Summary: What We Have Learned So Far

- **Real-world networks:**
  - small-world phenomenon:
    - Short path lengths
    - High clustering
  - Broad degree distributions, often power laws $P(k) \propto k^{-\gamma}$

- **Erdös-Renyi model:**
  - Short path lengths
  - Poisson degrees
  - No clustering

- **Watts-Strogatz Small World model:**
  - short path lengths
  - high clustering
  - almost constant degrees

- **Barabási-Albert scale-free model:**
  - short path lengths
  - power-law degrees
  - no clustering, but simple variants fix this

The Barabási-Albert Scale Free Model

- A model of **network growth**
- Based on the principle of **preferential attachment** - "rich get richer!"
- Yields networks with a power-law degree distribution

$$P(k) = \frac{2m^2}{k^3}$$

(average degree $\langle k \rangle = 2m$)

1. Take a small seed network, e.g. a few connected nodes
2. Let a new node of degree m enter the network
3. Connect the new node to existing nodes such that the probability of connecting to node i of degree k_i is

$$\pi_i = \frac{k_i}{\sum_i k_i}$$

4. Repeat 2.-3. until N nodes.

Scale-Free Network Models: Summary

- For growing networks, **preferential attachment** yields power-law degree distributions
- To be exact, it has to be linear:

$$\pi_i = \frac{k_i}{\sum_i k_i}$$

(If *superlinear*, "winner takes it all" and in the end one node has ALL the links! If *sublinear*, we get a stretched exponential degree distribution)

(if *mixed*, e.g. combination of linear preferential and random attachment, we get exponents larger than 3!)

- The fundamental model: Barabási-Albert, where

$$P(k) = \frac{2m^2}{k^3}$$

- Several mechanisms lead to the preferential attachment principle!

More Network Characteristics
&
Network Analysis

Degree Correlations

- "If the degree of a vertex is k, does this affect the degrees of its neighbours?"

- We could investigate the conditional probability

  $P(k'|k)$

  of the neighbour having degree k'

- In practice this is cumbersome to calculate (esp. in data analysis)

- Hence the average nearest-neighbour degree $k_{nn}(k)$ is typically used

$$k_{nn}(k) = \frac{1}{N_k} \sum_{i,k_i=k} \left[\frac{1}{k_i} \sum_{j \in \nu_i} k_j \right]$$

Degree Correlations

- In practical network analysis, $k_{nn}(k)$ is simply calculated by averaging over all neighbour degrees for each k

- **Assortativity:** positive degree correlations, $k_{nn}(k)$ increasing with k

- **Disassortativity:** negative degree correlations, $k_{nn}(k)$ decreasing with k

- Alternative method: the assortativity coefficient (Pearson correlation coefficient)

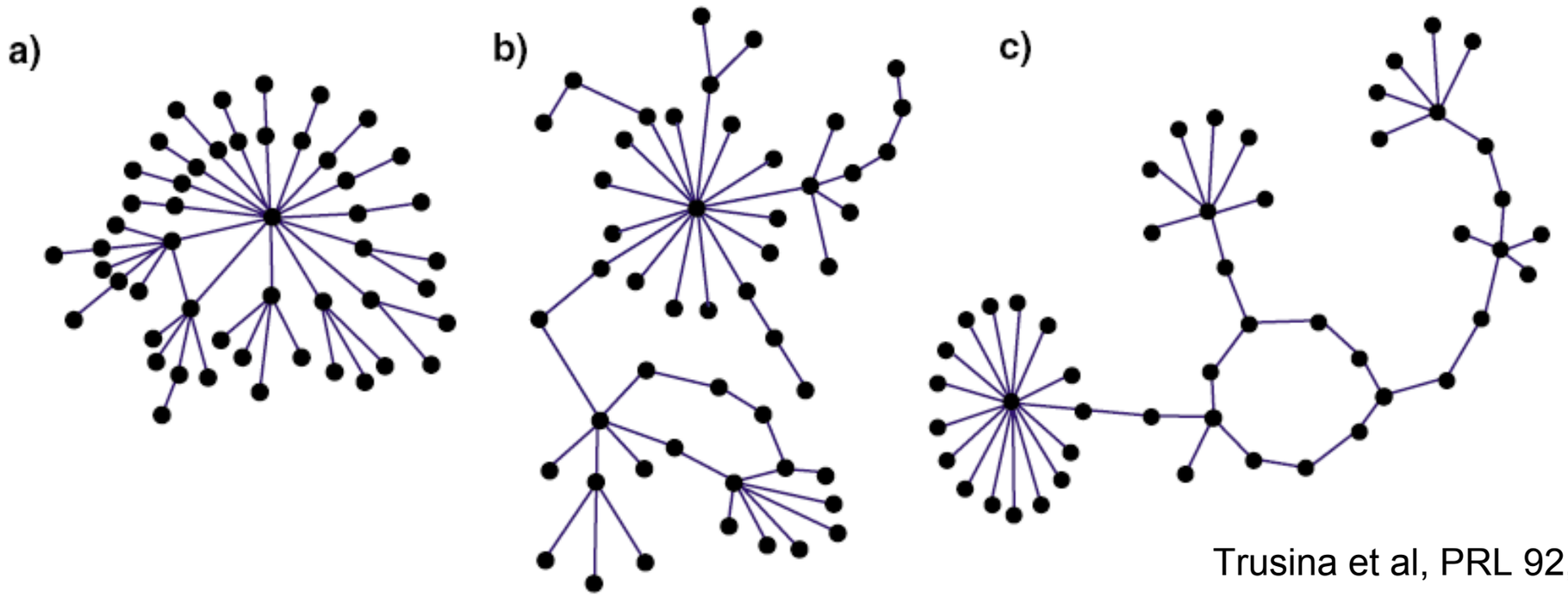
$$r = \frac{\langle k_i k_j \rangle - \langle k_i \rangle \langle k_j \rangle}{\sqrt{\langle k_i^2 \rangle - \langle k_i \rangle^2} \sqrt{\langle k_j^2 \rangle - \langle k_j \rangle^2}}$$

