Quantum k-Nearest Neighbours Algorithm

Afham

A dissertation submitted for the partial fulfilment of BS-MS dual degree in Science

Supervised by
Dr Sandeep K. Goyal

May 2020

Certificate of Examination

This is to certify that the dissertation titled **Quantum k-nearest neighbours algorithm** submitted by **Afham (Reg. No. MS15024)** for the partial fulfilment of BS-MS dual degree programme at Indian Institute of Science Education and Research, Mohali has been examined by the thesis committee duly appointed by the Institute. The committee finds the work done by the candidate satisfactory and recommends that the report be accepted.

Dr Kinjalk Lochan Dr Sanjib Dey Dr Sandeep K. Goyal
(Supervisor)

Declaration

The work presented in this dissertation has been carried out by me under the guidance of Dr Sandeep K. Goyal at the Indian Institute of Science Education and Research, Mohali.

This work has not been submitted in part or in full for a degree, a diploma, or a fellowship to any other university or institute. Whenever contributions of others are involved, every effort is made to indicate this clearly, with due acknowledgement of collaborative research and discussions. This thesis is a bonafide record of original work done by me and all sources listed within have been detailed in the bibliography.

Afham

(Candidate)

Date:

In my capacity as the supervisor of the candidate's project work, I certify that the above statements by the candidate are true to the best of my knowledge.

Dr Sandeep K. Goyal

(Supervisor)

Date

Acknowledgements

This work would not have been possible without the help and support of a lot of people. It would be difficult to put my gratitude in words, yet I try nonetheless.

Firstly, I must thank Dr Sandeep K. Goyal for prompt guidance, ample freedom, and insightful discussions. I must thank Afrad Basheer for not only the collaboration and hefty contributions to this work, but also the never-ending presence. I must also thank him for reining me in with the rigor of a mathematician whenever I let my *physicist's intuition* run amok. I thank Dr Sanjib Dey and Dr Kinjalk Lochan for the encouragement via evaluation.

I thank my parents and sisters for the support they've always provided me. I appreciate Afrad Basheer, Joseph Maliakkal, and Dany Joshua for the weekly *Age of Empires* nights. I thank the members of *Poker kings* - Asif, Ashish, Sidharth, Niteesh, Alan, and Athul for the card nights and the company. I am grateful to Mannathu Gopikrishnan and Bharadwaj Varma, my compatriots in the *Cabibbo Kobayashi Maskawa Matrix* for everything ranging from *chai chats* on physics to the ancient Achaemenid empire. I thank the members of our group *Quantum dreams* for all conversations and discussions, and Dr Sandeep K. Goyal and Anshu Mittal for hosting us. I thank Lipika, Amisha for the chats, and Ardra for the breakfast conversations. I thank Dr Dipanjan Chakraborty for all the chats, advice, and pep talks.

As my five years at IISER Mohali come to an end, I thank all friends, old and new, for all the memories, good and bad. IISER Mohali will always remain my home, and the people here, my family.

Abstract

Quantum computing (QC) and machine learning (ML) are two disciplines experiencing tremendous growth these days. Machine learning works through picking up patterns in huge amounts of data to build a model, which it uses upon unseen data to make predictions. Various ML algorithms are nothing but different ways through which the machine can find interesting patterns in data. Quantum computers promises a different paradigm of computing - one where certain problems, such as prime factorisation, could be solved faster than any classical computer. We propose a quantum analog of the classical k-nearest neighbour (kNN) machine learning algorithm. Our algorithm uses Fredkin gates and wavefunction collapse upon measurement to estimate the fidelity simultaneously between the test state and all the train states, which is advantageous over its classical counterpart in certain situations. The quantum kNN algorithm presented here is capable of dealing with completely unknown test states encoded in quantum systems. We discuss the cost and analysis of our algorithm and compare it with other similar methods. As an example, we test this algorithm on the problem of classifying n-qubit pure entangled states.

List of Figures

1.0.1 Schematic representation of different kinds of machine learning 3

2.0.1 Illustration of k-nearest neighbours algorithm 8

2.2.1 Circuit diagram for the Swap test 12

2.2.2 Fredkin gate decomposition 12

2.3.1 Illustration of entanglement classes for two qubits 13

3.1.1 Circuit diagram for Quantum kNN algorithm 15

4.2.1 Histogram of QKNN algorithm for entanglement classification 24

List of Tables

4.1 Entanglement classification using classical kNN classifier 23

4.2 Comparing classical and quantum kNN classifiers on entanglement classification 24

Contents

Acknowledgements **iv**

Abstract **v**

List of Figures **vi**

List of Tables **vii**

1 Introduction **1**

2 Background **7**

2.1 Classical kNN algorithm 7

2.2 Swap Test 9

2.3 Entanglement classes 12

3 Quantum k-Nearest neighbours Algorithm **15**

3.1 The algorithm 15

3.2 Cost and benefits 19

4 Entanglement classification using classical and quantum kNN **22**

4.1 Entanglement classification using classical kNN 22

4.2 Entanglement classification using QKNN 23

5 Conclusion **25**

Chapter 1

Introduction

Quantum theory, when it was first formulated in early 20th century, was no more than an attempt at explaining nature at a microscopic level, which would lead to the expansion of the collective human knowledge. Yet, as we learned quantum theory more and more, we also saw ways to exploit it to benefit (and also to besiege) humankind. Technological marvels such as transistors, nuclear weapons, nuclear energy had quantum theory as its theoretical underpinning. It was termed as the first quantum revolution. Today, it is the era of the second quantum revolution, where we seek to manipulate individual atoms and photons to achieve complex tasks. One could argue that the second quantum revolution was founded by Richard Feynman, when he said that to simulate quantum systems, one must have quantum computers as regular computers, now termed *classical* computers simply won't do. This thesis stems from the author's deep interest in quantum information, computing, and algorithms. In this thesis, we present a quantum machine learning algorithm - a machine learning algorithm that runs on a quantum computer - named quantum k-nearest neighbours algorithm.

Quantum computing is a research avenue that shows great promise to revolutionise the way we compute. The basic idea behind quantum computing is to use concepts from quantum physics, such as entanglement and superposition, to solve problems better than classical computers. This could be designing quantum algorithms for various tasks such as prime factorization [Shor 99] and matrix inversion [Harrow 09], quantum simulation of molecules (which can be intractable for classical computers) [O'Malley 16], quantum cryptography - which looks at methods of breaking current cryptography protocols using quantum computing and constructing cryptography protocols, classical and quantum, that are quantum-proof [Pirandola 19]. Another field that comes under the broad

umbrella of QC is quantum annealing, which looks at ways of solving optimisation problems [Santoro 06].

Currently, in quantum computing, we're at the stage where the technology is leaving laboratories and entering the industry. Recently Google's quantum computing team has demonstrated *quantum supremacy*, a point in technology where a quantum computer solves a particular task faster than any classical computer[Arute 19].

Machine learning, though has its theoretical underpinnings beginning from 1960s, has really exploded into our daily lives in the past couple of decades thanks to the immense advancement in hardware capabilities. Applications of ML are today ubiquitous, from everyday Netflix suggestions to Google's AlphaGo defeating the world's best Go player. Machine learning works through picking up patterns in huge amounts of data to build a model, which it uses upon unseen data to make predictions. Various ML algorithms such as neural networks, k-means classifier, support vector machines, random forests among others are nothing but different ways through which the machine can find interesting patterns in data.

