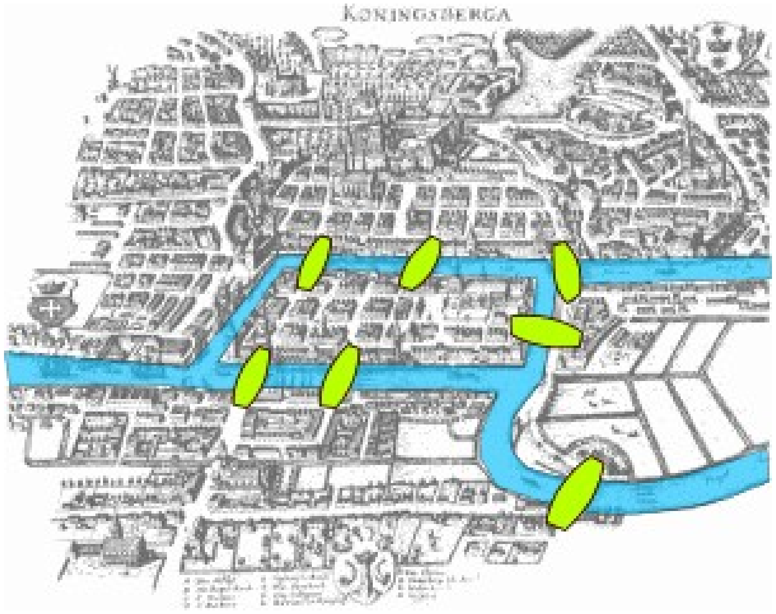
Networks – Some Graph Theory and Network Measures

Outline

- Directed and undirected Nws
- Degrees, components, paths
- Connectivity & cut sets, the graph Laplacian
- Centrality (Eigenvector, Katz, Pagerank, Closeness, Betweenness)
- Clustering, reciprocity, motifs
- Homophily
- Distances

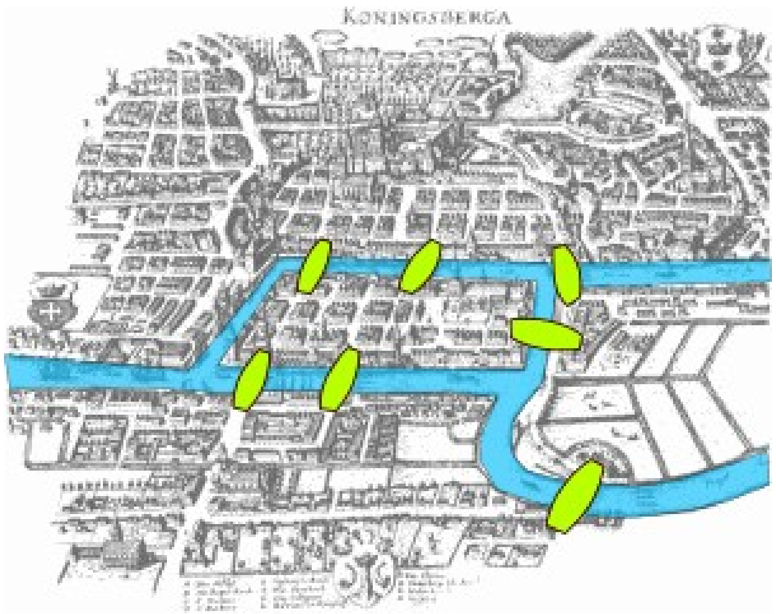
Konigsberg Bridge Problem

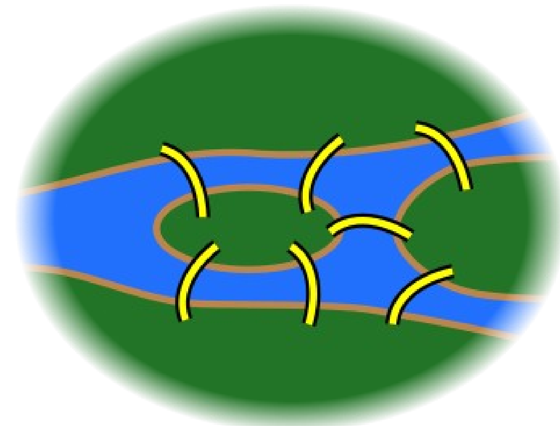
- Graph Theory goes back to Euler (1736) and the Konigsberg bridge problem

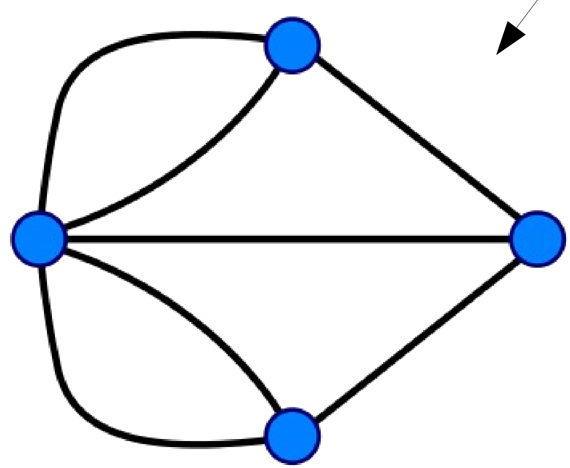


- Find a walk through city that crosses each bridge once and only once

Konigsberg Bridge Problem (2)

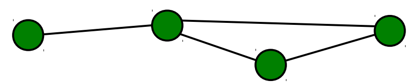


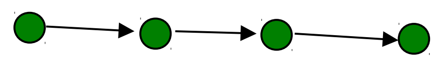




Some Formalities

- Graph G = (V,E) V … set of vertices, E … set of edges

undirected graph: 

directed graph: 

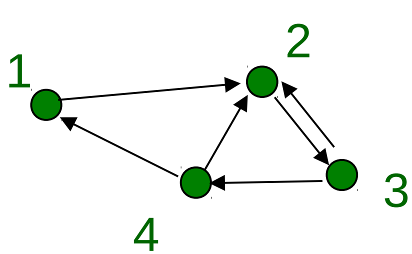
path from u to v P(u,v): sequence of edges (u,a), (a,b),…,(x,v)

Eulerian path: path that visits every edge exactly once

cycle: path that contains at least one vertex twice

length of a path: number of edges traversed along it

adjacency matrix $A = (a_{ij})$

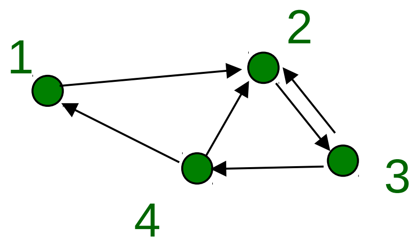


$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$(A^n)_{kl}$ … #paths of length n from k to l

How to “Store a Network”?

- Adjacency matrices (as in previous slide)
- Adjacency lists
  - In-lists
    - 1: 4; 2: 1,3,4; 3: 2; 4: 3
  - Out-lists:
    - 1: 2; 2:3; 3: 2,4; 4: 1,2
  - Both?



