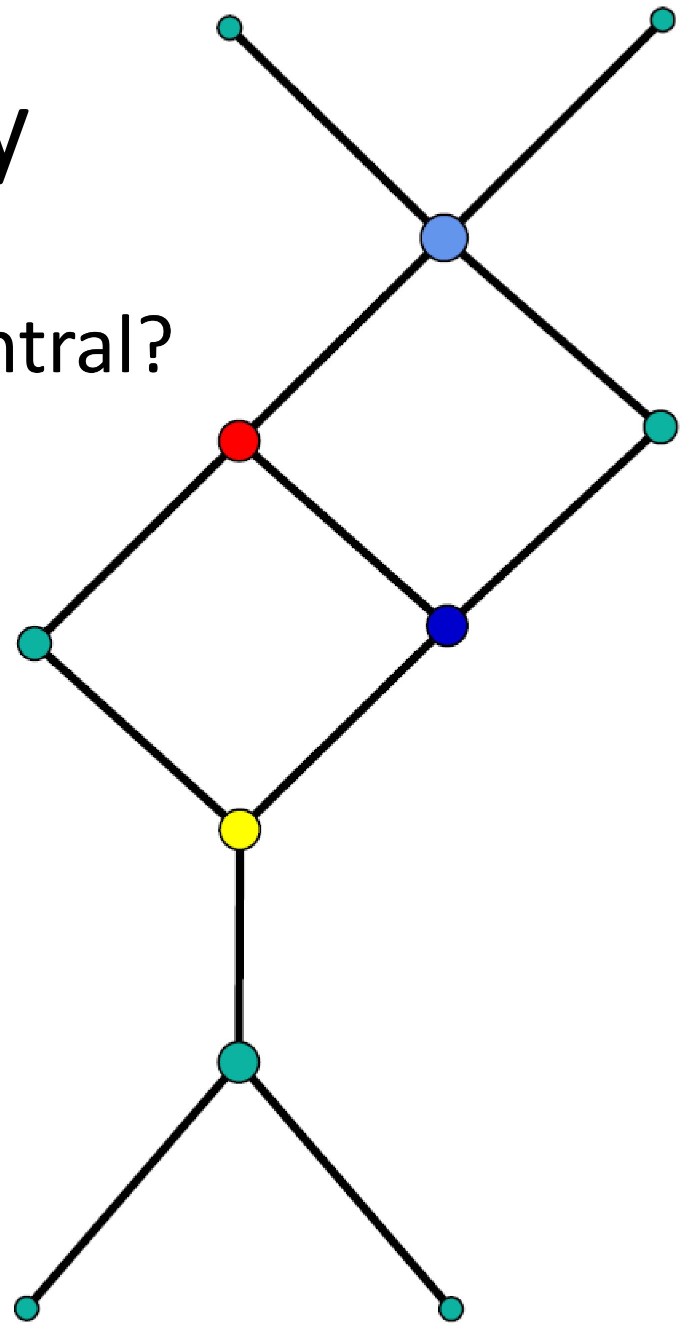
# Complex Networks
# measures and metrics

2018.12.6(Thu)

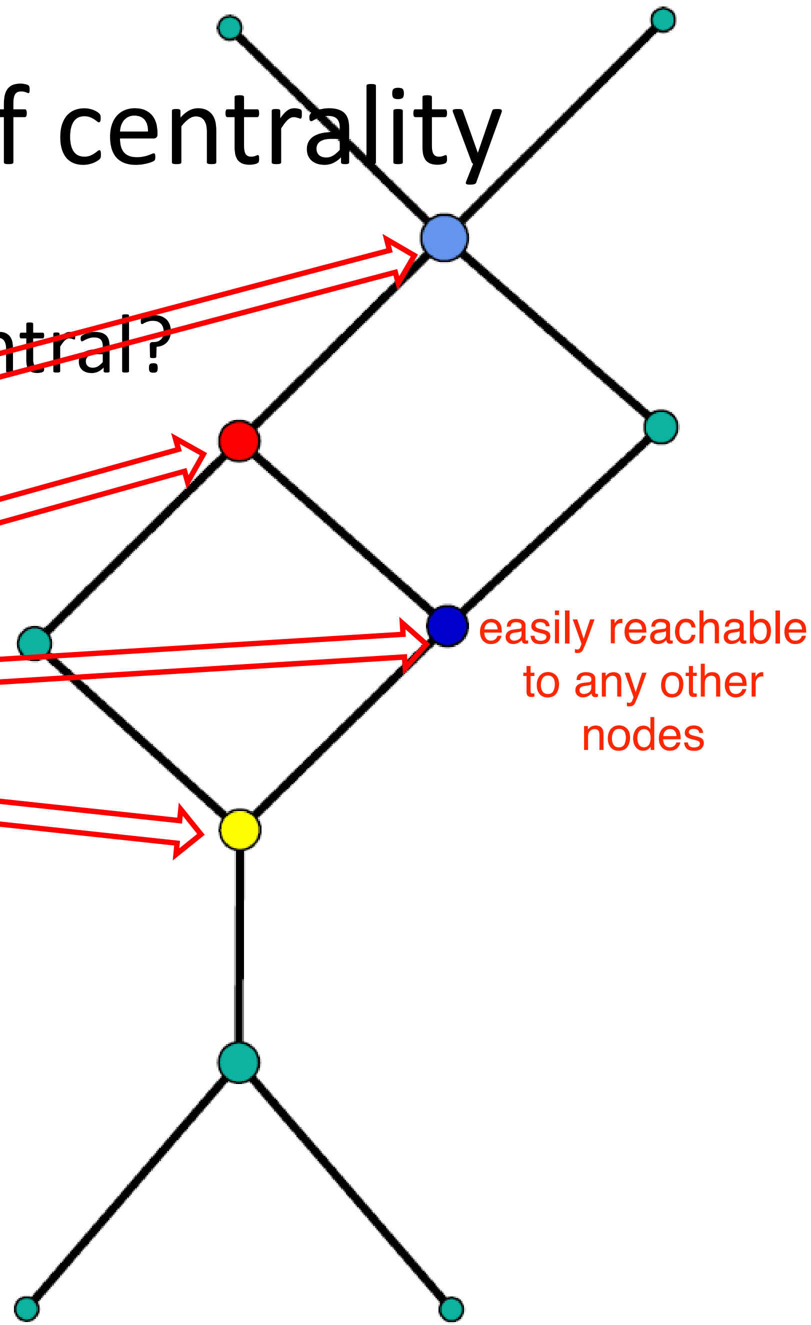
# centrality

- which vertex is the most central?
  - red?
  - blue?
  - green?
  - light blue?
  - yellow?



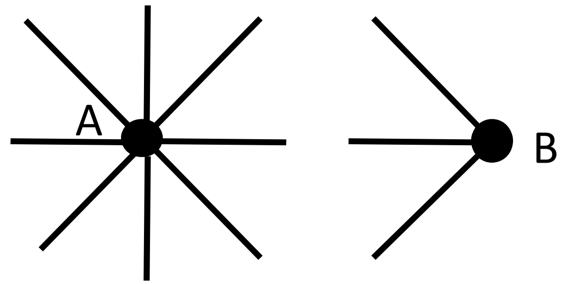
# many definitions of centrality

- which vertex is the most central?
  - degree centrality
  - eigenvector centrality
  - closeness centrality
  - betweenness centrality

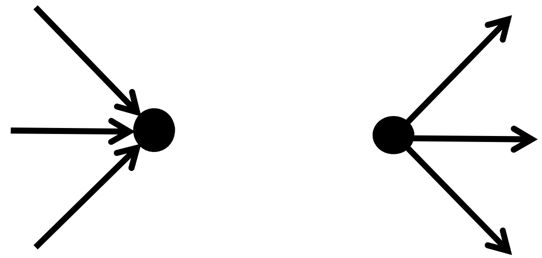


# degree centrality

- # of edges connected to a vertex
  - friendship
  - citation




- directed networks
  - in-degree centrality
  - out-degree centrality

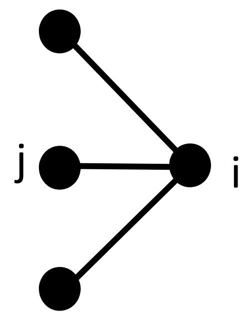


# eigenvector centrality (1)

- neighboring vertices are not equally important
- setting initial values ($x_i$ = 1 for all i)
- update by the sum of the centralities of the neighbors

$$x_i' = \sum_j A_{ij} x_j$$

$$\mathbf{x}' = \mathbf{A}\mathbf{x}$$




weight of each neighboring vertices inequal

- repeating this process

$$\mathbf{x}(t) = \mathbf{A}^t \mathbf{x}(0)$$

- write x(0) as a linear combination of eigenvectors

$$\mathbf{x}(0) = \sum_i c_i \mathbf{v}_i$$

$C_i$ : some appropriate choice of constant

# eigenvector centrality (2)

$$\mathbf{x}(t) = \mathbf{A}^t \sum_i c_i \mathbf{v}_i = \sum_i c_i \kappa_i^t \mathbf{v}_i = \kappa_1^t \sum_i c_i \left[\frac{\kappa_i}{\kappa_1}\right]^t \mathbf{v}_i \qquad \because \mathbf{A}^t \mathbf{v}_i = \kappa_i^t \mathbf{v}_i$$

