IAS/PCMI Graduate Summer School 2023

The computational complexity of near-term quantum experiments

Lecturer: Bill Fefferman *TA: Soumik Ghosh*

1 Problem Set

Problem 1: Recall in the lecture I introduced two problems that I called "classical" and "quantum" exact sum. The "classical" exact sum was the problem of, given as input an efficient classical circuit computing a function $f : \{0,1\}^n \to \{0,1\}$, *exactly* compute $\sum_{x\in\{0,1\}^n} f(x)$. The "quantum" exact sum was the same problem with respect to the input function $g : \{0,1\}^n \to \{\pm 1\}$. Prove that these problems are of the same difficulty – i.e., show that I can solve either problem if I have the ability to solve only one of the problems.

Problem 2: Now consider the *multiplicative approximation* analogues of the definitions from Problem 1 – that is define the "classical" approximate sum to be the problem of, given as input an efficient classical circuit computing a function $f : \{0,1\}^n \to \{0,1\}$, compute a multiplicative estimate α so that

$$(1-\epsilon)\sum_{x\in\{0,1\}^n} f(x) \leq \alpha \leq (1+\epsilon)\sum_{x\in\{0,1\}^n} f(x)$$

in time $poly(n, 1/\epsilon)$. Similarly, the "quantum" approximate sum was the same multiplicative estimation problem with respect to the input function $g : \{0,1\}^n \to \{\pm 1\}$. In lecture I claimed that the "quantum" approximate sum problem is strictly harder than the "classical" approximate sum problem. But consider the following claimed reduction: suppose I take an instance of the "quantum" approximate sum problem specified by an input function $g : \{0,1\}^n \to \{\pm 1\}$. Then I can define two efficiently computable functions $g_+ : \{0,1\}^n \to \{0,1\}$ and $g_- : \{0,1\}^n \to \{0,1\}$ so that

$$\sum_{x\in\{0,1\}^n} g(x) = \sum_{x\in\{0,1\}^n} g_+(x) - \sum_{x\in\{0,1\}^n} g_-(x)$$

I have written the "quantum" sum as the difference of two "classical" sums. Suppose I can solve the "classical" approximate sum problem. Then I can use this ability twice to get good multiplicative estimates to $\sum_{x\in\{0,1\}^n} g_+(x)$ and $\sum_{x\in\{0,1\}^n} g_-(x)$. Now I can subtract these estimates. What sort of additive error can this approximate procedure incur? Does this prove that the "quantum" approximate sum is no harder than "classical" approximate sum?

Problem 3: In the lecture we proved that the Polynomial time hierarchy collapses if we have a classical algorithm that samples exactly from the same output distribution as a quantum circuit, that is a classical algorithm S so that for all quantum circuits C and outcomes $y \in \{0,1\}^n$:

$$Pr_r\left[S(C,r) = y\right] = |\langle y|C|0^n\rangle|^2$$

But as we've discussed, there is a trivial *quantum* algorithm that samples from this distribution: just run C on $|0^n\rangle$ and measure all n qubits! Why doesn't this quantum sampling algorithm also collapse the Polynomial time hierarchy, by similar arguments involving Stockmeyer's result?

Problem 4: Recall our hardness of exact quantum sampling result worked by showing that, given a classical circuit computing a function $g : \{0,1\}^n \to \{\pm 1\}$ there is a quantum algorithm that, after measuring all n qubits in the standard basis, outputs a string $y \in \{0,1\}^n$ with probability

$p_y = \frac{\sum_x (-1)^{\langle x,y \rangle} g(x)}{2^{2n}}$. In particular, when $y = 0^n$, p_{0^n} is proportional to $(\sum_x g(x))^2$, which is an instance of squared quantum sum. Is this hardness proof resilient to classical sampling algorithms that, rather than sampling *exactly* from the same distribution, samples:

1. From any distribution X that assigns each string y a probability $\tilde{p}_y$ so that $(1-\epsilon)p_y \leq \tilde{p}_y \leq (1+\epsilon)p_y$ where $\epsilon = \frac{1}{poly(n)}$.
2. From any distribution X that is $\frac{1}{poly(n)}$-close in total variation distance to the outcome distribution of the quantum algorithm.

Problem 5: In the proof of hardness of exact sampling we used the fact that computing a single outcome probability of a worst-case quantum circuit (e.g., the probability that the 0^n outcome is observed, as discussed in Problem 4) is as hard as the approximate squared quantum sum problem, which is $\sharp P$-hard. To show that this implies the hardness of classical sampling from this distribution, we needed an argument involving Stockmeyer's algorithm. Why don't we instead do something much simpler – suppose there exists a classical sampling algorithm. Let's use this algorithm to repeatedly sample from the quantum distribution a polynomial number of times, then look at the fraction of outcomes equal to 0^n and use this as an estimate for the output probability of the 0^n outcome. Does this work to prove hardness of exact sampling with a classical algorithm? How accurate of an estimate does this procedure attain?

Problem 6: Recall we proved Lipton's theorem stating that the problem of computing the Permanent of a matrix (with entries over a sufficiently large finite field) is $\sharp\mathsf{P}$-hard *on average*.

Recall that the determinant of a matrix is defined to be:

$$Det[X] = \sum_{\sigma \in S_n} \left(sign(\sigma) \prod_{i=1}^{n} X_{i,\sigma(i)} \right)$$

Where S_n is the symmetric group on n elements and $sign(\sigma)$ is the function on permutations defined to be $+1$ if the permutation can be obtained with an even number of exchanges of two entries and otherwise is defined to be -1.

