Final Submission - Nov 20

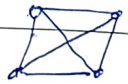
Oct - 10
abstract submission - • working title
• both groupmembers
• abstract - describe the dataset goal/aim (questions you are trying to solve)
• Type of analysis
• Source citation

DOMS | Page No. | Date

Python - Networkx package.
Zachary Karate Club dataset - 34 nodes, 78 edges
↳ very strong dataset (download .xml from Newman).
- importance / centrality
- density
- Adjacency matrix
- probability distribution

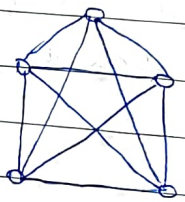
Group of nodes (Cohesive behavior)

compactness → everybody knows everyone else



reachability → within a few hops its possible to reach all other nodes.

Density → high density within a component, low density b/w 2 components. Used in clustering algorithms.
- microscopic - single level
- mesoscopic - a small section of the system
- macroscopic - entire system



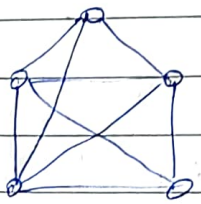
→ close tied structure
every node is connected with every other node
complete sub-graph. b/a Cliques

maximal subset of vertices which forms a complete graph.

to find all the cliques in a graph → the problem is NP-hard.

Instead of complete subgraph, we look for k-plex.

1-plex is a clique.



→ 2 plex

atleast connected with 3 other vertices. 5 nodes

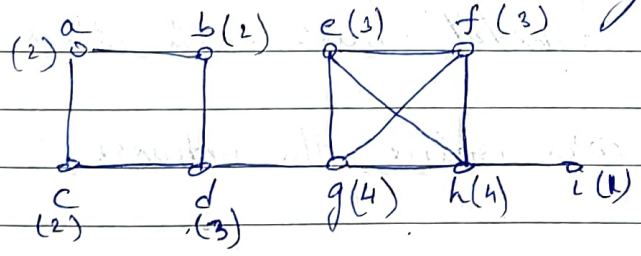
$\therefore k = 2$

$5 - 2 = 3$

n ↙ → k → no. of edges from each node.

k-core →

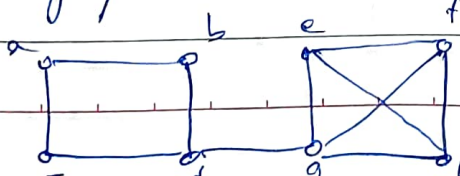
remove node with the lowest degree.
after doing that if any node has degree less than or equal to the removed node, remove that too. — recursive process.
all the nodes removed belong to the same k-core value.

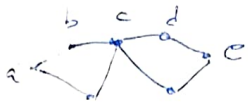


removing vertex i →

vertex i will belong to k=1.

new graph →

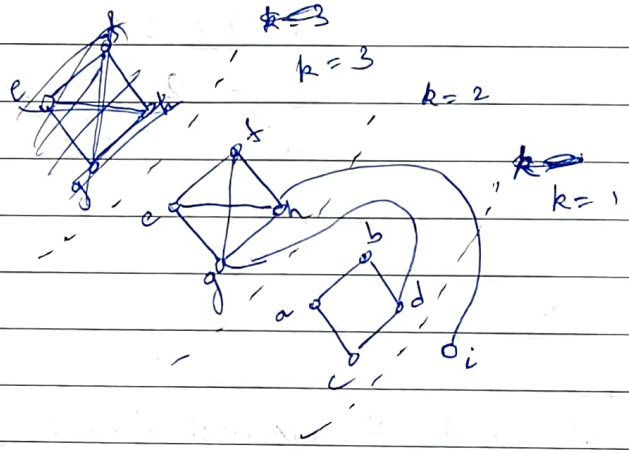




DOMS | Page No.
Date / /

next iteration we can remove a vertex a.

iteration	k = 1	k = 2	k = 3
1	{i}		
2		{a}	
3		{a, b, c}	
4		{a, b, c, d}	
5			{e, f, g, h}



as we move into the core, the density of graph increases. k = 3 → more towards the core.

We are trying to find group of nodes that are cohesive — highly connected with each other.

as we increase the core value, the network becomes more & more dense.

Here we have a group of nodes based centrality.

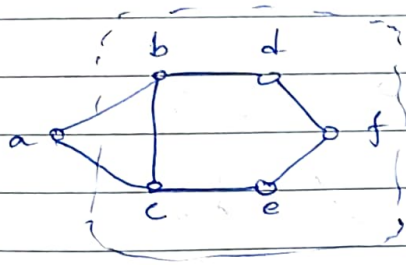
high degree nodes inside a higher core value are more influencial.

k-core allows us to identify more influencial spreaders. Node ~~with~~ inside a higher core value will be more influencial than a ~~the~~ node at lower core value no matter the degree of those nodes. ∴ Hubs are not the most important.

