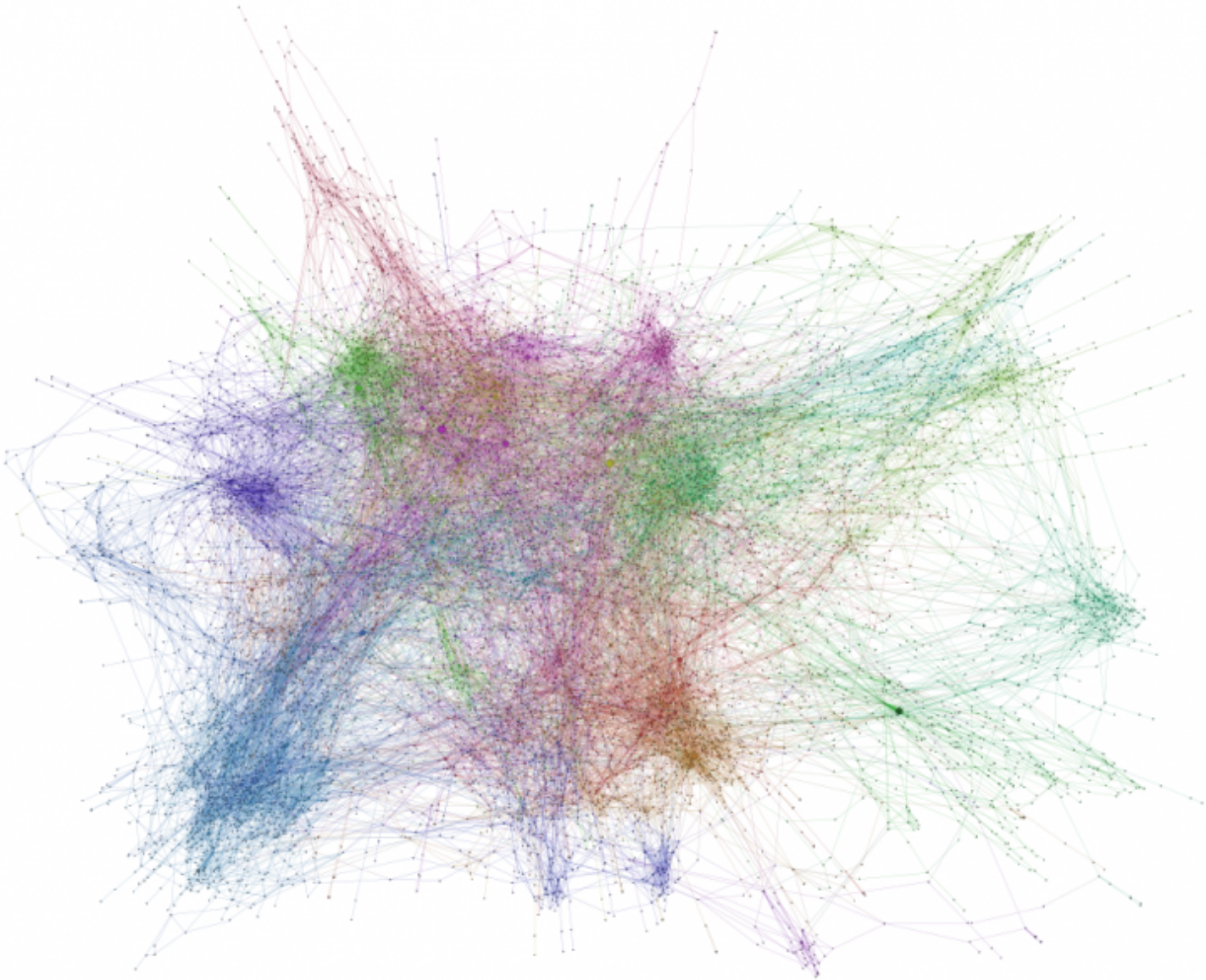
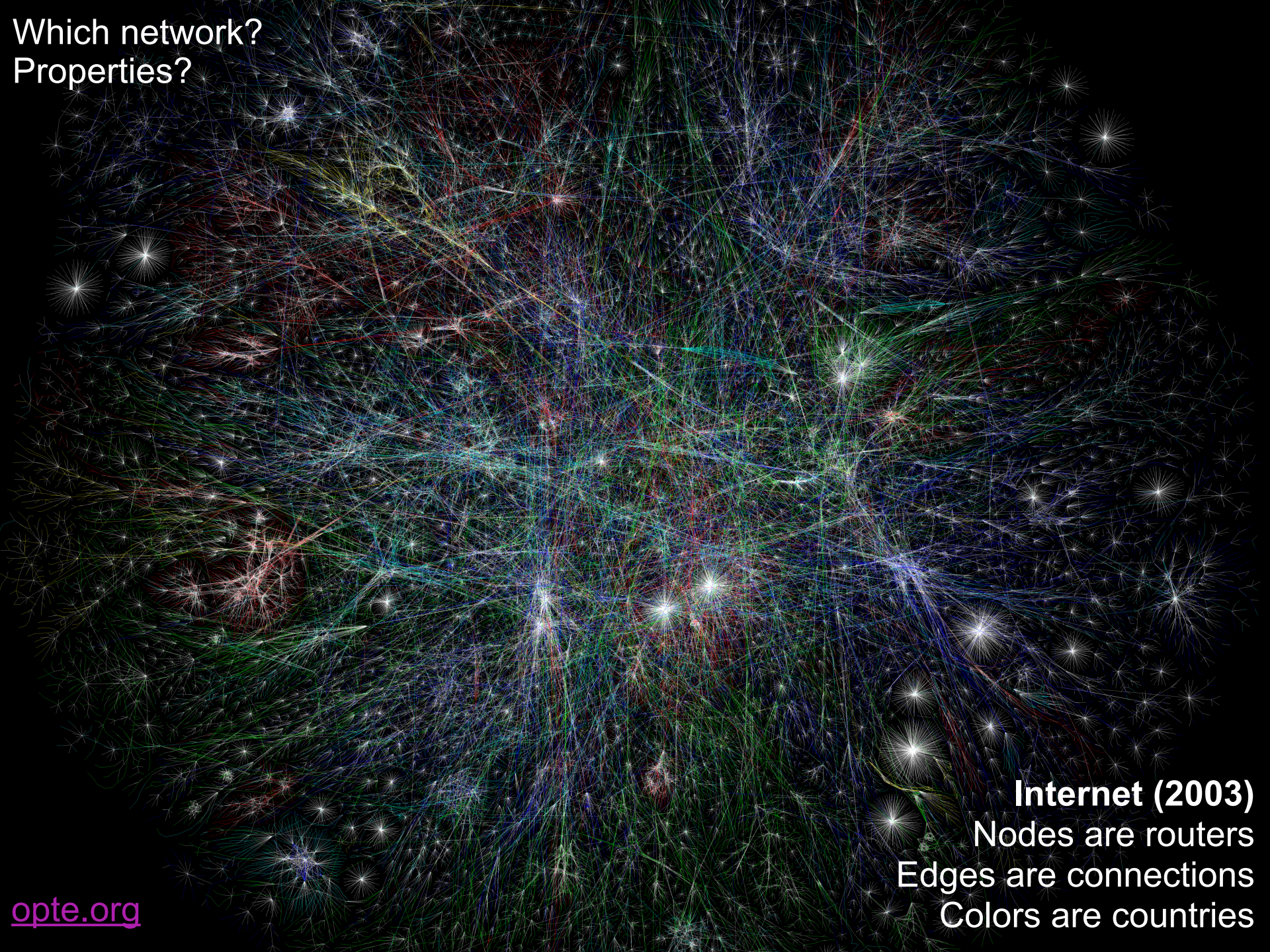
Network Analysis



Which network?
Properties?

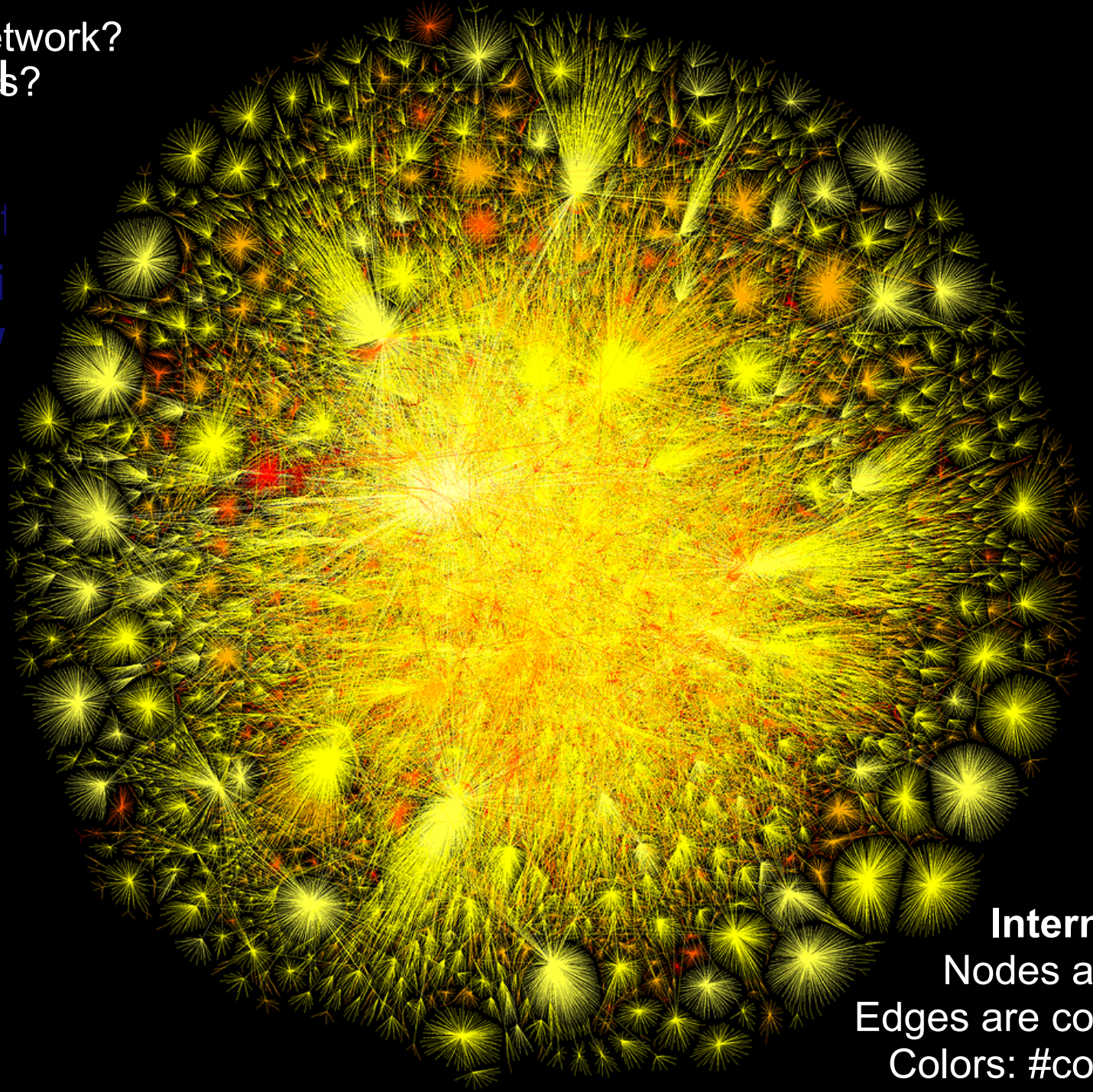


Internet (2003)
Nodes are routers
Edges are connections
Colors are countries

opte.org

Which network?
Properties?

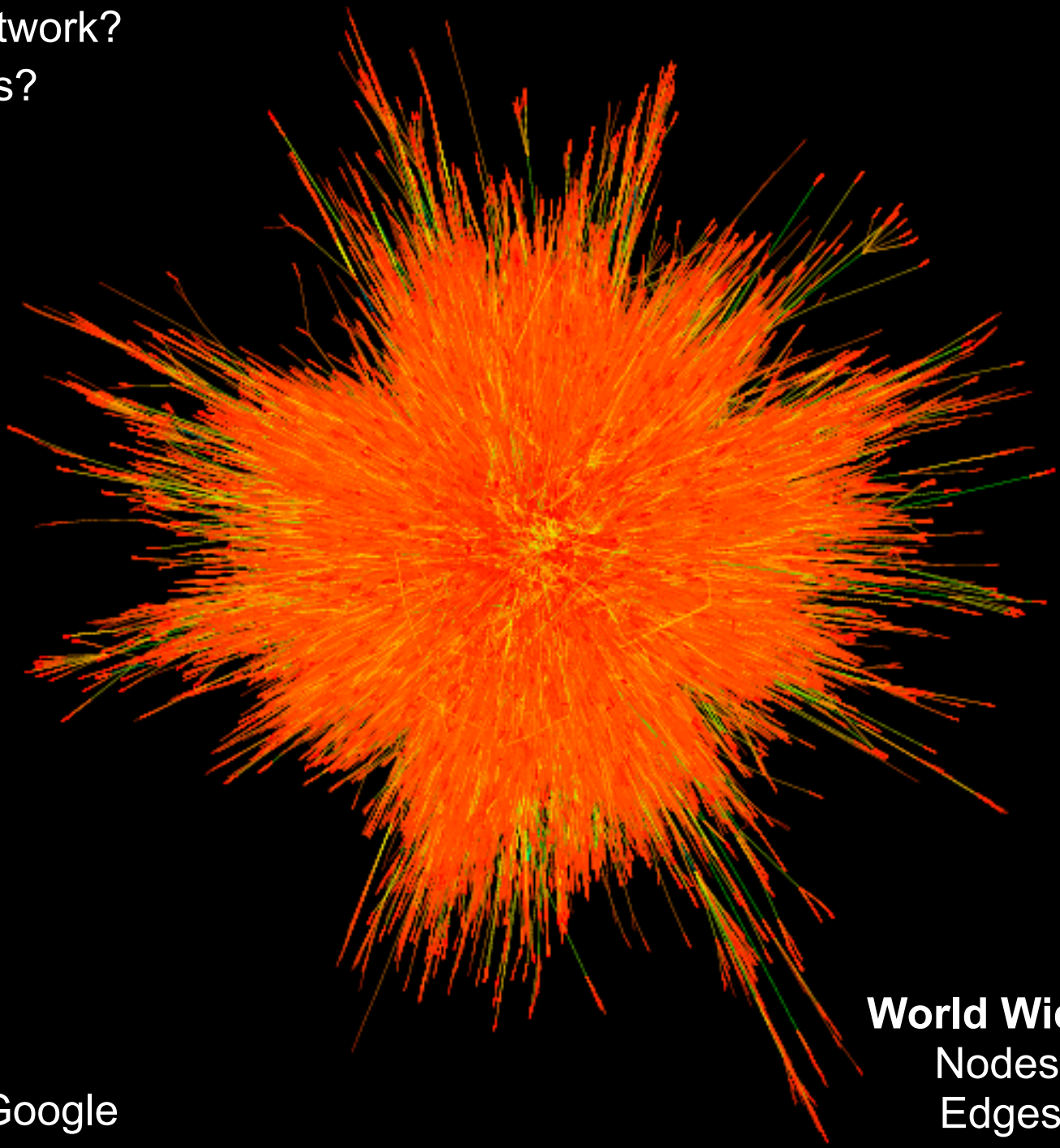
- Int
- Ci
- W



Internet (2010)
Nodes are routers
Edges are connections
Colors: #connections

opte.org

Which network?
Properties?