The Determinant, unlike the Permanent, is efficiently computable, so it shouldn't be $\sharp\mathsf{P}$-hard on average. Is the Determinant as hard on average as it is in the worst-case? Which step in the average-case hardness proof breaks for the Determinant?

Problem 7: Recall the setting of Lipton's average-case hardness result: we have a "faulty" algorithm that correctly computes the Permanent on average. Our goal is to use this faulty algorithm to correctly compute the Permanent on a "worst-case" (i.e., arbitrary) matrix. Lipton's argument proceeds by a polynomial extrapolation argument. Why do we need such an argument – in particular, since the faulty algorithm succeeds with high probability, say probability $1 - \frac{1}{6n}$, why can't we simply repeat the faulty algorithm on our worst-case matrix and then take a majority vote to compute the Permanent?

Problem 8: We are interested in classically *verifying* that a noisy quantum experiment samples from a distribution p_{exp} that is close in total variation distance to the ideal output distribution of a fixed but random quantum circuit, p_{ideal}, since our complexity theoretic hardness results work under this assumption. Recall we define total variation distance as:

$$TVD(p_{exp}, p_{ideal}) = \frac{1}{2} \sum_{x \in \{0,1\}^n} |p_{exp}(x) - p_{ideal}(x)|$$

On the other hand, modern quantum advantage experiments are verified using different closeness measures such as the so-called "cross-entropy metric", defined to be:

$$XEB(p_{exp}, p_{ideal}) = \sum_{x \in \{0,1\}^n} p_{exp}(x) \log(1/p_{ideal}(x))$$

We note that this is the "logarithmic" variant of the metric, which is different from the "linear" variant defined in lecture. While in general XEB and TVD can be quite different, prove that if we make the *assumption* that the Shannon entropy $H(p_{exp}) > H(p_{ideal})$ then we have that scoring $1-\epsilon$ on XEB implies a non-trivial upper bound on TVD (here we assume that ϵ is a sufficiently small constant and that a XEB score of 1 is the ideal score if no errors occured). *Hint: use Pinsker's inequality i.e., that* $TVD(p_{exp}, p_{ideal}) \leq \sqrt{\frac{1}{2}|p_{exp} - p_{id}|_{KL}}$ *and the fact that the KL divergence metric can be written in terms of cross-entropy.*

Is the assumption that we score $1-\epsilon$ as the system size increases on XEB reasonable for quantum experiments that are not error-corrected?

Problem 9: Consider a distribution $\mathcal{D}$ over quantum circuits C with Haar random two-qubit gates over a fixed architecture as defined in lecture. Calculate $E_{C\sim\mathcal{D}}\left[|\langle 0^n|C|0^n\rangle|^2\right] = E_{C\sim\mathcal{D}}\left[p_{0^n}(C)\right]$. Argue that this quantity would remain the same even if we replaced the 0^n outcome with any outcome $x \in \{0,1\}^n$.

Problem 10: Recall the Heavy Output Generation ("HOG") problem – given as input a random circuit C output strings $x_1, x_2, ..., x_k$ so that at least 2/3 are "heavy" (i.e., we call a string x_i *heavy* if the output probability $|\langle x_i|C|0^n\rangle|^2$ is greater than median in the output distribution of C). Also recall the Quantum Threshold Assumption or "$QUATH$": No efficient classical algorithm takes as input random C and decided if the output probability of 0^n outcome is heavy with probability $1/2 + \Omega(1/2^n)$, where probability is over both and internal randomness of the classical algorithm. Prove that $QUATH$ implies that HOG is hard.

Problem 11: Recall the "$XHOG$" problem is the variant of HOG in which we are given as input a random circuit C and are asked to output strings $x_1, x_2, ..., x_k$ so that $E_i\left[|\langle x_i|C|0^n\rangle|^2\right] \geq \frac{b}{2^n}$ where $b = 1 + \epsilon$ for some $\epsilon > 0$. Correspondingly we recall the "$XQUATH$" assumption: No efficient classical algorithm takes as input a random circuit C and outputs an estimate p to the output probability of the 0^n outcome, p_{0^n}, so that:

$$2^{2n}\left(E_C\left[\left(p_{0^n} - \frac{1}{2^n}\right)^2\right] - E_C\left[\left(p_{0^n} - p\right)^2\right]\right) = \Omega(2^{-n})$$

Describe the strategy to prove that $XQUATH$ implies that $XHOG$ is hard (i.e., while the proof itself is a bit technical, the big picture idea behind the reduction is straightforward and can be easily described.)

Problem 12: Prove that the probability of each outcome string $x \in \{0,1\}^n$ with respect to a quantum circuit $C = C_d C_{d-1} ... C_1$ can be expressed as a path integral over Pauli paths:

$$p_x = |\langle x|C|0^n\rangle|^2 = \sum_{s \in P_n^{d+1}} f(C, s, x)$$

Where the function $f(C, s, x)$ is a product of $d+1$ Pauli transition amplitudes of the circuit C:

$$f(C, s, x) = Tr(|x\rangle\langle x|s_d) Tr(s_d C_d s_{d-1} C_d^\dagger) ... Tr(s_1 C_1 s_0 C_1^\dagger) Tr(s_0 |0^n\rangle\langle 0^n|)$$

Problem 13: Let us use the same notation as the previous problem and assume the "orthogonality of Pauli paths" (recall this is "Fact 2" in the lecture and means that $E_C\left[f(C, s, x) f(C, s', x)\right] = 0$ for

any two Pauli paths $s \neq s'$ and for any $x \in \{0,1\}^n$). Use this to prove that $E_C[f(C,s,x)] = 0$ for any path $s \neq I_n^{\otimes d+1}$ where I is the identity operator.