- r positive: assortative mixing

- r negative: disassortative mixing

Degree Correlations: Visualized Example

maximal correlations **no correlations** **maximal anticorrelations**

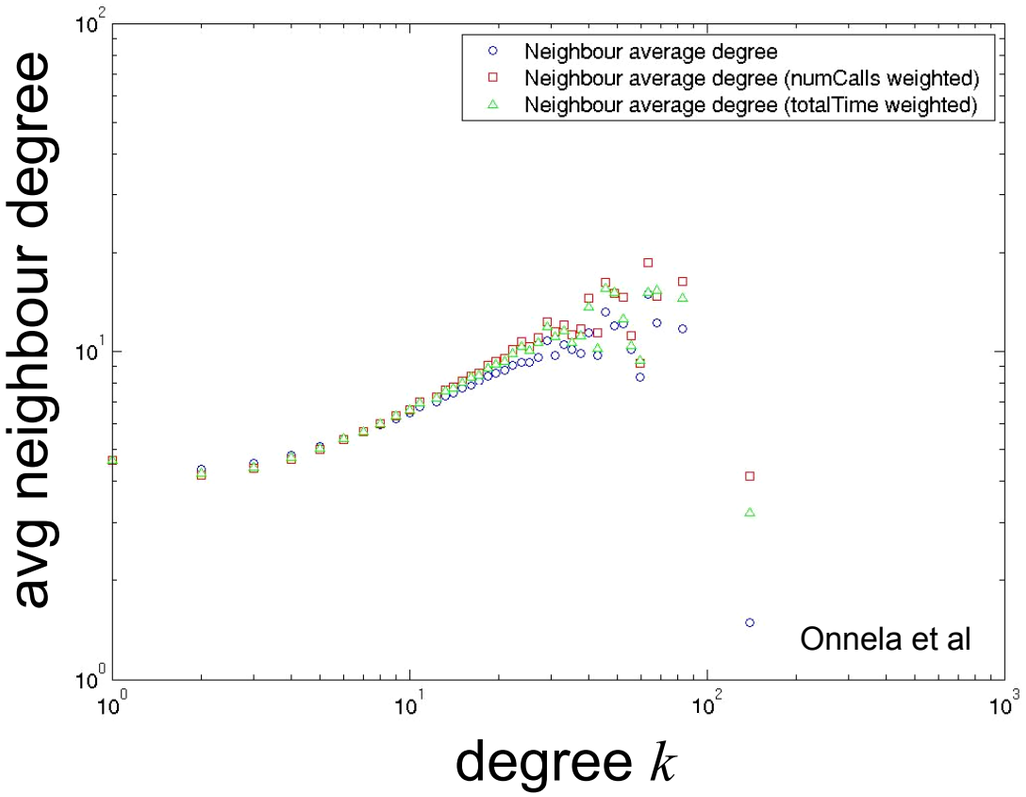


Trusina et al, PRL 92, 2004

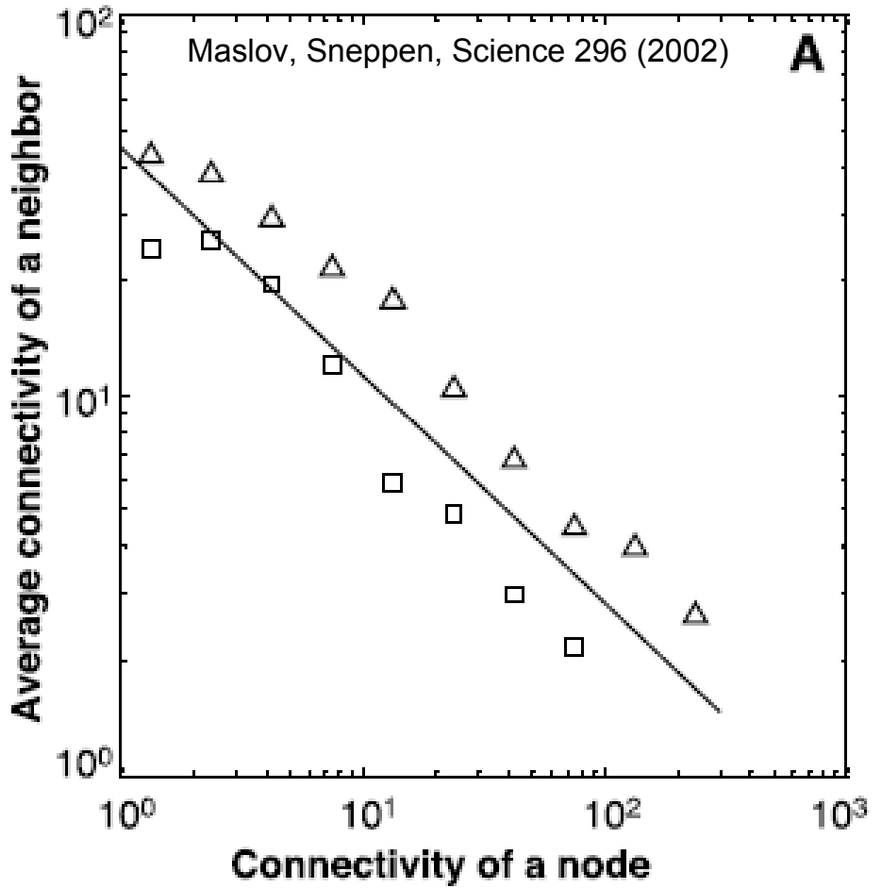
Three networks with same degree sequence, differently rewired

Degree Correlations: Examples

Social network based on mobile telephone calls



S. Cerevisiae protein interactions and genetic regulation



△ interactions
□ regulatory network

Social networks are almost always assortative, biological networks disassortative

$k_{nn}(k)$ for uncorrelated networks

- Let'a first calculate the probability that a vertex of degree k is connected to a vertex of degree k'

- In uncorrelated networks, this is equal to the probability that the vertex in the other end of a random link has degree k':

there are $P(k')N$ vertices of degree k', and k' times this number of edge ends points to them

$$P(k'|k) = \frac{k' P(k') N}{\langle k \rangle N} = \frac{k' P(k')}{\langle k \rangle}$$

there are altogether $N\langle k \rangle$ edge ends

- Now

$$\begin{aligned} k_{nn}(k) &= \sum_{k'} k' P(k'|k) \\ &= \sum_{k'} \frac{(k')^2 P(k')}{\langle k \rangle} \\ &= \frac{\langle k^2 \rangle}{\langle k \rangle} \end{aligned}$$

$k_{nn}(k)$ for the BA model

1. Take a small seed network
   - eg 4 connected vertices
2. Create a new vertex with m edges
3. Connect the m edges to existing vertices with a probability proportional to their degree k, ie the probability π_i of choosing vertex i is

$$\pi_i = \frac{k_i}{\sum_i k_i}$$

4. Repeat 2.-3. until the network has grown to desired size of N vertices

- First we need an expression for the degree of vertex i at time t
- Let's write the rate equation:

$$\frac{\partial k_i(t)}{\partial t} = m\pi_i$$

$$= m\frac{k_i}{\sum_i k_i}$$

$\sum_i k_i = 2mt + N_0 \sim 2mt$

$$= m\frac{k_i}{2mt}$$

$k_{nn}(k)$ for the BA model

- Now let's solve this by integration

$$\Rightarrow k_i(t) = m\left(\frac{t}{i}\right)^{\frac{1}{2}}$$

(i = "time of birth" of vertex i)

- Let us now assume that as far as correlations go, BA networks correspond to random nets:

$$k_{nn}(k) = \frac{\langle k^2 \rangle}{\langle k \rangle}$$

- Since

$$\begin{aligned}
\langle k^2 \rangle &= \frac{1}{N}\sum_{i=1}^{N}\left[k_i(t=N)\right]^2 \\
&= \frac{1}{N}m^2\sum_{i=1}^{N}\left(\frac{N}{i}\right) \\
&\approx m^2\int_{i=1}^{N}\frac{1}{i}di \\
&\approx m^2 \ln N \\
&= \frac{\langle k \rangle^2}{4}\ln N \qquad (\langle k\rangle = 2m)
\end{aligned}$$

we finally get

$$k_{nn}(k) = \frac{\langle k \rangle}{4}\ln N.$$

More correlation measures: the rich-club coefficient

- How connected are high-degree vertices among themselves?
- The rich-club coefficient