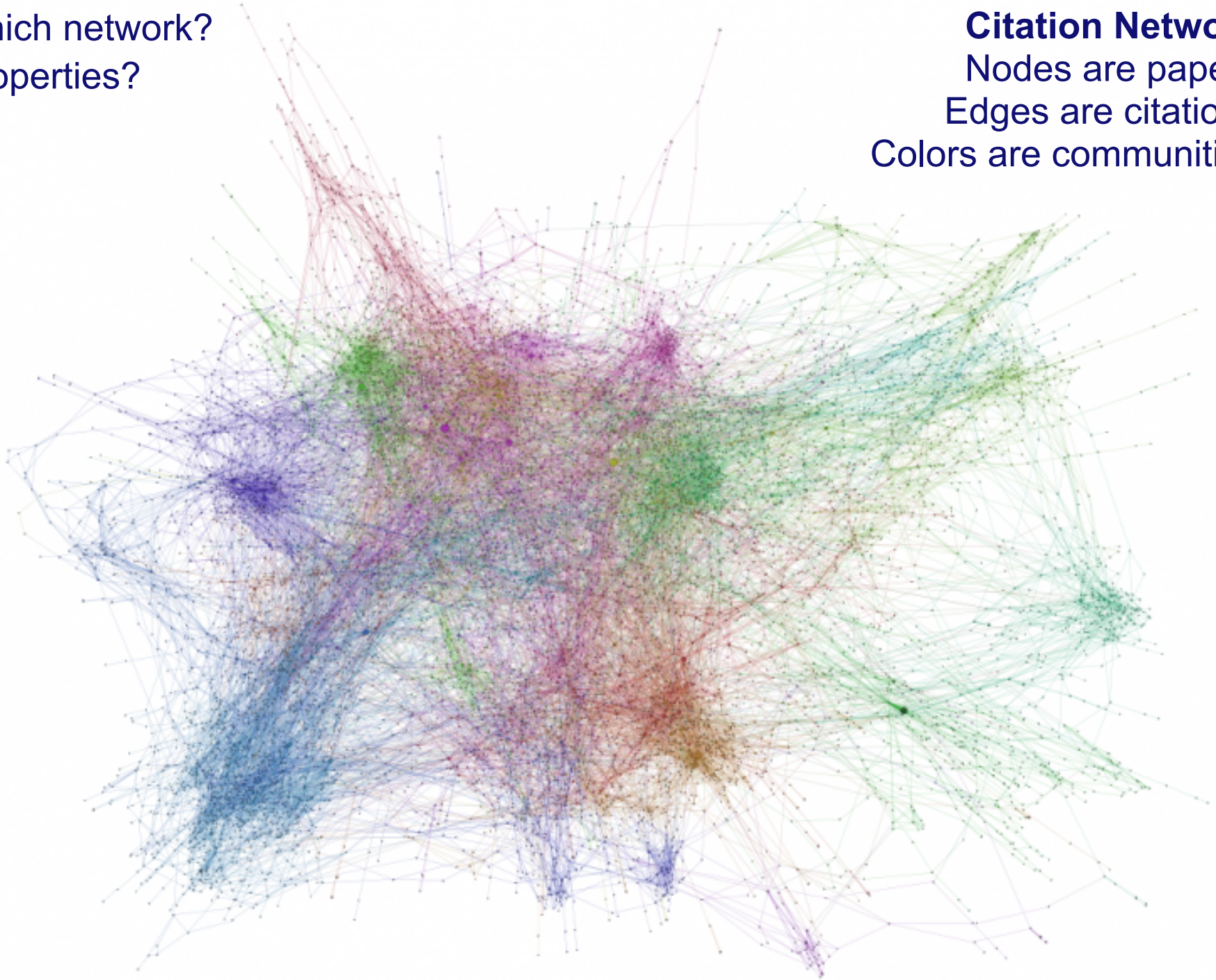


World Wide Web (2002)
Nodes are webpages
Edges are hyperlinks

SNAP - Google

Which network?
Properties?

Citation Network
Nodes are papers
Edges are citations
Colors are communities

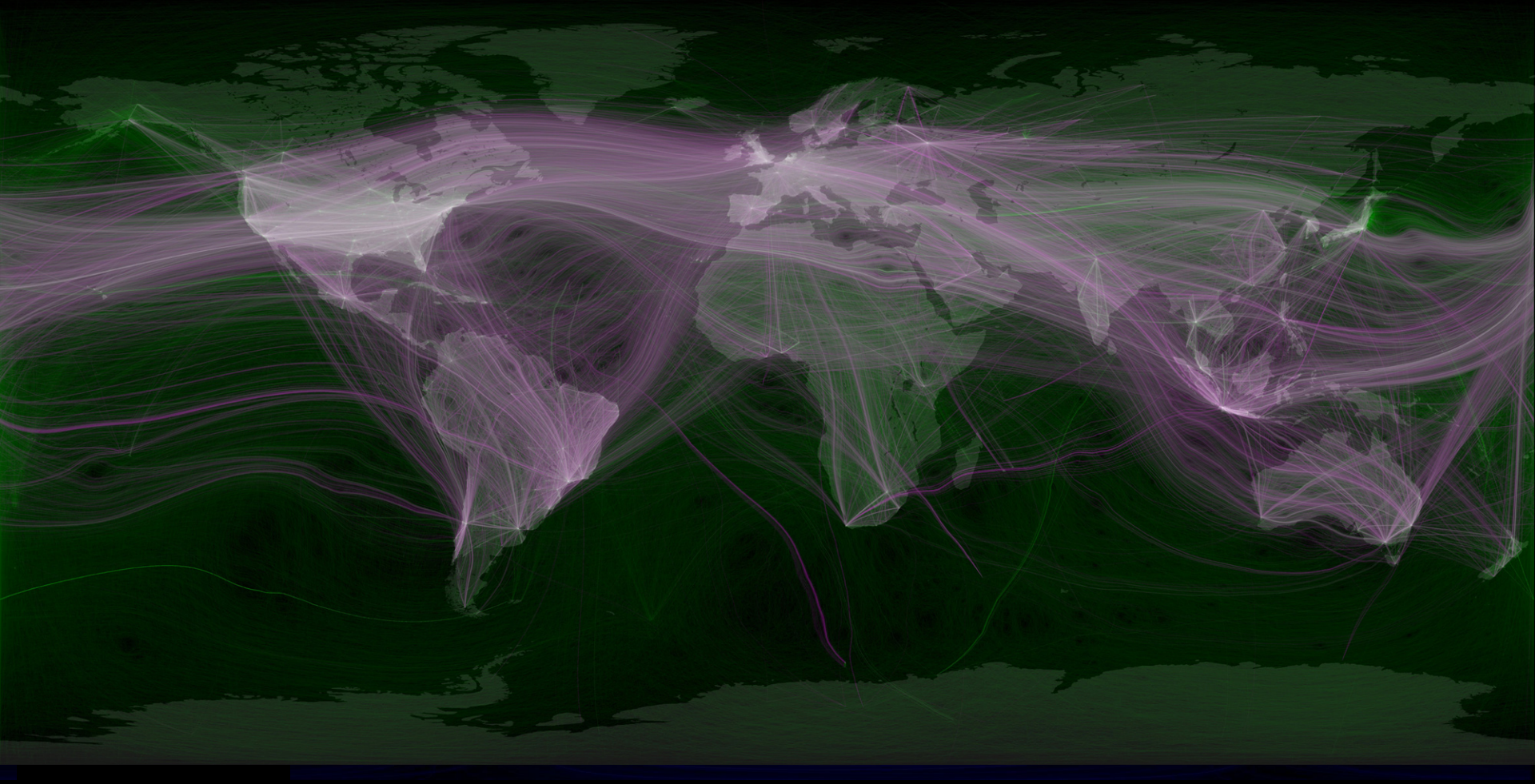


Which network?
Properties?

Facebook Graph
Nodes are people
Edges are friendships
Visualised by geolocation

Which network?
Properties?

Twitter Graph
Nodes are people
Edges are @mentions
Visualised by geolocation



Real time: http://tweetping.net/

Which network?

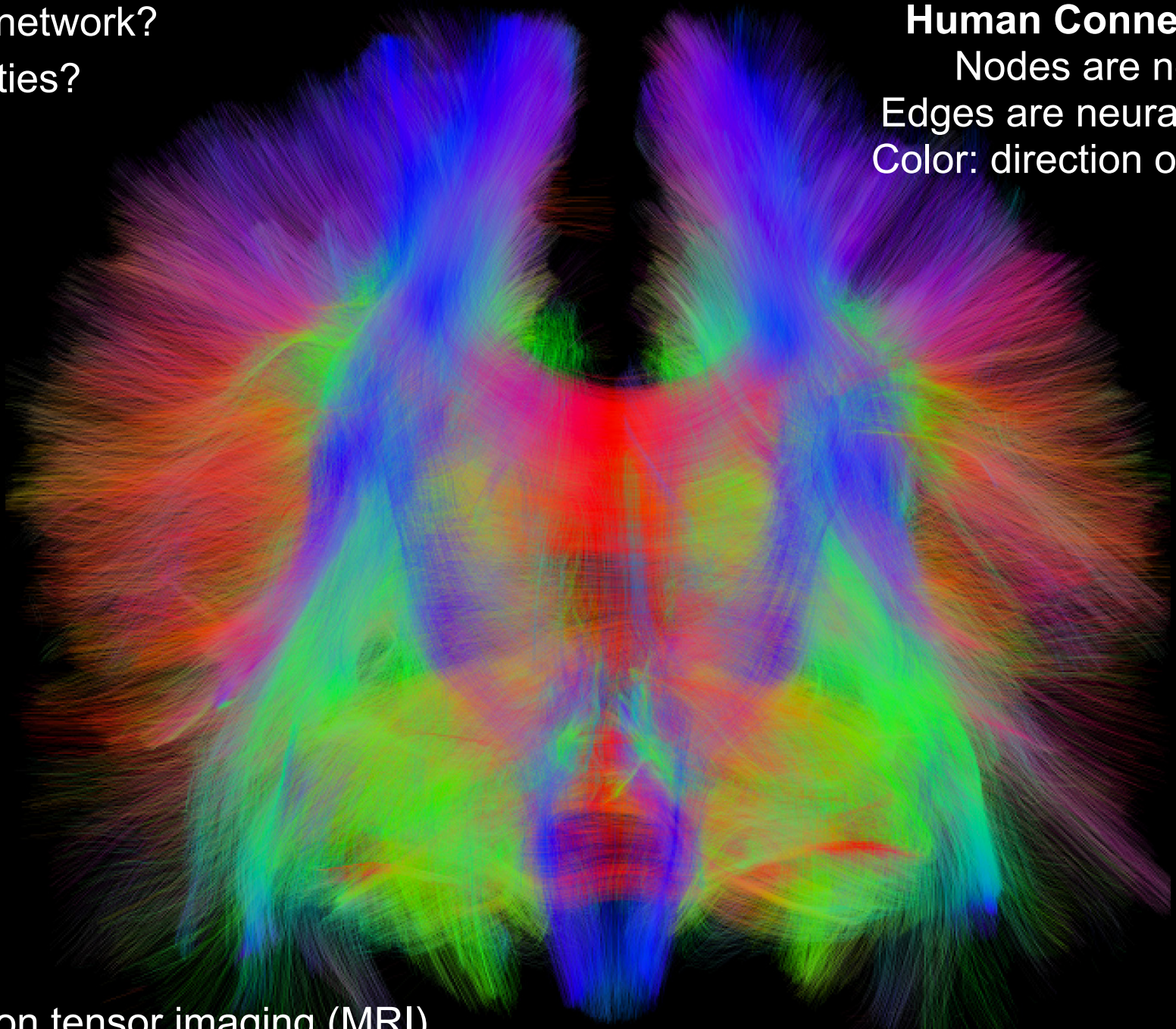
Properties?

Human Connectome

Nodes are neurons

Edges are neural tracts

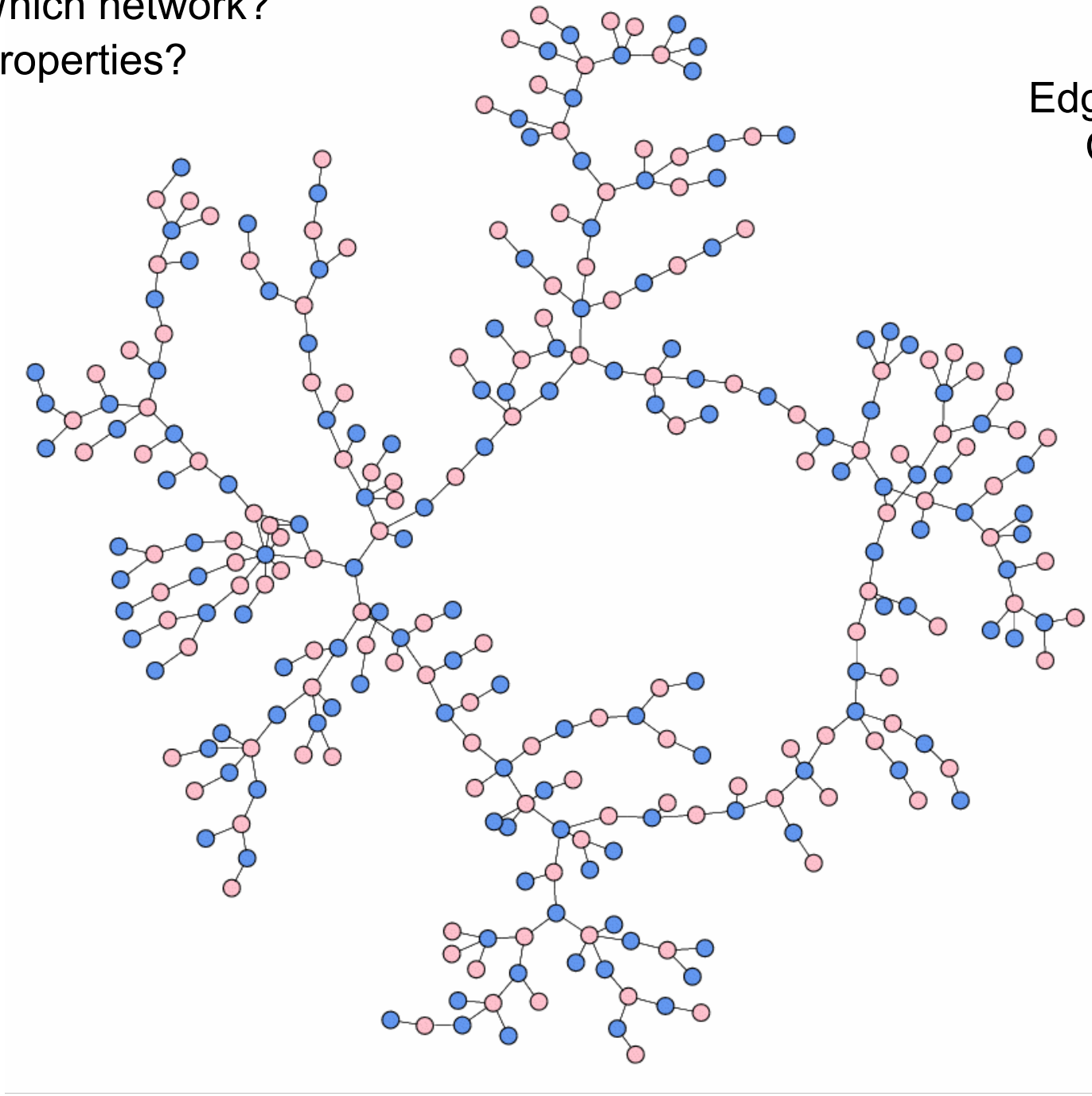
Color: direction of fibers



Diffusion tensor imaging (MRI)

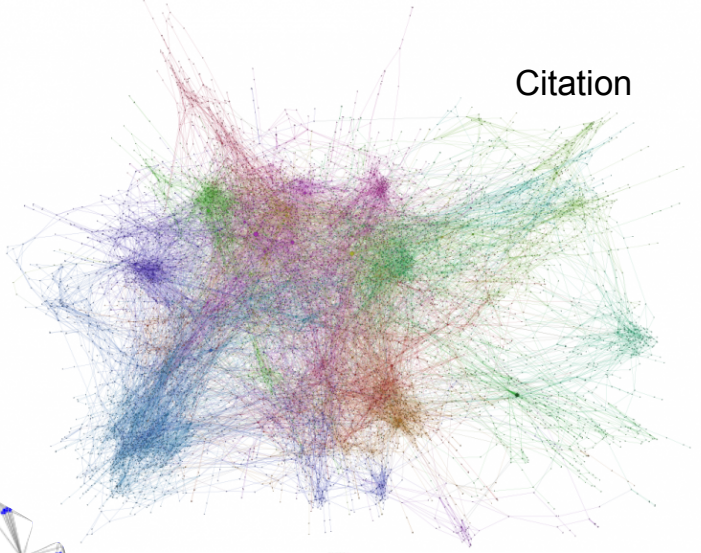
Which network?
Properties?