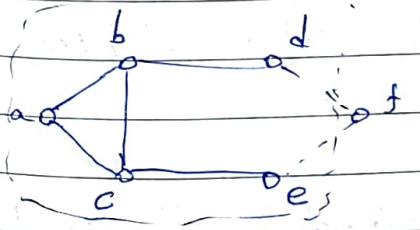
Distance based grouping →
- k-clique
- k-clan
- k-club

k-clique →

it puts all those verticise that are at a ditance k with each other.



every node is at a max. of dist. 2 from each other.
∴ they can be grouped at 2-clique.
distance is shortest path.



problem with k-clique.

→ this 2 clique includes a path from e to d passing through f. but f is not part of the said 2-clique.

k-component

DOMS | Page No.
Date / /

k-clan →

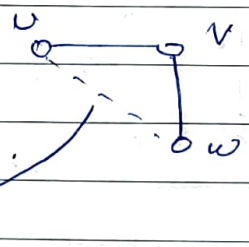
- extract the k-clique
- check the diameter of the k-clique graph.
- if diameter ≤ k
  then k-clan
  else
  not k-clan

(this is to avoid the problem with k-clique)

k-club →

- if diameter of a ~~max~~ maximal subset of a graph is k, then k-club.
  ↓
  no more nodes can be added.

Transitivity →
(Clustering)



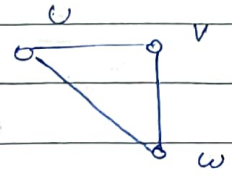
u — v
v — w
are u & w related?

(especially in the content of social networks)

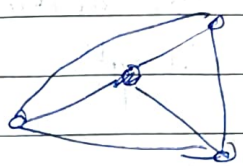
- Social networks have much higher clustering than any other network.

- If all your friends are friends with each other, then it will be a complete graph.

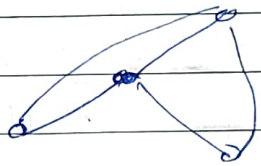
- each of these Triangles increase the transitivity



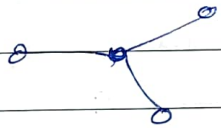
- case where all my friends are friends with each other (perfectly transitive)



- case where not all my friends are friends with each other



- case where none of my friends are friends with each other.



The total no. of triangles present in the network gives us how many of the connections are transitive.
We normalize the no. of triangles obtained by the total no. of triangles possible in the network.

This normalized quantity is called the clustering coefficient.

If clustering coefficient = 1, then completely connected graph. — perfect Transitivity

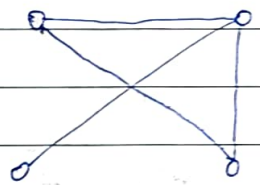
lack of transitivity → clustering coefficient = 0 (star graph).

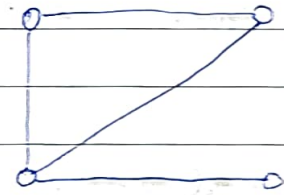
$$C = \frac{\text{no. of closed paths of length 2}}{\text{no. of paths of length 2}}$$

for every triangle, there are 6 paths of length 2 - (both sides)

$$C = \frac{6 \times \text{no. of } \Delta\text{'s}}{\text{no. of paths of length 2}}$$

$$C = \frac{3 \times \text{no. of } \Delta\text{'s}}{\text{no. of connected triplets}} \rightarrow \text{easiest to implement}$$





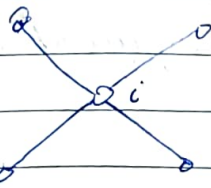
probability of connection of 2 nodes $= \frac{|E|}{\binom{n}{2}}$

local clustering coeff. $\rightarrow$

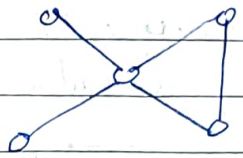
$$C_i = \frac{\text{no. of edges b/w the neighbours of } i}{\binom{k_i}{2}}$$

k_i : degree of vertex i

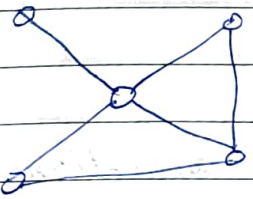
$\binom{k_i}{2}$ $\rightarrow$ no. of edges possible b/w neighbours



$C_i = \frac{0}{{}^4C_2} = 0$



$C_i = \frac{1}{{}^4C_2} = \frac{1}{6}$



$C_i = \frac{2}{{}^4C_2} = 1/3$

Local clustering coeff. is also a centrality measure, ∵ it can be calculated for each node.

