Introduction to

Information Retrieval

Hinrich Schütze and Christina Lioma

further modified by Razvan Bunescu

Lecture 12: Link Analysis

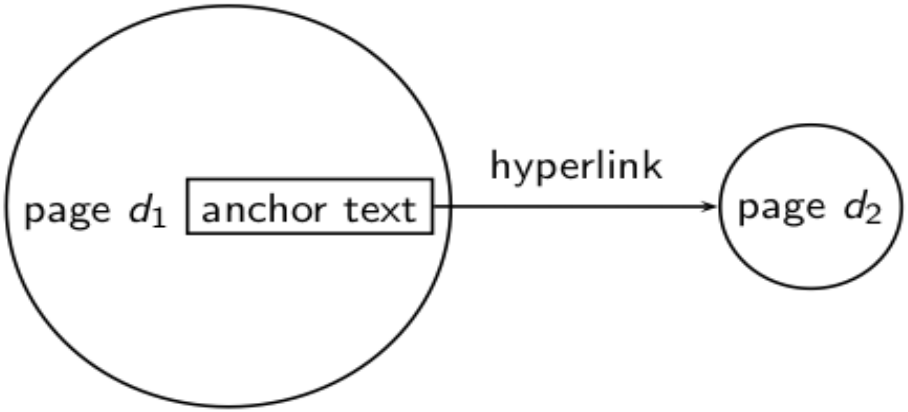
Outline

- **Anchor text**: What exactly are links on the web and why are they important for IR?
- **Citation analysis**: the mathematical foundation of PageRank and link-based ranking.
- **PageRank**: the original algorithm that was used for link-based ranking on the web.
- **Hubs & Authorities**: an alternative link-based ranking algorithm.

Outline

- **Anchor text**: What exactly are links on the web and why are they important for IR?
- **Citation analysis**: the mathematical foundation of PageRank and link-based ranking.
- **PageRank**: the original algorithm that was used for link-based ranking on the web.
- **Hubs & Authorities**: an alternative link-based ranking algorithm.

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 <a href =http://…>here . "
    - Anchor text: "You can find cheap here"

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → 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!

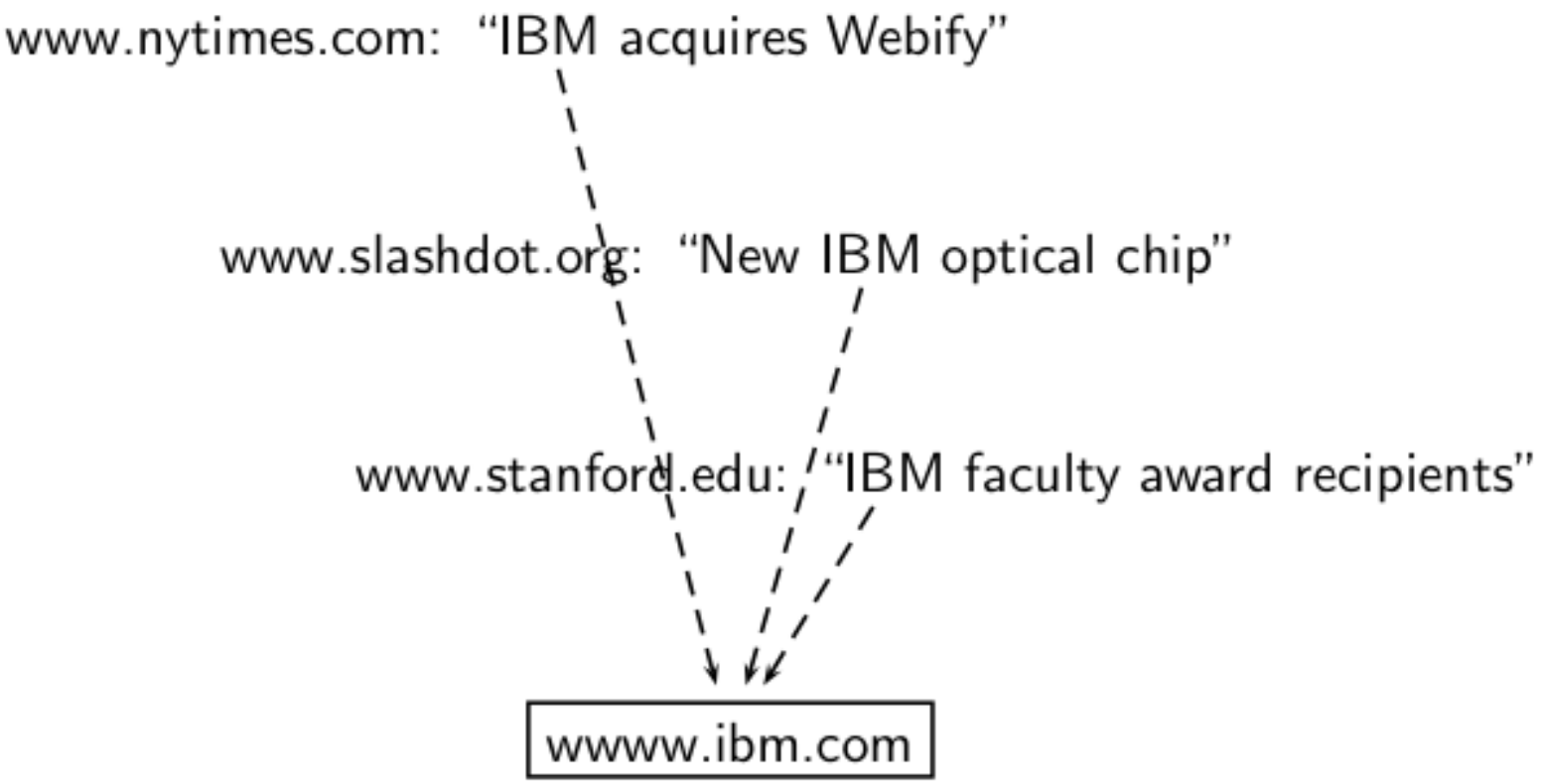
[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → 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

[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → 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 → d_2] is better for the query *IBM*.
  - In this representation, the page with most occurences of *IBM* is www.ibm.com

Anchor text containing *IBM* pointing to www.ibm.com



Indexing anchor text

- 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 Assumption 1&2)

Exercise: Assumptions underlying PageRank

- **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.
  - Is assumption 1 true in general?
- **Assumption 2**: The anchor text describes the content of the linked-to page.
  - Is assumption 2 true in general?

Google bombs

- A Google bomb is a search with "bad" results due to maliciously manipulated anchor text.
- Google introduced a new weighting function in January 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:
  - [dumb motherf…], [who is a failure?], [evil empire]
  - http://en.wikipedia.org/wiki/Google_bomb

Outline

- **Anchor text**: What exactly are links on the web and why are they important for IR?
- **Citation analysis**: the mathematical foundation of PageRank and link-based ranking.
- **PageRank**: the original algorithm that was used for link-based ranking on the web.
- **Hubs & Authorities**: an alternative link-based ranking algorithm.

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature.
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
  - We can view "Miller (2001)" as a hyperlink linking two scientific articles.

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature.
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
  - We can view "Miller (2001)" as a hyperlink linking two scientific articles.
- One application of these "hyperlinks" in the scientific literature:
  - Measure the similarity of two articles by the overlap of other articles citing them.
  - This is called cocitation similarity.
  - Cocitation similarity on the web: Google's "find pages like this" or "Similar" feature.

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article .
  - Simplest measure: Each article gets one vote – not very accurate.
- On the web: citation frequency = inlink count
  - A high inlink count does not necessarily mean high quality ...
    - ... mainly because of link spam.

Origins of PageRank: Citation analysis (2)

- Another application: Citation frequency can be used to measure the impact of an article.
  - Simplest measure: Each article gets one vote – not very accurate.
- On the web: citation frequency = inlink count
  - A high inlink count does not necessarily mean high quality ...
    - ... mainly because of link spam.
- Better measure: weighted citation frequency or citation rank
  - An article's vote is weighted according to its citation impact.
  - Circular? No: can be formalized in a well-defined way.

Origins of PageRank: Citation analysis (3)

- Better measure: **weighted citation frequency or citation rank**.
  - This is basically PageRank.
- **PageRank** was invented in the context of citation analysis by Pinsker and Narin in the 1960s.
  - Asked: which journals are authoritative?
- We can use the same formal representation 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.

