Introduction to Data Science and Analytics (DSC510)

Handling Networks

George Pallis

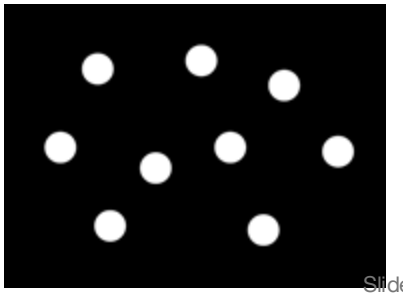
Beyond flat tables

People

id	name	age
1	Bob	36
2	Willy	32
…	…	…

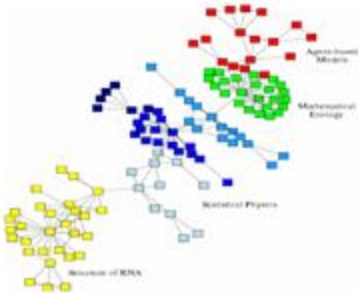
Marriages

husband_id	wife_id
1	34
2	5
2	87
…	…

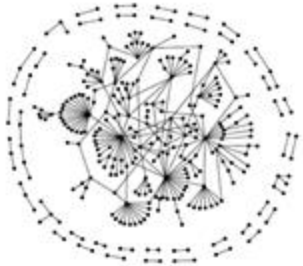


Examples

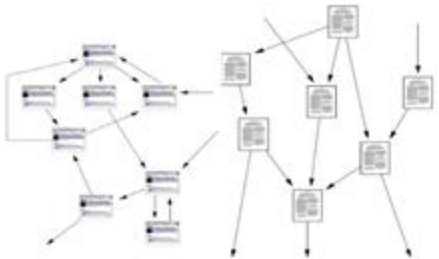
Social networks



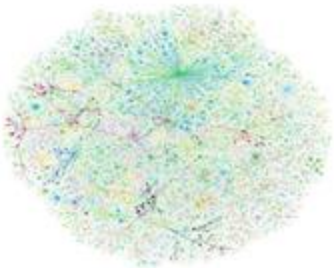
Economic networks



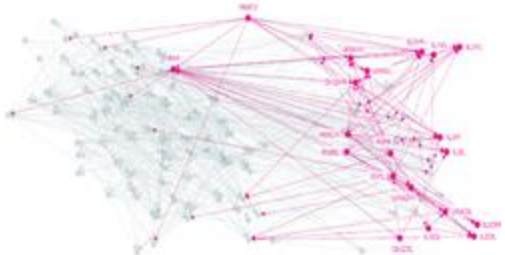
Communication graphs



Information networks:
Web & citations



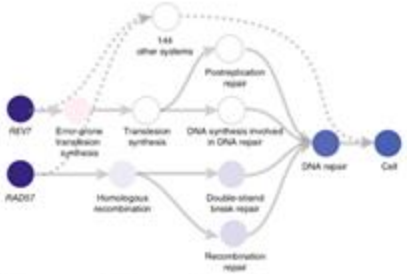
Internet



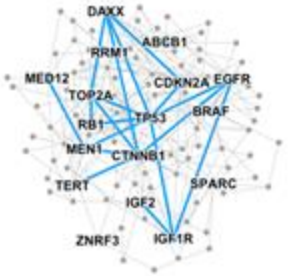
Networks of neurons

Examples

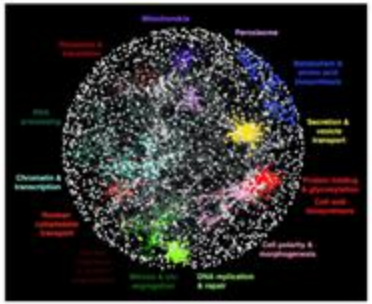
Patient networks



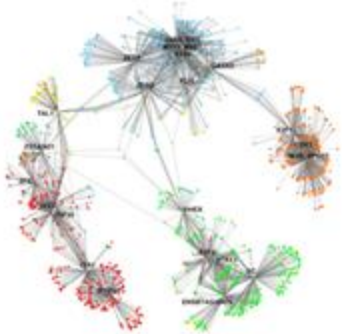
Hierarchies of cell systems



Disease pathways



Genetic interaction networks



Gene co-expression networks

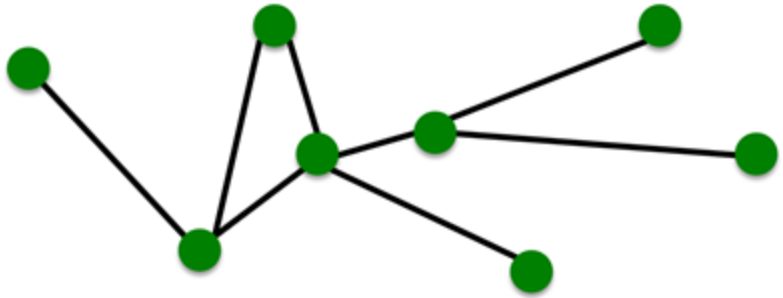
Cell-cell similarity networks

Networks as graphs

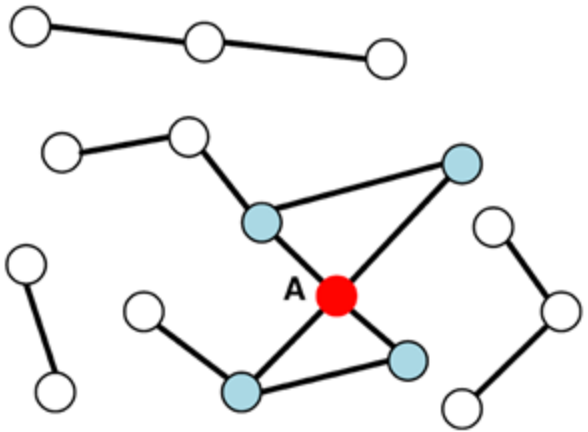
- **Network:** a real-world system of dependent variables, e.g.,
  - WWW is a network of hyperlinked documents
  - Society is a network of individuals linked by family, friendship, professional ties
  - Metabolic network is sum of all chemical reactions in a cell
- **Graph:** mathematical abstraction for describing networks
- In practice, “network” and “graph” are used interchangeably
- You can make a graph out of almost anything (e.g., connect all people whose name starts with the same letter), so must ask: Does this graph correspond to a meaningful network?

Most basic type: undirected graphs

- Entities:
  nodes/vertices *V*
- Relationships/interactions:
  edges/links *E*
- Entire system:
  network/graph *G* = (*V*, *E*)



Node degree



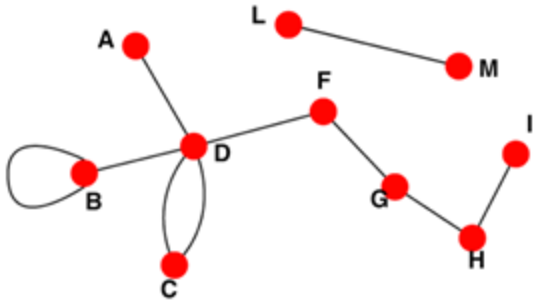
Node degree, k_i: the number of edges adjacent to node ***i***

$k_A = 4$

Types of graphs: undirected vs. directed

Undirected

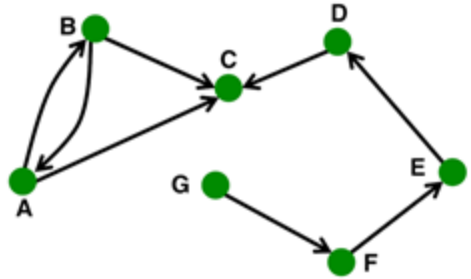
- **Links:** undirected (symmetrical, reciprocal)



- **Examples:**
  - Collaborations
  - Friendship on Facebook

Directed

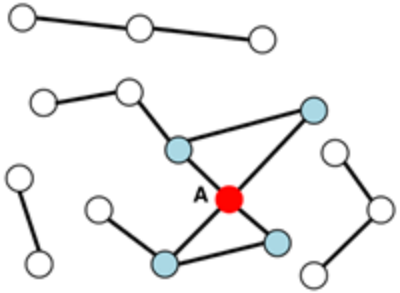
- **Links:** directed (arcs)



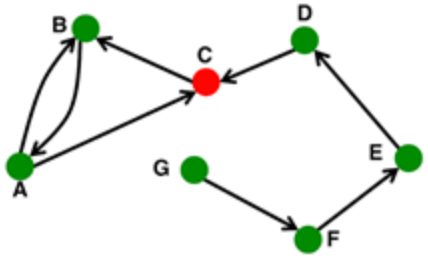
- **Examples:**
  - Phone calls
  - Following on Twitter

Average node degree

Undirected



Directed



Source: Node with $k^{in} = 0$
Sink: Node with $k^{out} = 0$

Node degree, k_i: the number of edges adjacent to node i

$k_A = 4$

Avg. degree: $\bar{k} = \langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i =$

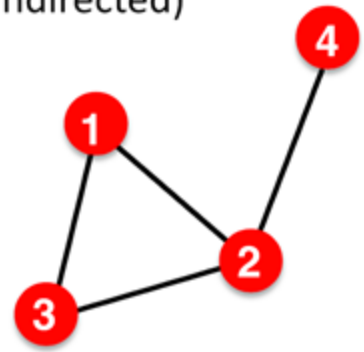
In directed networks we define an **in-degree** and **out-degree.**
The (total) degree of a node is the sum of in- and out-degrees.

