CS682 : Presentation

Abhinav Kumar(160019)
Rahul B S(160540)

Quantum Proofs for Classical Theorems
Andrew Drucker Ronald de Wolf

April 20, 2019

In this paper, the authors have surveyed various classical results and shown how ideas of quantum computing has been used to prove them. We will be representing some of these results.

Low Dimensional Encodings:

Suppose each x ∈ [N] has been encoded in a d dimensional quantum state $|\phi_x\rangle$ and $P_1, P_2, ..., P_N$ are the measurement operators used for decoding x, then we have:

p_x=P[correctly decoding x]=$||P_x|\phi_x\rangle||^2 <= trace(P_x)$

then $\sum_x p_x <= \sum_x trace(P_x) = trace(\sum_x P_x) = trace(I_d) = d$

So on average, we have the probability of decoding a particular x as d/N

Quantum Random Access Codes:

Suppose that instead of x, we wanted to decode one of the n bits of x that is the value of x_i , then if we have measurement operators $\{M_i, I - M_i\}$ such that $||M_i|\phi_x\rangle||^2 >= p$ if $x_i = 1$ and $||M_i|\phi_x\rangle||^2 <= 1 - p$ if $x_i = 0$ that is we can decode x_i correctly with probability at least p.

Bound on number of qubits(Due to Nayak):

Suppose we have a quantum random access code: x $|- > |\phi_x\rangle$ encoding n bit strings into m qubits states , and that we can decode x_i with success probability p, then we have $m >= (1 - H(p))n$ where H(.) is the binary entropy function.

$$H(p) = -plog(p) - (1 - p)log(1 - p)$$

Since H(p) measures the uncertainity in the number of bits, then if we have decoded x_i with success probability p, the uncertainity in the number of bits is H(p). So we have 1-H(p) bits of

information about x_i, so overall the state must have at least n(1-H(p)) bits of information about x which implies

$$m = \Omega(n(1 - H(p)))$$

It is because of the Holevo theorem which states that a m qubit states can encode at most 2m bits of information.

Locally Decodable Codes:

Suppose we have encoded a large string but at the time of recovering, we are interested in decoding small parts of the string. For example, a small piece of some large message.

Formally, Let C:$\{0,1\}^n -> \{0,1\}^N$ is a (q, δ, ϵ) LDC if there exists a classical randomized algorithm (A) such that:

1. A makes at most q queries to N bit string y (i.e accesses <= q bits of y)

2. For all n bit strings x and the index i $\in [n]$, we have
For all N bit strings y such that hamming distance d(C(x),y) <= δN,
the algorithm correctly computes the value of x_i with probability p >= $\frac{1}{2} + \epsilon$

Basically string y can be corrupted upto δ fraction of bits when compared to C(x)

Hadamard Code

Here $N = 2^n, q = 2$ and the mapping is defined as:
$C(x)_z = x.z \bmod 2$ where z $\in \{0,1\}^n$

To decode x_i from a N bit string y, following procedure is adopted:
Pick any random z $\in \{0,1\}^n$ and query the z and $z \oplus e_i$ bits of y and compute the xor of these bits, then we get:
$y_z \oplus y_{z \oplus e_i} = C(x)_z \oplus C(x)_{z \oplus e_i}$
if the queried bits are not corrupted which happens with a probability of $(1 - \delta)^2$

$x.z \oplus x.(z \oplus e_i) = x.z \oplus x.z \oplus x.e_i = x_i$

So we can decode x_i correctly with probability >= $1 - 2\delta$, Hence
Hadamard Code is $(2, \delta, \frac{1}{2} - 2\delta)$ LDC and encoded strings have exponential length.

Proof of Optimality of Hadamard Code:

Fact:
For every (q, δ, ϵ)-LDC C:$\{0,1\}^n -> \{0,1\}^N$, and for each $i \in [n]$, there exists a set M_i of $\Omega(\delta\epsilon N/q^2)$ disjoint tuples, each of at most q indices from [N], and a bit $a_{i,t}$ for each tuple t $\in M_i$, such that following holds for all n-bit strings x:

$$Pr\left[x_i = a_{i,t} \oplus \sum_{j \in t} C(x)_j\right] >= 1/2 + \Omega(\epsilon/2^q)$$

Now take any 2-query LDC, quantum method helps us to lower bound the value of codelength N

Consider the following quantum random access code:

$$x|- > |\phi_x\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} (-1)^{C(x)_j} |j\rangle$$

Now we want to recover the x_i from this state $|\phi_x\rangle$
So we convert the tuples from M_i into measurement operators: for each (j,k) $\in M_i$, define $P_{jk} = |j\rangle\langle j| + |k\rangle\langle k|$ and define $P_{rest} = \sum_{i \notin \cup t \in M_i} |i\rangle\langle i|$

So P_{jk} corresponds to the states we would want to see upon measurement.

