Economics of Networks

Econ 4v98/5v98
Prof. Scott Cunningham

January 28, 2014

Outline

Part 1: Background and Fundamentals

- Definitions and Characteristics of Networks
- Empirical Background.

Part 2: Network Formation

- Random Network models
- Strategic Network models

Part 3: Networks and Behavior

- Diffusion and Learning
- Games on Networks

Graphs and Networks

What is a graph? What is a network?

- A *graph* (N, g) and a *network* are largely synonymous terms in this class, and consist of
  - a set of nodes $N = \{1, ..., n\}$
  - a real-valued $n \times n$ *matrix*, g
- g_{ij} is the (i, j) element of the matrix.
- The matrix g is often referred to as the *adjacency matrix*, as it lists which nodes are linked to each other (or which nodes are adjacent to one another).

What is a Matrix?

- An $n \times m$ *matrix* is a rectangular array of numbers.
- Example:

$$a = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \tag{1}$$

- We refer to individual elements, a_{ij} by their row and column indices
- Ex: $a_{12} = 2$, $a_{22} = 5$, $a_{31} = 7$.
- Adjacency matrices for unweighted graphs have $g_{ij} = 1$ when there is an edge between i and j.
- $g_{ij} = 0$ otherwise

Graphs and Networks

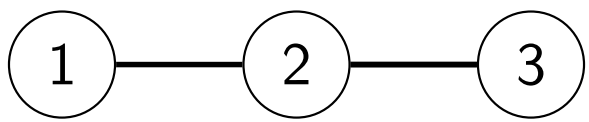
- We can allow g to take on more than two values, for example to keep track of the intensity of the level of relationships,
  - e.g. in a *weighted* graph.
- A network is *directed* if it is possible that $g_{ij} \neq g_{ji}$
- A network is *undirected* if it is not possible that $g_{ij} \neq g_{ji}$

Graphs and Networks

What is the adjacency matrix?

$$g = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

g is the *adjacency matrix* for the (undirected and unweighted) network on $N = \{1, 2, 3\}$:



Each element, g_{ij} is a zero or one. Diagonal is zero by convention as a node cannot be linked to itself.

- Links are often referred to as *edges* or *ties*; and as *arcs* in the case of directed graphs.
- Self-links or *loops* will often not have any real consequence. Unless otherwise indicated, assume that $g_{ii} = 0$ for all i.

Paths, Walks, and Cycles

Different ways of describing navigating a network:

- A *walk* is a sequence of links connecting a sequence of nodes. In a network $g \in G(N)$ between nodes i and j, a *walk* is a sequence of links $i_1 i_2, ..., i_{K-1} i_K$ such that $i_k i_{k+1} \in g$ for each $k \in \{1, ..., K-1\}$, with $i_1 = i$ and $i_K = j$.
- A *path* is a walk where a node appears at most once in the sequence. In a network $g \in G(N)$ between nodes i and j, a path is a sequence of links $i_1 i_2, ..., i_{K-1} i_K$ such that $i_k i_{k+1} \in g$ for each $k \in \{1, ..., K-1\}$, with $i_1 = i$ and $i_K = j$, and such that each node in the sequence $i_1, ..., i_K$ is distinct.
- A *geodesic* between nodes i and j is the **shortest path** between these nodes.

Paths, Walks, and Cycles

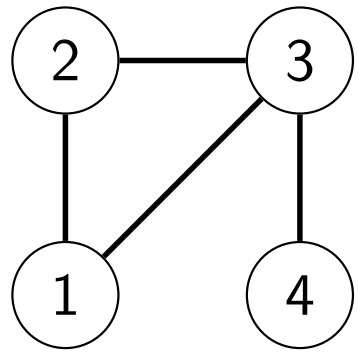
Different ways of describing navigating a network:

- A *cycle* is a walk that starts and ends on the same node, with all nodes appearing once except the starting node which also appears as the ending node.
- In a network $g \in G(N)$ between nodes i and j, a cycle is a walk $i_1 i_2, ..., i_{K-1} i_K$ such that $i_1 = i_K$, with all other nodes being distinct ($i_k \neq i_{k'}$ when $k < k'$ unless $k = 1$ and $k' = K$).

Paths, Walks, and Cycles

What else does the adjacency matrix tell us?

- If we have the following network, g



- And by convention let $g_{ii} = 0$, then the adjacency matrix is:

$$g = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Paths, Walks, and Cycles

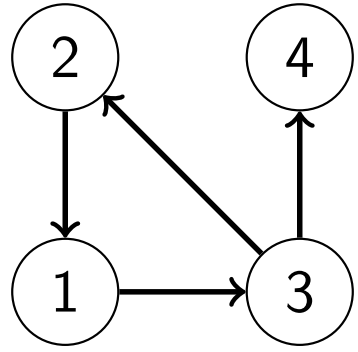
Matrix multiplication

- $g^2 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$
- $g_{12}^2 : (0 \times 1) + (1 \times 0) + (1 \times 1) + (0 \times 0) = 1$, etc.
- Interpretation: g^2 provides all possible walks of length 2 between any two nodes, including walks with many cycles within them
  - $g_{11} = 2$ because there are two walks of length 2 for node 1: From node 1 to 2 and back to 1, and from node 1 to 3 and back to 1.
  - $g_{32} = 1$ because there is only one walk of length 2 from node 3 to node 2: from node 3 to node 1, node 1 to node 2.
- For the kth power of the network, g^k would keep describe all possible walks of length k between any two nodes, including walks with many cycles within them.

Directed Paths, Walks, and Cycles

More formal definitions of directed networks

- A *directed walk* in a network $g \in G(N)$ is a sequence of links $i_1 i_2, ..., i_{K-1} i_K$ such that $i_k i_{k+1} \in g$ (that is $g_{i_k i_{k+1}} = 1$) for each $k \in \{1, ..., K-1\}$.
- A *directed path* in a network $g \in G(N)$ from node i to node j is a sequence of links $i_1 i_2, ..., i_{K-1} i_K$ such that $i_k i_{k+1} \in g$ (that is $g_{i_k i_{k+1}} = 1$) for each $k \in \{1, ..., K-1\}$, with $i_1 = i$ and $i_K = j$, and such that each node in the sequence $i_1, ..., i_K$ is distinct.
- A *directed cycle* in a network $g \in G(N)$ is a sequence of links $i_1 i_2, ..., i_{K-1} i_K$ such that $i_k i_{k+1} \in g$ (that is $g_{i_k i_{k+1}} = 1$) for each $k \in \{1, ..., K-1\}$, with $i_1 = i_K$.



A Directed Path from 2 to 4 (via 1 and 3); a Directed Cycle from 1 to 3 to 2 to 1; and a Directed Walk from 3 to 2 to 1 to 3 to 4

Undirected Paths, Walks, and Cycles

- In cases where the direction of the link just indicates who initiated the link, but where links can conduct in both directions, we can keep track of *undirected paths*. There we can think of i and j being linked if either $g_{ij} = 1$ or $g_{ji} = 1$.
- To be more specific, given a directed network g, let $\hat{g}$ denote the undirected network obtained by allowing an (undirected) link to be present whenever there is a directed link present in g. That is, let $\hat{g}_{ij} = max(g_{ij}, g_{ji})$. There is an undirected path between nodes i and j if there is a path between them in $\hat{g}$.
- Similarly, we defined an *undirected cycle* or *undirected walk*.

Quiz

Q1:Is the following matrix directed or undirected? Why?

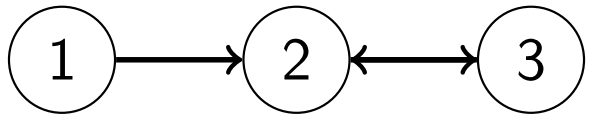
$$g = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Quiz

Q1:Is the following matrix directed or undirected? Why?

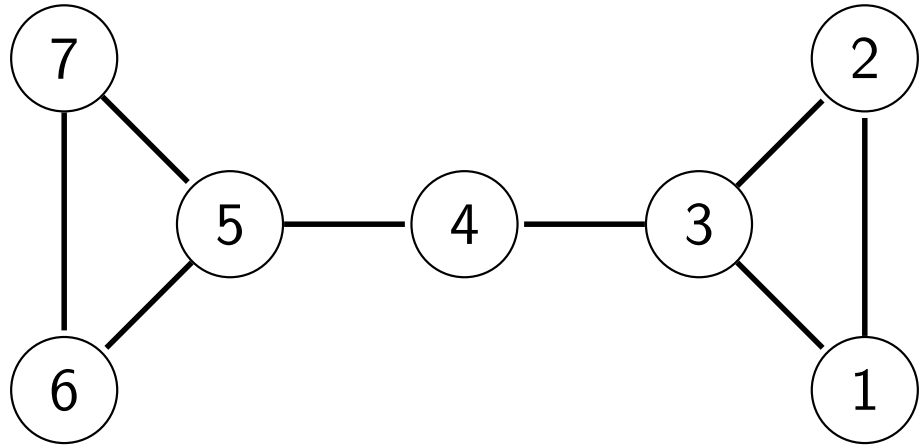
$$g = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

Ans: It's *directed* because $g_{21} \neq g_{12}$.



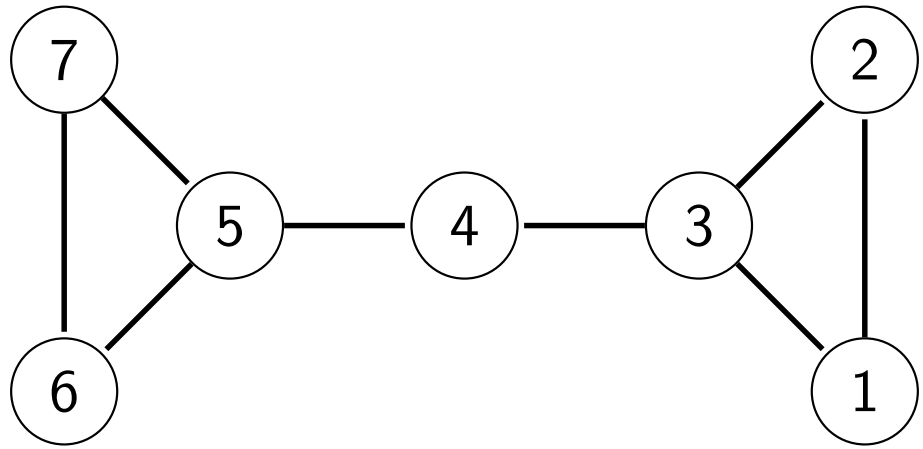
Quiz (cont)

Q2: Write the adjacency matrix for the following graph.



Quiz (cont)

Q2: Write the adjacency matrix for the following graph.



Ans:

$$g = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Components and Connected Subgraphs

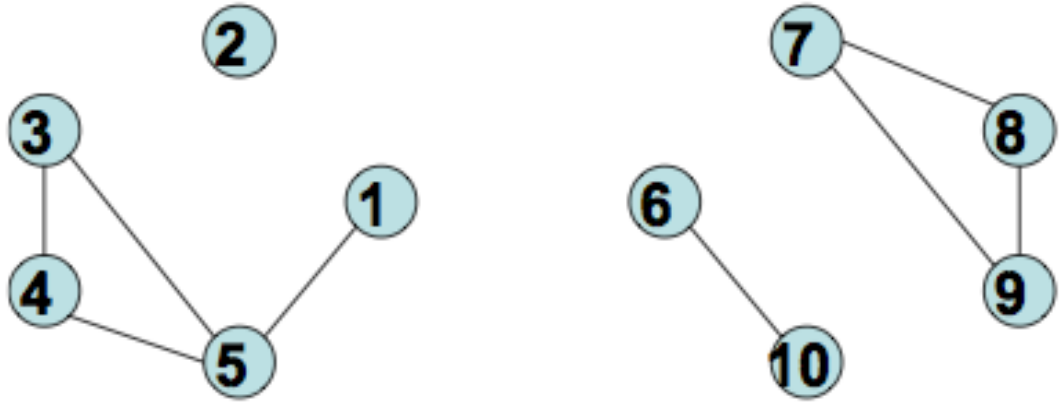
- We often care about the identify which nodes can reach which other nodes through paths in the network
    - Contagion, learning, diffusion
- Looking at path relationships in a network naturally partitions a network into different connected subraphs commonly called *components*
- A network (N, g) is *connected* (or path-connected) if every two nodes in the network are connected by some path in the network. That is, (N, g) is connected if for each $i \in N$ and $j \in N$ there exists a path in (N, g) between i and j.

Components and Connected Subgraphs

- A *component* of a network (N, g) is a nonempty subnetwork (N', g') such that $\emptyset \neq N' \subset N, g' \subset g$, and (N', g') is connected, and if $i \in N'$ and $ij \in g$, then $j \in N'$ and $ij \in g'$. In other words, the components of a network are the distinct maximal connected subgraphs of the network.
- The set of components of a network (N, g) is denoted $C(N, g)$. In cases where N is fixed of obvious, components can simply be denoted as $C(g)$.

Components and Connected Subgraphs

- Components of a network *partition* the nodes into groups within which nodes are path-connected. Let $\Pi(N, g)$ denote the partition of N induced by the network (N, g). That is, $S \in \Pi(N, g)$ if and only if $(S, h) \in C(N, g)$ for some $h \subset g$.
- A network is connected if any only if it consists of a single component, and so $\Pi(N, g) = \{N\}$.
- A link ij is a *bridge* in the network g if $g - ij$ has more components than g.
- In the case of a directed network, we refer to *strongly connected* graphs or subgraphs, so that each node can reach each other via a directed path.



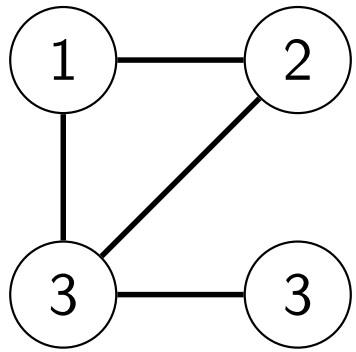
A Network with Four Components

Trees, Stars, Circles, and Complete Networks

- A *tree* is a connected network that has no cycles.
- A connected network is a tree if and only if it has $n - 1$ links.
- A tree has at least two *leaves*, where leaves are nodes that have exactly one link.
- In a tree, there is a unique path between any two nodes.

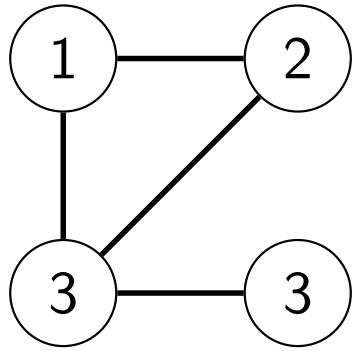
Quiz

Q3: Is the following network a tree? Why/why not? Provide a drawing that is a tree if not.

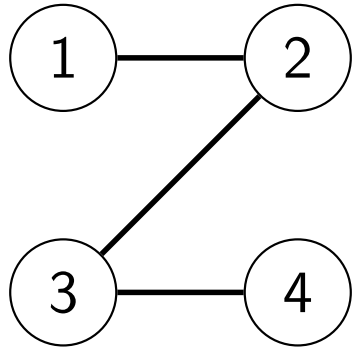


Quiz

Q3: Is the following network a tree? Why/why not? Provide a drawing that is a tree if not.

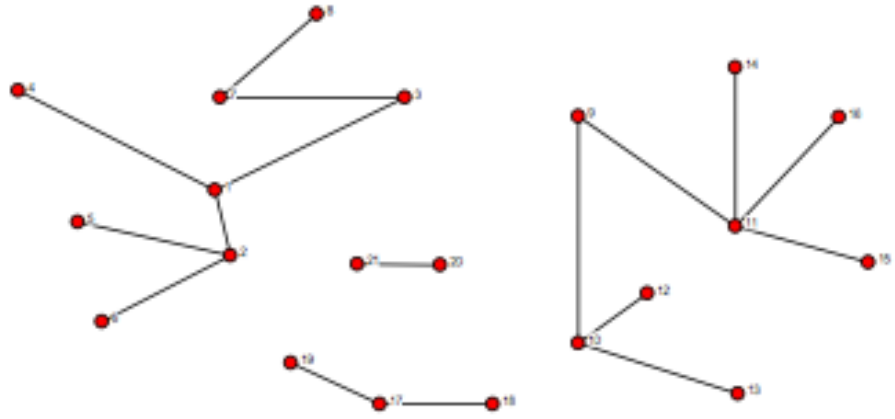


Ans: It is not, because trees have $n - 1$ edges, and this network has 4 nodes and 4 edges. The following is, though.