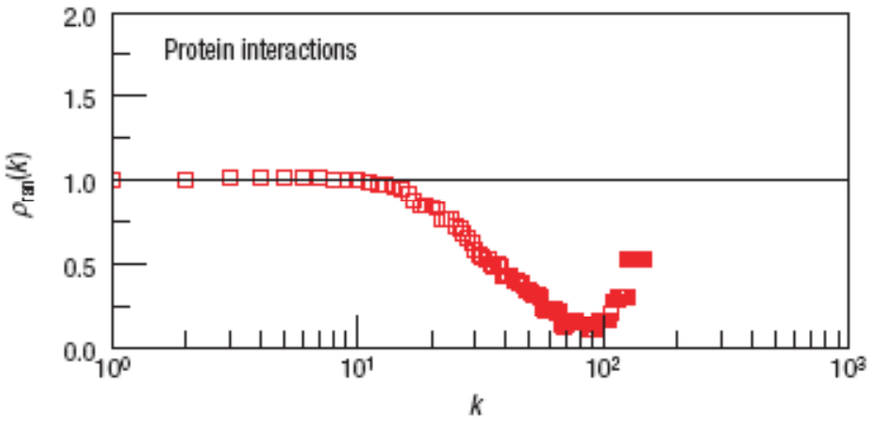
$$\phi(k) = \frac{2E_{>k}}{N_{>k}(N_{>k}-1)}$$

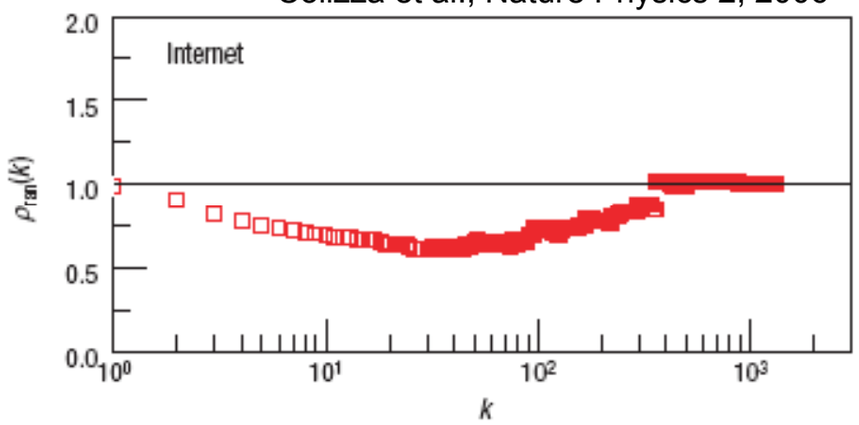
measures how many edges $E_{>k}$ exist among the $N_{>k}$ vertices of degrees higher than k, divided by the maximum possible number $N_{>k}(N_{>k}\text{-}1)/2$

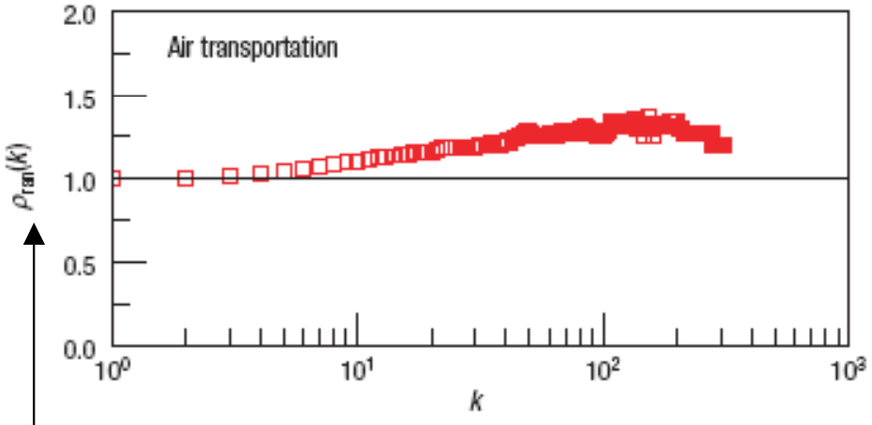
- It is useful to compare the values against randomized reference networks
- Randomized reference: rewire the network whilst retaining its degree sequence, getting rid of correlations (the configuration model)

