Quantum and quantum-inspired linear algebra: Some problem set solutions

Ewin Tang Christopher Kang

July 24–28, 2023

These are solutions to some of the problem sets in a 5-lecture mini-course I taught at the 2023 PCMI graduate summer school. These were written up during the week and haven't been edited since, so: beware of errors!

Contents

Problem Set 1: The block-encoding **2**

Problem Set 2: The QSVT **5**

Problem Set 3: Polynomial approximation **7**

Problem Set 4: Dequantizing QSVT **10**

Problem Set 5: The power of classical **12**

Problem Set 1: The block-encoding

Problem 1.1 (Taking tensor products of block-encodings)**.** Let U and V be Q-block encodings of A and B, respectively. Show how to get a Q-block-encoding of $A \otimes B$.

Solution. $U \otimes V$ is a block-encoding of $A \otimes B$. □

Problem 1.2 (Extensibility properties)**.** Prove Corollary 1.11 of the lecture notes. Specifically, show that the two extensibility properties allow us to convert a Q-block encoding of A to a nQ-block encoding of $p^{(\mathrm{SV})}(A)$.

Solution. We can construct a kQ-block encoding of $m_k^{(\mathrm{SV})}(A)$, for $m_k(x) = x^k$. The problem here is that the naïve approach – producing x^n and then adding with x^{n-1} – would require $\mathcal{O}(n^2 Q)$ complexity.

Instead, via Horner's rule, we may rewrite the polynomial in the following form:

$$a_0 + x(a_1 + x(a_2 + ... + x(a_{n-1} + xa_n))) \tag{1}$$

Precisely the sum of products of polynomials. It can be shown that the coefficients can be structured carefully so that they never exceed 1. □

Solution. [Angus Lowe's solution] Consider the following preparation unitaries:

$$\mathrm{PREP}\,|0\rangle = \sum_k \sqrt{\lambda_k}\,|k\rangle \tag{2}$$

$$\mathrm{SELECT} = \sum_{k=0}^{n} |k\rangle\langle k| \otimes A^k \tag{3}$$

Then, the application of $\mathrm{PREP}^\dagger \cdot \mathrm{SELECT} \cdot \mathrm{PREP}$ precisely implements a desired block encoding with λ_k chosen appropriately. This is a version of linear combinations of unitaries seen in Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity. SELECT can be implemented efficiently via using a binary encoding in the ancilla and using $\log_2 n$ controlled-A^{2^j} gates. □

Problem 1.3 (Extensibility properties do not suffice)**.** Let $p(x) = \sum_{k=0}^{n} a_k x^k$ be a polynomial whose coefficients satisfy $\sum |a_k| \leq 1$. Show that $p(x)$ cannot approximate $\sin(100x)$ for any choice of n. That is, show that there is some $x \in [-1, 1]$ such that

$$|p(x) - \sin(100x)| \geq 0.01.$$

We will see in Lecture 3 that $\sin(100x)$ *can in fact be approximated by a low-degree polynomial; it's just that this class of polynomials doesn't suffice.*

Solution. The key idea is straightforward: we want to show that any polynomial $p(x)$ has derivative $p'(x)$ that differs significantly from $\frac{d}{dx}\sin(100x)$ and use this to produce a contradiction.

First, consider $x = -\frac{\pi}{200}, x = \frac{\pi}{200}$. Then, $\sin(100x) = \pm 1$ at those points. Thus, by the Mean Value Theorem, p must at some point attain a derivative exceeding the following value:

$$\frac{0.99 - -0.99}{\frac{\pi}{200} - -\frac{\pi}{200}} = \frac{200 \cdot 0.99}{\pi} \geq 50 \tag{4}$$

Now, consider the maximum derivative attainable by the polynomial. Set $p(x) = \sum_{k=0}^{n} a_k x^k$ with $\sum |a_k| = 1$. Then,

$$|p'(x)| \leq \left| \sum_{k=1}^{d} a_k \cdot k x^{k-1} \right| \tag{5}$$

$$\leq \sum_{k=1}^{n} |a_k| k |x|^{k-1} \tag{6}$$

$$\leq \sum_{k=1}^{n} k |x|^{k-1} \tag{7}$$

Numerics can show that this function lies far below 50 for $x \in [\pm \frac{\pi}{200}]$.

