Lecture 6: Approximate Degree

Scribe: Aditya Ranjan   Lecture: Rajat Mittal

IIT Kanpur

1 Lower bounding Quantum Query Complexity

Let us recall a lower bound for the decision tree complexity of a Boolean function f: the **degree of** f. The proof sketch is as follows:

Proof. Consider the decision tree for f having depth $\mathrm{D}(f)$. Corresponding to each computation path (i.e. a path from the root to some output leaf) in the tree, we can get an indicator polynomial. The indicator polynomial corresponding to any computation path is the product of either x_i or $(1 - x_i)$, based on whether we set its value to be 1 or 0 respectively in the path, for all the variables x_i in the path. It can be seen that the degree of each of these indicator polynomials is equal to the depth of the leaf, and hence is at most $\mathrm{D}(f)$. The (unique) polynomial corresponding to f is the sum of these indicator polynomials, all having degree at most $\mathrm{D}(f)$. Hence, the degree of their sum, $\deg(f) \leqslant \mathrm{D}(f)$. □

1.1 Quantum Query Complexity

Though we have not defined quantum query complexity formally (which comes from the theory of Quantum Computation), we state a similar statement (without proof) for it as well (this will help us achieve lower bounds on the quantum query complexity):

Theorem 1. *The probability of* ***acceptance*** *of a quantum query algorithm on input $x = x_1x_2 \dots x_n$ is a polynomial in $x_1, x_2, \dots, x_n$ having degree at most* ***twice*** *the number of queries.*

Note 1. Acceptance here refers to the event of the Quantum algorithm giving 1 as the output (this is analogous to the definition of acceptance of a language in the Theory of Computation). There is no mention of any Boolean function f here, the theorem is independent of that.

Notice that this statement gives some sort of a lower bound on the number of queries (i.e. the Quantum query complexity). Let us see how we will use it to prove lower bounds.

Let f be a Boolean function. Now, say we want our Quantum algorithm Q to compute f. We must have $\Pr(\mathrm{Q}(x) = f(x)) \geqslant \frac{2}{3}$ for every input x. Thus, we can say that for each input x:

$$f(x) = 0 \implies \text{Probability of Acceptance} \leqslant \frac{1}{3} \qquad \text{(Acceptance is failure here)}$$

$$f(x) = 1 \implies \text{Probability of Acceptance} \geqslant \frac{2}{3} \qquad \text{(Acceptance is success here)}$$

As specified in Theorem **??**, let the probability of acceptance of Q on input x be a polynomial $P(x)$. The condition for Q computing f can be rewritten as:

$$|f(x) - P(x)| \leqslant \frac{1}{3} \text{ for all inputs } x \tag{1}$$

Note 2. As with all of our previous discussions, there is nothing extraordinarily special about $\frac{1}{3}$ here, we can replace it by any $\epsilon < \frac{1}{2}$.

Thus, if we have a quantum algorithm Q which computes f, then there is a corresponding polynomial P which satisfies (**??**) (and also the constraint $0 \leqslant P(x) \leqslant 1 \,\forall\, x$, as P represents a probability) such that the number of queries taken by Q is at least $\frac{1}{2}\deg(P)$.

This also means that for a given Boolean function f, if we look for **all** of the polynomials P *in general* which satisfy (**??**), then the lowest degree among all these polynomials will give us a lower bound for complexities of **all** quantum algorithms which compute f!

Note 3. Here we have removed the constraint $0 \leqslant P(x) \leqslant 1 \,\forall\, x$, as 1. We already had the constraint $-\frac{1}{3} \leqslant P(x) \leqslant \frac{4}{3} \,\forall\, x$ from (**??**) itself, and 2. This additional constraint does not really change the lower bound for complexities very much.

Let us study more about the polynomials satisfying (**??**).

2 Approximate Degree of a Boolean Function

Let's say that we want our quantum algorithm Q to compute f "perfectly" (with 0 error), i.e. we want $\Pr(\mathrm{Q}(x) = f(x)) = 1$. Then we get that $|f(x) - P(x)| \leqslant 0$, i.e. $P(x) = f(x) \,\forall\, x$, where P is Q's probability of acceptance. Then we must have that P is the (unique) polynomial representation of f, and so the number of queries used by such a 0-error quantum algorithm Q is $\geqslant \frac{1}{2}\deg(P) = \frac{1}{2}\deg(f)$. Hence, $\frac{1}{2}\deg(f)$ is a lower bound for $\mathrm{Q}_0(f)$. But for our purposes, we might not be so strict that we want to compute f perfectly.

