COMP6237 – Pagerank

Markus Brede

Brede.Markus@gmail.com

Lecture slides available here:

http://users.ecs.soton.ac.uk/mb8/stats/datamining.html

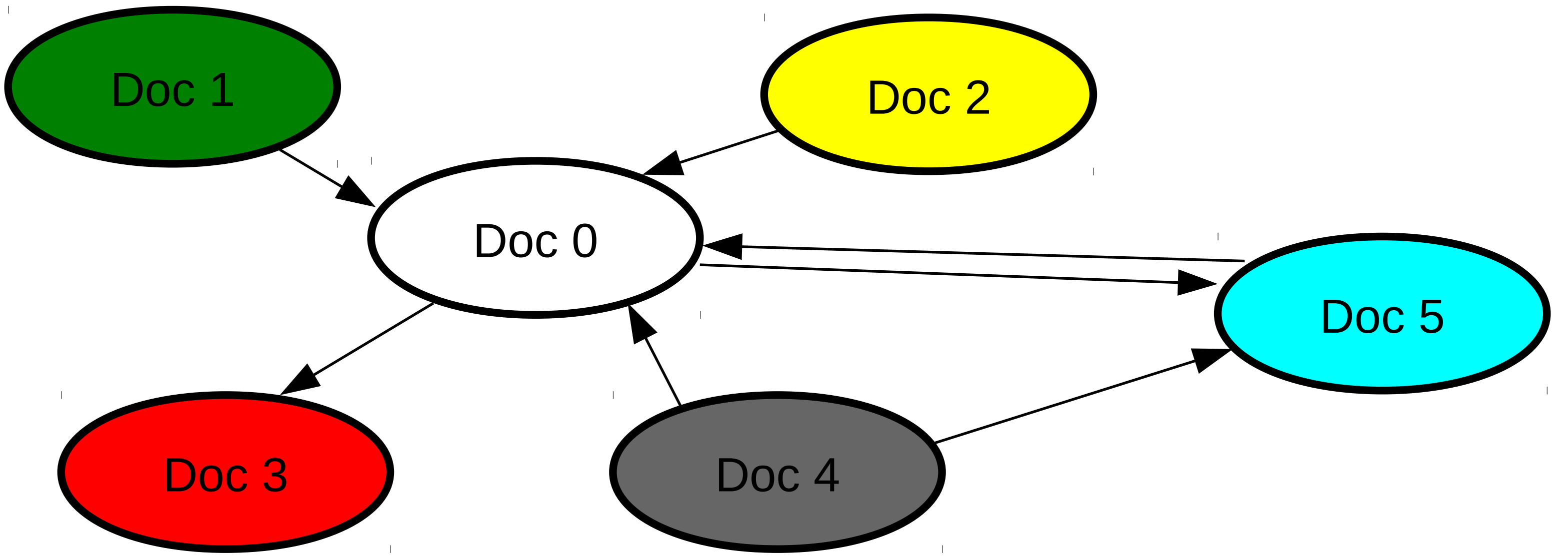
History

- Was developed by Larry Page (hence the name) and Sergey Brin
- First part of a research project about a new type of search engine. Started 1995, first prototype 1998.
- Shortly after Page and Brin founded Google …
- Work has been influenced by earlier work on citation analysis by Eugene Garfield in the 1950s
- At the same time as Page and Brin Kleinberg published a similar idea for web search, the HITS (Hyperlink-induced topic search) algorithm

