Applications of Markov Chains: PageRank and Random Walks

Josep Díaz Maria J. Serna Conrado Martínez
U. Politècnica de Catalunya

RA-MIRI 2023–2024

Part

1. Page Rank

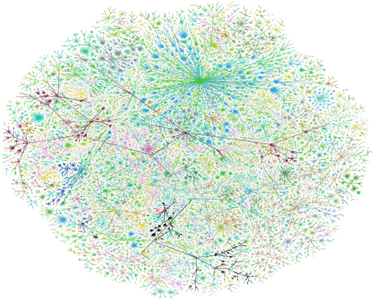
2. Gambler's ruin

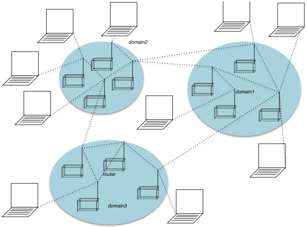
3. More on random walks

Application: Searching in the WWW

Complex network: Highly dynamic graphs with non-trivial topological features that model “real life systems”.

Internet: Undirected graph of server’s and computers connexions, with 12×10^8 vertices (2015)



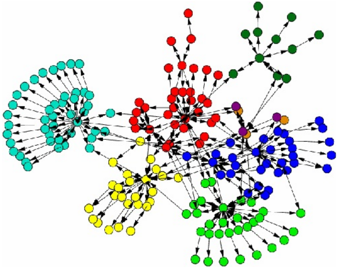


The actual internet is due to Vicent Cerf and Bob Kahn (in the 1980’s)

The Web

Directed graph representing web pages (vertices) and hyperlink connexions between web pages (edges).

Size: 5.28 × 10^9 pages (Nov. 2018)



Developed by Tim Berners-Lee and his team in the CERN (1989-1991)

Hypertext

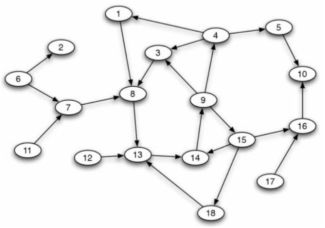
- An easy way to publish information. Users can make documents available to anyone in the internet.
- Each Web page is identify by its URL, so a page can be access following an hyperlink.
- Organizes the Web pages as a network linked by hyperlinks, which links parts of a text into another text, in other machines which could be located very far away.
- This organization transforms the set of documents in a directed graph, in which nodes is a Web document and direct edges.
- The Web is an hypertext system at a scale no one could have anticipated.

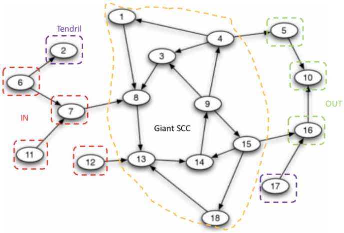
The Web as a graph: Strongly Connected Components

The Web is a directed graph, where the nodes are the Webpages and the edges are the hyperlinks.

In the center of the Web there is a giant Strongly Connected Components (SCC)

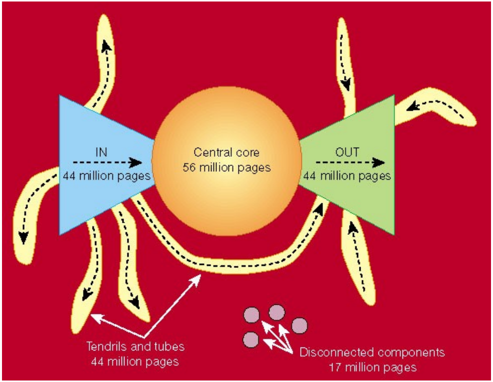
The main search engines and other “starting points” have links to directory type sites which link to all of the major universities, big companies, etc. And many pages in those sites point to the search engines.





The Web as a graph: The Bow-tie structure

A. Broder et all: *Graph structure in the Web*-2000



The details of the structure change continuously but the overall structure remains the same.

