CSE 203A: Randomized Algorithms

Spring 2026

Lecture 14: Random Walks, Return Times, Cover Times, and Matchings
Date: 29 April 2026
Instructor: Daniel M. Kane
Scribe: Youchen Qing

Lecture Overview

This lecture continued our study of random walks and Markov chains. We reviewed the steady-state distribution of a random walk on an undirected graph, proved the return-time identity $h_{uu} = 1/\pi_u$, and applied this identity to a randomized algorithm for finding perfect matchings in regular bipartite graphs. We then introduced hitting time, commute time, and cover time, and compared the cover times of complete graphs, paths, and lollipop graphs before proving the general upper bound $O(mn)$ for undirected graphs.

1 Random Walks on Graphs

A Markov chain is a random process $X_0, X_1, \ldots$ whose next state depends only on the current state. If

$$P_{ij} = \mathbb{P}(X_{t+1} = j \mid X_t = i)$$

is its transition matrix and $q_i^t = \mathbb{P}(X_t = i)$, then

$$q^{t+1} = q^t P.$$

A distribution π is stationary if $\pi = \pi P$. For a finite, connected, aperiodic chain, the distribution q^t converges to the unique stationary distribution π.

For a graph $G = (V, E)$, the random walk moves from the current vertex to a uniformly random neighbor. Thus

$$P_{uv} = \begin{cases} 0, & (u,v) \notin E, \\ \dfrac{1}{\deg(u)}, & (u,v) \in E. \end{cases}$$

Proposition 1.1. *For a random walk on an undirected graph $G = (V, E)$ with $m = |E|$, the stationary distribution is*

$$\pi_u = \frac{\deg(u)}{2m}.$$

Proof. For any vertex v,

$$\sum_{u \in V} \pi_u P_{uv} = \sum_{u \sim v} \frac{\deg(u)}{2m} \cdot \frac{1}{\deg(u)} = \sum_{u \sim v} \frac{1}{2m} = \frac{\deg(v)}{2m} = \pi_v.$$

Therefore $\pi P = \pi$. □

This says that the long-run probability of being at a vertex is proportional to its degree. This clean formula is special to undirected graphs. In directed graphs, steady-state probabilities can behave very differently; for example, a walk that repeatedly either advances down a path or resets to the start can have exponentially small probability of reaching far-away states.

2 Return Times

Definition 2.1 (Hitting and Return Times)**.** For vertices $u, v \in V$, the hitting time

$$h_{uv} = \mathbb{E}[\text{time to reach } v \mid X_0 = u].$$

The return time h_{uu} is the expected time to return to u, not counting time 0 as a return.

Theorem 2.2 (Return-Time Formula)**.** *For a finite irreducible Markov chain with stationary distribution π,*

$$h_{uu} = \frac{1}{\pi_u}.$$

Proof Sketch. Let R_i be the time between the i-th and $(i+1)$-st visits to u. By the Markov property, the R_i's are independent copies with mean h_{uu}. Thus after N visits to u, the elapsed time is about Nh_{uu}, so the fraction of time spent at u is about $1/h_{uu}$.

On the other hand, the walk converges to stationarity, so the long-run fraction of time spent at u is π_u. Hence $1/h_{uu} = \pi_u$. □

Equivalently, if $N_u(T)$ is the number of visits to u in the first T steps, then

$$\mathbb{E}[N_u(T)] = \sum_{t=1}^{T} \mathbb{P}(X_t = u) = \sum_{t=1}^{T} q_u^t \approx T\pi_u.$$

This gives the same identity. In an undirected graph,

$$h_{uu} = \frac{2m}{\deg(u)}.$$

3 Perfect Matching in Regular Bipartite Graphs

Let $G = (L \cup R, E)$ be a k-regular bipartite graph with $|L| = |R| = n$. The goal is to find a perfect matching. General bipartite matching algorithms run in about $O(m\sqrt{n}) = O(kn^{3/2})$ time here, but for regular bipartite graphs we can use random walks to get expected time $O(n \log n)$.

Suppose the current matching M has size t. An augmenting path starts at an unmatched left vertex, ends at an unmatched right vertex, and alternates between edges not in M and edges in M. Flipping the path increases the matching size by one.

3.1 Auxiliary Graph

Construct a directed graph G' encoding alternating paths. Let S be a special vertex representing all unmatched vertices on the left; the other vertices of G' correspond to matched vertices on the left.

From a left vertex u, take an original edge (u, v) to the right side. If v is unmatched, add an edge in G' from u to S. If v is matched to a left vertex u', add an edge from u to u'. Thus reaching a matched right vertex is immediately "folded back" through its matching edge. The same rule applies to S, except that S represents all unmatched left vertices.

A return path from S to S in G' corresponds to an augmenting path in G. Therefore, to find the next augmenting path, run a random walk in G' starting at S and wait for it to return to S.

