CS 880: Quantum Information Processing	10/14/2010
Lecture 16: Quantum Walks	
Instructor: Dieter van Melkebeek	Scribe: Brian Nixon

In this lecture we start our examination of quantum walks, the quantum equivalent of classic random walks. There are two types of quantum walk: discrete time (which is mainly considered by computer scientists) and continuous time (which is mainly considered by physicists). Here we focus on the former and leave the latter for future lectures.

1 Classic Random Walks

A random walk is performed on a graph. Given a starting vertex s we move to a randomly chosen neighboring vertex. From there we can step again to a new neighbor and so on. An example is provided by a man who steps out onto the street and walks either one block north or one block south depending on the flip of a coin (here the graph is a line). We can ask questions about his position after t steps. Random walks can be modelled as Markov chains.

There are two big motivations for random walks.

1. We can model random processes with a walk (e.g. there is an algorithm to solve 2-SAT that can be analyzed as a random walk on a line).

2. It also forms the body for some algorithms (e.g. sampling methods for statistical physics). For such an algorithm we might ask under what conditions a stationary distribution might exist. If it exists, what is required for such a distribution to be unique and how quickly do walks converge to a stationary distribution (if ever)?

We first consider regular graphs (all vertices have same degree). Let A be the normalized adjacency matrix. The distribution at step $k+1$ is the vector $p_{k+1} = Ap_k$ where $p_k = (a_s)_{s \in G}$ and $\sum a_s = 1$, $a_s \geq 0 \forall s$. We set our initial position vector p_0 centered on a single node s as $a_s = 1$, $a_t = 0$ for $t \neq s$. A stationary distribution is represented by a vector v such that $Av = v$.

Exercise 1. *Prove the following propositions for the above A.*

1. *The uniform distribution is an eigenvector with eigenvalue* 1.

2. *There is a multiplicity of eigenvectors over* 1 *iff G is disconnected.*

3. -1 *is an eigenvalue iff G is bipartite.*

Definition 1. *The spectral gap is $\delta = \min(\{1 - |\lambda| | \exists x \neq u \text{ s.t. } Ax = \lambda x\})$ where u is the uniform distribution.*

Finding δ amounts to finding the greatest magnitude eigenvalue less than 1. Note if -1 is not an eigenvalue and there is only one eigenvector for 1 then we will have convergence to a unique stationary distribution at a speed dictated by the spectral gap.

Given an initial vector p, we want to know $(A^k p - u) = A^k(p - u)$. We note that A is a symmetric matrix so we can build an orthonormal basis of eigenvectors $\{e_i\}$ that includes the

normalized uniform distribution $\sqrt{N}u = e_1$ given that it is an eigenvector of 1. Now for any probability distribution p, $\langle p, u\rangle$ equals the sum over each entry multiplied by $\frac{1}{N}$. As the inner product is bilinear we get $\langle (p-u), u\rangle = \langle p, u\rangle - \langle u, u\rangle = \frac{1}{N} - \frac{1}{N} = 0$. Since $(p-u)$ has no component in the u direction we get $A^k(p-u) = A^k \sum_{i=2}^{N} \gamma_i e_i = \sum_{i=2}^{N} \gamma_i \lambda_i^k e_i$ where λ_i is the eigenvalue corresponding to e_i. Thus $||A^k p - u||_2^2 = \sum_{i=2}^{N} |\gamma_i \lambda_i^k|^2 \leq (1-\delta)^k \sum_{i=2}^{N} |\gamma_i|^2 = (1-\delta)^k ||p-u||_2^2 \leq (1-\delta)^2$. This last step is reached by noting orthogonality gives us $||p-u||_2^2 + ||u||_2^2 = ||p||_2^2$ and as this is the sum of probabilities squared (all elements of $[0,1]$) we must have $||p||_2^2 \leq 1$. . Considering the 1-norm, we want $||A^k p - u||_1 \leq \sqrt{N}||A^k p - u||_2 \leq \sqrt{N}(1-\delta)^k \leq \epsilon$. To guarantee this we need to choose $k \geq \frac{\log(\sqrt{N}/\epsilon)}{\log(1/(1-\delta))}$ for our running time. This is $\geq \Omega(\frac{1}{\delta}\log(\sqrt{N}/\epsilon))$ for small δ. There are graphs where this bound is tight. The important part to take away is the dependency on spectral gap is $\frac{1}{\delta}$.

2 Quantum Model

When adapting to quantum we again need to pick neighbors at random. If we use the adjacency graph A we get $|v\rangle \to |N_v\rangle = \frac{1}{N_v} \sum_{(v,w)\in E} |w\rangle$ where N_v is the neighborhood of v. This is not unitary (consider a square graph).

Our solution is to distinguish between picking a nieghbor and moving to it. Let our state be $|v, e\rangle$ where e is an edge containing v. In phase I, we will perform a unitary operation such that an observation afterwards would yield a random choice of edge. Now there is no interference as we've saved the v information and the states that would interfere operate in different subspaces. This is enough to be unitary. Call this transition C_v. In phase II, implement $S|v, e\rangle = |w, e\rangle$ where $e = (v, w)$. Note this transition is its own inverse. If we observe after each pair of steps, this method provides a classic random walk.

We have options in choosing C_v. If the degree is two (i.e. a walk on a line or circle) could use the Hadamard matrix. This approach can be generalized to higher values for the degree, where each entry has same amplitude at a different phase.

