BIG DATA ANALYTICS
Link Analysis and PageRank

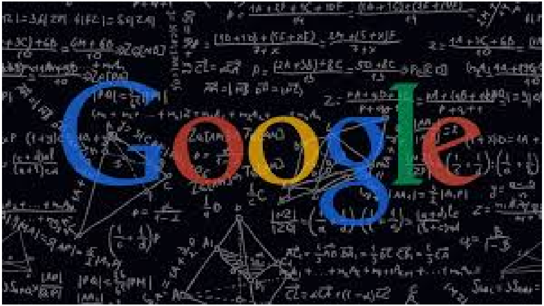
ESSEC
BUSINESS SCHOOL

Problem: efficient Web search

- The availability of efficient and accurate Web search
- Google: the first able to defeat the spammers
- The innovation provided by Google: **"PageRank"**

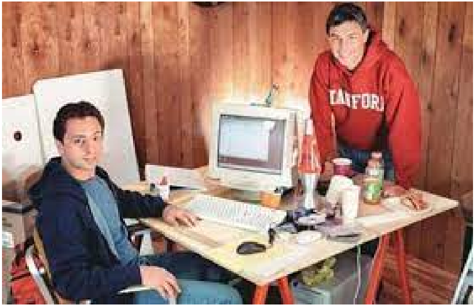
Solutions

- **PageRank**: an essential technique for a search engine
- Spammers invented ways to manipulate the PageRank
- $\Longrightarrow$ **TrustRank** (and other techniques) for preventing spammers from attacking PageRank



PageRank

- Larry Page, the co-inventor and a co-founder of Google
- PageRank: a tool for evaluating the importance of Web pages
    - Ideas: "random surfers" and "taxation"



Early Search Engines

Before Google:

- Crawling the Web
- listing the terms found in each page in an inverted index
  - makes it easy, given a term, to find all the places where that term occurs
- A search query is issued:
  - pages with those terms extracted from the inverted index
  - Ranked reflecting the use of the terms within the page:
    - presence of a term in a header
    - large numbers of occurrences

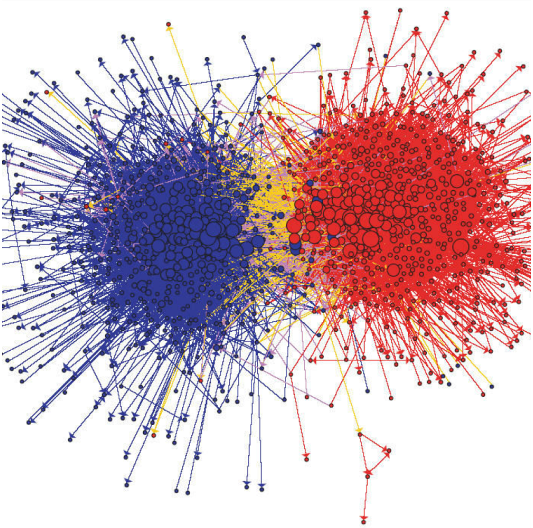
Term Spam

- How to fool search engines?
  - E.g., you were selling shirts on the Web
  - Add a term "movie" to your page thousands of times
  - Give it the same color as the background
  - When a user issued a search query with the term "movie", the search engine would list your page first
  - If simply adding "movie" to your page didn't do the trick:
    - give the query "movie"
    - copy the page that come back as the first choice into your own
    - use the background color to make it invisible.
- **Term spam** made early search engines almost useless...

Graph Data: Social Networks

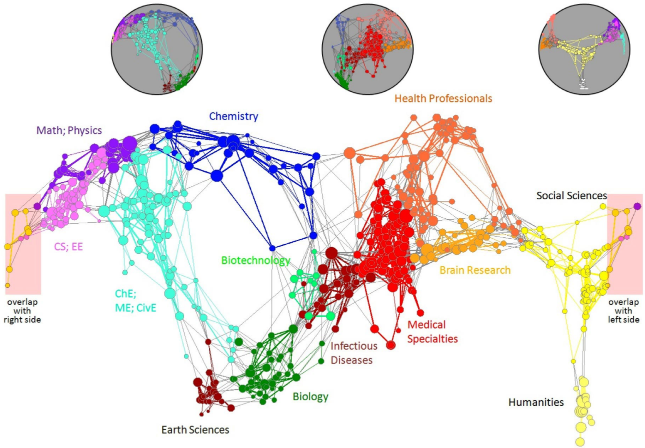
Facebook social graph, [Backstrom-Boldi-Rosa-Ugander-Vigna, 2011]

Graph Data: Media Networks



Connections between political blogs, [Adamic-Glance, 2005]

Graph Data: Information Nets



Citation networks and Maps of science, [Börner et al., 2012]

