Quantum algorithms (CO 781/CS 867/QIC 823, Winter 2011)

Andrew Childs, University of Waterloo

LECTURE 19: The polynomial method

In this lecture we discuss a second method for establishing lower bounds on quantum query complexity, the polynomial method. This can be more difficult to apply than the adversary method, but sometimes gives better lower bounds (than the adversary method with positive weights). We begin by discussing the basic approach and some cases in which its application is relatively easy.

Quantum algorithms and polynomials

The following shows a basic connection between quantum algorithms and polynomials.

Lemma. *The acceptance probability of a t-query quantum algorithm for a problem with black-box input $x \in \{0,1\}^n$ is a polynomial in $x_1, \ldots, x_n$ of degree at most $2t$.*

Proof. We claim that the amplitude of any basis state is a polynomial of degree at most t, so that the probability of any basis state (and hence the probability of success) is a polynomial of degree at most $2t$.

The proof is by induction on t. If an algorithm makes no queries to the input, then its success probability is independent of the input, so it is a constant, a polynomial of degree 0.

For the induction step, a query maps

$$|i,b\rangle \mapsto (-1)^{bx_i}|i,b\rangle \tag{1}$$

$$= (1-2bx_i)|i,b\rangle, \tag{2}$$

so it increases the degree of each amplitude by at most 1. $\square$

Consider a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$. We say a polynomial $p \in \mathbb{R}[x_1,\ldots,x_n]$ represents f if $p(x) = f(x)$ for all $x \in \{0,1\}^n$. Letting $\deg(f)$ denote the smallest degree of any polynomial representing f, we have $Q_0(f) \ge \deg(f)/2$.

To handle bounded-error algorithms, we introduce the concept of *approximate degree*. We say a polynomial p ϵ-represents f if $|p(x)-f(x)| \le \epsilon$ for all $x \in \{0,1\}^n$. Then the ϵ-approximate degree of f, denoted $\widetilde{\deg}_\epsilon(f)$, is the smallest degree of any polynomial that ϵ-represents f. Clearly, $Q_\epsilon(f) \ge \widetilde{\deg}_\epsilon(f)/2$. Since bounded-error query complexity does not depend strongly on the particular error probability ϵ, we can define, say, $\widetilde{\deg}(f) := \widetilde{\deg}_{1/3}(f)$.

Now to lower bound the quantum query complexity of a Boolean function, it suffices to lower bound its approximate degree.

Symmetrization

While polynomials are well-understood objects, the acceptance probability is a multivariate polynomial, so it can be rather complicated. Since $x^2 = x$ for $x \in \{0,1\}$, we can restrict our attention to multilinear polynomials, but it is still somewhat difficult to deal with such polynomials directly. Fortunately, for many functions it suffices to consider a related univariate polynomial obtained by symmetrization.

For a string $x \in \{0,1\}^n$, let $|x|$ denote the Hamming weight of x, the number of 1s in x.

Lemma. *Given any n-variate multilinear polynomial p, let $P(k) := \mathbb{E}_{|x|=k}[p(x)]$. Then P is a polynomial with* $\deg(P) \le \deg(p)$.

Proof. Since p is multilinear, it can be written as a sum of monomials, i.e., as

$$p(x) = \sum_{S \subseteq \{1,\ldots,n\}} c_S \prod_{i \in S} x_i \tag{3}$$

for some coefficients c_S. Then we have

$$P(k) = \sum_{S \subseteq \{1,\ldots,n\}} c_S \mathop{\mathbb{E}}_{|x|=k}\left[\prod_{i \in S} x_i\right] \tag{4}$$

and it suffices to compute the expectation of each monomial. We find

$$\begin{aligned}
\mathop{\mathbb{E}}_{|x|=k}\left[\prod_{i \in S} x_i\right] &= \Pr_{|x|=k}[\forall\, i \in S,\ x_i = 1] && (5)\\
&= \frac{\binom{n-|S|}{k-|S|}}{\binom{n}{k}} && (6)\\
&= \frac{(n-|S|)!\, k!\, (n-k)!}{(k-|S|)!\, (n-k)!\, n!} && (7)\\
&= \frac{(n-|S|)!}{n!} k(k-1)\cdots(k-|S|+1) && (8)
\end{aligned}$$

which is a polynomial in k of degree $|S|$. Since $c_S = 0$ whenever $|S| > \deg(p)$, we see that $\deg(P) \le \deg(p)$. □

Thus the polynomial method is a particularly natural approach for symmetric functions, functions that only depend on the Hamming weight of the input.

Parity

Let $\textsc{parity}\colon \{0,1\}^n \to \{0,1\}$ denote the symmetric function $\textsc{parity}(x) = x_1 \oplus \cdots \oplus x_n$. Recall that Deutsch's problem, which is the problem of computing the parity of 2 bits, can be solved exactly with only one quantum query. Applying this algorithm to a pair of bits at a time and then taking the parity of the results, we see that $Q_0(\textsc{parity}) \le n/2$.

