Quantum algorithms (CO 781/CS 867/QIC 823, Winter 2013)
Andrew Childs, University of Waterloo

LECTURE 12: Unstructured search

Now we begin to discuss applications of quantum walks to search algorithms. We start with the most basic of all search problems, the unstructured search problem (which is solved optimally by Grover's algorithm). We discuss how this problem fits into the framework of quantum walk search, and also describe amplitude amplification and quantum counting in this setting. We also discuss quantum walk algorithms for the search problem under locality constraints.

Unstructured search

In the unstructured search problem, we are given a black box function $f\colon S \to \{0,1\}$, where S is a finite set of size $|S| = N$. The inputs $x \in M$, where $M := \{x \in S\colon f(x) = 1\}$, are called *marked items*. In the decision version of the problem, our goal is to determine whether M is empty or not. We might also want to find a marked item when one exists.

It is quite easy to see that even the decision problem requires $\Omega(N)$ classical queries, and that N queries suffice, so the classical query complexity of unstructured search is $\Theta(N)$.

You should already be familiar with Grover's algorithm, which solves this problem using $O(\sqrt{N})$ quantum queries. Grover's algorithm works by starting from the state $|S\rangle := \sum_{x\in S} |x\rangle/\sqrt{N}$ and alternately applying the reflection about the set of marked items, $\sum_{x\in M} 2|x\rangle\langle x| - 1$, and the reflection about the state $|S\rangle$, $2|S\rangle\langle S| - 1$. The former can be implemented with two quantum queries to f, and the latter requires no queries to implement. It is straightforward to show that there is some $t = O(\sqrt{N/|M|})$ for which t steps of this procedure give a state with constant overlap on $|M\rangle$ (assuming M is non-empty), so that a measurement will reveal a marked item with constant probability.

It can be shown that unstructured search requires $\Omega(\sqrt{N/|M|})$ queries. We will prove this when we discuss adversary lower bounds.

Quantum walk algorithm

Consider the discrete-time random walk on the complete graph represented by the stochastic matrix

$$P = \frac{1}{N-1}\begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 1 & \cdots & 1 \\ 1 & 1 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \tag{1}$$

$$= \frac{N}{N-1}|S\rangle\langle S| - \frac{1}{N-1}I. \tag{2}$$

It has eigenvalues 1 (which is non-degenerate) and $-1/(N-1)$ (with degeneracy $N-1$). Since the graph is highly connected, its spectral gap is very large: we have $\delta = 1 - \frac{1}{N-1} = \frac{N}{N-1}$.

This random walk gives rise to a very simple classical algorithm for unstructured search. In this algorithm, we start from a uniformly random item and repeatedly choose a new item uniformly at random from the other $N-1$ possibilities, stopping when we reach a marked item. The fraction of marked items is $\epsilon = |M|/N$, so the hitting time of this walk is

$$O\left(\frac{1}{\delta\epsilon}\right) = \frac{(N-1)N}{N|M|} = O(N/|M|) \tag{3}$$

(this is only an upper bound on the hitting time, but in this case we know it is optimal). Of course, if we have no a priori lower bound on $|M|$ in the event that M is non-empty, the best we can say is that $\epsilon \geq 1/N$, giving a running time $O(N)$.

The corresponding quantum walk search algorithm has a hitting time of

$$O\left(\frac{1}{\sqrt{\delta\epsilon}}\right) = O(\sqrt{N/|M|}), \tag{4}$$

corresponding to the running time of Grover's algorithm. To see that this actually gives an algorithm using $O(\sqrt{N/|M|})$ queries, we need to see that a step of the quantum walk can be performed using only $O(1)$ quantum queries. In the case where the first item is marked, the modified classical walk matrix is

$$P' = \frac{1}{N-1}\begin{pmatrix} N-1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ 0 & 1 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 0 & 1 & \cdots & 1 & 0 \end{pmatrix}, \tag{5}$$

so that the vectors $|\psi_j\rangle$ are $|\psi_1\rangle = |1,1\rangle$ and $|\psi_j\rangle = |j, S\setminus\{j\}\rangle = \sqrt{\frac{N}{N-1}}|j,S\rangle - \frac{1}{\sqrt{N-1}}|j,j\rangle$ for $j = 2,\ldots,N$. With a general marked set M, the projector onto the span of these states is

