5 Quantum-inspired algorithms: sketching and beyond

Recall our definitions of oversampling and query access.

Definition 4.4. We have *ϕ-oversampling and query access* to a vector $v \in \mathbb{C}^n$, $\mathrm{SQ}_\phi(v)$, if:

1. we can query for entries of v, $\mathrm{Q}(v)$, and;
2. we have sampling and query access to an "entry-wise upper bound" vector $\tilde{v}$, $\mathrm{SQ}(\tilde{v})$, where $\|\tilde{v}\|^2 = \phi\|v\|^2$ and $|\tilde{v}(i)| \geq |v(i)|$ for all indices $i \in [n]$.

Let $\mathbf{sq}_\phi(v)$ denote the time cost of any query.

Intuitively speaking, estimators that use $\mathcal{D}_v$ can also use $\mathcal{D}_{\tilde{v}}$ via rejection sampling at the expense of a factor ϕ increase in the number of utilized samples. From this observation we can prove that oversampling access implies an approximate version of the usual sampling access:

Lemma 4.5. *Suppose we are given $\mathrm{SQ}_\phi(v)$ and some $\delta \in (0,1]$. Denote $\overline{\mathbf{sq}}(v) := \phi\,\mathbf{sq}_\phi(v)\log\frac{1}{\delta}$. We can sample from $\mathcal{D}_v$ with probability $\geq 1-\delta$ in $\mathcal{O}(\overline{\mathbf{sq}}(v))$ time. We can also estimate $\|v\|$ to ν multiplicative error for $\nu \in (0,1]$ with probability $\geq 1-\delta$ in $\mathcal{O}(\frac{1}{\nu^2}\,\overline{\mathbf{sq}}(v))$ time.*

Lemma 4.6 (Linear combinations, Proposition 4.3 of [Tan19])**.** *Given $\mathrm{SQ}_{\varphi_t}(v_t) \in \mathbb{C}^n$ and $\lambda_t \in \mathbb{C}$ for all $t \in [\tau]$, we have $\mathrm{SQ}_\phi(\sum_{t=1}^{\tau}\lambda_t v_t)$ for $\phi = \tau\frac{\sum\varphi_t\|\lambda_t v_t\|^2}{\|\sum\lambda_t v_t\|^2}$ and $\mathbf{sq}_\phi(\sum\lambda_t v_t) = \sum_{t=1}^{\tau}\mathbf{sq}(v_t)$ (after paying $\mathcal{O}(\sum_{t=1}^{\tau}\mathbf{sq}_{\varphi_t}(v_t))$ one-time pre-processing cost to query for norms).*

5.1 Oversampling and query access to matrices

Definition 5.1 (Oversampling and query access to a matrix)**.** For a matrix $A \in \mathbb{C}^{m\times n}$, we have $\mathrm{SQ}(A)$ if we have $\mathrm{SQ}(A(i,\cdot))$ for all $i \in [m]$ and $\mathrm{SQ}(a)$ for $a \in \mathbb{R}^m$ the vector of row norms ($a(i) := \|A(i,\cdot)\|$).

We have $\mathrm{SQ}_\phi(A)$ if we have $\mathrm{Q}(A)$ and $\mathrm{SQ}(\tilde{A})$ for $\tilde{A} \in \mathbb{C}^{m\times n}$ satisfying $\|\tilde{A}\|_{\mathrm{F}}^2 = \phi\|A\|_{\mathrm{F}}^2$ and $|\tilde{A}(i,j)|^2 \geq |A(i,j)|^2$ for all $(i,j) \in [m]\times[n]$.

Let $\mathbf{sq}_\phi(A)$ denote the cost of every query. We omit subscripts if $\phi = 1$.

