Web Dynamics

Part 2 – Modeling static and evolving graphs

2.1 The Web graph and its static properties

2.2 Generative models for random graphs

2.3 Measures of node importance

Notation: Graphs

We will drop G when the graph is clear from the context.

- G=(V(G),E(G))
  - directed graph: E(G)$\subseteq$V(G)xV(G)
  - undirected graph: E(G) $\subseteq$\{\{v,w\} $\subseteq$V(G)\}
- Degrees of nodes in directed graphs:
  - indegree of node n: indeg(n)=|\{(v,w)$\in$E(G):w=n\}|
  - outdegree of node n: outdeg(n)=|\{(v,w)$\in$E(G):v=n\}|
- Degree of node n in undirected graph:
  - deg(n)=|\{ e$\in$E(G):n$\in$e\}|
- Distributions of degree, indegree, outdegree

$$P_{deg,G}(k) = \frac{|\{n \in V(G) : \mathrm{deg(n)} = \mathrm{k}\}|}{|V(G)|}$$

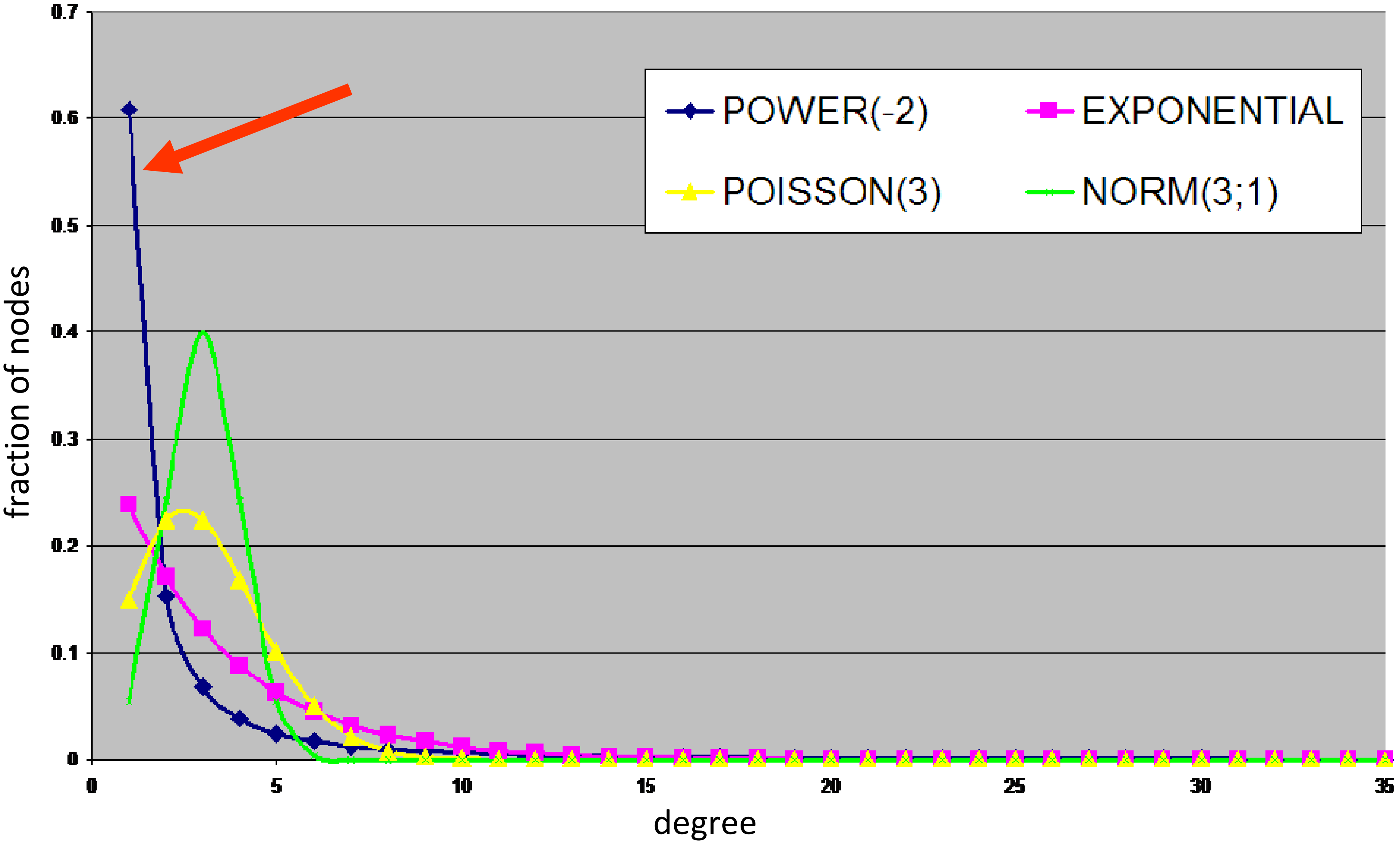
Web Graph W

- Nodes are URLs on the Web
  - No dynamic pages, often only HTML-like pages
- Edges correspond to links
  - directed edges, sparse
- Highly dynamic, impossible to grab snapshot at any fixed time
  $\Rightarrow$ large-scale crawls as approximation/samples

Degree distributions

- Assume the average indegree is 3, what would be the shape of $P_{in,W}$?

Degree distributions



Power Law Distributions

Distribution P(k) follows power law if

$$P(k) = C \cdot k^{-\beta}$$

for real constant C>0 and real coefficient β>0

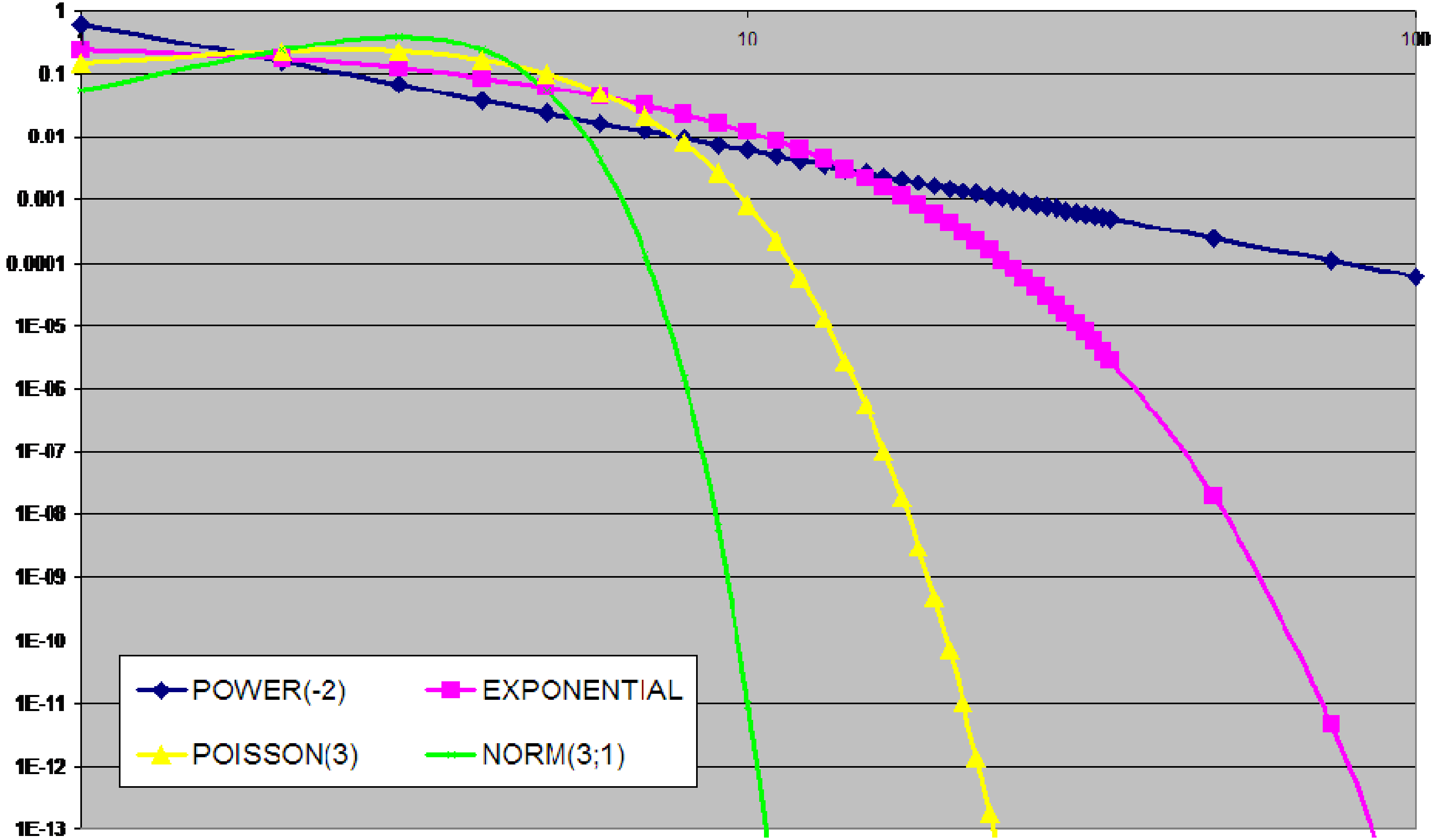
(needs normalization to become probability distribution)

