CS 388R: Randomized Algorithms, Fall 2023 November 1st, 2023

Lecture 20: Markov Chains II; Hitting and Cover Times

Prof. Eric Price *Scribe: Bennett Liu*

NOTE: THESE NOTES HAVE NOT BEEN EDITED OR CHECKED FOR CORRECTNESS

1 Overview

In the last lecture we finished up our discussion of graph sparsification and began the discussion of Markov Chains. Specifically, we defined the idea of an ergodic Markov Chain and the Fundamental Theorem of Markov Chains. We ended by briefly discussing random walks in undirected graphs.

In this lecture we continue the discussion of random walks, specifically providing an exact formula for commute time and bounding cover time within a factor of $O(\log n)$.

2 Some Definitions

We can begin by defining several values. Some of these were covered in the previous lecture but are included for completeness.

Definition 1. *P is a transition matrix, representing our graph. We define the weights of this transition matrix as:*

- $P_{uv} = \frac{1}{d(u)}$ *if $v \in N(u)$, where $d(u)$ is the degree of the vertex u.*
- *Alternatively, we can say that $P = D^{-1}A$.*

Definition 2. *The hitting time, h_{uv}, is the expected time to hit v on a random walk beginning from u.*

Definition 3. *The commute time c_{uv} is equal to the expected time to begin at u, hit v, then hit u.*

- *This can be represented as $c_{uv} = h_{uv} + h_{vu}$*
- *We will show that $c_{uv} = 2mR_{uv}$.*

Definition 4. *The cover time, $C_u(G)$ is the expected time to cover G starting from u.*

Specifically, the starting node matters in examples such as the lollipop (a fully connected graph connected to a chain of nodes). The cover time beginning at the end of the lollipop's stick is less than beginning in the candy part, as it is easier to move from the stick to the candy than vice versa.

Definition 5. *The effective resistance between u and v is $R_{uv} = (e_u - e_v)^\top L_G^+(e_u - e_v)$*

3 Computing Commute/Hitting Time

We seek to prove that $c_{uv} = 2mR_{uv}$.

Physical intuition Define $h_{uu} = 0$ and consider injecting $d(v)$ current at each vertex v.

Note that this adds a total of $\sum_v d(v) = 2m$ current injected into the graph. Given that our origin is u, we will remove $2m$ current at our source node u, balancing out the current. This current is represented by the vector i.

In this graph, the induced voltages are given by $x = L_G^+ i + c$, where c is some constant.

We claim that we can choose a c such that our voltage $x_u = 0$ and in general, the voltage $x_v = h_{vu}$ for all v.

We can prove this by analyzing the hitting time of u from v in terms of neighboring hitting times. We proceed to each neighbor with probability $\frac{1}{d(v)}$, thus:

$$h_{vu} = \sum_{w \in N(v)} \frac{1}{d(v)}(1 + h_{wu})$$

$$h_{vu} = 1 + \frac{1}{d(v)} \sum_{w \in N(v)} h_{wu}$$

$$1 = h_{vu} - \frac{1}{d(v)} \sum_{w \in N(v)} h_{wu}$$

$$d(v) = d(v)h_{vu} - \sum_{w \in N(v)} h_{wu}$$

$$d(v) = \sum_{w \in N(v)} (h_{vu} - h_{wu})$$

If $x_v = h_{vu}$, then the difference in voltages, $x_v - x_w = h_{vu} - h_{wu}$ is the current between v and w. Then the net current at node v is $\sum x_v - x_u = d(v)$, as defined, so we can say that $x_v = hvu$.

