

MODULE 1 : NETWORKS

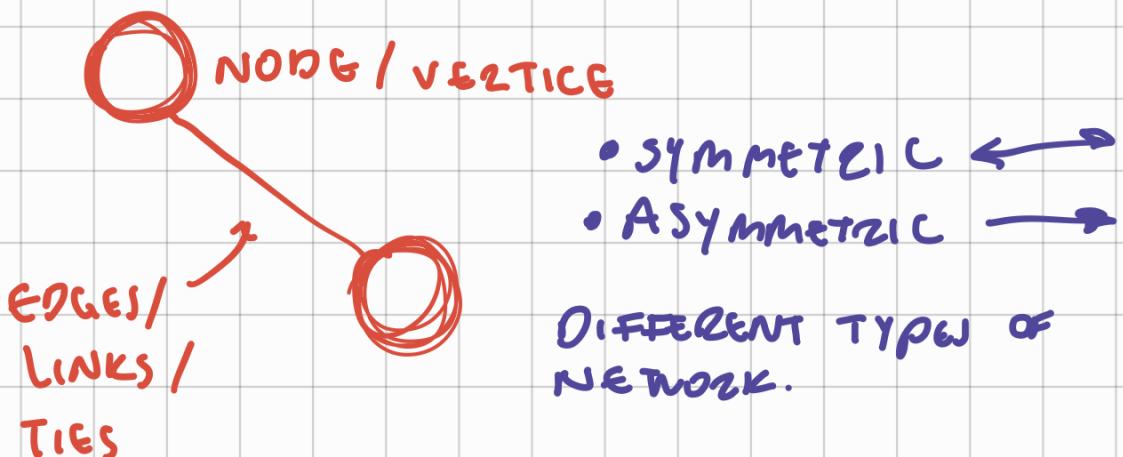
04.02.25

HAVE A LOOK AT NETWORKS

e.g.: NETWORK OF WIKIPEDIA ARTICLES
ABOUT CLIMATE CHANGE AND THEIR
CLUSTERING. [EMAPS].

A FOODWEB IS ALSO A NETWORK.

A NETWORK (OR A GRAPH) IS A REPRESENTATION
OF CONNECTIONS AMONG DIFFERENT ITEMS.



PYTHON: networkx as nx

UNDIRECTED NETWORK $g = nx.Graph()$
 $g.add_edge(.,.)$

DIRECTED NETWORKS $d = nx.DGraph()$
 $d.add_edge(.,.)$

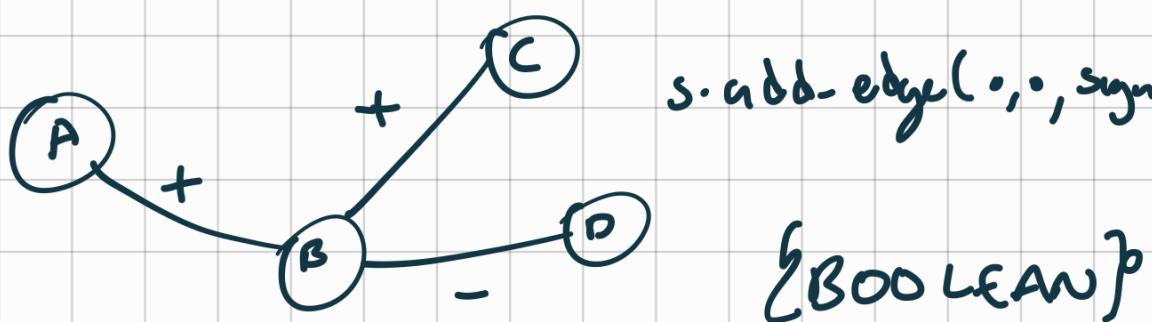
WEIGHTED NETWORKS: $w = nx.Graph()$



SIGNED NETWORKS: It's possible to study networks whose links carry a positive or negative connotation.

