Web Data Mining

Lecture 8: PageRank and HITS

Jaroslav Kuchař & Milan Dojčinovski
jaroslav.kuchar@fit.cvut.cz, milan.dojchinovski@fit.cvut.cz

Czech Technical University in Prague - Faculty of Information Technologies - Software and Web Engineering

EVROPSKÁ UNIE

Summer semester 2019/2020
Humla v0.3

Overview

- Web Structure Mining
- PageRank
- HITS

Web Structure Mining (Recall)

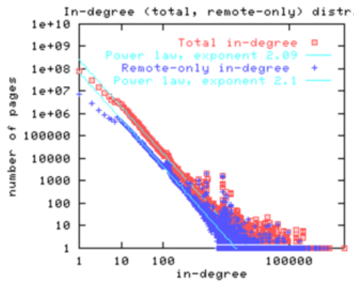
- Main ideas
  - *Use graph theory to analyze the node and connection structure of a web site.*
  - *Help the users to retrieve the relevant documents by analyzing the link structure of the Web.*
- Tasks
  - *Hyperlink analysis*
    → *Intra-page vs Inter-page.*
  - *Analysis of the tree-like structure of page structures*
- Applications
  - *Document retrieval and ranking*
  - *Discovery of hubs and authorities*
  - *Discovery of web communities*
  - *Citation networks*
  - *Social network analysis*
  - *Search engines, SEO, ...*

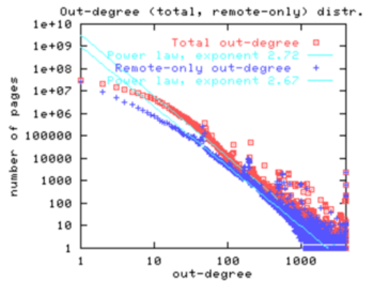
Web Graph

- Terminology
  - *Web graph*
    → *a directed graph representing the web*
  - *Node*
    → *web page in the graph*
  - *Edge*
    → *hyperlink on the web page*
  - *In-links (backlinks)*
    → *links pointing to the node*
  - *Out-Links*
    → *links generated from the node*
  - *In-Degree*
    → *number of links pointing to the node*
  - *Out-Degree*
    → *number of links generated from the node*

Web Graph Analysis

- Web graph statistics
  - *A. Broder et al., Graph structure in the web, 2000 - they analyzed the web graph consisting of 200 million pages and 1.5 billion links from AltaVista.*
- In (Out) Degree - Power law
  - *the probability that a node has in(out)-degree i is proportional to $\frac{1}{i^x}$, where x=2.1.*



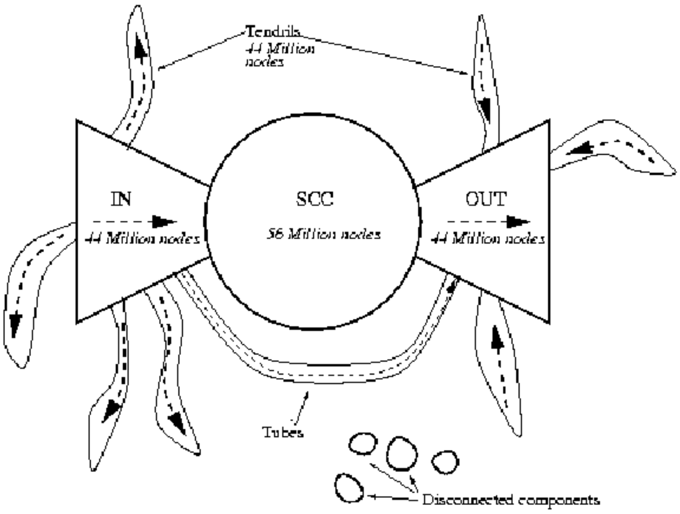


- Web graph
  - *Bow Tie Structure*
- Applications
  - *Important for designs and implementations of main crawlers, search engines, etc.*

