DD2476 Search Engines and Information Retrieval Systems

Lecture 6: Retrieval of Documents with Hyperlinks

Hedvig Kjellström
hedvig@kth.se
https://www.kth.se/social/course/DD2476/

Recap: Ranked Retrieval

We want top-ranking documents to be both relevant and authoritative

- Relevance – cosine scores
- Authority – query-independent property

Examples of authority signals

- Wikipedia pages (qualitative)
- Articles in certain newspapers (qualitative)
- A scientific paper with many citations (quantitative)
- **PageRank** (quantitative)

Today

PageRank (Manning Chapter 21)

- Measuring the authority of a document in corpus with hyperlinks

Monte Carlo methods (Bishop Chapter 11)

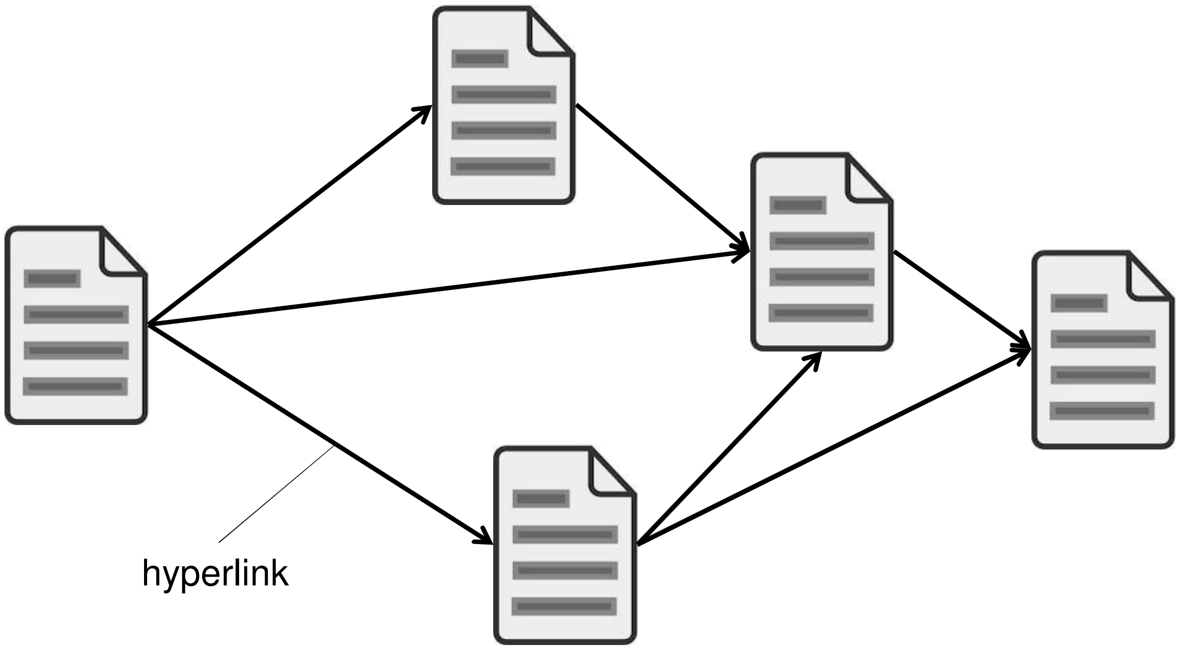
- Short recap of / intro to Monte Carlo methods

Monte Carlo approximations to PageRank (Avrachenkov et al Sections 1-2)

- Approximative and fast way to find PageRank

PageRank (Manning Chapter 21)

The web as a directed graph



Using link structure for ranking

Assumption: A link from X to Y signals that X's author perceives Y to be an authoritative page

– X "casts a vote" on Y

Simple suggestion: Rank = number of in-links

Discuss with your neighbor:
What is the problem with this approach?

