The University of Hong Kong

PHYS3999 Directed Studies in Physics

Title: Quantum Computing and its Physical Realization

Author:
Weihao Cao

Supervisor:
Prof. Zidan Wang

Author:Weihao Cao
UID:3035234727
Date:November 25, 2017

Contents

1 Abstract

Quantum computing has recently been a heated topic and research frontier, accompanied by the decreasing dimension of classical computing circuits where quantum effects will take effect. On the other hand, many featured properties of quantum computing has made it a possible candidate for the next generation of computing device to keep growth of computing power in accordance with Moore's Law. In specific, some quantum algorithms have shown exponential speedup compared with classical algorithms.

In this report, first I will review the formalism of quantum computing, and provide some mathematical proof to theorems or give some concrete examples. Next one of the most dramatic algorithm – Quantum Fourier Transform(QFT). Finally, an experimental confirmation using QFT and machine learning techniques to classify digits will be analyzed and some discussions will be proposed.

2 Formalism

First the elements of quantum computation will be introduced. Similar to the composition of classical computation, quantum computation may be decomposed into three parts: qubits, which are the "memory" and computational units of quantum computing; quantum gates, which implement unitary transformation on the quantum states; measurement, where the desired information will be extracted. What's more, a quantum computation generally involves three steps: encoding the input state, conducting unitary transformation on the previous state and measuring the output state.

2.1 Qubits

Analogous to a classical bit used in classical computers, a qubit is the basic two-level computational unit obeying quantum mechanics. A qubit is represented in two computational basis as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{1}$$

Here it can be readily seen that rather than only take values 0 and 1 as those in classical computers, the evolution coefficient, α and β, takes continuous values.

Unfortunately, measurement theory tells us that only one bit of information may be acquired when the state is measured, that is, either state $|0\rangle$ or state $|1\rangle$ may be detected in experiments.

More generally, an n-qubit state may be represented as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle \tag{2}$$

constrained by the normalization condition

$$\sum_{i=0}^{2^n-1} |c_i|^2 = 1 \tag{3}$$

It is worth mentioning here that with n qubits, the formed computational basis has 2^n states, and thus the potential storage capabilities have exponential increase over classical algorithms. However, the output of the measurement process is probabilistic and quantum mechanics tells us that the probabilities of achieving each possible output are proportional to the square of amplitudes. Therefore, in a quantum algorithm, the whole process must be repeated several times to obtain desired degree of accuracy of the solution.

2.2 Quantum Gates

First the concept of quantum gates should be introduced.

Quantum gates represent the unitary transformation between quantum states. Unitary comes from the fact that the norm of the states must be preserved. In single-qubit case, gates are described by 2×2 unitary matrices. One of the most important matrices is the Hadamard gate(denoted as H), and its matrix form for single qubit is

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{4}$$

One useful property of the Hadamard gate is that the superposition of all states in the computational basis can be constructed by applying Hadamard gate on the state $|000..\rangle$ with n qubits: (it is worth mentioning that H_n operates on n qubits,

and thus has dimension $2^n \times 2^n$.)

$$H_n |\psi_n\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} H_{n-1} & H_{n-1} \\ H_{n-1} & -H_{n-1} \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{0_{n-1}} \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \tag{5}$$

which will be useful in many algorithms.

Another important single qubit gate is the phase gate (S):

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \tag{6}$$

and it is apparent that the phase gate S transforms a generic state $|\psi\rangle = a|0\rangle + b|1\rangle$ into the state $|\psi\rangle = a|0\rangle + ib|1\rangle$.

Under the Bloch sphere representation of qubits, the effect of the phase gate may be visualized as the rotation of the state vector by an angle $\pi/2$ along the z axis. Furthermore, we may generalize the phase operator into rotation operators about the three axes:

$$R_x(\theta) = e^{-i\theta X/2} = \ R_z(\theta) = e^{-i\theta Z/2} = \ \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}, \tag{7}$$

and similar for rotation operators along the other two directions. More generally, the following property which I am going to prove guarantees the completeness of single qubit gates.

Claim. *Any single qubit gates may be represented in the form $U = e^{i\alpha} R_{\vec{n}}(\theta)$ for some real numbers α and θ, 3-D unit vector $\vec{n}$, and the rotation operator is $R_{\vec{n}}(\theta) = \cos\frac{\theta}{2} I - i \sin\frac{\theta}{2} \cdot (n_x X + n_y Y + n_z Z)$.*

Proof. To decompose any single qubit operator U, first we calculate the overall phase α. Consider

$$|\det U|^2 = \det U \cdot (\det U)^\dagger = \det U \det U^\dagger = detUU^\dagger = 1, \tag{8}$$

thus $\det U$ can be regarded a phase. We choose $e^{2i\alpha} = \det U$, then $U = e^{i\alpha} V$, and $\det V = 1$. Next we decompose the matrix V using the given Pauli matrices:

$$V = v_0 I + v_x X + v_y Y + v_z Z. \tag{9}$$

If we denote the matrix V as$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$, then by comparing the coefficients on both sides, we may get, $v_0 = (a+d)/2, v_x = (b+c)/2, v_y = (c-b)/2i, v_z = (a-d)/2$.

