Lecture 17.3

1

Review QSVT

Today discuss applications of QSVT.

Function Evaluation:

desire to implement $f(x)$

as $f(A)$ & use a polynomial

that approximates $f(A)$.

Key problems: Hamiltonian

simulation & matrix inversion.

1) Hamiltonian simulation:

simulate the time evolution of

a state $|\psi\rangle$ under the

Hamiltonian H.

From Schrödinger equation, we

know that $i\hbar \frac{d}{dt}|\psi(t)\rangle = H|\psi(t)\rangle$

has solution $|\psi(t)\rangle = e^{-iHt/\hbar}$

Set $\hbar = 1$

Assume access to a block encoding of H. This is possible for sparse Hamiltonians & linear combination of unitaries.

Aside: How does linear combination of unitaries work?

Suppose Hamiltonian is of the form

$$H = \sum_i \alpha_i W_i$$

where each W_i is a unitary (think Pauli matrices)

Suppose that we have a unitary $U_{\vec{\alpha}}$ that prepares the state $\sum_i \sqrt{\alpha_i}\,|i\rangle$ as

$$U_{\vec{\alpha}}|0\rangle = \sum_i \sqrt{\alpha_i}\,|i\rangle \Big/ \sqrt{\|\vec{\alpha}\|_1}$$

Suppose also a unitary that performs $\sum_i |i\rangle\langle i| \otimes W_i = \text{select}(W)$

Then H is block encoded in

$$\left(U_{\vec{\alpha}}^\dagger \otimes I\right)\text{select}(W)\left(U_{\vec{\alpha}} \otimes I\right)$$

To see this sandwich this by

$$\langle 0| \otimes I \quad (\cdots)\,|0\rangle \otimes I$$

It gives

$$\frac{1}{\sqrt{\|\vec{\alpha}\|_1}}\left(\sum_i \sqrt{\alpha_i}\,\langle i| \otimes I\right)\left(\sum_{i'} |i'\rangle\langle i'| \otimes W_{i'}\right)\frac{\left(\sum_{i''} \sqrt{\alpha_{i''}}\,|i''\rangle \otimes I\right)}{\sqrt{\|\vec{\alpha}\|_1}}$$

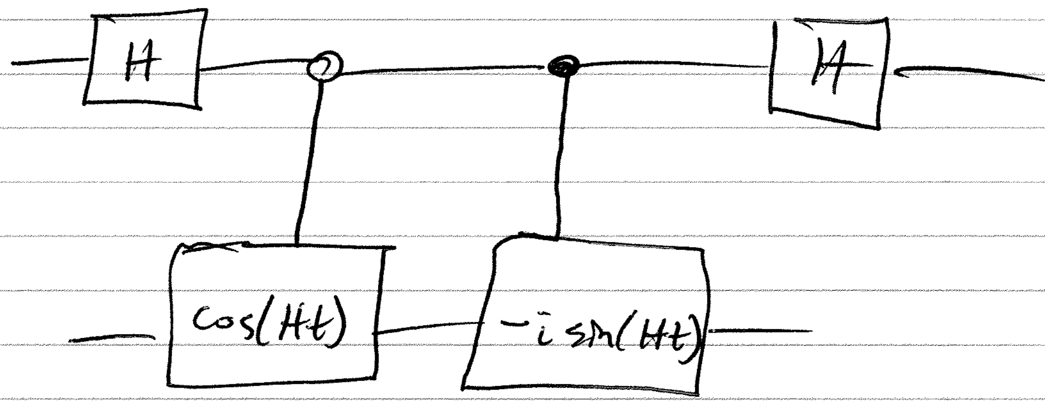
$$= \frac{1}{\|\vec{\alpha}\|_1} \sum_i \alpha_i W_i = \frac{1}{\|\vec{\alpha}\|_1} H$$

normalization factor

How to solve Hamiltonian simulation via QSVT?

- Naively, could try a poly. approx. of e^{-ixt}, but this ~~does~~ function does not have definite parity.
- can instead apply QSVT twice.
  - once w/ even polynomial to approximate $\cos(xt)$ & another w/ an odd polynomial to approximate $\sin(xt)$

can then use this circuit to simulate Hamiltonian:



Why does this work?

Consider that

$$|0\rangle|\psi\rangle \rightarrow \left(|0\rangle|\psi\rangle + |1\rangle|\psi\rangle\right)/\sqrt{2}$$

$$\rightarrow \frac{|0\rangle \cos(Ht)|\psi\rangle + |1\rangle \, {-i\sin(Ht)}|\psi\rangle}{\sqrt{2}}$$

$$= \frac{1}{2}\Big(\big(|+\rangle + |-\rangle\big) \otimes \cos(Ht)|\psi\rangle + \big(|+\rangle - |-\rangle\big) \otimes -i\sin(Ht)|\psi\rangle\Big)$$

