COMP38210
Workshop 5:
Web graph & ranking

John McNaught

The University of Manchester

Overview

- Cosine similarity based ranking
- Web as graph
- PageRank
- Combined ranking measures

(IIR book)

Reminder: Queries as vectors

- Key idea 1: Do the same for queries: represent them as vectors in the space
- Key idea 2: Rank documents according to their proximity to the query in this space
- proximity = similarity of vectors
- proximity ≈ inverse of distance
- Recall: We do this because we want to get away from the you're-either-in-or-out Boolean model.
- Instead: rank more relevant documents higher than less relevant documents

(IIR book)

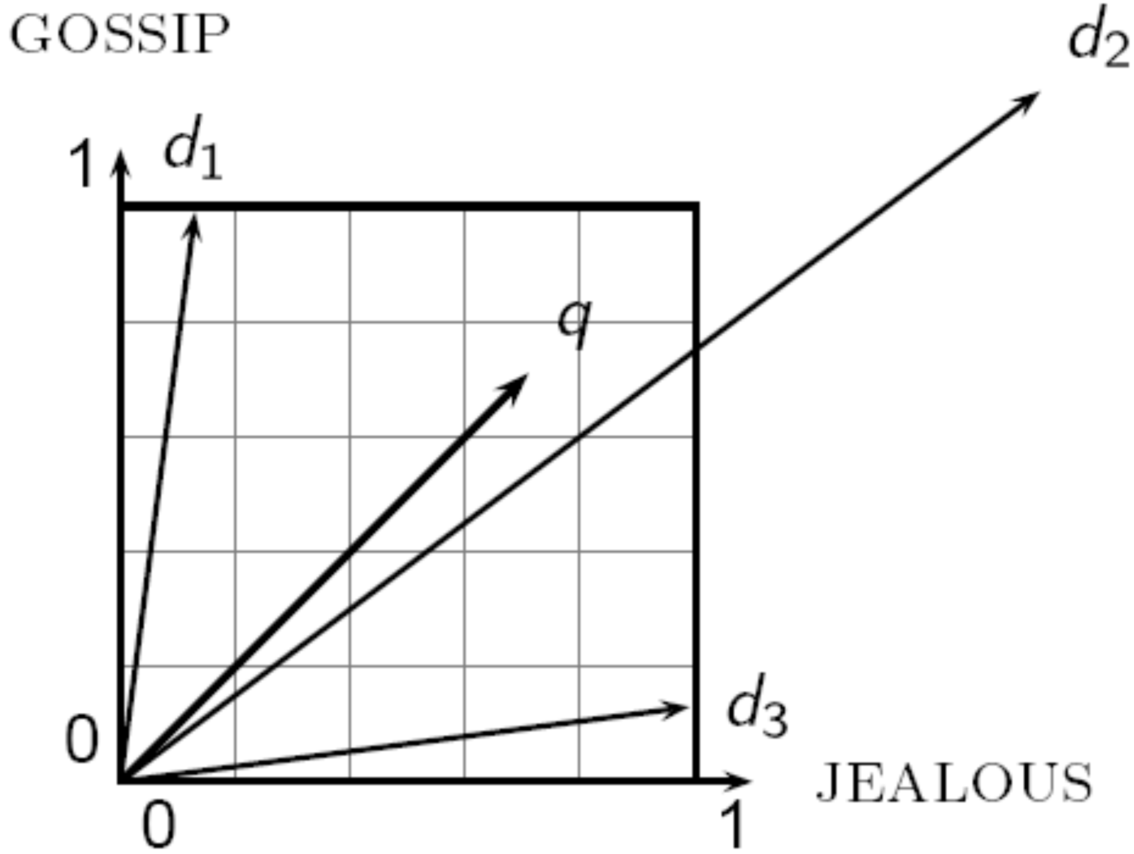
Formalizing vector space proximity

- First cut: distance between two points
  - (= distance between the end points of the two vectors)
- Euclidean distance?
- Euclidean distance is a bad idea . . .
- . . . because Euclidean distance is large for vectors of different lengths

(IIR book)

Why distance is a bad idea

The Euclidean distance between $\vec{q}$ and $\vec{d_2}$ is large even though the distribution of terms in the query $\vec{q}$ and the distribution of terms in the document $\vec{d_2}$ are very similar.



