CS 761 : Randomized Algorithms , Spring 2023, Waterloo

Lecture 12 : Random walks

Random walks is an important technique in random sampling, and also in designing fast algorithms.

We start with the basic questions and then see the fundamental theorem of Markov chains (without proofs).

Then, we see two applications of the fundamental theorem, one about pagerank and another about designing a fast algorithm for perfect matching in regular bipartite graph.

Overview

Given a graph, a random walk starts from a vertex, at each time step the walk moves to a uniformly random neighbor of the current vertex, and repeat.

Some of the basic mathematical questions are:

(1) Is there a limiting distribution of the random walk? If so, what is it? (stationary distribution)

(2) How long does it take before the walk approaches the limiting distribution? (mixing time)

(3) Starting from a vertex s, what is the expected number of steps to first reach t? (hitting time)

(4) How long does it take to reach every vertex at least once? (cover time)

There are two main approaches to questions (1) and (2), one is probabilistic and uses the idea of "coupling" two random processes. another is spectral and uses the eigenvalues of the adjacency matrix.

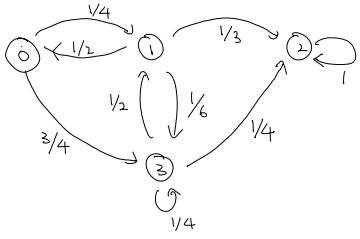
We will study the probabilistic approach in analyzing mixing time.

Questions (3) and (4) are best answered when we view the graph as an "electrical network", which we will study in the next lecture.

Today we just state a fundamental result about (1) and see some interesting applications.

Markov Chain

We analyze the general problem of random walk on a directed graph.



each vertex corresponds to a state

an arc corresponds to the transition prob. from state i to state j.

We can also formulate the problem as a matrix problem

$$\begin{array}{c} \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1/4 & 0 & 3/4 \\ 1/2 & 0 & 1/2 & 1/4 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1/4 & 0 & 3/4 \\ 1/2 & 0 & 1/3 & 1/6 \\ 0 & 0 & 1 & 0 \\ 0 & 1/2 & 1/4 & 1/4 \end{pmatrix} \end{array}$$, i.e. P_{ij} = transition prob. from state i to state j.

Let X_t be the state at time t.

Let $p_t(i)$ be the probability of being at state i at time t.

For example, $\vec{p_0} = (1, 0, \dots, 0)$ if the walk starts at state 0 at time 0,

or $\vec{p_0} = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$ if the walk starts at a random state at time 0.

By the definition, it follows that $p_{t+1}(j) = \sum_{i=0}^{n-1} p_t(i) \cdot P_{i,j} \quad \forall j$, and we can write it compactly as $\vec{p_{t+1}} = \vec{p_t} P$ and more generally $\vec{p_{t+m}} = \vec{p_t} P^m$, where the vectors are row vectors.

This random process is called a Markov chain because it forgets about the past, i.e.

$$Pr(X_t = a_t \mid X_{t-1} = a_{t-1}, X_{t-2} = a_{t-2}, \dots, X_1 = a_1, X_0 = a_0) = Pr(X_t = a_t \mid X_{t-1} = a_{t-1}) = P_{a_{t-1}, a_t}.$$

States

We will assume all Markov chains are finite in this course.

A Markov chain is <u>irreducible</u> if the underlying directed graph is strongly connected, i.e. there is a directed path from i to j and from j to i for every pair of vertices $i, j \in V$.

In other words, for every i, j, there exists ℓ such that $Pr(X_{t+\ell} = s_j \mid X_t = s_i) > 0$.

The <u>period</u> of a state i is defined as $d(s_i) := \gcd\{ t \mid P^t_{i,i} > 0 \}$.

A state i is <u>aperiodic</u> if $d(s_i) = 1$.

A Markov chain is aperiodic if all states are aperiodic; otherwise it is periodic.  is periodic.

For example, random walks on a bipartite graph is aperiodic as $d(s_i) = 2 \quad \forall i \in V$.

We use irreducibility and aperiodicity to get the following property.

<u>Lemma</u> For any finite, irreducible, and aperiodic Markov chain, there exists a $T < \infty$ such that

$(P^t)_{i,j} > 0$ for any i, j for any $t \geq T$.

Stationary distribution

A stationary distribution of a Markov chain is a probability distribution $\vec{\pi}$ such that $\vec{\pi} = \vec{\pi} P$.

Informally, $\vec{\pi}$ is a "steady / equilibrium / fixed point" state, as $\vec{\pi} = \vec{\pi} P^t$ for any $t \geq 0$.

Intuitively, given any finite, irreducible, and aperiodic Markov chain, after we run it long enough, then we will completely forget about the history and "converge" to the same distribution.