Outline

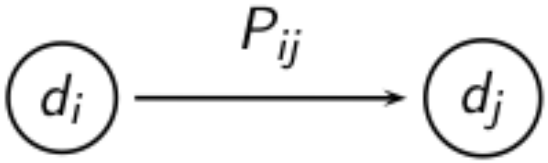
- **Anchor text**: What exactly are links on the web and why are they important for IR?
- **Citation analysis**: the mathematical foundation of PageRank and link-based ranking.
- **PageRank**: the original algorithm that was used for link-based ranking on the web.
- **Hubs & Authorities**: an alternative link-based ranking algorithm.

Model behind PageRank: Random walk

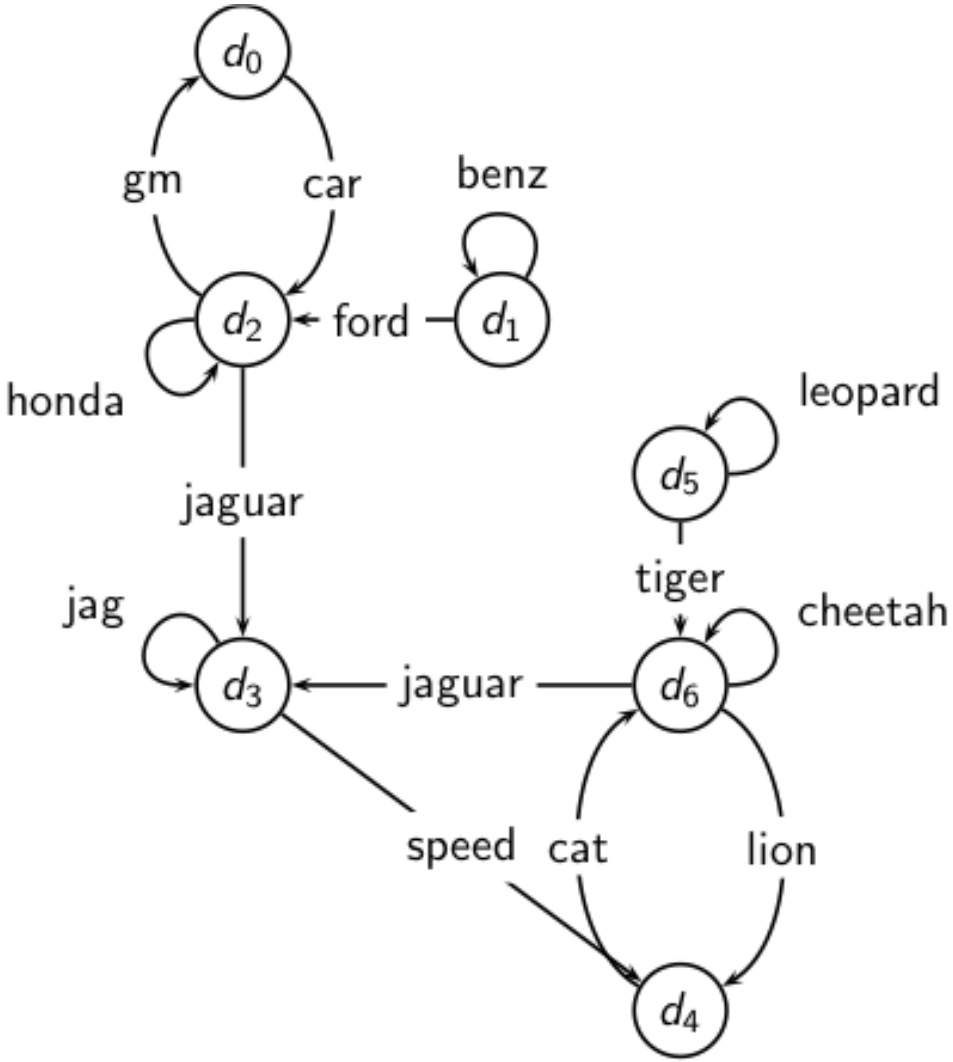
- Imagine a web surfer doing a **random walk** on the web:
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably
- In the steady state, each page has a long-term visit rate.
- This long-term visit rate is the page's PageRank.
- PageRank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

Formalization 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 $1 \le i, j \le N$, 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

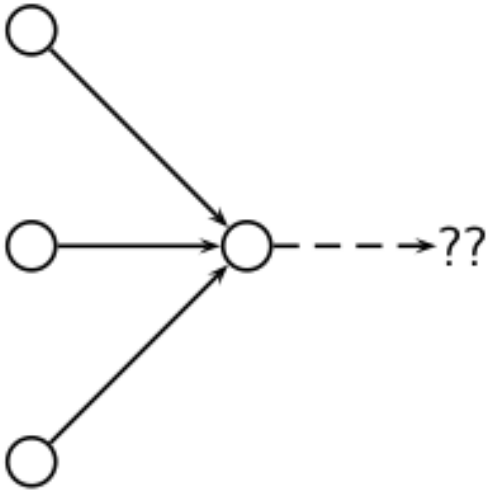
Transition probability matrix P for example

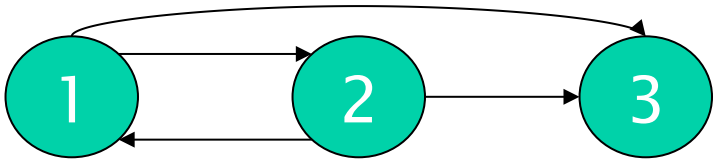
	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
d_1	0.00	0.50	0.50	0.00	0.00	0.00	0.00
d_2	0.33	0.00	0.33	0.33	0.00	0.00	0.00
d_3	0.00	0.00	0.00	0.50	0.50	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.50	0.50
d_6	0.00	0.00	0.00	0.33	0.33	0.00	0.33

Long-term visit rate

- 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.
- 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.
  - Special case: The web graph must not contain dead ends.

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 of dead ends

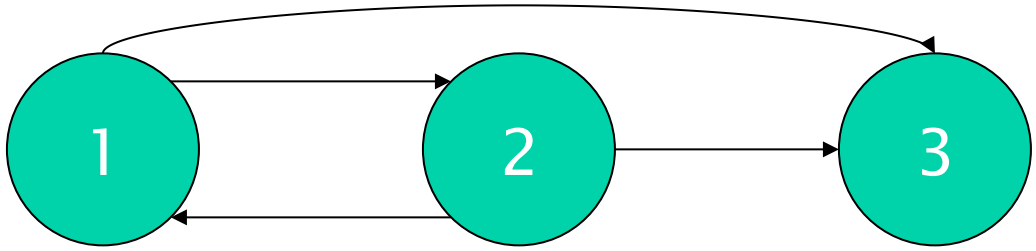
- At a dead end, jump to a random web page with prob. 1/*N*.
- At a non-dead end, with probability 10%, jump to a random web page:
  - To each with a probability of 0.1/*N*.
- With remaining probability (90%), go out on a random hyperlink:
  - For example, if the page has 4 outgoing links, randomly choose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

Build Transition Probability Matrix P

- Build adjacency matrix A such that:
  - A[i, j] = 1, if there is a link from page i to page j.
  - A[i, j] = 0, otherwise.
  - d[i] = # of out-links of page i.
    - d[i] is the sum of the row A[i].

- Build transition probability matrix P as follows:
  - P[i, j] = 1 / N, if d[i] == 0.
  - P[i, j] = α / N, if d[i] > 0 and A[i, j] = 0.
  - P[i, j] = α / N + (1 - α) / d[i], if d[i] > 0 and A[i, j] = 1.

Example: Teleportation Rate $\alpha = 0.15$



P[i, j] = 1 / N, if deg[i] == 0.

P[i, j] = α / N, if deg[i] > 0 and A[i, j] = 0.

P[i, j] = α / N + (1 - α) / d[i], if deg[i] > 0 and A[i, j] = 1.

