6
Random Walks

6.1 Introduction

One major abstraction used in the design and analysis of randomized algorithms is that of **Markov Chains (MCs)**. We can often model an algorithm's behavior as a process that moves between different "states" (e.g., the current configuration of the algorithm) via probabilistic "transitions".

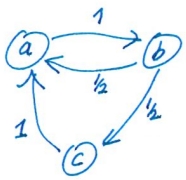

Figure 6.1: A simple Markov Chain.

When analyzing algorithms using this abstraction, we are interested in several key properties of the corresponding Markov Chain:

- **Hitting Time:** What is the expected time to reach a "final" state (e.g., termination of the algorithm)?
- **Long-term Behavior (Stationary Distribution):** What does the distribution over states look like after many steps?
- **Mixing Time:** How quickly does the short-term behavior approach the long-term behavior?

In this lecture, we will first cover some general theorems about Markov Chains. We will then focus on a special case: random walks on undirected graphs, and explore their surprising connection to electrical networks. We will also look at applications to algorithmic problems, including algorithms for 2-SAT and finding matchings.

6.2 Markov Chains

6.2.1 Definitions

A Markov Chain is defined by a set of states, which we can visualize as a directed graph $\vec{G} = (V, A)$.

Definition 6.1 (Markov Chain)**.** A (discrete-time) Markov Chain is defined on a set of states V. Each arc (i, j) has a weight $P_{ij} \geq 0$, representing the **transition probability** from state i to state j. These probabilities must satisfy:

$$\sum_{j \in V} P_{ij} = 1 \quad \forall i \in V.$$

The matrix $P = (P_{ij})_{i,j \in V}$ is the **transition matrix**. (Self-loops are allowed.)

This defines a stochastic process $X_0, X_1, X_2, \ldots, X_t, \ldots$, where $X_t \in V$ is the state at time t. The defining property of a Markov chain (the Markov property or memoryless property) is that the next state depends only on the current state, independently of the history:

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

6.2.2 Evolution of the Distribution

Let $\pi^{(t)}$ be the probability distribution over the states at time t, represented as a row vector, where $\pi_i^{(t)} = \Pr(X_t = i)$. The distribution evolves according to the transition matrix:

$$\pi_j^{(t+1)} = \sum_{i \in V} \Pr(X_t = i) \cdot \Pr(X_{t+1} = j \mid X_t = i) = \sum_{i \in V} \pi_i^{(t)} P_{ij}.$$

In vector-matrix notation, this is:

$$\pi^{(t+1)} = \pi^{(t)} P.$$

By induction, the distribution at time t is $\pi^{(t)} = \pi^{(0)} P^t$.

6.2.3 Stationary Distributions and Convergence

We are often interested in the long-term behavior of the chain.

Definition 6.2 (Stationary Distribution)**.** A probability distribution π is a **stationary distribution** for a Markov Chain with transition matrix P if it satisfies:

$$\pi = \pi P.$$

It is a fixed point for the mapping $x \mapsto xP$.

To understand when a stationary distribution exists and when the chain converges to it, we need a few structural definitions.

Definition 6.3. A Markov Chain is **strongly connected** (or **irreducible**) if for any two states i, j, there is a non-zero probability of eventually getting from i to j.

Definition 6.4. A Markov Chain is **aperiodic** if for all states i, the greatest common divisor (GCD) of the lengths of all paths from i back to i is 1. In other words, for any vertex i, if we start at i, there is a time t_0, so that for all $t \geq t_0$ a return is possible.

Figure 6.2: A simple example of a periodic Markov Chain.

A (finite) chain that is both strongly connected and aperiodic is called **ergodic**.

Theorem 6.5 (Fundamental Theorem of Markov Chains)**.** *For any strongly connected finite Markov Chain, there exists a unique stationary distribution π. Moreover, for any start distribution $\pi^{(0)}$, the time-averaged distribution converges to π:*

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \pi^{(t)} = \pi.$$

If the chain is also aperiodic (ergodic), then for any starting distribution $\pi^{(0)}$, the distribution converges to π:

$$\lim_{t\to\infty} \pi^{(t)} = \pi.$$

6.2.4 *Calculating the Stationary Distribution*

We can find the stationary distribution by solving the system of linear equations $\pi = \pi P$, along with the normalization constraint $\sum_i \pi_i = 1$.