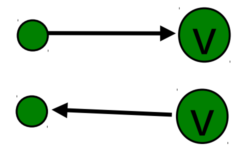
- Adjacency matrix vs adjacency lists
  - Memory: expensive efficient
  - Direct access: fast slow
  - Neighbour access: slow fast

Graph Theory

- Degree d(v) of a vertex v = #neighbours

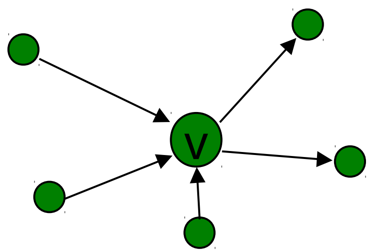
  In – degree $d_i(v)$ = #in-links

  Out-degree $d_o(v)$ = #out-links



(in-link)

(out-link)



$d(v) = 5$, $d_i(v) = 3$, $d_o(v) = 2$

Isolated vertex … vertex of degree 0, leaf … vertex of degree 1

Degree distribution: Pr {d(v) = n}

- Connected component (cluster) of v in V … maximal set of vertices v’ of V for which there is a path P(v,v’)

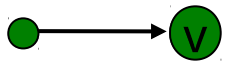
  In-cluster of v … for which there is an in-path $P_{in}(v,v')$

  Out-cluster of v …for which there is an out-path $P_{out}(v,v')$

Algorithms: e.g. breadth first search

Graph Theory

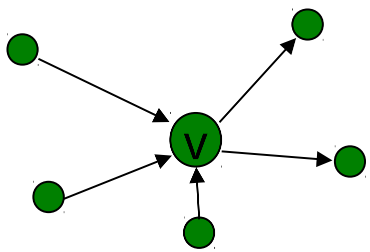
- Degree $d(v)$ of a vertex v = #neighbours

  In – degree $d_i(v)$ = #in-links 

(in-link)

  Out-degree $d_o(v)$ = #out-links 

(out-link)



$d(v) = 5$, $d_i(v) = 3$, $d_o(v) = 2$

  Isolated vertex … vertex of degree 0, leaf … vertex of degree 1

  Degree distribution: Pr $\{d(v) = n\}$

- Connected component (cluster) of v in V … maximal set of vertices v' of V for which there is a path P(v,v')

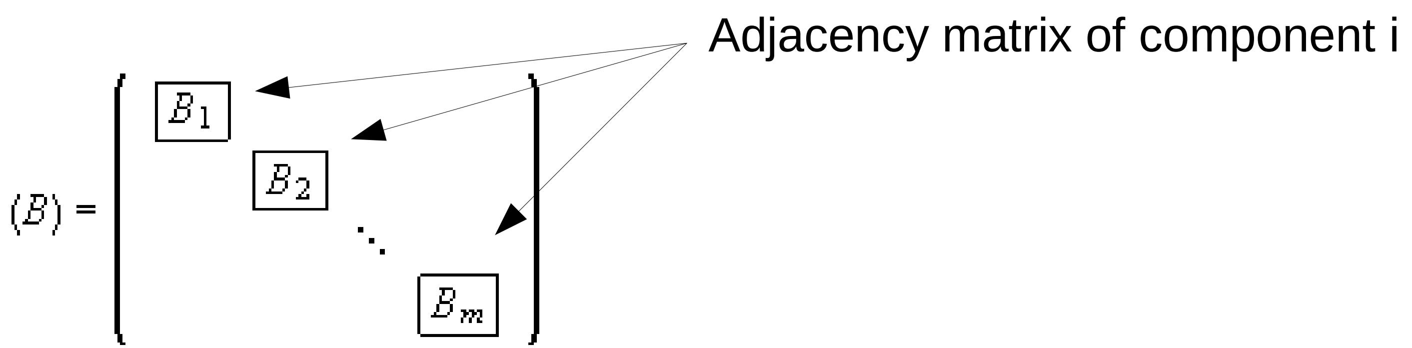
  In-cluster of v … for which there is an in-path $P_{in}(v,v')$

  Out-cluster of v …for which there is an out-path $P_{out}(v,v')$

  Strong components: maximal sets of vertices that can reach each other

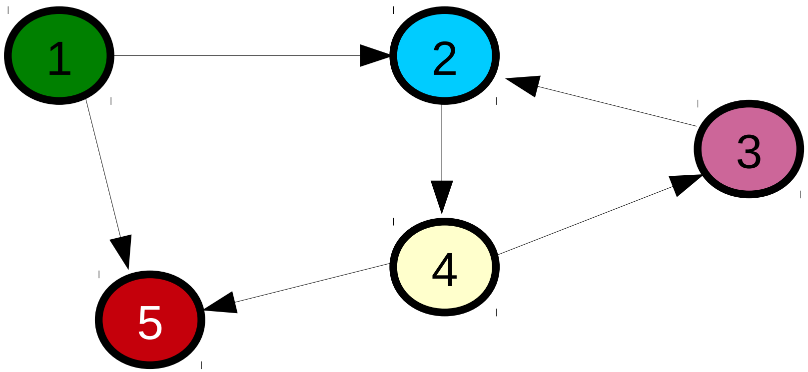
Components in Undirected NWs

- Show up in block diagonal form of the adjacency matrix (if indices are ordered correctly)



- Could easily be determined by, e.g., breadth first search

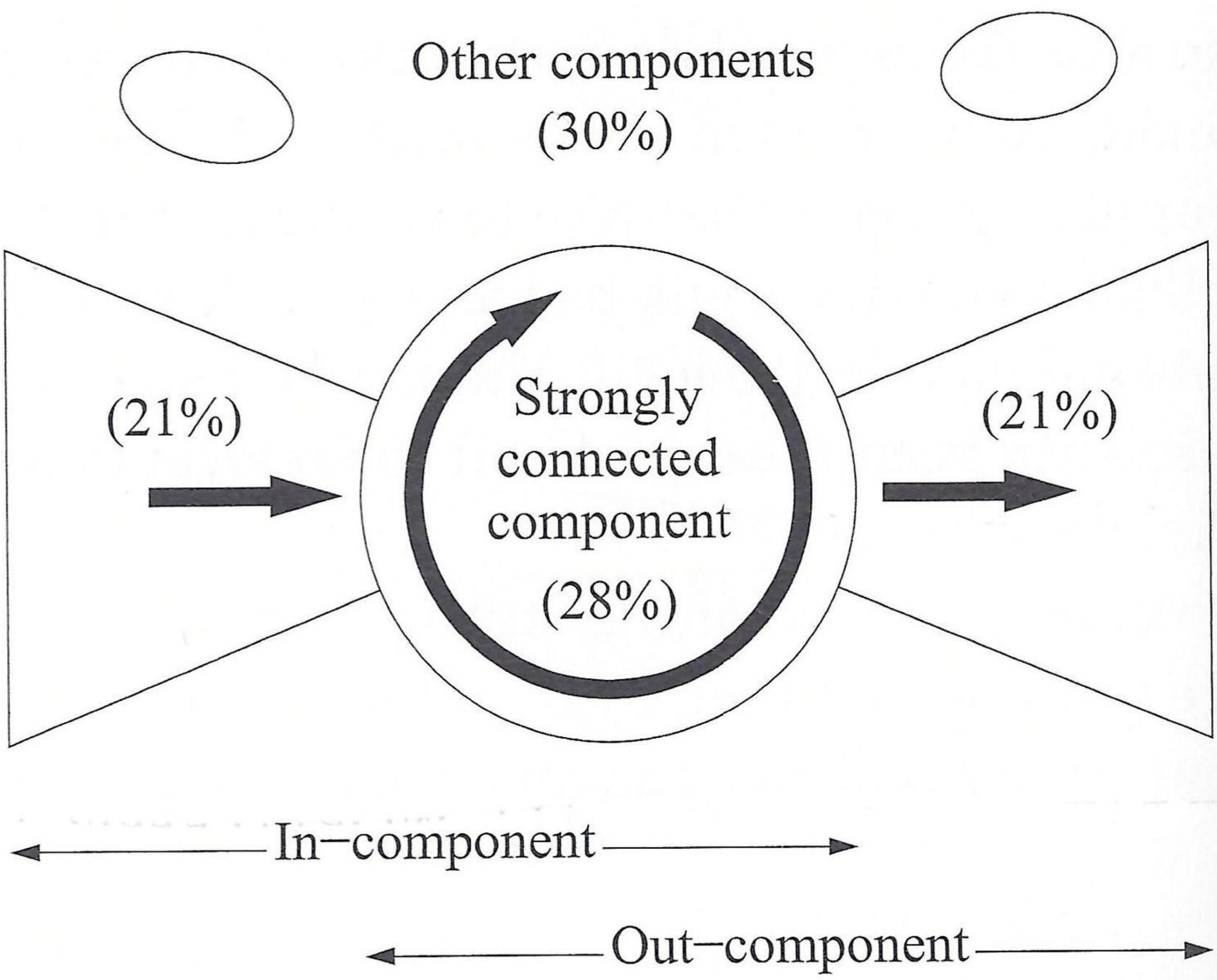
Some Examples



- In component of 5: 1 2 3 4
- Out-component of 2: 3 4 5
- Strong components? 2 3 4
- Path lengths