Adjacency Matrix

	deg	d_1	d_2	d_3
d_1	2	0	1	1
d_2	2	1	0	1
d_3	0	0	0	0

Transition Matrix

	d_1	d_2	d_3
d_1	0.05	0.475	0.475
d_2	0.475	0.05	0.475
d_3	1/3	1/3	1/3

Result 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:

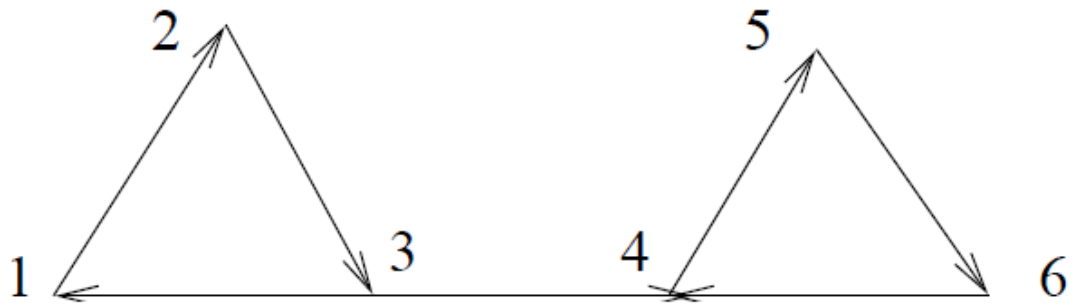


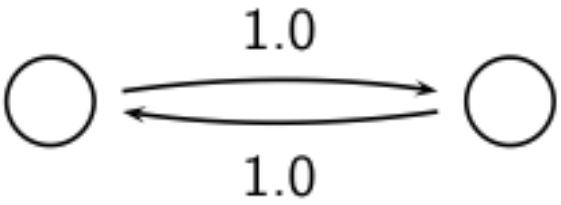
Figure 1: An example of rank sink.

- More generally, we require that the Markov chain be ergodic.

Ergodic Markov chains

- Definition: A Markov chain is **ergodic** if there exists a positive integer T_0 such that for any pair of states i, j, if $p(S_0 = i) = 1$ then $p(S_t = j) > 0$ for all $t > T_0$.

- A Markov chain is **ergodic** if it is irreducible and aperiodic.
  - **Irreducibility**: It is possible to get to any state from any state.
  - **Aperiodicity**. The pages cannot be partitioned such that the random walker visits the partitions sequentially.



a periodic Markov chain

Ergodic Markov chains

- **Theorem 21.1**: 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.

  ⇒ Web-graph + teleporting has a steady-state probability distribution.

  ⇒ Each page in the web-graph+teleporting has a PageRank.

Formalization of "visit": Probability vector

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

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

Formalization of "visit": Probability vector

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

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

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

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

- $\Sigma\, x_i = 1$

Change in probability vector

- If the probability vector is $\boldsymbol{x} = (x_1, \ldots, x_N)^T$ at this step, what is the probability vector $\boldsymbol{y}$ at the next step?
- Recall that row i of the transition probability matrix P tells us where we go next from state i.

Change in probability vector

- If the probability vector is $\boldsymbol{x} = (x_1, \ldots, x_N)^T$ at this step, what is the probability vector $\boldsymbol{y}$ 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 $\boldsymbol{x}$, our next state is distributed as $\boldsymbol{y}^T = \boldsymbol{x}^T P$.

$$y[j] = \sum_{i=1}^{N} x[i] * P[i, j]$$

Steady state in vector notation

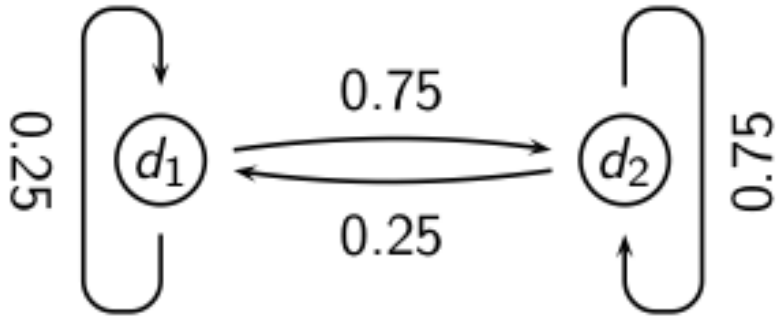
- The steady state in vector notation is simply a vector:

  $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)^T$ of probabilities.

  - We use $\boldsymbol{\pi}$ to distinguish it from the notation for the probability vector $\boldsymbol{x}$.
  - $\boldsymbol{\pi}^T = \boldsymbol{\pi}^T P$
- $\boldsymbol{\pi}$ 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 $P_t(d_2)$		
			$P_{11} = 0.25$ $P_{21} = 0.25$	$P_{12} = 0.75$ $P_{22} = 0.75$
t_0 t_1	0.25	0.75		

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

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Steady-state distribution: Example

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

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

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Steady-state distribution: Example

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

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

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Steady-state distribution: Example

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

PageRank vector = $\boldsymbol{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)^T$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$
$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

How do we compute the steady state vector?

- In other words: how do we compute PageRank?
- Recall: $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- … and if the distribution in this step is $\boldsymbol{x}$, then the distribution in the next step is $\boldsymbol{x}^T P$.

How do we compute the steady state vector?

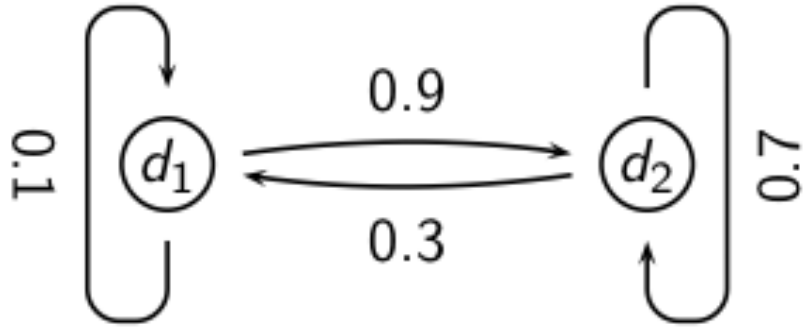
- In other words: how do we compute PageRank?
- Recall: $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$ is the PageRank vector, the vector of steady-state probabilities ...
- … and if the distribution in this step is $\boldsymbol{x}$, then the distribution in the next step is $\boldsymbol{x}^T P$.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

The Power Method for computing PageRank

- Start with any distribution $\boldsymbol{x}$, e.g., uniform distribution:
  - After one step, we're at $\boldsymbol{x}^T P$.
  - After two steps, we're at $\boldsymbol{x}^T P^2$.
  - After k steps, we're at $\boldsymbol{x}^T P^k$.
- Algorithm: multiply $\boldsymbol{x}$ by increasing powers of P until convergence:
  - $|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}| \,/\, N < \tau$
- Recall: regardless of where we start, we eventually reach the steady state π.

Power method: Example

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



Computing PageRank: Power Example

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

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

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

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

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

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
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					$= \vec{x}P^3$
t_3					$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
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			$= \vec{x}P^3$
t_3					$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
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					$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
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			$= \vec{x}P^4$
					. . .
t_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
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_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
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_∞					$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
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			$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
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	$= \vec{x}P^\infty$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$
$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Computing PageRank: Power Example

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$			
			$P_{11} = 0.1$ $P_{21} = 0.3$	$P_{12} = 0.9$ $P_{22} = 0.7$	
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	$= \vec{x}P^\infty$

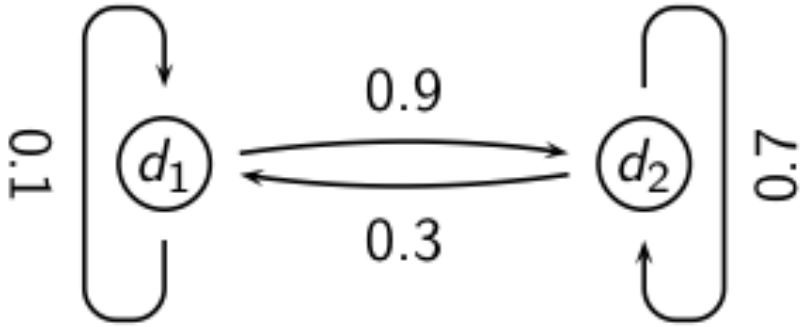
PageRank vector = $\boldsymbol{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

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

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

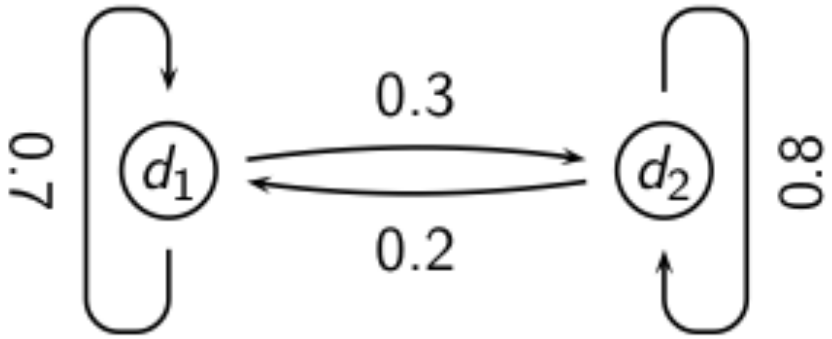
Power method: Example

- 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.

