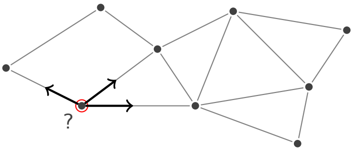
Random walks (and Markov chains)

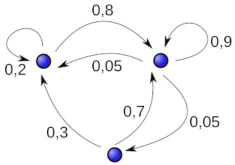
Arnaud Casteigts

Part of Graph Algorithms (14X061)
Masters of Computer Science

University of Geneva

Random Walk





The basic version:

- Choose a neighbor at random
- Move to it
- Repeat

General version:

- Same, but...
- The graph may be directed
- Outgoing probabilities are unequal (sum to 1)

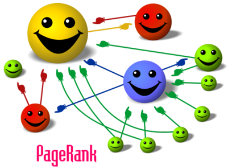
Main concepts

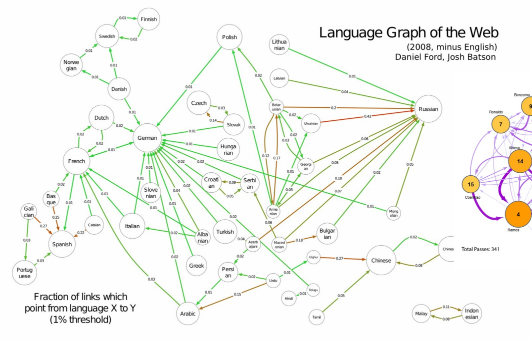
- **Hitting** time: How long until a given vertex is visited from another given vertex
- **Cover** time: How long until all the vertices are visited (from the worst starting vertex)
- **Mixing** time: How long before the probability to be on each vertex no longer depends on the starting position
- **Stationary** distribution: Probability to be on each vertex after the mixing time
- **Return** time: How long before the starting vertex is visited again
- **Recurrent** / **transient** / **absorbing**: a vertex/state is recurrent if the probability to return to it eventually is 1. It is transient otherwise. It is absorbing if we cannot leave it (no outgoing edges).

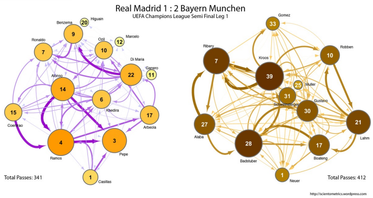
The above times are typically considered **on expectation** or **with high probability**, the latter being stronger. We will focus on expectation only (simpler to calculate).

Applications

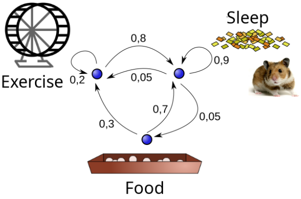
- **Finance**: Stock Market Analysis
- **Physics**: Brownian Motion, Diffusion
- **Computer science**: Graph traversal, Search Engines (→), Analysis of algorithms
- **Biology**: Animal movements, Spread of disease, Genetic evolution
- **Math**: Monte Carlo simulation, Laplace's equation







A concrete example



Transition Matrix:

$$M = \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.7 & 0 & 0.3 \\ 0.8 & 0 & 0.2 \end{bmatrix}$$

Probability distribution for the hamster at any time: $\mathbf{x} = [\ p_s \quad p_f \quad p_e\ ]$ (with $p_s + p_f + p_e = 1$)

Let $\mathbf{x}_0 = [\ 1 \quad 0 \quad 0\ ]$ (initially sleeping), what about the next step?

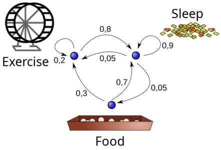
$$\mathbf{x}_1 = \mathbf{x}_0 M = [\ 1 \quad 0 \quad 0\ ] \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.7 & 0 & 0.3 \\ 0.8 & 0 & 0.2 \end{bmatrix} = [\ 0.9 \quad 0.05 \quad 0.05]$$

$$\mathbf{x}_2 = \mathbf{x}_1 M = \mathbf{x}_0 M^2 = [\ 1 \quad 0 \quad 0\ ] \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.7 & 0 & 0.3 \\ 0.8 & 0 & 0.2 \end{bmatrix}^2 = [\ 0.885 \quad 0.045 \quad 0.07]$$

$\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$ is a called **Markov chain**, and M is a **Markov model**.