PageRank: Basic idea

WWW’s particular structure can be exploited:

- pages have links to one another
- the more in-links, the higher rank
- in-links from pages having high rank are worth more than in-links from pages having low rank

This idea is the cornerstone of PageRank (Brin & Page 1998)

Way of formalizing:

A ”random surfer” that randomly follows links will spend more time on pages with high PageRank

First attempt

$$PR(D) = \sum_{D' \in in(D)} \frac{PR(D')}{L_{D'}}$$

D and *D′* are web pages

(documents in corpus with hyperlinks)

in(*D*) is the set of pages linking to *D*

L_D is the number of out-links from *D*

Discuss with your neighbor:
Something missing?
What happens when the page has no outlinks?
What happens when the page has no inlinks?

Random Surfer

Imagine a random surfer that **follows links**

- The link to follow is selected with uniform probability
- If the surfer reaches a sink (a page without links), they randomly restarts on a new page
- Every once in a while, the surfer jumps to a random page (even if there are links to follow)

Second attempt

With probability *1-c* the surfer is bored, stops following links, and restarts on a random page

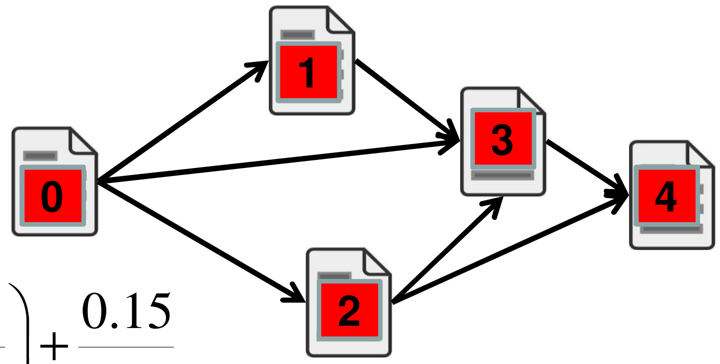
- Guess: Google used c=0.85

$$PR(D) = c\left(\sum_{D' \in in(D)} \frac{PR(D')}{L_{D'}}\right) + \frac{1-c}{N}$$

Without this assumption, the surfer will get stuck in a subset of the web.

Example

$$PR_4 = 0.85 \cdot \left(\frac{PR_2}{2} + PR_3 \right) + \frac{0.15}{5}$$



$$PR_3 = 0.85 \cdot \left(\frac{PR_0}{3} + PR_1 + \frac{PR_2}{2} + \frac{PR_4}{4} \right) + \frac{0.15}{5}$$

$$PR_2 = PR_1 = 0.85 \cdot \left(\frac{PR_0}{3} + \frac{PR_4}{4} \right) + \frac{0.15}{5}$$

$$PR_0 = 0.85 \cdot \left(\frac{PR_4}{4} \right) + \frac{0.15}{5}$$

Probability of moving from 4 to here since 4 is a sink

Interpretation

Authority / popularity / relative information value

PR_D = the probability that the random surfer will be at page D at any given point in time

This is called the stationary probability
(the left eigenvector of the transition matrix)

How do we compute it?

The random surfer as a Markov chain

The random surfer model suggests a Markov chain formulation:

N states (= documents)

$N \times N$ transition probability matrix G

At each step, the surfer is in exactly one of the states

Matrix entry G_{ij} = probability of j being the next state (doc), given we are currently in state (doc) i

Ergodic Markov chains

A Markov chain is ergodic if

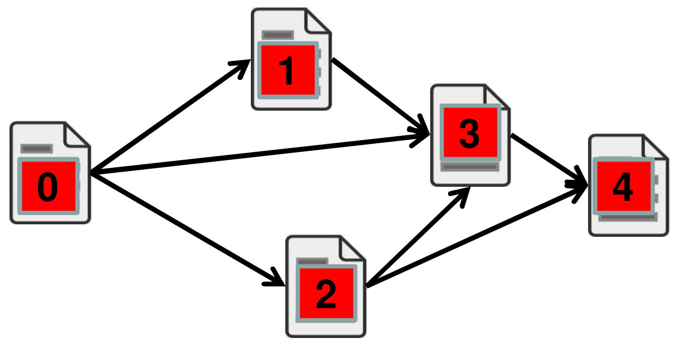
- you have a path from any state to any other
- For any start state, after a finite transient time T_0, the probability of being in any state at a fixed time $T > T_0$ is nonzero

