Lecture 9: Hidden subgroup problem

Rajat Mittal

IIT Kanpur

Shor's algorithm for integer factorization is one of the leading results in the field of quantum computing. Whenever someone talks about quantum computing, this is the first algorithm that comes to mind. It showed that a very natural mathematical problem (factoring) can be solved in polynomial time on a quantum computer; no such efficient classical algorithm is known for this problem.

We will go sequentially, first we will see Simon's algorithm which laid the foundation for Shor's algorithm. It will be generalized to an algorithm for *hidden subgroup problem (HSP)* over finite Abelian groups (Simon's problem is an HSP with group $\mathbb{Z}_2^n$).

The quantum part of Shor's algorithm can be seen as solving HSP over the group of integers $\mathbb{Z}$ (notice that this group is not finite). After looking at various building blocks, we will be prepared to look at Shor's algorithm.

1 Simon's algorithm

We saw two quantum algorithms in the last few lectures. Deutsch-Jozsa provided exponential separation between quantum and classical computers in the deterministic setting. Bernstein-Vazirani showed speed-up in the randomized setting (1 quantum query vs $\log n$ randomized queries, where n is the size of the input).

The next algorithm, *Simon's algorithm*, will give exponential separation in the randomized setting ($\log n$ as compared to $\Omega(n)$). Again, the speed-up will be in the query model.

Input: A string of length n, $x = (x_1, x_2, \cdots, x_n)$, where each x_i (not just i) is a k bit binary string ($n = 2^k$). You can also think of it as a function f from $\{0,1\}^k$ to $[n]$. So, x_i gives the function value $f(i)$ as before. Here, we use i for both, as an index between 1 to n and as a binary string of length k.

Promise: There exists a hidden shift $s \in \{0,1\}^k$ such that $x_i = x_j$ *if and only if* $i = j \oplus s$. Again, notice that when we write $i = j \oplus s$, i, j, s are viewed as binary strings and *XOR* is taken entrywise. On the other hand, when writing x_i, i corresponds to the index (a number) between 1 and n.

Output: We need to find the hidden shift s.

Note 1. The difference from previous settings is that the function range is $[n]$ instead of $\{0,1\}$. Actually, the range is not that important, it should be big enough to make sense of the promise.

We will assume that one query to the input gives the entire string x_i. The query oracle is in the standard form, $O_x|i,0\rangle \to |i,x_i\rangle$, where both parts of the register have k qubits.

Note 2. Even if we assume that there are $k2^k$ indices and every query outputs a bit, the exponential separation will not be affected.

Exercise 1. Why is the above statement true?

We will begin with our usual strategy of creating a superposition over all inputs and computing their function value in superposition.

So, the algorithm starts in the state $|0,0\rangle$, where both $|0\rangle$'s are in the state space of k qubits. Applying Hadamard transform to the first part will create a superposition over all inputs,

$$\frac{1}{\sqrt{2^k}} \sum_{j=0}^{n-1} |j,0\rangle.$$

Now we query the oracle (of the input) and get,

$$\frac{1}{\sqrt{2^k}} \sum_{j=0}^{n-1} |j, x_j\rangle.$$

Notice that the second part of the register is same iff the indices in the first part differ by s. If we measure the second register at this stage, we get the state

$$\frac{1}{\sqrt{2}}(|j\rangle + |j \oplus s\rangle), \tag{1}$$

for some j.

We have ignored the second part of the register, since that part of the register has been measured. How is this state useful to us?

Another viewpoint about Simon's problem will be of help. Our input is on $\mathbb{Z}_2^n$ and the hidden shift s generates a subgroup K_s of order 2 in this group. The promise can be reinterpreted as, $x_i = x_j$ if and only if i and j belong to the same coset with respect to K_s. We need to find the subgroup (or its generator) K_s to solve Simon's problem.

Exercise 2. Convince yourself that the above statement is true.

So, after measurement, we get a state of the form given in Eq. 1, called *coset states*. What information can we derive from these coset states about K_s? Let us apply the only tool we have, Fourier transform on $\mathbb{Z}_2^n$.

Applying Hadamard transform (Fourier transform on $\mathbb{Z}_2^n$) on the coset state in Eq. 1,

$$\frac{1}{\sqrt{2^{k+1}}} \left(\sum_{l \in \{0,1\}^n} ((-1)^{j.l} + (-1)^{(j \oplus s).l}) |l\rangle \right).$$

Exercise 3. When will state $|l\rangle$ have amplitude non-zero?

Looking at the formula, it is non-zero if and only if $s.l = 0 \mod 2$. Measuring this state, we will get an l such that $s.l = 0 \mod 2$. You can easily check that each such l has an equal probability to appear after the measurement.

Note 3. l specifies a character of $\mathbb{Z}_2^n$ given by $\chi_l(a) = (-1)^{l.a}$. Intuitively, Fourier transform puts weight only on those characters which are trivial on H.