Example: Consider the chain in Figure 6.1. The system of equations $\pi = \pi P$ is:

$$\begin{aligned}
\pi_a &= \frac{1}{2}\pi_b + \pi_c \\
\pi_b &= \pi_a \\
\pi_c &= \frac{1}{2}\pi_b
\end{aligned}$$

Substituting $\pi_b = \pi_a$ and $\pi_c = \frac{1}{2}\pi_a$ into the normalization constraint $\pi_a + \pi_b + \pi_c = 1$:

$$\pi_a + \pi_a + \frac{1}{2}\pi_a = 1 \implies \frac{5}{2}\pi_a = 1.$$

So, $\pi_a = 2/5$. The stationary distribution is $\pi = (2/5, 2/5, 1/5)$.

6.2.5 *The Return Time Theorem*

A very useful theorem relates the stationary distribution to the expected time it takes for the chain to return to its starting state.

Definition 6.6 (Return Time)**.** The (expected) **return time** for a state x, denoted r_x, is the expected number of steps to return to x, given that we start at $X_0 = x$.

Theorem 6.7 (Return Time Theorem)**.** *Suppose a (finite) Markov Chain is strongly connected. Then the expected return time for state x is:*

$$r_x = \frac{1}{\pi_x}.$$

Loosely speaking, the Markov chain spends a fraction π_x of its time at x, so it should take $1/\pi_x$ time on average to return.

6.3 *Application: Perfect Matchings in Bipartite Regular Graphs*

Let's use the Return Time Theorem to analyze a fast randomized algorithm for finding perfect matchings in bipartite regular graphs.

Let $G = (L \cup R, E)$ be a d-regular bipartite graph ($|L| = |R| = n$). It is known that such a graph always has a perfect matching.

We present the core idea of a randomized algorithm by Goel, Kapralov, and Khanna that runs in $O(n \log n)$ time. We remark that a deterministic algorithm that runs in linear time $O(nd)$ was discovered before by Cole, Ost, Schirra in 2001 but the randomized algorithm that we explain here is very clean. The algorithm uses the standard augmenting path approach, but finds the paths using random walks.

Lemma 6.8. *Suppose we have a matching M of size $n - k$ (for $1 \leq k \leq n$). We can find an augmenting path in expected $O(n/k)$ steps using a random walk on a suitably defined graph.*

If this lemma holds, the total expected length for all n augmentations is:

$$\sum_{k=1}^{n} O\left(\frac{n}{k}\right) = O\left(n \sum_{k=1}^{n} \frac{1}{k}\right) = O(n \log n).$$

Proof Sketch of Lemma. We construct an auxiliary directed graph G' based on G and the current matching M.

1. Start with G. Direct all edges from L to R.
2. Contract all matched edges in M.
3. Add a new "source" node s.
4. For every unmatched node $u \in L$, add d parallel edges directed from s to u.

5. For every unmatched node $v \in R$, add d parallel edges directed from v to s.

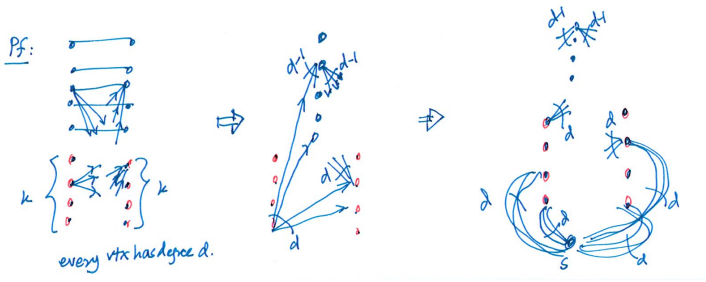

Figure 6.3: Illustration of the regular matching algorithm.

Crucially, this construction ensures that the in-degree equals the out-degree for every node (the graph is balanced or **Eulerian**). Let d_v denote this degree. For the source s, there are k unmatched nodes in L and k in R. So s has kd outgoing edges and kd incoming edges. $d_s = kd$.