Moments of order *m* are finite iff $\beta > m+1$:

$$E[X^m] = \sum_{k=1}^{\infty} k^m \cdot P(k) = \sum_{k=1}^{\infty} C \cdot k^{m-\beta} = C \cdot \zeta(\beta - m)$$

Heavy-tailed distribution: P(k) decays *polynomially* to 0

Power-Law-Distributions in log-log-scale

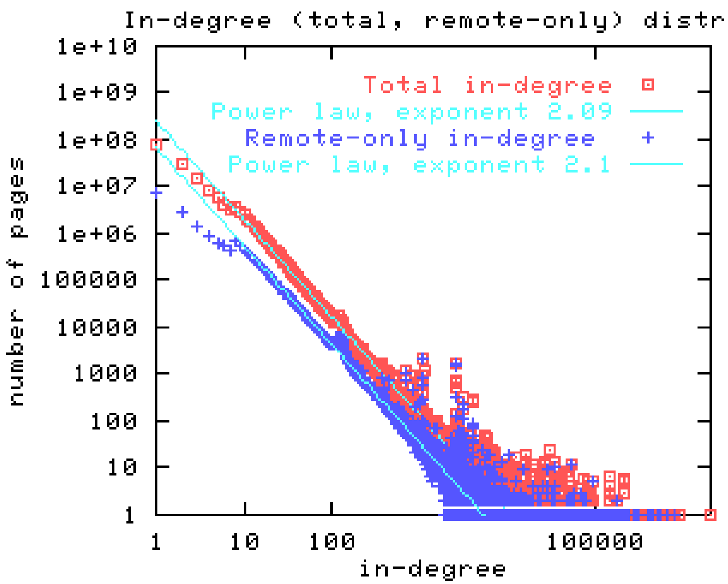


Parameter fitting in loglog-scale (fit linear function)

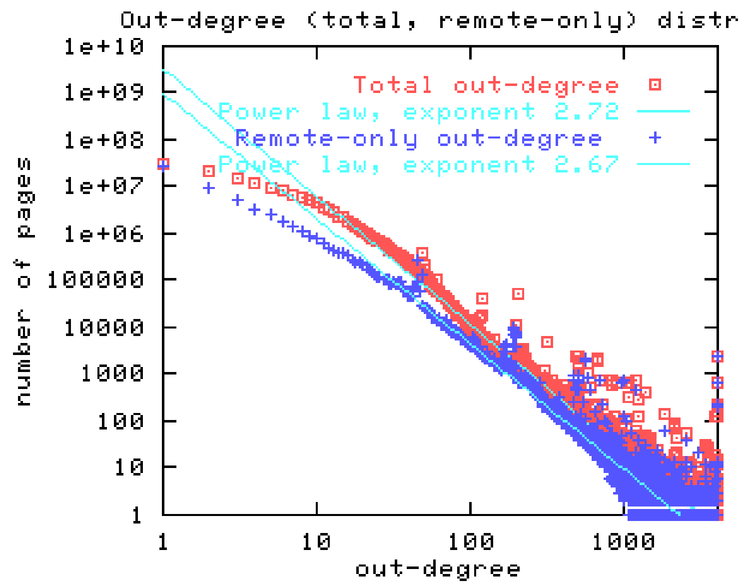
Degree distributions of the Web

Based on an Altavista crawl in May 1999

(203 million urls, 1466 million links)



$\beta = 2.1$



$\beta = 2.72$

A. Broder et al.: Grpah structure in the Web, Computer Networks 33:309—320, 2000

Examples for Power Laws in the Web

- Web page sizes
- Web page access statistics
- Web browsing behavior
- Web page connectivity
- Web connected components size

More graphs with Power-Law degrees

- Connectivity of Internet routers and hosts
- Call graphs in telephone networks
- Power grid of western United States
- Citation networks
- Collaborators of Paul Erdös
- Collaboration graph of actors (IMDB)

Scale-Freeness

Scaling *k* by a constant factor yields a proportional change in *P(k)*, independent of the absolute value of *k*:

$$P(ak) = C \cdot (ak)^{-\beta} = C \cdot a^{-\beta} \cdot k^{-\beta} = a^{-\beta} \cdot P(k)$$

(similar to 80/20 or 90/10 rules)

Additionally: results often independent of graph size (Web or single domain)

Zipfian vs. Power-Law

Zipfian distribution:

Power-law distribution of ranks, not numbers

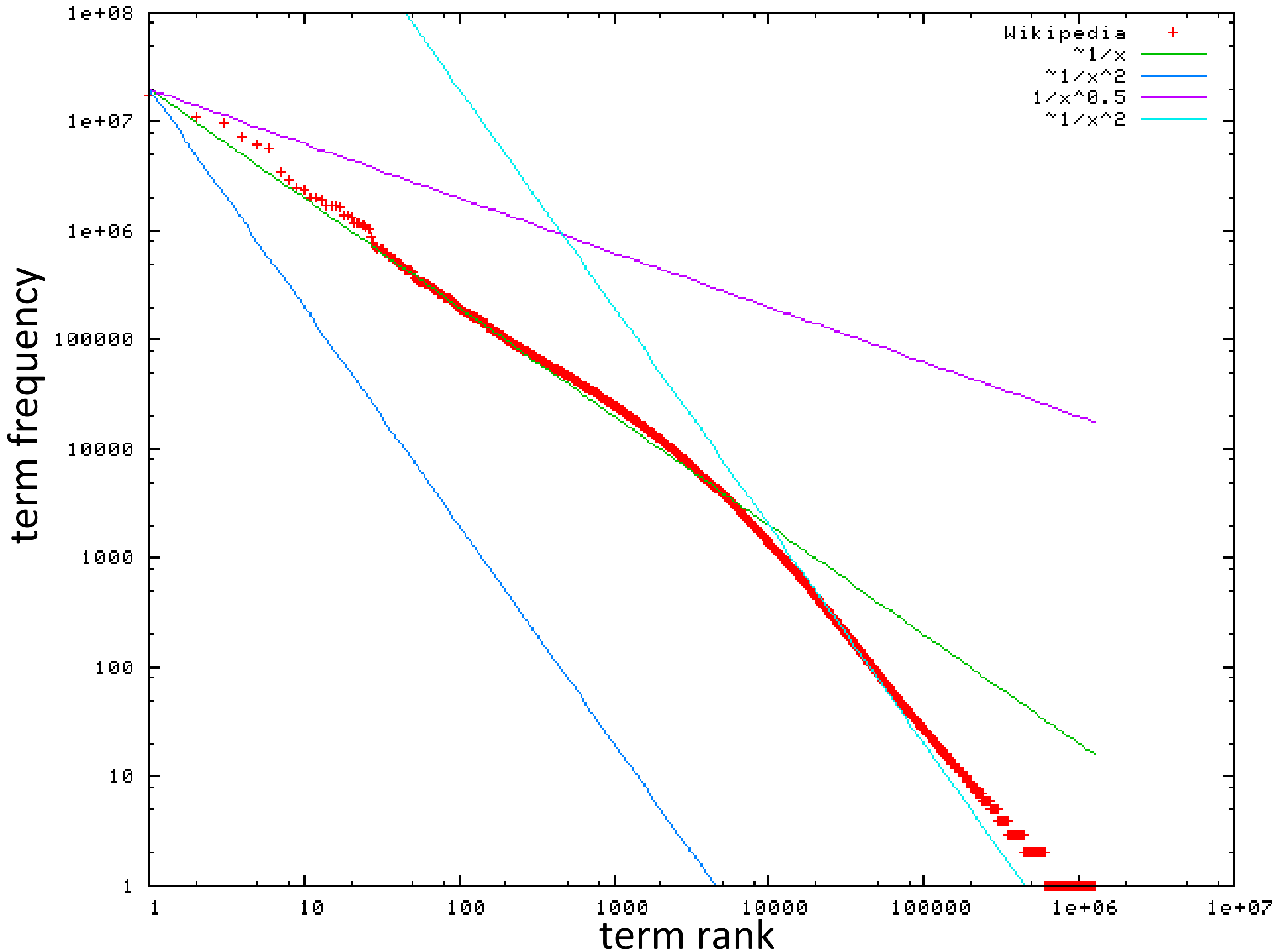
- Input: map item→value (e.g., terms and their count)
- Sort items by descending value (any tie breaking)
- Plot (k, value of item at position k) pairs and consider their distribution

Important example: Frequency of words in large texts (but: also occurs in completely **random** texts)

Other related Law:

- Benford‘s Law: distribution of first digits in numbers
- Heaps‘ Law: number of distinct words in a text

Example: Term distribution in Wikipedia



http://en.wikipedia.org/wiki/File:Wikipedia-n-zipf.png

Most popular words are “the”, “of” and “and” (so-called “stopwords”)

Diameters

How many clicks away are two pages?