The Bow-Tie Structure

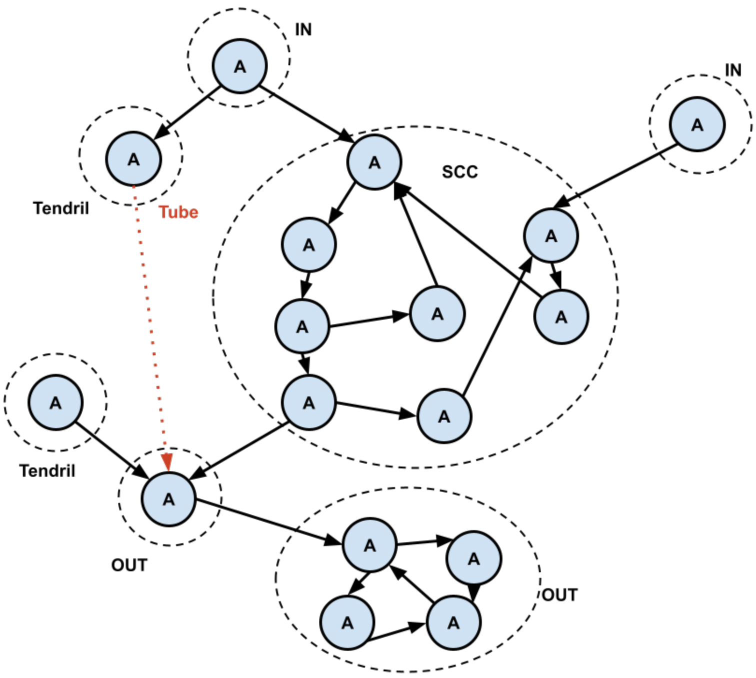
- Presents the connectivity of the web
  - *Web isn't the fully interconnected network*
- Components
  - *SCC - giant strongly connected component*
    - → *central core, all of whose pages can reach one another along directed links*
  - *IN*
    - → *pages that can reach the SCC, but cannot be reached from it.*
      - → *e.g. new pages not yet discovered*
  - *OUT*
    - → *pages that are accessible from the SCC, but do not link back to it.*
      - → *e.g. corporate pages with internal links only*
  - *Tendrils*
    - → *pages reachable from IN but cannot reach the SCC*
      - → *e.g. single page or document with no out-links*
    - → *pages that can reach the OUT but cannot be reached from the SCC*
  - *Tubes*
    - → *TENDRILS that fulfills both assumptions*
      - → *e.g. a single page linking only a blog post about a company that links to the pages with internal links*
  - *Disconnected*

The Bow-Tie Structure (cont.)



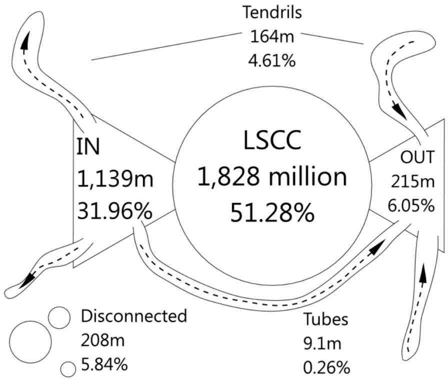
- "the chance of being able to surf between two randomly chosen pages is less than one in four"
  - A. Broder, R. Kumar, F. Maghoul, P. Ragha-van, S. Rajagopalan, S. Stata, A. Tomkins, and J.Wiener. Graph structure in the web. Computer Net-works, 33:309–320, June 2000

The Bow-Tie Structure Example



The Bow-Tie Structure Revisited

- Power law exponent (2000 vs 2012): 2.1 vs 2.24
- Average degree: 7.5 vs 36.8
- SCC: 27.7% vs 51.3%
- IN, OUT: 21%,21% vs 31%,6%
- Pairs of connected pages: 25% vs 48%



- Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, and Christian Bizer. 2014. Graph structure in the web — revisited: a trick of the heavy tail. In Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion). ACM, New York, NY, USA, 427-432.