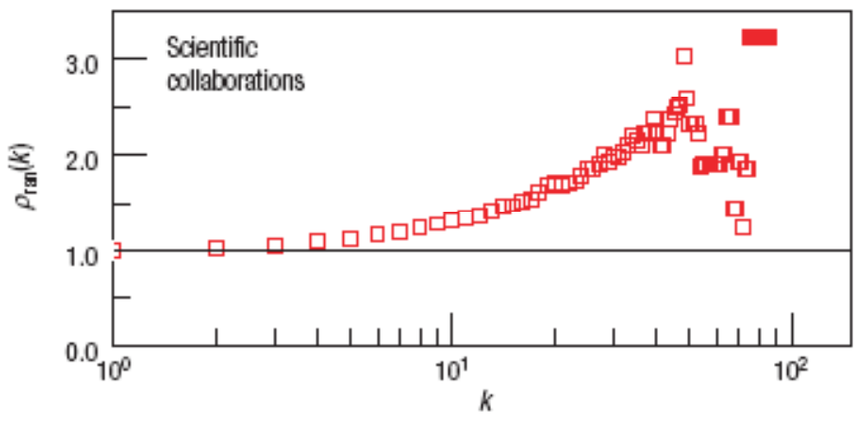
Rich-club coefficient: examples

Colizza et al., Nature Physics 2, 2006





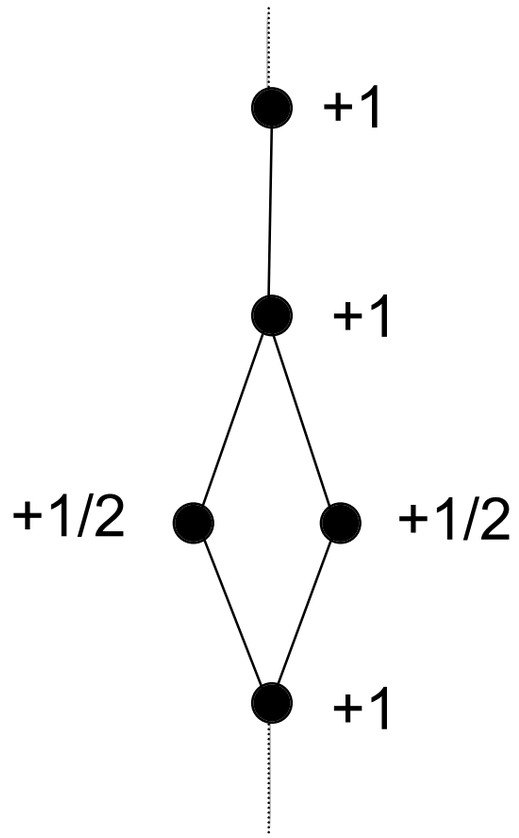




ratio of $\phi(k)$ in orig. nets to $\phi(k)$ in randomized counterparts

Betweenness Centrality

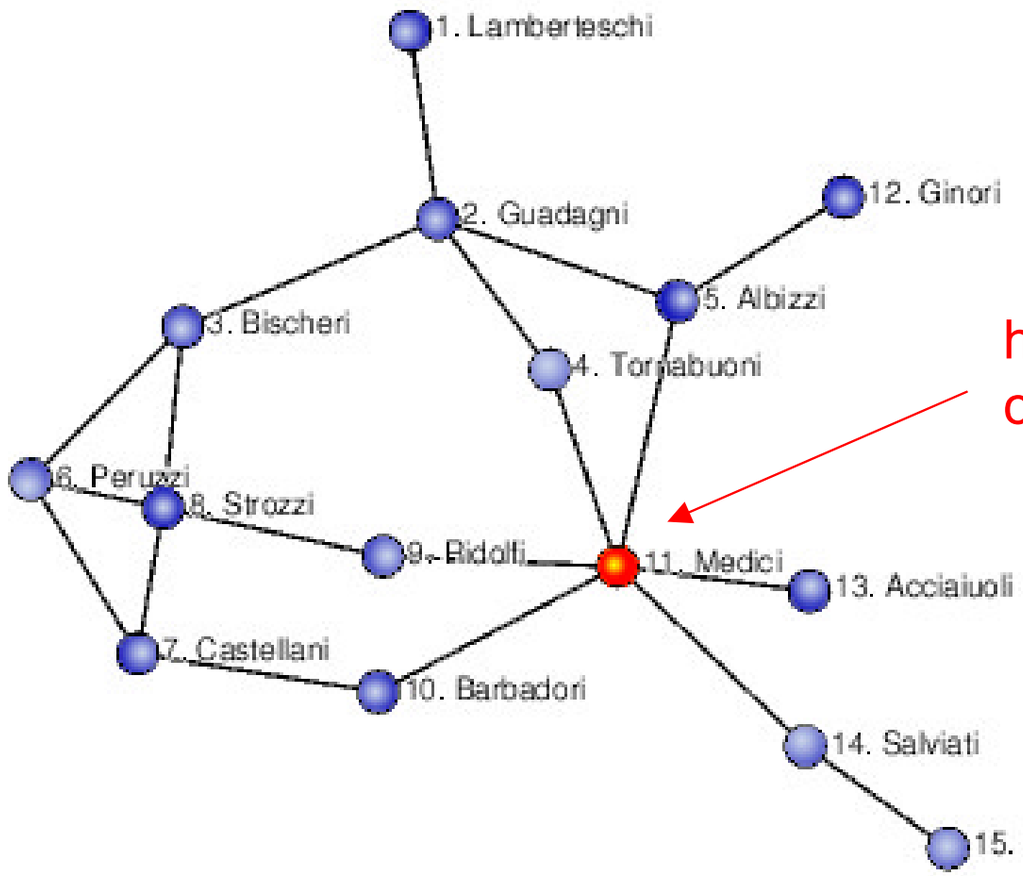
- Measures the amount of flow through a vertex (or an edge), if each vertex sends e.g. a signal through all other vertices via shortest paths



- Formally: number of **shortest** paths going through vertex/edge, such that the contribution of each path is divided by its multiplicity (if any)

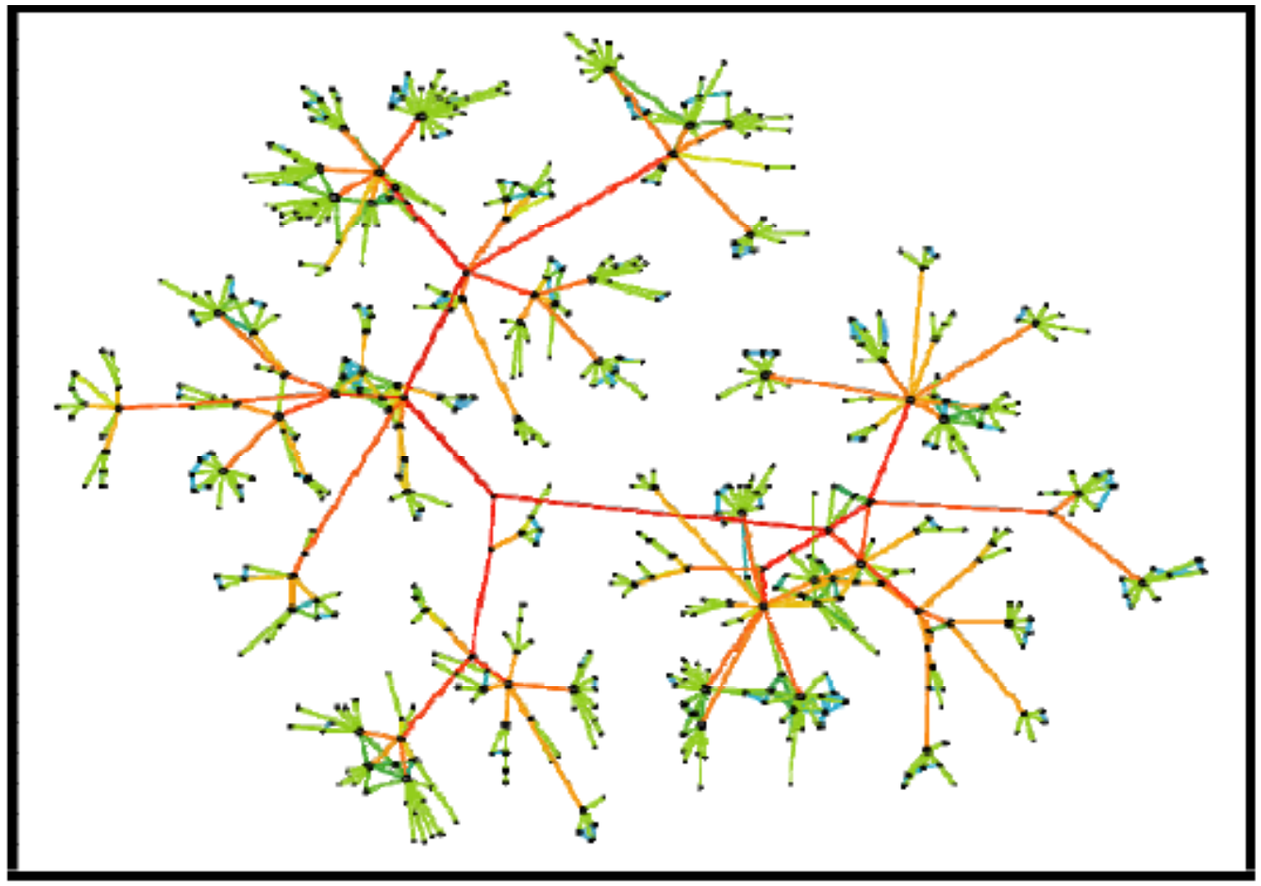
- Computationally demanding, for a good algorithm see M. E. J. Newman, *Phys. Rev. E* **64**, 016132 (2001)

