Network Structure of the Web

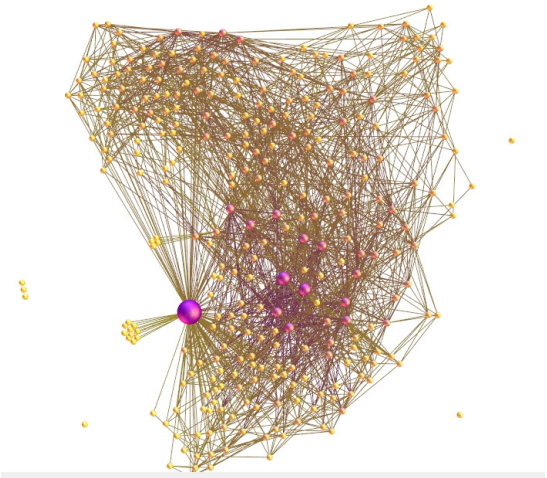
Chapter 21
(plus additional references)

The Web as a Graph

- Nodes: static Web pages
- Links: directed hyperlinks
- Notion of “in-links”, “out-links”, “in-degree”, “out-degree”.
- **Questions:**
  - What is the structure of this graph?
  - How did this structure come about?
  - How to best search, given this structure?
  - How to best crawl, given this structure?
  - How to mine the structure to find communities, themes?

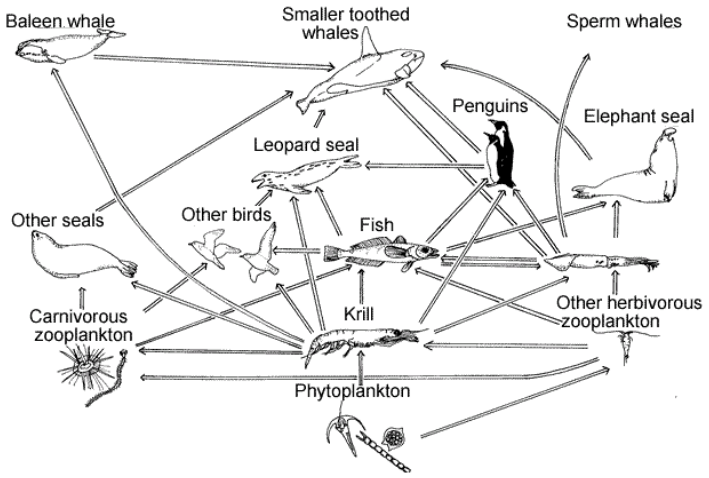
Sidetrack:
The Science of Networks

Examples of Complex Networks



Neural Network
(*C. Elegans*)

http://gephi.org/wp-content/uploads/2008/12/screenshot-celegans.png



Food Web

http://www.coolantarctica.com/Antarctica%20fact%20file/wildlife/whales/foodweb.gif

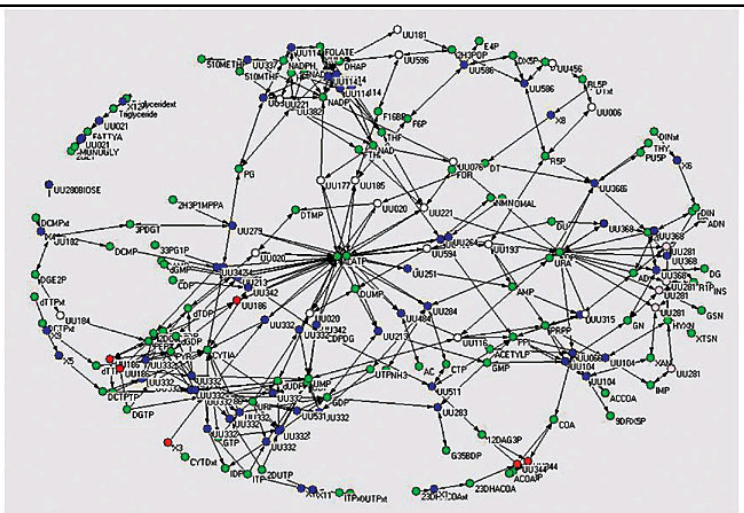
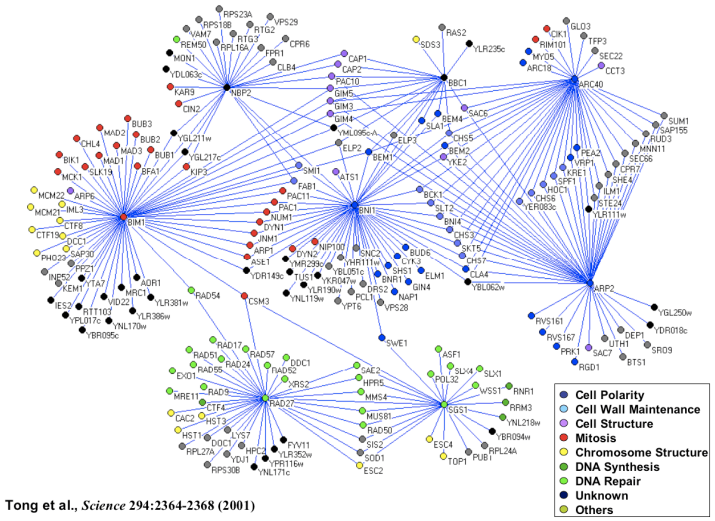


Figure 2. Bipartite graph of the metabolic network of *Ureaplasma urealyticum*. Dark gray and white nodes represent enzymes and light gray nodes represent metabolites (Lemke et al., 2004).

Metabolic Network

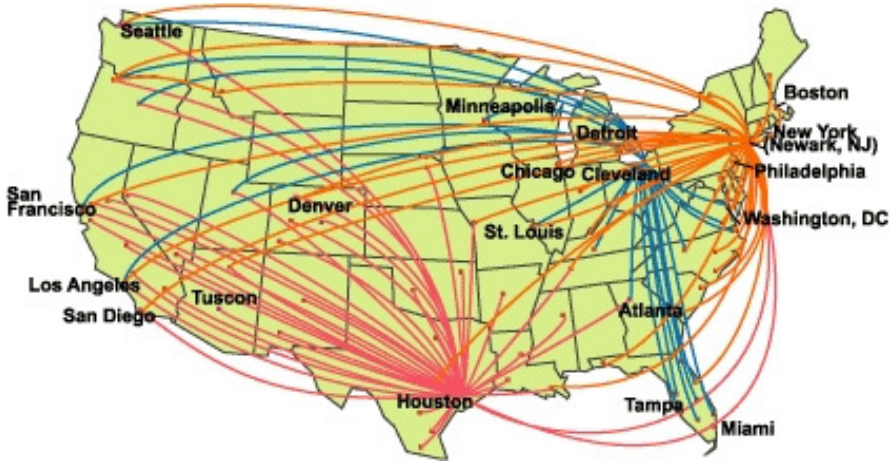
http://www.funpecrp.com.br/gmr/year2005/vol3-4/wob01_full_text.htm



