

Lecture 8: Linkage algorithms and web search

Information Retrieval
Computer Science Tripos Part II

Helen Yannakoudakis¹

Natural Language and Information Processing (NLIP) Group



**UNIVERSITY OF
CAMBRIDGE**

`helen.yannakoudakis@cl.cam.ac.uk`

2018

¹Based on slides from Simone Teufel and Ronan Cummins

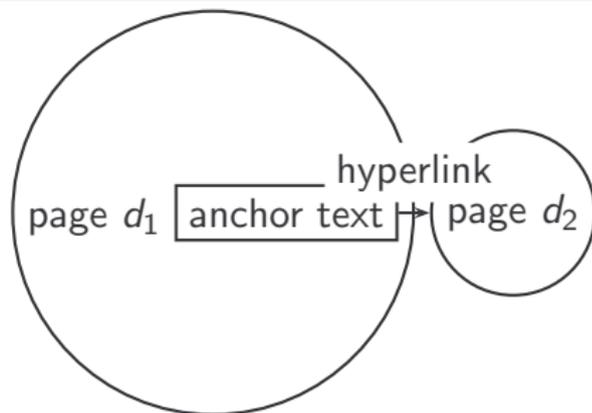
- Anchor text: What exactly are links on the web and why are they important for IR?
- PageRank: the original algorithm that was used for link-based ranking on the web
- How to compute PageRank

1 Anchor text

2 PageRank

3 Wrap up

The web as a directed graph



- Assumption 1: **A hyperlink is a quality signal.**
 - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1 's author deems d_2 high-quality and relevant.
- Assumption 2: **The anchor text describes the content of d_2 .**
 - We use anchor text somewhat loosely here for: the text surrounding the hyperlink.
 - Example: “**You can find cheap cars** **here**.”
 - Anchor text: “You can find cheap cars here”

[text of d_2] only vs. [text of d_2] + [anchor text $\rightarrow d_2$]

- Searching on [text of d_2] + [anchor text $\rightarrow d_2$] is often more effective than searching on [text of d_2] only.
- Example: Query *IBM*
 - Matches IBM's copyright page
 - Matches many spam pages
 - Matches IBM wikipedia article
 - May not match IBM home page!
 - ... if IBM home page is mostly graphics
- Searching on [anchor text $\rightarrow d_2$] is better for the query *IBM*.
 - In this representation, the page with the most occurrences of *IBM* is www.ibm.com.

www.nytimes.com: “IBM acquires Webify”

www.slashdot.org: “New IBM optical chip”

www.stanford.edu: “IBM faculty award recipients”

www.ibm.com

- Thus: Anchor text is often a better description of a page's content than the page itself.
- Anchor text can be weighted more highly than document text. (based on Assumptions 1&2)

- A Google bomb is a search with “bad” results due to maliciously manipulated anchor text.
- Google introduced a new weighting function in 2007 that fixed many Google bombs.
- Still some remnants: [dangerous cult] on Google, Bing, Yahoo
 - Coordinated link creation by those who dislike the Church of Scientology
- Defused Google bombs: [who is a failure?], [evil empire]

A historic google bomb



Web Images Groups News Froogle Local more »

miserable failure

Search

[Advanced Search](#)
[Preferences](#)

Web

Results 1 - 10 of about 969,000 for [miserable failure](#). (0.06 seconds)

[Biography of President George W. Bush](#)

Biography of the president from the official White House web site.

www.whitehouse.gov/president/gwbbio.html - 29k - [Cached](#) - [Similar pages](#)

[Past Presidents](#) - [Kids Only](#) - [Current News](#) - [President](#)

[More results from www.whitehouse.gov »](#)

[Welcome to MichaelMoore.com!](#)

Official site of the gadfly of corporations, creator of the film Roger and Me and the television show The Awful Truth. Includes mailing list, message board, ...

www.michaelmoore.com/ - 35k - [Sep 1, 2005](#) - [Cached](#) - [Similar pages](#)

[BBC NEWS | Americas | 'Miserable failure' links to Bush](#)

Web users manipulate a popular search engine so an unflattering description leads to the president's page.

news.bbc.co.uk/2/hi/americas/3298443.stm - 31k - [Cached](#) - [Similar pages](#)

[Google's \(and Inktomi's\) Miserable Failure](#)

A search for **miserable failure** on Google brings up the official George W. Bush biography from the US White House web site. Dismissed by Google as not a ...

searchenginewatch.com/sereport/article.php/3296101 - 45k - [Sep 1, 2005](#) - [Cached](#) - [Similar pages](#)

