Chapter 1

Quantum Nearest Neighbour Algorithm

*The first part of Section 1.1 (introducing the quantum nearest neighbour algorithm in basis encoding) has been published in Ref [**?**]: Schuld, Sinayskiy, Petruccione (2014) Quantum Computing for Pattern Classification, Lecture Notes in Computer Science Vol 8862, Springer, pp. 208-220.]. I was responsible for the design, analysis and write-up of the content of the publication.*

The k-nearest neighbour algorithm has been briefly introduced in Chapter **??** as one of the most simple but yet successful classifiers. It can be formulated as the following rule (illustrated in Figure **??**):

Predict the class for a new input that most of its k closest training vectors are assigned to.

Some versions of the model weigh the neighbours by their distance to the new input, such that closer neighbours have more influence on the prediction than those further away. With this adaptation one can in theory look at $k = M$ which takes into account the entire dataset. The weighing rule then defines how fast the 'influence' decreases with distance and can be understood as a kernel $\kappa(|\tilde{\mathbf{x}} - \mathbf{x}^m|)$. An illustration is presented in Figure 1.1.

To find a quantum algorithm for k-nearest neighbour and its variations has been done before. LITERATURE.

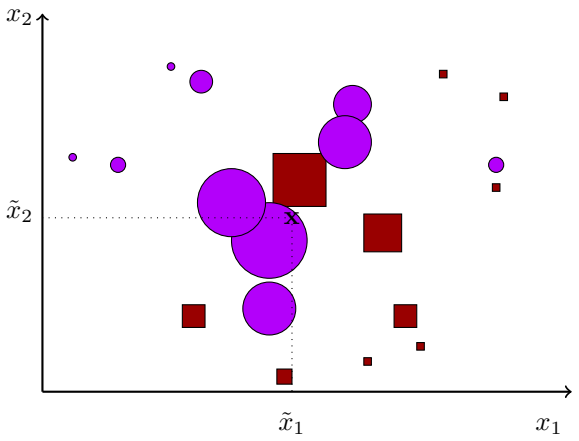

Figure 1.1: Illustration of all-nearest neighbour where the neighbours are weighted by the Euclidean distance to the new input. The symbols show the 2-dimensional inputs that have each a class attribute 'circle' or 'rectangle'. The new input is located at $\tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x_2})$, and as in the k-nearest neighbour illustration it will be classified as a circle.

1.1 Quantum nearest-neigbour algorithm

The idea of the quantum algorithm is to create a superposition in which each basis state corresponds to a training point, and to write the weighing factor (or the value of the kernel function) into the corresponding amplitude,

$$\sum_{m=1}^{M} \sqrt{\kappa(|\tilde{\mathbf{x}} - \mathbf{x}^m|)} |x^m, y^m\rangle. \tag{1.1}$$

For this a simple routine developed by Trugenberger [**?**] will be applied, which leads to a cosine kernel function (restricted to the interval $[0, \pi]$).

In the basis encoding version of the quantum machine learning algorithm, the data is represented by the basis states $|x^m, y^m\rangle$ (for details, see Section **??**). The way this works is that each data point (x^m, y^m) is translated to a bit sequence by representing each real number by τ bits and concatenating them to a $(N+1)\tau$-bit sequence $(x_1^m ... x_{N\tau}^m; y_1^m ... y_\tau^m)$. In the quantum state, only basis states of the form $|x_1^m ... x_{N\tau}^m; y_1^m ... y_\tau^m\rangle$ have a nonzero amplitude.

Doing a measurement on the last qubits of the superposition encoding the class label reveals the combined weight of each class. The measurement outcome will be 0 with a probability

$$\sum_{m|y^m=-1} \kappa(|\tilde{\mathbf{x}} - \mathbf{x}^m|),$$

and 1 with probability

$$\sum_{m|y^m=1} \kappa(|\tilde{\mathbf{x}} - \mathbf{x}^m|).$$

Alternatively, one can turn the algorithm into a k-nearest-neighbour version by measuring the entire data register and repeating the entire procedure a number of times. Training inputs that are closer will have a higher probability to be the outcome, and the state of their class qubit is recorded. The final prediction is the class of which more class qubits were measured. This is in a sense a probabilistic version of k-nearest neighbour. CALCULATE THE STATISTICS, WHAT IS k, A MEAN?.