For two nodes $u,v \in V$:

$d(u,v)$ minimal length of a path from u to v

Scale-free graphs: d has Normal distribution (Albert, 1999)

- Average path length
  - $E[d]=O(\log n)$, n number of nodes
  - For the Web: $E[d] \sim 0.35 + 2.06 * \log_{10} n$ (avg 21 hops distance)
  - Undirected: $O(\ln \ln n)$ (Cohen&Havlin, 2003)
- Maximal path length („diameter“)

Diameters

From Broder et al, 2000:

- only 24% of nodes are connected through directed path
- average connected directed distance: 16
- average connected undirected distance: 7

$\Rightarrow$ small world only for connected nodes!

Connected components

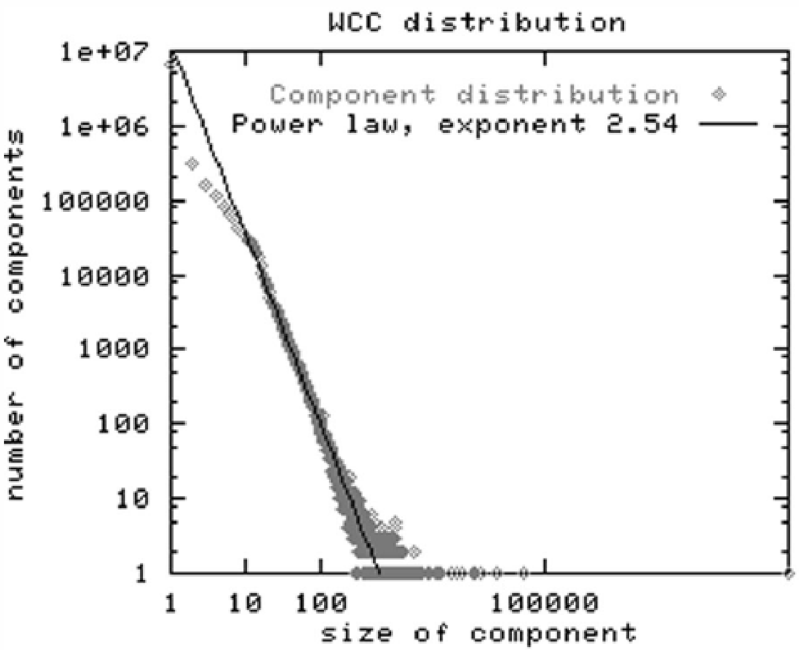


Fig. 5. Distribution of weakly connected components on the Web. The sizes of these components also follow a power law.

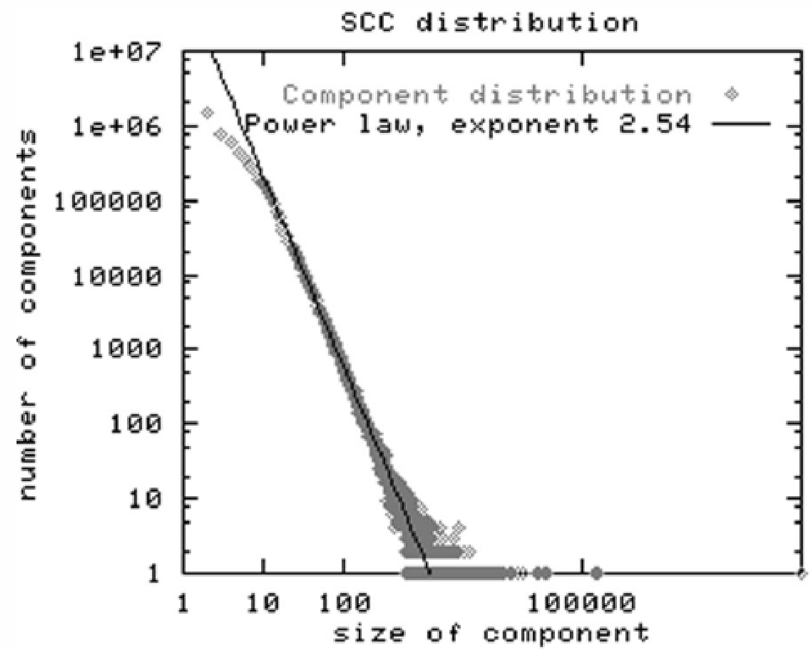


Fig. 6. Distribution of strongly connected components on the Web. The sizes of these components also follow a power law.

A. Broder et al.: Grpah structure in the Web, Computer Networks 33:309—320, 2000

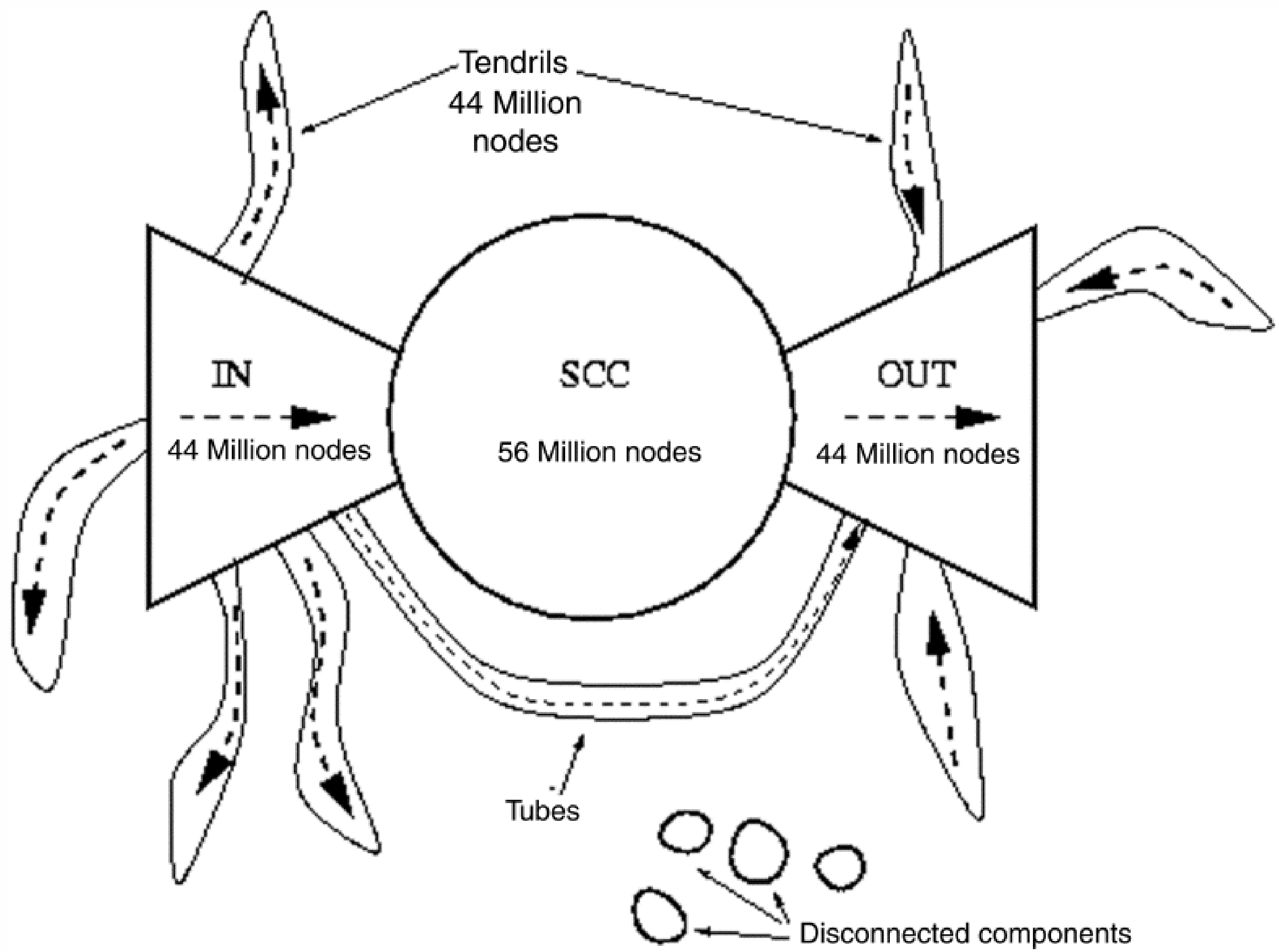
(Their sample of the) Web graph contains

- one giant weakly connected component with 91% of nodes
- one giant strongly connected component with 28% of nodes

(even after removing well-connected nodes)

