Lecture 8: Grover's Quantum Search Algorithm

Dr Iain Styles: I.B.Styles@cs.bham.ac.uk

Introduction

Last lecture, we:

- showed how to perform the Quantum Fourier Transform

In this lecture we will:

- introduce Grover's search algorithm
- discuss how Grover's algorithm performs in comparison with classical algorithms

The Search Problem

- Searching is an important task that modern computers can help us with
- Unfortunately, they're not very good at it!
- Classically, the only way to do a search is to systematically examine all the possibilities until you find the solution
- Clearly if the search space has N entries, then the time taken to complete a search is $O(N)$ (on average, $N/2$)
- No classical algorithm can do better than this
- Quantum search algorithms can do better than $O(N)$
- Grover's algorithm works in time $O(\sqrt{N})$
- This is the best that a quantum computer can do
- For large N, this could yield very large performance increases
- The key idea is that although finding a solution to the search problem is hard, *recognising* a solution is easy

Setting up the problem

- We wish to search through a list of N elements, y_x
- Each element has an index x in the range $0 \mapsto N-1$
- Assume $N = 2^n$, so we can store x in n qubits
- The search problem has M solutions, where $1 \leq M \leq N$
- Rather than dealing with the list itself, we focus on the *index* of the list, x
- The key idea is that given some value of x, we can tell whether y_x solves the search problem
- We assume that we can construct some device to tell us if y_x solves the search problem
- This device is called an **Oracle**

The Oracle

- The Oracle is simply a device with the ability to recognise solutions to the search problem
- When given an index x, the Oracle raises a flag if y_x solves the search problem
- For a given search problem, we can define some function $f(x)$ such that $f(x) = 1$ if y_x solves the search problem, and $f(x) = 0$ if it doesn't
- The Oracle takes as input an index value in a qubit register $|x\rangle$
- It also takes a single "Oracle qubit", $|q\rangle$
- The state given to the Oracle is thus $|\psi\rangle = |x\rangle |q\rangle$
- The Oracle is represented by a unitary operator, O
- If x indexes a solution to the search problem, O sets $f(x) = 1$, and $f(x) = 0$ if it doesn't index a solution
- If $f(x) = 1$, the Oracle flips the state of $|q\rangle$

The Oracle

- We write this as $O|x\rangle|q\rangle = |x\rangle X^{f(x)}|q\rangle$
- X is just our quantum NOT operator
- So if $f(x) = 1$, $|q\rangle \mapsto X|q\rangle$, else $|q\rangle \mapsto |q\rangle$
- We will choose to initially program $|q\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$
- Then $X|q\rangle = \frac{1}{\sqrt{2}}(-|0\rangle + |1\rangle) = -|q\rangle$
- And $O|x\rangle|q\rangle = (-1)^{f(x)}|x\rangle|q\rangle$
- The Oracle therefore takes $|x\rangle \mapsto (-1)^{f(x)}|x\rangle$
- So the term indexing the solution is marked with a $-$ sign

An Example of the Oracle

- We prepare a 2-qubit system in state $|\psi\rangle = \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle)$
- The solution to the search problem is given by $x = 2$
- So $O|\psi\rangle|q\rangle = \frac{1}{2}(|0\rangle + |1\rangle - |2\rangle + |3\rangle)$
- The Oracle does **not** find the solution to the problem
- It simply recogises the answer when it is presented to it
- So different search problems need different Oracles
- The key to quantum search is that we can look at all solutions simulataneously: the Oracle just manipulates the state coefficients using a unitary operator!
- But the Oracle by itself is not sufficient
- We can't find out what the answer is!

The Grover Iteration

- When we measure $|x\rangle$ after the Oracle has been applied, we won't be able to see the sign change
- What we want is to be able to measure $|x\rangle$ and get the index to the solution of the search problem ($x = 2$ in our example) with high probability
- The Grover iteration is a technique for achieving this result
- The essence of Grover is to ensure that when we measure the state of $|x\rangle$, the probability of getting the right answer is maximised
- Grover realised that this could be done by doing some rather simple manipulations on $|\psi\rangle$, and applying the QFT multiple times
- Here's how it works:

Grover's Algorithm

1. Begin with $|x\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle$
2. Apply the Oracle to $|x\rangle$:

$$|x\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} (-1)^{f(x)} |j\rangle$$

