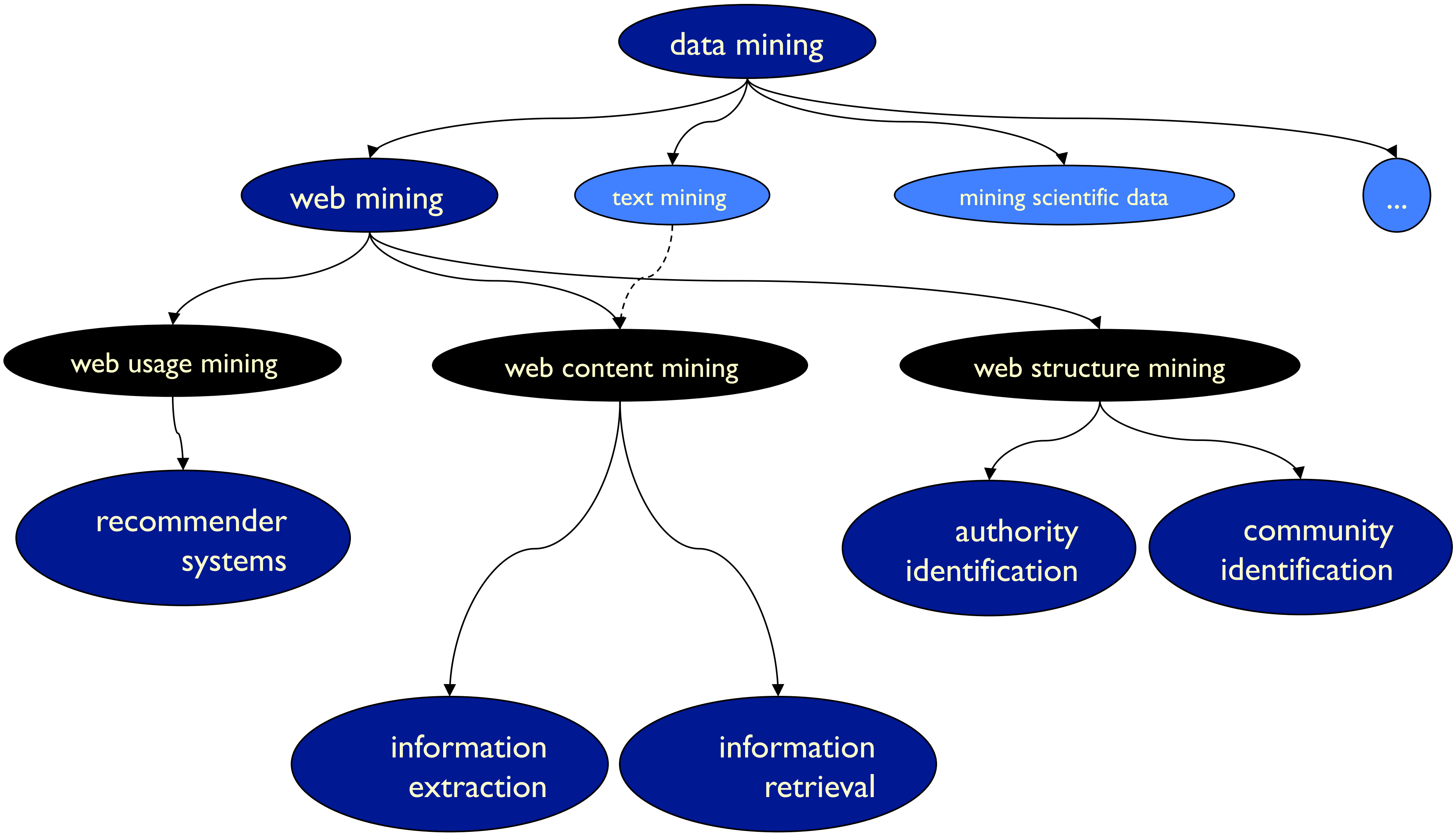
Web Mining: Structure: Link Analysis

Alípio Jorge, DCC-FC, Universidade do Porto
amjorge@fc.up.pt



amjorge@fc.up.pt

Web Mining

- Web Usage Mining
  - discovery from user access patterns from logs or alike
  - applications:
    - user segmentation, recommendation, personalization, adaptation, usability improvement

- Web Structure Mining
  - discovery of useful knowledge from hyperlinks
  - applications:
    - discover important pages (information retrieval)
    - discover communities

- Web Content Mining
  - extracts information from Web pages
  - applications
    - information extraction, summarization, topic extraction, discovering user emotions

amjorge@fc.up.pt

Web structure mining

- Take advantage of the information in web hyperlinks
  - links are created locally
  - web structure, as a whole, is not planned.
- Take advantage of the information in social links
  - social networks
- To understand the structure of the web
  - Link analysis
  - Analysis of the topology of connections

