Lecture 4: Linear combination of unitaries

Lecturer: Simon Apers

In the previous lecture we encoded a general Hermitian matrix H into a unitary quantum walk operator. We combined this operator with phase estimation to apply a function of H onto a quantum state (e.g., e^{iHt} for Hamiltonian simulation and H^{-1} for linear system solving). The cost scaled with some problem parameter, and was inversely proportional to the precision ε (cost $\sim t/\varepsilon$ for Hamiltonian simulation, $\sim \kappa^2/\varepsilon$ for linear system solving). This linear scaling in the error is intrinsic to the use of phase estimation. In this lecture we see that an alternative technique called *linear combination of unitaries* (LCU) [CW12] allows us to implement a function $f(H)$ more directly and more precisely.

1 Linear combination of unitaries

Consider a Hermitian matrix H with $\|H\|_1 < 1$, and let W be the QW operator based on H. Let $\Pi_0 = |0\rangle\langle 0|$. We saw in the last lecture that

$$(I \otimes \Pi_0) U_\psi^\dagger W^t U_\psi |\chi\rangle |0\rangle = (T_t(H) |\chi\rangle) |0\rangle . \tag{1}$$

I.e., if we project into the right subspace, then a quantum walk effectively applies the t-th Chebyshev polynomial $T_t(H)$ on $|\chi\rangle$. We will use that the Chebyshev polynomials form a basis for the polynomials.

Exercise 1. • *Verify that the Chebyshev polynomials form an orthogonal set w.r.t. the inner product*

$$\langle f, g\rangle = \int_{-1}^{+1} f(x)g(x)\frac{\mathrm{d}x}{\sqrt{1-x^2}}.$$

Hint: use that $T_t(\cos(\theta)) = \cos(t\theta)$*.*

- *Moreover, they form a basis: any degree-d polynomial can be described as a linear combination of the first $d+1$ Chebyshev polynomials, $x^\tau = \sum_{t=0}^{\tau} \alpha_t T_t(x)$, with coefficients*

$$\alpha_t = \begin{cases} \frac{1}{2^\tau}\binom{\tau}{\tau/2} & \text{if } t = 0 \text{ and } \tau = 0 \bmod 2 \\ \frac{1}{2^{\tau-1}}\binom{\tau}{(\tau-t)/2} & \text{if } t > 0 \text{ and } \tau = t \bmod 2 \\ 0 & \text{elsewhere.} \end{cases} \tag{2}$$

Prove this. Hint: use the fact that $\cos(\theta) = (e^{i\theta} + e^{-i\theta})/2$*.*

As a consequence, we can try to implement a general polynomial $f(H) = \sum_t \alpha_t T_t(H)$ to the quantum state $|\chi\rangle$ by implementing a linear combination $\sum_t \alpha_t W^t$ of the quantum walk evolution (1) for different times. However, this no longer corresponds to a unitary operator. The LCU technique allows us to bypass this restriction (again, by mapping the nonunitary dynamics to a subspace of certain unitary dynamics).

The technique uses the "controlled" version of the QW operator (which also shows up in phase estimation). For some integer $\tau \geq 0$, this operator is defined by

$$cW = \sum_{t=0}^{\tau} W^t \otimes |t\rangle \langle t| .$$

You can think of the cost of the controlled operator as essentially τ steps of the QW operator W. Applying this operator to a state $|\phi\rangle |t'\rangle$ gives

$$cW |\phi\rangle |t'\rangle = W^{t'} |\phi\rangle |t'\rangle ,$$

so it applies a number of QW steps given by the second register. Now assume that $\sum_t |\alpha_t| = 1$,[1] and define the "clock" state

$$U_{\mathrm{cl}} |0\rangle = \sum_t \sqrt{\alpha_t} |t\rangle .$$

The LCU technique puts this state into the clock register, applies the conditioned QW operator, and then inverts the clock operation. This yields the state $(I \otimes U_{\mathrm{cl}}^\dagger)\, cW\, (I \otimes U_{\mathrm{cl}}) |\phi\rangle |0\rangle$. Looking at this state in the right subspace reveals what we are interested in.

Exercise 2.

- *Analyze what the (projected) output of the LCU algorithm corresponds to, given by*

$$(I \otimes \Pi_0)(I \otimes U_{\mathrm{cl}}^\dagger)\, cW\, (I \otimes U_{\mathrm{cl}}) |\phi\rangle |0\rangle .$$

- *Now let cW be the controlled quantum walk operator, and let $|\chi\rangle |0\rangle$ be the initial state of the quantum walk. Analyze what the following state corresponds to:*

$$(I \otimes \Pi_0 \otimes \Pi_0)(I \otimes U_{\mathrm{cl}}^\dagger)(U_\psi^\dagger \otimes I)\, cW\, (U_\psi \otimes I)(I \otimes U_{\mathrm{cl}}) |\chi\rangle |0\rangle |0\rangle .$$

2 Matrix powering and quantum fast-forwarding

As an illustration of the LCU technique we consider the problem of matrix powering, where we wish to return a state of the form $H^\tau |\psi\rangle$ (up to normalization) for some integer $\tau \geq 0$. This corresponds to implementing the polynomial $f(x) = x^\tau$.