$s = nx.Graph()$

$s.add_edge(\cdot, \cdot, \text{sign} = "+")$
 $= "-")$

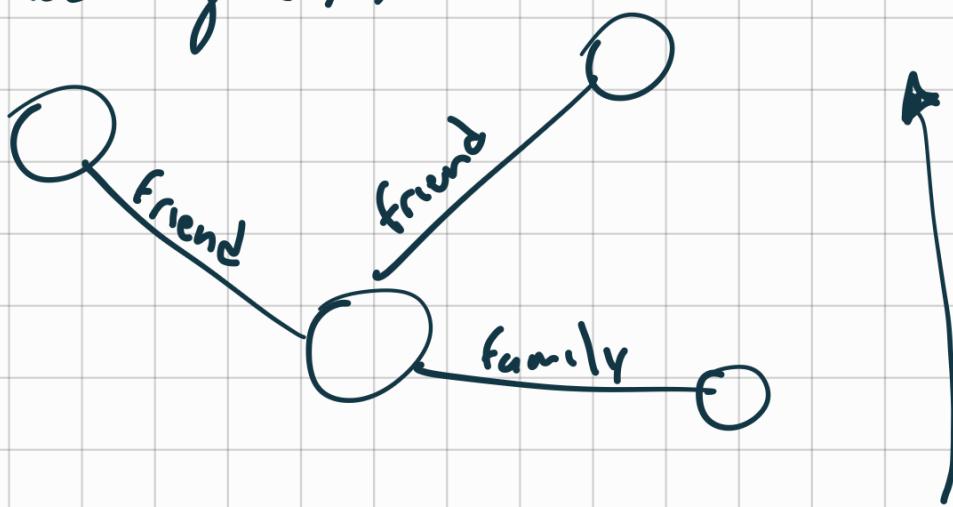


{BOOLEAN}

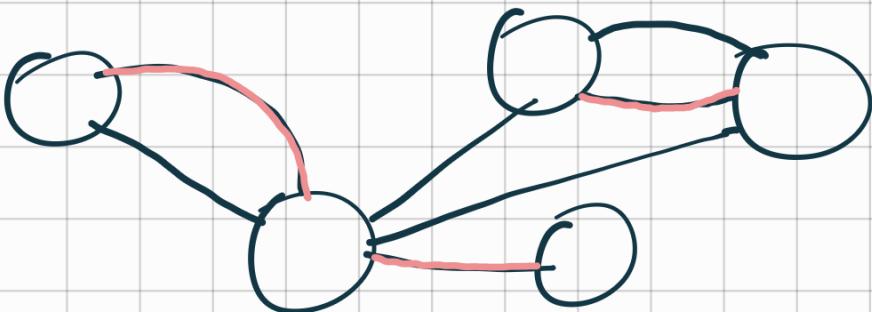
OTHER ATTRIBUTES

$g = nx.Graph()$

$g.add_edge(\cdot, \cdot, \text{relation} = "<\text{attribute}>")$



MULTIGRAPHS: A pair of nodes that can have different types of relationships simultaneously.



SUMMARY

[UNDIRECTED]

[DIRECTED]

[SIGNED]

[MULTIGRAPH]

[WEIGHTED]

PYTHON TOOLS

LIST OF EDGES

g.edges()

LIST OF EDGES WITH ALL ATTRIBUTES

g.edges(data=True)

SPECIFIC ATTRIBUTES; z.B.: "relation"

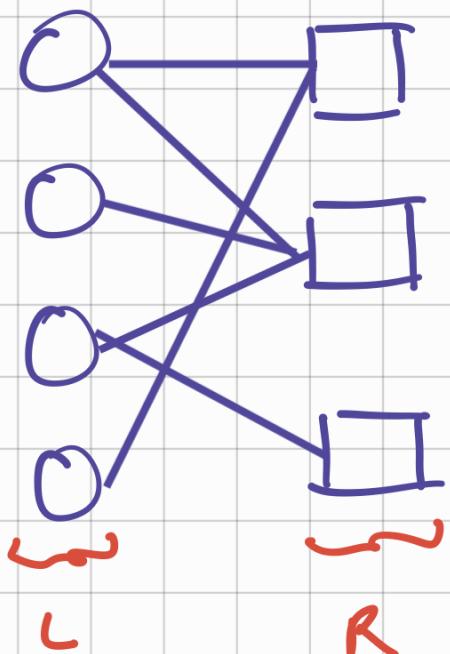
g.edges(data="relation")

ACCESSING A SPECIFIC EDGE

g.edge[·][·] # will print all attributes.

BIPARTITE GRAPHS

WHEN A GRAPH CAN BE SPLIT INTO TWO SETS, L AND R, AND EVERY EDGE CONNECTS A NODE IN L WITH A NODE IN R.



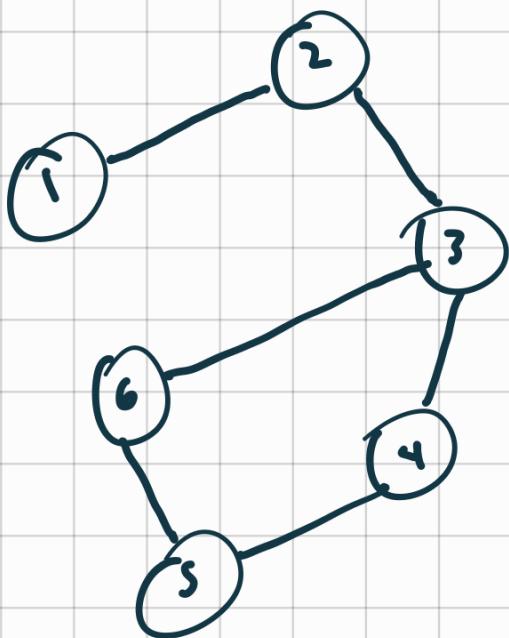
from networkx.algorithms
import bipartite

