CS 270 Combinatorial Algorithms and Data Structures, Spring 2015

Lecture 11: Random walks

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

We will study random walks in more depth in the third part of the course when we study eigenvalues and electrical networks.

Today we see some interesting applications using minimal background knowledge.

Overview

Given a graph, the 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.

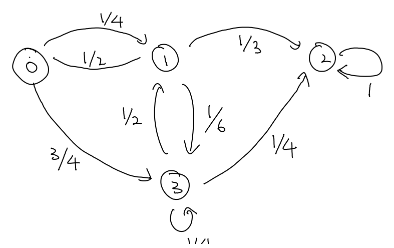
We will study the spectral approach to some depth in the third part of this course, and hopefully we will also have time to see the coupling argument in analyzing mixing time.

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 third 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

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

Let $X(t)$ be the state at time t.

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

e.g. $\vec{p}_0 = (1, 0, 0, \dots, 0)$ if the walk starts at state 0 in time 0,

or $\vec{p}_0 = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$ if the walk starts at a random state in time 0.

By the definition it follows that $\vec{p}_{t+1} = \vec{p}_t P$, and more generally $\vec{p}_{t+m} = \vec{p}_t P^{m-1}$

This random process is called a Markov chain because

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

States

We will assume the Markov chain is finite throughout this lecture.

A Markov chain is irreducible if the corresponding graph is strongly connected

(i.e there is a directed path from i to j and from j to i for every pair $i,j \in V$)

In other words, for every i,j, there exists an n s.t. $Pr(X(m+n)=s_j \mid X(m)=s_i) > 0$

The period of state i is $d(s_i) = \gcd\{n \geq 1 : P_{i,i}^n > 0\}$.

State i is aperiodic if $d(s_i) = 1$.

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

(e.g. random walk in a bipartite graph is periodic since $d(s_i) = 2$ for all i.)

We use irreducibility and aperiodicity to get the following property.

Thm For any finite, irreducible, aperiodic Markov chain, there exists $N < \infty$ s.t. $(P^n)_{i,j} > 0$ for all i,j and for all $n \geq N$.

Stationary distribution

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

Informally, $\vec{\pi}$ is a "steady" state or an "equilibrium" state or a "fixed-point" state, as

$\vec{\pi} = \vec{\pi} P^t$ for any $t \geq 0$.

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

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

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

One more definition: the hitting time is defined as $T_{i,j} = \min\{n \geq 1 : X_n = j \mid X_0 = i\}$.

The mean hitting time (or just hitting time) is defined as $h_{i,j} = E[T_{i,j}]$.

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

Theorem For any finite, irreducible and aperiodic Markov chain:

① There exists a stationary distribution $\vec{\pi}$.

② Any initial distribution $\vec{p}(0)$ will converge to $\vec{\pi}$.

③ There is a unique stationary distribution.

④ $\pi_i = \lim (P^t)_{i,i} = \frac{1}{h_{i,i}}$.

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

Roughly speaking, the probabistic 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 N 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 unique stationary distribution.

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 homepages, i.e. there is an arc from page i to page j if page i links to page j.

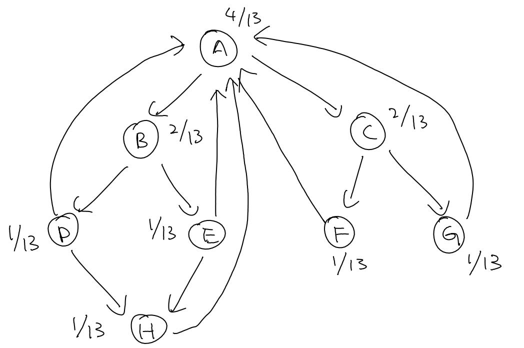
We want an algorithm to "rank" how important a page is.

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

This motivates the following algorithm:

PageRank : - initially every page has PageRank value $1/n$.

- in each step, each page divides its current PageRank value equally to its outgoing links and sends these equal shares to the pages that it points to. Each page updates its new PageRank value to be the sum of the shares it receives.



the "equilibrium" PageRank values.

It is not difficult to see that the equilibrium PageRank values are equal to the probabilities in the stationary distribution of the random walk when each entry $P_{i,j}$ in P is equal to $1/d^{out}(i)$ where $d^{out}(i)$ is the outdegree of i in the graph, since the PageRank values and the stationary probabilities satisfy the same equations: $\text{pagerank}^{(t+1)}(v) = \sum_{u:\, uv \in E} \text{pagerank}^{(t)}(u) \quad \forall v \iff \overrightarrow{\text{pagerank}}^{(t+1)} = \overrightarrow{\text{pagerank}}^{(t)} \cdot P$.

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.

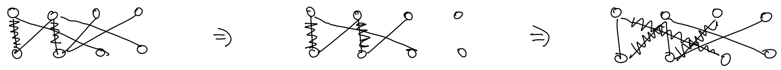
Perfect Matching in Regular Bipartite Graphs

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 $\underset{\text{regular}}{\wedge}$ 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 $v_1 - v_2 - \cdots - v_{2l+1}$, 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_{2l+1} are unmatched vertices.

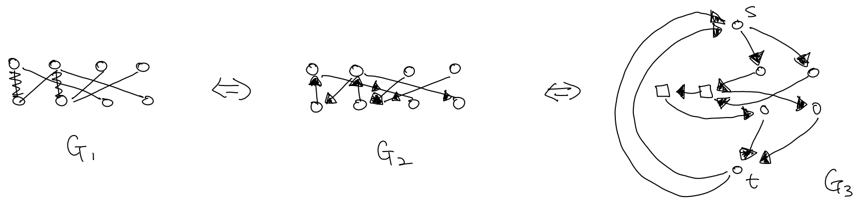
Theorem The current matching is maximum if and only if there is no augmenting path.

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 walk. Let me explain 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 show that ① G_1 has an augmenting path

⇔ ② G_2 has a directed path from a top unmatched vertex to a bottom unmatched vertex

⇔ 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.

<u>Claim</u> In an Eulerian directed graph, the Stationary distribution is $\pi_v = \frac{d^{out}(v)}{m}$ when edge uv is traversed with probability $\frac{1}{d^{out}(u)}$.

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

Since $\sum_v \pi_v = \sum_v \frac{d(v)}{m} = 1$, 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 $O(n \log n)$.

Opon 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 (yet).

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

Actually, as a disguise, we have seen the idea of dimension reduction when we saw the sublinear algorithm for estimating frequency moments, because there we use few numbers to preserve the norm of a vector, and very similar ideas are used for dimension reduction.

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

Next time we will start to study linear programming, but the powerful use of randomness will show again and again and also in spectral analysis and semidefinite programming.

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.