Betweenness Centrality: An Example



Marriages between influential families in 13th century Florence

Betweenness Centrality: An Example



Betweenness calculated for edges: **red=high**

Closeness & Eigenvector Centrality

- **Closeness:** measures how far, on the average, a vertex is from all other vertices:

$$C_{C,i} = \frac{1}{\sum_{j \neq i} d_{ij}}$$

where d_{ij} is the distance along links from i to j

- **Eigenvector:** assigns relative scores to all nodes in the network based on the principle that connections to nodes having a high score contribute more to the score of the node in question

- A measure of ”influence”

$$x_i = \frac{1}{\lambda} \sum_{j=1}^{N} A_{i,j} x_j$$

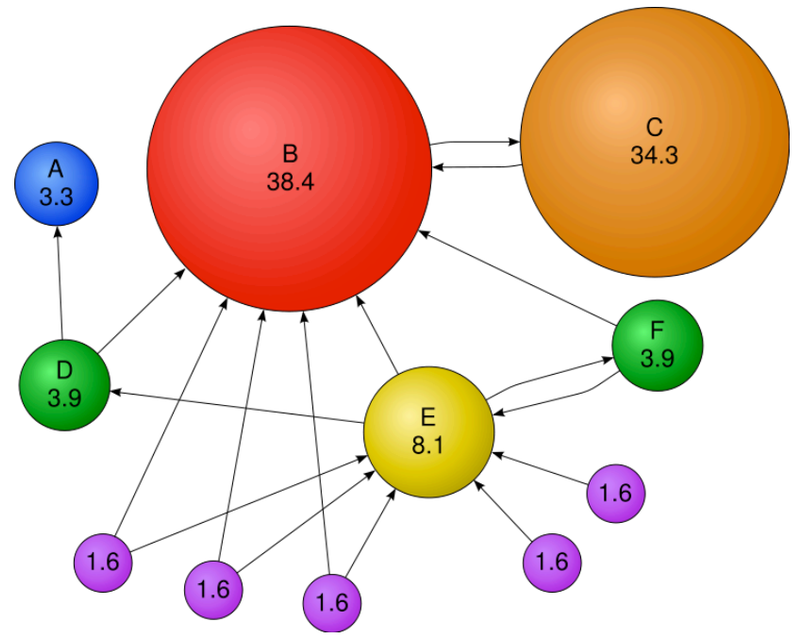
$$A\vec{x} = \lambda\vec{x}$$

PageRank

$$PR(i) = \frac{1-d}{N} + d \sum_{j \in \nu_i} \frac{PR(j)}{k_{out}(j)}$$

PR(i) = PageRank of *i*, *d* = damping factor, *N* = number of pages, ν_i= in-neighbourhood of *i*

- In essence, a "damped" version of eigenvector centrality
- Corresponds to a random surfer following hyperlinks, who continues with probability d (assumed for Google to be 0.85), and jumps to a random page with probability $1\text{-}d$



PageRank

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Brin, S. and Page, L. (1998) The Anatomy of a Large-Scale Hypertextual Web Search Engine. In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia.

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at http://google.stanford.edu/
To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been done on them. Furthermore, due to rapid advance in technology and web proliferation, creating a web search engine today is very different from three years ago. This paper provides an in-depth description of our large-scale web search engine -- the first such detailed public description we know of to date. Apart from the problems of scaling traditional search techniques to data of this magnitude, there are new technical challenges involved with using the additional information present in hypertext to produce better search results. This paper addresses this question of how to build a practical large-scale system which can exploit the additional information present in hypertext. Also we look at the problem of how to effectively deal with uncontrolled hypertext collections where anyone can publish anything they want.

Keywords

World Wide Web, Search Engines, Information Retrieval, PageRank, Google

Subgraphs & Motifs

- **Subgraph:** any set of nodes in the network, and the edges connecting them

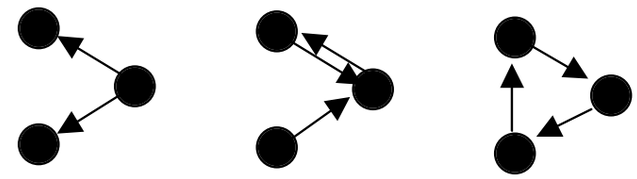
- **Clique:** a fully connected subgraph

- k-clique: clique with k vertices

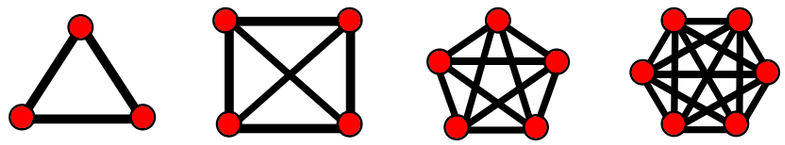
- **Motif:** subgraph occurring in a network at a number significantly higher than in randomized counterpart

randomization:
- pick two random links
- exchange endpoints
- repeat

directed subgraphs of order 3



cliques



Counting Motifs: Z-score

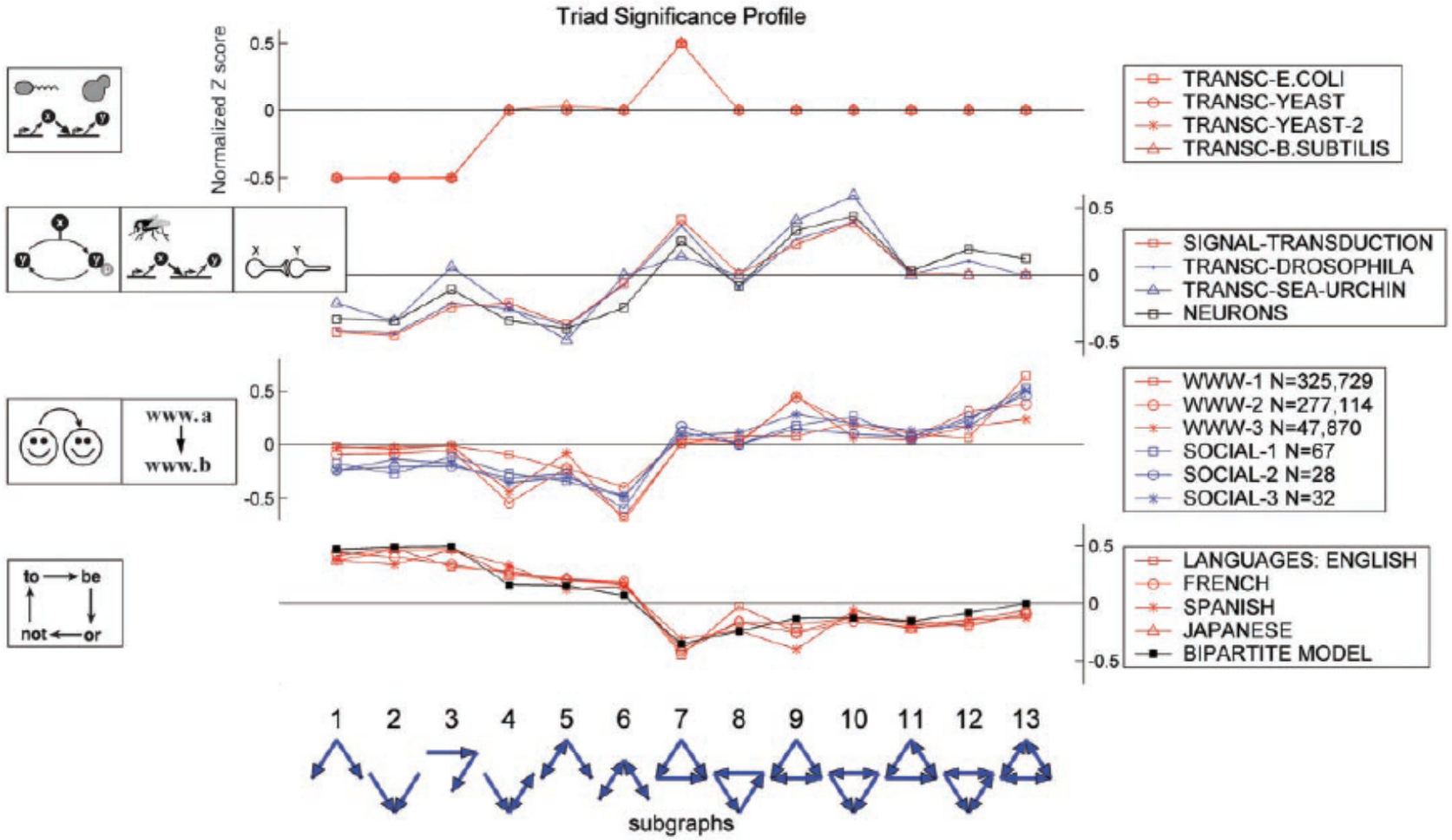
number of times the subgraph M occurs in an empirical network