Origins of PageRank: Citation Analysis

- We can use the same formal representation (as DAG) for
 - citations in the scientific literature
 - hyperlinks on the web
- Appropriately weighted citation frequency is an excellent measure of **quality** ...
 - ... both for web pages and for scientific publications.
- Next: PageRank algorithm for computing weighted citation frequency on the web

1 Anchor text

2 PageRank

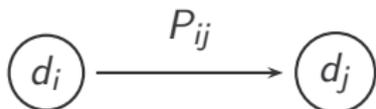
3 Wrap up

Model behind PageRank: Random walk

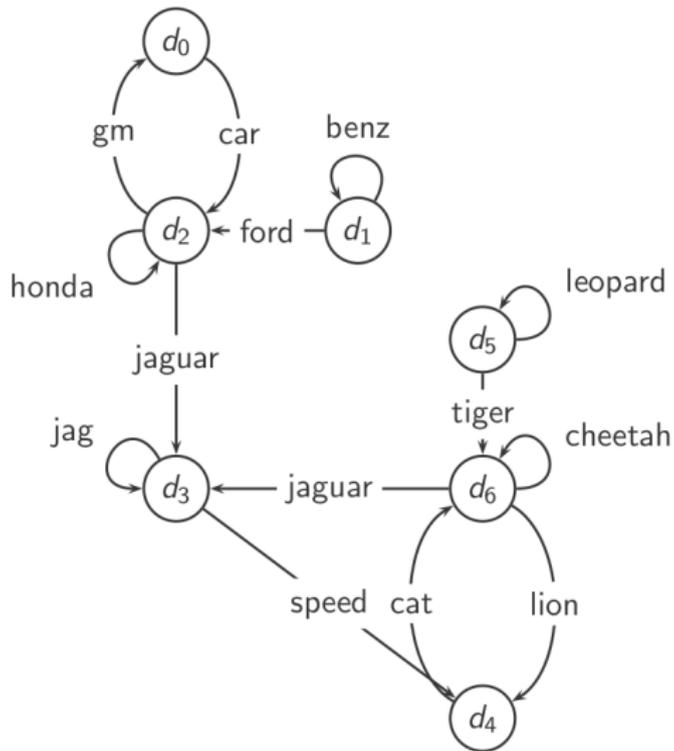
- Imagine a web surfer doing a random walk on the web
 - Start at a random page
 - At each step, go out of the current page along one of the links on that page, equiprobably
- In the steady state, each page has a **long-term visit rate**.
- This long-term visit rate is the page's **PageRank**.
- **PageRank = long-term visit rate = steady state probability**

Formalisation of random walk: Markov chains

- A Markov chain consists of N states, plus an $N \times N$ **transition probability matrix** P .
- **state = page**
- At each step, we are on exactly one of the pages.
- For $1 \leq i, j \leq N$, the matrix entry P_{ij} tells us the probability of j being the next page, given we are currently on page i .
- Clearly, for all i , $\sum_{j=1}^N P_{ij} = 1$



Example web graph



Link matrix for example

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0	0	1	0	0	0	0
d_1	0	1	1	0	0	0	0
d_2	1	0	1	1	0	0	0
d_3	0	0	0	1	1	0	0
d_4	0	0	0	0	0	0	1
d_5	0	0	0	0	0	1	1
d_6	0	0	0	1	1	0	1

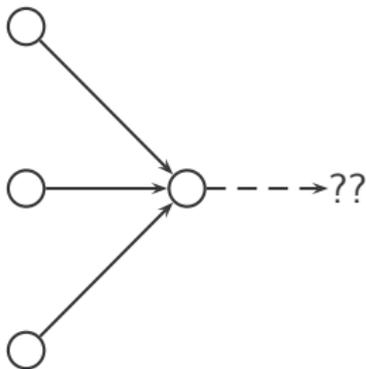
Link matrix for example

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0	0	1	0	0	0	0
d_1	0	1	1	0	0	0	0
d_2	1	0	1	1	0	0	0
d_3	0	0	0	1	1	0	0
d_4	0	0	0	0	0	0	1
d_5	0	0	0	0	0	1	1
d_6	0	0	0	1	1	0	1

Transition probability matrix P for example

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
d_1	0.00	0.50	0.50	0.00	0.00	0.00	0.00
d_2	0.33	0.00	0.33	0.33	0.00	0.00	0.00
d_3	0.00	0.00	0.00	0.50	0.50	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.50	0.50
d_6	0.00	0.00	0.00	0.33	0.33	0.00	0.33

- Recall: PageRank = long-term visit rate
- Long-term visit rate of page d is the probability that a web surfer is at page d at a given point in time.
- Next: what properties must hold of the web graph for the long-term visit rate to be well defined?
- The web graph must correspond to an **ergodic** Markov chain.
- First a special case: The web graph must not contain **dead ends**.



