CSE 632 (Fall 2019): Analysis of Algorithms II : Randomized Algorithms

Lecture 20 (11/1/2019): Random Walk on a Graph

Lecturer: Shi Li *Scriber: Yifu Yin*

1 Random Walk on a Graph

In the last two lectures, we talked about 2-SAT and 3-SAT. The analysis of the algorithms are based on *random walks* on a line, although we did not make the definition formal. In this lecture, we give some basic definitions related to random walks on undirected graphs.

Let $G = (V, E)$ be a undirected graph. For every $v \in V$, we let d_v be the degree of v in G. In the random walk in G, we can think of that there is a particle moving among vertices V of G in discrete time steps. If the particle is at position $v \in V$ at time t, then at time $t + 1$, it will be at a random chosen neighbor u of v. Thus, the transition matrix of the Markov chain is defined as $D^{-1}A$ where A is the adjacency matrix of G, and

$$D = \begin{pmatrix} d_{v_1} & & & & \\ & d_{v_2} & & & \\ & & d_{v_3} & & \\ & & & \ddots & \\ & & & & d_{v_n} \end{pmatrix}$$

is the diagonal matrix containing the degrees of the n vertices. In the lecture we assume the graph does not have self-loops and parallel edges, and every vertex has degree at least 1.

1.1 Stationary Distributions

Definition 1 (Stationary Distribution)**.** *For a Markov chain with transition matrix P, a stationary distribution π is one such that $\pi P = \pi$.*

In other words, if we start from an stationary distribution π at time 0, then the distribution of states for any time $t \geq 0$ in the Markov chain is π. For a general Markov chain, a stationary distribution can be computed using the following linear program:

$$\pi P = \pi, \quad \sum_v \pi_v = 1, \quad \pi_v \geq 0, \forall v.$$

Lemma 2. *For a random walk in a graph $G = (V, E)$, the following distribution π is a stationary distribution:*

$$\pi_v = \frac{d_v}{2m}, \textit{for every } v \in V,$$

where $m = |E|$ is the number of edges.

For example, consider the graph with the following adjacent matrix A and transition matrix P:

$$A = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}, \qquad P = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array} \begin{array}{c} \begin{array}{ccccc} a & b & c & d & e \end{array} \\ \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 & 0 \\ 1/3 & 0 & 1/3 & 0 & 1/3 \\ 0 & 1/2 & 0 & 1/2 & 0 \end{pmatrix} \end{array}$$

A stationary distribution will be $\left(\frac{3}{14}, \frac{3}{14}, \frac{3}{14}, \frac{3}{14}, \frac{2}{14}\right)$.

Proof of Lemma 2. Let $P = D^{-1}A$ be the transition matrix of the random walk. Then for every $v \in V$, we have $(\pi P)_v = (\pi D^{-1}A)_v = \sum_{u\sim v} \frac{1}{d_u}\pi_u = \sum_{u\sim v} \frac{1}{d_u}\frac{d_u}{2m} = \sum_{u\sim v}\frac{1}{2m} = \frac{d_v}{2m} = \pi_v$. $\square$

We state the following lemma without giving a proof:

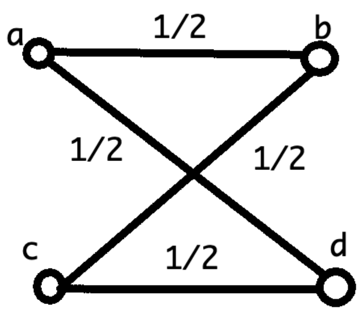
Lemma 3. *The station distribution for the random walk in G is unique if and only if G is connected.*

Moreover, if further we have that G is non-bipartite, then a random walk in G will converge to its stationary distribution.

Lemma 4. *Let $G = (V, E)$ be connected and non-bipartite graph and P be the transition matrix of the random walk on G. Then a random walk in G converges to the stationary distribution. Formally, for every initial distribution π', and every $v \in V$, we have*

$$\lim_{t\to\infty} (\pi' P^t)_v = \frac{d_v}{2m}.$$

We remark that in the above definition, the graph G being non-bipartite is necessary. Consider the following graph and we start from the distribution $\pi = [1, 0, 0, 0]$. Then the graph will never reach stationary distribution since the random walk will be "periodic".



	a	b	c	d
time 0	1	0	0	0
time 1	0	1/2	0	1/2
time 2	1/2	0	1/2	0
time 3	0	1/2	0	1/2
time 4	1/2	0	1/2	0
		...		

1.2 Hitting time and Covering Times

Below we assume we are given a connected graph $G = (V, E)$ and we consider a random walk on G.

Definition 5. *Let $u, v \in V$, the hitting time$H_{u,v}$ from u to v is defined as the expected number of steps we need to visit v if we start the random walk from u. The commute time $C_{u,v}$ from u to v is the expected number of steps the random walk from u takes to visit u and then come back to v.*

It is easy to see that $C_{u,v} = H_{u,v} + H_{v,u}$ for every $u, v \in V$.

Definition 6. *For every $u \in V$, the covering time C_u is defined as the expected number of steps it takes for a random walk from u to visit all verticies in V. The covering time of the graph G is defined as $C_G := \max_{u\in V} C_u$.*

Theorem 7. *If G is connected, then the covering time $C_G \leq 2m(n-1)$.*

The proof of the theorem is left as an exercise. The upper bound is indeed tight, as can be seen from a random walk on a line of n points. As we already showed, the expected number of steps it takes for a random walk from one end point of the path to visit the other end point is n^2.

1.3 s-t-connectivity using $O(\log n)$ space

Now we are concerned with the problem of solving the s-t connectivity using $O(\log n)$ space. We are given a graph $G = (V, E)$ and $s, t \in V$. The input is stored in a *read-only* memory. We need to decide if there is a path from s to t in G using only $O(\log n)$ extra number of bits. In this lecture, we show a randomized algorithm that achieves this goal, which is based on random walk:

Algorithm 1 s-t-connectivity using $O(\log n)$ space

1: let $v \leftarrow s$
2: **for** up to $10mn$ times **do**
3: let v be a random neighbor of v
4: **return** true if $v = t$ in some step of the random walk and false otherwise.

Notice that if s and t are disconnected in G, then our algorithm will always return false. Now assume s and t are connected in G. Then, the expected number of steps for the random walk to visit all verticies (which contains t) contained in the connected component of G containing s is at most $2m(n-1) \leq 2mn$. Thus, by Markov inequality, the probability that this did not happen in $10mn$ steps is at most $1/5$. Thus the success probability of the algorithm is at least $1 - \frac{1}{5} = \frac{4}{5}$. We can repeat the procedure multiple times to boost the success probability.