Tong et al., *Science* 294:2364-2368 (2001)

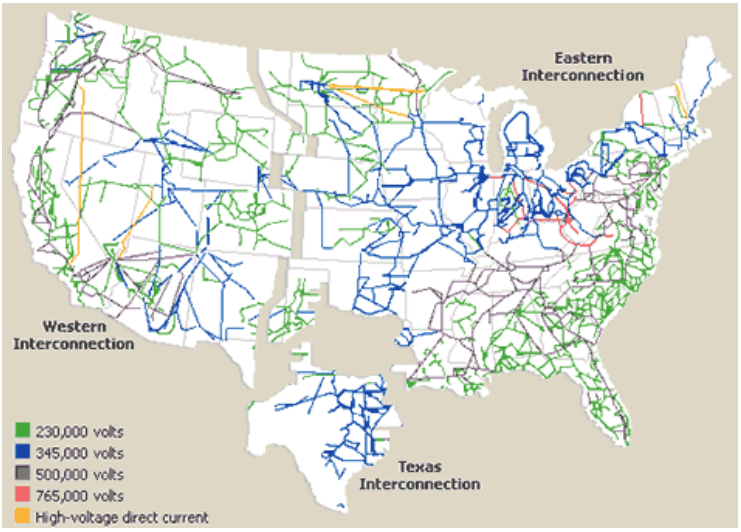
Genetic Regulatory Network

http://expertvoices.nsdl.org/cornell-info204/files/2009/03/figure-3.jpeg



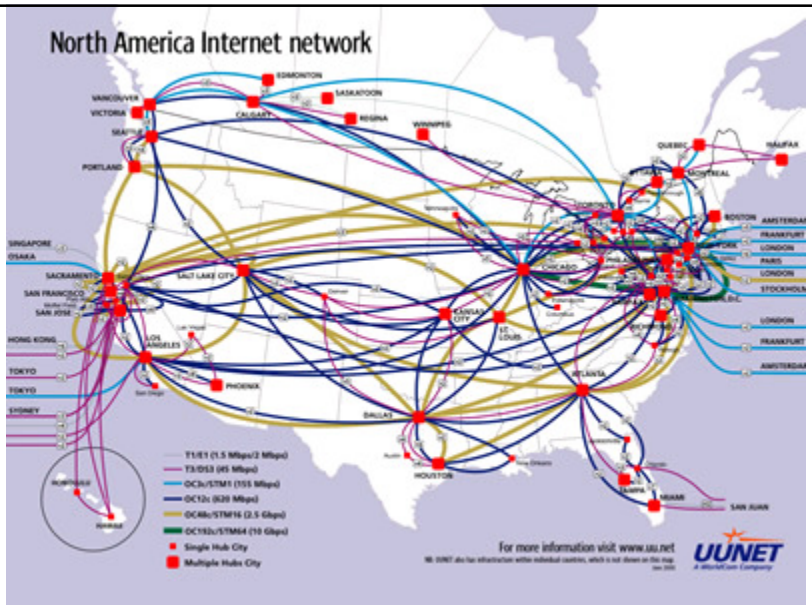
Airline Routes

http://virtualskies.arc.nasa.gov/research/tutorial/images/12routemap.gif



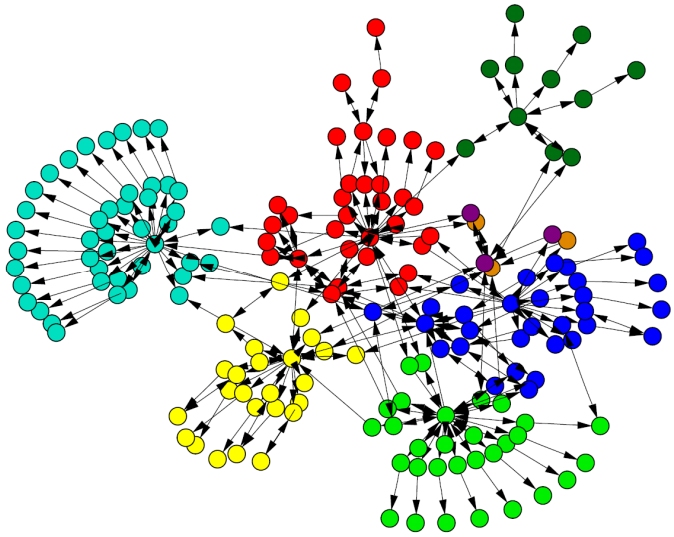
US Power Grid

http://images.encarta.msn.com/xrefmedia/aencmed/targets/maps/map/000a5302.gif



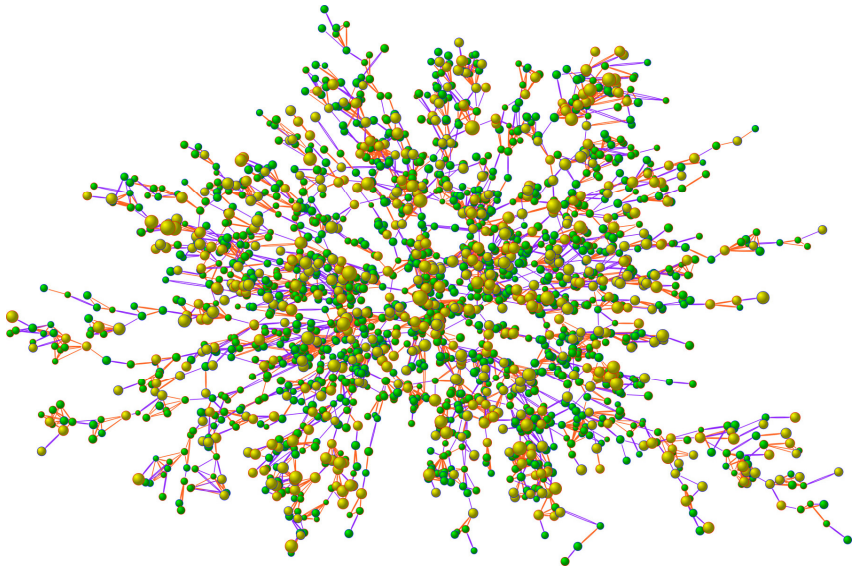
Internet

http://www.visualcomplexity.com/vc/images/270_big01.jpg



World Wide Web (small part)

From M. E. J. Newman and M. Girvin, Physical Review Letters E, 69, 026113, 2004.