$k_C^{in} = 2 \qquad k_C^{out} = 1 \qquad k_C = 3$

$\bar{k} =$

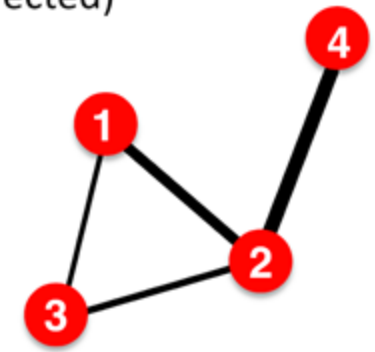
Types of graphs: weighted

- **Unweighted**
  (undirected)



Examples: Friendship, Hyperlink

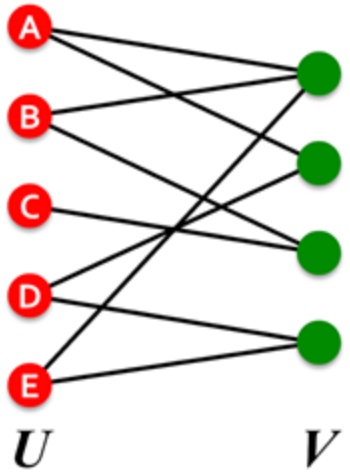
- **Weighted**
  (undirected)



Examples: Collaboration, Internet, Roads

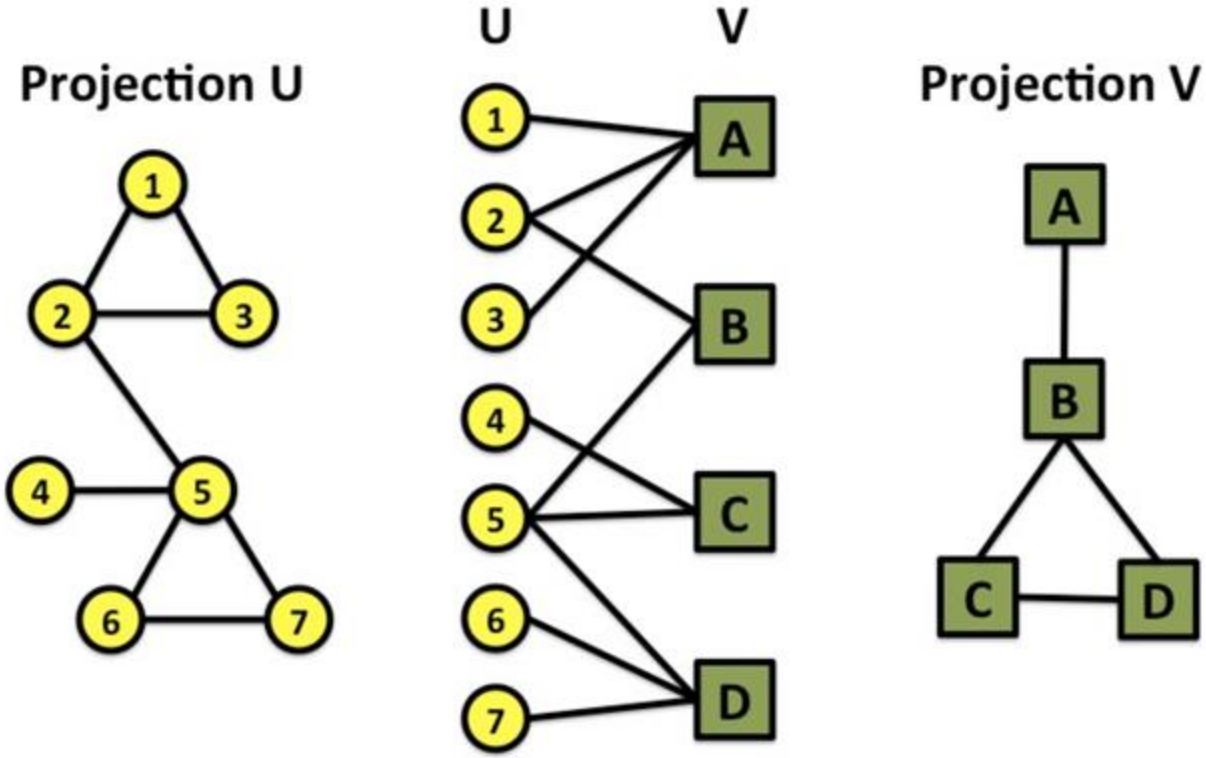
Types of graphs: bipartite

- **Bipartite graph** is a graph whose nodes can be divided into two disjoint sets U and V such that every link connects a node in U to one in V; that is, U and V are **independent sets**

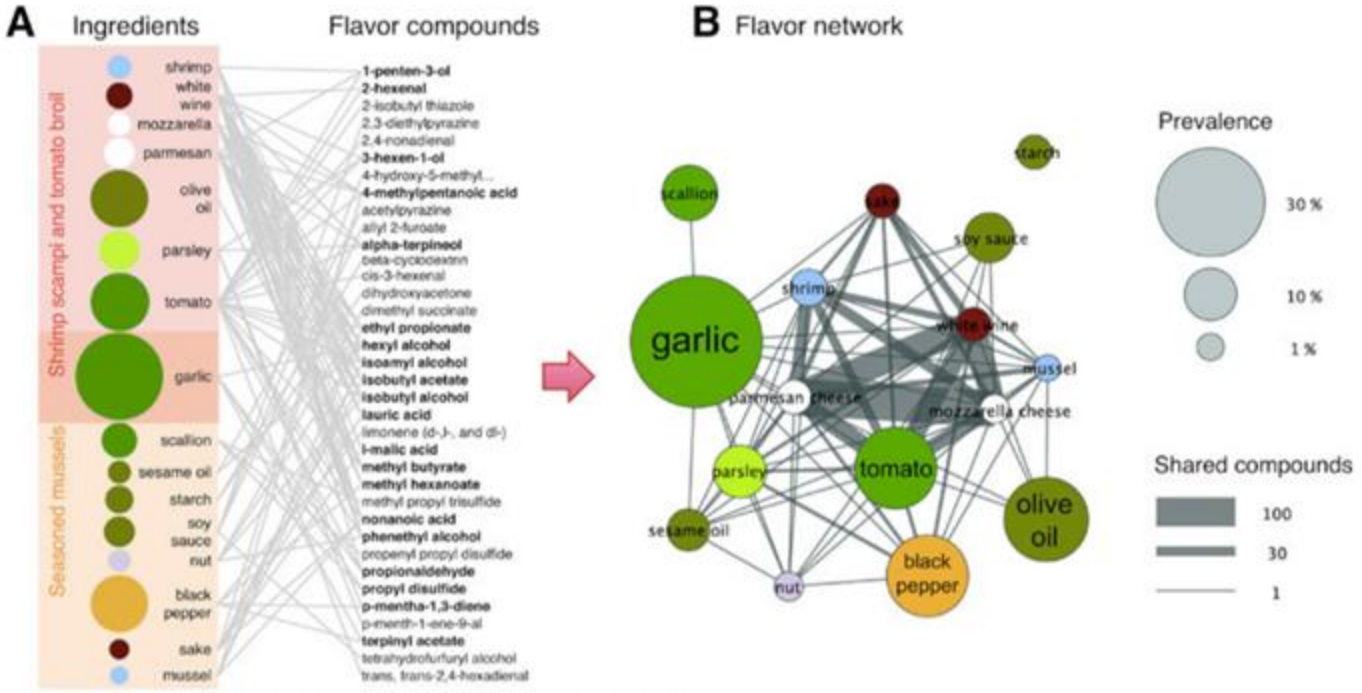


- **Examples:**
  - Authors-to-Papers (they authored)
  - Actors-to-Movies (they appeared in)
  - Users-to-Movies (they rated)
  - Recipes-to-Ingredients (they contain)
- **"Folded" networks:**
  - Author collaboration networks
  - Movie co-rating networks

Projections of bipartite graph



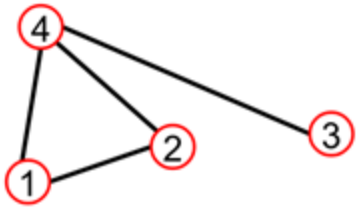
Example: flavor networks

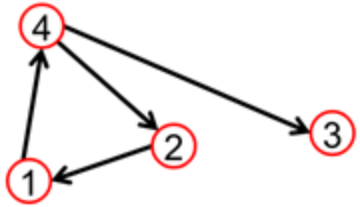


Y.-Y. Ahn, S. E. Ahnert, J. P. Bagrow, A.-L. Barabási
Flavor network and the principles of food pairing , Scientific Reports 196, (2011).

[paper]

Representing graphs on computers: adjacency matrix





$A_{ij} = 1$ if there is a link from node *i* to node *j*

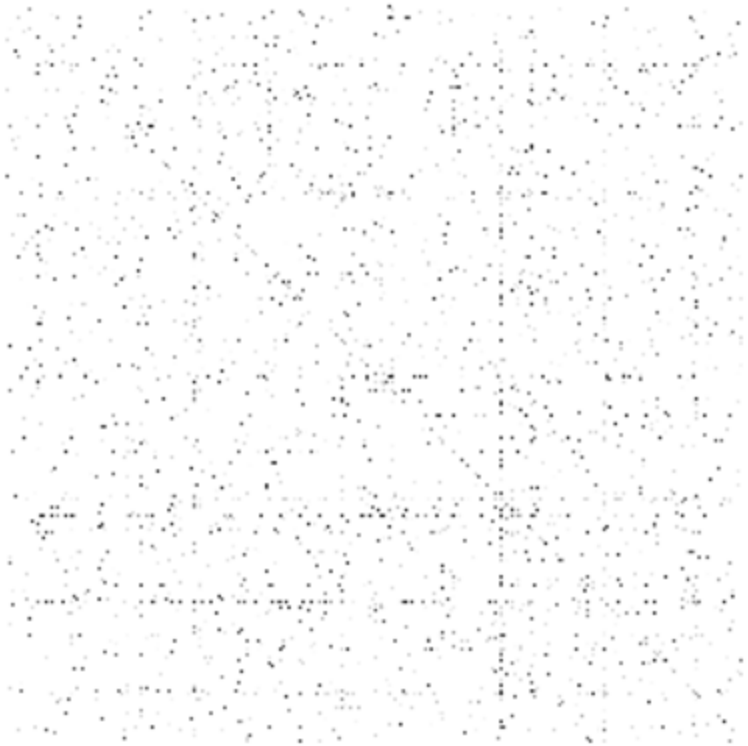
$A_{ij} = 0$ otherwise

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Note that for a directed graph (right) the matrix is not symmetric.

