Reading notes on quantum linear systems

Guo Yuanxin

February 2020

This reading note basically follows from [DHM$^+$18]

1 Quantum Computing

The computational model we will use throughout is *gate-model quantum computation*. In classical computing, the input is a classical bit string. Through the application of finitely many classical gates, a output bit string is produced. This framework is known as the classical circuit model. In quantum computing, we follow the quantum circuit model, which is the analogue of the classical circuit model.

The basic unit of information is the qubit, $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$, where $\alpha_0, \alpha_1 \in \mathbb{C}$ and $|\alpha_0|^2 + |\alpha_1|^2 = 1$. This is represented by a column vector. We denote by $\langle\psi|$ the complex conjugate of $|\psi\rangle$. The pure states can be regarded as the basis vectors:

$$|0\rangle \cong (1,0)^T, \quad |1\rangle \cong (0,1)^T.$$

Hence the qubit is the normalized complex superposition (i.e. linear combination) over these basis vectors.

Multiple qubits are combined by the tensor product. Thus, an n-bit quantum state $|\psi\rangle$ can be expressed as

$$|\psi\rangle = \sum_{i_1,\ldots,i_n} \alpha_{i_1\ldots i_n} |i_1 \ldots i_n\rangle, \tag{1}$$

where $i_k \in \{0,1\}$, $\sum |\alpha_{i_1\ldots i_n}|^2 = 1$, and $|i_1 \ldots i_n\rangle = \bigotimes_{k=1}^n |i_k\rangle$. We call $\{\, |i_1 \ldots i_n\rangle \mid i_k \in \{0,1\}\}$ the *computational basis*. Note that describing a n-bit quantum state requires 2^n complex coefficients.

There are two types of operations that we can perform on a quantum state: *unitary operators* and *measurements*. A unitary operator satisfies $U^{-1} = U^{\dagger}$. Also, a *unitary operator* is norm-preserving, hence maps quantum states to quantum states. A unitary operator on n qubits can be expressed as a matrix of size $2^n \times 2^n$. Moreover, unitary operators are closed under composition. A *measurement* is described by a collection of operators (not necessarily unitary) $\{M_k\}$, where the index k indicates a given measurement outcome. The operators satisfy the *completeness equation*,

$$\sum_k M_k^{\dagger} M_k = I. \tag{2}$$

For a quantum state $|\psi\rangle$, the probability of measuring outcome m is given by (similar to the form $\mathbf{x}^T M \mathbf{x}$)

$$p(m) = \langle\psi| M_m^{\dagger} M_m |\psi\rangle \tag{3}$$

and the resulting quantum state is the normalized vector produced by applying M_m on $|\psi\rangle$

$$\frac{M_m |\psi\rangle}{\sqrt{\langle\psi| M_m^{\dagger} M_m |\psi\rangle}}. \tag{4}$$

The completeness equation implies that the measurement probabilities sum to unity. A *computational basis measurement*, $\{M_x\}$ for $x \in \{0,1\}^n$ consists of **unitary** operators $M_x = |x\rangle\langle x|$ (similar to outer product), the projectors onto the computational basis states.

In the quantum circuit model, we are given an input $x \in \{0,1\}^n$, which is a **classical** bit string. The first step is to prepare a m-qubit quantum input state $|\psi\rangle$, where $m = \text{poly}(n)$. A unitary operator U is then applied to $|\psi\rangle$, and finally the output state is measured (customarily in the computational basis). The measurement outcome corresponds to a classical bit string $y \in \{0,1\}^m$, which is obtained with probability

$$\langle\psi| U^{\dagger} M_y^{\dagger} M_y U |\psi\rangle = |\langle y| U |\psi\rangle|^2.$$

In practice, a quantum computer will be built on a **finite** set of quantum gates which act on a **finite** number of qubits. Typically, we consider gates acting on either **one** or **two** qubits, leaving the others invariant. A set of quantum gates S is said to be *universal* if any unitary operator can be approximated "well-enough" using only gates from this set. Namely, for any $\epsilon > 0$, $\exists$ sequence of operators $\{U_k \in S\}_{k=1}^{L}$ such that $\|U - U_L \dots U_1\|_2 \leq \epsilon$. Some examples of the universal sets are *Toffoli gate* (which acts on three qubits), *Hadamard gate*, or *single qubit rotations + C-NOT*. By **Solovay-Kitaev theorem**, $L = \mathcal{O}(\log^2 \frac{1}{\epsilon})$, and thus **exponential accuracy** can be achieved using a **polynomial** number of gates.

