Modern Information Retrieval

Link Analysis[1]

Hamid Beigy

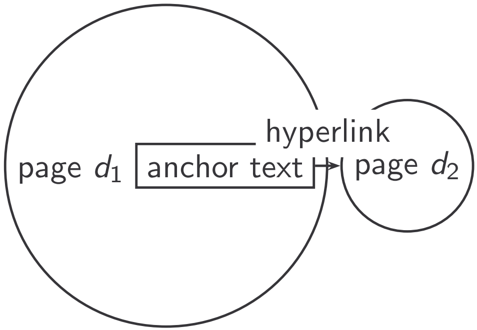
Sharif university of technology

January 2, 2023

[1] Some slides have been adapted from slides of Manning, Yannakoudakis, and Schütze.

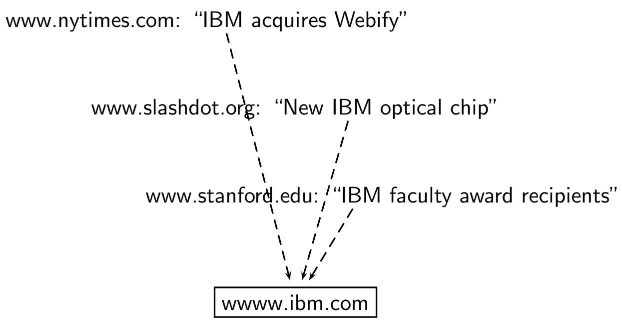
Table of contents

Anchor text



1. Assumption 1: A hyperlink is a quality signal.
   - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1's author seems d_2 high-quality and relevant.
2. 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.

1. Searching on [text of d_2] +[anchor text $\rightarrow d_2$] is often more effective than searching on [text of d_2] only.
2. 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
3. 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.



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

1. Assumption 1: A link on the web is a quality signal –the author of the link thinks that the linked-to page is high-quality.
2. Assumption 2: The anchor text describes the content of the linked-to page.
3. Is assumption 1 true in general?
4. Is assumption 2 true in general?

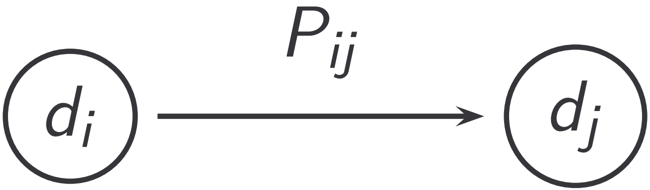
Citation analysis

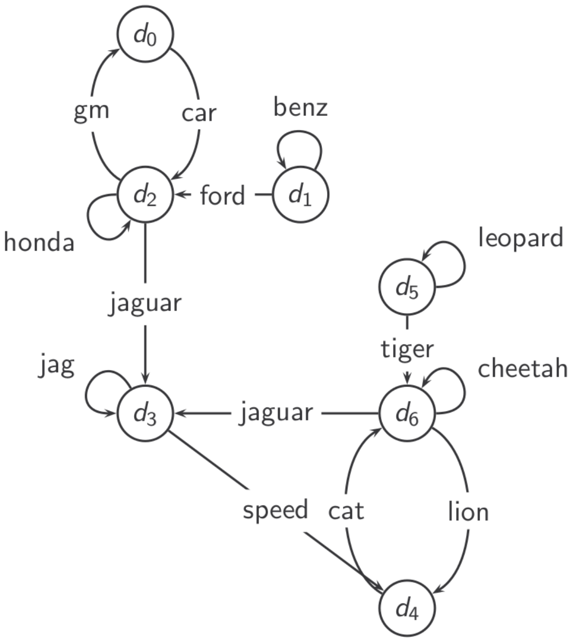
1. Citation analysis: analysis of citations in the scientific literature
2. Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
3. We can view "Miller (2001)" as a hyperlink linking two articles.
4. An application: Citation frequency can be used to measure the impact of a scientific article.
    - Simplest measure: Each citation gets one vote.
    - On the web: citation frequency = inlink count
5. However: A high inlink count does not necessarily mean high quality mainly because of link spam.
6. Better measure: weighted citation frequency or citation rank

PageRank

1. 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
2. In the steady state, each page has a long-term visit rate.
3. This long-term visit rate is the page's PageRank.
4. PageRank = long-term visit rate = steady state probability

1. A Markov chain consists of N states, plus an $N \times N$ transition probability matrix P.
2. state = page
3. At each step, we are on exactly one of the pages.
4. 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.
5. Clearly, for all i, $\sum_{j=1}^{N} P_{ij} = 1$





	PageRank
d_0	0.05
d_1	0.04
d_2	0.11
d_3	0.25
d_4	0.21
d_5	0.04
d_6	0.31

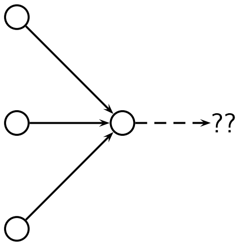
PageRank(d2)< PageRank(d6)

why?