Social Network

http://ucsdnews.ucsd.edu/graphics/images/2007/07-07socialnetworkmapLG.jpg

The Science of Networks

Are there properties common to all complex networks?

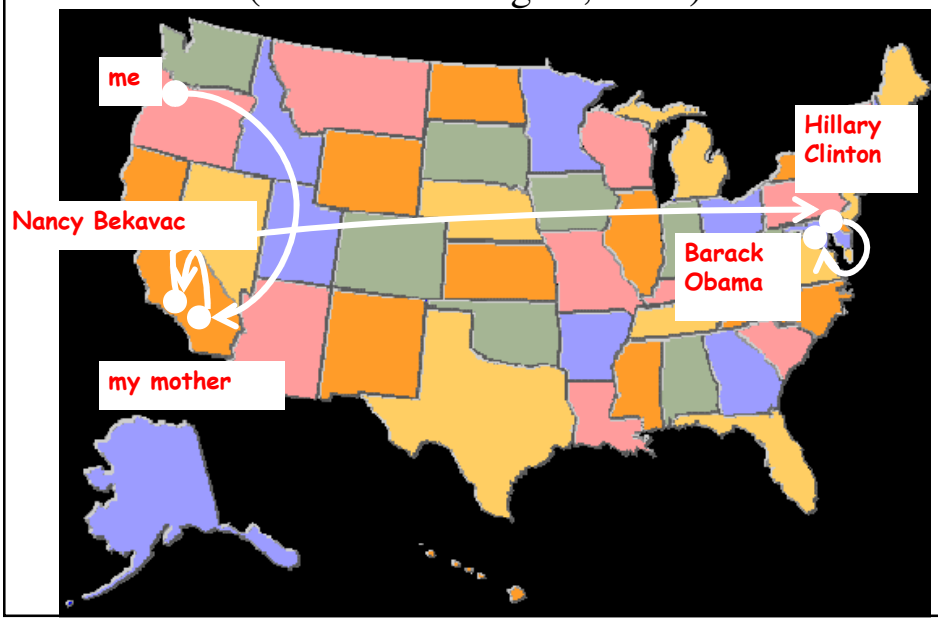
If so, why?

Observed common properties:

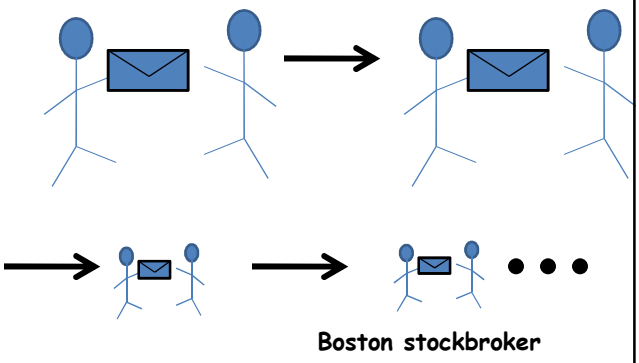
- **Small world property**
- **Scale-free structure**
- **Clustering and community structure**
- **Robustness to random node failure**
- **Vulnerability to targeted hub attacks**
- **Vulnerability to cascading failures**

Small-World Property
(Watts and Strogatz, 1998)

Small-World Property
(Watts and Strogatz, 1998)



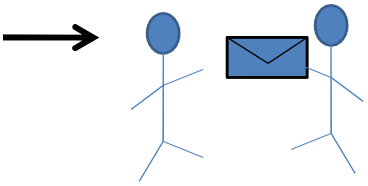
Nebraska farmer



Boston stockbroker

Stanley Milgram

On average: "six degrees of separation"



The Small-World Property (Watts and Strogatz)

The network has relatively few "long-distance" links but there are short paths between most pairs of nodes, usually created by "hubs".

Notion of *average path length*

Notion of *clustering coefficient*

Netlogo Demo

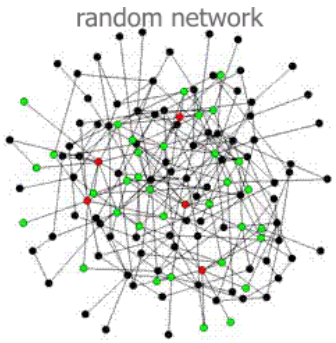
The Small-World Property (Watts and Strogatz)

The network has relatively few "long-distance" links but there are short paths between most pairs of nodes, usually created by "hubs".

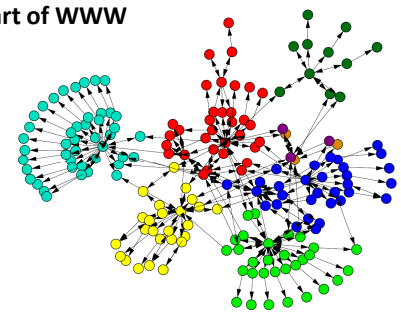
Most real-world complex networks seem to have the small-world property, and thus have short average path lengths and high clustering.

Why?

Scale-Free Structure (Albert and Barabási, 1998)