average number of times the subgraph M in randomized reference ensemble

$$Z_M = \frac{n_M - \left\langle n_M^{\mathrm{rand}} \right\rangle}{\sigma_{n_M}^{\mathrm{rand}}}$$

standard deviation of n_M in reference ensemble

Motifs: Example



Milo et al., Science 303, 1538 (2004)

Motifs: Conservation in Evolution

- Yeast (S. Cerevisiae) protein interactions
- Find out motifs in the interaction network
- Find out orthologs for proteins in higher eucaryotes (eg humans)
- Calculate fraction of motifs where each protein has an ortholog

Table 1 Evolutionary conservation of motif constituents

#	Motifs	Number of yeast motifs	Natural conservation rate	Random conservation rate	Conservation ratio
1		9,266	13.67%	4.63%	2.94
2		167,304	4.99%	0.81%	6.15
3		3,846	20.51%	1.01%	20.28
4		3,649,591	0.73%	0.12%	5.87
5		1,763,891	2.64%	0.18%	14.67
6		9,646	6.71%	0.17%	40.44
7		164,075	7.67%	0.17%	45.56
8		12,423	18.68%	0.12%	157.89
9		2,339	32.53%	0.08%	422.78
10		25,749	14.77%	0.05%	279.71
11		1,433	47.24%	0.02%	2,256.67

Wuchty et al., Nature Genetics 35, 176 (2003)

Weighted Networks

Weighted networks

- Elements ⇔ vertices
- Interactions ⇔ edges
- An edge between v_i and v_j means v_i and v_j interact
- In reality, interactions can have different strengths, leading to *weighted networks*

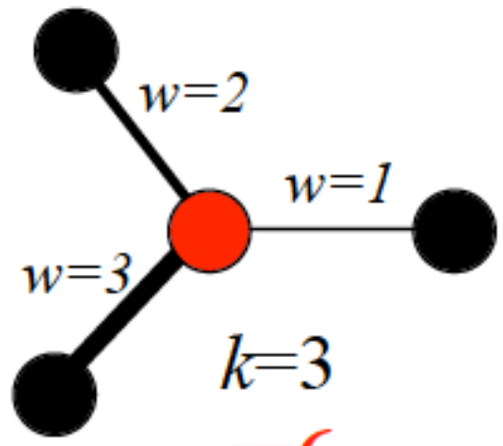
Vertex	Edge	Weight
person	friendship	closeness
neuron	synapse	synaptic strength
WWW	hyperlink	none
company	ownership	% owned
gene	regulation	level of regulation

Weighted Networks: Fundamentals

- Let us denote the weight between i and j by w_{ij}
- (Usually) $w_{ij} \geq 0$ and $w_{ij} = 0$ means that there is no edge
- The weights w_{ij} form **a weight matrix W**, analogous to the adjacency matrix
- For undirected networks, $W^T = W$
- If weights $w_{ij} = \{0,1\}$, W=A

- The notion of degree is readily generalized for weighted networks; the **strength** of a vertex is defined as

$$s_i = \sum_{j=1}^{N} w_{ij}$$



Strength Distributions of Real-World Networks

- Just as for the degree, we can investigate the strength distributions P(s) of networks

- And (unsurprisingly), these tend to be broad and have power-law-like tails

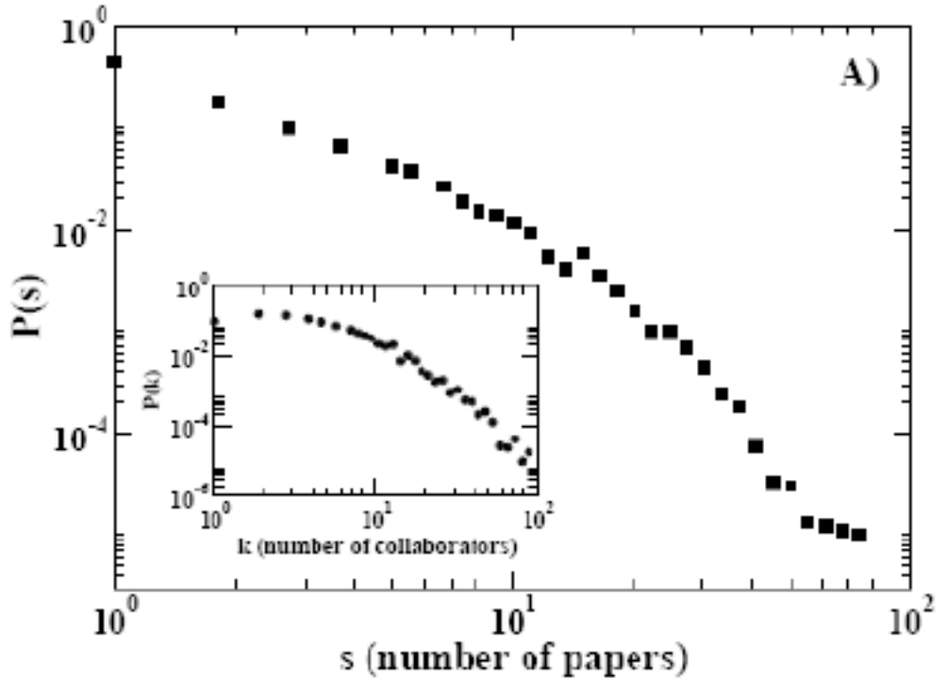
- Evidently this has to do with the fact that degrees and strengths are related