Use angle instead of distance

- Thought experiment: take a document *d* and append it to itself. Call this document *d′*
- "Semantically" d and d′ have same content
- Euclidean distance between the two documents can be quite large
- Angle between the two documents is 0, corresponding to maximal similarity
- Key idea: Rank documents according to angle with query
- We use cosine (higher value, greater similarity)

(IIR book)

Cosine for length-normalized vectors

- Need to normalise for length to ensure fair comparison
- Dividing a vector by its ***L_2 norm*** (see IIR book) makes it a unit (length) vector (on surface of unit hypersphere)

- For ***length-normalized vectors***, cosine similarity is simply the dot product (or scalar product):

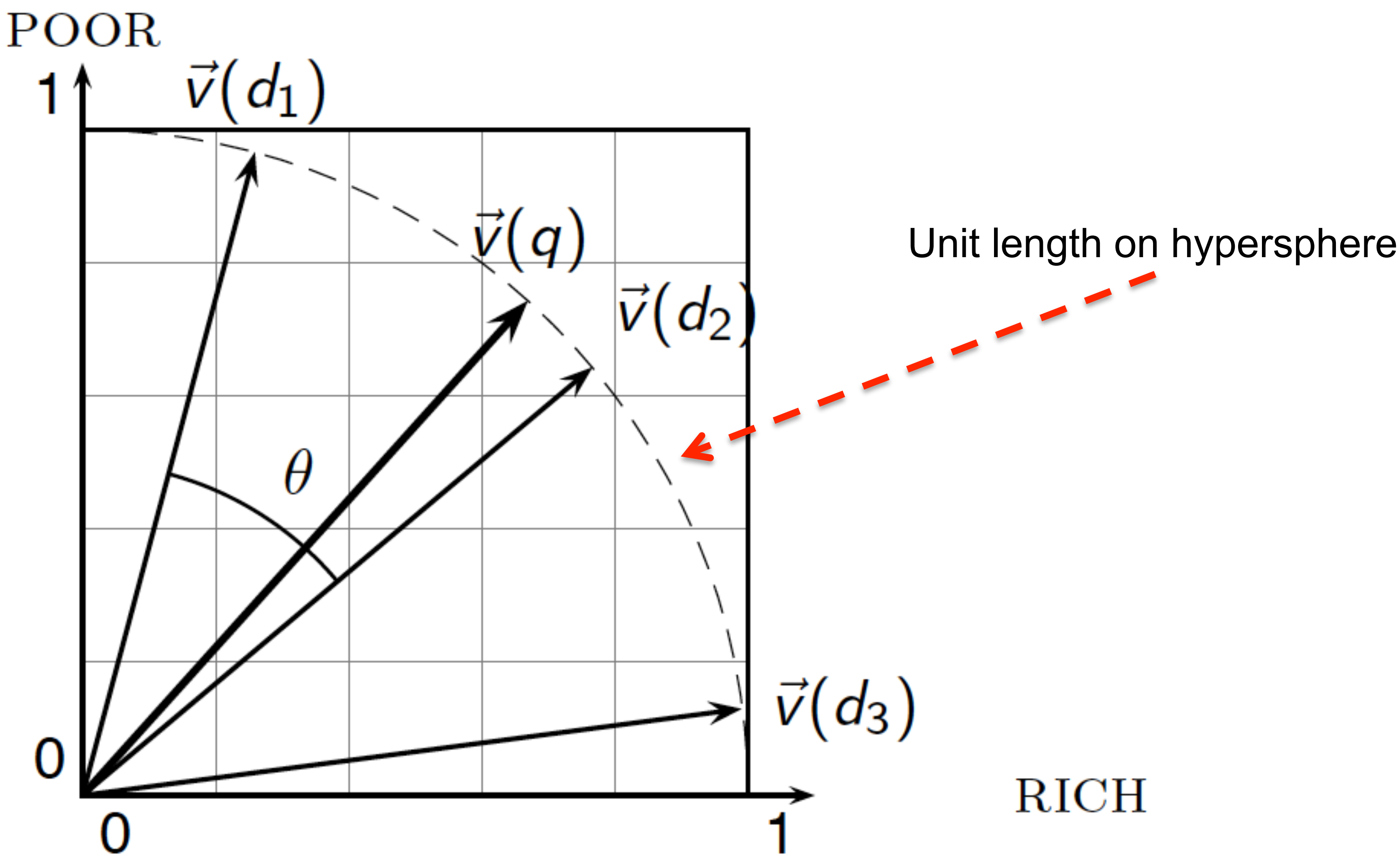
$$\cos(\vec{q},\vec{d}) = \vec{q} \bullet \vec{d} = \sum_{i=1}^{|V|} q_i d_i$$