Web graphs

- Internet can be seen as different interdependent graphs
  - pages and hyperlinks (web)
  - computers and communications between them (internet)
- Web graph
  - very large (2×10^{10}?)
  - dynamical (changes structure and content)
  - has virtual parts (dynamic pages, harder to analyse)
  - disconnected (has islands)
  - sparse (relatively few connections)

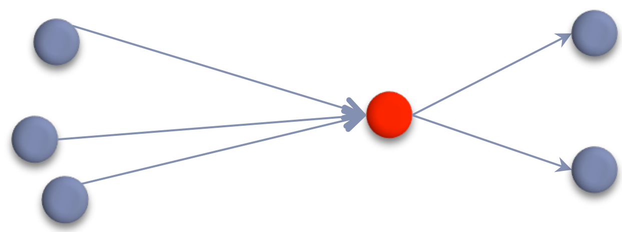
How can we use web structure?

- For finding
  - prestigious web pages
  - central links in social webs
  - communities
    - web page clusters pointing to each other
    - groups of people who change emails

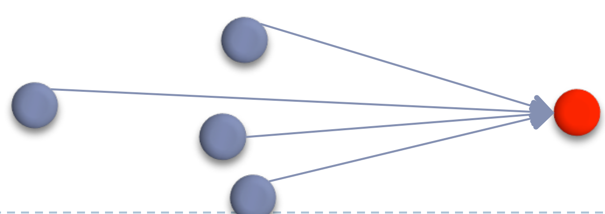
- Studying web structure is related to
  - social network analysis
    - e.g. package "sna" of R
  - complex networks

Network analysis

- Interesting phenomena in a network
  - central nodes
    - are important to connect two parts of the network
    - are involved in many indirect connections



  - prestigious nodes
    - tend to be referred to by many other nodes

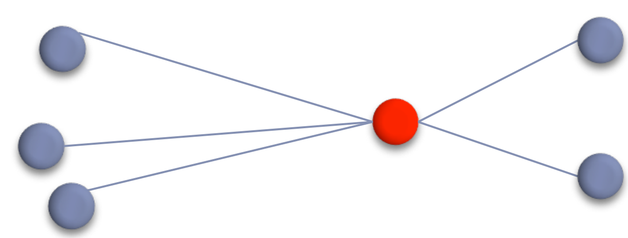


Network analysis: centrality measures

- Degree centrality of a node
  - network has *n* nodes (actors)
  - *d(i)* is the number of links of node *i* – node degree
  - the more links, the higher centrality
  - range [0,1]

  - Undirected graph

$$C_D(i) = \frac{d(i)}{n-1}$$

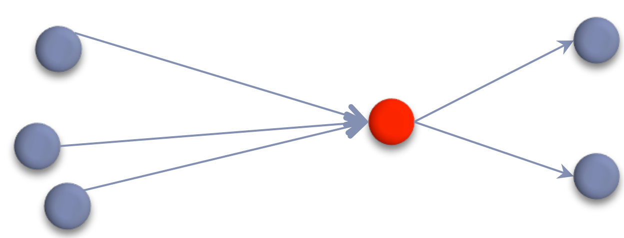


Network analysis: centrality measures

- Degree centrality of a node
  - network has *n* nodes (actors)
  - $d_o(i)$ is the number of **out-links** of node *i* – out-degree
  - the more links, the higher centrality
  - range [0,1]

  - Directed graph

$$C'_D(i) = \frac{d_o(i)}{n-1}$$

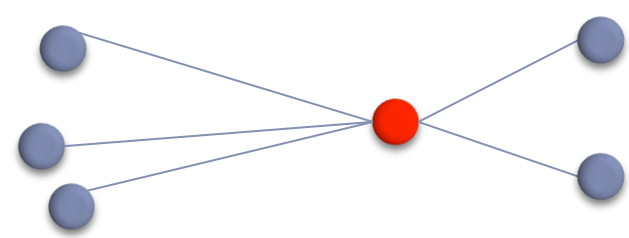


Network analysis: centrality measures

- Closeness centrality of a node
  - a node is important if it is closer to all other nodes
  - *d(i,j)* is the distance between nodes *i* and *j* – e.g. number of edges
  - range [0,1] (assuming a connected graph)