An important tool used in quantum computation is the *oracle*. Given a boolean function $f : \{0,1\}^n \to \{0,1\}^m$. The function is said to be queried via an *oracle* $\mathcal{O}_f$, if given the input $|x\rangle\,|q\rangle$ (where juxtaposition here means tensor product and $x \in \{0,1\}^n$, $y \in \{0,1\}^m$), we can prepare the output $|x\rangle\,|q \oplus f(x)\rangle$, where $\oplus$ denotes XOR. This can be implemented by a unitary circuit U_f, which takes the form

$$U_f = \sum_{x\in\{0,1\}^n} \sum_{q\in\{0,1\}^m} |x\rangle\langle x| \,\otimes\, |q \oplus f(x)\rangle\langle q|\,. \tag{5}$$

2 Quantum algorithms

2.1 Notations

Any integer between 0 and $N-1 = 2^n - 1$ can be expressed as an n-bit string $k = k_1 \dots k_n$ (binary representation). The Hadamard matrix is defined as

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{6}$$

The images of $|0\rangle$ and $|1\rangle$ are respectively denoted by $|+\rangle$ and $|-\rangle$.

2.2 Quantum Fourier Transform

2.2.1 DFT and FFT

The Fourier transform decomposes a signal into its fundamental components: frequencies. The DFT is the discrete counterpart of the continuous Fourier transform. Applying the DFT is equivalent to multiplying a **square, invertible** matrix D of size $N \times N$, where

$$D_{jk} = \frac{1}{\sqrt{N}}\omega^{jk}, \quad \omega = e^{i2\pi/N}. \tag{7}$$

It is not hard to verify that the columns of D are orthogonal ($D_j^\dagger D_k = \delta_{jk}$: Kronecker delta) and have unit length, thus the set of column vector are referred to as the *Fourier basis*.

If the DFT is applied to a vector using matrix multiplication, then the time complexity scales as $\mathcal{O}(N^2)$. Particularly, using a divide and conquer approach, this can be improved to $\mathcal{O}(N \log N)$, which is referred to as fast Fourier transform (FFT).

2.2.2 QFT

In a similar way, the QFT is defined by mapping each **computational basis state** (common approach in linear algebra) to a new quantum state. For convenience, we denote $\exp{(i2\pi/N)}$ by ω:

$$\text{QFT}: |x\rangle \mapsto |f_x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega^{x \cdot k} \, |k\rangle. \tag{8}$$

(Recall j is a integer between 0 and $N-1$.) Correspondingly, the inverse QFT is then defined as:

$$\text{QFT}^\dagger: |k\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \omega^{-k \cdot x} \, |x\rangle. \tag{9}$$

We decompose the two equations. Recall that $k = \sum_{l=1} n k_l 2^{n-l}$, and thus

$$\begin{align}
|f_x\rangle &= \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp(i2\pi x \sum_{l=1}^{n} k_l 2^{-l}) \, |k\rangle \notag \\
&= \frac{1}{\sqrt{N}} \sum_{k_1,\ldots,k_n} \exp(i2\pi x k_1 2^{-1}) \ldots \exp(i2\pi x k_n 2^{-n}) \, |k_1 \ldots k_n\rangle \tag{10} \\
&= \frac{1}{\sqrt{N}} \big(|0\rangle + \exp(i2\pi x k_1 2^{-1}) \, |1\rangle \big) \otimes \cdots \otimes \big(|0\rangle + \exp(i2\pi x k_1 2^{-n}) \, |1\rangle \big) \tag{11}
\end{align}$$

Note that $\frac{x}{2^m} \bmod 1 = 0.x_{n-m+1} \ldots x_n$ (since $\exp{(i2\pi)}$ is unity), we can finally write

$$|f_x\rangle = \frac{1}{\sqrt{N}} (|0\rangle + \exp(i2\pi 0.x_n) \, |1\rangle) \otimes \cdots \otimes (|0\rangle + \exp(i2\pi 0.x_1 \ldots x_n) \, |1\rangle) \tag{12}$$

From (12), we see that the information pertaining to the input x is disseminated throughout the relative phase on each individual qubit. (Each qubit contains different amount of information of x). Given the final state $|f_x\rangle$, applying the inverse QFT would yield the input string x. Equivalently, performing a measurement in the *Fourier basis* (where $M_k = |k\rangle\langle f_k|$) also yields x.

2.2.3 Implementation of the QFT

Our goal is to obtain the quantum state $|x\rangle$ after applying a quantum circuit to an all-zero input state. From (12) we observe that $|f_x\rangle$ is the tensor product of n qubits that are initialized in the state

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}},$$

which is the image of $|0\rangle$ under the Hadamard gate, and subsequently acquires a relative phase (regarded as a rotation) of $\exp(i2\pi x 2^{-k})$, where $1 \leq k \leq n$.