$\langle C \rangle = \frac{1}{N} \sum_i C_i$ ⟶ clustering coeff. for the entire network

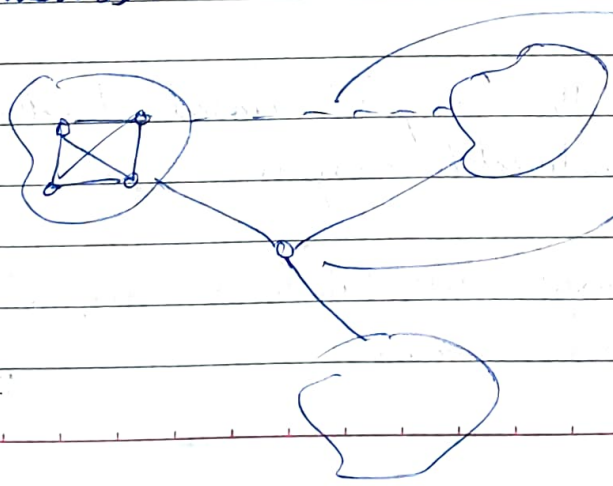
$= C_{WOS}$ (Watts & Strogatz)

N ↳ no. of nodes (to normalize).

the more correct definition of clustering coeff. is the previous one, but this is more used.

structural holes → absence of links to connect clusters.

highly correlated with the idea of Betweenness Centrality



→ degree based

Clustering however is computationaly efficient as compared to Betweenness centrality.

→ path based.

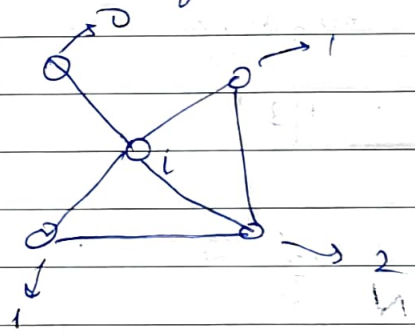
Redundancy → The redundancy

mean no. of connections b/w neighbours.

Redundancy (R_i) is the mean no. of connections b/w neighbours of i.

$$R_i = \frac{1}{4}[0+1+2+1]$$

4 neighbours



$$R_i = 1$$

Total no. of connections b/w neighbours of verten i

$$= \frac{k_i R_i}{2}$$

k_i → degree of verten i

——→ all connections are counted twice.

$$C_i = \frac{k_i R_i / 2}{\binom{k_i}{2}} = \frac{R_i}{k_i - 1}$$

→ total connections

triangles are the simplest closed structure possible in ~~undirected~~ undirected graphs.

For directed graph, we can have simplest closed structure with 2 nodes.

DOMS | Page No.
Date / /

Reciprocity → counts the total no. of such structures & then normalizes it by dividing by total no. of edges.

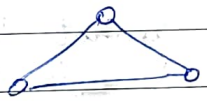
$R_i = \frac{1}{m} \sum_j A_{ij} A_{ji}$ if i, j are nodes.

for all the vertices → $R_i = \frac{1}{m} Tr[A^2]$

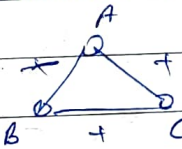
Signed Networks

relationship → friends & enemy

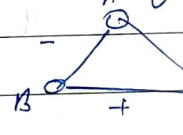
again the simplest structure that can be formed →



4 of such networks are possible for signed structures →

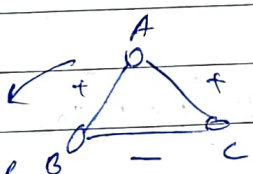


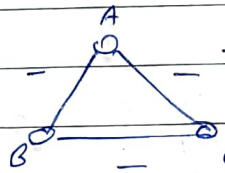
friendship network



→ enemy of an enemy is a friend.

not a structurally stable scenario



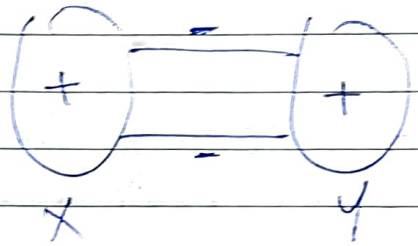


→ structurally unstable

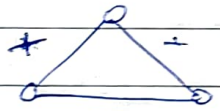
1st 2 are structurally stable.

Cartwright - Harary Theorem →

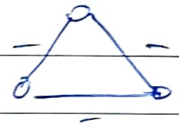
For a structurally stable complete graph, we can divide the graph into 2 groups where all the links within a group are friend links & links b/w 2 groups are enemy links.