Exercise: Compute PageRank using power method



Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1		
t_1				
t_2				
t_3				
t_∞				

PageRank vector = $\pi = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1				
t_2				
t_3				
t_∞				

PageRank vector = $\pi = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8		
t_2				
t_3				
t_∞				

PageRank vector = $\pi = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2				
t_3				
t_∞				

PageRank vector = $\pi = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7		
t_3				
t_∞				

PageRank vector = $\pi = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3				
t_∞				

PageRank vector = $\pi = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65		
t_∞				

PageRank vector = $\pi = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
t_∞				

PageRank vector = $\boldsymbol{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
			. . .	
t_∞				

PageRank vector = $\boldsymbol{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
			. . .	
t_∞	0.4	0.6		

PageRank vector = $\boldsymbol{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$ $P_{21} = 0.2$	$P_{12} = 0.3$ $P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
			. . .	
t_∞	0.4	0.6	0.4	0.6

PageRank vector = $\boldsymbol{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

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

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

PageRank summary

- PageRank computation:
  - Given graph of links, build transition probability matrix P
    - Apply teleportation.
  - From transition matrix P, compute vector π.
    - π is the dominant eigenvector of P.
    - π_i is the PageRank of page i.
- Query processing:
  - Retrieve pages satisfying the (Boolean) query.
  - Rank them by their PageRank.
  - Return reranked list to the user.

Practical Issues

1) Do not explicitly store transition matrix P for large graphs:
   - May not fit into memory.
   - Compute entries P[i, j] dynamically, based on deg[i] and adjaceny list representation of the graph:

     P[i, j] = 1 / N, if deg[i] == 0.

     P[i, j] = α / N, if deg[i] > 0 and j $\notin$ adj(i).

     P[i, j] = α / N + (1 - α) / d[i], if deg[i] > 0 and j $\in$ adj(i).
   - $\boldsymbol{y} = \boldsymbol{x}^T P$ is equivalent with:

$$y[j] = \sum_{i=1}^{N} x[i] * P[i, j]$$

Practical Issues

2) When using the Power method, check for convergence:
   - $|x^{(k+1)} - x^{(k)}| / N < \tau$
   - Can use either 1-norm or 2-norm.
   - Also set a maximum number of iterations.

PageRank issues

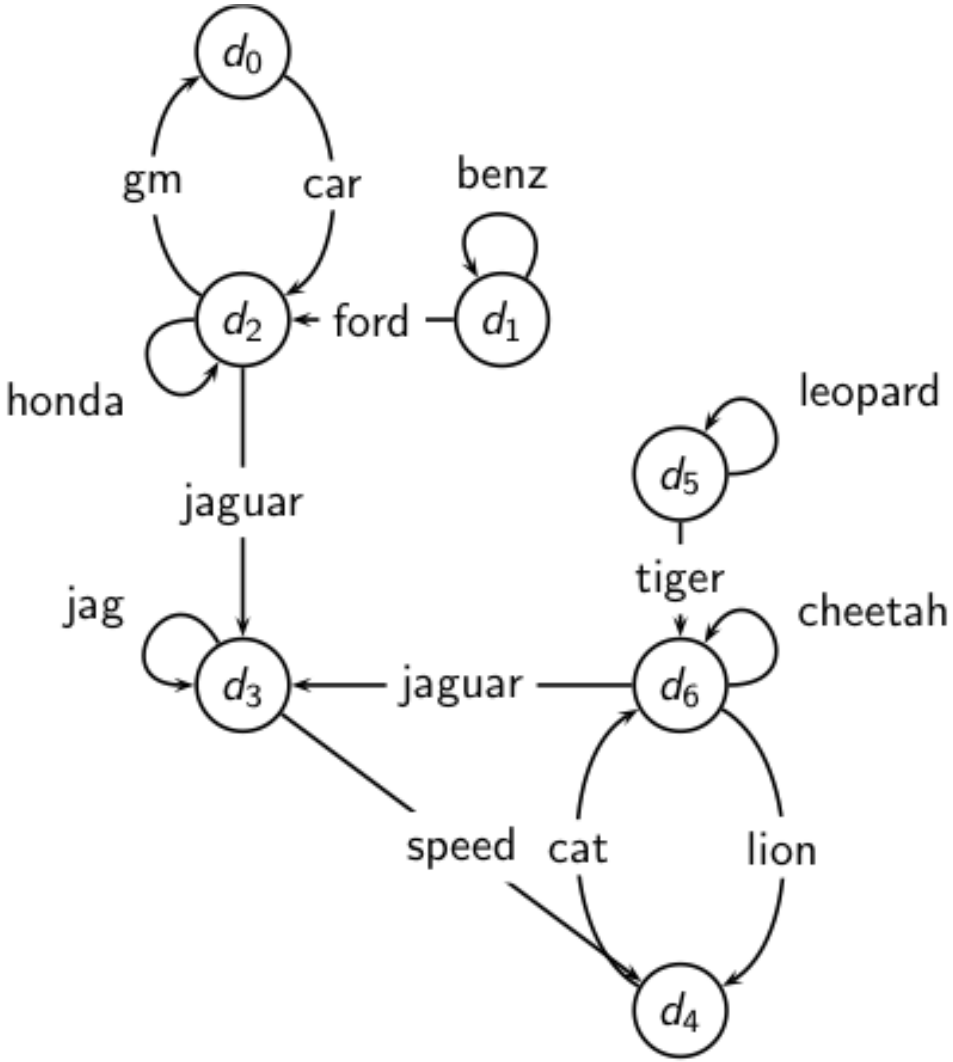
- Real surfers are not random surfers:
  - Examples of nonrandom surfing: back button, short vs. long paths, bookmarks, directories – and search!

    => Markov model is not a good model of surfing.
  - But it's good enough as a model for our purposes.
- Simple PageRank ranking 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.

PageRank issues

- **In practice**: rank according to weighted combination of raw text match, anchor text match, PageRank & other factors.
  - See lecture on Learning to Rank.

Example web graph



Transition matrix without teleporting

	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

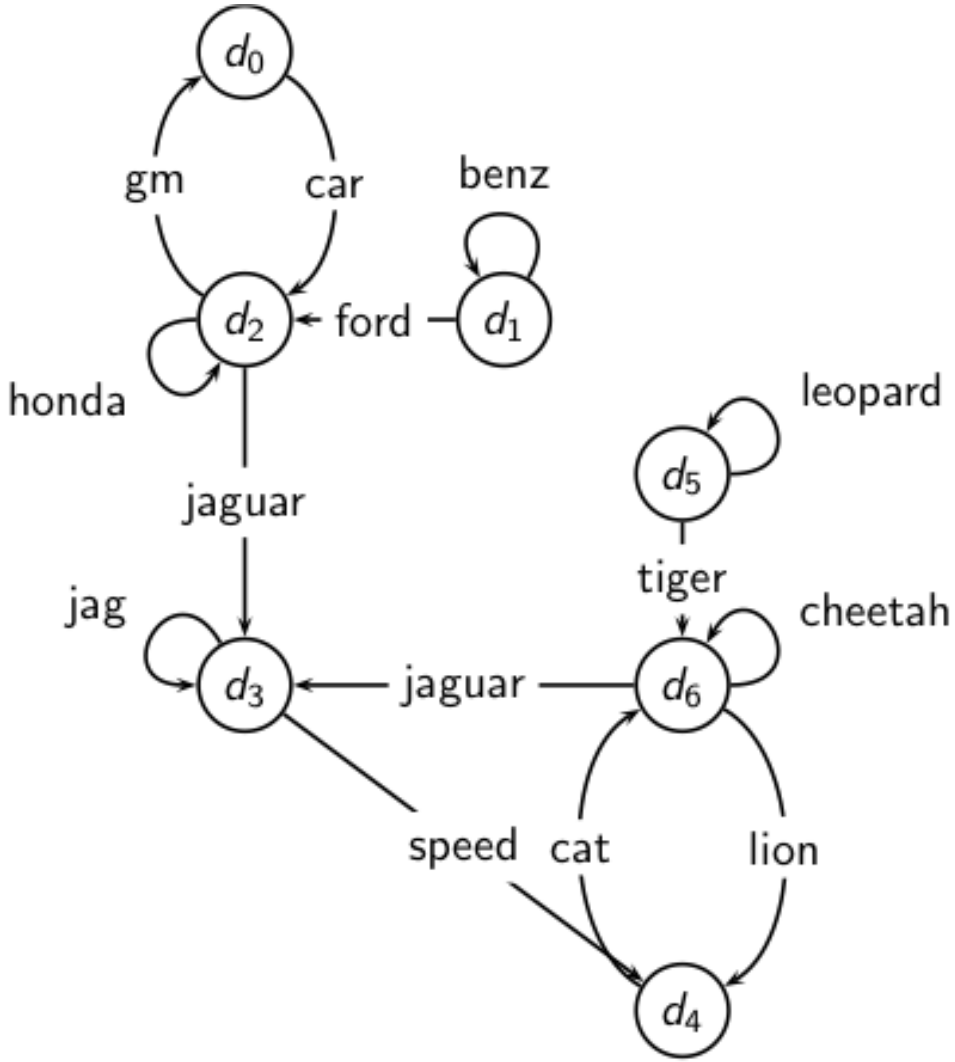
Transition matrix with teleporting $\alpha = 0.14$