Scientific collaboration "weights" from the bipartite co-authorship network:

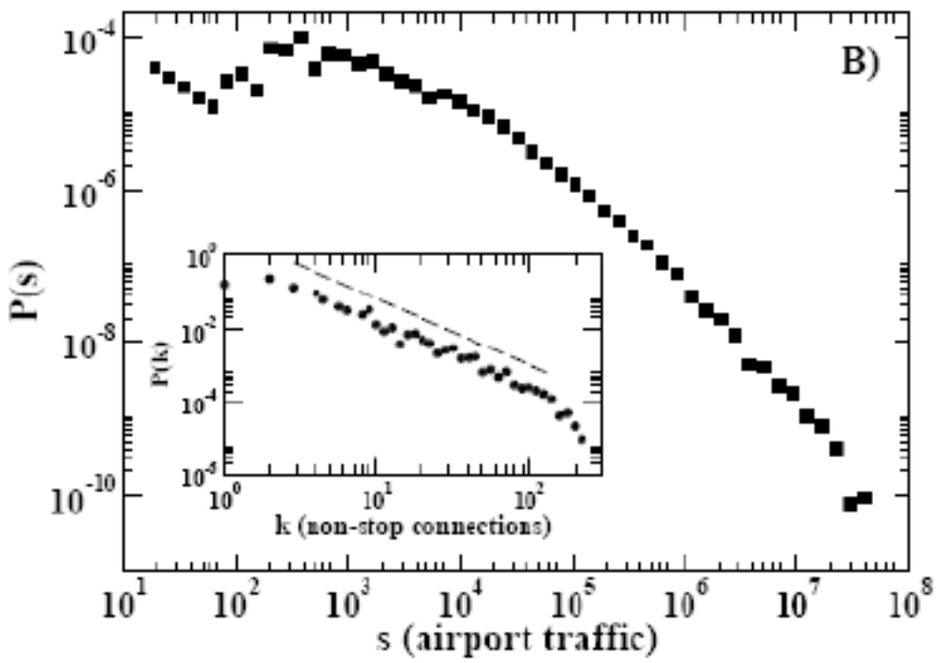
$$w_{ij} = \sum_p \frac{\delta_i^p \delta_j^p}{n_p - 1}$$

p runs over all papers
n_p = # of authors of p

scientific collaborations



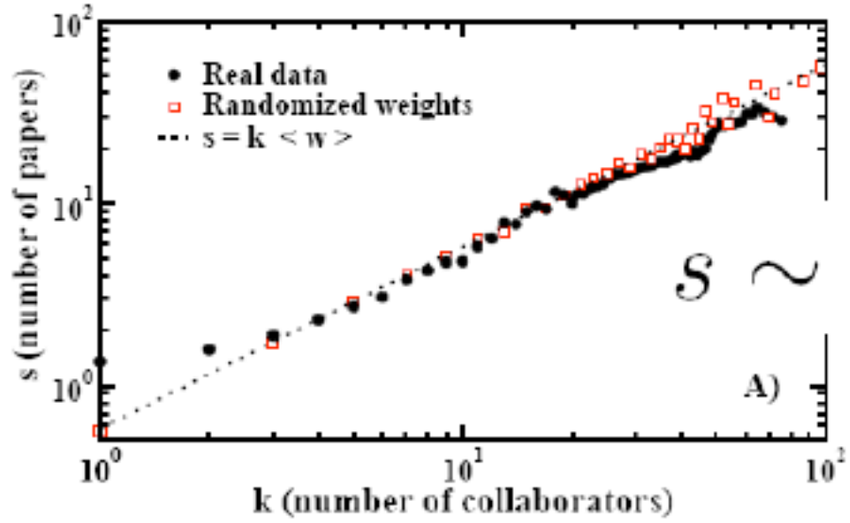
world-wide airport network



Strength-Degree Correlations

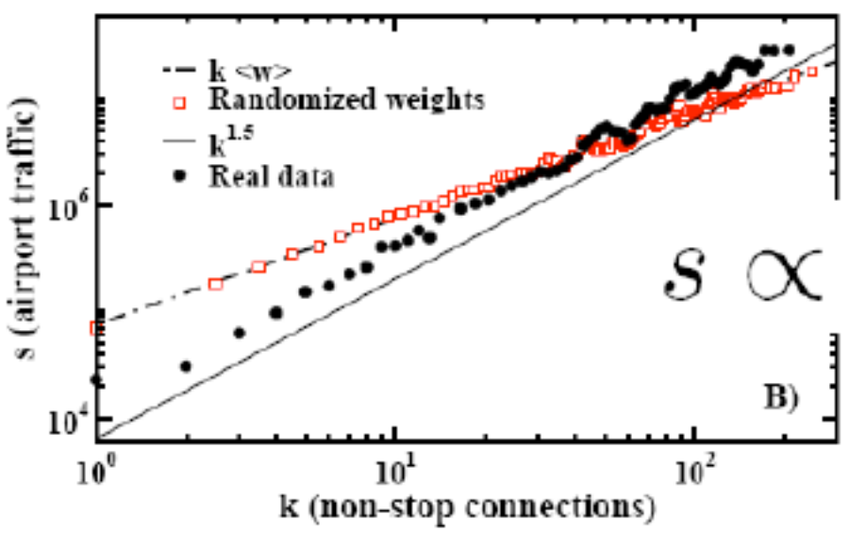
- How does the (average) strength behave as function of degree?
- If $\langle s|k\rangle = k\langle w\rangle$, i.e. the dependency is linear, weights and degrees are uncorrelated
- However other kinds of dependencies have been observed; often $\langle s|k\rangle \propto k^{\beta}$
- For $\beta > 1$, high-degree nodes attract higher-weight edges

scientific collaborations



$$s \sim k \langle w \rangle$$

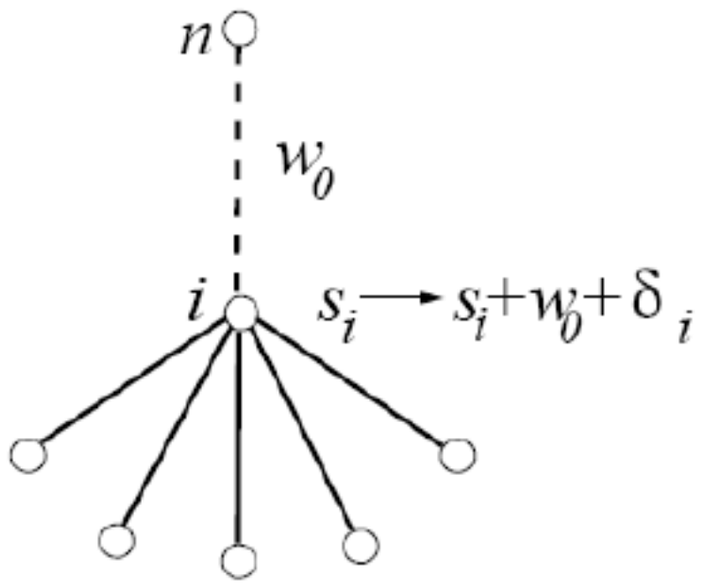
world-wide airport network



$$s \propto k^{1.5}$$

The Barrat-Barthélemy-Vespignani Model

- A ”generalized” version of the BA model
- Power-law distributions of degrees, strengths and weights