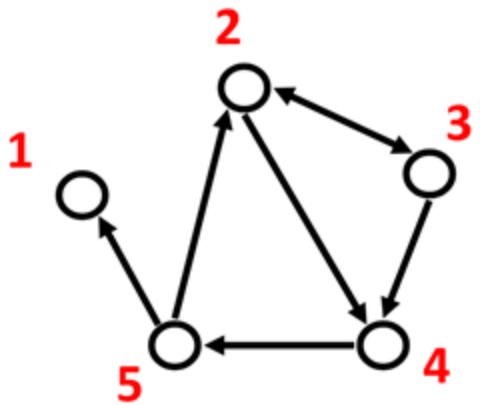
Slides by Dr. Robert West, Applied Data Analysis

Adjacency matrix usually sparse



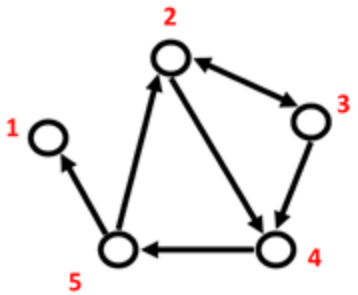
Representing graphs on computers: edge list

- **Represent graph as a set of edges:**
  - (2, 3)
  - (2, 4)
  - (3, 2)
  - (3, 4)
  - (4, 5)
  - (5, 2)
  - (5, 1)



Representing graphs on computers: adjacency list

- **Adjacency list:**
  - Easier to work with if network is
    - Large
    - Sparse
  - Allows us to quickly retrieve all neighbors of a given node
    - 1:
    - 2: 3, 4
    - 3: 2, 4
    - 4: 5
    - 5: 1, 2

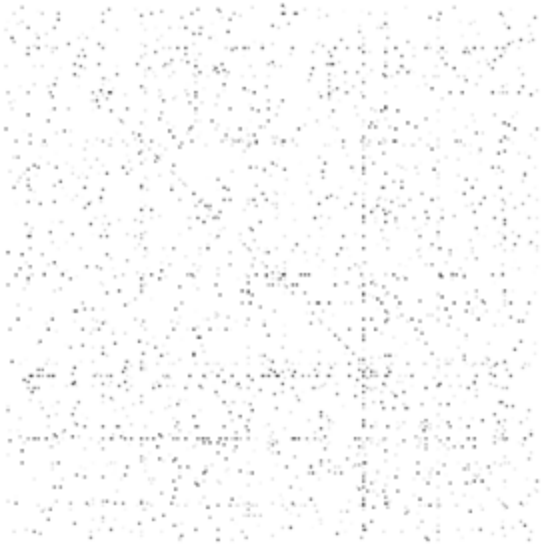


Properties of real-world networks

- Real networks are different from arbitrary graphs
- Real networks tend to share certain properties
- Remarkable, given the diversity of networks
  - Information networks (e.g., Web graph, knowledge graphs)
  - Social networks (e.g., Facebook, sexual networks)
  - Biological networks
  - …

Properties of real-world networks: sparsity

- Every node connected to only small fraction of all other nodes
- i.e., $k_i \ll N$
- Often bounded by a constant
  - e.g., social networks: Dunbar's number (cognitive limit to the number of people with whom one can maintain stable social relationships; allegedly 150)



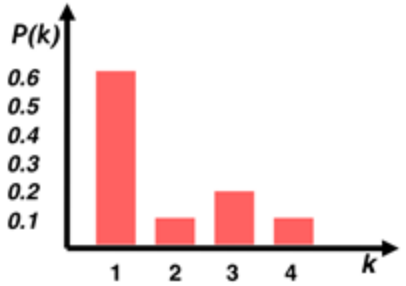
Properties of real-world networks: degree distribution

- **Degree distribution $P(k)$:** Probability that a randomly chosen node has degree k

  N_k = # nodes with degree k

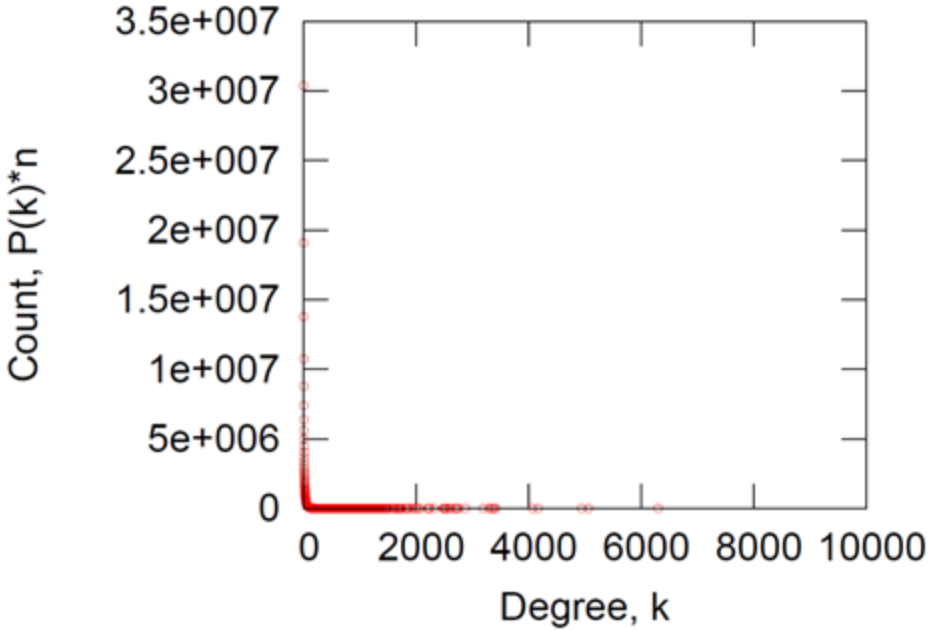
- Normalized histogram:

  $P(k) = N_k / N$ → **plot**

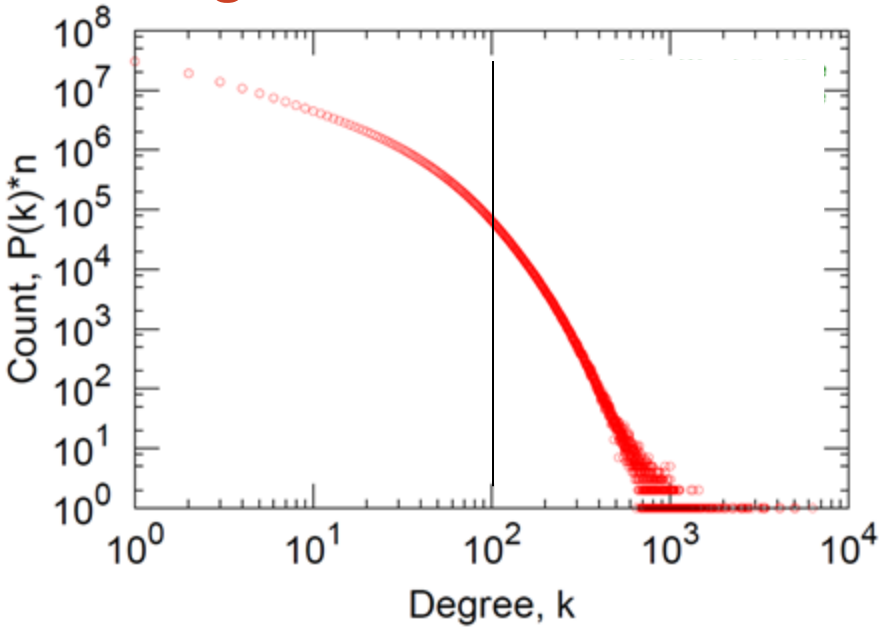


Properties of real-world networks: degree distribution

Linear axes

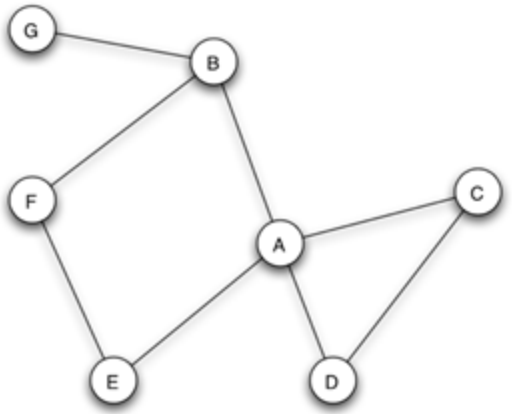


Logarithmic axes

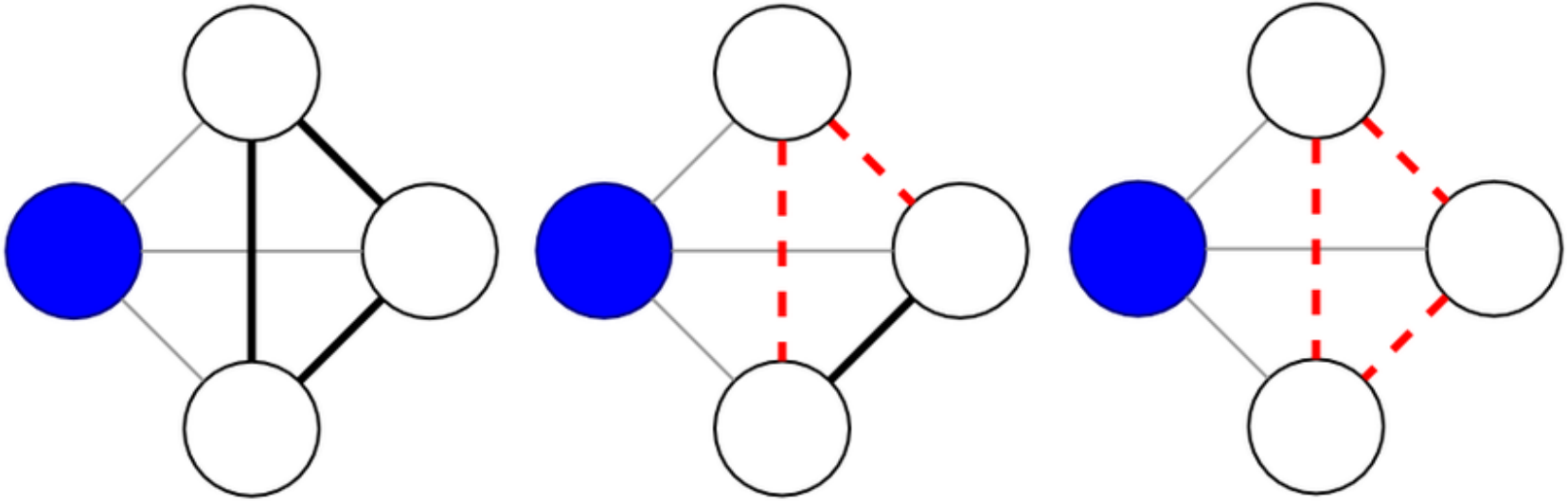


Properties of real-world networks: triadic closure

- "A friend of my friend is my friend"
- Measured via clustering coefficient C_i of node i:
  C_i = (#edges among neighbors of i) / (#potential edges among neighbors of i)
- #potential edges among neighbors of i in undirected graph: $k_i (k_i - 1) / 2$
- $C_A = 1 / (4*3/2) = 1/6$
- In real networks, nodes have large clustering coefficients



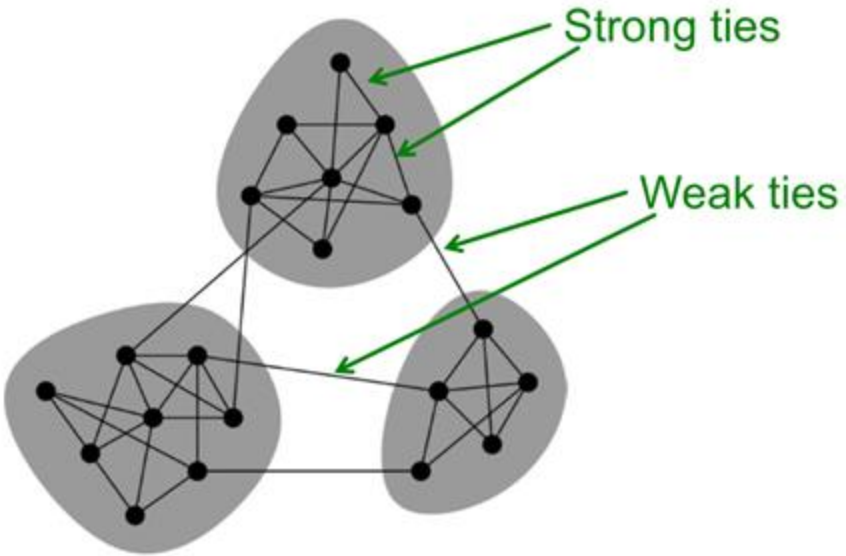
Clustering Coefficient



$C(i) = 1$ $C(i) = 1/3$ $C(i) = 0$

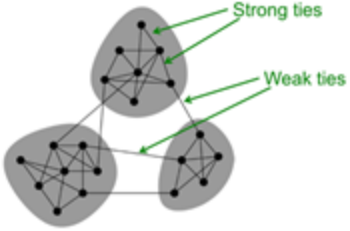
Properties of real-world networks: community structure

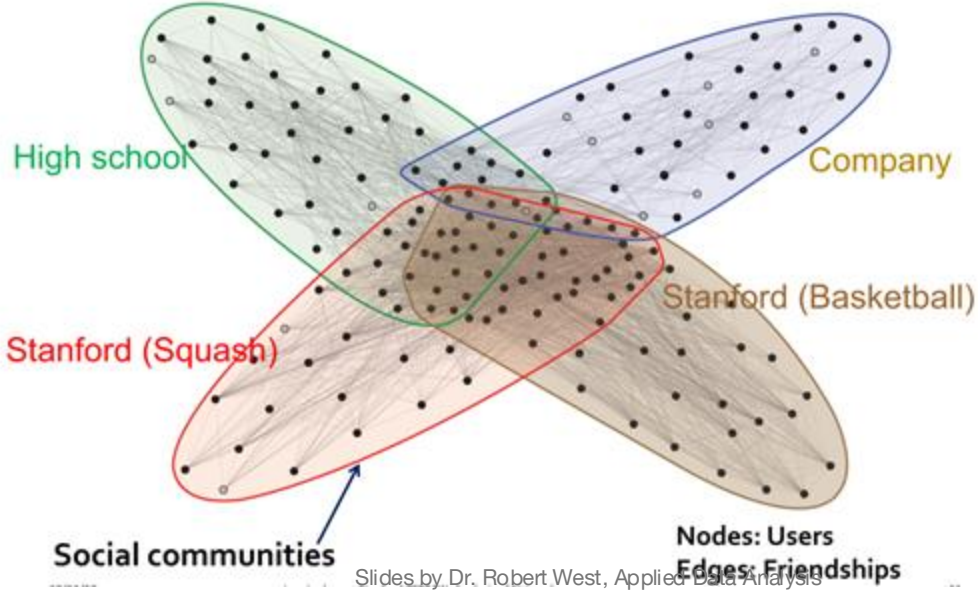
- Triadic closure makes real networks cluster into locally dense "communities"
- Communities connected via "weak ties"
- "Strength of weak ties"
- "Structural holes"



Properties of real-world networks: community structure

- In real life, communities are often not disjoint, But overlapping:

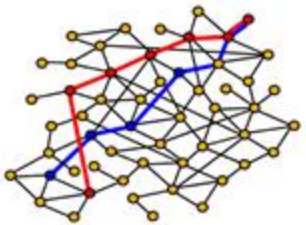




Properties of real-world networks: average shortest-path length

- **What is the typical shortest path length between any two people?**
  - **Experiment on the global friendship network**
    - Can't measure, need to probe explicitly
- **Small-world experiment** [Milgram '67]
  - Picked 300 people in Omaha, Nebraska and Wichita, Kansas
  - Ask them to get a letter to a stock-broker in Boston by passing it through friends
- **How many steps did it take?**

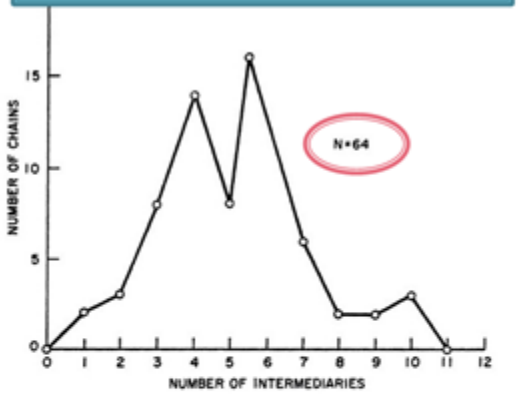
[Movie]



Properties of real-world networks: average shortest-path length

- **64 chains completed:**
  (i.e., 64 letters reached the target)
  - It took 6.2 steps on the average, thus **"6 degrees of separation"**
- **Further observations:**
  - People who owned stock had shorter paths to the stockbroker than random people: 5.4 vs. 6.7
  - People from the Boston area have even closer paths: 4.4

Milgram's small world experiment



2008 2011

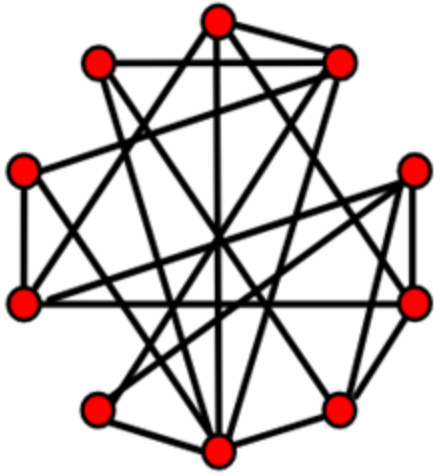
TECH • FACEBOOK

It's Actually 3.5 Degrees of Separation, Says Facebook

BY **JONATHAN CHEW**
February 5, 2016 3:56 PM GMT+2

Properties of real-world networks: navigability

- For decades, people focused on the fact that short paths exist in social networks
- But this is true even in random graphs (e.g., Erdős-Rényi model)
- The truly amazing fact is not that short paths exist, but that they are discoverable via greedy decentralized routing (as in Milgram's experiment)
- Intrigued? Read Jon Kleinberg: "The Small-World Problem: An Algorithmic Perspective"

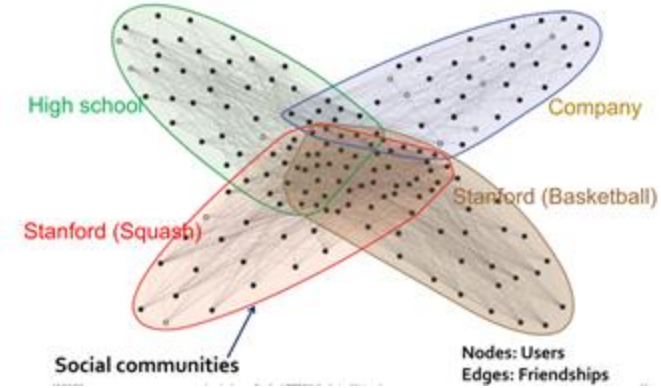


Don’t believe me?