Imposing the condition that V is unitary,

$$\begin{aligned} V^\dagger V =&(v_0 I + v_x X + v_y Y + v_z Z)^\dagger (v_0 I + v_x X + v_y Y + v_z Z) \\ =&|v_0|^2 + |v_x|^2 + |v_y|^2 + |v_z|^2) I + (v_0^* v_x + v_x^* v_0 + i v_y^* v_z - i v_z^* v_y) X + \\ &(v_0^* v_y + v_y^* v_0 + i v_z^* v_x - i v_x^* v_z)) Y + (v_0^* v_z + v_z^* v_0 + i v_x^* v_y - i v_y^* v_x) Z \\ =&I. \end{aligned} \tag{10}$$

To satisfy the up-left element, we should have $|v_0|^2 + |v_x|^2 + |v_y|^2 + |v_z|^2 = 1$. Define $\cos(\theta/2) = |v_0|$, then $|v_x|^2 + |v_y|^2 + |v_z|^2 = \sin^2(\theta/2)$. The symmetric first two terms of XYZ and the asymmetric last two terms stimulate us to reformulate them using cross product:

$$2Re(v_0)\ \vec{Re(v)} + 2Im(v_0)\ \vec{Im(v)} + i\vec{v^*} \times \vec{v} = \vec{0}. \tag{11}$$

Borrow the the rotation vector notation $\vec{n}$, then we have,

$$(n_x, n_y, n_z) = \frac{i}{\sin(\theta/2)}(v_x, v_y, v_z). \tag{12}$$

First $|n_x|^2 + |n_y|^2 + |n_z|^2 = 1$, thus the normalization condition is achieved; v_0 is real, v is purely imaginary, and $\vec{v^*}$ is parallel to $\vec{v}$, thus the Equation 11 is satisfied. Finally, consider the determinant of V:

$$V = \begin{bmatrix} \cos\frac{\theta}{2} - i\sin\frac{\theta}{2} n_z & -i\sin\frac{\theta}{2} n_x - \sin\frac{\theta}{2} n_y \\ -i\sin\frac{\theta}{2} n_x + \sin\frac{\theta}{2} n_y & \cos\frac{\theta}{2} + i\sin\frac{\theta}{2} n_z \end{bmatrix}, \tag{13}$$

thus $\det(V) = \cos^2\frac{\theta}{2} + \sin^2\frac{\theta}{2}(n_x^2 + n_y^2 + n_z^2) = 1$, and thus the requirement that $\det(V) = 1$ is fulfilled. Thus all conditions are satisfied.

In summary, for any unitary 2×2 matrix, we can find an angle θ, phase α and a unit vector $\vec{n}$ so that the matrix can be decomposed as

$$U = e^{i\alpha} R_{\vec{n}}(\theta) = e^{i\alpha}[\cos\frac{\theta}{2} I - i\sin\frac{\theta}{2} \cdot (n_x X + n_y Y + n_z Z)]. \tag{14}$$

■

2.3 Multiple Qubit Gates

We claim that entanglement states cannot be generated simply with single qubit gates: assume the system starts with a separable state, $|\psi\rangle = |\psi_{n-1}\rangle \otimes \cdots \otimes |\psi_0\rangle$, and multiple single-qubit gates are implemented: $U = U_{n-1} \otimes \cdots \otimes U_0$, then the final state can still be represented as $|\psi'\rangle = \left|\psi'_{n-1}\right\rangle \otimes \cdots \otimes |\psi'_0\rangle$, and the qubits are still disentangled. Thus we need to introduce multiple-qubit operations. Here we restrict to two qubit gates, and higher dimension gates are easy to generalize.

First I will mention a trivial gate, the swap gate, which may switch the state function of two qubits, and the operator can be represented as

(15)

The gate swaps the state of the two qubits, and the matrix representation is

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{16}$$

One key two-qubit gate is the controlled-NOT gate, which conducts the NOT operation on the second qubit conditioning on the truth of the first qubit. The representation of $CNOT$ is

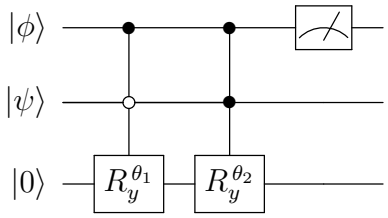
(17)

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{18}$$

Next I will elaborate on the controlled operations by referring to a specific circuit using Controlled-Phase Shift operators to prepare a general superposition state, which is a core experimental step in later parts of the report.