Machine learning can be broadly divide into two categories. Supervised and unsupervised machine learning. In supervised machine learning, one builds a model by first feeding it with labelled data called training data.

For example, if one were to build a model to distinguish between physicists and biologists based on their facial features, one would first feed the model, say, a 100 images of each class. Once this training phase is completed, we then test the accuracy of the model by feeding it previously unseen pictures of physicists and biologists. If the model can distinguish between physicists and biologists with an acceptable degree of accuracy, then one might make the conclusion that there is some visible yet hidden features on our faces that screams whether we're physicists or biologists. Or it could be something as simple as that perhaps in most of the pictures the biologists were in their labs and the physicists were next to their whiteboards and the model simply decided based on the background images and not on the faces.

This example is well suited to explain another aspect of machine learning - we know it works, but for algorithms such as neural networks, we do not know *why* it works. Thereby, choosing a model has multiple aspects - one cannot always go behind accuracy. Simpler models maybe less accurate, but might be more explainable. In unsupervised machine learning, the algorithm is just fed data with no labels. The algorithm must decide what how many classes to create and the membership of each

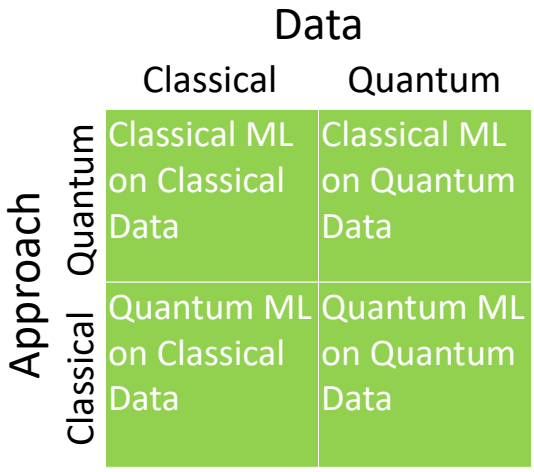

Figure 1.0.1: Machine learning can be roughly divided on the basis of the data it's working on and the approach it's taking. Both the approach and the data can be classical or quantum. Hence we can divide it into four different kinds.

class. Examples include k-means clustering among others.

Recent years have seen significant advancements in the fields of quantum computing (QC) [Steane 98, O'brien 07, Li 01, Ladd 10, Häffner 08, Kok 07] and machine learning (ML) [LeCun 15, Le 13, Voulodimos 18, Simeone 18, Dey 16, Benuwa 16]. While quantum computing enables us with a new paradigm for computing, machine learning, armed with big data and powerful hardware, shows the depth of classical computing. The union of these two fields recently has led to the birth of a new field — quantum machine learning (QML) [Biamonte 17, Wittek 14, Schuld 14, Arunachalam 17].

QML aims to tackle the ever-growing big data by employing quantum computers in hopes that its surreal properties, such as superposition and entanglement would lead to methods that can process data much faster than classical computers. Not only is QML capable of providing massive speedup over the classical counterparts, it can also handle quantum data efficiently [Carrasquilla 17, Wang 17, Lu 17].

Quantum machine learning can be neatly summarised into a matrix (Fig:1.0.1). The most ubiquitous approach is of course using classical ML on classical data. Recently, we have seen classical ML being used on quantum data to solve problems in physics. Using quantum ML on classical data would be among the most useful applications of quantum machine learning when the technology comes of age. It is expected that the quantum computers will be able to speed up machine learning algorithms hence will be able to process data faster[Biamonte 17]. The last kind we can think of is quantum

ML approach on quantum data. The quantum kNN algorithm presented here falls under this kind and we also present the simulations of this kind - where we use our QKNN algorithm to classify quantum pure states on the basis of its entanglement.

Several classical ML algorithms have been ported to quantum versions, such as quantum principle component analysis [Lloyd 14], which allows one to extract the principle components (the eigenvectors corresponding to the largest eigenvalues) of the convariance matrix which is encoded as a density matrix through density matrix exponentiation, quantum k means clustering [Lloyd 13, Biamonte 17], where the aim is to cluster unlabelled data in to k clusters. Other ported algorithms include quantum support vector machines [Rebentrost 14], which offers a mapping to an exponentially large feature space to compute the inner product as compared to classical method. It is also conjectured that the method they provide is impossible to simulate classically. Each of the ports presented here has its vices and virtues. In this article, we propose a quantum version of the k-nearest neighbour (kNN) algorithm.

kNN algorithm is a simple supervised ML algorithm used extensively for pattern recognition and classification [Cover 67, Samworth 12, Nigsch 06]. This algorithm rest on the assumption that two states close to each other are more likely to belong to the same class or pattern. In this algorithm, the computer is trained with a set of train states whose class labels are known. The test state with the unknown label is compared with the train states, and a k number of the nearest neighbours from the train states are identified for the given test state. The label of the test state is determined upon majority voting.

The most computationally expensive step in the kNN or classical kNN algorithm is to determine the distance between the test state and all the train states, which makes the kNN algorithm slow. Each state (train or test) is represented by a vector of complex numbers. As the number of train states and the size of the state vectors increases, kNN becomes more expensive. To classify vector of dimension N by comparing it to a set of train vectors of cardinality M, we need to carry out MN multiplication operations. Multiplications and the sorting in order to get the nearest neighbours gives classical kNN algorithm a complexity of $\mathcal{O}(MN)$.

Several quantum machine learning algorithms have been proposed which exploit the broad concept of nearest neighbours; for example, quantum nearest neighbour algorithm [Wiebe 15] and quantum k-nearest neighbour algorithm using Hamming distance [Ruan 17]. Although these quantum versions of classical kNN algorithms have

their merits, they also have severe limitations. For example, the method presented in [Wiebe 15] requires two oracles and multiple calls from these oracles. This also implicitly requires knowledge of the state to be classified, hence limited to only classical data. Furthermore, this algorithm is restricted to a single neighbour for classification, which limits its accuracy. The quantum ML algorithm presented in [Ruan 17] requires complete knowledge of the test state which restricts its impact.

Here, we propose a novel quantum k-nearest neighbour (QKNN), a quantum analog of classical kNN algorithm. In this algorithm, we exploit the superposition properties of the quantum states and collapse of the wavefunction upon measurement to calculate the distance between the test state and all the train states simultaneously. In particular, we use the Swap test [Buhrman 01] to calculate the fidelity simultaneously between the test state and all the train states which makes our algorithms much faster than its classical counterpart. Another important advantage of QKNN is that it does not require any kind of information about the test state. Therefore, it is eligible to handle quantum as well as classical data. As an example, we test QKNN on the problem of classifying pure multipartite entangled states. We compare the results with the classical kNN algorithm and find that both the algorithms yield the same accuracy; however, the classical algorithm requires the knowledge of the test state, whereas QKNN does not.

QKNN is capable of estimating the distance and find the nearest neighbours for any unknown quantum test state; therefore, it can handle quantum as well as classical data. Unlike existing quantum nearest neighbour algorithms, our algorithm has the capability of classifying unknown states, thereby would be of great use in situations where it is costly to learn the states. These situations include cases involving quantum data, where expensive processes such as quantum state tomography, whose complexity grows exponentially with the number of quantum systems [Aaronson 07], are required to gain complete knowledge about the states. This feature invariably makes QKNN better than the existing quantum kNN algorithms. The advantage of QKNN can be seen from the example of classification of multipartite pure entangled states where it is capable of classifying the entangled states without any prior information about the given state.