Of course, in both versions the measurement destroys the state and for each classification it has to be re-prepared, which requires to store and access the entire dataset as in the classical method. The runtime of the routine depends on the resources needed for state preparation, and this is what has to be compared with the fastest classical methods which are known to take time in ????.[1]

Representing the dataset in basis encoding is rather expensive in terms of spatial resources, especially when dealing with continuous features: The number of qubits needed if every real entry of the input vector is encoded in τ qubits is of the order $N\tau$. For example, if the features are encoded as binary fractions, a precision of τ would imply an error of less than $\frac{1}{2^\tau}$. To have an error of less than 0.01, we need at least $\tau = 7$ qubits per feature. Considering the technological challenges to scale quantum computing to a large number of qubits, this is quickly prohibitive for large input spaces. It is therefore interesting to develop a similar routine based on amplitude encoding which will be considered further below.

1.1.1 Quantum nearest neighbour routine in basis encoding

I will first discuss the quantum nearest neighbour algorithm with the training set given in basis encoding. For simplicity, I assume in the following that the features as well as the class are binary, and choose the distance measure to be the Hamming distance. The m'th basis state in the superposition then becomes $|x_1^m, ..., x_N^m; y^m\rangle$. The algorithm can easily be adapted to the case of continuous numbers and multilabel classification, with the only difference being the routine to extract the distance measure.

A state preparation procedure (such as presented in Section **??**) is used to encode the training set into a uniform superposition $\sum_{m=1}^{M} |x_1^m, ..., x_N^m; y^m\rangle$ joined with a register encoding the new input as

[1] For kNN The fastest sorting algorithms are in $\mathcal{O}(\mathcal{M} \log \mathcal{M})$.

well as an ancilla in superposition:

$$\sum_{m=1}^{M} |x_1, ..., x_N \; ; \; y^m\rangle |\tilde{x}_1, ..., \tilde{x}_N\rangle (|0\rangle + |1\rangle).$$

Here and unless stated different also in the following, global normalisation constants will be ignored since the desired result is a relative value. An XOR gate applied to every qubit x_j and $\tilde{x}_j$ for $j = 1...N$ compares the two vectors and writes the result

$$d_j^m = \begin{cases} 1, & \text{if } x_j^m = \tilde{x}_j, \\ 0, & \text{else,} \end{cases}$$

into the register previously containing the new input,

$$\sum_{m=1}^{M} |x_1, ..., x_N \; ; \; y^m\rangle |d_1^m, ..., d_N^m\rangle (|0\rangle + |1\rangle).$$

A unitary $e^{-i\frac{\pi}{2N}H}$ with the Hamiltonian

$$H = 1 \otimes 1 \otimes \sum_j \left(\frac{(\sigma_z)_j + 1}{2} \right) \otimes \sigma_z$$

which $(\sigma_z)_j$ acting on d_j, has the effect of writing the sum of the d_j, in other words the Hamming distance, into the phase with a sign conditioned on the last ancilla qubit.

$$\sum_{m=1}^{M} e^{i\frac{\pi}{2N}\sum_j d_j^m} |x_1, ..., x_N \; ; \; y^m\rangle |d_1^m, ..., d_N^m\rangle |0\rangle + \sum_{m=1}^{M} e^{-i\frac{\pi}{2N}\sum_j d_j^m} |x_1, ..., x_N \; ; \; y^m\rangle |d_1^m, ..., d_N^m\rangle |1\rangle$$

A Hadamard gate on the last ancilla qubit then interferes the two terms and results in

$$\sum_{m=1}^{M} \cos(\frac{\pi}{2N} \sum_j d_j^m) |x_1, ..., x_N \; ; \; y^m\rangle |d_1^m, ..., d_N^m\rangle |0\rangle + \sum_{m=1}^{M} \sin(\frac{\pi}{2N} \sum_j d_j^m) |x_1, ..., x_N \; ; \; y^m\rangle |d_1^m, ..., d_N^m\rangle |1\rangle.$$

A conditional measurement on the ancilla selects the cosine term. The probability of acceptance is given by

$$p(0) = \frac{1}{M} \sum_{m=1}^{M} \cos^2 \left[\frac{\pi}{2N} \sum_j d_j^m \right],$$

where the previously ommitted normalisation constant of M^{-1} was included. This probability is also a measure of how close the data is to the new input: If the collective Hamming distance is large,

the sine branch of the superposition will have a larger probability to be measured. In the worst case scenario, all training vectors have a Hamming distance close to N and the probability of the conditional measurement to succeed will be close to zero. However, in this case the data might not reveal a lot of information for the classification of the new input anyways, and the probability can therefore be seen as a measure of how well-posed the classification problem is in the first place.