- Play a game and see for yourself how well you can navigate an a-priori unknown network:
  - Wikispeedia.net

Properties of real-world networks: homophily/heterophily

- "Birds of a feather flock together"
- Especially in social networks
- Big confound and cause of debate: Influence vs. homophily
- E.g., obese people's friends are more likely to also be obese
  - Influence: I copy eating behavior of those around me [video]
  - Homophily: people with similar eating behavior prone to become friends



Properties of real-world networks: summary

Real-world networks (across many types)

- are sparsely connected,
- but some nodes are much more connected than most others (i.e., skewed degree distribution);
- form locally dense clusters via triadic closure,
- which leads to community structure;
- have short paths between random node pairs (partly due to "hubs" [skewed degree distribution!]),
- and the short paths are easily discoverable.

How to measure “importance” of a node?

- Formalized via *centrality measures*
- Map each node *i* to a scalar value *C*(*i*) capturing its importance in the overall network

Degree centrality

- Simplest centrality measure
- Many neighbors → important node
  - $C(i)$ = number of neighbors of i
- Very brittle, easy to “rig”
  - E.g., Twitter: scam account, followed by 100,000 other scam accounts

Closeness centrality

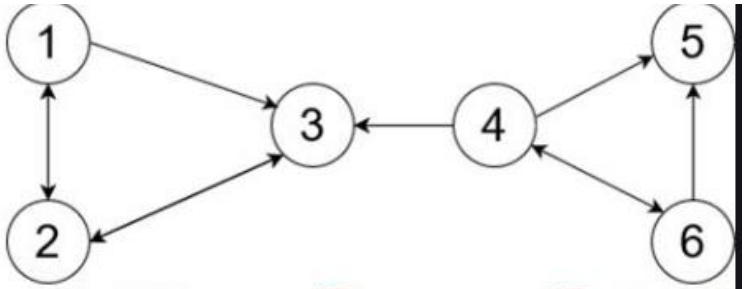
- Farness(*x*) = average (or cumulative) distance to *x* from other nodes
- *C*(*x*) = 1 / Farness(*x*)

$$C(x) = \frac{1}{\sum_y d(y,x)}$$

  - Reciprocal to turn farness into closeness
  - Only defined for connected graphs (otherwise *d* infinite)
- Under closeness centrality, nodes that are easy to reach from anywhere in the network are considered important
- Variant: harmonic centrality: switch sum and reciprocal
  - *C*(*x*) = average reciprocal distance of *x* to other nodes
  - Defined even for disconnected graphs (define 1/*d* of disconnected nodes as 0)

$$C(x) = \sum_{y \neq x} \frac{1}{d(y,x)}$$

CLOSENESS CENTRALITY



- Undirected graph:

$$C_C(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)}$$

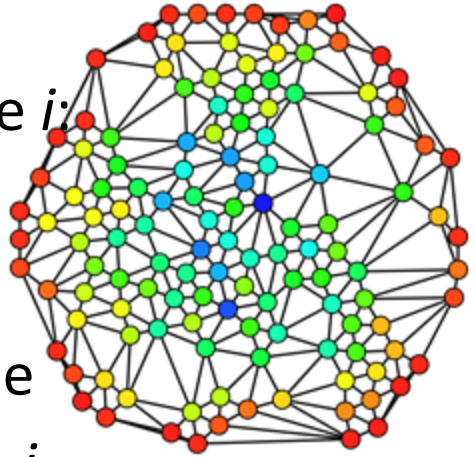
$C_c(1) = \frac{5}{1+1+2+3+3} = 5/10 = 0.5$

$C_c(4) = \frac{5}{2+2+1+1+1} = 5/7$

- Directed graph: same equation but distance with directed edges only

Betweenness centrality

- $C(i)$ = fraction of all shortest paths in the network that pass through node i
- Computation of betweenness centrality of node i:
  - For each pair of vertices (s, t), compute all shortest paths between them.
  - For each pair of vertices (s, t), determine the fraction of shortest paths that pass through i.
  - Average this fraction over all pairs of vertices (s, t).
- Expensive to compute



Katz centrality

- Generalization of degree centrality
- Degree centrality counts only number of direct neighbors (i.e., neighbors at distance 1)
- Katz centrality also counts neighbors at distances 2, 3, …
- More precisely, number of paths from other nodes to *i*, giving less weight to larger distances:

$$C(i) = \sum_{k=1}^{\infty} \sum_{j=1}^{N} \alpha^k (A^k)_{ji}$$

k-th power of adjacency matrix *A* contains number of length-*k* paths for each node pair

- More robust than degree centrality

PageRank centrality

- Recursive definition: my centrality $C(i)$ =: x_i is high if I receive inlinks from many other central nodes:

$$x_i = \sum_j a_{ji} \frac{x_j}{L(j)}$$

$$L(j) = \sum_i a_{ji}$$

a_{ji}: entry (j, i) of adjacency matrix A (1 if j links to i, else 0)
$L(j)$: out-degree of j

- Some extra tweaks to make it work with any graph (e.g., disconnected)

PageRank centrality

$$x_i = \sum_j a_{ji} \frac{x_j}{L(j)}$$

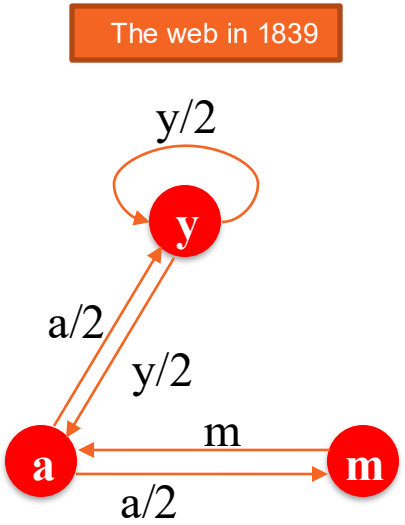
- Matrix notation: $x = M x$

  (where *M* is computed from adjacency matrix *A*)
- Do you recognize this?
  - *x* is eigenvector of *M* with eigenvalue 1 → we're in linear-algebra land
  - *x* is the steady state the Markov chain induced by the network: x_i is fraction of time a random walker will spend in node *i*

PageRank: The "Flow" Model

- **A "vote" from an important page is worth more**
- **A page is important if it is pointed to by other important pages**
- **Define a "rank" r_j for node j**

$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d}_{\mathrm{out}}(\mathrm{i})}$$



Flow equations:

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

Solving the Flow Equations

Flow equations:

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

- **3 equations, 3 unknowns, no constants**
  - No unique solution
- **Additional constraint forces uniqueness**
  - $r_y + r_a + r_m = 1$
  - $r_y = 2/5,\ r_a = 2/5,\ r_m = 1/5$
- Gaussian elimination method works for small examples, but we need a better method for large web-size graphs

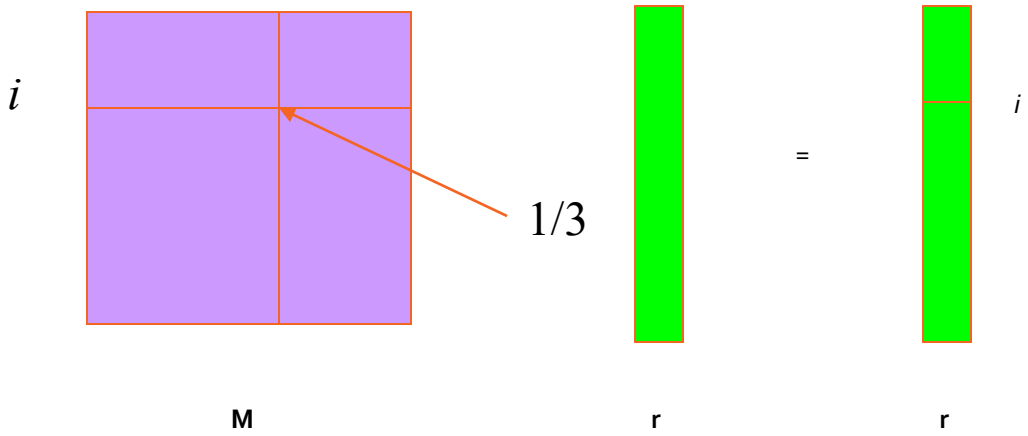
PageRank: Matrix Formulation

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

$$r = M\,r$$

Example

- **Suppose page j links to 3 pages, including i**



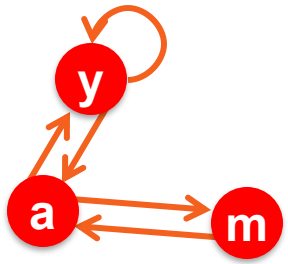
Eigenvector Formulation

- **The flow equations can be written**

$$r = M \cdot r$$

- So the rank vector is an eigenvector of the stochastic web matrix
  - In fact, its first or principal eigenvector, with corresponding eigenvalue 1

Example: Flow Equations & M



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

r = Mr

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

$$\begin{matrix} y \\ a \\ m \end{matrix} = \begin{bmatrix} ½ & ½ & 0 \\ ½ & 0 & 1 \\ 0 & ½ & 0 \end{bmatrix} \begin{matrix} y \\ a \\ m \end{matrix}$$

Power Iteration Method

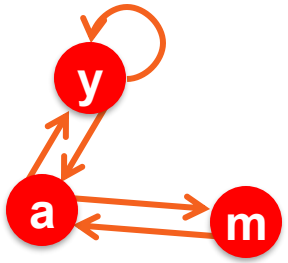
- **Given a web graph with n nodes, where the nodes are pages and edges are hyperlinks**
- **Power iteration:** a simple iterative scheme
  - Suppose there are *N* web pages
  - Initialize: $\mathbf{r}^{(0)} = [1/N,....,1/N]^T$
  - Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M} \cdot \mathbf{r}^{(t)}$
  - Stop when $|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}|_1 < \varepsilon$
    - $|\mathbf{x}|_1 = \sum_{1 \le i \le N} |x_i|$ is the L_1 norm
    - Can use any other vector norm e.g., Euclidean

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

d_i .... out-degree of node i

PageRank: How to solve?