Lemma 5.2. *Given vectors $\mathrm{SQ}_{\varphi_u}(u) \in \mathbb{C}^m$ and $\mathrm{SQ}_{\varphi_v}(v) \in \mathbb{C}^n$, we have $\mathrm{SQ}_\phi(A)$ for their outer product $A := uv^\dagger$ with $\phi = \varphi_u\varphi_v$ and $\mathbf{sq}_\phi(A) = \mathbf{sq}_{\varphi_u}(u) + \mathbf{sq}_{\varphi_v}(v)$.*

Proof. We can query an entry $A(i,j) = u(i)v(j)^\dagger$ by querying once from u and v. Our choice of upper bound is $\tilde{A} = \tilde{u}\tilde{v}^\dagger$. Clearly, this is an upper bound on $uv^\dagger$ and $\|\tilde{A}\|_{\mathrm{F}}^2 = \|\tilde{u}\|^2\|\tilde{v}\|^2 = \varphi_u\varphi_v\|A\|_{\mathrm{F}}^2$. We have $\mathrm{SQ}(\tilde{A})$ in the following manner: $\tilde{A}(i,\cdot) = \tilde{u}(i)\tilde{v}^\dagger$, so we have $\mathrm{SQ}(\tilde{A}(i,\cdot))$ from $\mathrm{SQ}(\tilde{v})$ after querying for $\tilde{u}(i)$, and $\tilde{a} = \|\tilde{v}\|^2\tilde{u}$, so we have $\mathrm{SQ}(\tilde{a})$ from $\mathrm{SQ}(\tilde{u})$ after querying for $\|\tilde{v}\|$. □

Using the same ideas as in Lemma 4.6, we can extend sampling and query access of input matrices to linear combinations of those matrices.

Lemma 5.3. *Given* $\mathrm{SQ}_{\varphi^{(t)}}(A^{(t)}) \in \mathbb{C}^{m\times n}$ *and* $\lambda_t \in \mathbb{C}$ *for all* $t \in [\tau]$*, we have* $\mathrm{SQ}_\phi(A) \in \mathbb{C}^{m\times n}$ *for* $A := \sum_{t=1}^{\tau} \lambda_t A^{(t)}$ *with* $\phi = \tau \frac{\sum_{t=1}^{\tau} \varphi^{(t)} \|\lambda_t A^{(t)}\|_{\mathrm{F}}^2}{\|A\|_{\mathrm{F}}^2}$ *and* $\mathbf{sq}_\phi(A) = \sum_{t=1}^{\tau} \mathbf{sq}_{\varphi^{(t)}}(A^{(t)})$ *(after paying* $\mathcal{O}(\sum_{t=1}^{\tau} \mathbf{sq}_{\varphi^{(t)}}(A^{(t)}))$ *one-time pre-processing cost).*

5.2 Sketching to estimate matrix products

We now introduce the workhorse of our algorithms: the matrix sketch. Using sampling and query access, we can generate these sketches efficiently, and these allow one to reduce the dimensionality of a problem, up to some approximation. Most of the results presented in this section are known in the classical sketching literature.

Definition 5.4. For a distribution $p \in \mathbb{R}^m$, we say that a matrix $S \in \mathbb{R}^{s\times m}$ is *sampled according to* p if each row of S is independently chosen to be $e_i/\sqrt{s \cdot p(i)}$ with probability $p(i)$, where e_i is the vector that is one in the ith position and zero elsewhere.

We call S an *importance sampling sketch* for $A \in \mathbb{C}^{m\times n}$ if it is sampled according to A's row norms a, and we call S a *ϕ-oversampled importance sampling sketch* if it is sampled according to the bounding row norms from $\mathrm{SQ}_\phi(A)$, $\tilde{a}$ (or, more generally, from a ϕ-oversampled importance sampling distribution of a).

One should think of S as a description of how to sketch A down to SA. In the standard algorithm setting, computing an importance sampling sketch requires reading all of A, since we need to sample from $\mathcal{D}_a$. If we have $\mathrm{SQ}_\phi(A)$, though, we can efficiently create a ϕ-oversampling sketch S in $\mathcal{O}(s\,\mathbf{sq}_\phi(A))$ time: for each row of S, we pull a sample from $\tilde{a}$, and then compute $\sqrt{\tilde{a}(i)}$. After finding this sketch S, we have an implicit description of SA: it is a normalized multiset of rows of A, so we can describe it with the row indices and corresponding normalization, $(i_1, c_1), \dots, (i_s, c_s)$.