	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 $\boldsymbol{x}^T P^k$

	$\boldsymbol{x}^T$	$\boldsymbol{x}^T P^1$	$\boldsymbol{x}^T P^2$	$\boldsymbol{x}^T P^3$	$\boldsymbol{x}^T P^4$	$\boldsymbol{x}^T P^5$	$\boldsymbol{x}^T P^6$	$\boldsymbol{x}^T P^7$	$\boldsymbol{x}^T P^8$	$\boldsymbol{x}^T P^9$	$\boldsymbol{x}^T P^{10}$	$\boldsymbol{x}^T P^{11}$	$\boldsymbol{x}^T P^{12}$	$\boldsymbol{x}^T P^{13}$
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



	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

Personalized PageRank

- **Original**: At a dead end, jump to a random web page with prob. 1/*N*.

- **Personalized**: At a dead end, jump to ?

How important is PageRank?

- **Frequent claim**:
  - PageRank is the most important component of web ranking.

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:
    - anchor text, phrases, proximity, tiered indexes, ...
  - Rumor has it that PageRank in his 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.
    - Adressing link spam is difficult and crucial.

Outline

- **Anchor text**: What exactly are links on the web and why are they important for IR?
- **Citation analysis**: the mathematical foundation of PageRank and link-based ranking.
- **PageRank**: the original algorithm that was used for link-based ranking on the web.
- **Hubs & Authorities**: an alternative link-based ranking algorithm.

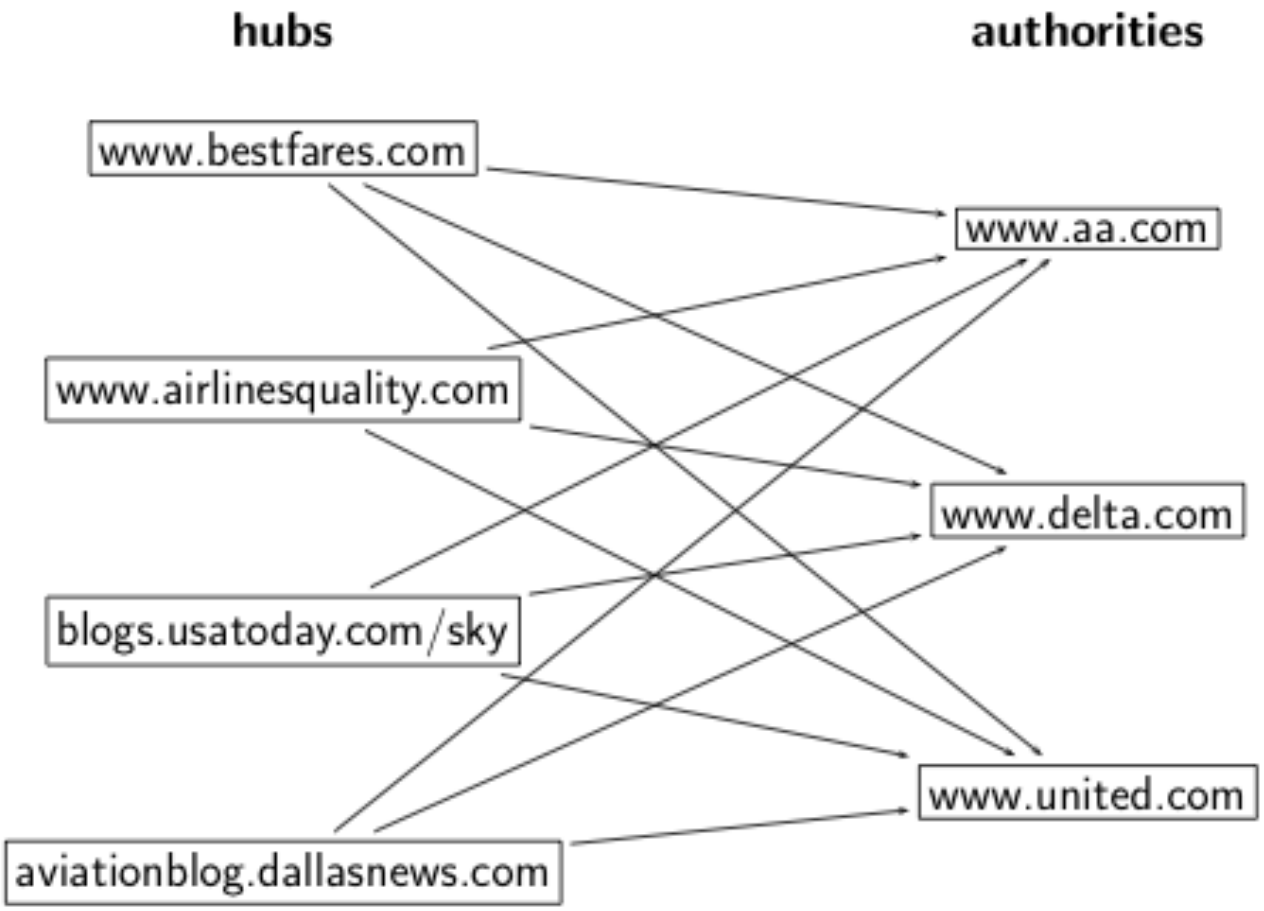
Hyperlink-Induced Topic Search (HITs)

- Premise = there are two different types of relevant pages:
  - Hubs. A hub page is a good list of links to pages answering the information need.
    - http://nlp.stanford.edu/IR-book/information-retrieval.html
  - Authorities. Pages that are recognized as providing significant, trustworthy, and useful information on a topic.
    - http://nlp.stanford.edu/IR-book
    - Links to authority pages occur repeatedly on hub pages.
- Most approaches to search (including PageRank) don't make the distinction between these two very different types of relevance.

Definition: Hubs and Authorities

- A good hub page for a topic links to many authority pages for that topic.
- A good authority page for a topic is linked to by many hub pages for that topic.
- This is a circular definition => we will turn this into an iterative computation.

Example for Hubs and Authorities



HITS

- Algorithm developed by Kleinberg in 1998.
- Based on mutually recursive facts:
  - Hubs point to lots of authorities.
  - Authorities are pointed to by lots of hubs.

  $\Rightarrow$ tend to form a bipartite graph.