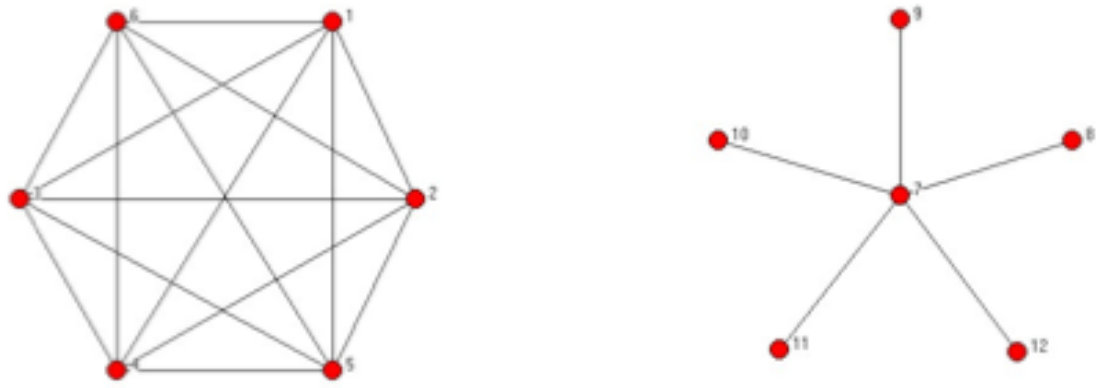


Trees, Stars, Circles, and Complete Networks

- A *forest* is a network such that each component is a tree. Any network that has no cycles is a forest.
- A *star* is a network such that there exists some node i such that every link in the network involved node i. In this case, i is referred to as the *center* of the star.
- A *circle*, or cycle-graph, is a network that has a single cycle and such that each node in the network has exactly two neighbors.
- A *complete network* is one where all possible links are present, so one where $g_{ij} = 1$ for all $i \neq j$.



Four Trees in a Forest



A Complete Network on Six Nodes and a Star Network on Six Nodes

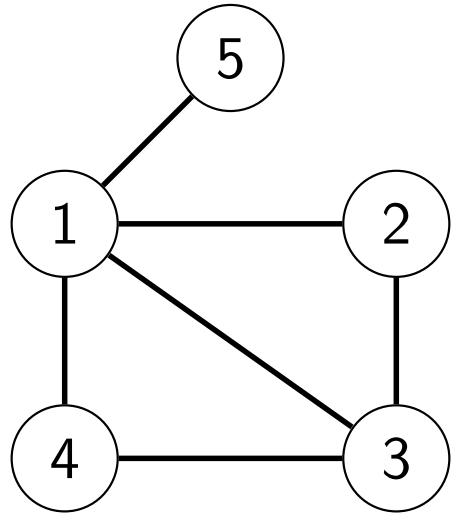
Centrality

Measures of centrality can be categorized into four main groups:

1. degree: how connected a node is;
2. closeness: how easily a node can reach other nodes;
3. betweenness: how important a node is in terms of connecting other nodes;
4. neighbors' characteristics: how important, central, or influential a node's neighbors are.

Neighborhoods and Degree

- The *neighborhood of a node i*, $N_i(g)$, is the set of nodes that i is linked to. That is $N_i(g) = \{j : g_{ij} = 1\}$..
- The *degree* of a node, $d_i(g)$, is the number of links that involve that node, which is the cardinality of i's neighborhood. Thus we define node i's degree in network g as $d_i(g) = \#\{j : g_{ji} = 1\} = \#N_i(g)$.
- *Network density* keeps track of the relative fraction of links that are present, and is equal to the average degree divided by $n - 1$, $\frac{\sum_i^n d_i(g)}{(n-1)}$



- The above undirected network, g, has five 5 nodes.
  - Node 2's neighborhood is $N_2(g) = \{1, 2\}$.
  - Node 1 has a degree of 4 ($d_1(g) = 4$),
  - Node 3 has degree 3 ($d_3(g) = 3$)

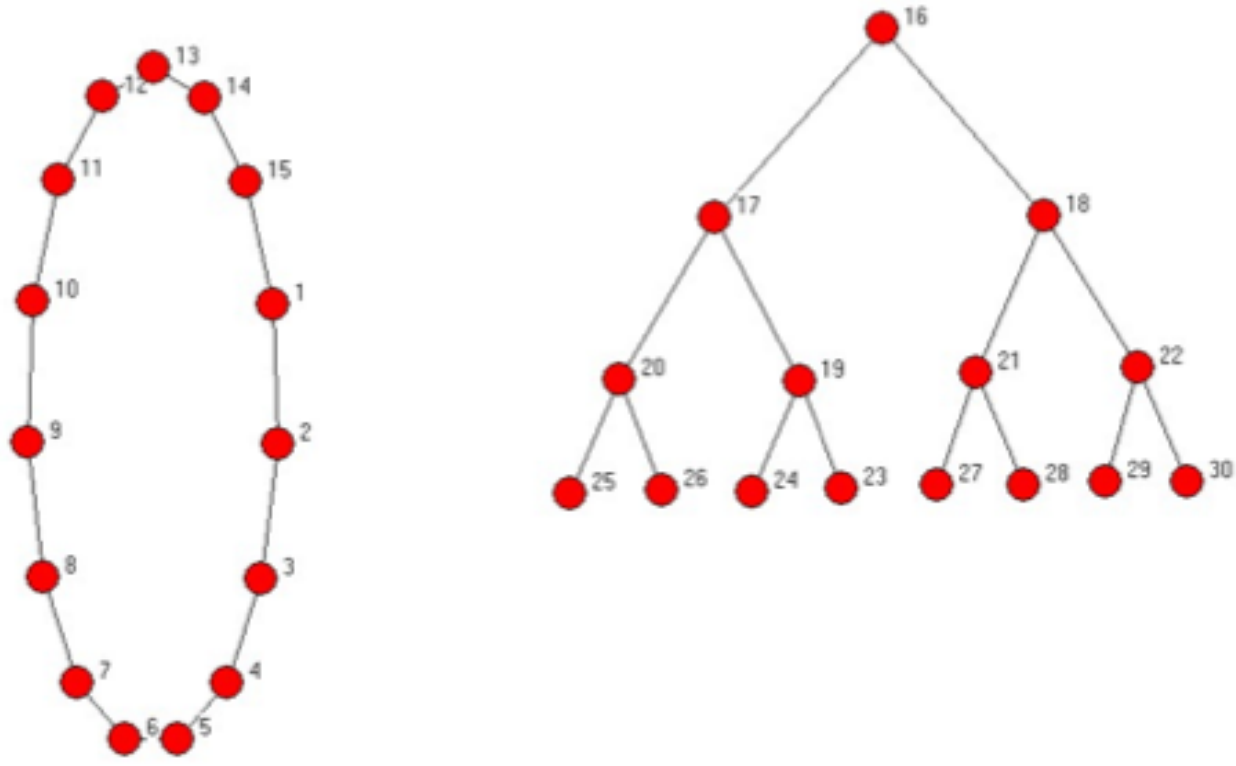
Distance and Diameter

More definitions!

- The *diameter* of a network is the largest geodesic in the network.
  - where *geodesic* is the largest shortest path between any two nodes in the network.
- The *average path length* (or characteristic path length) between nodes takes the average over geodesics. The average path length will be bounded above by the diameter.
- Networks that are similar in degree may be very different in structure.
  - Diameter of a circle is either $\frac{n}{2}$ or $\frac{(n-1)}{2}$ depending on even or odd number of nodes
  - Diameter of a tree, there are $n = 2^{K+1} - 1$ nodes in a binary tree and K levels. First solve for K

$$\begin{aligned} n &= 2^{K+1} - 1 \\ n+1 &= 2^{K+1} \\ log_2(n+1) &= K + 1 log_2(2) \\ log_2(n+1) &= K + 1 \\ K &= log_2(n+1) - 1 \end{aligned}$$

  - diameter$= 2K$ for tree. Can you see why?



Circle vs. Binary Tree

- Diameter of circle network is $\frac{(n-1)}{2} = \frac{14}{2} = 7$.
- Diameter of tree network is
  $2K = 2(log_(n + 1) - 1) = 2(log_2(16) - 1) = 2(4 - 1) = 6$

Calculating Shortest Path Lengths

- The shortest path length between nodes i and j can be found by finding the smallest ℓ such that the ij-th entry g^ℓ is positive, and then that entry is the number of shortest paths between those nodes.
- Calculating shortest path lengths for all pairs of nodes provides a basic method of calculating or estimating diameter.

Degree Centrality

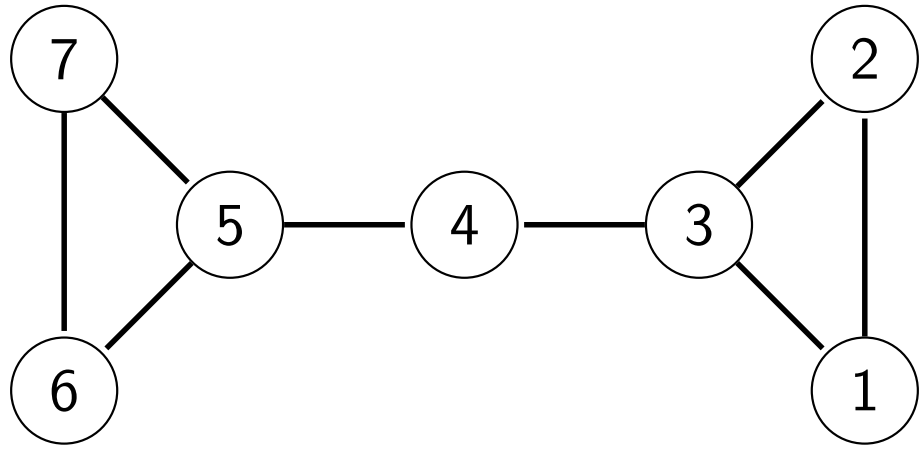
How connected is this node within the network?

- The *degree centrality* of a node tells us how well a node is connected in terms of direct connections:

$$\frac{d_i(g)}{(n-1)},$$

  ranges from 0 to 1
- Degree centrality misses any aspect of how well located a node is in a network. It might be that a node has relatively few links, but lies in a critical location of the network.

Degree Centrality



A Central Node with Low Degree Centrality

- Node 1: $\frac{d_1}{n-1} = \frac{2}{6} = 0.33$
- Node 2:$= \frac{2}{6} = 0.33$
- Node 3:$= \frac{3}{6} = 0.50$
- Node 4:$= \frac{2}{6} = 0.33$
- Node 5:$= \frac{3}{6} = 0.5$
- Node 6:$= \frac{2}{6} = 0.33$
- Node 7:$= \frac{2}{6} = 0.33$

Closeness Centrality

Another obvious way to measure centrality is measure how long it takes for the node to reach other nodes

- *closeness centrality*: the inverse of the average distance between i and any other node:

$$\frac{(n-1)}{\sum_{j \neq i} \ell(i,j)},$$

where $\ell(i,j)$ is the number of links in the geodesic between i and j.

Decay centrality

Variations on "closeness" exist – weight the further geodesics less than the closer ones

- *Decay centrality*: weight each node's decay parameter, δ, where $0 < \delta < 1$, by the geodesic corresponding to path *ij*:

$$\Sigma_{j \neq i} \delta^{\ell(i,j)},$$

  where $\ell(i, j)$ is set to infinity if *i* and *j* are not path-connected.
- As δ gets close to one, decay centrality measures how the size of the component a node lies in. As δ gets close to zero, decay centrality gives infinitely more weight to closer nodes than farther nodes and becomes proportional to degree centrality.

Betweenness Centrality

What about the importance of being unavoidable?

- *Betweenness centrality* equals:

$$C_{e_i}^{B}(g) = \frac{P_i(kj)}{P(kj)}.$$

  where $P_i(kj)$ denote the number of geodesics between node k and j that node i lies on, and $P(kj)$ the total number of geodesics between k and j.
- Betweenness centrality will usually be a fraction, and when it is close to 1, it means node i lies on most of the shortest paths connecting k to j, but if close to 0, then i avoidable on geodesics from k and j.

Betweenness Centrality

- Averaging across all pairs of nodes, the *betweenness centrality* of a node i is

$$Ce_i^B(g) = \Sigma_{k \neq j; i \notin \{k,j\}} \frac{P_i(kj)/P(kj)}{(n-1)(n-2)/2}.$$

Eigenvector Centrality

- Degree centrality doesn't capture the importance of one's friends; idea of *eigenvector centrality* is that importance comes from being connected to other important nodes
- The *eigenvector centrality* of a node is proportional to the sum of the centrality of its neighbors:

$$\lambda C_i^e(g) = \Sigma_j g_{ij} C_j^e(g).$$

- In terms of matrix notation,

$$\lambda C^e(g) = g C^e(g),$$

  where λ is a proportionality factor. Thus, $C^e(g)$ is an eigenvector of g, and λ is its corresponding eigenvalue.
- If $g_{ij} = 1$, then C_j is counted, and if $g_{ij} = 0$, it's not counted

Eigenvector centrality

- This is a system of equations in a system of unknowns
- To figure out C_i, we have to figure out C_j
- Eigenvectors have many possible solutions – generally up to n different solutions; look for the solution with the largest eigenvalue (Perron-Frobenius theorem which says if we are dealing with a non-negative matrix, then the eigenvector associated with the largest eigenvalue is going to have non-negative values)
- Use software to calculate eigenvalues for you, as they get annoying pretty fast otherwise
- Before calculating it, let's look at some not so distant history – Google PageRank

Eigenvector centrality example: PageRank

- PageRank is an algorithm used by Google Search to rank websites in their search engine results; did you know it was named after Larry Page, one of Google's founders?
- PageRank is a way of measuring importance of website pages

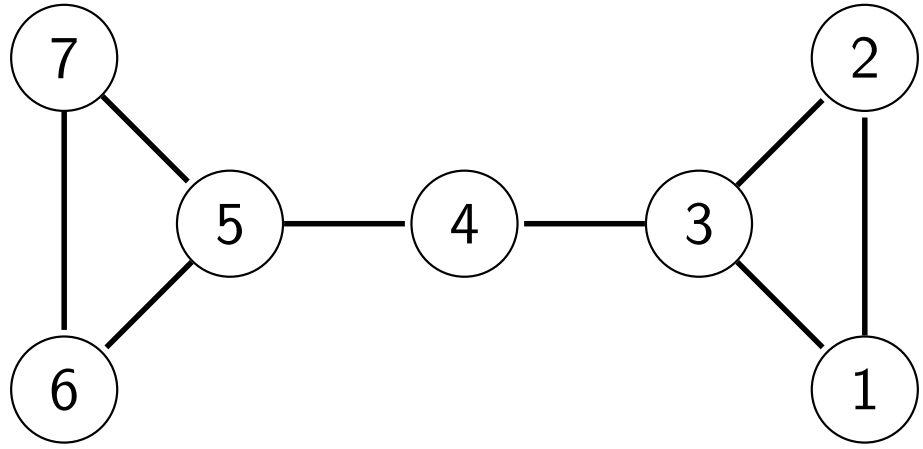
  > *"PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites."*

- It's not the only algorithm Google uses now – they use many – but it was the first one, and is the cause of the company's initial success in search

PageRank

- The score of a page is proportional to the sum of the scores of pages linked to it
- This helped Google dominate the search market early on
- So when you'd search, a bunch of pages would come up – but the order in which those came up was ordered by Google according to the ones with the largest eigenvector centrality values
- PageRank was looking for pages that were important, and eigenvector centrality was the measure of its importance
- Go back to the bowtie network from earlier and write down the adjacency matrix

Bowtie and adjacency matrix



$$g = \begin{pmatrix} 0 & 1 & 1 & & & & \\ 1 & 0 & 1 & & & & \\ 1 & 1 & 0 & 1 & & & \\ & & 1 & 0 & 1 & & \\ & & & 1 & 0 & 1 & 1 \\ & & & & 1 & 0 & 1 \\ & & & & 1 & 1 & 0 \end{pmatrix} \tag{2}$$

Eigenvector centrality

- We are trying to solve the eigenvalue problem:

$$agC = C$$

- In eigenvalue problems, the constant is usually on the side opposite the matrix; so define $a = \frac{1}{\lambda}$ and rewrite the above equation as

$$\frac{1}{\lambda} = C$$

,

$$gC = \lambda C$$

- Use MATLAB to calculate eigenvalues:
  - `www.compileonline.com/execute_matlab_online.php`
- We want the sum to add to one, so we divide the last column by the sum of itself (2.34292) to find centralities by node

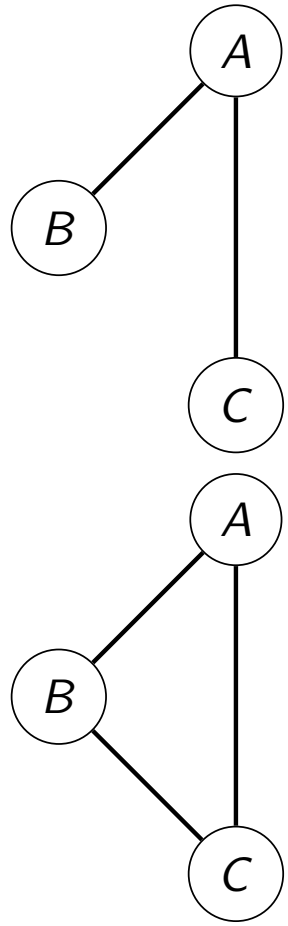
$$\text{eigenvalues} = \begin{pmatrix} 0.12768 \\ 0.12768 \\ 0.17146 \\ 0.14637 \\ 0.17146 \\ 0.12768 \\ 0.12768 \end{pmatrix} \tag{3}$$

Centrality

- As with "clustering", theory tells us that network position matters.
  - Structural holes
  - embedded links
  - local bridges
- Again, how to consistently measure "important" positions is less clear.