Given two probability distributions $\vec{p} = (p_1, p_2, \ldots, p_n)$ and $\vec{q} = (q_1, q_2, \ldots, q_n)$, the total variation distance of $\vec{p}$ and $\vec{q}$ is defined as $d_{TV}(p,q) = \frac{1}{2}\sum_{i=1}^{n} |p_i - q_i|$.

We say that $\vec{p}_t$ converges to $\vec{q}$ if $\lim_{t\to\infty} d_{TV}(\vec{p}_t, \vec{q}) = 0$.

One more definition: the return time from i to i is defined as $H_i := \min\{ t \geq 1 \mid X_t = s_i, X_0 = s_i \}$.

The expected return time is defined as $h_i := E[H_i]$.

The following theorem is known as the fundamental theorem of Markov chains.

Theorem For any finite, irreducible, and aperiodic Markov chain, the followings hold.

1. There exists a stationary distribution $\vec{\pi}$.
2. The distribution $\vec{p}_t$ will converge to $\vec{\pi}$ as $t \to \infty$, no matter what is the initial distribution $\vec{p}_0$.
3. There is a unique stationary distribution.
4. $\pi_i = \lim_{t\to\infty} (P^t)_{i,i} = \frac{1}{h_i}$.

We won't prove this theorem today, and will prove it later using the coupling method.

Roughly speaking, the probabilistic approach goes as follows: Two random walks are indistinguishable after they "meet" at the same vertex at some time step. (This statement can be made precise by the "coupling" argument) By the previous lemma using irreducibility and aperiodicity, the two random walks will meet with some positive probability after T steps no matter where they were. This would imply that the two random walks will meet with probability one eventually, and it follows that they would converge to the same stationary distribution, and thus the uniqueness.

It is intuitive to see that $\pi_i = \frac{1}{h_i}$, as the average return time is h_i.

Pagerank

Suppose we have a directed graph describing the relationships between a set of webpages, i.e. there is an arc from page i to page j iff there is a link from page i to page j.

We want an algorithm to "rank" how important a page is.

Intuitively, a page being linked by many other pages is important, and a page linked by an important page is important.

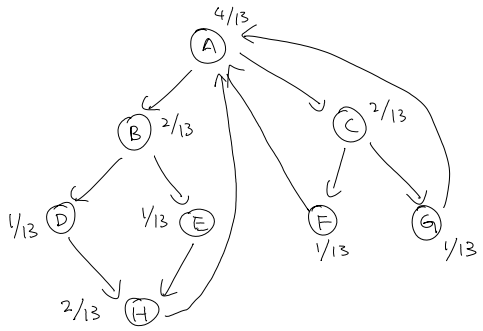
This motivates the following "iterative refinement" algorithm.

Pagerank algorithm:

- Initially, each page has pagerank value $\frac{1}{n}$.
- In each step, each page divides its pagerank value equally to its outgoing link,

and send these equal shares to the pages that it points to.

Each page update its new pagerank value to be the sum of the shares that it receives.



the "equilibrium" pagerank values.

It is not difficult to see that the equilibrium pagerank values are equal to the probabilities of the stationary distribution of the random walk when each entry P_{ij} in P is equal to $\frac{1}{deg^{out}(i)}$, where $deg^{out}(i)$ is the outdegree of vertex i in the graph.

This is because the pagerank values and the stationary probabilities satisfy the same equations: $pagerank(j) = \sum_{i:ij \in E} pagerank(i) / deg^{out}(i)$, which implies that $\vec{pagerank} = \vec{pagerank} \cdot P$, and thus $\vec{pagerank}$ is a stationary distribution of P.

When the graph is finite, irreducible and aperiodic, this is uniquely defined by the fundamental thm.

Therefore, we know that if the graph is strongly connected and aperiodic, then the "equilibrium" pagerank values are unique regardless of the initial values.

In other words, the pagerank values are a function of the graph structure and are well-defined.

In practice, the directed graph may not satisfy these conditions and the following modified process is used:

Fix a number $s > 0$, divide s fraction of its pagerank values to its neighbors, and divide $1-s$ fraction of its pagerank value to all nodes evenly.

This is equivalent to the random walk process that with probability s, go to a random neighbor, and with probability $1-s$, go to a random vertex.

With this modification, the resulting graph is strongly connected and aperiodic, and hence a unique stationary distribution exists.

At the same time, intuitively, this modification does not change the "relative importance" of the vertices.

Furthermore, it also shows the relation between the pagerank value and the return time.

Perfect matching in regular bipartite graphs

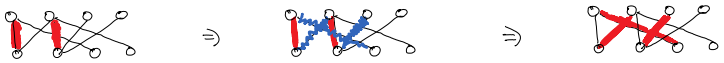
The bipartite matching problem is to find a maximum number of vertex disjoint edges in a bipartite graph.

By Hall's theorem on bipartite matching, it is well-known that a regular bipartite graph always has a

perfect matching (i.e. a regular bipartite graph with $2n$ vertices has a matching of size n).