for q, d length-normalized

The University of Manchester

(IIR book)

Cosine similarity illustrated



Using cosine similarity

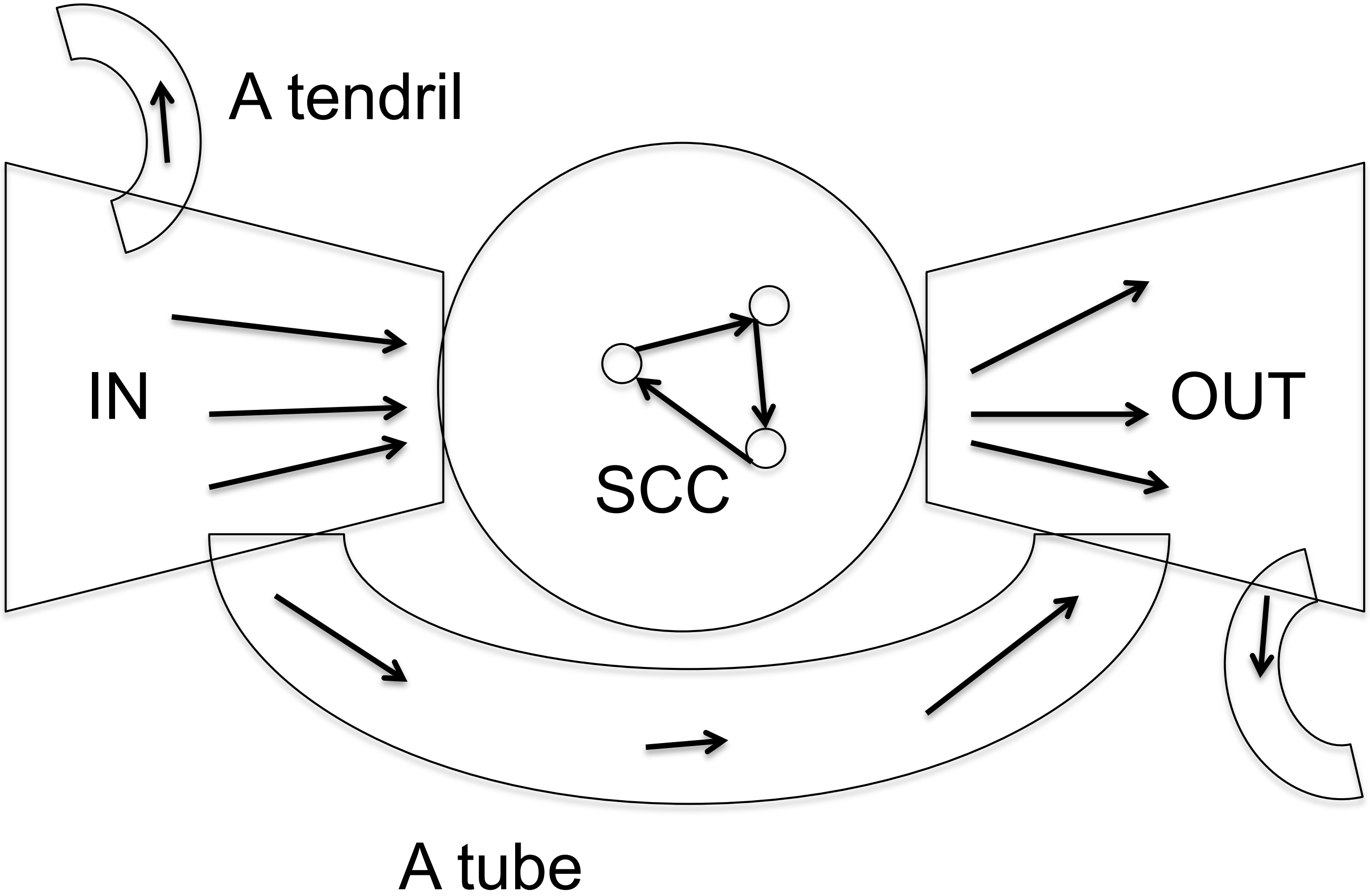
- Rank documents (their vectors)
  - Vector for Doc_i against vector for Doc_j
  - How similar are documents?

