CS 466 / 666 : Algorithm Design and Analysis . Spring 2018 . Waterloo.

Lecture 10: Random walks

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

Through the study of random walks, we will begin the second part of the course, where we use linear algebraic ideas to design and analyze algorithms.

Today, we will first see some interesting applications using minimal background knowledge.

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) What is the limiting distribution of the random walk? (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.

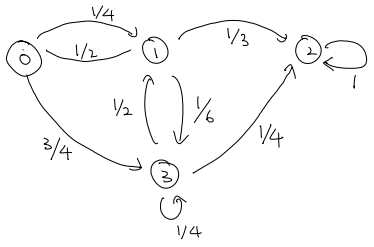
We will study the spectral approach to some depth in the second part of this course, and hopefully we will also have time to see the coupling argument in analyzing mixing time.

Questions (3) and (4) are best answered by viewing the graph as an electrical network, and we will also study this view point to some depth in the second part of the course.

Today we just assume 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

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

, 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 $Pr(X_{t+l} = s_j \mid X_t = s_i) > 0$.

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

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

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 <u>total variation distance</u> 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 $\vec{p}_t$ <u>converges</u> to $\vec{q}$ if $\lim_{t \to \infty} d_{TV}(\vec{p}_t, q) = 0$.

One more definition: the <u>hitting time</u> from i to j is defined as $H_{i,j} := \min \{ t \geq 1 \mid X_t = S_j, X_0 = S_i \}$.

The <u>expected hitting time</u> is defined as $h_{i,j} := E[H_{i,j}]$.

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

<u>Theorem</u> 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. $\pi_i = \lim_{t \to \infty} (P^t)_{i,i} = \frac{1}{h_{i,i}}$.

We won't prove this theorem today, as we will do a spectral proof of a special case later.

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 theorem 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 easy to see that $\pi_i = \frac{1}{h_{i,i}}$, as the average return time is $h_{i,i}$.

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.

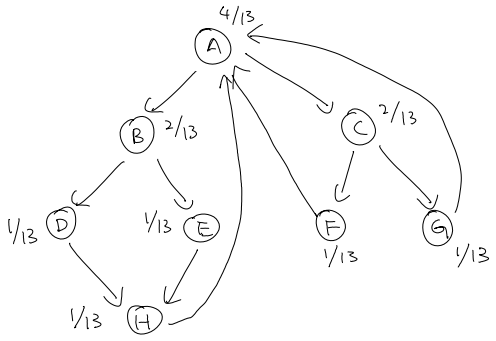
Intuitively, a page being linked by many other pages is important, and a page linked by an important page is important.

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 distribution.

In practice, the directed graph may not satisfy that condition, and the following modified process is used: fix a number $s > 0$, divide s fraction of its pagerank value to its neighbors, divide $1-s$ fraction of its pagerank to all nodes evenly.

This is equivalent to the random walk that with probability s, go to a random neighbor, and

with probability $1-s$, go to a random vertex. Then the resulting graph is strongly connected and aperiodic, and hence a unique stationary distribution exists.

Having a unique "equilibrium" pagerank value shows that the ranking only depends on the graph structure, but not on the initial pagerank values.

It also shows the relation between the pagerank value and the hitting time.

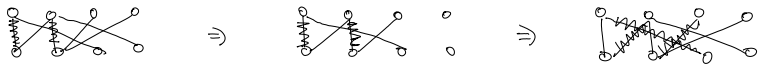
Perfect Matching in Regular Bipartite Graphs

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 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 matching.

Pictorically,



current matching ⇒ augmenting path ⇒ larger matching

An augmenting path is a path $v_1 - v_2 - \cdots - v_{2\ell+1}$, 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 $v_{2\ell+1}$ are unmatched vertices.

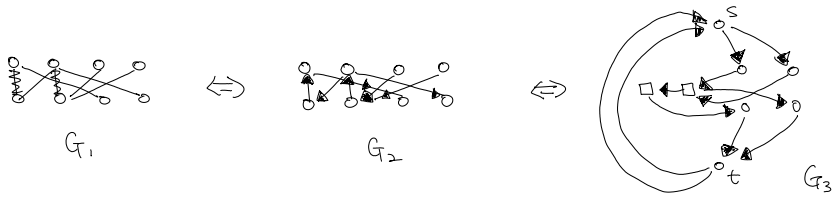
Theorem The current matching is maximum if and only if there is no augmenting path.

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 a larger matching M^* and M. □

So, 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 a BFS (breadth first search).

The new idea is to replace BFS 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 pointing upward, while every other edge (not in the matching) is pointing 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 in bottom has $d^{in}(u)$ edges to the sink t. And t has $d^{in}(t)$ edges to s.

It is not difficult to see that:

① G_1 has an augmenting path

$\Leftrightarrow$ ② G_2 has a directed path from a top unmatched vertex to a bottom unmatched vertex.

$\Leftrightarrow$ 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. the indegree is equal to the outdegree for every vertex) if G_1 is a regular bipartite 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 hitting time $H_{s,s}$ in G_3.

Recall that $H_{s,s} = 1/\pi_s$, 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, the stationary distribution is $\pi_v = \frac{d^{out}(v)}{m}$.

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

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

Therefore, $H_{s,s} = m/d^{out}(s)$.

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

So, in the i-th iteration, $H_{s,s} \leq dn/d(n-i) = n/(n-i)$.

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

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

Open problem: Can you extend this approach to non-regular bipartite graphs?

End of first part

In these lectures, we have seen the basic tools and ideas in the design and analysis of randomized algorithms, and some interesting applications including sublinear algorithms, parallel algorithms, distributed algorithms, etc.

There are some notable topics that we have not covered.

On the algorithmic side, dimension reduction and more generally metric embedding have various important applications and beautiful mathematics, similarly compressive sensing is another interesting area.

On the complexity side, interactive proofs and PCP theorems are very intersting results with profound implications in complexity and hardness of approximations.

In the second part of the course, we will study linear algebraic techniques in the design and analysis of algorithms. First, we will continue our study of random walks. Then, we will move to other topics.

References

Basics of random walks can be found in chapter 7 of "probability and computing" and chapter 6 of "randomized algorithms".

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

The regular bipartite matching algorithm is from the paper "perfect matching in $O(n \log n)$ time in regular bipartite graphs" by Goel, Kapralov and Khanna.

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