Thus, for the polynomial to observe our requirements, it must attain a derivative of at least 50 at some point. However, on this interval, it has derivative far less. Thus, we have obtained a contradiction and p does not exist. □

Solution. [Zachary Stier's solution] Suppose we have a polynomial $p(x) = a_0 + \sum_{k=1}^{n} a_k x^k$. Then, because $|p(0)| \leq \frac{1}{100}$ by our constraint, we need $|a_0| \leq \frac{1}{100}$. Then, observe that, on $x \in [0, 1/2]$:

$$p(x) \leq |a_0| + \sum_{k=1}^{n} |a_k| |x|^k \tag{8}$$

$$\leq \frac{1}{100} + \frac{1}{2} \tag{9}$$

Thus, the maximum attainable value of $p(x)$ is $\frac{51}{100}$. However, $x = \frac{\pi}{200}$ would mean $\sin(100x) = 1$, so $p(x)$ and $\sin(100x)$ differ from a quantity much greater than 0.01, a contradiction. □

Problem 1.4 (Oblivious amplitude amplification)**.** QSVT is a unifying technique which includes many major quantum algorithms, including amplitude amplification [MRTC21]. In this problem, we show that Oblivious Amplitude Amplification (OAA), as described in [BCCKS17, Lemma 3.6], can be written in our block-encoding framework.

Identify the block-encoding within the aforementioned unitary. What polynomial would effect the same transformation as described in [BCCKS17, Lemma 3.6]?

Solution. The state preparation unitary mentioned in [BCCKS17] performs the following transformation:

$$U |0\rangle^{\mu} |\psi\rangle = \sin\theta |0\rangle^{\mu} V |\psi\rangle + |\Phi^{\perp}\rangle \tag{10}$$

Where $|\Phi^{\perp}\rangle$ is an orthogonal component such that $\langle 0|^{\mu} \otimes I |\Phi^{\perp}\rangle = 0$. Then, U is a block-encoding of $\sin\theta V$, i.e.:

$$U = \begin{bmatrix} \sin\theta V & \cdot \\ \cdot & \cdot \end{bmatrix} \tag{11}$$

In fact, the net unitary we would like to implement is the following:

$$S^{\ell} U = \begin{bmatrix} \sin(2\ell+1)\theta V & \cdot \\ \cdot & \cdot \end{bmatrix} \tag{12}$$

Thus, we see that $S^\ell U$ actually implements a polynomial (Chebyshev polynomial) taking $\sin\theta$ to $\sin(2\ell+1)$. However, we need not use Chebyshev polynomials if we may tolerate a different construction. In particular, $\sin\theta$ will typically be known, so implementing any polynomial taking the specific value of $\sin\theta$ to $\sin(2\ell+1)\theta$ will suffice. □

Remark 1.1. See [Ral20] for more information on how to get block-encodings of density matrices and observables, and how to use this to estimate physical quantities like expectations of Gibbs states. See [BCCKS17] for further discussion of Hamiltonian simulation, placing it in the context of the more general problem of understanding the "fractional query model", "discrete query model", and "continuous query model". See [LC19] (the original paper) or [GSLW19] for a more thorough explanation of the Hamiltonian simulation algorithm.

Problem Set 2: The QSVT

Problem 2.1 (When will my reflection show who I am inside?)**.** QSVT achieves polynomials by interspersing phase operators with signal rotation operators. However, these rotation operators may look different in the literature. Consider two potential operators, $W(x), R(x)$, with the following matrix forms:

$$W(x) = \begin{pmatrix} x & i\sqrt{1-x^2} \\ i\sqrt{1-x^2} & x \end{pmatrix} \qquad R(x) = \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \tag{13}$$

Where W is the rotation operator while R is the reflection operator. We can define two different kinds of QSP, $\mathbf{QSP}_W(\Phi, x)$ and $\mathbf{QSP}_R(\Phi, x)$ for these two different operators. For example,

$$\mathbf{QSP}_W(\Phi, x) \coloneqq \Big(\prod_{j=1}^{n} e^{\mathrm{i}\phi_j\sigma_z} W(x)\Big) e^{\mathrm{i}\phi_0\sigma_z}.$$

Suppose we have some series of phases $\Phi = (\phi_0, \ldots, \phi_n)$ such that $\mathbf{QSP}_W(\Phi, x)$ forms a desired polynomial $p(x)$. Can we find a Φ' such that $\mathbf{QSP}_R(\Phi', x)$ performs the same polynomial? If so, find a formula for Φ' in terms of Φ; if not, prove why.