Application: Improving Search Results

- Web Search
  - *Can build on top of existing boolean and vector models from Information Retrieval.*
  - *Vector based model was used in AltaVista.*
- Issues of basic IR models
  - *Results are too large that the user can explore.*
  - *All documents are treated equally according to the relevance point of view.*
  - *Results are returned only using the text based matching approaches.*
  - *Heavily influenced by many spam techniques*
    → e.g. keyword stuffing
- Need for other relevance/popularity scores
  - *Web structure is the most well known source of additional information about popularity of web pages.*

Overview

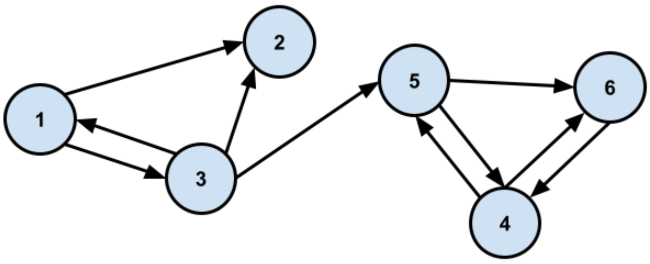
- Web Structure Mining
- PageRank
- HITS

PageRank

- Introduced in April, 1998 at WWW98 by Sergey Brin and Larry Page in a paper titled "The anatomy of a large-scale hypertextual Web search engine."
  - *Uses link structure as an indicator of an individual page's quality.*
  - *The prestige of a page is proportional to the sum of the prestige scores of pages linking to it.*
  - *Prestige is independent of any information need or query.*
- Main formula
  - $\pi^{(k+1)T} = \pi^{(k)T}(\alpha S + (1 - \alpha)E)$
- Characteristics
  - *ability to fight spam, global measure and is query independent, computed off-line, very efficient at the query time.*

PageRank Computation

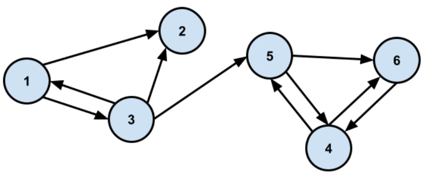
- Main idea
  - *If a web page is pointed to by other, important pages, then it's also an important page.*
  - *Think as kind of "fluid" that circulates through networks.*
- PageRank for one page
  - $r(P_i) = \sum_{P_j \in B_{P_i}} \frac{r(P_j)}{|P_j|}$
    - $\rightarrow B_{P_i}$ *- set of pages linking to* P_i
    - $\rightarrow |P_j|$ *- number of outinks from* P_j
- Examples:
  - $r(P_1) = \frac{r(P_3)}{3}$, $r(P_2) = \frac{r(P_1)}{2} + \frac{r(P_3)}{3}$



Iterative computation of the PageRank

- Next iteration (k+1) uses states from the previous one (k)
  - $r_{k+1}(P_i) = \sum_{P_j \in B_{P_i}} \frac{r_k(P_j)}{|P_j|}$
- PageRank is initialised with a predefined value
  - $\forall i : r_0(P_i) = \frac{1}{n}$

Node	Iteration 0	Iteration 1	Iteration 2	Order (after 2nd iteration)
P1	$\frac{1}{6}$	$\frac{1}{18}$	$\frac{1}{12} \times \frac{1}{3} = \frac{1}{36}$	5.
P2	$\frac{1}{6}$	…	$\frac{1}{18}$	4.
P3	$\frac{1}{6}$	$\frac{1}{6} \times \frac{1}{2} = \frac{1}{12}$	$\frac{1}{36}$	5.
…	…	…	…	…
P6	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{14}{72}$	2.