Bow-Tie Structure of the Web

A. Broder et al. / Computer Networks 33 (2000) 309–320



A. Broder et al.: Grpah structure in the Web, Computer Networks 33:309—320, 2000

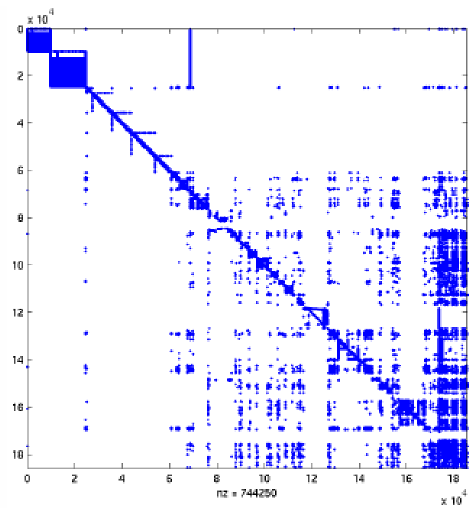
Connectivity of Power-Law Graphs

(Undirected) connectivity depends on β:

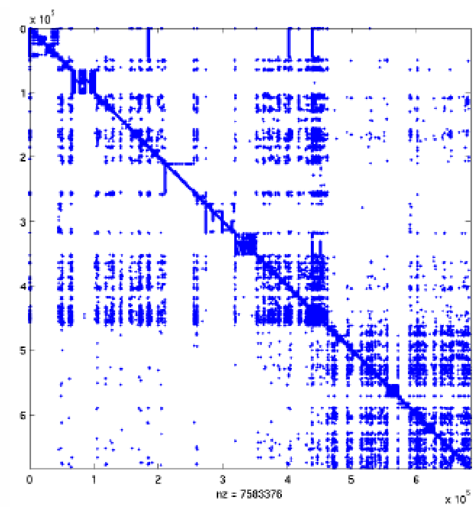
- $\beta<1$: connected with high probability
- $1<\beta<2$: one giant component of size O(n),
  all others size O(1)
- $2<\beta<\beta_0=3.4785$: one giant component of size O(n),
  all others size O(log n)
- $\beta>\beta0$: no giant component with high probability

(Aiello et al, 2001)

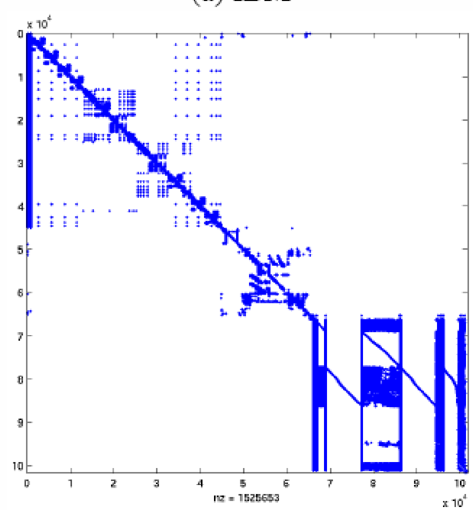
Block structure of Web links



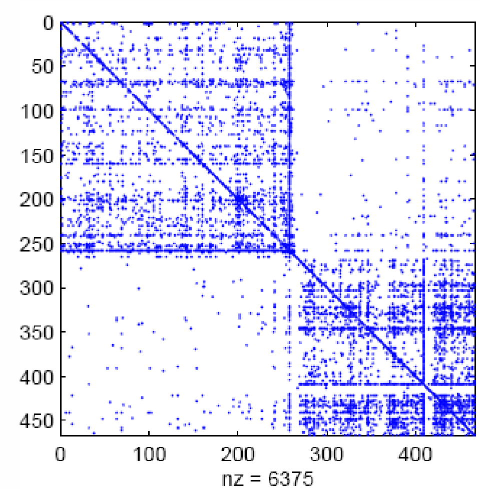
(a) IBM



(b) Stanford/Berkeley



(c) Stanford-50



(d) Stanford/Berkeley Host Graph

Figure 1: A view of 4 different slices of the web: (a) the IBM domain, (b) all of the hosts in the Stanford and Berkeley domains, (c) the first 50 Stanford domains, alphabetically, and (d) the host-graph of the Stanford and Berkeley domains.

S.D. Kamvar et al.: **Exploiting the block structure of the Web for computing Pagerank**, WWW conference, 2003

Neighborhood sizes

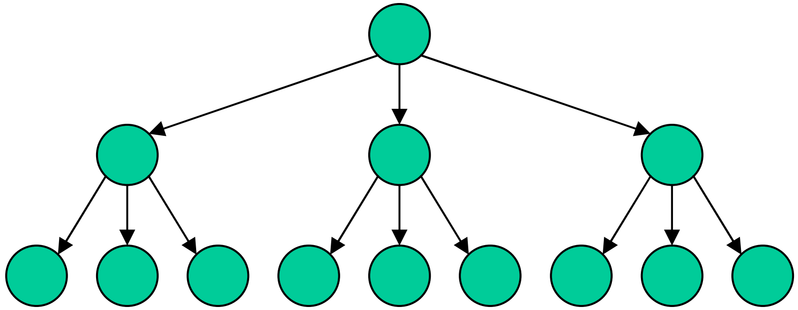
N(h): number of pairs of nodes at distance <=h

When average degree=3, how many neighbors can be expected at distance 1,2,3,…?

1 hop: 3 neighbors

2 hops: 3*3=9 neighbors

h hops: 3^h neighbors



Neighborhood sizes

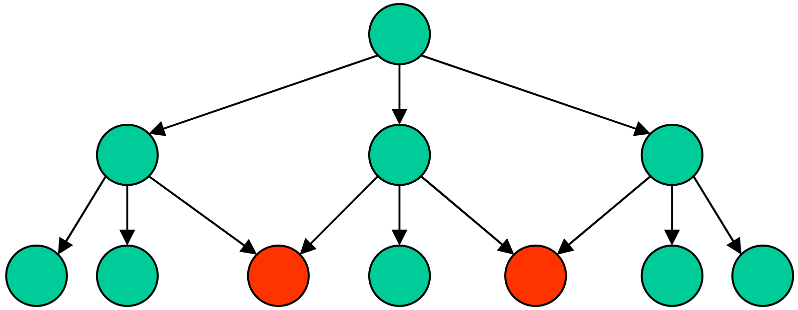
N(h): number of pairs of nodes at distance <=h

When average degree=3, how many neighbors can be expected at/up to distance 1,2,3,…?

1 hop: 3 neighbors

2 hops: 3*3=9 neighbors

h hops: 3^h neighbors

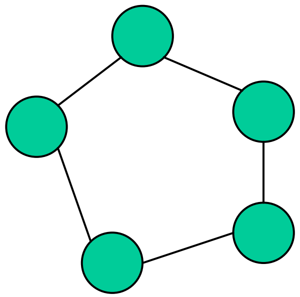


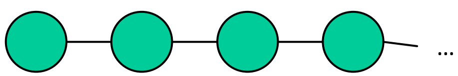
Not true in general! (duplicates ⇒ over-estimation)

$N(h) \propto h^H$ (hop exponent) [Faloutsos et al, 1999]

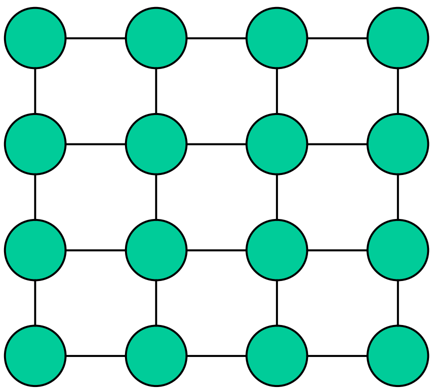
Neighborhood sizes

Intuition: H ~ „fractal dimensionality“ of graph





$N(h) \propto h^1$



$N(h) \propto h^2$

Web Dynamics

Part 2 – Modeling static and evolving graphs

2.1 The Web graph and its static properties

2.2 Generative models for random graphs

2.3 Measures of node importance

Requirements for a Web graph model

- **Online**: number of nodes and edges changes with time
- **Power-Law**: degree distribution follows power-law, with exponent $\beta>2$
- **Small-world**: average distance much smaller than O(n)
- Possibly more features of the Web graph…

Random Graphs: Erdös-Rénji

G(n,p) for undirected random graphs:

- Fix n (number of nodes)
- For each pair of nodes, independently add edge with uniform probability p