- The web is full of dead ends.
- Random walk can get stuck in dead ends.
- If there are dead ends, long-term visit rates are not well-defined (or non-sensical).

Teleporting – to get us out of dead ends

- At a **dead end**, jump to a random web page with prob. $1/N$.
- At a **non-dead end**, with probability 10%, jump to a random web page (to each with a probability of $0.1/N$).
- With remaining probability (90%), follow a random hyperlink on the page.
 - For example, if the page has 4 outgoing links: randomly choose one with probability $(1-0.10)/4=0.225$
- 10% is a parameter, the **teleportation rate**.
- Note: “jumping” from dead end is independent of teleportation rate.

$$P' = (1 - \alpha) \cdot P + \alpha \cdot T \quad (1)$$

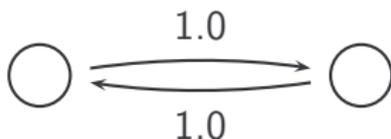
where T is the teleportation matrix and P is a stochastic matrix

- what is T ?
- An $N \times N$ matrix full of $1/N$
- α is the probability of teleporting

- With teleporting, we cannot get stuck in a dead end.
- But even without dead ends, a graph may not have well-defined long-term visit rates.
- More generally, we require that the Markov chain be **ergodic**.

Ergodic Markov chains

- A Markov chain is ergodic iff it is irreducible and aperiodic.
- **Irreducibility.** Roughly: there is a path from any page to any other page.
- **Aperiodicity.** Roughly: The pages cannot be partitioned such that the random walker visits the partitions sequentially.
- A non-ergodic Markov chain:



Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
- This is the **steady-state probability distribution**.
- Over a long time period, we visit each state in proportion to this rate.
- It doesn't matter where we start.
- **Teleporting makes the web graph ergodic.**
- \Rightarrow **Web-graph+teleporting has a steady-state probability distribution.**
- \Rightarrow **Each page in the web-graph+teleporting has a PageRank.**

- We now know what to do to make sure we have a well-defined PageRank for each page.
- Next: how to compute PageRank

Formalization of “visit”: Probability vector

- A probability (row) vector $\vec{x} = (x_1, \dots, x_N)$ tells us where the random walk is at any point.

- Example:
$$\begin{pmatrix} 0 & 0 & 0 & \dots & 1 & \dots & 0 & 0 & 0 \\ 1 & 2 & 3 & \dots & i & \dots & N-2 & N-1 & N \end{pmatrix}$$

- More generally: the random walk is on page i with probability x_i .

- Example:
$$\begin{pmatrix} 0.05 & 0.01 & 0.0 & \dots & 0.2 & \dots & 0.01 & 0.05 & 0.03 \\ 1 & 2 & 3 & \dots & i & \dots & N-2 & N-1 & N \end{pmatrix}$$

- $\sum x_i = 1$

Change in probability vector

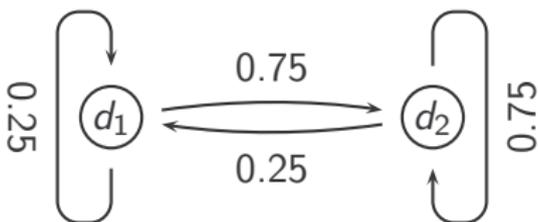
- If the probability vector is $\vec{x} = (x_1, \dots, x_N)$ at this step, what is it at the next step?
- Recall that row i of the transition probability matrix P tells us where we go next from state i .
- So from \vec{x} , our next state is distributed as $\vec{x}P$.

Steady state in vector notation

- The steady state in vector notation is simply a vector $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ of probabilities.
- (We use $\vec{\pi}$ to distinguish it from the notation for the probability vector \vec{x} .)
- π_i is the long-term visit rate (or PageRank) of page i .
- So we can think of PageRank as a very long vector – one entry per page.

Steady-state distribution: Example

What is the PageRank / steady state in this example?



Steady-state distribution: Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.25$	$P_{12} = 0.75$
			$P_{21} = 0.25$	$P_{22} = 0.75$
t_0	0.25	0.75		
t_1	0.25	0.75	(convergence)	

$$P_t(d_1) = P_{t-1}(d_1) \cdot P_{11} + P_{t-1}(d_2) \cdot P_{21}$$
$$0.25 \cdot 0.25 + 0.75 \cdot 0.25 = 0.25$$

$$P_t(d_2) = P_{t-1}(d_1) \cdot P_{12} + P_{t-1}(d_2) \cdot P_{22}$$
$$0.25 \cdot 0.75 + 0.75 \cdot 0.75 = 0.75$$

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \dots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- ... and if the distribution in this step is \vec{x} , then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi}P$
- Solving this matrix equation gives us $\vec{\pi}$.
- $\vec{\pi}$ is the principal left eigenvector for P ...
- ... that is, $\vec{\pi}$ is the left eigenvector with the largest eigenvalue.
- All transition probability matrices have largest eigenvalue 1.