Triadic closure

- *Triadic closure*: If two people in a social network have a friend in common, then there's a higher probability that they will become friends themselves at some point in time.
- New structure is called *triangle* – triadic closure "closes" the third side of the triangle



Clustering

- The *individual* clustering coefficient is

$$Cl_i(g) = \frac{\sum_{j \neq i; k \neq j; k \neq i} g_{ij} g_{ik} g_{jk}}{\sum_{j \neq i; k \neq i} g_{ij} g_{ik}}.$$

- In words... the probability that two randomly selected friends of node A are friends with each other; equal to the fraction of pairs of A's friends *already* connected by edges

Clustering

Two options for measuring clustering of the network,

1. *Overall clustering*,

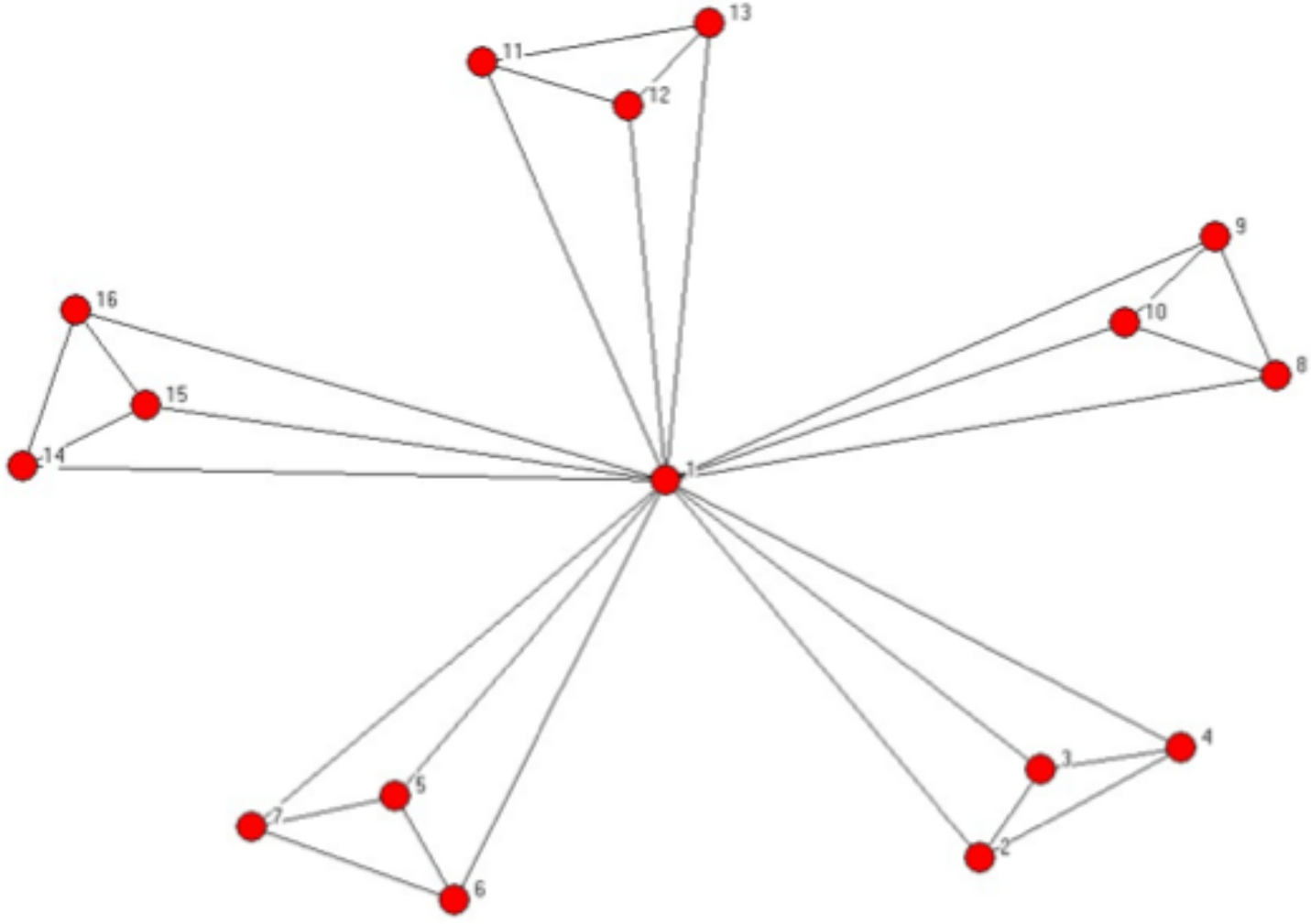
$$Cl(g) \quad = \quad \frac{\Sigma_i \#\{kj \text{ in } g | k, j \text{ in } N_i(g)\}}{\Sigma_i \#\{| k, j \text{ in } N_i(g)\}}$$

   This repeats the individual clustering calculation across all pairs of edges (not just those involving i).

2. *Average clustering coefficient*

$$Cl^{Avg}(g) = \Sigma_i Cl_i(g)/n.$$

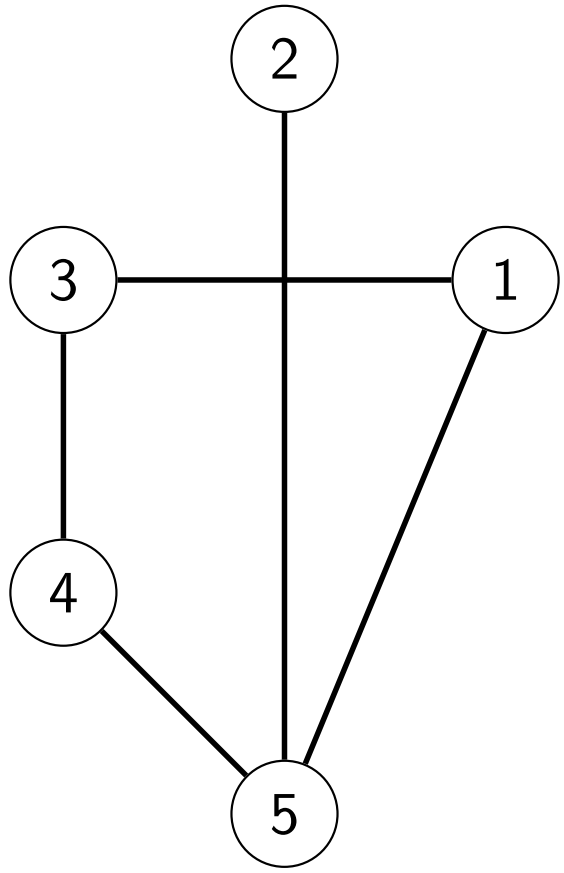
   Under average clustering, one gives more weight to the low-degree nodes than the clustering coefficient.



Differences in Clustering Measures

Quiz

- Q4: Calculate the clustering coefficient, $Cl_1(g)$ for Node 1.
- Q5: Calculate the average clustering, $Cl^{avg}(g)$ for the whole network, g.
- Q6: Calculate the overall clustering, $Cl(g)$.



Reasons for triadic closure

1. More opportunities to meet
2. Bayesian Trust – the fact that A and B are friends, and A and C are friends, then B and C may have information about each other based on the observed friendship with A.
    - I'll accept a friend request from a friend of a friend on Facebook if we are both commenting in his thread for almost no other reason than that I figure I'll like this guy – a friend of a friend is my friend?
3. Incentive and stress reduction
    - My son experiences genuine stress whenever he has a party because inevitably he'll have to invite people across his social network who don't know one another; much easier if they just knew each other

Bearman and Moody found that teenage girls with low clustering coefficients in their network of friends are significantly more likely to contemplate suicide than those whose clustering coefficient is high. What else might explain this? (Causality and Correlation are not the same...)

Strength of Weak Ties

- Mark Granovetter 1960s sociology student did his thesis on job referrals for people who just got new jobs and found
  1. 16.7% found new job through a strong tie (i.e., at least 2 interactions per week)
  2. 55.7% via medium tie (i.e., at least 1 interaction per year)
  3. 27.6% via weak tie (i.e., $<$1 interaction per year)
- Theory: weak ties form "bridges". Even though weak ties were people with whom you had little interaction, they still accounted for over a quarter of the information people were getting in how they managed to find their jobs.

Strength of Weak Ties

- Think about information flow throughout network as moving from node to node
- The people with whom you have strong ties – your spouse, your children, your colleagues, people at church, your best friends – are a very small number probably, maybe a dozens or less
- But if you think of the number of people who you have ever known in your life – weak ties – then the number is much much higher, maybe on the order of thousands
- So you have many, many weak ties but only a few strong ties, and even though you don't interact as much with your weak ties, it could be there is just more of them

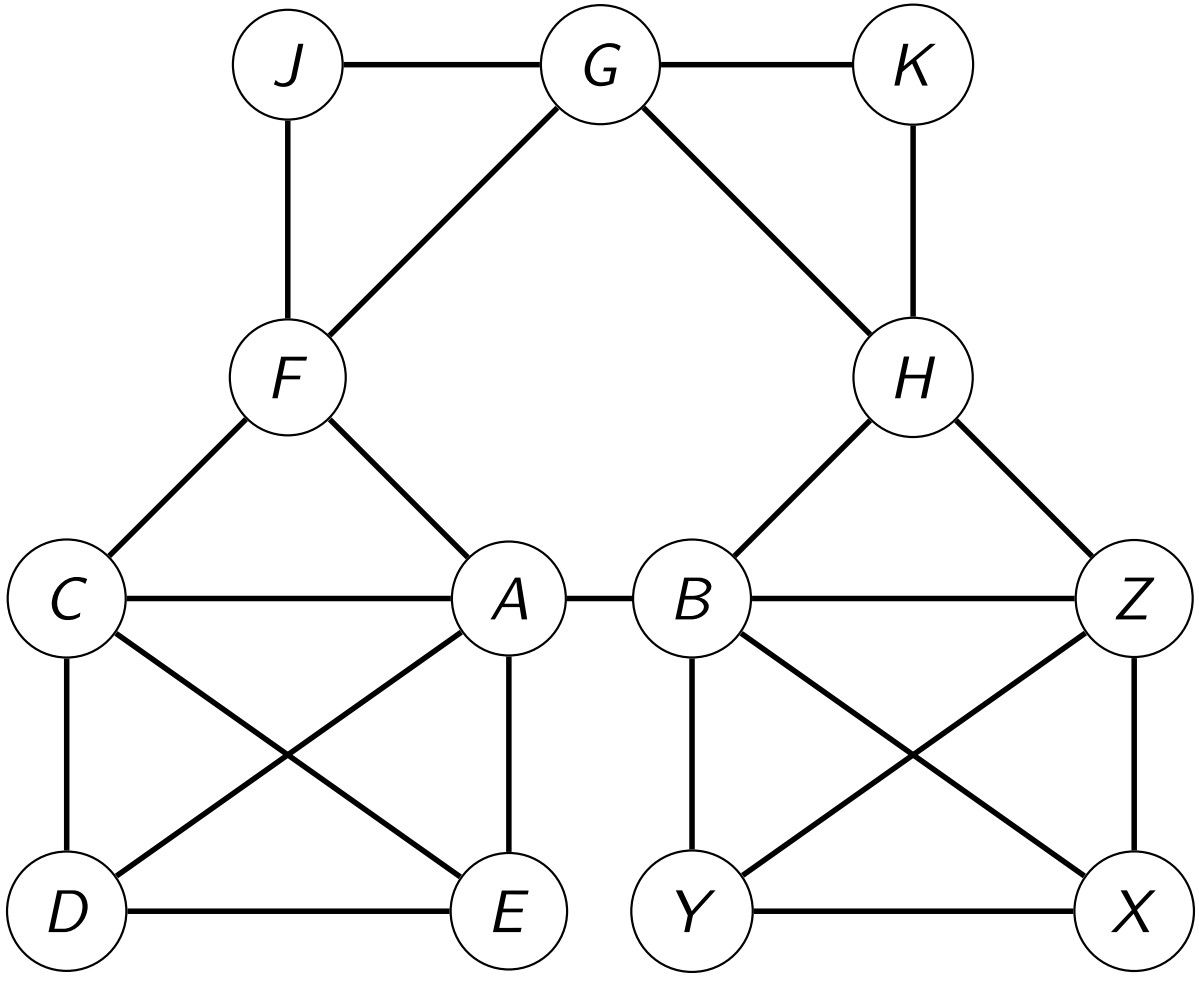
Strength of Weak Ties

- The interesting point that Granovetter was making, though, wasn't that weak ties were greater in number – it was a network explanation too
- Weak ties could be connecting you to another part of the network
- When you look at friendships by race, you typically find a tremendous amount of clustering by race, but not across race – weak ties are often bridging racial networks
- Bridging might occur more often with weak ties, in other words, and even though you interact with them relatively infrequently, they are important because weak ties connect you to another part of the network you wouldn't otherwise access
- Accessing information isn't redundant with respect to weak ties, but may be with respect to strong ties

Bridges

- An edge that joins two nodes A and B in a graph is called a *bridge* if deleting the edge would cause A and B to lie in two different components
- In other words, this edge is literally the only path between its endpoints, A and B.
- Small world theory (to be discussed later), though, suggests bridges are probably rare phenomena – if we are all separated by six path lengths, then by definition there are no bridges
- But this idea of a bridge can have global and local properties – we may say that an edge between two nodes in a graph is a *local bridge* if its endpoints A and B have no friends in common.
- In other words, deleting that edge would increase the geodesic between A and B to a value strictly more than 2.
- We say that the *span* of a local bridge is the distance its endpoints would be from one another if the edge were deleted (see next slide for en example)

Local bridges



Strong Triadic Closure Property

- The geodesic path length from A to B falls from 1 to 4 if we remove AB
- A to B is a local bridge across this semi-tree network shortening the geodesic, and depending on the location of the edge, could reduce the diameter of a network if strategically placed

Strong Triadic Closure Property

- Strong Triadic Closure Property
  - assume all edges are either strong or weak, where strong ties are friends and weak ties are acquaintances, and relabel a network accordingly
  - Moves towards a weighted graph as opposed to undirected graphs
  - If a node A has edges to nodes B and C, then the B-C edge is especially likely to form if A's edges to B and C are both strong ties
  - Granovetter extreme version: "We say that node A violates the Strong Triadic Closure property if it has strong ties to two other nodes B and C, and there is no edge at all (either a strong or weak tie) between B and C. We say that a node A satisfies the Strong Triadic Closure property if it does not violate it."
- No node in the previous figure will violates the Strong Triadic Closure Property but if AF was strong instead of weak, then nodes A and F would both violate the Strong Triadic Closure Property

Local Bridges, Weak Ties

- This gave us a micro and macro measurement
  - Strong and weak ties is a micro concept describing interpersonal nodes
  - Local bridges is a structural, global concept
  - On the surface, no direct connection either between these two concepts
- Using triadic closure, we have a claim: If a node A in a network satisfies the Strong Triadic Closure property, and it is involved in at least two strong ties, then any local bridge it is involved in must be a weak tie.
  - In other words, under the assumption of Strong Triadic Closure property and a sufficient number of strong ties, the local bridges in a network are *necessarily weak ties* under this claim

Proof by contradiction

- Take some network (N, g) where node $A \in N$. Let A satisfy strong triadic closure property and is involved in at least two strong ties.
- Suppose A is involved in a local bridge to B in a strong tie
- Contradiction
  - Strong Triadic Closure says AB must exist
  - By definition of a local bridge, it cannot exist

- Small networks can easily be described by g
  - Easily illustrated in a figure
  - or looking at adjacency matrix
- Larger networks are hard to envision and describe..
- We want statistics that describe recurring patterns and overall features of the network.

Simplifying Observed Complexity

- Global patterns of networks
  - degree distribution, path length, clustering, ...
- Segregation patterns
  - node characteristics and *homophily*
- Local patterns
  - Clustering, cycles, cliques
- Position in networks
  - Centrality, influence, bridges/structural holes

Erdos-Renyi

- Start with n nodes.
- Each potential link is formed with probability p
- This is the benchmark model for network formation in the literature (see Ch. 2 of *Linked*).