Web as a Graph

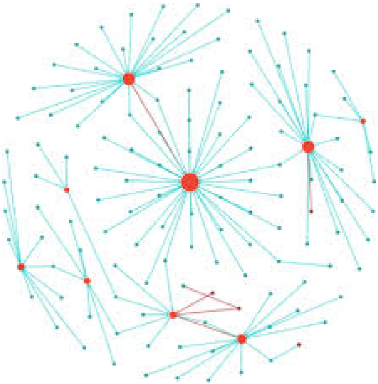
- Web is a directed graph
- Nodes: Webpages
- Edges: Hyperlinks

Web search: challenges

- **Web contains many sources of information. Who to trust?**
  - Trick: trustworthy pages may point to each other!
- **What is the best answer to query ”newspaper”?**
  - No single right answer
  - Trick: pages that actually know about newspapers might all be pointing to many newspapers

Ranking Nodes on the Graph

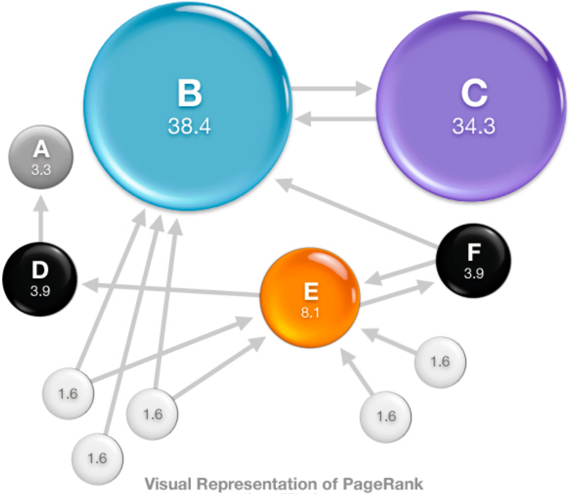
- All web pages are not equally "important"
- There is large diversity in the web-graph node connectivity
- Let's rank the pages by the link structure!



Links as Votes

- Idea: links are votes
- Page is more important if it has more links
- Are all in-links equal?
  - Links from important pages count more

PageRank Scores



Visual Representation of PageRank
*Source: Wikipedia.

Recursive Formulation

- Each link's vote is proportional to the importance of its source page
- If page j with importance r_j has n out-links, each link gets r_j/n votes
- Page j's own importance is the sum of the votes on its in-links

The "flow" model

- A "vote" from an important page is worth more
- A page is important if it is pointed to by other important pages
- The rank r_j of page j:

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

d_i out-degree of node i

Solving the flow equations

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

- No unique solution
- Additional constraint forces uniqueness: $\sum r_i = 1$
- Gaussian elimination method works for small examples
- A better method for large web-size graphs?

Matrix Formulation

- Stochastic adjacency matrix M:
  - Let page i has d_i out-links
  - If $i \rightarrow j$, then $M_{ji} = \frac{1}{d_i}$ else $M_{ji} = 0$
  - M is a column stochastic matrix
    - Columns sum to 1
- Rank vector r: vector with an entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$
- **The flow equation**:

$$r = M \cdot r$$

Eigenvector Formulation

- The flow equation:

$$r = M \cdot r$$

- The rank vector r is an eigenvector of the stochastic matrix M
- We can now efficiently solve it!
- **Power iteration**

Random Walk Interpretation

- Imagine a random web surfer:
  - At any time t, surfer is on some page i
  - At time $t+1$, the surfer follows an out-link from i uniformly at random
  - Ends up on some page j linked from i
  - Process repeats indefinitely

The stationary Distribution

- Where is the surfer at time $t + 1$?
- Follows a link:

$$p(t+1) = Mp(t)$$

- Suppose the random walk reaches a state

$$p(t+1) = Mp(t) = p(t)$$

  then $p(t)$ is stationary distribution of random walk

- our original rank vector r satisfies $r = Mr$
- So, r is a stationary distribution for the random walk

PageRanking: three questions

$$r = Mr$$

- Does this converge?
- Does it converge to what we want?
- Are results reasonable?

