac{\sum_{j=x_L^i}^{(x_R^i-x_L^i)/2} p_j}{\sum_{j=x_L^i}^{x_R^i} p_j}, \tag{14}$$

here x_L^i and x_R^i are respectively a left boundary and a right boundary. Value of $f(i)$ represents the probability of some j to lie in the left half of the region i. Therefore, if there is an efficient classical algorithm to caluculate the sub-norms $\sum_{i=a}^{b} p_i$, then we can efficiently create a quantum state on $m+1$ qubits. First, we apply the following evolution:

$$\sqrt{p_i^{(m)}}\,|i\rangle\,|0\ldots0\rangle \to \sqrt{p_i^{(m)}}\,|i\rangle\,|\theta\rangle\,, \tag{15}$$

where $\theta = \arccos\sqrt{f(i)}$. Second, we add new qubit and perform its controlled rotation as follows:

$$\sqrt{p_i^{(m)}}\,|i\rangle\,|\theta\rangle\,|0\rangle \to \sqrt{p_i^{(m)}}\,|i\rangle\,|\theta\rangle\,(\cos\theta\,|0\rangle + \sin\theta\,|1\rangle) \tag{16}$$

We simply bring the register containing θ to its initial state $|0\ldots0\rangle$, and we receive the state of the exactly desired form.

As it was mentioned before, given vector $\vec{x} = (x_1, x_2, \ldots, x_N)$ (assume $N = 2^n$ for the sake of convenience) we would like to construct the following state:

$$|x\rangle = \frac{\vec{x}}{|\vec{x}|} \tag{17}$$

Therefore, to prepare such a quantum state efficiently by the procedure described above we need to be able to calculate sub-norms $n_k = \sum_{i=1}^{k} |x_i|$ efficiently.