Erdos-Renyi (1959,1960) Random Graphs

- Even this simple model has some striking features that can be characterized formally with probability theory.
- The presence (or absence) of a particular link is a *Bernoulli* random variable (coin toss).
- This means that the degree of a node is a Binomial random variable
  - Recall that a binomial rv measures the number of successful coin tosses
  - Here, for a given link, measures the number of "successful" edges.
- When n gets very large relative to p, the expected degree can be approximated by a Poisson random variable.
- So, sometimes ER random graphs are called *Poisson random graphs*.

Erdos-Renyi (1959,1960) Random Graphs

- ER random graph model predicts that we will all be connected...

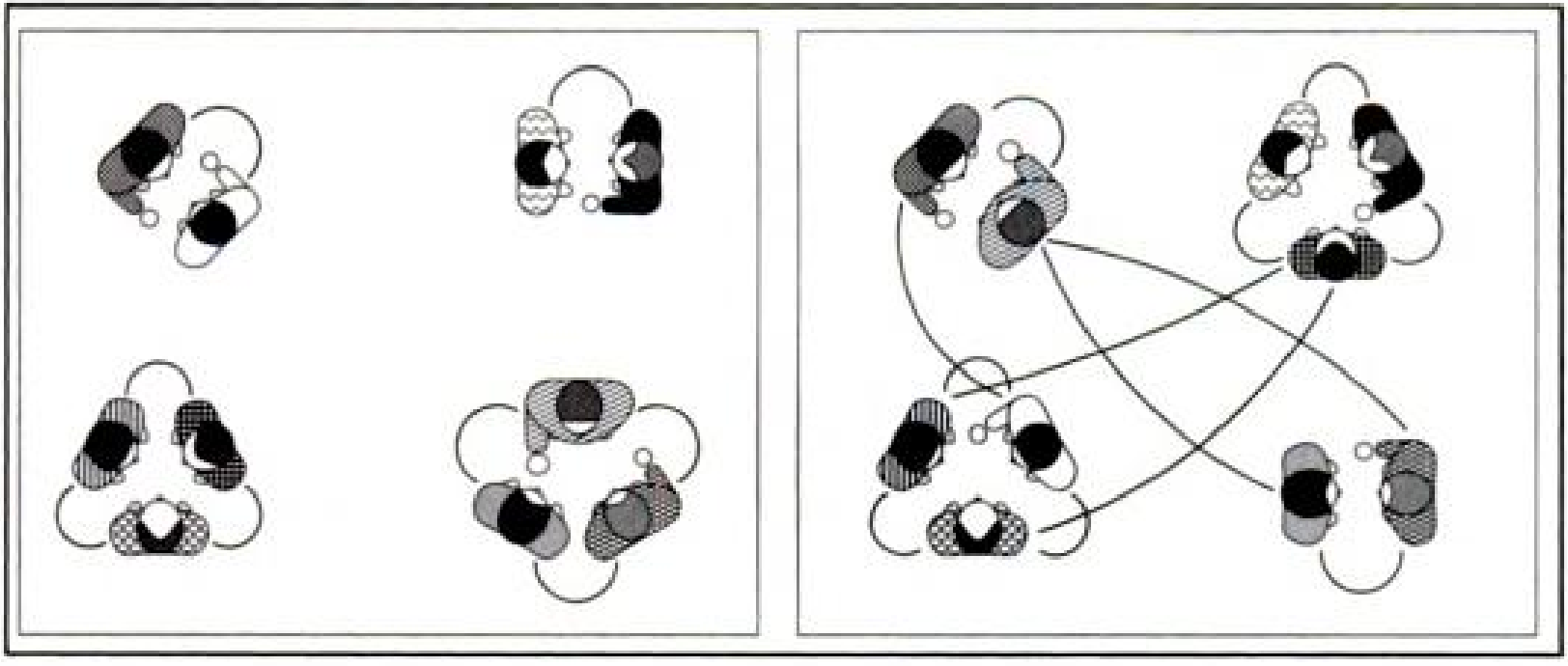


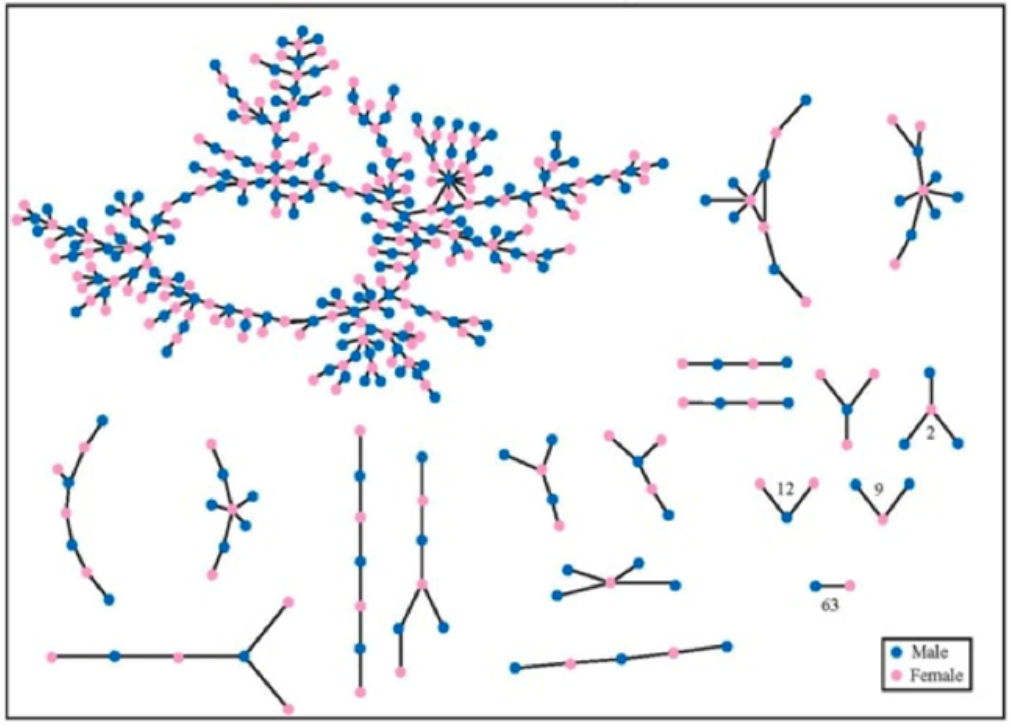
Figure 2.2 **The Party.** *At a party with ten guests, none of whom initially knows*

Erdos-Renyi (1959,1960) Random Graphs

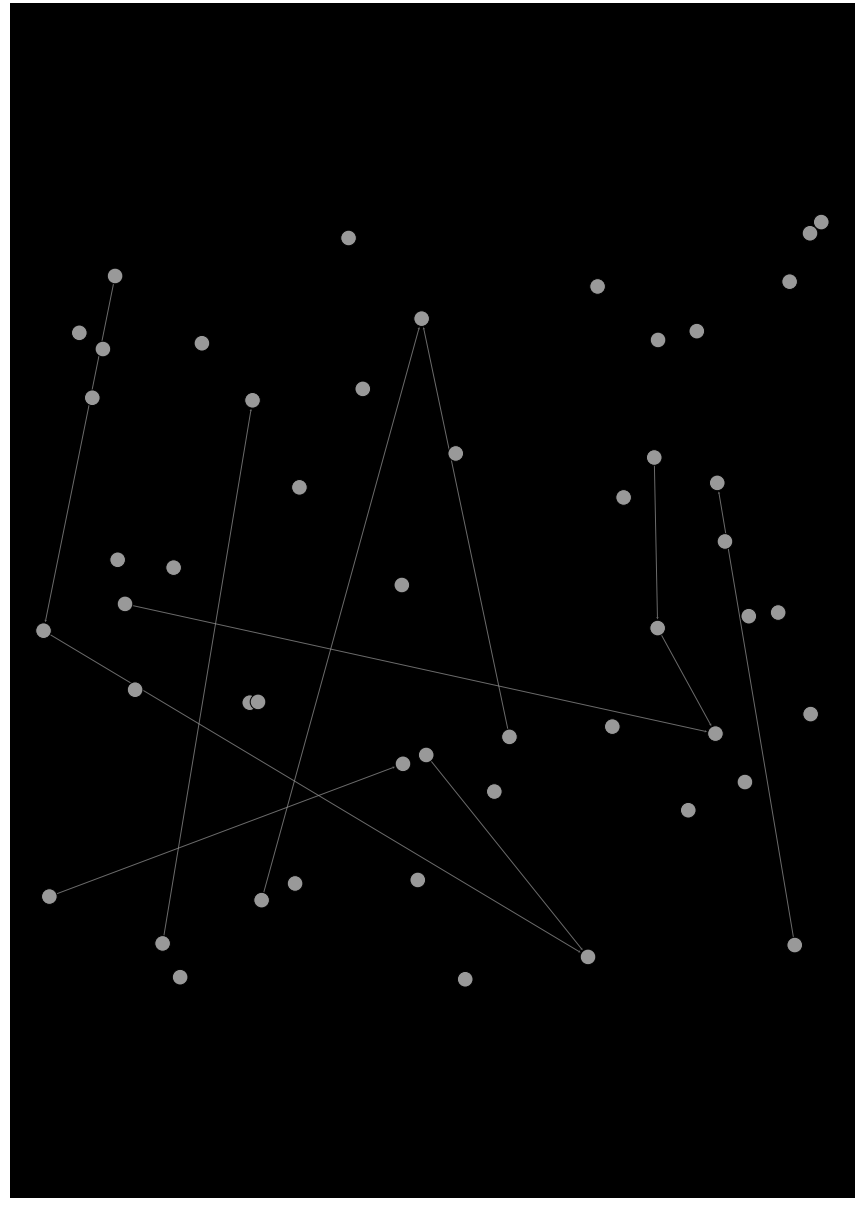
- ER random graph model predicts that we will all be connected...
- As the probability of connection, p, increases
  - The number of edges goes up, (obviously)
  - The size of components gets small
  - i.e. a giant component emerges *abruptly*
  - WHen $p > \frac{1}{n}$ (expected degree is 1)
  - The giant component contains (almost) all nodes beyond a higher threshold
- Metaphors
  - Phase transition from water to ice (physics)
  - Community formation (sociology)

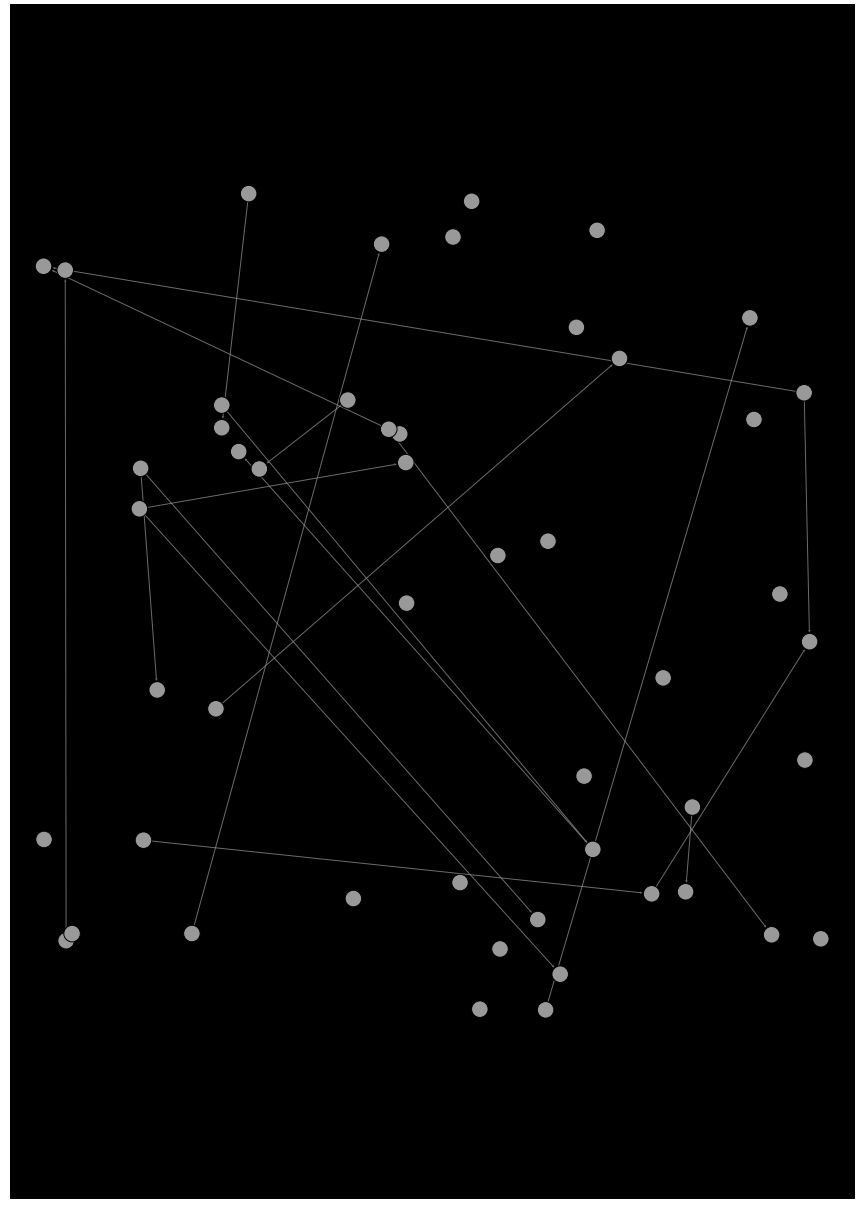
Density in the Real World

- In the Facebook, density is about 120.
- Romantic relationships, density = 0.8.