We say that a polynomial P **approximates** f, if

$$|f(x) - P(x)| \leqslant \frac{1}{3} \text{ for all inputs } x$$

Note 4. This is just one notion of approximation, there are many more such notions in mathematics. For example, another notion is that $P(x) = f(x)$ for a high fraction of inputs x. But this notion is not useful for us, as for e.g. if f is OR, then the constant (zero degree) polynomial 1 approximates OR (as it is equal to OR on almost all inputs).

For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, its approximate degree, $\widetilde{\deg}_{1/3}(f)$ is the **minimum** possible degree of a polynomial p which approximates f.

$$\widetilde{\deg}_{1/3}(f) = \min_{\substack{\text{polynomial } p: \\ |p(x)-f(x)| \leqslant 1/3 \;\forall\, x}} \deg(p) \tag{2}$$

It is not difficult to see that $\widetilde{\deg}_{1/3}(f) \leqslant \deg(f)$, as the polynomial representation for f trivially approximates f. Infact, if $\epsilon \geqslant \delta$, then $\widetilde{\deg}_{\epsilon}(f) \leqslant \widetilde{\deg}_{\delta}(f)$.

2.1 Approximate degree & $\mathbf{Q_{1/3}(f)}$

Similar to how we defined $\mathrm{R}_{1/3}(f)$, $\mathrm{Q}_{1/3}(f)$ refers to the smallest number of queries used by any quantum algorithm which computes f.

Now, if there is a quantum algorithm for f which uses t queries, then $\exists$ an approximating polynomial for f with degree at most $2t$. This comes from Theorem **??**, and here the approximating polynomial mentioned is infact the probability of acceptance of that quantum algorithm.

Now consider the optimal quantum algorithm which computes f, it uses $\mathrm{Q}_{1/3}(f)$ queries, so there is an approximating polynomial for f with degree $\leqslant 2 \cdot \mathrm{Q}_{1/3}(f)$. The degree of this approximating polynomial is at least the approximate degree of f.

So finally, even without knowing about quantum computation in general and just using Theorem **??**, we get a lower bound using the approximate degree of f:

$$\boxed{\mathrm{Q}_{1/3}(f) \geqslant \frac{1}{2}\widetilde{\deg}_{1/3}(f)} \tag{3}$$

Notice how this is analogous to the statement $\mathrm{D}(f) \geqslant \deg(f)$.

3 Techniques for getting Approximate Degrees

3.1 Approximate Degree of PARITY

We wish to find out the approximate degree of PARITY, but going by the definition, we need to find out the minimum degree among *all* polynomials which approximate PARITY. As usual, this lower bounding seems to be a difficult job, so we must use technique(s) in order to do it. We used the adversary arguments and hard distributions to lower bound $\mathrm{D}(f)$ and $\mathrm{R}_{1/3}(f)$ respectively. But here we are lower bounding degrees of polynomials, which are nicer mathematical objects than Decision Trees and Randomized Decision Trees.

One way of lower bounding the degree of a polynomial can be to argue about its number of roots. But here, our polynomials are multivariate polynomials, not univariate ones, so we cannot use the number of roots directly. Fortunately for us, PARITY is symmetric. We had seen before that a symmetric function/polynomial (in $x \in \{0,1\}^n$) can be thought of as a **univariate** polynomial (in $|x| \in \{0,1,\ldots,n\}$) having the **same** degree as the original polynomial. But though f here is symmetric, our polynomial p approximating f need not be a symmetric polynomial. Thus, we use a **symmetrization trick** to convert p into a symmetric polynomial first.

Symmetrization Trick (by Minsky & Papert)

Theorem 2. *Let p be a polynomial (and a function $p : \{0,1\}^n \to \mathbb{R}$) and let p_{symm} be the symmetric polynomial*

$$p_{\text{symm}}(x) = \frac{1}{n!}\sum_{\sigma \in S_n} p(\sigma(x))$$

If p approximates a symmetric Boolean function f, p_{symm} approximates f too.

Proof. As f is symmetric, we can write f as

$$f(x) = \frac{1}{n!}\sum_{\sigma \in S_n} f(\sigma(x))$$

Thus, we have

$$|p_{\text{symm}}(x) - f(x)| = \frac{1}{n!}\left|\sum_{\sigma \in S_n} p(\sigma(x)) - f(\sigma(x))\right| \leqslant \frac{1}{n!}\sum_{\sigma \in S_n} |p(\sigma(x)) - f(\sigma(x))| \leqslant \frac{1}{n!}\sum_{\sigma \in S_n} \frac{1}{3} = \frac{1}{3}$$

Here, we have used the triangle inequality. Thus, p_{symm} approximates f if p approximates f. □

Note that p_{symm} has the same degree as that of p (this can be proven).