We first consider the first qubit. It is easy to see that a state of the form

$$\frac{1}{\sqrt{2}} \left(|0\rangle + \exp(i2\pi 0.x_n) |1\rangle \right)$$

corresponds to either $|+\rangle$ or $|-\rangle$, depending on the value of $x_n \in \{0, 1\}$, or compactly written $H |x_n\rangle$. Next, the state of the second qubit is given by

$$\frac{1}{\sqrt{2}} \left(|0\rangle + \exp(i2\pi 0.x_{n-1}x_n) |1\rangle \right),$$

which can be re-expressed as

$$\frac{1}{\sqrt{2}} \left(|0\rangle + \exp(i2\pi 0.x_{n-1}) \exp(i2\pi 0.0x_n) |1\rangle \right).$$

This corresponds to first preparing the state $H |x_{n-1}\rangle$, then applying a controlled rotation to the qubit, where the controlled rotation operator R_2 is given by

$$R_2 = \begin{bmatrix} e^{i2\pi \frac{0}{2^k}} & 0 \\ 0 & e^{i2\pi \frac{1}{2^k}} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i2\pi \frac{1}{2^k}} \end{bmatrix},$$

where the control is the n-th qubit. Now we extend it to the entire input state $|x\rangle$. First apply the Hadamard gate and appropriate controlled rotation controlled on qubits $|x_2\rangle, \ldots, |x_n\rangle$, we get

$$|x\rangle \mapsto \frac{1}{\sqrt{2}} \left(|0\rangle + \exp(i2\pi 0.x_1 \ldots x_n) |1\rangle \right) \otimes |x_2 \ldots x_n\rangle. \tag{13}$$

We proceed to the next qubit and repeatedly carry on this process, we have

$$|x\rangle \mapsto \frac{1}{\sqrt{2^n}} \left(|0\rangle + \exp(i2\pi 0.x_1 \dots x_n) |1\rangle \right) \otimes \cdots \otimes \left(|0\rangle + \exp(i2\pi 0.x_n) |1\rangle \right). \tag{14}$$

Now we have the output but in reverse order. Adding $\lfloor \frac{n}{2} \rfloor$ SWAP gates would yield the correct result. The total number of gates scales as $\mathcal{O}(n^2)$, and thus QFT can be efficiently implemented in a quantum circuit.

2.3 Hamiltonian simulation

Hamiltonian simulation is a important subroutine for HHL algorithm and a variety of quantum algorithms for ML.

In quantum mechanics, time evolution of the wave function $|\psi(t)\rangle$ is governed by the Schrödinger equation,

$$i\hbar \frac{\mathrm{d}}{\mathrm{d}t} |\psi(t)\rangle = \hat{H}(t) |\psi(t)\rangle. \tag{15}$$

Here $\hat{H}(t)$ is the *Hamiltonian*, an operator with units of energy, and $\hbar$ is Planck's constant. For $\hat{H}$ independent of time, the solution of the Schrödinger equation is

$$|\psi(t)\rangle = e^{-i\hat{H}t} |\psi(0)\rangle.$$

Our goal is to determine a quantum circuit which implements the unitary operator $U = e^{-i\hat{H}t}$.

The challenge is due to the fact that the application of matrix exponentials are computationally expensive. Naïve methods require $\mathcal{O}(N^3)$ for a $N \times N$ matrix, and in the quantum case, matrix size grows exponentially with number of qubits. As a consequence, a more efficient method called *Hamiltonian simulation* is introduced.

Definition 1 (Hamiltonian Simulation)**.** We say that a Hamiltonian $\hat{H}$ that acts on n qubits can be *efficiently simulated* if for any $t > 0$, $\epsilon > 0$, there exists a quantum circuit $U_{\hat{H}}$ consisting of $\mathrm{poly}(n, t, 1/\epsilon)$ gates such that $\left\| U_{\hat{H}} - e^{-i\hat{H}t} \right\| < \epsilon$.

Simulating Hamiltonians in general is BQP-hard, where BQP refers to the complexity class of decision problems efficiently solvable on a universal quantum computer.

Note that the dependency on time t is important, and at least time $\omega(t)$ is required to simulate $\hat{H}$ for time t, which is known as the **no fast-forwarding theorem**. However, there are no nontrivial lower bounds on the error dependency ϵ, hence making it barely possible to simulate an arbitrary Hamiltonian efficiently. However, certain classes of Hamiltonians are efficiently simulatable, which we will see.

Finally, since any unitary operator $U_{\hat{H}}$ can be expressed as $\exp(i\hat{H})$, we can similarly speak of an efficiently simulatable unitary operator.

2.3.1 Trotter-Suzuki methods

For any efficiently simulatable *unitary* operator U, we can always simulate $\hat{H}$ in a transformed basis $U\hat{H}U^\dagger$, since

$$e^{-iU\hat{H}U^\dagger t} = Ue^{-i\hat{H}t}U^\dagger. \tag{16}$$

This equation comes from the fact that $(U\hat{H}U^\dagger)^m = U\hat{H}^m U^\dagger$, given U unitary.