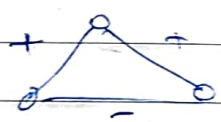
Strong Structural Balance



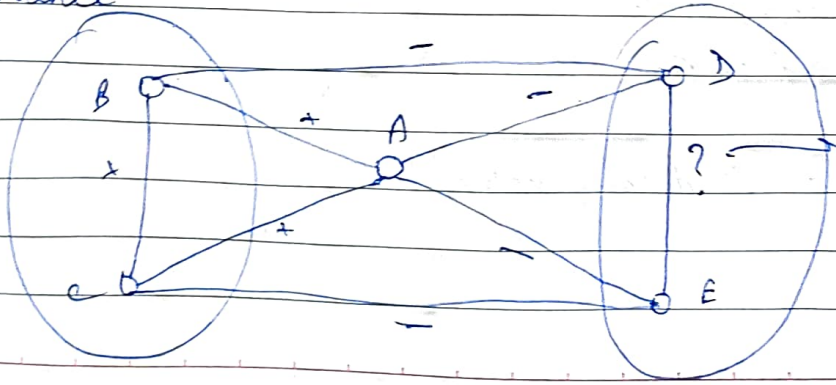
→ in time the negative is influenced to be +ve.



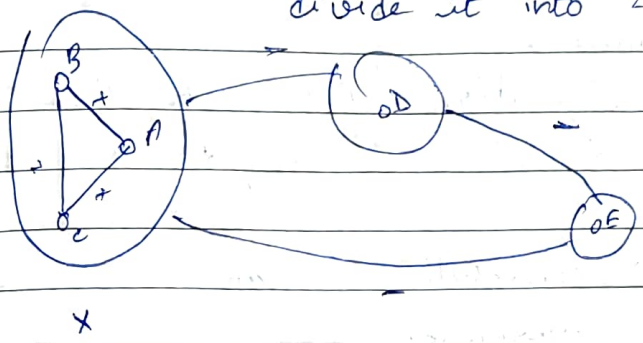
→ ~~cannot be influenced~~
∴ leads to
weak structural balance

assumption is that  would not exist.

Now if the network permits weak structural balance



if relationship b/w D & E is -ve, then we divide it into 2 clusters



X

∴ For weak structural balance we can divide the graph into multiple structures instead of 2.
(graph coloring problem)
↳ This is used for link prediction.

Similarity

Nodes that are connected to each other in networks tend to be similar in their features.

assortativity - measuring similarity - high degree connected to high degree
disassortative - converse of assortativity

How do we quantify similarity?
We determine if any similarity is present by checking if the similarity obtained is greater than the similarity of a completely random network.

→ Random Variable
→ Probability Distribution (PDF, CDF, etc)
→ Moments $\langle x^n \rangle = \sum_x x^n p\{x = n\}$

$n = 1$ → expectation (mean)
$n = 2$ → variance.
$n = 3$ → skewness
$n = 4$ → Kurtosis

→ Limit Theorems
- Markov's inequality
- Chebyshev's inequality
- ~~Weak~~ Strong Law of Large no.'s
- Weak law of Large no.'s
- Central Limit Theorem.

→ Jointly distributed Random Variables (Covariance, Indep. Correlation).

Syllabus In-Sem →

Ch.6 → 6.1, 6.2, 6.3, 6.4, 6.4.1, 6.4.2, 6.6, 6.9, 6.10, 6.10.1, 6.11, 6.11.1, 6.14

Problems → 6.1, 6.2, 6.3, 6.4, 6.8

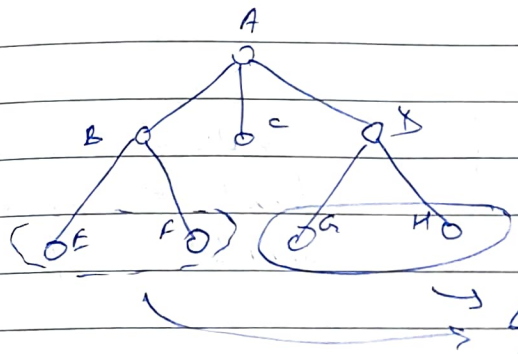
Ch.7 → 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7

Problems → 7.1, 7.2, 7.3, 7.4

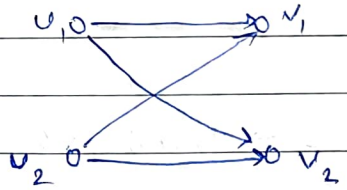
Core periphery structure → If high degree nodes are connected with other high degree nodes & low degree nodes are connected with other low degree nodes.

Disassortative network → high degree nodes connected with low degree nodes & vice-versa.

2 ways in which one can think of similarity →
- Structural Equivalence
- Regular Equivalence → has to do with ~~goals~~ roles.

