Randomized Algorithms 2019A – Lecture 2
Random Walks on Graphs*

Robert Krauthgamer

1 Random Walks on Graphs

Let $G = (V, E)$ be an undirected graph on n vertices. Throughout, we shall assume that G is connected.

A random walk on G is the following random process that proceeds in discrete steps. Start at some initial vertex $v_0 \in V$, then at each time step, pick a random neighbor (same as random incident edge) of the current vertex and move to that vertex.

Formally, for each vertex $v \in V$ let $N(v) \subset V$ be the set of its neighbors, and let $\deg(v) = |N(v)|$ be its degree. Now define random variables $X_0, X_1, \ldots$ where $X_0 = v_0$, and for each $t \geq 0$, set X_{t+1} to each $w \in N(X_t)$ with probability $1/\deg(X_t)$.

Remark: Given X_t, we know the distribution of future steps $(X_{t+1}, X_{t+2}, \ldots)$ and it will not change if we are also given any additional information about earlier steps $(X_{t-1}, X_{t-2}, \ldots)$. This is called a Markovian process.

Potential usage: We will see how random walks can be used to design various algorithms. For example, to check if $u, v \in V$ are connected, we could start a random walk at u and see if it reaches v within a reasonable amount of time. We need to analyze the probability to reach v, but implementing the walk surely requires very little storage!

2 Hitting Time

The *hitting time* from vertex u to vertex v, denoted H_{uv}, is the expected number of steps for a random walk that starts at u until it hits v. Formally, define the random variable $T = \min\{t \geq 0 : X_t = v\}$ and let $H_{uv} = \mathbb{E}[T]$.

Notice that H_{uv} depends on G, but it is not a random variable (despite capital letter notation). Notice also that it is not symmetric, i.e., in some cases $H_{uv} \neq H_{vu}$.

*These notes summarize the material covered in class, usually skipping proofs, details, examples and so forth, and possibly adding some remarks, or pointers. The exercises are for self-practice and need not be handed in. In the interest of brevity, most references and credits were omitted.

Example: Consider an n-clique, i.e., $G = K_n$. Then $H_{uv} = n - 1$ for all $u \neq v$, because T is a geometric random variables with parameter $p = 1/(n-1)$. And by definition $H_{uu} = 0$ (for every G).

Lemma 1: We have the directed triangle inequality

$$\forall u, v, w \in V, \qquad H_{uw} \leq H_{uv} + H_{vw}.$$

Proof: Was seen in class, using one random walk that starts at u.

Exer: Let $G = K_{n_1,n_2}$ be a complete bipartite graph with n_1 and n_2 vertices. Analyze H_{uv} for all possible $u, v \in V$.

Exer: Let G be a path on n vertices. Give an explicit formula for H_{uv} for all possible $u, v \in V$, and show in particular that $H_{uv} = O(n^2)$.

Hint: Denote the vertices $1, 2, \ldots, n$, and write linear equations $H_{uv} = 1 + \frac{1}{2}H_{u-1,v} + \frac{1}{2}H_{u+1,v}$ and solve these $\binom{n}{2}$ equations over $\binom{n}{2}$ variables. A simpler version is to consider h_{uv} only for $u < v$ (the other case follows by symmetry), express each $H_{uv} = H_{u,u+1} + H_{u+1,u+2} + \cdots + H_{v-1,v}$, and now the earlier equations give us $n-1$ equations using $n-1$ variables.

We will soon see that the hitting time is always (for every connected G) bounded by a polynomial in n. The next exercise shows this is not true for directed graphs.

Exer: Consider the analogous definitions of random walks and hitting time for *directed* graphs, and show (that for every n) there exists a directed graph on n vertices and two vertices u, v such that $H_{uv} = 2^{\Omega(n)}$.

3 Commute Time

The *commute time* between vertices u and v is defined as $C_{uv} = H_{uv} + H_{vu} = C_{vu}$. It can be viewed as the expected time for a random walk that starts at u, to return to u after at least one visit to v. It is sometimes viewed as a symmetric version of the hitting time.