One way of computing the PageRank $\vec{\pi}$

- Start with any distribution \vec{x} , e.g., uniform distribution
- After one step, we're at $\vec{x}P$.
- After two steps, we're at $\vec{x}P^2$.
- After k steps, we're at $\vec{x}P^k$.
- Algorithm: multiply \vec{x} by increasing powers of P until convergence.
- This is called the [power method](#).
- Recall: regardless of where we start, we eventually reach the steady state $\vec{\pi}$.
- Thus: we will eventually (in asymptotia) reach the steady state.

Computing PageRank: Power method

	x_1	x_2			
	$P_t(d_1)$	$P_t(d_2)$			
			$P_{11} = 0.1$	$P_{12} = 0.9$	
			$P_{21} = 0.3$	$P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748	0.2496	0.7504	$= \vec{x}P^4$
			
t_∞	0.25	0.75	0.25	0.75	$= \vec{x}P^\infty$

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

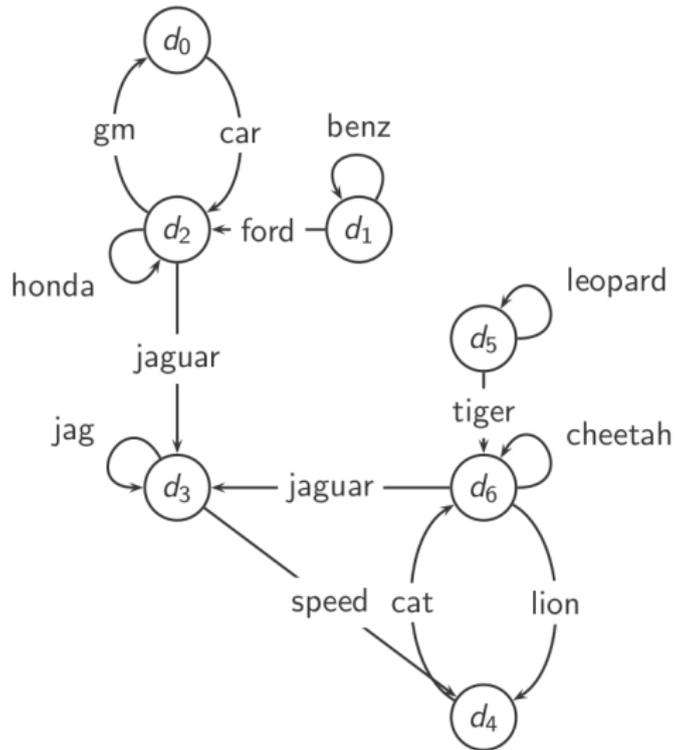
$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$

$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

- Preprocessing
 - Given graph of links, build initial matrix P
 - Ensure all rows sum to 1.0 to update P (for nodes with no outgoing links use $1/N$ for each element)
 - Apply teleportation with parameter α
 - From modified matrix, compute $\vec{\pi}$
 - π_i is the PageRank of page i .
- Query processing
 - Retrieve pages satisfying the query
 - Rank them by their PageRank (or at least a combination of PageRank and the relevance score)
 - Return reranked list to the user

- Real surfers are not random surfers.
 - Examples of non-random surfing: back button, short vs. long paths, bookmarks, directories – and search!
 - → Markov model is not a good model of surfing.
 - But it's good enough as a model for our purposes.
- Simple PageRank ranking (as described on previous slide) produces bad results for many pages.
 - Consider the query [video service]
 - The Yahoo home page (i) has a very high PageRank and (ii) contains both *video* and *service*.
 - If we rank all Boolean hits according to PageRank, then the Yahoo home page would be top-ranked.
 - Clearly not desirable
- In practice: rank according to weighted combination of raw text match, anchor text match, PageRank & other factors

Example web graph



Transition (probability) matrix

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
d_1	0.00	0.50	0.50	0.00	0.00	0.00	0.00
d_2	0.33	0.00	0.33	0.33	0.00	0.00	0.00
d_3	0.00	0.00	0.00	0.50	0.50	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.50	0.50
d_6	0.00	0.00	0.00	0.33	0.33	0.00	0.33