Solution. (From [MRTC21, Appendix A.2]) We can notice that

$$\begin{aligned} W(x) &= \begin{pmatrix} 1 & \\ & i \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} 1 & \\ & i \end{pmatrix} \\ &= e^{\mathrm{i}\pi/2} \begin{pmatrix} e^{-\mathrm{i}\pi/4} & \\ & e^{\mathrm{i}\pi/4} \end{pmatrix} \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix} \begin{pmatrix} e^{-\mathrm{i}\pi/4} & \\ & e^{\mathrm{i}\pi/4} \end{pmatrix} \\ &= e^{\mathrm{i}\pi/2} e^{-\mathrm{i}\frac{\pi}{4}\sigma_z} R(x) e^{-\mathrm{i}\frac{\pi}{4}\sigma_z} \end{aligned}$$

So, if $\Phi = (\phi_0, \phi_1, \ldots, \phi_n)$ is the phase sequence for W, then $\Phi - (\pi/4, \pi/2, \pi/2, \ldots, \pi/2, \pi/4 - d\pi/2)$ is the phase sequence for R. □

Problem 2.2 (Perfectly balanced, as all things should be)**.** The Chebyshev polynomials of the first and second kind are functions such that, for all $z \in \mathbb{C}$,

$$\begin{aligned} T_n(\tfrac{1}{2}(z + z^{-1})) &= \tfrac{1}{2}(z^n + z^{-n}) \\ U_n(\tfrac{1}{2}(z + z^{-1})) &= (z^{n+1} - z^{-(n+1)})/(z - z^{-1}) \end{aligned}$$

Prove that T_n and U_n are polynomials. Then, prove that

$$T_n(x)^2 + (1 - x^2)U_{n-1}(x)^2 = 1. \tag{14}$$

Just a little more and we have a proof that these can be used in QSP/QSVT!

Problem 2.3 (They're the same picture!)**.** Return to [BCCKS17, Lemma 3.6]. What are the angles of the phase operators? What are the polynomials that are being computed with these phase operators? (A recursive definition is fine.)

Solution. The key idea here is that the phase unitaries applied take the form $2\Pi - I$ for some projector Π. Thus, this is equivalent to performing a rotation of $\phi = \frac{\pi}{2}$. They are creating a Chebyshev polynomial taking $\sin\theta \mapsto \sin(2\ell + 1)\theta$. □

Problem 2.4 (Block-encodings for any matrix)**.** Given a matrix $A \in \mathbb{C}^{d\times d}$ such that $\|A\| \leq 1$, show there exists a unitary $U \in \mathbb{C}^{2d\times 2d}$ such that U is a block-encoding of A:

$$U = \begin{pmatrix} A & \cdot \\ \cdot & \cdot \end{pmatrix}.$$

Prove that $2d$ is tight, i.e., there is some matrix A such that any unitary with A as a submatrix must be size at least $2d \times 2d$. *Note: this is true for non-square A as well, but the argument might get more annoying.*

Solution. Consider the singular value decomposition $A = VDW^\dagger$. Then

$$\begin{pmatrix} V & \\ & I \end{pmatrix}\begin{pmatrix} D & \sqrt{1-D^2} \\ \sqrt{1-D^2} & -D \end{pmatrix}\begin{pmatrix} W^\dagger & \\ & I \end{pmatrix}$$

is a product of unitary matrices, where the top-left block is A. *For A non-square, this works via mimicking the structure CS decomposition.* If A is the zero matrix, then we need U to be size $2d \times 2d$; smaller matrices containing A must have linearly dependent columns.

(Alternative solution from [AA11, Lemma 29]) Since $A^\dagger A$ is a positive semi-definite matrix such that $\|A^\dagger A\| \leq 1$, then $I - A^\dagger A$ is also positive semi-definite, so it has a Hermitian square root $I - A^\dagger A = B^2 = B^\dagger B$. Since $A^\dagger A + B^\dagger B = I$,

$$\begin{pmatrix} A \\ B \end{pmatrix}^\dagger \begin{pmatrix} A \\ B \end{pmatrix} = I,$$

so this stacked $2d \times d$ matrix has orthonormal columns. Consequently, we can complete it to a $2d \times 2d$ unitary matrix. □

Problem 2.5 (It's just a phase)**.** In our QSVT algorithm, we needed to apply gates of the form $e^{\mathrm{i}\phi(2\Pi - I)}$, where $\Pi = (|0\rangle^{\otimes a}\langle 0|^{\otimes a}) \otimes I$. How do you implement these?

Solution. A single ancilla coupled with Π-controlled nots are sufficient.

