CAP4770/5771
Introduction to Data Science
Fall 2015

University of Florida, CISE Department
Prof. Daisy Zhe Wang

Based on notes from CS194 at UC Berkeley by Michael Franklin, John Canny, and Jeff Hammerbacher

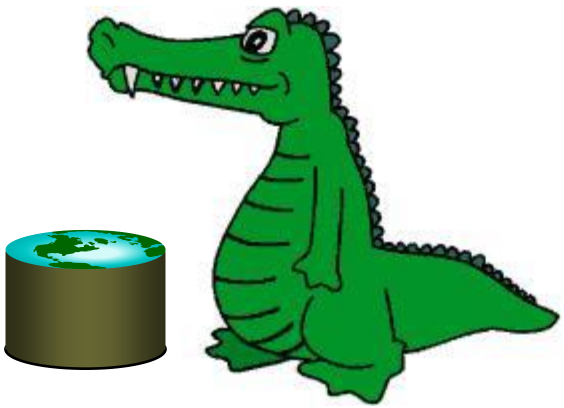
Page Rank:
Link Analysis over Large Graphs

Web Graph and Link Analysis

Page Rank Algorithm

Dead Ends and Spider Traps

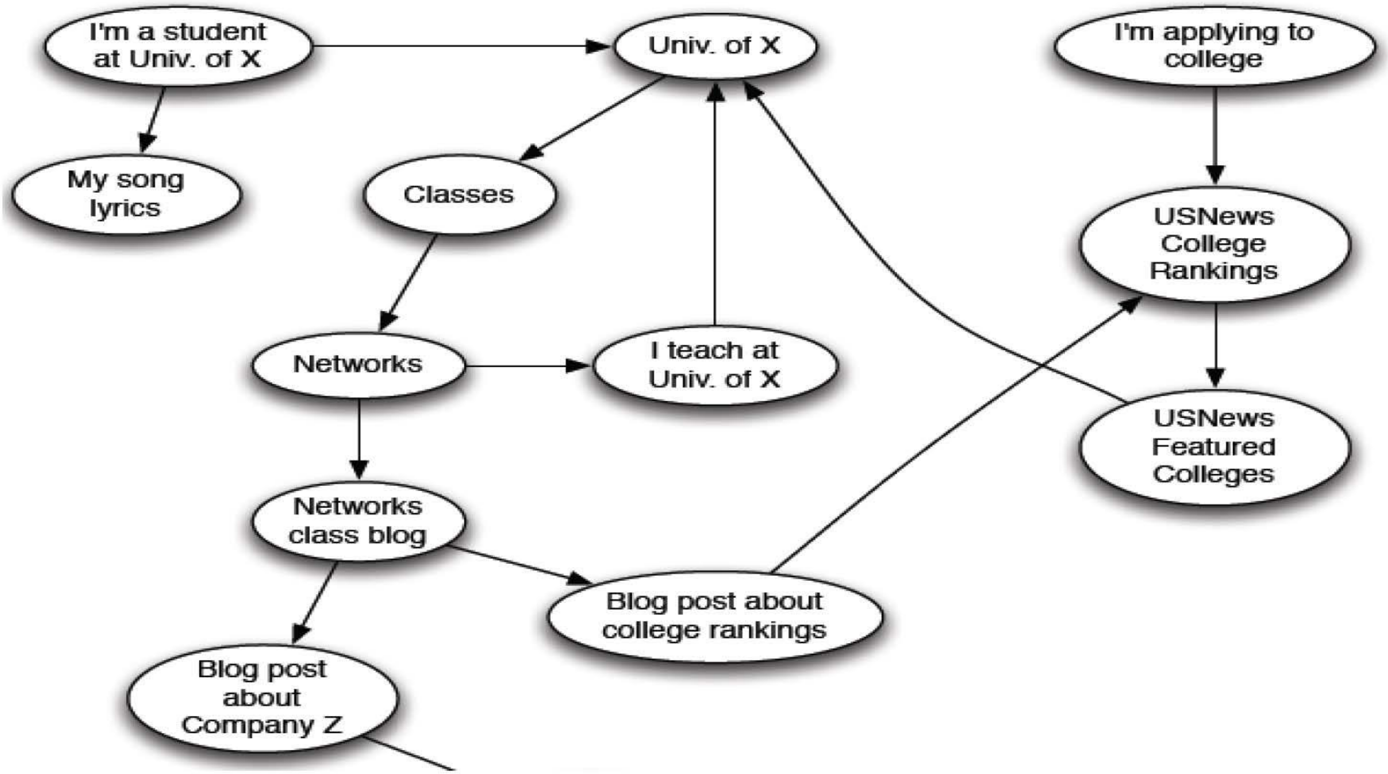
Other Types of Graph Data



Adapted Slides from Jeff Ullman, Anand Rajaraman and Jure Leskovec from Stanford

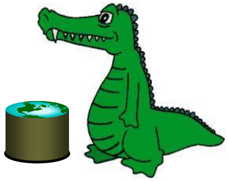
Web as a Graph

- Web as a directed graph:
  - Nodes: Webpages
  - Edges: Hyperlinks



How to organize the Web?

- First try: Human curated Web directories (e.g., Yahoo)
- Second try: Web Search