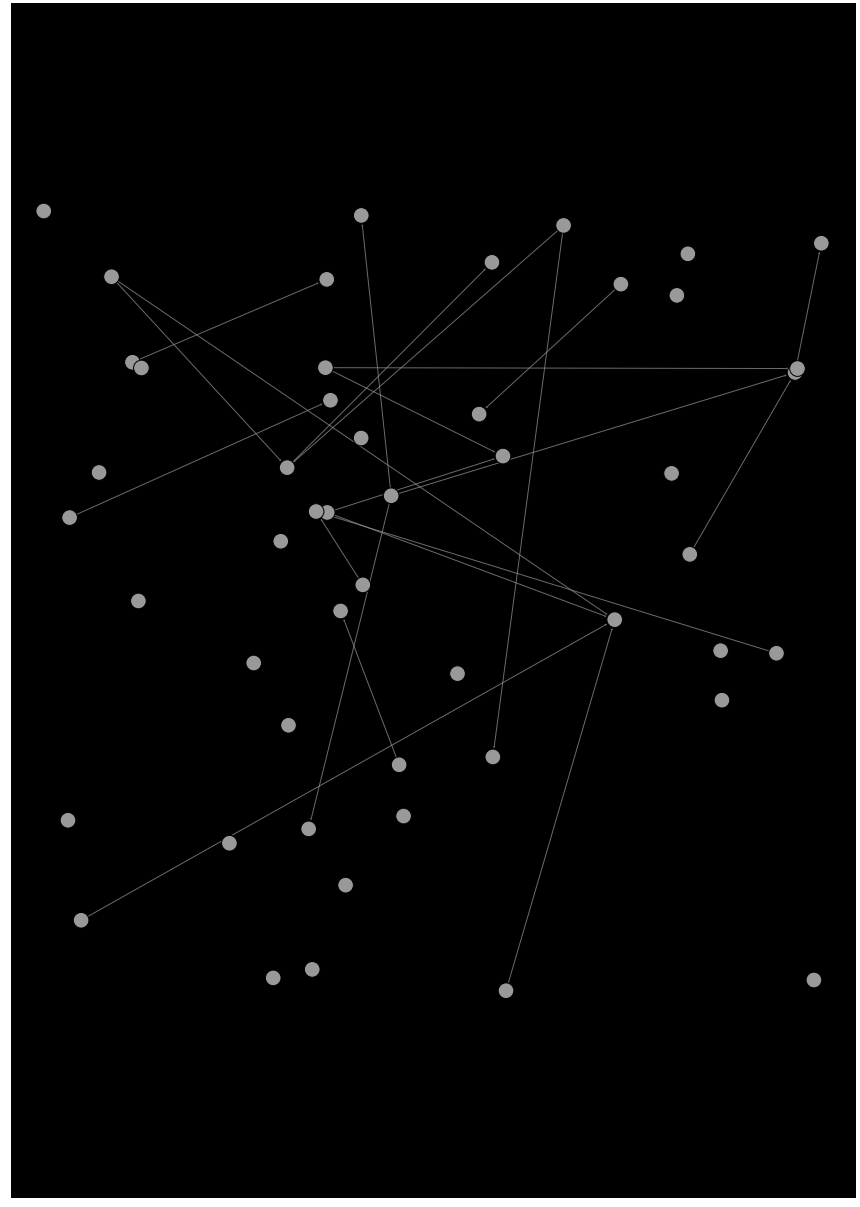


$N = 50, p = 0.01.$

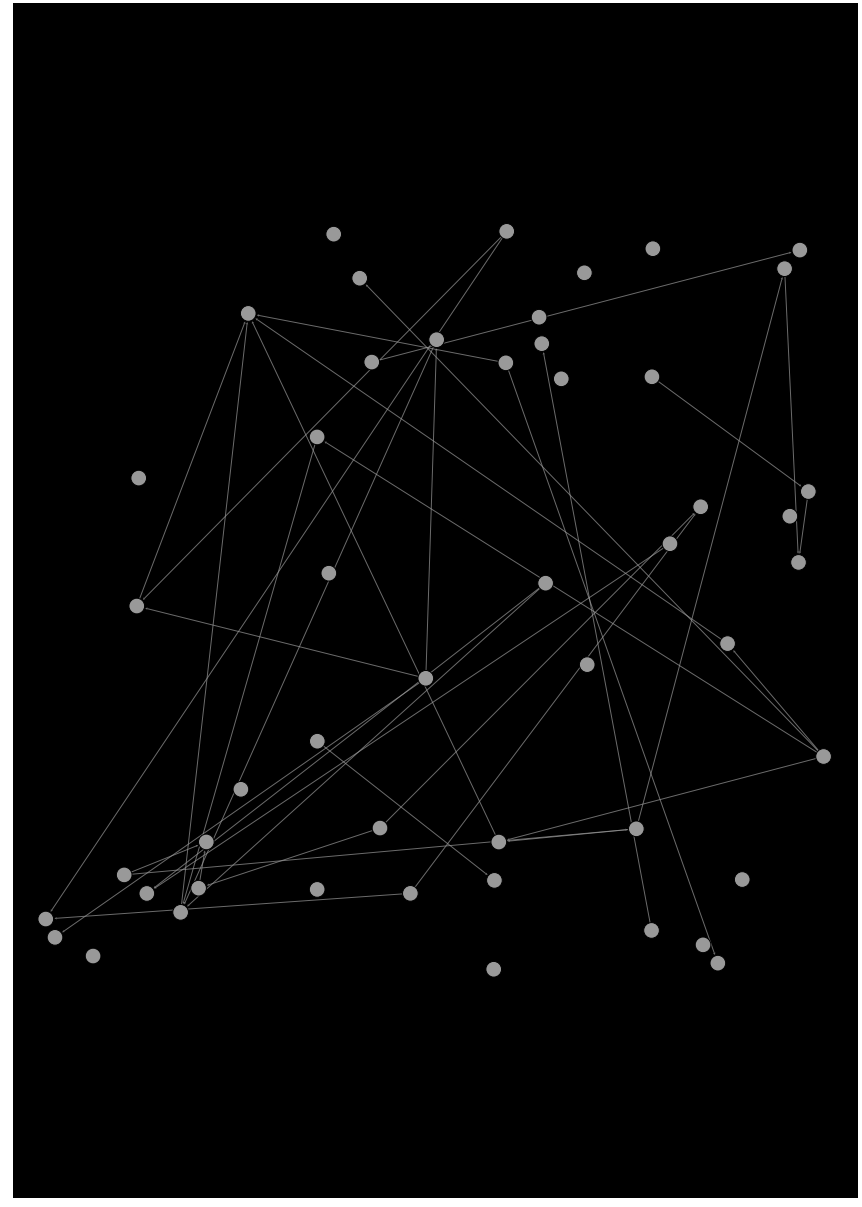




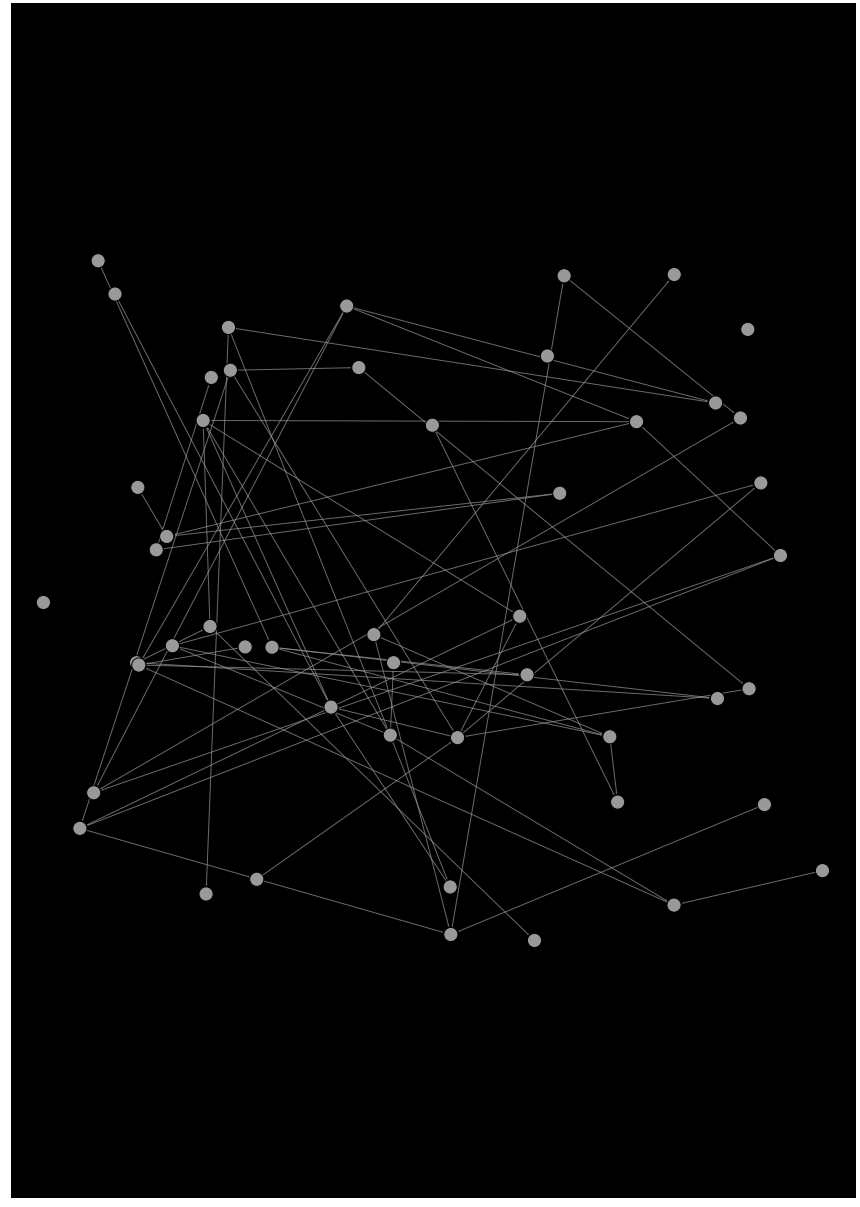
$N = 50, p = 0.015$



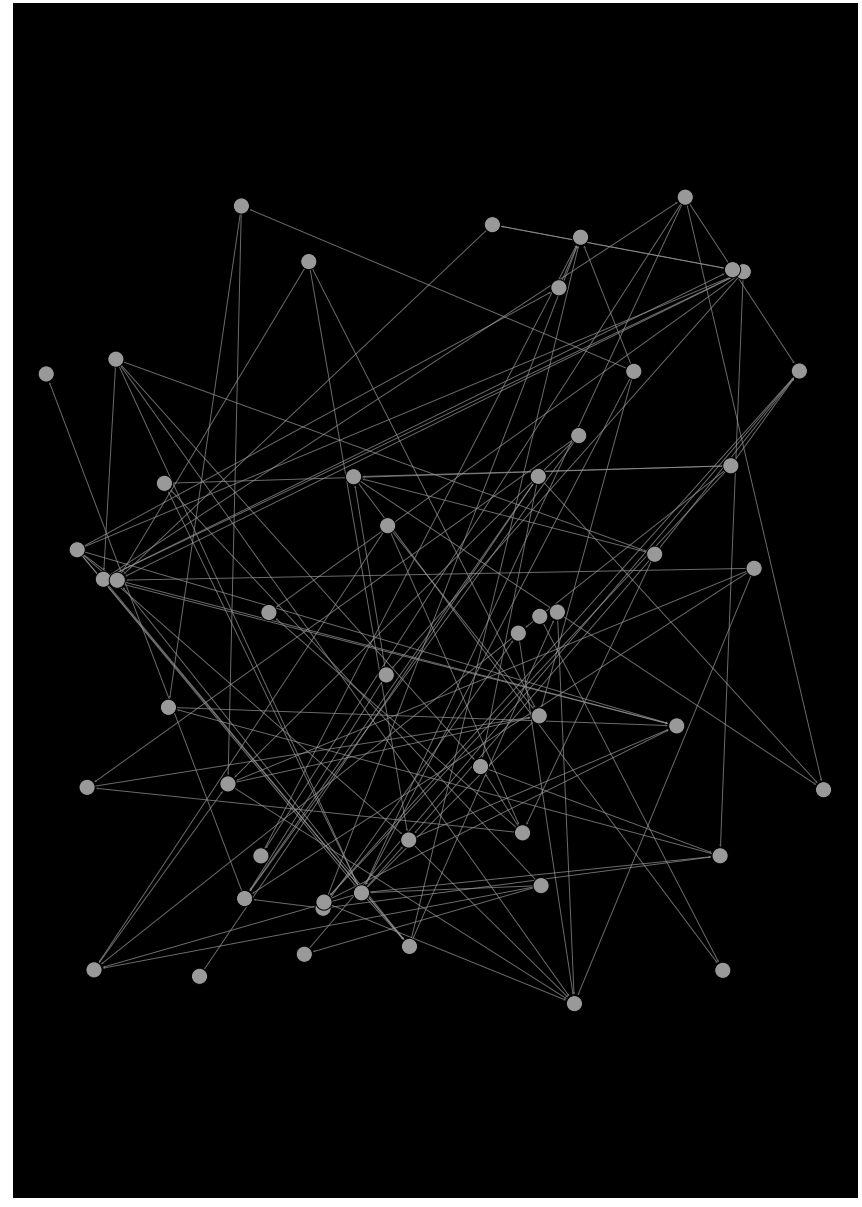
$N = 50, p = 0.02$



$N = 50, p = 0.03$



$N = 50, p = 0.05$

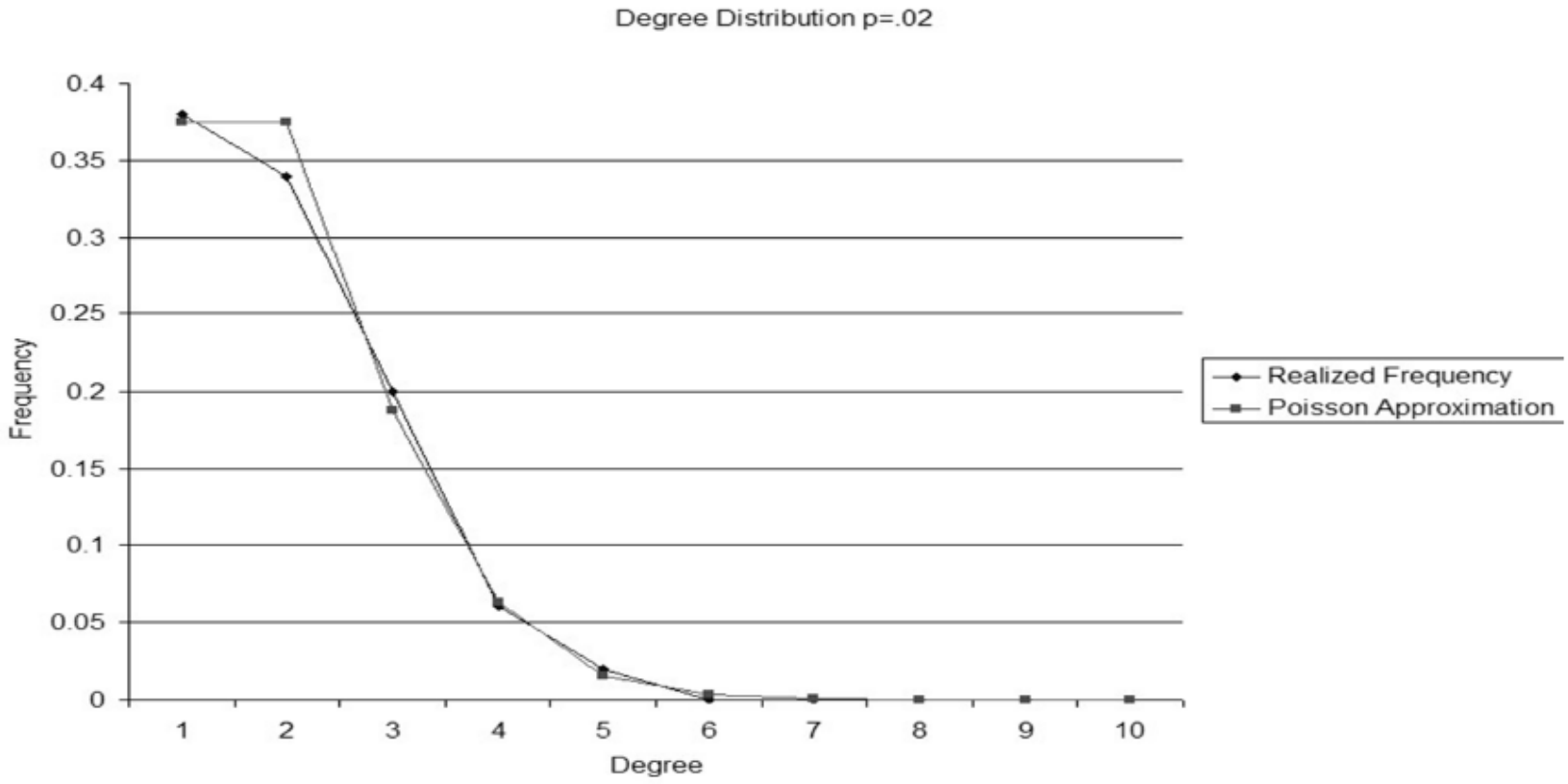


$N = 50, p = 0.08$

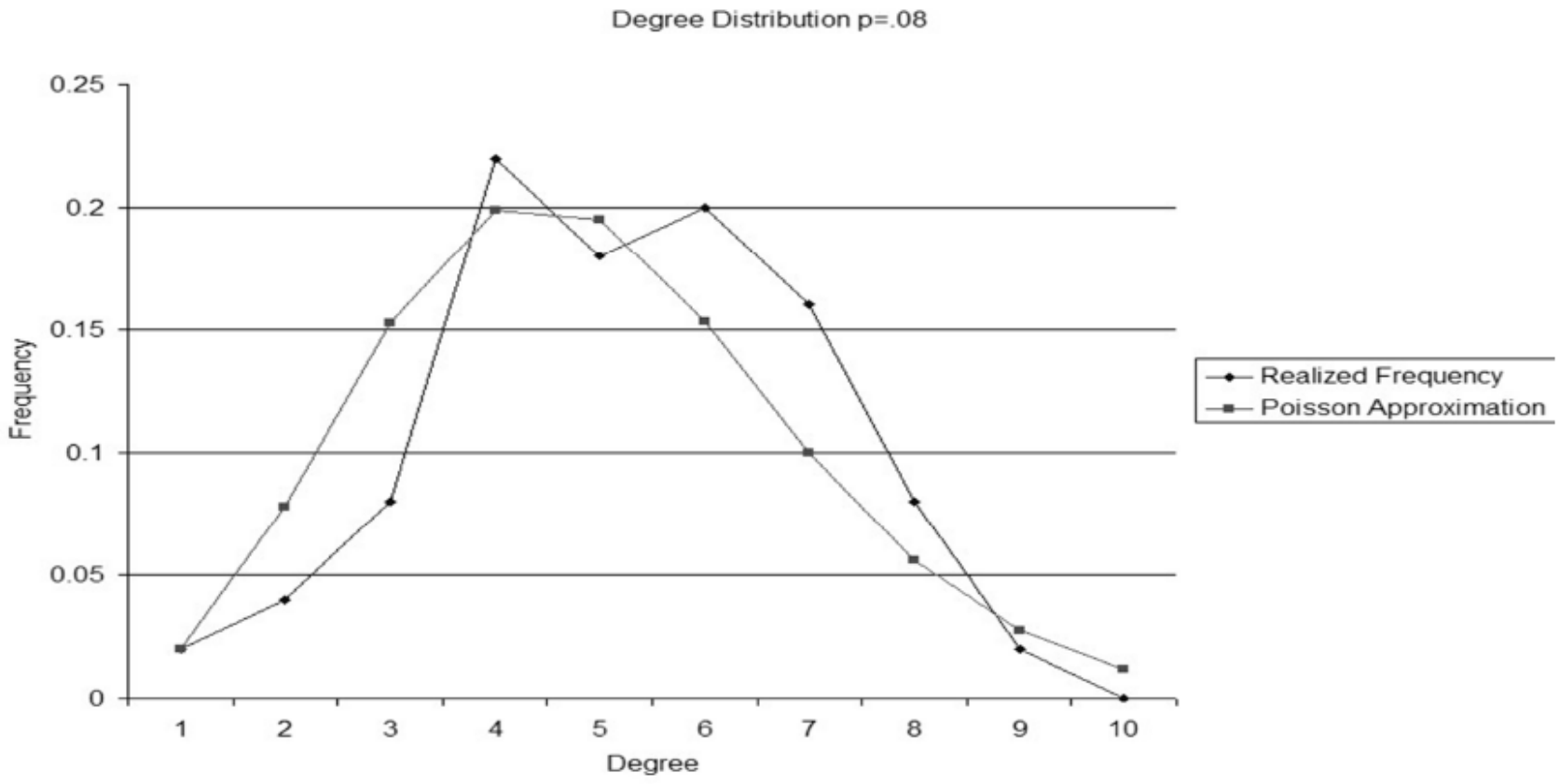
Degree Distributions

- The *degree distribution* of a network, $P(d)$, is the fraction of nodes that have degree d.
- A network is *regular of degree* k if $P(k) = 1$ and $P(d) = 0$ for all $d \neq k$.
- Other prominent degree distributions include a *degenerate degree distribution* associated with a regular network, the *Poisson degree distribution* associated with Poisson random networks, and a *scale-free distribution* (or power distribution).