  - Undirected graph

$$C_D(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)}$$

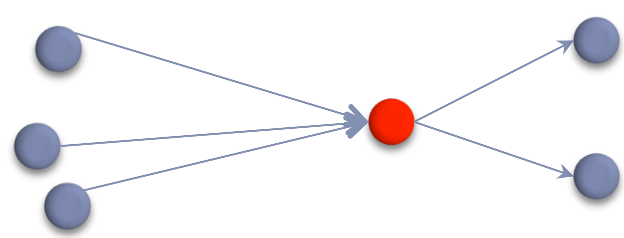


Network analysis: centrality measures

- Closeness centrality of a node
  - a node is important if it is closer to all other nodes
  - *d(i,j)* is the distance between nodes *i* and *j* – e.g. number of edges
  - range [0,1] (assuming a connected graph)

  - Directed graph – distance now considers direction

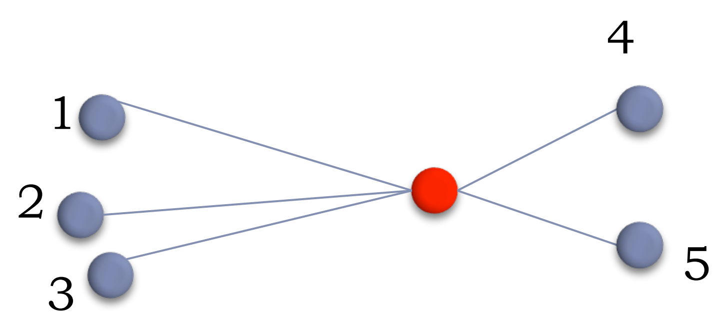
$$C_D(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)}$$



Network analysis: centrality measures

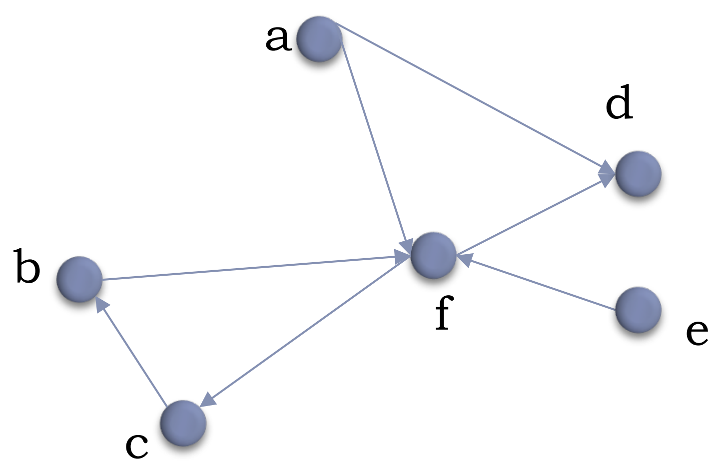
- **Betweeness centrality of a node**
  - a node is important if it is between other nodes
  - p_{jk} is the number of shortest paths between j and k
  - $p_{jk}(i)$ is the number of shortest paths between j and k that go through i (i≠j, i≠k)
    - range [0, (n-1)(n-2)/2]
  - Undirected graph

$$C_B(i) = \sum_{j<k,\, j\neq i,\, k\neq i} \frac{p_{jk}(i)}{p_{jk}}$$



Network analysis: centrality measures

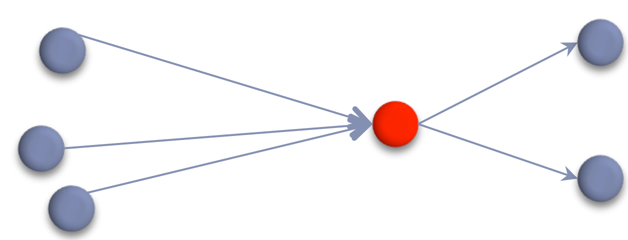
- There is data about friendship requests in a social net
  - Who would you pick as a marketing mate: f or d?
  - Who would you pick for collecting information?
  - Who would you pick for distribution of goods?



Network analysis: prestige measures

- Degree prestige of a node
  - a node is prestigious if it is referred by other nodes
  - Directed graph
  - $d_i(i)$ is the number of **in-links** of node i – in-degree

$$P_D(i) = \frac{d_i(i)}{n-1}$$



Network analysis: prestige measures

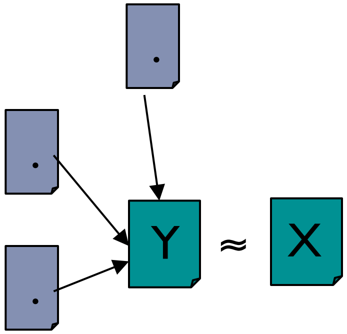
- node A is referred by n ordinary nodes
- node B is referred by n nodes, k of which prestigious
  - which node has higher prestige?