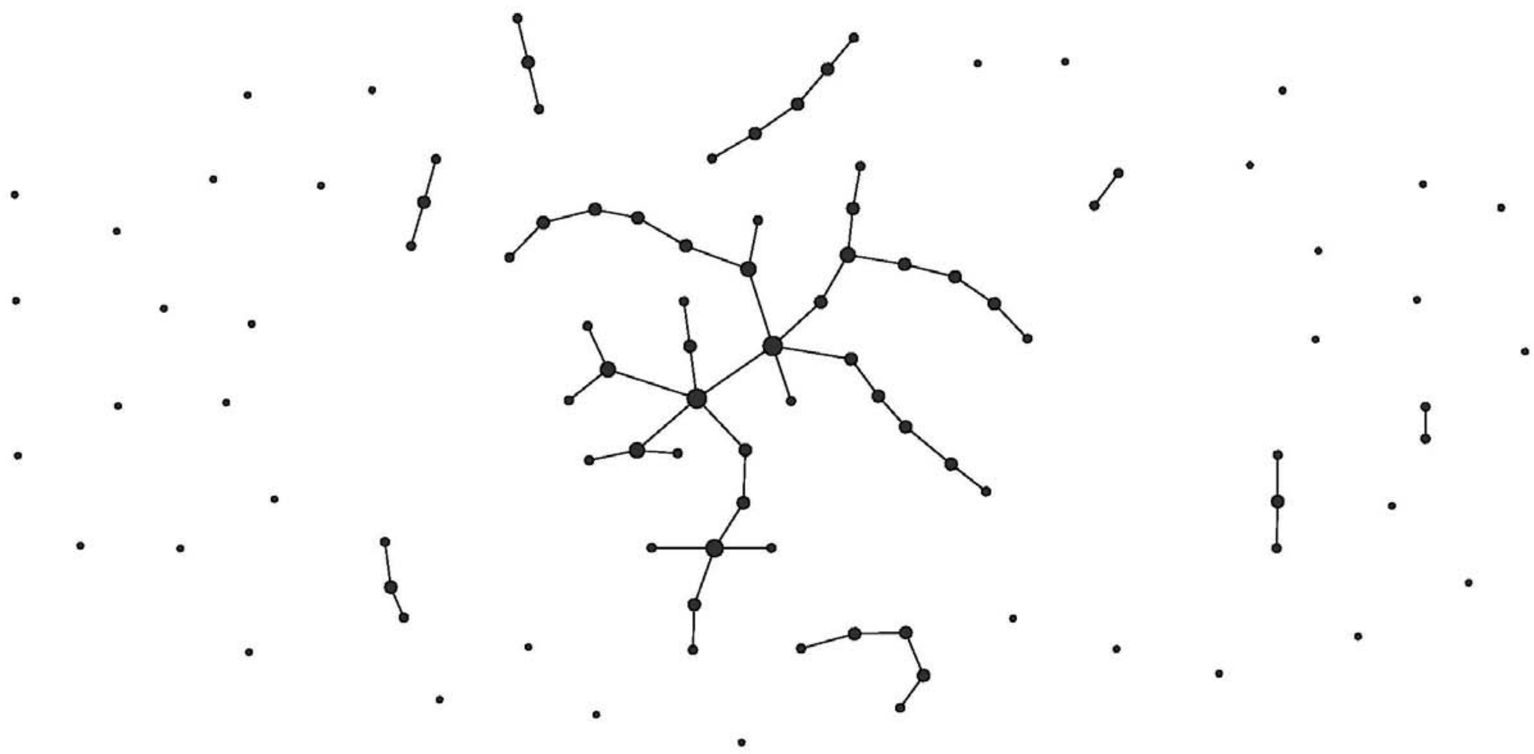
Degree distribution: binomial

$$P_{\deg}(k) = \underbrace{\binom{n-1}{k}}_{\text{Pick k out of n-1 targets}} \underbrace{p^k (1-p)^{n-1-k}}_{\text{Probability to have exactly k edges}}$$

$\frac{\ln n}{n}$ threshold for the connectivity of G(n,p)

$\Rightarrow$ cannot be used to model the Web graph

Example: p=0.01



http://upload.wikimedia.org/wikipedia/commons/1/13/Erdos_generated_network-p0.01.jpg

Preferential attachment

Idea: Barabasi&Albert, 1999

- mimic creation of links on the Web
- Links to „important“ pages are more likely than links to random pages

Generation algorithm:

- Start with set of M_0 nodes
- When new node is added, add $m \leq M_0$ random edges
  probability of adding edge to node v: $\frac{\deg(v)}{\sum \deg(w)}$

Result: Power-law degree distribution with β=2.9 for M_0=m=5
(from simulation)

Analysis of Preferential Attachment

(Using „mean field" analysis and assuming continuous time, see Baldi et al.)

After *t* steps: M_0+t nodes, *tm* edges

Consider node *v* with $k_v(t)$ edges after step *t*

$$k_v(t+1) - k_v(t) = m\frac{k_v(t)}{2mt} = \frac{k_v(t)}{2t}$$ (considering expectations, allowing multiple edges)

$$\frac{\partial k_v}{\partial t} = \frac{k_v}{2t}$$ (assuming continous time, considering differential equation)

with initial condition $k_v(t_v) = m$ (t_v: time when *v* was added)

This can be solved as

$$k_v(t) = m\sqrt{\frac{t}{t_v}}$$ (older nodes grow faster than younger ones)

Further analysis shows that $P(k) = \frac{2m^2}{k^3}$

Properties and extensions

- Diameter of generated graphs:
  - O(log n) for m=1
  - O(log n/log logn) for m$\geq$2
- Extension to directed edges:
  - randomly choose direction of each added edge
  - consider indegree and outdegree for edge choice
- Extensions to generate different distributions (where $\beta \neq 3$): mixtures of operations
  - Allow addition of edges between existing nodes
  - Allow rewiring of edges
- Extensions for node and edge deletion required

Copying

Idea: Kleinberg et al., 1999

- mimic creation of *pages* on the Web
- links are partially copied from existing pages

Generation algorithm:

- When new node is added, pick random (uniform) existing node *u* and add *d* edges as follows
  - Add edge to random (uniform) node with probability p
  - Copy random (uniform) existing edge from u with probability 1-p

Prefers nodes with high indegree (similar to preferential attachment)

Generates Power-law degree distribution with $\beta = \frac{2-p}{1-p}$

Other Generative Models

- Watts and Strogatz model:
  - Fix number of nodes n and degree k
  - Start with a regular ring lattice with degree k
  - Iterate over nodes, rewire edge with probability p
  - $\Rightarrow$ Degree distribution similar to random graph (for p>0), infeasible to model the Web graph
- Growth-Deletion Models:
  - Generative model (like PA or Copying)
  - Generate new node + *m* PA-style edges with probability p_1
  - Generate m PA-style edges with probability p_2
  - Delete existing node (uniform, random) with probability p_3
  - Delete m edges (uniform, random) with probability $1-p_1-p_2-p_3$

  Generates power-law degree distribution with $\beta = 2 + \frac{p_1 + p_2}{p_1 + 2p_2 - p_3 - p_4}$

Web Dynamics

Part 2 – Modeling static and evolving graphs

2.1 The Web graph and its static properties

2.2 Generative models for random graphs

2.3 Measures of node importance

More networks than just the Web

- Citation networks (authors, co-authorship)
- Social networks (people, friendship)
- Actor networks (actors, co-starring)
- Computer networks (computers, network links)
- Road networks (junctions, roads)

Characteristics are similar to the Web:

- Degree distribution
- (strongly, weakly) connected components
- Diameters
- ***Centrality of nodes***: how important is a node

Assume undirected graphs for the moment

Clustering: Edge density in neighborhood

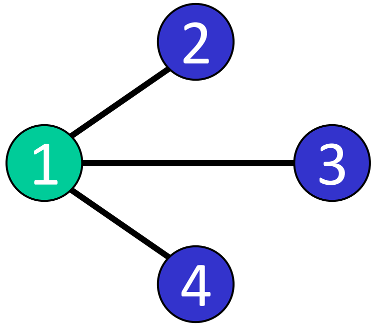
For each node *v* having at least two neighbors:

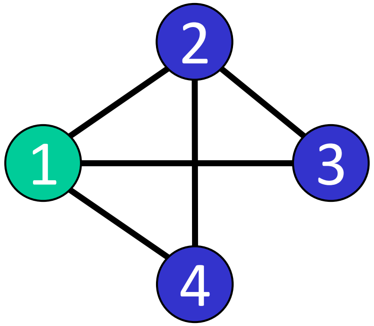
$$C^v = \frac{\left|\{\{j,k\} \in E : \{v,j\} \in E \wedge \{v,k\} \in E\}\right|}{\frac{\deg(v)(\deg(v)-1)}{2}}$$

For each node *v* having less than two neighbors:

$$C^v = 0$$

Clustering index of the network: $C = \frac{1}{|V|}\sum_{v \in V} C^v$





Degree centrality

General principle:

Nodes with many connections are important.

$$C_D(v) = \frac{\deg(v)}{|V| - 1}$$

But: too simple in practice, link targets/sources matter!

Closeness centrality

Total distance for a node *v*:

$$\sum_{w \in V} d(v, w)$$

Closeness is defined as:

$$C_C(v) = \frac{1}{\sum_{w \in V} d(v,w)}$$

Helps to find central nodes w.r.t. distance

(e.g., useful to find good location for service stations)

But: what happens with nodes that are (almost) isolated?