This thesis is structured as follows. First, we begin by laying out the necessary background, which includes introduction to the Swap test, classical kNN algorithm, and pure state entanglement classes. We then present our quantum kNN algorithm in chapter 3 and in chapter 4 we present entanglement classification using QKNN algorithm. We

then conclude on chapter 5.

Chapter 2

Background

In this chapter, we present the relevant background of the classical kNN algorithm. We also present the entanglement classes, and the Swap test which is an integral part of our QKNN algorithm.

2.1 Classical kNN algorithm

Classical k-nearest neighbour or kNN algorithm is a supervised classical machine learning algorithm to classify test states (say $\{u_n\}$) whose labels are to be determined, by comparing their distance to the train states (say $\{v_m\}$), whose labels are known to us [Cover 67]. kNN has been applied successfully to a multitude of problems [Liao 02, He 07, Mani 03, Imandoust 13, Bijalwan 14]. Being a simple algorithm, kNN also allows us to reason about the structure of the data we are working with.

Both the test states and the trains states are r-dimensional real or complex vectors. Any bona fide definition of a distance measure can be used for the purpose of kNN algorithm. Most common distance measures include Euclidean distance $d(\mathbf{u}, \mathbf{v})$ and cosine similarity $(\mathbf{u}, \mathbf{v})$ (which reduces to inner product for normalised states), which are defined as:

$$d(\mathbf{u}, \mathbf{v}) = \left(\sum_{i}^{r} |u_i - v_i|^2 \right)^{1/2}, \tag{2.1}$$

$$(\mathbf{u}, \mathbf{v}) = \frac{\sum_i^r u_i^* v_i}{\sqrt{\sum_i^r u_i^2} \sqrt{\sum_i^r v_i^2}}. \tag{2.2}$$

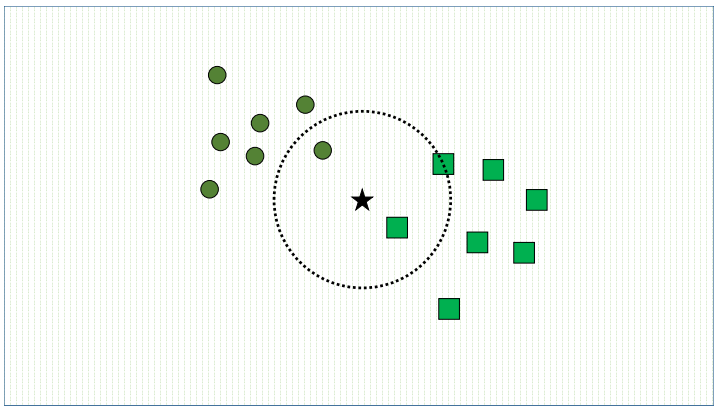

Figure 2.0.1: Choosing a k = 3 neighbourhood. Here circle and square represents two different classes and star represents the unknown state whose label is to be determined. On choosing $k = 3$, we classify it as a 'square' point.

Here, **u** and **v** are r-dimensional complex vectors and u_i, v_i are their components, respectively.

Another popular choice for the distance measure is fidelity $F(u, v)$ which is the square modulus of the cosine similarity, i.e., $F(u, v) = |(u, v)|^2$. In quantum setup, the states are represented in the Dirac notation by $|u\rangle$ and their duel space vectors by $\langle u|$. Fidelity between two such states $|u\rangle$ and $|v\rangle$ is simply [Jozsa 94]:

$$F(u, v) = |\langle u|v\rangle|^2. \tag{2.3}$$

Fidelity arises naturally as a criterion to determine neighbours in any quantum protocol. For normalized states $|u\rangle$ and $|v\rangle$, one convenient measure of distance between them can be

$$D(u, v) = 1 - F(u, v). \tag{2.4}$$

Therefore, higher the fidelity between the two states, closer they will be.

The rationale behind kNN is that data points that are close together, with respect to some distance measure, must be similar. Formally, the kNN algorithm consists of the following steps:

1. For each test state (whose label is to be determined), compute its distance to the train states whose labels are known.

2. Choose the k number of neighbours which are nearest to the test point.

3. Conduct a majority voting and assign the label of the majority to the test point.

Although the kNN algorithm is simple to understand and easy to implement, there are several limitations and shortcomings of the algorithm. As the number of train data points and the dimension of the state vectors grows, kNN can quickly turn intractable for classical computers. Classification of an N dimensional test state by comparing with M train states requires $\mathcal{O}(MN)$ multiplication operations. Finding the nearest neighbours will require sorting of M number of distance which requires $\mathcal{O}(M \log M)$ operations. Furthermore, the choice of the number k is also highly debated. There is no general way of choosing k and usually, hyperparameter tuning is done to choose the best possible k. For a dicussion on how to learn the best value of k, see [Zhang 17].

2.2 Swap Test

Since computing distance between the test states and the train states is an integral part of the kNN algorithm, we require a quantum subroutine, which can estimate the distance between two quantum states. The swap test [Buhrman 01] is a quantum algorithm that can be used to statistically estimate the fidelity of two pure states $|\psi\rangle$ and $|\phi\rangle$, i.e., $F = |\langle\psi|\phi\rangle|^2$.

In order to implement the swap test, we need three registers prepared in states $|0\rangle$, $|\psi\rangle$ and $|\phi\rangle$, respectively (see Fig. 2.2.1). The initial combined state of the three registers is

$$|R\rangle = |0\rangle \otimes |\psi\rangle \otimes |\phi\rangle \,. \tag{2.5}$$

Next we apply a Hadamard operation H on the first register followed by a control swap C_S on the other two registers where the first register serves as the control system. The action of the Hadamard operation H on $|0,1\rangle = (|0\rangle \pm |1\rangle)/\sqrt{2}$. Whereas the action of C_S reads

$$\begin{aligned} C_S\,|0\rangle\,|a\rangle\,|b\rangle &= |0\rangle\,|a\rangle\,|b\rangle\,, \\ C_S\,|1\rangle\,|a\rangle\,|b\rangle &= |1\rangle\,|b\rangle\,|a\rangle\,. \end{aligned} \tag{2.6}$$

The total state of the system after these two operations reads

$$|\bar{R}\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|\psi\rangle\,|\phi\rangle + |1\rangle\,|\phi\rangle\,|\psi\rangle). \tag{2.7}$$

Applying another Hadamard operation H on the first qubit followed by a measurement on the first qubit in the $\{|0\rangle, |1\rangle\}$ results in 0 and 1 with probabilities

$$P(0) = \frac{1}{2} + \frac{1}{2}|\langle\psi|\phi\rangle|^2, \tag{2.8}$$

$$P(1) = \frac{1}{2} - \frac{1}{2}|\langle\psi|\phi\rangle|^2. \tag{2.9}$$

The quantity $P(0) - P(1)$ gives us the desired fidelity.

In this whole protocol to estimate the fidelity between two n-qubit states, the more resource-intensive component is the controlled swap operation. A controlled swap operation on n-qubit system can be realized using n number of Fredkin gates. A Fredkin gate is a three-qubit gate where all three registers - the control and the two registers to be swapped, are single qubits.

A Fredkin gate can be decomposed into two-qubit gates as shown in Fig. 2.2.2, where V is the single-qubit gate [Smolin 96]:

$$V = \frac{e^{i\pi/4}}{\sqrt{2}} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}. \tag{2.10}$$

We can achieve the control swap operation on n qubits with no more than n Fredkin gates by using each of the Fredkin gate to swap corresponding qubits in the two registers $|a\rangle$ and $|b\rangle$ with the first register being the control qubit for all the Fredkin gates. This is done as shown below.