Degree Distributions



Degree Distributions

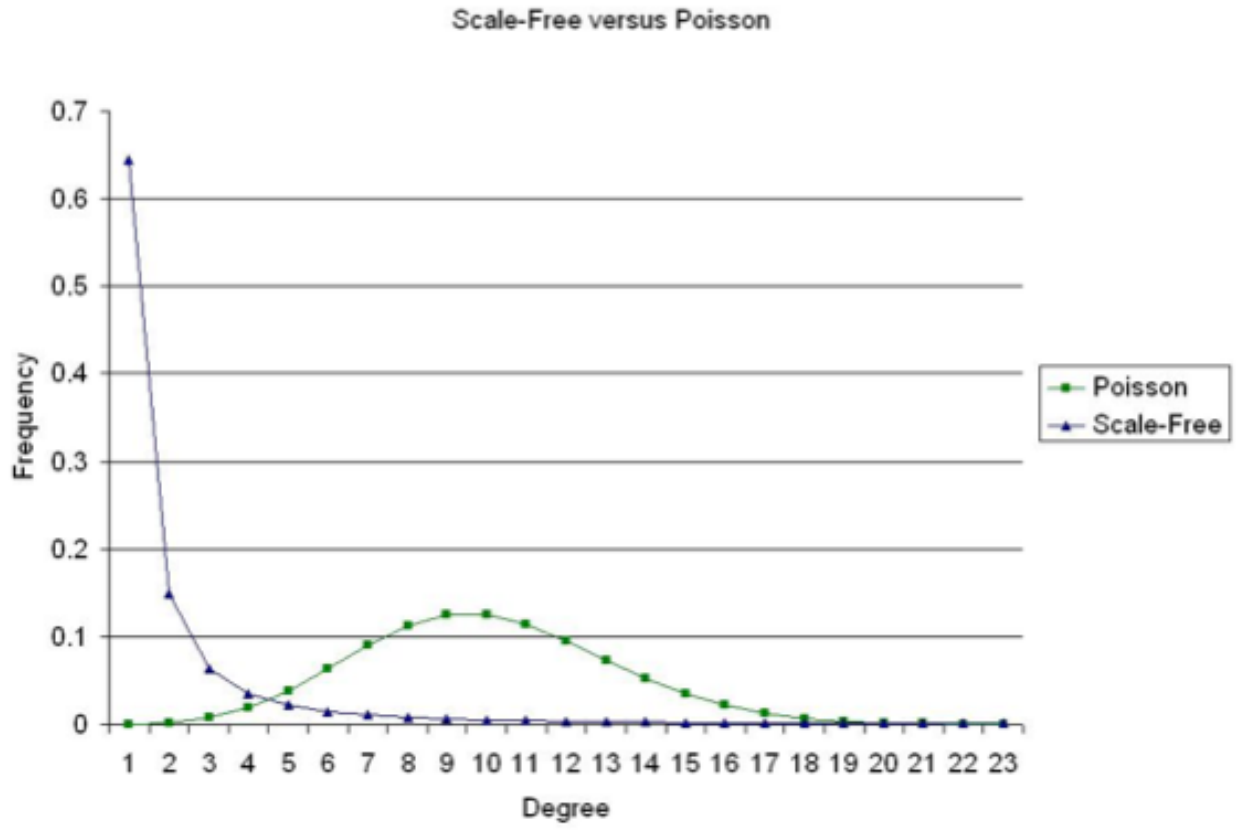


Scale-Free Distributions

- A scale-free distribution $P(d)$ satisfies

$$P(d) = cd^{-\gamma}$$

  where $c > 0$ is a scalar which depends on the support of the distribution.
- If we increase the degree by a factor k, then we end up with a frequency that goes down by a factor of $k^{-\gamma}$. That is, $P(2)/P(1)$ is the same as $P(20)/P(10)$.
- Scale-free distributions are often said to exhibit a *power law*, with reference to the power function $d^{-\gamma}$.



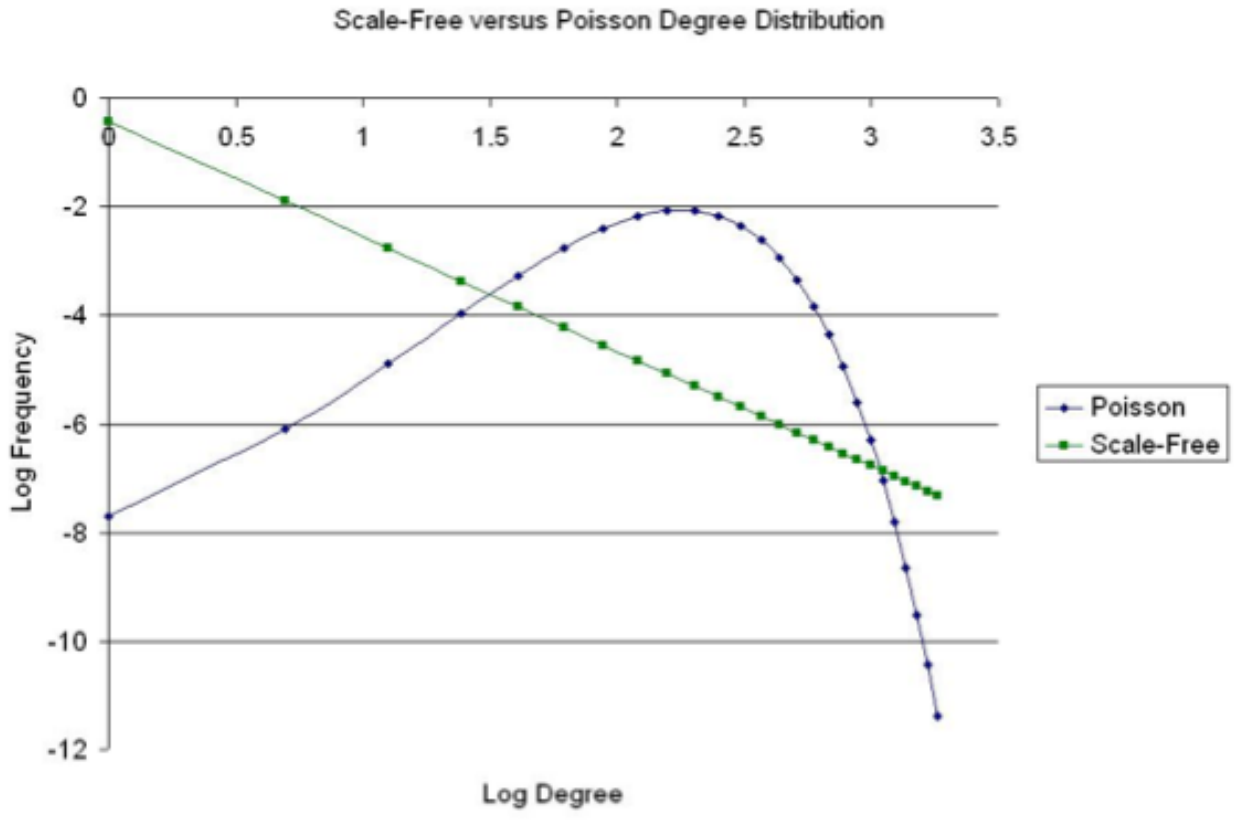
Comparing a Scale-Free Distribution to a Poisson Distribution

Scale-Free Distributions

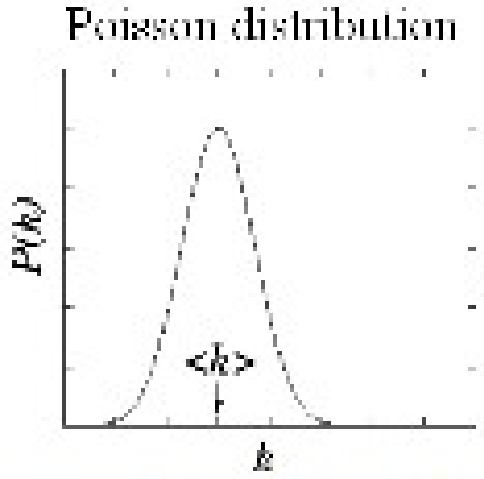
- Scale-free distributions have "fat tails," as there tend to be many more nodes with very small and very large degrees than one would see if the links were formed completely independently.
- Scale-free distributions are linear when plotted on a log-log plot. If we rewrite $P(d) = cd^{-\gamma}$ by taking logs of both sides, we obtain

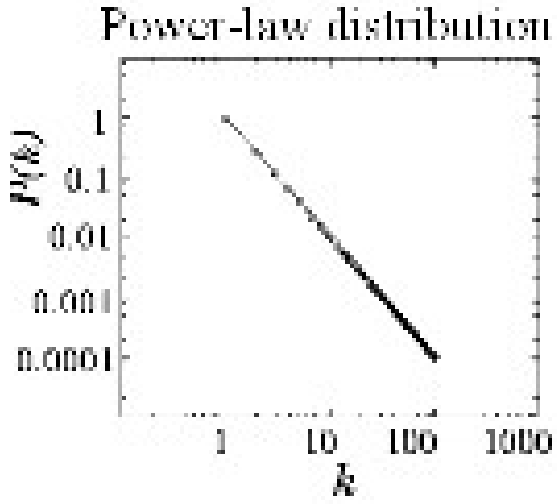
$$log(f(d)) = log(c) - \gamma log(d).$$

  This allows us to estimate γ from data, as a linear regression can be used.

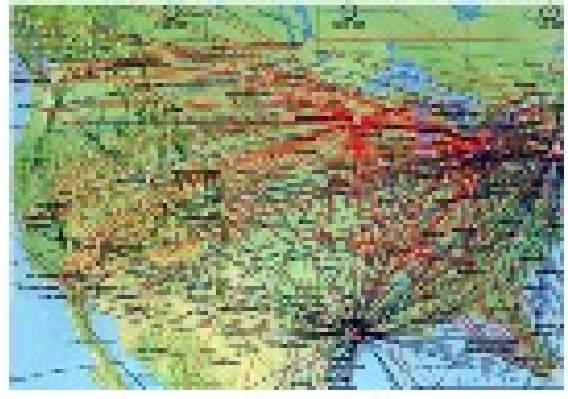


Comparing a Scale-Free Distribution to a Poisson Distribution: LOG-LOG Plot





Exponential Network



Scale-free Network

Comparing a Scale-Free Distribution to an Exponential Network: LOG-LOG Plot

Detour on Web search: PageRank (Ch 14.2 in Easley and Kleinberg)

Influence and Web Search

- When you search for something online what happens?
  1. You ask for a list of pages with information about some term
     - "University of Georgia"
     - "Matrix multiplication"
     - "aquarium plants dying"
  2. your search engine gathers up pages with the corresponding term
  3. the results are displayed to you in some order

  Question: How to order the search results in a useful way?

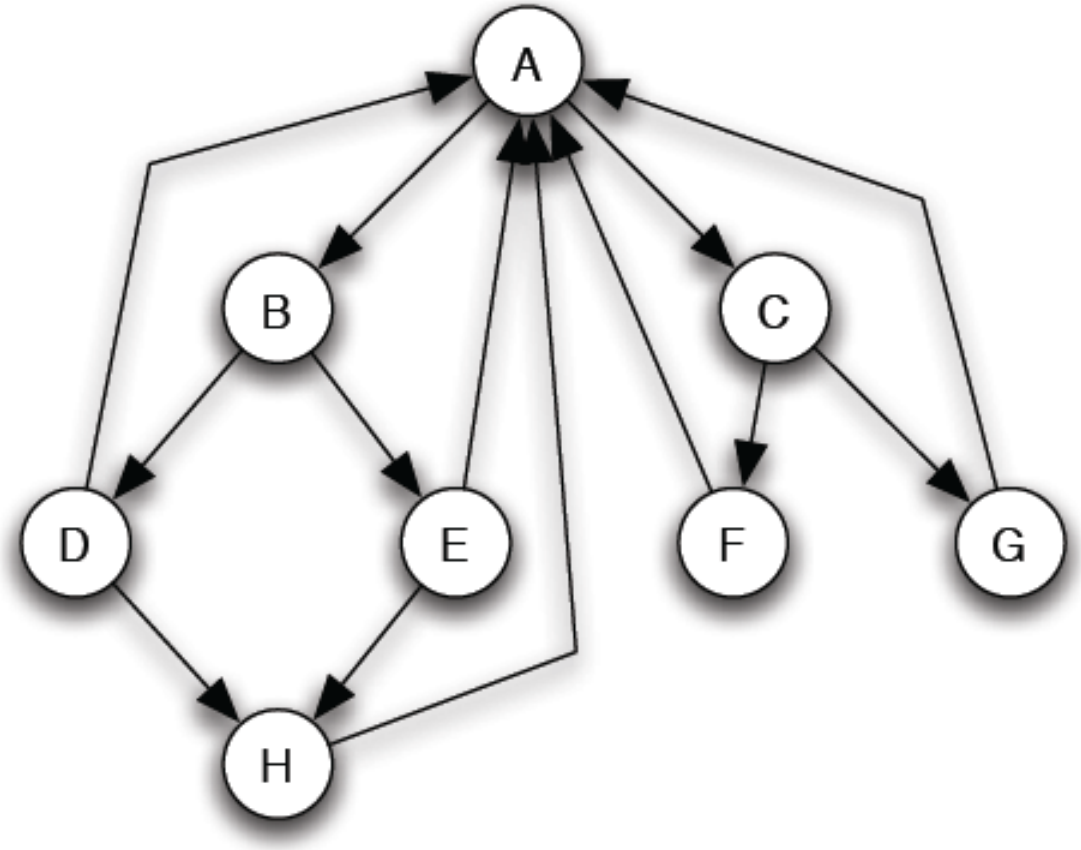
Influence begets influence

- **Idea:** We want to see the most influential pages first
- Why?
  - Important pages are most frequently cited by other pages
  - This is analogous to citation in the scientific literature
  - court cases; patent decisions
- **Problem:** Any page can get lots of links
- **Solution:** Weight links (citations) from highly cited nodes (papers) more heavily
- This idea forms the basis for the *PageRank* measure of node importance/influence.

Influence begets influence

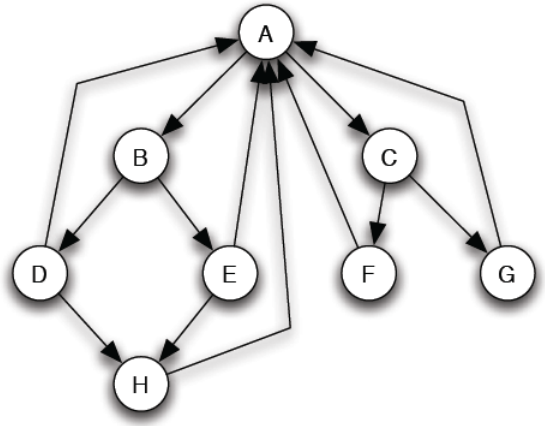
- So the proposal is to define influence (PageRank) in terms of the influence of the pages that link to it.
- The logic seems circular, but can work
- PageRank was the dominant algorithm for ranking pages at Google for many years.
- Ranking algorithms are a moving target
    - manipulation is possible if you know the algorithm
    - big incentives to engage in manipulation

A Network of Webpages



A collection of 8 pages. A has the largest PageRank, followed by B and C, which collect endorsements from A.

