Quantum Computing 3/10/2022

Lecture 13: Fast Forwarding

Instructor: Dieter van Melkebeek Scribe: Jonah Liu, Tiger Ji

We discuss the fast forwarding property of quantum random walks, a key property in searching. To demonstrate this, we involve block encodings, Chebyshev polynomials, and linear combinations of unitaries (LCU). Finally, we introduce interpolating walks.

1 Recap

1.1 From Random to Quantum Walk

Consider a weighted graph $G = (V, E)$ without any isolated vertices (i.e., the sum of all the weights of all the edges incident to any vertex is positive). In the classical setting, when at vertex u, move to vertex v with probability proportional to the weight of the corresponding edge. This can be modeled as a Markov chain with transition matrix T such that:

$$T_{uv} = \Pr[\text{move to } u | \text{at } v]$$

There is a unique stationary distribution π provided that G is connected, and convergence to the distribution is guaranteed given G is connected and non-bipartite.

In a quantum walk, we act on pairs of vertices, namely $|u, v\rangle$ where u represents the previous or next vertex and v represents the current one. Each step of the quantum walk can be decomposed into two parts: a coin flip C, where we decide the next node to go to, and a swap, where we actually "move" from one vertex to the next. While there are several choices for the coin flip, the reflection coin enables the square root speed-up results.

Specifically, the two steps can be formalized as:

- Reflection Coin C: Reflects the first component of $|u, v\rangle$ about $|N_v\rangle \doteq \sum_{u'} \sqrt{T_{u'v}} |u'\rangle$. Assuming there exists some unitary $U : |0^n\rangle |v\rangle \mapsto |N_v\rangle |v\rangle$, then C can be expressed as $C = U R_{|0^n *\rangle} U^*$
- Swap S: Swap vertex u and v to realize $|u, v\rangle \mapsto |v, u\rangle$

Theorem 1. *There exists a quantum algorithm that takes an undirected weighted graph $G = (V, E)$ without isolated vertices and a black-box for $f : V \to \{0, 1\}$, and outputs $v \in V$ with $f(v) = 1$ in expected time $\tilde{O}(S + \sqrt{H(T, f)}(U + C))$.*

The quantum walk versions of S, U, C are similar to the classical random walk. Here are the precise definitions of the quantities in Theorem 1.

- $H(T, f)$: smallest t such that Pr[random walk visits v such that $f(x) = 1$ within t steps]
- S: Time to sample from π: Create $|\pi\rangle \doteq \sum_{v \in V} \sqrt{\pi(v)} |v\rangle$
- U: Time for one step of the quantum walk/application of SC:

- S: $|u,v\rangle \mapsto |v,u\rangle$
- C: $|u,v\rangle \mapsto R_{|N_v\rangle}|u\rangle|v\rangle$
- $C = UR_{|0^n *\rangle}U^*$ where $U : |0^n\rangle|v\rangle \mapsto |N_v\rangle|v\rangle$.

◦ C: Time to check $f(v) = 1$: Apply $|v\rangle|b\rangle \mapsto |v\rangle|b \oplus f(v)\rangle$

1.2 Block Encoding

Block encodings are essential for fast forwarding, so we will review them here.

Definition 1. *A block encoding of a matrix M acting on m qubits is a unitary (circuit) A acting on $l+m$ qubits such that*

$$A = \begin{bmatrix} M & * \\ * & * \end{bmatrix}$$

The block encoding can be used as a probabilistic encoding of M with a success indicator as follows:

1. Apply A to state $|0^l\rangle|\psi\rangle$ representing 0 in all ancillas and $|\psi\rangle$ being the state to which we want to apply M.
2. Measure the first register (first l qubits).
3. If the outcome is 0^l, the second register is in the state $M|\psi\rangle / ||M|\psi\rangle||_2$. For this encoding to be useful, $||M|\psi\rangle||_2^2$, the probability of observing 0^l, should be sufficiently large.