To show two n-qubit registers can be control-swapped using n Fredkin gates, it is sufficient to show that two n-qubit registers can be swapped using n swap gates. The action of swap gate S is defined as $S|a\rangle|b\rangle = |b\rangle|a\rangle$ where $|a\rangle$ and $|b\rangle$ are single qubit states.

Let $|x\rangle$ and $|y\rangle$ be n-qubit pure states which can be expanded in the standard basis

as:

$$|x\rangle = \sum_{i=0}^{2^n-1} x_i|i\rangle, \tag{2.11}$$

$$|y\rangle = \sum_{i=0}^{2^n-1} y_i|i\rangle. \tag{2.12}$$

Each basis state $|i\rangle$ can be expressed in its binary decomposition as

$$|i\rangle = |i_1\rangle|i_2\rangle \dots |i_n\rangle = |i_1 i_2 \dots i_n\rangle, \tag{2.13}$$

where each $|i_j\rangle$ are single qubits and i_j can take a value of 0 or 1.

Let S_k be a swap gate acting on the k^{th} qubits of the two registers. The action of S_k on the basis states $|i\rangle$ and $|j\rangle$ is

$$\begin{aligned} S_k|i\rangle|j\rangle &= S_k|i_1 i_2 \dots i_k \dots i_n\rangle|j_1 j_2 \dots j_k \dots j_n\rangle \\ &= |i_1 i_2 \dots j_k \dots i_n\rangle|j_1 j_2 \dots i_k \dots j_n\rangle. \end{aligned} \tag{2.14}$$

Hence, the action of $\bar{S} = S_1 S_2 \dots S_n$ is

$$\begin{aligned} \bar{S}|i\rangle|j\rangle &= \bar{S}|i_1 i_2 \dots i_n\rangle|j_1 j_2 \dots j_n\rangle \\ &= |j_1 j_2 \dots j_n\rangle|i_1 i_2 \dots i_n\rangle \\ &= |j\rangle|i\rangle. \end{aligned} \tag{2.15}$$

For two general n-qubit states $|x\rangle$ and $|y\rangle$, we have

$$\begin{aligned} \bar{S}|x\rangle|y\rangle &= \bar{S} \sum_{i,j=0}^{2^n-1} x_i y_j |i\rangle|j\rangle \\ &= \sum_{i,j=0}^{2^n-1} x_i y_j \bar{S}|i\rangle|j\rangle = \sum_{i,j=0}^{2^n-1} x_i y_j |j\rangle|i\rangle \\ &= \sum_{i,j=0}^{2^n-1} y_j x_i |j\rangle|i\rangle = |y\rangle|x\rangle. \end{aligned} \tag{2.16}$$

So we may swap two quantum registers of equal size n swapping corresponding qubits of the two registers using two qubit swap gates. Hence, we may control-swap

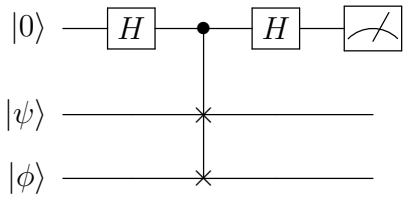

Figure 2.2.1: Circuit diagram for Swap test. Here H is the Hadamard operation.

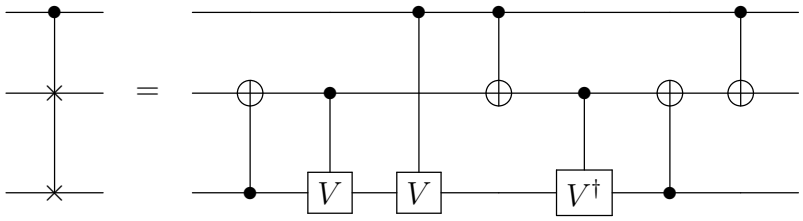

Figure 2.2.2: Decomposition of Fredkin gate in two-qubit operations. Apart from the standard CNOT gate, we use the control V (C_V) gate where V is a $\pi/4$ rotation about σ_x.

two quantum registers of equal size by control swapping corresponding qubits of the two registers using Fredkin gates, with the same qubit as control qubit for all n Fredkin gates.

Note that the swap test requires no knowledge of the states whose overlap is being measured. Hence, in principle, it is possible to compute the fidelity between two unknown n qubit states with a total number of $2n + 1$ qubits.

2.3 Entanglement classes

In this section, we discuss the entanglement classes in pure n-partite quantum states. For simplicity, we restrict ourselves to n-qubit systems only. We begin with $n = 2$ case. A pure two-qubit quantum state $|\Phi\rangle$ is called separable or product state if and only if it can be written as a tensor product of two pure states corresponding to individual subsystems, i.e.,

$$|\Phi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle \,. \tag{2.17}$$

If the state $|\Phi\rangle$ is not of the form (2.17) then its an entangled states. In two-qubit or bipartite systems a pure state is either separable or entangled. However, the same statement is not true in multipartite systems. A pure n-qubit quantum state $|\Psi\rangle$ is separable

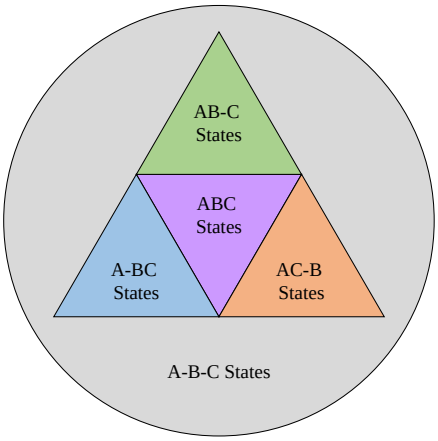

Figure 2.3.1: Representative diagram of entanglement classes in three qubits. Subspaces separated by a hyphen ('-') are separable subspaces and those that are not are entangled subspaces.

only if it can be written as the tensor product of n quantum states as

$$|\Psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle \,. \tag{2.18}$$

Such states are also called n-separable states [Horodecki 09]. Another way to look at these states is the following: a pure state $|\Psi\rangle$ is an n-separable state if it is separable across all the possible bipartitions of the n qubits. If this condition is violated then the state is no longer n-separable. Some states can be entangled in certain bipartitions and separable in others. Some states are entangled in all the bipartitions. This motivates a classification of n-partite quantum states on the bases of entanglement.

For two-qubit states, there are only two classes – separable and entangled states. Three qubit states can be divided into three classes: (i) three-separable states, (ii) states that are separable in two bipartitions and entangled in one and (iii) the states which are entangled in all the three bipartitions. The class (ii) can further be divided into three subclasses depending on which bipartition is entangled. If A, B, and C represent the three-qubits, and if we represent two subsystems that are separable as A-B and two subsystems that are entangled as AB, then the entanglement classes can be written as {A-B-C, AB-C, A-BC, AC-B, ABC} (Fig: 2.3.1). Note that we do not distinguish between W states and GHZ states defined in [Dür 00] and keep them in the same class ABC.

The same classification of the entanglement can be extended to n number of qubits. The question that is relevant to us is the following: given an n-qubit arbitrary state, is there a way to label it according to its entanglement class. In the chapter 4, we show that classical kNN algorithm can classify these state with very high accuracy. Furthermore, the same accuracy can be achieved by our QKNN algorithm without the knowledge of the given quantum state, establishing the advantage of QKNN over classical kNN algorithm.