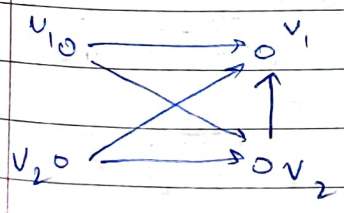


→ structurally equivalent
(share the same neighbours)
(complete overlap b/w neighbours of a set of nodes)



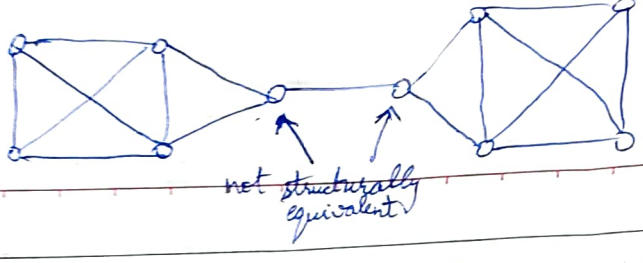
$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_1 \\ v_2 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_1 & v_2 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

If 2 vertices are structurally equivalent then the rows & columns of the adjacency matrix with respect to the 2 vertices will be identical.
v_1 & v_2 are structurally equivalent.



$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_1 \\ v_2 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_1 & v_2 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \end{array}$$

v_1 & v_2 are not structurally equivalent here.
a little change leads to a change in structural equivalence.



DOMS | Page No.
Date / /

Even a complete graph is not structurally equivalent.
If we add self loops in a complete graph, then all nodes will be structurally equivalent.

$\sigma_{ij} = [\quad]$ matrix element will denote the similarity in neighbours b/w nodes i

n_{ij} = no. of common neighbours of vertex i & j.
$= \sum_k A_{ik} A_{kj}$ → for undirected graph.

now we normalize n_{ij} to get σ_{ij}.

Partial Overlap →

(1) Jaccard Similarity

$$\frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$$ N → neighbours.

$$= \frac{n_{ij}}{\sum_k A_{ik} + \sum_k A_{jk} - n_{ij}} = \frac{n_{ij}}{k_i + k_j - n_{ij}} = \sigma_{ij}$$ k_i, k_j → degrees.

(2) Cosine Similarity

A → adjacency matrix

$\bar{n} \cdot \bar{y} = |\bar{n}||\bar{y}| \cos\theta$

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

$= \frac{n_{ij}}{\sqrt{k_i k_j}}$

$A_{ik} = [\quad]$
$A_{jk} = [\quad]$

↓
cosine similarity

(3) Pearson

(4) Hamming Dist.

Expected overlap if the connection is purely at random.

Now we calculate the actual overlap.
If the value is greater than the expected value, then there is a preference. Likes connected likes.
If the value is less than the expected value then there is a preference of likes being connected to dislikes.

Let's say we have n R.V. $\rightarrow$

$$X_1 \; - \; - \; - \; X_n$$

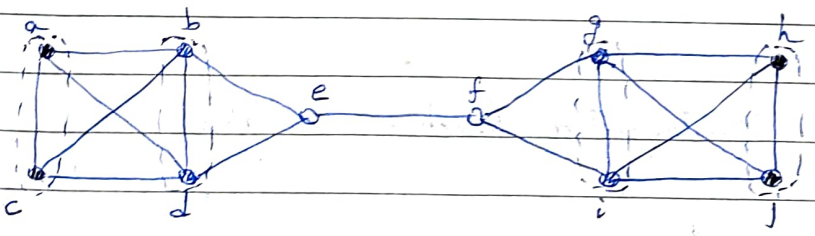
$$E[X] = \frac{1}{n} \sum_i x_i$$

$$Var[X] = E[(X - E[X])^2]$$

$$Y_1 \; - \; - \; - \; Y_n$$

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])]$$

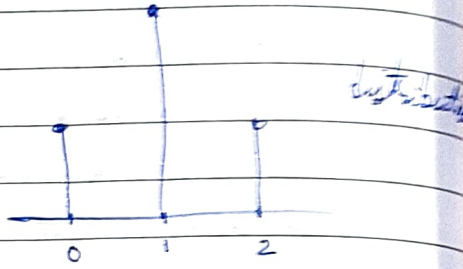
if both X & Y are on the same side of the mean then product will be +ve. $\rightarrow$ +ve relationship.
if they are on opposite side of the mean then product will be -ve $\rightarrow$ -ve relationship.