Further, we can chain sketches using the lemma below, which shows that from $\mathrm{SQ}_\phi(A)$, we have $\mathrm{SQ}_{\leq 2\phi}((SA)^\dagger)$, under a mild assumption on the size of the sketch S. This can be used to find a sketch $T^\dagger$ of $(SA)^\dagger$. The resulting expression SAT is small enough that we can compute functions of it in time independent of dimension, and so will be key for us. When we discuss sketching A down to SAT, we are referring to the below lemma for the method of sampling T.

Lemma 5.5. *Consider* $\mathrm{SQ}_\varphi(A) \in \mathbb{C}^{m\times n}$ *and* $S \in \mathbb{R}^{r\times m}$ *sampled according to* $\tilde{a}$*, described as pairs* $(i_1, c_1), \dots, (i_r, c_r)$*. If* $r \geq 2\varphi^2 \ln \frac{2}{\delta}$*, then with probability* $\geq 1-\delta$*, we have* $\mathrm{SQ}_\phi(SA)$ *and* $\mathrm{SQ}_\phi((SA)^\dagger)$ *for some* ϕ *satisfying* $\phi \leq 2\varphi$*. If* $\varphi = 1$*, then for all* r*, we have* $\mathrm{SQ}(SA)$ *and* $\mathrm{SQ}((SA)^\dagger)$*.*

The runtimes for $\mathrm{SQ}_\phi(SA)$ *are* $\mathbf{sq}(SA) = \mathbf{sq}(A)$*. The runtimes for* $\mathrm{SQ}_\phi((SA)^\dagger)$ *are* $\mathbf{sq}((SA)^\dagger) = r\,\mathbf{sq}(A)$*.*

Proof. We will only prove this for $\varphi = 1$. We have $\mathrm{SQ}(SA)$. Because the rows of SA are rescaled rows of A, we have SQ access to them from SQ access to A. Because $\|SA\|_{\mathrm{F}}^2 = \|A\|_{\mathrm{F}}^2$ and $\|[SA](i,\cdot)\|^2 = \|A\|_{\mathrm{F}}^2/r$, we have SQ access to the vector of row norms of SA (pulling samples simply by pulling samples from the uniform distribution).

We have $\mathrm{SQ}((SA)^\dagger)$. (This proof is similar to one from [FKV04].) Since the rows of $(SA)^\dagger$ are length r, we can respond to SQ queries to them by reading all entries of the row and performing some linear-time computation. $\|(SA)^\dagger\|_{\mathrm{F}}^2 = \|A\|_{\mathrm{F}}^2$, so we can respond to a norm query by querying the norm of A. Finally, we can sample according to the row norms of $(SA)^\dagger$ by first querying an index $i \in [r]$ uniformly at random, then outputting the index $j \in [n]$ sampled from $[SA](i, \cdot)$ (which we can sample from because it is a row of A). The distribution of the samples output by this procedure is correct: the probability of outputting j is

$$\frac{1}{r}\sum_{i=1}^{r} \frac{|[SA](i,j)|^2}{\|[SA](i,\cdot)\|^2} = \sum_{i=1}^{r} \frac{|[SA](i,j)|^2}{\|SA\|_{\mathrm{F}}^2} = \frac{\|[SA](\cdot,j)\|^2}{\|SA\|_{\mathrm{F}}^2}. \qquad \square$$

We will show below that SA can be used in place of A in matrix products. We begin with a fundamental observation: given sampling and query access to a matrix A, we can approximate the matrix product $A^\dagger B$ by a sum of rank-one outer products. We formalize this with two variance bounds, which we can use together with Chebyshev's inequality.