Web search

- Information Retrieval investigates how to find relevant docs in a large database, from which the Web is a particular case, we make queries to find answers in the Web but
- The Web is huge, full of untrusted documents random things, false information, web spam, as well as very interesting information.
- In the Web every body is an author and a searcher.
- The Web contains many sources of information, but not all are trustful, which one to trust?
- Two characteristics of the Web are the synonymy (different thing to express the same) and polysemy (multiple meanings for the same word) searching can yield a myriad of different answer, which one is the best one for our needs?

Web search: Link analysis

Instead of using textual match use the link structure of the Web to rank the pages by some measure of authority w.r.t the search.

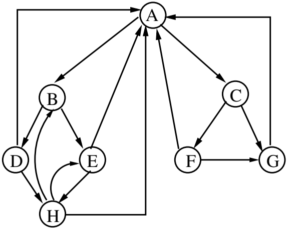
- G. Pinski, F. Narin: *Citation aggregates for journal aggregates of scientific publications: Theory, with application to the literature of physics*. Information Processing and Management, 12 (5), 297-312, 1976. Study the digraph of papers/citations, to rank the relevance of the papers.
- Jon Kleinberg (1996-97), because of his contributions got the MacArthur Genius Prize and the Nevanlinna Prize.
- Larry Page, Sergey Brin (1996-97), implemented the PageRank algorithm and founded Google.

Web search: Hyperlink analysis

- Understanding the Web structure and taking advantage of its link structure is basic to develop good search algorithms.
- hyperlink analysis: take advantage of in and out links to rank the Web's pages.
- Not all Web pages are equally important: There are hubs and authority pages and nodes without in or out links.
- Web pages are important if people visit them a lot, (but we want to avoid self-spamming).
- For rank the importance of a page, it is even more important important to consider the authority of the pages into the page.
- We can think of in-hyperlinks as a flux of votes.
- To define a ranking in importance of a page, we consider that votes from important pages have more votes.
- How do we get to know the important pages?

Main ideas behind PageRank

1− If a page has d out-links, each of its hyperlinks should count as $1/d$ votes.
In the figure, the importance of A would depend on
$1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{2} + 1 = \frac{10}{3}$



2− Key idea: If a page $u \to v$ then u contributes to v with $1/d(u)$, but this does not take into account the authority of u:
Let $\pi[v]$ define the authority of page v, and let P_{uv} be the the fraction out-links from u pointing to v. Then

$$\pi[v] = \sum_{u \to v} \pi[u] P_{uv},$$

which is a a recursive definition!.

PageRank as a Markov chain

- We have a finite MC, where the states= Web pages and transitions= hyperlinks.
- As we saw, if $\pi[v]$ is the authority of Web page v, and $P_{u,v}$ is the fraction of hyperlinks $u \to v$ then $\pi[v] = \sum_{u \to v} \pi[u] P_{u,v}$
- At each step, we transition according to a random hyperlink on the page.
- The importance, (PageRank) of a page is just its probability under the stationary distribution. That is, the long-term fraction of time a random surfer is at the page.

Problems with the MC approach: Dangling pages

The Web graph is not a Markov Chain:

- Dangling nodes: pages have no outgoing links (or links which haven't been crawled yet).
- PageRank considers that every dangling page is connected to every page in the Web and jumps out.
- The influence of that node is over the other pages is not significant $1/(5 \times 10^9)$

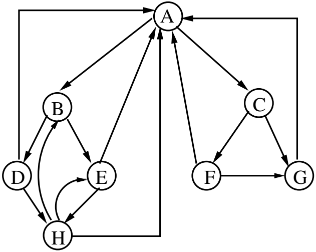
Problems with the MC approach: Rank sinks

- Rank sinks: group of pages which only have links to each other; once you go in, there is no way to escape.
- This is actually a major pitfall in practice, spammers put cliques of pages with a few in-links and no out-links.
- PageRank solution: Consider a scaling parameter α modify PageRank:
    - On each step, with probability α follow a random link on the current page
    - On each step, with probability $1 - \alpha$ move to a random page
- So at each step, PageRank jumps to a totally random page, with probability $1 - \alpha$ and follows the links with probability α.
- In early versions of PageRank, Google used a value $\alpha = 0.85$