- we must take the prestige of pointing nodes into account

- HITS and PageRank do just that

Using web structure for information retrieval

- Search
  - Search a page about topic X
  - Each page Y is relevant according to
    - similarity between the content of X and Y

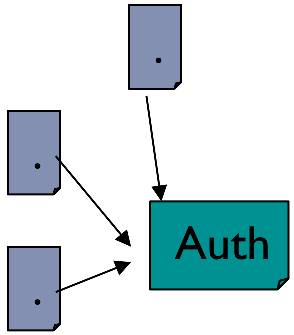


- Link analysis
  - Each page Y is relevant according to
    - number of references to page Y
    - content of pages which refer to Y

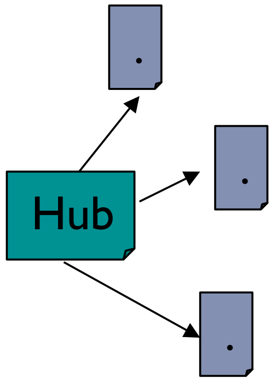
- Pages linked to pages with interesting content are also potentialy interesting

HITS (hyperlink induced topic search) [Chakrabarti et al.]

- Discovery of two kinds of pages
- Authorities
  - pages referred to by many other in a specific topic



- Hubs
  - pages that refer to many others



- In a first stage we use text similarity then we use link structure

Hubs e Authorities

- Relevance of an Authority
  - if a page is referred to by many others, then it must be relevant
  - it enables search more robust to variation in terms
    - example "data mining" and "machine learning"
- Quality of a Hub
  - if a hub refers to many important authorities then it is a good hub
- The relevance of an Authority and the quality of a Hub are interdependent

Link analysis with HITS

- Community discovery about a topic by computing hubs and authorities to that topic
  1. given a query (topic) Q, collect a set of seed pages S = {s1, s2, ..., sn } (*root set*)
  2. S is expanded to $T = S \cup \{d \mid s \rightarrow d \text{ or } d \rightarrow s, s \in S\}$
  3. initially, each page $r \in T$ has
     authority weight **a(r)** = 1,
     hub weight **h(r)**=1

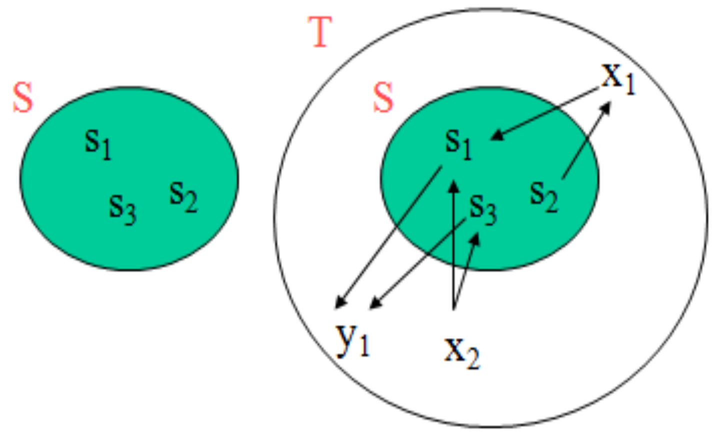
     $$a(r) = \sum_{d \to r} h(d)$$

     For each page we update the values of **a** and **h**

     $$h(r) = \sum_{r \to d} a(d)$$

     Normalize **a** and **h** and repeat step 3 until convergence (typically 10 it.)
  4. The community corresponds to the k top pages with highest **a** and **h**

Link analysis



x1→s1	a(s1)=1, h(s1)=1
x2→s1	a(s2)=1, h(s2)=1
s1→y1	a(s3)=1, h(s3)=1
s2→x1	a(x1)=1, h(x1)=1
x2→s3	a(x2)=1, h(x2)=1
s3→y1	a(y1)=1, h(y1)=1

iteração 1
a1(s1)=2, h1(s1)=1
a1(s2)=0, h1(s2)=1
a1(s3)=1, h1(s3)=1
a1(x1)=1, h1(x1)=1
a1(x2)=0, h1(x2)=2
a1(y1)=2, h1(y1)=0

iteração 1 (norm.)
a2(s1)=1, h2(s1)=0,5
a2(s2)=0, h2(s2)=0,5
a2(s3)=0,5, h2(s3)=0,5
a2(x1)=0,5, h2(x1)=0,5
a2(x2)=0, h2(x2)=1
a2(y1)=1, h2(y1)=0

iteração 2
a2(s1)=1,5 h2(s1)=1
a2(s2)=0 h2(s2)=0,5
a2(s3)=1 h2(s3)=1
a2(x1)=0,5, h2(x1)=1
a2(x2)=0, h2(x2)=1,5
a2(y1)=1 h2(y1)=0

iteração 2 (norm.)
a2(s1)=1 h2(s1)=0,66
a2(s2)=0 h2(s2)=0,33
a2(s3)=0,66 h2(s3)=0,66
a2(x1)=0,33 h2(x1)=0,66
a2(x2)=0, h2(x2)=1
a2(y1)=0,66 h2(y1)=0