We now discuss the case where $\hat{H}$ is diagonal. Given efficient access to any diagonal element $H_{kk} = \langle k| \hat{H} |k\rangle$. We can simulate the Hamiltonian through the following steps:

$$\begin{aligned} |k,0\rangle &\rightsquigarrow |k, H_{kk}\rangle && (17)\\ &\rightsquigarrow e^{-iH_{kk}t} |k, H_{kk}\rangle && (18)\\ &\rightsquigarrow e^{-iH_{kk}t} |k, 0\rangle && (19)\\ &= e^{-i\hat{H}t} |k\rangle \otimes |0\rangle. && (20) \end{aligned}$$

In words, we first load the second register with the entry H_{kk}, and then apply a conditional phase, and finally reverse the loading process. This together with the change of basis argument enables the efficient simulatability of any diagonalizable matrix.

More generally, any k*-local* Hamiltonian, i.e., a sum of polynomially many terms in the number of qubits that each act only on $k = \mathcal{O}(1)$ qubits, can be simulated efficiently. This is because we can efficiently diagonlize each of the terms.

In general, we argue that if $\hat{H}_1$ and $\hat{H}_2$ are efficiently simulatable, $\hat{H}_1 + \hat{H}_2$ is efficiently simulatable. If the two Hamiltonians commute, this is direct from the **binomial theorem** and **Cauchy product formula** (for the product of two infinite series). However, this is not for the general case, i.e., when the operators don't commute. Here we need to use the

Lie product formula,

$$e^{-i(\hat{H}_1+\hat{H}_2)t} = \lim_{m\to\infty}\left(e^{-i\hat{H}_1 t/m}e^{-i\hat{H}_2 t/m}\right)^m. \tag{21}$$

The proof is given in the appendix.

If we want to restrict the simulation to a certain error ϵ, it suffices to limit m, which we call the number of steps, such that

$$\left\| e^{-i(\hat{H}_1+\hat{H}_2)t} - (e^{-i\hat{H}_1 t/m}e^{-i\hat{H}_2 t/m})^m \right\|_2 \le \epsilon, \tag{22}$$

By Taylor expansion,

$$\begin{aligned}(e^{-i\hat{H}_1 t/m}e^{-i\hat{H}_2 t/m})^m &= \left(I - i(\hat{H}_1+\hat{H}_2)\frac{t}{m} + \mathcal{O}(\frac{t^2\max(\hat{H}_1,\hat{H}_2)^2}{m^2})\right)^m \\ &= \left(e^{-i(\hat{H}_1+\hat{H}_2)\frac{t}{m}} + \mathcal{O}(\frac{t^2\max(\hat{H}_1,\hat{H}_2)^2}{m^2})\right)^m.\end{aligned}$$

Now let $A = e^{-i(\hat{H}_1+\hat{H}_2)t/m}$ and $B = \mathcal{O}(\frac{t^2\max(\hat{H}_1,\hat{H}_2)^2}{m^2})$. Expand the expression $(A+B)^m$, we have m first order terms in B, in the form of

$$A^{m-1-k}BA^k,$$

with k iterating from 0 to $m-1$. The higher order terms are absorbed in the $\mathcal{O}(\frac{t^2\max(\hat{H}_1,\hat{H}_2)^2}{m^2})$ term. Furthermore, by unitarity of A, they will not play a role in bounding the error. Hence

$$(e^{-i\hat{H}_1 t/m}e^{-i\hat{H}_2 t/m})^m = e^{-i(\hat{H}_1+\hat{H}_2)t} + \mathcal{O}(\frac{t^2\max(\hat{H}_1,\hat{H}_2)^2}{m}). \tag{23}$$

Combine (22) with (23), we have

$$\mathcal{O}\left(\frac{t^2\max(\|\hat{H}_1\|_2,\|\hat{H}_2\|_2)^2}{m}\right) \le \epsilon,$$

or

$$m = \mathcal{O}\left(\frac{t^2\max(\|\hat{H}_1\|_2,\|\hat{H}_2\|_2)^2}{\epsilon}\right). \tag{24}$$

This scheme is naïve and non-optimal. One can use higher-order approximation schemes such that $\hat{H}_1+\hat{H}_2$ can be simulated for time t in $t^{1+\delta}$ for any positive but arbitrarily small δ.

These **Trotter-Suzuki** schemes can be generalized to an arbitrary sum of Hamiltonians.