- Rank query against documents
  - Vector for query against vector for Doc_i
  - How similar is query to some document?

Web as directed graph

- Web page = vertex (node)
- Hyperlink = edge (arc)
- How can we use graph structure to help in ranking?
  - Measure “quality” or “importance” of web page in structure
- On what basis can we measure this?

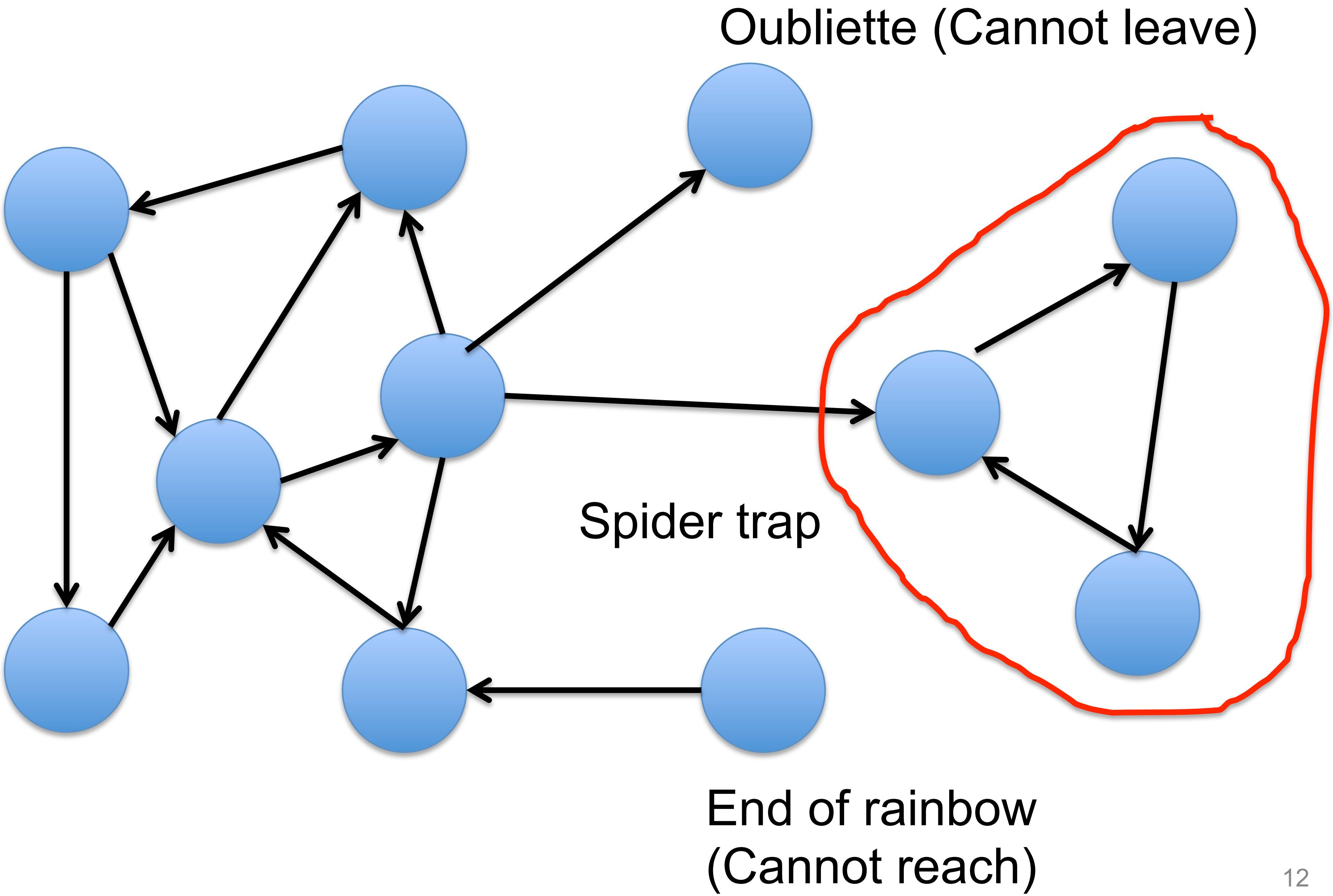
Bowtie structure of Web



Broder et al. (2000)

SCC=strongly connected component

Web graph walking



The University of Manchester

PageRank (Brin & Page, 1998)

- Calculates a score for each page
  - In practice, modulo internal pages
- Uses hyperlinked structure of Web
- Notion of random walk
  - Go from current page to a random one that the page links to
- Will visit some nodes more than others
  - Such nodes have many IN links from other frequently visited nodes
  - Pages visited more often are more important
- PageRank gives *probability distribution* over pages, i.e., likelihood random surfer will arrive at some page