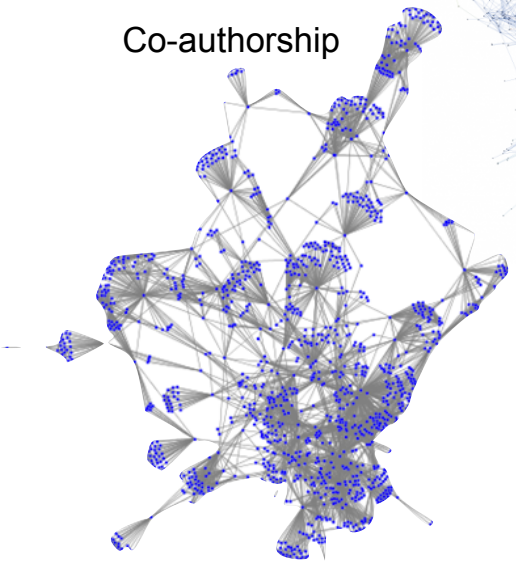
Dating Graph
Nodes are people
Edges are relationships
Colors are boys/girls

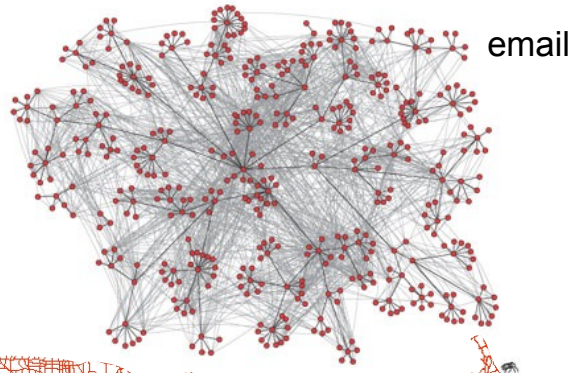


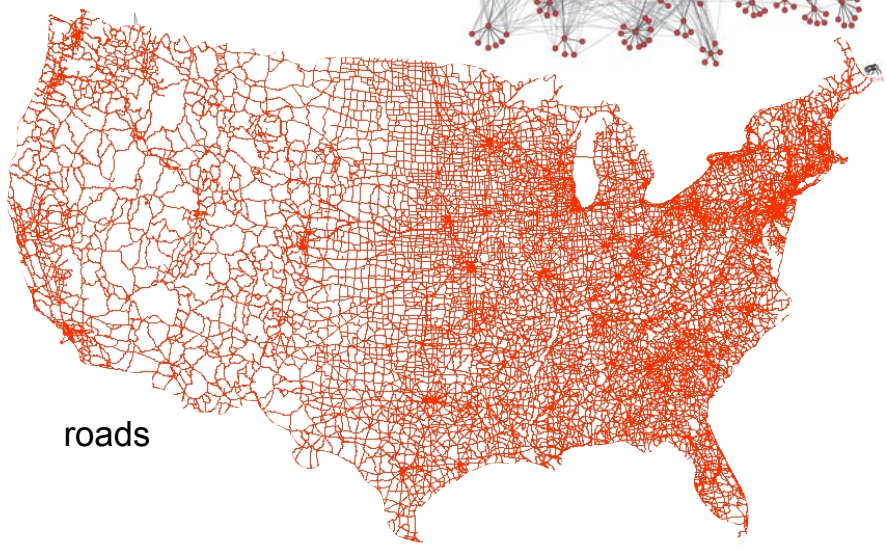
Types of real-world networks

- **Information networks:**
  - World Wide Web: hyperlinks
  - Citation networks
  - Blog networks
- **Social networks:**
  - Organisational networks
  - Communication networks
  - Collaboration networks
  - Sexual networks
  - Email
- **Technological networks:**
  - Power grid
  - Airline, road, river networks
  - Telephone networks
  - Internet
  - Autonomous systems



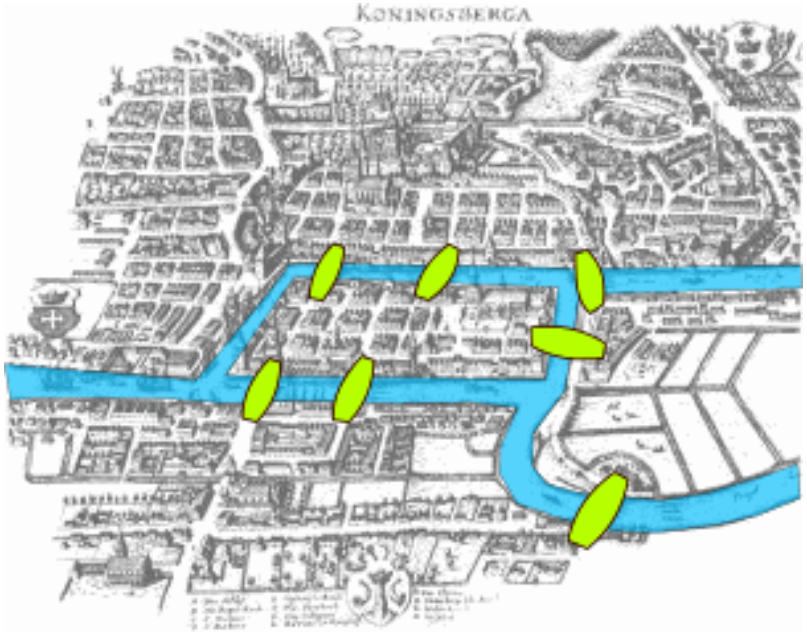


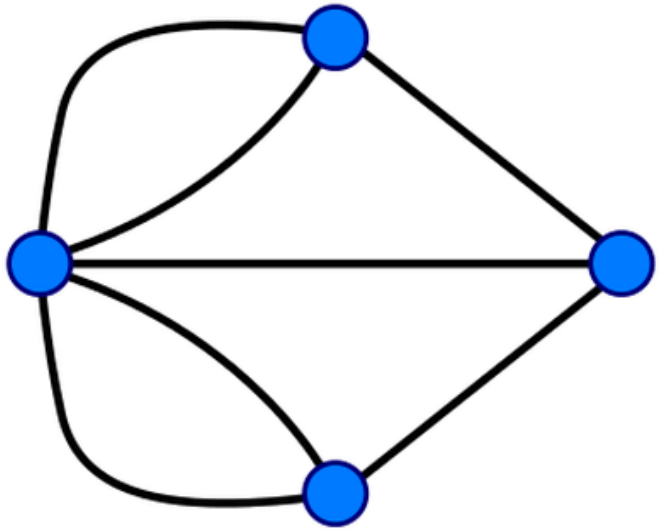




First graph formulations

- Seven Bridges of Königsberg, over the river Pregel
- Find a walk that would cross each bridge once and only once
- Euler proved this is impossible using 'graph' theory (1735)
- *Euler walks* are possible if exactly zero or two nodes have an odd number of edges.





Random graphs (1959)

- Random graph model (Erdos-Renyi model, Poisson random graph model):
  - Given n vertices connect each pair i.i.d. with probability p
- Very nice theoretical results
  - E.g. cocktail party, guests mix randomly, how fast does information spread?
- How good ("realistic") is this graph generator?

Small-world effect (1963)

- Six degrees of separation [Milgram 60s]
  - 300 random people in Nebraska were asked to send letters to stockbrokers in Boston
  - Letters can only be passed to first-name acquaintances
  - Only 25% letters reached the goal
  - But they reached it in about **6** steps

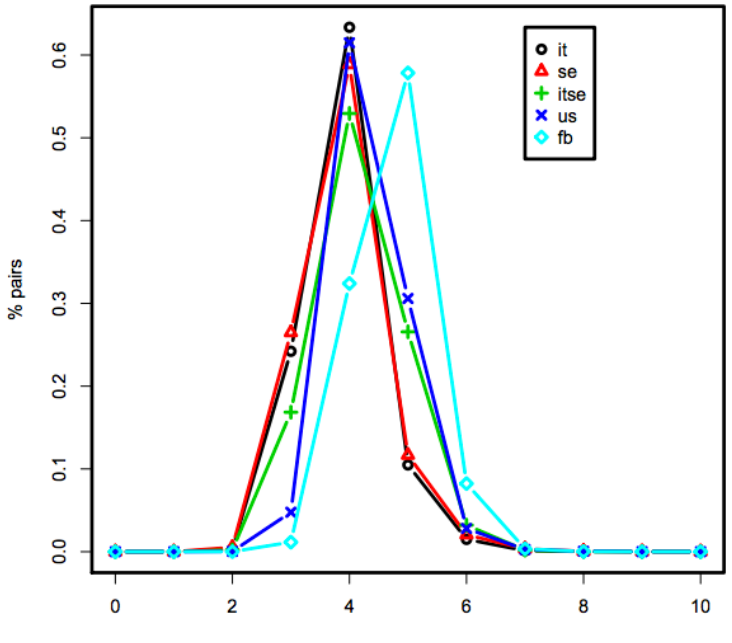
What do we measure?

- Measuring path lengths:
  - Diameter (longest shortest path): $max\ d_{ij}$
  - Effective diameter: distance at which 90% of all connected pairs of nodes can be reached
  - Mean geodesic (shortest) distance l

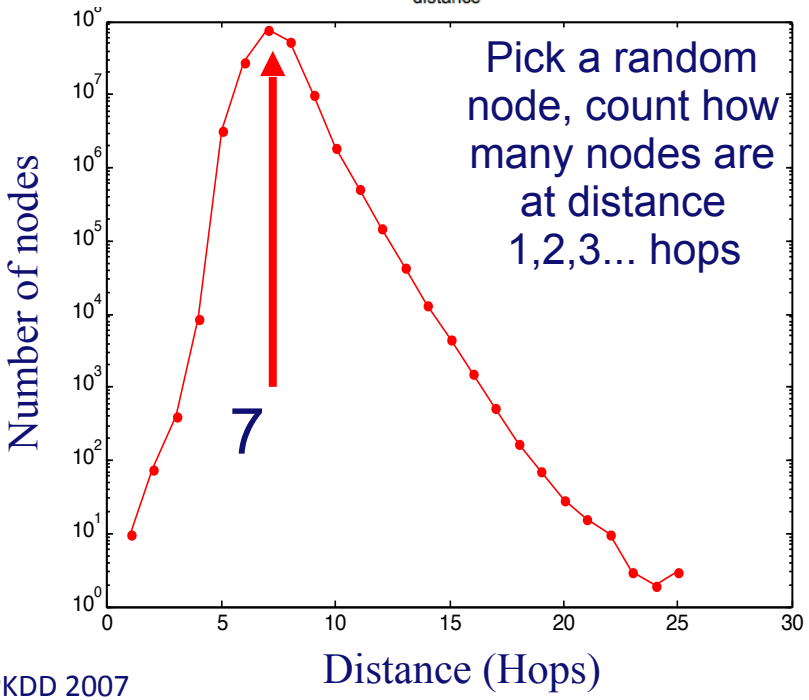
$$\ell = \frac{1}{\frac{1}{2}n(n+1)} \sum_{i \geq j} d_{ij}$$

Small-world effect

- Complete Facebook graph
  - 721 million people, 69 billion edges
  - Edge if two people are friends
- Average distance is 4.74, thus 3.74 degrees of separation, decreasing



- Microsoft Messenger network (2006)
  - 180 million people, 1.3 billion edges
  - Edge if two people exchanged at least one message in one month
  - Average around 5.5



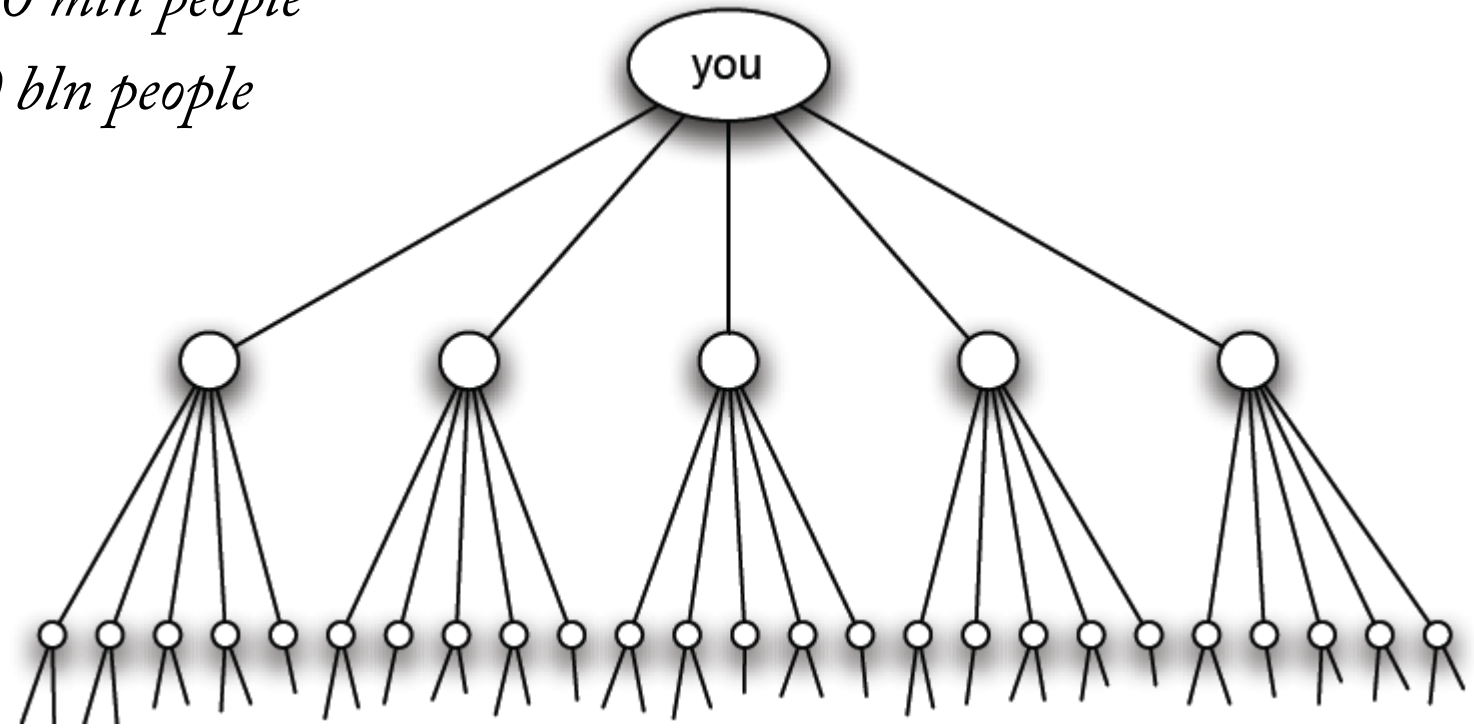
Small-world effect

- Implications (why is this interesting?):
  - Information (viruses) spread quickly
  - Robust networks (biological systems)
  - Synchronization occurs easily (brain, hands clapping)
  - Erdos numbers are small
- Shortest paths exists, humans are able to find it:
  - People only know their friends
  - People do not have the global knowledge of the network
- Suggests something about structure of network
  - In random network, shortest paths would be hard to find

Is it surprising that diameter is so small?

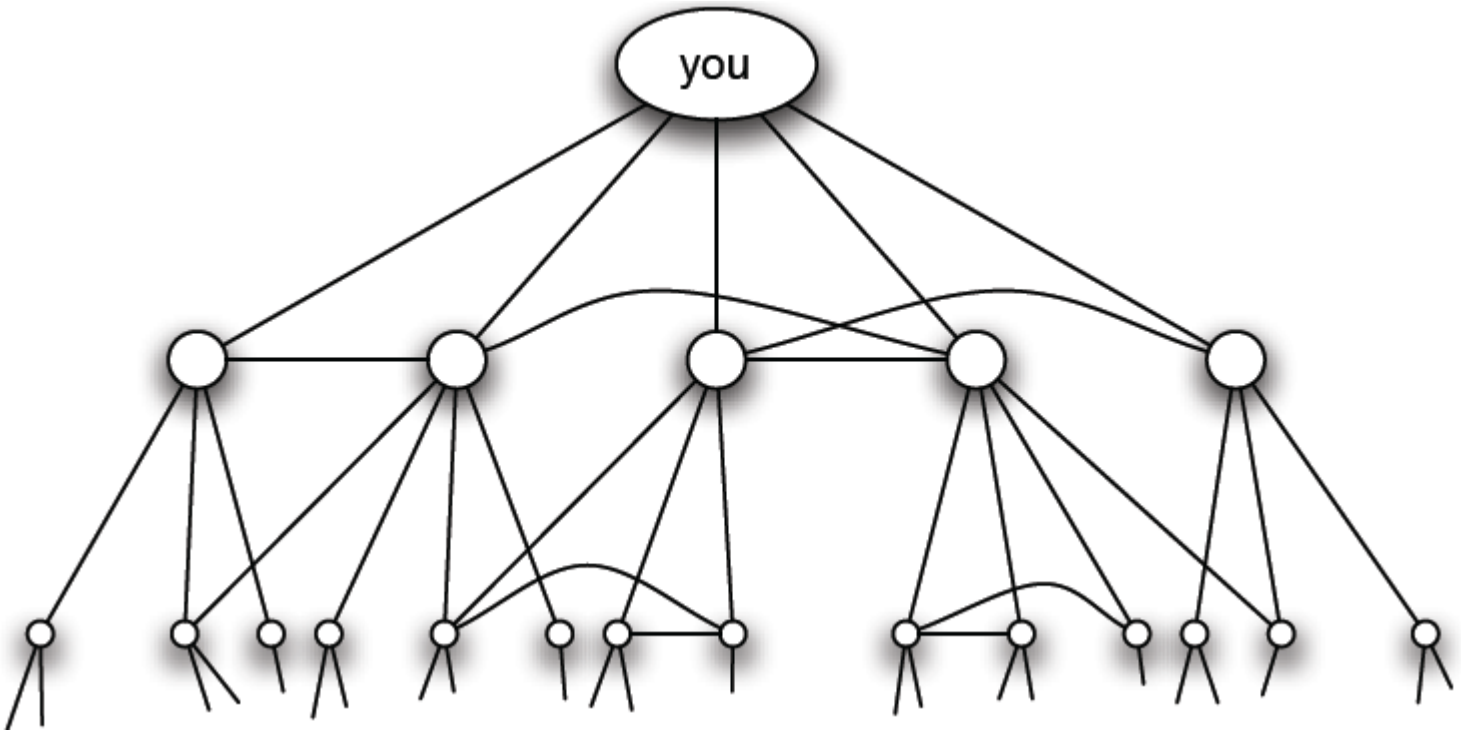
Pure exponential growth produces a small world