	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

	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

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



1. The web is full of dead ends.
2. Random walk can get stuck in dead ends.
3. If there are dead ends, long-term visit rates are not well-defined.

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

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

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

Theorem (Ergodic Markov chains)

For any ergodic Markov chain, there is a unique long-term visit rate for each state.

1. This is the steady-state probability distribution.
2. Over a long time period, we visit each state in proportion to this rate.
3. It doesn't matter where we start.
4. Teleporting makes the web graph ergodic.
5. $\Rightarrow$ Web-graph+teleporting has a steady-state probability distribution.
6. $\Rightarrow$ Each page in the web-graph+teleporting has a PageRank.

1. A probability (row) vector $\vec{x} = (x_1, \ldots, x_N)$ tells us where the random walk is at any point.
2. Example:

(	0	0	0	...	1	...	0	0	0	)
	1	2	3	...	*i*	...	N-2	N-1	N	

3. More generally: the random walk is on page i with probability x_i.
4. Example:

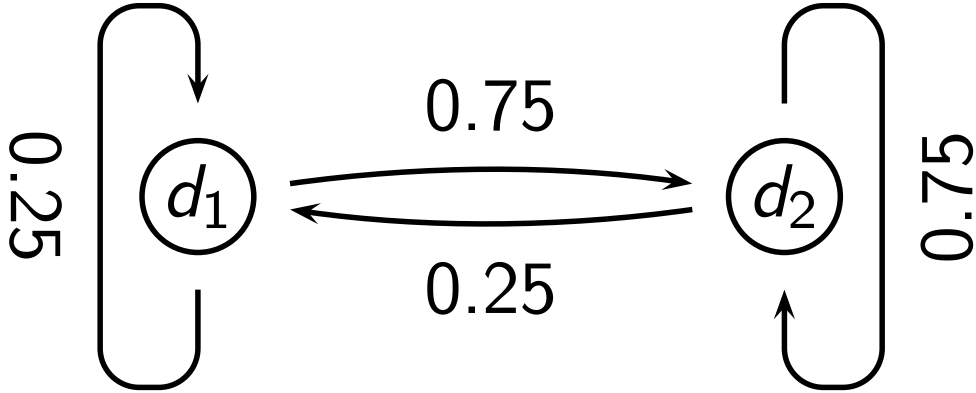
(	0.05	0.01	0.0	...	0.2	...	0.01	0.05	0.03	)
	1	2	3	...	*i*	...	N-2	N-1	N	

5. $\sum x_i = 1$

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

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

- What is the PageRank / steady state in this example?



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

PageRank vector = $\vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$ $P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

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

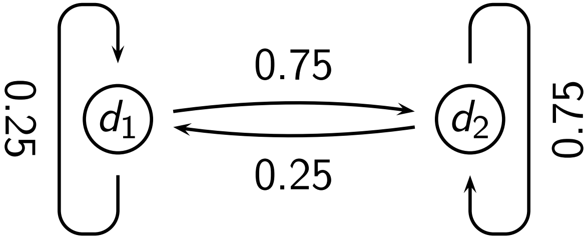
- In other words: how do we compute PageRank?
- Recall: $\vec{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector.
- If the distribution in this step is $\vec{x}$, then the distribution in the next step is $\vec{x}P$.
- But $\vec{\pi}$ is the steady state!
- So: $\vec{\pi} = \vec{\pi}P$
- Solving 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.

$$\lambda\vec{\pi} = \vec{\pi}P$$

- All transition probability matrices have largest eigenvalue 1.

- 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.

- What is the PageRank / steady state in this example?

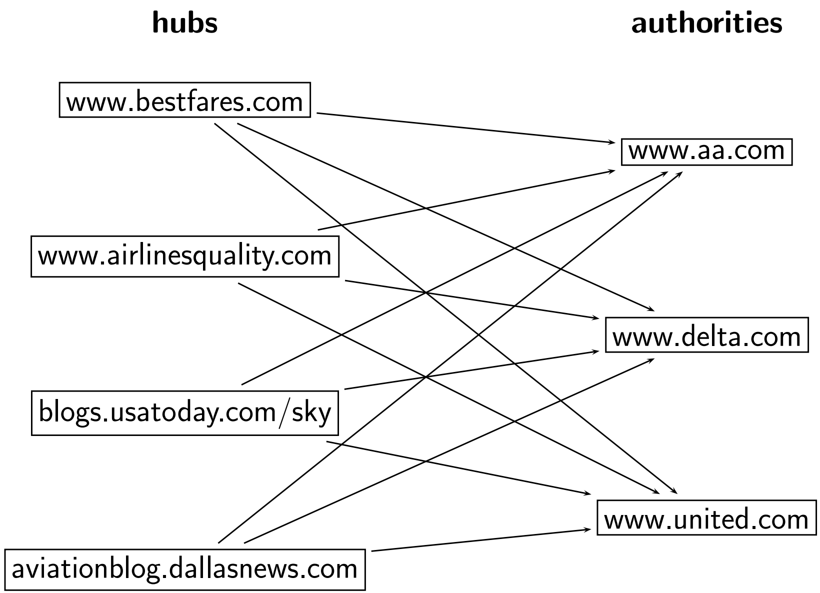


- The steady state distribution (= the PageRanks) in this example are 0.25 for d_1 and 0.75 for d_2.