  - Information Retrieval using inverted index
  - Good for finding relevant docs in a small and trusted set (e.g., Newspaper articles, Patents)
  - But: Web is huge, full of untrusted documents, random things, web spam, etc.
    - E.g., Word Spam



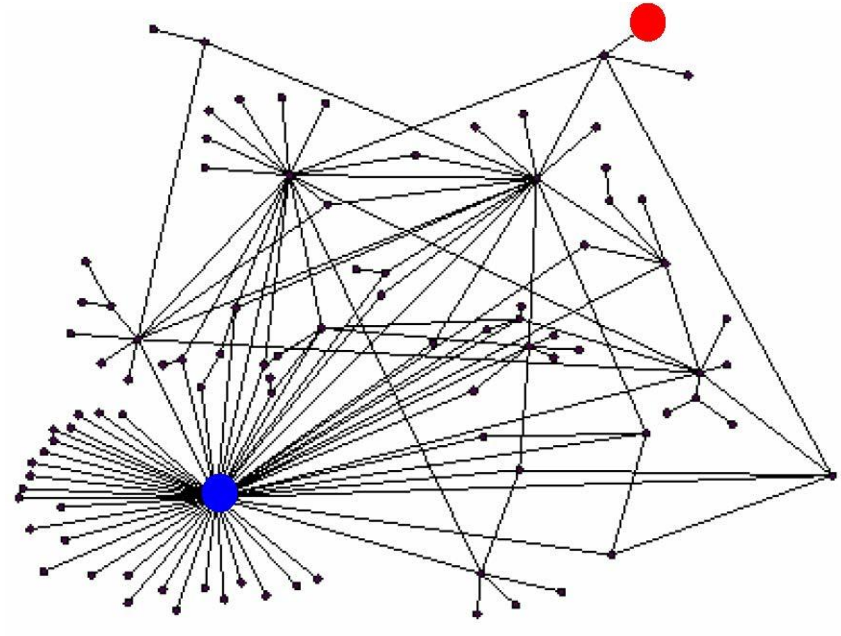
2 Challenges of Web Search

1. Web contains many sources of information. Who to "trust"?
   - Observation: Trustworthy pages point to each other! (in&out-links)
2. What is the "best" answer to query "newspaper"?
   - No single right answer
   - Observation: Pages that actually know about newspapers might all be pointing to many newspapers (outlink)
   - Observation: a good newspaper is pointed to from many sources (inlink)

Solution: Ranking nodes on the Graph Based on Link Structures!