- $\kappa_i$ : eigenvalue of A,   $\kappa_1$ : the largest one
- $\kappa_i / \kappa_1 < 1$ for all i≠1
- when $t \to \infty$ , $\mathbf{x}(t) \to c_1 \kappa_1^t \mathbf{v}_1$
- the centrality x satisfies $\mathbf{Ax} = \kappa_1 \mathbf{x}$   $x_i = \kappa_1^{-1} \sum_j A_{ij} x_j$
  - proposed by Bonacich in 1987
- eigenvector centralities are non-negative

# eigenvector centrality for directed networks

- [problem1]adjacency matrix is asymmetric -> two sets of eigenvectors
  - left eigenvectors and right eigenvectors

    $\mathbf{xA} = \lambda \mathbf{x}$ $\qquad$ $\mathbf{Ax} = \lambda \mathbf{x}$

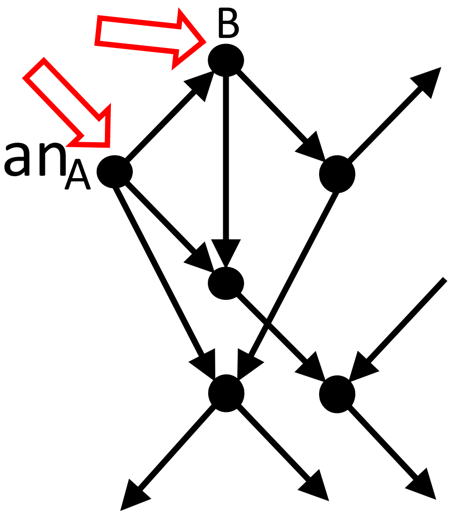
Centrality is determined by other vertices pointing towards you

- in most cases, right eigenvectors are used

$$x_i = \kappa_1^{-1} \sum_j A_{ij} x_j \qquad \mathbf{Ax} = \kappa_1 \mathbf{x}$$

- [problem2] no incoming edges

  -> centrality will be zero
  - only SCCs and their out-components can have non-zero centralities



# Katz centrality

- simply give each vertex a small amount of centrality

$$x_i = \alpha \sum_j A_{ij} x_j + \beta \qquad \mathbf{x} = \alpha \mathbf{A}\mathbf{x} + \beta \mathbf{1} \qquad \mathbf{1} = (1,1,1,\ldots)$$

$$\mathbf{x} = \beta(\mathbf{I} - \alpha\mathbf{A})^{-1} \cdot \mathbf{1}$$ β = 1 (absolute value of x is not important)

$$\mathbf{x} = (\mathbf{I} - \alpha\mathbf{A})^{-1} \cdot \mathbf{1}$$

- α:balance between the eigenvector term and constant term
- if α→0, all vertices have the same centrality β
- as we increase α, x diverges when $(\mathbf{I} - \alpha\mathbf{A})^{-1}$ diverges $\det(\mathbf{A} - \alpha^{-1}\mathbf{I}) = 0$

$\alpha^{-1} = \kappa_1$ the largest eigenvector of A

α should be less than $1/\kappa_1$ if we wish the centrality converge

# calculating Katz centrality

- inverting matrix : (O($n^3$)) slow

$$\mathbf{x} = (\mathbf{I} - \alpha\mathbf{A})^{-1} \cdot \mathbf{1}$$

# of vertices

- update x repeatedly: (rm)

$$\mathbf{x}' = \alpha\mathbf{A}\mathbf{x} + \beta\mathbf{1}$$

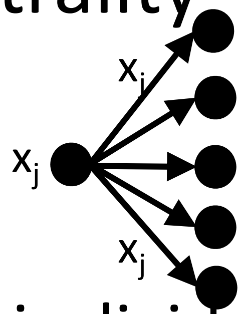
# of iteration

# of edges

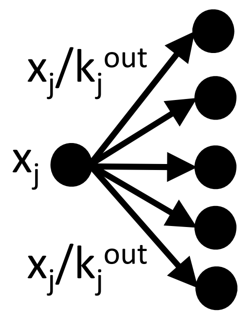
# PageRank (1)

$$x_i = \alpha \sum_j A_{ij} x_j + \beta$$

- weakness of Katz centrality: if a vertex with high Katz centrality points to many others, then those others also get high centrality
  - centrality should be diluted




- PageRank
  - the centrality derived from neighbors is divided by their out-degree




$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j^{out}} + \beta$$

for the vertices with zero outdegree ($k_i^{out}=0$), we artificially set $k_i^{out}=1$

# PageRank (2)

$$x_i = \alpha \sum_j A_{ij} \frac{x_j}{k_j^{out}} + \beta$$

$$\mathbf{x} = \alpha \mathbf{A}\mathbf{D}^{-1}\mathbf{x} + \beta \mathbf{1}$$

D: diagonal matrix with elements $D_{ii}=\max(k_i^{out},1)$

$$\mathbf{x} = \beta(\mathbf{I} - \alpha \mathbf{A}\mathbf{D}^{-1})^{-1} \cdot \mathbf{1}$$

β is set to 1

$$\mathbf{x} = \mathbf{D}(\mathbf{D} - \alpha \mathbf{A})^{-1} \cdot \mathbf{1}$$

- Google uses it as a central part of their Web ranking technology
- α should be less than the inverse of the largest eigenvalue of $AD^{-1}$
- α=0.85 is often used

# summary of centrality measures

solve heavy centrality problem

| | with constant term | without constant term |
|---|---|---|
| divide by out-degree | $\mathbf{x} = \mathbf{D}(\mathbf{D} - \alpha\mathbf{A})^{-1} \cdot \mathbf{1}$ PageRank | $\mathbf{x} = \mathbf{A}\mathbf{D}^{-1}\mathbf{x}$ degree centrality |
| no division | $\mathbf{x} = (\mathbf{I} - \alpha\mathbf{A})^{-1} \cdot \mathbf{1}$ Katz centrality | $\mathbf{x} = \kappa_1^{-1}\mathbf{A}\mathbf{x}$ eigenvector centrality |

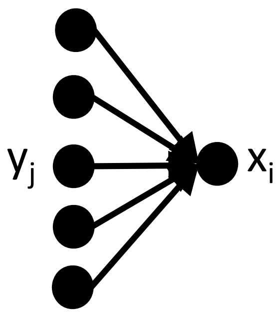
solve 0 centrality problem

# hubs and authorities (1)

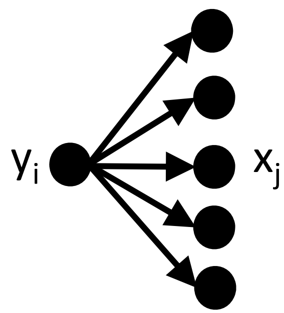
- two types of important vertices
  - authorities: vertices that contain useful information
  - hubs: vertices that tell us where the best authorities are to be found

eg. webpages contain link to the best authorities

- HITS (hyperlink-induced topic search) : search authority centrality ($x_i$) and hub centrality ($y_i$)




good authority is pointed by many good hubs




good hub points to many good authorities

# hubs and authorities (2)

- authority centrality ($x_i$) and hub centrality ($y_i$) are mutually recursive

$$x_i = \alpha \sum_j A_{ij} y_j \qquad y_i = \beta \sum_j A_{ji} x_j$$

$$\mathbf{x} = \alpha \mathbf{A} \mathbf{y} \qquad \mathbf{y} = \beta \mathbf{A}^T \mathbf{x}$$

$$\mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda \mathbf{x} \qquad \mathbf{A}^T \mathbf{A} \mathbf{y} = \lambda \mathbf{y} \qquad \lambda = (\alpha\beta)^{-1}$$