Outline

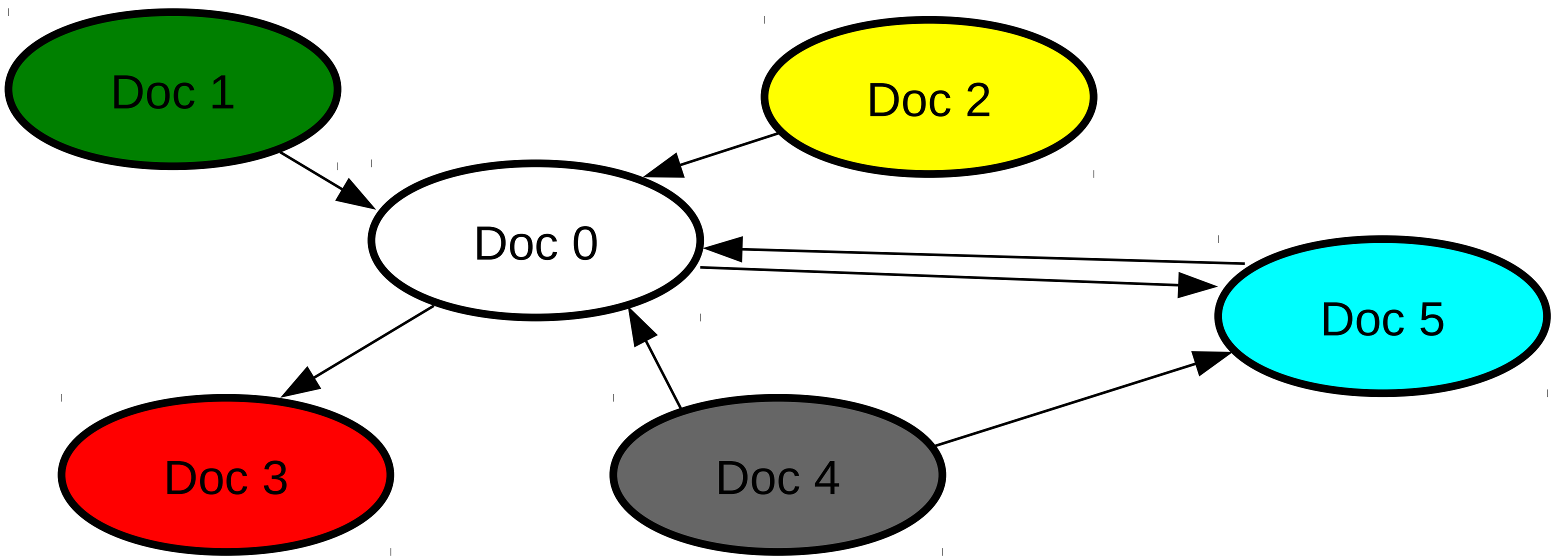
- Why?
  - Could use bag of words representation, cosine similarity and inverse document frequency weighting for search – works pretty well
  - There is often more information about documents
  - Web pages contain links to other pages. These reflect judgments about relevance – page rank aims to exploit this!
- Agenda:
  - Ideas to rank importance of webpages – “centrality measures”
    - Degree centrality, eigenvector centrality, Katz centrality, … pagerank
    - Page rank and random walks
    - Calculating page rank
    - Kleinberg's HITS algorithm
  - Summary

Main Idea



- Documents (web pages) refer to each other in some way
- Links are endorsements of relevance (i.e. if a links to b the creator of a thinks that b is relevant to the topic of a)
- Surely, pages with many incoming links are more relevant than such with less incoming links
- Want to exploit this link structure in a systematic way to **rank pages according to importance**, but when is a page/node important?!
- This is also useful for a lot of other data mining in social networks