1 to 3 3 3 to 1 infinite

Components in Directed NWs



The Graph Laplacian

- Say every node is associated with some amount of substance ϕ
- The substance would diffuse on the network

$$\frac{d\phi_i}{dt} = C\sum_{ij} a_{ij}(\phi_j - \phi_i) = C\sum_{ij}(a_{ij} - k_i\delta_{ij})\phi_j = C\underbrace{(A-D)}_{-L}\phi$$

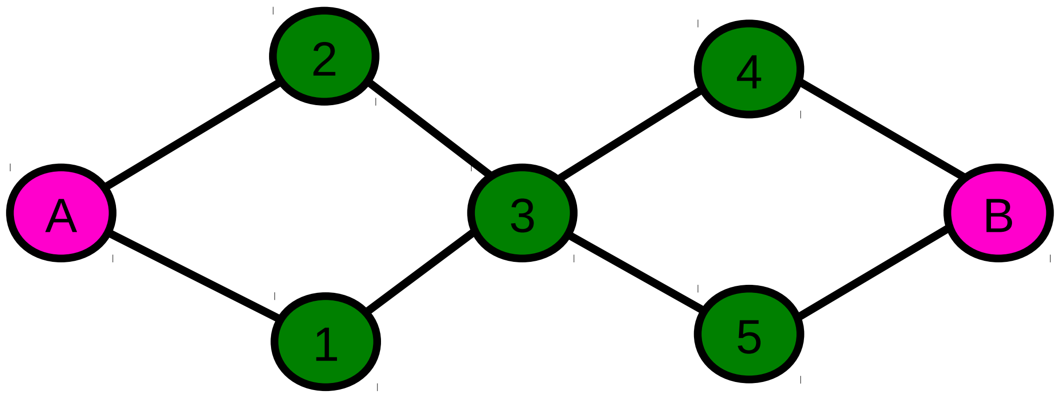
$$\longrightarrow \frac{d\phi_i}{dt} + C \underbrace{L}_{\text{this is usually the Laplace operator } \nabla^2} \phi = 0$$

- Eigenvalues:
  - Always positive, at least one zero eigenvalue (because 1 is eigenvector)
  - Number of zero eigenvalues = number of components
  - First non-zero EV is “algebraic connectivity”

Types of Paths

- Geodesic paths – shortest paths
- (node/edge) Independent paths
  - Paths that share no node/edge apart from start and end
  - Number of independent paths between i and j
    - Node/edge connectivity of i and j
    - measures “how strongly” two nodes are connected
    - Can you find an example of a graph with two nodes that have vertex connectivity 1 and edge connectivity 2?
- Cut set:
  - set of nodes whose removal will disconnect a pair of nodes
  - Minimum ~ : smallest cut set for a pair of nodes

Cut sets – Examples

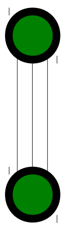


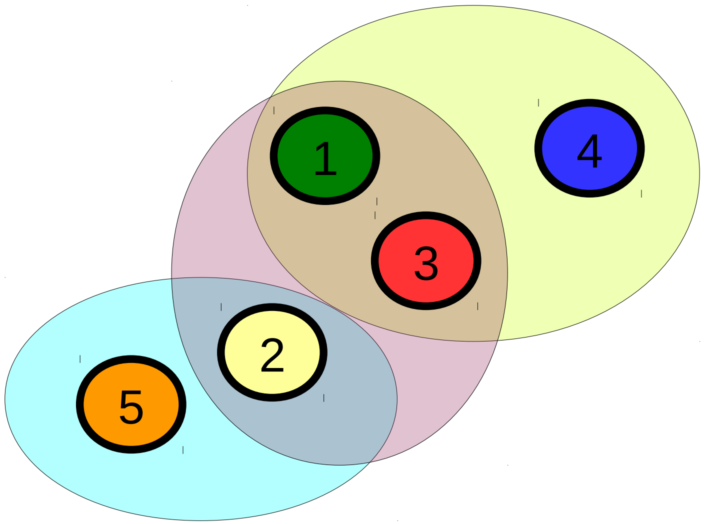
- Edge/vertex connectivity between A and B?
- Vertex cut sets between A and B?
- Minimum vertex cut set A,B?
- Minimum edge cut set A,B?

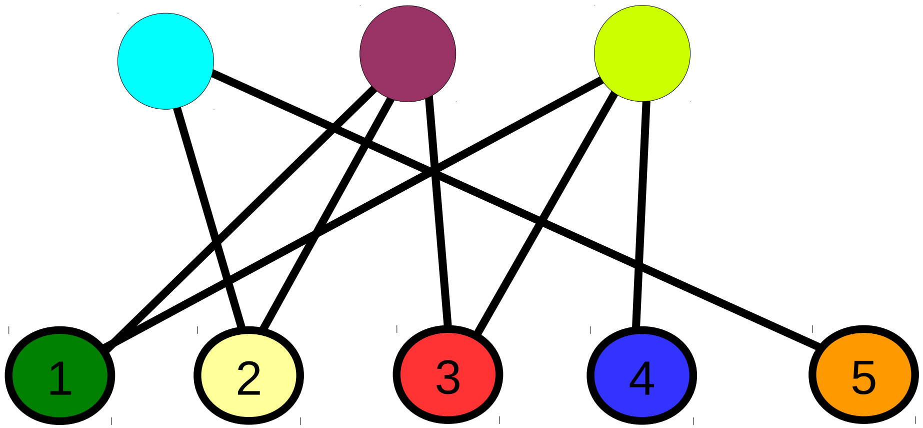
→ Menger: if there is no cut set of size less than n between two given vertices, then there are at least n independent paths between them

Some Special Types of Graphs

- Directed/undirected
- Binary/weighted – graphs with multi-edges?
- Hypergraphs – links can join more than two nodes
- Bipartite networks – two classes of nodes without intra-class links

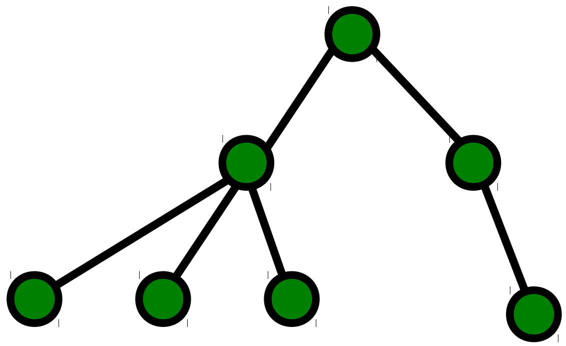






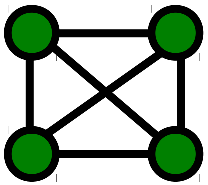
Some Special Types of Graphs (2)

- Trees – networks with no cycle



- Planar networks
  - Can be drawn in a plane without crossing edges

    Is this graph planar?



  - Kuratowski's theorem to determine planarity
- Spatial networks
  - Every node is assigned a location in space

Network Metrics

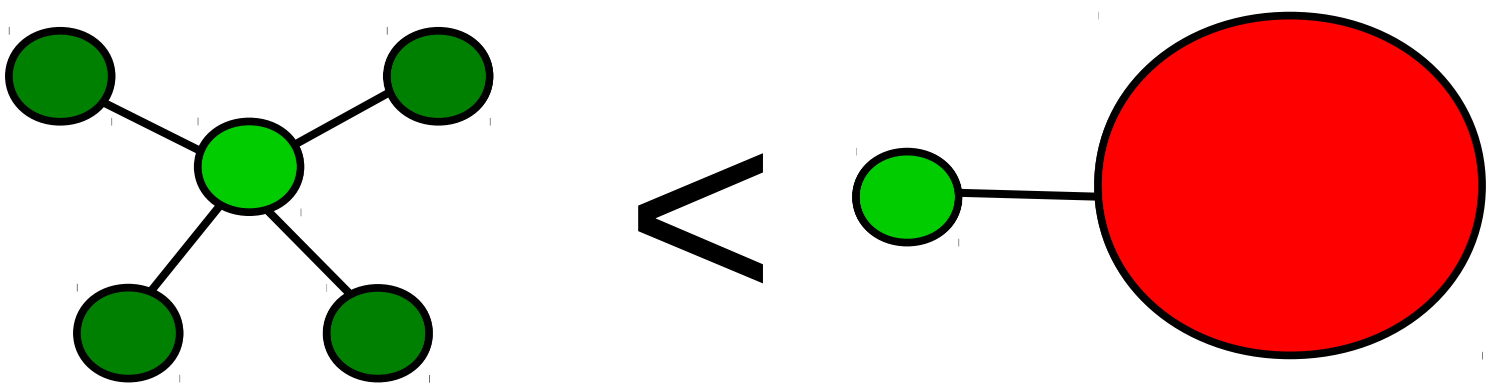
- How would we "measure" a network?
  - Number of nodes/density of connections
    - Sparse vs. dense?
  - Number of components?
  - Importance of nodes (centrality)?
  - "Distances" – Average pathlengths/diameters – large vs. small?
  - Heterogeneity
    - Degree distributions? Narrow vs. broad?
  - Mixing
    - Assortative vs. disortative
  - Local structure
    - Local coherence (clustering), motifs ...

