Quantum Query Complexity

Akash Kumar Singh, Snehal Raj, Siddhant Kar

June 2020

Chapter 1

Introduction

This report focuses on some recent work regarding a quantum query problem, namely FORRELATION. This problem has achieved the largest known gap between classical and quantum complexity yet among promise problems. There has also developed a new notion of block multilinear polynomials, which has helped in solving quantum query problems, including FORRELATION, in sub-linear classical time. We then explore this notion further by understanding block multilinear degree and its relation to standard degree.

1.1 Quantum Query Model

The quantum query model is used to answer some question about a boolean function, either partial (promise problem) or total (total problem). A partial function has some condition on the outputs and hence can be constructed using fewer inputs, whereas a total function has no such restriction. For example, Simon's problem is a promise problem and Grover's total search is total.

For our convenience, we assume these are functions from $\{-1,1\}^n$ to $\{-1,1\}$. Let us represent them in the standard basis as $x = (x_1, \ldots, x_N)$, where $N = 2^n$ and $\forall i\ x_i \in \{-1,1\}$ corresponds to the ith output. We want to check a property f, which answers a question about x with yes or no (decision problem). We have

$$f : \{-1,1\}^N \longrightarrow \{0,1\}.$$

We are given access to an oracle or black box O_x which is a quantum gate that, on input $|i\rangle$, does either of the following.

$$\begin{array}{cc} & \text{[wires=2]}i \quad \text{[wires=3]O}_x[wires = 2]\text{i} \\ b \quad b \oplus x_i & \text{[wires=2]}i \quad \text{[wires=3]O}_x[wires = 3]\text{x}_i ib \\ & b \end{array}$$

Both the oracles shown above are equivalent. For example, to convert the first to the second, we set the control bit $|b\rangle$ to $|-\rangle$. We will only use the latter oracle that takes $|i\rangle$ and returns $x_i|i\rangle$. Our quantum query algorithm can call the oracle several times and also perform other linear operations in between as shown below.

$$[\text{wires=2}]i \quad [\text{wires=4}]\mathrm{U}_0[wires = 4]O_x[wires = 4]U_1[wires = 4]O_x \ \ldots \ [wires = 4]U_{t-1}[wires = 4]O_x[wires = 4]U_t$$
$$\ldots$$
$$[\text{wires=2}]w \quad \ldots$$
$$\ldots$$

The gates in between the oracles are fixed and do not vary with x. Measuring the final qubits should result in an accepted state with high probability if $f(x) = 1$ and with low probability if $f(x) = 0$. The number of calls t to the oracle O_x is called the quantum query complexity of the algorithm.

Definition 1.1.1. *A block multilinear polynomial on $\{-1, 1\}^N$ is a polynomial of the form*

$$p(x) = p(x_{1,1}, x_{1,2}, \ldots) = \sum_{(i_1,\ldots,i_k)} a_{i_1\ldots i_k} x_{1,i_1} \ldots x_{k,i_k}$$

where its N variables can be partitioned into k disjoint blocks $B_i, i \in [k]$ such that $x_{i,j} \in B_i \ \forall j$.

Theorem 1.1.1. *The probability that a quantum query algorithm of complexity t accepts a function x is given by $p(x, x, \ldots)$, where p is a block multilinear polynomial with $2t$ blocks of size N each, and $p(\bar{x})$ is bounded in $[-1, 1]$.*

Proof. Consider the query model discussed above. Let $x_{j,i}$ denote the value of x_i returned on the jth query to the oracle. We first prove that the amplitudes of the final state after t queries are block-multilinear in $x_{1,1}, \ldots, x_{t,N}$. We prove this using induction on t.

When $t = 0$, the amplitudes are simply given by constant polynomials. Let the result hold for $t - 1$ queries. So now, the state of the qubits after the gate U_{t-1} is given by

$$\sum_{i,w} a_{i,w}(x_{1,1}, \ldots, x_{t-1,N})|i\rangle|w\rangle.$$

After applying O_x, the state is

$$\sum_{i,w} x_{t,i} a_{i,w}(x_{1,1}, \ldots, x_{t-1,N})|i\rangle|w\rangle.$$

It is easy to see that the amplitudes are still block-multilinear, now with t blocks, even after the linear transformation U_t. The squares of the amplitudes are also block-multilinear, but with $2t$ blocks. We simply introduce a duplicate variable $x_{t+j,i}$ for each $x_{j,i}$ and then square amplitudes as shown below. The final polynomial $p(\bar{x})$ is given by

$$\sum_{i,w \in Acc} |a_{i,w}(x_{1,1}, \ldots, x_{t,N})|^2 = \sum_{i,w \in Acc} a^*_{i,w}(x_{1,1}, \ldots, x_{t,N}) a_{i,w}(x_{t+1,1}, \ldots, x_{2t,N})$$

which is clearly block multilinear. The probability that the algorithm accepts is simply $p(x, x, \ldots)$, which means that the norm of the vector of accepted state amplitudes is at most 1. Thus, $p(\bar{x})$ being an inner product lies in $[-1, 1]$. □

1.2 Forrelation

Forrelation is a promise problem which measures the correlation between a function f and the fourier transform of a second function g.