{ VERY EASY TO
CREATE :) }

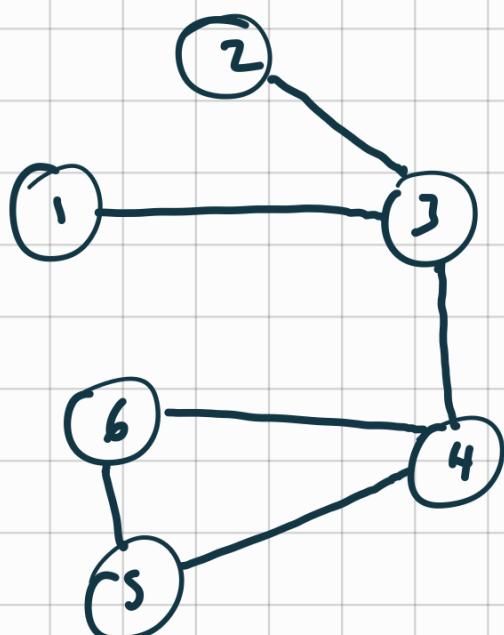
MODULE 7, VIDEO
 $t = 3:40$

REMARKS OF A BIPARTITE GRAPH:

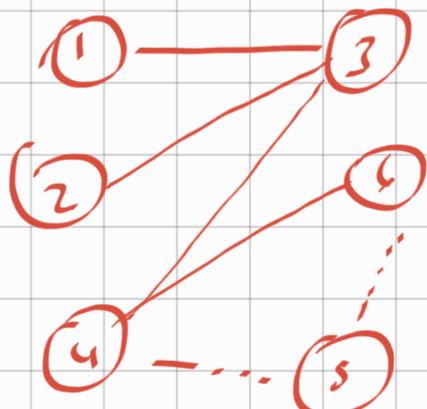
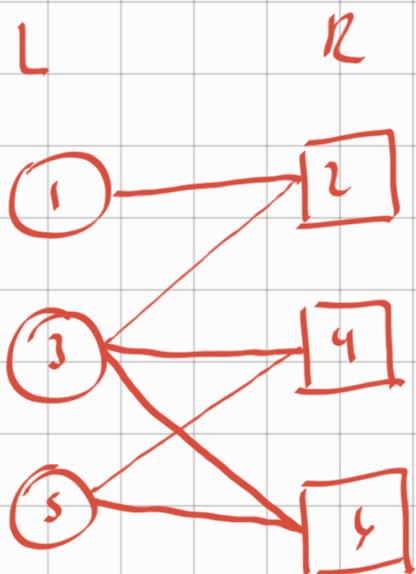
- CANNOT CONTAIN A CYCLE
OF AN ODD NUMBER OF NODES; z.B.:



$\{1, 3, 5\}; \{2, 4, 6\}$



NOT BIPARTITE.



IN PYTHON, IT'S POSSIBLE TO CHECK IF A GRAPH
1) BIPARTITE:

`g.is_bipartite(g)`

} For a specific node.

`g.is_bipartite_node_set(g, c)`

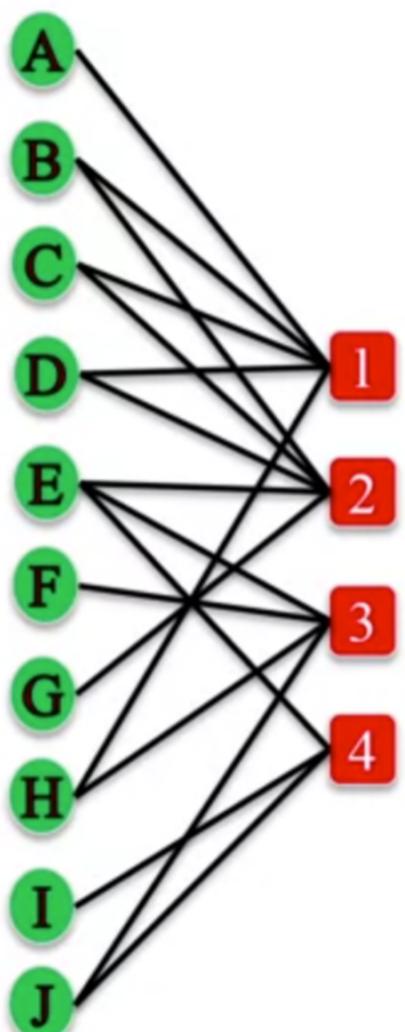
} For a specific set of nodes.