Statistical Methods (Comparision)

Covariance

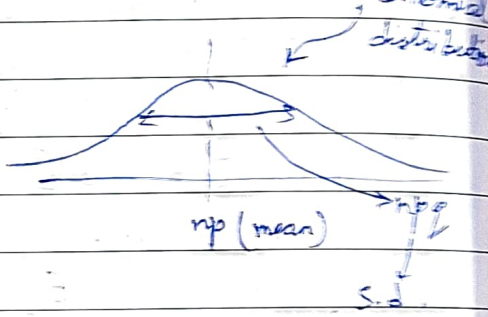
X : no. of heads
toss a coin twice
$P\{X=0\} = 1/4$
$P\{X=1\} = 2/4$
$P\{X=2\} = 1/4$



if we toss the coin n times, X : no of heads

n flips x heads

$$\binom{n}{x} p^x (1-p)^{n-x}$$



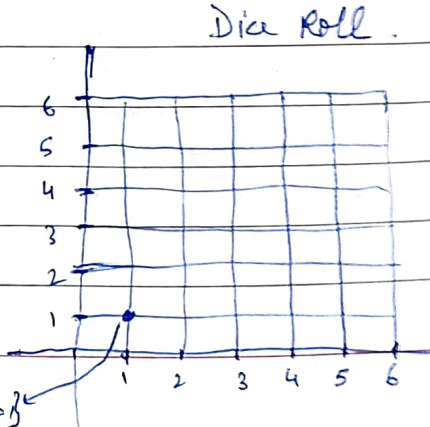
$$E[X] = \sum_x x \cdot p\{X=x\}$$

$$E[X^2] = \sum_x x^2 p\{X=x\}$$

$$Var(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

Jointly Distributed Random Variable

$P\{X=x, Y=y\}$



$$Cov(X,Y) = E[(X - E[X])(Y - E[Y])]$$

↳ Statistical dependence b/w X & Y

if Cov is +ve — +ve dependence
if cov is -ve — -ve dependence.
if cov is 0 — random

maximum possible value of Cov is obtained when X & Y are essentially the same (obtain same distribution)
∴ we can normalize Cov(x,Y) by variance.

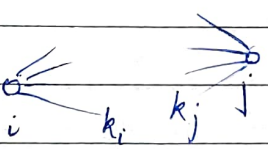
∴ ~~$E[(X - E[X])$~~

$$\rho = \frac{E[(X - E[X])(Y - E[Y])]}{\sqrt{Var(X)\,Var(Y)}}$$

Pearson correlation coefficient

Expected number of common neighbours if the connections are at random

probability of picking one of the neighbours of j $\Rightarrow \frac{k_j}{n-1} \approx \frac{k_j}{n}$



this is being calculated for 1 edge.

large graph

∴ expected no. of common neighbours = $\frac{k_i k_j}{n}$
if the connections are drawn at random

~~if random~~

$$n_{ij} - \frac{k_i k_j}{n}$$

+ve → likes connected
-ve → dislikes connected
about 0 → random

$$\sigma_{ij} = n_{ij} - \frac{k_i k_j}{n} = \left(\sum_k A_{ik} A_{kj}\right) - \frac{k_i k_j}{n}$$

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

$$\boxed{\langle A_i \rangle = \frac{1}{n} \sum_k A_{ik}}$$

$A_{ik} \rightarrow [\quad\quad] \quad \langle A_i \rangle$
$A_{jk} \rightarrow [\quad\quad] \quad \langle A_j \rangle$

∵ undirected graph $A_{ik} = A_{ki}$
$A_{ik}^2 = A_{ik}$ (0 & 1st only)
$k_i = \sum_k A_{ik}$

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

$$\left(\sum_k A_{ik} A_{kj}\right) - \frac{k_i k_j}{n} \cdot \frac{n}{n} = \left(\sum_k A_{ik} A_{kj}\right) - n \langle A_i \rangle \langle A_j \rangle$$

$$= \sum_k \left[A_{ik} A_{kj} - \langle A_i \rangle \langle A_j \rangle\right]$$

$$n\langle A_i\rangle\langle A_j\rangle = n \cdot \frac{1}{n}\sum_k A_{ik}\langle A_j\rangle$$

$$= \sum_k A_{ik}\langle A_j\rangle$$

DOMS | Page No.
Date / /

$$= \sum_k \left[A_{ik}A_{kj} - \langle A_i\rangle\langle A_j\rangle + \langle A_iA_j\rangle - \langle A_i\rangle\langle A_j\rangle \right]$$

$\langle A_i\rangle\langle A_j\rangle \rightarrow \sum_k A_{ik}\langle A_j\rangle$

$\langle A_iA_j\rangle - \langle A_i\rangle\langle A_j\rangle \rightarrow \sum_k A_{jk}\langle A_i\rangle$

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

$$\sigma_{ij} = \sum_k (A_{ik} - \langle A_i\rangle)(A_{jk} - \langle A_j\rangle) \longleftrightarrow \text{Covariance.}$$

$A =$ [ ] $\rightarrow$ $\sigma =$ [ ]

we can make a heat map for σ.

