Link analysis

Giorgio Gambosi

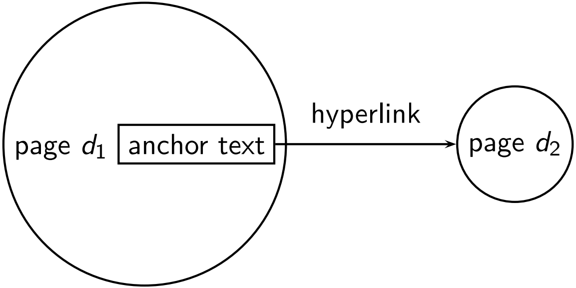
Course of Information Retrieval
CdLM in Computer Science
University of Rome Tor Vergata

Derived from slides produced by C. Manning and by H. Schütze

Link analysis

- The existence of hyperlinks between documents adds information to the collection
- The relevance (absolute or related to a query) of a document can be estimated by considering its relation with other documents
- 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.

Anchor text

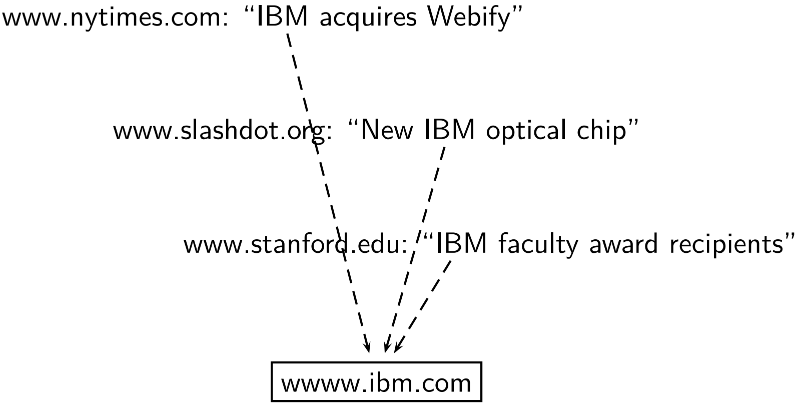


- Assumption 2: The anchor text describes the content of d_2.
  - We use anchor text somewhat loosely here for: the text surrounding the hyperlink.
  - Example: “You can find cheap cars here.”
  - Anchor text: “You can find cheap cars here”

[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.

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 Assumptions 1&2)

Origins of PageRank: Citation analysis

- 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

- 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
- Technique introduced by Pinsker and Narin in the 1960s.
  - 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

- Citation system = weighted directed graph
- nodes = papers
- edges = there is an edge from paper i to paper j if i cites j
- Let $c_{i,j} = 1$ if there exists and edge from i to j
- Let $c_i = \sum_j c_{i,j}$ (total number of references from i)

Origins of PageRank: Citation analysis

- Citation matrix H such that $h_{i,j} = c_{i,j}/c_i$ (fraction of references to j among all the ones declared in i)
  - $h_{i,j} = 1/c_i$ if i cites j
  - $h_{i,j} = 0$ otherwise
- Influence score measures the relevance of i in terms of the number of paper citing it, the number of their references, and their relevance

$$\pi_j = \sum_i \pi_i h_{i,j} = \sum_i \pi_i c_{i,j}/c_i$$

  - $\pi_i c_{i,j}/c_i$ is the amount of influence score of i received by j
  - $\sum_i \pi_i c_{i,j}/c_i$ is the overall amount of influence score received by j
- in matrix notation: $\pi = \pi H$

Origins of PageRank: Citation analysis

The influence of all papers is given by the vector π solution of the matrix equation

$$\pi = \pi H$$

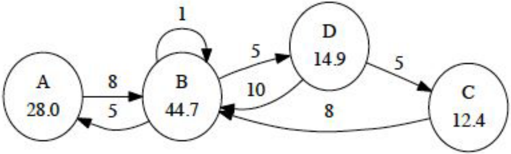
that is, π is the left eigenvector of H associated to eigenvalue $\lambda = 1$

Problem: does such a vector exist for all H?
Does it exist for some special H?

Origins of PageRank: Citation analysis

The same holds for journals:

- Let T_1, T_2 time intervals
- $c_{i,j}$ number of references from papers edited by journal i in T_1 to papers edited by journal j in T_2
- c_i total number of references from papers edited by i in T_1
- again, $\pi = \pi H$



Origins of PageRank: Sociometry

Measuring people prestige through endorsements.

Hubble (1965):

- set of members of a social context
- matrix W, where $w_{i,j}$ is the strength at which i endorses j ($w_{i,j}$ possibly negative)
- prestige π_i of member i defined in terms of the prestige of the endorsers and of their endorsement strengths
- some prestige v_i can be pre-assigned
- in matrix form:

$$\pi = \pi W + v$$

Origins of PageRank: Sociometry

Ranking football teams

Keener (1993):

- set of football teams
- $a_{ij} \geq 0$ score depending on the result of match i vs. j (for example, 1 i won, $1/2$ tie, 0 i lost)
- matrix A, where $a_{i,j}$ is the score of i vs. j
- rank ρ_i of team i defined in terms of the rank of the opponents and of the match result
- $\rho_i = \sum_{j=1}^{n} a_{i,j}\rho_j$ (assume $a_{i,i} = 0$
- in matrix form:

$$\rho = \rho A$$

Origins of PageRank: Econometrics

- economy divided in a number of sectors (industries) producing different goods
- an industry requires a certain amount of inputs to produce a unit of goods
- an industry sells the produced goods to other industries at a certain prize
- equilibrium: each industry balances the costs of production (buying goods) to its revenues (selling products)
- which product prizes guarantee equilibrium (if any)?