Lemma 5.6 (Asymmetric matrix multiplication to Frobenius norm error, **[DKM06]**)**.** *Consider $X \in \mathbb{C}^{m\times n}, Y \in \mathbb{C}^{m\times p}$, and take $S \in \mathbb{R}^{r\times m}$ to be sampled according to $p \in \mathbb{R}^m$ a ϕ-oversampled importance sampling distribution from X or Y. Then,*

$$\mathrm{E}[\|X^\dagger S^\dagger SY - X^\dagger Y\|_{\mathrm{F}}^2] \leq \frac{\phi}{r}\|X\|_{\mathrm{F}}^2\|Y\|_{\mathrm{F}}^2 \quad \text{and} \quad \mathrm{E}\Big[\sum_{i=1}^{r}\|[SX](i,\cdot)\|^2\|[SY](i,\cdot)\|^2\Big] \leq \frac{\phi}{r}\|X\|_{\mathrm{F}}^2\|Y\|_{\mathrm{F}}^2.$$

Proof. To show the first equation, we use that $\mathrm{E}[\|X^\dagger S^\dagger SY - X^\dagger Y\|_{\mathrm{F}}^2]$ is a sum of variances, one for each entry (i,j), since $\mathrm{E}[X^\dagger S^\dagger SY - XY]$ is zero in every entry. Furthermore, for every entry (i,j), the matrix expression is the sum of r independent, mean-zero terms, one for each row of S:

$$[X^\dagger S^\dagger SY - XY](i,j) = \sum_{s=1}^{r}\Big([SX](s,i)^\dagger[SY](s,j) - \frac{1}{r}[X^\dagger Y](i,j)\Big).$$

So, we can use standard properties of variances[1] to conclude that

$$\begin{aligned}\mathrm{E}[\|X^\dagger S^\dagger SY - X^\dagger Y\|_{\mathrm{F}}^2] &= r\cdot\mathrm{E}[\|[SX](1,\cdot)^\dagger[SY](1,\cdot) - \tfrac{1}{r}X^\dagger Y\|_{\mathrm{F}}^2] \leq r\cdot\mathrm{E}[\|[SX](1,\cdot)^\dagger[SY](1,\cdot)\|_{\mathrm{F}}^2] \\ &= r\sum_{i=1}^{m} p(i)\frac{\|X(i,\cdot)^\dagger Y(i,\cdot)\|_{\mathrm{F}}^2}{r^2 p(i)^2} = \frac{1}{r}\sum_{i=1}^{m}\frac{\|X(i,\cdot)\|^2\|Y(i,\cdot)\|^2}{p(i)} \leq \frac{\phi}{r}\|X\|_{\mathrm{F}}^2\|Y\|_{\mathrm{F}}^2.\end{aligned}$$

The second other inequality follows by the same computation:

$$\mathrm{E}\Big[\sum_{i=1}^{r}\|[SX](i,\cdot)\|^2\|[SY](i,\cdot)\|^2\Big] = r\cdot\mathrm{E}[\|[SX](1,\cdot)\|^2\|[SY](1,\cdot)\|^2] \leq \frac{\phi}{s}\|X\|_{\mathrm{F}}^2\|Y\|_{\mathrm{F}}^2. \qquad \square$$

The above result shows that, given $\mathrm{SQ}(X)$, $X^\dagger Y$ can be approximated by a sketch with constant failure probability. If we have $\mathrm{SQ}(X)$ *and* $\mathrm{SQ}(Y)$, we can make the failure probability exponential small.

Lemma 5.7 (Approximating matrix multiplication to Frobenius norm error; corollary of **[DKM06]**)**.** *Consider $X \in \mathbb{C}^{m\times n}, Y \in \mathbb{C}^{m\times p}$, and take $S \in \mathbb{R}^{r\times m}$ to be sampled according to $q := \frac{q_1+q_2}{2}$, where $q_1, q_2 \in \mathbb{R}^m$ are ϕ_1, ϕ_2-oversampled importance sampling distributions*

[1] See the proof of **??** in **??** for this kind of computation done with more detail.

from x, y, the vector of row norms for X, Y, respectively. Then S is a $2\phi_1, 2\phi_2$-oversampled importance sampling sketch of X, Y, respectively. Further,

$$\Pr\left[\|X^\dagger S^\dagger SY - X^\dagger Y\|_{\mathrm{F}} < \sqrt{\frac{8\phi_1\phi_2 \log 2/\delta}{r}}\|X\|_{\mathrm{F}}\|Y\|_{\mathrm{F}}\right] > 1 - \delta.$$

