CS 683: Advanced Algorithms	**Random Walks on Graphs**
03/15/01	**Scribe: Chek Beng Chua, Ranjith Rajagopalan**

1 Random Walks

We want to study the properties of random walks on graphs. From last time, we noted that connected, non-bipartite graphs have the property that any probability distribution will converge to the stationary distribution, so we'll focus on this class of graphs for this discussion.

In order to formalize the notion of a random walk on a graph, we'll construct a Markov chain which has the nodes of the graph as its states. For any edge (u, v), the probability of moving to state v from state u is $1/d(u)$, where $d(u)$ is the degree of vertex v. That is, the chain moves to any neighbor of u with equal probability.

There are two pieces of information that we want to know about this Markov chain. The first is the "hitting time" h_{uv} which is the expected time to reach node u starting from node v. The second is the "covering time" C_u which is the expected time to reach every node in the graph, starting from node u.

For various reasons, we would like to know the stationary distribution of this chain.

Claim 1.1 *The stationary distribution $\pi_v = d(v)/2m$.*

Proof. Let $\Gamma(v)$ be the neighbor set of v. Then the probability of being at node v after one step, starting from the distribution π_v is

$$\sum_{u\in\Gamma(v)} \pi_u \frac{1}{d(u)} = \sum_{u\in\Gamma(v)} \frac{d(u)}{2m}\frac{1}{d(u)} = \frac{d(v)}{2m}.$$

■

We would be happier if this were the uniform distribution, because it would make our analysis simpler. Thus, it makes sense to look at a slightly altered Markov chain where we consider edges as states instead of nodes. Specifically, we have $2m$ states, one for each edge in each direction. The probability of moving from state (u, v) to (v, w) is again $1/d(v)$. This Markov chain is almost identical to our original, with additional information about where the chain has been.

Claim 1.2 *The stationary distribution for the altered chain is $\pi_{(u,v)} = 1/2m$.*

Proof. The probability of being in state (v, w) after one step is

$$\sum_{u\in\Gamma(v)} \pi_{(u,v)} \frac{1}{d(v)} = \sum_{u\in\Gamma(v)} \frac{1}{2m}\frac{1}{d(v)} = \frac{1}{2m}.$$

■

From last time, we know that if $\pi_{(u,v)}$ is the stationary distribution, then the expected time to return to state (u, v) starting from state (u, v) is $1/\pi_{(u,v)}$. Thus, the expected time to return to an edge is $2m$. However, since our transition probabilities only depended on the head of the edge, this tells us that the expected time to traverse edge (u, v) starting at node v is $2m$.

Lemma 1.3 *If (u, v) is an edge of G, then $h_{uv} \leq 2m$.*

Proof. We know that the expected time to traverse the edge (v, u) starting from node u is $2m$. But if the chain traverses that edge, it must have reached node v at that point (possibly earlier). Thus, $h_{uv} \leq 2m$. ■

2 Examples

Example 1. Consider the path on n nodes. Using the previous lemma, we know that $h_{12} \leq 2m$, $h_{23} \leq 2m$ etc. which tells us that $h_{1n} \leq 2(n-1)(n-1) = 2(n-1)^2$.

It turns out that this is correct up to a constant factor, even though the $2m$ upper bound we used was very crude.

For the covering time, we know that C_1 is exactly the time to reach node n, so $C_1 \leq 2(n-1)^2$. In fact, C_v is at most twice this, since one possible way to cover every vertex is to move to node 1, and then to node n, and using our $2m$ upper bound for each step as before, this is at most twice the time to move from node 1 to node n.

Example 2. Consider the complete graph on n nodes, K_n. Since the chain moves to any (other) node with equal probability, the covering time for this graph is similar to the coupon collector problem, and thus $C_v = O(n \log n)$.

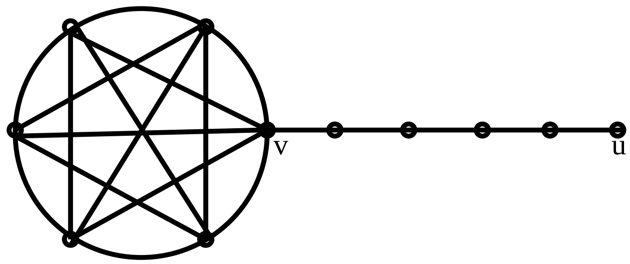
Consider $C_v(G)$ for arbitrary G. One way to upper bound the covering time is to specify an order to visit the nodes, and then use the $2m$ bound on each individual edge. Thus, by constructing a spanning tree, and then walking "around" it, the chain visits every vertex using approximately $2n$ edges, and each individual edge takes time $2m$, so we have that $C_v(G) \leq 4nm = O(nm)$.