Our transition matrix G is non-zero everywhere ↔ the graph is strongly connected ↔
the Markov chain is ergodic ↔
unique stationary probabilities π exist

Example: Transition matrices

P

0	0.33	0.33	0.33	0
0	0	0	1	0
0	0	0	0.5	0.5
0	0	0	0	1
0.25	0.25	0.25	0.25	0



J

0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2
0.2	0.2	0.2	0.2	0.2

G = cP+(1-c)J

0.0300	0.3105	0.3105	0.3105	0.0300
0.0300	0.0300	0.0300	0.8800	0.0300
0.0300	0.0300	0.0300	0.4550	0.4550
0.0300	0.0300	0.0300	0.0300	0.8800
0.2425	0.2425	0.2425	0.2425	0.0300

Pagerank = probability vector

A probability (row) vector $x=(x_1,...,x_N)$ tells us where the walk is at any point

One step of the random surfer:

$$x' = xG$$

Pagerank (let's call it π) is the stationary probability vector for G:

$$\pi \mathbf{G} = \pi$$

π is a left eigenvector of G

So, let's do SVD on G! Or, what could be the problem?

Power iteration

Method of finding dominant eigenvector

Eigenvector with largest eigenvalue

Recall, regardless of where we start, we eventually reach the stationary vector π

Start with any distribution (say x=(1,0,...,0)).
– After one step, we're at xG;
– after two steps at (xG)G, then ((xG)G)G and so on

"Eventually", for "large" k, $xG^k=\pi$

k is the number of steps taken by the random surfer

Power iteration algorithm

```
Let x=(0,…,0) and x′ an initial
  state, say (1,0,…,0)

while ( |x−x′| > ε ):
     x = x′
     x′= xG
```

Converges very slowly!

Monte Carlo Methods (Bishop Chapter 11)

Approximate Solutions

Huge #docs -> exact inference very expensive

- Matrix factorization takes us part of the way
- But eventually…

Better solution: find approximation

One way: Monte Carlo sampling

The Monte Carlo principle

State space *z*

Imagine that we can sample $z^{(l)}$ from the pdf $p(z)$ but that we do not know its functional form

Might want to estimate for example:

$$E[z] = \sum z\, p(z)$$

$p(z)$ can be approximated by a histogram over $z^{(l)}$:

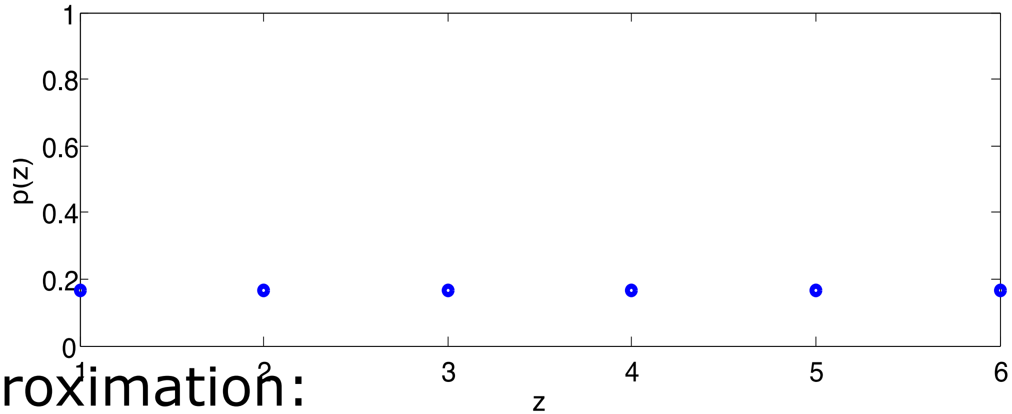
$$\hat{q}(z) = \frac{1}{L} \sum_{l=1}^{L} \delta_{z^{(l)}=z}$$

Example: Dice Roll

The probability of outcomes of dice rolls: $p(z) = \frac{1}{6}$

Exact solution:

What would happen if the dice was bad?