- *Assume every person knows 100 people*
- *2nd step 10 thousand people*
- *3rd step 1 mln people*
- *4th step 100 mln people*
- *5th step 10 bln people*



Triadic closure reduces the growth rate

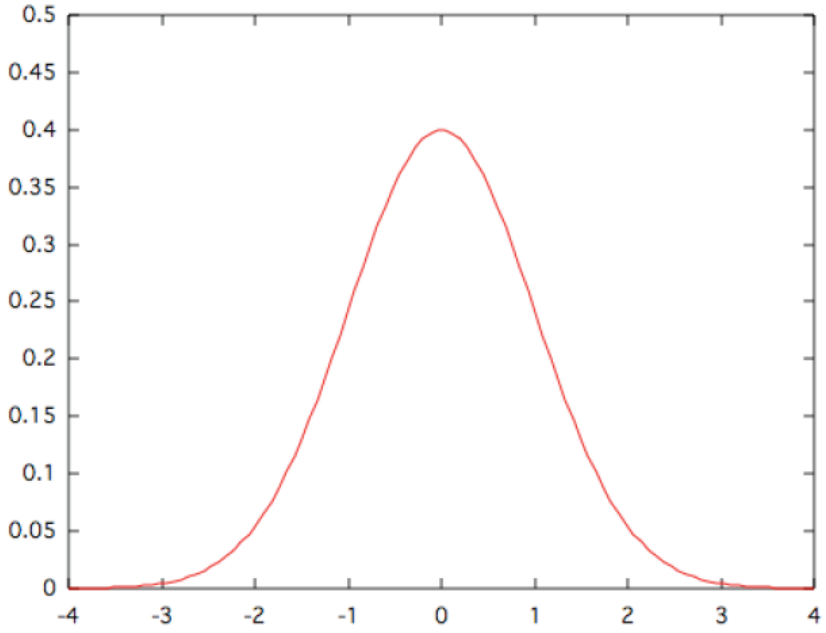
- many of the edges go from one friend to another, not to the rest of world
- social networks tend to be highly clustered, do not exhibit massively branching structure



Expectations about popularity of nodes in G

- As a function of k, what fraction of pages on the Web have k in-links?
- A Simple Hypothesis: The Normal Distribution
  - a natural guess in our case, since it is ubiquitous across the natural sciences
  - *Central Limit Theorem*
    - any quantity that can be viewed as the sum of many small independent *random* effects will be well-approximated by the normal distribution

    - E.g. Height of people

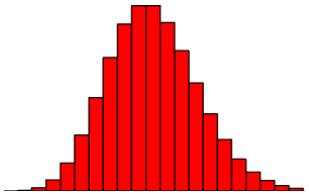


Expectations vs. reality

- If we assume that each page decides at random whether to link to any other page, then we'd expect it to be normally distributed.
- In this model, the number of pages with *k in-links should decrease exponentially in k, as k grows large.*
- *But taking a snapshot of the Web we can see that* the fraction of Web pages that have *k in-links is* approximately proportional to $1/k^2$ *(vs.* 2^{-k}*)*

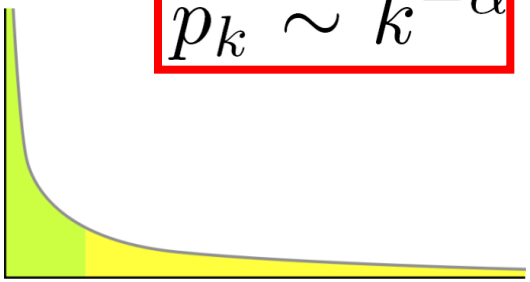
Degree distributions (1)

- Let p_k denote a fraction of nodes with degree k *(nr links)*
- We can plot a histogram of p_k vs. k
- In a (Erdos-Renyi) random graph degree distribution follows Poisson distribution



- Degrees in real networks are heavily skewed to the right
- Distribution has a long tail of values that are far above the mean
  - Power-law, Zipf's law, Long tail, Heavy-tail
- Many things follow Power-law:
  - Amazon sales,
  - word length distribution,
  - Wealth, Earthquakes, …

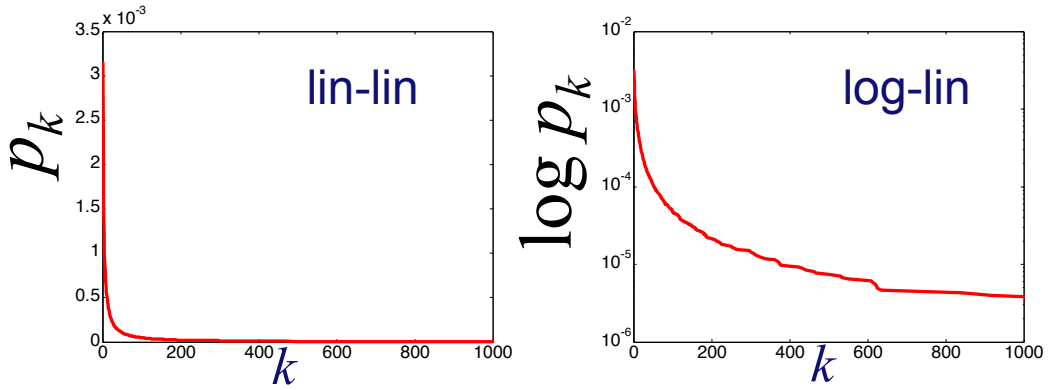
$$p_k \sim k^{-\alpha}$$

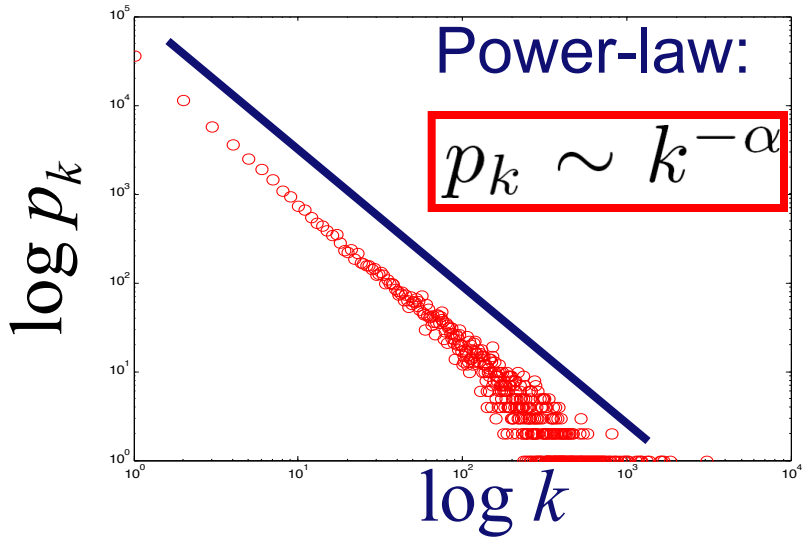


Degree distributions (2)

- Many real world networks contain hubs: highly connected nodes
- We can easily distinguish between exponential and power-law tail by plotting on log-lin and log-log axis
- Power-law is a line on log-log plot

Degree distribution in a blog network
(plot the same data using different scales)





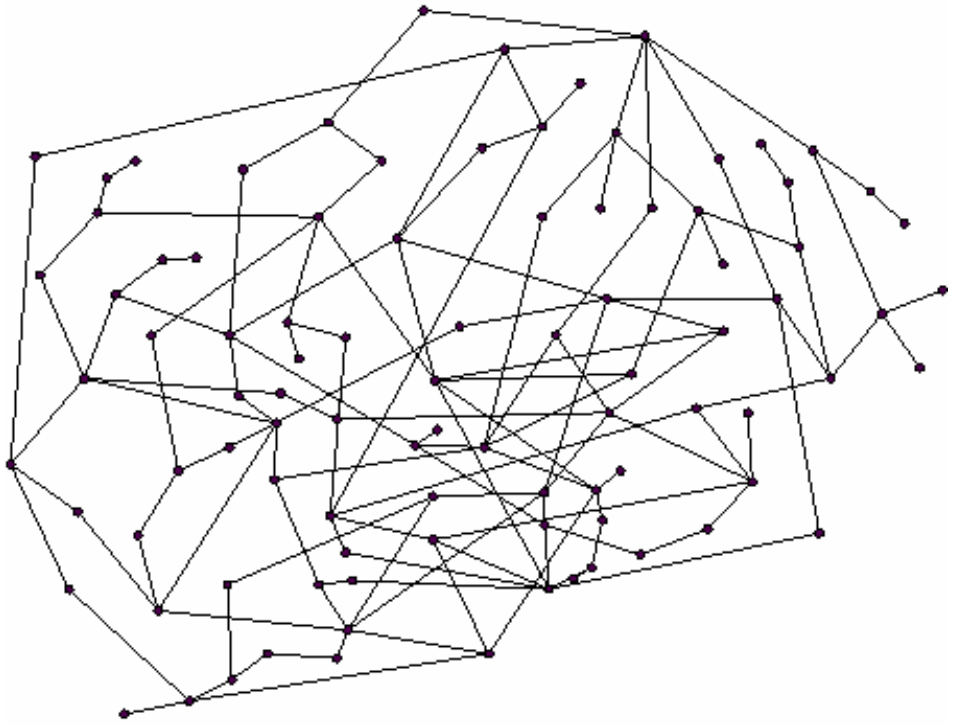
How can this phenomenon be explained?

- What underlying process is keeping the line so straight?
- Rich-get-richer models
  - Preferential attachment
    - Links are formed "preferentially" to pages that already have high popularity
- What are the implications for
  - browsing behavior?
  - finding information with SE?
  - Amazon-like stores and their business models?

Power laws and long tails: reflections

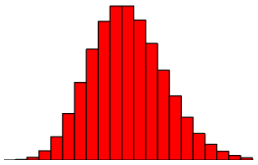
- Popularity as a Network Phenomenon
- Power laws vs. normal distribution
  - the number of highly *popular items* the total sales volume of *unpopular items* are much higher than normal distribution may suggest
- Preferential attachment paradigm (rich-get-richer)
  - should be used instead of the central limit theorem for the worlds affected by popularity
- The Unpredictability of Rich-Get-Richer Effects
  - 9 "parallel" copies of the site with music download
    - Feedback influences people
    - Success of one phenomenon (e.g. Harry Potter) cannot simply be 'repeated'

Poisson vs. Scale-free network

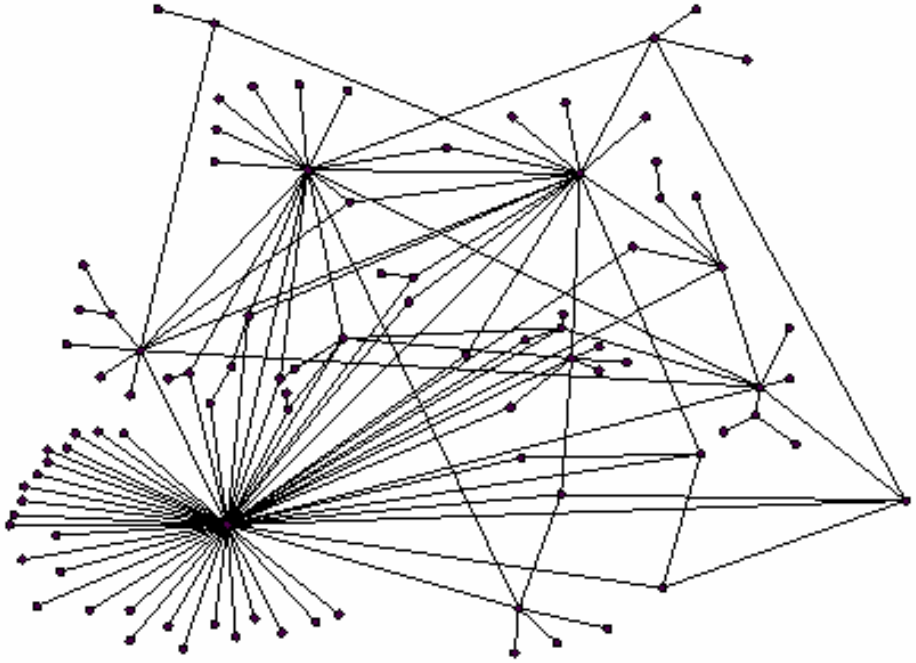


Poisson network

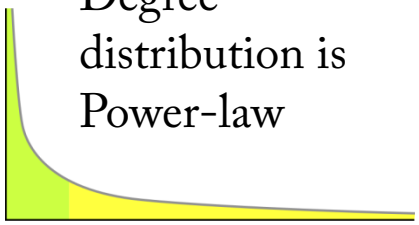
(Erdos-Renyi random graph)



Degree distribution is Poisson



Scale-free (power-law) network



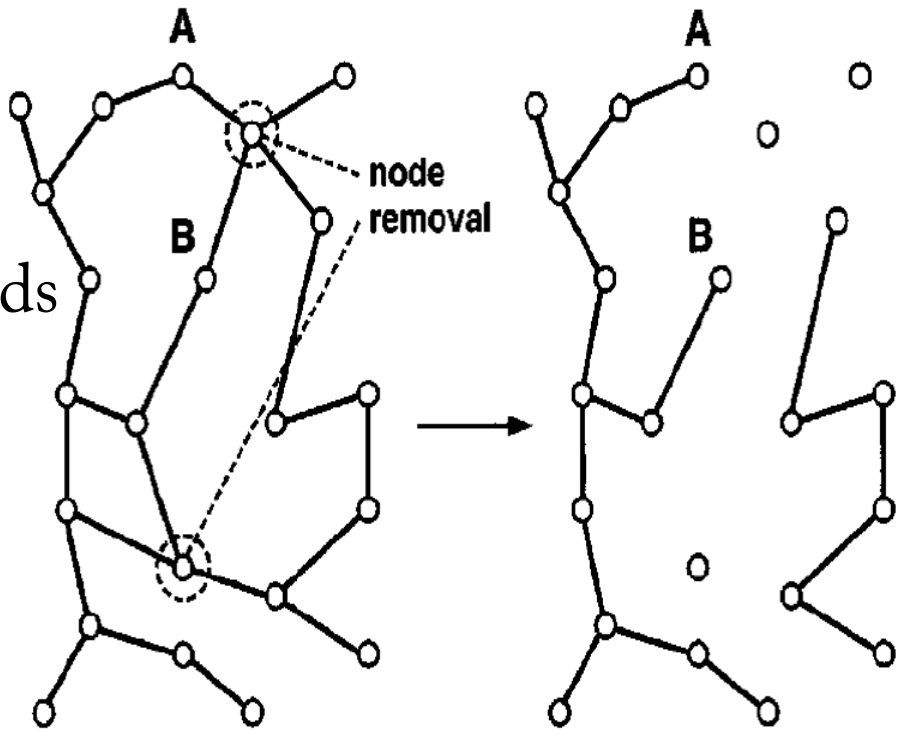
Degree distribution is Power-law

Function is scale free if:

$f(ax) = c\,f(x)$

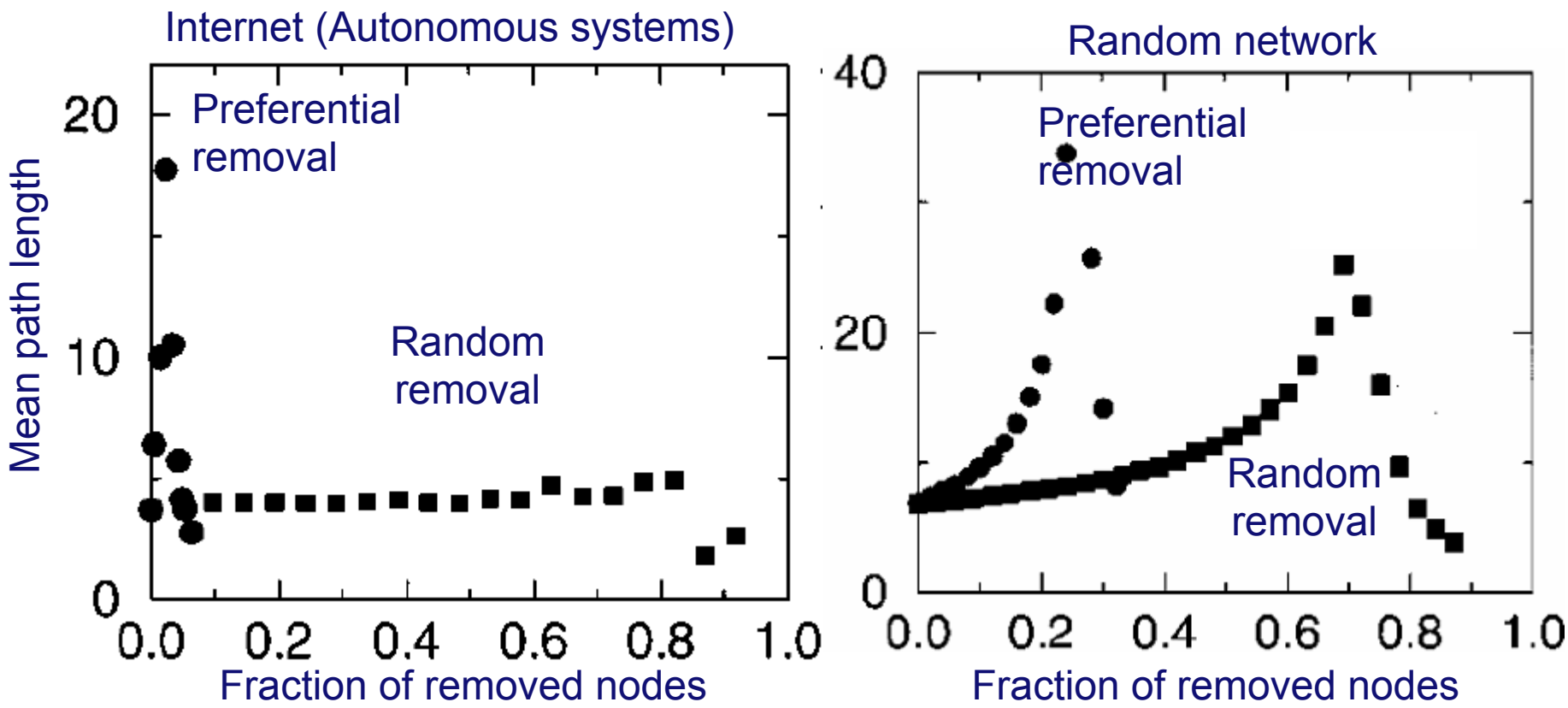
Network resilience (1)

- How the connectivity (length of the paths) of the network changes as the vertices get removed
- Vertices can be removed:
  - Uniformly at random
  - In order of decreasing degree
- Important e.g. for epidemiology
  - Removal of vertices corresponds to vaccination



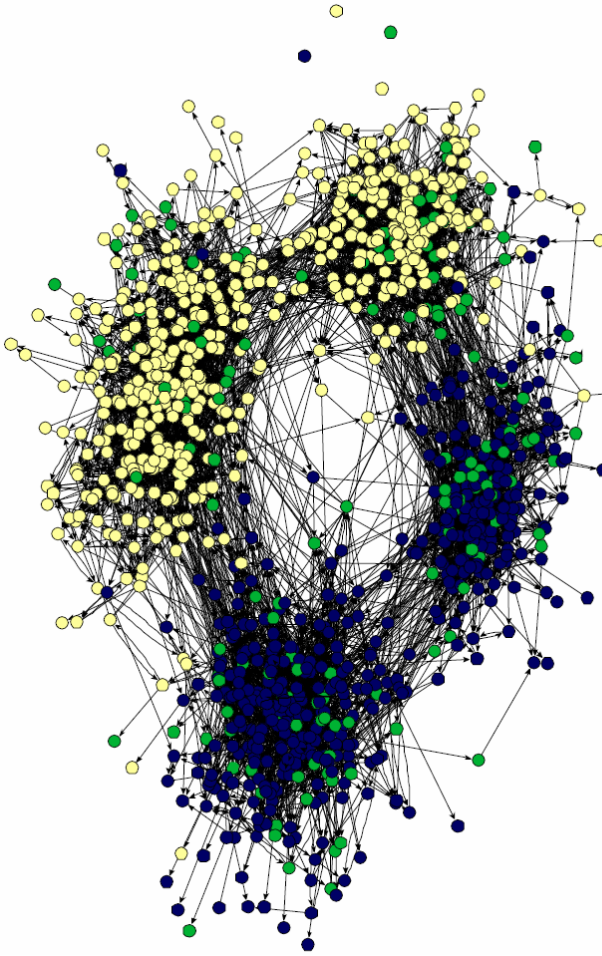
Network resilience (2)

- Real-world networks are resilient to random attacks
  - One has to remove all web-pages of degree > *5* to disconnect the web
  - But this is a very small percentage of web pages
- Random network has better resilience to targeted attacks



Community structure

- Most social networks show community structure
  - groups have higher density of edges within than across groups
  - People naturally divide into groups based on interests, age, occupation, …
- How to find communities (not our focus):
  - Spectral clustering (embedding into a low-dim space)
  - Hierarchical clustering based on connection strength
  - Combinatorial algorithms (min cut style formulations)
  - Block models
  - Diffusion methods



Friendship network of children in a school

What about evolving graphs?

- Conventional wisdom/intuition:
  - Every new node brings about same number of new links
  - Constant average degree: the number of edges grows linearly with the number of nodes
  - Slowly growing diameter: as the network grows the distances between nodes grow
  - **But** analysis of real large scale networks shows that this is not true!

Networks over time: Densification

- A simple question: What is the relation between the number of nodes and the number of edges in a network over time?
- Let:
  - $N(t)$ … nodes at time t
  - $E(t)$ … edges at time t
- Suppose that:

$$N(t+1) = 2 * N(t)$$

- Q: what is your guess for

$$E(t+1) =? 2 * E(t)$$

- A: over-doubled!
  - But obeying the Densification Power Law

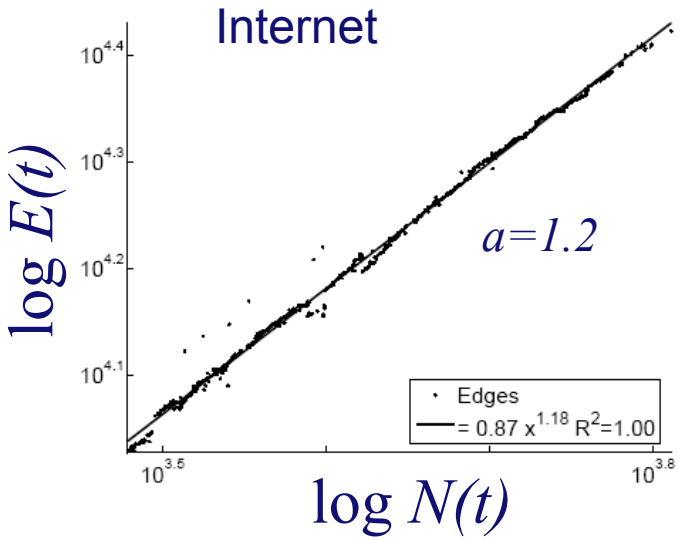
Networks over time: Densification

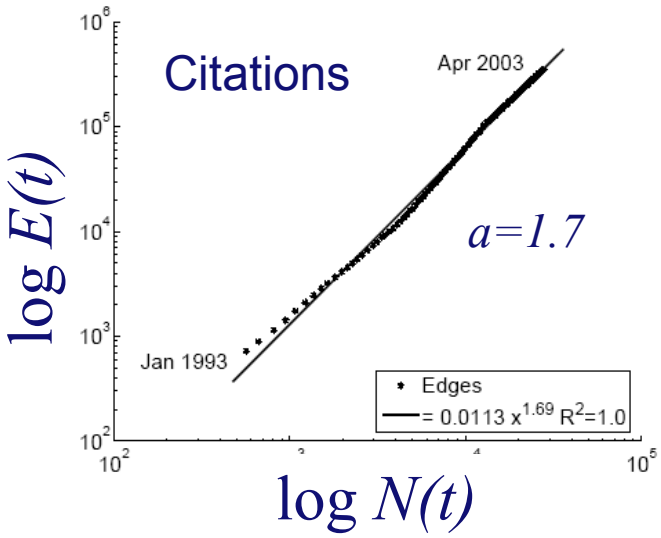
- Networks become denser over time
- The number of edges grows faster than the number of nodes – average degree is increasing

$$E(t) \propto N(t)^a$$

a -> densification exponent

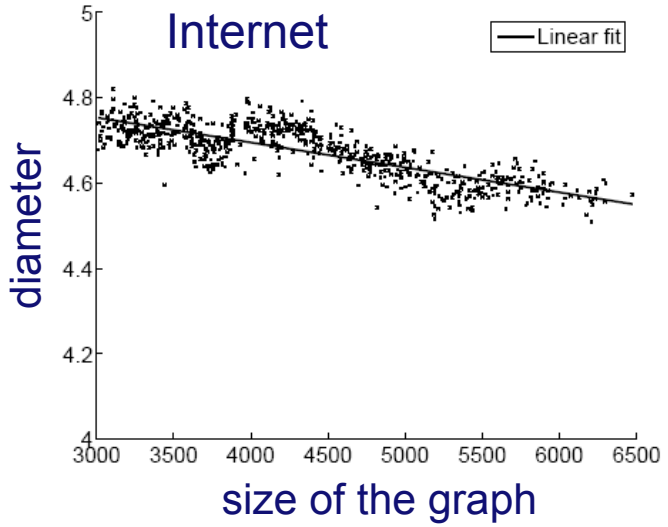
- *1* ≤ *a* ≤ *2*:
  - *a=1*: linear growth – constant out-degree, assumed in (outdated) literature
  - *a=2*: quadratic growth – everyone connected to everyone (clique)

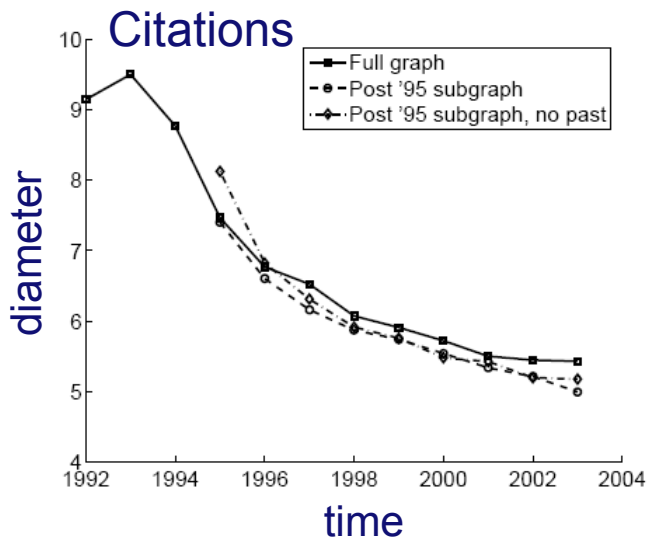




Shrinking diameters

- Intuition and prior work say that distances between the nodes slowly grow as the network grows (like *log n*):
  - *d* ~ *O(log N)*
  - *d* ~ *O(log log N)*
- Diameter Shrinks/Stabilizes over time
  - as the network grows the distances between nodes slowly decrease [KDD 05]





Properties hold in many graphs

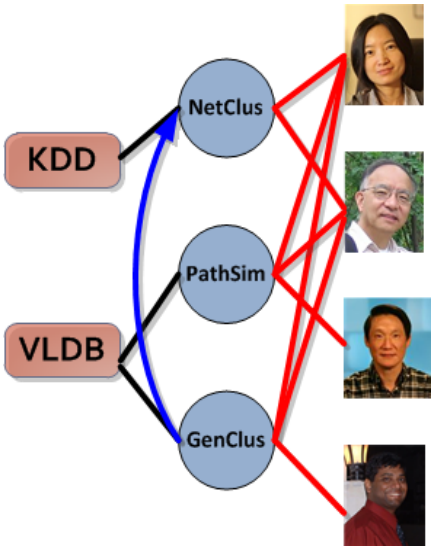
- These patterns can be observed in many real world networks:
  - World wide web [Barabasi]
  - On-line communities [Holme, Edling, Liljeros]
  - Who call whom telephone networks [Cortes]
  - Internet backbone – routers [Faloutsos, Faloutsos, Faloutsos]
  - Movies to actors network [Barabasi]
  - Science citations [Leskovec, Kleinberg, Faloutsos]
  - Click-streams [Chakrabarti]
  - Autonomous systems [Faloutsos, Faloutsos, Faloutsos]
  - Co-authorship [Leskovec, Kleinberg, Faloutsos]
  - Sexual relationships [Liljeros]

Similarity/Distances in networks

- Popularity measures: **PageRank**
- Homogeneous networks: **SimRank**
  - x and y are similar if they are related to similar objects
- Heterogeneous networks: **PathSim**
  - x and y are similar if there are many paths between them following a given meta-path (e.g. image-tag-image)
- Ranking-based clustering:
  - Bipartite networks: **RankClus**
    - Build clusters, rank items within cluster, iterate
  - General networks: **NetClus**
    - Build clusters, build model to predict clusters, iterate

Heterogeneous networks

- Multiple object types and/or multiple link types



Venue Paper Author

DBLP Bibliographic Network

The Facebook Network

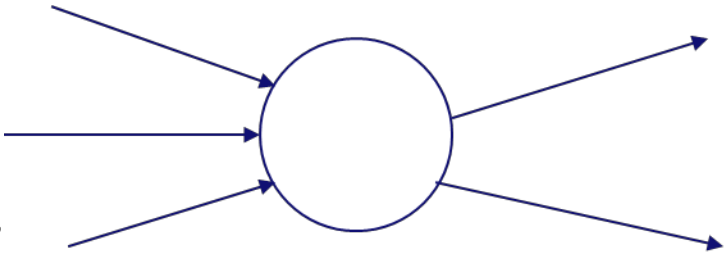
The IMDB Movie Network

Homo/heterogeneous networks

- Homogeneous networks can often be derived from their original heterogeneous networks
  - E.g., coauthor networks can be derived from author-paper-conference networks by projection on authors only
- Heterogeneous network carries richer information than its corresponding projected homogeneous networks
- Typed heterogeneous networks vs. non-typed networks. (i.e., not distinguishing different types of nodes)
  - Typed nodes and links imply a more structured network, and thus often lead to more informative discovery

Query-independent scoring/ranking

- 1st generation: using link counts as simple measures of popularity; two basic suggestions:
  - Undirected popularity:
    - Score of a page = # in-links + # of out-links (3+2=5)
  - Directed popularity:
    - Score of a page = # of its in-links (3).



- Query processing:
  - retrieve all pages meeting the text query;
  - order these by their link popularity.
- Spamming with such simple heuristics (so your page gets a high score) is easy

First ranking algorithms based on link analysis

Boolean spread

- given page *p* in result set,
- extend result set with pages that point to and are pointed by page *p*

Vector spread

- given page *p* in result set,
- a page *p* is assigned a score proportional to number of query words contained in other pages that point to *p*

Most-cited

- Simply order the results matching a query by the number of in-links to each page

PageRank

- PageRank interprets a hyperlink from page v to page u as a vote (an implicit conveyance of authority) by v, for u.
  - The more in-links that u receives, the more prestige it has.
- Pages that point to page u also have their own prestige scores.
  - A page of a higher prestige pointing to u is more important than a page of a lower prestige pointing to u.
  - In other words, a page is important if it is pointed to by other important pages.

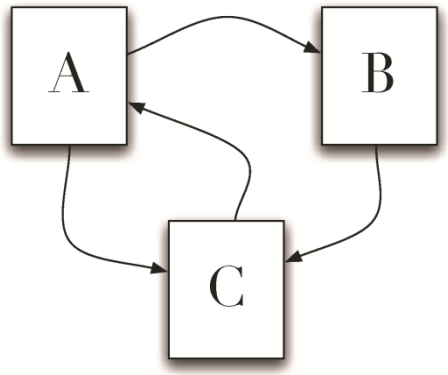
$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L_v}$$

L_v is the number of outgoing links from v

B_u contains all pages linking to u

PageRank toy example

- PR(A) = PR(C)/1
- PR(B) = PR(A)/2
- PR(C) = *PR*(A)/2 + *PR*(B)/1



PageRank

- Don't know PageRank values at start
- Assume equal values (1/3 in this case)
- Then iterate:
  - first iteration:

PR(C) = 0.33/2 + 0.33 = 0.5, PR(A) = 0.33, PR(B) = 0.17

  - second:

PR(C) = 0.33/2 + 0.17 = 0.33, PR(A) = 0.5, PR(B) = 0.17

  - third:

PR(C) = 0.42, PR(A) = 0.33, PR(B) = 0.25

  - Converges to

PR(C) = 0.4, PR(A) = 0.4, and PR(B) = 0.2

PageRank scoring

- Imagine a browser doing a random walk on web:
  - Start at a random page
  - At each step, go out of the current page along one of the links on that page, equiprobably
- "In the steady state" each page has a long-term visit rate - use this as the page's score.
- PageRank of a page is the probability that the "random surfer" will be looking at that page
  - links from popular pages will increase PageRank of pages they point to

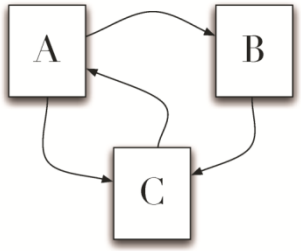
What is missing?