Centrality

- Which are the most important nodes in the network?
- Simplest idea: degree centrality
  - Nodes with the most connections are the most important ones …
  - This is actually used quite prominently in citation networks (in-degree = #citations of a paper)

Eigenvector Centrality

- Score "centrality points" for being connected to "important" nodes (Bonacich 1987)

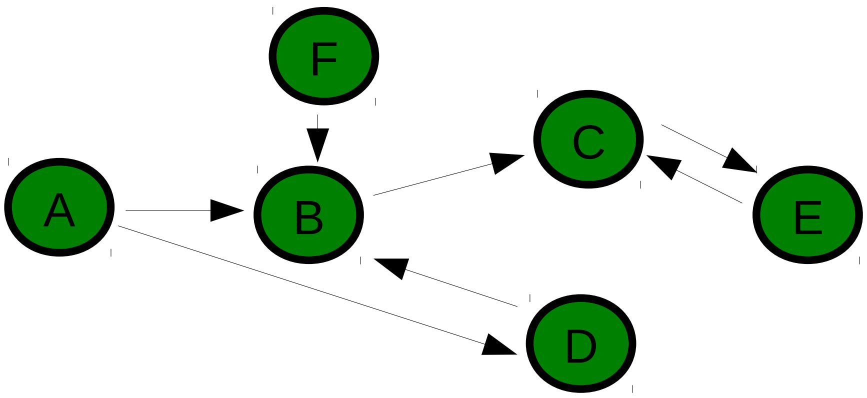


- Imagine experiment:
  - Assign all nodes importance 1.
  - Then update $x'_i = \sum_j a_{ij} x_j \longrightarrow x(t) = A^t x(0)$
  - Say $x(0) = \sum_i c_i v_i \longrightarrow x(t) = \sum_i c_i k_i^t v_i = k_1^t \sum_i c_i \left(\frac{k_i}{k_1}\right)^t v_i \rightarrow c_1 k_1^t v_1$

Eigenvectors of {a}

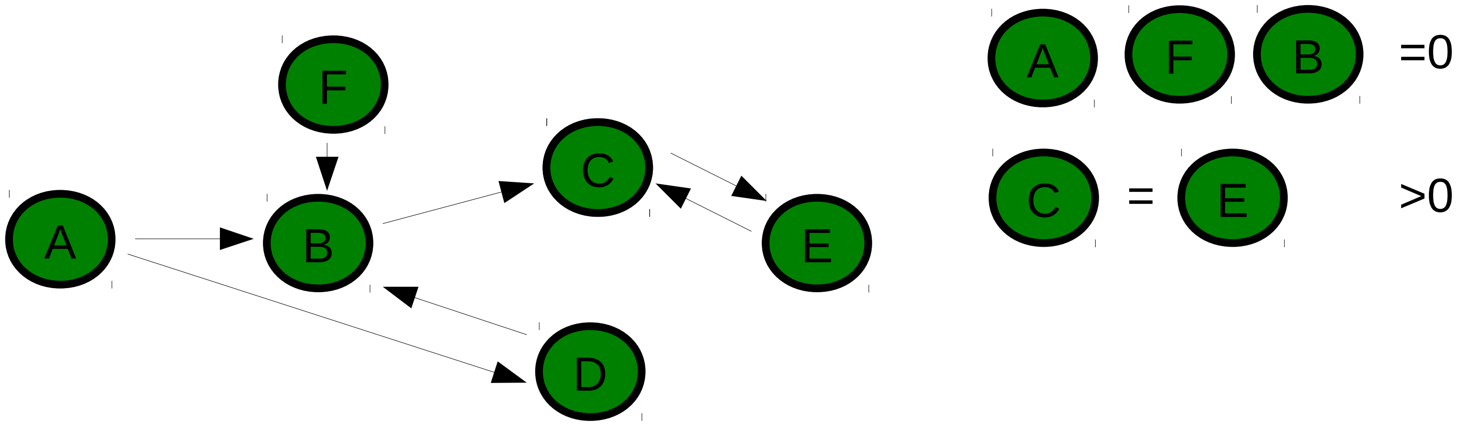
Eigenvector Centrality

- Problems:
  - Normalisation?
  - Directed networks, left or **right** eigenvectors?
  - What about this?



Eigenvector Centrality

- Problems:
  - Normalisation?
  - Directed networks, left or **right** eigenvectors?
  - What about this?



  - According to this definition only nodes in strong components or their out-components have centrality > 0!

Katz Centrality

- Katz (1953); every node gets some amount of centrality for “free” $x = \alpha A x + \beta 1$
- Re-arranging: $x = (I - \alpha A)^{-1} 1$
  - Where alpha balances relative importance of eigenvector component and “free” component
  - Alpha should be between 0 and 1/k1
  - In practice: better solve by iteration than by inverting the adjacency matrix
- Potential problem:
  - All nodes pointed to by a high centrality node receive high centrality!

Pagerank

- To overcome the problem of Katz centrality we could consider:

$$x_i = \alpha \sum_j a_{ij} x_j / k_j^{out} + \beta$$

- In matrix form: $x = \alpha A D^{-1} x + \beta 1$ with $D_{ii} = max(k_i^{out}, 1)$
- Conventionally beta=1: $x = (I - \alpha A D^{-1})^{-1} 1 = D(D - \alpha A)^{-1} 1$
- In principle this is what google uses with alpha=0.85
- Could give nodes different intrinsic importance beta

$$\longrightarrow \quad x = D(D - \alpha A)^{-1} \beta$$

Closeness Centrality

- Nodes that have short geodisic paths d to most other nodes might have better access to information …
- Measure: $l_i = \frac{1}{n} \sum_i d_{ij}$
  - Gives low values for most central nodes
- Hence: $$C_i = \frac{1}{l_i} = \frac{n}{\sum_i d_{ij}}$$
- Problems:
  - Typically spans small range of values (1 … 1/log(n)), so extreme values are not very robust to minor changes in network organisation
  - Disconnected components … C=0
    - Maybe use harmonic means? $C'_i = 1/(n-1) \sum_j 1/d_{ij}$

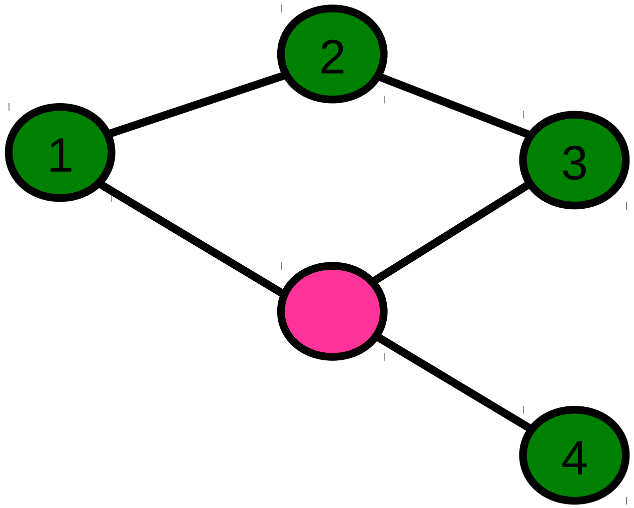
Betweenness Centrality

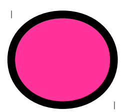
- Idea goes back to Freeman (1977)
  - Suppose we have a network in which messages are passed around, let's say with equal probability between each pair of nodes
  - Assume that messages are passed along the shortest routes
  - Number of messages through each vertex proportional to number of geodesic paths through this vertex. Call this betweenness.
  - If there are g(s,t) geodesic paths between s and t and n(v,s,t)=1 if a geodesic path between s and t goes through v and 0 otherwise, then:

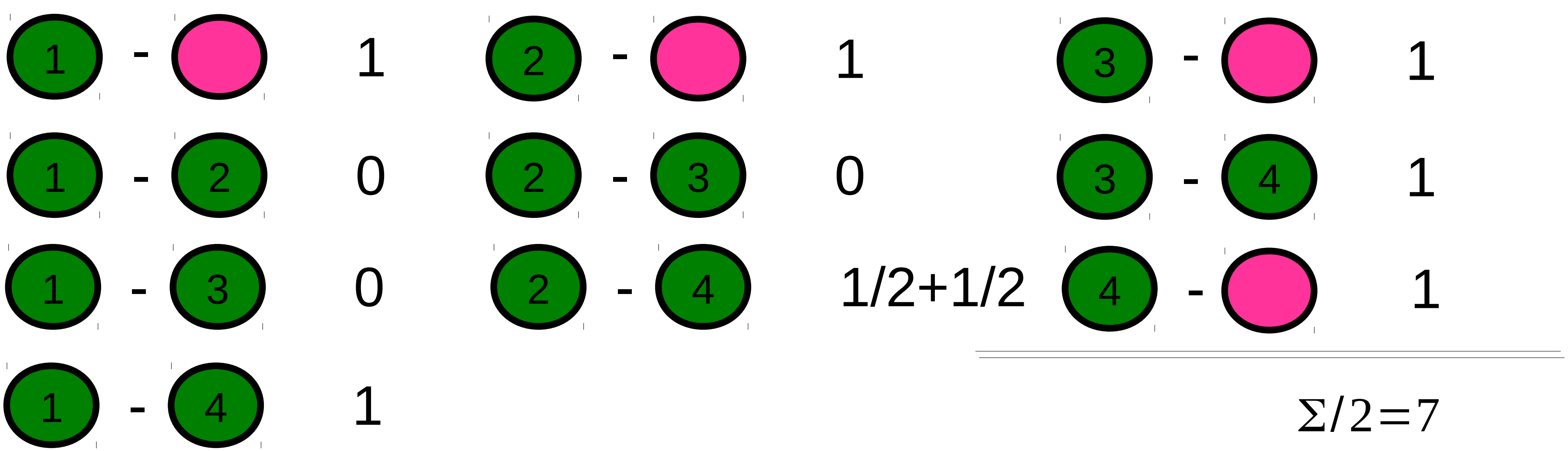
$$b_v = \sum_{s,t} n(v,s,t)/g(s,t)$$

Betweenness Centrality (2)

- Sounds more complicated than it is …

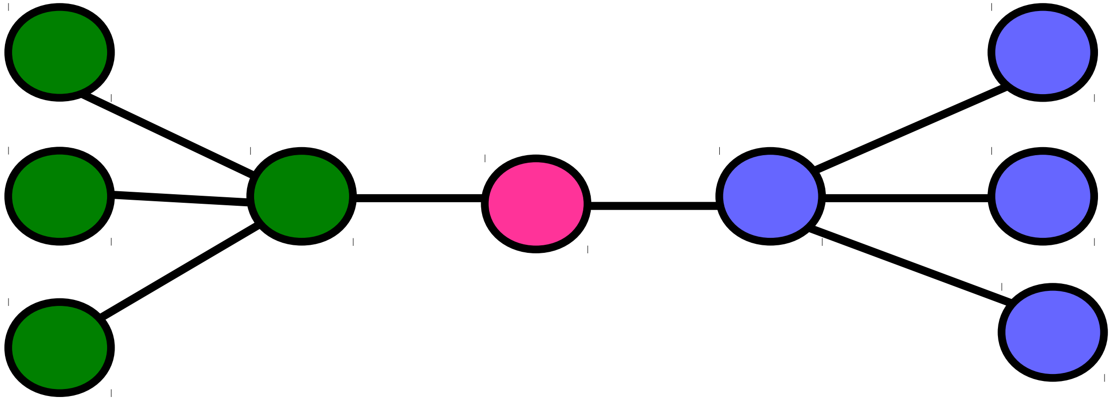


- Centrality of  ?



Betweenness (3)

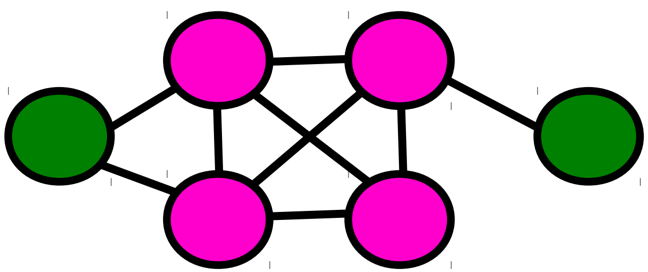
- Not necessarily a measure of how well connected a node is, rather how much a node is “between” others



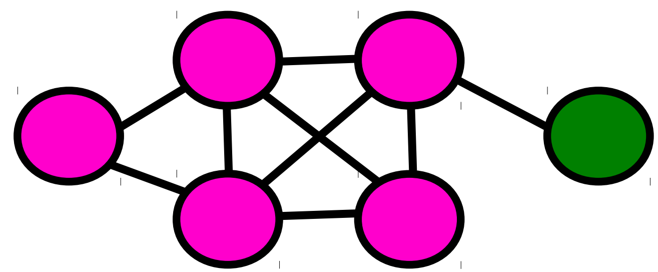
- Sometimes normalised $b_v = 1/n^2 \sum_{s,t} n(v,s,t)/g(s,t)$
- Typically spans large range of values
- Various generalisations, e.g. flow betweenness

Cliques and Cores

- Clique (undirected nw)
  - Maximal set of vertices such that every member is connected by an edge to every other
  - Indication of cohesive subgroups



Example of a 4 clique

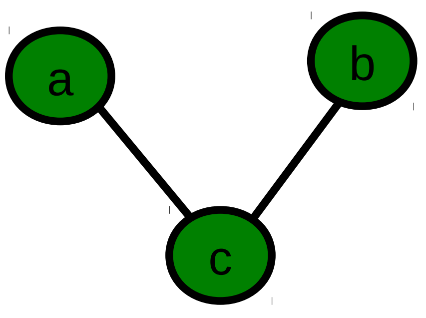


Example of a 2 core

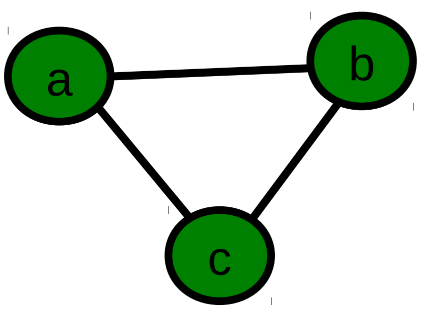
- K-Core:
  - Maximal subset such that each vertex is connected to at least k others
  - Easy to construct by successively eliminating nodes of degree less than k

Transitivity/Clustering

- In maths transitivity usually implies: a ~ b and b ~ c → a ~ c
- In network science one often uses "~" = connected by an edge, i.e. if a is connected to b and b to c then also a should be connected to c



(open triad)

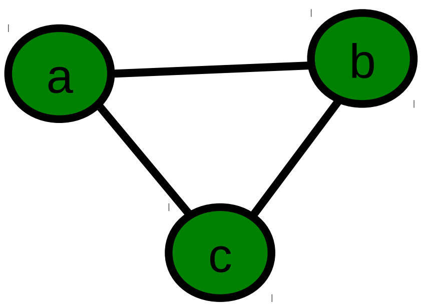


(closed triad)