- All web pages are not equally “importantce” can be captured by link structures

- There is large diversity in the web-graph node connectivity.
- Let’s rank the pages by the link structure!
  - Link Spam also possible but harder



Link Analysis

- Link Analysis algorithms: for computing importance of nodes in a graph
  - Page Rank
  - Topic-Specific (personalized) Page Rank
  - Mining for Communities
  - Web Spam Detection Algorithms

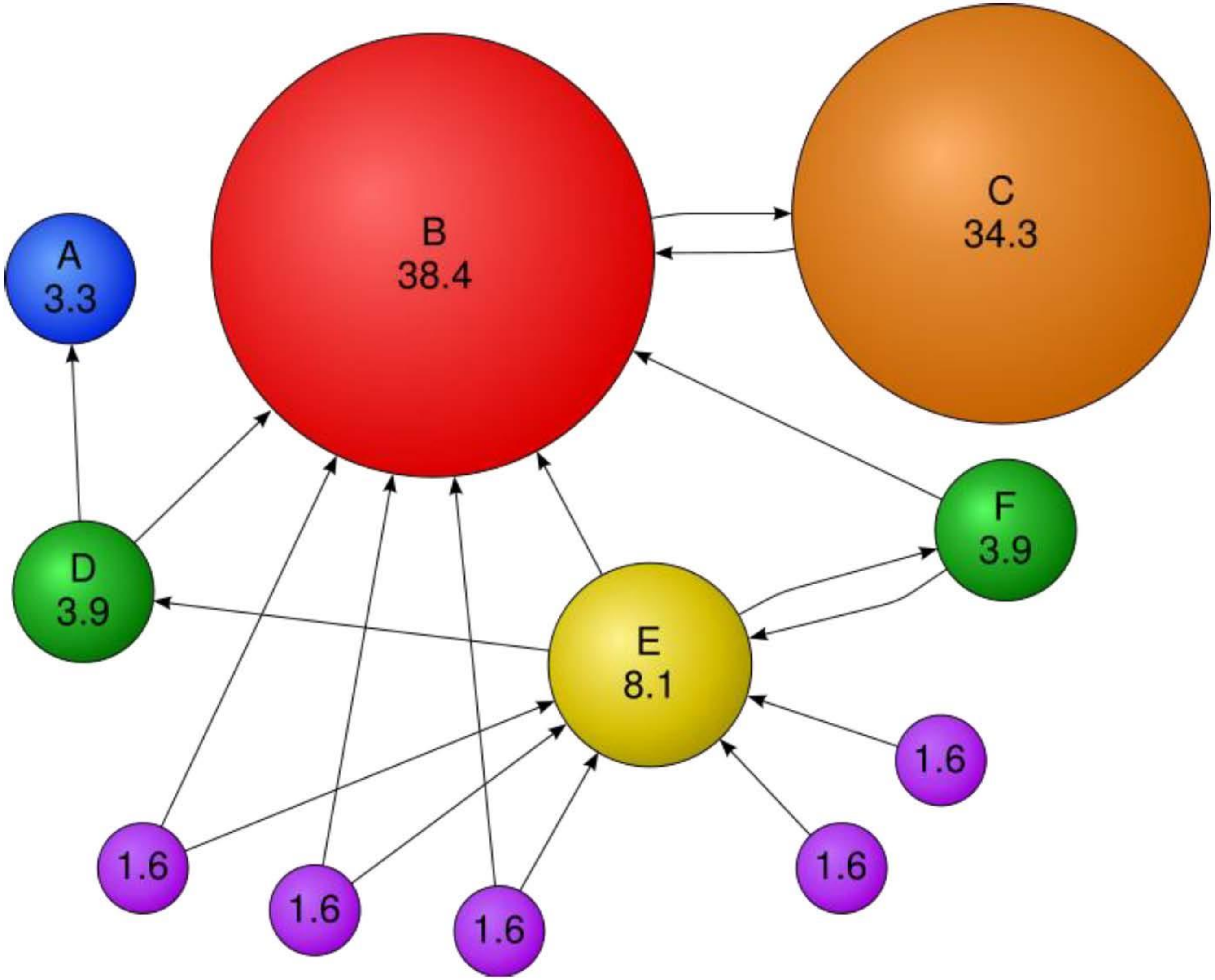
How to Rank Web Pages Based on Link Analysis

- Web pages are not equally “important”
  - www.joe-schmoe.com **vs.** www.ufl.edu
  - Page is more important if it has more inlinks

- Inlinks as votes
  - www.ufl.edu has 23,400 inlinks
  - www.joe-schmoe.com has 1 inlink

- Are all inlinks equal?
  - Recursive question!
  - Links from important pages count more