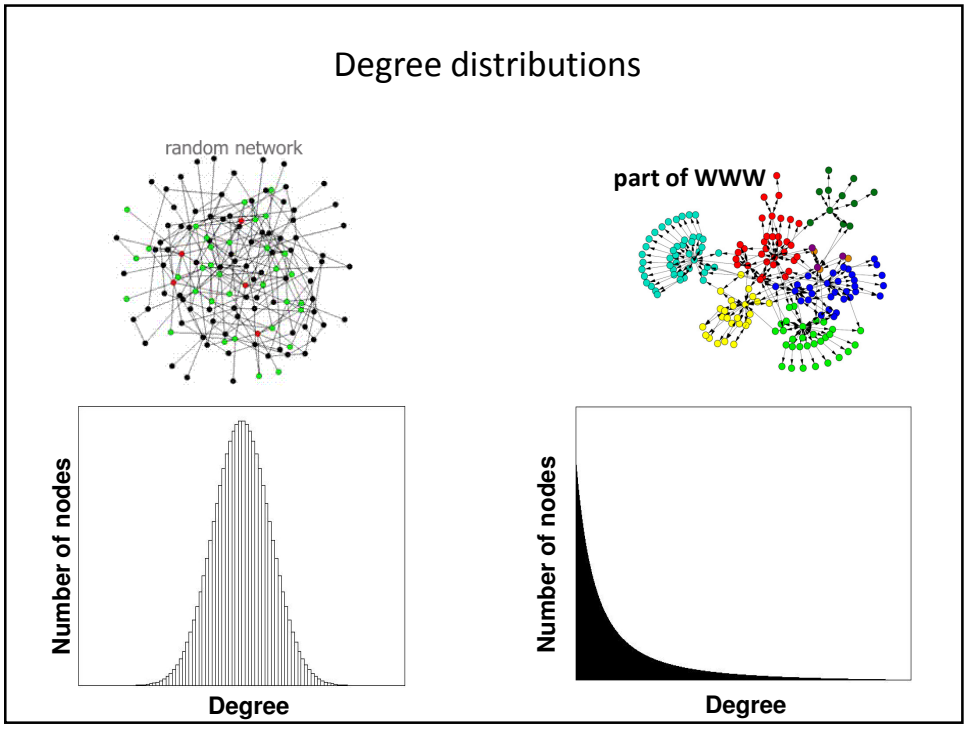


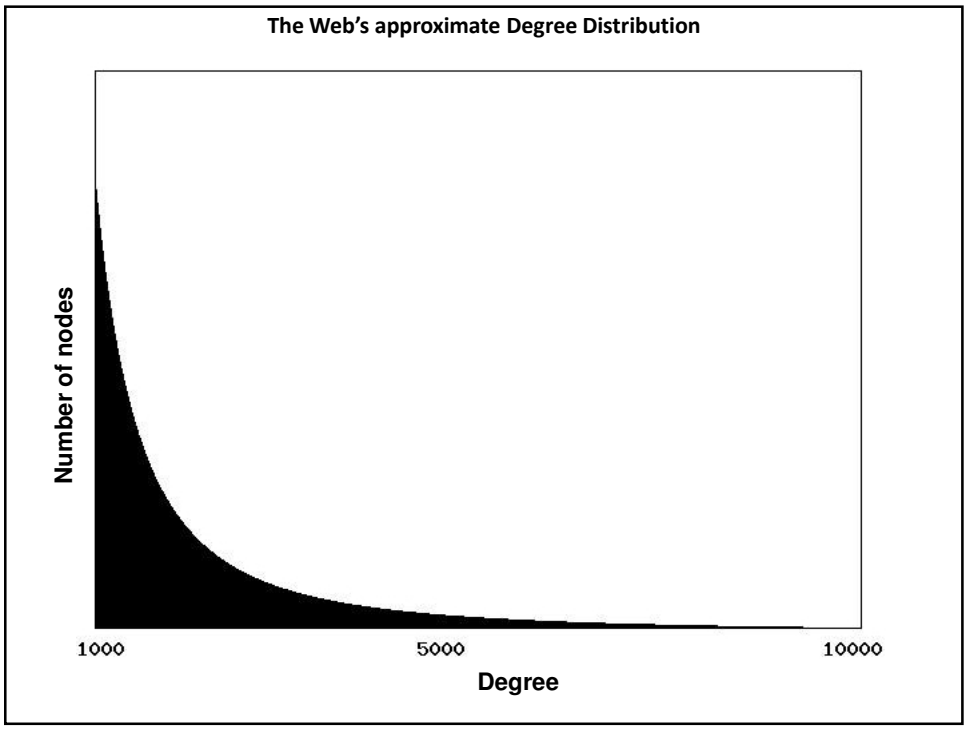
Typical structure of a randomly connected network