Origins of PageRank: Econometrics

- $q_{i,j}$: quantity produced by industry i and used by industry j
- $q_i = \sum_{i=1}^{n} q_{i,j}$: total quantity produced by industry i
- matrix A, where $a_{i,j} = q_{i,j}/q_j$: amount of i's product necessary for a unit of j's product
- π_j: price per unit of the product produced by j
- $c_j = \sum_{i=1}^{n} \pi_i q_{i,j}$ total cost for j
- $r_j = \sum_{i=1}^{n} \pi_j q_{j,i} = \pi_j \sum_{i=1}^{n} q_{j,i} = \pi_j q_j$ total revenue for j

Origins of PageRank: Econometrics

- equilibrium: costs=revenues

$$c_j = \sum_{i=1}^{n} \pi_i q_{i,j} = \pi_j q_j = r_j$$

- divide both sides by q_j

$$\pi_j = \sum_{i=1}^{n} \pi_i \frac{q_{i,j}}{q_j} = \sum_{i=1}^{n} \pi_j a_{i,j}$$

- in matrix notation: $\pi = \pi A$

Idea of Pagerank

- Set of hyperlinked documents
- $a_{i,j} = 1$ if there exists a hyperlink from document i to document j (seen as declaration of interest of j)
- $a_{i,j} = 0$ otherwise
- matrix A: incidence matrix of the web graph
- $a_i = \sum_{j=1}^{n} a_{i,j}$ number of documents hyperlinked from i (outdegree in the graph)
- $a_{i,j}/a_i$ fraction of i expressed judgement of relevant documents assigned to j
- π_i: relevance of document i (assumed also as relevance judge)
- $\pi_i a_{i,j}/a_i$ fraction of i authority assigned to j
- $\pi_j = \sum_{i=1}^{n} \pi_i a_{i,j}/a_i$ total relevance obtained by j from other documents hyperlinking it
- in matrix form: $\pi = \pi A$

Idea of Pagerank

So, a document is relevant if:

- it is linked (voted) by many documents
- these documents cast few votes
- these documents are relevant

A bit of history

- introduced by S. Brin, L. Page (Ph.D. students), R. Motwani and T. Winograd (professors), at Stanford University
  - S. Brin, L. Page "The Anatomy of a Large-Scale Hypertextual Web Search Engine." Proceedings of the 7th international conference on World Wide Web (1998)
  - S. Brin, L. Page,, R. Motwani and T. Winograd"The PageRank Citation Ranking: Bringing Order to the Web." Technical Report. Stanford InfoLab (1999)
- made it possible to automatically rank web pages
- previously, human-based cathegorization (Yahoo!, Altavista)
- IR techniques alone were not satisfactory
- other papers considering citation analysis techniques as a reference for web ranking appeared in the same period
  - M. Marchiori "The Quest for Correct Information on the Web: Hyper Search Engines." Proceedings of the 6th international conference on World Wide Web (1997)
  - J. Kleinberg "Authoritative sources in a hyperlinked environment" Journal of the ACM 46 (5). (1999)

Pagerank

Basic Pagerank formula

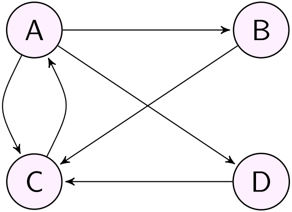
$$\pi(v) = (1-\delta) + \delta \sum_{i=1}^{n} \frac{\pi(v_i)}{o(v_i)}$$

- v is the page of interest
- $v_1, v_2, \ldots, v_n$ pages with a hyperlink to v
- $\pi(v_i)$ Pagerank value of page v_i
- $o(v_i)$ overall number of hyperlinks from v_i
- δ, thedamping factor, controls the amount of Pagerank deriving from hyperlinks (usually $\delta = 0.85$)

Pagerank

- Each page v_i distributes only a fraction δ of its Pagerank, divided by the number of exit hyperlinks.
- The term $(1-\delta)$ can be seen as the Pagerank assigned to a page even if it is not referenced by any other page.
- Recursive formula: iterative update
  - convergence?
  - initial values?

Pagerank computing example



Assuming $\delta = 0.85$, the following holds for all pageranks:

$$\pi_A = 0.15 + 0.85\pi_C$$
$$\pi_B = 0.15 + 0.85\frac{\pi_A}{3}$$
$$\pi_C = 0.15 + 0.85(\frac{\pi_A}{3} + \pi_B + \pi_D)$$
$$\pi_D = 0.15 + 0.85\frac{\pi_A}{3}$$

Pagerank computing example

In matrix form: $\pi = d + 0.85 * \pi A$, where

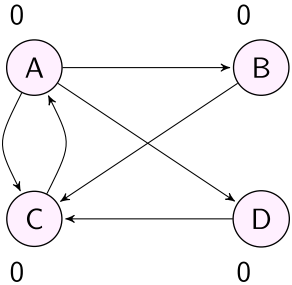
$$\pi = [\pi_A, \pi_B, \pi_C, \pi_D]$$

$$d = [0.15, 0.15, 0.15, 0.15]$$

$$A = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Pagerank computing example

Assume an initial pagerank $\pi = 0$ for all nodes.



We have:

$$\pi_A = 0.15 + 0.85 * 0 = 0.15$$

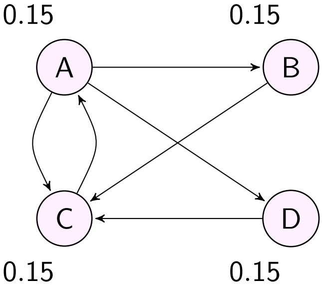
$$\pi_B = 0.15 + 0.85\frac{0}{3} = 0.15$$

$$\pi_C = 0.15 + 0.85\left(\frac{0}{3} + 0 + 0\right) = 0.15$$

$$\pi_D = 0.15 + 0.85\frac{0}{3} = 0.15$$

Pagerank computing example

After 1 step.



We have:

$$\pi_A = 0.15 + 0.85 * 0.15 = 0.2775$$

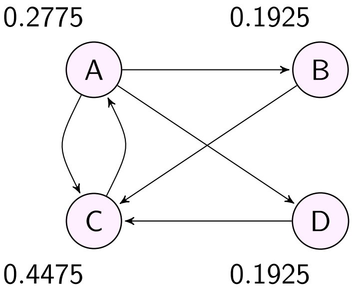
$$\pi_B = 0.15 + 0.85 \frac{0.15}{3} = 0.1925$$

$$\pi_C = 0.15 + 0.85 \left(\frac{0.15}{3} + 0.15 + 0.15 \right) = 0.4475$$

$$\pi_D = 0.15 + 0.85 \frac{0.15}{3} = 0.1925$$

Pagerank computing example

After 2 steps.