After a successful conditional measurement, the state becomes proportional to

$$\sum_{m=1}^{M} \cos(\frac{\pi}{2N} \sum_j d_j^m) |x_1^m, ..., x_N^m \ ; \ y^m\rangle,$$

where the $|d_1, ..., d_N\rangle|0\rangle$ registers were ommitted. This is precisely the form of 1.1. The kernel function used is a cosine, and the factor of $\frac{\pi}{2N}$ normalises the argument in the cosine to the interval $[0, \pi]$ where it is a monotonically decreasing function, similar to a Gaussian. As required, training vectors with a larger Hamming distance to the new input get a smaller weight than those closer to it.

The two versions of the nearest neighbour algorithm can now be implemented by choosing the measurement. A measurement on the class qubit y^m will have a probability of

$$p(\tilde{y} = -1) = p(|y^m\rangle = |0\rangle) = \sum_{m|y^m=0} \cos^2(\frac{\pi}{2N} \sum_j d_j^m)$$

to predict the class -1 and a complementary probability of

$$p(\tilde{y} = 1) = p(|y^m\rangle = |1\rangle) = \sum_{m|y^m=1} \cos^2(\frac{\pi}{2N} \sum_j d_j^m)$$

to predict class 1. Alternatively, measuring the entire basis state $|x_1^m, ..., x_N^m \ ; \ y^m\rangle$ has a probability of

$$p(\mathbf{x}^m) = \cos^2(\frac{\pi}{2N} \sum_j d_j^m)$$

to pick the mth training vector, and closer training vectors are thus preferred. Doing this repeatedly can be understood as a statistical procedure to select the closest vectors and to assign the new class via majority vote.

Routine 1.1.1: QUANTUM LINEAR REGRESSION ALGORITHM (BASIS ENCODING)

Goal: Implement the weighed nearest neighbour algorithm for binary pattern classification on a quantum computer. For simlicity assume that the features and class are of binary form.

Runtime: $\mathcal{O}(\frac{N}{p_{\text{acc}}})$ excluding state preparation [Number of features N]

Input: A quantum superposition of training vectors in basis encoding joined with a register containing the new input.

Output: Class of new output $\tilde{y}$

PROCEDURE

Given: $\sum_{m=1}^{M} |x_1^m, ..., x_N^m \,;\, y^m\rangle|\tilde{x}_1, ..., \tilde{x}_N\rangle(|0\rangle + |1\rangle)$

1: $\sum_{m=1}^{M} |x_1^m, ..., x_N^m \,;\, y^m\rangle|d_1^m, ..., d_N^m\rangle(|0\rangle + |1\rangle)$ ▷ $d_j = x_j \oplus \tilde{x}_j$

2: $\sum_{m=1}^{M} e^{i\frac{\pi}{2n}\sum_k d_k}|x_1^m, ..., x_N^m \,;\, y^m\rangle|d_1^m, ..., d_N^m\rangle|0\rangle +$
$\sum_{m=1}^{M} e^{-i\frac{\pi}{2n}\sum_k d_k}|x_1^m, ..., x_N^m \,;\, y^m\rangle|d_1^m, ..., d_N^m\rangle|1\rangle$ ▷ after unitary

3: $\sum_{m=1}^{M} \cos(\frac{\pi}{2n}\sum_k d_k)|x_1^m, ..., x_N^m \,;\, y^m\rangle|d_1^m, ..., d_N^m\rangle|0\rangle +$
$\sum_{m=1}^{M} \sin(\frac{\pi}{2n}\sum_k d_k)|x_1^m, ..., x_N^m \,;\, y^m\rangle|d_1^m, ..., d_N^m\rangle|1\rangle$ ▷ after Hadamard