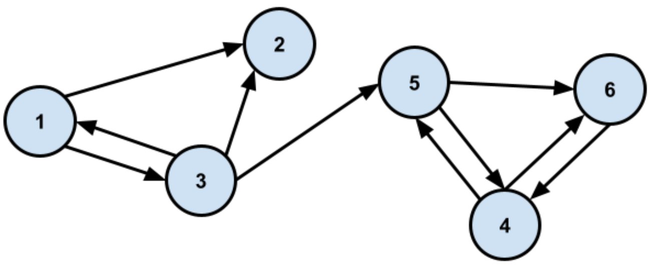


Matrix Representation

- Mathematically
  - *a system of n linear equations with n unknown variables.*
  - *Can be represented as a matrix.*
- PageRank vector
  - $\pi = (r_0(P_1), r_0(P_2), \ldots, r_0(P_n))$
- Use matrix H ($n \times n$)
  - $H_{ij} = \frac{1}{|P_i|}$ *if there is a link from* P_i *to* P_j
  - $H_{ij} = 0$ *otherwise*
- Circular definition, where the iterative algorithm is used to solve
  - $\pi^{(k+1)} = \pi^{(k)} H$
  - *The equation is the characteristic equation used for finding the eigensystem of the matrix.*
  - π *is an eigenvector with the corresponding eigenvalue of 1.*
  - *1 is the largest eigenvalue and the PageRank vector P is the principal eigenvector*
  - *Also called power method*
- Issues:
  - *the Web graph does not meet all conditions*
    - → *There are many pages without any out-links, as well as directed paths leading into a cycle, ...*

Matrix Representation (cont.)

	P1	P2	P3	P4	P5	P6
P1	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
P2	0	0	0	0	0	0
P3	$\frac{1}{3}$	$\frac{1}{3}$	0	0	$\frac{1}{3}$	0
P4	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
P5	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$
P6	0	0	0	1	0	0



Iterative computation using Matrix

- Using the equation:
  - $\pi^{(k+1)} = \pi^{(k)} H$
- $\pi^{(0)} = \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right)$

	P1	P2	P3	P4	P5	P6
P1	0	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
P2	0	0	0	0	0	0
P3	$\frac{1}{3}$	$\frac{1}{3}$	0	0	$\frac{1}{3}$	0
P4	0	0	0	0	$\frac{1}{2}$	$\frac{1}{2}$
P5	0	0	0	$\frac{1}{2}$	0	$\frac{1}{2}$
P6	0	0	0	1	0	0

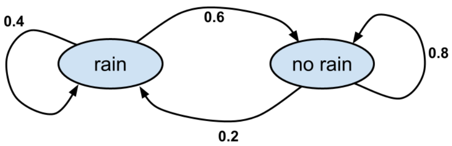
- $\pi^{(1)} = \pi^{(0)} H = \left(\frac{1}{18}, \frac{5}{36}, \frac{1}{12}, \frac{1}{4}, \frac{5}{36}, \frac{1}{6}\right)$

Matrix Representation and Computation

- Complexity
  - *Every iteration requires $O(n^2)$*
  - *Multiplication of PageRank vector of size n and matrix of size $n \times n$*
- The matrix is sparse
  - *Most of the elements are zero*
  - *Efficient memory representations using LIL (List of List), CSR (Compressed Sparse Row) or CSC (Compressed Sparse Column), ...*
  - *There are many efficient algorithms for sparse matrix multiplication with complexity O(nnz), where nnz is number of non-zero elements.*
- The matrix is close to the stochastic (transition) matrix of probabilities in Markov chain models.
  - *Fulfills the "memorylessness" Markov property*
    - *→ If one can make predictions for the future without knowing history.*
  - *Except dangling pages - pages that have no out-links!*

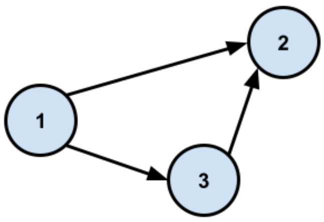
Markov Chains

- Markov Chains
  - *named after Andrey Markov*
  - *mathematical systems that hop from one state to another*
  - *special type of stochastic model*
    - → *the simplest from Markov models*
    - → *the future state depends only on the present state and not on the history*
- Example
  - *Weather*
    - → *raining today*
      - → *40% rain tomorrow*
      - → *60% no rain tomorrow*
    - → *not raining today*
      - → *20% rain tomorrow*
      - → *80% no rain tomorrow*
  - $P = \begin{bmatrix} 0.4 & 0.6 \\ 0.2 & 0.8 \end{bmatrix}$