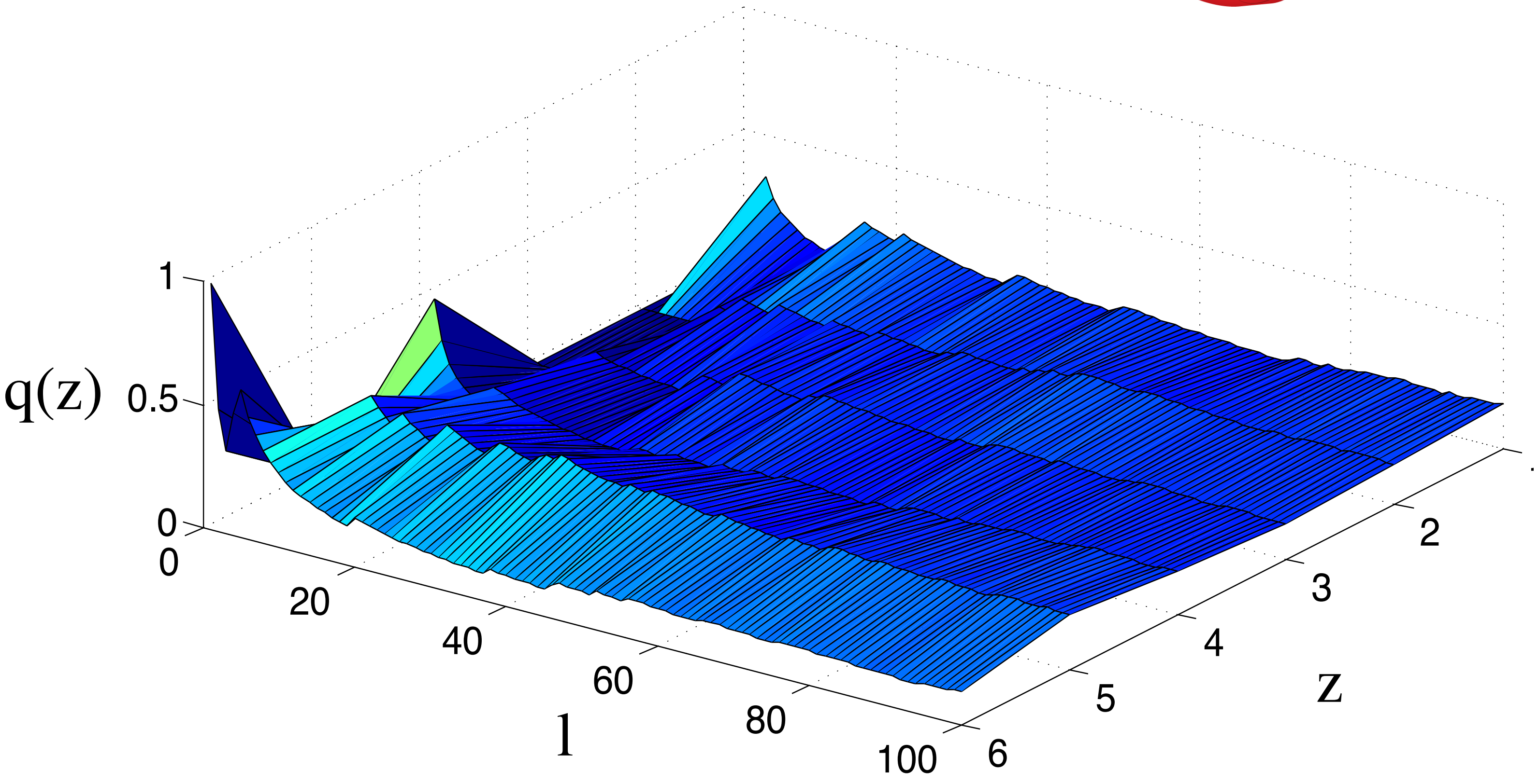


Monte Carlo approximation:

- Roll a dice a number of times, might get

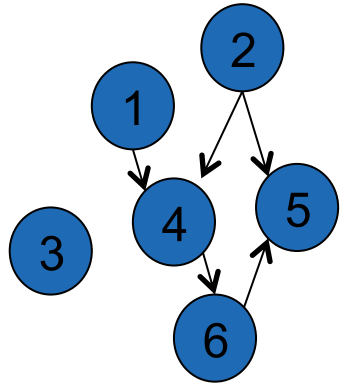
$$z^{(1)} = 6 \quad z^{(2)} = 4 \quad z^{(3)} = 1 \quad z^{(4)} = 6 \quad z^{(5)} = 6$$

Example: Dice Roll



The Law of Large Numbers

What is *p* and *q* for PageRank?



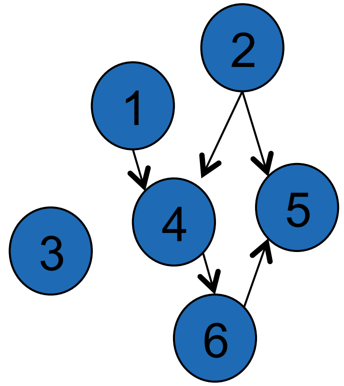
Discuss with your neighbor (5 mins)

- Graph of connected documents
- Look at each document *z*, compute PageRank

Quest: Find $p(z)$ = prob that the document *z* is visited = PageRank score of document *z*

Monte Carlo approach: find approximate PageRank $\hat{q}(z)$ by sampling from $p(z)$

How do we sample from *p* without knowing *p*?



Discuss with your neighbor (5 mins)

Simulate a ”random surfer” walking in the graph

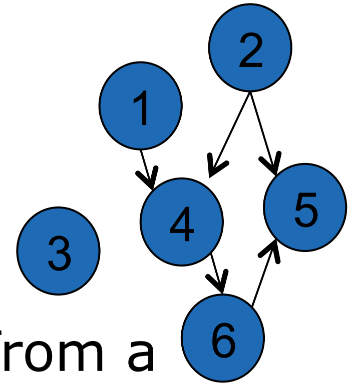
- Equal probability c/<#links> of selecting any of the <#links> links in a document D
- Probability (1 - c) of not following links, but jumping to an unlinked document in the graph

Record location $z^{(l)}$ at each step *l*

$$\hat{q}(z) = \frac{1}{L} \sum_{l=1}^{L} \delta_{z^{(l)}=z}$$

Monte Carlo Approximations to PageRank (Avrachenkov et al Sections 1-2)

Monte Carlo Idea



z above same as D here

D = document id

Consider a random walk $\{D_t\}_{t \geq 0}$ that starts from a randomly chosen page.

At each step t:

- Prob c: D_t = one of the documents with edges from D_{t-1}
- Prob (1 – c): The random walk terminates, and D_t = random node

Endpoint D_T is distributed as PageRank π

Sample from π = do many random walks

Advantages

Exact method: precision improves linearly for all docs

Monte Carlo method: precision improves faster for high-rank docs

Exact method: computationally expensive

Monte Carlo method: parallel implementation possible

Exact method: must be redone when new pages are added

Monte Carlo method: continuous update

1. MC end-point with random start

Simulate N runs of the random walk $\{D_t\}_{t\geq 0}$ initiated at a **randomly chosen page**

PageRank of page j = 1,...,n:

π_j = (#walks which end at j)/N

Worst case: N = O(n^2)