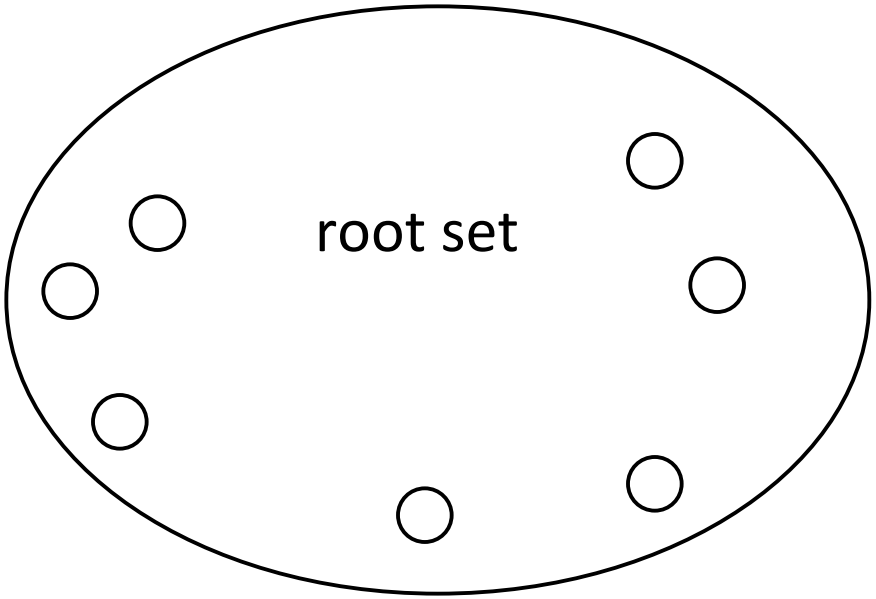
1) First determines a set of relevant pages for the query called the base set S.
2) Analyze the link structure of the web subgraph defined by S to find authority and hub pages in S.

Constructing a Base Set

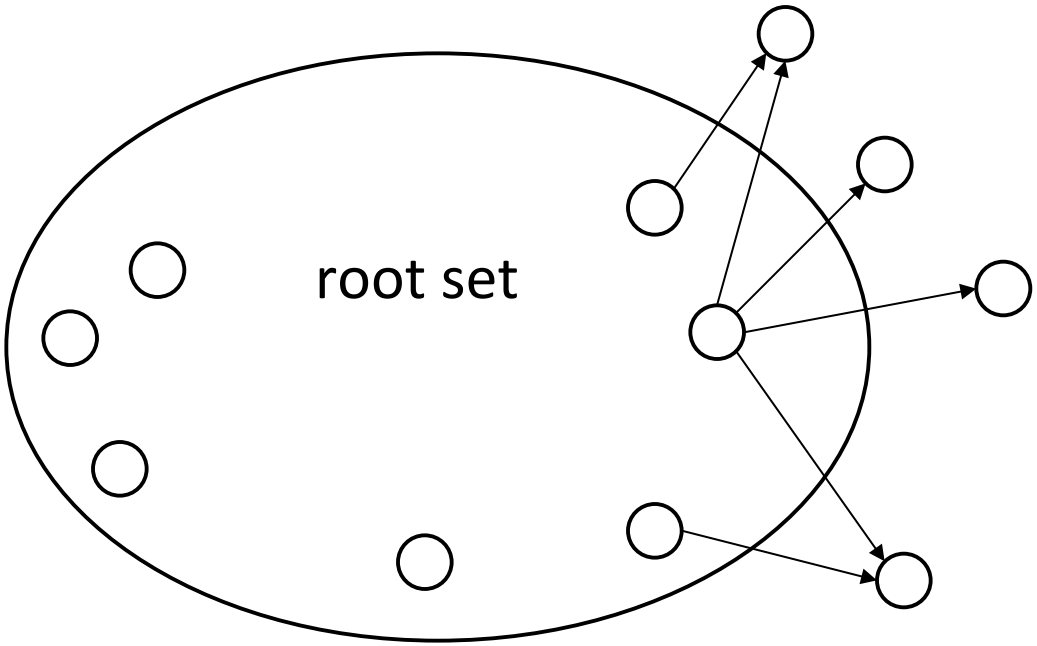
- Given a query Q, do a regular web search first for Q.
- Call the search result the root set.
- Add in all pages that either:
  - link to a page in the root set.
  - are linked from a page in the root set.
- Call this larger set the base set.

- Get in-links (and out-links) from a *connectivity server*.

Constructing a Base Set

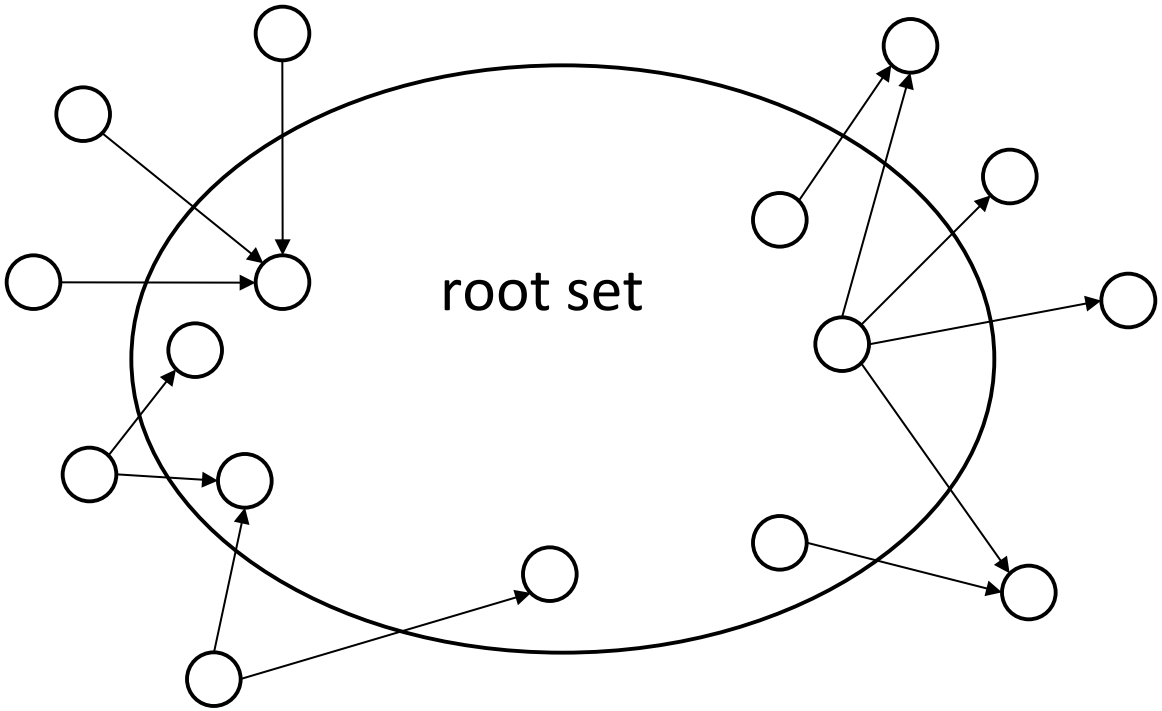


Constructing a Base Set



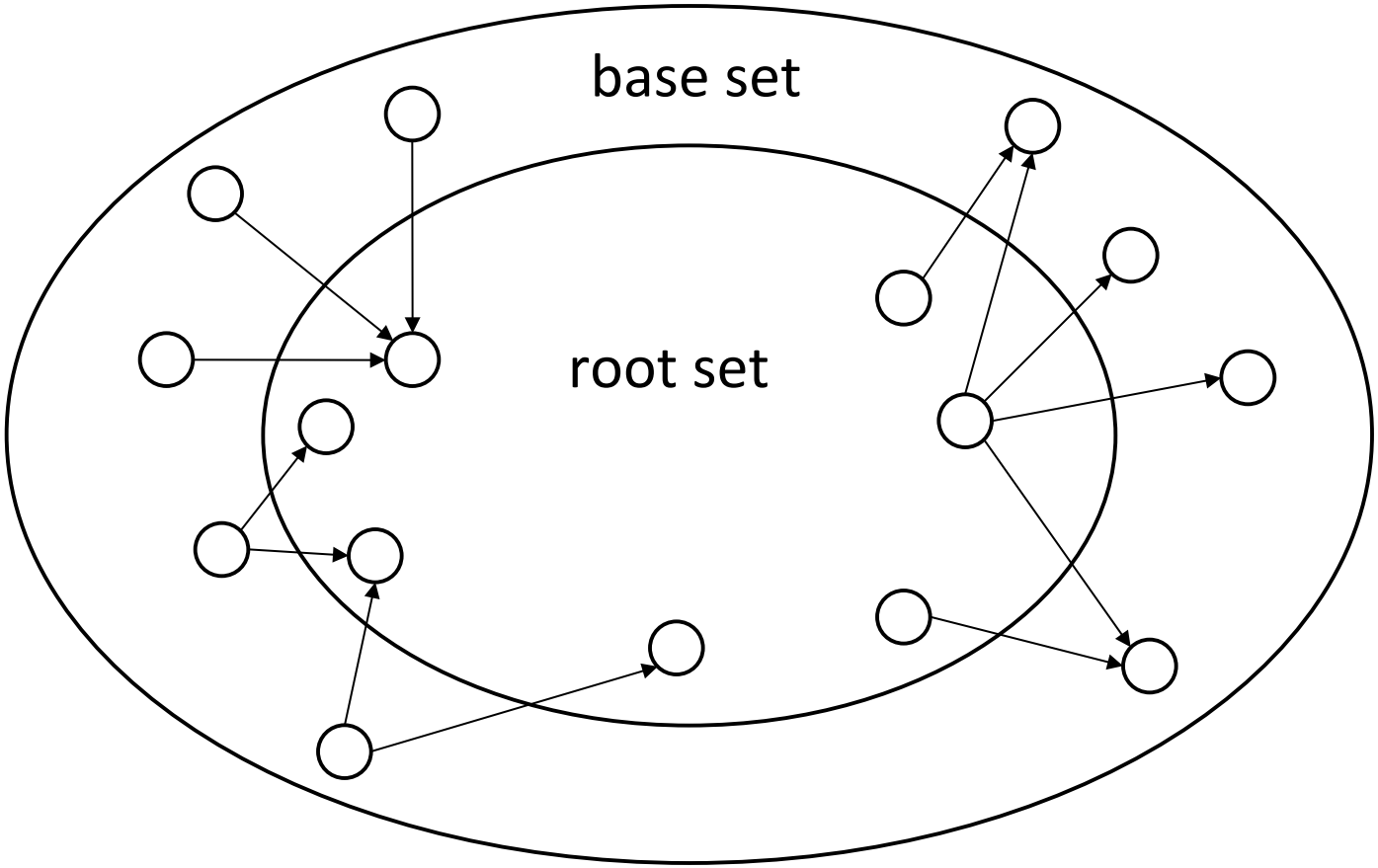
Nodes that root set nodes link to.