Chapter 3

Quantum k-Nearest neighbours Algorithm

In this chapter, we introduce our new QKNN machine learning algorithm. We analyze the cost and benefits of the QKNN over classical kNN algorithm. As an example, in the next chapter, we simulate this algorithm on classical computers for the problem of classifying bipartite entangled states.

3.1 The algorithm

Let $|\psi\rangle$ be the n-qubit test state, whose label is to be determined. The set $\{|\phi_i\rangle\}$ contains all the train states of the same dimension. Each of the train states is indexed, which we

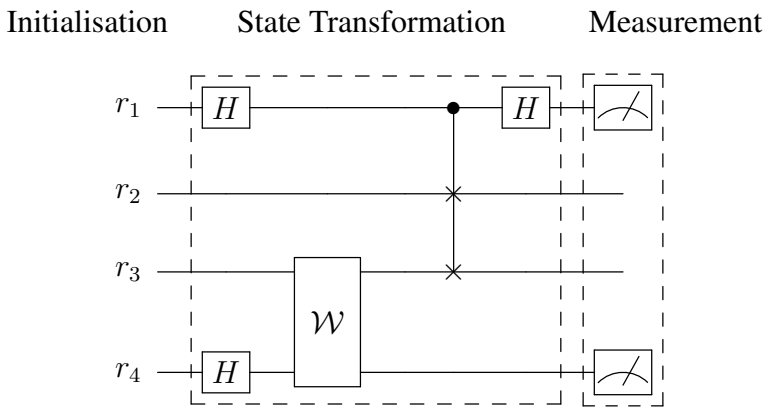

Figure 3.1.1: Circuit for the quantum kNN algorithm. The three steps of the algorithm are demarcated. $\mathcal{W}$ is the oracle as defined in equation 3.2.

refer to as i which need not represent their label. Two or more states with different indices i can have the same label.

Implementation of our algorithm requires four registers r_1, r_2, r_3, and r_4. The r_1 register is a single-qubit system, r_2 and r_3 are n-qubit systems and r_4 is an m-qubit system where its dimension $2^m = M$ is the cardinality of the set $\{|\phi_i\rangle\}$. The QKNN algorithm consists of three major steps:

1. Initialization: initialize the registers in the required state vectors $|R\rangle$.
2. State transformation: transforming the initial state to arrive at the state $|\bar{R}\rangle$, which is suitable for fidelity estimation.
3. Measurements: performing measurements to estimate the fidelity.

We present each step in detail below.

Initialisation

In this step of the algorithm, we prepare the four registers in a suitable state. For our purpose, we prepare the r_1 is the state $|0\rangle$, r_2 is prepared in the test state, i.e., $|\psi\rangle$, r_3 and r_4 are prepared in the states $|0\rangle^{\otimes n}$ and $|0\rangle^{\otimes m}$, respectively, where $n = \log N$ and $m = \log M$. Hence, the initial state of the total system is $|R\rangle = |0\rangle |\psi\rangle |0\rangle^{\otimes n} |0\rangle^{\otimes m}$. Although, for the sake of this algorithm we prepare the register r_2 is the state $|\psi\rangle$ but in real situations we are given an n-qubit system in an unknown state $|\psi\rangle$. The advantage of our QKNN algorithm is that it does not require the knowledge of the state $|\psi\rangle$.

State transformation

In the second step of the algorithm, we apply a set of quantum operations that are independent of the given test state. We first apply a Hadamard gate H to the first register, r_1 and $H^{\otimes m}$ to the r_4 register, after which the state $|R\rangle$ transforms to $|R'\rangle = H \otimes 1_{r_2} \otimes 1_{r_3} \otimes H^{\otimes m} |R\rangle$:

$$|R'\rangle = \frac{1}{\sqrt{2M}} \sum_{i=1}^{M} (|0\rangle + |1\rangle) |\psi\rangle |0\rangle^{\otimes n} |i\rangle , \tag{3.1}$$

where $|i\rangle$ is the m-qubit basis state in the computational basis. Next we apply a quantum oracle $\mathcal{W}$ of the form

$$\mathcal{W}|0\rangle|i\rangle = |\phi_i\rangle|i\rangle. \tag{3.2}$$

on the registers r_3 and r_4, where $|\phi_i\rangle$ is the train state indexed by i. Applying this oracle to the coherent superposition in $|R'\rangle$, we obtain

$$|R''\rangle = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} |0\rangle\,|\psi\rangle\,|\phi_i\rangle\,|i\rangle\,. \tag{3.3}$$

We now implement a control swap C_S (2.6) with r_1 as the control qubit and r_2 and r_3 as the target registers. The total state of the system reads

$$|R'''\rangle = \frac{1}{\sqrt{2M}} \sum_{i=1}^{M} (|0\rangle\,|\psi\rangle\,|\phi_i\rangle + |1\rangle\,|\phi_i\rangle\,|\psi\rangle)\,|i\rangle\,. \tag{3.4}$$

This is followed by another Hadamard operation on the r_1 register. After all these we get the final state $|\bar{R}\rangle$ given by

$$|\bar{R}\rangle = \frac{1}{2\sqrt{M}} \sum_{i=1}^{M} \big(\,|0\rangle\,[|\psi\rangle\,|\phi_i\rangle + |\phi_i\rangle\,|\psi\rangle] + |1\rangle\,[|\psi\rangle\,|\phi_i\rangle - |\phi_i\rangle\,|\psi\rangle]\big)\,|i\rangle\,. \tag{3.5}$$

Measurements

We now measure the registers r_1 and r_4. Let $X \in \{0, 1\}$ be the random variable that indicates the state of the r_1 after measurement and Y be the random variable that denotes the resultant index after the measurement of r_4 register, which takes values from $\{0, ..., 2^m - 1\}$.

After measurement, the joint probability distribution $\mathrm{p}(Y = i, X = a)$ (where $a \in \{0, 1\}$) is as follows:

$$p_a(i) \equiv \mathrm{P}(Y = i, X = a) = \frac{1 + (-1)^a F_i}{2M}. \tag{3.6}$$

We may sum over this distribution to verify that we are still handling probabilities

$$\sum_{i=0}^{M} \sum_{a\in\{0,1\}} \mathrm{P}(Y=i, X=a) = \sum_{i=0}^{M-1} \frac{1+F_i}{2M} + \frac{1-F_i}{2M} = 1. \tag{3.7}$$

Indeed, the probabilities do add up to unity. We now construct the quantity $q(i)$ which we call the *contrast* as the difference between the probabilities $p_0(i)$ and $p_1(i)$ for the i-th outcome, i.e.,

$$\begin{aligned} q(i) &= p_0(i) - p_1(i) \\ &= \frac{1+F_i}{2M} - \frac{1-F_i}{2M} = \frac{F_i}{M}. \end{aligned} \tag{3.8}$$

The quantity $q(i)$ is simply the desired fidelity scaled by a factor of $1/M$ and is the quantity of interest in QKNN algorithm. However, this can not be estimated by performing measurement only once. We need to initialize the system in the state $|R\rangle$ and transform it into the state $|\bar{R}\rangle$ and perform the measurement for a sufficiently large number of times. In each run of the algorithm, we acquire a click in the register r_1 and a click in the register r_4.

Let $\{c_{a,i}\}$ be the number of times (a, i) occur in a T-trial (T number of trials). $\{c_{a,i}\}$ follows a multinomial distribution ($p(Y = a, X = 1), T$). That is, the *experiment* has T number of trials with each trial having $2M$ possible outcomes with probabilities $p_a(i)$ with $a \in 0, 1$ and $i \in 0, ..., M-1$.