Applying Hadamard on the coset states and measuring, we get an l for which $s.l = 0 \mod 2$. So, one coset state gives us one linear equation for s. If we get $k-1$ *linearly independent* l's, we can find s. The obvious choice is to create lot ($O(k)$) of cosets states and hope that we have $k-1$ linearly independent l's, then s can be found using Gaussian elimination.

Exercise 4. What is Gaussian elimination. What field are we working over?

The only thing to show is, if we have $O(k)$ coset states then we get $k-1$ linearly independent l's with high probability.

Exercise 5. Suppose we have h linearly independent l's. What is the probability that we get a linearly independent l in the next coset state measurement?

Note 4. We are working over the field $\mathbb{F}_2$.

This probability is at least half, because the vector space spanned by h vectors will cover 2^h vectors out of 2^k vectors. Hence, at every step, we get a linearly independent vector with probability at least $\frac{1}{2}$. So in $O(k)$ iterations, we will have $k-1$ linearly independent vectors with high probability.

Exercise 6. Make the previous statement precise and prove it.

Hence with $O(k)$ queries (since it takes one query to prepare a coset state) we can give a randomized algorithm for Simon's problem on a quantum computer.

Simon [2] also showed that any randomized algorithm will take take at least $\Omega(\sqrt{2^k})$ queries to solve this problem with high probability (calculations similar to birthday paradox). Hence, we get an instance where quantum and classical query complexity differ exponentially. Notice, this does not imply exponential separation in the usual circuit model.

2 Hidden subgroup problem (HSP)

The ideas of this section are heavily borrowed from Andrew Childs' course notes [1]. The strategy for the algorithms seen till now is very similar. Most of these problems can be framed in terms of *Hidden subgroup problem*.

For a hidden subgroup problem, we are given a group G and a function which is constant on the cosets of G with respect to some subgroup H. That means, any two elements of the same coset have same function value and any two elements of two different cosets have different value. In this case too, the function is given as a blackbox, i.e., there is an oracle to find the value of a function on any element of G.

The problem is to find the hidden subgroup H (or its generators, subgroup might be too big to output). In some cases, the hidden subgroups are restricted, that means, they only come for a subset of all possible subgroups.

Exercise 7. Read about cosets of a group.

Definition 1 (Hidden Subgroup Problem). *For a group G and its subgroup H (hidden), we are given a function f (as oracle), constant on cosets and different on different cosets of H. Find generators of H.*

Some of the problems we have covered can be seen as examples of hidden subgroup problem.

If we look at Deutsch's problem, it is a hidden subgroup problem with respect to the group $\mathbb{Z}_2$. When the function is constant it corresponds to the entire group being the subgroup. The balanced case is when $H = \{0\}$ is the subgroup.

In the assignment you will prove that Simon's problem is an example of HSP on $\mathbb{Z}_2^n$ with restricted subgroups. The strategy to solve these problems can be generalized to many more groups.

In particular, we know how to solve the HSP on finite abelian groups using generalization of techniques from Simon's algorithm. You might have guessed that Fourier transform is going to be an important tool for this.

QFT over a finite Abelian group:
One of the main tools to solve HSP for a finite Abelian group G is the quantum Fourier transform (QFT) over G.

Exercise 8. If you are not comfortable with groups and characters of a finite Abelian group, please read about these concepts.

Remember that a character is a function $\chi : G \to \mathbb{C}$ such that

- $|\chi(g)| = 1$ for all $g \in G$,
- and $\chi(g)\chi(h) = \chi(gh)$ for all $g, h \in G$.

The set of characters of a group G themselves form a group and that group is denoted by $\hat{G}$. These characters form an orthonormal basis of the space $\mathbb{C}^{|G|}$.

The Fourier transform over G converts a function in the standard basis to the character basis. Let F_G denote the QFT over G, we specify F_G by its action on the basis.

$$F_G|x\rangle = \frac{1}{\sqrt{|G|}} \sum_{y \in \hat{G}} \chi_y(x)|y\rangle \quad \forall x \in G.$$

Exercise 9. Show that the expression above computes the Fourier transform of x.

We have only learnt to apply QFT for $\mathbb{Z}_{2^k}$ in previous lectures. This can be extended (non-trivial) to QFT over general $\mathbb{Z}_n$. The QFT for a general group is obtained by taking tensor product of these QFT's over $\mathbb{Z}_n$ (follows from decomposition of any finite Abelian group). We will skip those details, interested reader can refer to [1]. For the rest of this lecture, we will assume access to QFT over any finite Abelian group.

The idea behind the algorithm to solve HSP is described below, try to fill the details on your own. For full algorithm, see Andrew Childs course notes [1].