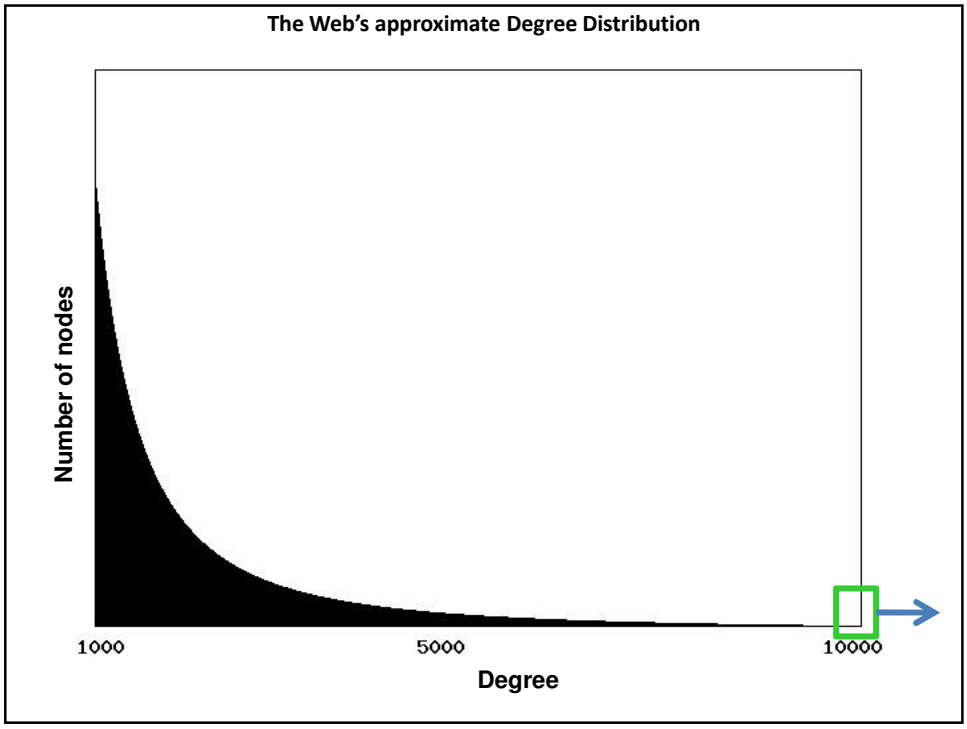


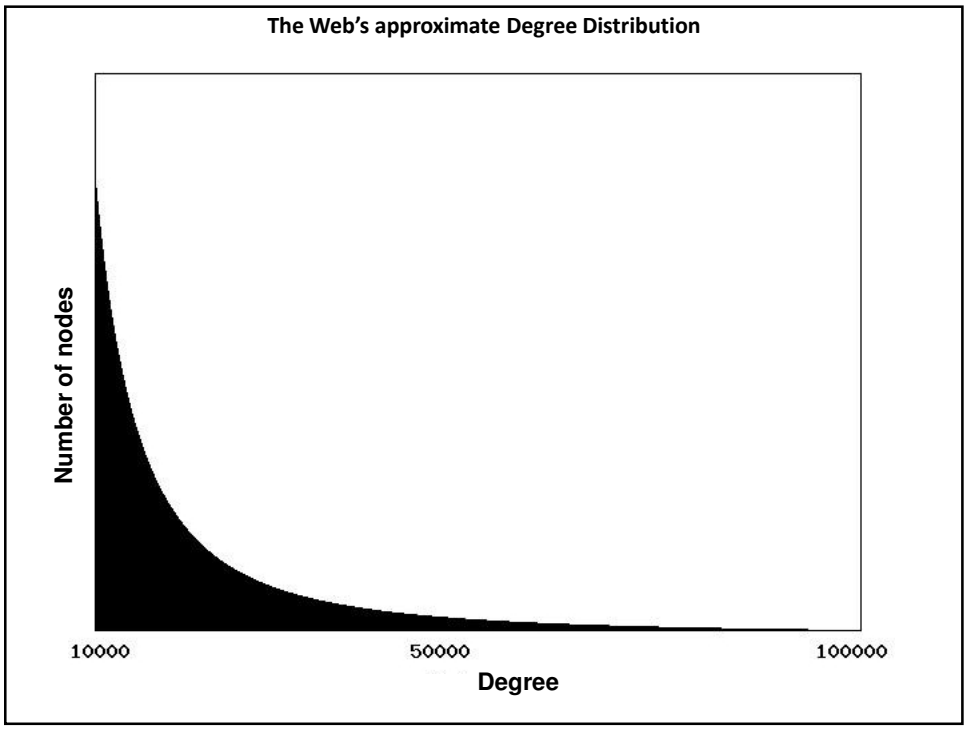
Typical structure of World Wide Web (nodes = web pages, links = links between pages)

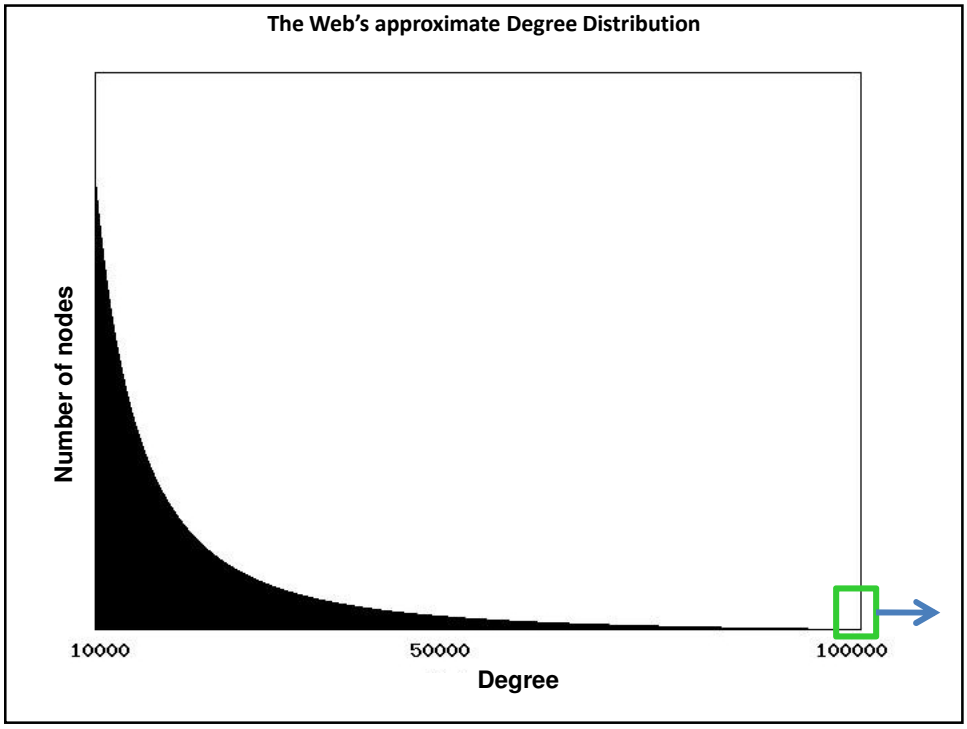
http://www.dichotomistic.com/images/random%20network.gif

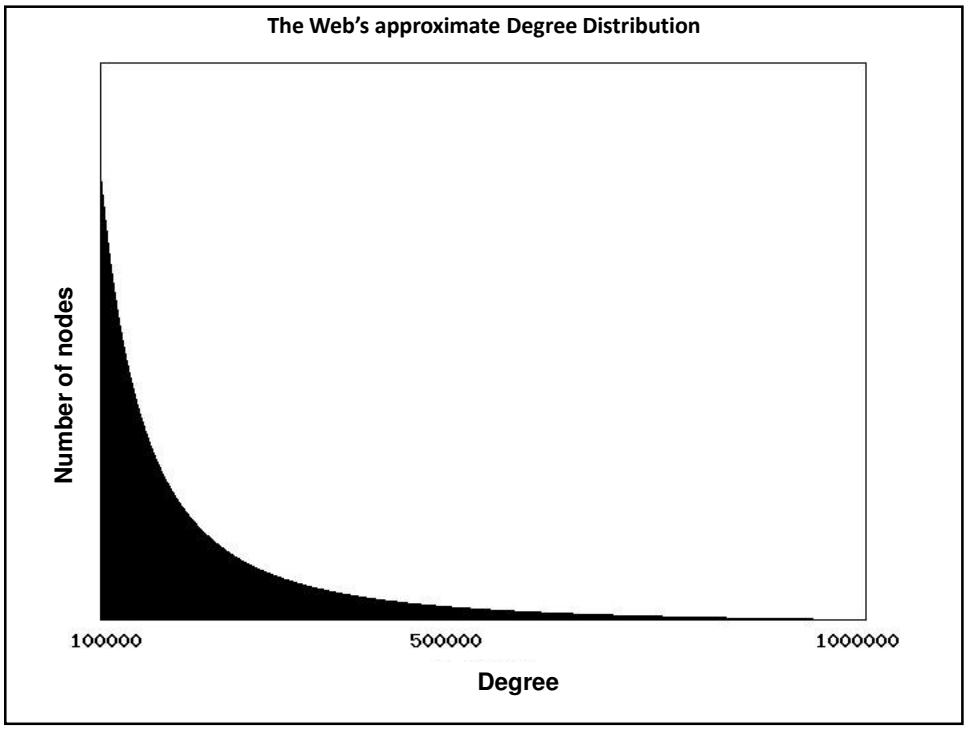

Degree distributions
random network
part of WWW
Number of nodes
Degree
Number of nodes
Degree