Issues of the Matrix Representation

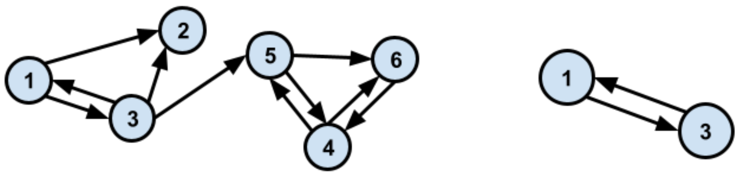
- Rank sinks
  - *pages that have no out-links*
  - *it does not distribute the PageRank to others*
  - *continuously decrease the overall PageRank in the graph*



- Example
  - $\pi^{(0)} = (1/3, 1/3, 1/3)$
  - $(1/3, 1/3, 1/3) \times \begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = (0, 1/2, 1/6) = \pi^{(1)}$
  - $(0, 1/2, 1/6) \times \begin{bmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = (0, 1/6, 0) = \pi^{(2)}$
  - $(0, 0, 0) = \pi^{(3)}$

Issues of the Matrix Representation (cont.)

- Link farms
  - *group of pages that link to every other page in the group*
  - *a link farm is a clique*
  - *they support each other*
- Cycles
  - *cause oscillation of the PageRank between them*



- Example
  - $\pi^{(0)} = (0, 1)$
  - $(0, 1) \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = (1, 0) = \pi^{(1)}$
  - $(1, 0) \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = (0, 1) = \pi^{(2)}$
  - ...

Alternative PageRank Definition

- Random walk/random surfer
  - *Someone who is randomly browsing a network.*
  - *Choosing a page at random, picking each page with equal probability.*
  - *Follow links for a sequence of k steps.*
    - → *In each step, they pick a random out-going link from their current page, and follow it to where it leads.*
- Randomly following links is called a random walk
- Claim
  - *The probability of being at a page X after k steps of this random walk is precisely the PageRank of X after k applications of the Basic PageRank Update Rule.*
- Issue
  - *Rank sink and cycles*
- Solution
  - *Teleportation to a random node*

Transition Probability Matrix

- Stochasticity adjustment of matrix H to matrix S
  - *Update of the dangling node row*
    - *→ setting all the zeros to 1/n*
  - *Random teleport/jump*
- $S = H + a(\frac{1}{n}e^T)$
  - *a is a vector of length n*
    - → $a_i = 1$ *if there is no outlink from* P_i
    - → $a_i = 0$ *otherwise*
  - $e^T = (1, 1, 1, 1, 1, 1)$

$$\begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} = S$$

Transition Probability Matrix (cont.)

- S matrix is stochastic
  - *sum of values in row is equal to 1*
  - *non-negative and square*
- Transition matrix for a finite Markov chain
  - *Probability of using the link for the random walk*
- Issue
  - *It is not irreducible*
    - *→ Web graph is strongly connected*
    - *→ for each pair of nodes, there is a path from one to another one*
  - *It is not aperiodic*
    - *→ periodic - all paths leading from one node back to that node*
  - *Convergence issue!*

$$S = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Google Matrix

- Solution for irreducible and aperiodic situation
  - *Primitivity adjustment*
  - *We add a link from each page to every page and give each link a small transition probability controlled by a parameter* damping factor d *(e.g. 0.85)*
- Updated model
  - *Random surfer has two options*
    - → *With probability d, he randomly chooses an out-link to follow.*
    - → *With probability 1-d, he jumps to a random page without a link.*
      - → *Surfer may get bored, or interrupted*
- Google matrix
  - *Becomes strongly connected*
    - → *link from each page to every page*
  - *Becomes aperiodic*
    - → *random surfer does not have to traverse a fixed cycle*
- $G = d \times S + (1-d)\frac{E}{n}$
  - *d is damping factor*
  - *E is* $e \times e^T$ *- is a* $n \times n$ *square matrix of all 1*

Google Matrix (cont.)