iteração 3
a2(s1)=1,66 h2(s1)=0,66
a2(s2)=0 h2(s2)=0,33
a2(s3)=1 h2(s3)=0,66
a2(x1)=0,33 h2(x1)=1
a2(x2)=0 h2(x2)=1,66
a2(y1)=1,33 h2(y1)=0

iteração 3 (norm.)
a2(s1)=1 h2(s1)=0,4
a2(s2)=0 h2(s2)=0,2
a2(s3)=0,6 h2(s3)=0,4
a2(x1)=0,2 h2(x1)=0,6
a2(x2)=0 h2(x2)=1
a2(y1)=0,8 h2(y1)=0

topo authority: **s1** (1), **y1** (0,8), **s3** (0,6)

topo hub: **x2** (1) **x1**(0,6), **s1** (0,4)

HITS with an adjacency matrix

- The graph of connections / links can be represented by an adjacency matrix A
- Where
  - a not normalized (ann) is

    ann = $A^T h$

  - a normalized (a) is

    a = ann/max(ann)

  - similarly
    hnn = A.a;   h = hnn / max(hnn)

	[s1]	[s2]	[s3]	[x1]	[x2]	[y1]
[s1]	0	0	0	0	0	1
[s2]	0	0	0	1	0	0
[s3]	0	0	0	0	0	1
[x1]	1	0	0	0	0	0
[x2]	1	0	1	0	0	0
[y1]	0	0	0	0	0	0

HITS in pseudo-code

- Graph of connections given as an adjacency matrix
- Given a number o f iterations

```
> A
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    1
[2,]    0    0    0    1    0    0
[3,]    0    0    0    0    0    1
[4,]    1    0    0    0    0    0
[5,]    1    0    1    0    0    0
[6,]    0    0    0    0    0    0
```

hits-iterate(A)

$a_0 \leftarrow h_0 \leftarrow (1,1,...,1)$

$k \leftarrow 1$

Repeat

$hnn_k \leftarrow A \,.\, a_{k-1}$

$a_k \leftarrow ann_k \,/\, max(ann_k)$

$h_k \leftarrow hnn_k \,/\, max(hnn_k)$

Until $|a_k - a_{k-1}| < ea$ and $|h_k - h_{k-1}| < eh$

return ann_k , a_k , hnn_k , h_k

```
> hits(A,4)
     [,1] [,2]  [,3]  [,4]
[1,]  1.6  0.8 1.000 0.500
[2,]  0.0  0.2 0.000 0.125
[3,]  1.0  0.8 0.625 0.500
[4,]  0.2  1.0 0.125 0.625
[5,]  0.0  1.6 0.000 1.000
[6,]  0.8  0.0 0.500 0.000
>
```

HITS by eigenvectors

$$a = A^T h \qquad h = Aa$$

$$a = A^T A a \qquad h = AA^T h$$

- with normalization

$$a = A^T .k_1 h \qquad h = A.k_2 a$$

$$a = A^T A.k_2 a \qquad h = AA^T .k_1 h$$

- a is the largest eigenvector of $A^T A$
- h is the largest eigenvector of $A.A^T$

  - x eigenvector of M if Mx=k.x, where k is a scalar

And on R we get

```
> t(eigen(t(A)%*%A)$vectors[,1])
          [,1] [,2]      [,3] [,4] [,5] [,6]
[1,] 0.8506508    0 0.5257311    0    0    0
```

- i.e., authorities are s1 and s3

```
> t(eigen(A%*%t(A))$vectors[,1])
     [,1] [,2] [,3]      [,4]      [,5] [,6]
[1,]    0    0    0 0.5257311 0.8506508    0
```

- hubs are x2 and x1

- Function eigen computes eigenvectors (among other things)

Comments on eigenvectors

- Iterative algorithm finds the principal eigenvectors
  - Major communities
- Other eigenvectors
  - Alternative communities
    - E.g. query "classification" or "football"

- Convergence
  - HITS always converges
  - Different initializations may give different results
    - If there are repeated principal eigenvalues
    - If A^TA is reducible
      - the graph is not strongly connected

Notes

- Search in HITS starts by CONTENT relevance
  - root set
- then content is ignored
  - only links are exploited