Assumes connected graph

Betweenness centrality

Centrality of a node *v*:

- which fraction of shortest paths through *v*
- Probability that an arbitrary shortest path passes through *v*

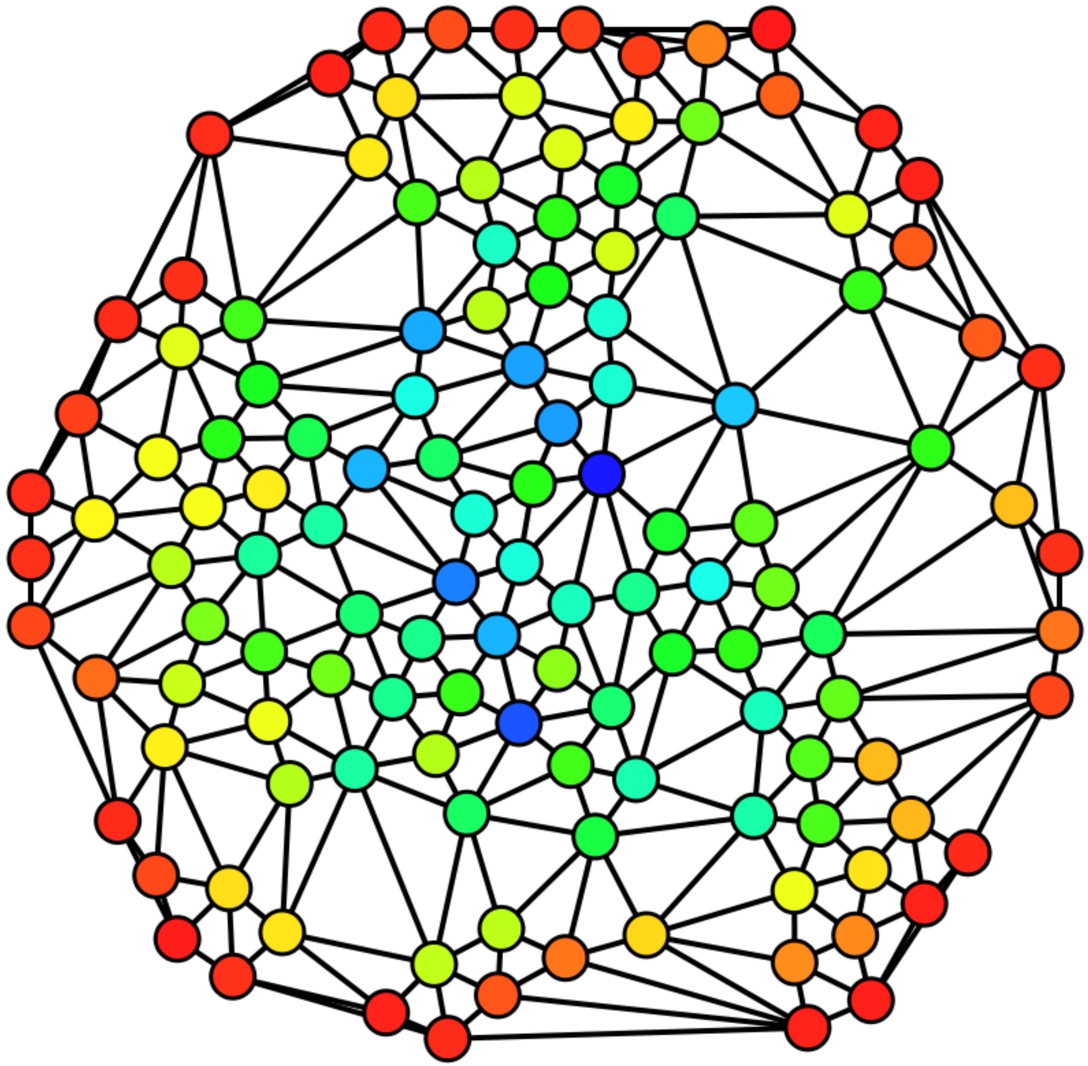
Number of shortest paths between *s* and *t*: σ_{st}

Number of shortest paths between *s* and *t through v*: $\sigma_{st}(v)$

Betweenness of node *v:* $C_B(v) = \sum_{s \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$

Can be computed in O(|V|·|E|) using per-node BFS plus clever tricks (to account for overlapping paths) [Brandes,2001]

Example: Betweenness

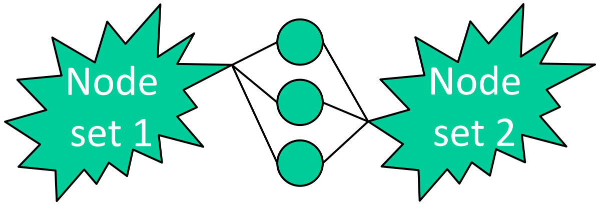


http://en.wikipedia.org/wiki/File:Graph_betweenness.svg

red=0, blue=max

Betweenness: Properties & Extensions

- Node with high betweenness may be crucial in communication networks:
  - May intercept and/or modify many messages
  - Danger of congestion
  - Danger of breaking connectivity if it fails
- But: No information how messages really flow!
- Extension: take network flow into account („flow betweenness“)



Authority Measures for the Web

Goal:

Determine **authority** (prestige, importance) of a page

with respect to

- volume
- significance
- freshness
- authenticity

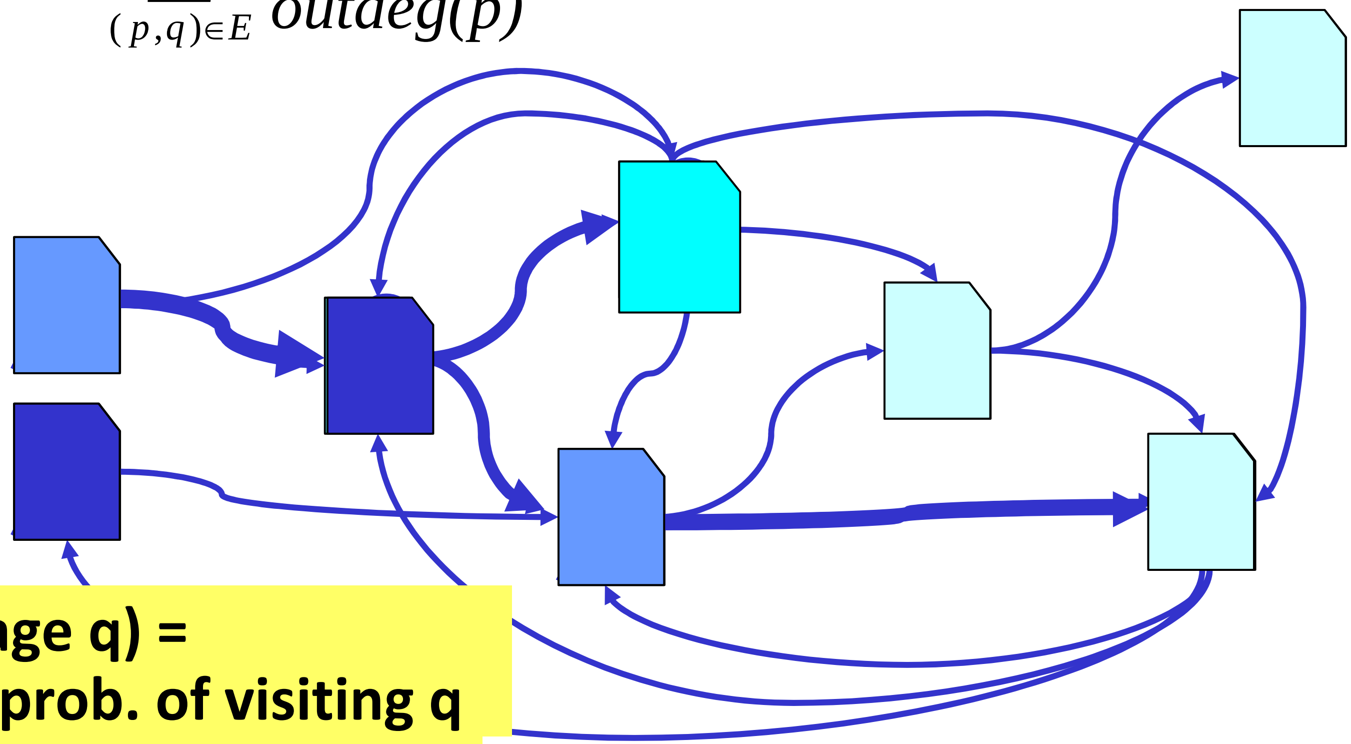
of its information content

Approximate authority by (modified) centrality measures in the (directed) Web graph

PageRank

<u>Idea:</u> incoming links are endorsements & increase page authority, authority is higher if links come from high-authority pages

$$PR(q) = \frac{\varepsilon}{|V|} + (1-\varepsilon) \cdot \sum_{(p,q) \in E} \frac{PR(p)}{outdeg(p)}$$



Authority (page q) =
stationary prob. of visiting q

Random walk: uniformly random choice of links + random jumps

PageRank

Input: directed Web graph $G=(V,E)$ with $|V|=n$ and
adjacency matrix E: $E_{ij} = 1$ if $(i,j) \in E$, 0 otherwise