IT'S POSSIBLE TO GET THE SET OF NODES THAT BIPARTITE.

bipartite.sets(g)

L-BIPARTITE GRAPH PROJECTION

Network of nodes in group Γ , where a pair of nodes is connected if they have a common neighbour in Λ in the bipartite graph. e.g.:



$$\Lambda = \{1, 2, 3, 4\}$$

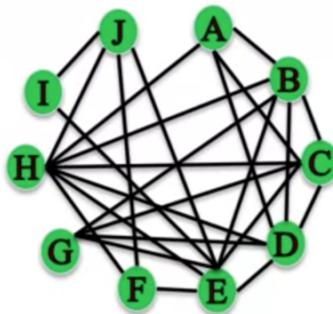
$$\Gamma = \{A, B, C, D, E, F, G, H, I, J\}$$

Projection on Γ

$\gamma = \text{bipartite.projected-graph}(g, \Gamma)$

Projection on Λ

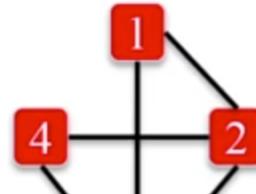
$\lambda = \text{bipartite.projected-graph}(g, \Lambda)$



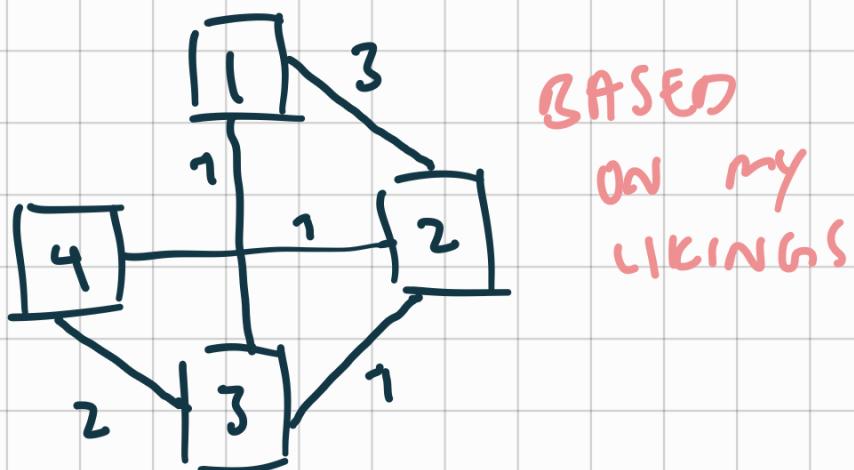
g

γ

λ



You can see that there's a bit of loss of information on λ when reducing g since there are more than one repetition for the connection of certain nodes. To preserve this information, one can consider the creation of a L-Bipartite weighted graph projection; 2.5: here:



EXAMPLE OF USAGE: RECOMMENDATION OF ARTISTS BASED ON USERS THAT LIKE SIMILAR ARTISTS.

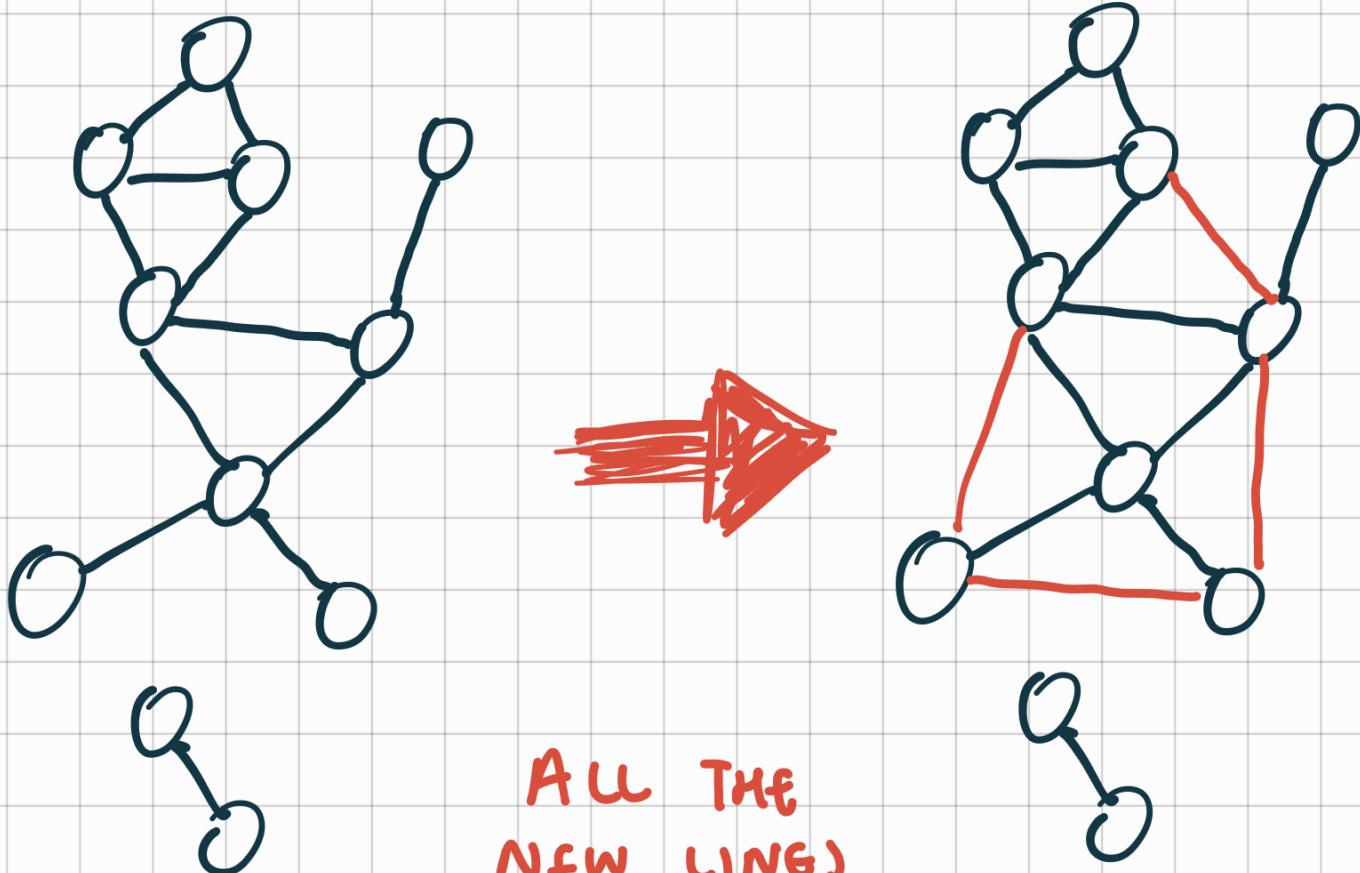
- (1) SANTA FE KLAN
 - (2) LEFTY SM
 - (3) STILO
 - (4) DOUBLE ONE FLOW LBL.
- NEW USER
- NEXT RECOMMENDATION BASED ON λ IS THEN
- THEN