- The web is full of dead-ends, webpages that
  - do not have outgoing links or
  - contain only (broken) links that no longer point to other pages
  - may also be links to pages that have not yet been crawled
  - have links forming a loop
- Random walk can get stuck in dead-ends or loops.
  - Makes no sense to talk about long-term visit rates.

Teleporting

- Teleporting: at a dead end, jump to a random web page.
  - At any non-dead end, with probability 10%, jump to a random web page.
  - With remaining probability (90%), go out on a random link.
  - Now cannot get stuck locally.
- Teleporting makes Web graph ergodic
  - There is a path from any page to any other page
  - Disregarding where we start, the probability of being at any page at a fixed time is nonzero
  - Important for having theoretical guarantees

PageRank



- Taking random page jump into account, for our toy example consisting of 3 nodes, there is 1/3 chance of going to any page when we do teleporting
  - PR(C) = λ/3 + (1 – λ) · (PR(A)/2 + PR(B)/1)
- More generally,

$$PR(u) = \frac{\lambda}{N} + (1-\lambda).\sum_{v \in B_u} \frac{PR(v)}{L_v}$$

  - where N is the number of pages, λ typically {0.1, 0.15}

Markov chains (abstraction of random walks)

- A Markov chain consists of n states, plus an $n \times n$ transition probability matrix **P**.
- At each step, we are in exactly one of the states.
- For $1 \le i,j \le n$, the matrix entry P_{ij} tells us the probability of j being the next state, given we are currently in state i.

$$\sum_{j=1}^{n} P_{ij} = 1.$$

Computing P itself is rather trivial

- The adjacency matrix A of the web graph:
  - if there is a hyperlink from page i to page j, then $Aij = 1$, otherwise $Aij = 0$.
- If a row of A has no 1's, then replace each element by 1/N. For all other rows proceed as follows:
- Divide each 1 in A by the number of 1's in its row.
  - Thus, if there is a row with three 1's, then each of them is replaced by 1/3.
- Multiply the resulting matrix by $1 - \lambda$.
- Add λ/N to every entry of the resulting matrix, to obtain P.

Ergodic Markov chains

- A Markov chain is ergodic if
  - you have a path from any state to any other
  - For any start state, after a finite transient time T_0, the probability of being in any state at a fixed time $T > T_0$ is nonzero.

- For any ergodic Markov chain, there is a unique long-term visit rate for each state.
  - *Steady-state probability distribution*.
- Over a long time-period, we visit each state in proportion to this rate.
- It doesn't matter where we start.

Probability vectors

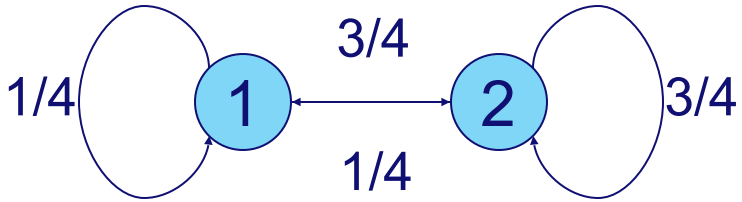
- A probability (row) vector $\mathbf{x} = (x_1, \ldots x_n)$ tells us where the walk is at any point
  - the walk is in state i with probability x_i

$$\sum_{i=1}^{n} x_i = 1.$$

- Row i of the transition probabilities matrix $\mathbf{P}$ tells us where we go next from state i.
  - From $\mathbf{x}$, our next state is distributed as $\mathbf{xP}$.

Steady state

- The steady state looks like a vector of probabilities $\mathbf{a} = (a_1, \ldots a_n)$:
  - a_i is the probability that we are in state i.



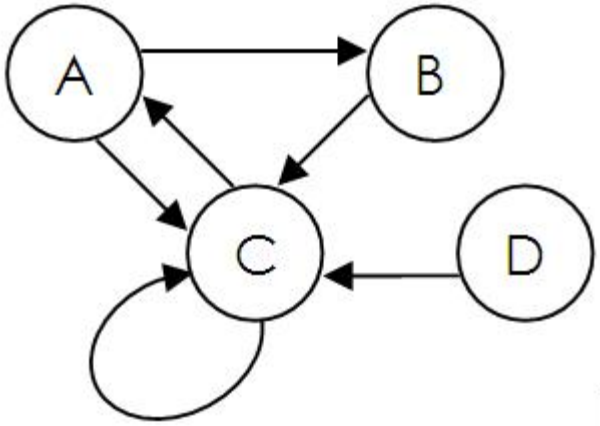
How do we compute steady state vector?

- Let **a** = $(a_1, \ldots a_n)$ denote the row vector of steady-state probabilities.
- If our current position is described by **a**, then the next step is distributed as **aP**.
- But **a** is the steady state, so **a**=**aP**.
- Solving this matrix equation gives us **a**.
  - So **a** is the (left) eigenvector for **P**.
  - (Corresponds to the *principal eigenvector* of **P** with the largest eigenvalue, which is in this case always equal to 1)
- How to solve linear equations?
  - Gaussian elimination? $O(n^3)$, n – number of equations
    - for 10s-100s billion nodes, it is not feasible

Another way of computing *a*

- Recall, regardless of where we start, we eventually reach the steady state **a**.
- Start with any distribution (say **x**=(*10…0*)).
- After one step, we're at **xP**;
- after two steps at $\mathbf{xP}^2$, then $\mathbf{xP}^3$ and so on.
- "Eventually" means for "large" k, $\mathbf{xP}^k = \mathbf{a}$.
- Algorithm: multiply **x** by increasing powers of **P** until the product looks stable.

What is PageRank of C if λ = 0.4?



\+
=>

	A	B	C	D
A	0	0.5	0.5	0
B	0	0	1	0
C	0.5	0	0.5	0
D	0	0	1	0

	A	B	C	D
A	0.1	0.4	0.4	0.1
B	0.1	0.1	0.7	0.1
C	0.4	0.1	0.4	0.1
D	0.1	0.1	0.7	0.1

Let's use power method.
We can from any state.
Let's start with A, so $\mathbf{x}0 = (1;0;0;0)$
$\mathbf{x}_0\mathbf{P} = (0.1;0.4;0.4;0.1) = \mathbf{x}_1$;
$\mathbf{x}_1\mathbf{P} = (0.01 + 0.04 + 0.16 + 0.01;$
$0.04+ 0.04 + 0.04 + 0.01;$
$0.04 + 0.28 + 0.16 + 0.07;$
$0.01 + 0.04 + 0.04 + 0.01)$
$= (0.22; 0.13; 0.55; 0.01) = \mathbf{x}_2$

x0	1	0	0	0
x1	0.1	0.4	0.4	0.1
x2	0.22	0.13	0.55	0.1
x3	0.265	0.166	0.469	0.1
x4	0.2407	0.1795	0.4798	0.1
x5	0.24394	0.17221	0.48385	0.1
x6	0.245155	0.173182	0.481663	0.1
x7	0.244499	0.173547	0.481955	0.1
x8	0.244586	0.17335	0.482064	0.1
x9	0.244619	0.173376	0.482005	0.1

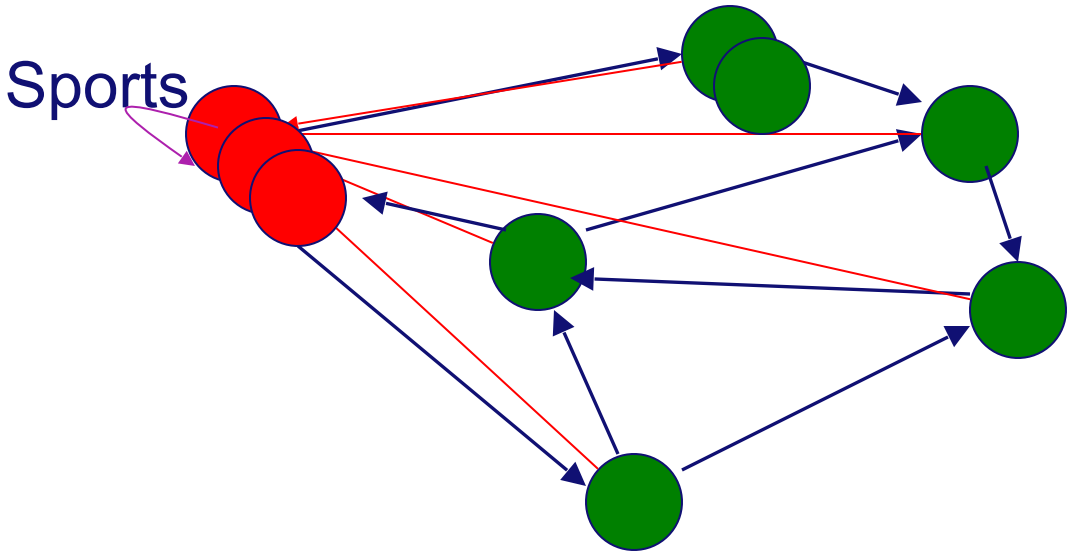
PageRank summary

- Preprocessing:
  - Given graph of links, build matrix **P**
  - From it compute **a**
  - The entry a_i is a number between 0 and 1: the pagerank of page i
- Example query processing:
  - Retrieve pages meeting query (Boolean)
  - Rank them by their pagerank
  - Order is query-*independent*
- Pagerank is used in Google, but so are many other clever heuristics

Topic Specific PageRank

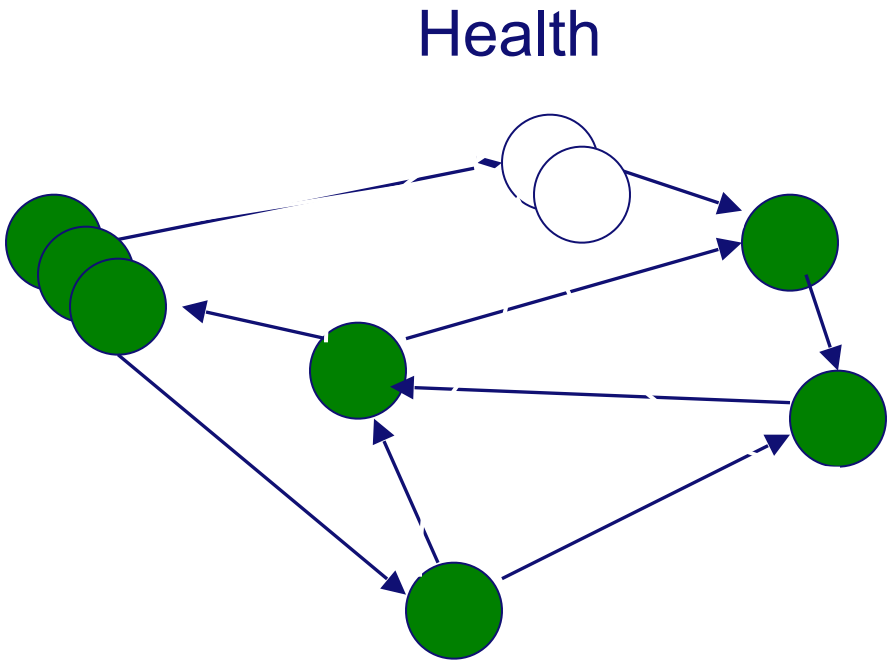
- Conceptually, teleport a random surfer as follows:
  - select a category *c* based on query/user info
  - teleport to a page uniformly at random within *c*
- Can/should we compute PR at query time?
  - **offline**: compute PR wrt to *individual* categories
    - query independent model as before
    - each page has multiple PR_c scores – one for each *c*, with teleportation only to that *c*
  - **online**: distribution of weights over categories computed by query context classification
    - Generate a dynamic PR score for each page - weighed sum of category-specific PR's

Non-uniform Teleportation



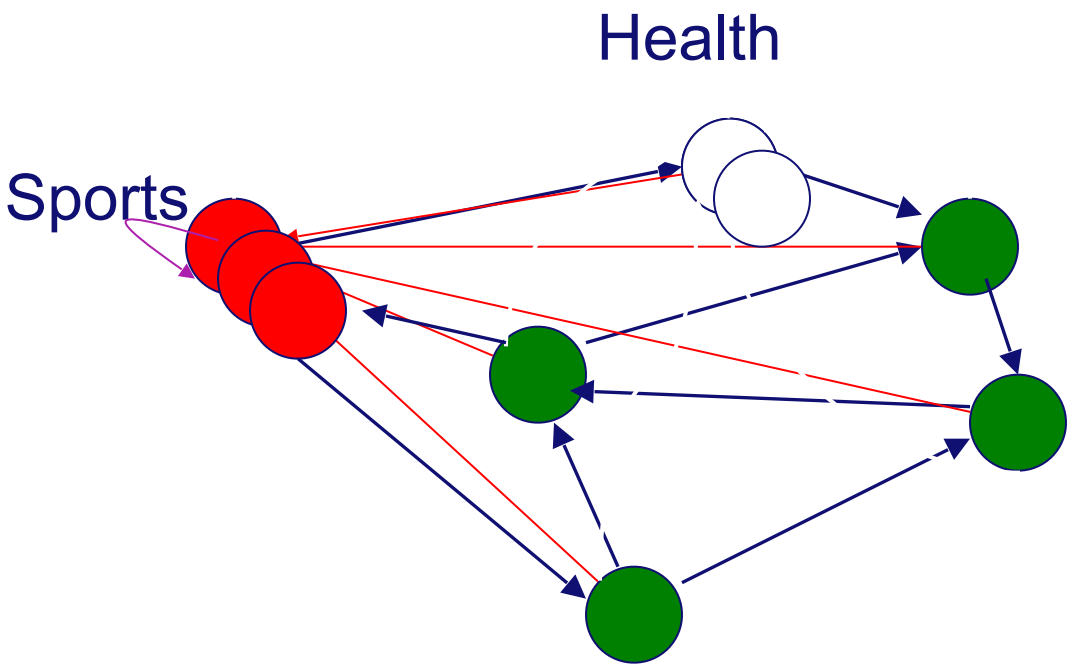
Teleport with 10% probability to a Sports page

Non-uniform Teleportation



10% Health teleportation

Personalized PageRank



pr = (0.9 PR_{sports} + 0.1 PR_{health}) gives you:
9% sports teleportation, 1% health teleportation

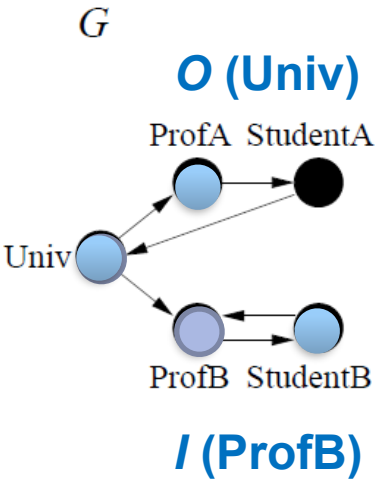
Homogeneous: SimRank (KDD'02)

- SimRank: two objects are similar if they are related to similar objects, i.e.
  - measures similarity of the structural context in which objects occur, based on their relationships with other objects
  - its score relates to the expected distance for two random surfers to first meet at the same node.
  - can be applied to any domain with object-to-object relationships.
  - takes into account both direct and indirect connections.
  - can be combined with traditional feature-based similarity, i.e. content descriptors, demographics, rating etc.