We define our estimates $\hat{p}_0(i)$ and $\hat{p}_1(i)$ for $p_0(i)$ and $p_1(i)$ as follows

$$\hat{p}_0(i) = \frac{c_{0,i}}{T} \tag{3.9}$$

$$\hat{p}_1(i) = \frac{c_{1,i}}{T}. \tag{3.10}$$

Our estimate for the contrast $q(i)$ is simply

$$\hat{q}(i) = \hat{p}_0(i) - \hat{p}_1(i) \tag{3.11}$$

As we know larger values of fidelity yields larger estimate of contrast $\hat{q}(i)$; hence, running the QKNN algorithm a sufficient number of times, we can find the k states

which are closest to $|\psi\rangle$, i.e., the k number of indices having highest $q(i)$. We assign $|\psi\rangle$ a label after conducting the majority voting. In the next section, we discuss the costs and benefits of the algorithm, including how the standard error scales with the number of maeasurement shots (trials).

3.2 Cost and benefits

The QKNN algorithm offers two main advantages over its classical counterpart. Firstly, it offers the capability to classify unknown states. This is advantageous when we deal with quantum data as we get to bypass the expensive process of quantum state tomography. Any classical kNN method will require the complete description of the quantum state.

The second advantage is obtained through the inherent natures of quantum physics. In classical kNN methods, one requires to compute the distance of the test state with every train state, even far off states, to obtain the k nearest neighbours. In our QKNN algorithm, through quantum parallelism and the probabilistic nature of quantum measurement, only those train states which have high Fidelity with the train states will have high probability of getting detected upon measurement. Therefore, in a limited number of trails only the states which are closer to the train state will appear in the measurement hence fewer resources are spent on them. Furthermore, the measurement results yield the neighbours of the test states with high probability, no sorting is required to determine the neighbours of the test state.

Moreover, in classical kNN, for classifying an N dimensional vector by comparing it with M train states, one requires to have $\mathcal{O}(MN)$ multiplication operations. This also requires $\mathcal{O}(MN)$ space complexity. In our kNN, we require $\log N$ number of Fredkin gates to compute the circuit. Since each Fredkin gate can be realized using seven two-qubit gates as shown in Fig 2.2.2, we require a total of $7 \log N$ two-qubit gates and only two Hadamard gates, which gives QKNN a gate complexity of $\mathcal{O}(\log N)$. We also require $2 \log(N)$+$\log(M) + 1$ qubits, making the space complexity $\mathcal{O}(\log(MN))$.

Let us now look at how the standard error scales with number of trials (measurement shots). Recall that the standard error of $\hat{q}(i)$ is simply the square root of the variance of $\hat{q}(i)$.

$$\begin{aligned}\text{Var}(\hat{q}(i)) &= \text{Var}(\hat{p}_0(i) - \hat{p}_1(i)) \\ &= \text{Var}(\hat{p}_0(i)) + \text{Var}(\hat{p}_1(i)) - 2\,\text{Cov}(\hat{p}_0(i), \hat{p}_1(i)).\end{aligned} \tag{3.12}$$

Using the identity $\text{Var}(X_i) = Tp_i(1-p_i)$ for a multinomial distribution (where $p_i \text{P}(X = i)$ is the probability that a random variable X takes the value i), the variance terms are:

$$\text{Var}(\hat{p}_0(i)) = \frac{1}{T^2}\text{Var}(c_{0,i}) = \frac{1}{T^2}T\frac{1+F_i}{2M}\left(1 - \frac{1+F_i}{2M}\right), \tag{3.13}$$

$$\text{Var}(\hat{p}_1(i)) = \frac{1}{T^2}\text{Var}(c_{1,i}) = \frac{1}{T^2}T\frac{1-F_i}{2M}\left(1 - \frac{1-F_i}{2M}\right), \tag{3.14}$$

and the covariance term is:

$$\begin{aligned}\text{Cov}(\hat{p}_0(i), \hat{p}_1(i)) &= \frac{1}{T^2}\text{Cov}(c_{0,i}, c_{1,i}) = \frac{1}{T^2}(-T)\left(\frac{1+F_i}{2M}\right)\left(\frac{1-F_i}{2M}\right) \\ &= \frac{F_i^2 - 1}{4TM^2},\end{aligned} \tag{3.15}$$

where we've used the identities

$$\begin{aligned}\text{Cov}(aX, bY) &= ab\,\text{Cov}(X, Y), \\ \text{Cov}(X_i, X_j) &= Tp_ip_j.\end{aligned} \tag{3.16}$$

where T is the number of trials and p_i, p_j are the probabilities $P(X = i), P(X = j)$ respectively.

Substituting it all in 3.12, we obtain the variance to be

$$\text{Var}(\hat{q}(i)) = \frac{1}{M^2T}(M - F_i^2), \tag{3.17}$$

taking the square root, we obtain the standard error σ to be

$$\sigma(\hat{q}(i)) = \frac{1}{M\sqrt{T}}\sqrt{M - F_i^2}. \tag{3.18}$$

This tells us that to approximate the distribution to a standard error of σ, we require $T \sim 1/\sigma^2$ order of measurements. Since $F_i < 1 << M$, we can safely ignore that term

to bring the standard error to the form

$$\sigma = \sigma(\hat{q}(i)) = \frac{1}{\sqrt{MT}}. \tag{3.19}$$

We drop the state dependence from σ since we've ignored the fidelity F_i dependence. We may rearrange the equation to get a relation between number of required measurements T and number of train states M.

$$T = \frac{1}{M\sigma^2} \tag{3.20}$$

To put in some concrete numbers, lets say our tolerance of error in the estimate of F_i is 10%. Then an upperbound for 10% error tolerance in F_i/M is $0.1/M$. Substituting this in the above equation, we obtain that we need $T = \frac{1}{M(0.01/M^2)} = 100\ M$ order of measurements.

In the next subsection, we apply the QKNN algorithm on the entanglement classification problem and compare the results with the ones we achieve with classical kNN algorithm.

Chapter 4

Entanglement classification using classical and quantum kNN

We conduct entanglement classification in two ways - first by using classical kNN with the distance function $D(\psi, \phi) = 1 - F(\psi, \phi)$ defined in equation 2.4, and second by complete simulation of the quantum algorithm. We denote the first method by label 'classical' and second method by the label 'quantum' under the algorithm type in tables 4.1 and 4.2. In all classifications, we use $k = 3$ nearest neighbours for classification purposes.

4.1 Entanglement classification using classical kNN

We first classify two and three-qubit quantum states based on their entanglement using classical kNN. Here, we consider three cases: a) separable vs. entangled states (in two qubits), b) separable vs. maximally entangled states (in two qubits), and c) three qubit classification. In (c), we have five classes. In all three cases, we have 10^5 train states in each class generated randomly. Classical kNN allows us to show how the principle of kNN can be used to solve the problem. The results are tabulated in table 4.1.

From table 4.1, we can see that the classical kNN works perfectly for entanglement classification in two-qubit case. In the case of three-qubit case the accuracy we achieve is little over 82%. This accuracy can be increased by increasing the number of k and by increasing the size of the set of train states.