2.4 An Example Circuit

Assume we want to encode the superposition of two states $\Phi_1 = \begin{bmatrix} 0.987 \\ 0.159 \end{bmatrix}$, and $\Phi_2 = \begin{bmatrix} 0.354 \\ 0.935 \end{bmatrix}$ into the state $|\beta_1\rangle |\Phi_1\rangle + |\beta_2\rangle |\Phi_2\rangle$, which is part of the preparation process in the experiment[1], and I will give the mathematical deduction of the circuit operations. e Consider the circuit below, where the first qubit starts from the state $|\phi\rangle = \alpha_1 |0\rangle + \alpha_2 |1\rangle$, and the second qubit starts from the state $|\psi\rangle = \beta_1 |0\rangle + \beta_2 |1\rangle$:



(19)

The circuit operates as follows:

$$\begin{aligned} |\phi\rangle |\psi\rangle |0\rangle &= \big[\alpha_1 |0\rangle + \alpha_2 |1\rangle\big] \otimes \big[\beta_1 |0\rangle + \beta_2 |1\rangle\big] \otimes |0\rangle \\ &\to \alpha_1 |000\rangle + \alpha_2 |1\rangle \otimes \Bigg[\beta_1 |0\rangle \otimes \big[\cos\frac{\theta_1}{2} |0\rangle + \sin\frac{\theta_1}{2} |1\rangle\big] + \\ &\qquad \beta_2 |1\rangle \otimes \big[\cos\frac{\theta_2}{2} |0\rangle + \sin\frac{\theta_2}{2} |1\rangle\big]\Bigg] \end{aligned} \tag{20}$$

Conditioning on the measurement result of the first qubit is $|1\rangle$, and the first rotation operator contains the rotation angle θ_1 satisfies $\tan\frac{\theta_1}{2} = \frac{0.159}{0.987}$, and the second θ_2 satisfies $\tan\frac{\theta_2}{2} = \frac{0.935}{0.354}$, then the state of the qubits reduces to

$$\beta_1 |0\rangle \begin{bmatrix} \cos\frac{\theta_1}{2} \\ \sin\frac{\theta_1}{2} \end{bmatrix} + \beta_2 |1\rangle \begin{bmatrix} \cos\frac{\theta_2}{2} \\ \sin\frac{\theta_2}{2} \end{bmatrix}, \tag{21}$$

as desired.

3 Quantum Algorithms

In the past few decades, many quantum algorithms solving specific fields of questions have been proposed, and some of them achieves impressive speedup over their

classical counterparts. In specific, three most prominent algorithms are quantum search, quantum Fourier transform and Quantum simulation of dynamic systems. Here I will mainly focus on quantum Fourier transform algorithm, which is a crucial step in later discussions.

3.1 Quantum Fourier Transform

3.1.1 Quantum Fourier Transform Formalism

Analogous to the classical discrete Fourier transform, quantum Fourier transform samples the orthogonal basis states with varying frequencies. With an orthogonal basis $|0\rangle, ..., |N-1\rangle$, the quantum Fourier transform acting on the quantum state $|j\rangle$ outputs the state

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle \tag{22}$$

For a general initial state $\sum_{j=0}^{N-1} c_j |j\rangle$, the quantum Fourier transform performs as

$$\sum_{j=0}^{N-1} c_j |j\rangle \to \sum_{k=0}^{N-1} \tilde{c_k} |k\rangle, \quad where \quad \tilde{c_k} = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi ijk/N} c_j. \tag{23}$$

Before moving on, I will first prove that quantum Fourier transform is a unitary operation, which is a requirement for quantum gates.

Claim. *Quantum Fourier transform is a unitary transformation.*

Proof. For clarity reasons, first I will rewrite the transform in outer product forms:

$$F = \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \frac{e^{2\pi ijk/N}}{\sqrt{N}} |k\rangle \langle j|. \tag{24}$$

Consider

$$\begin{aligned} F^\dagger F &= \Big(\sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \frac{e^{-2\pi ijk/N}}{\sqrt{N}} |j\rangle \langle k| \Big) \Big(\sum_{j'=0}^{N-1} \sum_{k'=0}^{N-1} \frac{e^{2\pi ijk/N}}{\sqrt{N}} |k'\rangle \langle j'| \Big) \\ &= \frac{1}{N} \sum_{j,k,j',k'} e^{2\pi i(j'k'-jk)/N} |j\rangle \langle j'| \delta_{kk'} \\ &= \frac{1}{N} \sum_{j,k,j'} e^{2\pi ik(j'-j)/N} |j\rangle \langle j'|. \end{aligned} \tag{25}$$

When $j = j'$, $e^{2\pi ik(j'-j)/N} \equiv 1$, thus $\sum_k e^{2\pi ik(j'-j)/N} = N$; If $j \neq j'$, then $\sum_k e^{2\pi ik(j'-j)/N} = (1 - (\frac{2\pi(j'-j)}{N})^n)/(1 - \frac{2\pi(j'-j)}{N}) = 0$, thus

$$\frac{1}{N} \sum_{j,k,j'} e^{2\pi ik(j'-j)/N} |j\rangle \langle j'| = \sum_{j,j'} |j\rangle \langle j'| \delta_{jj'} = \sum j |j\rangle |j\rangle = I. \tag{26}$$

Thus F is a unitary operation. ■

Another more useful representation of the algorithm utilizes the binary representation of quantum states: Assume the computational basis are of dimension $N = 2^n$, then a state $|j\rangle$ may be decomposed as $j = j_1 2^{n-1} + \cdots + j_n 2^0$, and represented as $j = j_1 \dots j_n$.