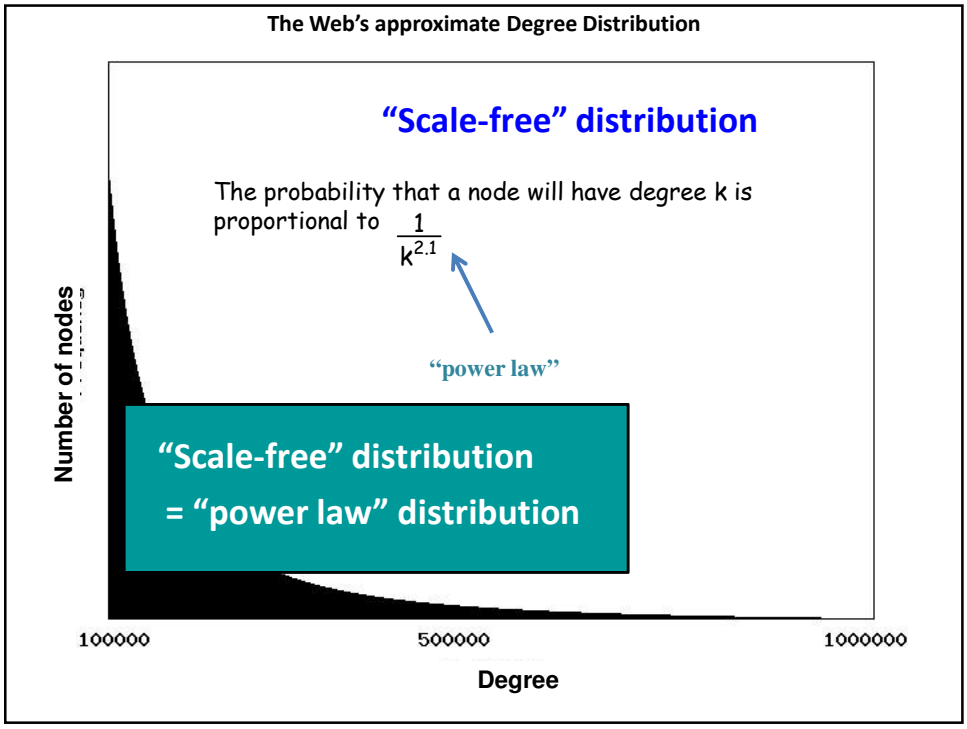

The Web's approximate Degree Distribution
Number of nodes
1000
5000
10000
Degree


The Web's approximate Degree Distribution
Number of nodes
1000
5000
10000
Degree


The Web's approximate Degree Distribution
Number of nodes
10000
50000
100000
Degree

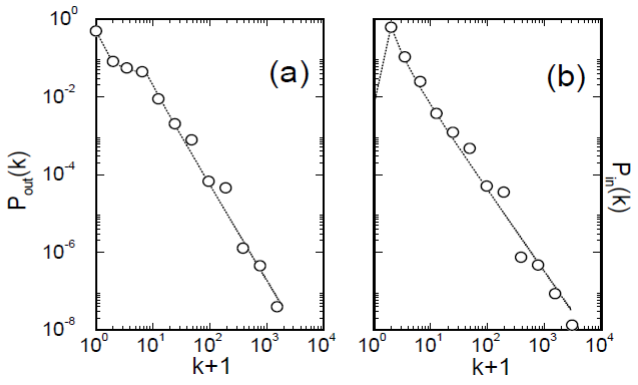

The Web's approximate Degree Distribution
Number of nodes
10000
50000
100000
Degree


The Web's approximate Degree Distribution
Number of nodes
100000
500000
1000000
Degree



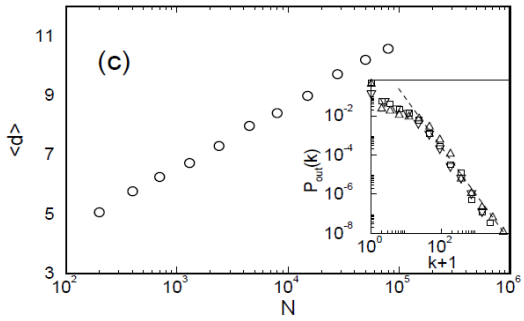
Degree distribution of the Web

- Out and in degree distributions (from sampling) with exponents 2.45 and 2.1 respectively, from Albert et al., "Diameter of the World-Wide Web".



The Web as a small-world network

From Albert et al., "Diameter of the World-Wide Web".



Path length: Average of shortest path between two documents as function of system size. <d> = 0.35 + 2.06 log(N)

"Diameter of the web"

(Albert and Barabasi)

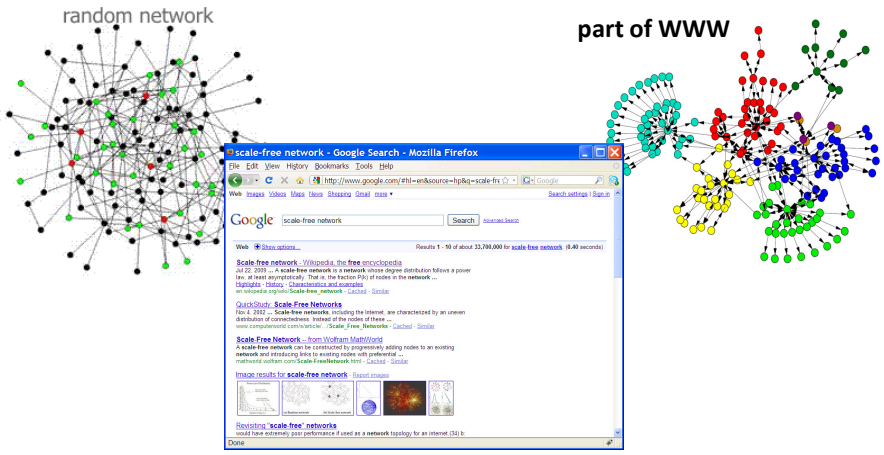
Average distance over all pairs of vertices: approx 19.

As N grows, d won't change much, given logarithmic dependence.

E.g., if web increases 1000%, d will change from 19 to 21.

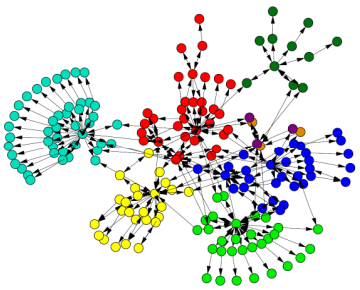
Problem: how to find shortest paths?

The scale-free structure of the Web explains why Google works so well



It also explains some of the success of other scale-free networks in nature!