A Π-controlled not takes the following form:

$$C_\Pi NOT = \Pi \otimes X + (I - \Pi) \otimes I \tag{15}$$

So that $C_\Pi NOT e^{i\phi Z} C_\Pi NOT$ when applied to an ancilla of $|0\rangle$ is precisely the required circuit. (See [MRTC21] for more circuits). □

Problem Set 3: Polynomial approximation

Problem 3.1 (Polynomial approximation of monomials)**.** First, compute the Chebyshev coefficients of the monomial $m^{(n)}(x) = x^n$. (Doing this via $T_k(\frac{1}{2}(z + z^{-1})) = \frac{1}{2}(z^n + z^{-n})$ formulation may be easiest.) How small can k be such that the Chebyshev truncation $m_k^{(n)}$ a good approximation of $m^{(n)}$:

$$\|m^{(n)} - m_k^{(n)}\|_{[-1,1]} \leq \varepsilon?$$

Solution. Substituting in $x = \frac{1}{2}(z + z^{-1})$, we get that

$$x^n = \frac{1}{2^n}(z + z^{-1})^n \tag{16}$$

$$= \frac{1}{2^n} \sum_{k=0}^{n} \binom{n}{k} z^{k-(n-k)} \tag{17}$$

$$= \frac{1}{2^n} \sum_{k=0}^{n} \binom{n}{k} z^{2k-n} \tag{18}$$

There's some annoyance involving parity. If n is odd, then

$$= \frac{1}{2^n} \Big(\sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{k} z^{2k-n} + \sum_{k=\lfloor n/2 \rfloor + 1}^{n} \binom{n}{k} z^{2k-n} \Big) \tag{19}$$

$$= \frac{1}{2^n} \sum_{k=\lfloor n/2 \rfloor + 1}^{n} \binom{n}{k} 2T_{2k-n}(x) \tag{20}$$

$$= \frac{1}{2^{n-1}} \sum_{k=0}^{\lfloor n/2 \rfloor} \binom{n}{k} T_{n-2k}(x) \tag{21}$$

If n is even, then we get a constant term.

$$= \frac{1}{2^n} \Big(\binom{n}{n/2} + \sum_{k=0}^{n/2-1} \binom{n}{k} z^{2k-n} + \sum_{k=n/2+1}^{n} \binom{n}{k} z^{2k-n} \Big) \tag{22}$$

$$= \frac{1}{2^n} \Big(\binom{n}{n/2} + \sum_{k=n/2+1}^{n} \binom{n}{k} 2T_{2k-n}(x) \Big) \tag{23}$$

$$= \frac{1}{2^n} \Big(\binom{n}{n/2} + \sum_{k=0}^{n/2-1} \binom{n}{k} 2T_{n-2k}(x) \Big) \tag{24}$$

Roughly, the Chebyshev coefficient corresponding to a_ℓ is $2^{1-n}\binom{n}{(n-\ell)/2}$, up to parity issues. So, for the truncation $m_{2\ell}^{(n)}$, the tail bound is (again, morally),

$$m_{2\ell}^{(n)} = \sum_{k \geq \ell} \binom{n}{n/2 - \ell} = \Pr[\mathrm{Bin}(n, 1/2) \leq n/2 - \ell]. \tag{25}$$

By a Chernoff bound, it suffices to choose $\ell = \mathcal{O}(\sqrt{n \log(1/\varepsilon)})$. See [SV14] for a more careful version of this argument. □

Problem 3.2 (Chebyshev interpolation [Tre19])**.** The *Chebyshev interpolant* of a function f, denoted p_n, is the unique degree-n polynomial such that $p_n(x_j) = f(x_j)$ for all $x_j = \cos(j\pi/n)$, $j = 0, 1, \ldots, n$. Prove that[1]

$$\|f(x) - p_n(x)\|_{[-1,1]} \leq 2\sum_{\ell \geq n} |a_\ell|.$$

Hint: when is $T_k(x_j) = T_\ell(x_j)$ for all points $\{x_j\}$?

Solution. We will build the Chebyshev interpolant of the function and identify the maximal error associated with this interpolant.

First, a detour: observe that the following Chebyshev polynomials have the same value for $x = \frac{z+z^{-1}}{2}$ for $z^{2\nu n} = 1$ for any integer ν.

$$T_m, T_{2n-m}, T_{2n+m}, T_{4n-m}, T_{4n+m}, \ldots \tag{26}$$

This follows from the observation that $T_k(\frac{z+z^{-1}}{2}) = \frac{z^k+z^{-k}}{2}$ ([Tre19, Theorem 4.1]).