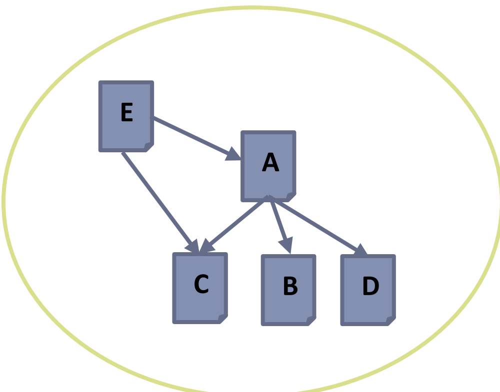
- Example:
  - look for pages of "japanese car manufacturers"
    - the page of Honda will not have this page
  - software companies...
  - pages are not typically self descriptive

How to identify inlinks

- google.com
  - Query “link: <url>”
  - no space after :

- results are a sample of the actual set of links

Activity



- Determine the most interesting hub
- Determine the most important authority
- Suppose we are looking for information about a car model X and page A contains that model, how would that change your previous results?
- Think of way for enhancing recommendation using hits

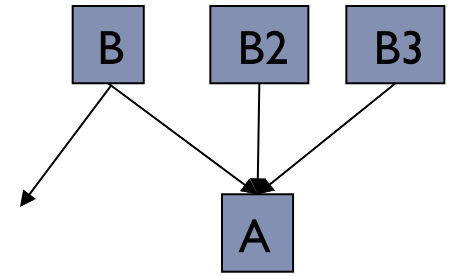
PageRank

- HITS was proposed in January 1998
  - Kleinberg
- PageRank was proposed in April 1998 and is used by Google
  - Sergey Brin and Larry Page

- HITS and PageRank have many similarities
- but they have very important differences
  - computational
  - robustness of results

- The idea of PageRank
  - rank pages according to their prestige
  - prestige is (mainly) determined by **inlinks** and their prestige

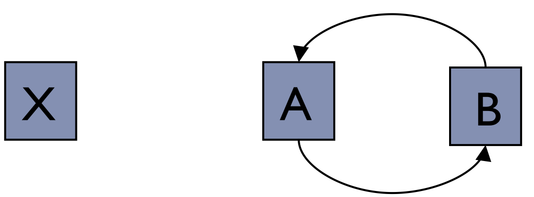
PageRank: The idea



- We consider a random robot
  - prob(página B → página A) = 1/n
    - n is the number of outlinks from B
  - prob(being on page A comming from B) = prob(B)/n
  - prob(A) =

    prob(B)/Out(B) + prob(B2)/Out(B2) + prob(B3)/Out(B3)

- Most important pages will have higher probability
- What about loops and direct accesses?



PageRank Mathematics

- R(i) = R(j1) / Oj1 + R(j2) / Oj2 + …
- How can we determine R(i) ?
  - system of n equations and n unknowns
- R = <R(1), R(2),…,R(n)>
- Aij = 1/Oi OR zero
- $R = A^T.R$
- This could be enough, but…

```
> Adjacency matrix
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    1
[2,]    0    0    0    1    0    0
[3,]    0    0    0    0    0    1
[4,]    1    0    0    0    0    0
[5,]    1    0    1    0    0    0
[6,]    0    0    0    0    0    0
```

```
> A
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    1
[2,]    0    0    0    1    0    0
[3,]    0    0    0    0    0    1
[4,]    1    0    0    0    0    0
[5,]  0,5    0  0,5    0    0    0
[6,]    0    0    0    0    0    0
```

PageRank – some problems

- $R = A^T.R$

- For the above to have a unique solution A must be
  - **stochastic** (all rows must sum 1)
    - often it is not: there are nodes with no outlinks
    - solution 1: remove nodes without outlinks
    - solution 2: artificially insert equal weights into a row with zeros

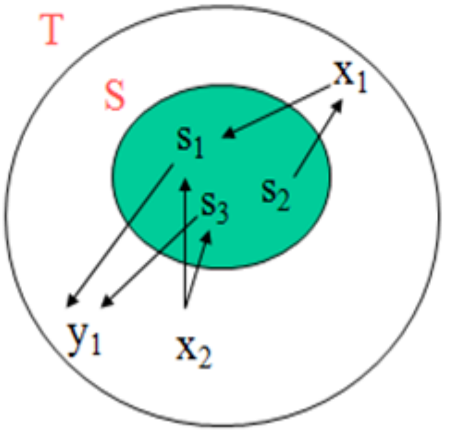
```
> A
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    1
[2,]    0    0    0    1    0    0
[3,]    0    0    0    0    0    1
[4,]    1    0    0    0    0    0
[5,]  0,5    0  0,5    0    0    0
[6,]    0    0    0    0    0    0
```

```
> A
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    1
[2,]    0    0    0    1    0    0
[3,]    0    0    0    0    0    1
[4,]    1    0    0    0    0    0
[5,]  0,5    0  0,5    0    0    0
[6,]  1/6  1/6  1/6  1/6  1/6  1/6
```