MODULE 2

TRIADIC CLOSURE:

THE TENDENCY FOR NODES WHO SHARE CONNECTIONS IN A NETWORK TO BECOME CONNECTED.

FOR INSTANCE



ALL THE
NEW LINES
ARE LIKELY TO
OCUR AS THEY FORM
A TRIANGLE

LOCAL CLUSTERING COEFFICIENTS OF A NODE λ

FRACTION OF PAIRS OF THE NODE'S

CONNECTION THAT ARE CONNECTED WITH
EACH OTHER.

Z.B.: $\lambda \rightarrow C$

$$K_\lambda = \frac{M_\lambda}{N_\lambda}$$

λ ≡ NODE TAG

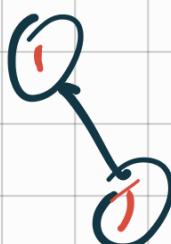
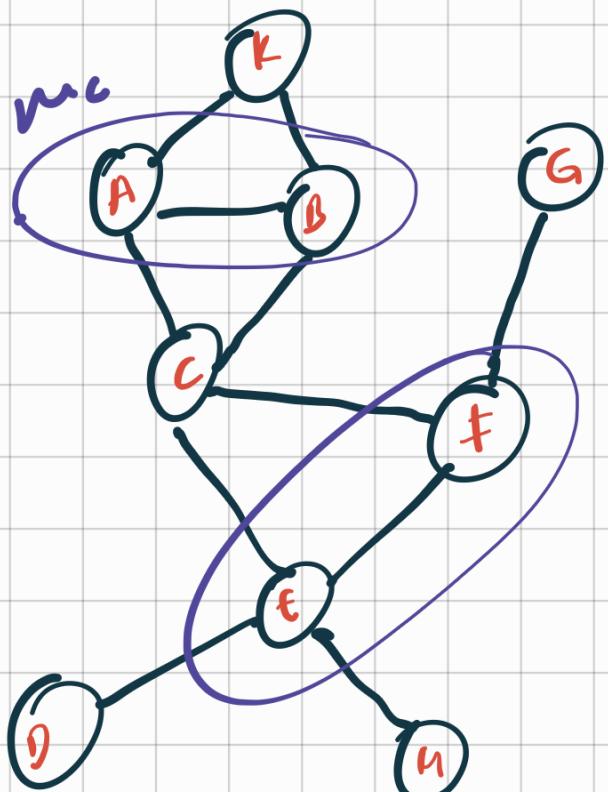
K_λ ≡ LOCAL CLUSTERING
COEFFICIENT OF NODE λ

M_λ ≡ # OF PAIRS OF λ 's
CONNECTIONS THAT ARE ALSO
CONNECTED

d_λ ≡ # OF DIRECT CONNECTIONS
TO λ

N_λ ≡ # OF PAIRS OF λ 's CONNECTIONS

$$N_\lambda \hat{=} \frac{d_\lambda(d_\lambda - 1)}{2}$$



$$K_c = \frac{m_c}{n_c} = \frac{2}{\left[\frac{4(4-1)}{2} \right]} = \frac{4}{4(4-3)}$$