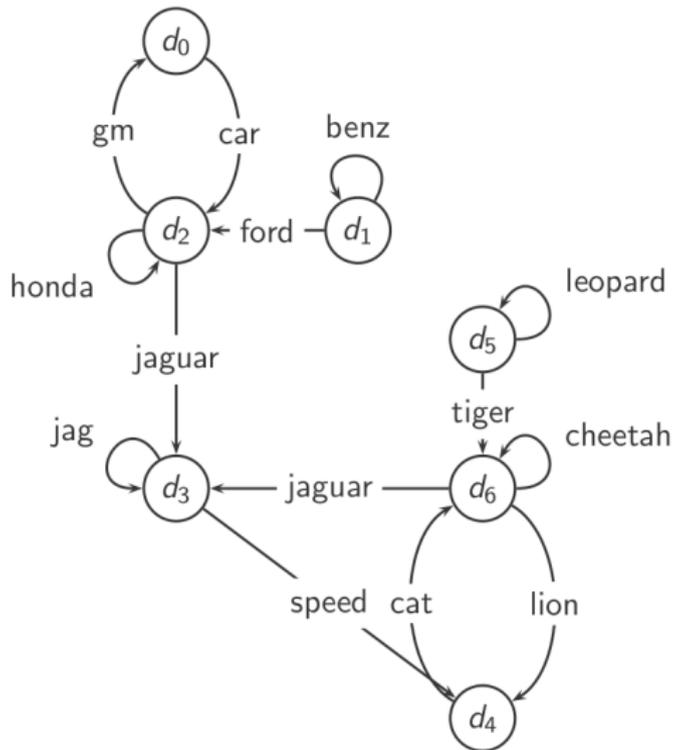
Transition matrix with teleporting ($\alpha = 0.14$)

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.02	0.02	0.88	0.02	0.02	0.02	0.02
d_1	0.02	0.45	0.45	0.02	0.02	0.02	0.02
d_2	0.31	0.02	0.31	0.31	0.02	0.02	0.02
d_3	0.02	0.02	0.02	0.45	0.45	0.02	0.02
d_4	0.02	0.02	0.02	0.02	0.02	0.02	0.88
d_5	0.02	0.02	0.02	0.02	0.02	0.45	0.45
d_6	0.02	0.02	0.02	0.31	0.31	0.02	0.31

Power method vectors $\vec{x}P^k$

	\vec{x}	$\vec{x}P^1$	$\vec{x}P^2$	$\vec{x}P^3$	$\vec{x}P^4$	$\vec{x}P^5$	$\vec{x}P^6$	$\vec{x}P^7$	$\vec{x}P^8$	$\vec{x}P^9$	$\vec{x}P^{10}$	$\vec{x}P^{11}$	$\vec{x}P^{12}$	$\vec{x}P^{13}$
d_0	0.14	0.06	0.09	0.07	0.07	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05
d_1	0.14	0.08	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
d_2	0.14	0.25	0.18	0.17	0.15	0.14	0.13	0.12	0.12	0.12	0.12	0.11	0.11	0.11
d_3	0.14	0.16	0.23	0.24	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.25	0.25	0.25
d_4	0.14	0.12	0.16	0.19	0.19	0.20	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
d_5	0.14	0.08	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
d_6	0.14	0.25	0.23	0.25	0.27	0.28	0.29	0.29	0.30	0.30	0.30	0.30	0.31	0.31

Example web graph



How important is PageRank?

Frequent claim: PageRank is the most important component of web ranking. The reality:

- There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes . . .
- Rumour has it that PageRank in its original form (as presented here) now has a negligible impact on ranking
- However, variants of a page's PageRank are still an essential part of ranking.
- Google's official description of PageRank:

"PageRank reflects our view of the importance of web pages by considering more than 500 million variables and 2 billion terms. Pages that we believe are important pages receive a higher PageRank and are more likely to appear at the top of the search results."

- Addressing link spam is difficult and crucial.

1 Anchor text

2 PageRank

3 Wrap up

- PageRank is topic independent
- We also need to incorporate topicality (i.e. relevance)
- There is a version called Topic Sensitive PageRank
- And also Hyperlink-Induced Topic Search (HITS)

- Anchor text is a useful descriptor of the page it refers to
- Links can be used as another useful retrieval signal - one indicating authority
- PageRank can be viewed as the stationary distribution of a Markov chain
- Power iteration is *one simple method* of calculating the stationary distribution
- Topic sensitive variants exist

- MRS Chapter 21, excluding 21.3.
- MRS 21.3 on HITS algorithm – *optional*