4: $\sum_{m=1}^{M} \cos(\frac{\pi}{2N}\sum_j d_j^m)|x_1^m, ..., x_N^m \,;\, y^m\rangle|d_1^m, ..., d_N^m\rangle|0\rangle$ ▷ after conditional measurement

5: **return** $\begin{cases} -1 \text{ if } p(|y^m\rangle = |0\rangle) > p(|y^m\rangle = |1\rangle) \\ 1 \text{ else} \end{cases}$ ▷ measure the class qubit y^m

1.1.2 Quantum nearest neighbour routine in amplitude encoding

As discussed above, the linear dependency of runtime and qubits needed on the input dimension can still be prohibitive, and it is worth to look for alternatives based on amplitude encoding. For this section, I will relax the assumption of binary features and consider continuous features for a binary classification problem. For this one requires the (normalised) data set and new input to be encoded

in the amplitudes of a quantum state,

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

with

$$|\psi_{\mathbf{x}^m}\rangle=\sum_{i=1}^{N}x_i^m|i\rangle,\quad |\psi_{\tilde{\mathbf{x}}}\rangle=\sum_{i=1}^{N}\tilde{x}_i|i\rangle.$$

A Hadamard gate on the ancilla interferes the two states and results in

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

A conditional measurement to find the ancilla in $|0\rangle$ succeeds with probability $p_{\text{acc}} = \frac{1}{4}\sum_m\sum_i|\tilde{x}_i+x_i^{(m)}|^2$ which is equal to $1-\frac{1}{4}\sum_m\sum_i|\tilde{x}_i-x_i^{(m)}|^2$, i.e. it is more likely to succeed if the collective Euclidean distance of the training set to the new input is small. This is similar to the conditional measurement in the basis encoding routine. In the worst case, $\tilde{x}_i\approx -x_i^{(m)}$ for all $m=1...M$, and acceptance will be very unlikely. However, this means also here that the new input is 'far away' from the dataset, an indicator for the low expressive power of a classification algorithm based on distances.

If the conditional measurement was successful, the result is proportional to

$$\frac{1}{2}\sum_{m=1}^{M}\sum_{i=1}^{N}\left(\tilde{x}_i+x_i^{(m)}\right)|i\rangle|y^{(m)}\rangle|m\rangle.$$

The probability of measuring the class qubit $|y^{(m)}\rangle$ in state 0 and predict class -1 is given by

$$p(\tilde{y}=-1)=p(|y^{(m)}\rangle=|0\rangle)=\frac{1}{4}\sum_{m|y^m=0}|\tilde{\mathbf{x}}+\mathbf{x}^{(m)}|^2=\sum_{m|y^m=0}1-\frac{1}{4}|\tilde{\mathbf{x}}-\mathbf{x}^{(m)}|^2.$$

The last equality has been already used above and is easy to show:

$$\begin{aligned}
\frac{1}{4}|\tilde{\mathbf{x}}+\mathbf{x}^{(m)}| &= \frac{1}{4}\sum_{i=1}^{N}|\tilde{x}_i+x_i^{(m)}|^2 \\
&= \frac{1}{4}\sum_{i=1}^{N}|\tilde{x}_i|^2+\frac{1}{2}\sum_{i=1}^{N}(\tilde{x}_i)^*x_i^{(m)}+\frac{1}{2}\sum_{i=1}^{N}\tilde{x}_i(x_i^{(m)})^*+\frac{1}{4}\sum_{i=1}^{N}|x_i^{(m)}|^2 \\
&= \frac{1}{2}+\frac{1}{2}(\tilde{x}_i)^*x_i^{(m)}+\frac{1}{2}\tilde{x}_i(x_i^{(m)})^* \\
&= 1-\frac{1}{4}\sum_{i=1}^{N}|\tilde{x}_i|^2+\frac{1}{2}\sum_{i=1}^{N}(\tilde{x}_i)^*x_i^{(m)}+\frac{1}{2}\sum_{i=1}^{N}\tilde{x}_i(x_i^{(m)})^*-\frac{1}{4}\sum_{i=1}^{N}|x_i^{(m)}|^2 \\
&= \frac{1}{4}\sum_{i=1}^{N}|\tilde{x}_i-x_i^{(m)}|^2 \\
&= 1-\frac{1}{4}|\tilde{\mathbf{x}}-\mathbf{x}^{(m)}|
\end{aligned}$$

Expressing the probability to predict class -1 by the squared distance shows that it is higher the closer the class -1 training vectors are to the input. The kernel is hence given by $\kappa(\tilde{\mathbf{x}}-\mathbf{x}^{(m)})=1-\frac{1}{4}|\tilde{\mathbf{x}}-\mathbf{x}^{(m)}|^2$, and is therefore a polynomial kernel.