$$\Pi = \sum_{j\in M} |j,j\rangle\langle j,j| + \sum_{j\notin M} |j,S\setminus\{j\}\rangle\langle j,S\setminus\{j\}|, \tag{6}$$

so the operator $2\Pi - 1$ acts as Grover diffusion over the neighbors when when the vertex is unmarked, and as a phase flip when the vertex is marked. (Note that since we start from the state $|\psi\rangle = \sum_{j\notin M}|\psi_j\rangle$, we stay in the subspace of states $\mathrm{span}\{|j,k\rangle\colon (j,k)\in E\}$, and in particular have zero support on any state $|j,j\rangle$ for $j \in V$, so $2\Pi - 1$ acts as -1 when the first register holds a marked vertex.) Each such step can be implemented using two queries of the black box, one to compute whether we are at a marked vertex and one to uncompute that information; the subsequent swap operation requires no queries. Thus the query complexity is indeed $O(\sqrt{N/|M|})$.

This algorithm is not exactly the same as Grover's; for example, it works in the Hilbert space $\mathbb{C}^N \otimes \mathbb{C}^N$ instead of $\mathbb{C}^N$. Nevertheless, it is clearly closely related. In particular, notice that in Grover's algorithm, the unitary operation $2|S\rangle\langle S| - 1$ can be viewed as a kind of discrete-time quantum walk on the complete graph, where in this particular case no coin is necessary to define the walk.

The algorithm we have described so far only solves the decision version of unstructured search. To find marked item, we could use bisection, but this would introduce a logarithmic overhead. In fact, it can be shown that the final state of the quantum walk algorithm actually encodes a marked item when one exists.

Amplitude amplification and quantum counting

We briefly mention some other concepts related to unstructured search that provide useful tools for quantum algorithms in general. These ideas are typically presented in the context of Grover's algorithm; he were describe them in the framework of quantum walk search. This is slightly less space efficient, but the essential ideas are the same.

Amplitude amplification is a general method for boosting the success probability of a (classical or quantum) subroutine. It can be implemented by quantum walk search as follows. Suppose we have a procedure that produces a correct answer with probability p (i.e., with an amplitude of magnitude $\sqrt{p}$ if we view it as a quantum process). From this procedure we can define a two-state Markov chain that, at each step, moves from the state where the answer is not known to the state where the answer is known with probability p, and then remains there. This walk has the transition matrix

$$P' = \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix},$$

so $P_M = 1-p$, giving a quantum hitting time of $O(1/\sqrt{1-\|P_M\|}) = O(1/\sqrt{p})$.

For some applications, it may be desirable to estimate the value of p. Quantizing the above two-state Markov chain gives eigenvalues in the non-marked subspace of $e^{\pm i \arccos(1-p)} = e^{\pm i\sqrt{2p}+O(p^{3/2})}$. By applying phase estimation, we can determine $\sqrt{p}$ aproximately. Recall that phase estimation gives an estimate with precision μ using $O(1/\mu)$ applications of the given unitary (assuming we cannot apply high powers of the unitary any more efficiently than simply applying it repeatedly). An estimate of $\sqrt{p}$ with precision μ gives an estimate of p with precision $\mu\sqrt{p}$ (since $(\sqrt{p}+O(\mu))^2 = p + O(\mu\sqrt{p})$), so we can produce an estimate of p with precision ν in $O(\sqrt{p}/\nu)$ steps.

In particular, if the Markov chain is a search of the complete graph as described the previous section, with $|M|$ marked sites out of N, then $p = |M|/N$, and this allows us to count the number of marked items. We obtain an estimate of $|M|/N$ with precision ν in $O(\sqrt{|M|/N}/\nu)$ steps. If we want a multiplicative approximation of $|M|$ with precision ρ, this means we need $O(\sqrt{N/|M|}/\rho)$ steps.

Note that for exact counting, no speedup is possible in general. If $|M| = \Theta(N)$ then we need to estimate p within precision $O(1/N)$ to uniquely determine $|M|$, but then the running time of the above procedure is $O(N)$. In fact, it can be shown that exact counting requires $\Omega(N)$ queries.