Because p_{symm} is symmetric, we can further convert it into a **univariate** polynomial (in $|x|$) which has the same degree as p_{symm}, i.e. same degree as p.

Corollary. *If f is symmetric and $\widetilde{\deg}_{1/3}(f) = d$, then there exists a univariate polynomial $P(w)$ of degree d such that*

$$|P(w) - f(w)| \leqslant \frac{1}{3} \qquad \forall\, w \in \{0, 1, \ldots, n\} \tag{4}$$

Now, let us consider such an optimal degree univariate polynomial P for PARITY. Note that,

$$\text{PARITY}(w) = \begin{cases} 0, & \text{if } w \text{ is even} \\ 1, & \text{if } w \text{ is odd} \end{cases}$$

This means that $-\frac{1}{3} \leqslant P(w) \leqslant \frac{1}{3}$ if w is even, and $\frac{2}{3} \leqslant P(w) \leqslant \frac{4}{3}$ if w is odd. So P roughly looks like this:

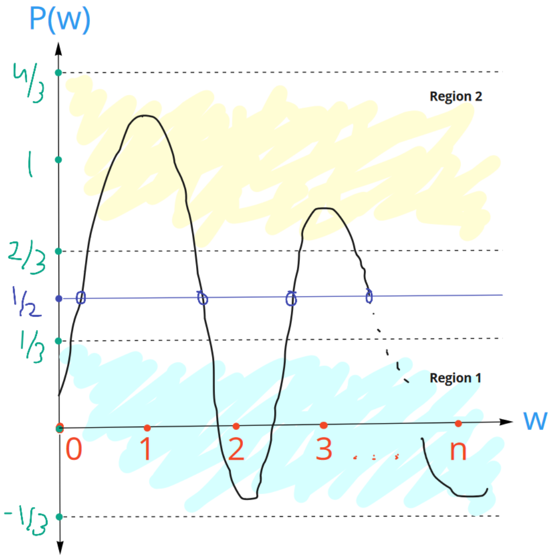

Fig. 1. Rough sketch of P for PARITY

So $P(w)$ goes up and down from Region 1 to Region 2 on each successive step. Let us see its intersection with a line. Let us choose the line $y = \frac{1}{2}$ (any value between $\frac{1}{3}$ and $\frac{2}{3}$ works). We can see that the line intersects the polynomial many times. More formally, it can be seen that $\forall 1 \leqslant k \leqslant n$, $P(k-1)$ and $P(k)$ are in different regions (among region 1 & 2), i.e. one of $P(k-1)$ and $P(k)$ lies in $\left[-\frac{1}{3}, \frac{1}{3}\right]$, and the other one lies in $\left[\frac{2}{3}, \frac{4}{3}\right]$. Since polynomials are continuous, we can say that $\exists\, t \in (k-1, k)$ such that $P(t) = \frac{1}{2}$, and we can find such a $t \in (k-1, k)$ for every $1 \leqslant k \leqslant n$. This means $y = P(w)$ intersects $y = \frac{1}{2}$ at least n times, i.e. $P(w) - \frac{1}{2}$ has at least n roots, and hence $P(w)$ has at least n roots. Hence $\deg(P) \geqslant n$. The P we chose came from the optimal polynomial approximating PARITY, and hence $\widetilde{\deg}_{1/3}(\text{PARITY}) \geqslant n$. We already knew that $\widetilde{\deg}_{1/3}(\text{PARITY}) \leqslant \deg(\text{PARITY}) = n$, and hence this shows that $\widetilde{\deg}_{1/3}(\text{PARITY}) = n$.

3.2 Approximate Degree of OR

We saw how doing the symmetrization trick for PARITY, we got a univariate polynomial having degree equal to the approximate degree of PARITY, and then we lower bound the degree of this univariate polynomial using observations on this polynomial.

Because OR is also a symmetric Boolean function, we can use the same strategy as above for OR. Now, there already exists a quantum algorithm for OR which has $O(\sqrt{n})$ complexity (this is the Grover's Search algorithm). So we know that $\widetilde{\deg}_{1/3}(\mathrm{OR})$ can be at most $O(\sqrt{n})$. We are going to show that it is also $\Omega(\sqrt{n})$, i.e. the bound is tight.