Exercise 3.

- *Let W be the QW operator associated to H. Explicitly describe the LCU algorithm based on W for constructing the state $H^\tau |\psi\rangle / \|H^\tau |\psi\rangle\|_2$. What is the cost of this algorithm?*
- ***Quantum fast-forwarding.*** *We can implement an approximation of H^τ more efficiently by truncating the expansion $x^\tau = \sum_t \alpha_t T_t(x)$.*
  - *Consider independent and uniformly distributed random variables $X_1, \ldots, X_\tau \in \{+1, -1\}$, and let $Y_\tau = \sum_{k=1}^{\tau} X_k$. Prove that the coefficients in Eq. (2) satisfy $\alpha_t = \Pr(|Y_\tau| = t)$.*
  - *By Hoeffding's theorem we know that $\Pr(|Y_\tau| > r) \leq 2\exp(-r^2/(2\tau))$ for any $r \geq 0$. Use this to prove that there exists $d \in O(\sqrt{\tau \log(1/\varepsilon)})$ such that the degree-d polynomial*

$$h(x) = \sum_{t=0}^{d} \alpha_t T_t(x)$$

satisfies $|h(x) - x^\tau| \leq \varepsilon$ for all $x \in [-1, 1]$.

[1] In fact, a similar argument applies as long as $\sum_t |\alpha_t| \in \Theta(1)$.

– *What is the cost of the LCU algorithm based on the polynomial h for implementing the function H^τ with error ε?*

This technique was first described in [AS19]. It was used in the recent resolution [AGJK20] of a long-standing problem related to quantum walk search: can we find an element from a marked set M in time $\widetilde{O}(\sqrt{HT(M)})$? In previous lectures we proved the weaker bound $\widetilde{O}(1/\sqrt{\delta\pi(M)})$.

3 Hamiltonian simulation

In the case of Hamiltonian simulation, we are given some initial state $|\chi\rangle$, a Hermitian matrix H and a time $t \geq 0$, and we wish to output the state $e^{iH\tau}|\chi\rangle$. This corresponds to implementing the function $f(x) = e^{ix\tau}$. For the case where $\|H\|_1 \leq 1$, we described an algorithm based on quantum walks and phase estimation with complexity $\widetilde{O}(\tau/\varepsilon)$. We will use LCU to improve the error dependency to $\log(1/\varepsilon)$.

We focus on the case where $\|H\|_1 < 1$ and $\tau \leq 1$. To use LCU, we must find a polynomial $h(x) = \sum_t \alpha_t T_t(x)$ such that $\sum_t |\alpha_t| \in \Theta(1)$ and $|h(x) - e^{ix\tau}| \leq \varepsilon$ for $|x| < 1$.

Exercise 4.

- *Find an expansion $e^{ix\tau} = \sum_{t\geq 0} \alpha_t T_t(x)$ using the Jacobi-Anger expansion*

$$e^{i\cos(\theta)\tau} = J_0(\tau) + 2\sum_{k=1}^{+\infty} i^k J_k(\tau)\cos(k\theta).$$

Here the function $J_k(y)$ corresponds to the k-th Bessel function of the first kind.

- *Use that $|J_k(\tau)| \leq \frac{1}{k!2^k}$ for $\tau \leq 1$ to show that $\sum_t |\alpha_t| \in \Theta(1)$.*
- *Show that there exists $d \in O(\log(1/\varepsilon))$ such that $h(x) = \sum_{t=0}^d \alpha_t T_t(x)$ satisfies $|h(x) - e^{ix}| \leq \varepsilon$ for $|x| < 1$.*[2]
- *Describe the LCU algorithm based on h for Hamiltonian simulation with $\|H\|_1 < 1$, $\tau \leq 1$ and error $\varepsilon > 0$. What is its cost?*

Using standard tools, this algorithm can be extended to ε-approximate general Hamiltonians for arbitrary times $\tau \geq 0$ with cost $O(\tau\|H\|_1 \log(1/\varepsilon))$.

References

[AGJK20] Andris Ambainis, András Gilyén, Stacey Jeffery, and Martins Kokainis. Quadratic speedup for finding marked vertices by quantum walks. In *Proceedings of the 52nd ACM Symposium on Theory of Computing (STOC)*, pages 412–424. ACM, 2020. arxiv:1903.07493.

[AS19] Simon Apers and Alain Sarlette. Quantum fast-forwarding Markov chains and property testing. *Quantum Information & Computation*, 19(3&4):181–213, 2019. arXiv:1804.02321.

[CW12] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Information & Computation*, 12(11–12):901–924, 2012.

[2] While harder to prove, you could even choose $d \in O(\log(1/\varepsilon)/\log\log(1/\varepsilon))$.