2.4 Quantum phase estimation

The goal of quantum phase estimation is to obtain a good approximation of an operator's eigenvalue given the associated eigenstate. Particularly, we consider a **unitary operator** U acting on an m-qubit state, with one of the eigenvectors being $|\psi\rangle$ and associated unknown eigenvalue being $\lambda = e^{i2\pi\phi}$, where $0 \leq \phi \leq 1$ is referred to as the phase. We wish to determine a good n-bit approximation $\tilde{\phi} = 0.\tilde{\phi}_1 \ldots \tilde{\phi}_n$ of ϕ, which will give a good n-bit approximation $\tilde{\lambda}$ of λ.

Our intuition is to use n ancillary qubits and encode the approximation within. Note that

$$U^j |\psi\rangle = \lambda^j |\psi\rangle = e^{i2\pi\phi j} |\psi\rangle.$$

We first prepare the $(n+m)$-qubit state

$$|0\rangle^{\otimes n} \otimes |\psi\rangle.$$

Next, n Hadamard gates are applied to the first n qubits, resulting in

$$\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle \otimes |\psi\rangle.$$

Then, in order to get an approximation, we apply n unitary operators to the last m qubits $|\psi\rangle$. More specifically, these are controlled-U^{2^k} gates controlled on each qubit of x. This gives

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{2^n-1} |x\rangle \left(U^x |\psi\rangle\right) = \frac{1}{\sqrt{N}} \sum_{x=0}^{2^n-1} \exp(i2\pi\phi x) |x\rangle |\psi\rangle.$$

Recall the approximation $0.\tilde{\phi}_1 \ldots \tilde{\phi}_n$. We can write $\phi = 0.\tilde{\phi}_1 \ldots \tilde{\phi}_n + \delta$. When $\delta = 0$, we can write the previous formula as

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{2^n-1} \exp(i2\pi \sum_{s=1}^{n} \tilde{\phi}_s 2^{-s} x) |x\rangle |\psi\rangle.$$

Observe that this is the QFT applied to the state $|\tilde{\phi}_1 \ldots \tilde{\phi}_n\rangle$. Performing a measurement recovers the qubit string. If $\delta \neq 0$, the measurement will have to be repeated in order to guarantee a certain level of error. More details are given in [Kit95].

Crucially, we do not need to be given access to the eigenvectors of the operator. Since any state $|u\rangle$ can be decomposed in the eigenbasis of the operator

$$|u\rangle = \sum_j \langle u|v_j\rangle\,|v_j\rangle = \sum_j \alpha_j\,|v_j\rangle.$$

Hence using the previous procedure, for an arbitrary state $|u\rangle$, we have

$$|0\rangle\,|u\rangle \mapsto \sum_j \alpha_j\,|\tilde{\theta}_j\rangle\,|v_j\rangle,$$

which gives the eigenbasis and estimated eigenvalues.

2.5 Phase kickback

We consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$. The function is queried via an oracle $\mathcal{O}_f$, and so

$$|x\rangle\,|q\rangle \to\, |x\rangle\,|q \oplus f(x)\rangle,$$

where $|x\rangle$ is the input state and $|q\rangle$ is an ancillary register. This operation can be implemented by a unitary operator U_f.

If the ancillary qubit is in the state $|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$, then by applying the oracle we obtain the state:

$$|x\rangle\frac{|f(x)\rangle - |\overline{f(x)}\rangle}{\sqrt{2}},$$

where adding a bar means flipping a bit. Readily, we see that this is equivalent to

$$(-1)^{f(x)}\,|x\rangle\,|-\rangle.$$

Thus, inputs which evaluate to 1 acquire a relative phase of -1. This process is referred to as phase kickback.

2.6 Amplitude amplification

Amplitude amplification is an extension of Grover's search algorithm. We first present Grover's search algorithm.

We are given a set $\{1, \ldots, N\}$ $(N = 2^n)$ where exactly one element is marked. Otherwise put, we are given a function $f : \{1, \ldots, N\} \to \{0, 1\}$ defined by

$$f(x) = \begin{cases} 1, & x = a \in \{1, \ldots, N\}, \\ 0, & \text{otherwise.} \end{cases}$$

This is a Boolean function, hence with an oracle using the phase kickback technique, we can evaluate $f(x)$ with unit computational cost.

Grover's algorithm speeds up the root finding of $\neg f(x)$ from $\mathcal{O}(N)$ (brute-force) oracle queries to $\mathcal{O}(\sqrt{N})$ oracle queries. Furthermore, it has been shown that this is the optimal scaling of the number of queries.