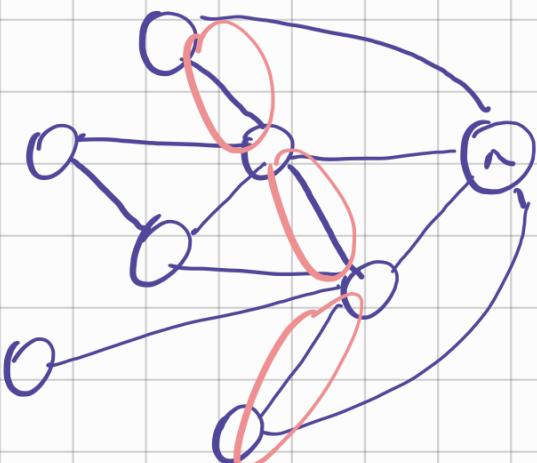
$$K_c = \frac{1}{3}$$

↳ LOCAL CLUSTERING COEFFICIENT FOR NODE C IN THE GIVEN NETWORK IS $\frac{1}{3}$. THIS MEANS THAT ONE-THIRD OF ALL THE POSSIBLE PAIRS OF CONNECTIONS TO C, WHICH COULD BECOME CONNECTED, ARE ACTUALLY CONNECTED.

REMARKS. IF YOU WERE TO GET THE LOCAL CLUSTERING COEFF OF J, YOU WOULD GET A MATHEMATICAL INCONSISTENCY. THEN ONE SETS:

$$\text{IF } n_j = 0 \rightarrow K_j \stackrel{!}{=} 0$$

QUICK EXAMPLE



$$K_j = \frac{2 \cdot m_j}{d_j(d_j-1)}$$

$$d_j = 4; m_j = 3$$

$$K_j = \frac{2 \cdot 3}{4(4-1)} = \frac{1}{2}$$

IT'S POSSIBLE TO GET THESE COEFFICIENTS IN NETWORK X.

GLOBAL CLUSTERING COEFF.

↳ ONE CAN TAKE THE AVERAGE LOCAL CLUSTERING COEFF. OF ALL NODES WITHIN THE NETWORK.

OR

↳ COMPUTE THE TRANSITIVITY OF A NETWORK

WHAT IS TRANSITIVITY?

↳ PERCEIVED AS THE PERCENTAGE OF OPEN TRIADS THAT ARE TRIANGLES IN A NETWORK.

TRIANGLES

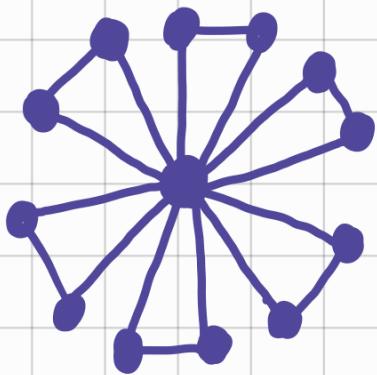


OPEN TRIADS



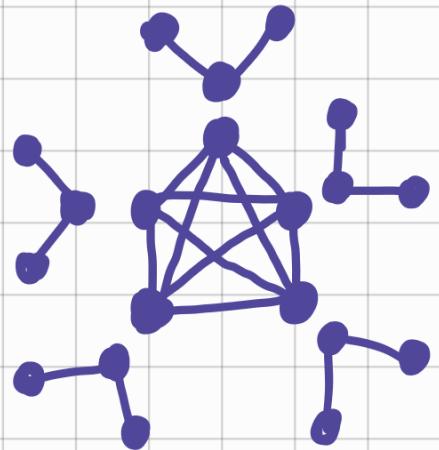
$$\text{TRANSITIVITY} = \frac{3 * \# \text{ OF CLOSED TRIADS}}{\# \text{ OF OPEN TRIADS}}$$

BOTH CONCEPTS CAN BE USED TO GAIN AN IDEA OF HOW THE GLOBAL CLUSTERING LOOKS LIKE. HOWEVER, HAVE A LOOK AT THE FOLLOWING TWO CASES.



MOST NODES HAVE HIGH LOCAL CLUSTERING COEFFICIENT BUT THE HIGH-DEGREE NODE HAS A LOW LOCAL CLUSTERING COEFFICIENT.

- AVE. CLUSTERING COEFF = 0.93
- TRANSITIVITY = 0.23



MOST NODES HAVE LOW LCC BUT THE HIGH DEGREE NODES HAVE HIGH LCC.

- AVE. CLUSTERING COEFF = 0.93
- TRANSITIVITY = 0.86

- THE AVERAGE LOCAL CLUSTERING COEFFICIENT OF ALL NODES IN NETWORKX
↳ `nx.average_clustering(g)`

- TRANSITIVITY IN NETWORKX
↳ `nx.transitivity(g)`

DISTANCE - RELATED CONCEPTS

- DISTANCE BETWEEN TWO NODES: length of shortest path
- ECCENTRICITY: a list with the max distance there's to reach another node + nodes.
- DIAMETER: $\max(\text{eccentricity})$
- RADIUS: $\min(\text{eccentricity})$

- PERIPHERY: all nodes whose eccentricity = DIAMETER
- CENTER: all nodes whose eccentricity = RADIUS

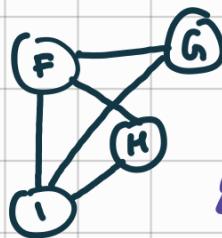
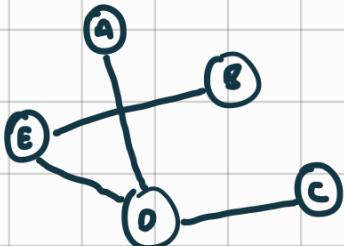
You can get an idea of how close are the nodes with each other by taking the average distance between every pair of nodes.