PageRank – some problems

- $R = A^T.R$

- ...A must be
  - **irreducible** (in the graph there is a path from any node to any other node)
    - often it is not the case (there is no path from S1 to S2)
  - **aperiodic** (the greatest common divisor of all cycles for each node is 1)
    - A→B, B→C, C→A: the cycle has period 3
    - No loop traps



- Solution
  - add a link to every two pages
    - in fact, if one is in one page can go directly to any other by typing its URL
    - the probability of transition is controlled by a parameter d

PageRank: Teleportation

- When on a page B, there is a certain probability (ex: 0.1) of teleporting to a page A which has no direct connection to B
  - prob(getting to A) = 0.1/(number of nodes) + 0.9 * prob(direct access)

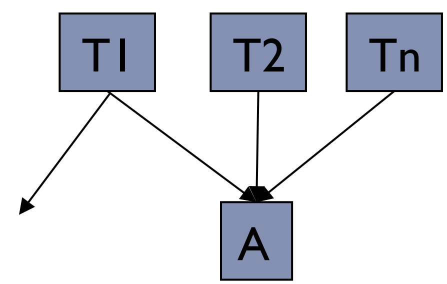
- R(A) is proportional to this probability

PageRank

http://www-db.stanford.edu/~backrub/google.html
http://www.iprcom.com/papers/pagerank/

$$R(A) = (1-d) + d*(R(T1)/Out(T1) + \ldots + R(Tn)/Out(Tn))$$

- R(X) page rank of page X
- d damp factor (solves connectivity problems and models direct accesses)
- Out(X) number of outlinks of X



PageRank additional criteria*

- (*improving the user model*)
- Visibility of a link
- Position of a link within a document
- Distance between web pages
  - same server, same domain, same region
- Importance of a linking page
- Up-to-dateness of a linking page

*http://pr.efactory.de/e-further-factors.shtml

amjorge@fc.up.pt

Algorithm

- We can solve a system of equations
- We can calculate R iteratively
  - assign initial R values to pages
  - calculate new values for R
  - iterate (number of iterations depends on the size of the network)

PageRank and HITS

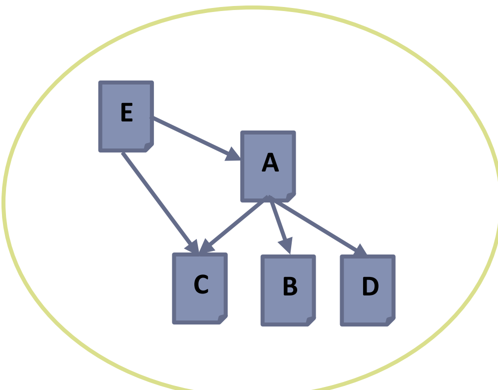
- PageRank can be computed offline
  - it is query independent
    - which can be a disadvantage: a page can be an authority in a topic but not in general
      - comparar www.publico.pt ou www.oftalmologia.pt
- PageRank is more robust to SPAM
  - importance of a page depends on inlinks not outlinks
- PageRank does not consider time
- PageRank is more robust to perturbations in the input than HITS

Web Spamming

- Artificially increasing the rank of a page without increasing its specific information value
  - Search Engine Optimization can be spam or not
    - debateable (http://www.webworkshop.net/ethical-search-engine-optimization.html)

- Content Spamming
  - insert popular words (even if unrelated)
  - repeating important terms
  - dumping many unrelated terms
- Link Spamming
  - outlink spamming: directory cloning
  - inlink spamming
    - honey-pot
    - submit URLs to Web Directories
    - Posting links to forums or the like
    - link exchange schemes
    - spam farms

Activity



- Assume damp factor of 0.9
- Suppose the PageRank of A is 1, what is the PR of B?
- and of C?
- Determine the PageRank of the pages of the graph

Community Discovery

- Community: group of entities (people, organizations) sharing common interests.
  - Users who like metal music
  - Treckies
- What for?
  - Source of resources for users with similar interests
  - Sociology of the web: we know better, we can exploit better
  - Target advertising

amjorge@fc.up.pt

Community Discovery

- Given a set of entities S
  - Of the same type
- A community is
  - A pair C = (Theme ,Group)
- Example
  - Users who like metal music