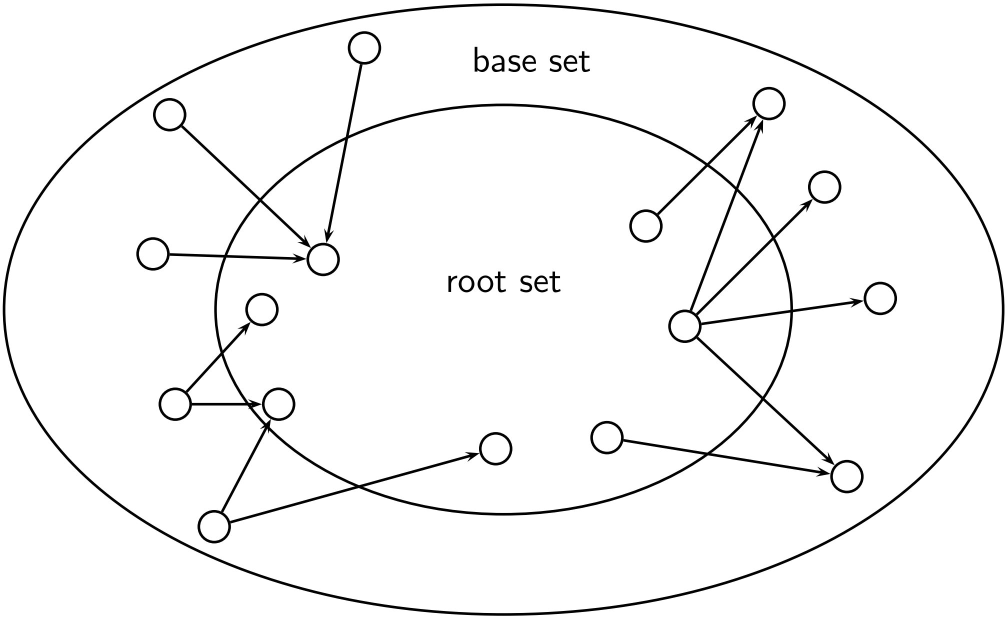
HITS: Hubs & Authorities

- Premise: there are two different types of relevance on the web.
- Relevance type 1: Hubs. A hub page is a good list of [links to pages answering the information need].
- Relevance type 2: Authorities. An authority page is a direct answer to the information need.
- Most approaches to search (including PageRank ranking) don't make the distinction between these two very different types of relevance.

1. A good hub page for a topic links to many authority pages for that topic.
2. A good authority page for a topic is linked by many hub pages for that topic.
3. Circular definition – we will turn this into an iterative computation.


hubs
authorities
www.bestfares.com
www.airlinesquality.com
blogs.usatoday.com/sky
aviationblog.dallasnews.com
www.aa.com
www.delta.com
www.united.com

1. Do a regular web search first
2. Call the search result the root set
3. Find all pages that are linked to or link to pages in the root set
4. Call this larger set the base set
5. Finally, compute hubs and authorities for the base set (which we'll view as a small web graph)


base set
root set

1. Root set typically has 200–1000 nodes.
2. Base set may have up to 5000 nodes.
3. Computation of base set, as shown on previous slide:
   - Follow outlinks by parsing the pages in the root set
   - Find d's inlinks by searching for all pages containing a link to d

1. Compute for each page d in the base set a hub score $h(d)$ and an authority score $a(d)$
2. Initialization: for all d: $h(d) = 1$, $a(d) = 1$
3. Iteratively update all $h(d), a(d)$
4. After convergence:
   - Output pages with highest h scores as top hubs
   - Output pages with highest a scores as top authorities
   - So we output two ranked lists

1. For all d: $h(d) = \sum_{d \mapsto y} a(y)$
2. For all d: $a(d) = \sum_{y \mapsto d} h(y)$
3. Iterate these two steps until convergence
4. Scaling
   - To prevent the $a()$ and $h()$ values from getting too big, can scale down after each iteration
   - Scaling factor doesn't really matter.
   - We care about the relative (as opposed to absolute) values of the scores.
5. In most cases, the algorithm converges after a few iterations.

References

1. Chapter 21 of Introduction to Information Retrieval[2]

[2] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press.

Questions?