CONNECTIVITY - RELATED CONCEPTS

CASE: UNDIRECTED NETWORKS

- A SUBSET OF NODES IS CONSIDERED A CONNECTED COMPONENT IFF:
 - EVERY NODE IN THE SUBSET HAS A PATH TO EVERY OTHER NODE (NO MISSING LINKS).
 - NO OTHER NODE HAS A PATH TO ANY NODE IN THE SUBSET. (MORE LINKS THAN NECESSARY).

Ex.:



$\{A, B, C, I\}$ violates (1).

$\{F, H, I\}$ violates (2).

$\{A, B, C, D, E\}$ is ✓

$\{F, G, H, I\}$ is ✓

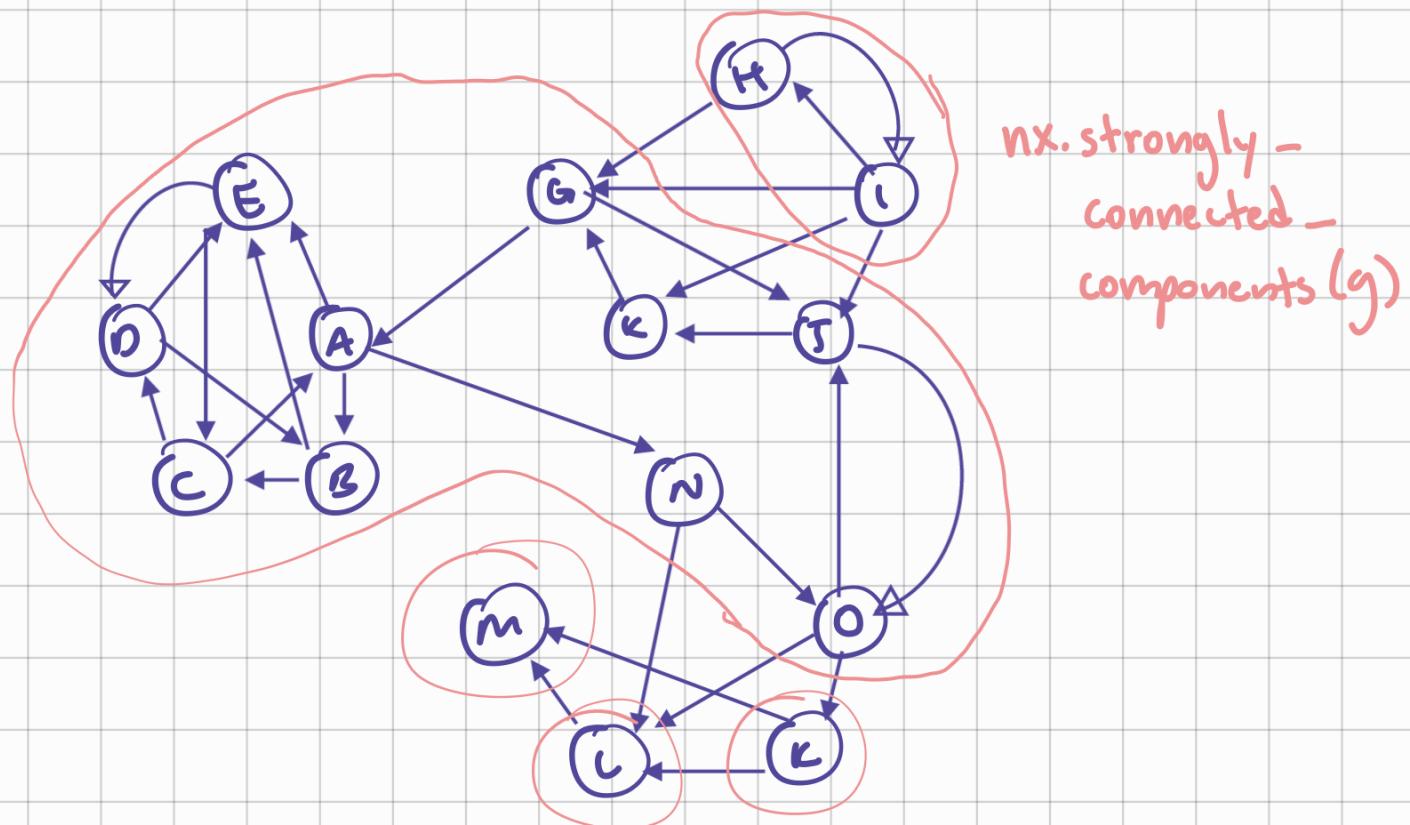
▷ POSSIBLE TO CHECK
O WITH NETWORKX.