First, we prepared n qubits in the $|0\rangle^{\otimes n}$ state. By applying $H^{\otimes n}$ gate, we acquire the **uniform superposition state** $|s\rangle$,

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle.$$

An ancillary qubit is prepared in the $|-\rangle$ state. Applying the phase kickback protocol to $|s\rangle$, we obtain

$$\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{f(x)} |x\rangle |-\rangle.$$

We now discard the second register and denote the state in the first register by $|\psi\rangle$,

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{f(x)} |x\rangle$$

Specially, we denote by U_ω the unitary operator that queries the oracle, that is

$$U_\omega : \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \quad \mapsto \quad \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} (-1)^{f(x)} |x\rangle.$$

Now it's time for intuition to kick in. We know that our last step is to do a measurement. In order to yield the marked state with high probabilty, we aim to **dampen** the coefficients associated with unmarked items, and **strengthen** the coefficient corresponding to the marked item. The *Grover diffusion operator* does the trick.

The *Grover diffusion operator* is defined by

$$U_s := 2\left|s\right\rangle\!\left\langle s\right| - I.$$

It is closely related with the **Householder transformations**, which reflects a vector with respect to a hyperplane. To see why the operator works, we first apply the operator to basis states:

$$U_s\left|x\right\rangle = (2\left|s\right\rangle\!\left\langle s\right| - I)\left|x\right\rangle = \frac{2}{N}\left(\sum_x \left|x\right\rangle\right) - \left|x\right\rangle.$$

Thus for any state $\psi = \sum_x \alpha_x \left|x\right\rangle$,

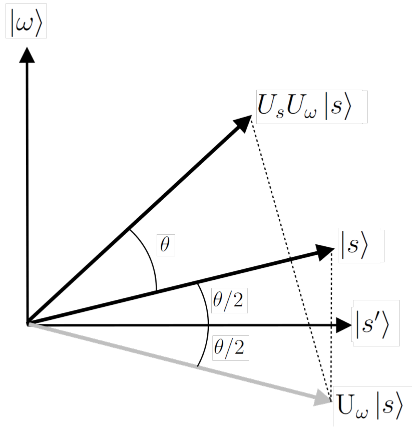
$$U_s\left|\psi\right\rangle = \sum_x \left(2\langle\alpha\rangle - \alpha_x\right)\left|x\right\rangle,$$

where $\langle\alpha\rangle = \sum_x(\frac{\alpha_x}{N})$ is the mean of the coefficients. In Grover's algorithm, the input of the diffusion operator is the output of U_ω:

$$\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}(-1)^{f(x)}\left|x\right\rangle.$$

Here $\langle\alpha\rangle = \frac{N-2}{N\sqrt{N}}$. The unmarked items have coefficients $\frac{1}{\sqrt{N}}$, and an application of U_s reduces them to $\frac{N-4}{N\sqrt{N}}$, while the marked item has coefficient $\frac{3N-4}{N\sqrt{N}}$. This illustrates the operator indeed does what we claim of it. Applying $G = U_sU_\omega$ repeatedly, the positive coefficients will eventually dampen to 0 while the negative efficient will be magnified and become positive.

We use a geometric approach to further illustrate.



$\left|\omega\right\rangle$
$U_sU_\omega\left|s\right\rangle$
θ
$\left|s\right\rangle$
$\theta/2$
$\left|s'\right\rangle$
$\theta/2$
$\mathrm{U}_\omega\left|s\right\rangle$

Figure 1: An illustration of Grover's operator.

For the first iteration, we start with the uniform superposition state s. Applying U_ω, it is equivalent to a reflection about $|s'\rangle$, where

$$|s'\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq \omega} |x\rangle$$

Clearly, $|s'\rangle$ is perpendicular to the basis state that is marked. Then applying U_s is equivalent to reflecting about the state $|s\rangle$. Combining these two operations, the net effect is a rotation by an angle θ towards the marked state $|\omega\rangle$. Obviously, we have the following geometric relationship:

$$\sin\frac{\theta}{2} = \frac{1}{\sqrt{N}}.$$

Rotating r times, the probability of the measurement yields $|\omega\rangle$ is $\sin((r+\frac{1}{2})\theta)$. To have this close to one, we need $(r+\frac{1}{2})\theta \approx \frac{\pi}{2}$. Using the small angle approximation, $\frac{\theta}{2} \approx \sin\frac{\theta}{2} = \frac{1}{\sqrt{N}}$. Thus Grover's algorithm scales as $\mathcal{O}(\sqrt{N})$.

This algorithm can be extended to the case where there are M marked items out of the N. The state $|\omega\rangle$ is replaced by the state $|m\rangle = \frac{1}{\sqrt{M}}\sum_{x_m}|x_m\rangle$, where x_m's are the marked items, and the state $|s'\rangle$ is replaces by the state $|u\rangle = \frac{1}{\sqrt{N-M}}\sum_{x_u}|x_u\rangle$, where x_u's are the unmarked items. The geometric intuition of this generalization is splitting the total space $\mathcal{H}$ as two subspaces,

$$\mathcal{H} = \mathcal{H}_m \oplus \mathcal{H}_u,$$

where $\oplus$ here means direct sum. Applying the Grover's operator rotates the quantum state towards the marked subspace.