Using tensor products to dissemble each binary bits into individual qubits, we may get the product representation of quantum Fourier transform:

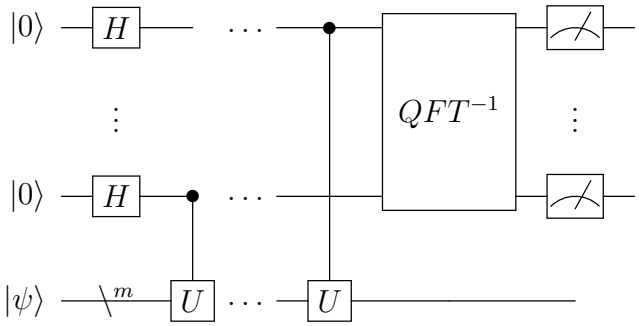
$$\begin{aligned} |j\rangle &\to \frac{1}{2^N} \sum_{k=0}^{2^n-1} e^{2\pi ijk/N} |k\rangle \\ &= \frac{1}{2^N} \bigotimes_{l=1}^{n} [|0\rangle + e^{2\pi ij2^{-l}} |1\rangle]. \end{aligned} \tag{27}$$

It can be easily seen that if the system starts from the state $|j_1 \dots j_n\rangle$, we can use some Hadamard gates and general phase-shift gates to complete the algorithm. A concrete example is integrated into the next section for a more general illustration.

3.1.2 Phase Estimation

The phase estimation algorithm enables us to find the eigenvalue of a unitary operator U. First I will phrase the problem: Let U be a unitary operator acting on m qubits with an eigenvector $|\psi\rangle$ and eigenvalue $e^{2\pi i\theta}$ such that $U|\psi\rangle = e^{2\pi i\theta} |\psi\rangle$ ($0 \leq \theta < 1$), where the value θ is to be determined.

The algorithm contains two registers. The first ensemble contains t qubits, and the initial state is $|000\dots\rangle$. The value t depends on the expected estimation accuracy of θ. The second register contains and is initialized to U.



(28)

Phase estimation procedure contains two steps. First, each qubit in the first register is super-positioned using Hadamard gate. Then controlled-U operations are applied to the second register, with times of increasing powers of two. The output state is

$$\frac{1}{2^N}(|0\rangle + e^{2\pi i 2^{t-1}\theta}|1\rangle)(|0\rangle + e^{2\pi i 2^{t-2}\theta}|1\rangle)\ldots(|0\rangle + e^{2\pi i 2^{0}\theta}|1\rangle). \tag{29}$$

Then the inverse quantum Fourier transform (the reverse procedure of the quantum Fourier transform) is applied to the first register

$$\frac{1}{2^N}\sum_{j=0}^{2^t-1}\sum_{k=0}^{2^t-1} e^{-2\pi i k j/2^t} e^{2\pi i k\theta}|j\rangle . \tag{30}$$

If we measure the state of the first register, then we can obtain θ to n bits with chance of success at least $1-\epsilon$ if the number t satisfies

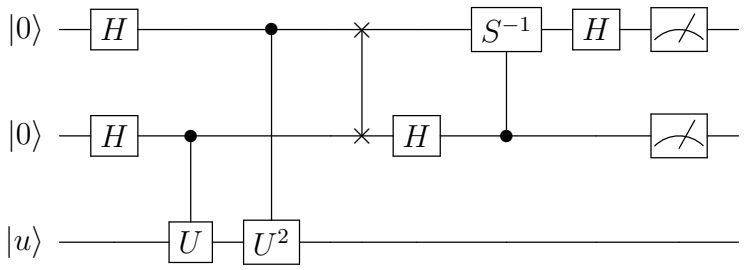
$$t = n + \left\lceil \log(2 + \frac{1}{2\epsilon}) \right\rceil . \tag{31}$$

More generally, if the second register is not initially in an eigenstate of U, say the state of the register is $|\psi\rangle$, then we may expand this state in terms of the eigenstates $|u\rangle$ of U $|\psi\rangle = \sum c_u |u\rangle$. Suppose the eigenstate $|u\rangle$ has eigenvalue $e^{2\pi i\phi_u}$, then measuring the first register will give us with probability $|c_u|^2$ the eigenstate $|u\rangle$'s corresponding phase ψ_u.