Random surfer page-visiting probability after i +1 steps:

$$p^{(i+1)}(y) = r_y + \sum_{x=1..n} C_{yx}\, p^{(i)}(x)$$

$$p^{(i+1)} = r + C\, p^{(i)}$$

with conductance matrix C:
$C_{yx} = (1-\varepsilon)E_{xy} / \text{outdeg}(x)$
and random jump vector r:
$r_y = \varepsilon/n$

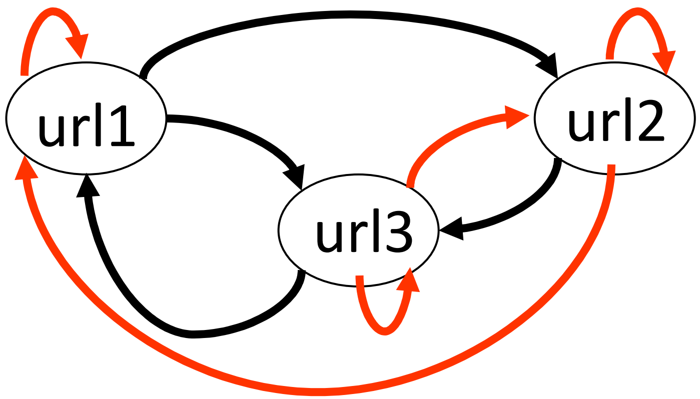
Finding solution of fixpoint equation suggests **power iteration:**
initialization: $p^{(0)}(y) = 1/n$ for all y
repeat until convergence (L_1 or L_∞ of diff of $p^{(i)}$ and $p^{(i+1)}$ < threshold)

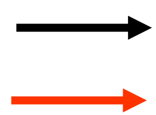
$p^{(i+1)} := r + Cp^{(i)}$

(typically ~50 iterations until convergence of top authorities)

PageRank: Foundations

Random walk can be cast into ergodic Markov chain:





Transition probability (from state i to state j):

$$p_{i,j} = \underbrace{\frac{\varepsilon}{n^2}}_{\text{random jump } i \to j} + \underbrace{(1-\varepsilon)\frac{E_{i,j}}{outdeg(i)}}_{\text{move along link}}$$

Probability $\pi_i^{(t+1)}$ for being in state i in step t+1:

$$\pi_i^{(t+1)} = \sum_n p_{ji} \cdot \pi_j^{(t)}$$

$\Rightarrow$ Fixpoint equation: $\pi = P\pi$ ($\sum \pi_i = 1$)

PageRank: Extensions

Principle: Adapt random jump probabilities

- **Personal PageRank**: Favour pages with „good“ content (personal bookmarks, visited pages)
- **Topic-specific PageRank**:
  - Fix set of topics
  - For each topic, fix (small) set of authoritative pages
  - For each topic, compute PR_t with random jumps only to authoritative pages of that topic
  - Compute query-specific topic probability $P[t|q]$ and query-specific pagerank $PR(d,q)=\sum P[t|q] \cdot PR_t(d)$

HITS (Hyperlink Induced Topic Search)

Idea: determine

- Pages with good content (***authorities***): many inlinks
- Pages with good links (***hubs***): many outlinks

Mutual reinforcement:

- good authorities have good hubs as predecessors
- good hubs have good authorities as successors

Define for nodes x, y $\in$V in Web graph W = (V, E)

authority score $$a_y \sim \sum_{(x,y)\in E} h_x$$

hub score $$h_x \sim \sum_{(x,y)\in E} a_y$$

HITS as Eigenvector Computation

Authority and hub scores in matrix notation:

$$\vec{a} = E^T \vec{h} \qquad \vec{h} = E \vec{a}$$

Iteration with adjacency matrix A:

$$\vec{a} = E^T \vec{h} = E^T E \vec{a} \qquad \vec{h} = E \vec{a} = E E^T \vec{h}$$

a and h are **Eigenvectors** of $E^T E$ and $E E^T$, respectively

<u>Intuitive interpretation:</u>

$M^{(auth)} = E^T E$ is the cocitation matrix: $M^{(auth)}_{ij}$ is the number of nodes that point to both i and j

$M^{(hub)} = E E^T$ is the bibliographic-coupling matrix: $M^{(hub)}_{ij}$ is the number of nodes to which both i and j point

HITS Algorithm

Compute fixpoint solution by **iteration with length normalization:**

initialization: $a^{(0)} = (1, 1, ..., 1)^T$, $h^{(0)} = (1, 1, ..., 1)^T$

repeat until sufficient convergence

$h^{(i+1)} := E\, a^{(i)}$

$h^{(i+1)} := h^{(i+1)} / ||h^{(i+1)}||_1$

$a^{(i+1)} := E^T\, h^{(i)}$

$a^{(i+1)} := a^{(i+1)} / ||a^{(i+1)}||_1$

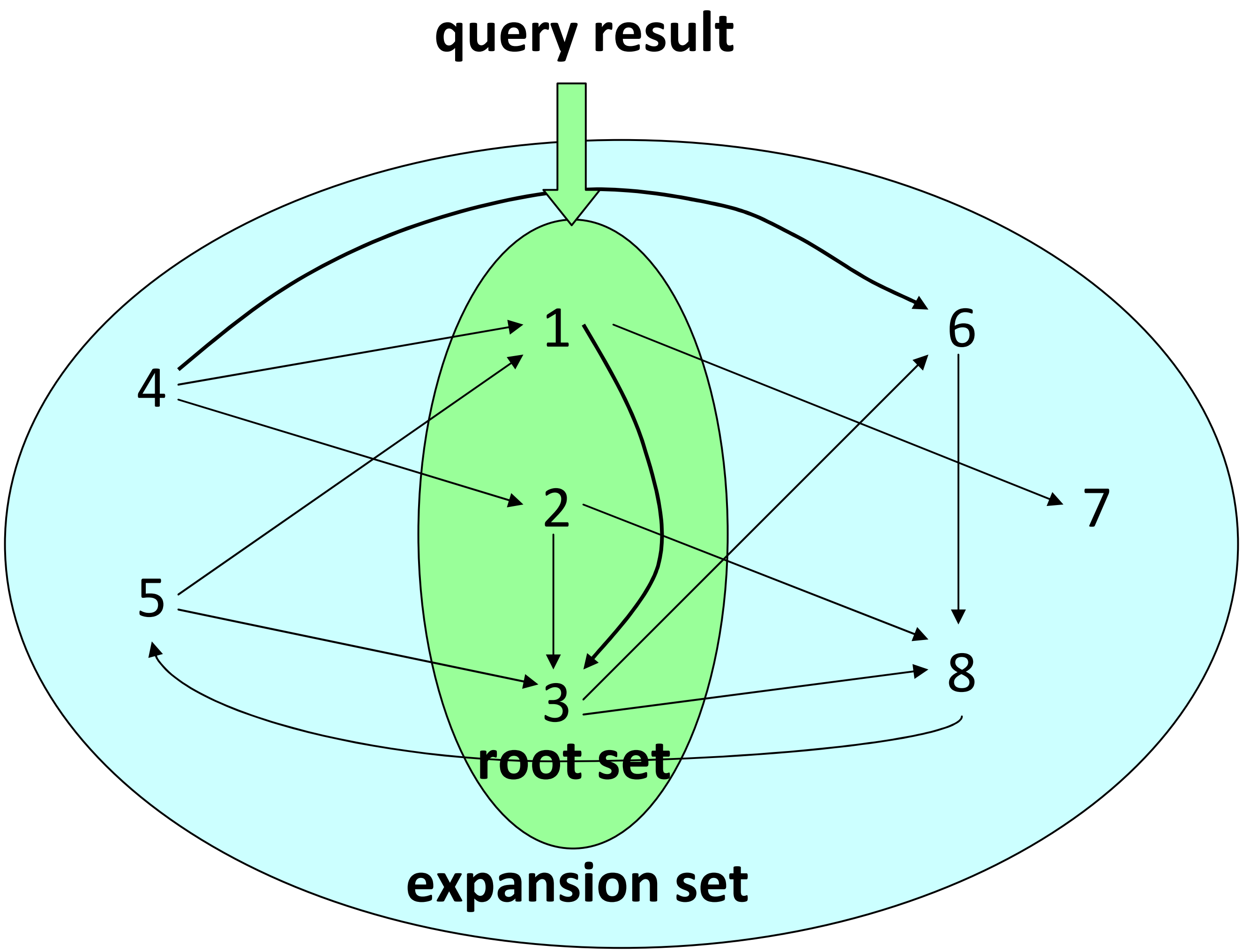
convergence guaranteed under fairly general conditions

HITS for Ranking Query Results

1) Determine sufficient number (e.g. 50-200) of „root pages“ via relevance ranking (using any content-based ranking scheme)
2) Add all successors of root pages
3) For each root page add up to *d* predecessors
4) Compute iteratively authority and hub scores of this „expansion set“ (e.g. 1000-5000 pages)
   $\rightarrow$ converges to principal Eigenvector
5) Return pages in descending order of authority scores (e.g. the 10 largest elements of vector a)