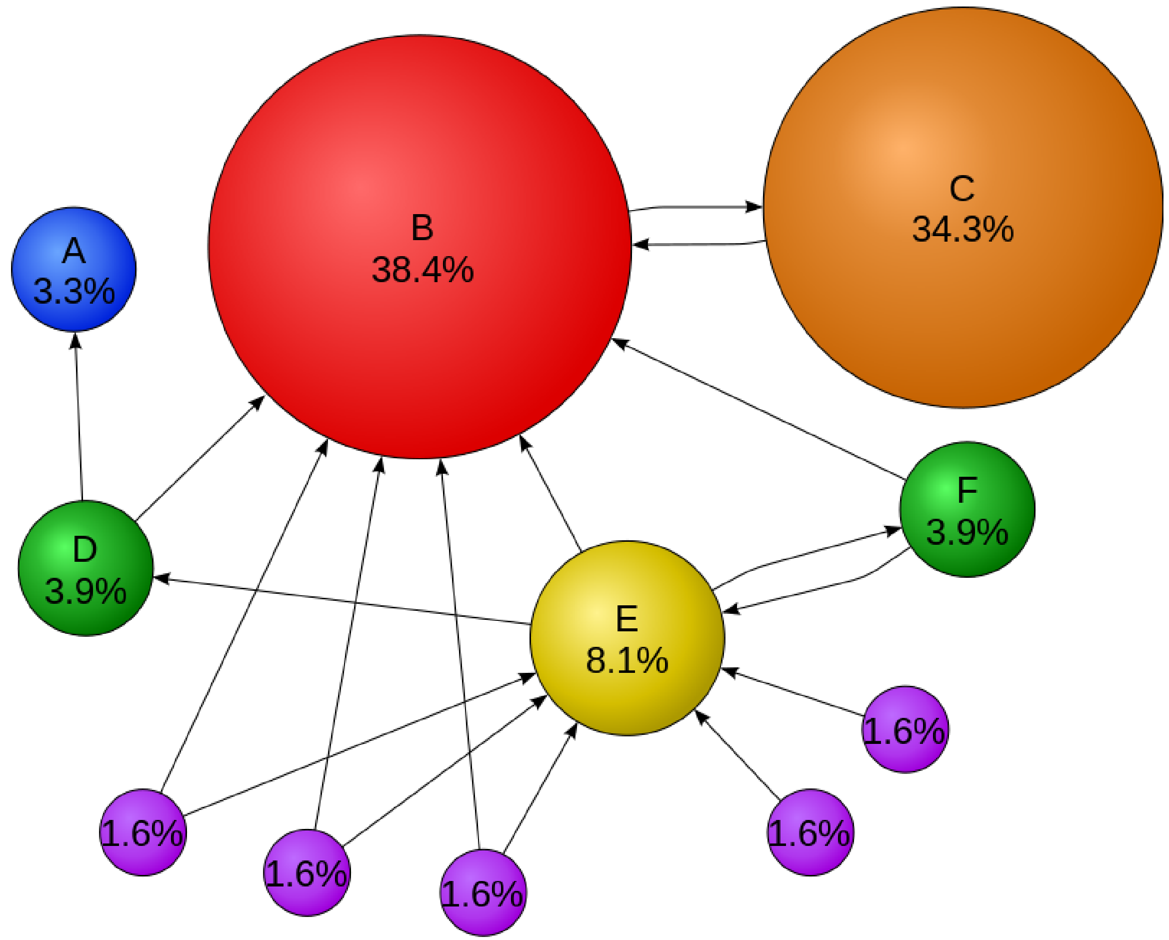
Reminder



- Suppose we have such a network, how to represent it on a computer?
  - Can label nodes by numbers 1 … n
  - Network is given by an adjacency matrix A, entries of which are 1 if there is a connection between the respective nodes and zero otherwise

Degree Centrality

- Simplest idea:
  - Importance of a page = number of incoming links (“in-degree”)
  - This is actually used quite often to evaluate scientific papers
    - Papers link to each other when they cite each other
    - In-degree = number of citations of a paper
- Advantage: very easy to calculate, e.g. $d_i = \sum_j a_{ji}$
- Problems:
  - A paper might be very important because it is cited by one very influential study (rather than by thousands of largely ignored low level papers)
  - Overlooks the global picture (a paper might be a very influential link between different disciplines, but not cited very much)