- authority and hub centralities are given by eigenvectors of $AA^T$ and $A^TA$ with the same eigenvalue (leading eigenvalue should be used)
- $AA^T$ and $A^TA$ have the same eigenvalues

$$\mathbf{A}\mathbf{A}^T \mathbf{x} = \lambda \mathbf{x}$$

$A^T x$ is an eigenvector of $A^TA$ with the same eigenvalue λ

$$\mathbf{A}^T \mathbf{A}(\mathbf{A}^T \mathbf{x}) = \lambda (\mathbf{A}^T \mathbf{x}) \qquad \mathbf{y} = \mathbf{A}^T \mathbf{x}$$

# hubs and authorities (3)

- $AA^T$ is cocitation matrix
- $A^TA$ is bibliographic coupling matrix
- hub and authority centralities circumvent the problems of eigenvector centrality with directed network

  strong connected

  – problem: vertices outside of SCC or out-components always have centrality zero
  – vertices not cited by any others have authority centrality zero, but they can still have no-zero hub centrality

HITS is used as the basis for the Web search engines Teoma and Ask.com

# closeness centrality

- mean distance from a vertex to other vertices

$$l_i = \frac{1}{n}\sum_j d_{ij}$$ $d_{ij}$ : length of geodesic path from i to j

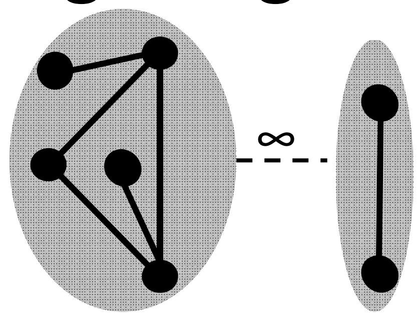
- <u>low</u> values for vertices that are close to others
- closeness centrality : inverse of $l_i$

$$C_i = \frac{1}{l_i} = \frac{n}{\sum_j d_{ij}}$$

- problems of closeness centrality
  – span a rather small range from largest to smallest
  – vertices in smaller component will get higher value

# problems of closeness centrality

- span a rather small range from largest to smallest
  - difficult to distinguish between central and less central ones (small fluctuations can change the order)
  - Internet Movie Database: half a million actors
    - smallest centrality 2.4138, largest centrality 8.6681
- vertices in smaller component will get higher value
  - redefine closeness: $C_i' = \frac{1}{n-1} \sum_{j(\neq i)} \frac{1}{d_{ij}}$




when disconnected, 1/d_ij -> 0

# mean geodesic distance

- for a network with only one component

$$l = \frac{1}{n^2}\sum_{ij} d_{ij} = \frac{1}{n}\sum_i l_i$$ mean of $l_i$ over all vertices

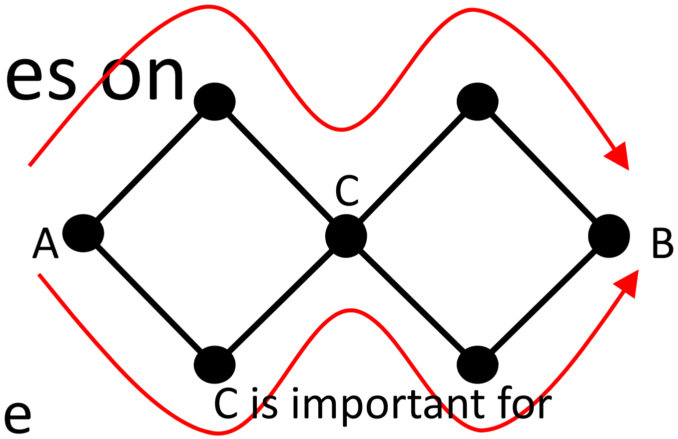
- for a network with more than one component

$$l = \frac{\sum_m \sum_{ij \in \mathscr{C}_m} d_{ij}}{\sum_m n_m^2}$$ $n_m$ : # of vertices in component $\mathscr{C}_m$

average only over the paths in the same component

- alternative approach : harmonic mean distance $$\frac{1}{l'} = \frac{1}{n(n-1)}\sum_{i \neq j} \frac{1}{d_{ij}} = \frac{1}{n}\sum_i C_i'$$

# betweenness centrality (1)




- # of geodesic paths a vertex lies on

$$n_{st}^{i} = \begin{cases} 1 & \text{i is on the path from s to t} \\ 0 & \text{otherwise} \end{cases}$$

- betweenness centrality $x_i$

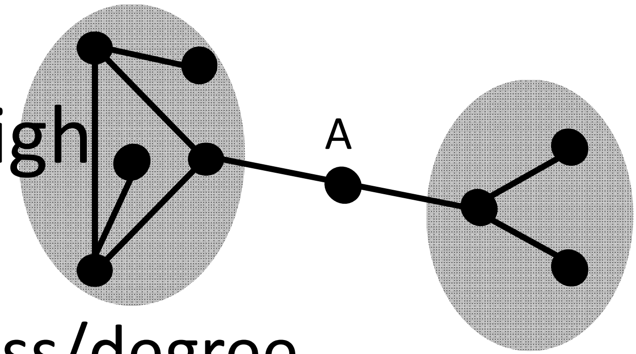
$$x_i = \sum_{st} n_{st}^{i}$$ counts each vertex pair twice

- plural paths -> give weight (=1/(# of paths))

$$x_i = \sum_{st} \frac{n_{st}^{i}}{g_{st}}$$ $g_{st}$ : # of geodesic paths from s to t

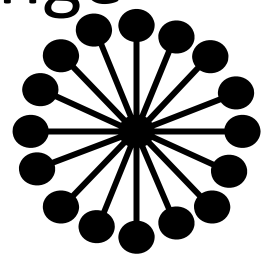
- good also for directed networks

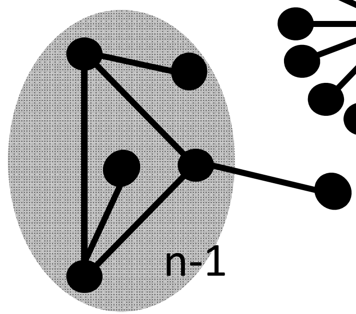
# betweenness centrality (2)




- a vertex on a bridge acquires high betweenness
  - although its eigenvector/closeness/degree centrality is low
- its values are distributed over a wide range
  - maximum : star graph ($n^2$-n+1)
  - minimum : leaf (2n-1)
  - ratio : $$\frac{n^2-n+1}{2n-1} \cong \frac{1}{2}n$$

  large dynamic range -> clear winners/losers





# variation of betweenness centrality

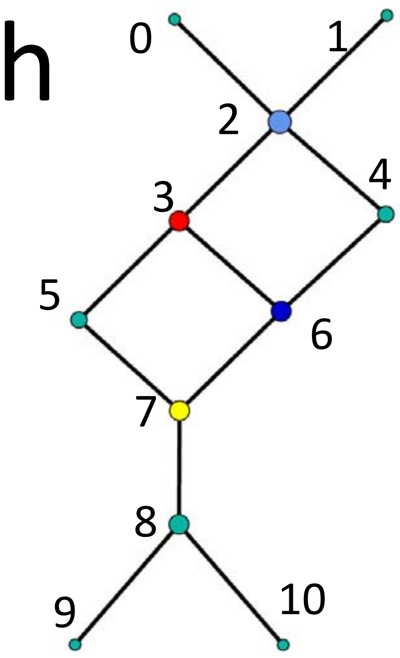
- normalization: $x_i = \frac{1}{n^2}\sum_{st}\frac{n_{st}^i}{g_{st}}$
- flow betweenness: $n_{st}^i$ -> # of <u>independent</u> paths between s and t that run through i
- random-walk betweenness:

  $x_i = \sum_{st} n_{st}^i$ $n_{st}^i$ : # of times that the random walk from s to t passes through i

  – in general, $n_{st}^i \neq n_{ts}^i$

  – random-walk betweenness and shortest-path betweenness often give quite similar results