The basic PageRank algorithm

Given a Web graph $\vec{W} = (V, \vec{E})$, and a number k of iterations.

```
**PageRank** $(\vec{W}, k), |V| = n$
Assign to $v \in V$, $\pi[v] = 1/n$
For $k$ iterations, for each $v \in V$ modify $\pi[v]$
    Each $v$ divides its current $\pi[v]$
        equally between its out-links
    Each $v$ updates its $\pi[v]$ to
        the sum on the shares it receives.
```



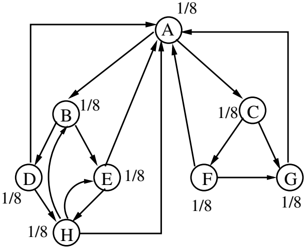
At iteration i, to update the authority $\pi^i[v]$ for each $v \in V$

$$\pi^i[v] = \sum_{(u \to v)} \frac{\pi^{i-1}[u]}{d(u)}.$$

The basic PageRank algorithm: π^0

Given a Web graph $\vec{W} = (V, \vec{E})$, and a number k of iterations.

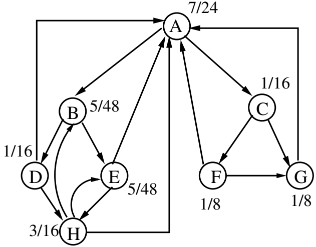
```
**PageRank** $(\vec{W}, k), |V| = n$
Assign to $v \in V$, $\pi[v] = 1/n$
For $k$ iterations, for each $v \in V$ modify $\pi[v]$
    Each $v$ divides its current $\pi[v]$
        equally between its out-links
    Each $v$ updates its $\pi[v]$ to
        the sum on the shares it receives.
```



The basic PageRank algorithm: π^1

Given a Web graph $\vec{W} = (V, \vec{E})$, and a number k of iterations.

```
**PageRank** $(\vec{W}, k), |V| = n$
Assign to $v \in V$, $\pi[v] = 1/n$
For $k$ iterations, for each $v \in V$ modify $\pi[v]$
    Each $v$ divides its current $\pi[v]$
        equally between its out-links
    Each $v$ updates its $\pi[v]$ to
        the sum on the shares it receives.
```

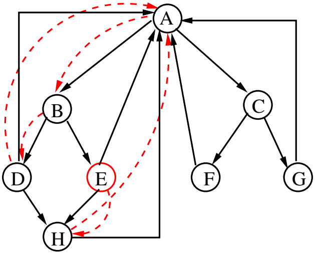


Notice: For each $0 \leqslant i \leqslant k \sum_v \pi^i[v] = 1 \Rightarrow$ if $\vec{W}$ is SCC ,
$\lim_{k \to \infty} \to \pi$, where π is the stationary PageRank's values.
which is the value of π for our toy example of $\vec{W}$?

PageRank as a Random Walk in the Web's graph

A discrete time stochastic process is a sequence of random variables $\{X_0, X_1, \ldots, X_t, \ldots\}$ where the subindex represent discrete points in time.

A random walk is a stochastic process, that describes a path that consists of a succession of random steps on some mathematical space such as the edges of a graph.



T:	0	1	2	3	4	5
P:	E	H	A	B	D	A
Pr:		1/2	1	1/3	1/2	1/2

Final considerations

- Recall that measuring the quality or authority of a web page in an automatic way by using textual analysis was not a good choice.
- Kleinberg and Brin-Page created the basic theory to develop the PageRank algorithm, taking into account the direct hyperlink structure of the Web.
- PageRank is just a Markov Chain random walk of the whole Web graph.
- Scaling replace the transition probability $P_{u,v}$ with $\alpha P_{u,v} + (1 - \alpha)/N$, where N is the total number of pages.
- Therefore, each entry of P becomes > 0, and the Fundamental Theorem MC implies that there is a unique stationary π, which gives the ranking of web pages. i.e. The random distribution on visited pages converges to π.