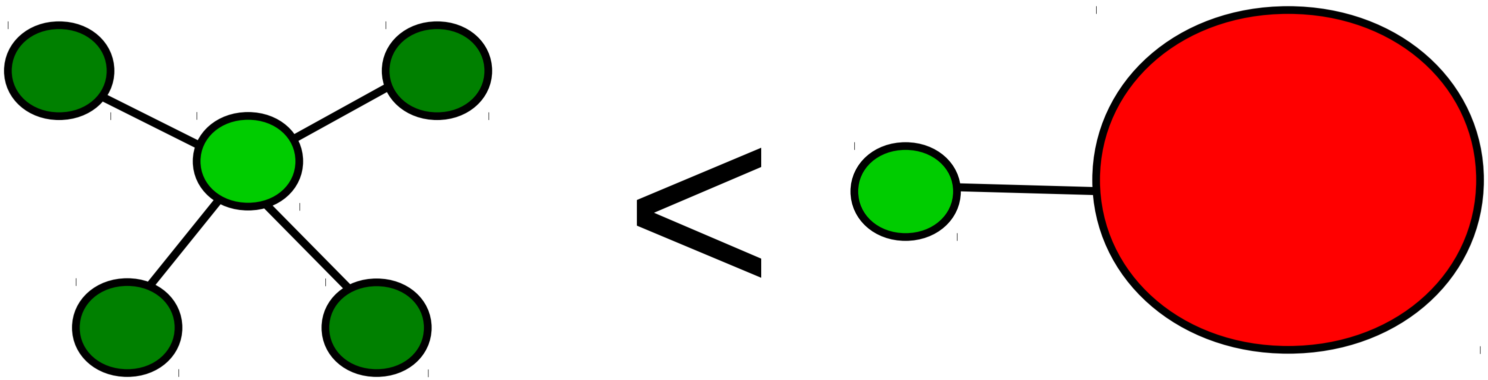
Link structure of web pages and a measure of their importance, page rank, discussed Later. Having many incoming links does not always mean a page is important.

Other Centrality Measures

- Quite a number of measures has been developed in network theory to overcome some of these problems of degree centrality, e.g.:
  - Closeness centrality
    - centrality of a node related to average graph distance to all other nodes on network
  - Betweenness centrality
    - Centrality of a node related to how many paths pass through the node if messages are passed along shortest paths between randomly selected source/target nodes
- Some of these are computationally quite expensive, so not straightforward to use for very large networks. What is used in web search nowadays builds on eigenvector centrality ...

Eigenvector Centrality

- Score “centrality points” for being connected to “important” nodes (Bonacich 1987)



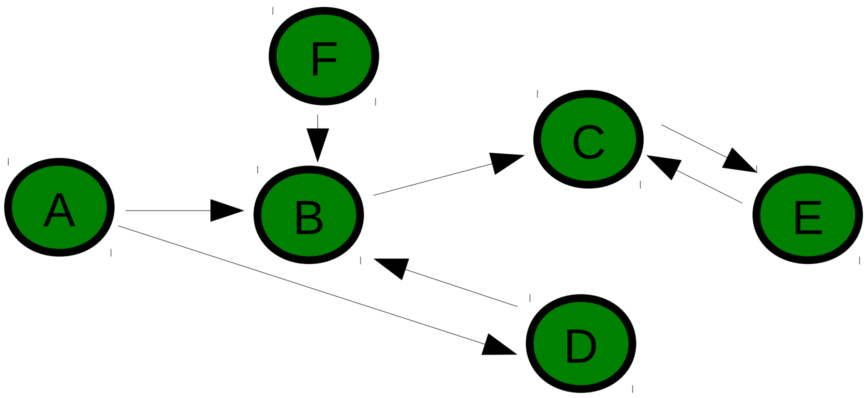
- Imagine experiment:
  - Assign all nodes importance 1.
  - Then update $x'_i = \sum_j a_{ij} x_j \longrightarrow x(t) = A^t x(0)$
  - Say $x(0) = \sum_i c_i v_i \longrightarrow x(t) = \sum_i c_i k_i^t v_i = k_1^t \sum_i c_i \left(\frac{k_i}{k_1}\right)^t v_i \rightarrow c_1 k_1^t v_1$