Mean case: N = O(n)

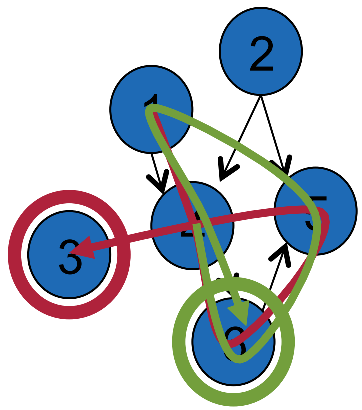
Example:

1 link 4 link 6 link 5 jump 3

4 link 6 link 5 jump 1 link 4 link 6

π = [0, 0, 0.5, 0, 0, 0.5]

2 walks not enough



2. MC end-point with cyclic start

Simulate N = mn runs of the random walk $\{D_t\}_{t \geq 0}$ initiated at **each page exactly m times**

PageRank of page j = 1,...,n:

$$\boldsymbol{\pi_j = (\#\text{walks which end at } j)/N}$$

3. MC complete path

Simulate N = mn runs of the random walk $\{D_t\}_{t\geq 0}$ of length T, initiated at **each page exactly m times**

PageRank of page j = 1,…,n:

$$\boldsymbol{\pi_j = (\#\text{visits to node j during walks})/(NT_j)}$$

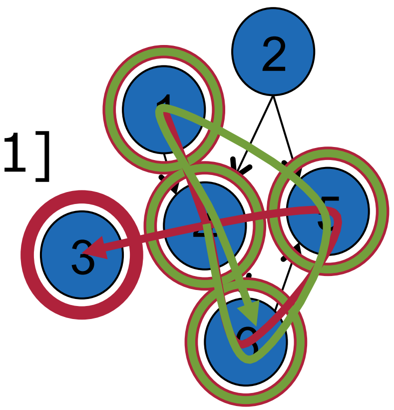
Example:

1 link 4 link 6 link 5 jump 3

4 link 6 link 5 jump 1 link 4 link 6

π = [2/11, 0, 1/11, 3/11, 2/11, 3/11]

2 walks almost enough



4. MC complete path stopping at dangling nodes

Simulate N = mn runs of the random walk $\{D_t\}_{t\geq 0}$ initiated at **each page exactly m times** and **stopping when it reaches a dangling node**

PageRank of page j = 1,...,n:

$$\boldsymbol{\pi_j = (\#\text{visits to node } j \text{ during walks}) / (\text{total } \#\text{visits during walks})}$$

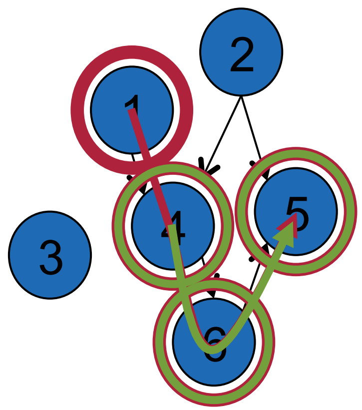
Example:

1 link 4 link 6 link 5

4 link 6 link 5

π = [1/7, 0, 0, 2/7, 2/7, 2/7]

2 walks not enough



5. MC complete path with random start

Simulate N runs of the random walk $\{D_t\}_{t\geq 0}$ initiated at a **randomly chosen page** and **stopping when it reaches a dangling node**

PageRank of page j = 1,…,n:

$$\boldsymbol{\pi_j = (\#\text{visits to node } j \text{ during walks}) / (\text{total } \#\text{visits during walks})}$$

Next

Assignment 1 left?

- You can present it at the session for Assignment 2
- Reserve two slots, one for each assignment!

Lecture 7 (February 24, 10.15-12.00)

- B3
- Readings: Manning Chapter 11, 12

Computer hall session (March 8, 13.00-…)

- Orange (Osquars Backe 2, level 4)
- Examination of computer Assignment 2

Doodle to come!