- Perfect transitivity only in cliques, partial transitivity is of more interest

Clustering Coefficients

- Want to measure the degree to which a network is transitive
- Clustering coefficient $= \dfrac{\text{\#closed paths of length 2}}{\text{\#paths of length 2}}$



abc
acb
bac
bca
cab
cba

$$\frac{6\ \text{*\#triangles}}{\text{\#paths of length 2}}$$

$$\frac{3\ \text{*\#triangles}}{\text{\#connected triples}}$$

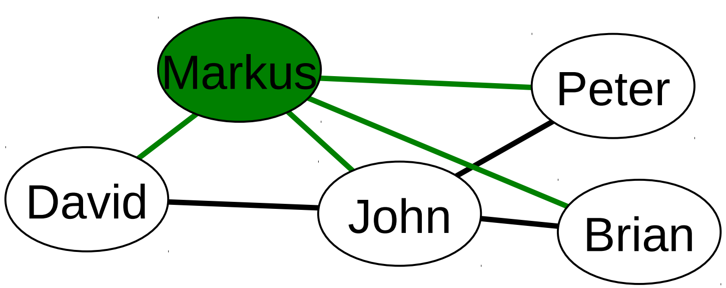
- Remarks:
  - 0<=C<=1
  - Directed networks are often considered undirected

Localised Clustering Coefficients

- Clustering coefficient of a node
  - CC(node)=fraction of pairs of friends that are friends of each other

$$CC=\frac{2 * \#\text{links between friends}}{k\,(k-1)}$$

- Example:



k(Markus)=4
#links between friends=2
CC(Markus)=4/(4*3)=1/3

- In the context of social networks missing triadic closures are called structural holes
  - Might impede flow of information or give power to a node

Clustering Coefficients

- Remarks:
  - Local CC often used to calculate a global CC via

    $$C = 1/n \sum_i C_i$$

    (it is important to note that this definition differs from the first definition, the second measure is dominated by low degree nodes)
  - Typical social networks: C~0.1...0.6
  - Q: Is C=0.1 high for a network with 1000 nodes and 2000 links?

Answer

- Suppose everybody has the same number of friends

$$2L = \sum_i k_i \approx Nk \longrightarrow k \approx 2L/N = 4$$

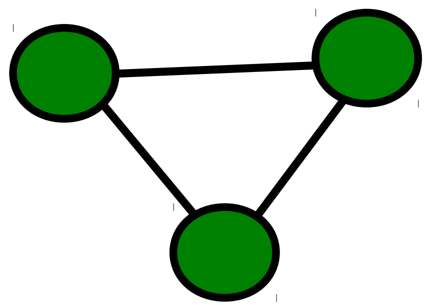
- Suppose everyone picks friends at random from the entire population, i.e. the chance that two friends of mine are connected is approximately

$$CC \approx k/N = 0.004$$

- So yes!

Reciprocity

- Why focus on triangles (=cycles of length 3)?
  - They are the shortest for undirected networks
  - Not so for directed networks!

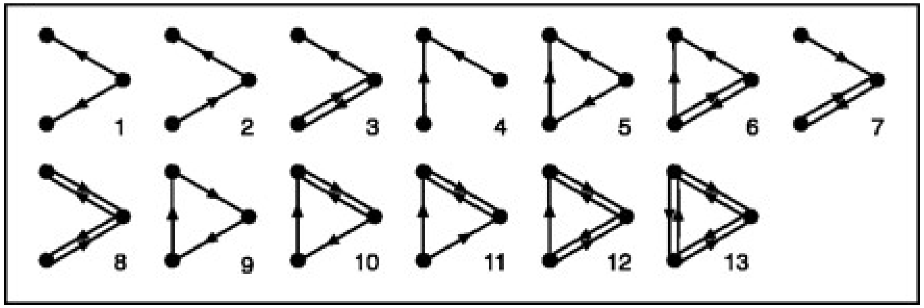


- Measure density of cycles of length 2 (or **reciprocated** connections)

$$r = 1/L \sum_{i,j} a_{ij} a_{ji} = 1/L \, Tr \, A^2$$

Motifs

- No reason to stop with 2-cycles or 3-cycles (triangles)
- In particular for directed networks, we can count frequencies of small subgraphs (=motifs)

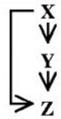
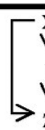
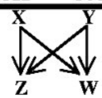
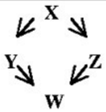
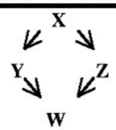
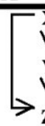
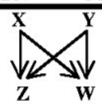
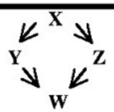
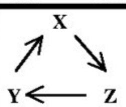
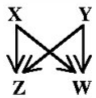
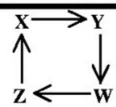
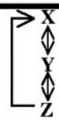
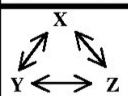
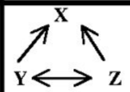


Motifs involving 3 nodes
(from Milo R et al. (2002). Science 298 (5594): 824–827.)

- Two steps:
  - Frequency counting
  - Significance evaluation

$$Z(G') = \frac{F_G(G') - \mu_R(G')}{\sigma_R(G')}$$

Motifs in Complex Networks

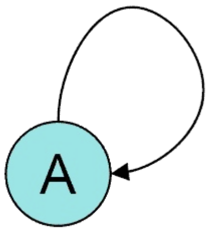
Network	Nodes	Edges	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score	N_{real}	$N_{rand} \pm SD$	Z score
Gene regulation (transcription)					**Feed-forward loop**			**Bi-fan**			
E. coli	424	519	40	7 ± 3	10	203	47 ± 12	13			
*S. cerevisiae**	685	1,052	70	11 ± 4	14	1812	300 ± 40	41			
Neurons					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
C. elegans†	252	509	125	90 ± 10	3.7	127	55 ± 13	5.3	227	35 ± 10	20
Food webs					**Three chain**			**Bi-parallel**			
Little Rock	92	984	3219	3120 ± 50	2.1	7295	2220 ± 210	25			
Ythan	83	391	1182	1020 ± 20	7.2	1357	230 ± 50	23			
St. Martin	42	205	469	450 ± 10	NS	382	130 ± 20	12			
Chesapeake	31	67	80	82 ± 4	NS	26	5 ± 2	8			
Coachella	29	243	279	235 ± 12	3.6	181	80 ± 20	5			
Skipwith	25	189	184	150 ± 7	5.5	397	80 ± 25	13			
B. Brook	25	104	181	130 ± 7	7.4	267	30 ± 7	32			
Electronic circuits (forward logic chips)					**Feed-forward loop**			**Bi-fan**			**Bi-parallel**
s15850	10,383	14,240	424	2 ± 2	285	1040	1 ± 1	1200	480	2 ± 1	335
s38584	20,717	34,204	413	10 ± 3	120	1739	6 ± 2	800	711	9 ± 2	320
s38417	23,843	33,661	612	3 ± 2	400	2404	1 ± 1	2550	531	2 ± 2	340
s9234	5,844	8,197	211	2 ± 1	140	754	1 ± 1	1050	209	1 ± 1	200
s13207	8,651	11,831	403	2 ± 1	225	4445	1 ± 1	4950	264	2 ± 1	200
Electronic circuits (digital fractional multipliers)					**Three-node feedback loop**			**Bi-fan**			**Four-node feedback loop**
s208	122	189	10	1 ± 1	9	4	1 ± 1	3.8	5	1 ± 1	5
s420	252	399	20	1 ± 1	18	10	1 ± 1	10	11	1 ± 1	11
s838‡	512	819	40	1 ± 1	38	22	1 ± 1	20	23	1 ± 1	25
World Wide Web					**Feedback with two mutual dyads**			**Fully connected triad**			**Uplinked mutual dyad**
nd.edu§	325,729	1.46e6	1.1e5	2e3 ± 1e2	800	6.8e6	5e4±4e2	15,000	1.2e6	1e4 ± 2e2	5000

(from Milo R et al. (2002). Science 298 (5594): 824–827.)