5.3 Proving extensibility from sketching

Remark 5.8. Lemma 5.7 implies that, given $\mathrm{SQ}_{\phi_1}(X)$ and $\mathrm{SQ}_{\phi_2}(Y)$, we can get $\mathrm{SQ}_{\phi}(M)$ for M a sufficiently good approximation to $X^\dagger Y$, with $\phi \leq \phi_1\phi_2\frac{\|X\|_{\mathrm{F}}^2\|Y\|_{\mathrm{F}}^2}{\|M\|_{\mathrm{F}}^2}$. This is an approximate closure property for oversampling and query access under matrix products.

Given the above types of accesses, we can compute the sketch S necessary for Lemma 5.7 by taking $p = \mathcal{D}_{\tilde{x}}$ and $q = \mathcal{D}_{\tilde{y}}$), thereby finding a desired $M := X^\dagger S^\dagger SY$. We can compute entries of M with only r queries each to X and Y, so all we need is to get $\mathrm{SQ}(\tilde{M})$ for $\tilde{M}$ the appropriate bound. We choose $|\tilde{M}(i,j)|^2 := r\sum_{\ell=1}^r |[S\tilde{X}](\ell,i)^\dagger [S\tilde{Y}](\ell,j)|^2$; showing that we have $\mathrm{SQ}(M)$ follows from the proofs of Lemmas 5.2 and 5.3, since M is simply a linear combination of outer products of rows of $\tilde{X}$ with rows of $\tilde{Y}$. Finally, this bound has the appropriate norm. Notating the rows sampled by the sketch as $s_1, \ldots, s_r$, we have

$$\begin{aligned}\|\tilde{M}\|_{\mathrm{F}}^2 = r\sum_{\ell=1}^r \|[S\tilde{X}](\ell,\cdot)\|^2\|[S\tilde{Y}](\ell,\cdot)\|^2 &= r\sum_{\ell=1}^r \frac{\|\tilde{X}(s_\ell,\cdot)\|^2\|\tilde{Y}(s_\ell,\cdot)\|^2}{r^2(\frac{\|\tilde{X}(s_\ell,\cdot)\|^2}{2\|\tilde{X}\|_{\mathrm{F}}^2} + \frac{\|\tilde{Y}(s_\ell,\cdot)\|^2}{2\|\tilde{Y}\|_{\mathrm{F}}^2})^2} \\ &\leq \sum_{\ell=1}^r \frac{\|\tilde{X}(s_\ell,\cdot)\|^2\|\tilde{Y}(s_\ell,\cdot)\|^2}{r(\frac{\|\tilde{X}(s_\ell,\cdot)\|\|\tilde{Y}(s_\ell,\cdot)\|}{\|\tilde{X}\|_{\mathrm{F}}\|\tilde{Y}\|_{\mathrm{F}}})^2} = \|\tilde{X}\|_{\mathrm{F}}^2\|\tilde{Y}\|_{\mathrm{F}}^2 = \phi_1\phi_2\|X\|_{\mathrm{F}}^2\|Y\|_{\mathrm{F}}^2.\end{aligned}$$

Assuming A and b are in appropriate data structures in QRAM, we can implement a $\|A\|_{\mathrm{F}}$-block-encoding of A and prepare copies of $|b\rangle$ efficiently, so we can quantumly produce a sample from Ab in $\mathcal{O}(\frac{\|A\|_{\mathrm{F}}^2\|b\|^2}{\|Ab\|^2})$ time. We can dequantize this algorithm! Classically, under identical assumptions, we can produce a sample from a v such that $\|v - Ab\| \leq \varepsilon\|Ab\|$ in $\mathcal{O}(\frac{\|A\|_{\mathrm{F}}^4\|b\|^4}{\varepsilon^2\|Ab\|^4})$ time, only polynomially slower than quantum.