Again, as an alternative the m-register could be measured to extract training inputs and their classes with a probability depending on their squared distance to the new input.

Routine 1.1.2: Quantum Linear Regression Algorithm (amplitude encoding)

Goal: Implement the weighed nearest neighbour algorithm for binary pattern classification on a quantum computer.

Runtime: $\mathcal{O}(\frac{1}{p_{\text{acc}}})$ excluding state preparation [Acceptance of conditional measurement p_{acc} as described in text]

Input: A quantum superposition of training vectors as well as the new input in amplitude encoding.

Output: Class of new output $\tilde{y}$

PROCEDURE

Given: $\frac{1}{\sqrt{2}} \sum_{m=1}^{M} \left(|0\rangle|\psi_{\tilde{\mathbf{x}}}\rangle + |1\rangle|\psi_{\mathbf{x}^m}\rangle\right)|y^m\rangle|m\rangle$

1: $\frac{1}{2} \sum_{m=1}^{M} \Big(|0\rangle\big[|\psi_{\tilde{\mathbf{x}}}\rangle + |\psi_{\mathbf{x}^m}\rangle\big] + |1\rangle\big[|\psi_{\tilde{\mathbf{x}}}\rangle - |\psi_{\mathbf{x}^m}\rangle\big]\Big)|y^m\rangle|m\rangle$ ▷ after Hadamard on ancilla

2: $\frac{1}{2} \sum_{m=1}^{M} |\psi_{\tilde{\mathbf{x}}+\mathbf{x}^{(m)}}\rangle|y^{(m)}\rangle|m\rangle$ ▷ after conditional measurement

3: **return** $\begin{cases} -1 \text{ if } p(|y^m\rangle = |0\rangle) > p(|y^m\rangle = |1\rangle) \\ 1 \text{ else} \end{cases}$ ▷ measure the class qubit y^m

Bibliography

[1] Lev Landau and Evgeny Lifshitz. *Quantum Mechanics: Non-Relativistic Theory*, volume 3 of *Course of Theoretical Physics*. Butterworth-Heinemann, 2005.

[2] Andrew Whitaker. *Einstein, Bohr and the quantum dilemma: From quantum theory to quantum information*. Cambridge University Press, 2006.

[3] Albert Einstein. Über die von der molekularkinetischen theorie der wärme geforderte bewegung von in ruhenden flüssigkeiten suspendierten teilchen. *Annalen der Physik*, 322(8):549–560, 1905.

[4] Arnab Das and Bikas K Chakrabarti. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80(3):1061, 2008.

[5] Kristen L Pudenz and Daniel A Lidar. Quantum adiabatic machine learning. *Quantum Information Processing*, 12(5):2027–2070, 2013.

[6] Vasil Denchev, Nan Ding, Hartmut Neven, and Svn Vishwanathan. Robust classification with adiabatic quantum optimization. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 863–870, 2012.

[7] Hartmut Neven, Geordie Rose, and William G Macready. Image recognition with an adiabatic quantum computer i. mapping to quadratic unconstrained binary optimization. *arXiv preprint arXiv:0804.4457*, 2008.

[8] Bryan OGorman, R Babbush, A Perdomo-Ortiz, Alan Aspuru-Guzik, and Vadim Smelyanskiy. Bayesian network structure learning using quantum annealing. *The European Physical Journal Special Topics*, 224(1):163–188, 2015.

[9] Steven H Adachi and Maxwell P Henderson. Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*, 2015.

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

[11] Rodion Neigovzen, Jorge L Neves, Rudolf Sollacher, and Steffen J Glaser. Quantum pattern recognition with liquid-state nuclear magnetic resonance. *Physical Review A*, 79(4):042321, 2009.