$\frac{n_{ij}}{k_i k_j / n}$ $\Rightarrow$ if 1 $\rightarrow$ random
if > 1 $\rightarrow$ similarity
0 $\rightarrow$ not connected
if < 1 $\rightarrow$ dislikes connected

Hamming Distance $\rightarrow$

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

$$k_{nn}(i) = \frac{1}{k_i}\sum_j A_{ij}k_j$$

$k_{nn}(i)$: i → nearest neighbour

k_j → degrees of the neighbours only for immediate neighbours.

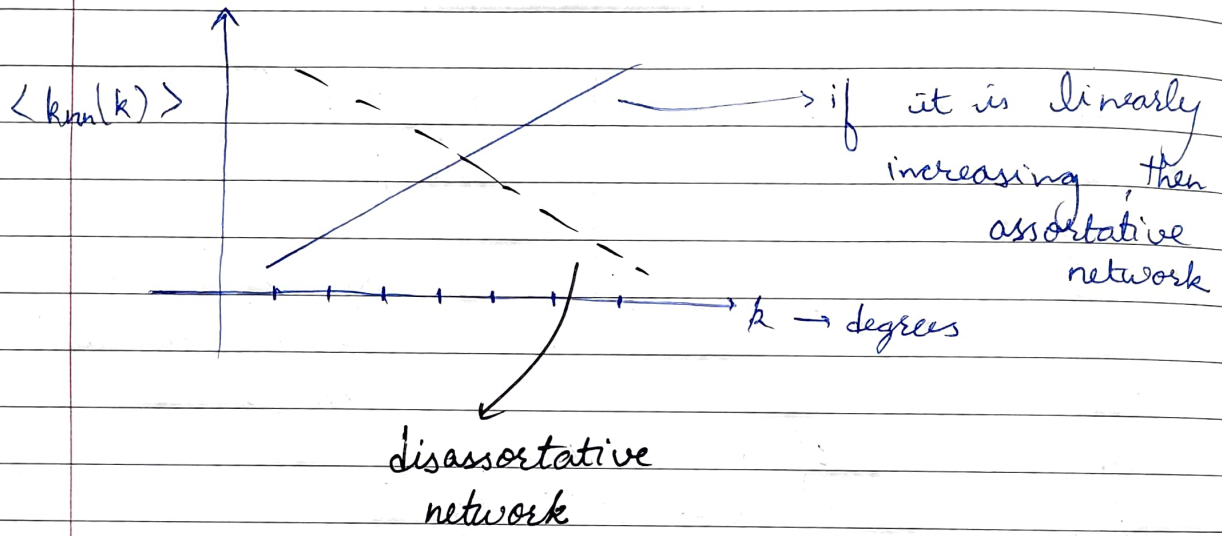
$\langle k_{nn}(k)\rangle$ → avg. of $k_{nn}(i)$ of all vertices with degree k.

$$\langle k_{nn}(k) \rangle = \frac{1}{N_k} \sum_{i=1}^{n} k_{nn}(i)\, \delta_{k_i, k}$$

k_i = degree of vertex i.

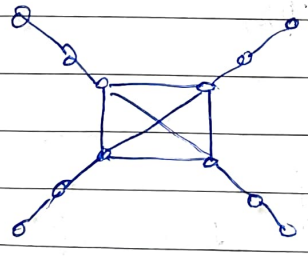
$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \rightarrow \text{Kronecker delta function}$$

network assortative or not ?



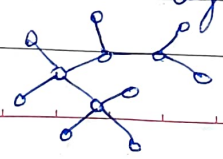
assortative network → core-periphery structure
(core is densely connected & sparsely connected periphery).

eg →



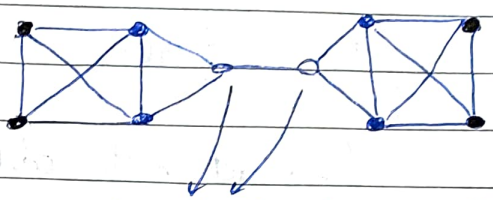
disassortative network → low degree nodes connect to high degree nodes. Many star nodes.

eg →



Regular Equivalence

There need not be overlap b/w the neighbours of the nodes.

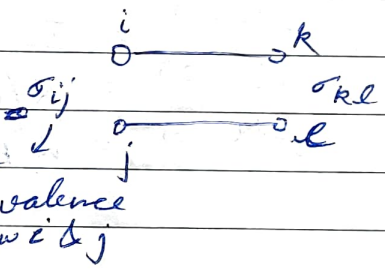


Regularly Equivalent

calculated based on eigen vectors.

