CS787: Advanced Algorithms	
Scribe: Evan Driscoll, Daniel Wong	**Lecturer:** Shuchi Chawla
Topic: Random Walks & Markov chains: the resistance method.	**Date:** December 7, 2007

28.1 Introduction

For this lecture, consider a graph $G = (V, E)$, with $n = |V|$ and $m = |E|$. Let d_u denote the degree of vertex u.

Recall some of the quantities we were interested in from last time:

Definition 28.1.1 *The transition matrix is a matrix P such that P_{uv} denotes the probability of moving from u to v: $P_{uv} = \mathbf{Pr}[\text{random walk moves from } u \text{ to } v \text{ given it is at } u]$.*

Definition 28.1.2 *Stationary Distribution for the graph starting at v, π^*, is the distribution over nodes such that $\pi^* = \pi^* P$*

Definition 28.1.3 *The hitting time from u to v, h_{uv}, is the expected number of steps to get from u to v.*

Definition 28.1.4 *The commute time between u and v, C_{uv}, is the expected number of steps to get from u to v and then back to u.*

Definition 28.1.5 *The cover time of a graph, $C(G)$, is the maximum over all nodes in G, of the expected number of steps starting at a node and walking to every other node in G.*

Last time we showed the following:

- For a random walk over an undirected graph where each vertex v has degree d_v, $\pi^*_{2m} = \frac{d_v}{2m}$ is a stationary distribution
- In any graph, we can bound the commute time between adjacent nodes: if $(u, v) \in E$ then $C_{uv} \le 2m$
- In any graph, we can bound the cover time: $C(G) \le 2m(n - 1)$

28.2 Resistive Networks

Recall the relationships and values in electrical circuits.

The 3 principle values we will be looking at are voltage, V, current, i, and resistance r. For more information on the intuition of these properties I direct the reader to the wikipeida entry on Electrical circuits, http://en.wikipedia.org/wiki/Electrical_circuits.

Ohm's Law:

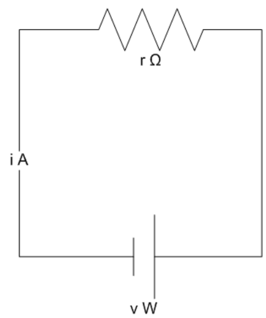
Definition 28.2.1 $V = ir$

Kirchoff's Law:

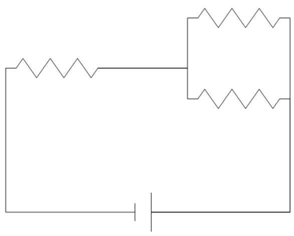
Definition 28.2.2 *At any junction, $i_{in} = i_{out}$*

In other words, current is conserved.

Here is an example circuit diagram.



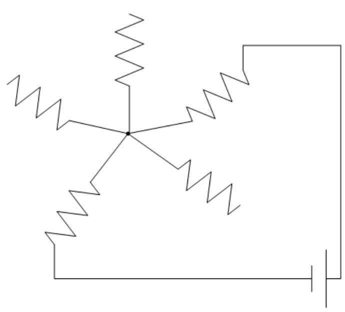
Slightly more complicated circuit:



Recall that for resistors in series the net resistance is the sum of the individual resistances, $r_{net} = \sum_{r \text{ in series}} r$. Similarly for resistors in parallel the multiplicative inverse of the net resistance is the sum of the multiplicative inverses of each individual resistor, $\frac{1}{r_{net}} = \sum_{r \text{ in parallel}} \frac{1}{r}$.

28.3 Analysis of Random Walks with Resistive Networks

Consider an undirected unweighted graph G. Replace every edge by a 1Ω resistor. Let R_{uv} be the effective resistance between nodes u and v in this circuit.



As you can see by applying these operations to a star graph, the effective resistance is 2Ω between the indicated u and v.

Lemma 28.3.1 $C_{uv} = 2mR_{uv}$

Proof:

The proof of Lemma 28.3.1 will be shown by considering two schemes for applying voltages in resistive networks and then showing that the combination of the two schemes show the lemma.

Part 1. We will analyze what would happen if we connect v to ground, then apply a current to each other vertex w of amount d_w amps. (d_w is the degree of w.) The amount of current that flows into the ground at v is $2m - d_v$, since each edge contributes one amp at each end. Let ϕ_w be the voltage at node w.