- $G = d \times S + (1-d)\frac{E}{n}$
  - $d = 0.9$

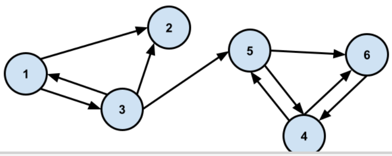
$$S = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

- $G = 0.9 \times S + 0.1\frac{E}{6}$

$$G = \begin{bmatrix} 1/60 & 7/15 & 7/15 & 1/60 & 1/60 & 1/60 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 19/60 & 19/60 & 1/60 & 1/60 & 19/60 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 7/15 & 7/15 \\ 1/60 & 1/60 & 1/60 & 7/15 & 1/60 & 7/15 \\ 1/60 & 1/60 & 1/60 & 11/12 & 1/60 & 1/60 \end{bmatrix}$$

PageRank Computation

- Power Iteration Method
  - $\pi^{(k+1)} = \pi^{(k)} G$
- Google matrix
  - *Stochastic, Irreducible, Aperiodic, Primitive*
  - *No-zero elements*
    - → *It is not sparse any more!*
- Computation
  - *Complexity* $O(n^2)$
- Example
  - *50 iterations*
  - $\pi = (0.03721, 0.05396, 0.04151, 0.3751, 0.206, 0.2862)$
  - *Order: 4, 6, 5, 2, 3, 1*

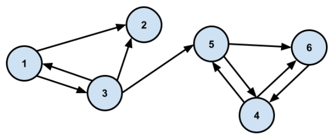


PageRank Computation (cont)

- Convert to operations with sparse matrix
  - $\pi^{(k+1)} = \pi^{(k)} G$
  - $\pi^{(k+1)} = d\pi^{(k)} H + (d\pi^{(k)} a + 1 - d)\frac{e^T}{n}$
- The most computational intensive operation
  - *multiplication of vector and matrix uses sparse matrix H*
- Convergence criteria
  - *1-norm*
    - → *the iteration ends after the 1-norm of the residual vector is less than a pre-specified threshold* δ
    - → *1-norm for a vector is simply the sum of all the components*
  - *page-order*
    - → *no significant change of the page order between iterations*
  - *usually around 50*

PageRank Example

- Example 1
  - *iterations: 50*
  - *damping factor: 1.0*
    - $\rightarrow$ *following links*
  - $\pi = (7.18e-10, 1.24e-09, 8.36e-10, 0.44, 0.22, 0.33)$
- Example 2
  - *iterations: 50*
  - *damping factor: 0.0*
    - $\rightarrow$ *random choosing*
  - $\pi = \left(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}\right)$
- Google
  - *damping factor* ≈ 0.85



PageRank Modifications

- Intelligent surfer
  - *Modification of probabilities in transition matrix*
    - $\rightarrow$ *Analysis of users behavior*
      - $\rightarrow$ *Using click logs, ...*
    - $\rightarrow$ *Similarities of pages*
      - $\rightarrow$ *Using cosine similarity*
    - $\rightarrow$ *Anchor text, or the surrounding information*
- Personalization
  - *Modification of the teleportation*
    - $\rightarrow G = d \times S + (1-d)\frac{E}{n}$
      - $\rightarrow$ *E is* $e \times e^T$ *- is a* $n \times n$ *square matrix of all 1*
    - $\rightarrow$ *Change* $e \times e^T$ *to* $e \times v^T$*, where* v^T *provides information about preferences for specific pages*