3.1.3 A concrete example

Next let's consider a quantum circuit illustrating the phase estimation scheme using part of the circuits from [1]. Below I will give the mathematical deduction

of the quantum states involved:



(32)

First, Hadamard gates and the controlled operations transform inject increasing length of bits of the eigenvalue of the operator U into the first two registers. Then inverse Fourier transform is applied to the first two qubits to filter the phase factor. If then we measure the state of the qubits, we can get an approximation of the phase.

Here the operator U acts on the eigenstate $|u\rangle$ as $U|u\rangle = e^{2\pi i\phi}|u\rangle$, and the term ϕ is the phase to be determined. In binary representations, ϕ may be written as $\phi = 0.\phi_1\phi_2\ldots$.

$$
\begin{aligned}
|00u\rangle &\to \frac{1}{2}[|0\rangle|0\rangle|y\rangle + |0\rangle|1\rangle e^{2\pi i\phi}|y\rangle + |1\rangle|0\rangle e^{2\pi i\times 2\phi}|y\rangle + |1\rangle|1\rangle e^{2\pi i\times 3\phi}|y\rangle] \\
&= \frac{1}{2}\big[\,|0\rangle + e^{2\pi i 0.\phi_1\cdots}\big] \otimes \big[\,|0\rangle + e^{2\pi i 0.\phi_2\cdots}\big] \otimes |y\rangle \quad \text{after } U \text{ gates} \\
&\to \frac{(1+e^{2\pi i 0.\phi_1\cdots})(1+e^{2\pi i 0.\phi_2})}{4}|00\rangle + \frac{(1-e^{2\pi i 0.\phi_1\cdots})(1+e^{2\pi i 0.\phi_2})}{4}|10\rangle\ldots \\
&\quad + \frac{(1-ie^{2\pi i 0.\phi_1\cdots})(1-e^{2\pi i 0.\phi_2})}{4}|01\rangle + \frac{(1+ie^{2\pi i 0.\phi_1\cdots})(1-e^{2\pi i 0.\phi_2})}{4}|11\rangle
\end{aligned}
\tag{33}
$$

In general, none of the four coefficients corresponding to different states is zero, and thus we can only correctly extract the first two bits of the phase ϕ with accuracy

$$
|c|^2 = \left|\frac{(1+e^{\frac{\pi i}{2}0.\phi_3\cdots})(1+e^{\pi i 0.\phi_3\cdots})}{4}\right|^2. \tag{34}
$$

In the special case when $\phi = 0.\phi_1\phi_2 = 0.01$, we can immediately get that the state after the inverse Fourier transform is $|0\rangle|1\rangle|y\rangle$, and thus the measurement yields 0 and 1 individually, as desired.

4 Quantum Computer Realization

Besides building quantum algorithms superior to their classical counterparts, experimentally realizing large-scale quantum computers is another obstacle to harnessing the power of quantum computation. Indeed, the main requirements to realize quantum computation include: [4]

1. Find a well-behaved system with characterized qubits to represent quantum information.
2. Manipulate the state of the qubits into a fiducial state (such as $|000\rangle$).
3. Conduct a broad/complete family of unitary operations.
4. Long enough significant decoherence time to do gate operations.
5. Acquire a measurement method with high-fidelity.

In this report, I will focus on the realization method based on nuclear magnetic resonance(NMR) processors.

4.1 Nuclear Magnetic Resonance

NMR - based quantum computation is performed over a system of spin-$\frac{1}{2}$ nuclei (the qubits) of molecules in some solution. In the following, we will first introduce the Hamiltonian governing the states of the nuclei, and then how quantum computation can be implemented with NMR will be summarized.

4.1.1 Hamiltonian of the system

Quantum mechanics has shown that for a single spin-$\frac{1}{2}$ nucleus, its Hamiltonian in a static magnetic field $\mathbf{B_0}$ is $H_0 = -\mu \cdot \mathbf{B_0} = -\gamma \mathbf{S} \cdot \mathbf{B_0}$, where γ is the gyromagnetic ratio of the nucleus and $\mu = \gamma S$ its magnetic moment, $S = \frac{1}{2} \times \sigma$ being the spin operator.

In practice, a quantum register is represented by several spin-$\frac{1}{2}$ atomic nuclei (the qubits) in a molecule. Due to the shielding effects by other electrons with

different degrees, generally we may denote the Hamiltonian for an n-qubit non-interacting molecule as follows(the system is in a static magnetic field directed along z):

$$H_0 = -\sum_{i=1}^{n} (1-\alpha^i)\gamma^i \ B_0 S_z^i = \sum_{i=1}^{n} \omega_0^i S_z^i \tag{35}$$

If the scalar-coupling effect is also taken into consideration, then the total Hamiltonian for a system with n qubits of coupled nuclear spins is

$$H_{sys} = \sum_i \omega_0^i S_z^i + \frac{2\pi}{\hbar} \sum_{i<j} J_{ij} S_z^i S_z^j, \tag{36}$$

where J_{ij} reflects the coupling strength between the ith and jth nuclei.