To express that A is a block encoding of M, we project onto the subspace where the first l qubits are set to 0. This projection can be realized by:

$$P \doteq P_{|0^l *\rangle} = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}$$

As a result, we have

$$PAP = P\begin{bmatrix} M & * \\ * & * \end{bmatrix}P = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}$$

2 Fast Forwarding

Now we state the fast forwarding property of quantum walks, which is the key ingredient for speedups compared to classical random walks. For simplicity, we consider the case where T is symmetric, but this generalizes to non-symmetric transition matrices as well.

Lemma 2. *There exists a quantum algorithm that for any $t \in \mathbb{N}$ and $\eta > 0$, realizes a block encoding of matrix M such that $||T^t - M||_2 \leq \eta$ in time $O(\sqrt{t\log(1/\eta)}\mathrm{U})$, where* U *represents the cost of one application of the quantum walk operator ($U^*, R_{|0^n *\rangle}, U$, and S).*

Here, t is the number of steps to simulate and η is the accuracy parameter. The idea behind the lemma is that we can find an approximation for T^t to simulate t random walk steps in time $O(\sqrt{t})$. The outline of the proof is as follows:

- Connection with Chebyshev polynomial T_d of degree d: $(SC)^d$ encodes $UT_d(T)U^*$
- Approximation of T^t by linear combination of $T_d(T)$ for d around $\Theta(\sqrt{t})$
- Linear combination of unitaries (LCU) yields desired block encoding

2.1 Iterates of Quantum Walk Operator

For the following section, recall the notation/definitions:

- S: $|u,v\rangle \mapsto |v,u\rangle$
- U: $|0^n\rangle |v\rangle \mapsto |N_v\rangle |v\rangle$
- $R \doteq R_{|0^n*\rangle}$: reflection about $|0^n*\rangle$

From the previous section, we have that

$$\begin{aligned} SC &= SURU^* && (C = URU^*) \\ &= (UU^*)SURU^* && (UU^* = I) \\ &= U(U^*SU)RU^* \\ &= U\tilde{S}RU^* && (\tilde{S} \doteq U^*SU) \\ &= UQU && (Q \doteq \tilde{S}R) \end{aligned}$$

where $\tilde{S} \doteq U^*SU$ and $Q \doteq \tilde{S}R$. Notice that

$$\begin{aligned} (SC)^2 &= (UQU^*)(UQU^*) \\ &= UQ^2U^* \end{aligned}$$

In general, we have that

$$(SC)^d = UQ^dU^*$$

2.2 Block of Q

Consider the top left block of $\tilde{S} \doteq U^*SU$, where $U : |0^n\rangle |v\rangle \mapsto |N_v\rangle |v\rangle$. To obtain the entry $\tilde{S}_{0^nu,0^nv}$, we multiply $\tilde{S}$ to the right with the basis vector corresponding to the column index, and to the left with the complex conjugate transpose of the basis vector corresponding to the row index:

$$\begin{aligned} \tilde{S}_{0^nu,0^nv} &= \langle 0^nu| \tilde{S} |0^nv\rangle \\ &= \langle 0^nu| U^*SU |0^nv\rangle && (\tilde{S} \doteq U^*SU) \\ &= (U |0^nu\rangle)^* S(U |0^nv\rangle) && ((U |0^nu\rangle)^* = \langle 0^nu| U^*) \\ &= (|N_u\rangle |u\rangle)^* S(|N_v\rangle |v\rangle) && (U : |0^n\rangle |v\rangle \mapsto |N_v\rangle |v\rangle) \\ &= (|N_u\rangle |u\rangle)^* (|v\rangle |N_v\rangle) && (S : |u,v\rangle \mapsto |v,u\rangle) \\ &= \sqrt{T_{vu}} \cdot \sqrt{T_{uv}} \\ &= T_{uv} \end{aligned}$$