Two vertices are regularly equivalent if they are connected to vertices that are regularly equivalent.

σ_{ij} will depend on σ_{kl}.



$$\sigma_{ij} \propto \sum_{k,l} A_{ik} A_{jl} \sigma_{kl}$$

$$\sigma_{ij} = \alpha \sum_{k,l} A_{ik} A_{jl} \sigma_{kl} + \delta_{ij}$$ (just to ensure that we get a larger value when $i=j$).

$$\boxed{\sigma = \alpha A \sigma A}$$ $A \rightarrow$ symmetric (undirected graph).

$$\boxed{\sigma = \alpha A \sigma A + I}$$

we want α to be smaller than 1, otherwise σ will diverge.

if $\sigma^{(t=0)} = \vec{0}$

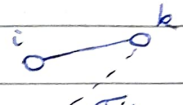
$\sigma(t=1) = I$

$\sigma(t=2) = \alpha A^2 + I$

$\sigma(t=3) = \alpha^2 A^4 + \alpha A^2 + I$

⋮

another formalism →



$$\sigma_{ij} = \alpha \sum_{i,k} A_{ik} \sigma_{kj} + \delta_{ij}$$

(we get both odd & even powers of A here)
∴ paths of all lengths could be taken into consideration.

$$\boxed{\sigma = \alpha A \sigma + I}$$

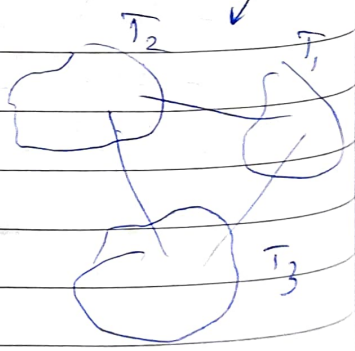
$$\sigma(t=0) = \bar{0}$$
$$\sigma(t=1) = I$$
$$\sigma(t=2) = \alpha A + I$$
$$\sigma(t=3) = \alpha^2 A^2 + \alpha A + I$$

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

(almost similar to Katz centrality)
& α should be smaller than the largest eigen value of the adjacency matrix

- In real networks are the connections based on similarity in terms of attributes? Homophily / assortative
- How do we quantify it?
- Which ones in which groups?

graph partioning



(community detection)

BETWEENESS

DOMS | Page No.
Date / /

m_c -> edges that connect nodes with the same attribute

$\langle m_c \rangle$ -> expected no. of if the nodes were connected at random

m -> total no. of links.

c : class, cluster, group, attribute, etc.

$$\frac{m_c - \langle m_c \rangle}{m} = Q$$

↳ Modularity

we try to maximize modularity in order to obtain a good optimal solution.

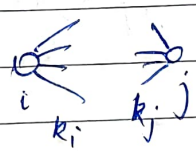
let C_i represent class $i = 1, \ldots n$.

m_c = number of links between nodes with the same attribute

$$m_c = \frac{1}{2} \sum_{ij} A_{ij} \delta_{c_i, c_j}$$

$\langle m_c \rangle$: Expected number of links between nodes of same type if connections are drawn at random.

$$\langle m_c \rangle = \sum_{i,j} \frac{k_i k_j}{2m} \delta_{c_i, c_j}$$



$\frac{k_i k_j}{2m}$ -> total no. of edges b/w vertex i & j.

probability that edge from vertex i will connect to vertex j = $\frac{k_j}{2m-1} \approx \frac{k_j}{2m}$

$$\langle m_c \rangle = \frac{1}{2} \sum_{ij} \frac{k_i k_j}{2m} \delta_{c_i, c_j}$$

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

Types / Networks — Matrix Representation
└ centrality (node level description)
group of nodes.

Power - law

$p_k \sim k^{-\alpha}$ ~~(a node having degree k)~~

(probability of a node having degree k)

$p_k = c k^{-\alpha}$

$\sum_k p_k = 1$

$\sum_k k^{-\alpha} = \zeta(\alpha)$

Discrete Continuum

Riemman - Zeta function

$p(k) = c k^{-\alpha}$

normalization →

$$\int p(k)\,dk = c \int_{k_{min}}^{\infty} k^{-\alpha} dk = \frac{c k^{1-\alpha}}{1-\alpha} \Big|_{k_{min}}^{\infty}$$

$(\alpha > 1)$

$p_k = \frac{N_k}{N}$

$$\frac{c\, k_{min}^{1-\alpha}}{\alpha - 1} = 1$$

$c = (\alpha - 1)\, k_{min}^{\alpha - 1}$

else cannot solve the problem, it will diverge

$\log(p_k)$ → linear if power law distribution

$\log(k)$