- **Power Iteration:**
  - Set $r_j = 1/N$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate
    - $r_i = \sum_j M_{ij} \cdot r_j$
- **Example:**



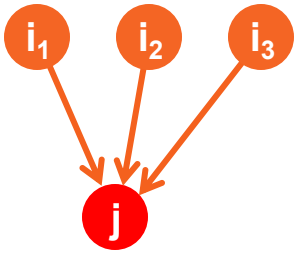
	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

r_y		1/3	1/3	5/12	9/24		6/15
r_a	=	1/3	3/6	1/3	11/24	…	6/15
r_m		1/3	1/6	3/12	1/6		3/15

Iteration 0, 1, 2, …

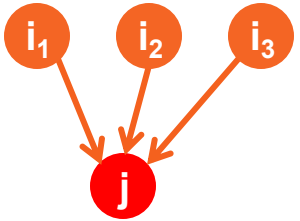
Random Walk Interpretation



$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d}_{\mathrm{out}}(\mathrm{i})}$$

- **Imagine a random web surfer:**
  - At any time t, surfer is on some page u
  - At time $t+1$, the surfer follows an out-link from u uniformly at random
  - Ends up on some page v linked from u
  - Process repeats indefinitely
- **Let:**
  - $\boldsymbol{p}(t)$ … vector whose i^{th} coordinate is the prob. that the surfer is at page i at time t
  - $\boldsymbol{p}(t)$ is a probability distribution over pages

The Stationary Distribution



$$p(t+1) = \mathbf{M} \cdot p(t)$$

- **Where is the surfer at time *t+1*?**
  - Follows a link uniformly at random

$$\boldsymbol{p}(t+1) = \boldsymbol{M} \cdot \boldsymbol{p}(t)$$

- Suppose the random walk reaches a state $\boldsymbol{p}(t+1) = \boldsymbol{M} \cdot \boldsymbol{p}(t) = \boldsymbol{p}(t)$
  - then $\boldsymbol{p}(t)$ is stationary distribution of a random walk
  - **Our rank vector $\boldsymbol{r}$** satisfies $\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$
    - So, it is a stationary distribution for the random walk

PageRank: Three Questions

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

or equivalently

$$r = Mr$$

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

Does This Converge?

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

- **Example:**

=

r_a	1	0	1	0
r_b	0	1	0	1

Iteration 0, 1, 2, ...

Does it Converge to What We Want?

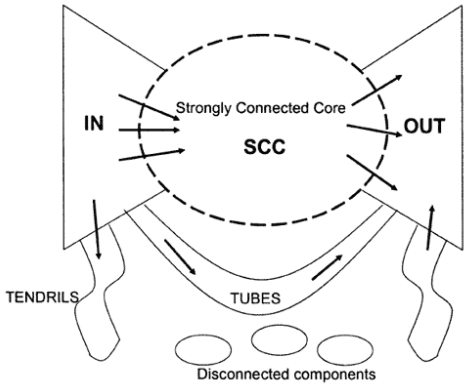
$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

- **Example:**

r_a =	1	0	0	0
r_b	0	1	0	0

Iteration 0, 1, 2, …

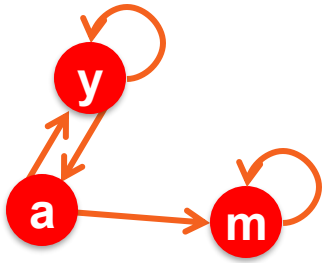
Problems with the "Flow" Model



2 problems:

- Some pages are "**dead ends**" (have no out-links)
  - Such pages cause importance to "leak out"
- **Spider traps** (all out links are within the group)
  - Eventually spider traps absorb all importance

Problem: Spider Traps



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2$$
$$r_m = r_a/2 + r_m$$

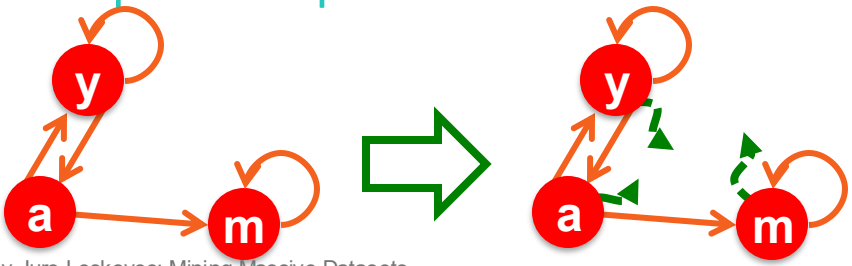
- **Example:**

r_y		1/3	2/6	3/12	5/24	0
r_a	=	1/3	1/6	2/12	3/24 …	0
r_m		1/3	3/6	7/12	16/24	1

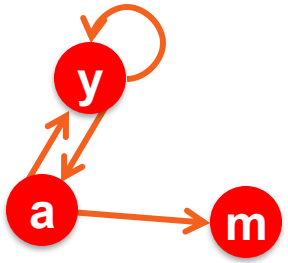
Iteration 0, 1, 2, …

Solution: Random Teleports

- **The Google solution for spider traps:** At each time step, the random surfer has two options:
  - With probability β, follow a link at random
  - With probability 1-β, jump to some page uniformly at random
  - Common values for β are in the range 0.8 to 0.9
- Surfer will teleport out of spider trap within a few time steps



Problem: Dead Ends



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

- **Power Iteration:**
  - Set $r_j = 1$
  - $r_j = \sum_{i \to j} \frac{r_i}{d_i}$
    - And iterate

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2$$
$$r_m = r_a/2$$

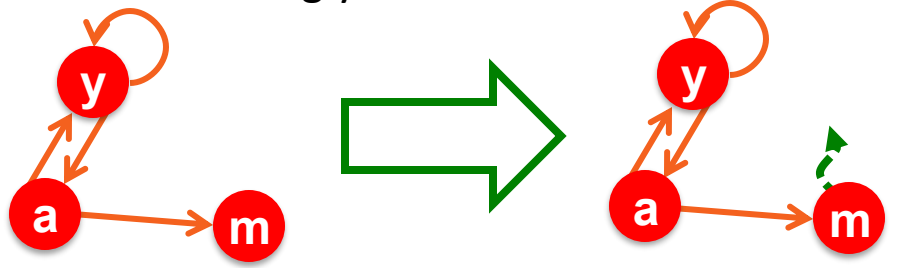
- **Example:**

r_y		1/3	2/6	3/12	5/24	0
r_a	=	1/3	1/6	2/12	3/24	… 0
r_m		1/3	1/6	1/12	2/24	0

Iteration 0, 1, 2, …

Solution: Dead Ends

- **Teleports:** Follow random teleport links with probability 1.0 from dead-ends
  - Adjust matrix accordingly



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

Why Teleports Solve the Problem?

$$r^{(t+1)} = Mr^{(t)}$$

Markov Chains

- Set of states X
- Transition matrix P where $P_{ij} = P(X_t=i \mid X_{t-1}=j)$
- π specifying the probability of being at each state $x \in X$
- Goal is to find π such that $\pi = P\,\pi$

Why is This Analogy Useful?

- **Theory of Markov chains**

- Fact: For any start vector, the power method applied to a Markov transition matrix P will converge to a unique positive stationary vector as long as P is **stochastic**, **irreducible** and **aperiodic**.