[12] Hans J Briegel and Robert Raussendorf. Persistent entanglement in arrays of interacting particles. *Physical Review Letters*, 86(5):910, 2001.

[13] Michael A Nielsen. Cluster-state quantum computation. *Reports on Mathematical Physics*, 57(1):147–161, 2006.

[14] Philip Walther, Kevin J Resch, Terry Rudolph, Emmanuel Schenck, Harald Weinfurter, Vlatko Vedral, Markus Aspelmeyer, and Anton Zeilinger. Experimental one-way quantum computing. *Nature*, 434(7030):169–176, 2005.

[15] Robert Prevedel, Andr Stefanov, Philip Walther, and Anton Zeilinger. Experimental realization of a quantum game on a one-way quantum computer. *New Journal of Physics*, 9(6):205, 2007.

[16] Sang Min Lee, Hee Su Park, Jaeyoon Cho, Yoonshik Kang, Jae Yong Lee, Heonoh Kim, Dong-Hoon Lee, and Sang-Kyung Choi. Experimental realization of a four-photon seven-qubit graph state for one-way quantum computation. *Opt. Express*, 20(7):6915–6926, Mar 2012.

[17] M. S. Tame, R. Prevedel, M. Paternostro, P. Böhi, M. S. Kim, and A. Zeilinger. Experimental realization of deutsch's algorithm in a one-way quantum computer. *Phys. Rev. Lett.*, 98:140501, 2007.

[18] M. S. Tame, B. A. Bell, C. Di Franco, W. J. Wadsworth, and J. G. Rarity. Experimental realization of a one-way quantum computer algorithm solving simon's problem. *Phys. Rev. Lett.*, 113:200501, Nov 2014.

[19] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015.

[20] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 2003.

[21] Dan Ventura and Tony Martinez. Quantum associative memory. *Information Sciences*, 124(1):273–296, 2000.

[22] Carlo A Trugenberger. Probabilistic quantum memories. *Physical Review Letters*, 87:067901, Jul 2001.

[23] I. M. Georgescu, S. Ashhab, and Franco Nori. Quantum simulation. *Review of Modern Physics*, 86:153–185, 2014.

[24] Phillip Kaye and Michele Mosca. Quantum networks for generating arbitrary quantum states. In *Proceedings of the International Conference on Quantum Information*. ICQI, 2001. preprint quant-ph/0407102v1.

[25] Barbara Kraus, HP Büchler, S Diehl, A Kantian, A Micheli, and P Zoller. Preparation of entangled states by quantum markov processes. *Physical Review A*, 78(4):042307, 2008.

[26] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009.

[27] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Phys. Rev. Lett.*, 113:130503, Sep 2014.

[28] Anupam Prakash. *Quantum Algorithms for Linear Algebra and Machine Learning*. PhD thesis, EECS Department, University of California, Berkeley, Dec 2014.

[29] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical Review Letters*, 109(5):050505, 2012.

[30] B David Clader, Bryan C Jacobs, and Chad R Sprouse. Preconditioned quantum linear system algorithm. *Physical review letters*, 110(25):250504, 2013.

[31] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *arXiv preprint quant-ph/0005055*, 2000.

[32] Matthew McKague. On the power of quantum computation over real hilbert spaces. *International Journal of Quantum Information*, 11(01):1350001, 2013.

[33] Artur K. Ekert, Carolina Moura Alves, Daniel K. L. Oi, Michał Horodecki, Paweł Horodecki, and L. C. Kwek. Direct estimations of linear and nonlinear functionals of a quantum state. *Phys. Rev. Lett.*, 88:217901, 2002.

[34] Hirotada Kobayashi, Keiji Matsumoto, and Tomoyuki Yamakami. Quantum merlin-arthur proof systems: Are multiple merlins more helpful to arthur? In *Algorithms and Computation*, pages 189–198. Springer, 2003.

[35] Daniel S Abrams and Seth Lloyd. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Physical Review Letters*, 83(24):5162, 1999.

[36] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10:631–633, 2014.

[37] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. Prediction by linear regression on a quantum computer. *Physical Review A*, 94(2):022342, 2016.

[38] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.