Constructing a Base Set



Nodes that link to root set nodes.

Constructing a Base Set



Reasons for Expansion into Base Set

1) A good authority page may not contain the query text:
   - example: “search engine”.

2) If the text query manages to capture a good hub page v_h in the root set, then the inclusion of all pages linked to by v_h will capture other good authorities into the base set.

3) Conversely, if the text query manages to capture a good authority page v_a in the root set, then the inclusion of pages which point to v_a will bring other good hubs into the base set.

=> the “expansion” of the root set into the base set enriches the common pool of good hubs and authorities.

Constructing a Base Set

- To limit computational expense:
  - Limit number of root pages to the top 200 – 1000 pages retrieved for the query.
  - Limit number of "back-pointer" pages to a random set of at most 50 pages returned by a "reverse link" query.
- To eliminate purely navigational links:
  - Eliminate links between two pages on the same host.
- To eliminate "non-authority-conveying" links:
  - Allow only *m* (4-8) pages from a given host as pointers to any individual page.

Distilling Hubs and Authorities

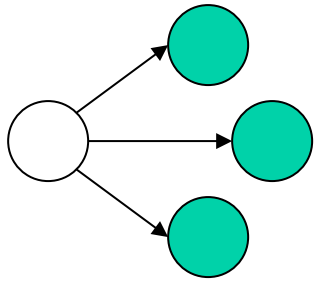
- Compute for each page *d* in the base set:
  - a hub score *h*(*d*)
  - an authority score *a*(*d*)
- Initialization:
  - for all *d*, *h*(*d*) = 1, *a*(*d*) = 1
- **Iteratively update** all *h*(*d*), *a*(*d*).

- After convergence, output two ranked lists:
  - Output pages with highest *h* scores as top hubs
  - Output pages with highest *a* scores as top authrities

Iterative Update

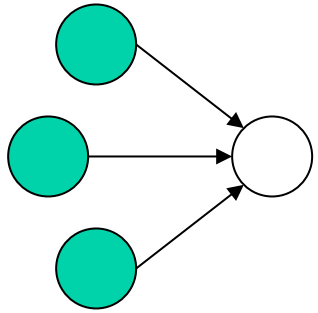
- Repeat the following updates, for all *x*:
  - update hub scores:

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$



  - update authority scores:

$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$

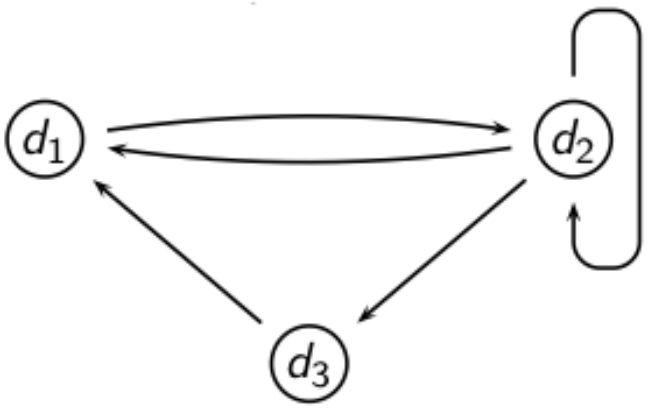


  - normalize using L2 norm such that:

$$\sum_x a(x)^2 = \sum_x h(x)^2 = 1$$

Proof of convergence

- We define an $N \times N$ adjacency matrix A.
- For $1 \leq i, j \leq N$, the matrix entry A_{ij} tells us whether there is a link from page i to page j ($A_{ij} = 1$) or not ($A_{ij} = 0$).



	d_1	d_2	d_3
d_1	0	1	0
d_2	1	1	1
d_3	1	0	0

Write update rules as matrix operations

- Define the hub vector **h** = $(h_1, \ldots, h_N)^T$ as the vector of hub scores.
- Similarity, define **a** as the vector of authority scores

$$h(x) \leftarrow \sum_{x \mapsto y} a(y) \quad \Big\} \Rightarrow \mathbf{h} = A\mathbf{a}$$

$$a(x) \leftarrow \sum_{y \mapsto x} h(y) \quad \Big\} \Rightarrow \mathbf{a} = A^T\mathbf{h}$$

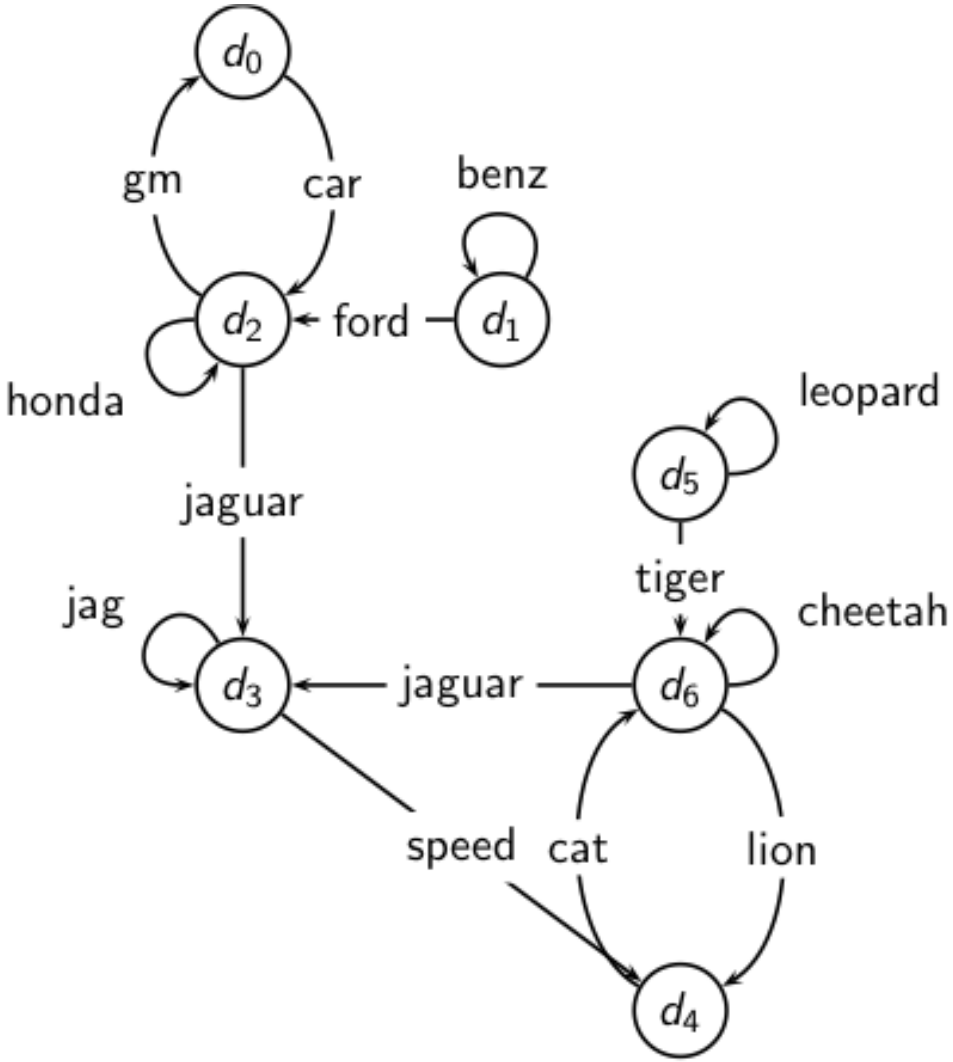
- HITS algorithm in matrix notation:
  1. Compute **h** = A**a**
  2. Compute **a** = A^T**h**
  3. Normalize **h** and **a** using L_2 norm.