Basic Graph Model

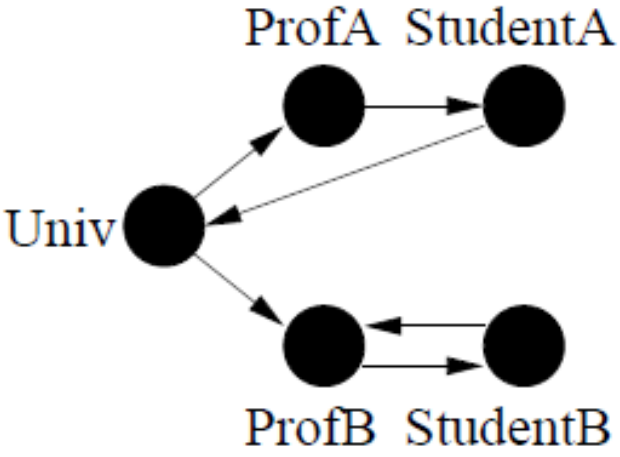
- $G = (V, E)$ [vertex, edge]
  - Nodes in V: objects in the domain
  - Directed edges in E: relationships between objects
  - <p, q> : from object p to object q

- For a node v, denote:
  - $I(v)$: the set of in-neighbors of v
  - $O(v)$: the set of out-neighbors of v
  - $I_i(v)$: individual in-neighbor ($1 \le i \le |I(v)|$)
  - $O_i(v)$: individual out-neighbor ($1 \le i \le |O(v)|$)



SimRank

- Motivation
  - Two objects are ***similar*** if they are referenced by ***similar*** object
  - Object always maximally similar to itself (similarity score of 1)



Similar nodes:
{ProfA, ProfB},
{StudentA, StudentB},
{Univ, ProfB},
…

Basic SimRank Equation

- The similarity between objects *a* and *b*: **s(*a*, *b*) ∈ [0, 1]**

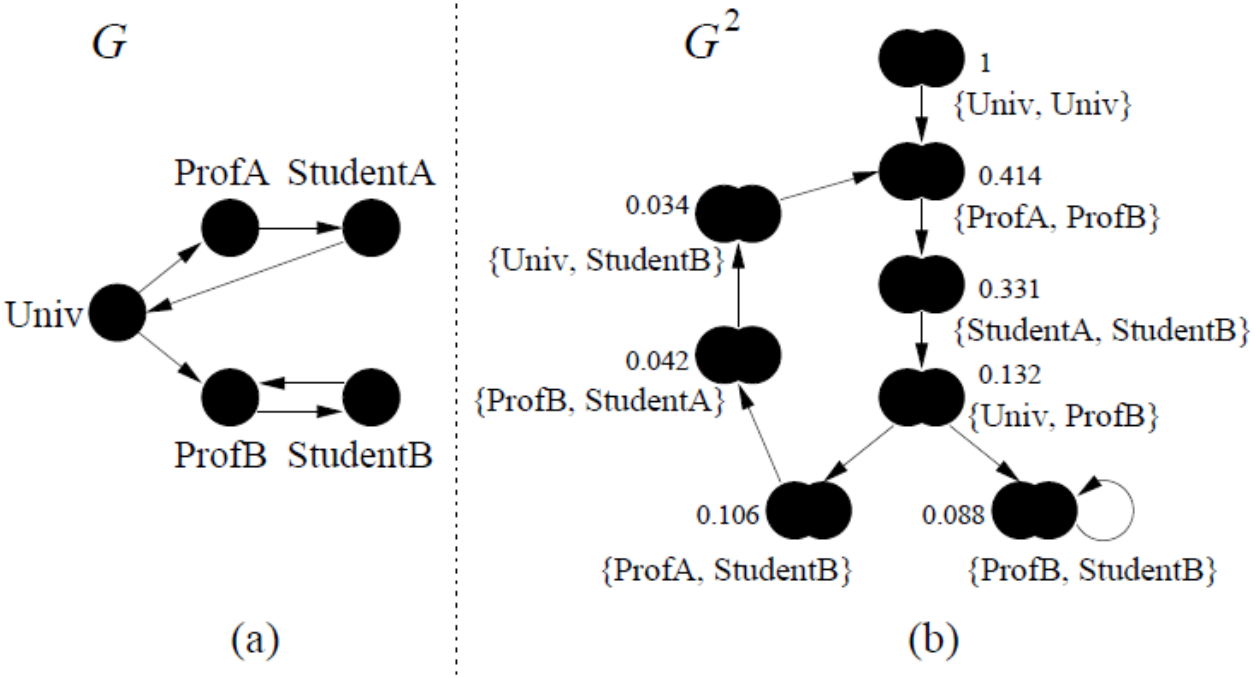
$$s(a,b) = \begin{cases} 1 & (\text{if } a = b) \\ \dfrac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} s(I_i(a), I_j(b)) & (\text{if } a \neq b) \end{cases}$$

  - *C:* confidence level or decay factor
    - constant between 0 and 1
    - gives the rate of decay as similarity flows across edges
  - If *a* or *b* do not have any in-neighbors, s(*a*,*b*) = 0
  - SimRank scores are symmetric, i.e., s(*a*,*b*) = s(*b*,*a*)

- Similarity between *a* and *b* is **the average similarity** between in-neighbors of *a* and in-neighbors of *b*

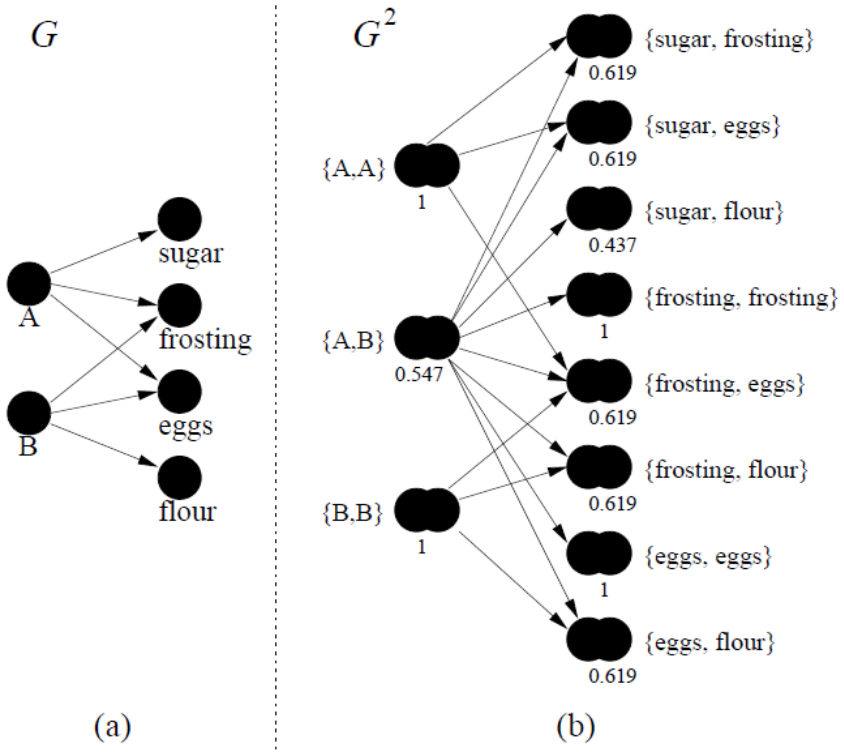
Basic SimRank Equation

- Similarity – as "propagating" from pair to pair
  - Consider the derived graph $G^2=(V^2, E^2)$ where
    - $V^2=V$ x V, represents a pair (a,b) of nodes in G
    - An edge from (a,b) to (c,d) exists in E^2, iff the edges $\langle a,c\rangle$ and $\langle b,d\rangle$ exist in E

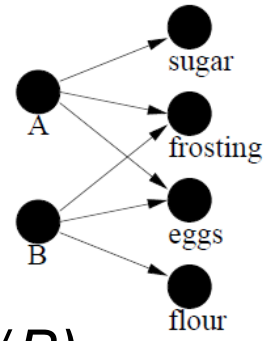


Bipartite SimRank

- Bipartite domains consist of two types of objects
- Recommender system
  - People are *similar* if they purchase *similar* items
  - Items are *similar* if they are purchased by *similar* people
  - Similarity of items and similarity of people are mutually reinforcing notions



Bipartite SimRank



- Bipartite Equation
  - Directed edges go from people to items
  - s(*A*,*B*) denote the similarity between <u>persons</u> *A* and *B*, *(A≠B)*

$$s(A,B) = \frac{C_1}{|O(A)||O(B)|} \sum_{i=1}^{|O(A)|} \sum_{j=1}^{|O(B)|} s(O_i(A), O_j(B))$$

  - s(*c*,*d*) denote the similarity between <u>items</u> *c* and *d*, *(c≠d)*

$$s(c,d) = \frac{C_2}{|I(c)||I(d)|} \sum_{i=1}^{|I(c)|} \sum_{j=1}^{|I(d)|} s(I_i(c), I_j(d))$$

  - The similarity between persons *A* and *B* is the average similarity between the items they purchased
  - The similarity between items *c* and *d* is the average similarity between the people who purchased them

Computing SimRank - Naïve Method

- $R_k(a,b)$ gives the score between a and b on iteration k

$$R_0(a,b) = \begin{cases} 0 \,(if\ a \neq b) \\ 1 (if\ a = b) \end{cases}$$

$$R_{k+1}(a,b) = \frac{C}{|I(a)||I(b)|} \sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} R_k(I_i(a), I_j(b))$$

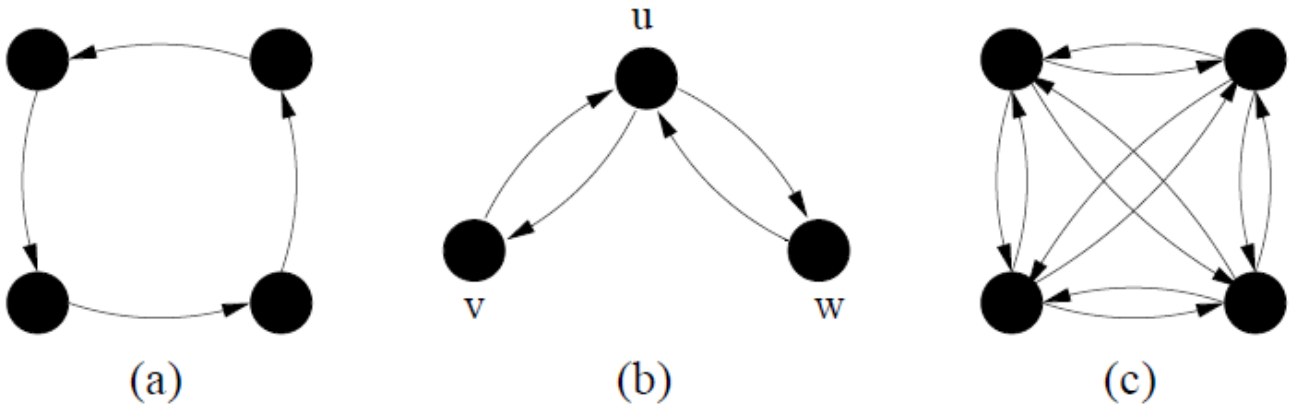
- The values $R_k(*,*)$ are non-decreasing as k increase

  $\lim_{k \to \infty} R_k(a,b) = s(a,b)$

- In experiments, when K = 5, R_k is rapidly converged
- Many faster/scalable/non-iterative ways proposed

SimRank as Random Surfer-Pairs Model

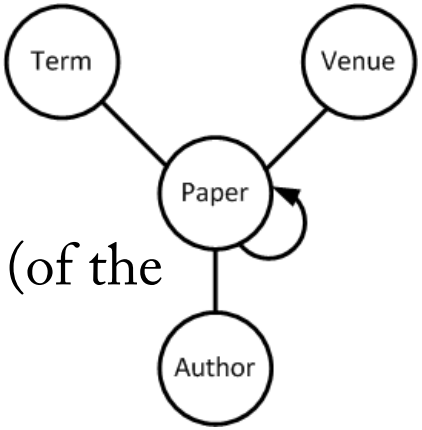
- Shows that SimRank score s(a,b) measures how soon 2 random surfers are expected to meet at the same node



- Expected Distance
  - u and v are nodes in strongly connected graph
  - ED(u, v) is exactly the expected number of steps a random surfer would take before he first reaches v, starting from u

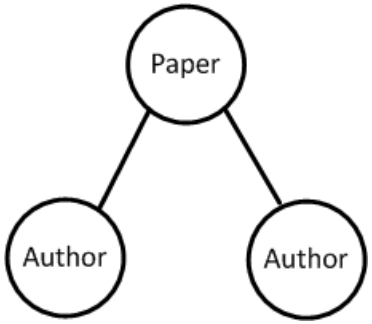
Heterogeneous: Meta-Path

- Network schema
  - Meta-level description of a network
- Meta-Path
  - **Meta-level description** of a path between two objects (of the same type)
  - **A path** on network schema
  - Denote an existing or concatenated **relation** between two object types



"Jim-P1-Ann"
"Mike-P2-Ann"
"Mike-P3-Bob"
...

Path Instances



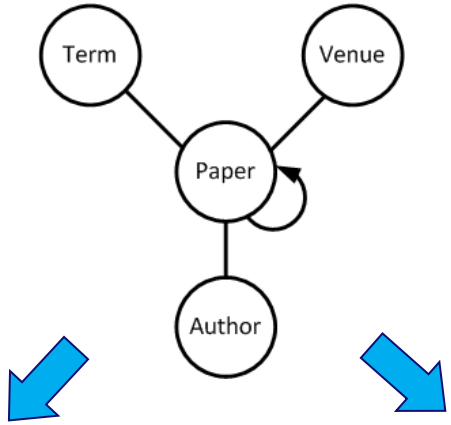
Meta-Path

Co-authorship

Relation: Describe the Type of Relationships

Different Meta-Paths: Different Semantics

- Who are most similar to Christos Faloutsos?



Meta-Path: Author-Paper-Author

Rank	Author	Score
1	Christos Faloutsos	1
2	Spiros Papadimitriou	0.127
3	Jimeng Sun	0.12
4	Jia-Yu Pan	0.114
5	Agma J. M. Traina	0.110
6	Jure Leskovec	0.096
7	Caetano Traina Jr.	0.096
8	Hanghang Tong	0.091
9	Deepayan Chakrabarti	0.083
10	Flip Korn	0.053

Christos's students or close collaborators

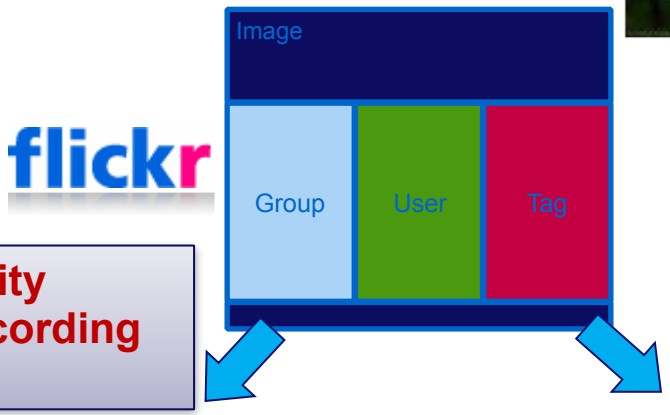
Meta-Path: Author-Paper-Venue-Paper-Author

Rank	Author	Score
1	Christos Faloutsos	1
2	Jiawei Han	0.842
3	Rakesh Agrawal	0.838
4	Jian Pei	0.8
5	Charu C. Aggarwal	0.739
6	H. V. Jagadish	0.705
7	Raghu Ramakrishnan	0.697
8	Nick Koudas	0.689
9	Surajit Chaudhuri	0.677
10	Divesh Srivastava	0.661

Work on similar topics and have similar reputation

Some Meta-Paths are "Better" Than Others

- Which pictures are most similar to

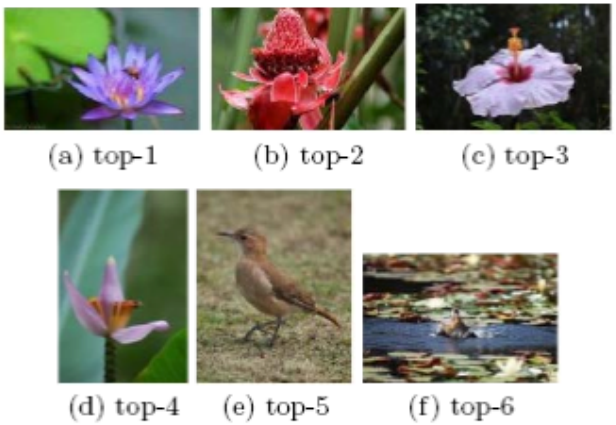


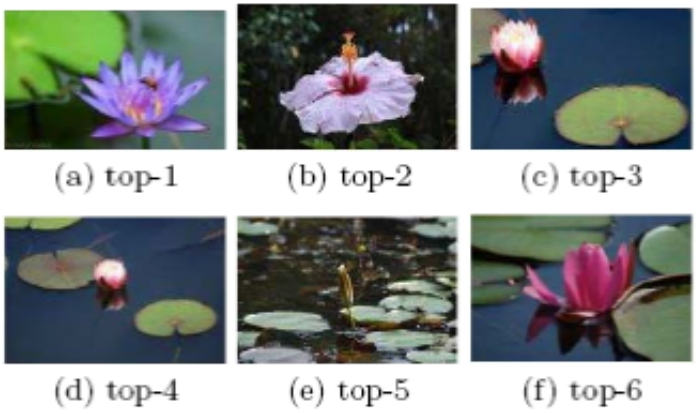
Evaluate the similarity between images according to their linked tags