Question (Tom Wexler): Is this bound close for trees? Consider the star, this is the coupon collector, so $C_v(G) = O(n \log n)$.

It's possible that $O(mn) = O(n^3)$, but it isn't this bad in the two examples that we've seen. Is it ever this bad?

Example 3. Consider the lollipop, a graph consisting of a complete graph on $n/2$ nodes connected to a path on $n/2$ nodes. For this graph, $C_v(G) = \theta(n^3)$ since $h_{vu} \sim n^3$.

Notice that $h_{uv} \sim n^2$, but $h_{vu} \sim n^3$, which shows that the hitting times aren't necessarily symmetric.



3 Randomized Connectivity Algorithm

This analysis gives rise to a randomized test for connectivity. Specifically, the algorithm tests if some nodes s and t are connected. It always returns false if they are not connected, and it returns true with probability at least 1/2 if they are connected.

Algorithm. Start a random walk at node s. Proceed for $8mn$ steps. If t has been reached by that point, return true, otherwise return false.

Claim 3.1 *$Pr(t$ was not reached in $8mn$ steps$) \leq 1/2$ if s is connected to t*

Proof. The expected time to cover all nodes in the component containing s and t is no more than $4mn$. Thus, Markov's Inequality gives us that

$$Pr(\text{All nodes are not covered in } 2C_s \text{ steps}) \leq 1/2.$$

Specifically, that gives us that the probability that t is not seen is less than 1/2. ■

There are certainly more reliable tests for connectivity, but this algorithm has the advantage that it runs in log space, since it only needs to keep track of the current node and the number of steps taken, which is bounded by $8mn$.

4 Connection to Electric Networks

We need to begin with some basic facts about electricity. First, if current is flowing through a network, it must obey flow conservation. Second, if the difference in voltage between points u and v is V, the resistance between u and v is R, and the current flowing between u and v is I, then $V = IR$. Using these two facts, we can calculate the resistance of a network between nodes u and v by solving a system of equations that define these constraints, corresponding to sending one unit of current from u to v. Call this resistance R_{uv}.

An alternate method to calculate the resistance between nodes u and v is as follows. Insert $d(w)$ units of flow to every node w, this adds a total of $2m$ units of flow to the network. Let all $2m$ units leave the network from node v.

Consider the equations that govern this flow. Let Φ_{uv} be the voltage level at u if v is grounded. Φ_{uu} must be zero. Since each individual edge has resistance one, if (u_1, u_2)

is an edge, then the current on that edge must be $\Phi_{u_1 v} - \Phi_{u_2 v}$. Since we must have flow conservation, that gives us:

$$\Phi_{uu} = 0,$$
$$d(u) = \sum_{w \in \Gamma(u)} \Phi_{uv} - \Phi_{wv}.$$

Now, consider the hitting times h_{uv}. We can describe h_{uv} in terms of h_{wv} for the neighbors $w \in \Gamma(u)$. The chain must take a step in some direction, each with probability $1/d(u)$. If it moves to node w, then it will take h_{wv} more steps to reach v. Specifically,

$$h_{uu} = 0,$$
$$h_{uv} = \frac{1}{d(u)} \sum_{w \in \Gamma(u)} (1 + h_{wv}).$$

Rearranging this equation,

$$\begin{aligned} d(u)h_{uv} &= \sum_{w \in \Gamma(u)} (1 + h_{wv}) \\ &= d(u) + \sum_{w \in \Gamma(u)} h_{wv} \\ \Rightarrow d(u) &= d(u)h_{uv} - \sum_{w \in \Gamma(u)} h_{wv} \\ &= \sum_{w \in \Gamma(u)} (h_{uv} - h_{wv}) \end{aligned}$$

Thus, the Φ variables are exactly the hitting times because they correspond to the solution to the same set of equations.

Finally, we noted earlier that hitting times aren't necessarily symmetric. However, if we make them symmetric by considering $h_{uv} + h_{vu}$, we get the following theorem.

Theorem 4.1 $h_{uv} + h_{vu} = 2mR_{uv}$.

Proof. (sketch) : One possible way to send $2m$ units of current from u to v is to first send $2m$ units of current into u, and let $d(w)$ units exit every node w. Then, let $d(w)$ units of current enter every node w, and let $2m$ units of flow exit from v. By the analysis we have above, the second stage gives a voltage difference of h_{uv}. If we reverse the flow in the first stage then node v has a voltage h_{vu} above node u. So, for the actual flow, node u has a voltage h_{vu} above node v. Superimposing the two stages gives a voltage difference of $h_{uv} + h_{vu}$. The voltage drop from u to v must be $2mR_{uv}$ since we sent $2m$ units of current from u to v, but it's also $h_{uv} + h_{vu}$, so $h_{uv} + h_{vu} = 2mR_{uv}$. ∎