    Eigenvectors of {a}

  - EV centrality = eigenvector for largest eigenvalue of adjacency matrix

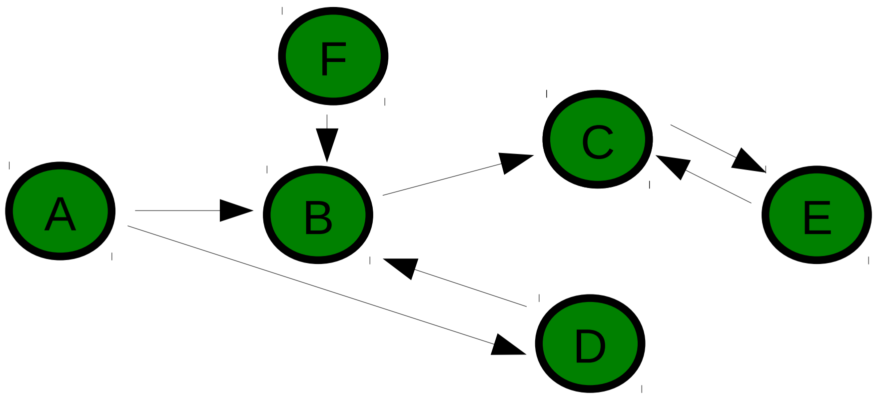
Eigenvector Centrality

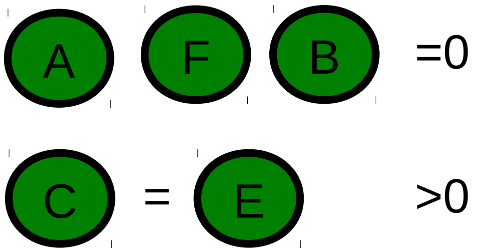
- Problems:
  - Normalisation?
  - Directed networks, left or **right** eigenvectors?
  - What about this?



Eigenvector Centrality

- Problems:
  - Normalisation? – We only care about rankings.
  - Directed networks, left or **right** eigenvectors?
  - What about this?





  - According to this definition only nodes in strong components or their out-components have centrality > 0!

Katz Centrality

- Katz (1953); every node gets some amount of centrality for "free" $x = \alpha A x + \beta \mathbf{1}$
- Re-arranging: $x = (I - \alpha A)^{-1} \mathbf{1}$
  - Where alpha balances relative importance of eigenvector component and "free" component
  - α should be between 0 and 1/kmax
  - In practice: better solve by iteration than by inverting the adjacency matrix
- Potential problem:
  - All nodes pointed to by a high centrality node receive high centrality! (i.e. if I am one of a million guys a big guy points to I become big myself ...)

Pagerank

- To overcome the problem of Katz centrality we could consider:

$$x_i = \alpha \sum_j a_{ij} x_j / k_j^{out} + \beta$$

- In matrix form: $x = \alpha A D^{-1} x + \beta 1$ with $D_{ii} = max(k_i^{out}, 1)$
- Conventionally β=1-α: $x = (I - \alpha A D^{-1})^{-1}(1-\alpha) = D(D - \alpha A)^{-1}(1-\alpha)$
- In principle this is what google uses with α=0.85
- Could give nodes different intrinsic importance β

$$\longrightarrow \quad x = D(D - \alpha A)^{-1} \beta$$

Pagerank and Random Walkers

- Imagine a random walker on a network
  - From each node one outgoing link is chosen at random to continue the walk
  - If there is no outgoing link the walk continues at a randomly chosen node
  - Let N(i,t) be the number of times page i is visited until time t
  - Then: $x_i = \lim_{t \to \infty} \frac{N(i,t)}{t}$
  - Can see this by writing down the transition matrix for the above Markov process, i.e. $P_{ij} = 1/k(i)_{out}$ for j linked to by I or 1/n if there is no outgoing link
  - Consider a vector v of probabilities of staying at node i, then: $v_{t+1} = P v_t$ (→ see previous slide!)

How to use Page Rank in Web Search?

- Simplest form:
  - Crawl links between pages to construct adjacency matrix
  - Calculate page rank once
  - Given a query Q, find all pages that contain all words in Q.
  - Return the page with the highest page rank among those (or the k pages with largest pagerank)
- Problems with this …
  - Pages are scored mainly on the basis of link structure. This can be exploited quite easily ...