Motifs (2)

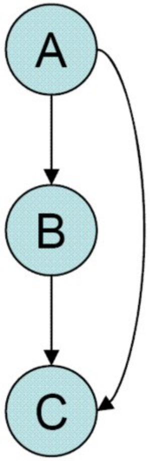
- Especially in GNRs motifs have been linked to certain functions

Self-loop



Positive/negative auto-regulation

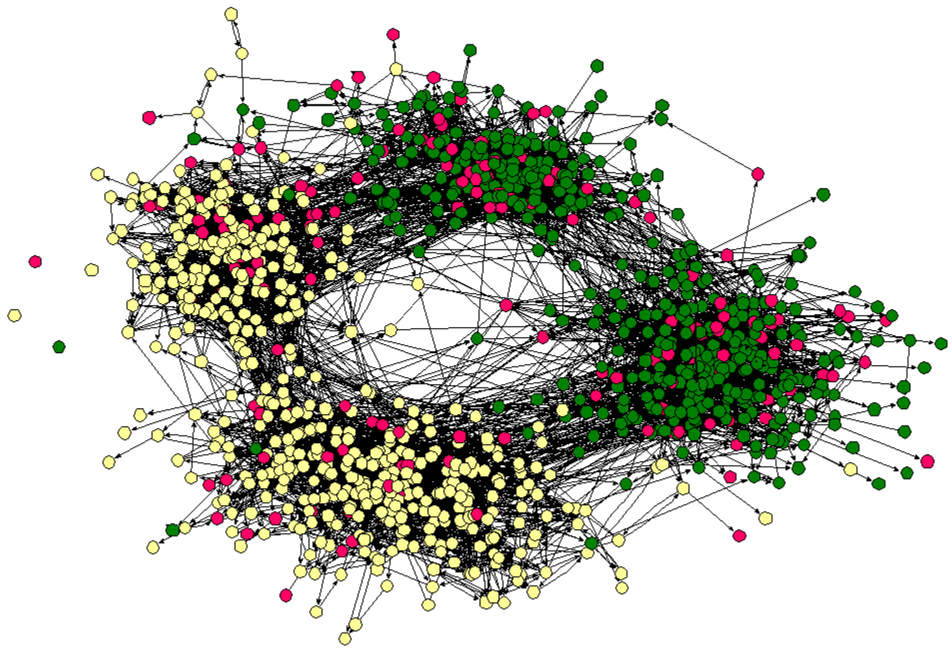
Feed-forward loops



May serve as AND or OR

Homophily

Nodes are colour coded by race (black, white, other)



Mixing is clearly not random.

Most “social” characteristics In social networks mix like this.

How do we measure such mixing patterns?

High school friendship: James Moody, Race, school integration, and friendship segregation in America, American Journal of Sociology 107, 679-716 (2001).

Homophily (1)

- Idea:
  - Say every node I has some classifier c(i)
  - Count edges that connect nodes of same type, subtract number of edges that we would expect to connect nodes of same type in a random arrangement

$$L_1 = 1/2 \sum_{ij} a_{ij} \delta_{c_i, c_j}$$

← Kronecker delta: =1 if $c_i = c_j$, 0 otherwise

$$L_2 = 1/2 \sum_{ij} \frac{k_i k_j}{2L} \delta_{c_i, c_j}$$

(chance that one edge from i connects to j is kj/2L, hence kikj/2L is chance that any of i's ki edges connects to j)

Homophily (2)

- Conventionally this is normalised by L:

$$Q = 1/L(L_1 - L_2) = 1/2m \sum_{ij} (a_{ij} - \frac{k_i k_j}{2L}) \delta_{c_i, c_j}$$

(Q is often called “modularity”: measures the extent to which like is connected to like)

- Value of Q is between -1 and 1, but often does not assume 1 even for a perfectly assorted NW
- Assortativity coefficient: Q/Q_{max}

$$Q/Q_{max} = \frac{\sum_{ij} (a_{ij} - k_i k_j / 2L) \delta_{c_i, c_j}}{2L - \sum_{ij} (k_i k_j / 2L) \delta_{c_i, c_j}}$$

Homophily (3)

- What if the node characteristics c are ordinal?
- Define mean of c at the end of an edge as

$$\langle c \rangle = \frac{\sum_{ij} a_{ij} c_i}{\sum_{ij} a_{ij}}$$

(this is not the usual mean, but degree weighted)

- Then calculate covariance over edges

$$cov(c_i, c_j) = \frac{\sum_{ij} a_{ij} (c_i - \langle c \rangle)(c_j - \langle c \rangle)}{\sum_{ij} a_{ij}}$$

Homophily (4)

- Can rewrite this a bit …

$$cov(c_i, c_j) = \frac{\sum_{ij} a_{ij}(c_i - \langle c \rangle)(c_j - \langle c \rangle)}{\sum_{ij} a_{ij}} = 1/2L \sum_{ij} a_{ij} c_i c_j - \langle c \rangle^2$$

$$= 1/2L \sum_{ij} a_{ij} c_i c_j - 1/(2L)^2 \sum_{ij} k_i k_j c_i c_j$$

$$= 1/2L \sum_{ij} \left(a_{ij} - \frac{k_i k_j}{2L}\right) c_i c_j$$

(reminds strongly of the modularity!)

- Normalisation?
  - Perfect assortment for:

$$1/2L \sum_{ij} a_{ij} c_i^2 - 1/(2L)^2 \sum_{ij} k_i k_j c_i c_j = 1/2L \sum_{ij} (a_{ij} \delta_{ij} - k_i k_j / 2L) c_i c_j$$

  - And the normalised ratio is:

$$r = \frac{\sum_{ij} (a_{ij} - k_i k_j / 2L) c_i c_j}{\sum_{ij} (k_i \delta_{ij} - k_i k_j / 2L) c_i c_j}$$

(this turns out to be a type of **Pearson correlation coefficient**)

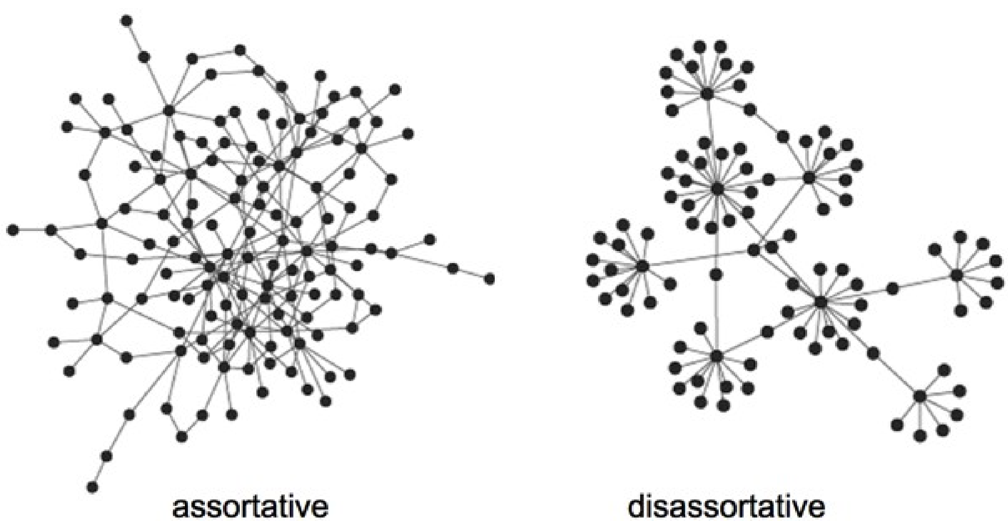
Assortment by Degree

- One of the properties c we might be interested in is degree.
- In how far does degree dictate position of edges?