What can we say about lower bounds for computing parity? Symmetrizing $\textsc{parity}$ gives the function $P\colon \{0,1,\ldots,n\} \to \mathbb{R}$ defined by

$$P(k) \begin{cases} 0 & \text{if } k \text{ is even} \\ 1 & \text{if } k \text{ is odd.} \end{cases} \tag{9}$$

Since P changes direction n times, $\deg(P) \ge n$, so we see that $Q_0(\textsc{parity}) \ge n/2$. Thus Deutsch's algorithm is tight among zero-error algorithms.

What about bounded-error algorithms? To understand this we would like to lower bound the approximate degree of PARITY. If $|p(x) - f(x)| \le \epsilon$ for all $x \in \{0,1\}^n$, then

$$|P(k) - F(k)| = \left| \mathop{\mathbb{E}}_{|x|=k} (p(x) - f(x)) \right| \le \epsilon \tag{10}$$

for all $k \in \{0, 1, \ldots, n\}$, where P is the symmetrization of p and F is the symmetrization of f. Thus, a multilinear polynomial p that ϵ-approximates PARITY implies a univariate polynomial P satisfying $P(k) \le \epsilon$ for k even and $P(k) \ge 1 - \epsilon$ for k odd. For any $\epsilon < 1/2$, this function still changes direction n times, so in fact we have $\widetilde{\deg}_\epsilon(f) \ge n$, and hence $Q_\epsilon(\text{PARITY}) \ge n/2$.

This shows that the strategy for computing parity using Deutsch's algorithm is optimal. This is an example of a problem for which a quantum computer cannot get a significant speedup—here the speedup is only by factor of 2. In fact, we need at least $n/2$ queries to succeed with any bounded error, even with very small advantage (e.g., even if we only want to be correct with probability $\frac{1}{2} + 10^{-100}$). In contrast, while the adversary method can prove an $\Omega(n)$ lower bound for parity, the constant factor that it establishes is error-dependent.

Note that this also shows we need $\Omega(n)$ queries to exactly count the number of marked items in an unstructured search problem, since exactly determining the number of 1s would in particular determine whether the number of 1s is odd or even.

Unstructured search

Next we will see how the polynomial method can be used to prove the $\Omega(\sqrt{n})$ lower bound for computing the logical OR of n bits. Symmetrizing OR gives a function $F(k)$ with $F(0) = 0$ and $F(1) = 1$. We also have $F(k) = 1$ for all $k > 1$, but we will not actually need to use this. This function is monotonic, so we cannot use the same simple argument we applied to parity. Nevertheless, we can prove that $\widetilde{\deg}(\text{OR}) = \Omega(\sqrt{n})$ using the following basic fact about polynomials, due to Markov.

Lemma. *Let $P \colon \mathbb{R} \to \mathbb{R}$ be a polynomial. Then*

$$\max_{x\in[0,n]} \frac{\mathrm{d}P(x)}{\mathrm{d}x} \le \frac{\deg(P)^2}{n} \left(\max_{x\in[0,n]} P(x) - \min_{x\in[0,n]} P(x) \right). \tag{11}$$

In other words, if we let

$$h := \max_{x\in[0,n]} P(x) - \min_{x\in[0,n]} P(x) \tag{12}$$

denote the "height" of P in the range $[0, n]$, and

$$d := \max_{x\in[0,n]} \frac{\mathrm{d}P(x)}{\mathrm{d}x} \tag{13}$$

denote the largest derivative of P in that range, then we have $\deg(P) \ge \sqrt{nd/h}$.

Now let P be a polynomial that ϵ-approximates OR. Since $P(0) \le \epsilon$ and $P(1) \ge 1 - \epsilon$, P must increase by at least $1 - 2\epsilon$ in going from $k = 0$ to $k = 1$, so $d \ge 1 - 2\epsilon$.

We have no particular bound on h, since we have no control over the value of P at non-integer points; the function could become arbitrarily large or small. However, since $P(k) \in [0, 1]$ for $k \in \{0, 1, \ldots, n\}$, a large value of h implies a large value of d, since P must change fast enough to start from and return to values in the range $[0,1]$. In particular, P must change by at least $(h-1)/2$ over a range of k of width at most $1/2$, so we have $d \geq h-1$. Therefore,

$$\deg(P) \geq \sqrt{\frac{n \max\{1-2\epsilon, h-1\}}{h}} \tag{14}$$
$$= \Omega(\sqrt{n}). \tag{15}$$

It follows that $Q(\textsc{or}) = \Omega(\sqrt{n})$.

Note that the same argument applies for a function that takes the value 0 whenever $|x| = w$ and the value 1 whenever $|x| = w+1$, for any w; in particular, it applies to any non-constant symmetric function. (Of course, we can do better for some symmetric functions, such as PARITY and also MAJORITY, among others.)