3.2 Return-Time Analysis

Although G' is directed, every vertex has equal in-degree and out-degree. If u is a matched left vertex, then one of its k original edges is its matching edge, so

$$\text{outdeg}(u) = k - 1.$$

Its incoming edges correspond to original edges reaching the right vertex matched to u, except for the matching edge itself, so

$$\text{indeg}(u) = k - 1.$$

For the special vertex S, there are $n - t$ unmatched vertices on each side, each of degree k, hence

$$\text{outdeg}(S) = \text{indeg}(S) = k(n - t).$$

For a directed graph with $\text{indeg}(u) = \text{outdeg}(u)$ for all u, the stationary distribution is again proportional to degree. The total out-degree in G' is

$$k(n - t) + t(k - 1) = kn - t.$$

Thus

$$\pi_S = \frac{k(n-t)}{kn-t} \geq \frac{k(n-t)}{kn} = \frac{n-t}{n}.$$

By the return-time formula,

$$h_{SS} = \frac{1}{\pi_S} \leq \frac{n}{n-t}.$$

Therefore, when the matching size is t, the expected time to find the next augmenting path is at most $n/(n - t)$. Summing over all t,

$$\sum_{t=0}^{n-1} \frac{n}{n-t} = n \sum_{r=1}^{n} \frac{1}{r} = nH_n = O(n \log n).$$

Algorithm 1: Randomized Perfect Matching Algorithm for Regular Bipartite Graphs

Input: A k-regular bipartite graph $G = (L \cup R, E)$, where $|L| = |R| = n$
Output: A perfect matching M
Initialize $M \leftarrow \emptyset$
while $|M| < n$ **do**
  Construct the auxiliary graph G' from M
  Run a random walk from S until it returns to S
  Extract the corresponding augmenting path in G
  Flip the augmenting path to increase $|M|$ by 1
return M

4 Hitting, Commute, and Cover Times

The hitting time h_{uv} is the expected time to reach v starting from u. The commute time between u and v is

$$C_{uv} = h_{uv} + h_{vu}.$$

The cover time starting from u, denoted $C_u(G)$, is the expected time for a random walk starting at u to visit every vertex at least once.

4.1 Examples

For the complete graph K_n, the walk is close to repeatedly sampling independent uniform vertices. This is essentially the coupon collector problem, so the cover time is

$$O(n \log n).$$

This suggests that $n \log n$ is a natural benchmark for random exploration, since even nearly independent samples take this long to see all vertices.

For the path P_n, covering the graph requires traveling from one end to the other. A standard one-dimensional random-walk (gambler's-ruin) calculation gives cover time

$$O(n^2).$$

The lollipop graph consists of a clique of size about $n/2$ attached to a path of length about $n/2$. Its cover time is much worse. To leave the clique, the walk must first hit the special clique vertex attached to the path. By the return-time intuition inside the clique, a fixed clique vertex has stationary probability about $2/n$, so hitting such a vertex takes $\Theta(n)$ steps on average. From there, the walk chooses the single edge into the path with probability $\Theta(1/n)$. Hence one serious attempt to enter the stem costs $\Theta(n^2)$ steps on average.

Once the walk enters the stem, it behaves like an unbiased walk on a path. Starting near one end of a path of length $\Theta(n)$, the probability of reaching the far end before returning to the clique is only $O(1/n)$. Therefore about $\Theta(n)$ attempts are needed. Combining the two estimates gives cover time

$$\Omega(n^3).$$

5 General Cover-Time Upper Bound

Theorem 5.1. *For any connected undirected graph $G = (V, E)$ with $n = |V|$ and $m = |E|$,*

$$C_u(G) = O(mn)$$

for every starting vertex u.

Proof Sketch. Consider the Markov chain whose states are directed edges of G. If the current state is (x, u), then the walk is at u, and the next state is (u, v), where v is chosen uniformly among the neighbors of u.

There are $2m$ directed edges. The stationary distribution on directed edges is uniform:

$$\pi_e = \frac{1}{2m}.$$

Indeed, for a fixed directed edge $e = (u, v)$, the previous edge must be one of the $\deg(u)$ directed edges entering u. Therefore

$$\sum_{\text{directed } f \text{ entering } u} \frac{1}{2m} \cdot \frac{1}{\deg(u)} = \deg(u) \cdot \frac{1}{2m} \cdot \frac{1}{\deg(u)} = \frac{1}{2m}.$$

By the return-time formula, the expected return time to any directed edge is $2m$.

Now let u and v be adjacent. If the walk has just traversed (u, v), then it is at v. Returning to the same directed edge requires the walk to go from v back to u and then traverse (u, v) again, which contains a commute across the edge. Hence the commute time across an edge is at most $2m$.

Take a spanning tree T of G. Fix a depth-first traversal of T that starts and returns to the initial vertex. This traversal crosses each tree edge once in each direction, so its expected length under the random walk is bounded by the sum of the corresponding commute times over the tree edges. Thus

$$C_u(G) \le \sum_{e \in T} C_e \le 2m(n-1),$$

since T has $n-1$ edges and each tree-edge commute time is at most $2m$. Hence $C_u(G) = O(mn)$. □

6 Further Remarks

- The $O(n \log n)$ cover time of K_n gives a useful benchmark for fast cover times. In K_n, the walk behaves almost like drawing independent random vertices, so the coupon-collector phenomenon gives the $n \log n$ scale. The lecturer noted that one can beat this on certain directed graphs, such as a directed cycle, whose cover time is $\Theta(n)$. For undirected random walks, this $n \log n$ scale is a useful baseline for fast exploration.
- The $\Omega(n^3)$ cover-time behavior of the lollipop graph is not limited to one specially chosen starting vertex. It holds for essentially any starting vertex that is not already very close to the end of the stem.
- The next lecture will introduce expander graphs, which are graphs where random walks mix rapidly and converge to their stationary distribution very quickly.

7 Summary

The main takeaways are:

- For an undirected graph, $\pi_u = \deg(u)/(2m)$.
- The expected return time satisfies $h_{uu} = 1/\pi_u$.
- This identity gives an $O(n \log n)$-expected-time randomized algorithm for finding a perfect matching in a k-regular bipartite graph.
- Cover time varies greatly by graph: K_n has $O(n \log n)$, paths have $O(n^2)$, and lollipop graphs have $\Omega(n^3)$.
- Every connected undirected graph satisfies $C_u(G) = O(mn)$.

References

[1] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.