Observation: $\mathbf{x}_i$ depends only on $\mathbf{x}_{i-1}$, not on earlier information (Markovian property)

Stationary distribution



How to compute the stationary distribution?

We know that $\mathbf{x}_{i+1} = \mathbf{x}M$.

$\rightarrow$ we are looking for $\mathbf{s}$ such that $\mathbf{s}M = \mathbf{s}$

$$\begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix} \begin{bmatrix} 0.9 & 0.05 & 0.05 \\ 0.7 & 0 & 0.3 \\ 0.8 & 0 & 0.2 \end{bmatrix} = \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix}$$

$$\begin{array}{cccc} 0.9s_1 & +0.7s_2 & +0.8s_3 & = s_1 \\ 0.05s_1 & +0s_2 & +0s_3 & = s_2 \\ 0.05s_1 & 0.3s_2 & 0.2s_3 & = s_3 \\ s_1 & +s_2 & +s_3 & = 1 \end{array}$$

$$\begin{array}{cccc} 0.1s_1 & -0.7s_2 & -0.8s_3 & = 0 \\ 0.05s_1 & -s_2 & & = 0 \\ 0.05s_1 & 0.3s_2 & -0.8s_3 & = 0 \\ s_1 & +s_2 & +s_3 & = 1 \end{array}$$

$$\implies \mathbf{s} = \begin{bmatrix} 0.884 & 0,0442 & 0,0718 \end{bmatrix}$$

The hamster spends $\sim 88.4\%$ of its time sleeping.

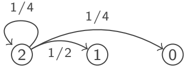
Remark 1: $\mathbf{s}$ is the eigen vector of M corresponding to the eigen value 1.

Remark 2: In the particular case that the graph is undirected and unweighted, each entry s_i of the stationary vector corresponds to $d(v_i)/2m$ (the probability of being at a given vertex is proportional to its degree).

Remark 3: Some random walks may never converge to their stationary distribution (e.g. bipartite graphs).

Analysis of algorithms

Algorithm 1: We want to run an election among n candidates. In each step, each candidate flips a coin. The players having `tails` are eliminated. The players having `heads` play another round. The algorithm succeeds if at some point only one candidate remains.



What is the probability that the algorithm succeeds if $n = 2$?

Let P_n be the probability of success if there are n candidates.

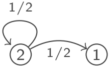
$P_0 = 0$

$P_1 = 1$

$P_2 = \frac{1}{4}P_2 + \frac{1}{2}P_1 + \frac{1}{4}P_0$

$\implies \frac{3}{4}P_2 = \frac{1}{2} \implies \boxed{P_2 = \frac{2}{3}}$

Algorithm 2: Same algorithm, but we restart from scratch whenever it fails.



Now, the probability of success is 1. But how long (# steps) does it take **on expectation**?

Let T_n be the remaining time if there are n candidates.

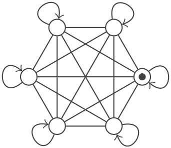
$T_1 = 0$

$T_2 = 1 + \frac{1}{2}T_2 + \frac{1}{2}T_1 \implies \frac{1}{2}T_2 = 1 \implies \boxed{T_2 = 2}$

Remark: We just analysed the **expected hitting time** from vertex 2 to vertex 1 in this graph!

Complete graph

Let's consider a random walk in a complete graph on n vertices (with loops)

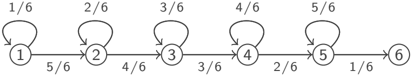


Expected **hitting** time from the stating vertex to a given vertex?

$\rightarrow$ In each step, $1/n$ chances of success
$\rightarrow$ expected time of n.

Expected **cover** time?