Since $|N_v\rangle \doteq \sum_{u'} \sqrt{T_{u'v}}\,|u'\rangle$, the inner product of $|u\rangle$ and $|N_v\rangle$ is $\sqrt{T_{uv}}$, since all terms besides $u = u'$ disappear. The same reasoning can be applied to the inner product of $|v\rangle$ and $|N_u\rangle$. The last step is due to the symmetry of T. In the general case, this will be some discriminant matrix instead. This means that $\tilde{S}$ is a block encoding of T:

$$P\tilde{S}P = \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix}$$

To find the top left block of Q, we first argue that R and $2P - I$ have the same effect. For all vectors v in the subspace $|0^n{*}\rangle$, which is the axis we reflect about, both R and P have no effect on them. Therefore, $(2P-I)v = 2Pv - Iv = 2v - v = v$. However, if the vector v is orthogonal to the subspace $|0^n{*}\rangle$, R flips the sign and P projects to 0. Therefore, $(2P-I)v = 2Pv - Iv = 0 - v = -v$. In both cases, R behaves the same as $2P - I$, so

$$PQP = P\tilde{S}RP = P\tilde{S}(2P-I)P = 2P\tilde{S}P^2 - P\tilde{S}P = P\tilde{S}P = \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix}$$

This comes from the fact that P^2 projects the vector twice, which is the same as projecting it once, thus $P^2 = P$. As a result, we have shown that $\tilde{S}$ and Q both block encode T.

2.3 Recurrence for Blocks of Q's Iterates

We are interested in the blocks for the higher powers of Q, so we setup a recurrence which turns out to be the same one as for the Chebyshev polynomials:

$$\begin{aligned}
PQ^{d+1}P &= P(\tilde{S}R)Q^dP && (Q \doteq \tilde{S}R, Q^{d+1} = QQ^d = (\tilde{S}R)Q^d) \\
&= P\tilde{S}(2P-I)Q^dP && (R \text{ behaves the same as } 2P-I) \\
&= 2P\tilde{S}PQ^dP - P\tilde{S}Q^dP \\
&= 2(P\tilde{S}P)PQ^dP - P\tilde{S}(\tilde{S}R)Q^{d-1}P && (P = P^2, Q \doteq \tilde{S}R) \\
&= 2(P\tilde{S}P)PQ^dP - PRQ^{d-1}P && (\tilde{S}^2 = (U^*SU)(U^*SU) = U^*S^2U = U^*U = I) \\
&= 2(P\tilde{S}P)PQ^dP - PQ^{d-1}P && (PR = P \text{ since reflect has no effect after projection}) \\
&= 2\begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix}PQ^dP - PQ^{d-1}P && (P\tilde{S}P = \begin{bmatrix} T & 0 \\ 0 & 0 \end{bmatrix})
\end{aligned}$$

Since multiplying by P on the left and right of Q^d produces a matrix with the top left block of Q^d, we can define $M_d \doteq$ top left block of Q^d. Therefore, the recurrence for M_d is:

$$M_0 = I$$

$$M_1 = T$$

$$M_{d+1} = 2 \cdot T \cdot M_d - M_{d-1} \text{ for } d \geq 1$$

2.4 Connection with Chebyshev Polynomials

Chebyshev polynomials are defined by the following recurrence relation:

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_{d+1}(x) = 2 \cdot x \cdot T_d(x) - T_{d-1}(x) \text{ for } d \geq 1$$

Therefore, from the recurrence, we have shown that Q^d is a block encoding of the d-th Chebyshev polynomial applied to the transition matrix T, $T_d(T)$. Recall that, up to the basis transformation, $Q^d = U^*(SC)^d U$. From this definition, we also have that $U^*(SC)^d U$ block encodes $T_d(T)$.

2.5 Properties of Chebyshev Polynomials

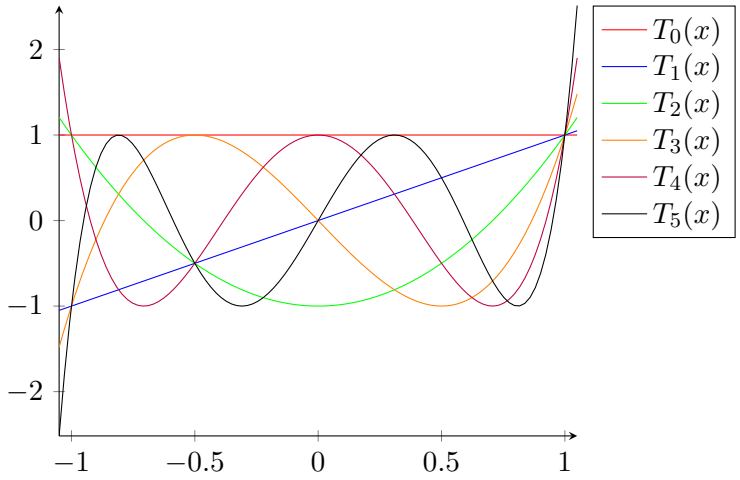
Now that we are given the definition of Chebyshev polynomials, we observe a few properties:

- T_d is a polynomial of degree d
  - This follows from the recurrence relation for T_d.
- $T_d(\cos(\theta)) = \cos(d\theta)$ for $\theta \in \mathbb{R}$.
  - This follows by induction. For $d = 0$, $T_0(x) = 1 = \cos(0\theta)$. Assume that $T_d(\cos(\theta)) = \cos(d\theta)$ for some arbitrary $d > 0$. Therefore:

$$\begin{aligned}
T_{d+1}(\cos(\theta)) &= 2\cos(\theta)T_d(\cos(\theta)) - T_{d-1}(\cos(\theta)) && \text{(by definition of } T_d(x)) \\
&= 2\cos(\theta)\cos(d\theta) - \cos((d-1)\theta) && \text{(by inductive hypothesis)} \\
&= 2\cos(\theta)\cos(d\theta) - (\cos(d\theta)\cos(\theta) + \sin(d\theta)\sin(\theta)) && \text{(trigonometric identity)} \\
&= \cos(\theta)\cos(d\theta) - \sin(d\theta)\sin(\theta) \\
&= \cos((d+1)\theta) && \text{(trigonometric identity)}
\end{aligned}$$

- $T_d(x) \in [-1, 1]$ for $x \in [-1, 1]$
  - For any $x \in [-1, 1]$, there exists $\theta \in \mathbb{R}$ such that $\cos(\theta) = x$. Therefore, $T_d(x) = T_d(\cos(\theta)) = \cos(d\theta) \in [-1, 1]$.

Here are the first few Chebyshev polynomials:



Any polynomial of degree t can be expressed as a linear combination of the Chebyshev polynomials up to degree t. This will be helpful later on when attempting to approximate T^t, which is closely related. To find the Chebyshev expansion of x^t, we recall the recurrence of Chebyshev Polynomials:

$$\begin{aligned} T_{d+1}(x) &= 2xT_d(x) - T_{d-1}(x) \\ xT_d(x) &= \frac{1}{2}(T_{d+1}(x) + T_{d-1}(x)) \\ &= E_\Delta[T_{d+\Delta}(x)] \qquad \text{where } \Delta \in_u \pm 1 \end{aligned}$$

Note that this relationship extends to $d \in \mathbb{Z}$ when $T_{-d}(x) \doteq T_d(x)$. The following expansion property allows us to write x^t as a linear combination of Chebyshev polynomials up to degree d.

Theorem 3. *For every $t \in \mathbb{N}$, $x^t = E_{d_t}[T_{d_t}(x)]$, where d_t is the sum of t independent uniform ± 1 random variables.*

Proof. Base case: $t = 0$

Induction step $t \to t+1$ for $t \in \mathbb{N}$:

$$x^{t+1} = x \cdot x^t = x \cdot E_{d_t}[T_{d_t}(x)] = E_{d_t}[xT_{d_t}(x)] = E_{d_t}[E_\Delta[T_{d_t+\Delta}(x)]] = E_{d_{t+1}}[T_{d_{t+1}}(x)] \qquad \square$$

In this linear combination, the weight will be concentrated on polynomials up to degree $\sqrt{t}$

2.6 Chebyshev Approximation to x^t

Expansion property For every $t \in \mathbb{N}$, $x^t = E_{d_t}[T_{d_t}(x)]$ where d_t is the sum of t independent uniform ± 1 random variables.

Chernoff bound For each $a \in (0, \infty), \Pr[d_t \geq a] \leq \exp(-a^2/(2t))$.

Truncated expansion To achieve an approximation for a given accuracy at degree a:
Let $p_{t,a}(x) \doteq \Sigma_{|d|<a} q_{t,d} T_d(x)$ where $q_{t,d} \doteq \Pr[d_t = d]$.