Solving HSP over a finite Abelian group:

We give an outline of the algorithm to solve HSP over a finite Abelian group. Notice the similarity between Simon's algorithm and the algorithm given below. A similar idea is going to work for factorization (Shor's algorithm). The steps to solve HSP on a finite Abelian group G is,

1. *Starting state:* Start with the state $|0, 0\rangle$, where the first part (register) is for the domain of f (group elements) and second register is for the range of the function f. In particular, first register will have $\log(|G|)$ qubits.
2. *Equal superposition:* Create an equal superposition over the group elements, resulting state is
$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g, 0\rangle.$$
3. *Applying function oracle:* Apply the function oracle, resulting state is
$$\frac{1}{\sqrt{|G|}} \sum_{g \in G} |g, f(g)\rangle.$$
Notice that this is the strategy adopted in almost all the algorithms you have seen till now. After this step, we have the function value of all the elements of G in superposition. This is a highly entangled state.
4. *Creating coset states:* From the definition of HSP problem, second register contains the same value if the value in the first register belongs to the same coset. By measuring the second register, we pick one of the function values, and resulting state is an equal superposition over all elements of the corresponding coset.
$$|gH\rangle := \frac{1}{\sqrt{|H|}} \sum_{h \in H} |gh\rangle$$
The new state mentioned above is only on the first register, we have measured (and discarded) the second register. These states are known as *coset states* (similar to the definition of coset states in Eq. 1, those states were defined for a particular subgroup of $\mathbb{Z}_2^n$).
5. *Apply QFT over G:* To obtain information from a coset state, we apply quantum Fourier transform (QFT) over the group G. After applying the Fourier transform, we get
$$|\widehat{gH}\rangle = \frac{1}{\sqrt{|H||G|}} \sum_{y \in \hat{G}} \sum_{h \in H} \chi_y(gh)|y\rangle.$$
Exercise 10. How did we get the above state?
6. *Rewrite the state:* This is not really a step in the quantum algorithm, but a simplification of the state for our understanding. The above state can be simplified to,
$$\sqrt{\frac{|H|}{|G|}} \sum_{y \in \hat{G}} \chi_y(g) \left(\frac{1}{|H|} \sum_{h \in H} \chi_y(h) \right) |y\rangle.$$
Since χ_y is a character of H too (why?), the quantity in parenthesis gives zero except when χ_y is trivial on H.

Exercise 11. What is the quantity in parenthesis when χ_y is trivial on H?

So, the state reduces to,

$$\sqrt{\frac{|H|}{|G|}} \sum_{y \in \hat{G}: \chi_y(h)=1 \ \forall h \in H} \chi_y(g)|y\rangle.$$

We get an equal superposition over characters which are trivial on H. For Simon's algorithm, we got l such that $s.l = 0$ for the hidden shift s. In that case, hidden s corresponds to a hidden subgroup and l corresponds to a character.

7. *Measure the first register:* The measurement gives rise to character of G which is trivial on H. It can be argued that this measurement does not result in lose of information, see [1]. This justifies our choice of QFT over G and measurement.
8. *Classical post-processing:* If we repeat the above process multiple times, we will get multiple characters which are trivial on H. That means, the kernel for each of these characters contain H. The quantum part of the algorithm has finished. We need to find H using this classical information: a string of characters trivial on H where each entry is picked independently.

   Exercise 12. Given multiple such characters, what heuristic you can follow to get H?

   It seems that an intersection of the kernels of such characters should give H. The intersection will contain H (why?), can it be much bigger?
   We can show that the size of the intersection will decrease with every new sample (character trivial on H). After $O(\log(|G|))$ such samples, the intersection will become H with high probability. The last two statements requires a proof and follows from group properties [1].
   You can again observe the similarity with the last step of Simon's algorithm. We get lots of l's such that $s.l = 0$. Each new l takes us closer to the hidden shift s.

If you closely see the Fourier transform of coset states, it shows one more property: amplitudes on Fourier transform of two coset states are same except a phase of $\chi_y(g)$. In that sense, it does not matter which coset state we get after the first measurement. The measurement statistics will be the same for any g.

Exercise 13. How is this similar to Simon's algorithm? What is the group and what are the subgroups?

Exercise 14. What is the Fourier transform over $\mathbb{Z}_2^n$.

The HSP on non-abelian groups, like dihedral and symmetric groups, have various applications. Unfortunately, no efficient algorithm to solve these, even on a quantum computer, is known.

Note 5. Most of the known techniques (not efficient) for non abelian HSP's also produce coset states and then do classical/quantum post-processing on them.

3 Assignment

Exercise 15. Show that Simon's problem is the Hidden subgroup problem on $\mathbb{Z}_2^n$.

Exercise 16. Suppose in the Simon's problem, $x_j = x_i$, iff there exist $v \in V$ for which $j = i \oplus v$. Here V is a subspace of $\mathbb{Z}_2^n$. What l will we get at the end of the Simon's algorithm?

Hint: The usual case corresponds to the one dimensional subspace.

Exercise 17. Read about subgroups, cosets and abelian groups.

Exercise 18. What is the size of $\hat{G}$? Can you prove it? Is there an easy way to see its size?

References

1. A. Childs. Quantum algorithms, 2013.
2. D. R. Simon. On the power of quantum computation. *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on: 116–123*, pages 116–123, 1994.