# centrality with R+igraph




```
library(igraph)
g0 <- graph(c(0,2,1,2,2,3,2,4,3,5,3,6,4,6,5,7,6,7,7,8,8,9,8,10), directed=FALSE)
tkplot(g0)
degree(g0)
 [1] 1 1 4 3 2 2 3 3 3 1 1
betweenness(g0)
 [1]  0.00000  0.00000 17.83333 13.66667  5.50000  6.00000 15.16667 21.83333
 [9] 17.00000  0.00000  0.00000
closeness(g0)
 [1] 0.2941176 0.2941176 0.4000000 0.4545455 0.4166667 0.4347826 0.4761905
 [8] 0.4545455 0.3703704 0.2777778 0.2777778
evcent(g0)$vector
 [1] 0.3609833 0.3609833 0.9416624 1.0000000 0.7344577 0.6926947 0.9742468
 [8] 0.8069662 0.4381138 0.1679495 0.1679495
page.rank(g0)$vector
 [1] 0.04789965 0.04789965 0.16123899 0.11361066 0.07996125 0.07917508
 [7] 0.11315861 0.11770244 0.13537082 0.05199143 0.05199143
> authority.score(g0)$vector
 [1] 0.2950560 0.2950560 0.9665543 0.8173675 0.6003218 0.7110054 1.0000000
 [8] 0.6595879 0.4496949 0.1372765 0.1372765
> hub.score(g0)$vector
 [1] 0.2950560 0.2950560 0.9665543 0.8173675 0.6003218 0.7110054 1.0000000
 [8] 0.6595879 0.4496949 0.1372765 0.1372765
```

degree centrality: 2 is the biggest

betweenness centrality: 7 is the biggest

closeness centrality: 6 is the biggest

eigenvector centrality: 3 is the biggest

PageRank: 2 is the biggest

authority: 6 is the biggest

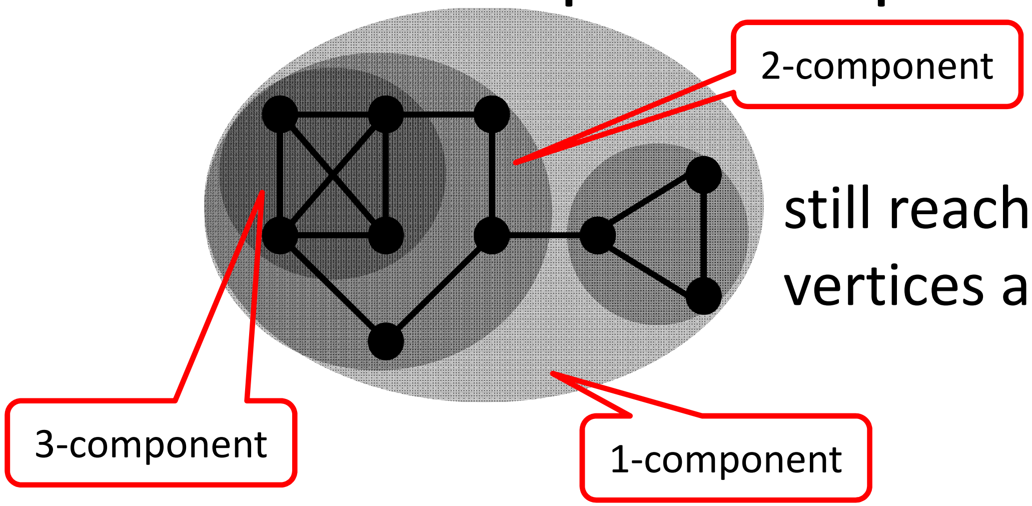
hub: 6 is the biggest

# groups of vertices

- clique : maximal subset of vertices such that every vertex is connected to every other
- k-plex : maximal subset of n vertices such that each vertex is connected to at least n-k of the others
  - 1-plex is clique
- k-core : maximal subset of vertices such that each is connected to at least k others in the subset
  - k-core is (n-k)-plex
- k-clique : maximal subset of vertices such that each is no more than a distance k away from any of the others
- k-clan (k-club) : same as k-clique, but paths should run within the subset

# components and k-components

- components: maximal subset of vertices such that each is reachable from each of the others
- k-component: maximal subset of vertices such that each is reachable from each of the others by at least k vertex-independent paths

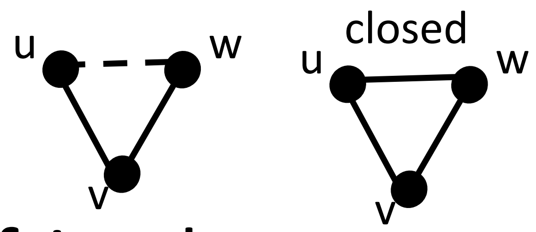



still reachable even if k-1 vertices are removed

strong, not easy to be damaged

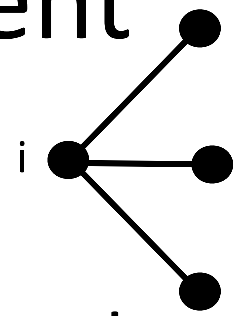
# transitivity

if a and b are friends, b and c are friends, it's highly likely a and c are friends

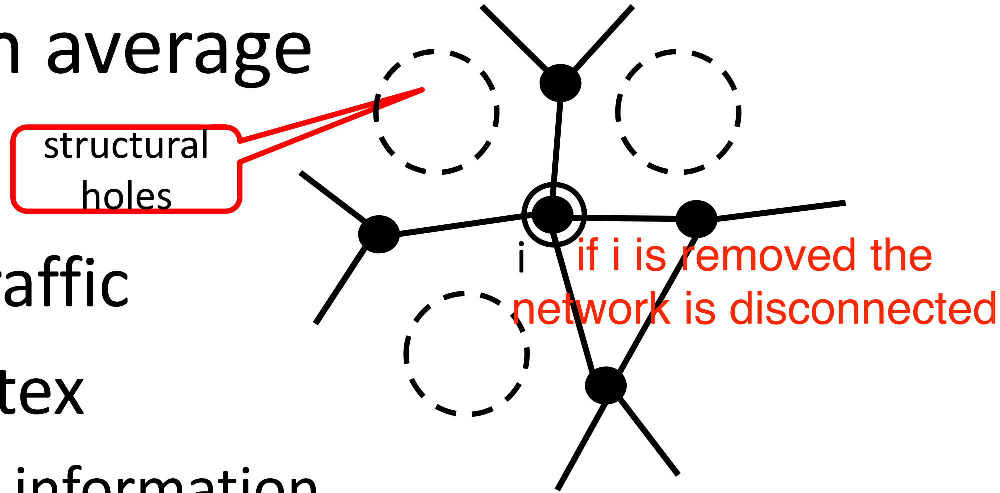



- a•b and b•c -> a•c
- u & v are friends and v & w are friends
- clustering coefficient: $C = \dfrac{\text{(\# of closed paths of length two)}}{\text{(\# of paths of length two)}}$
  - C=1:clique all of the paths of length two are closed
  - C=0:tree, square lattice no triangles
- $C = \dfrac{\text{(\# of triangles)} \times 6}{\text{(\# of paths of length two)}} = \dfrac{\text{(\# of triangles)} \times 3}{\text{(\# of connected triples)}}$
- social networks tend to have high values

# local clustering coefficient




- $C_i = \dfrac{\text{(\# of pairs of neighbors of i that are connected)}}{\text{(\# of pairs of neighbors of i)}}$
- vertices with higher degree have lower local clustering coefficient on average
- structural holes
  - bad for info spread or traffic
  - good for the central vertex
    - it can control the flow of information
- similar to betweenness centrality

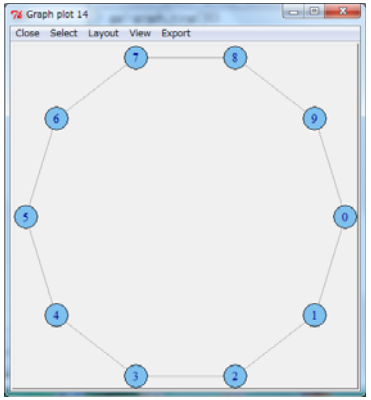


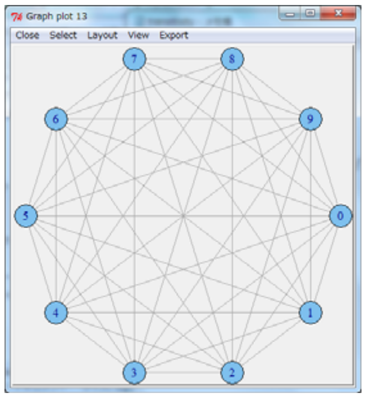
# clustering coefficient with R+igraph

clustering coefficient

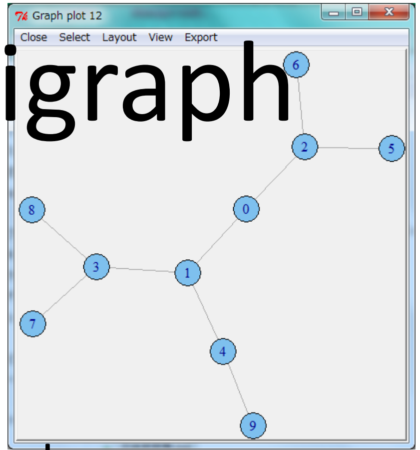
```
> library(igraph)
> gg<-graph.ring(10)
> transitivity(gg)
[1] 0
> gg2<-graph.full(10)
> transitivity(gg2)
[1] 1
> transitivity(gg2,type="global")
[1] 1
> transitivity(gg2,type="local")
 [1] 1 1 1 1 1 1 1 1 1 1
> largest.cliques(gg2)
[[1]]
 [1] 0 1 2 3 4 5 6 7 8 9
```