We note here that a dependence on error ε appears here where it does not in the quantum setting. However, this is not a realizable quantum speedup (except possibly for sampling tasks) since the output is a quantum state: estimating some statistic of the quantum state requires incurring a polynomial dependence on ε. For example, if the goal is to estimate $|\langle v|Ab\rangle|^2$, where v is a given vector, then this can be done with $1/\varepsilon^2$ invocations of a swap test (or $1/\varepsilon$ if one uses amplitude amplification). More generally, distinguishing a state from one ε-far in trace distance requires $\Omega(1/\varepsilon)$ additional overhead, even when given an oracle efficiently preparing that state, so estimating quantities to this sensitivity requires polynomial dependence on ε.

To see this, we first consider a simple case: where b is a constant-sized vector, so $n = \mathcal{O}(1)$. Then we simply wish to sample from a linear combination of columns of A, since $Ab = \sum_{t=1}^n b(t)A(\cdot,t)$. If A is in the QRAM data structure (i.e. storing $A^\dagger$ in **??**), then this means its columns are in the vector QRAM data structures (**??**), so classically we have sampling and query access to the columns of A, $\mathrm{SQ}(A(\cdot,t))$ for all $t \in [n]$. This implies we have sampling and query access to Ab, up to some overhead.

Given access to a constant number of vectors $\mathrm{SQ}(A(\cdot,1)),\dots,\mathrm{SQ}(A(\cdot,n))$, we have access to linear combinations $\mathrm{SQ}_\phi(Ab)$ with $\phi = n\frac{\sum_{t=1}^n |b(t)|^2\|A(\cdot,t)\|^2}{\|Ab\|^2} \leq \frac{n\|A\|_{\mathrm{F}}^2\|b\|^2}{\|Ab\|^2}$ and $\mathbf{sq}_\phi(Ab) = \mathcal{O}(n)$ (Lemma 4.6; the inequality follows from Cauchy-Schwarz). Finally, from $\mathrm{SQ}_\phi(Ab)$ we can perform approximate versions of all the queries of $\mathrm{SQ}(Ab)$ with a factor ϕ of overhead (Lemma 4.5). This is possible with rejection sampling: given $\mathrm{SQ}_\phi(v)$, pull a sample i from $\tilde{v}$; accept it with probability $|v(i)|^2/|\tilde{v}(i)|^2$, and restart otherwise; the output will be a sample from v. In particular, we can sample from Ab in $\phi n = \mathcal{O}(n^2\frac{\|A\|_{\mathrm{F}}^2\|b\|^2}{\|Ab\|^2})$ time in expectation, which is good when $n = \mathcal{O}(1)$.

Now, consider when n is too large to iterate over in our linear combination of vectors. In this setting, we can use the *approximate matrix product* property of importance sampling to reduce the number of vectors under consideration. Consider pulling a sample $s \in [n]$ where we sample i with probability $p(i)$. Then $\frac{1}{p(s)}b(s)A(\cdot,s)$, a rescaled random column of A, has expectation $\sum_i b(i)A(\cdot,i) = Ab$. If the sampling distribution is chosen to be $p(i) = \frac{|b(i)|^2}{\|b\|^2}$, an importance sample from $\mathrm{SQ}(b)$, then a variance computation shows that the average of $\tau = \Theta(\frac{\|A\|_{\mathrm{F}}^2}{\varepsilon^2\|A\|^2})$ copies of this random vector is $\varepsilon\|A\|\|b\|$-close to Ab with probability ≥ 0.9 (Lemma 5.6). This average, which we denote v, is now a linear combination of only τ columns of A, each of which we have sampling and query access. So, we can use the closure properties mentioned before to get $\mathrm{SQ}_\phi(v)$ for $\phi = \mathcal{O}(\frac{\|A\|_{\mathrm{F}}^2\|b\|^2}{\|v\|^2})$ and $\mathbf{sq}_\phi(v) = \mathcal{O}(\tau)$, and a sample from v in $\phi\tau = \mathcal{O}(\frac{\|A\|_{\mathrm{F}}^4\|b\|^2}{\varepsilon^2\|A\|^2\|v\|^2})$ time in expectation. Rescaling ε by a factor of $\frac{\|Ab\|}{\|A\|\|b\|}$ gives the result stated above.