  Iterate until convergence:
  - usually gets fairly close to fixed-point solution after 20 iterations.

HITS as an Eigenvector or SVD Problem

- When converged, HITS algorithm outputs **h**, **a** satisfying:
  - $\mathbf{h} = A\mathbf{a}$
  - $\mathbf{a} = A^T\mathbf{h}$
- By substitution we get: $\mathbf{h} = AA^T\mathbf{h}$ and $\mathbf{a} = A^TA\mathbf{a}$.
  - $\Rightarrow$ **h** is the dominant eigenvector of AA^T
    - $\Rightarrow$ h is the leading left singular vector of A.
  - $\Rightarrow$ **a** is the dominant eigenvector of A^TA.
    - $\Rightarrow$ **a** is the leading right singular vector of A.
- So the HITS algorithm is actually a special case of the power method for computing hub and authority scores as eigenvectors.

Example web graph



Raw matrix *A* for HITS

	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	2	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	2	1	0	1

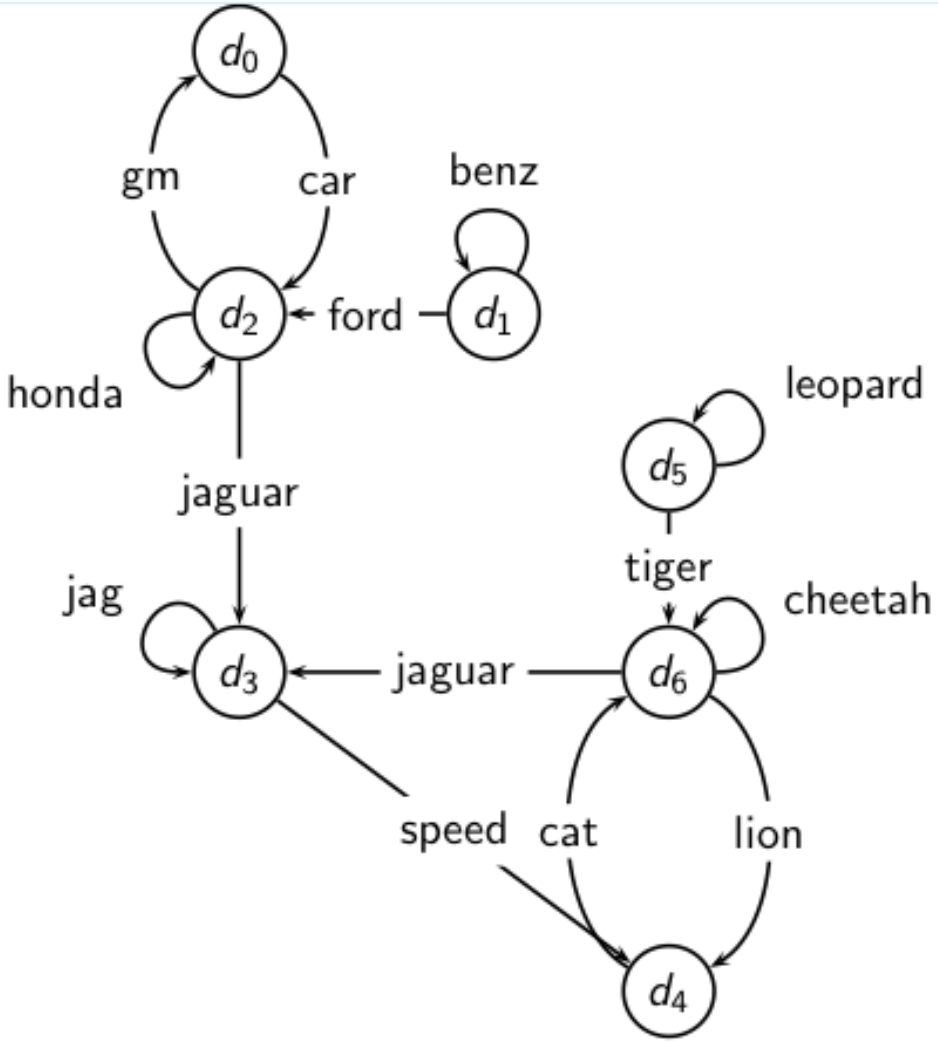
Hub vectors $\mathbf{h}_0$ to $\mathbf{h}_5$

	$\mathbf{h}_0$	$\mathbf{h}_1$	$\mathbf{h}_2$	$\mathbf{h}_3$	$\mathbf{h}_4$	$\mathbf{h}_5$
d_0	0.14	0.06	0.04	0.04	0.03	0.03
d_1	0.14	0.08	0.05	0.04	0.04	0.04
d_2	0.14	0.28	0.32	0.33	0.33	0.33
d_3	0.14	0.14	0.17	0.18	0.18	0.18
d_4	0.14	0.06	0.04	0.04	0.04	0.04
d_5	0.14	0.08	0.05	0.04	0.04	0.04
d_6	0.14	0.30	0.33	0.34	0.35	0.35

Authority vectors $\mathbf{a}_1$ to $\mathbf{a}_7$

	$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathbf{a}_4$	$\mathbf{a}_5$	$\mathbf{a}_6$	$\mathbf{a}_7$
d_0	0.06	0.09	0.10	0.10	0.10	0.10	0.10
d_1	0.06	0.03	0.01	0.01	0.01	0.01	0.01
d_2	0.19	0.14	0.13	0.12	0.12	0.12	0.12
d_3	0.31	0.43	0.46	0.46	0.46	0.47	0.47
d_4	0.13	0.14	0.16	0.16	0.16	0.16	0.16
d_5	0.06	0.03	0.02	0.01	0.01	0.01	0.01
d_6	0.19	0.14	0.13	0.13	0.13	0.13	0.13

Example web graph



	a	h
d_0	0.10	0.03
d_1	0.01	0.04
d_2	0.12	0.33
d_3	0.47	0.18
d_4	0.16	0.04
d_5	0.01	0.04
d_6	0.13	0.35

Top-ranked pages

- Pages with highest in-degree: d_2, d_3, d_6
- Pages with highest out-degree: d_2, d_6
- Pages with highest PageRank: d_6
- Pages with highest in-degree: d_6 (close: d_2)
- Pages with highest authority score: d_3
- Pages with highest hub score: d_2

Finding Similar Pages using HITS

- Create a root set R from t (e.g. 200) pages that point to a given pageP.
- Grow a base set S from R, and run HITS on S.
- Return the top authorities in S as the most similar-pages for P.
  - Authorities in the "link neighbor-hood" of P.

- Given honda.com, algorithm finds:
  - toyota.com, ford.com, bmwusa.com, saturncars.com, nissanmotors.com, audi.com, volvocars.com, ...

Clustering Pages using HITS

- An ambiguous query can result in the principal eigenvector only covering one of the possible meanings.
- Non-principal eigenvectors may contain hubs & authorities for other meanings.

- Example query "jaguar":
  - Atari video game (dominant eigenvector).
  - NFL Football team (2nd dominant eigenvector).
  - Automobile (3rd dominant eigenvector).

PageRank vs. HITS

- PageRank can be precomputed, HITS has to be computed at query time.
  - HITS is too expensive in most application scenarios.
- PageRank and HITS make two different design choices concerning (i) the eigenproblem formalization (ii) the set of pages to apply the formalization to.
- These two are orthogonal: We could also apply HITS to the entire web and PageRank to a small base set.
- Claim: On the web, a good hub almost always is also a good authority.
  - The actual difference between PageRank ranking and HITS ranking is therefore not as large as one might expect.