Robustness of Scale-Free Networks



- **Robust** to random node failure
- **Vulnerable** to targeted "hub" failure
- If failing nodes can **cause** other nodes to fail

  Can result in ***cascading failure***

Other examples of power-laws in nature

- Magnitude vs. frequency of earthquakes
- Magnitude vs. frequency of stock market crashes
- Income vs. frequency (of people with that income)
- Populations of cities vs. frequency (of cities with that population)
- Word rank vs. frequency in English text

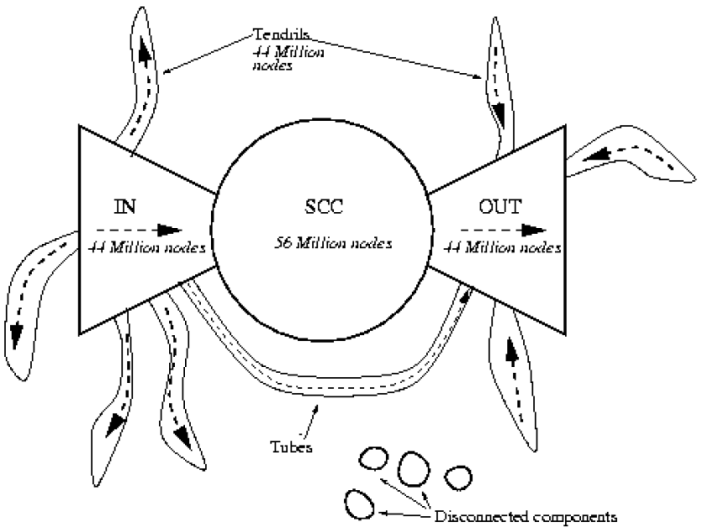
"More normal than 'normal'?

How are scale-free networks created?

Barabàsi and Albert: **Preferential attachment**

Netlogo demo

From Broder et al., Graph structure in the web, 2000

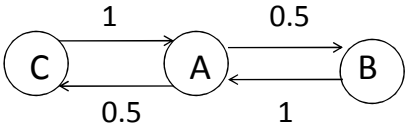


PageRank

- Named after Larry Page, patented by Stanford University
- General idea:
  - Consider surfer who begins at one page (node) and executes random walk on the graph. At each time step, surfer goes from page A to randomly chosen page that A links to, with equal probability.
  - If A does not link to any pages, or if surfer gets "bored" (with probability α) following hyperlinks, then surfer "teleports" to randomly chosen page in collection.
  - PageRank (page) = fraction of time page will be visited in this stochastic process.
  - Pages visited more often in this process are more important and thus ranked higher.

Markov Chains

- Stochastic process with *states* and *transition probabilities.*
- Probability of transitioning from state A to State B depends only on state A (current state), not on previous states.
- E.g.:



Markov Chain

$$\begin{array}{c} \\ A \\ B \\ C \end{array} \begin{array}{c} \begin{array}{ccc} A & B & C \end{array} \\ \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix} \end{array}$$

Transition Matrix

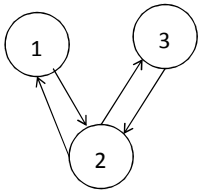
Web Surfing as a Markov Chain Process

- Random web surfer viewed as Markov chain
- State = web page
- Transition probability: probability of moving from one web page to another.

Web Surfing as a Markov Chain Process, continued

- Adjacency Matrix **A**:
  - If there is a hyperlink from page *i* to page *j*, then **A***ij* = 1, otherwise **A***ij* = 0.
- Transition Matrix **P**:
  - If a row of **A** has no 1s (i.e., no out-links), then insert 1/*N* for each element in that row in **P** (uniform teleporting probability)
  - Otherwise, divide each 1 in the row in **A** by the number of 1s in its row. (uniform probability of going to out-link)
  - Multiply the resulting matrix by (1- α) (probability of going to that linked page by *not* teleporting)
  - Add α /*N* to every entry of the resulting matrix (probability of going to that each by teleporting)

- Exercise 21.6: Consider the following web graph. What are the transition matrices for $\alpha = 0$ and 0.5?



Ergodicity in Markov Chains

- Markov chain is ergodic if there is some non-zero probability of visiting every state.

- **Theorem**: For any ergodic Markov chain with transition matrix **P**, there is a unique steady-state probability vector $\boldsymbol{\pi}$ that is the principal left eigenvector of **P**, such that if $\eta(i,t)$ is the number of visits to state i in t steps, then

$$\lim_{t\to\infty}\frac{\eta(i,t)}{t}=\pi(i),$$

where $\pi(i) > 0$ is the steady-state probability for state i.

Corollary: The "random walk with teleporting" process described above results in a unique distribution of steady state probabilities over the states of the Markov chain.

Definition:

PageRank(i) = $\pi(i)$.

Computing PageRank

- http://www.page-rank-calculator.com/

- By definition of left principal eigenvector:

$$\boldsymbol{\pi}\mathbf{P}=\lambda\boldsymbol{\pi}$$

- If $\boldsymbol{\pi}$is steady-state distribution, then

$$\boldsymbol{\pi}\mathbf{P}=1\boldsymbol{\pi}$$

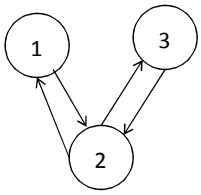
- Thus 1 is an eigenvalue of **P**. $\boldsymbol{\pi}$ gives the PageRank values of all the pages.

- How to compute $\boldsymbol{\pi}$?

Computing PageRank, continued

- Suppose alpha = 0.5. Let $\mathbf{x}_t$ be the probability distribution over the states at time t.

  Suppose surfer starts in state 1. I.e., $\boldsymbol{x}_0 = (1\ 0\ 0)$.



$$\mathbf{P} = \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 5/12 & 1/6 & 5/12 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}$$

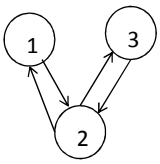
After one time step, we have $\mathbf{x}_1 = \mathbf{x}_0\,\mathbf{P} = (1/6 \quad 2/3 \quad 1/6)$

After two time steps, $\mathbf{x}_2 = \mathbf{x}_1\,\mathbf{P} = (1/3 \quad 1/3 \quad 1/3)$.

Keep going. Finally reach steady state of (5/18 4/9 5/18). **[Show this is a steady state]**

Alternative explanation from Wikipedia

- Recursive calculation:



$$\mathbf{P} = \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 5/12 & 1/6 & 5/12 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}$$

$$PageRank(i) = \sum_{j \to i} \frac{PageRank(j)}{\text{Number of outbound links from } j}$$

E.g.,

- PageRank(2) = PageRank(1) + PageRank(3)
- PageRank(1) = PageRank(2) / 2
- PageRank(3) = PageRank(2)/2

- With **damping factor** α: $PageRank(i) = \frac{(1-\alpha)}{N} + \alpha \sum_{j \to i} \frac{PageRank(j)}{\text{Number of outbound links from } j}$

where N is number of pages in the collection.