"Link Farms"

- Collections of artificially created nonsensical pages that link to each other and acquire importance this way.
- Can than be used to boost importance of desired other pages.
- Not so easy to distinguish those from "real pages" like wiki pages
- Proprietary fine tuning by google ...

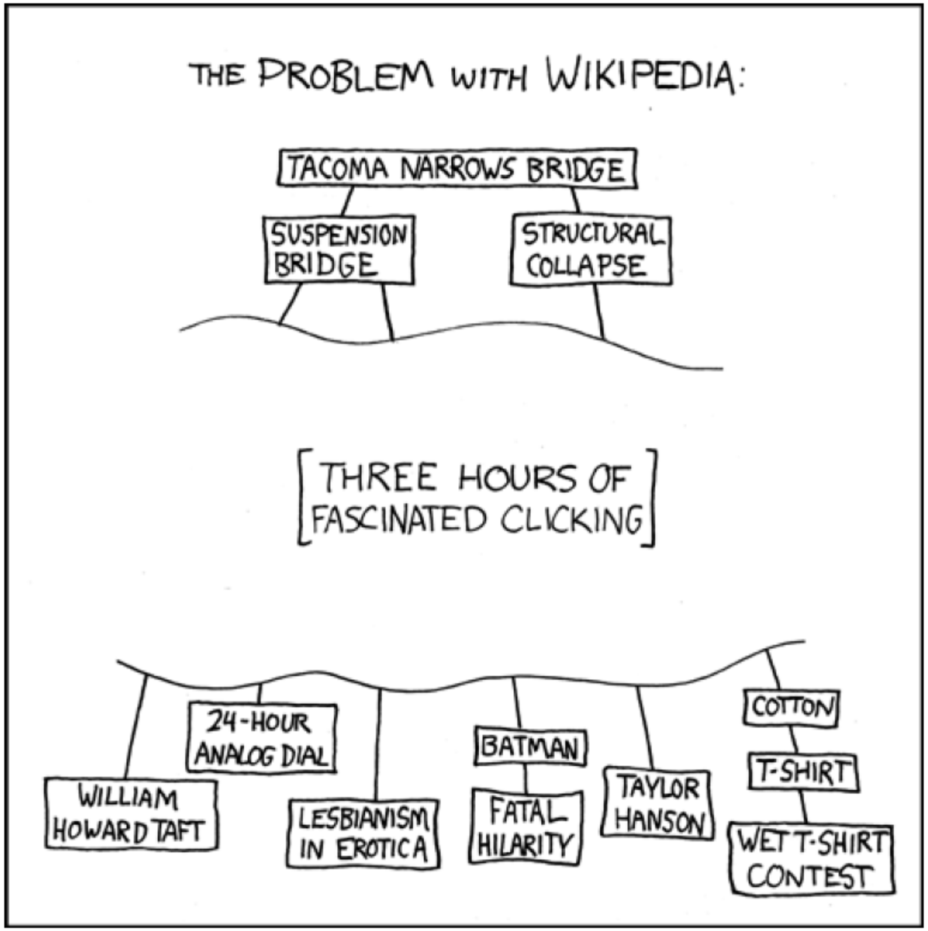
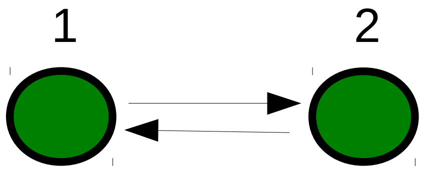


Figure 1: How do we distinguish this, automatically, from a link farm? (By Randall Munroe, http://xkcd.com/214/.)