Solution: Random Jumps

- **Google's solution that does it all:**
  - Makes ***M*** stochastic, aperiodic, irreducible
- At each step, random surfer has two options:
  - With probability 1-β, follow a link at random
  - With probability β, jump to some random page
- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1 - \beta)\frac{1}{n}$$

From now on: We assume M has no dead ends
That is, we follow random teleport links
with probability 1.0 from dead-ends

d_i … out-degree of node i

The Google Matrix

- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \to j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{n}$$

- **The Google Matrix *A*:**

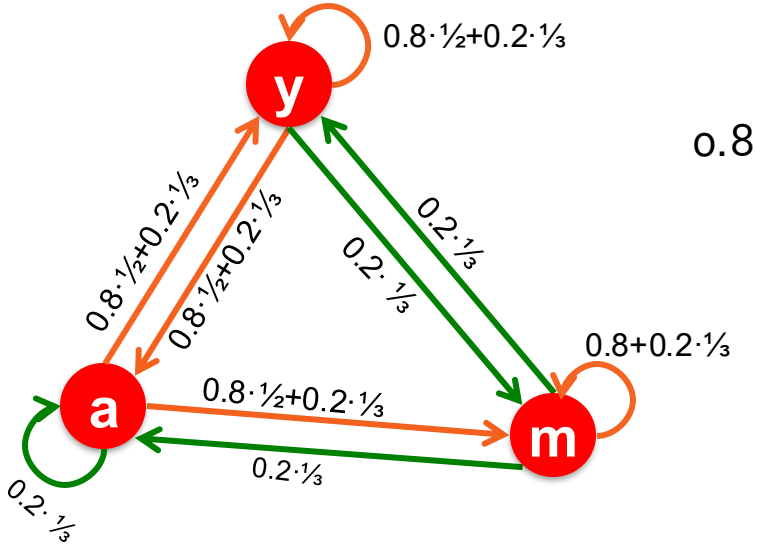
$$A = \beta\, S + (1-\beta)\frac{1}{n}\mathbf{1} \cdot \mathbf{1}^T$$

- ***G* is stochastic, aperiodic and irreducible, so**

$$r^{(t+1)} = A \cdot r^{(t)}$$

- **What is β?**
  - In practice $\beta = 0.85$ (make 5 steps and jump)

Random Teleports ($\beta = 0.8$)



S **1/n·1·1ᵀ**

$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$\begin{matrix} y \\ a \\ m \end{matrix} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

A

$$\begin{matrix} y \\ a \\ m \end{matrix} = \begin{matrix} 1/3 & 0.33 & 0.24 & 0.26 & \\ 1/3 & 0.20 & 0.20 & 0.18 & \ldots \\ 1/3 & 0.46 & 0.52 & 0.56 & \end{matrix} \begin{matrix} \\ \\ \\ \end{matrix} \begin{matrix} 7/33 \\ 5/33 \\ 21/33 \end{matrix}$$

PageRank centrality

- The technology that made Google huge
  - "Page" in PageRank for Larry Page
  - MapReduce (next lecture!) was invented to compute PageRank on ful
  - "The $25,000,000,000 eigenvector" [link]
- Bottom line:
  Pay attention in your linear algebra class

Some Problems with Page Rank

Measures generic popularity of a page

Biased against topic-specific authorities

Solution: Topic-Specific PageRank (next)

Susceptible to Link spam

Artificial link topographies created in order to boost page rank

Solution: TrustRank (next)

Uses a single measure of importance

Other models e.g., hubs-and-authorities

Solution: Hubs-and-Authorities (next)

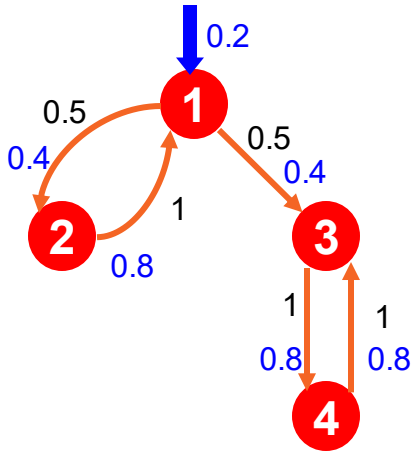
Topic-Specific PageRank

- **Instead of generic popularity, can we measure popularity within a topic?**
- **Goal:** Evaluate Web pages not just according to their popularity, but by how close they are to a particular topic, e.g. “sports” or “history.”
- **Allows search queries to be answered based on interests of the user**
  - **Example:** Query “Trojan” wants different pages depending on whether you are interested in sports or history.

Topic-Specific PageRank

- Assume each walker has a small probability of "teleporting" at any step
- **Teleport can go to:**
  - Any page with equal probability
    - To avoid dead-end and spider-trap problems
  - A topic-specific set of "relevant" pages (teleport set)
    - For topic-sensitive PageRank.
- **Idea: Bias the random walk**
  - When walked teleports, she pick a page from a set *S*
  - *S* contains only pages that are relevant to the topic
    - E.g., Open Directory (DMOZ) pages for a given topic
  - For each teleport set *S*, we get a different vector r_S

Example



Suppose $S = \{1\}$, $\beta = 0.8$

Node	Iteration				
	0	**1**	**2**	**…**	**stable**
1	1.0	0.2	0.52		0.294
2	0	0.4	0.08		0.118
3	0	0.4	0.08		0.327
4	0	0	0.32		0.261

Note how we initialize the PageRank vector differently from the unbiased PageRank case.

Discovering the Topic

- **Create different PageRanks for different topics**
  - The 16 DMOZ top-level categories:
    - arts, business, sports,…
- **Which topic ranking to use?**
  - User can pick from a menu
  - Classify query into a topic
  - Can use the context of the query
    - E.g., query is launched from a web page talking about a known topic
    - History of queries e.g., “basketball” followed by “Jordan”
  - User context, e.g., user’s bookmarks, …

What is Web Spam?

- **Spamming:**
  - any deliberate action to boost a web page's position in search engine results, incommensurate with page's real value
- **Spam:**
  - web pages that are the result of spamming
- This is a very broad definition
  - SEO industry might disagree!
  - SEO = search engine optimization
- Approximately 10-15% of web pages are spam

Web Search

- **Early search engines:**
  - Crawl the Web
  - Index pages by the words they contained
  - Respond to search queries (lists of words) with the pages containing those words
- **Early page ranking:**
  - Attempt to order pages matching a search query by “importance”
  - First search engines considered:
    - 1) Number of times query words appeared.
    - 2) Prominence of word position, e.g. title, header.

First Spammers

- As people began to use search engines to find things on the Web, those with commercial interests tried to exploit search engines to bring people to their own site – whether they wanted to be there or not.
- **Example:**
  - Shirt-seller might pretend to be about “movies.”
- **Techniques for achieving high relevance/importance for a web page**

First Spammers: Term Spam

- **How do you make your page appear to be about movies?**
  - **1)** Add the word movie 1000 times to your page
  - Set text color to the background color, so only search engines would see it
  - **2)** Or, run the query “movie” on your target search engine
  - See what page came first in the listings
  - Copy it into your page, make it “invisible”
- **These and similar techniques are term spam**

Google's Solution to Term Spam

- **Believe what people say about you, rather than what you say about yourself**
  - Use words in the anchor text (words that appear underlined to represent the link) and its surrounding text

- PageRank as a tool to measure the "importance" of Web pages

Why It Works?

- Our hypothetical shirt-seller **loses**
  - Saying he is about movies doesn't help, because others don't say he is about movies
  - His page isn't very important, so it won't be ranked high for shirts or movies
- **Example:**
  - Shirt-seller creates 1000 pages, each links to his with "movie" in the anchor text
  - These pages have no links in, so they get little PageRank
  - So the shirt-seller can't beat truly important movie pages like IMDB

Google vs. Spammers: Round 2

- Once Google became the dominant search engine, spammers began to work out ways to fool Google
- **Spam farms** were developed to concentrate PageRank on a single page
- **Link spam:**
  - Creating link structures that boost PageRank of a particular page

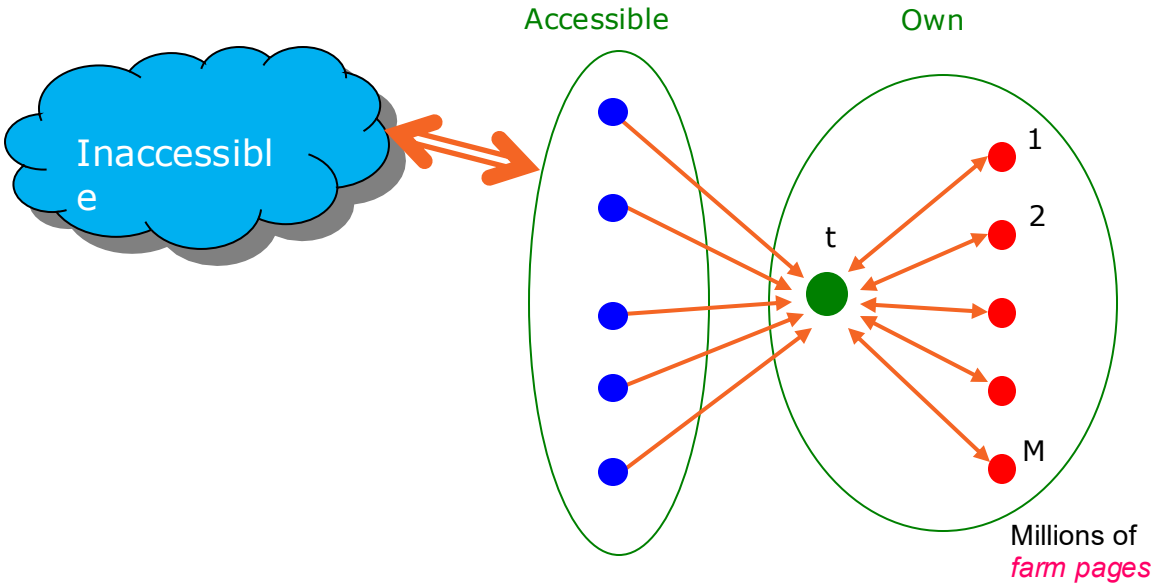
Link Spamming

- **Three kinds of web pages from a spammer's point of view:**
  - **Inaccessible pages**
  - **Accessible pages:**
    - e.g., blog comments pages
    - spammer can post links to his pages

Link Farms

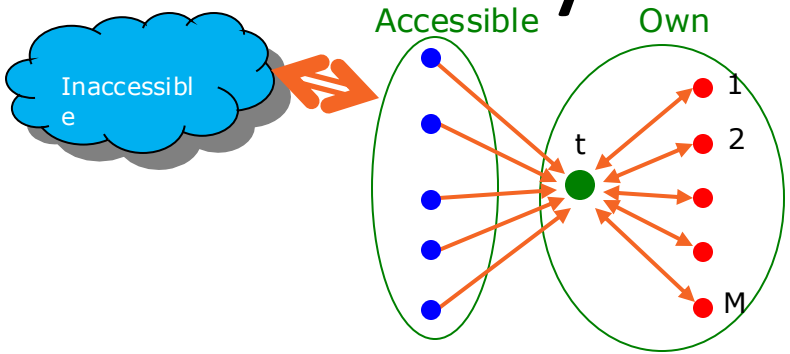
- **Spammer's goal:**
  - Maximize the PageRank of target page *t*
- **Technique:**
  - Get as many links from accessible pages as possible to target page *t*