Google and PageRank

- **Google:** "PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves 'important' weigh more heavily and help to make other pages 'important'."
- Google recalculates PageRank each time it crawls the web and rebuilds its index.
- PageRank tends to favor older pages (more links)
- http://directory.google.com/

From http://www.geek.com/articles/chips/googles-pagerank-algorithm-traced-back-to-the-1940s-20100217/

Earlier forerunner to PageRank in the work of the Harvard economist Wassily Leontief:

"In 1941, Leontief published a paper in which he divides a country's economy into sectors that both supply and receive resources from each other, although not in equal measure. One important question is: what is the value of each sector when they are so tightly integrated? Leontief's answer was to develop an iterative method of valuing each sector based on the importance of the sectors that supply it. Sound familiar? In 1973, Leontief was awarded the Nobel Prize in economics for this work."

Other Uses for Page Rank

- http://www.eigenfactor.org (for journal impact)
- Ranking doctoral programs (network: departments are nodes, one node links to another if it hires faculty from that dept.)
- Food webs – species that are essential to an ecosystem

Googling Food Webs: Can an Eigenvector Measure Species' Importance for Coextinctions?

Stefano Allesina[1]*, Mercedes Pascual[2,3,4]

1 National Center for Ecological Analysis and Synthesis, Santa Barbara, California, United States of America, **2** Department of Ecology and Evolutionary Biology, Universi of Michigan, Ann Arbor, Michigan, United States of America, **3** Santa Fe Institute, Santa Fe, New Mexico, United States of America, **4** Howard Hughes Medical Institu

Abstract

A major challenge in ecology is forecasting the effects of species' extinctions, a pressing problem given current human impacts on the planet. Consequences of species losses such as secondary extinctions are difficult to forecast because species are not isolated, but interact instead in a complex network of ecological relationships. Because of their mutual dependence, the loss of a single species can cascade in multiple coextinctions. Here we show that an algorithm adapted from the one Google uses to rank web-pages can order species according to their importance for coextinctions, providing the sequence of losses that results in the fastest collapse of the network. Moreover, we use the algorithm to bridge the gap between qualitative (who eats whom) and quantitative (at what rate) descriptions of food webs. We show that our simple algorithm finds the best possible solution for the problem of assigning importance from the perspective of secondary extinctions in all analyzed networks. Our approach relies on network structure, but applies regardless of the specific dynamical model of species' interactions, because it identifies the subset of coextinctions common to all possible models, those that will happen with certainty given the complete loss of prey of a given predator. Results show that previous measures of importance based on the concept of "hubs" or number of connections, as well as centrality measures, do not identify the most effective extinction sequence. The proposed algorithm provides a basis for further developments in the analysis of extinction risk in ecosystems.

Googling Food Webs: Can an Eigenvector Measure Species' Importance for Coextinctions?

Stefano Allesina[1]*, **Mercedes Pascual**[2,3,4]

1 National Center for Ecological Analysis and Synthesis, Santa Barbara, California, United States of America, **2** Department of Ecology and Evolutionary Biology, Universi of Michigan, Ann Arbor, Michigan, United States of America, **3** Santa Fe Institute, Santa Fe, New Mexico, United States of America, **4** Howard Hughes Medical Institu

Abstract

A major challenge in ecology is forecasting the effects of species' extinctions, a pressing problem given current human impacts on the planet. Consequences of species losses such as secondary extinctions are difficult to forecast because species are not isolated, but interact instead in a complex network of ecological relationships. Because of their mutual dependence, the loss of a single species can cascade in multiple coextinctions. Here we show that an algorithm adapted from t oviding the se the gap betwe simple algorit condary extinc specific dynam models, those revious measu do not identif s in the analys

"Here we show that an algorithm adapted from the one Google uses to rank web-pages can order species according to their importance for coextinctions, providing the sequence of losses that results in the fastest collapse of the network."

How to spoof/manipulate PageRank

Hubs and Authorities (HITS Algorithm)

- Proposed by Jon Kleinberg (Cornell) at same time Brin and Page were developing PageRank
- HITS: Hyperlinked-induced topic search
- Supposedly used by Teoma and Ask.com

Hubs and Authorities Main ideas

Each node has a hub score and an authority score

Hub: Web site that points to a lot of good authorities

Authority: Web site that is pointed to by a lot of good hubs

Circular definition ⟹ iterative computation

$$h(v) \leftarrow \sum_{v \mapsto y} a(y)$$

$$a(v) \leftarrow \sum_{y \mapsto v} h(y)$$

where $a \mapsto b$ means a links to b

Hubs and Authorities
How to compute scores

Let **h** = vector of hub scores of web pages in collection
Let **a** = vector of authority scores of web pages in collection

Let **A** denote the adjacency matrix of the web pages in collection

Then: $\mathbf{h} \leftarrow \mathbf{Aa}$ $\quad \mathbf{a} \leftarrow \mathbf{A}^T\mathbf{h}$ and $\mathbf{h} \leftarrow \mathbf{AA}^T\mathbf{h}$ $\quad \mathbf{a} \leftarrow \mathbf{A}^T\mathbf{Aa}$

Can turn these into eigenvector equations:

$$\lambda_h \mathbf{h} = \left(\mathbf{AA}^T\right)\mathbf{h}$$
$$\lambda_h \mathbf{a} = \left(\mathbf{A}^T\mathbf{A}\right)\mathbf{a}$$

Hubs and Authorities
How to compute scores, continued

- HITS Algorithm:
  - Assemble the target subset of web pages from a given query, form the graph induced by their hyperlinks, and compute $\mathbf{AA}^T$ and $\mathbf{A}^T\mathbf{A}$
  - Compute the principal eigenvectors of $\mathbf{AA}^T$ and $\mathbf{A}^T\mathbf{A}$ to form the vector of hub scores **h** and the vector of authority scores **a**.
  - Output the top-scoring hubs and top-scoring authorities

Choosing the subset of the web

(One proposed method, described in textbook)

1. Given a query , use a text index to get all pages containing the terms of the query. Call this the *root* set of pages.

2. Build the *base* set of pages to include the root set as well as any page that either links to a page in the root set, or is linked to by a page in the root set.

Use the base set to compute hub and authority scores.

Differences between PageRank and HITS

- PageRank computers one score per document. HITS computers two.

- PageRank executed at indexing time, HITS executed at query time. Hub and authority scores are query-specific, whereas PageRank scores are query-independent.

- PageRank is assigned to all documents in collection. HITS scores are assigned only to relevant subset.

In-Class Exercises

Draw a picture of a Web structure containing nodes with high hub score and high authority score.

Draw two different pictures of Web structures containing nodes with high PageRank.