Examples

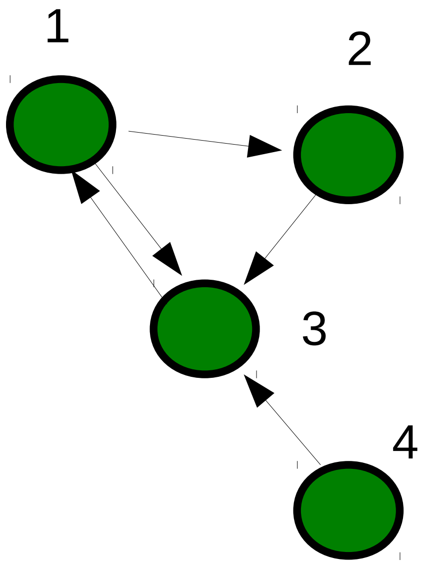
- What is the page rank of all nodes in the following situations?



$$x_1 = (1-\alpha) + \alpha\, x_2 / 1$$

$$x_2 = (1-\alpha) + \alpha\, x_1 / 1$$

$$\rightarrow \quad x_1 = x_2 = 1$$



$$x_1 = (1-\alpha) + \alpha\, x_3$$

$$x_2 = (1-\alpha) + \alpha\, x_1 / 2$$

$$x_3 = (1-\alpha) + \alpha\, (x_1 / 2 + x_2 + x_4)$$

$$x_4 = (1-\alpha)$$

$$\rightarrow$$

$$x_1 = 1.49$$

$$x_2 = 0.78$$

$$x_3 = 1.58$$

$$x_4 = 0.15$$

- More examples, see, e.g.:

http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm

Calculating Page Rank in Practice

- Equation for page rank defines a linear system of equations (which can be millions of equations for practical applications!)
  - Could solve those exactly, e.g. Gauss algorithm or similar
    - $O(n^3)$, i.e. maybe impractical
  - Could simulate a random walker on the network
    - Takes forever …
  - Best way is to solve system iteratively, i.e. guess a solution (say x=1) and then iterate

$$x_i = \alpha \sum_j a_{ij} x_j / k_j^{out} + (1-\alpha)$$

until convergence.

Calculating Page Rank in Practice

- Equation for page rank defines a linear system of equations (which can be millions of equations for practical applications!)
  - Could solve those exactly, e.g. Gauss algorithm or similar
    - $O(n^3)$, i.e. maybe impractical
  - Could simulate a random walker on the network
    - Takes forever …
  - Best way is to solve system iteratively, i.e. guess a solution (say x=1) and then iterate

    $$x_i = \alpha \sum_j a_{ij} x_j / k_j^{out} + (1-\alpha)$$

    until convergence.

Problems with Pagerank

- Good:
  - Robust to spam
  - Global measure
- Problems:
  - Favours older pages
  - Link farms
  - Buying links from high pagerank websites?

Kleinberg's HITS Algorithm

- Based on the idea that there are two kinds of useful web pages for broad topic search
  - Authoritative sources of information – **authorities** (e.g. a medical research institute)
  - Hand-compiled lists of authoritative sources – **hubs** (e.g. an association promoting health care)
- Basic properties:
  - Good hubs point to many good authorities
  - Good authorities are pointed to by many hubs
  - (But authorities will not necessarily be linked!)
- Idea:
  - Give each node two scores, a hub score (**h**) and an authority score (**a**)

HITS (2)

- Start with a set S of web pages (composed of most relevant pages for the search query, usually around 200 and those linked to by it), initially set a(v)=h(v)=1 for all members v of this set
- Consider the following iteration:

$$a_{t+1}(v) = \sum_{\text{y points to v}} h_t(y)$$ (authority update)

A page gets good authority if pointed to by many hubs.

$$h_{t+1}(v) = \sum_{\text{v points to y}} a_t(y)$$ (hub update)

A page is a good hub if pointing to many good authorities.

Normalise by respective square roots of sums of squares.

Problems of HITS

- Calculated on the fly, query time evaluation is slow
- Easily spammed – it is easy to create out-links on ones page
- Has problems with advertisements

Summary

- Idea: exploit link structure between documents as indications of relevance
- How to measure centrality
  - Eigenvector, Katz, Pagerank
- The HITS algorithm

- Original paper on HITS:

  http://www.cs.cornell.edu/home/kleinber/auth.pdf
- Original paper on pagerank:

  http://www-db.stanford.edu/~backrub/google.html