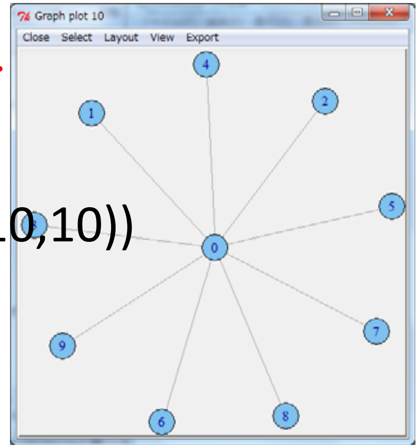
local clustering coefficient

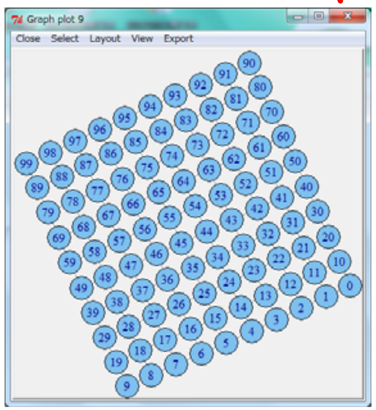



```
> gg3<-graph.tree(10)
> transitivity(gg3)
[1] 0
> gg4<-graph.star(10,mode=
"undirected")
> transitivity(gg4)
[1] 0
> gg5<-graph.lattice(c(10,10))
> transitivity(gg5)
[1] 0
```

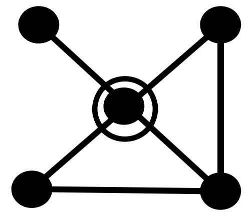


# redundancy

- redundancy of i ($R_i$) : the mean number of connections from a neighbor of i to other neighbors of i
  $$R_i = \frac{1}{4}(0+1+1+2) = 1$$
  - minimum : 0
  - maximum : $k_i - 1$



redundancy will be small if the i is bridge or has many structural holes

mutually connected edges is connected twice

$$C_i = \frac{\frac{1}{2}k_i R_i}{\frac{1}{2}k_i(k_i-1)} = \frac{R_i}{k_i - 1}$$

total number of connections between friends

total number of pairs of friends of i

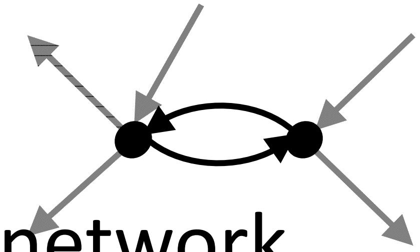
# another clustering coefficient

- $C_{WS}$: the mean of the local clustering coefficients for each vertex

$$C_{WS} = \frac{1}{n}\sum_{i=1}^{n} C_i$$

- We need to be aware of both definitions and clear which is being used

# reciprocity

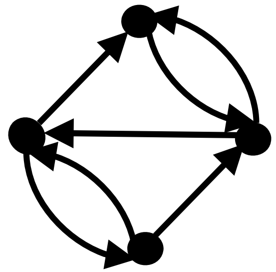


percentage of

- a loop of length two in a directed network

$$r = \frac{1}{m}\sum_{ij} A_{ij} A_{ji} = \frac{1}{m} Tr\mathbf{A}^2$$

m : # of edges

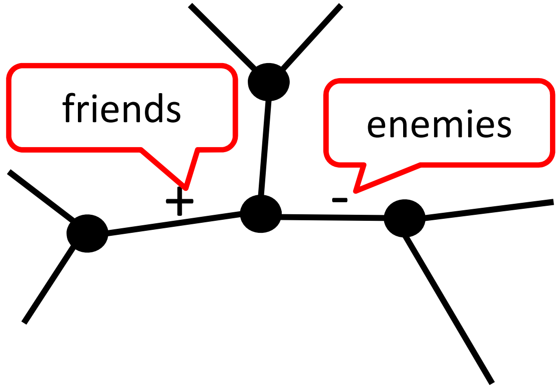


- example: r = 4/7

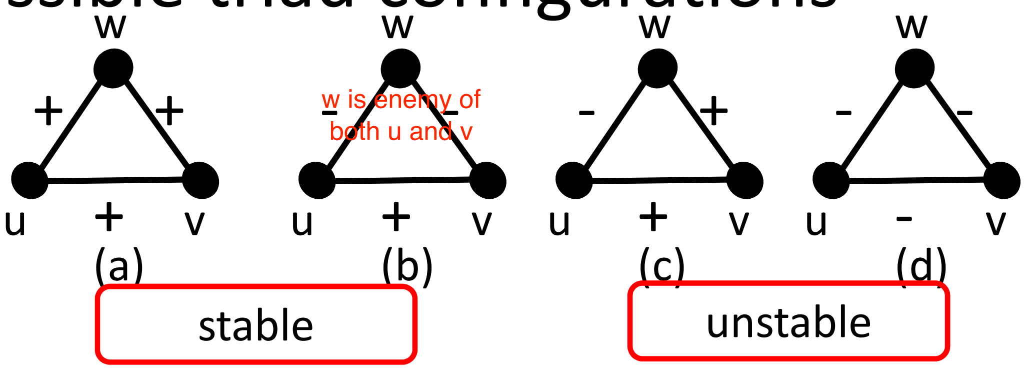
seven directed edges
four are reciprocated

- WWW : r=57%
- Email address book : r=23%

# signed edges



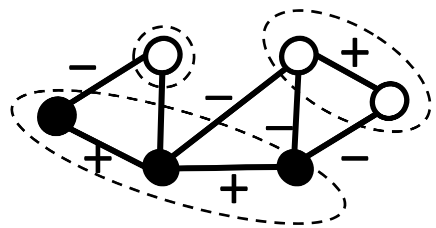

- positive/negative edges
- negative edge ≠ absence of edge
- possible triad configurations




  - stable : even number of minus signs
  - unstable configurations occur far less often in real social networks than stable configurations

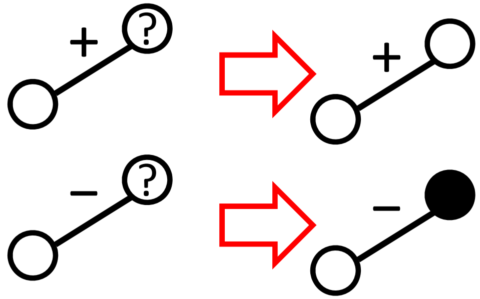
# structural balance

- balanced network : containing only loops with even numbers of minus signs
- Harary's theorem: a balanced network can be divided into connected groups of vertices such that all connection between members of the same group are positive and all connections between members of different groups are negative
  - such network is clusterable

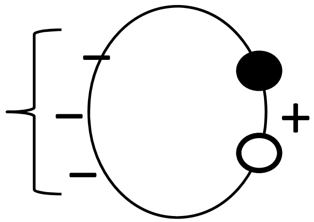


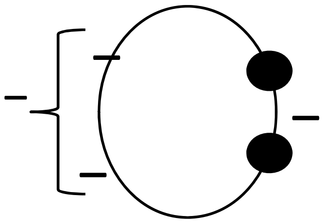
# proof of Harary's theorem

- color in the vertices according to the following algorithm:
  - connected by + : same color
  - connected by − : different color