Lemma 2: We have the triangle inequality

$$\forall u, v, w \in V, \qquad C_{uw} \leq C_{uv} + C_{vw}.$$

The proof follows immediately from Lemma 1.

Theorem 3: For all $(u, v) \in E$, we have $C_{uv} \leq 2|E|$.

We will prove it in the next class, for now let's see some consequences.

Corollary 4: For all $u, v \in V$, we have $C_{uv} \leq 2(n-1)|E| < n^3$ (recall G is connected).

Proof: Follows from Lemma 2 (the triangle inequality) along a shortest path between u and v, and then applying Theorem 3.

4 Undirected Connectivity

Undirected st-connectivity (USTCON): In this problem, the input is a undirected graph G and two vertices s, t and the goal is to determine if s, t are the in the same connected component (equivalently, there is a path between them).

Theorem 5 [Aleliunas, Karp, Lipton, Lovasz, and Rackoff, 1979]: $USTCON \in RL$, i.e., USTCON can be solved by a randomized algorithm (Turing machine) that uses $O(\log n)$ bits of space and has one-sided error.

Proof: Was seen in class.

Remark: It was a big open problem to solve USTCON in deterministic logarithmic space, and Reingold proved it in 2005.

Exer: Show similarly how to decide whether all of G is connected (i.e., G has only one connected component) in randomized log-space.

5 Cover Time

The *cover time* from vertex u, denoted $\text{cov}_u(G)$, is the expected number of steps until a random walk that starts at u has visited all vertices of G. Formally, let $T' = \min\{t \geq 0 : \{X_0, X_1, \ldots, X_t\} = V\}$ and let $\text{cov}_u(G) = \mathbb{E}[T']$.

The cover time of a graph G is defined as $\text{cov}(G) = \max_u \text{cov}_u(G)$, i.e., according to the "worst-case" starting vertex.

Example: In the n-clique, the cover time is $O(n \log n)$. (It is just the coupon collector problem, as we will soon see).

Exer: Prove it.

Theorem 6: $\text{cov}(G) \leq 2(n-1)|E|$.

Proof: Was seen in class, using a spanning tree T of the graph.

Theorem 7 (Matthews' bound): Let G be a connected graph on n vertices, and let $H_{\max} = \max\{H_{uv} : u, v \in V\}$. Then

$$H_{\max} \leq \text{cov}(G) \leq O(\log n) H_{\max}.$$

Proof: Was seen in class.

Exer: Show that each of these inequalities is tight (up to constants) for some graph G.

6 Electrical Networks

It turns out that random walks are "equivalent" to electrical networks (composed of resistors), and this "physical" interpretation gives alternative ways to prove things. We first introduce the basic concept.

Given an undirected graph $G = (V, E)$, we think of it as an electrical circuit with unit resistors. The basic property of electrical circuits is that current flows when there is a potential difference (e.g., between the endpoints of a resistor).

What happens when two vertices are connected to the positive and negative terminals of a battery? We create a "potential difference" between these two vertices, which induces a current (or electrical flow) in the network, which satisfies the following laws:

Kirchhoff's Current Law: At every vertex, the total incoming flow equals the total outgoing flow.

Kirchhoff's Voltage Law: The sum of potential differences along every (directed) cycle is zero.

Ohm's Law: The current flowing from u to v through an edge $\{u, v\}$ of resistance r_{uv} is exactly $\frac{\phi_{uv}}{r_{uv}}$, where ϕ_{uv} is the potential difference on (the endpoints of) the resistor.

We assumed unit resistors, but in general, if G has edge weights, then each edge e would have resistance $r_e = 1/w_e$ (i.e., its conductance is $c_e = 1/r_e = w_e$), and this corresponds to a random walk according to the edge weights, i.e., each outgoing edge is picked with probability proportional to w_e.

Remark: (KVL) explains why we call it "potential difference". It implies that we can assign a potential to each vertex, i.e., define $\phi' : V \to \mathbb{R}$, such that $\phi_{uv} = \phi'_u - \phi'_v$ for every edge. Obviously, this map is unique up to translation (if G is connected).