There is a recent result (2010) showing how to find a perfect matching in such graphs in $O(n \log n)$ time. Note that this is sublinear time when the graph has much more than $n \log n$ edges (e.g. $\Omega(n^2)$ edges). The algorithm and the analysis are very elegant.

The traditional approach is to repeatedly find an "augmenting path" to enlarge the current matching. Pictorically.



current matching augmenting path larger matching

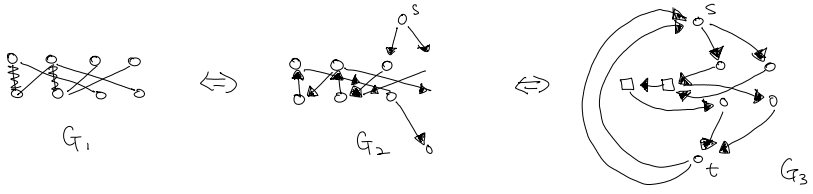
An augmenting path is a path $v_1 - v_2 - \cdots - v_{2k+1}$, where $v_{2i-1} - v_{2i}$ is an edge not in the current matching, $v_{2i} - v_{2i+1}$ is an edge in the current matching, and v_1 and v_{2k+1} are unmatched vertices.

Theorem The current matching is maximum if and only if there is no augmenting path.

Proof idea One direction is easy: If there is an augmenting path, then we can use it to enlarge the matching. Another direction is to show that if the current matching M is not maximum, then there is an augmenting path, by considering the union of M and a larger matching M^*. □

With the theorem, the maximum bipartite matching problem can be reduced to at most n subroutines of finding an augmenting path, which can be done in $O(m)$ time by BFS or DFS.

The new idea is to replace BFS/DFS by random walks. Let me illustrate the algorithm by pictures.



G_1 is the original undirected bipartite graph with a matching M.

G_2 is the directed graph where each edge in the matching M is pointed upward, while all other edges not in the matching are pointed downward.

G_3 is the directed graph obtained from G_2, by contracting each edge in the matching M into a single square node. The source s has $d^{out}(v)$ edges to every unmatched vertex v on top, and every unmatched vertex u at bottom has $d^{in}(u)$ edges to the sink t. Finally, the sink t has $d^{in}(t)$ edges to the source s.

It is not difficult to see that:

① G_1 has an augmenting path.

$\Leftrightarrow$ ② G_2 has a directed path from s to t.

$\Leftrightarrow$ ③ G_3 has a cycle from s to s.

Also, it is not difficult to verify that G_3 is an Eulerian directed graph (i.e. indegree = outdegree for every vertex) if G_1 is a regular graph.

So, if we do a random walk in G_3, then the expected time to find an augmenting path is equal to the expected return time h_s in G_3.

Recall that $h_s = 1/\pi_s$, where π_s is the probability of being in s in the stationary distribution, which is easy to compute in Eulerian directed graphs.

Claim In an Eulerian directed graph $G=(V,E)$, the stationary distribution π is given by $\pi_v = \frac{d^{out}(v)}{|E|}$ $\forall v \in V$.

Proof $\pi_v = \sum_{u:uv\in E} \pi_u P_{u,v} = \sum_{u:uv\in E} \frac{d^{out}(u)}{|E|} \cdot \frac{1}{d^{out}(u)} = \frac{d^{in}(v)}{|E|} = \frac{d^{out}(v)}{|E|} = \pi_v$.

Since $\sum_v \pi_v = \sum_v \frac{d^{out}(v)}{|E|} = 1$, this is the unique stationary distribution. □

Therefore, $h_s = |E(G_3)| / d^{out}(s)$ using the above claim.

In the i-th iteration when there are only i edges in the matching, $d^{out}(s) = (n-i)\cdot d$ assuming G_1 is a d-regular graph.

Note that $|E(G_3)| \le 4|E(G_1)| = 4dn$.

So, in the i-th iteration, $h_s \le 4dn / d(n-i) = 4n/(n-i)$.

Therefore, the total running time is $\sum_{i=0}^{n-1} 4n/(n-i) = O(n\log n)$.

With appropriate simple data structures, the total complexity of the algorithm is $O(n\log n)$.

Note that we don't need to construct G_3 in the algorithm. It is only used in analyzing the expected time to find a path from s to t in G_2, as the random walks in G_2 and G_3 have the same behaviour.

In the algorithm, we just need to keep track of the change in G_2 after updating the partial matching.

Open: Can this approach be extended to non-regular graphs?

References

Basics of random walks can be found in chapter 7 of [MU].

A full proof of the fundamental theorem can be found in the short book "finite Markov chains and algorithmic applications" by Häggström.

The regular bipartite matching algorithm is from the paper "Perfect matching in O(nlogn) time in

regular bipartite graphs" by Goel, Kapralov and Kannan.

Pagerank is from chapter 14 of "Networks, crowds and markets" by Easley and Kleinberg.