We have:

$$\pi_A = 0.15 + 0.85 * 0.4475 = 0.530375$$

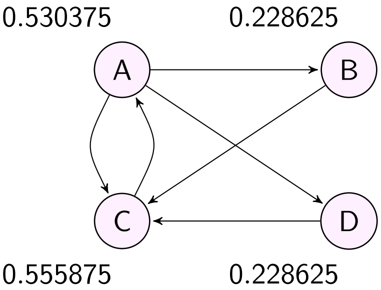
$$\pi_B = 0.15 + 0.85\frac{0.2775}{3} = 0.228625$$

$$\pi_C = 0.15 + 0.85\left(\frac{0.2775}{3} + 0.1925 + 0.1925\right) = 0.555875$$

$$\pi_D = 0.15 + 0.85\frac{0.2775}{3} = 0228625$$

Pagerank computing example

After 3 steps.



We have:

$$\pi_A = 0.15 + 0.85 * 0.555875 \simeq 0.6$$

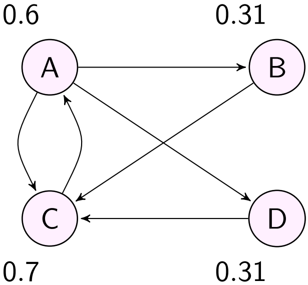
$$\pi_B = 0.15 + 0.85 \frac{0.530375}{3} \simeq 0,31$$

$$\pi_C = 0.15 + 0.85 \left(\frac{0.530375}{3} + 0,228625 + 0,228625 \right) \simeq 0.7$$

$$\pi_D = 0.15 + 0.85 \frac{0.530375}{3} \simeq 0,31$$

Pagerank computing example

After 4 steps.



We have:

$$\pi_A = 0.15 + 0.85 * 0.7 \simeq 0.75$$

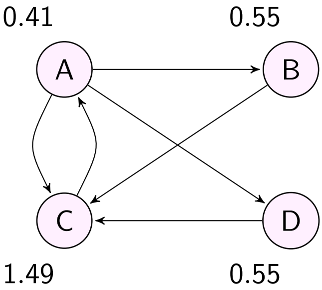
$$\pi_B = 0.15 + 0.85\frac{0.6}{3} \simeq 0,32$$

$$\pi_C = 0.15 + 0.85\left(\frac{0.6}{3} + 0,31 + 0,31\right) \simeq 0.85$$

$$\pi_D = 0.15 + 0.85\frac{0.6}{3} \simeq 0,32$$

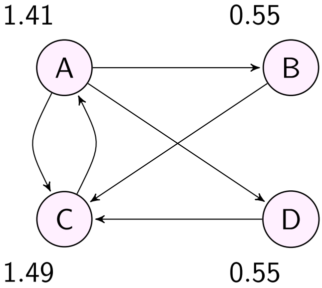
Pagerank computing example

After 100 steps.



Pagerank computing example

After 200 steps.



It converged. Does it always happen?

Pagerank computing example

Different initialization

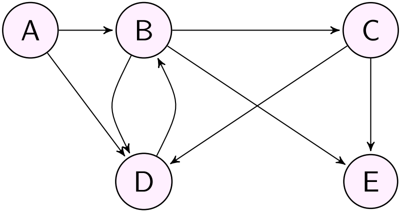
# iterations	π_A	π_B	π_C	π_D
0	1	0.4	0.8	1.5
1	0.83	0.43	2.05	0.43
2	1.89	0.39	1.12	0.39
3	1.1	0.69	1.34	0.69
4	1.29	0.46	1.63	0.46
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	1.41	0.55	1.49	0.55
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
200	1.41	0.55	1.49	0.55

Pagerank computing example

One more initialization

# iterations	π_A	π_B	π_C	π_D
0	0.1	4	0	30
1	0.15	0.18	29.08	0.18
2	24.87	0.19	0.5	0.19
3	0.57	7.2	7.52	7.2
4	6.54	0.31	12.54	0.31
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	1.41	0.55	1.49	0.55
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
200	1.41	0.55	1.49	0.55

A different example



Assuming $\delta = 0.85$, the following holds for all pageranks:

$$\pi_A = 0.15$$
$$\pi_B = 0.15 + 0.85\left(\frac{\pi_A}{2} + \pi_D\right)$$
$$\pi_C = 0.15 + 0.85\frac{\pi_B}{3}$$
$$\pi_D = 0.15 + 0.85\left(\frac{\pi_A}{2} + \frac{\pi_B}{3} + \frac{\pi_C}{2}\right)$$
$$\pi_E = 0.15 + 0.85\left(\frac{\pi_B}{3} + \frac{\pi_C}{2}\right)$$

New pagerank computing example

# iterations	π_A	π_B	π_C	π_D	π_E
0	0	0.4	0.2	1.6	2.1
1	0.15	1.51	0.26	0.35	0.35
2	0.15	0.51	0.58	0.75	0.69
3	0.15	0.85	0.29	0.6	0.54
4	0.15	0.73	0.39	0.58	0.52
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	0.15	0.68	0.34	0.55	0.49
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
200	0.15	0.68	0.34	0.55	0.49

The importance of δ

Let $\delta = 0.2$

# iterations	π_A	π_B	π_C	π_D	π_E
0	0	0.4	0.2	1.6	2.1
1	0.8	1.12	0.83	0.85	0.85
2	0.8	1.05	0.87	1.04	0.96
3	0.8	1.09	0.87	1.04	0.96
4	0.8	1.09	0.87	1.04	0.96
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
100	0.8	1.09	0.87	1.04	0.96
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
200	0.8	1.09	0.87	1.04	0.96