Consider each neighbor w' of w. There is a 1Ω resistor going between them. By Ohm's law, the current across this resistor is equal to the voltage drop from w to w', which is just $\phi_w - \phi_{w'}$. Look at the sum of this quantity across all of w's neighbors:

$$d_w = \sum_{w':(w,w')\in E} (\phi_w - \phi_{w'}) = d_w\phi_w - \sum_{w':(w,w')\in E} \phi_{w'}$$

Rearranging:

$$\phi_w = 1 + \frac{1}{d_w} \sum_{w':(w,w')\in E} \phi_{w'} \tag{28.3.1}$$

At this point, we will take a step back from the interpretation of the graph as a circuit. Consider the hit time h_{wv} in terms of the hit time of w's neighbors, $h_{w'v}$. In a random walk from w to v, we will take one step to a w' (distributed with probability $1/d_w$ to each w'), then try to get from w' to v. Thus we can write h_{uv} as:

$$h_{wv} = 1 + \frac{1}{d_w} \sum_{w':(w,w')\in E} h_{w'v} \tag{28.3.2}$$

However, note that equation (28.3.2) is the same as equation (28.3.1)! Because of this, as long as these equations have a unique solution, $h_{uv} = \phi_w$. We will argue that this is the case. The voltage at a node is one more than the average voltage of its neighbors. Consider two solutions $\phi^{(1)}$ and $\phi^{(2)}$. Look at the vertex w where $\phi^{(1)}_w - \phi^{(2)}_w$ is largest. Then one of the neighbors of w must also have a large difference because of the average. In both solutions, $\phi_v = 0$, so the difference in v's neighbors has to average out to zero.

Part 2. We will now analyze what happens with a different application of current. Instead of applying current everywhere (except v) and drawing from v, we will apply current at u and draw from everywhere else.

We are going to apply $2m - d_u$ amps at u, and pull d_w amps for all $w \neq u$. (We continue to keep v grounded.) Let the voltage at node w under this setup be ϕ'_w.

Through a very similar argument, $h_{wu} = \phi'_u - \phi'_w$. Thus $h_{vu} = \phi'_u - 0 = \phi'_u$.

Part 3. We will now combine the conclusions of the two previous parts. At each node w, apply $\phi_w + \phi'_w$ volts. We aren't changing resistances, so currents also add. This means that each w ($\neq u$

and $\neq v$) has no current flowing into or out of it, and the only nodes with current entering or exiting are u and v.

At v, $2m - d_v$ amps were exiting during part 1, and d_v amps were exiting during part 2, which means that now $2m$ amps are exiting. By a similar argument (and conservation of current), $2m$ amps are also entering u.

Thus the voltage drop from u to v is given by Ohm's law:

$$(\phi_u + \phi'_u) - 0 = R_{uv} \cdot 2m$$

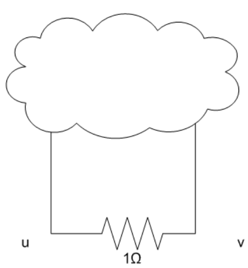
But $\phi_u = h_{uv}$ and $\phi'_u = h_{vu}$, so that gives us our final goal:

$$h_{uv} + h_{vu} = C_{uv} = 2mR_{uv}$$

■

28.4 Application of the resistance method

A couple of the formulas we developed last lecture can be re-derived easily using lemma 28.3.1. Lemma 27.5.1 says that, for any nodes u and v, if there is an edge $(u, v) \in E$, then $C_{uv} \leq 2m$. This statement follows immediately by noting that $R_{uv} \leq 1\Omega$. If a 1Ω resistor is connected in parallel with another circuit (for instance, see the following figure), the effective resistance R_{uv} is less than the minimum of the resistor and the rest of the circuit.



In addition, last lecture we showed (in lemma 27.6.1) that $C(G) \leq 2m(n-1)$. We can now develop a tighter bound:

Theorem 28.4.1 *Let $R(G) = \max_{u,v \in V} R_{u,v}$ be the maximum resistance between any two points. Then $mR(G) \leq C(G) \leq mR(G)2e^3 \ln n + n$.*

Proof: The lower bound is fairly easy to argue. Consider a pair of nodes, (u, v), that satisfy $R_{uv} = R(G)$. Then $\max\{h_{uv}, h_{vu}\} \geq C_{uv}/2$ because either h_{uv} or h_{vu} makes up at least half of the commute time. Lemma 28.3.1 and the above inequality shows the lower bound.