We have an error of

$$\begin{aligned}
|x^t - p_{t,a}(x)| &= |\Sigma_{|d|\geq a} q_{t,d} T_d(x)| \\
&\leq \Sigma_{|d|\geq a} q_{t,d} |T_d(x)| \text{ , through triangle inequality} \\
&\leq \Sigma_{|d|\geq a} q_{t,d} \text{ for } x \in [-1,1] \\
&= \Pr[|d_t| \geq a] \\
&\leq 2\exp\left(-a^2/(2t)\right) \\
&\leq \eta \text{ provided } a \in \Omega(\sqrt{t\log(1/\eta)}.
\end{aligned}$$

2.7 Chebyshev Approximation to T^t

We can now use what we have proved to approximate T^t. Symmetric transition matrix T has a full orthonormal basis of eigenvectors $|\psi_i\rangle$, $i \in [N]$: $T|\psi_i\rangle = \lambda_i |\psi_i\rangle$.
Then $T^t |\psi_i\rangle = \lambda_i^t |\psi_i\rangle$ and $p_{t,a}(T)|\psi_i\rangle = p_{t,a}(\lambda_i)|\psi_i\rangle$.

To see how close our approximation, $p_{t,a}(T)$, is, we use;

$$||(T^t - p_{t,a}(T))|\psi_i\rangle||_2 = |\lambda_i^t - p_{t,a}(\lambda_i)| \leq \eta \text{ as } \lambda_i \in [-1,1]$$

We now have a full orthonormal basis of eigenvectors. To find the difference, we use the decomposition:

$$\begin{aligned}
\text{For arbitrary } |\psi\rangle &\doteq \Sigma_{i\in[N]} \alpha_i |\psi_i\rangle \\
||(T^t - p_{t,a}(T))|\psi\rangle||_2^2 &= ||\Sigma_{i\in N} \alpha_i (T^t - p_{t,a}(T))|\psi_i\rangle||_2^2 \\
&= ||\Sigma_{i\in N} \alpha_i (\lambda_i^t - p_{t,a}(\lambda_i))|\psi_i\rangle||_2^2, \text{ because of eigenvalues} \\
&= \Sigma_{i\in N} ||\alpha_i (\lambda_i^t - p_{t,a}(\lambda_i))|\psi_i\rangle||_2^2, \text{ due to Pythagorean Theorem} \\
&\leq \Sigma_{i\in[N]} |\alpha_i|^2 \eta^2 = \eta^2, \text{ as each term is at most } \eta \\
&\therefore ||(T^t - p_{t,a}(T))||_2 \leq \eta
\end{aligned}$$

If you recall, $U^*(SC)^d U$ block encodes $T_d(T)$ for each $d \in \mathbb{N}$. We then have the next part towards proving Lemma 2:

Lemma 4. *For $a \in \Omega(\sqrt{t\log(1/\eta)})$, $\Sigma_{|d|<a} q_{t,d} U^*(SC)^d U$ block encodes some matrix M such that $||T^t - M||_2 \leq \eta$.*

With the block encoding for unitaries, we then need a block encoding for the entire linear combination. This becomes the last component for our fast forwarding proof.

2.8 Linear Combination of Unitaries

Given unitaries U_i on n qubits, for $q_i \in \mathbb{C}$, we need a block encoding of linear combination $L \doteq \Sigma_i q_i U_i$. However, it is difficult to get a block encoding of L as we cannot grow the vectors in our

unitaries by some factor q_i.
Instead, we may also settle for L/q, for small q (q being too large decreases our probability of success) using the unitary (and as a multiplexor) $V : |i\rangle |\psi\rangle \mapsto |i\rangle U_i |\psi\rangle$.

Without loss of generality, we can assume $q_i \in (0, \infty)$ by dropping zero terms and incorporating the phases into the unitaries U_i.