Different score, same ranking

The importance of δ

Let $\delta = 1$

# iterations	π_A	π_B	π_C	π_D	π_E
0	0	0.4	0.2	1.6	2.1
1	0	1.6	0.13	0.23	0.23
2	0	0.23	0.53	0.6	0.6
3	0	0.6	0.08	0.34	0.34
4	0	0.34	0.2	0.24	0.24
5	0	0.24	0.11	0.21	0.21
6	0	0.21	0.08	0.14	0.14
7	0	0.14	0.07	0.11	0.11
8	0	0.11	0.05	0.08	0.08
9	0	0.08	0.04	0.06	0.06
10	0	0.06	0.03	0.05	0.05
11	0	0.05	0.02	0.03	0.03
12	0	0.03	0.02	0.03	0.03
13	0	0.03	0.01	0.02	0.02
14	0	0.02	0.01	0.01	0.01
15	0	0.01	0.01	0.01	0.01
16	0	0.01	0	0.01	0.01
17	0	0.01	0	0.01	0.01
18	0	0.01	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0

Markov chains

- A stochastic process is a set $\mathcal{X}$ of random variables defined on the same domain $\mathcal{S}$ (state space)
- Can be interpreted as a single r.v. evolving on time
- We are interested in the case $\mathcal{X} = \{X_0, X_1, X_2, \ldots\}$ (discrete stochastic process) and $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ (finite state space)
- A Markov chain is a discrete stochastic process on a finite space such that for all $n = 0, 1, 2, \ldots$

$$p(X_n = s_n | X_{n-1} = s_{n-1}, \ldots, X_0 = s_0) = p(X_n = s_n | X_{n-1} = s_{n-1})$$

- In a Markov chain X_n depends only on X_{n-1} (memoryless)

Stationary markov chains

- If $p(X_n|X_{n-1})$ does not depend on n (the probability distribution of states is the same for each transition), the chain is stationary
- transition matrix M, with $M_{i,j} = p(X_n = s_i|X_{n-1} = s_j)$
- equivalent, weighted directed graph

$$\begin{aligned} N &= \mathcal{S} \\ E &= \{< s_i, s_j | p(X_n = s_i|X_{n-1} = s_j) > 0\} \\ w(< s_i, s_j >) &= p(X_n = s_i|X_{n-1} = s_j) \end{aligned}$$

MC example: weather in Oz

- In the Land of Oz day can be nice (n), rainy (r), snowy (s)
- Tuesday's weather depends (in probability) only on Monday's one according to the following transition matrix

$$M = \begin{array}{c c} & \begin{array}{ccc} \text{r} & \text{n} & \text{s} \end{array} \\ \begin{array}{c} \text{r} \\ \text{n} \\ \text{s} \end{array} & \begin{pmatrix} .5 & .25 & .25 \\ .5 & 0 & .5 \\ .25 & .25 & .5 \end{pmatrix} \end{array}$$

- That is, for example,

$$p(T = r | M = n) = .5$$

MC example: marginal probabilities

Clearly,

$$
\begin{aligned}
p(T = r) = p(T = r|M = r)p(M = r)+ \\
p(T = r|M = n)p(M = n)+ \\
p(T = r|M = s)p(M = s)
\end{aligned}
$$

That is, if $\pi^{(0)} = [p(M = r), p(M = n), p(M = s)]$ and $\pi^{(1)} = \pi^{(0)}M$, then we have

$$p(T = r|M = n) = \pi_1^{(1)}$$

MC example: deriving probabilities

Note that Wednesday's weather indirectly depends on Monday's one. In fact,

$$\begin{aligned} p(W=r|M=n) &= p(W=r|T=r)p(T=r|M=n)+ \\ & p(W=r|T=n)p(T=n|M=n)+ \\ & p(W=r|T=s)p(T=s|M=n) \\ &= M_{11}M_{12}+M_{12}M_{22}+M_{13}M_{32} \\ &= M^2_{12} \end{aligned}$$

In general, $p(X_n=s_i|X_{n-2}=s_j)=M^2_{ij}$

MC example: deriving probabilities

The same holds for any probability $p(X_n|X_{n-k})$

$$p(X_n = s_i|X_{n-k} = s_j) = M_{ij}^k$$

Given an initial probability distribution $\pi^{(0)}$, it results that the probability distribution after k transitions is

$$\pi^{(k)} = \pi^{(0)} M^k$$

MC example: deriving probabilities

For example, if $\pi^{(0)} = [.5, .25, .25]$

$$\pi^{(1)} = \pi^{(0)} M = [.5, .25, .25] \begin{bmatrix} .5 & .25 & .25 \\ .5 & 0 & .5 \\ .25 & .25 & .5 \end{bmatrix} = [.4375, .1875, .375]$$

$$\pi^{(2)} = \pi^{(0)} M^2 = [.5, .25, .25] \begin{bmatrix} .4375 & .1875 & .375 \\ .375 & .25 & .375 \\ .375 & .1875 & .4375 \end{bmatrix} = [.40, .21, .39]$$

$$\pi^{(3)} = \pi^{(0)} M^3 = [.5, .25, .25] \begin{bmatrix} .4 & .2 & .4 \\ .4 & .2 & .4 \\ .4 & .2 & .4 \end{bmatrix} = [.4, .2, .4]$$

MC example: deriving probabilities

Since

$$\begin{bmatrix} .4 & .2 & .4 \\ .4 & .2 & .4 \\ .4 & .2 & .4 \end{bmatrix} \begin{bmatrix} .5 & .25 & .25 \\ .5 & 0 & .5 \\ .25 & .25 & .5 \end{bmatrix} = \begin{bmatrix} .4 & .2 & .4 \\ .4 & .2 & .4 \\ .4 & .2 & .4 \end{bmatrix}$$

we have that

$$[.5, .25, .25] \begin{bmatrix} .4 & .2 & .4 \\ .4 & .2 & .4 \\ .4 & .2 & .4 \end{bmatrix} = [.4, .2, .4] = [.4, .2, .4] \begin{bmatrix} .5 & .25 & .25 \\ .5 & 0 & .5 \\ .25 & .25 & .5 \end{bmatrix}$$

that is, after a certain number of transition, the resulting probability distribution $[.4, .2, .4]$ is stationary (remains unchanged). This the long term probability of all states.