$$r = \frac{\sum_{ij} (a_{ij} - k_i k_j / 2L) k_i k_j}{\sum_{ii} (k_i \delta_{ii} - k_i k_i / 2L) k_i k_i}$$

r>0

(**assortative** network)



r<0

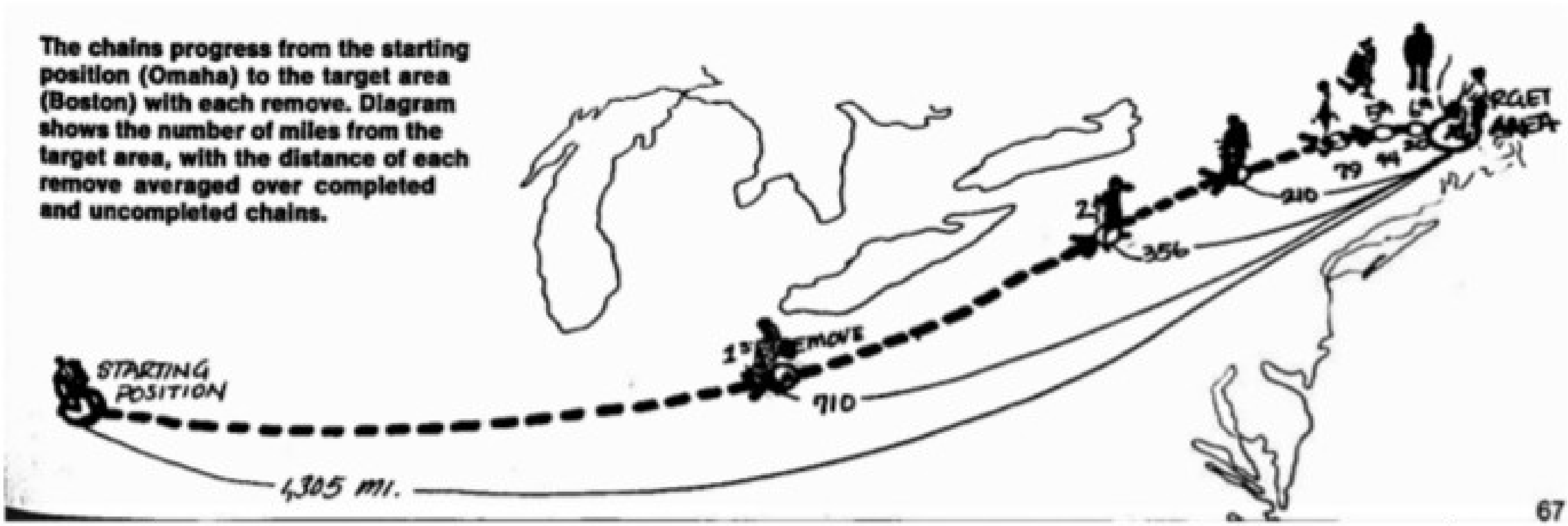
(**disassortative** network)

→ Tends to have core-periphery structure

Distances

- Milgram's small-world experiment (1960s)
  - Milgram sent 96 packages to randomly selected individuals from the phone directory in Omaha; package contained the name of a target individual and its address in Boston
  - Individuals were asked to pass package on to someone they new on first name basis who might be closer to target
  - 18 found their way back
  - Mean lengths of paths was 5.9! (the famous “6 degrees of separation”)
  - This is quite remarkable also in terms of navigation …

Milgram's Experiment



Actual path of the letter traveling from Nebraska to Boston [Milgram '67]

Distances (2)

- Can now quantify this for a given network by average shortest path lengths or mean geodesic distances

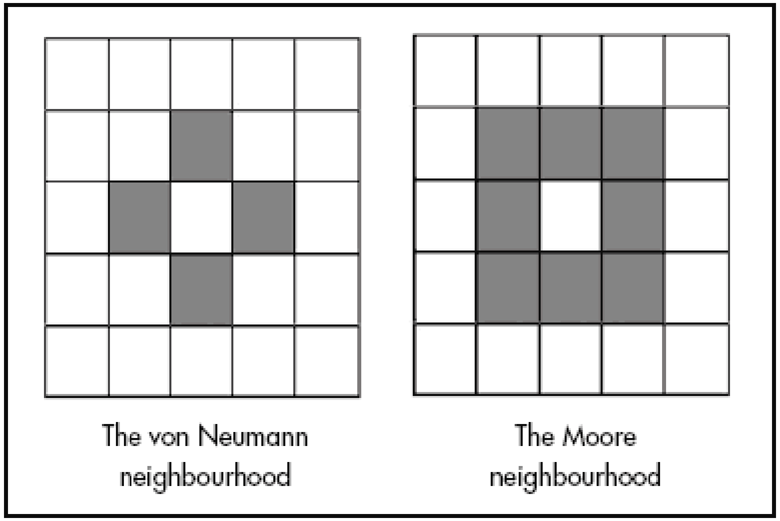
$$d = \frac{1}{N(N-1)} \sum_{ij} d(i,j)$$

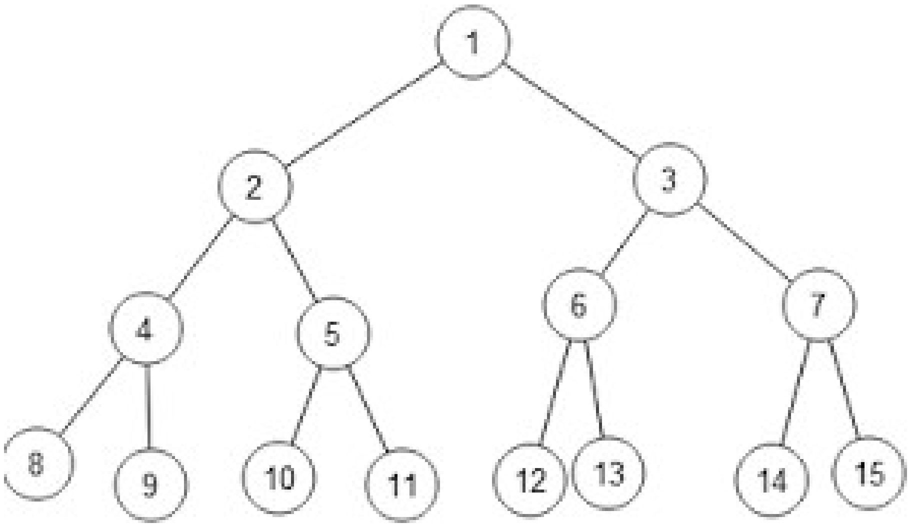
- Sometimes people also measure “diameters”

$$D = max_{i,j} d(i,j)$$

Distances (3)

- What do you expect for:
  - An L by L 2d grid (Moore) with N=L^2 nodes
  - A tree with coordination number 2 composed of N nodes?





Solution – Grid

- Grid (roughly)
  - D=L-1
  - d? – Number of nodes at distance l from node in left hand corner:

$$n_l = 2l + 1$$

$$d \leq 1/L^2 \sum_{l=1}^{L-1} l\, n_l = 1/L^2 \underbrace{\sum_{l=0}^{L-1} l(2l+1)}_{\approx \int_0^{L-1} (4l^2 + l)\, dl \propto L^3} \propto L = N^{1/2}$$

- On d dimensional grids distances scale with

$d \propto N^{1/d}$ (scales algebraically with size, not really “small”...)

Solution – Tree

- Nodes at distance l from root: $n_l = 2^l$

$$N = \sum_{l=0}^{D} n_l = 2^{D+1} - 1$$

$$\longrightarrow \quad D = \log_2(N+1) - 1$$

- Similarly:

$$d = 1/N \sum_{l=0}^{D} l\, n_l = \left(\sum_{l=0}^{D} l\, 2^{\alpha l}\right)(\alpha = 1)$$

$$= 1/N\, 1/\ln 2 \frac{\partial}{\partial \alpha}\left(\sum_{l=0}^{D} 2^{\alpha l}\right)(\alpha = 1) = ¿$$

$$= 1/N\, 1/\ln 2 \frac{\partial}{\partial \alpha}\left(\frac{2^{(D+1)\alpha} - 1}{2^{\alpha} - 1}\right)(\alpha = 1) = \frac{D\, 2^{D+1} + 1}{2^{D+1} - 1}$$

$$\approx D \propto \log N$$

- Distances scale logarithmically with system size