Markov model:
(States = #visited)



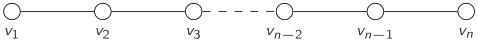
Remark: **cover** time in the original graph = **hitting** time from 1 to 6 in the second graph.

If k vertices remain to visit, what is the probability to move to a new vertex? Answer: k/n.

$\rightarrow$ Expected time before visiting the *next* vertex: n/k.

Total time: $\sum_{k=1}^{n-1} \frac{n}{k} = n \cdot \sum_{k=1}^{n-1} \frac{1}{k} = n \cdot \Theta(\log n) = \Theta(n \log n)$ ("harmonic" numbers)

Path graph



Expected **return** time of v_n?

Recall that stationary distribution of a vertex v is $\frac{d(v)}{2m}$.
$\rightarrow$ We visit it on average every $2m/d(v)$ steps.
$\rightarrow$ The expected return time must be the same, so for v_n: $2m/1 = 2(n-1) = 2n-2$.

Expected **hitting** time from v_1 to v_n?

$H(v_1 \rightarrow v_n) = H(v_1 \rightarrow v_2) + H(v_2 \rightarrow v_3) + \cdots + H(v_{n-1} \rightarrow v_n)$ (linearity of expectation)

How much is $H(v_{n-1} \rightarrow v_n)$?
$\rightarrow$ One less than the return time of v_n (why?)
$\rightarrow$ $2n-3$

How much is $H(v_{n-2} \rightarrow v_{n-1})$?
$\rightarrow$ Same argument, in a graph without v_n (why?)
$\rightarrow$ $2n-5$

$\rightarrow$ $H(v_1 \rightarrow v_n) = 1 + 3 + \cdots + (2n-5) + (2n-3)$
$\rightarrow$ $H(v_1 \rightarrow v_n) = (n-1)^2 = \Theta(n^2)$.

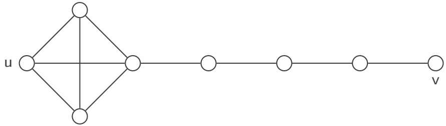
Expected **cover** time of the graph?

$\rightarrow$ at most twice the worst hitting time (why?)
$\rightarrow$ the cover time of the path graph is $\leq 2(n-1)^2 = \Theta(n^2)$

Worst cases

Undirected graphs:

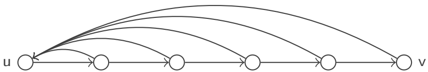
→ Lollipop graph (a $n/2$-clique, connected to a $n/2$-path).



→ Expected **hitting** time from u to v is $\Theta(n^3)$.

Directed graphs:

→ A directed path + return to origin in each vertex



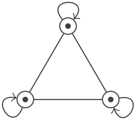
→ Expected **hitting** time from u to v is $\Theta(2^n)$

Coalescing time

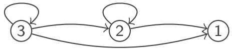
A token on every vertex. Each token performs a random walk. When two tokens meet, they merge.

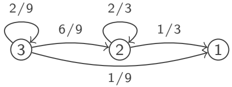
Coalescing time: How long until a single token remains?

Example: on a complete graph with loops:



Markov model:





Coalescing time of the original problem = **hitting** time from 3 to 1 in this graph.

3 choices for three tokens $\rightarrow$ 27 combinations of choices:

- 6 don't merge. $6/27 = 2/9$
- 18 merge two tokens. $18/27 = 6/9$
- 3 merge three tokens. $3/27 = 1/9$

3 choices for two tokens $\rightarrow$ 9 combinations of choices:

- 6 don't merge. $6/9 = 2/3$
- 3 merge. $3/9 = 1/3$

$T_1 = 0$

$T_2 = 1 + \frac{2}{3} T_2 + \frac{1}{3} T_1$

$T_3 = 1 + \frac{2}{9} T_3 + \frac{6}{9} T_2 + \frac{1}{9} T_1$

$\frac{1}{3} T_2 = 0 \implies T_2 = 3$

$\frac{7}{9} T_3 = 1 + \frac{6}{9} T_2 \implies \frac{7}{9} T_3 = 3 \implies \boxed{T_3 = 27/7}$