Algorithm We create the superposition $|q\rangle \doteq \frac{1}{\sqrt{\Sigma_i q_i}} \Sigma_i \sqrt{q_i} |i\rangle$

Then we apply the multiplexer $V |q\rangle |\psi\rangle = \frac{1}{\sqrt{\Sigma_i q_i}} \Sigma_i |i\rangle \sqrt{q_i} U_i |\psi\rangle$

We project on that superposition,

$$\langle q| \langle \phi| V |q\rangle |\psi\rangle = \frac{1}{\Sigma_i q_i} \Sigma_i \sqrt{q_i}\sqrt{q_i} \langle \phi| U_i |\psi\rangle \text{, by application of } \langle q| \text{ to the previous multiplexor application}$$
$$= \frac{1}{\Sigma_i q_i} \langle \phi| L |\psi\rangle \text{, by the previous definition of L}$$

Lemma 5. *Let $\tilde{U}$ be a unitary such that $\tilde{U} : |0^m\rangle \mapsto |q\rangle$. Then $\tilde{U}^* V \tilde{U}$ block encodes $L/\Sigma_i q_i$*

In summary, given the block encodings for unitaries, we can find a block encoding for a linear combination of them. If you recall Lemma 2, we now know there is a fast forwarding algorithm in time $O(\sqrt{t\log(1/\eta)}\mathrm{U})$. We can use this to measure how close our approximation to classical walks are after t steps, measuring renormalized $M |\pi\rangle$ to yield distributions close to renormalized $T - t |\pi\rangle$. We have reached a good vertex if we measure π, or a bad superposition if we measure 0. Additionally, random walks for $t = H$ steps from any start distribution has a probability of at least 0.5 of visiting a good vertex.

2.9 Interpolating Walks

However, for Lemma 2, another issue arises. The block encoding is a probabilistic method of realizing T^t, so if $||T^t |pi\rangle||_2$ is small, there is a small probability of success. Additionally, we cannot measure success after each step and thus do not know how many steps are needed for fast forwarding. We handle this by observing the differences between the actual walk and the absorbing walk:

Absorbing walk This walk stops when we reach a good vertex. As we are using a Markov chain with transition matrix T_{abs}, there is no time reversibility and this is not a random walk on a graph as we do not leave the good vertex.

Interpolating walks For some parameter $a \in [0, 1]$, we then observe the difference between the actual random walk T and the absorbing walk using the Markov chain with transition matrix $T(a) \doteq (1 - a) \cdot T + a \cdot T_{abs}$. If $a = 1$, we get the absorbing walk, otherwise we get valid random walks that we can fast forward.

With the following lemma, we then know that we won't get small probabilities when block encoding and reach a good number of steps with good probability.

Lemma 6. *If $\epsilon \doteq \Pr_{v\ \pi}[goodv] < \frac{1}{10}$, then for any integer $\tau \in \Omega(H)$, $E[||P_{good} T(a)^t |B\rangle||_2^2] = \Omega(\frac{1}{\log(\tau)})$ for $a \doteq 1 - \frac{1}{2^r}$ when $r \in_u [O(\log(\tau))]$ and $\tau \in_u [\tau]$*

We can then use the following algorithm to find how many steps of fast forwarding are needed to find a block encoding for Chebyshev approximations of a matrix.

Quantum Walk Search Algorithm First create a superposition $|\pi\rangle$ based on the uniform distribution. Then evaluate the goodness in the ancilla once. If we are successful, we stop here and measure the register and output the result. Otherwise we are in $|B\rangle$. Then pick a and t for τ steps, by Lemma 6. We then run fast forwarding for the random walk with transition matrix $T(a)$ and apply the resulting matrix A to $|B\rangle$. We evaluate the goodness of the ancilla again. If successful, we measure the register and output the result, otherwise we report a failure.

Analysis for $\tau = \Theta(H)$ A single run costs $O(S + \sqrt{H}(U + C))$ and has a success probability of $min(\frac{1}{10}, \Omega(\frac{1}{\log(H)}))$, due to t, which is bound by the hitting time. Using amplitude amplification to boost confidence yields an additional factor of $\sqrt{\log(H)}$ because we only need a quantum walk of roughly $\sqrt{t}$ steps by the Chebyshev polynomial.

Conclusion In summary, we can complete a quantum walk in a quadratically fewer amount of steps when compared to the classical hitting time of a random walk.