- conflict of coloring
  - the # of − in the loop is odd -> unbalanced

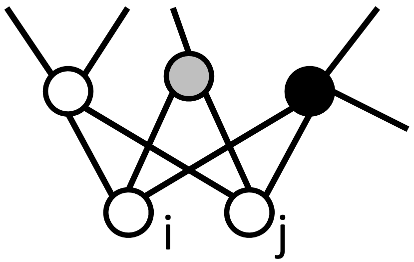
odd # of −



even # of −


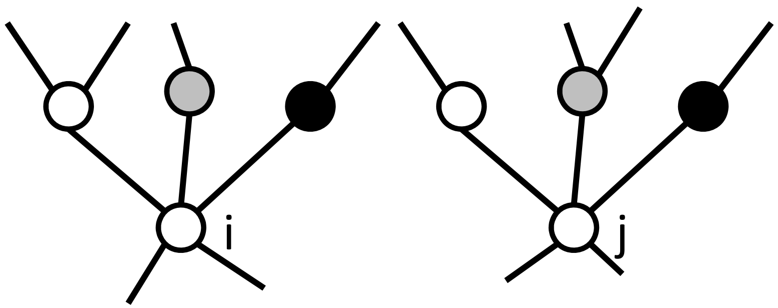

- remove all − edges -> groups connected by +

# similarity between vertices

- structural equivalence
  - sharing many of the same network neighbors
- regular equivalence
  - having neighbors who are themselves similar




structural equivalence




regular equivalence

# cosine similarity

- # of common neighbors of vertices i and j

$$n_{ij} = \sum_k A_{ik} A_{kj} = [\mathbf{A}^2]_{ij}$$

  – normalization is required for the varying degrees of vertices

- cosine similarity: $\cos\theta = \frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}||\mathbf{y}|}$ ith and jth rows of adjacency matrix

$$\sigma_{ij} = \cos\theta = \frac{\sum_k A_{ik} A_{kj}}{\sqrt{\sum_k A_{ik}^2}\sqrt{\sum_k A_{jk}^2}}$$

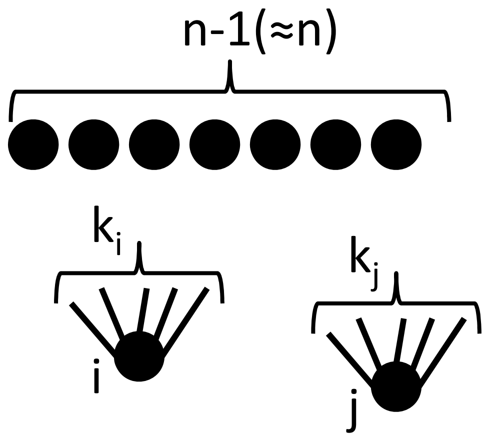
- unweighted simple graph -> $A_{ij}$ = 1 or 0

$A_{ij}^2 = A_{ij}$ for all i and j

$$\sum_k A_{ik}^2 = \sum_k A_{ik} = k_i \quad \Rightarrow \quad \sigma_{ij} = \frac{\sum_k A_{ik} A_{kj}}{\sqrt{k_i k_j}} = \frac{n_{ij}}{\sqrt{k_i k_j}}$$

# Pearson correlation coefficient (1)

- normalize by the expected number of common neighbors if connections are made at random
- vertices i and j have degrees $k_i$ and $k_j$




$$\left.\begin{array}{l} \boxed{\text{probability that the 1st neighbor that j chooses is one of } k_i \text{ vertices -> } k_i/n} \\ \vdots \\ \boxed{\text{probability that the } k_j\text{th neighbor that j chooses is one of } k_i \text{ vertices -> } k_i/n} \end{array}\right\} k_j$$

expected # of common neighbors = $k_i k_j/n$

(We neglect the possibility of choosing the same neighbor twice, since it is small for a large networks)

# Pearson correlation coefficient (2)

- (actual # of common neighbor) – (expected number if chosen randomly)

$$\sum_k A_{ik}A_{kj} - \frac{k_i k_j}{n} = \sum_k A_{ik}A_{jk} - \frac{1}{n}\sum_k A_{ik}\sum_l A_{jl}$$

$$\langle A_i \rangle = n^{-1}\sum_k A_{ik}$$

$$= \sum_k A_{ik}A_{jk} - n\langle A_i\rangle\langle A_j\rangle$$

$$= \sum_k [A_{ik}A_{jk} - \langle A_i\rangle\langle A_j\rangle]$$

$$= \sum_k (A_{ik} - \langle A_i\rangle)(A_{jk} - \langle A_j\rangle)$$

$$= \sum_k A_{ik}A_{jk} - \langle A_j\rangle\sum_k A_{ik} - \langle A_i\rangle\sum_k A_{jk} + n\langle A_i\rangle\langle A_j\rangle$$

$$= \sum_k A_{ik}A_{jk} - n\langle A_i\rangle\langle A_j\rangle - n\langle A_i\rangle\langle A_j\rangle + n\langle A_i\rangle\langle A_j\rangle$$

$$= \sum_k A_{ik}A_{jk} - n\langle A_i\rangle\langle A_j\rangle$$

positive -> i & j are similar

negative -> i & j are dissimilar

# Pearson correlation coefficient (3)

$$\sum_k (A_{ik} - \langle A_i \rangle)(A_{jk} - \langle A_j \rangle) = n \cdot \mathrm{cov}(A_i, A_j)$$

- normalize -> Pearson correlation coefficient

$$r_{ij} = \frac{\mathrm{cov}(A_i, A_j)}{\sigma_i \sigma_j} = \frac{\sum_k (A_{ik} - \langle A_i \rangle)(A_{jk} - \langle A_j \rangle)}{\sqrt{\sum_k (A_{ik} - \langle A_i \rangle)^2} \sqrt{\sum_k (A_{jk} - \langle A_j \rangle)^2}}$$

$$-1 \leq r_{ij} \leq 1$$

# other measures of structural equivalence

- normalize $n_{ij}$ by dividing by (not by subtracting) the expected value ($k_i k_j/n$)

$$\frac{n_{ij}}{k_i k_j / n} = n \frac{\sum_k A_{ik} A_{jk}}{\sum_k A_{ik} \sum_k A_{jk}}$$

alternative to cosine similarity

=1 : # of common neighbors is exactly as expected
>1 : more common neighbors than expected
<1 : less common neighbors than expected
=0 : vertices i & j have no common neighbors
non-negative

- Euclidean distance: # of neighbors that differ between vertices i & j

$$d_{ij} = \sum_k \left( A_{ik} - A_{jk} \right)^2$$

normalize by dividing by its possible maximum value

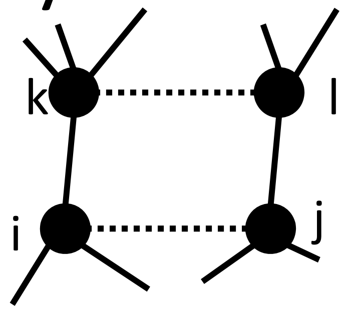
$$\frac{\sum_k \left( A_{ik} - A_{jk} \right)^2}{k_i + k_j} = \frac{\sum_k \left( A_{ik} + A_{jk} - 2 A_{ik} A_{jk} \right)}{k_i + k_j} = 1 - 2 \frac{n_{ij}}{k_i + k_j}$$

# regular equivalence

- define similarity score $\sigma_{ij}$ such that i and j have high similarity if they have neighbors k and l that themselves have high similarity

$$\sigma_{ij} = \alpha \sum_{kl} A_{ik} A_{jl} \sigma_{kl}$$

$$\boldsymbol{\sigma} = \alpha \mathbf{A} \boldsymbol{\sigma} \mathbf{A}$$




- problems
  - not necessary give a high value for self-similarity ($\sigma_{ii}$ )

$$\sigma_{ij} = \alpha \sum_{kl} A_{ik} A_{jl} \sigma_{kl} + \delta_{ij}$$

$$\boldsymbol{\sigma} = \alpha \mathbf{A} \boldsymbol{\sigma} \mathbf{A} + \mathbf{I}$$

# regular equivalence (2)

- another problem: repeated iteration of σ

$$\boldsymbol{\sigma}^{(0)} = 0$$

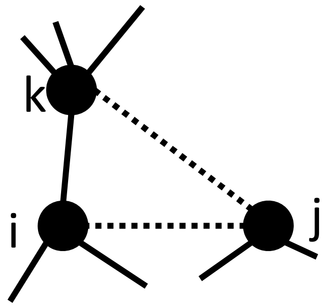
$$\boldsymbol{\sigma}^{(1)} = \mathbf{I}$$

$$\boldsymbol{\sigma}^{(2)} = \alpha\mathbf{A}^2 + \mathbf{I}$$

$$\boldsymbol{\sigma}^{(3)} = \alpha^2\mathbf{A}^4 + \alpha\mathbf{A}^2 + \mathbf{I}$$

sum over even powers only.
why not consider paths of all lengths?