Stationary distribution

Given a Markov chain on n states, with transition matrix M, and given an initial distribution $\pi^{(0)}$, the stationary distribution (or steady state) π of the MC (if it exists) is given by

$$\lim_{k\to\infty} \pi^{(k)} = \pi^{(0)} \lim_{k\to\infty} M^k$$

equivalently,

$$\pi = \pi M$$

Open problems:

- does the stationary distribution always exist?
- if not, when does it exist?
- if it exists, how to compute it?
- does it depends on $\pi^{(0)}$?

Usefulness of MC

Why are we interested in Markov chains?

- 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

But we would like that

- the steady state indeed exists
- it is independent from the initial page

Perron Frobenius theory

- Developed 1 century ago (1907, 1912) by Oskar Perron and Georg Frobenius
  - applied to positive and non negative square matrices
  - spectral (eigenvalues, eigenvectors) characterization of the matrices

Perron theorem

For any $A^{n \times n} > 0$, we have that:

1. the (right) spectral radius is positive, $\rho(A) = \max_{\lambda_i} |\lambda| > 0$, where λ_i is a (right) eigenvalue of A ($Aw_i = \lambda_i w_i$ for some w_i)
2. $r = \rho(A)$ is a (right) eigenvalue of A, denoted as (right)Perron root
3. r is a simple (right) eigenvalue, that is, it is a simple root of the characteristic polynomial $|\lambda I - A|$: this implies that there exists only one (right) eigenvector associated to r
4. as a consequence, there exists a unique vector p (named right Perron vector) such that
    - $Ap = rp$
    - $p > 0$
    - $|p|_1 = \sum_{i=1}^{n} p_i = 1$
5. r is the only (right) eigenvalue on the spectral circle (such that $|r| = \rho(A)$)

Perron theorem: left eigenspace case

The same properties hold also for left eigenvalues/eigenvectors: recall that for a left eigenvalue/eigenvector pair (λ, w) we have

$$wA = \lambda w \Longrightarrow A^T w^T = \lambda w^T$$

Then, for any $A^{n\times n} > 0$, we have that:

1. $\rho(A^T) = \max_{\lambda_i} |\lambda| > 0$, where λ_i is a (left) eigenvalue of A
2. $r' = \rho(A^T)$ is an eigenvalue of A^T, (left) Perron root
3. r' is a simple eigenvalue, that is, it is a simple root of the characteristic polynomial $|\lambda I - A^T|$
4. there is a unique vector p (named left Perron vector) such that
   - $A^T p = r' p$
   - $p > 0$
   - $|p|_1 = \sum_{i=1}^n p_i = 1$

   and there exist no other eigenvector $w \geq 0$ of A^T (except cp)
5. r' is the only eigenvector such that $|r| = \rho(A^T)$

Why is Perron theorem interesting?

Let us return to Markov chains:

- the i-th row of M lists the probabilities $p(X_{n+1} = s_j | X_n = s_i)$, then
  - $M_{ij} \geq 0$ for all i, j
  - $\sum_{j=1}^{n} M_{ij} = 1$ for all i
- the matrix is said (row) stochastic; observe that in general, it is not column stochastic
- then we could prove that $\rho(M) = 1$ and that $e^T = [1, \dots, 1]$ is the corresponding (right) eigenvector, that is $Me = e$

So what?

- Perron theorem is not applicable to M, since M is just non negative (it should be positive)
- Even if we could apply it, it would result that $r = 1$ is a simple (right) eigenvalue with Perron vector e/n: in fact, $Me/n = e/n$, with $|e/n|_1 = 1$
- But we are interested in finding π such that $\pi = \pi M$, that is, $M^T \pi^T = \pi^T$
- that is, we are interested in the left Perron vector (the steady state distribution)

We need something more

Under some conditions (to be stated later) the following holds

$$\lim_{k\to\infty}\left(\frac{A}{r}\right)^k = \frac{pq^T}{qp^T}$$

where

- A is a square matrix
- $r = \rho(A)$
- p is the right Perron vector of A: $p \in \mathbb{R}^{n\times 1}$
- q is the left Perron vector of A: $q \in \mathbb{R}^{1\times n}$

Exploiting the new property

Since, for a stochastic matrix M, $(1, e)$ is a right Perron pair and $(1, \pi)$ is a left Perron pair, it would result

$$\lim_{k\to\infty}\left(\frac{M}{1}\right)^k = \frac{e\pi}{\pi^T e^T} = e\pi = \begin{pmatrix} \pi_1 & \pi_2 & \cdots & \pi_n \\ \pi_1 & \pi_2 & \cdots & \pi_n \\ \vdots & \vdots & \ddots & \vdots \\ \pi_1 & \pi_2 & \cdots & \pi_n \end{pmatrix}$$

For the steady state distribution we would get

$$\lim_{k\to\infty} \pi^{(k)} = \pi^{(0)} \lim_{k\to\infty} M^k = \pi^{(0)} e\pi = \pi$$

that is, independent from the initial distribution $\pi^{(0)}$

Exploiting the new property

We also obtain an indication on how to compute π

- choose any initial distribution π_0 (for example $[1/n, \ldots, 1/n]$)
  - set $M' \leftarrow M$
  - iterate
    - $M \leftarrow M'$
    - $M' \leftarrow M^2$
  - until $\text{dist}(M, M') < \epsilon$
- set $\pi = \pi_0 M$

This is called power method

What conditions we need?

$$\lim_{k\to\infty}\left(\frac{A}{r}\right)^k = \frac{pq^T}{qp^T}$$

holds iff:

1. A is non negative (holds)
2. A has exactly one eigenvalue λ on the spectral circle (that is s.t $|\lambda| = \rho(A)$) (holds)
3. A is irreducible

If this case the matrix is said primitive.