Now, consider the Chebyshev series associated with f:

$$f(x) = \sum_{k=0}^{\infty} a_k T_k(x) \tag{27}$$

Then, to produce an interpolant, we need to enforce the condition that $p_n(x_j) = f(x_j)$. This can be done by recognizing that $T_k(x), T_j(x)$ coincide for specific values of k, j depending on x. Then, at these values, you could rewrite the function as follows:

$$f(x_j) = \sum_{k=0}^{n} c_k \sum_{m \in S_k} T_m(x_j) \tag{28}$$

Where S_k are the set of Chebyshev polynomials taking the same value at x_j. We've already defined this set above, and can find an explicit form for c_k as follows ([Tre19, Theorem 4.2]):

$$c_0 = a_0 + a_{2n} + a_{4n} + \ldots \tag{29}$$
$$c_n = a_n + a_{3n} + \ldots \tag{30}$$
$$c_k = a_k + (a_{k+2n} + a_{-k+2n}) + (a_{k+4n} + a_{-k+4n}) + \ldots \tag{31}$$

Therefore, the error in a nth degree truncation can be seen as follows:

$$f(x) - p_n(x) = \sum_{k=0}^{\infty} a_k T_k(x) - \sum_{k=0}^{n} c_k T_k(x) \tag{32}$$
$$= \sum_{k=n+1}^{\infty} a_k (T_k(x) - T_m(x)) \tag{33}$$
$$\leq \sum_{k=n+1}^{\infty} 2|a_k| \tag{34}$$

[1] Recall that our approximation results used that $\|f(x) - f_n(x)\|_{[-1,1]} \leq \sum_{\ell \geq n} |a_\ell|$. So, Chebyshev interpolants p_n give the same results as Chebyshev truncations f_n, up to a constant factor. Interpolants have the advantage of being computable in $n+1$ function evaluations.

For $m = m(k,n)$. The second step follows because each of the terms between $0 \le k \le n$ cancel directly (each c_k contains an a_k within it), and the terms $k \ge n+1$ occur because the coefficient of a_m within some c_k is still unmodified, just associated with a lower order Chebyshev polynomial $T_{m(k,n)}$ which coincides with T_k at the provided values of x_j. □

Problem 3.3 (Jackson theorems, [Tre19])**.** Let $f : [-1,1] \to \mathbb{R}$ be absolutely continuous and suppose f is of bounded variation, meaning that $\int_{-1}^{1} |f'(x)| \mathrm{d}x \le V$. Then show that the Chebyshev coefficients of f satisfy

$$|a_k| \le \frac{2V}{\pi k}.$$

Solution. See [Tre19, Theorem 7.1]; it's integration by parts on the integral equation for a_k. □

Problem 3.4 (Optimal polynomial approximations; upper and lower bounds)**.** Consider a function $f : [-1,1] \to \mathbb{R}$ with a Chebyshev expansion $f(x) = \sum_{k \ge 0} a_k T_k(x)$. Prove that

$$\Big(\frac{1}{2}\sum_{k=n+1}^{\infty} a_k^2\Big)^{\frac{1}{2}} \le \min_{\substack{p \in \mathbb{R}[x] \\ \deg p = n}} \|f(x) - p(x)\|_{[-1,1]} \le \sum_{k=n+1}^{\infty} |a_k|$$

For what kind of Chebyshev coefficient decay is this characterization tight up to constants?

Solution. We follow [AA22, Proposition 2.2], but get an improved bound. The upper bound follows by taking $p(x) = f_n(x)$. The lower bound follows by bounding the max by the integral. Let $p(x) = \sum_{k=0}^{n} b_k T_k(x)$ be a degree-n polynomial. Take $b_k = 0$ for all $k > n$. Then

$$\begin{aligned}
\|f(x) - p(x)\|_{[-1,1]} &\ge \frac{1}{2\pi}\int_{-\pi}^{\pi} (f(\cos(\theta)) - p(\cos(\theta)))^2 \mathrm{d}\theta \\
&\ge \frac{1}{2\pi}\int_{-\pi}^{\pi} \Big(\sum_{k=0}^{\infty} (a_k - b_k) T_k(\cos(\theta))\Big)^2 \mathrm{d}\theta \\
&\ge \frac{1}{2\pi}\int_{-\pi}^{\pi} \Big(\sum_{k=0}^{\infty} (a_k - b_k) \cos(k\theta)\Big)^2 \mathrm{d}\theta
\end{aligned}$$

This is expression is the squared norm of the function $f(x) - p(x)$ under the inner product where $\cos(k\theta)$'s are orthogonal. So, this gives us the sum of squares of the coefficients.