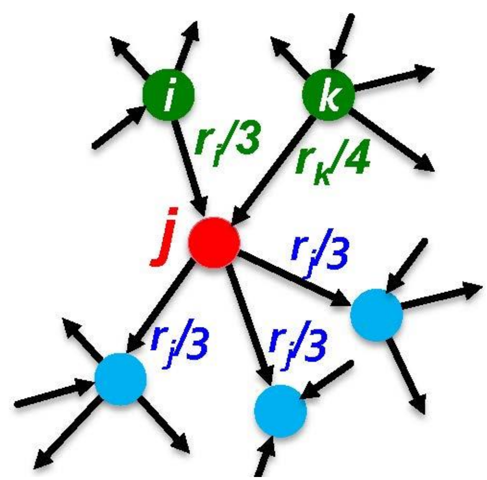
Example PageRank scores



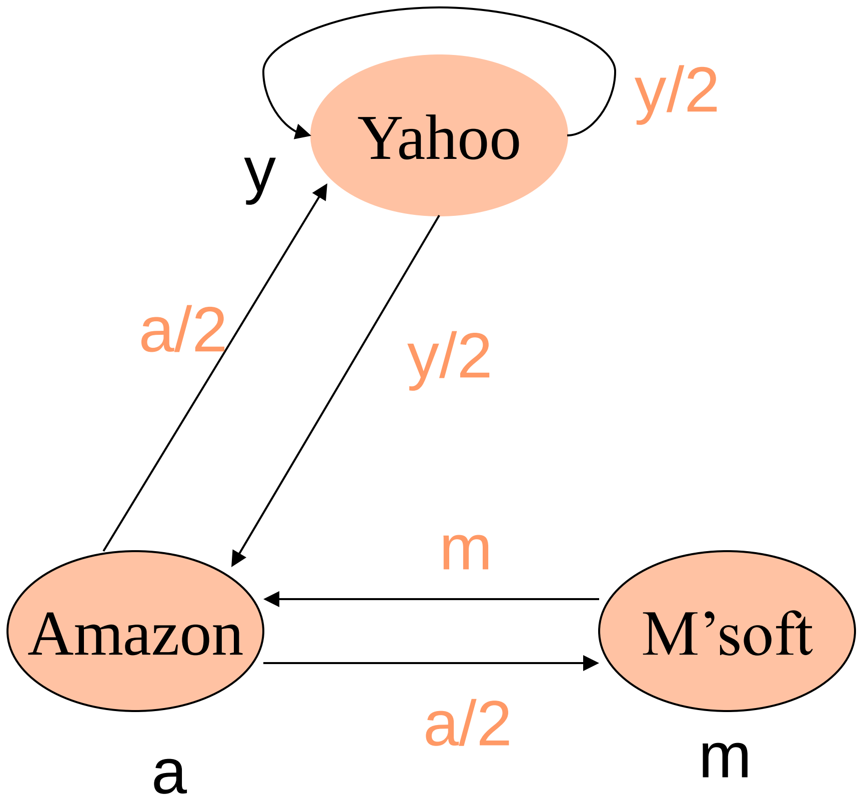
Simple recursive formulation

- Each link's vote is proportional to the importance of its source page
- If page j with importance r_j has n outlinks, each link gets r_j / n votes
- Page j's own importance is the sum of the votes on its inlinks
  - $r_j = ?$

    $= r_i /3 + r_k /4$



Simple “flow” model



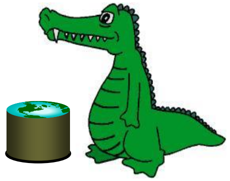
- A “vote” from an important page worth more
- A page is important if it is pointed to by other important pages
- Define a “rank” r_j for page j

$$r_j = \sum_{i \to j} r_i / d_i$$

d_i *is out-degree of node i*

"flow" equations:

$$y = y/2 + a/2$$
$$a = y/2 + m$$
$$m = a/2$$

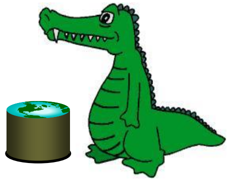


Solving the flow equations

flow” equations:

$$y = y/2 + a/2$$
$$a = y/2 + m$$
$$m = a/2$$

- 3 equations, 3 unknowns, no constants
  - No unique solution
  - All solutions equivalent modulo scale factor
- Additional constraint forces uniqueness
  - $y+a+m = 1$
  - $y = 2/5$, $a = 2/5$, $m = 1/5$
- Gaussian elimination method works for small examples, but we need a better method for large graphs