1. Take a small seed network
2. Create a new vertex with m edges of weight w_0
3. Connect the m edges to existing vertices with a probability proportional to their strength s:

$$\pi_i = \frac{s_i}{\sum_j s_j}$$

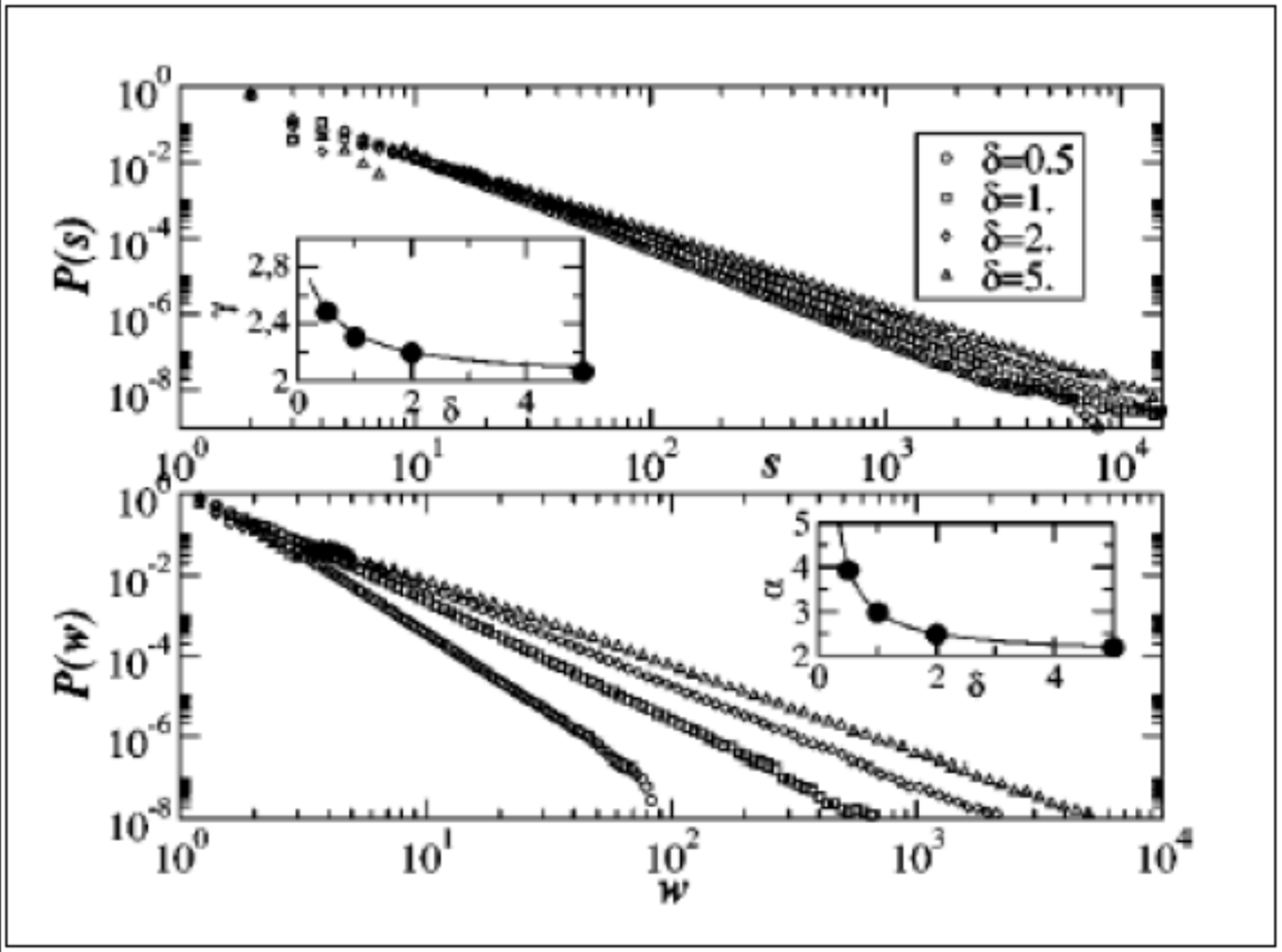
4. Update weights of edges of the selected vertices according to

$$\Delta w_{ij} = \delta_i \frac{w_{ij}}{s_i}$$

5. Repeat 2-4.

Barrat, Barthelemy, Vespignani, Phys Rev Lett **92**, 228701 (2004)

The Barrat-Barthélemy-Vespignani Model



- The exponents of strength, degree, and weight power laws depend on the weight addition parameter δ
- The strength and degree power-law exponents $\gamma \in [2,3]$
- Recall that for natural networks with $P(k) \propto k^{-\gamma}$, $\gamma \in [2,3]$, whereas

  (almost) all unweighted models yield $\gamma \geq 3$
- ...so one possible explanation for real-world exponents is that weights are involved!

How To Generalize Other Quantities

- For some quantities simply weighting the contributions of edges by $w_{ij}/(\sum_j w_{ij})$ works fine
- For example, one can consider the weighted average nearest-neighbour degree

$$k^w_{nn,i} = \frac{1}{s_i} \sum_{j=1}^{N} a_{ij} w_{ij} k_j$$

- Evidently, all weighted quantities should be equal to their unweighted counterparts if $w_{ij} = \{0,1\}$
- Often, the actual meaning and definition of weights plays a role
- E.g. should "weighted path lengths" be defined as $l=\sum w_{ij}$ or $l=\prod w_{ij}$?
- Some other quantities can be defined in numerous ways (which might indicate that the wrong question is being asked...)

Case Example: The Weighted Clustering Coefficient

Barrat & Barthélemy & Vespignani:

$$\tilde{C}_{i,B} = \frac{1}{s_i(k_i-1)} \sum_{j,k} \frac{w_{ij}+w_{ik}}{2} a_{ij}a_{jk}a_{ik}$$

Onnela, Saramäki, Kaski, Kertész:

$$\tilde{C}_{i,O} = \frac{1}{k_i(k_i-1)} \sum_{j,k} (\hat{w}_{ij}\hat{w}_{ik}\hat{w}_{jk})^{1/3}$$

Zhang et al:

$$\tilde{C}_{i,Z} = \frac{\sum_{j,k} \hat{w}_{ij}\hat{w}_{jk}\hat{w}_{ik}}{\left(\sum_k \hat{w}_{ik}\right)^2 - \sum_k \hat{w}_{ik}^2}$$

Holme et al:

$$\tilde{C}_{i,H} = \frac{\sum_{j,k} w_{ij}w_{jk}w_{ki}}{\max(w)\sum_{j,k} w_{ij}w_{ki}} = \frac{\mathbf{W}^3_{ii}}{(\mathbf{W}\mathbf{W}_{\max}\mathbf{W})_{ii}}$$

...same idea, different formulas, different behaviour...