The University of Manchester

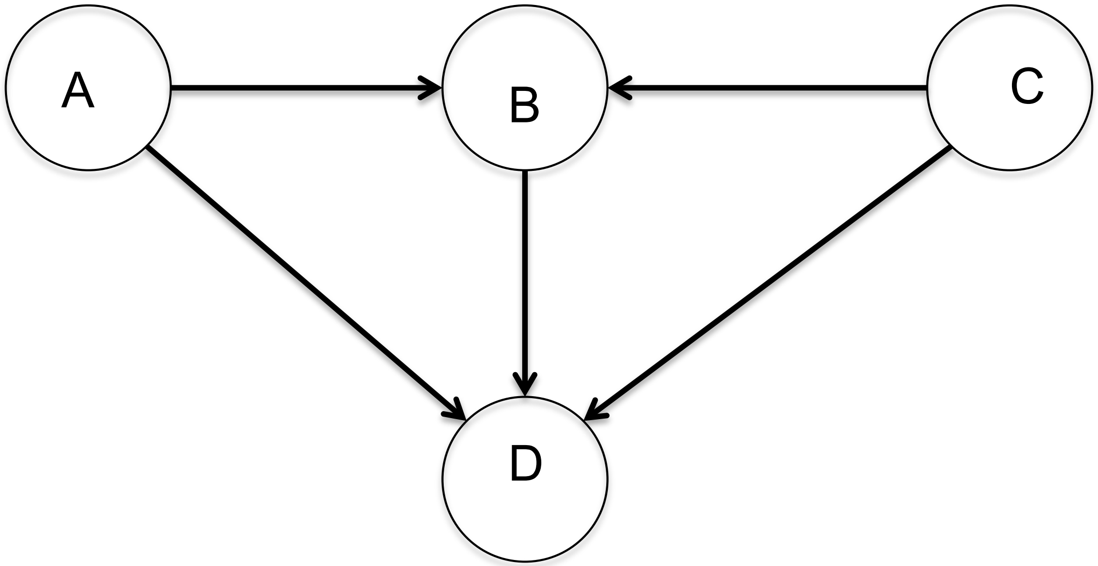
PageRank

- PageRank of page A recursively defined by PR of those pages that link to A
  - Link from B to A: a vote by B for A
  - BUT: vote weighted by analysis of page B’s PR
- Random surfer chooses 1 of
  - Random click on OUT link from page (85%)
  - Teleport (jump to unlinked page) (15%)

PageRank

- Iterative calculation until convergence
  - steady state
  - no great change in probabilities (threshold)
  - Surprisingly few iterations needed for Web
- Spreading of “probability mass” via OUT links