We now continue with our analysis of the cover time. By definition, $c_{uu'} = h_{uu'} + hu'u$. Let $x_v = h_{vu}$ and $x'_v = h_{vu'}$.

$$X = L_G^+(d - 2me_u) + c$$

$$X' = L_G^+(d - 2me_{u'}) + c'$$

This means that

$$(x - x') = 2mL_G^+(e_{u'} - e_u) + (c - c')$$

We can then substitute this into our definition of commute time.

$$c_{uu'} = h_{uu'} + h_{u'u}$$

Note that we defined $h_{u'u'} = h_{uu} = 0$.

$$c_{uu'} = (h_{u'u} - h_{u'u'}) - (h_{uu} - h_{uu'})$$

$$c_{uu'} = (x_{u'} - x'_{u'}) - (x_u - x'_u)$$
$$c_{uu'} = (x - x')^\top (e_{u'} - e_u)$$

We can now substitute the previous definition $(x - x') = 2mL_G^+(e_{u'} - e_u) + (c - c')$. Given that the constants $c - c'$ have the same result when they are multiplied by $e_{u'}$ and e_u, they cancel out.

$$c_{uu'} = 2m(e_{u'} - e_u)^\top L_G^+(e_{u'} - e_u)$$

Notably, we defined effective resistance to be $R_{uv} = (e_u - e_v)^\top L_G^+(e_u - e_v)$, so we can substitute to get that $c_{uu'} = 2mR_{u'u}$.

Bounding Cover Time

Next, we investigate bounding $C_u(G)$.

A loose bound For any spanning tree, one way to tour a graph is to walk around a spanning tree. For each edge in the spanning tree, we can view this as walking across, then later walking back. Thus, this would take $\sum_{(u,v)\in tree} h_{uv} + h_{vu} = \sum_{(u,v)\in tree} c_{uv}$ time.

Given that the effective resistance is always ≤ 1, we can bound $c_{uv} = 2mR_{uv} \leq 2m$. This means that $C_u(G) \leq 2m(n-1)$

This is tight for lollipop and line but not for a clique, where a spanning tree is far more restrictive than the graph itself.

Tighter bounds As the previous bound isn't very tight for some examples, we seek to establish tighter bounds that also work for the case of denser graphs like cliques.

Define R_{max} to be the maximum resistance in the graph, $R_{max} = max_{u,v} R_{(u,v)\in edges(G)}$.

We claim that we can bound $C(G)$ with

$$mR_{max} \leq C(G) \lesssim mR_{max} logn$$

Note that in a line, R_{max} is $O(n)$, but for a clique, it is much smaller, providing a better bound.

Lower bound proof Let $(u*, v*)$ be nodes that have $R_{u*v*} = R_{max}$. As established previously, we know the cover time:

$$c_{u*v*} = 2mR_{max} = h_{u*v*} + h_{v*u*}$$

Thus, we can guarantee that

$$\max(h_{u*v*}, h_{v*u*}) \geq mR$$

This shows that there is at least one pair u, v such that $h_{uv} \geq mR$. For any such pair, we know that any cover beginning at u must travel to v, so we can say that $C_u(G) \geq h_{uv} \geq mR_{max}$.

Upper bound proof For all u, v, we know that $c_{uv} = 2mR_{uv}$, so

$$c_{uv} \leq 2mR_{max}$$

Since $h_{uv} \leq c_{uv}$, we can also say that

$$h_{uv} \leq 2mR_{max}$$

We claim that after $O(mR_{max} \log n)$ time, we have visited an arbitrary node v with high probability.

We already know that the expected time to reach v from u is less than $2mR_{max}$, the maximal commute time.

Thus, the probability to reach v in the first $4mR_{max}$ steps $\geq \frac{1}{2}$. If this fails, we can repeat the process and we again reach v with probability $\geq \frac{1}{2}$.

If we repeat this process $O(\log(n))$ times, we have a $(\frac{1}{2})^{c\log(n)} \approx \frac{1}{n^c}$ probability of failure.

Union bounding this failure probability over all nodes, we have a $\frac{1}{n^{c-1}}$ probability of failure.

Thus, after repeating this $O(mR_{max} \log n)$ times, we succeed with high probability.

This means that $C(G) \lesssim mR_{max} \log n$.

Application: $s - t$ Connectivity

We can apply the first result to determine if two nodes s and t are connected within a small space.

The traditional method for accomplishing this task is to run a DFS or BFS from one node, searching for the other. This takes $O(m)$ time (as we may need to consider crossing each edge) and $O(n)$ space.

However we can instead perform this with much less space by performing a random walk beginning at s. This allows us to only store the specific node that our algorithm occupies, which uses $O(\log(n))$ memory. If t is reachable, we expect to see it after $C(G)^{\leq O(mn)}$ time.