$$\begin{aligned}
&= \frac{1}{2\pi}\sum_{k=0}^{\infty}\sum_{\ell=0}^{\infty} (a_k - b_k)(a_\ell - b_\ell) \int_{-\pi}^{\pi} \cos(k\theta)\cos(\ell\theta)\mathrm{d}\theta \\
&= \frac{1}{2\pi}\sum_{k=0}^{\infty} (a_k - b_k)^2 \pi \\
&\ge \frac{1}{2}\sum_{k=n+1}^{\infty} (a_k - b_k)^2 \\
&= \frac{1}{2}\sum_{k=n+1}^{\infty} a_k^2.
\end{aligned}$$

□

Problem Set 4: Dequantizing QSVT

Before you begin, recall the definitions of sampling and query access for vectors and matrices $(\mathrm{SQ}(v), \mathrm{SQ}(A))$ and *oversampling* and query access $(\mathrm{SQ}_\phi(v), \mathrm{SQ}_\phi(A))$ [CGLLTW22, Definition 3.2]. Below, time complexities are in the word RAM model: basically, assume that reading input numbers, and performing operations on those numbers, cost $\mathcal{O}(1)$.

Problem 4.1 (Errare humanum est...)**.** Suppose we have $\mathrm{SQ}_{\phi_u}(u), \mathrm{SQ}_{\phi_v}(v)$ for vectors u, v. Show that we have $\mathrm{SQ}_\phi(A)$ for $A := uv^\dagger$ and $\phi = \phi_u\phi_v$ with cost $\mathbf{sq}_\phi(A) = \mathbf{sq}_{\phi_u}(u) + \mathbf{sq}_{\phi_v}(v)$.

Solution. The sampling algorithm is straightforward:

1. Query an index i_u of u via $\mathrm{SQ}_{\phi_u}(u)$
2. Query an index i_v of v via $\mathrm{SQ}_{\phi_v}(v)$
3. Return $u(i_u) \cdot v(i_v)^\dagger$

This clearly has complexity in $\mathrm{SQ}_{\phi_u}(u) + \mathrm{SQ}_{\phi_v}(v)$.. How can we show that this is $\phi = \phi_u\phi_v$ oversampling? Observe that any row of the oversampled matrix $\tilde{A}$ takes the form $\tilde{A}(i, \cdot) = \tilde{u}(i)\tilde{v}^\dagger$ so that $\tilde{A} = \tilde{u}\tilde{v}^\dagger$. Thus, $\|\tilde{A}\|_F^2 = \|\tilde{u}\|^2\|\tilde{v}\|^2 = \phi_u\phi_v\|u\|^2\|v\|^2 = \phi_u\phi_v\|A\|_F^2$ □

Problem 4.2 (...sed perseverare (non?) diabolicum.)**.** Suppose we are given a matrix $A \in \mathbb{C}^{m\times m}$ with at most s non-zero entries per row, and suppose all entries are bounded by c. We are given this matrix as a list of non-zero entries $(i, j, A(i, j))$. Show how to perform $\mathrm{SQ}_\phi(A)$ queries for $\phi = c^2\frac{sm}{\|A\|_{\mathrm{F}}^2}$ with $\mathbf{sq}_\phi(A) = s$.[2] This means that we can run "dequantized" algorithms on sparse matrices with condition number κ; why doesn't this imply that QSVT admits no exponential speedup for sparse matrices?

Solution. For example, for $\mathrm{SQ}(\tilde{a})$ we can set $\tilde{a}(i) := c\sqrt{s}$, and for $\tilde{A}(i, \cdot)$ we use the vector with entries c at the non-zeros of $A(i, \cdot)$ (potentially adding some "dummy" zero locations to have exactly s non-zeroes).

Note that similar sparse-access assumptions are often seen in the QML and Hamiltonian simulation literature [HHL09]. Also, if A is not much smaller than we expect, then ϕ can be independent of dimension. For example, if A has exactly s non-zero entries per row and $|A(i, j)| \geq c'$ for non-zero entries, then $\phi \leq (c/c')^2$. □

Problem 4.3 (The alias method [Vos91])**.** Let $p = (p_1, \ldots, p_m)$ be a set of probabilities, so $p_i \geq 0$ and $\sum p_i = 1$. Suppose also that all of the p_i's are described in binary with $\mathcal{O}(1)$ bits.