4.1.2 Measures to represent quantum computing

Finally, instead of measuring the states of ..., readout scheme:

4.1.3 Properties of NMR systems

NMR system has many features distinct from other quantum computation apparatus, for example,

1. Instead of measuring the output from a single copy of quantum register, we average over a large number of molecules to read the magnetization signal.

2. The experiment is conducted at room temperature, and thus on highly mixed (thermal) states rather than pure states.

3. The interaction between qubits is persistent, so that the Hamiltonian evolution due to undesired couplings has be removed manually (by refocusing techniques).

Nevertheless, NMR systems are still of particular interest, as many quantum algorithms and quantum properties including entanglement have been implemented and confirmed through NMR systems. Next we will discuss one particular experimental realization.

5 Application: NMR computer and quantum support vector machines

In this section, we discuss an experimental realization of a quantum algorithm – quantum support vector machine[1]. First the scheme of the experiment will be explained, and then some discussions on the experiment will be proposed.

5.1 General Theory

The experiment tries to classify digits (in this paper, the digits 6 and 9) using a quantum machine learning algorithm – support vector machine. First, digits are represented by feature vectors, and the classification result are represented by variable $y(= \pm 1)$, where that of 6 is denoted as "positive" (y=1) and that of 9 "negative". The algorithm completes the task to classify the class of a vector $\vec{x_0}$, i.e., decide the value of the corresponding variable $\vec{y_0}$.

The core of the algorithm is a hyperplane with $\vec{w} \cdot \vec{x} + b = 0$, subject to the assumption that $\vec{w} \cdot \vec{x_i} + b \geq 1$ for $\vec{x_i}$ with positive classification results, and $\vec{w} \cdot \vec{x_i} + b \leq -1$ for negative ones. The normal vector $\vec{w}$ is weighted sum of training vectors, i.e.,

$$\vec{w} = \sum_{i=1}^{M} \alpha_i \vec{x_i}, \tag{37}$$

where M is the number of training data, and in this paper M=2. The machine learning comes exactly from the objective to optimize $\vec{w}$ and b so that the distance between two classes$(2/|\vec{w}|)$ is maximal.

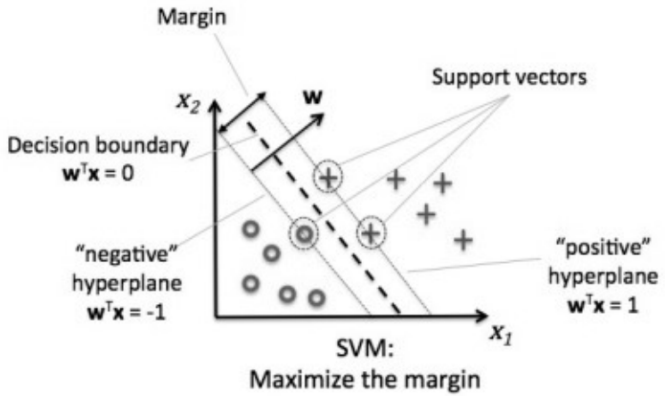


Figure 1: Illustration of support vector machines. [6]

The parameters $\vec{w}$ and b can be acquired by solving the equation

$$\tilde{F}(b, \alpha_1, ..., \alpha_M)^T = (0, y_1, ..., y_M)^T \tag{38}$$

following the least-squares approximation of SVM[5]. $\tilde{F}$ is a $(M+1) \times (M+1)$ matrix K entailing the similarity between the vectors. In this experiment, the intercept b is set to zero, thus the matrix is reduced to $dim(F) = M \times M$, and in this case 2×2.

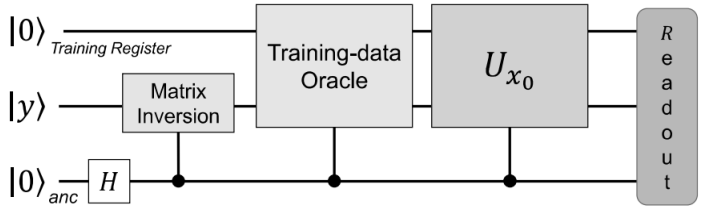


Figure 2: Experiment Scheme of Quantum SVM [1]

The whole algorithm contains roughly three parts:

1. Preparing the pseudo-pure state of the quantum system (which will not be discussed here),

2. extracting the feature vectors of both training and testing images, and preparing the kernel matrix K,

3. acquiring the hyperplane parameters $\vec{w}$ and classifying testing images.

The original paper illustrated the state flows with equations, and below I will fill in the explicit values of the matrices and quantum states for clarity and further discussion.