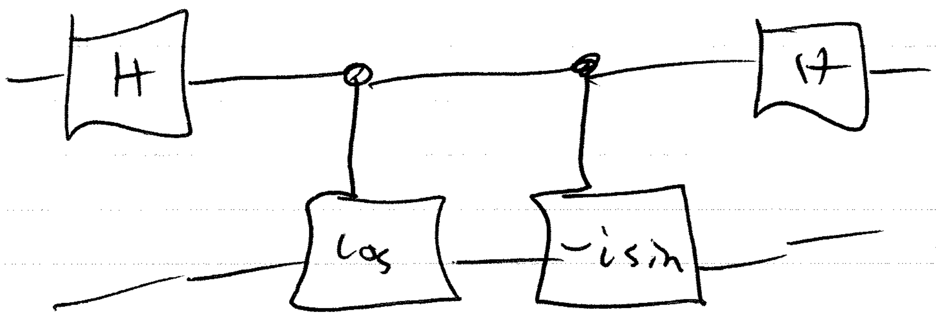
$$= \frac{1}{2}\Big(|+\rangle \otimes \big(\cos(Ht) - i\sin(Ht)\big)|\psi\rangle +$$

$$|-\rangle \otimes \big(\cos(Ht) + i\sin(Ht)\big)|\psi\rangle\Big)$$

$$= \frac{1}{2}\left(|+\rangle \otimes e^{-iHt}|\psi\rangle + |-\rangle \otimes e^{iHt}|\psi\rangle\right) \quad (6)$$

$$\to \frac{1}{2}\left(|0\rangle \otimes e^{-iHt}|\psi\rangle + |1\rangle \otimes e^{iHt}|\psi\rangle\right)$$

Thus, taking



+ sandwiching by

$$\langle 0| \otimes I \;(\dots)\; |0\rangle \otimes I$$

shows that it block encodes

$$e^{-iHt}.$$

In the above, we assumed that

$$\cos^{(SV)}(Ht) - i\sin^{(SV)}(Ht) = e^{-iHt}$$

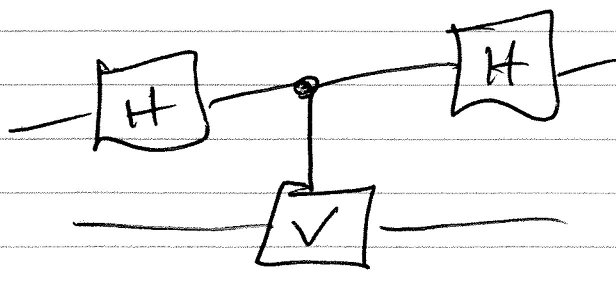
- This is only true if the eigenvalues of H are positive, so that singular values are equal to eigenvalues.

- If this is not the case, then use block encoding of $H/\|H\|_\infty$ to construct block encoding of

$$\frac{1}{2}(H/\|H\|_\infty + I) \text{ which is PSD.}$$

then timeevolve for time $2\|H\|_\infty t$ to get e^{-iHt}

can block encode $\frac{1}{2}\left(\frac{H}{\|H\|_\infty} + I\right)$ using



where V block encodes $H/\|H\|_\infty$

returning to approximation of $\cos(xt)$ + $\sin(xt)$, can use the Jacobi-Anger expansion

$$\cos(xt) = J_0(t) + 2\sum_{k=1}^{\infty}(-1)^k J_{2k}(t)\, T_{2k}(x)$$

$$\sin(xt) = 2\sum_{k=0}^{\infty}(-1)^k J_{2k+1}(t)\, T_{2k+1}(x)$$

where $J_i(x)$ is a Bessel function of order i + $T_i(x)$ is a Chebyshev polynomial of order i.

can get ε-approximations of $\cos(xt)$ + $\sin(xt)$ by truncating expressions at large index k.

Skipping details, this algorithm achieves ε-approx. + queries U

$$O\left(\|H\|t + \frac{\log 1/\varepsilon}{\log\left(e + \frac{\log(1/\varepsilon)}{\|H\|t}\right)}\right)$$

times. has state-of-the-art scaling in t & ε.

Matrix inversion using QSVT

Given access to a square matrix A, one wishes to prepare A^{-1}

- Given $N \times N$ square matrix w/ SVD $A = W \Sigma V^{\dagger}$.
- Suppose that singular values of A obey $\sigma_i \in [\kappa^{-1}, 1]$ for condition number $\kappa \geq 1$

(if not, can rescale A)

$\Rightarrow$ inverse of A exists & is given by $A^{-1} = V \Sigma^{-1} W^{+}$

Since $A^{+} = V \Sigma W^{+}$, this is the same as

$$A^{-1} = f^{(SV)}(A^{+}) \text{ where}$$

$$f(x) = 1/x \, .$$

Goal is then to find an odd polynomial $P(x)$ such that $f(x) = 1/x$ over $[\kappa^{-1}, 1]$ & then use QSVT to construct $P^{(SV)}(A^{+})$

- Since the polynomial for QSVT requires that $|P(x)| \leq 1 \quad \forall x \in [-1,1]$ we cannot use $P(x) \approx 1/x$

- We instead seek an approximation to $\frac{1}{2\kappa x}$ on range $[-1,-\kappa^{-1}] \cup [\kappa^{-1},1]$.

- This will invert each singular value & is bounded by $1/2$ in this range.

- procedure outputs an approximation of $\frac{1}{2\kappa} A^{-1}$

- Then desire an $\frac{\varepsilon}{2\kappa}$ approximation to $\frac{1}{2\kappa} A^{-1}$ in order to get ε approx. of A^{-1}

Then seek a polynomial that is an $\frac{\varepsilon}{2k}$ approx. to $\frac{1}{2kx}$

- Construction is somewhat complicated but now known.

- It has degree $d = O(k \log (k/\varepsilon))$ + this is thus the complexity.