  - Each node sums up PR contributions from its IN link nodes and computes own PR score

Basic ideas



- Probability of surfer at A or C to reach B is 1/2
- Probability of reaching B if one link away is

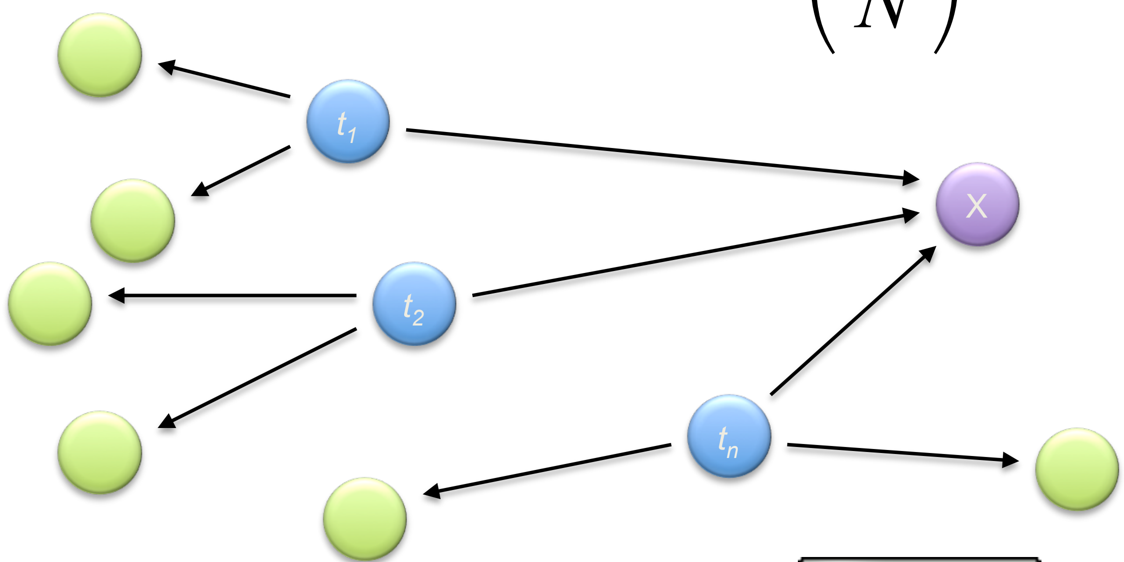
$$p(B) = 1/2p(A) + 1/2p(C)$$

PageRank: Defined

Given page x with inlinks $t_1 \ldots t_n$, where

- $C(t)$ is the out-degree of t
- α is probability of random jump (~15% of time)
- N is the total number of nodes in the graph

$$PR(x) = \alpha\left(\frac{1}{N}\right) + (1-\alpha)\sum_{i=1}^{n}\frac{PR(t_i)}{C(t_i)}$$