Link Farms



One of the most common and effective organizations for a link farm

Analysis



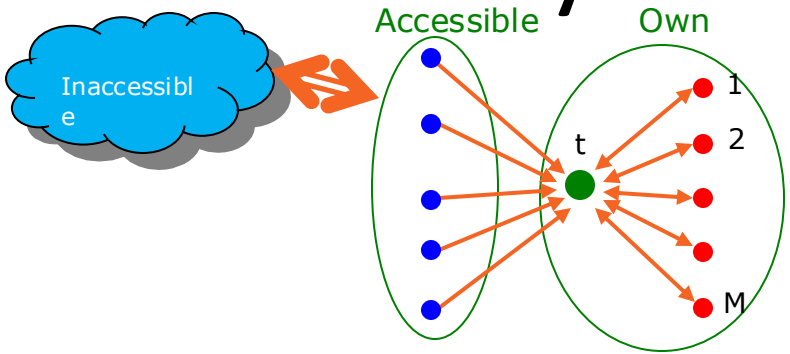
N…# pages on the web
M…# of pages spammer owns

- x: PageRank contributed by accessible pages
- y: PageRank of target page t
- Rank of each "farm" page $= \frac{\beta y}{M} + \frac{1-\beta}{N}$
- $y = x + \beta M\left[\frac{\beta y}{M} + \frac{1-\beta}{N}\right] + \frac{1-\beta}{N}$

  $= x + \beta^2 y + \frac{\beta(1-\beta)M}{N} + \frac{1-\beta}{N}$

  Very small; ignore
  Now we solve for y
- $y = \frac{x}{1-\beta^2} + c\frac{M}{N}$ where $c = \frac{\beta}{1+\beta}$

Analysis

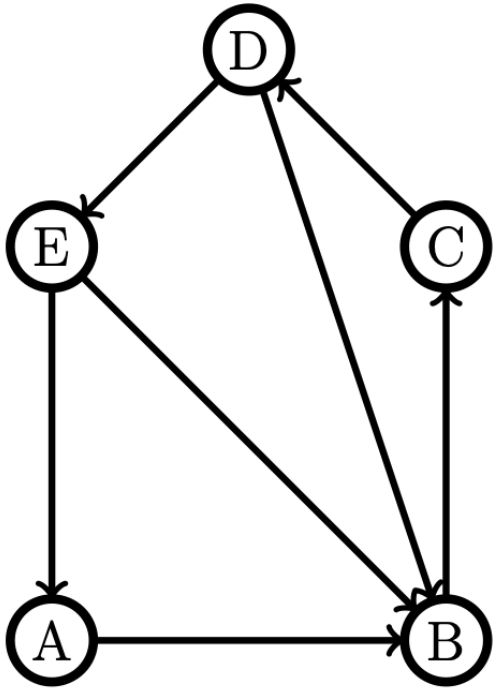


N…# pages on the web
M…# of pages spammer owns

- $y = \frac{x}{1-\beta^2} + c\frac{M}{N}$ where $c = \frac{\beta}{1+\beta}$
- For $\beta = 0.85$, $1/(1-\beta^2) = 3.6$

- Multiplier effect for "acquired" PageRank
- By making *M* large, we can make *y* as **large as we want**

Determine the nodes with the smallest and largest rank after running the PageRank algorithm on the network. Briefly justify your answer.



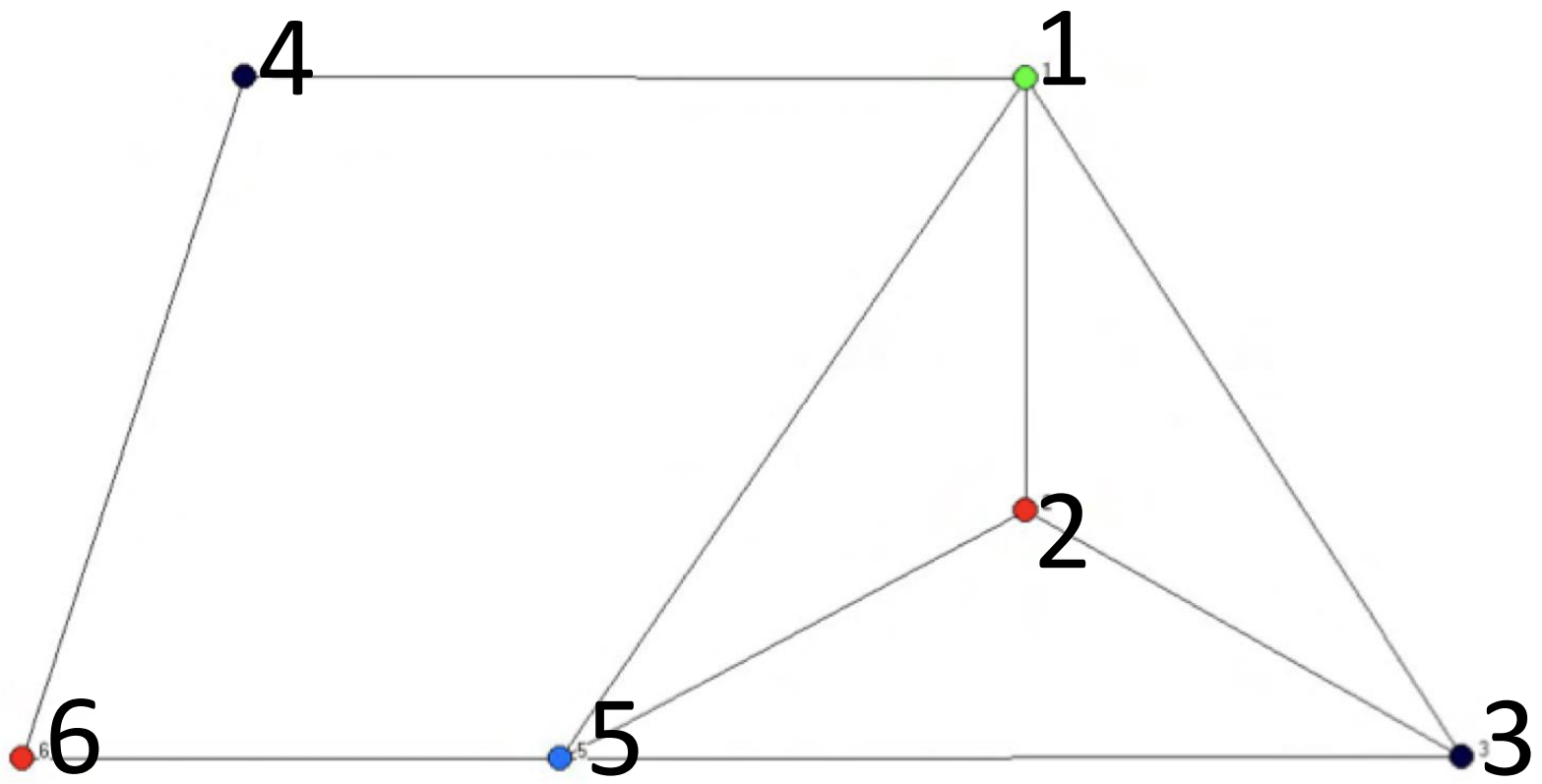
Node B has the largest page rank because it gets in-flows from nodes A, E and D. A has the smallest page rank because its in-flow comes from the stream connecting D to E and then A and half of this flow is already going to B in each step before getting delivered to A.

True or False?

- Homophily refers to a tendency of various types of individuals to associated with others who are similar to themselves and always contributes to a faster spread of information.

- **False -** *Homophily does not always contribute to faster spread of information. In fact, sometime the information may never reach a group of people if the network has a very strong component structures (formed based on individuals preferences) with very few or no links between components.*

Calculate Betweenness Centrality for individuals 2 & 5.
Calculate Closeness Centrality for individuals 3 & 1.



Answer: *Definition of betweenness centrality is:*

$$C_i^B(g) = \frac{2}{(n-1)(n-2)} \sum_{(j,k),j\neq k} \frac{P_i(j,k)}{P(j,k)}$$

Where $P_i(j,k)$ *is the number of shortest path between* j,k *that goes through* i*, and* $P(j,k)$ *is the number of shortest path between* j,k*. Using this equation for agent 2, we see that no shortest path between any two agents in this network goes through agent 2. Therefore, agent 2's betweenness centrality is equal to zero.*

Agent 5, however, is on several shortest path from agent $1,2,3$ *to agent* 6*, thus his centrality is:*

$$C_5^B(g) = \frac{1}{10}(\frac{P_5(16)}{P(16)} + \frac{P_5(26)}{P(26)} + \frac{P_5(36)}{P(36)}) = \frac{1}{10}(\frac{1}{2} + 1 + 1) = \frac{5}{20}$$

Answer: *Definition of closeness centrality is:*

$$C_i^C(g) = \frac{n-1}{\sum_j l(i,j)}$$

Where $l(i,j)$ *is the length of shortest path between* i, j*. Therefore,*

$$C_1^C(g) = \frac{5}{1+1+1+1+2} = \frac{5}{6}$$

$$C_3^C(g) = \frac{5}{1+1+1+2+2} = \frac{5}{7}$$