Computing PageRank



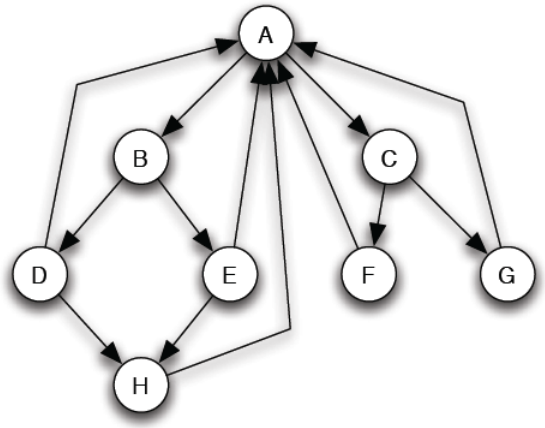
Computing PageRank

1. Assign all nodes, i, PageRank $P_i = \frac{1}{n}$.
2. Choose number of steps k
3. Perform k updates, where each update does the following:
   - Each page "sends" an equal share of its existing PageRank to the pages it points to.
   - If a page points to no other pages, it keeps its PageRank for itself.
   - The PageRank of each page is recomputed as the sum of all the PageRank it just received over its incoming links.

Remarks

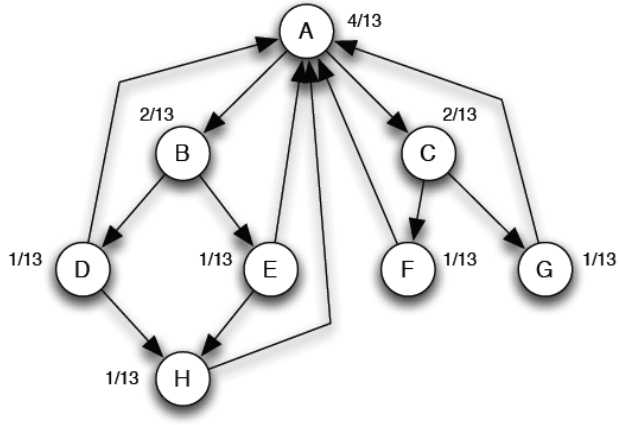
- Network starts with one unit of PageRank
- PageRank is "conserved" as it "flows" through the update steps.
- Intuitively, equilibrium is achieved when the flow in equals flow out for every node.

Computing PageRank: Example



Step	A	B	C	D	E	F	G	H
1	1/2	1/16	1/16	1/16	1/16	1/16	1/16	1/8
2	3/16	1/4	1/4	1/32	1/32	1/32	1/32	1/16

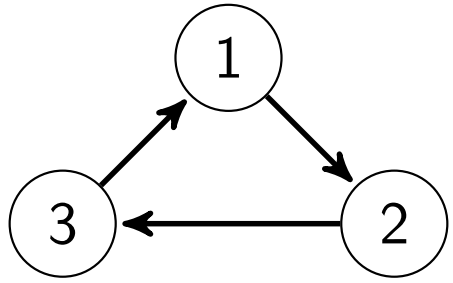
Equilibrium PageRank



- Equilibrium concept
  - If we start the nodes at the equilibium
  - and apply one update step
  - they end with the same PageRank they started with

  Claim: There is always an equilibrium distribution of PageRank (except in certain degenerate cases.)

QUIZ



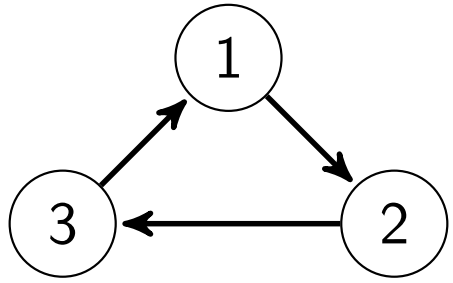
Is the following an equilibrium allocation of PageRank?

node	PageRank
1	1/2
2	1/2
3	0

- NO.
- In the next update we get

node	PageRank
1	0
2	1/2
3	1/2

QUIZ



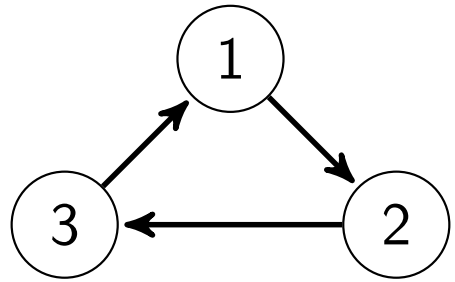
Is the following an equilibrium allocation of PageRank?

node	PageRank
1	1/2
2	1/2
3	0

- NO.
- In the next update we get

node	PageRank
1	0
2	1/2
3	1/2

QUIZ



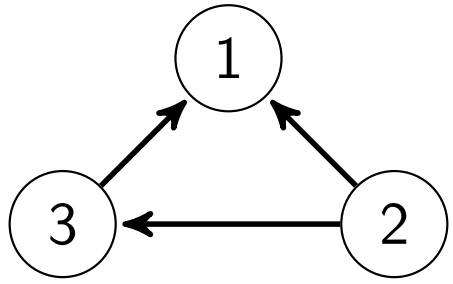
Is the following an equilibrium allocation of PageRank?

node	PageRank
1	1/2
2	1/2
3	0

- NO.
- In the next update we get

node	PageRank
1	0
2	1/2
3	1/2

Sinks and Degenerate Equilibria



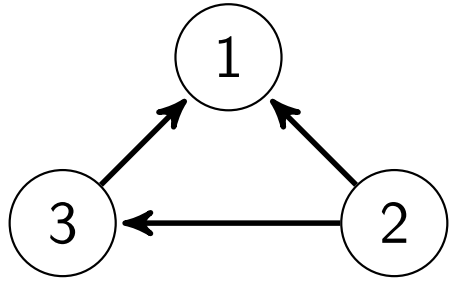
QUESTION: What is the unique equilibrium distribution of PageRank?

- **Answer:**

node	PageRank
1	1
2	0
3	0

- All of the PageRank "pools" at Node 1.

Sinks and Degenerate Equilibria



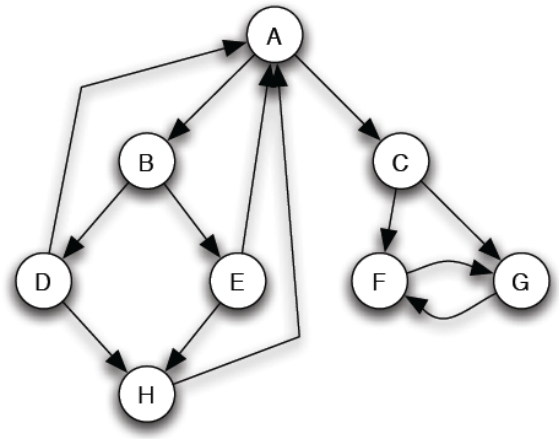
QUESTION: What is the unique equilibrium distribution of PageRank?

- **Answer:**

node	PageRank
1	1
2	0
3	0

- All of the PageRank "pools" at Node 1.

Uniqueness of Equilibrium



- You can prove the following:
- If the network is *strongly connected*, then the equilibrium PageRank distribution
  - exists,
  - and is unique.
- (Recall: *Strongly connected* means there is a directed path between from any node to any other node.)
- As the graph above indicates, strong connectivity is easy to violate

Scaled PageRank Update

The solution? **Make it rain!**
Specifically, modify the update step as follows

- After the update, take a fraction, α, of PageRank from every node
- Reallocate the α PageRank uniformly to all nodes.

This is called the *scaled PageRank Update*

Equilibrium in Scaled PageRank

It can be proven that...

- If you repeat the scaled PageRank update k times, as k goes to infinity
- The distribution of PageRank values
  - Will converge
  - the set of values it converges to will be unique
- Note that the equilibrium depends on the scaling factor, α.

Random Web Surfer

- Alternative derivation / interpretation: A Random Surfer
  - Imagine someone surfing the web, starting from a random page
  - She takes a random walk that follows k links
  - **Claim:** The probability of being at page i after k steps is the PageRank of i after k applications of the basic PageRank update.
  - Proof in Section 14.6 (advanced).

Page ranking in real life

- Google, Ask, Bing, etc. now use highly complex search and reporting algorithms
- Changes to these algorithms have big costs for companies
  - Moving from the first to second page of search results is costly
  - Moving from second to first result has a big payoff
- *Search Engine Optimization (SEO)* is a major industry.
- The "perfect" ranking is always a moving target, because the structure of the web evolves as pages (nodes) game the system.

Eigenvector Centrality

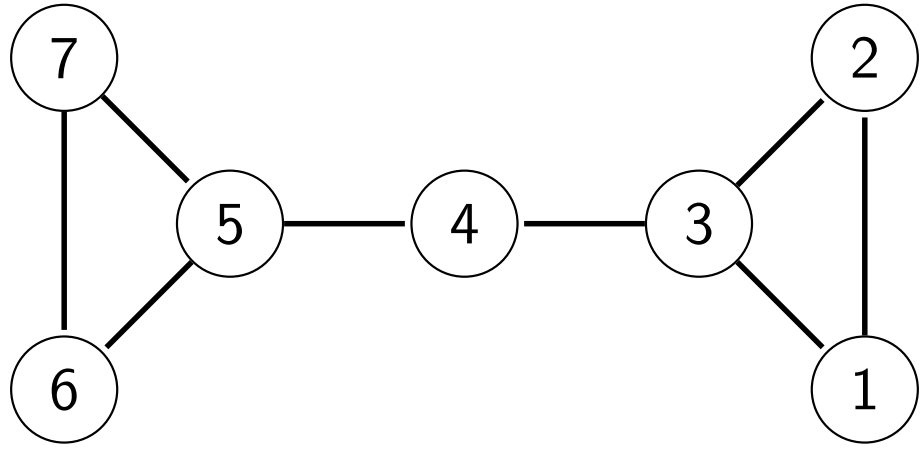
- The *eigenvector centrality* of a node is proportional to the sum of the centrality of its neighbors:

$$\lambda C_i^e(g) = \Sigma_j g_{ij} C_j^e(g).$$

- In terms of matrix notation,

$$\lambda C^e(g) = g C^e(g),$$

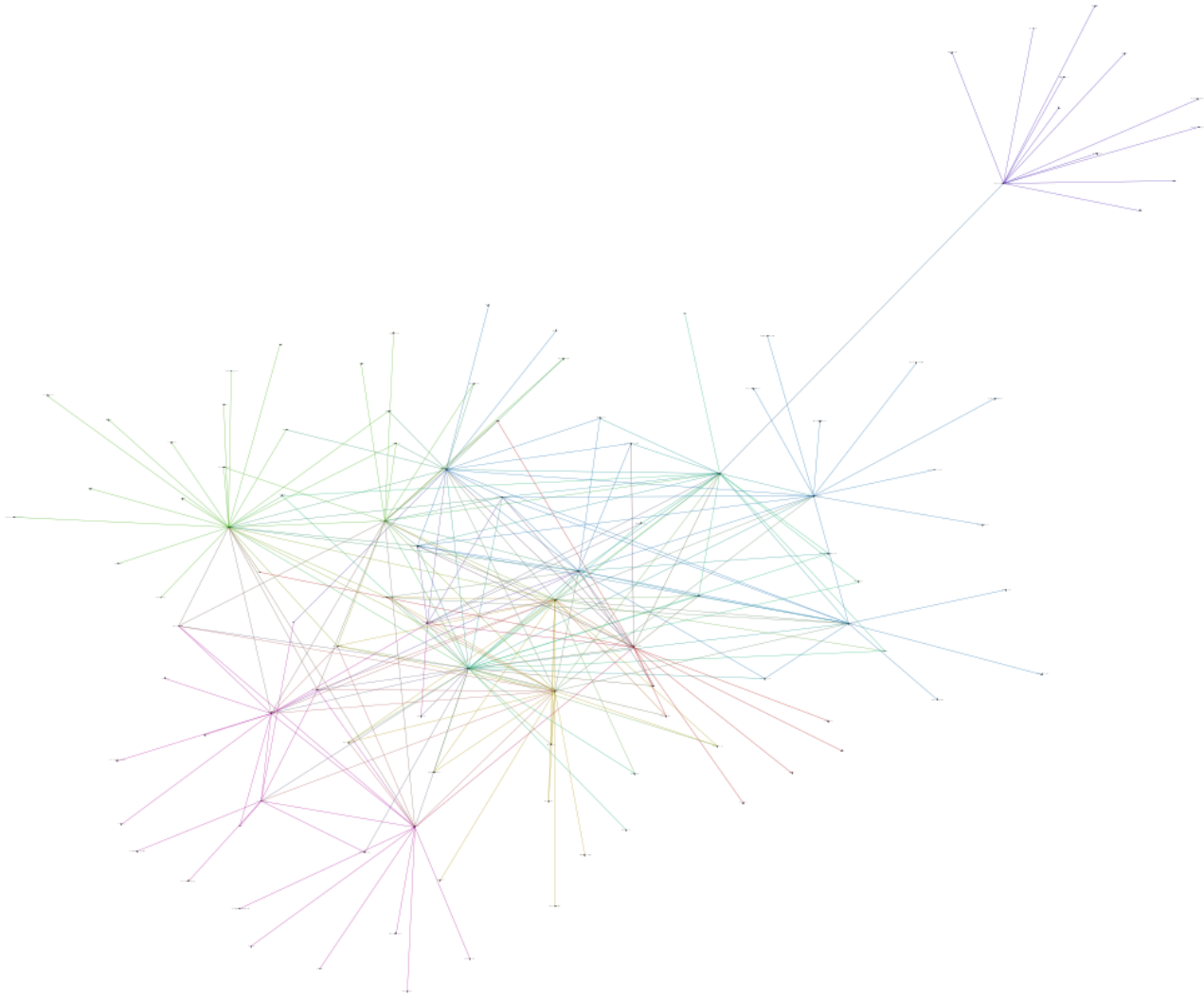
  where λ is a proportionality factor. Thus, $C^e(g)$ is an eigenvector of g, and λ is its corresponding eigenvalue.



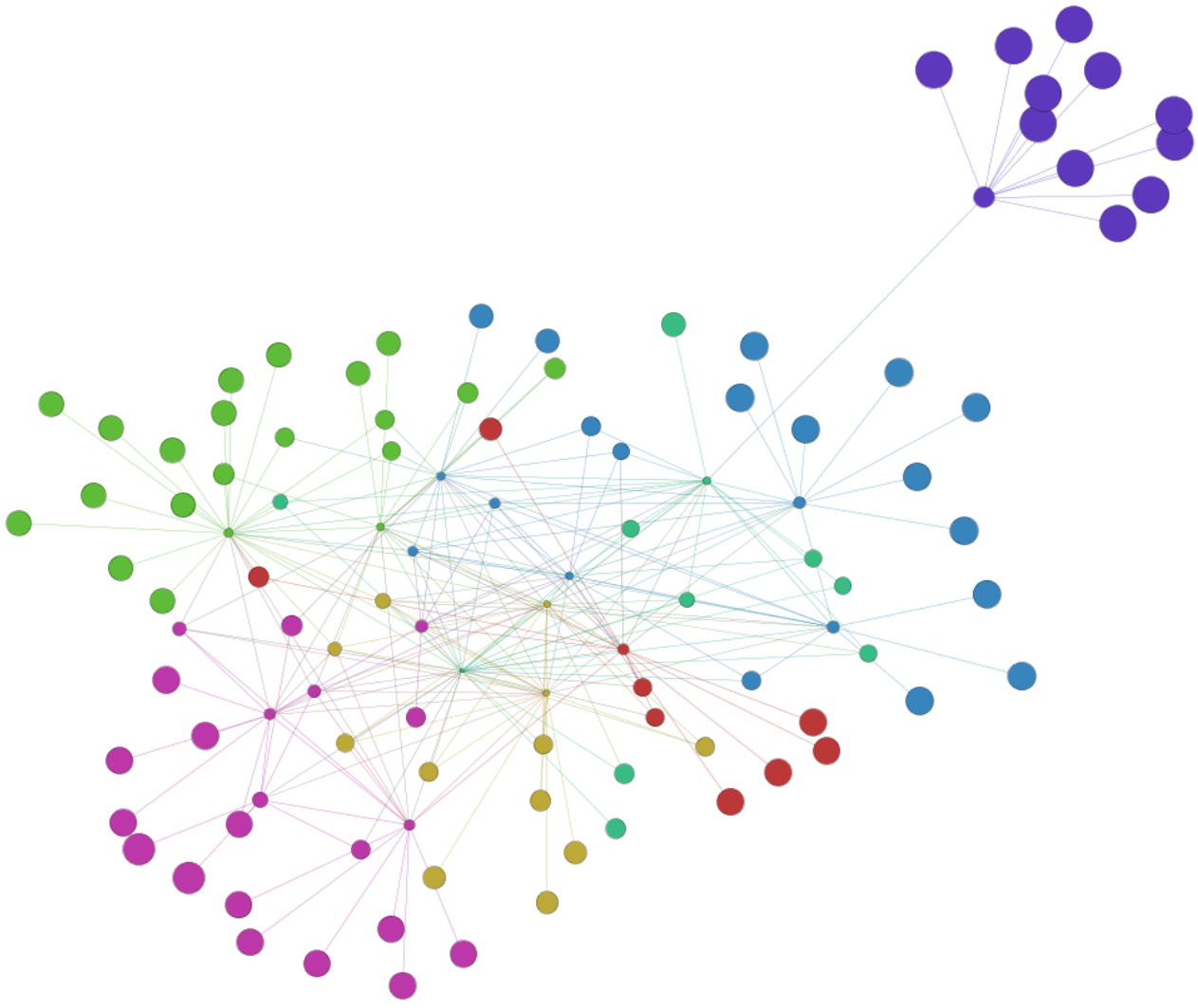
	Nodes 1,2,6,7	Nodes 3 and 5	Node 4
Degree (and Katz Prestige P^K)	.33	.50	.33
Closeness	.40	.55	.60
Decay Centrality ($\delta = .5$)	1.5	2.0	2.0
Decay Centrality ($\delta = .75$)	3.1	3.7	3.8
Decay Centrality ($\delta = .25$)	.59	.84	.75
Betweenness	0.0	.53	.60
Eigenvector Centrality	.47	.63	.54
Katz Prestige-2 P^{K2}, $a = 1/3$	3.1	4.3	3.5
Bonacich Centrality $b = 1/3$, $a = 1$	9.4	13	11
Bonacich Centrality $b = 1/4$, $a = 1$	4.9	6.8	5.4