Overview

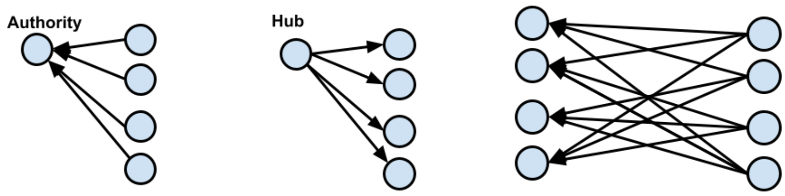
- Web Structure Mining
- PageRank
- HITS

HITS

- HITS
  - *Hypertext Induced Topic Search*
  - *Presented by Jon Kleinberg in January, 1998 at the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms.*
    - → *utilizes the web structure as important aspect*
    - → *a query is used to select a subgraph from the Web*
- Main characteristics
  - *Search query dependent*
  - *Two rankings*
    - → *authority ranking and hub ranking*
- Approach
  - *For a search query, HITS first expands the list of relevant pages returned by a search engine and then produces rankings of the expanded set of pages.*

Hubs and Authorities

- Hub
  - *page with many outlinks*
  - *page is a source of many important links to authority pages relevant for the topic*
- Authority
  - *a page with many inlinks*
  - *if people trust the page, they link to it at it becomes the authority*
- The goal
  - *Find best hubs and authorities*
    - *→ Good authorities are linked by good hubs*
    - *→ Good hubs link to good authorities*



HITS Algorithm

- Collecting pages
  - *HITS sends a query to a search engine and collects top t highest ranked pages that are relevant to the query (e.g. t=200)*
    - *→ Called root set W*
  - *Grows W by including pages that link to any page in W or are linked by any page from W. At most k per page. (e.g. k=50)*
    - *→ Called base set S (size 1000-5000)*
- Graph
  - *HITS works with the graph G(V,E) composed from all pages in the base set S.*
  - *L is the adjacency matrix of the graph G.*
- Scores
  - *Authority score*
    - → $a(i)^k = \sum_{(j,i) \in E} h(j)^{(k-1)}$
  - *Hub score*
    - → $h(i)^k = \sum_{(i,j) \in E} a(j)^{(k-1)}$

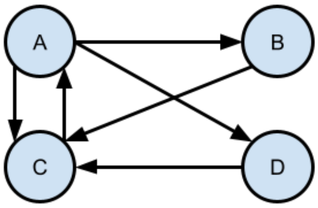
HITS Algorithm (cont.)

- Matrix representation
  - *Similar to PageRank*
    - $\rightarrow a = L^T h$
    - $\rightarrow h = La$
- Iterative computation
  - *using the power iteration method*
    - $\rightarrow a_k = L^T L a_{k-1}$
    - $\rightarrow h_k = L L^T h_{k-1}$
    - $\rightarrow a_0 = h_0 = (1, 1, 1, \ldots)$
  - *normalization*
    - $\rightarrow \sum_{i=1}^{n} a_i = 1$
    - $\rightarrow \sum_{i=1}^{n} h_i = 1$
  - *ends after the 1-norms of the residual vectors are less than some thresholds (e.g. 5 iteration)*
- Return top ranked pages as authorities and hubs.

HITS Algorithm (cont.)

- Convergence issues
  - *HITS will always converge*
  - *can provide different hub and authority vectors*
    - *→ depending on the initialization*
    - *→ caused by the problem that $L^T L$ (respectively LL^T) is reducible*
- Modification
  - *When pages are relevant to the query, but they can be separated in the graph G*
    - *→ e.g. words with different meaning*
  - *Compute HITS on smaller communities*
- Characteristics
  - *ability to rank pages according to the query topic*
    - *→ more relevant hubs and authorities*
  - *query time execution*
    - *→ time consuming operation*
  - *does not have the anti-spam capability*
    - *→ a simple page with many links can easily become a hub*
  - *topic drift*
    - *→ expanded pages are not relevant*

HITS Example



$$L = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, L^T = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, h_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, a_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$a_1 = L^T h_0 = \begin{bmatrix} 1 \\ 1 \\ 3 \\ 1 \end{bmatrix}, h_1 = L a_0 = \begin{bmatrix} 3 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$