Final considerations

- How Google computed the PageRank?
- In practice Google has crawled the whole web and has a copy of the Web graph in their machines. Computing the stationary distribution using the balance equations is isn too slow ($|V| = n \sim 5.28 \times 10^9$ pages).
- So to compute the PageRank, Google starts with π^0 as uniform, and then compute $\pi_0 P, \pi_0 P^2, \pi_0 P^3, \ldots \pi_0 P^k$.
- To get convergence to the stationary π needs $t \sim 100$. Still everything can be done quite quickly (about 4 hours for each computation of PageRank)
- It seems Google updates the Web graph and re-computes PageRank once a month.
- The details of the actual Google searchers is a very well kept secret!

Part

1. Page Rank

2. Gambler’s ruin

3. More on random walks

A MC with absorbing states: Gambler's Ruin

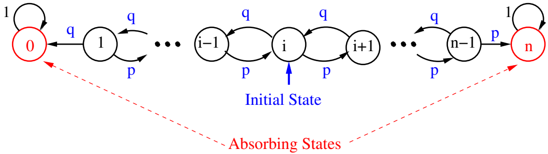
Model used to evaluate insurance risks.

- You place bets of 1.€ With probability p, you gain 1 ,€ and with probability $q = 1 - p$ you loose your 1 b€et.
- You start with an initial amount of m .€
- You keep playing until you loose all your money or you arrive to have n .€
- Define bias factor $\alpha = q/p$; If $\alpha = 1$ then $p = q = 1/2$, so it is fair game. If $\alpha > 1$ you are more likely to loose than win; if $\alpha < 1$, the game is bias against you.
- The goal is finding the probability of winning i.e. starting in state m reaching state n.

Notice in this chain, once we enter in state 0 or in state n, we can't leave the state. Those states are called absorbing states.

Gambler's Ruin

The chain can be given either by a $(n+1) \times (n+1)$ transition matrix P, where for $0 \leqslant i \leqslant n$: $P_{i,(i+1)} = p$ and $P_{i,(i-1)} = q$, $P_{0,0} = P_{n,n} = 1$.



$$
\begin{array}{c|cccccc c}
 & 0 & 1 & 2 & 3 & \cdots & n-1 & n \\
\hline
0 & 1 & q & 0 & 0 & \cdots & 0 & 0 \\
1 & 0 & 0 & q & 0 & \cdots & 0 & 0 \\
2 & 0 & p & 0 & q & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
n-2 & 0 & 0 & 0 & 0 & \cdots & q & 0 \\
n-1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
n & 0 & 0 & 0 & 0 & \cdots & p & 1
\end{array}
$$

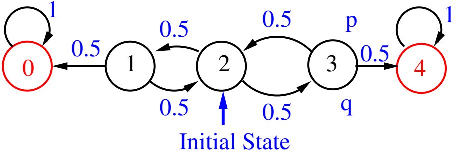
Gambler’s Ruin

In a MC an absorbing state i is one for which $p_{i,i} = 1$.

The chain has two absorbing states, when the system arrives to one of them it never exit, it is absorbed.
Some of the questions to be asked about such a chain are:

- What is the probability that the process will eventually reach an absorbing state? absorption probability.
- On the average, how long will it take for the process to be absorbed? expected absorption probability.

Gambler's Ruin Example with 4 states



$$\begin{array}{c|ccccc} & 0 & 1 & 2 & 3 & 4 \\ \hline 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0.5 & 0 & 0.5 & 0 & 0 \\ 2 & 0 & 0.5 & 0 & 0.5 & 0 \\ 3 & 0 & 0 & 0.5 & 0 & 0.5 \\ 4 & 0 & 0 & 0 & 0 & 1 \end{array}$$

$$\pi^0 = (0, 0, 1, 0, 0)$$
$$\pi^1 = (0, 1/2, 0, 1/2, 0)$$
$$\pi^2 = (1/4, 0, 2/4, 0, 1/4)$$
$$\vdots$$

Notice in this case the states 1,2, and 3 are transient and 0, 4 are absorbing states

Gambler's Ruin

Let $P_{i,n}$ denote the probability that the gambler with i euros arrives to n euros, before going broke.