- better definition: i and j are similar if i has a neighbor k that is itself similar to j




$$\sigma_{ij} = \alpha\sum_k A_{ik}\sigma_{kj} + \delta_{ij}$$

$$\boldsymbol{\sigma} = \alpha\mathbf{A}\boldsymbol{\sigma} + \mathbf{I}$$

$$\boldsymbol{\sigma} = \sum_{m=0}^{\infty}(\alpha\mathbf{A})^m = (\mathbf{I} - \alpha\mathbf{A})^{-1}$$

# regular equivalence (3)

$$\boldsymbol{\sigma} = \sum_{m=0}^{\infty} (\alpha \mathbf{A})^m = (\mathbf{I} - \alpha \mathbf{A})^{-1}$$

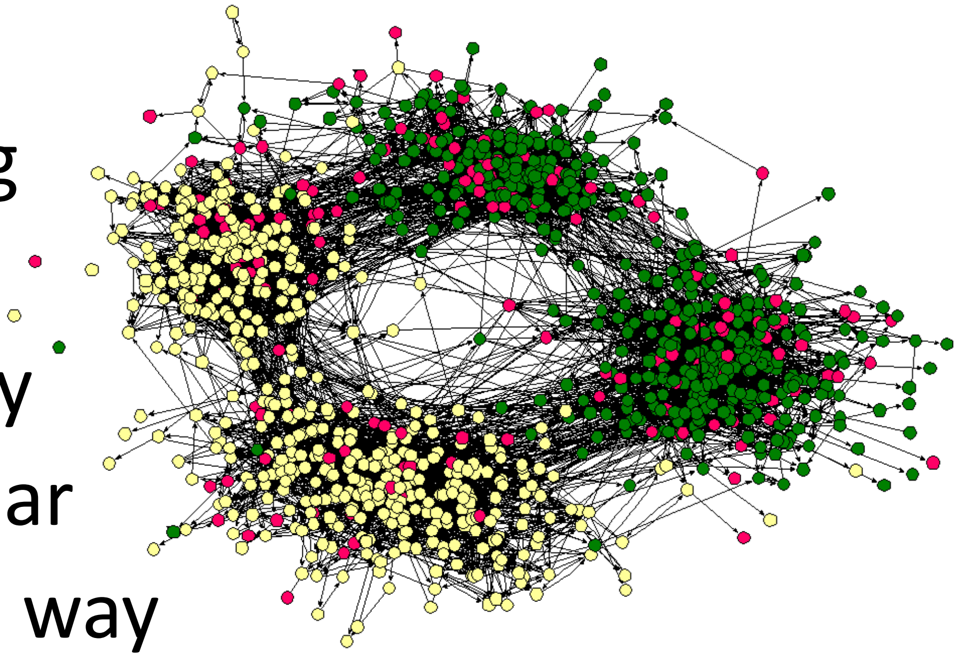
- longer paths will get less weight than shorter ones
- closely related to Katz centrality
- a generalization of structural equivalence
  - structural equivalence : # of paths of length two
  - regular equivalence : # of paths of all length
- variation
  - penalize vertices of high degree

$$\sigma_{ij} = \frac{\alpha}{k_i} \sum_k A_{ik} \sigma_{kj} + \delta_{ij} \qquad \boldsymbol{\sigma} = \alpha \mathbf{D}^{-1} \mathbf{A} \boldsymbol{\sigma} + \mathbf{I}$$

$$\boldsymbol{\sigma} = (\mathbf{I} - \alpha \mathbf{D}^{-1} \mathbf{A})^{-1} = (\mathbf{D} - \alpha \mathbf{A})^{-1} \mathbf{D}$$

# friendship network at a US high school

- the split from left to right is clearly primarily along lines of race
- people have a strong tendency to associate with others whom they perceive as being similar to themselves in some way ->“homophily”,“assortative mixing”



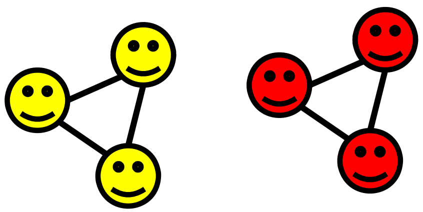
Yellow - White Race
Green - Black Race
Pink - Other

http://www-personal.umich.edu/~mejn/networks/

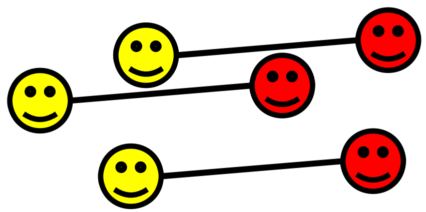
# assortative mixing by enumerative characteristics

- vertices are classified according to some enumerative values
  - nationality, race, gender, language,...
- network is assortative if a significant fraction of the edges run between vertices of the same type

not good measure : the fraction is 1 if all vertices belong to the same single type



assortative



disassortative

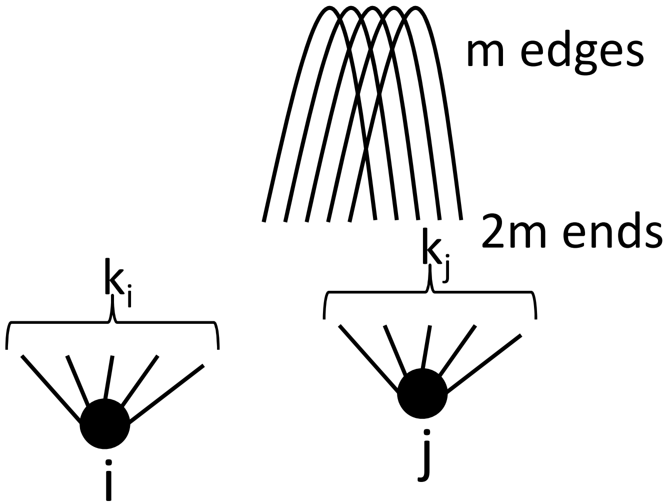
# better definition of assortative mixing

- (fraction of edges that run between vertices of the same type)-(expected fraction of edges if they are positioned at random)
- $c_i$ : class(type) of vertex i $(1,..,n_c)$
- (# of edges that connect the vertices of the same type) : $\sum_{edges(i,j)} \delta(c_i, c_j) = \frac{1}{2}\sum_{ij} A_{ij}\delta(c_i, c_j)$

C: class type
delta: 0 or 1(the same class or not for i&j)
A: adj matrix of these two vertices

# expected # of edges if connections are at random

- (expected # of edges between i and j if they are positioned at random) :




expected # of edges

probability that the other end of a particular edge = $k_j/2m$

counting all $k_i$ edges attached to i, the total expected # of edges between i and j = $k_i k_j/2m$

- (expected # of edges between all pairs of vertices of the same type) : $\frac{1}{2}\sum_{ij}\frac{k_i k_j}{2m}\delta(c_i, c_j)$

# modularity (1)

- (# of edges that run between vertices of the same type)-(expected # of edges if they are positioned at random)

$$\frac{1}{2}\sum_{ij} A_{ij}\delta(c_i, c_j) - \frac{1}{2}\sum_{ij}\frac{k_i k_j}{2m}\delta(c_i, c_j) = \frac{1}{2}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(c_i, c_j)$$

- divided by the # of edges

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(c_i, c_j)$$

- modularity: measure of the extent to which like is connected to like in a network
  - less than 1
  - positive if there are more edges than expected, negative if there are less edges