Centrality Comparisons of Previous Figure

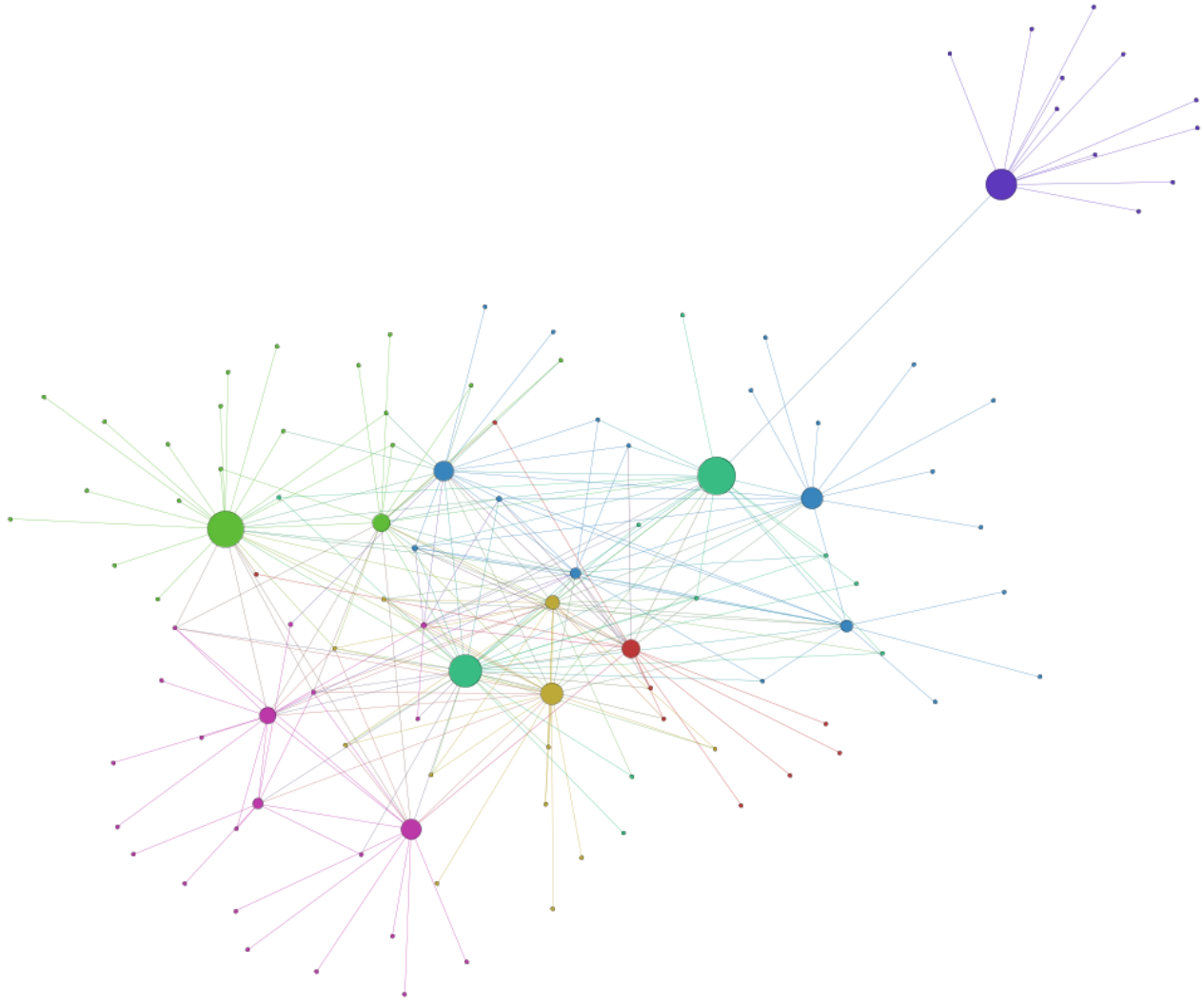
Comparison of Centrality Measures



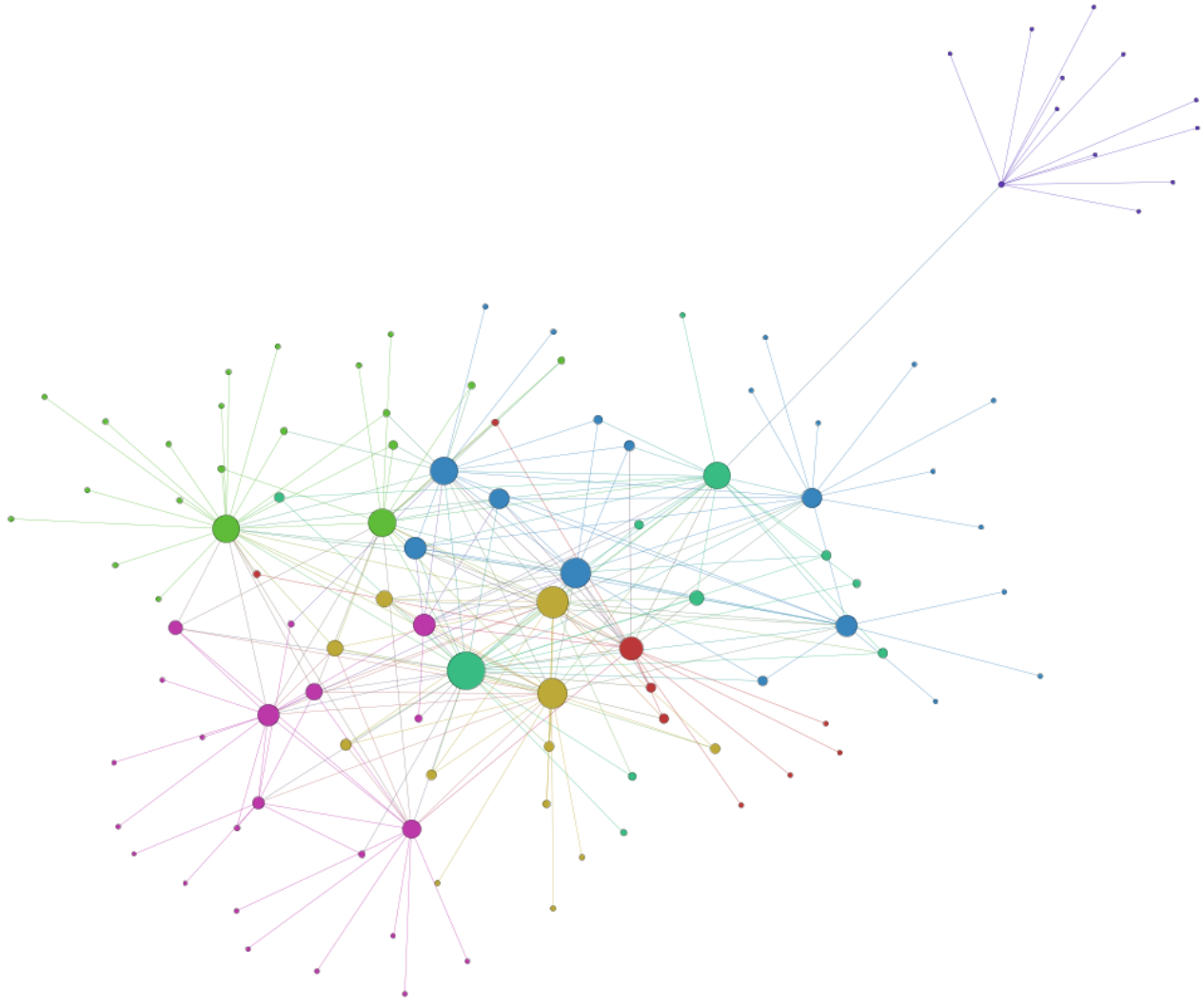
Closeness Centrality



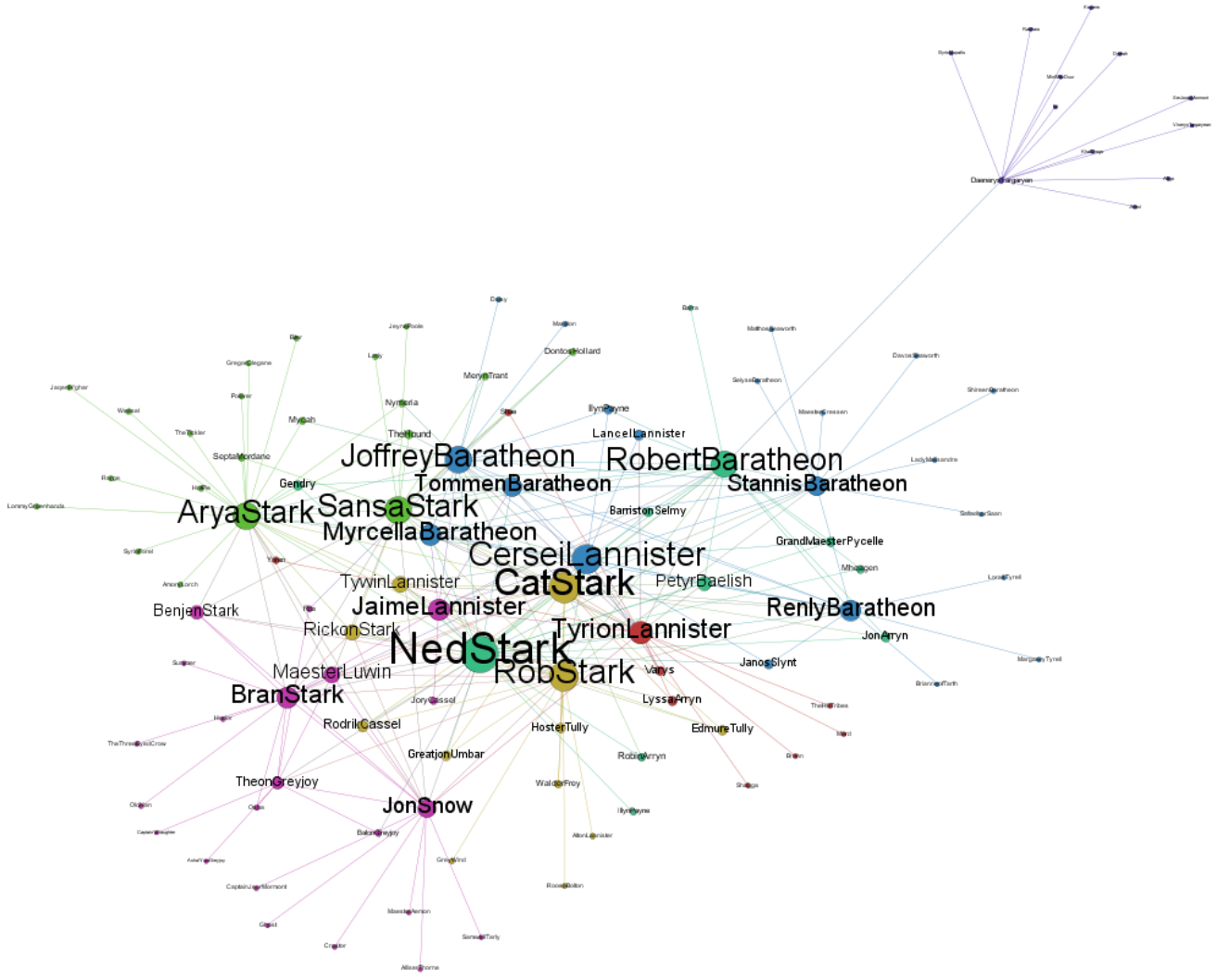
Betweeness Centrality



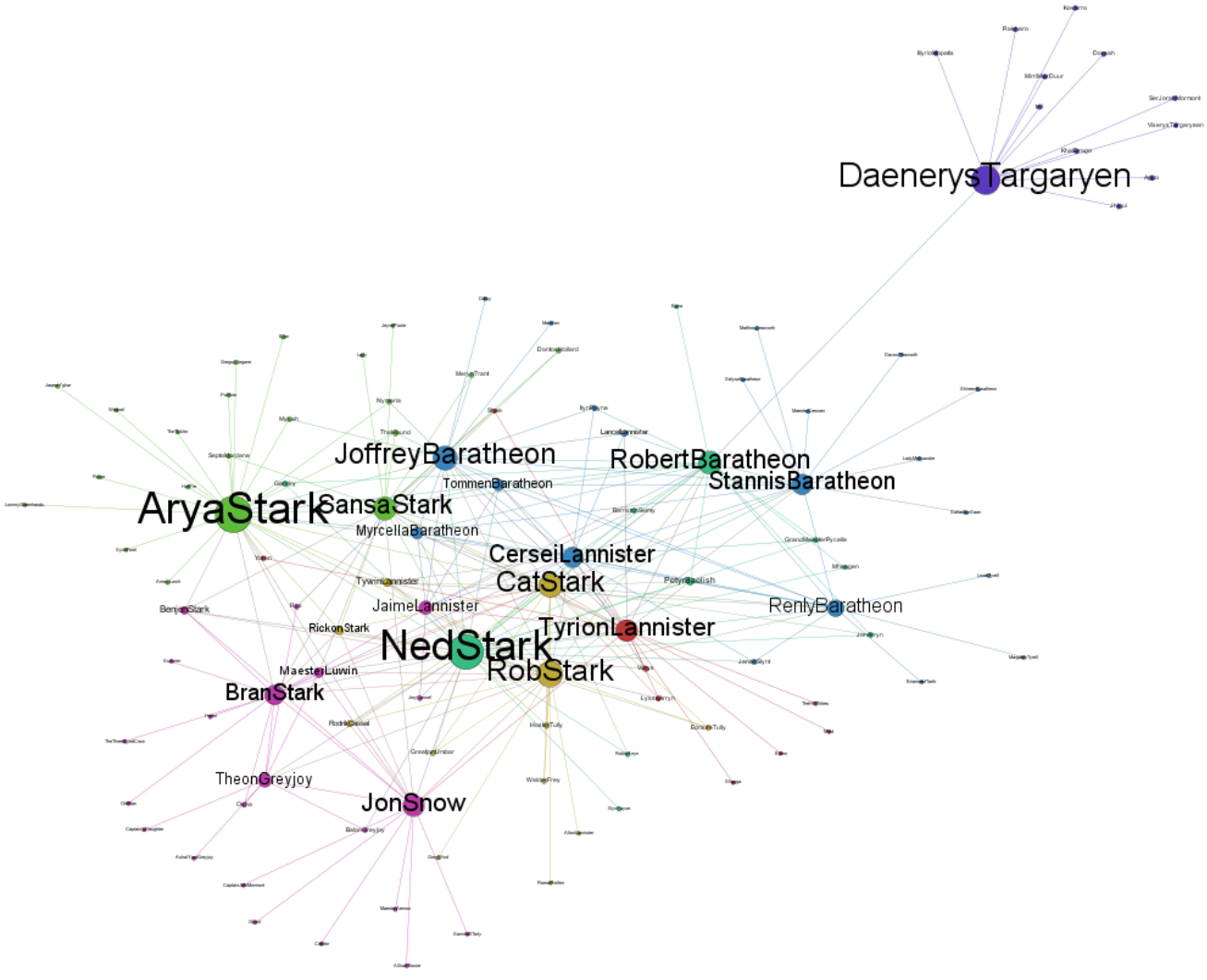
Eigenvector Centrality



The Graph of Thrones!



PageRank


DaenerysTargaryen
JoffreyBaratheon
RobertBaratheon
StannisBaratheon
TommenBaratheon
AryaStark
SansaStark
MyrcellaBaratheon
CerseiLannister
CatStark
TywinLannister
JaimeLannister
RenlyBaratheon
BenjenStark
RickonStark
TyrionLannister
NedStark
RobStark
MaesterLuwin
BranStark
TheonGreyjoy
JonSnow