Note that $1 - P_{i,n}$ is the corresponding probably that the gambler ruins.

Let us compute $P_{i,n}$:
Notice $P_{0,n} = 0$, $P_{n,n} = 1$ and $P_{i,n} = pP_{i+1,n} + qP_{i-1,n}$.

As $P_{i,n} = pP_{i,n} + qP_{i,n}$ then $P_{i+1,n} - P_{i,n} = \frac{q}{p}(P_{i,n} - P_{i-1,n})$.

In particular $P_{2,n} - P_{1,n} = \frac{q}{p}P_{1,n}$ (as $P_{0,n} = 0$)
and $P_{3,n} - P_{2,n} = \frac{q}{p}(P_{2,n} - P_{1,n}) = (\frac{q}{p})^2 P_{1,n}$
so $P_{i+1,n} - P_{i,n} = (\frac{q}{p})^i P_{1,n}$..

Gambler’s Ruin

On the other hand

$P_{i+1,n} = \sum_{k=0}^{i}(P_{k+1,n} - P_{k,n}) = \sum_{k=1}^{i}(P_{k+1,n} - P_{k,n}) + P_{1,n}$

$\Rightarrow P_{i+1,n} - P_{1,n} = \sum_{k=1}^{i}(P_{k+1,n} - P_{k,n}) = \sum_{k=1}^{i}(\frac{q}{p})^k P_{1,n}$

$\Rightarrow P_{i+1,n} = P_{1,n} + P_{1,n}\sum_{k=1}^{i}(\frac{q}{p})^k = \sum_{k=0}^{i}(\frac{q}{p})^k$

Using the geometric series equation $\sum_{j=0}^{i} x^j = P_{1,n}\frac{1-x^{i+1}}{1-x}$.

$$P_{i+1,n} = \begin{cases} P_{1,n}\frac{1-(q/p)^{i+1}}{1-(q/p)}, & \text{if } p \neq q; \\ P_{1,n} \cdot (i+1) & \text{if } p = q = 1/2. \end{cases}$$

Gambler's Ruin

Choosing $i = n - 1$ and as $P_{n,n} = 1$, then

$$P_{1,n} = \begin{cases} \frac{1-(q/p)}{1-(q/p)^n}, & \text{if } p \neq q; \\ 1/n & \text{if } p = q = 1/2. \end{cases}$$

Therefore , $P_{i,n} = \begin{cases} \frac{1-(q/p)^i}{1-(q/p)^n}, & \text{if } p \neq q; \\ i/n & \text{if } p = q = 1/2. \end{cases}$

□

Becoming rich or getting ruined

Using the deduced eq. for $P_{i,n}$:

- If $p > 1/2$ then $\frac{q}{p} < 1$ and $\lim_{n\to\infty} P_{i,n} = \lim_{n\to\infty}(1 - (q/p)^n) = 1$. In this case, the gambler will become rich with probability 1.
- If $p < 1/2$ then $\frac{q}{p} > 1$ and $\lim_{n\to\infty} P_{i,n} = 0$. So with probability 1 the gambler will get ruined
- If $p = q = 1/2$ then $P_{i,n} = i/n$ and $\lim_{n\to\infty} P_{i,n} = 0$ (if $i = o(n)$); again the gambler gets ruined with probability 1.

For ex. if Bob starts with 2 euros and $p = 0.6$, what is the probability that he gets $n = 5$ euros?
$P_{2,5} = \frac{1-(2/3)^2}{1-(2/3)^5} = 0.64$.

What is the probability that he will become infinitely rich?
$(n \to \infty)$ $P_{2,\infty} = 1 - (2/3)^2 = 0.56$.