Fact 6.9. In a strongly connected, balanced directed graph, the stationary distribution of a standard random walk (choosing an outgoing edge uniformly) is given by:

$$\pi_v = \frac{d_v}{\sum_w d_w}.$$

(This can be verified by checking the balance equations $\pi P = \pi$).

A random walk starting at s and returning to s corresponds exactly to traversing an augmenting path in G.

We use the Return Time Theorem to bound the expected time to return to s, which is $r_s = 1/\pi_s$. The total degree $\sum_w d_w$ is the total number of edges in G'. The original graph G has $m = nd$ edges. We added $2kd$ edges incident to s and removed $n - k$ edges in the contractions.

$$\sum_w d_w = nd + 2kd - (n-k) \leq nd + 2kd.$$

$$E[\text{Return time to } s] = \frac{1}{\pi_s} = \frac{\sum_w d_w}{d_s} = \frac{nd + 2kd}{kd} = \frac{n}{k} + 2.$$

Thus, a random walk will find an augmenting path in expected $O(n/k)$ steps. □

6.4 *Random Walks on Graphs*

We now focus on random walks on undirected graphs $G = (V, E)$. This is a special, very common case of Markov Chains.

The process is simple: if we are at vertex $X_t = v$, we move to a neighbor u chosen uniformly at random in the next step.

$$\Pr(X_{t+1} = u \mid X_t = v) = \frac{1}{\deg(v)} \quad \text{if } (u, v) \in E.$$

We are interested in several fundamental questions:

1. **Hitting Probability:** What is the chance of reaching state A before state B?
2. **Hitting Time (H_{uv}):** How long does it take on average to reach v starting from u?
3. **Stationary Distribution (π):** In the long run, what fraction of time is spent at each vertex?
4. **Cover Time:** How long does it take to visit all vertices in the graph?

6.4.1 *Example: The Line Graph (Gambler's Ruin)*

Consider the simplest setting: a path graph (the line) with vertices $\{0, 1, \ldots, n\}$.

Figure 6.4: Random walk on the line graph.

If $X_t = k$ (where $0 < k < n$), then $X_{t+1} = k + 1$ w.p. $1/2$, and $X_{t+1} = k - 1$ w.p. $1/2$. We analyze this assuming the endpoints 0 and n are absorbing barriers (once reached, the walk stops). This is often called the "Gambler's Ruin" problem: starting with k dollars, what is the probability of reaching n dollars (jackpot) before reaching 0 (bankrupt)?

Q1: Hitting Probability. Let p_j be the probability of hitting n before 0, starting at j. The boundary conditions are $p_n = 1$ and $p_0 = 0$. For $0 < j < n$, we have the recurrence:

$$p_j = \frac{1}{2} p_{j-1} + \frac{1}{2} p_{j+1}.$$

This is a system of linear equations. The unique solution is:

$$p_j = \frac{j}{n}.$$

Q2: Hitting Time. Let T_j be the expected time to hit one of the endpoints (0 or n), starting at j. The boundary conditions are $T_0 = T_n = 0$. For $0 < j < n$, the recurrence is (taking one step first):

$$T_j = 1 + \frac{1}{2}T_{j-1} + \frac{1}{2}T_{j+1}.$$

Again, we solve the linear system. The unique solution is:

$$T_j = j(n - j).$$

The maximum expected time occurs starting near the middle ($j = n/2$), where $T_{n/2} \approx n^2/4$.

6.5 *Application: 2-SAT*

We can use the analysis of the random walk on the line to analyze a simple randomized algorithm for the 2-Satisfiability problem (2-SAT), due to Papadimitriou.

Algorithm 2: Randomized Algorithm for 2-SAT (Papadimitriou)

2.1 Start with an arbitrary truth assignment X.
2.2 **while** *φ is not satisfied by X* **do**
2.3 Pick any unsatisfied clause $C = (l_1 \vee l_2)$.
2.4 Pick one of the literals in C uniformly at random, and flip the value of its underlying variable in X.
2.5 **return** X

Theorem 6.10. *If the formula φ is satisfiable, the algorithm finds a satisfying assignment in expected $O(n^2)$ time steps.*

Proof. Let X^* be a satisfying assignment (which exists by assumption). We analyze the progress of the algorithm by tracking the Hamming distance between the current assignment X_t and X^*.