Another option would be to ask for two amplitudes, one for the edge to change and one for the edge to go to itself. This is modelled by a square matrix of side length $N = |N_v|$ with value a on the diagonal and value b off. These variables are subject to the conditions $a^2 + (N-1)b^2 = 1$ and $2ab + (N-2)b^2 = 0$ by our orthonormality requirement. $b = 0$ returns the trivial case $\pm I$. Another solution is $b = \pm\frac{2}{N}$, $a = \pm(\frac{2}{N} - 1)$. This particular choice of C_v has 1 eigenvector for 1 (the uniform vector) and $(N-1)$ eigenvectors for -1. This corresponds to the Grover iterate where we are reflecting around $\sum_{w\in N_v} |w\rangle$. This suggests a connection between quantum walks and quantum search that we'll explore in a bit. Note that when $N = 2$ this corresponds to movement in one direction only.

Compare classic and quantum walks in one dimension. For the classic model what we get will resemble a normal distribution with width $\Theta(\sqrt{k})$. For quantum the resulting distribution after observation will depend on the coin operation, C_v, we chose. If the gate was symmetric we can expect to have two bulges in the distribution that are $\Theta(k)$ apart. This is why we expect to find answers quicker with quantum.

3 Search

Suppose we have a blackbox on the vertices of a graph that separates "good" vertices from "bad" vertices. Given a random starting vertex we want to walk until we find a good vertex.

Definition 2. *The hitting time is the number of steps at which the probability of hitting a good value rises above a certain threshold. Alternatively, we can consider it the expected number of steps before getting our first good hit.*

Theorem 1. *Let G be a regular graph with spectral gap δ. Let ϵ be the fraction of good vertices. Classically the hitting time is $O(\frac{1}{\delta\epsilon})$ and with a quantum random walk based on Grover diffusion it is $O(\frac{1}{\sqrt{\delta\epsilon}})$. With a quantum algorithm, the hitting time will depend on the choice of C_v.*

Proof. Coming in the next lecture. □

3.1 Applications

1. Grover search. Here G is the complete graph with 2^n vertices as we allow transition from possible state to all other states uniformly at random. $\delta \sim 1$, $\epsilon = \frac{t}{N}$ so we get the expected speed up to $\mathrm{O}(\sqrt{N/t})$.

   Exercise 2. *Show $\delta = 1 - \frac{1}{N-1}$ and determine the eigenstructure for Grover search.*

2. Spatial search. This differs from Grover in that we only allow certain transitions. Examples include grids (1 dimensional, 2D, 3D, ...). This happens in appears in cases such as database search.

   Exercise 3. *For the circle graph show determine the eigenvalues and show that $\delta = O(\frac{1}{N^2})$ and $\epsilon = \frac{1}{N}$.*

   Classically, spatial search on 1 dimensional graphs takes $\mathrm{O}(N^2)$, quantum takes $\mathrm{O}(N)$. With 2D, classic takes $\mathrm{O}(N \log N)$ and quantum takes $\mathrm{O}(\sqrt{N \log N})$. For 3D and higher, classic is $\mathrm{O}(N)$ and quantum $\mathrm{O}(\sqrt{N})$.

3. Element distinctness problem: given N elements answer if they are distinct by returning a boolean value (we can also look at a search version which returns a matching pair if one exists). Our initial attempt will be to pick ℓ elements uniformly at random. Test that no element collides within and use Grover to test that no outside element collides. The number of queries is then $\ell + \sqrt{N}$ with probability of success $\geq \frac{\ell}{N}$. We can optimize it by choosing $\ell = \sqrt{N}$ but this leads to an expected linear running time which doesn't offer any improvement on the classic model. We can modify this idea to boost success rate using amplitude amplification. So the expected number of trials goes from $\frac{N}{\ell}$ to $\sqrt{\frac{N}{\ell}}$. $\ell = \sqrt{N}$ is still optimal so we get a running time of $\mathrm{O}(N^{3/4})$. We can further modify our algorithm by not clearing all l elements every time we start over but instead swap out a single element chosen uniformly at random. This corresponds to a walk on a Johnson graph, a regular graph of $|V| = \binom{N}{\ell}$ with degree $\ell(n-\ell)$ as we have ℓ elements in our current subset node to switch and $(n-\ell)$ options to switch to. It is well known for the Johnson graph that $\epsilon = \frac{\binom{N-2}{\ell-2}}{\binom{N}{\ell}} = \Theta(\frac{\ell^2}{N^2})$ and $\delta = \frac{N}{\ell(N-\ell)}$.

So cost is $O(\frac{1}{\sqrt{\delta\epsilon}}) + \ell = O(\frac{N}{\sqrt{\ell}}) + \ell$. The optimal choice is $\ell = \Theta(N^{2/3})$ yielding a $O(N^{2/3})$ running time. . Classically we don't get the square root so this method yields $O(N)$ running time.

We note that this method offers another way of seeing the solution to question 6 in assignment 1, which was stated: given a function f on $\{0,1\}^n$ that is two-to-one give an algorithm that returns two inputs that map to the same element in $O(\sqrt{3}N)$ time. By the birthday paradox we have a high probability of finding a collision amongst $\sqrt{N}$ elements so let $N' = \sqrt{N}$. Running the element distinctness algorithm returns an answer in $O((N')^{2/3}) = O(N^{1/3})$ time. This demonstrates that we have in fact found the lower bound for solving this particular problem.