1. Suppose we are given a uniformly random number $x \in [0, 1]$ as a stream of random bits. Show how to sample $i \in [m]$ such that $\Pr[\text{sample } \ell] = p_\ell$ in $\mathcal{O}(m)$ operations.
2. Suppose we are given $p = (p_1, \ldots, p_m)$ in the following form: we get a list of m probability distributions $d_1, \ldots, d_m$ such that $\frac{1}{m}(d_1 + \cdots + d_m) = p$ and every d_i is supported on at most two outcomes. Show that we can sample $i \in [m]$ according to p in $\mathcal{O}(1)$ time.

[2] Hint: We immediately have query access to A. What's a good upper bound that's easy to sample from?

3. Prove that we can convert any distribution p into the form described above. Prove that we can do this in $\mathcal{O}(m)$ time.[3]

[3]This implies that, if we get time to pre-process, we can get a data structure such that we can respond to $\mathrm{SQ}(v)$ queries in $\mathcal{O}(1)$ time (in the word RAM access model).

Problem Set 5: The power of classical

For this problem set, you'll need the following result about importance sampling sketches, strengthening Lemma 5.9 from the lecture notes.

Lemma 5.2 (Approximating matrix multiplication to spectral norm error [RV07, Theorem 3.1])**.** *Suppose we are given $A \in \mathbb{R}^{m\times n}, \varepsilon > 0, \delta \in [0,1]$, and $S \in \mathbb{R}^{r\times n}$ a ϕ-oversampled importance sampling sketch of A. Then*

$$\Pr\left[\|A^\dagger S^\dagger SA - A^\dagger A\| \lesssim \sqrt{\frac{\phi^2 \log r \log 1/\delta}{r}}\|A\|\|A\|_{\mathrm{F}}\right] > 1-\delta.$$

Problem 5.1 ([CGLLTW22, Lemma 4.9])**.** Given $\mathrm{SQ}(A) \in \mathbb{C}^{m\times n}$ and $\varepsilon \in (0,1]$, we can form importance sampling sketches $S \in \mathbb{R}^{r\times m}$ and $T^\dagger \in \mathbb{R}^{c\times n}$ in $\mathcal{O}(rc\,\mathbf{sq}(A))$ time. Let σ_i and $\hat{\sigma}_i$ denote the singular values of A and SAT, respectively (where $\hat{\sigma}_i = 0$ for $i > \min(r,c)$). How big does our sketch ($r \times c$) need to be for the following property to hold with probability 0.9?

$$\left(\sum_{i=1}^{\min(m,n)} (\hat{\sigma}_i^2 - \sigma_i^2)^2\right)^{1/2} \leq \varepsilon\|A\|_{\mathrm{F}}^2. \tag{$\star$}$$

Problem 5.2 ([CGLLTW22, Corollary 6.12])**.** We now show that the previous problem implies a dequantization of QPCA [LMR14]. Given a matrix $\mathrm{SQ}(X) \in \mathbb{C}^{m\times n}$ such that $X^\dagger X$ has top k eigenvalues $\{\lambda_i\}_{i=1}^k$, along with a lower bound ν such that $\lambda_1, \ldots, \lambda_k \geq \nu$, compute eigenvalue estimates $\{\hat{\lambda}_i\}_{i=1}^k$ such that, with probability 0.9,

$$\sum_{i=1}^k |\hat{\lambda}_i - \lambda_i| \leq \varepsilon \operatorname{tr}(X^\dagger X). \tag{35}$$

What is the runtime of this classical algorithm?

Bonus: how would you design a quantum algorithm to solve this task? Suppose we are given a state prep unitary that prepares a purification of $\rho = X^\dagger X$ (i.e. the vectorized version of X), which implies both the ability to prepare ρ and a 1-block encoding of ρ.

Problem 5.3 ([Van11; GL22])**.** Suppose we are given SQ access to the vector corresponding to the n-qubit state $|\psi\rangle$ and a description of $H = \frac{1}{s}\sum_{i=1}^s \lambda_a E_a$, where $\lambda_a \in [-1,1]$ and E_a are Pauli matrices. Show how to estimate $\langle\psi| H^k |\psi\rangle$ to ε error in $\mathrm{poly}(n, s^k, 1/\varepsilon)$ time.

Bonus: prove you can still perform the above estimate if $|\psi\rangle$ is given as a matrix product state with polynomial bond dimension, meaning that, for some $2n\,\mathrm{poly}(n)\times\mathrm{poly}(n)$ matrices $A_i[0], A_i[1]$, $\psi_{b_1\cdots b_n} = \operatorname{tr}(A_1[b_1]\cdots A_n[b_n])$. Here, $b_1\cdots b_n$ are bits.