5.2 Numerical Analysis

First the feature vectors of the training data is encoded into two qubits to extract information about the density matrix K and the matrix F to be inverted, and we may get from Section 2.4,

$$K = \begin{bmatrix} 0.99945 & 0.49806 \\ 0.49806 & 0.99954 \end{bmatrix}, F = \begin{bmatrix} 1.49945 & 0.49806 \\ 0.49806 & 1.49954 \end{bmatrix}. \tag{39}$$

Next I will first give the result of the matrix inversion through classical methods, and then discuss the quantum approach for verification and comparison.

The matrix F may be decomposed by $F = VDV^{-1}$, where V's columns are eigenvectors and D's diagonal terms are eigenvalues:

$$F = \begin{bmatrix} -0.70714 & 0.70707 \\ 0.70707 & 0.70714 \end{bmatrix} \begin{bmatrix} 1.00143 & 0 \\ 0 & 1.99755 \end{bmatrix} \begin{bmatrix} -0.70714 & 0.70707 \\ 0.70707 & 0.70714 \end{bmatrix}, \tag{40}$$

and as the value $\mathbf{b} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, the hyperplane parameters are

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \tag{41}$$

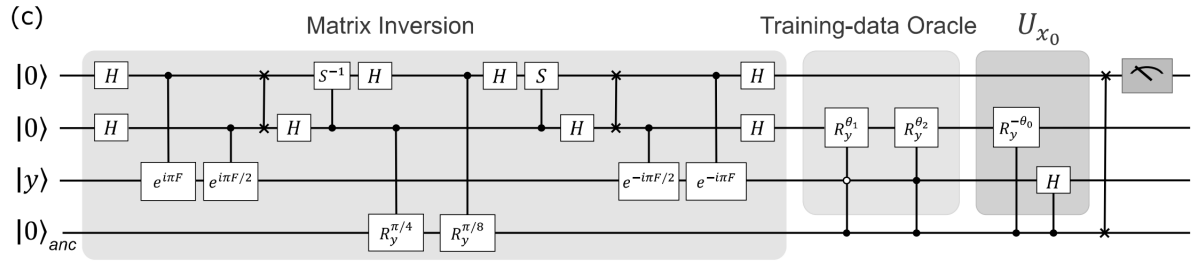


Figure 3: The quantum circuit for classification [1]

Then we shall discuss the approach of matrix inversion circuit.

First we decompose $|b\rangle$ in the eigenvector basis

$$|b\rangle = \beta_1 |v_1\rangle + \beta_2 |v_2\rangle = -0.99999 \begin{bmatrix} -0.70714 \\ 0.70707 \end{bmatrix} - 0.00005 \begin{bmatrix} 0.70707 \\ 0.70714 \end{bmatrix}, \tag{42}$$

using the results in Section 3.1.3, we get the state after Fourier transformation,

$$\sum_{\psi_1,\psi_2=0}^{1} \left[\sum \alpha_{\psi_1\psi_2|\beta_i} \beta_i |v_i\rangle \right] |\psi_1\rangle |\psi_2\rangle , \tag{43}$$

whose coefficients are,

$\alpha_{\psi_1\psi_2\|\beta_1}$	$\alpha_{\psi_1\psi_2\|\beta_2}$
$-0.001127600504415 + 0.001120014228117i$	$-0.000000000503854 - 0.000000087585601i$
$-0.999991138328462 - 0.003375226617191i$	$0.000000087249051 - 0.000000088258696i$
$0.001122537281808 + 0.001130140647694i$	$-0.000045675776705 + 0.000000262759379i$
$-0.000003797405739 + 0.001125071741380i$	$-0.000000087920863 - 0.000000086915082i.$

It can be immediately spotted that, the only nontrivial term in the superposition states is $|\psi_1\psi_2|\beta_1\rangle = |0\rangle |1\rangle |\beta_1\rangle$.

Rotating conditioned on the Fourier basis states, and then uncomputing the circuits, we may get the state

$$\sum \alpha_{\psi_1\psi_2|\beta_i} \beta_i |\psi_1\rangle |\psi_2\rangle |v_i\rangle \left(\sqrt{1 - \frac{C^2}{\lambda_j^2}} |0\rangle + \frac{C}{\lambda_j} |0\rangle \right), \tag{44}$$

$\alpha_{\psi_1\psi_2\|\|0\rangle}$	$\alpha_{\psi_1\psi_2\|\|1\rangle}$
$-0.5303067744762 + 0.1744070949332i$	$-0.7071005150385 - 0.002386645629057i$
$-0.1767951937665 - 0.176363194246662i$	$-3.508345439e^{-6} + 0.00103943075446800i$
$-0.1768010459975 - 0.1746293316798i$	$-1.454841077e^{-6} + 0.000431031261037493i$
$0.1767105347122 + 0.1767934597212i$	$0.000431031261037493 + 1.454841077e^{-6}i.$

It is worth mentioning that the circuit is successful conditioning on the fourth qubit reducing into $|1\rangle$, thus the first column's nontrivial results have no influence on the result, and the second column, parameters for $\alpha_{\psi_1\psi_2||1\rangle}$, concentrates on the state $|0\rangle |0\rangle$, whose fidelity is $(1 - 3 \times 10^{-6})$.