Reducible matrices

A square matrix A is reducible if it is possible to permutate its rows to obtain a new matrix

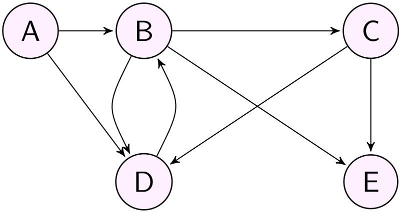
$$A' = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix}$$

where

- X and Z are $m \times m$ and $(n-m) \times (n-m)$ matrices, with $0 < m < n$
- 0 is the null matrix

Reducible Markov chains

- If A is the transition matrix of a Markov chain, reducibility means that there exists a subset of states (corresponding to the rows in Z) from which the chain cannot exit



- A markov chain is irreducible if it is always possible to go from each state to any other state

Primitivity

A simple condition:

A matrix A is primitive iff there exists $m > 0$ such that $A^m > 0$

Corollary: a positive matrix is primitive

A Markov chain with primitive transition matrix is said ergodic

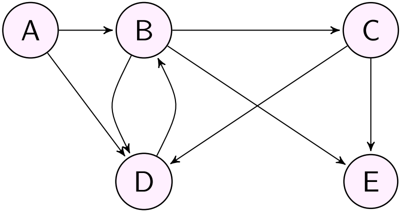
Where are we now?

Everything ok if we had a positive stochastic matrix

- Perron theorem: there exists a unique left Perron vector, corresponding to the greatest eigenvalue, equal to 1
- Convergence condition: the left Perron vector can be computed by the power method

But. . .

Let A be the matrix of the web graph.



$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

It has some drawbacks:

1. A has elements equal to 0
2. A cannot even be made stochastic: rows with some 1 can be transformed to sum 1 by scaling its values by the number of 1s, but there may exist rows with no 1 (dangling nodes)

Getting a stochastic matrix

We modify A to obtain a new stochastic positive matrix.
Null rows, corresponding to dangling nodes are modified from

$$[0, 0, \ldots, 0]$$

to

$$\left[\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right]$$

A uniform teleportation probability to any node is introduced.

$$P = \begin{pmatrix} 0 & .5 & 0 & .5 & 0 \\ 0 & 0 & .33 & .33 & .33 \\ 0 & 0 & 0 & .5 & .5 \\ 0 & 1 & 0 & 0 & 0 \\ .2 & .2 & .2 & .2 & .2 \end{pmatrix}$$

Getting a positive matrix

The application of the idea is extended: a teleportation probability is introduced for all nodes.

This can be done introducing a teleportation matrix T

$$T = \frac{1}{n} ee^T = \begin{pmatrix} 1/n & 1/n & \cdots & 1/n \\ 1/n & 1/n & \cdots & 1/n \\ \vdots & \vdots & \ddots & \vdots \\ 1/n & 1/n & \cdots & 1/n \end{pmatrix}$$

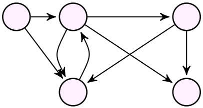
with $e^T = [1, 1, \dots, 1]$

A linear combination of A and T is then performed

$$H = \alpha P + (1 - \alpha) T$$

α is the damping factor

Getting a positive matrix



Let $\alpha = .8$, then

$$H = \begin{pmatrix} 0 & .4 & 0 & .4 & 0 \\ 0 & 0 & .266 & .266 & .266 \\ 0 & 0 & 0 & .4 & .4 \\ 0 & .8 & 0 & 0 & 0 \\ .16 & .16 & .16 & .16 & .216 \end{pmatrix} + \begin{pmatrix} .04 & .04 & .04 & .04 & .04 \\ .04 & .04 & .04 & .04 & .04 \\ .04 & .04 & .04 & .04 & .04 \\ .04 & .04 & .04 & .04 & .04 \\ .04 & .04 & .04 & .04 & .04 \end{pmatrix}$$

$$= \begin{pmatrix} .04 & .44 & .04 & .44 & .04 \\ .04 & .04 & .306 & .306 & .306 \\ .04 & .04 & .04 & .44 & .44 \\ .04 & .84 & .04 & .04 & .04 \\ .2 & .2 & .2 & .2 & .2 \end{pmatrix}$$

In terms of Markov chain

According to H the random surfer, at each node, chooses the next node as follows:

- if the current node v_i is dangling, apply teleporting: the next node is chosen with uniform probability $1/n$
- otherwise, flip a α-biased coin.
  - with probability α, follow an outlink chosen with uniform probability $1/o_i$, where o_i is the number of outlinks of v_i
  - with probability $1 - \alpha$, apply teleporting: the next node is chosen with uniform probability $1/n$

Computing Pagerank

$$H = \begin{pmatrix} .04 & .44 & .04 & .44 & .04 \\ .04 & .04 & .306 & .306 & .306 \\ .04 & .04 & .04 & .44 & .44 \\ .04 & .84 & .04 & .04 & .04 \\ .2 & .2 & .2 & .2 & .2 \end{pmatrix} \qquad H^2 = \begin{pmatrix} .0464 & .4144 & .1634 & .1954 & .1794 \\ .0888 & .03496 & .0995 & .2379 & .2219 \\ .1104 & .4784 & .121 & .153 & .137 \\ .0464 & .0944 & .2698 & .3018 & .2858 \\ .072 & .312 & .1252 & .2852 & .2052 \end{pmatrix}$$

$$H^4 = \begin{pmatrix} .079 & .3167 & .1438 & .2428 & .2153 \\ .0732 & .2984 & .1533 & .2509 & .2207 \\ .0779 & .3281 & .1387 & .2391 & .2144 \\ .0749 & .299 & .1668 & .2454 & .2111 \\ .0729 & .2897 & .1606 & .252 & .2229 \end{pmatrix} \qquad H^{256} = \begin{pmatrix} .0641 & .259 & .133 & .212 & .186 \\ .0641 & .259 & .133 & .212 & .186 \\ .0641 & .259 & .133 & .212 & .186 \\ .0641 & .259 & .133 & .212 & .186 \\ .0641 & .259 & .133 & .212 & .186 \end{pmatrix}$$

The resulting pagerank vector is then

$$[.0641, .259, .133, .212, .186]$$

Efficiency and sparsity

- H is a dense matrix
- this is bad in terms of efficiency
- but observe that

$$
\begin{aligned}
H &= \alpha P + (1-\alpha)\frac{1}{n}ee^T \\
&= \alpha(A + \frac{1}{n}de^T) + (1-\alpha)\frac{1}{n}ee^T \\
&= \alpha A + (\alpha d + (1-\alpha)e)\frac{1}{n}e^T
\end{aligned}
$$

where $d \in \{0,1\}^n$ has $d_i = 1$ if v_i is a dangling node and $d_i = 0$ otherwise.