CASE: DIRECTED NETWORKS

- A DIRECTED GRAPH IS STRONGLY CONNECTED IFF FOR EVERY PAIR OF NODES (i, j) , THERE'S A DIRECTED PATH FROM i TO j , AND A DIRECTED PATH FROM j TO i .
- A DIRECTED GRAPH IS WEAKLY CONNECTED IFF REPLACING ALL DIRECTED EDGES WITH UNDIRECTED EDGES PRODUCES A CONNECTED UNDIRECTED GRAPH.

- A STRONGLY CONNECTED COMPONENT IS A SUBSET OF NODES THAT:
 - EVERY NODE IN THE SUBSET HAS A DIRECTED PATH TO EVERY OTHER NODE.
 - NO OTHER NODE HAS A DIRECTED PATH TO AND FROM EVERY NODE IN THE SUBSET.

Q.2.: WHAT ARE THE STRONGLY CONNECTED NODES IN THE FOLLOWING NETWORK?



ROBUSTNESS - RELATED CONTENTS

NETWORK ROBUSTNESS IS THE ABILITY OF A NETWORK TO MAINTAIN ITS GENERAL STRUCTURAL PROPERTIES WHEN IT FACES FAILURES OR ATTACKS. (ATTACKS $\hat{=}$ REMOVAL OF EDGES OR NODES).

Por ejemplo, si una tuviera que desmantelar un grupo racista

- o que hace profit violando derechos humanos, ¿qué link o node se perderían de dar más atención para fraccionar el grupo
- o debilitarlo?

CONCEPTO FUNDAMENTAL PARA ESTRATEGIAS Y TOMA DE DECISIONES.

IN NETWORKX, IT IS POSSIBLE TO ASK "WHAT IS THE SMALLEST NUMBER OF NODES OR LINKS THAT CAN BE REMOVED IN ORDER TO DISCONNECT THE GRAPH?"

NODES

AMOUNT: `nx.node_connectivity(g)`

WHICH: `nx.minimum_node_cut(g)`

EDGES

AMOUNT: `nx.edge_connectivity(g)`

WHICH: `nx.minimum_edge_cut(g)`

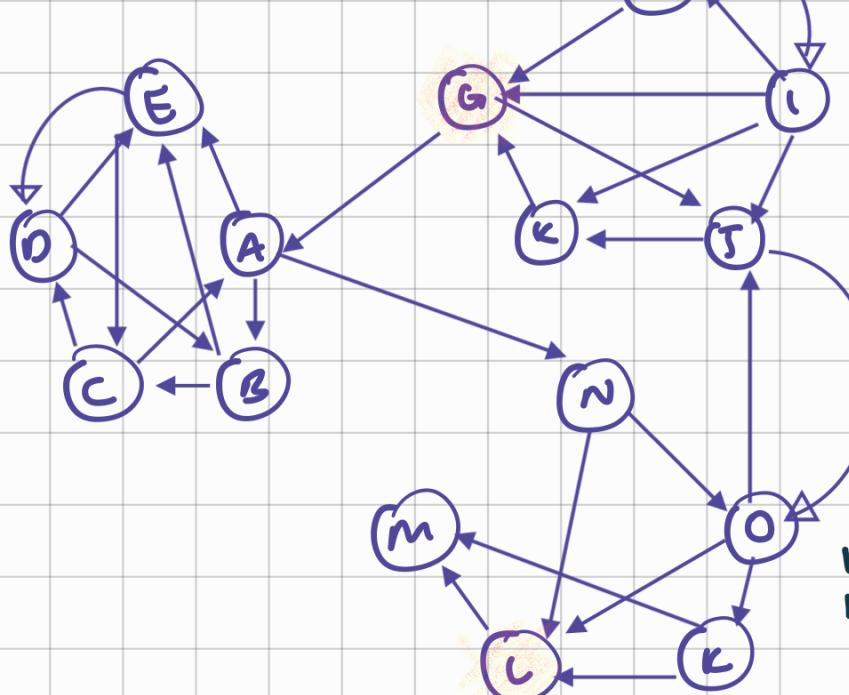
ROBUST NETWORKS HAVE LARGE MINIMUM NODE AND EDGE CUTS.

WHAT ABOUT SENDING MESSAGES? WHAT WOULD BE ALL THE OPTIONS OR SHORTEST PATH TO ACHIEVE SO? E.G.: SENDING A MESSAGE FROM G TO L.

`nx.all_simple_paths(g, "G", "L")`

```
>>> [[G, A, N, L],
      [G, A, N, O, K, L],
      [G, A, N, O, L],
      [G, J, O, K, L],
      [G, J, O, L]]
```

BUT NOW, IF THE OBJECTIVE WAS TO BLOCK ALL COMMUNICATION BETWEEN NODES (G, L), WHAT WOULD BE THE MINIMUM EDGE OR LINK CUT?



`nx.minimum_node_cut(g, "G", "L")`

```
>>> {"N", "O"}
```

`nx.minimum_edge_cut(g, "G", "L")`

```
>>> {("A", "N"), ("J", "O")}
```

