Lecture 14: Random Walks in Undirected Graphs

1 Random Walks in Undirected Graphs

A finite state Markov chain corresponds to a random walk in a weighted directed graph. Random walks in undirected graphs have many nice properties and a number of applications. They are also closely related to reversible Markov chains.

Suppose $G = (V, E)$ is an undirected graph. We let $\vec{G} = (V, \vec{E})$ be the corresponding bidirected graph.

We can consider weights on the edges, but for simplicity we assume all are 1 (we allow multi-graphs).

1.1 Definition of Random Walk

A random walk on G is the following stochastic process: Start at some random vertex given by a probability distribution π_0 on V. In each step, if we are at vertex v, pick a uniform random edge in $\delta(v)$ and go to the endpoint of v.

Note that if the edge is a self-loop, we stay at v.

We can think of this random walk as a Markov chain on V where each edge (v, u) is given probability $\frac{1}{d(v)}$.

Lemma 1. *Suppose G is a loopless connected graph. Then G is aperiodic iff G is not bipartite.*

Proof. If G is bipartite, the underlying chain has period 2, since all cycles and closed walks have even length.

If G is not bipartite, G has an odd length cycle. In $\vec{G}$ we have that each vertex is in a closed walk of even length and one with odd length. By gcd, the period is 1. □

We can either assume G is not bipartite or add self-loops on each vertex and make the walk lazy. This will ensure the walk is aperiodic (ergodic).

Lemma 2. *A random walk on G converges to a stationary distribution π where $\pi(v) = \frac{d(v)}{2m}$.*

Proof. Exercise: Verify that this satisfies $\pi P = \pi$ for the underlying Markov chain. □

1.2 Hitting Times and Commute Times

Let $h_{u,v}$ be the expected time to reach state v when starting at u.

Hitting time is not necessarily symmetric.

Example: Lollipop graph L_n: $h_{a,b} = \Theta(n^2)$ and $h_{b,a} = \Theta(n)$.

Also, L_n shows that adding edges can increase $h_{u,v}$ and $C_{u,v}$.

Commute time is $C_{u,v} = h_{u,v} + h_{v,u}$, which is symmetric.

1.3 Basic Results

We will prove two basic results using elementary methods.

Lemma 3. *For any edge $uv \in E$: $h_{u,v} + h_{v,u} \leq 2m$.*

Proof. Consider: We can view the random walk on G as a random walk on $\vec{E}$. That is, the state space is $\vec{E}$. Consider this claim.

Consider the transition matrix Q for this chain. It turns out to be doubly stochastic.

For a normal transition matrix, row sum is 1, but here column sum is also 1. Easy to verify: $(1, 1, \ldots, 1)^T$ is a left eigenvector of Q. By normalizing, the stationary distribution of Q is $\frac{1}{2m}$, the uniform distribution.

$h_{u,v} + h_{v,u} \leq 2m$ where $h_{(u,v),(u,v)}$ is the expected time in the edge walk chain to start on edge (u, v) and revisit (u, v). We can interpret such a walk as giving an upper bound on $h_{u,v} + h_{v,u}$.

Claim: $h_{u,v} + h_{v,u} \leq 2m$. If the original random walk traversed the edge (u, v), then the expected time to traverse (u, v) again is $2m$.

Note: Since the original walk is memoryless, once it reaches v, it shows that the expected time to visit u and take edge (u, v) is at most $2m$.

But this walk is only one way to start at v and reach u and back to v: $h_{v,u} + h_{u,v} \leq 2m$. □

Caveat: Note the above holds only for $u, v \in E$. We will later see a more refined version when (u, v) is not necessarily an edge.

1.4 Cover Time

Definition 1. The cover time of a graph $G = (V, E)$ is the max over all $v \in V$ of the expected time to visit all the vertices. $C(v)$ is cover time starting at v. $C(G) = \max_v C(v)$.

Theorem 1. $C(G) \leq 2m(n - 1)$.

Proof. Consider a spanning tree T of G.

We can consider an Eulerian walk on T. Say it is $v_1, v_2, v_3, \ldots, v_{2n-2}, v_1$.

We can upper bound $C(v_1)$ by:

$$h_{v_1,v_2} + h_{v_2,v_3} + \cdots + h_{v_{2n-2},v_1} = \sum_{uv \in E(T)} (h_{u,v} + h_{v,u}) \leq 2m(n - 1).$$

□

One can prove another interesting upper bound on cover time:

Theorem 2. $C(G) \leq (n - 1) \max_{u,v \in V} h_{u,v}$.

2 Applications

2.1 s-t Connectivity in $O(\log n)$ Space

Suppose we are given an undirected graph written on read-only memory in adjacency list/matrix format. We want to use very little extra memory to decide if some given s can reach t. We can easily do this using $O(n)$ space by using graph search (BFS/DFS).

Can we do this with $O(\log n)$ space? Note that writing s or t takes $O(\log n)$ bits.

Yes, if we allow randomization!

How: Start a random walk at s. Because $C(G) = O(mn)$, if we don't see t after $O(mn \log n)$ steps, we know w.h.p. that s is not connected to t.

Can implement random walk in $O(\log n)$ space.

G can be bipartite, so need to use lazy random walk. Doesn't change details too much.

2.2 2-SAT

2-SAT: Given a Boolean formula $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ where each clause has exactly 2 variables. Can check if ϕ is satisfiable, e.g., $\phi = (x_5 \vee x_3) \wedge (x_1 \vee \bar{x}_2) \wedge (\bar{x}_3 \vee x_7)$.

2-SAT is solvable in P. How? One nice way to see it is via random walks.

Algorithm:

1. Let $a = (a_1, a_2, \ldots, a_n) \in \{0,1\}^n$ be an arbitrary assignment to x_i.
2. While a does not satisfy ϕ do:
   - Let C_i be an arbitrary clause that is not satisfied by a.
   - Pick a literal of C_i uniformly at random.
   - Flip the assignment for the chosen literal and update a.

Lemma 4. *If ϕ is satisfiable, the algorithm terminates in $O(n^2)$ steps.*

Proof. Suppose s is a fixed satisfying assignment. Let a^t be the assignment after t steps. Let $d_t = \text{dist}(s, a^t)$ be the Hamming distance between s and a^t. That is, the number of variables in which a^t differs from s.

If $d_t = 0$ then algorithm terminates.

The algorithm can be viewed as doing a random walk on state space $\{0, 1, 2, \ldots, n\}$ and starting at position $\text{dist}(s, a^0)$.

Since only one variable is changed, distance changes by $+1$ or -1.

Since C_i is picked as an unsatisfied clause, at least one literal is incorrect, and hence with probability at least $\frac{1}{2}$ we will reduce distance.

Thus we can view this as a walk on $\{0, 1, 2, \ldots, n\}$. In the worst case, it starts at n on each side. Can view it as a random walk on a finite line.

Cover time of line is $O(n^2)$; will visit 0 in $O(n^2)$ in expectation. □

3 Electrical Networks and Random Walks

Ohm's law: $V = IR$ (voltage = current $\times$ resistance).

For resistors in series: effective resistance is $R = R_1 + R_2$.

For resistors in parallel: effective resistance is $R = \frac{R_1 R_2}{R_1 + R_2}$.

Let $R_{u,v}$ be effective resistance between u and v.

Theorem 3. $C_{u,v} = h_{u,v} + h_{v,u} = 2m \cdot R_{u,v}$.

Corollary 1. *If $uv \in E$, then $C_{u,v} \leq 2m$.*