Search on graphs

We can also consider a variant of unstructured search with additional locality constraints. Suppose we view the items in S as the vertices of a graph $G = (S, E)$, and we require the algorithm to be local with respect to the graph. More concretely, we require the algorithm to alternate between queries and unitary operations U constrained to satisfy $U|j,\psi\rangle = \sum_{k\in j\cup\partial(j)} \alpha_k |k,\phi_k\rangle$ for any $j \in S$ (where the second register represents possible ancillary space, and recall that $\partial(j)$ denotes the set of neighbors of j in G).

Since we have only added new restrictions that an algorithm must obey, the $\Omega(\sqrt{N})$ lower bound from the non-local version of the problem still applies. However, it is immediately clear that this bound cannot always be achieved. For example, if the graph is a cycle of N vertices, then simply propagating from one vertex of the cycle to an opposing vertex takes time $\Omega(N)$. So we would like to know, for example, how far from complete the graph can be such that we can still perform the search in $O(\sqrt{N})$ steps.

First, note that any expander graph (a graph with degree upper bounded by a constant and second largest eigenvalue bounded away from 1 by a constant) can be searched in time $O(\sqrt{N})$. Such graphs have $\delta = \Omega(1)$, and since $\epsilon \ge 1/N$ when there are marked items, the quantum hitting time is $O(1/\sqrt{\delta\epsilon}) = O(\sqrt{N})$ (whereas the classical hitting time is $O(1/\delta\epsilon) = O(N)$).

There are also many cases in which a quantum search can be performed in time $O(\sqrt{N})$ even though the eigenvalue gap of P is non-constant. For example, consider the n-dimensional hypercube (with $N = 2^n$ vertices). Recall that since the adjacency matrix acts independently as σ_x on each coordinate, the eigenvalues are equally spaced, and the gap of P is $2/n$. Thus the general bound in terms of the eigenvalues of P shows that the classical hitting time is $O(nN) = O(N \log N)$. In fact, this bound is loose; the hitting time is actually $O(N)$, which can be seen by directly computing $\|P_M\|$ with one marked vertex. So there is a local quantum algorithm that runs in the square root of this time, namely $O(\sqrt{N})$.

Perhaps the most interesting example is the d-dimensional square lattice with N sites (i.e., with linear size $N^{1/d}$). This case can be viewed as having N items distributed on a grid in d-dimensional space. For simplicity, suppose we have periodic boundary conditions; then the eigenstates of the adjacency matrix are given by

$$|\tilde{k}\rangle := \frac{1}{\sqrt{N}} \sum_x e^{2\pi i k\cdot x/N^{1/d}} |x\rangle \tag{7}$$

where k is a d-component vector of integers from 0 to $N^{1/d} - 1$. The corresponding eigenvalues are

$$2 \sum_{j=1}^{d} \cos \frac{2\pi k_j}{N^{1/d}}. \tag{8}$$

Normalizing to obtain a stochastic matrix, we simply divide these eigenvalues by $2d$. The 1 eigenvector has $k = (0, 0, \ldots, 0)$, and the second largest eigenvalue comes from (e.g.) $k = (1, 0, \ldots, 0)$, with an eigenvalue

$$\frac{1}{d}\left(d - 1 + \cos \frac{2\pi}{N^{1/d}}\right) \approx 1 - \frac{1}{2d}\left(\frac{2\pi}{N^{1/d}}\right)^2. \tag{9}$$

Thus the gap of the walk matrix P is about $\frac{2\pi^2}{2dN^{2/d}} = O(N^{-2/d})$. This is another case in which the bound on the classical hitting time in terms of eigenvalues of P is too loose (it gives only $O(N^{1+2/d})$), and instead we must directly estimate the gap of P_M. One can show that the classical hitting time is $O(N^2)$ in $d = 1$, $O(N \log N)$ in $d = 2$, and $O(N)$ for any $d \ge 3$. Thus there is a local quantum walk search algorithm that saturates the lower bound for any $d \ge 3$, and one that runs in time time $O(\sqrt{N \log N})$ for $d = 2$. We already argued that there could be no speedup for $d = 1$, and indeed we see that the quantum hitting time in this case is $O(N)$.