5.4 General singular value transformation

So far, we have shown extensibility properties of SQ access like that of the block-encoding. However, as we saw in a problem set, this does not suffice to implement all polynomials. We will now discuss the broad approach for producing $p^{(\mathrm{SV})}(A)b$ from $\mathrm{SQ}(A)$ and $\mathrm{SQ}(b)$.

First, we give an improved version of the above results on approximation of matrix products, from scaling with $\|X\|_{\mathrm{F}}\|Y\|_{\mathrm{F}}^2$ to $\|X\|_{\mathrm{F}}^2 + \|Y\|_{\mathrm{F}}^2$.

Theorem 5.9 (Asymmetric Approximate Matrix Multiplication [BT23])**.** *Given matrices $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{n\times d}$, consider S sampled according to $p_i \geq \frac{1}{2\phi}(\frac{\|A_{*,i}\|^2}{\|A\|_{\mathrm{F}}^2} + \frac{\|B_{*,i}\|^2}{\|B\|_{\mathrm{F}}^2})$ for some $\phi \geq 1$. Let $\mathrm{sr} = \frac{\|A\|_F^2}{\|A\|^2} + \frac{\|B\|_F^2}{\|B\|^2}$. Then, with probability at least $1-\delta$,*

$$\|ASS^\dagger B^\dagger - AB^\dagger\| \leq \sqrt{\frac{2}{s}\log\Big(\frac{\mathrm{sr}}{\delta}\Big)\phi(\|A\|_F^2\|B\|^2 + \|A\|^2\|B\|_F^2)} + \frac{1}{s}\log\Big(\frac{\mathrm{sr}}{\delta}\Big)\phi\|A\|_{\mathrm{F}}\|B\|_{\mathrm{F}}.$$

This means that a sketch of size $s = \widetilde{\Theta}(\frac{\phi\,\mathrm{sr}}{\varepsilon^2}\log\frac{1}{\delta})$ suffices to get an approximation of $\varepsilon\|A\|\|B\|$.

Lemma 5.10 (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*

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

Recall our goal of simulating QSVT: given a matrix $\mathrm{SQ}(A) \in \mathbb{C}^{m\times n}$, a vector $\mathrm{SQ}(b) \in \mathbb{C}^n$, and a polynomial $p : [-1,1] \to \mathbb{R}$, compute a description of a vector y such that $\|y - p(A)b\| \leq \varepsilon\|p\|_{[-1,1]}\|b\|$. Specifically, we aim for our algorithm to run in $\mathrm{poly}(\|A\|_{\mathrm{F}}, \frac{1}{\varepsilon}, d)$ time, and our description to be some sparse vector x such that $y = Ax$, since this allows us to get $\mathrm{SQ}_\phi(y)$.

Given a degree-d polynomial p given in terms of its Chebyshev coefficients a_ℓ (i.e. $p(x) = \sum_{\ell=0}^{d} a_\ell T_\ell(x)$, where $T_\ell(x)$ is the degree ℓ Chebyshev polynomial) and a value $x \in [-1,1]$, the Clenshaw recurrence computes $p(x)$. Concretely, our recurrence for p odd (so that $a_\ell = 0$ for ℓ even), is the following.

$$\begin{aligned} q_{(d-1)/2} &= q_{(d+1)/2} = 0 \\ q_k &= 2(2x^2-1)q_{k+1} - q_{k+2} + 2a_{2k+1}x \\ p(x) &= \tfrac{1}{2}(q_0 - q_1) \end{aligned}$$

The scalar recurrences we discuss lift to computing matrix polynomials in a natural way:

$$\begin{aligned} u_{(d-1)/2} &= u_{(d+1)/2} = 0 \\ u_k &= 2(2AA^\dagger - I)u_{k+1} - u_{k+2} + 2a_{2k+1}Ab \\ p(A)b &= \tfrac{1}{2}(u_0 - u_1) \end{aligned}$$