The probability to see the (good)state corresponding to P_{jk} for a particular (j,k) is 2/N but there are $|M_i| = \Omega(\delta\epsilon N)$ such pairs, so probability to see any of the required (j,k) is $\frac{2}{N}\Omega(\delta\epsilon N) = \Omega(\delta\epsilon)$ and we get the state corresponding to P_{rest} with probability $1 - \Omega(\delta\epsilon)$

Measurement using the above projectors will yield:

The state

$$\frac{1}{\sqrt{2}}\Big((-1)^{C(x)_j}|j\rangle + (-1)^{C(x)_k}|k\rangle\Big) = \frac{(-1)^{C(x)_j}}{\sqrt{2}}\Big(|j\rangle + (-1)^{C(x)_j \oplus C(x)_k}|k\rangle\Big)$$

for a particular (j,k) with probability 2/N and again measuring this state in basis $(1/\sqrt{2})(|j\rangle + |k\rangle)$ and $(1/\sqrt{2})(|j\rangle - |k\rangle)$

This will always give us value of $C(x)_j \oplus C(x)_k$ and then the randomized algorithm will decode it with probability at least $1/2 + \Omega(\epsilon)$
If we do not get one of the good states(with probability=1-$\Omega(\delta\epsilon)$), then we flip a fair coin to decide x_i.

So the success probability of recovering x_i is:

$$p >= \frac{1}{2}(1 - \Omega(\delta\epsilon)) + \Omega(\delta\epsilon)(\frac{1}{2} + \Omega(\epsilon)) = \frac{1}{2} + \Omega(\delta\epsilon^2)$$

So $\log N >= n(1 - H(1/2 + \Omega(\delta\epsilon^2))) \rightarrow N = 2^{\Omega(n\delta^2\epsilon^4)}$

So this means for any two query LDC, the codelength N must be exponential in n

PP and PostBQP:

The complexity class PP consists of all languages L for which there exists a probabilistic polynomial time algorithm that accepts input x with probability at least 1/2 if x $\in$ L and with probability less than 1/2 if x $\notin$ L.

BQP is the class of languages computable with bounded error by a uniform family of polynomial-size quantum circuits with error probability at most 1/3.

The Complexity class PostBQP is the class of languages computable with bounded by a uniform family of polynomial-size quantum circuits that is allowed to post-select that is it can decide to do different steps depending upon the outcome of a particular measurement.

Aaronson showed that PP complexity is actually same as the PostBQP. Closure properties of PP follow from closure properties of PostBQP.

In the paper, it has been shown that PP $\subseteq$ PostBQP by giving a quantum circuit that uses postselection for accepting a language belonging to PP.

Suppose L $\in$ PP, then let M be a probabilistic polynomial time algorithm that uses m = poly(n) random bits on an input x of length n. So depending on the random string, the algorithm decides whether to output 0(rejects) or 1(accepts).

This defines a function $g_x : \{0,1\}^m \to \{0,1\}$

Let $s = |g^{-1}(1)|$, then by definition of PP: $s >= 2^{m-1} \iff x \in L$

Aaronson gave a postselection quantum circuit to determine $s >= 2^{m-1}$ or not.
The algorithm has two parts. The first one is as follows:

A1. Initialize a $|0^{m+1}\rangle$ qubit register. Apply $H^{\otimes m} \otimes I_2$ and then compute the function using oracle, which gives the state:

$$\frac{1}{\sqrt{2^m}} \sum_x |x\rangle|g(x)\rangle$$

A2. Again apply Hadamard on the first m qubits to get

$$\frac{1}{\sqrt{2^m}} \sum_x \Big(\frac{1}{\sqrt{2^m}} \sum_w (-1)^{w.x} |w\rangle \Big) |g(x)\rangle$$

Now the state where first m qubits are zero in the superposition is:

$$\frac{1}{2^m} \sum_x |0^m\rangle|g(x)\rangle = \frac{1}{2^m}|0^m\rangle((2^m - s)|0\rangle + s|1\rangle)$$

A3. Now postselect on firts m qubits being zero, we have the state:

$$|\psi\rangle = \frac{(2^m - s)|0\rangle + s|1\rangle}{\sqrt{(s^2 + (2^m - s)^2}}$$

Now the second part is as follows: Let (α, β) be a pair of positive reals such that $\alpha^2 + \beta^2 = 1$

B1. Prepare a qubit in the state: $|z\rangle = \alpha|0\rangle + \beta|1\rangle$.