# modularity (2)

- modularity matrix $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$
  - used for community detection
- normalizing modularity :assortative coefficient

$$Q_{\max} = \frac{1}{2m}\left(2m - \sum_{ij} \frac{k_i k_j}{2m} \delta(c_i, c_j)\right) \qquad \frac{Q}{Q_{\max}} = \frac{\sum_{ij} (A_{ij} - k_i k_j / 2m)\delta(c_i, c_j)}{2m - \sum_{ij} (k_i k_j / 2m)\delta(c_i, c_j)}$$

normalized version is rarely used

# modularity with R+igraph

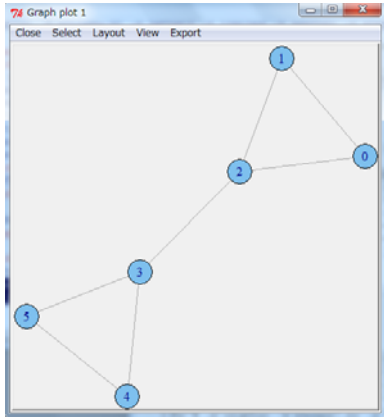
modularity can be used to evaluate partition

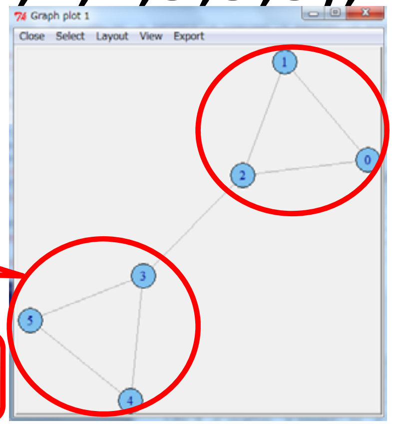
```
> g0 <- graph(c(0,1,1,2,2,0,2,3,3,4,4,5,5,3), directed=FALSE)
> tkplot(g0)
> modularity(g0,c(0,0,0,1,1,1))
[1] 0.3571429
```

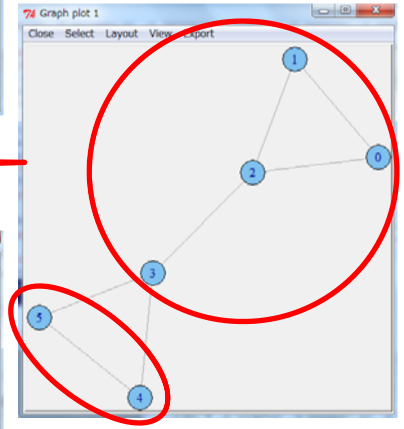
dense inside, sparse outside
→ high value

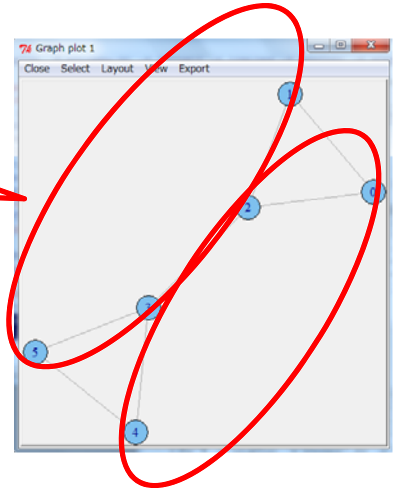
```
> modularity(g0,c(0,0,0,0,1,1))
[1] 0.1224490
> modularity(g0,c(0,1,0,1,0,1))
[1] -0.2142857
```

sparse inside, dense outside
→ low value



# alternative form of modularity

$$e_{rs} = \frac{1}{2m}\sum_{ij} A_{ij}\delta(c_i, r)\delta(c_j, s)$$

fraction of edges that join vertices of type r to vertices of type s

$$a_{rs} = \frac{1}{2m}\sum_{i} k_i\delta(c_i, r)$$

fraction of ends of edges attached to vertices of type r

$$\delta(c_i, c_j) = \sum_{r}\delta(c_i, r)\delta(c_j, r)$$

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\sum_{r}\delta(c_i, r)\delta(c_j, r)$$

$$= \sum_{r}\left[\frac{1}{2m}\sum_{ij} A_{ij}\delta(c_i, r)\delta(c_j, r) - \frac{1}{2m}\sum_{i} k_i\delta(c_i, r)\frac{1}{2m}\sum_{j} k_j\delta(c_j, r)\right]$$

$$= \sum_{r}(e_{rr} - a_r^2)$$

useful when we have no explicit data on vertex degrees

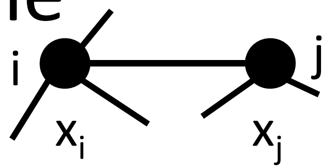
# assortative mixing by scalar characteristics

- vertices are classified according to some scalar values (age, income,...)
  - “assortatively mixed by age”, “stratified by age”
- the same approach as enumerative values will miss much of the point about scalar characteristics
  - group vertices into bins (age 0-9,10-19,20-29,...) and treat the bins as separate type

(age 8 and 9) are similar, but (age 9 and 10) are entirely dissimilar

# covariance measure

- $x_i$ : value of vertex i of the scalar quantity
- consider the pairs of values $(x_i, x_j)$ for the vertices at the end of each edge (i,j)




$$\mu = \frac{\sum_{ij} A_{ij} x_i}{\sum_{ij} A_{ij}} = \frac{\sum_i k_i x_i}{\sum_i k_i} = \frac{1}{2m} \sum_i k_i x_i$$

μ:mean of value of $x_i$ at the end of an edge (average over edges, not vertices)

- covariance of $x_i$ and $x_j$ over edges

$$\mathrm{cov}(x_i, x_j) = \frac{\sum_{ij} A_{ij} (x_i - \mu)(x_j - \mu)}{\sum_{ij} A_{ij}} = \frac{1}{2m} \sum_{ij} A_{ij} (x_i x_j - \mu x_i - \mu x_j + \mu^2)$$

$$= \frac{1}{2m} \sum_{ij} A_{ij} x_i x_j - \mu^2$$

$$= \frac{1}{2m} \sum_{ij} A_{ij} x_i x_j - \frac{1}{(2m)^2} \sum_{ij} k_i k_j x_i x_j$$

$$= \frac{1}{2m} \sum_{ij} \left( A_{ij} - \frac{k_i k_j}{2m} \right) x_i x_j$$

positive if values at either end of an edge tend to be both large or both small

# normalizing covariance

- $\mathrm{cov}(x_i,x_j)$ is maximum when $x_i=x_j$

$$\frac{1}{2m}\sum_{ij}\left(A_{ij}-\frac{k_i k_j}{2m}\right)x_i^2=\frac{1}{2m}\sum_{ij}\left(k_i\delta_{ij}-\frac{k_i k_j}{2m}\right)x_i x_j$$

$$\boxed{\sum_{ij}k_i\delta_{ij}x_j=\sum_{i=j}k_i\delta_{ij}x_j+\sum_{i\neq j}k_i\delta_{ij}x_j=\sum_{i}k_i x_i=\sum_{ij}A_{ij}x_i}$$

- normalize covariance

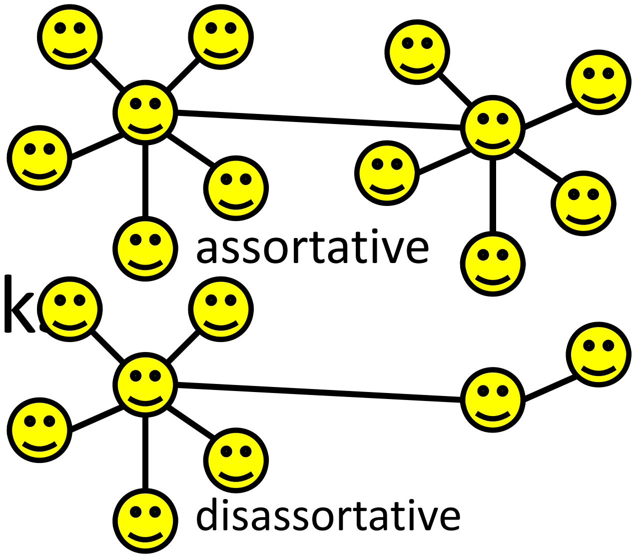
$$r=\frac{\sum_{ij}(A_{ij}-k_i k_j/2m)x_i x_j}{\sum_{ij}(k_i\delta_{ij}-k_i k_j/2m)x_i x_j}$$

$$-1\leq r\leq 1$$

# assortative mixing by degree

- assortative: high-degree vertices connect to other high-degree vertices
- core/periphery structure :

common feature of social networks




- covariance

$$\mathrm{cov}(k_i, k_j) = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)k_i k_j$$

- correlation coefficient (assortativity coefficient)

$$r = \frac{\sum_{ij}(A_{ij} - k_i k_j / 2m)k_i k_j}{\sum_{ij}(k_i \delta_{ij} - k_i k_j / 2m)k_i k_j}$$