Each iteration (to get u_k from u_{k+1} and u_{k+2}) can be performed in $\mathcal{O}(\mathrm{nnz}(A))$ arithmetic operations, so this can be used to compute $p(A)b$ in $\mathcal{O}(d\,\mathrm{nnz}(A))$ operations. We would like to do this approximately in time independent of $\mathrm{nnz}(A)$ and n. We begin by sketching down our matrix and vector: we show that it suffices to maintain a sparse description of u_k of the form $u_k = Av_k$ where v_k is sparse. In particular, we produce sketches $S \in \mathbb{C}^{n\times s}$ and $T \in \mathbb{C}^{t\times m}$ such that

1. $\|AS(AS)^\dagger - AA^\dagger\| \leq \varepsilon$,
2. $\|ASS^\dagger b - Ab\| \leq \varepsilon\|b\|$,
3. $\|TAS(TAS)^\dagger - AS(AS)^\dagger\| \leq \varepsilon$

In the pre-processing phase, we can produce these sketches of size $s, t = \widetilde{\mathcal{O}}(\frac{\|A\|_{\mathrm{F}}^2}{\varepsilon^2}\log\frac{1}{\delta})$, and then compute TAS. If the input is given in the quantum-inspired access model of *oversampling and query access*, this can even be done in time independent of dimension. All of these guarantees follow from Theorem 5.9, which shows ℓ_2^2 sampling gives an asymmetric approximate matrix product property (in operator norm). We do not need this generalization (prior "symmetric" results suffice), but we use it for convenience.

Using these guarantees we can sketch the iterates as follows:

$$\begin{aligned} u_k &= 2(2AA^\dagger - I)u_{k+1} - u_{k+2} + 2a_{2k+1}Ab \\ &= 4AA^\dagger A v_{k+1} - 2Av_{k+1} - Av_{k+2} + 2a_{2k+1}Ab \\ &\approx AS[4(TAS)^\dagger(TAS)v_{k+1} - 2v_{k+1} - v_{k+2} + 2a_{2k+1}S^\dagger b]. \end{aligned} \tag{1}$$

Therefore, we can interpret Clenshaw iteration as the recursion on the dimension-independent term $v_k \approx 4(TAS)^\dagger(TAS)v_{k+1} - 2v_{k+1} - v_{k+2} + 2a_{2k+1}S^\dagger b$, and then applying AS on the left to lift it back to m dimensional space. As desired, we can perform the iteration to produce v_k in $\mathcal{O}(st) = \widetilde{\mathcal{O}}(\frac{\|A\|_{\mathrm{F}}^4}{\varepsilon^4}\log^2\frac{1}{\delta})$ time, which is independent of dimension, at the cost of incurring $\mathcal{O}(\varepsilon(\|v_{k+1}\| + \|v_{k+2}\| + a_{2k+1}\|b\|))$ error. To bound the effect of these per-iteration errors on the final output, we need a stability analysis of the Clenshaw recurrence. We can prove that this gives a $d^3\varepsilon\|p\|_{[-1,1]}\|b\|$ scaling on the final bound, so we rescale ε by a factor of d^2 to get a final runtime of

$$d\frac{\|A\|_{\mathrm{F}}^4}{(\varepsilon/d^3)^4}\log^2\frac{1}{\delta} = d^{13}\frac{\|A\|_{\mathrm{F}}^4}{\varepsilon^4}\log^2\frac{1}{\delta}.$$

If we allow linear-time pre-processing, this can be improved further by a factor of ε^2/d^2 [BT23].

References

[BT23] Ainesh Bakshi and Ewin Tang. *An improved classical singular value transformation for quantum machine learning*. Mar. 2, 2023. arXiv: 2303.01492 [quant-ph] (pages 5, 7).

[FKV04] Alan Frieze, Ravi Kannan, and Santosh Vempala. "Fast Monte-Carlo algorithms for finding low-rank approximations". In: *Journal of the ACM* 51.6 (Nov. 2004), pp. 1025–1041. DOI: 10.1145/1039488.1039494 (page 2).

[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 5).

[Tan19] Ewin Tang. "A quantum-inspired classical algorithm for recommendation systems". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing - STOC 2019*. ACM Press, 2019, pp. 217–228. DOI: 10.1145/3313276.3316310. arXiv: 1807.04271 [cs.IR] (page 1).