Meta-Path: Image-Tag-Image

Evaluate the similarity between images according to tags and groups

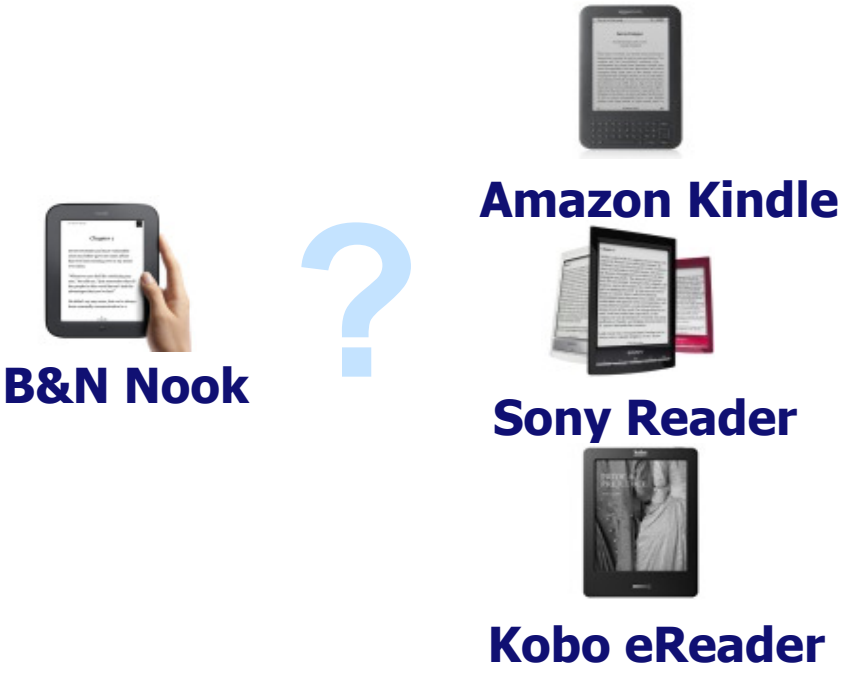
Meta-Path: Image-Tag-Image-Group-Image-Tag-Image





PathSim: Similarity in Terms of "Peers"

- Why peers?
  - Strongly connected, while **similar visibility**



- In addition to meta-path
  - Need to consider **similarity measures**

Personalized PageRank

We can find communities using *Personalized PageRank (PPR)*
[Andersen et al. 2006]

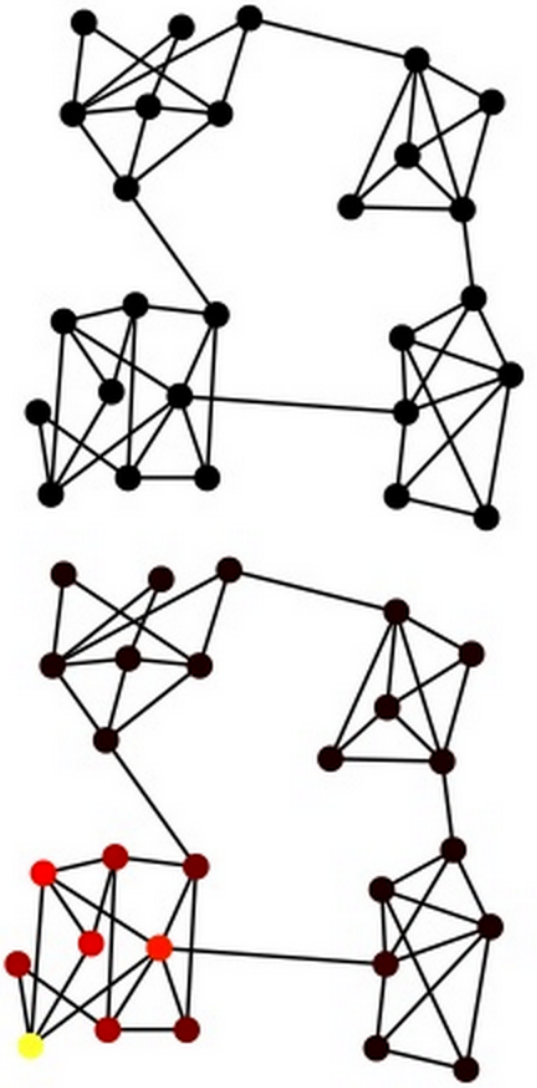
PPR is a Markov chain on nodes

1. with probability α, follow a random edge
2. with probability $1-\alpha$, restart at a seed

aka *random surfer*

aka *random walk with restart*

unique stationary distribution



Existing Similarity Measures

- Random walk: (Personalized) PageRank - used by Twitter
  - Favor **highly visible** objects (large degrees)
- Pairwise random walk: SimRank
  - Favor **"pure"** objects (highly skewed distribution in their in-links or out-links)
- PathSim
  - Favor **"peers"**, i.e. objects with strong connectivity and similar visibility under the given meta-path
- Note: PageRank and SimRank do not distinguish object type and relationship type

Comparison with Other Measures: A Toy Example

(a) Adjacency matrix W_{AC}.

	SIGMOD	VLDB	ICDE	KDD
Mike	2	1	0	0
Jim	50	20	0	0
Mary	2	0	1	0
Bob	2	1	0	0
Ann	0	0	1	1

(b) Similarity between Mike and other authors.

	Jim	Mary	Bob	Ann
P-PageRank	**0.3761**	0.0133	**0.0162**	0.0046
SimRank	**0.7156**	0.5724	**0.7125**	0.1844
RW	**0.8983**	0.0238	**0.0390**	0
PRW	**0.5714**	0.4444	**0.5556**	0
PathSim	0.0826	**0.8**	**1**	0

Ranking and clustering in SNA

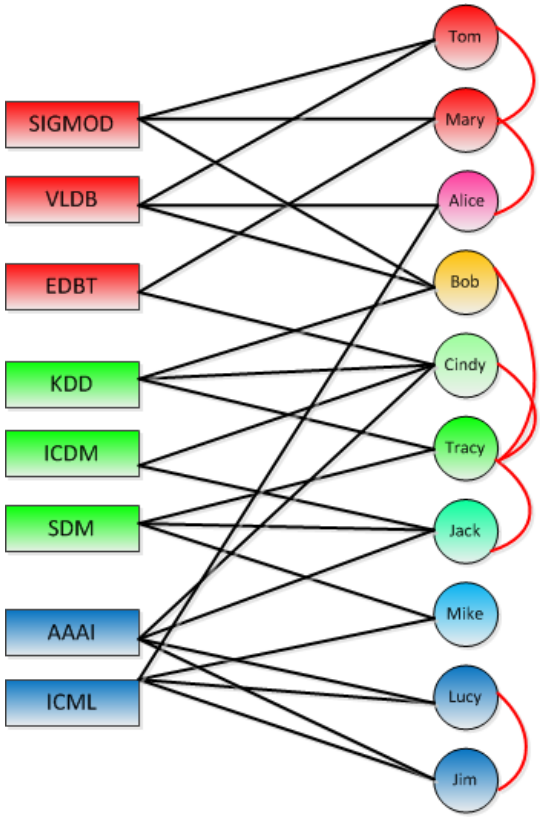
- Ranking evaluates objects based on some ranking function that mathematically demonstrates characteristics of objects
  - PageRank, HITS
- Clustering groups objects based on a certain proximity measure so that similar objects are in the same cluster, whereas dissimilar ones are in different clusters
- Ranking and clustering: often regarded as orthogonal techniques, but applying either of them on networks may lead to suboptimal results:
  - Ranking objects globally over whole network (e.g. politics, sports and fashion blogs all-together)
  - Clustering many objects together treating all equally important (hubs, authorities, nodes at periphery)

RankClus: Integrating Clustering with Ranking

- Ranking-based clustering
  - Ranking is conditional on a specific cluster
    - E.g., VLDB's rank in Theory vs. its rank in the DB area
  - The distributions of ranking scores over objects are different in each cluster
- Clustering and ranking are mutually enhanced
  - Better clustering: rank distributions for clusters are more distinguishing from each other
  - Better ranking: better metric for objects is learned from the ranking
- Not every object should be treated equally in clustering!

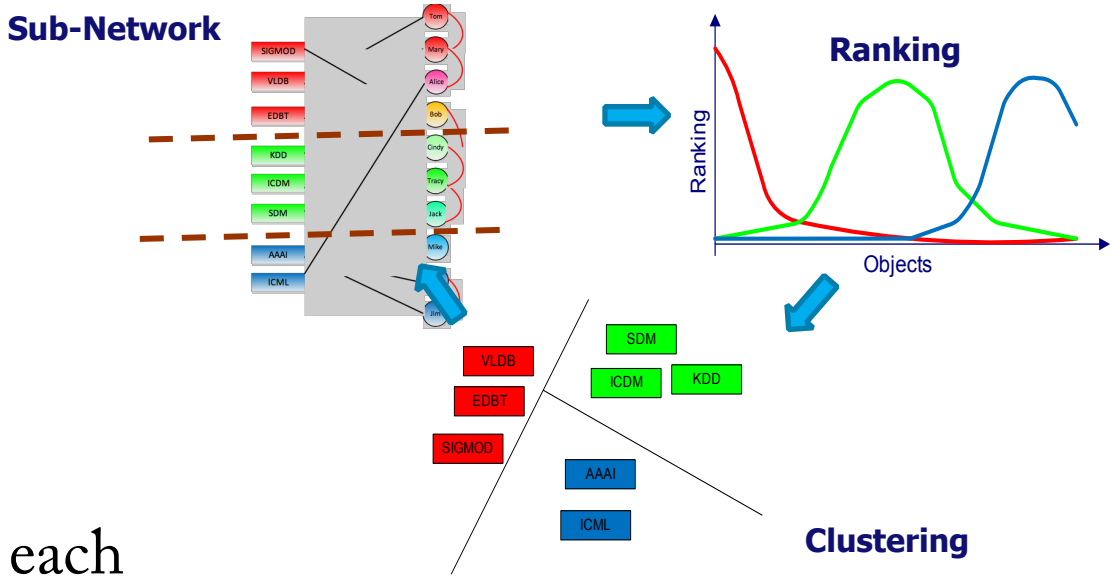
RankClus: Integrating Clustering with Ranking

- A case study on bi-typed DBLP network
- Links exist between
  - Conference (X) and author (Y)
  - Author (Y) and author (Y)
- A matrix denoting the weighted links
  - $W = \begin{bmatrix} W_{XX} & W_{XY} \\ W_{YX} & W_{YY} \end{bmatrix}$
- Goal:
  - Clustering and ranking conferences via authors



Naive solution: Project the bi-typed network into homogeneous conference network?
Information-loss projection!

RankClus: Algorithm Framework



- Initialization
  - Randomly partition
- Repeat
  - Ranking
    - Ranking objects in each sub-network induced from each cluster
  - Generating new measure space
    - Estimate model parameters (e.g. mixture model coefficients) for each target object
  - Adjusting cluster
- Until stable

Simple Ranking vs. Authority Ranking

- Simple Ranking
  - Proportional to # of publications of an author / a conference
  - Considers only **immediate neighborhood** in network

What about an author publishing 1000 papers in very weak conferences?

- Authority Ranking:
  - More sophisticated "rank rules" are needed
  - **Propagate** the ranking scores in the network over different types

Rules for Authority Ranking

Rule 1: Highly ranked authors publish *many* papers in highly ranked conferences

$$\vec{r}_Y(j) = \sum_{i=1}^{m} W_{YX}(j,i)\vec{r}_X(i)$$

Rule 2: Highly ranked conferences attract *many* papers from *many* highly ranked authors

$$\vec{r}_X(i) = \sum_{j=1}^{n} W_{XY}(i,j)\vec{r}_Y(j)$$

Rule 3: The rank of an author is enhanced if he or she co-authors with *many* highly ranked authors

$$\vec{r}_Y(i) = \alpha \sum_{j=1}^{m} W_{YX}(i,j)\vec{r}_X(j) + (1-\alpha)\sum_{j=1}^{n} W_{YY}(i,j)\vec{r}_Y(j)$$

What Can be Mined from Heterogeneous Networks?

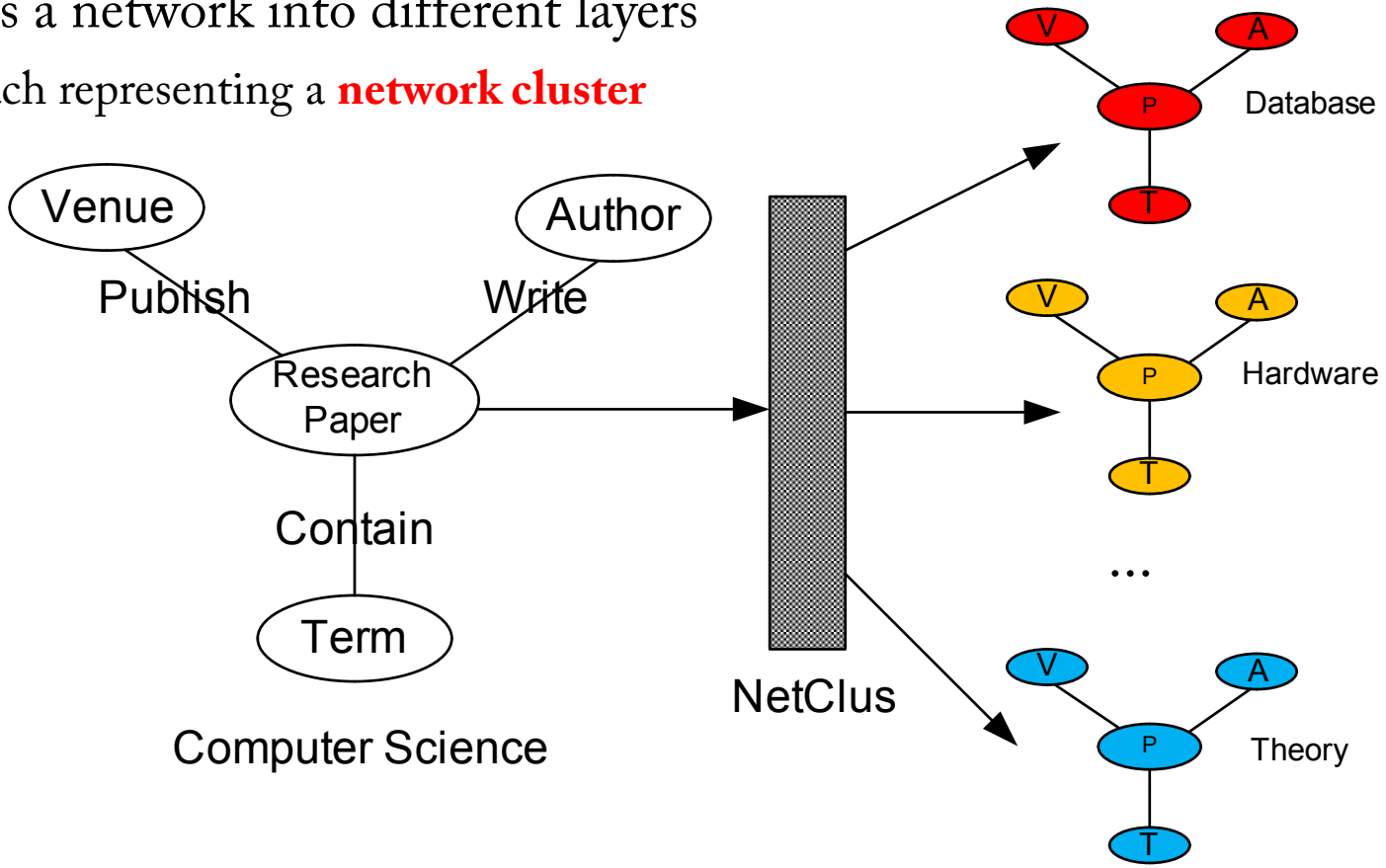
- DBLP: A Computer Science bibliographic database

A sample publication record in DBLP (>1.8 M papers, >0.7 M authors, >10 K venues), …

Knowledge hidden in DBLP Network	Mining Functions
How are CS research areas structured?	*Clustering*
Who are the leading researchers on Web search?	*Ranking*
What are the most essential terms, venues, authors in AI?	*Classification + Ranking*
Who are the peer researchers of Jure Leskovec?	*Similarity Search*
Whom will Christos Faloutsos collaborate with?	*Relationship Prediction*
Which types of relationships are most influential for an author to decide her topics?	*Relation Strength Learning*
How was the field of Data Mining emerged or evolving?	*Network Evolution*
Which authors are rather different from his/her peers in IR?	*Outlier/anomaly detection*

NetClus: Beyond Bi-Typed Networks

- Beyond bi-typed information network
  - A Star Network Schema [**richer information**]
- Take initial clustering, build model to predict clusters, build new clusters based on that model's predictions, repeat
- Splits a network into different layers
  - Each representing a **network cluster**



Summary on similarity in networks

- SimRank: x and y are similar if they are related to similar objects; basic and bipartite cases.
  - Jeh&Widom. SimRank: a measure of structural-context similarity. In KDD'02 http://doi.acm.org/10.1145/775047.775126
- PathSim: accounts for connectivity of x and y as number of paths between them following meta-path (specified by a user) and for the balance of their visibility; meaning/interpretation is reflected in a chosen meta-path.
  - Sun et al. PathSim: Meta PathBased TopK Similarity Search in Heterogeneous Information Networks, VLDB'11, http://www.cs.uiuc.edu/~hanj/pdf/vldb11_ysun.pdf
- RankClus: integrating clustering with ranking for bi-typed networks, ranking as a feature of clustering
  - Sun et al. RankClus: integrating clustering with ranking for heterogeneous information network analysis. In EDBT'09, http://doi.acm.org/10.1145/1516360.1516426
- NetClus: beyond bi-typed networks, star network schema
  - Sun et al. Ranking-based clustering of heterogeneous information networks with star network schema. In KDD'09