Perform controlled Hadamard on $|\psi\rangle$ using $|z\rangle$ as control qubit and we get the state:
$\alpha|0\rangle|\psi\rangle + \beta|1\rangle H|\psi\rangle$ which makes the state:

$$H|\psi\rangle = \frac{\frac{1}{\sqrt{2}}(2^m|0\rangle + (2^m - 2s)|1\rangle)}{\sqrt{(s^2 + (2^m - s)^2}}$$

B2. Postselect on second qubit measuring to 1 to get the following state(unnormalized):

$$|\phi\rangle = s\alpha|0\rangle + \beta\frac{1}{\sqrt{2}}(2^m - 2s)|1\rangle$$

Now perform a projective measurement relative to Hadamard Basis.

Perform steps B1-B2, for different choices for $\{\alpha_i, \beta_i\}_{-m}^{+m}$ chosen to satisfy $\alpha_i = \beta_i 2^i$ and for each i upto O(log m) trials to estimate the probability of getting the state $|+\rangle$.

If $s >= 2^{m-1}$ then $2s >= 2^m$ that mean that the state lies in the 4th quadrant and will never be close to the $|+\rangle$ and probability of measuring in $|+\rangle$ is at max 1/2 whereas if it it is not the case, then for suitable choice of α and β, it will be very close to $|+\rangle$ and probability of measuring in $|+\rangle$ will be near to 1, so we will be able to estimate with high probability whether $s >= 2^{m-1}$ or not. So we can conclude whether the input belongs to the language or not with high probability. Thus PP $\subseteq$ PostBQP.

ϵ approximating polynomials for symmetric boolean functions:

A symmetric boolean function map f:$\{0,1\}^n \to \{0,1\}$ such that the value depends only on the number of 1's that is the hamming weight of input.
For some specified approximation error ϵ, $\deg_\epsilon$(f) denotes the minimal degree of all n-variate multilinear polynomials p satisfying $|p(x) - f(x)| <= \epsilon \ \forall \ x \in \{0,1\}^n$. Paturi characterized the 1/3-error approximate degree: if t $\in$ (0,n/2] is the smallest integer such that f is constant for $|x| \in \{t, t+1, ..., n-t\}$, then $\deg_\epsilon$(f) = $\Theta(\sqrt{tn})$

de Wolf gave the characterization of $\deg_\epsilon$(f) using a quantum query algorithm which is as follows:
Let t=t(f) as in paturi's result.
1. Use t-1 applications of Grover's search to try to find upto t-1 1's in x assuming t 1's in the beginning and crossing out indices that we have already found to be 1. This costs $\sum_{i=1}^{t-1} O(\sqrt{n/i}) = O(\sqrt{tn})$ = O($\deg_{1/3}$(f)).

2. Now use $\epsilon/2-$error Grover search to try to find one more solution. This costs $O(\sqrt{n\log(1/\epsilon)})$ queries.

3. Perform step 1, now checking for 0's instead of 1's
4. Perform step 2, now checking for 0's instead of 1's

5. If step 2 did not find another 1, output the corresponding value of f(x)
Else if step 4 did not find another zero, output the corresponding value of f(x),
Else assume $|x| \in \{t, t+1, ..., n-t\}$ and output f(x).

Now the query complexity of the algorithm is O($deg_{1/3}(f) + \sqrt{n\log(1/\epsilon)}$)

If $|x| < t$, then step 1 will find all 1's and f(x) is computed with zero error
If $|x| > n-t$, then step 2 finds another 1 with probability at least 1-$\epsilon/2$ and step 4 will not find another zero, so f(x) is computed with error probability at most $\epsilon/2$,
Otherwise, step 2 and step 4 will find another 1 and 0 respectively, with probabilities at least 1-$\epsilon/2$ each, so we can detect $|x| \in \{t, t+1, ..., n-t\}$ with probability 1-ϵ. So in this case f(x) is computed with error probability at most ϵ.

Since it took O($deg_{1/3}(f) + \sqrt{n\log(1/\epsilon)}$) queries to compute f with error probability at most ϵ, the approximating polynomial's degree $deg_\epsilon(f) = O(deg_{1/3}(f) + \sqrt{n\log(1/\epsilon)})$