3. Apply the QFT to $|x\rangle$
4. Reverse the sign of all terms in $|x\rangle$ except for the term $|0\rangle$
5. Apply the Inverse QFT
6. Return to step 2 and repeat

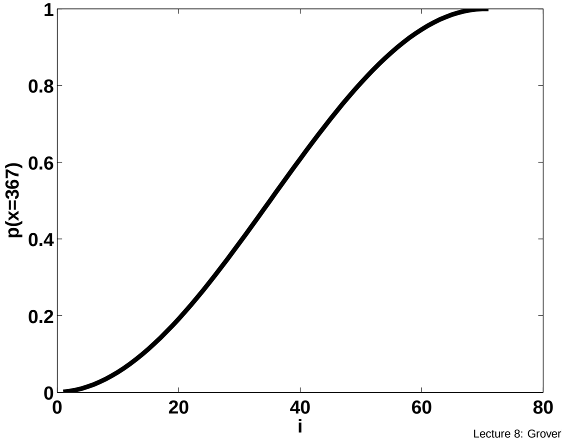
A Very Simple Example

Apply Grover's algorithm to $N = 4$ with solution $x = 2$

- We start with $|x\rangle = \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle)$
- Apply the Oracle: $|x\rangle \mapsto \frac{1}{2}(|0\rangle + |1\rangle - |2\rangle + |3\rangle)$
- Apply the QFT: $F\,|x\rangle = \frac{1}{2}(|0\rangle + |1\rangle - |2\rangle + |3\rangle)$
- Flips signs of all terms except $|0\rangle$: $F\,|x\rangle \mapsto \frac{1}{2}(|0\rangle - |1\rangle + |2\rangle - |3\rangle)$
- Inverse QFT: $|x\rangle = |2\rangle$
- So when we measure $|x\rangle$ we are guaranteed the right answer!
- This is a slightly artificial example
- Try a more complex example: search a list of $N = 8192$ items where the solution is indexed by $x = 367$
- Write a program to do this!

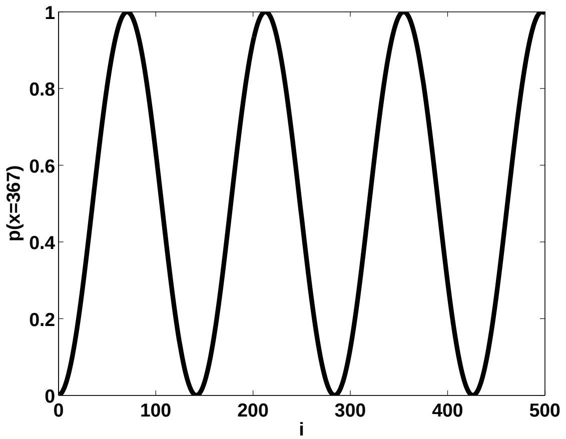
More Grover

- We construct the state $|x\rangle = \frac{1}{\sqrt{8192}} \sum_{j=0}^{8191} |j\rangle$
- Then run the Grover iteration and keep track of $p(367)$
- Running for 71 iterations, we get:



Some Problems

- So if we measure $|x\rangle$ after ≈ 71 iterations, we get the right answer
- But if we let it run for too long (say, 500 iterations)...



Performance of Grover's Algorithm

- The point at which we terminate Grover's algorithm and measure the result is critical
- It has been shown that the optimum number of iterations is $\approx \frac{\pi}{4}\sqrt{\frac{N}{M}}$, where M is the number of solutions
- It has also been shown that this is the best that any quantum search algorithm can do
- It's much better than classical search algorithms which take $O(N)$ steps
- So there are huge potential benefits when searching very large data sets
- Note that we could solve any problem where finding answers is hard, but recognising them is easy
- As long as we can construct an Oracle, we can use Grover's algorithm

Conclusions

In this lecture we have:

- Defined the concept of an Oracle as something which can recognise the solution to a problem
- Shown how Grover's algorithm uses the Oracle to search a list in $O(\sqrt{N})$ operations
- Pointed out some problems with Grover's algorithm: taking the result at the right time

Next lecture we will:

- Introduce some key ideas in Quantum Communication