No. of Qubits	No. of classes	Entanglement classes	Accuracy	Class size	Algorithm type
2	2	Separable, Entangled	100%	10^5	Classical
2	2	Separable, Maximally entangled	100%	10^5	Classical
3	5	1-2-3, 12-3, 1-23, 13-2, 123	82.2%	10^5	Classical

Table 4.1: Entanglement classification using classical kNN classifier. Cardinality of the set of train states is simply $M =$ (No. of classes)$\times$(Class size).

4.2 Entanglement classification using QKNN

Next we simulate the QKNN algorithm and classify two-qubit states in two scenarios. First, when the classification is between separable states and maximally entangled states and next when the two classes are separable states and general entangled states. In the simulation of QKNN algorithm, we have $n = 2$ number of qubits and the cardinality M of the set of train states to be 32 (16 train states in each of the two classes) and hence $m = \log M = 5$. So we simulate a quantum circuit of $1 + 2 + 2 + 5 = 9$ qubits. Each simulation have been performed for 10^4 measurements and each result (accuracy) has been averaged over ten different simulations with test and train states generated randomly. To compare like with like, we also run classical kNN on the same dataset (which is of the same cardinality) and display results in table 4.2. We see that QKNN achieves accuracy quite close to classical kNN.

It is clear from table 4.2 that the QKNN and classical kNN algorithm performs almost equally well (given the limitations). At first, it seems like we need large resources to perform simple classification in QKNN as compare to classical kNN. In the case presented we need 9-qubit register with 14 two-qubit operations, thousands of times in order to perform two-qubit entanglement classification, which can be done rather easily using classical computers. However, the resource requirement in QKNN increases linearly with the number of qubit as opposed to the classical kNN algorithm where the operations grow exponentially. Furthermore, we require no knowledge of the test state prior to performing the algorithm, however, we will need an arbitrary copies of the test state for arbitrarily high accuracy of classification.

No. of Qubits	No. of classes	Entanglement classes	Accuracy	Class size	Algorithm type
2	2	Separable, Maximally entangled	96.67%	16	Classical
2	2	Separable, Maximally entangled	95.67%	16	Quantum 10^4 shots
2	2	Separable, Entangled	80.1%	16	Classical
2	2	Separable, Entangled	80.67%	16	Quantum 10^4 shots

Table 4.2: Entanglement classification using quantum kNN classifier compared with classical kNN classifier. Here, shots indicate the number of measurement shots performed over each quantum circuit simulation. Cardinality of the set of train states is simply $M =$ (No. of classes)$\times$(Class size).

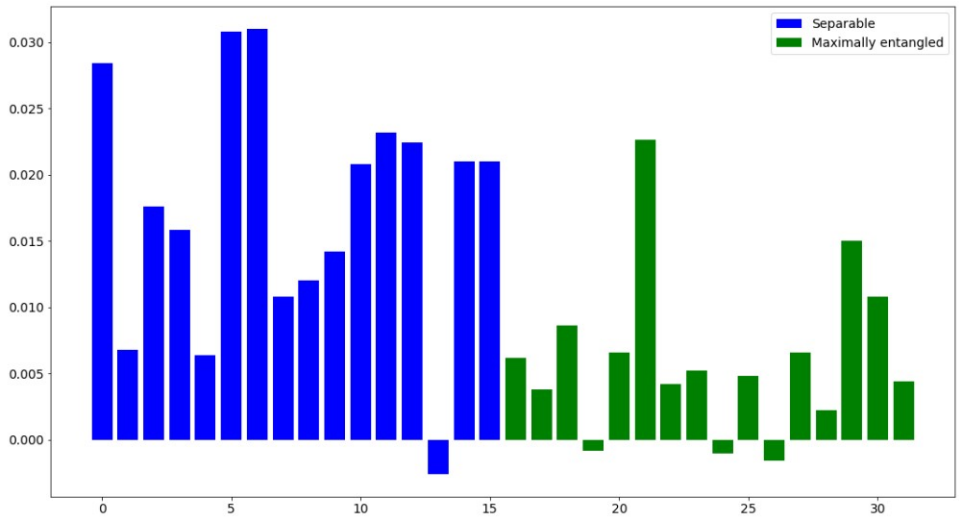

Figure 4.2.1: $\hat{q}(i)$ vs i for a separable test state $|\psi\rangle$. Of the 32 test states, the first 16 are separable and the rest are maximally entangled. The negative values for certain states is due to error in estimation.

Chapter 5

Conclusion

To conclude, we have presented a novel QKNN algorithm, which is a quantum analog of classical kNN algorithm. Our algorithm uses the Swap test and wavefunction collapse along with a single oracle to achieve high speedup as compare to its classical counterpart. The number of gates required to implement QKNN is linear in the n where $N = 2^n$ is the dimension of the test state vector. In terms of the number of additional qubit, QKNN requires $2n + m + 1$ number of qubits where $2^m = M$ is the cardinality of the set of train states.

One of the most important advantages of QKNN is that it is capable to handle unknown quantum test states. This feature is entirely missing in classical kNN where one needs to have complete knowledge of the test state. Furthermore, unlike classical kNN, QKNN does not need to calculate the distance between the test state and all the train states. Since the quantum measurements result in stochastic outcomes, only the most likely outcomes will be observed upon measurements yielding the closest neighbours. As an example, we simulate QKNN on classical computer for the problem of classifying multipartite entangled states. We show that QKNN yields as high accuracy in classifying the states as classical kNN algorithm with the additional advantage of not requiring the information about the test state. The application of QKNN is endless, and it is straight forward to implement on any platform.

There are a number of ways to extend the results from our work. An important question is the comparison with the swap test. Here we must look at how better is our QKNN algorithm at classification as compared to classification done through first Swap test-ing all the train states with the test state to estimate the Fidelity, and then rank accordingly. There is a qubit overhead with our algorithm (an overhead of $m = \log(M)$

but does it endow us with any advantage? Finally, another avenue to go about is whether we can construct a operation such that its (repeated) application can amplify the amplitude of the nearer states while reducing the amplitude of the far away states. If such an operation can be found, then it shows a certain advantage over individual swap test.

Bibliography

[Aaronson 07] Scott Aaronson. *The learnability of quantum states*. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 463, no. 2088, page 3089–3114, Sep 2007.

[Arunachalam 17] Srinivasan Arunachalam & Ronald de Wolf. *A Survey of Quantum Learning Theory*, 2017.

[Arute 19] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell*et al.* *Quantum supremacy using a programmable superconducting processor*. Nature, vol. 574, no. 7779, pages 505–510, 2019.

[Benuwa 16] Ben Bright Benuwa, Yong Zhao Zhan, Benjamin Ghansah, Dickson Keddy Wornyo & Frank Banaseka Kataka. *A review of deep machine learning*. In International Journal of Engineering Research in Africa, volume 24, pages 124–136. Trans Tech Publ, 2016.

[Biamonte 17] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe & Seth Lloyd. *Quantum machine learning*. Nature, vol. 549, no. 7671, pages 195–202, 2017.

[Bijalwan 14] Vishwanath Bijalwan, Vinay Kumar, Pinki Kumari & Jordan Pascual. *KNN based machine learning approach for text and document mining*. International Journal of Database Theory and Application, vol. 7, no. 1, pages 61–70, 2014.

[Buhrman 01] Harry Buhrman, Richard Cleve, John Watrous & Ronald de Wolf. *Quantum Fingerprinting*. Phys. Rev. Lett., vol. 87, page 167902, Sep 2001.

[Carrasquilla 17] Juan Carrasquilla & Roger G Melko. *Machine learning phases of matter*. Nature Physics, vol. 13, no. 5, pages 431–434, 2017.