As we did before, we choose a univariate polynomial approximating OR having optimal degree, say P. Now,

$$\mathrm{OR}(w) = \begin{cases} 0, & \text{if } w = 0 \\ 1, & \text{if } w = 1, 2, \ldots, n \end{cases}$$

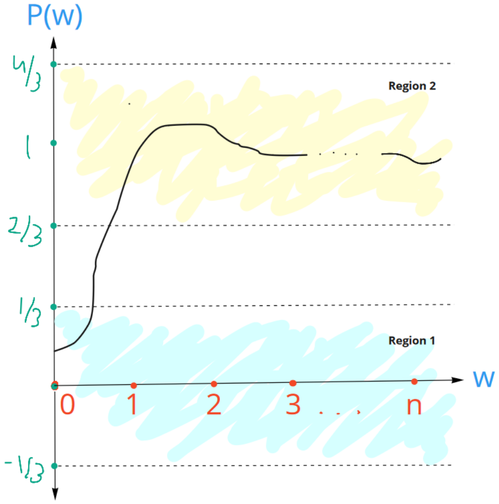

Fig. 2. Rough sketch of P for OR

This means that $P(w)$ remains in Region 1 ($\left[-\frac{1}{3}, \frac{1}{3}\right]$) at $w = 0$, and then shifts to Region 2 ($\left[\frac{2}{3}, \frac{4}{3}\right]$) at $w = 1, 2, \ldots n$. Thus, there is a sharp (i.e. non-zero) derivative of $P(w)$ in $(0, 1)$, and after that P remains bounded, at least on the integer points. We can then try the following inequality to lower bound the degree of P:

Markov Brothers' Inequality

Theorem 3. *For a polynomial P, if $b_1 \leqslant P(x) \leqslant b_2$ in the interval $a_1 \leqslant x \leqslant a_2$, then*

$$|P'(x)| \leqslant \frac{d^2(b_2 - b_1)}{a_2 - a_1} \qquad \forall\ x \in [a_1, a_2] \tag{5}$$

where d is the degree of P.

We know that using Lagrange's theorem, there is some $x \in (0, 1)$ such that $P'(x) = \frac{P(1)-P(0)}{1-0} \geqslant \frac{1}{3}$. One can then think of using Markov Brothers' Inequality for the interval $x \in [0, n]$ in which $P(x) \in [-\frac{1}{3}, \frac{4}{3}]$, which would indeed give us that $\deg(P) \geqslant c\sqrt{n}$ for some constant $c > 0$. But there is a mistake here, we cannot use the Markov Brothers' Inequality directly here, because even though we know that $P(x)$ is bounded (by bound of constant size) on the integer points in $[0, n]$, there is no guarantee that it will remain bounded **by a bound of constant size** in the whole interval $[0, n]$.

Note 5. We know that since P is a polynomial, P is bounded on any interval and hence we can use the inequality for the interval $[0, n]$. But then it can happen that the size of the bound is not a constant. For e.g. if the size of the bound is n instead of a constant, then we get an asymptotically smaller bound than our required lower bound of $\sqrt{n}$ for $\deg(P)$.

However, there is another result (which is derived from Markov Brothers' Inequality itself) that lower bounds the degree of a polynomial which is bounded on just integer points:

Degree Lower Bound for Polynomial bounded at integer points (Ehlich & Zeller)

Theorem 4. *If P is a polynomial such that $b_1 \leqslant P(i) \leqslant b_2 \ \forall \ i \in \{0, 1, \ldots, n\}$, and $\exists \, 0 \leqslant x \leqslant n$ such that $|P'(x)| \geqslant c$, then*

$$\deg(P) \geqslant \sqrt{\frac{cn}{c + b_2 - b_1}} \tag{6}$$

We can use this result for our case: we can take $c = \frac{1}{3}, b_1 = -\frac{1}{3}, b_2 = \frac{4}{3}$ and apply this result directly to get $\deg(P) = \Omega(\sqrt{n})$, and hence we proved that $\widetilde{\deg}_{1/3}(\mathrm{OR}) = \Omega(\sqrt{n})$. Thus, the Grover's Search quantum algorithm is asymptotically the most optimal quantum algorithm for searching.

Infact, using the exact same strategy as we did for OR (getting univariate P and using Ehlich & Zeller on it), the following statement holds:

Corollary. *If f is a non-constant symmetric Boolean function, then $\widetilde{\deg}_{1/3}(f) = \Omega(\sqrt{n})$*