Amplitude amplification is the process of applying **Grover's algorithm** to tasks where we have an oracle for a function f and we need to sample x from a "good" subset, i.e. $\{x \mid f(x) = 1\}$. This give us the following lemma:

Lemma 1 (Amplitude amplification)**.** *Suppose we have an algorithm $\mathcal{A}$ that succeeds with probability ϵ. Using amplitude amplification, we can take $\mathcal{O}(1/\sqrt{\epsilon})$ repetitions of $\mathcal{A}$ to yield an algorithm $\mathcal{A}'$ with success probability arbitrarily close to one.*

Note that $\mathcal{A}$ can either be a classical or quantum algorithm. We can realize that by taking the "good" subspace as successful bitstring outputs of $\mathcal{A}$. A simple way is to append a 1 at the end of successful evaluations and append a 0 otherwise.

2.7 The uncompute trick

In quantum computing, we usually need to set an arbitrary state to $|0\rangle$. Yet there's no single-qubit unitary operator can do this. Consider the function $f : \{0,1\}^n \to \{0,1\}^m$ A more common case is the map

$$|x\rangle |0\rangle |0\rangle \mapsto |x\rangle |g(x)\rangle |f(x)\rangle,$$

where $|g(x)\rangle$ is an **garbage state** in a **working register**. We assume that the working register should be initialized to $|0\rangle$ for any quantum algorithms, otherwise the following computation will be messed up. Even worse, in the case the mapping $|x\rangle |0\rangle |0\rangle \mapsto |x\rangle |g(x)\rangle |f(x)\rangle$ isn't **perfect**, the working register and the output register are **entangled**, i.e.

$$|x\rangle |0\rangle |0\rangle \mapsto \sum_{f_y \in \{0,1\}^m} \alpha_y |x\rangle |y\rangle |f_y\rangle.$$

This implies operations on the garbage register will affect the output register. Furthermore, we want to keep the input state $|x\rangle$.

Suppose we have an unitary operator U_f that realizes the previous mapping. We first consider the ideal case, where $f_y \equiv f(x)$. In this case the output is given by

$$\sum_y \alpha_y |x\rangle |y\rangle |f(x_0)\rangle = |x\rangle \left(\sum_y \alpha_y |y\rangle \right) |f(x_0)\rangle.$$

Here $\sum_y \alpha_y |y\rangle = g(x)$. We appending an additional register and apply the CNOT gate (fundamentally, bitwise-XOR gate) controlled on the third register on the newly added register, we get

$$\sum_y \alpha_y |x\rangle |y\rangle |f(x_0)\rangle |f(x_0)\rangle.$$

Then apply the inverse operator U_f^{-1}. The state evolves to

$$|x\rangle |0\rangle |0\rangle |f(x_0)\rangle.$$

Finally, applying the SWAP operator on the last two registers, we get $|x\rangle |0\rangle |f(x_0)\rangle |0\rangle$, where it is now safe to discard the working register and the ancillary register. This is the *uncompute* trick.

The scenario where U_f is unrealistic. Assume the error probability is ϵ,

$$\sum_{y:f_y=f(x)} |\alpha_y|^2 = 1-\epsilon.$$

After applying U_f, appending the ancillary register, and applying the CNOT, we have the state

$$|\phi'\rangle = \sum_y \alpha_y |x\rangle |y\rangle |f_y\rangle |f_y\rangle,$$

where the ideal final state $|\phi\rangle$ is given by

$$|\phi\rangle = \sum_y \alpha_y |x\rangle |y\rangle |f(x)\rangle |f(x)\rangle.$$

The inner product is

$$\langle \phi' | \phi \rangle = \sum_y |\alpha_y|^2 = 1-\epsilon$$

Finally, applying the unitary operator U_f^{-1} preserves the inner product. Therefore, we guarantee if the unitary operator acts to within ϵ, the mapping $|x\rangle |0\rangle \mapsto |x\rangle |f(x)\rangle$ can be enacted to within error ϵ.

A Proof of Lie product formula

This proof basically follows from [Her14].

Theorem 1. *For A and $B \in \mathbb{C}^{d\times d}$,*

$$\lim_{k\to\infty} (\exp(A/k)\exp(B/k))^k = \exp(A+B)$$

We prove a stronger result:

Theorem 2. *Let f, g be* ***entire*** *functions such that $f(0) = g(0) = 1$. Then*

$$\lim_{k\to\infty} (f(A/k)g(B/k))^k = \exp(f'(0)A + g'(0)B)$$

for A and $B \in \mathbb{C}^{d\times d}$.