References

[AA11] Scott Aaronson and Alex Arkhipov. "The computational complexity of linear optics". In: *Proceedings of the forty-third annual ACM symposium on Theory of computing*. ACM, June 2011. DOI: 10.1145/1993636.1993682. arXiv: 1011.3245 [quant-ph] (page 6).

[AA22] Amol Aggarwal and Josh Alman. "Optimal-degree polynomial approximations for exponentials and gaussian kernel density estimation". In: *37th Computational Complexity Conference, CCC 2022*. Vol. 234. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 22:1–22:23. DOI: 10.4230/LIPIcs.CCC.2022.22 (page 9).

[BCCKS17] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. "Exponential improvement in precision for simulating sparse Hamiltonians". In: *Forum of Mathematics, Sigma* 5 (2017), e8. DOI: 10.1017/fms.2017.2. arXiv: 1312.1414 [quant-ph] (pages 3–5).

[CGLLTW22] Nai-Hui Chia, András Pal Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. "Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning". In: *Journal of the ACM* 69.5 (Oct. 2022), pp. 1–72. DOI: 10.1145/3549524. arXiv: 1910.06151 [cs.DS] (pages 10, 12).

[GL22] Sevag Gharibian and François Le Gall. "Dequantizing the quantum singular value transformation: hardness and applications to quantum chemistry and the quantum pcp conjecture". In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2022. Rome, Italy: Association for Computing Machinery, 2022, pp. 19–32. ISBN: 9781450392648. DOI: 10.1145/3519935.3519991 (page 12).

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. "Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics". In: *Proceedings of the 51st ACM Symposium on the Theory of Computing (STOC)*. ACM, June 2019, pp. 193–204. DOI: 10.1145/3313276.3316366. arXiv: 1806.01838 (page 4).

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum algorithm for linear systems of equations". In: *Physical Review Letters* 103 (15 Oct. 2009), p. 150502. DOI: 10.1103/PhysRevLett.103.150502 (page 10).

[LC19] Guang Hao Low and Isaac L. Chuang. "Hamiltonian simulation by qubitization". In: *Quantum* 3 (July 2019), p. 163. DOI: 10.22331/q-2019-07-12-163 (page 4).

[LMR14] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. "Quantum principal component analysis". In: *Nature Physics* 10.9 (July 2014), pp. 631–633. DOI: 10.1038/nphys3029. arXiv: 1307.0401 [quant-ph] (page 12).

[MRTC21] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. "Grand unification of quantum algorithms". In: *PRX Quantum* 2 (4 Dec. 2021), p. 040203. DOI: 10.1103/PRXQuantum.2.040203. arXiv: 2105.02859 [quant-ph] (pages 3, 5, 6).

[Ral20] Patrick Rall. "Quantum algorithms for estimating physical quantities using block encodings". In: *Physical Review A* 102.2 (Aug. 2020), p. 022408. DOI: 10.1103/physreva.102.022408. arXiv: 2004.06832 [quant-ph] (page 4).

[RV07] Mark Rudelson and Roman Vershynin. "Sampling from large matrices: an approach through geometric functional analysis". In: *Journal of the ACM* 54.4 (July 2007), 21–es. ISSN: 0004-5411. DOI: 10.1145/1255443.1255449. URL: https://doi.org/10.1145/1255443.1255449 (page 12).

[SV14] Sushant Sachdeva and Nisheeth K. Vishnoi. "Faster algorithms via approximation theory". In: *Foundations and Trends in Theoretical Computer Science* 9.2 (2014), pp. 125–210. ISSN: 1551-305X. DOI: 10.1561/0400000065 (page 7).

[Tre19] Lloyd N. Trefethen. *Approximation theory and approximation practice, extended edition*. Extended edition [of 3012510]. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2019, pp. xi+363. ISBN: 978-1-611975-93-2. DOI: 10.1137/1.9781611975949 (pages 8, 9).

[Van11] Maarten Van den Nest. "Simulating quantum computers with probabilistic methods". In: *Quantum Information and Computation* 11.9&10 (Sept. 2011), pp. 784–812. ISSN: 1533-7146. DOI: 10.26421/qic11.9-10-5. arXiv: 0911.1624 [quant-ph] (page 12).

[Vos91] Michael D. Vose. "A linear algorithm for generating random numbers with a given distribution". In: *IEEE Transactions on Software Engineering* 17.9 (1991), pp. 972–975. DOI: 10.1109/32.92917 (page 10).