Markov chains with absorbing states

- The Gambler ruin's Markov chain is an example of a Markov chain with one or more absorbing states, where the process stops.
- An absorbing state u has $p_{u,u} = 1$. In many application the absorbing MC has two states: 0 and n.
- Those MC are not irreducible (states 0 and n do not exit)
- Those MC play an important role in many "practical" stochastic processes: Biological, economical, and others.
- The limit probability distribution π of an absorbing MC has the absorbing probabilities for the absorbing state, and 0 for the other states. In the ex. of the Gambler's ruin if $p \neq q$, $\pi = \left(1 - (\frac{1-(q/p)^i}{1-(q/p)^n}), 0, \ldots, 0, \frac{1-(q/p)^i}{1-(q/p)^n}\right)$.
- The two important quantities in those absorbing MC are:
  1. The absorption probability.
  2. The absorption time.

Part

1. Page Rank

2. Gambler’s ruin

3. More on random walks

Random walks

- An algorithmic paradigm.
- Given a finite, connected graph $G = (V, E)$ with $|E| = m$, $|V| = n$, a random walk on G is a MC defined by the sequence of moves of a particle between the vertices of G.
- A random walk on G, probability starts from a given $v \in V$, and if v has $d(v)$ outgoing neighbors, then the probability that the walk moves to u is $1/d(v)$, where $\mathcal{N}(u) =$ set of neighbors of u and $|\mathcal{N}(u)| = d(u)$.

The generic algorithm:

```
Given $G = (V, E)$, $v \in V$
**for** T times **do**
    Choose u.a.r. (with probability $= 1/d(v)$) a $u \in \mathcal{N}(v)$
    $v := u$
**end for**
```

Random walks: Definitions

Given a connected graph $G = (V, E)$ define:

1. The hitting time $h_{v,u}$ from v to u, that is the expected number of steps for the random walk to go from v to u (for first time).
2. The cover time $C_{v,u}$ from v as the expected number of steps that a walk will take in starting from v visiting all vertices in G.
3. The cover time of G, C_G as $\max_{v \in V} C_v$.

Random walks and Markov Chains

Theorem

A random walk on an undirected G is aperiodic iff G is not bipartite.

Proof

G is bipartite iff it does not have cycles with odd number of edges.
In an undirected G there is always a path of length 2 from $v \to v$.
If G is bipartite the RW is periodic with period 2.
If G is not bipartite it has an odd cycle, and traversing that cycle we have an odd-length path $v \quad \to \quad v \Rightarrow$ the Markov chain is aperiodic. □

Random walks and Markov Chains

From now on, we assume the given undirected G **is not bipartite and it is connected**.

Then the MC defined by RW on G is irreducible and aperiodic, by the Fundamental Theorem the random walk converges to a stationary distribution π.

Random walks and Markov Chains

The next theorem shows the stationary distribution π only depends of sequence degree in G.

Theorem

A random walk on G converges to a stationary distribution $\pi = (\pi[u])_{u \in V}$, where $\pi[u] = d(u)/2m$.

Proof

First we prove π is a true distribution: For $G = (V, E)$:

$\sum_{u \in V} d(u) = 2|E| \Rightarrow \sum_{u \in V} \pi[u] = \sum_{u \in V} d(u)/2|E| = 1$.

Let P be the trasition matrix of the MC, then $\forall u \in V$, as $\pi = \pi P \Rightarrow \pi[u] = \sum_{v \in \mathcal{N}(u)} \frac{d(v)}{2|E|} \frac{1}{d(v)} = \frac{d(u)}{2|E|}$. □

Random walks and Markov Chains

As we already know that if P is regular, $h_{u,u} = 1/\pi[u]$, then

Corollary

For $u \in V$, $h_{u,u} = 2|E|/d(u)$.