In the proof of Theorem 2, we make use of the formula

$$\lim_{k\to\infty}(I+\frac{A}{k})^k = \exp(A), \tag{25}$$

which can be proved in matrix (or Banach) algebra as well as for complex numbers. Moreover, we utilize the following lemma:

Lemma 2. *If A and $B \in \mathbb{C}^{d\times d}$, $k \in \mathbb{N}$*

$$\|(A+B)^k - A^k\| \leq (\|A\|+\|B\|)^k - \|A\|^k$$

Proof. We use a combinatorial approach to prove this. Let $\mathcal{T}$ be the set of all k-tuple $\Gamma = (\Gamma_1, \ldots, \Gamma_k)$, $\Gamma_j \in \{A, B\}$ for all $1 \leq j \leq k$ with $\Gamma_j = B$ *for at least one* j. We have

$$\|(A+B)^k - A^k\| = \left\|\sum_{\Gamma\in\mathcal{T}}\prod_{j=1}^{k}\Gamma_j\right\| = \sum_{\Gamma\in\mathcal{T}}\prod_{j=1}^{k}\|\Gamma_j\| = (\|A\|+\|B\|)^k - \|A\|^k$$

□

Proof of Theorem 2. Since f, g are entire, let

$$f(z) = \sum_{n=0}^{\infty}\alpha_n z^n \qquad \text{and} \qquad g(z) = \sum_{n=0}^{\infty}\beta_n z^n$$

be the power series expansions of f and g. For each $k \in \mathbb{N}$,

$$f(\frac{A}{k}) = I + f'(0)\frac{A}{k} + U_k,$$

where

$$U_k = \sum_{n=2}^{\infty}\alpha_n\frac{A^n}{k^n},$$

and hence

$$\|k^2 U_k\| \leq \sum_{n=2}^{\infty}|\alpha_n|\frac{\|A\|^n}{k^{n-2}} \leq \sum_{n=2}^{\infty}|\alpha_n|\|A\|^n.$$

Analogously,

$$g(\frac{B}{k}) = I + g'(0)\frac{B}{k} + V_k \qquad \text{and} \qquad \|k^2 U_k\| \leq \sum_{n=2}^{\infty}|\alpha_n|\|B\|^n.$$

Thus,

$$\begin{aligned} f(\frac{A}{k}) \cdot g(\frac{B}{k}) &= (I + f'(0)\frac{A}{k} + U_k)(I + g'(0)\frac{B}{k} + V_k) \\ &= I + \frac{f'(0)A + g'(0)B}{k} + W_k, \end{aligned}$$

where $k^2 W_k$ is bounded because f, g are entire:

$$\|W_k\| \leq \frac{b}{k^2}$$

Setting $C = f'(0)A + g'(0)B$, and by applying Lemma 2 and the **Mean Value Theorem**, we obtain

$$\begin{aligned} &\left\| \left(f(\frac{A}{k}) \cdot g(\frac{B}{k})\right)^k - \left(I + \frac{C}{k}\right)^k \right\| \\ &\qquad = \left\| \left(I + \frac{C}{k} + W_k\right)^k - \left(I + \frac{C}{k}\right)^k \right\| \\ &\qquad \leq \left(\left\|I + \frac{C}{k}\right\| + \frac{b}{k^2}\right)^k - \left\|I + \frac{C}{k}\right\|^k \\ &\qquad = k\xi_k^{k-1}\frac{b}{k^2} \\ &\qquad = \frac{b\xi_k^{k-1}}{k} \end{aligned}$$

for some

$$\xi_k \in \left(\left\|I + \frac{C}{k}\right\|, \left\|I + \frac{C}{k}\right\| + \frac{b}{k^2}\right).$$

We plan to bound

$$\lim_{k\to\infty} \xi_k^{k-1}$$

by a constant. Note that

$$0 \leq \xi_k^{k-1} \leq \left(\left\|I + \frac{C}{k}\right\| + \frac{b}{k^2}\right)^{k-1} \leq \left(1 + \frac{\|C\| + b}{k}\right)^k,$$

and

$$\lim_{k\to\infty} \left(1 + \frac{\|C\| + b}{k}\right)^k = \exp(\|C\| + b).$$

Hence

$$\lim_{k\to\infty} \frac{b\xi_k^{k-1}}{k} = 0.$$

Plug in (25), we finish the proof. □

References

[DHM$^+$18] Danial Dervovic, Mark Herbster, Peter Mountney, Simone Severini, Naïri Usher, and Leonard Wossnig. Quantum linear systems algorithms: a primer, 2018.

[Her14] Gerd Herzog. A proof of lie's product formula. *The American Mathematical Monthly*, 121(3):254–257, 2014.

[Kit95] A. Yu. Kitaev. Quantum measurements and the abelian stabilizer problem, 1995.