Formula breakdown

$$PR(x) = \boxed{\alpha\left(\frac{1}{N}\right)} + \boxed{(1-\alpha)}\boxed{\sum_{i=1}^{n}\frac{PR(t_i)}{C(t_i)}}$$

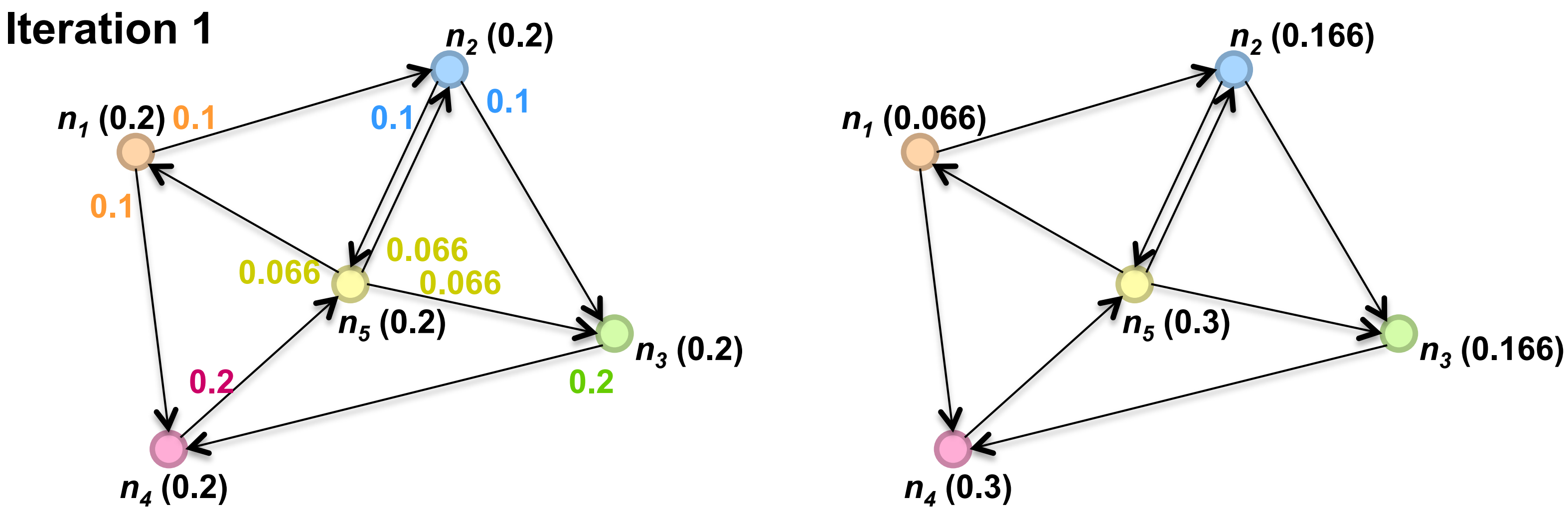
Probability of random jump

Damping factor (probability of normal click on link)

Sum up contributions of all pages contributing IN links to page X based on number of OUT links for each such page and its own PageRank score (RECURSION)

Sample PageRank Iteration 1 (no random jumps for simplicity)