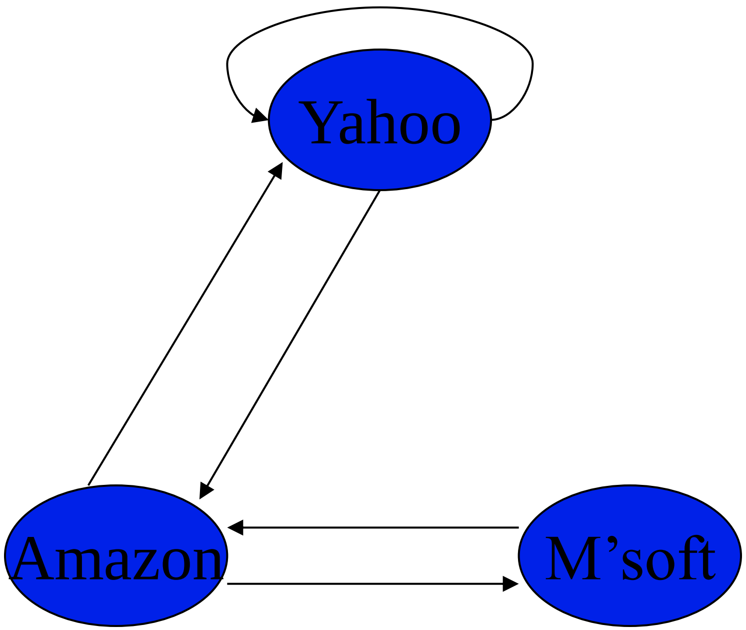


Matrix formulation

- Stochastic adjacency matrix **M**
  - Matrix **M** has one row and one column for each web page
  - Let page i has di outlinks
  - If i → j, then M_{ij}=1/di, Else M_{ij}=0
  - Each columns in **M** sum to 1
- Rank vector **r** : vector with one entry per web page
  - r_i is the importance score of page i
  - $\sum_i \mathbf{r}_i = 1$
- The flow equations can be written as

$$\mathbf{r} = \mathbf{Mr}$$

Matrix Formulation Example

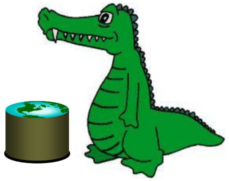


$$
\begin{array}{c|ccc|}
 & y & a & m \\
\hline
y & 1/2 & 1/2 & 0 \\
a & 1/2 & 0 & 1 \\
m & 0 & 1/2 & 0 \\
\hline
\end{array}
$$

r = Mr

$$
\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}
$$

$$
\begin{aligned}
y &= y/2 + a/2 \\
a &= y/2 + m \\
m &= a/2
\end{aligned}
$$



Rank Vector r = Eigenvector of M

- The flow equations can be written

$$\mathbf{r} = \mathbf{M}\mathbf{r}$$

- The rank vector **r** is an eigenvector of the stochastic web matrix **M**
  – with corresponding eigenvalue 1

- We can now efficiently solve for **r**!
  – The method is called Power iteration

Power Iteration method

- Given a web graph with N nodes, where the nodes are pages and edges are hyperlinks
- Power iteration: a simple iterative scheme
  - Suppose there are N web pages
  - Initialize: $\mathbf{r}^0 = [1/N,\ldots.,1/N]^T$
  - Iterate: $\mathbf{r}^{k+1} = \mathbf{M}\mathbf{r}^k$
  - Stop when $|\mathbf{r}^{k+1} - \mathbf{r}^k|_1 < \varepsilon$
    - $|\mathbf{x}|_1 = \sum_{1 \le i \le N} |x_i|$ is the L_1 norm
    - Can use any other vector norm e.g., Euclidean norm

$$r^{(t+1)}_j = \sum_{i \to j} r^{(t)}_i / d_i$$

d_i *is out-degree of node i*