To show the upper bound on $C(G)$, we proceed as follows. Consider running a random walk over G starting from node u. Run the random walk for $2e^3mR(G)$ steps. For some vertex v, the chance that we have not seen v is $1/e^3$. We know that from 28.3.1 the hitting time from any u to v is at most $2mR(G)$. From Markov's inequality:

$$\begin{aligned}
\mathbf{Pr}\big[\#\text{ of steps it takes to go from } u \text{ to } v \geq 2e^3 mR(G)\big] &\leq \frac{\mathbf{E}[\#\text{ of steps it takes to go from } u \text{ to } v]}{2e^3 mR(G)} \\
&\leq \frac{2mR(G)}{2e^3 mR(G)} \\
&\leq \frac{1}{e^3}
\end{aligned}$$

(Note that this holds for any starting node $u \in V$.)

If we perform this process $\ln n$ times — that is, we perform $\ln n$ random walks starting from u ending at u' the probability that we have not seen v on *any* of the walks is $(1/e^3)^{\ln n} = 1/n^3$. Because $h_{uv} \leq 1/e^3$ for all u, we can begin each random walk at the last node of the previous walk. By union bound, the chance that there exists a node that we have not visited is $1/n^2$.

If we have still not seen all the nodes, then we can use the algorithm developed last time (generating a spanning tree then walking it) to cover the graph in an expected time of $2n(m-1) \leq 2n^3$.

Call the first half of the algorithm (the $\ln n$ random walks) the "goalless portion" of the algorithm, and the second half the "spanning tree portion" of the algorithm.

Putting this together, the expected time to cover the graph is:

$$\begin{aligned}
C(G) &\leq \mathbf{Pr}[\text{goalless portion reaches all nodes}] \cdot (\text{time of goalless portion}) \\
&\quad + \mathbf{Pr}[\text{goalless portion omits nodes}] \cdot (\text{time of spanning tree portion}) \\
&\leq \left(1 - \frac{1}{n^2}\right) \cdot (2e^3 mR(G) \cdot \ln n) + (1/n^2) \cdot (n^3) \\
&\leq 2e^3 mR(G) \ln n + n
\end{aligned}$$

■

28.5 Examples

28.5.1 Line graphs

Last lecture we said that $C(G) = O(n^2)$. We now have the tools to show that this bound is tight. Consider u at one end of the graph and v at the other; then $R_{uv} = n - 1$, so by lemma 28.3.1, $C_{uv} = 2mR_{uv} = 2m(n-1)$, which is exactly what the previous bound gave us.

28.5.2 Lollipop graphs

Last lecture we showed that $C(G) = O(n^3)$. Consider u at the intersection of the two sections of the graph, and v at the end. Then $R_{uv} = n/2$, so $C_{uv} = 2m\frac{n}{2} = 2\Theta(n^2)\frac{n}{2} = \Theta(n^3)$. Thus again our previous big-O bound was tight.

28.6 Application of Random Walks

We conclude with an example of using random walks to solve a concrete problem. The 2-SAT problem consists of finding a satisfying assignment to a 2-CNF formula. That is, the formula takes the form of $(x_1 \vee x_2) \wedge (x_3 \vee \overline{x_1}) \wedge \ldots$. Let n be the number of clauses.

The algorithm works as follows:

1. Begin with an arbitrary assignment
2. If the formula is satisfied, halt
3. Pick any unsatisfied clause
4. Pick one of the variables in that clause UAR and invert it's value
5. Return to step 2

Each step of this algorithm is linear in the length of the formula, so we just need to figure out how many iterations we expect to have before completing.

This algorithm can be viewed as performing a random walk on a line graph with $n + 1$ nodes. Each node corresponds to the number of variables in the assignment that differ from a satisfying assignment (if one exists). When we invert some x_i, either we change it from being correct to incorrect and we move one node away from 0, or we change it from being incorrect to being correct and move one step closer to the 0 node.

However, there is one problem with this statement, which is that our results are for random walks where we take any outgoing edge uniformly. Thus we should argue that the probability of taking each edge out of a node is $\frac{1}{2}$. In the case where the algorithm chooses a clause with both a correct and incorrect variable, the chances in fact do work out to be $\frac{1}{2}$ in each direction. In the case where the algorithm chooses a clause where both variables are incorrect, it will *always* move towards the 0 node. Thus the probability the algorithm moves toward 0 is at least $\frac{1}{2}$. It may be better, but that only biases the results in favor of shorter running times.

Thus the probability of the random walk proceeding from i to $i-1$, and hence closer to a satisfying assignment, is at least $1/2$. Because of this, we can use the value of the hitting time we developed for line graphs earlier. Hence the number of iterations of the above we need to perform, is $O(n^2)$.

28.7 Next time

Next time we will return to a question brought up when we were beginning discussions of random walks, which is how fast do we converge to a stationary distribution, and under what conditions are stationary distributions unique.