Random walks and Markov Chains

Lemma

Given $G = (V, E)$*, if* $(u, v) \in E$ *then* $h_{u,v} < 2|E|$*.*

Proof

Let $u \in V$, from the previous corollary $h_{u,u} = \frac{2|E|}{d(u)}$,
On the other hand we also know $h_{u,u} = \frac{1}{d(u)} \sum_{w \in \mathcal{N}(u)} (1 + h_{w,u})$, since $P_{u,w} = 1/d(u)$ for all $w \in \mathcal{N}(u)$
Therefore, $\frac{2|E|}{d(u)} = \frac{1}{d(u)} \sum_{w \in \mathcal{N}(u)} (1 + h_{w,u})$

$\Rightarrow 2|E| = \sum_{w \in \mathcal{N}(u)} (1 + h_{w,u})$

So $h_{u,v} < 2|E|$. □

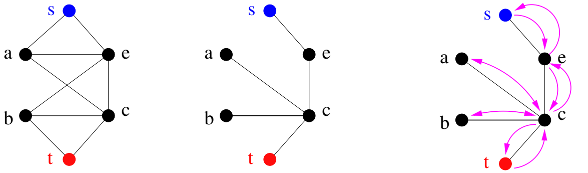
Random walks and Markov Chains

Corollary

Given $G = (V, E)$*, its cover time* $C_G \leqslant 4|E||V|$.

Proof

Given G with $|V| = n, |E| = m$, find a spanning tree T_G with $n - 1$ edges of G, then traverse T_G using a cyclic Eulerian tour, and the number of steps to traverse it is an upper bound to C_T.
That can be done in $\mathcal{O}(m + n)$ using BFS.



G

Spanning Tree

Eulerian tour: n−1

Random walks and Markov Chains

Proof (cont'd)

Let $v_0, v_1, \ldots v_{2n-2} \quad = \quad v_0$ the resulting sequence in the tour.

The expected time of going through the tour is $\leqslant C_G$, so

$$\sum_{i=0}^{2n-3} h_{v_i, v_{i+1}} < (2n-2)2m < 4nm$$

□

Algorithm to check $s - t$ connectivity in undirected G

Given a $G = (V, E)$, with $|V| = n, |E| = m$, and $s, t \in V$ we want to find a path from $s \to t$.

Deterministically we can do it in $\mathcal{O}(n + m)$ (DFS or BFS), however they need $\Omega(n)$ space.

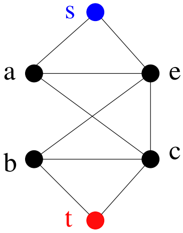
We produce a randomized algorithm, based in RW, that uses $\mathcal{O}(n^3)$ steps and $\mathcal{O}(\lg n)$ bits of space. At each step only needs to remember the last position

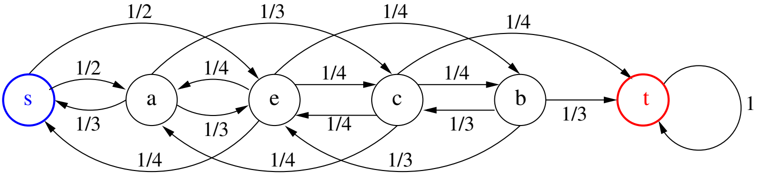
Moreover, no need of large or complicated data structure.

The clock on the number of steps is due to the fact that a Markovian RW does not know where it has visited the whole graph (because the Markovian property).

Algorithm to check $s - t$ connectivity in undirected G

s-t **Connectivity** $G = (V, E), s, t$
Start a RW from s
If the RW reaches t in $\leqslant 4n^3$ steps,
there is a path $s - t$





$s - t$ connectivity algorithm

Theorem

The $s - t$ connectivity algorithm returns the correct answer with probability $\geqslant 1/2$ and in $\mathcal{O}(n^3)$ steps using $\mathcal{O}(\lg n)$ bits of memory.

Using Markov's inequality:
$\mathbb{P}[$RW has not visited all vertices after $2C_G$ steps$] \leqslant \frac{1}{2}$,
by the previous corollary $C_G \leqslant 4|E||V| = 4nm \leqslant 4n^3$.

Notice if we set the clock to $200nm$ the above theorem tells us that the failure probability is reduced to $1/200$.