Potential problem of HITS algorithm:
Relevance ranking within root set is not considered

Example: HITS Construction of Graph



Improved HITS Algorithm

Potential weakness of the HITS algorithm:

- irritating links (automatically generated links, spam, etc.)
- topic drift (e.g. from „Jaguar car“ to „car“ in general)

Improvement:

- Introduce **edge weights**:
  0 for links within the same host,
  1/k with *k* links from *k* URLs of the same host to 1 URL *(aweight)*
  1/m with *m* links from 1 URL to *m* URLs on the same host (*hweight)*
- Consider **relevance weights** w.r.t. query (*score*)

→ Iterative computation of

authority score
$$a_q := \sum_{(p,q) \in E} h_p \cdot score(p) \cdot aweight(p,q)$$

hub score
$$h_p := \sum_{(p,q) \in E} a_q \cdot score(q) \cdot hweight(p,q)$$

Efficiently Computing PageRank

(Selected) Solutions:

- Compute Page-Rank-style authority measure online without storing the complete link graph
- Exploit block structure of the Web
- Decentralized, *synchronous* algorithm
- Decentralized, *asynchronous* algorithm

Online Link Analysis

Key ideas:

- Compute small fraction of authority as crawler proceeds **without storing the Web graph**
- Each page holds some **„cash“** that reflects its importance
- When a page is visited, it **distributes its cash** among its successors
- When a page is not visited, it can still accumulate cash
- This random process has a **stationary limit** that captures **importance of pages**

OPIC (Online Page Importance Computation)

Maintain for each page i (out of n pages):

- **C[i]** – cash that page i currently has and distributes
- **H[i]** – history of how much cash page has ever had in total

plus global counter

- **G** – total amount of cash that has ever been distributed

```
for each i do { C[i] := 1/n; H[i] := 0 }; G := 0;
do forever {
  choose page i (e.g., randomly);
  H[i] := H[i] + C[i];
  for each successor j of i do C[j] := C[j] + C[i] / outdegree(i);
  G := G + C[i];
  C[i] := 0; };
```

Note: 1) every page needs to be visited infinitely often (fairness)
2) the link graph is assumed to be strongly connected

OPIC Importance Measure

At each step *t* an estimate of the importance of page *i* is:

$(H_t[i] + C_t[i]) / (G_t + 1)$ (or alternatively: $H_t[i] / G_t$)

<u>Theorem:</u>

Let $X_t = H_t / G_t$ denote the vector of cash fractions accumulated by pages until step t.

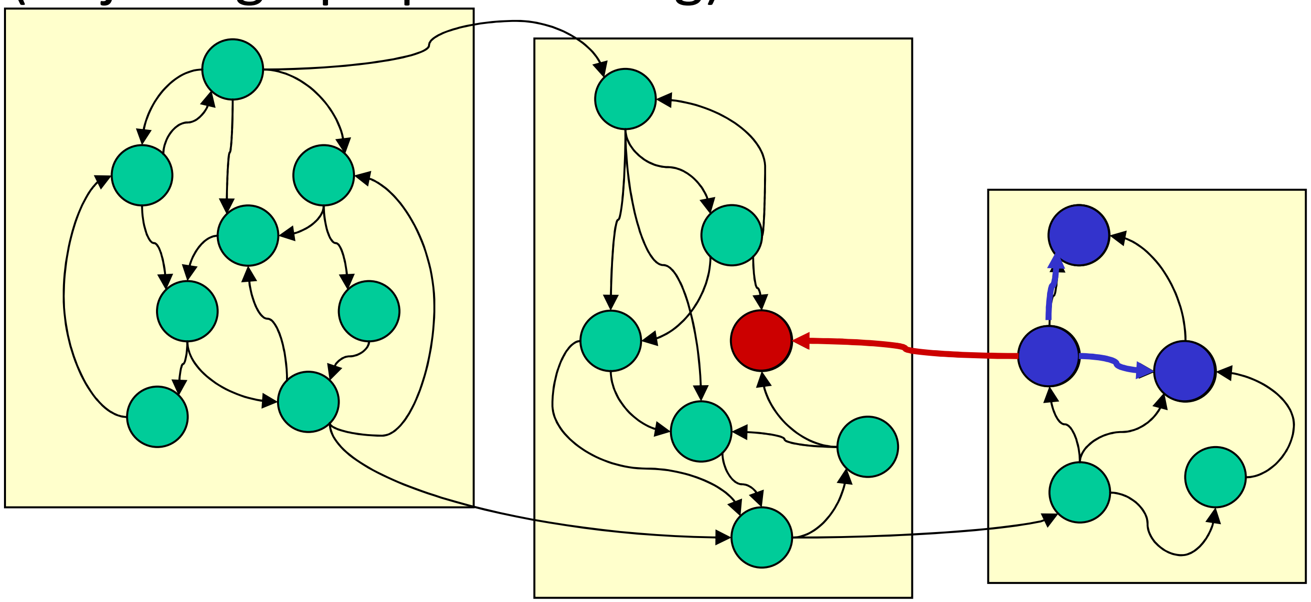
The limit $X = \lim_{t \to \infty} X_t$ exists with $||X||_1 = \Sigma_i X[i] = 1$.

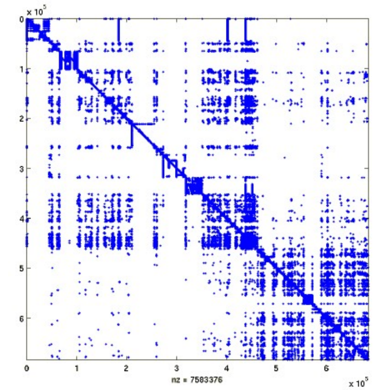
with crawl strategies such as:

- random
- greedy: read page *i* with highest cash *C[i]*
  (fair because non-visited pages accumulate cash until eventually read)
- cyclic (round-robin)

Exploiting Web structure

Exploit locality in Web link graph: construct block structure (disjoint graph partitioning) based on sites or domains





1) Compute local per-block pageranks
2) Construct block graph B with aggregated link weights proportional to sum of local pageranks of source nodes
3) Compute pagerank of B
4) Rescale local pageranks of pages by global pagerank of their block
5) Use these values as seeds for global pagerank computation

Decentralized synchronous computation

PageRank computation highly local:

needs only previous ranks of adjacent nodes

$\Rightarrow$ Apply distributed computing framework like MapReduce

References

Main references:

- A. Z. Broder et al.: **Graph structure in the Web**, Computer Networks 33, 309—320, 2000
- A. Bonato: **A survey of models of the Web graph**, Combinatorial and Algorithmic Aspects of Networking, 2005
- P. Baldi, P. Frasconi, P. Smyth: **Modeling the Internet and the Web**, chapters 1.7, 3, A

Additional references:

- A.-L. Barabasi, R. Albert: **Emergence of scaling in random networks**, Science 286, 509—512, 1999
- W. Aiello et al.: **A random graph model for massive graphs**, ACM STC, 2000
- W. Aiello et al.: **A random graph model for power-law graphs**, Experimental Math 10, 53—66, 2001
- R. Albert et al.: **Diameter of the World Wide Web**, Nature 401, 130—131, 1999
- M. Mitzenbacher: **A brief history of generative models for power law and lognormal distributions**, Internet Mathematics 1(2), 226—251, 2004
- R. Kumar et al.: Stochastic model for the Web graph, FOCS, 2000
- R. Cohen, S. Havlin: **Scale-free networks are ultrasmall**, Phys. Rev. Lett. 90, 058701, 2003
- A. Bonato, J. Janssen: **Limits and power laws of models for the Web graph and other networked information spaces**. Combinatorial and Algorithmic Aspects of Networking, 2005
- S.D. Kamvar et al.: **Exploiting the block structure of the Web for computing Pagerank**, WWW conference, 2003
- M. Faloutsos et al.: **On Power-Law relationships of the Internet topology**, SIGCOMM conference, 1999
- J. Kleinberg et al.: **The Web as a graph: Measurements, models, and methods**. Conference on Combinatorics and Computing, 1999
- D.J. Watts, S.H. Strogatz: **Collective dynamics of small-world networks**, Nature 393(6684), 409–410, 1998
- U. Brandes: **A Faster Algorithm for Betweenness Centrality**, Journal of Mathematical Sociology 25, 163—177, 2001
- S Brin, L. Page: **The Anatomy of a Large-Scale Hypertextual Web Search Engine**, WWW 1998
- T.H. Haveliwala: **Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search,** IEEE Trans. Knowl. Data Eng. 15(4), 784–796, 2003
- G. Jeh, J. Widom: **Scaling personalized web search**. WWW Conference, 2003
- J. Kleinberg: **Authoritative sources in a hyperlinked environment**, Journal of the ACM 36(5), 604–632, 1999
- S. Abiteboul, M. Preda, G. Cobena: **Adaptive on-line page importance computation**, WWW Conference 2003