Let $D_t = \|X_t - X^*\|_{\text{Hamming}}$ be the number of variables where X_t and X^* differ. We want to find the expected time until $D_t = 0$.

Suppose $D_t = k > 0$. The algorithm picks an unsatisfied clause C. Since X^* satisfies C but X_t does not, at least one of the variables in C must have different values in X_t and X^*.

The algorithm randomly flips one variable in C.

- With probability $p \geq 1/2$, it flips a variable whose value in X_t disagrees with X^*. This decreases the distance: $D_{t+1} = D_t - 1$. (If both variables disagree, $p = 1$).

- With probability $1 - p \leq 1/2$, it flips a variable whose value in X_t agrees with X^* (this might happen if only the other variable disagreed). This increases the distance: $D_{t+1} = D_t + 1$.

This process can be modeled as a (biased) random walk on the line $\{0, 1, \ldots, n\}$. The state space is the distance D_t. The process stops when it hits 0.

The expected time to reach 0 in this biased random walk is bounded by the expected time for an unbiased random walk (where probabilities are exactly $1/2$) to hit 0 or n. As we saw in the analysis of the line graph, the maximum expected time for the unbiased walk is $O(n^2)$.

Since the actual process is biased towards 0, and we might also stop early if we hit a different satisfying assignment, the expected stopping time is only decreased. Therefore, the expected number of steps is $O(n^2)$. □

This algorithm is (almost) memoryless and very simple. It also suggests ways to extend to 3-SAT and get fast (but still exponential) algorithms.

6.6 *Random Walks and Electrical Networks*

There is a deep and beautiful connection between random walks on undirected graphs and electrical networks. This analogy provides powerful tools for analysis.

Consider an undirected graph $G = (V, E)$. We transform it into an electrical network by replacing every edge with a 1-Ohm resistor.

6.6.1 *Electrical Laws*

We review the fundamental laws governing electrical flow:

- **Ohm's Law:** If the voltages (potentials) at nodes u and v are ϕ_u and ϕ_v, the current I_{uv} flowing from u to v is $I_{uv} = (\phi_u - \phi_v)/R_{uv}$. Since $R_{uv} = 1$ here, $I_{uv} = \phi_u - \phi_v$.

- **Kirchoff's Law:** For any node (except the source and sink), the total current flowing into the node equals the total current flowing out. The net flow is zero.

6.6.2 *Hitting Probabilities and Voltages*

Let's revisit the hitting probability question. Let p_v be the probability that a random walk starting at v hits s before hitting t. We have the boundary conditions $p_s = 1$ and $p_t = 0$. For any other node $v \notin$

$\{s, t\}$, the probability is the average of its neighbors (the harmonic property):

$$p_v = \frac{1}{\deg(v)} \sum_{u \sim v} p_u.$$

Now consider the electrical network. Suppose we attach a 1-volt battery between s and t, setting the voltage $\phi_s = 1$ and $\phi_t = 0$. Let's analyze the voltages ϕ_v at the other nodes.

By Kirchoff's law, for $v \notin \{s, t\}$, the net flow is zero:

$$\sum_{u \sim v} I_{uv} = 0 \implies \sum_{u \sim v} (\phi_u - \phi_v) = 0.$$

$$\sum_{u \sim v} \phi_u = \deg(v) \cdot \phi_v \implies \phi_v = \frac{1}{\deg(v)} \sum_{u \sim v} \phi_u.$$

The voltages satisfy the exact same system of linear equations as the hitting probabilities, with the same boundary conditions. Since the solution to this system is unique, we have the following remarkable equivalence:

Theorem 6.11. *The probability p_v that a random walk starting at v hits s before t is equal to the voltage ϕ_v when we set $\phi_s = 1$ and $\phi_t = 0$.*

6.6.3 Commute Time and Effective Resistance

This analogy extends to hitting times. We define the following quantities:

- **Hitting Time** H_{uv}: The expected time to reach v starting from u.
- **Commute Time** C_{uv}: The expected time to go from u to v and back to u. $C_{uv} = H_{uv} + H_{vu}$.