amjorge@fc.up.pt

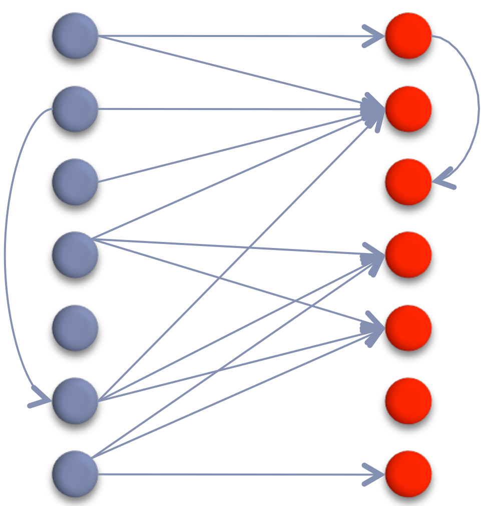
Communities

- Web pages
  - Users in the same community are usually interconnected through hyperlinks
  - Pages contain words that reveal the theme
- Emails
  - Members of a community exchange emails (links)
  - Emails contain words revealing the theme
- Documents
  - Members of a community are more likely to appear together in the same sentences or documents (this is the link)
  - Words indicate the community theme

amjorge@fc.up.pt

Algorithm: Bipartite Core Communities

- Bipartite graph of Fans and Centers
  - Music fans and band pages
- Identifying (i,j) cores
  - i fans and j centers
  - Fans ~ Hubs, Centers ~Authorities
- We could use HITS
  - But computing eigenvalues is relatively inefficient
  - We will describe an algorithm by R. Kumar

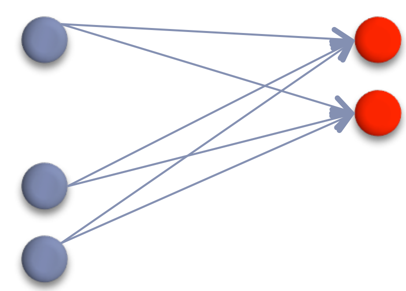


Bipartite graph

amjorge@fc.up.pt

Algorithm: Bipartite Core Communities

- Pruning
  - Delete pages that are too highly referenced
    - Inlink > 500
  - Prune fans and centers
    - Fans with outdegree < j
    - Centers with indegree < i
    - Example for (i=3,j=2)

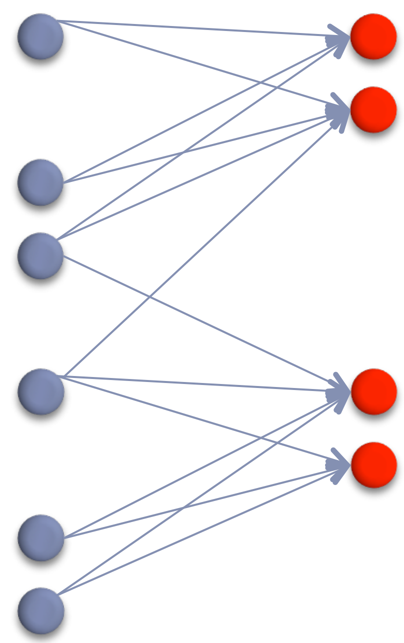


After pruning

amjorge@fc.up.pt

Algorithm: Generating all (i,j) cores

- After Pruning
  - Fix j, start with all (1,j) cores
    - Set of fans with outdegree at least j
  - Look for (2,j) cores by checking every fan that points to a center in a (1,j) core
  - Similarly for (3,j) in a APRIORI fashion
- Note
  - This algorithm finds cores of communities, not the whole community

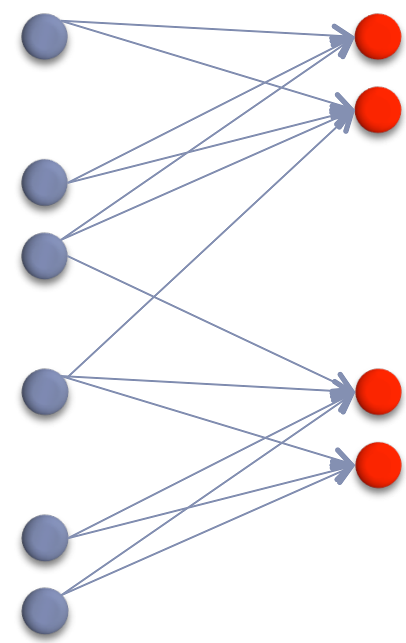


Suppose this was the result of pruning

amjorge@fc.up.pt

Algorithm: Generating all (i,j) cores

- Example
  - Find all (3,2) cores
    - Identify (1,2) cores
      - Fans with 2 outlinks (min)
    - Identify (2,2) cores
      - Combine pairs of fans to find larger cores
    - Identify (3,2) cores
  - Find all (3,3) cores



Suppose this was the result of pruning

amjorge@fc.up.pt

Resources

- Books
  - Web Data Mining, Bing Liu, Springer, 2007
  - Mining the World Wide Web, Chang, G., Healey, M., McHugh, J., Wang, J., Kluwer Academic Press, 2001.

- Article
  - Google's PageRank Explained and how to make the most of it, Phil Craven, http://www.webworkshop.net/pagerank.html