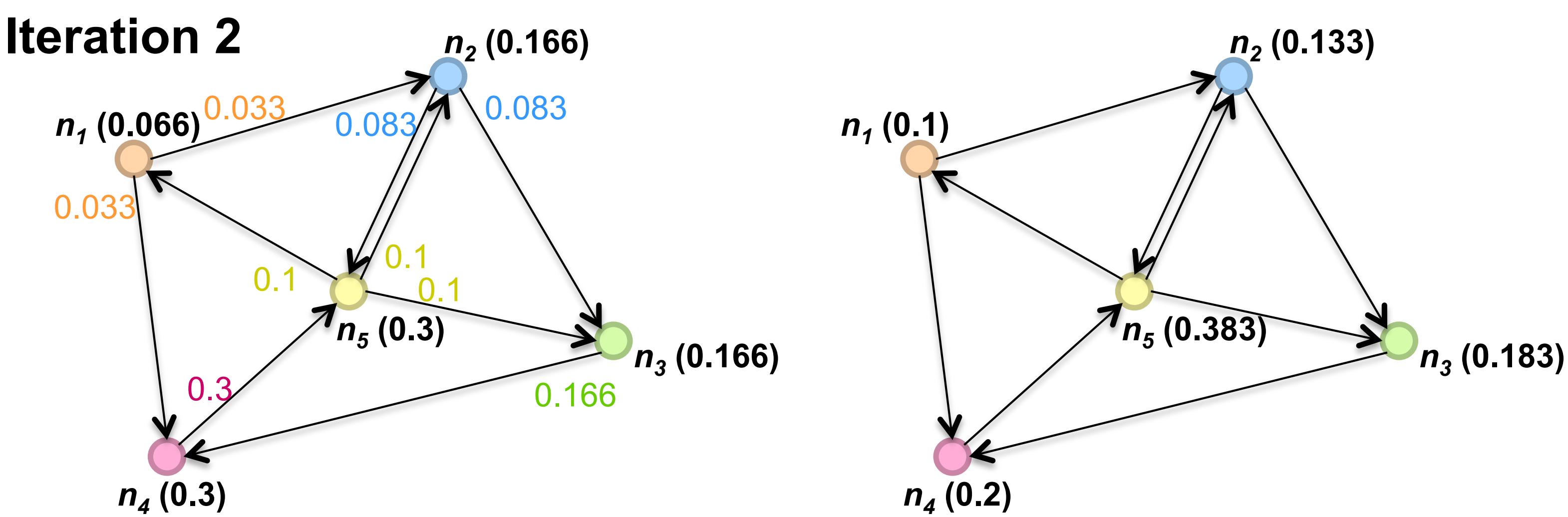
Start with uniform distribution across nodes.
At beginning of each iteration, PR values of all nodes sum to one



Send uniform partial PR contribution to each next neighbour

- Each node sums up PR contribution from neighbours
- May end up with less than gave away

Sample PageRank Iteration 2 (no random jumps)



Process repeats (until convergence, however defined)

Lost mass and random jumps

- Previous simple example ignores
  - Random jumps
  - Lost PageRank mass due to dangling nodes (no OUT links): distribute lost mass evenly over all nodes
- Full PageRank approach takes these into account

The University of Manchester

How PageRank is used

- Assigns global importance score to each page on Web
  - Independent of any query
- Upon query, can improve ranking of results by combining tf-idf score with PageRank score
  - weight(term, doc) = tf-idf(term,doc) x PR(doc)
- Typically combine with many other scores (e.g., cosine score)
- Google uses >200 measures to rank

The University of Manchester

Resources

- Chapter 6, section 6.2, Chapter 19, section 19.2 and Chapter 21 of Manning et al., Introduction to Information Retrieval
- Abiteboul et al. (2011) Web data management. (on syllabus) Chapter 13, Section13.4
- Broder et al. (2000)
  – http://www9.org/w9cdrom/160/160.html
- Brin & Page (1998)
  – http://infolab.stanford.edu/~backrub/google.html

Resources

- Worked examples with commentary
  - https://www.youtube.com/watch?v=4c3DAxQXzLI
- “PageRank explained with bright colours”
  - http://www.pagerank.dk/
- “How Google finds your needle in the Web's haystack”(a more mathematical explanation)
  - http://www.ams.org/samplings/feature-column/fcarc-pagerank
- Page ranking and search engines
  - https://www.youtube.com/watch?v=v7n7wZhHJj8
- How search engines treat data (indexes and tf-idf)
  - https://www.youtube.com/watch?v=vrjAIBgxm_w
- Demos:
  - https://learnforeverlearn.com/pagerank/
  - http://netlogoweb.org