Efficiency and sparsity

One step of the power method

$$\begin{aligned}\pi^{(k+1)} &= \pi^{(k)}H \\ &= \alpha\pi^{(k)}P + \frac{1-\alpha}{n}\pi^{(k)}ee^T \\ &= \alpha\pi^{(k)}A + (\alpha\pi^{(k)}d + 1 - \alpha)e^T\end{aligned}$$

- $\pi^{(k)}H$ is the product of an n-dimensional vector with a very sparse $n \times n$ matrix (this may require $O(n)$ steps)
- $\pi^{(k)}d = \sum_{v_i \text{dangling}} \pi_i^{(k)}$ clearly requires $O(n)$ steps

Convergence

Question: how fast (how may iterations) does the power method converge to the stationary distribution?

- A matrix $A \in \mathbb{R}^{n \times n}$ has n independent unitary (left) eigenvectors $u_1, \ldots, u_n$
- $u_1, \ldots, u_n$ form a basis of $\mathbb{R}^n$, then $\pi^{(0)} = \sum_{i=1}^{n} a_i u_i$ for suitable reals $a_1, \ldots, a_n$

Convergence

- let $\lambda_1, \ldots \lambda_n$ be the left eigenvalues of A (assume $|\lambda_1| \geq \ldots \geq |\lambda_n|$)
- then, since for any eigenvector u_i, $u_i A^k = u_i A A^{k-1} = \lambda_i u_i A^{k-1} = \lambda_i^k u_i$

$$\begin{aligned}
\pi^{(0)} A^k &= \left(\sum_{i=1}^{n} a_i u_i \right) A^k = \sum_{i=1}^{n} a_i u_i A^k \\
&= \sum_{i=1}^{n} a_i u_i \lambda_i^k = a_1 \lambda_1^k \left(u_1 + \sum_{i=2}^{n} \frac{a_i}{a_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k u_i \right)
\end{aligned}$$

Convergence

Then,

- $\pi^{(0)}A^k \rightarrow a_1\lambda_1^k u_1$
- the difference

$$|a_1\lambda_1^k u_1 - \pi^{(0)}A^k| = \left| a_1\lambda_1 \sum_{i=2}^{n} \frac{a_i}{a_1} \left(\frac{\lambda_i}{\lambda_1}\right)^k u_i \right|$$

  goes to 0 as k increases
- the slowest decreasing term is the largest one λ_2/λ_1
- since in our case $\lambda_1 = 1$, the convergence rate is determined by λ_2
- smaller λ_2: faster convergence

Convergence

In the case of the Google matrix

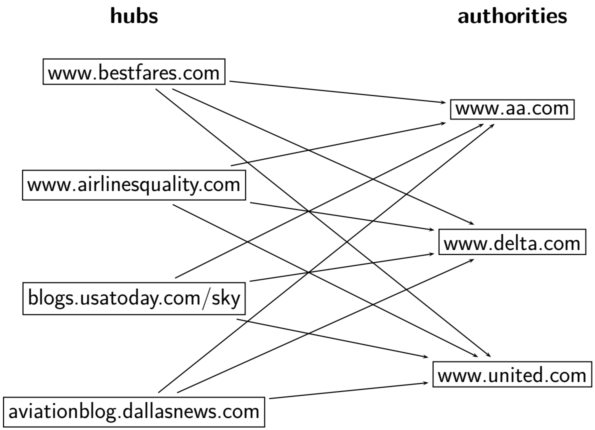
$$H = \alpha P + (1 - \alpha) T$$

it is possible to prove that $\lambda_2 = \alpha$

Hubs and authorities: Definition

- 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.
- Circular definition – we will turn this into an iterative computation.

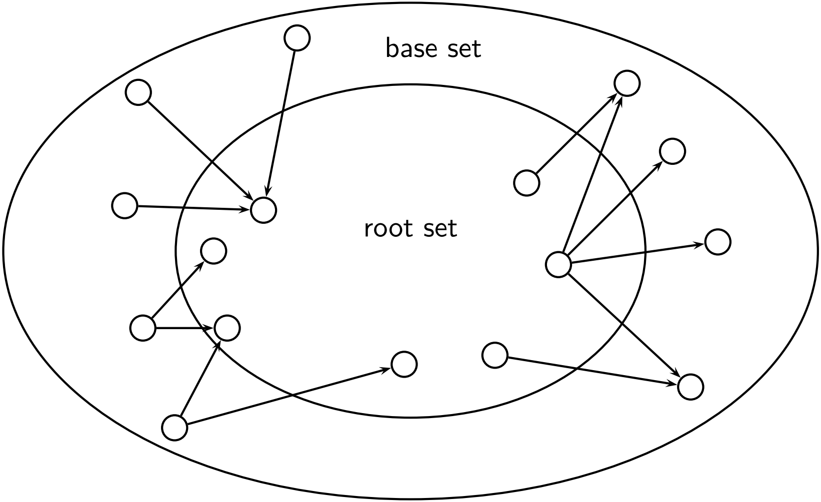
Example for hubs and authorities



How to compute hub and authority scores

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

Root set and base set (1)



The root set Nodes that root set nodes link to Nodes that link to root set nodes The base set

Root set and base set (2)

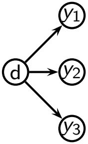
- Root set typically has 200–1000 nodes.
- Base set may have up to 5000 nodes.
- Computation of base set, as shown on previous slide:
    - Follow outlinks by parsing the pages in the root set
    - Find d's inlinks by searching for all pages containing a link to d

Hub and authority scores

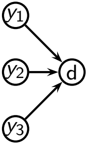
- Compute for each page d in the base set a hub score $h(d)$ and an authority score $a(d)$
- Initialization: for all d: $h(d) = 1$, $a(d) = 1$
- Iteratively update all $h(d), a(d)$
- After convergence:
    - Output pages with highest h scores as top hubs
    - Output pages with highest a scores as top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

Iterative update

- For all d: $h(d) = \sum_{d \mapsto y} a(y)$



- For all d: $a(d) = \sum_{y \mapsto d} h(y)$



- Iterate these two steps until convergence

Details

- 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.
- In most cases, the algorithm converges after a few iterations.