Definition 6.12. The **effective resistance** $R_{\text{eff}}(u, v)$ between nodes u and v is the voltage difference required to drive 1 unit of current from u to v.

The connection between these concepts is captured by the following central theorem:

Theorem 6.13 (Commute Time Theorem)**.** *For a graph with m edges, the commute time between u and v is given by:*

$$C_{uv} = 2m \cdot R_{\mathit{eff}}(u, v).$$

Proof. We prove this using the principle of superposition for electrical flows. Let $d_v = \deg(v)$.

Setting 1: Flow to t. Set the potential at node v to be $\phi_v = H_{vt}$. Note that $\phi_t = H_{tt} = 0$. Let's look at the net current flowing out of a node $v \neq t$.

$$I_{\text{out}}(v) = \sum_{u \sim v} (\phi_v - \phi_u) = \sum_{u \sim v} (H_{vt} - H_{ut}).$$

Recall the definition of hitting time: $H_{vt} = 1 + \frac{1}{d_v}\sum_{u\sim v} H_{ut}$. Rearranging this: $d_v H_{vt} - \sum H_{ut} = d_v$. So, $I_{\text{out}}(v) = d_v$.

A current of d_v leaves every node $v \neq t$. All this current must reach t. The total current entering t is $\sum_{v\neq t} d_v = 2m - d_t$.

Setting 2: Flow from s. Set the potential at node v to be $\phi'_v = -H_{sv}$. Note that $\phi'_s = -H_{ss} = 0$. By an identical calculation, a current of d_v must *enter* every node $v \neq s$. The total current leaving s is $2m - d_s$.

Setting 3: Superposition. Define the new potential $\Psi_v = \phi_v + \phi'_v = H_{vt} - H_{sv}$.

The voltage difference between s and t is:

$$\begin{aligned}\Psi_s - \Psi_t &= (H_{st} - H_{ss}) - (H_{tt} - H_{ts}) \\ &= (H_{st} - 0) - (0 - H_{ts}) = H_{st} + H_{ts} = C_{st}.\end{aligned}$$

Now let's look at the current flow. The net current under Ψ is the sum of the currents under ϕ and ϕ'. The total current leaving s is d_s (from Setting 1) $+(2m - d_s)$ (from Setting 2) $= 2m$. Similarly, the total current entering t is $2m$.

We have constructed a flow where $2m$ units of current flow from s to t, driven by a potential difference of C_{st}. By the definition of effective resistance ($R = V/I$):

$$R_{\text{eff}}(s,t) = \frac{\text{Voltage Difference}}{\text{Current Flow}} = \frac{C_{st}}{2m}.$$

Rearranging gives $C_{st} = 2m \cdot R_{\text{eff}}(s,t)$. □

6.6.4 *Applications of the Electrical Analogy*

Revisiting the Line Graph. Consider the line graph $\{0, 1, \ldots, n\}$. It has $m = n$ edges.

1. Hitting time from 0 to n. The effective resistance between 0 and n is the sum of the resistors in series: $R_{\text{eff}}(0,n) = n$. The commute time is $C_{0n} = 2m \cdot R_{\text{eff}}(0,n) = 2n \cdot n = 2n^2$. By symmetry, $H_{0n} = H_{n0} = \frac{1}{2}C_{0n} = n^2$.

2. Hitting time from i to 0. $R_{\text{eff}}(0,i) = i$. The commute time is $C_{0i} = 2n \cdot i$. $C_{0i} = H_{0i} + H_{i0}$. The walk from 0 to i on the line $0 \ldots n$ is the same as on the line $0 \ldots i$ (as we must hit i before going further), so $H_{0i} = i^2$. Therefore, $H_{i0} = C_{0i} - H_{0i} = 2ni - i^2$. This can also be written as $n^2 - (n-i)^2$.

Cover Time. The **cover time** is the expected time for a random walk to visit all vertices in the graph.

Theorem 6.14. *In any connected graph with n vertices and m edges, the expected time to visit all vertices (cover time), starting from any vertex, is at most* $2m(n-1)$.

This implies a randomized, memoryless algorithm for checking graph connectivity.