Moreover, the matrix inversion result is $\vec{a} = 0.99912 |v_1\rangle + 0.04197 |v_2\rangle$, which approximates to our result using classical methods.

Again calling the training-data oracle, from Section 2.4,

5.3 Discussion

First, I will discuss some of the improvements needed for the circuit to be used for practical classification.

By simple algebra, we can show that for a matrix of the form $A = \begin{bmatrix} m & n \\ n & m \end{bmatrix}$, the solution to the equation $A\vec{x} = \vec{b}$ is always $\vec{x} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, i.e., the vector $\vec{b}$ is an eigenvector of the matrix A, thus the matrix inversion part did not show its function here. However, this is not the general case for classification with larger training sets or feature vectors.

Furthermore, from previous analysis we have seen that, because the eigenvalue of the operator

Finally, we discuss another task of distinguishing the handwritten characters 0 and o. The task can be quite difficult for humans, while machine learning methods may offer us some insights into the problem. Below I will continue to use the support vector machine method to deal with the task. Apparently the left/right and up/down ratio used in the previous paper [1] is not suitable here due to the symmetric behaviors of the two objects. Instead, I introduced the parameter $\Delta y/\Delta x$, which is the farthest vertical distance over the horizontal one, and the parameter circum-radius over inscribed-radius R_o/R_i to represent the feature of the handwritten characters.

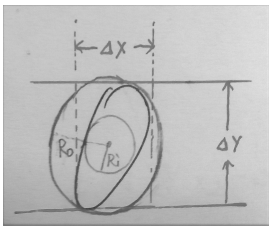


Figure 4: 1:Illustration of the parameters. 2: Handwritten'0'. 3: Handwritten'o'.

Using the two parameters to vectorize hand-written images, the following distribution is found:

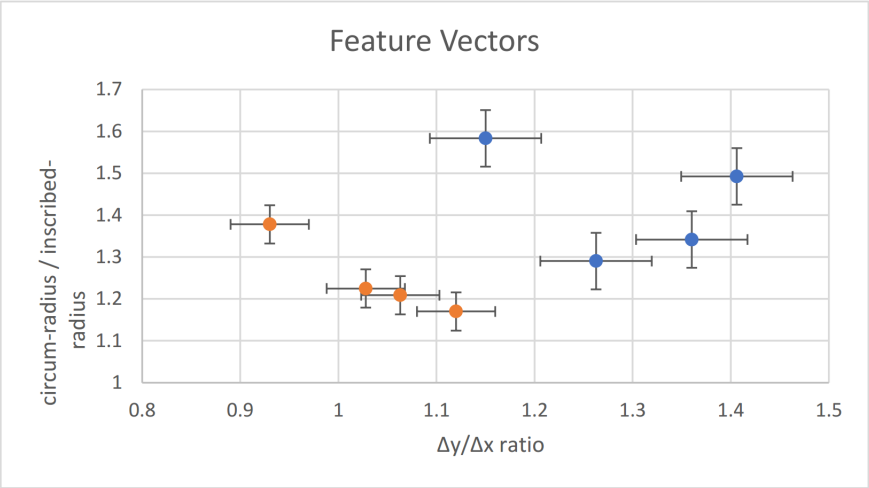


Figure 5: The features of training images. Blue dots represent '0', and orange dots represent 'o'.

We can find that, the coefficients for '0' are generally larger than that for 'o', which is in conformity with the definition of the two coefficients. Thus we can construct the decision boundary and normal vector $\vec{w}$. By using the same procedures as in the paper [1], we can train the support vector machine to classify the characters.

6 Conclusion

References

[1] Li, Zhaokai, et al. "Experimental realization of a quantum support vector machine." Physical review letters 114.14 (2015): 140504.

[2] Nielsen, Michael A., and Isaac Chuang. "Quantum computation and quantum information." (2002): 558-559.

[3] Benenti, Giuliano, Giulio Casati, and Giuliano Strini. Principles of Quantum Computation and Information: Basic tools and special topics. Vol. 2. World Scientific, 2007.

[4] Benenti, Giuliano, Giulio Casati, and Giuliano Strini, eds. Principles of quantum computation and information: Volume II: Basic Tools and Special Topics. World Scientific Publishing Co Inc, 2007.

[5] Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." Neural processing letters 9.3 (1999): 293-300.

[6] Maximum margin classification with support vector machines. [online] Available at: https://www.safaribooksonline.com/ library/view/python-deeper-insights/9781787128576/ch03s04.html [Accessed 13 Nov. 2017].