Existence and Uniqueness

For graphs that satisfy certain conditions, the stationary distribution is unique and eventually will be reached no matter what the initial probability distribution at time $t = 0$

PageRank

PageRank simulates where Web surfers, starting at a random page, would tend to congregate if they followed randomly chosen outlinks

- Pages with a large number of surfers considered more "important"
- Google prefers important pages to unimportant pages

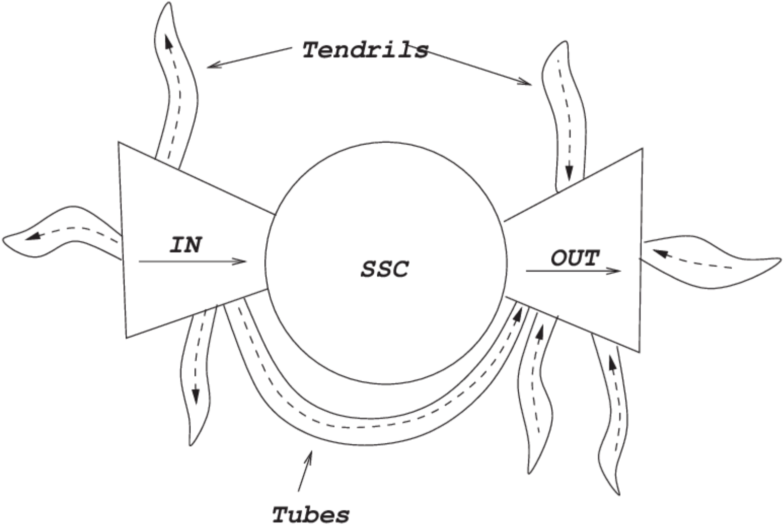
Simplified PageRank?

- Computing PageRank by simulating random surfers is a time-consuming process...
  - Simply counting the number of in-links for each page ??
    - "Spam farm" of a million pages, each of which linked to his shirt page
    - $\Longrightarrow$ the shirt page looks very important...

Why does it work?

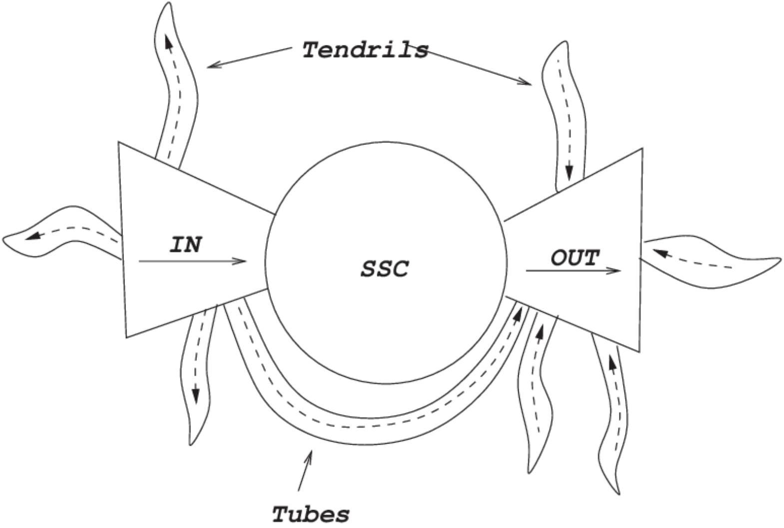
- Hard to fool Google
  - E.g., the shirt-seller can still add "movie" to his page
  - Google believed what other pages say about him
  - Create many pages of his own, and link to his shirt- selling page ??
  - Those pages would not be given much importance by PageRank...

Structure of the Web



"Analysis of the Greek Web-space" [T. Mchedlidze et al,]

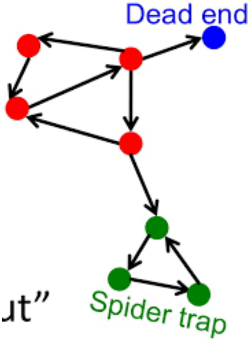
Structure of the Web



"Analysis of the Greek Web-space" [T. Mchedlidze et al,]

PageRank: problems

- **Dead ends** (have no out-links):
  - Random walk has “nowhere” to go to
  - Such pages cause importance to “leak out”
- (2) **Spider traps**: (all out-links are within the group):
  - Random walked gets “stuck” in a trap
  - Spider traps absorb all importance...



J. Leskovec, et al: Mining of Massive Datasets.

Solution: teleports!

- The Google solution for spider traps: At each time step, the random surfer has two options:
    1. with probability β follow a link at random
    2. with probability $1 - \beta$ jump to some random page
    3. Common values for β are in the range 0.8 to 0.9
- **Surfer will teleport out of spider trap or a dead end within a few time steps**

Using PageRank in a Search Engine

- A secret formula that decides the order in which to show pages to the user
- Google: over 250 different properties of pages
  - A page has to have at least one of the search terms in the query
    - Normally, unless all the search terms are present, a page has very little chance of being in the top ten
  - A score is computed for each qualified page
  - An important component: the PageRank of the page
  - Other components: the presence or absence of search terms in prominent places
  - ...

References

- J. Leskovec, A. Rajaraman and J. D. Ullman *Mining of Massive Datasets* (2014), Chapter 5
- S. Brin and L. Page, "Anatomy of a large-scale hypertextual web search engine", Proc. 7th Intl. World-Wide-Web Conference, pp. 107 - 117, 1998.