[Cover 67] T. Cover & P. Hart. *Nearest neighbor pattern classification*. IEEE Transactions on Information Theory, vol. 13, no. 1, pages 21–27, January 1967.

[Dey 16] Ayon Dey. *Machine learning algorithms: a review*. International Journal of Computer Science and Information Technologies, vol. 7, no. 3, pages 1174–1179, 2016.

[Dür 00] W. Dür, G. Vidal & J. I. Cirac. *Three qubits can be entangled in two inequivalent ways*. Physical Review A, vol. 62, no. 6, Nov 2000.

[Häffner 08] Hartmut Häffner, Christian F Roos & Rainer Blatt. *Quantum computing with trapped ions*. Physics reports, vol. 469, no. 4, pages 155–203, 2008.

[Harrow 09] Aram W Harrow, Avinatan Hassidim & Seth Lloyd. *Quantum algorithm for linear systems of equations*. Physical review letters, vol. 103, no. 15, page 150502, 2009.

[He 07] Q Peter He & Jin Wang. *Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes*. IEEE transactions on semiconductor manufacturing, vol. 20, no. 4, pages 345–354, 2007.

[Horodecki 09] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki & Karol Horodecki. *Quantum entanglement*. Reviews of Modern Physics, vol. 81, no. 2, page 865–942, Jun 2009.

[Imandoust 13] Sadegh Bafandeh Imandoust & Mohammad Bolandraftar. *Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background*. International Journal of Engineering Research and Applications, vol. 3, no. 5, pages 605–610, 2013.

[Jozsa 94] Richard Jozsa. *Fidelity for Mixed Quantum States*. Journal of Modern Optics, vol. 41, no. 12, pages 2315–2323, 1994.

[Kok 07] Pieter Kok, William J Munro, Kae Nemoto, Timothy C Ralph, Jonathan P Dowling & Gerard J Milburn. *Linear optical quantum computing with photonic qubits*. Reviews of Modern Physics, vol. 79, no. 1, page 135, 2007.

[Ladd 10] Thaddeus D Ladd, Fedor Jelezko, Raymond Laflamme, Yasunobu Nakamura, Christopher Monroe & Jeremy Lloyd O'Brien. *Quantum computers*. Nature, vol. 464, no. 7285, pages 45–53, 2010.

[Le 13] Quoc V Le. *Building high-level features using large scale unsupervised learning*. In 2013 IEEE international conference on acoustics, speech and signal processing, pages 8595–8598. IEEE, 2013.

[LeCun 15] Yann LeCun, Yoshua Bengio & Geoffrey Hinton. *Deep learning*. nature, vol. 521, no. 7553, pages 436–444, 2015.

[Li 01] Shu-Shen Li, Gui-Lu Long, Feng-Shan Bai, Song-Lin Feng & Hou-Zhi Zheng. *Quantum computing*. Proceedings of the National Academy of Sciences, vol. 98, no. 21, pages 11847–11848, 2001.

[Liao 02] Yihua Liao & V Rao Vemuri. *Use of k-nearest neighbor classifier for intrusion detection*. Computers & security, vol. 21, no. 5, pages 439–448, 2002.

[Lloyd 13] Seth Lloyd, Masoud Mohseni & Patrick Rebentrost. *Quantum algorithms for supervised and unsupervised machine learning*. arXiv preprint arXiv:1307.0411, 2013.

[Lloyd 14] Seth Lloyd, Masoud Mohseni & Patrick Rebentrost. *Quantum principal component analysis*. Nature Physics, vol. 10, no. 9, pages 631–633, 2014.

[Lu 17] Sirui Lu, Shilin Huang, Keren Li, Jun Li, Jianxin Chen, Dawei lu, Zhengfeng Ji, Yi Shen, Duanlu Zhou & Bei Zeng. *Separability-Entanglement Classifier via Machine Learning*. Physical Review A, vol. 98, 05 2017.

[Mani 03] Inderjeet Mani & I Zhang. *kNN approach to unbalanced data distributions: a case study involving information extraction*. In Proceedings of workshop on learning from imbalanced datasets, volume 126, 2003.

[Nigsch 06] Florian Nigsch, Andreas Bender, Bernd van Buuren, Jos Tissen, Eduard Nigsch & John BO Mitchell. *Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization*. Journal of chemical information and modeling, vol. 46, no. 6, pages 2412–2422, 2006.

[O'brien 07] Jeremy L O'brien. *Optical quantum computing*. Science, vol. 318, no. 5856, pages 1567–1570, 2007.

[O'Malley 16] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik & J. M. Martinis. *Scalable Quantum Simulation of Molecular Energies*. Phys. Rev. X, vol. 6, page 031007, Jul 2016.

[Pirandola 19] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. Pereira, M. Razavi, J. S. Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi & P. Wallden. *Advances in Quantum Cryptography*, 2019.

[Rebentrost 14] Patrick Rebentrost, Masoud Mohseni & Seth Lloyd. *Quantum Support Vector Machine for Big Data Classification*. Physical Review Letters, vol. 113, no. 13, Sep 2014.

[Ruan 17] Yue Ruan, Xiling Xue, Heng Liu, Jianing Tan & Xi Li. *Quantum Algorithm for K-Nearest Neighbors Classification Based on the Metric of Hamming Distance*. International Journal of Theoretical Physics, vol. 56, pages 3496–3507, 2017.

[Samworth 12] Richard J Samworth*et al. Optimal weighted nearest neighbour classifiers*. The Annals of Statistics, vol. 40, no. 5, pages 2733–2763, 2012.

[Santoro 06] Giuseppe E Santoro & Erio Tosatti. *Optimization using quantum mechanics: quantum annealing through adiabatic evolution*. Journal of Physics A: Mathematical and General, vol. 39, no. 36, pages R393–R431, aug 2006.

[Schuld 14] Maria Schuld, Ilya Sinayskiy & Francesco Petruccione. *An introduction to quantum machine learning*. Contemporary Physics, vol. 56, no. 2, page 172–185, Oct 2014.

[Shor 99] Peter W Shor. *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*. SIAM review, vol. 41, no. 2, pages 303–332, 1999.

[Simeone 18] Osvaldo Simeone. *A very brief introduction to machine learning with applications to communication systems*. IEEE Transactions on Cognitive Communications and Networking, vol. 4, no. 4, pages 648–664, 2018.

[Smolin 96] John A. Smolin & David P. DiVincenzo. *Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate*. Phys. Rev. A, vol. 53, pages 2855–2856, Apr 1996.

[Steane 98] Andrew Steane. *Quantum computing*. Reports on Progress in Physics, vol. 61, no. 2, pages 117–173, feb 1998.

[Voulodimos 18] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis & Eftychios Protopapadakis. *Deep learning for computer vision: A brief review*. Computational intelligence and neuroscience, vol. 2018, 2018.

[Wang 17] Bingjie Wang. *Learning to Detect Entanglement*, 2017.

[Wiebe 15] Nathan Wiebe, Ashish Kapoor & Krysta M. Svore. *Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsu-*

pervised Learning. Quantum Info. Comput., vol. 15, no. 3–4, page 316–356, March 2015.

[Wittek 14] Peter Wittek. Quantum machine learning: what quantum computing means to data mining. Academic Press, 2014.

[Zhang 17] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu & Debo Cheng. *Learning k for KNN Classification*. ACM Trans. Intell. Syst. Technol., vol. 8, no. 3, January 2017.