Authorities for query [Chicago Bulls]

0.85	www.nba.com/bulls
0.25	www.essex1.com/people/jmiller/bulls.htm “da Bulls”
0.20	www.nando.net/SportServer/basketball/nba/chi.html “The Chicago Bulls”
0.15	users.aol.com/rynocub/bulls.htm “The Chicago Bulls Home Page”
0.13	www.geocities.com/Colosseum/6095 “Chicago Bulls”

(Ben-Shaul et al, WWW8)

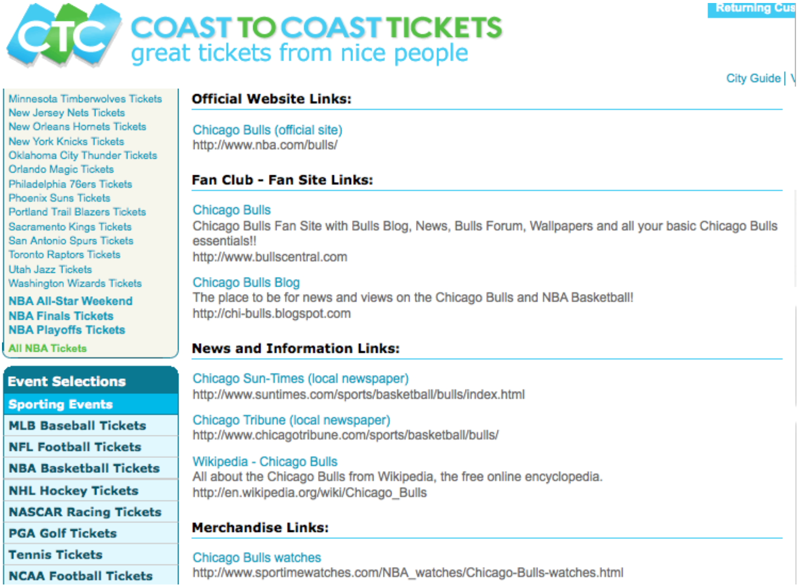
The authority page for [Chicago Bulls]

Hubs for query [Chicago Bulls]

1.62	www.geocities.com/Colosseum/1778 "Unbelieveabulls!!!!!"
1.24	www.webring.org/cgi-bin/webring?ring=chbulls "Erin's Chicago Bulls Page"
0.74	www.geocities.com/Hollywood/Lot/3330/Bulls.html "Chicago Bulls"
0.52	www.nobull.net/web_position/kw-search-15-M2.htm "Excite Search Results: bulls"
0.52	www.halcyon.com/wordsltd/bball/bulls.htm "Chicago Bulls Links"

(Ben-Shaul et al, WWW8)

A hub page for [Chicago Bulls]

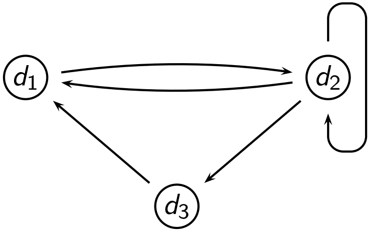


Hubs & Authorities: Comments

- HITS can pull together good pages regardless of page content.
- Once the base set is assembled, we only do link analysis, no text matching.
- Pages in the base set often do not contain any of the query words.
- In theory, an English query can retrieve Japanese-language pages!
  - If supported by the link structure between English and Japanese pages
- Danger: topic drift – the pages found by following links may not be related to the original query.

Proof of convergence

- We define an $N \times N$ adjacency matrix A. (We called this the link matrix earlier.
- 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 ($A_{ij} = 0$).
- Example:



	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 $\vec{h} = (h_1, \ldots, h_N)$ as the vector of hub scores. h_i is the hub score of page d_i.
- Similarly for $\vec{a}$, the vector of authority scores
- Now we can write $h(d) = \sum_{d \mapsto y} a(y)$ as a matrix operation: $\vec{h} = A\vec{a}$ ...
- ...and we can write $a(d) = \sum_{y \mapsto d} h(y)$ as $\vec{a} = A^T\vec{h}$
- HITS algorithm in matrix notation:
  - Compute $\vec{h} = A\vec{a}$
  - Compute $\vec{a} = A^T\vec{h}$
  - Iterate until convergence

HITS as eigenvector problem

- HITS algorithm in matrix notation. Iterate:
  - Compute $\vec{h} = A\vec{a}$
  - Compute $\vec{a} = A^T\vec{h}$
- By substitution we get: $\vec{h} = AA^T\vec{h}$ and $\vec{a} = A^TA\vec{a}$
- Thus, $\vec{h}$ is an eigenvector of AA^T and $\vec{a}$ is an eigenvector of A^TA.
- So the HITS algorithm is actually a special case of the power method and hub and authority scores are eigenvector values.
- HITS and PageRank both formalize link analysis as eigenvector problems.

Raw matrix A for HITS

$$
\begin{array}{c|ccccccc}
 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\
\hline
d_0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
d_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
d_2 & 1 & 0 & 1 & 2 & 0 & 0 & 0 \\
d_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
d_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
d_5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
d_6 & 0 & 0 & 0 & 2 & 1 & 0 & 1
\end{array}
$$

Hub vectors $h_0, \vec{h}_i = \frac{1}{d_i} A \cdot \vec{a}_i, i \geq 1$

	$\vec{h}_0$	$\vec{h}_1$	$\vec{h}_2$	$\vec{h}_3$	$\vec{h}_4$	$\vec{h}_5$
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 $\vec{a}_i = \frac{1}{c_i} A^T \cdot \vec{h}_{i-1}, i \geq 1$

	$\vec{a}_1$	$\vec{a}_2$	$\vec{a}_3$	$\vec{a}_4$	$\vec{a}_5$	$\vec{a}_6$	$\vec{a}_7$
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

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 hub score: d_6 (close: d_2)
- Pages with highest authority score: d_3

PageRank vs. HITS: Discussion

- 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.