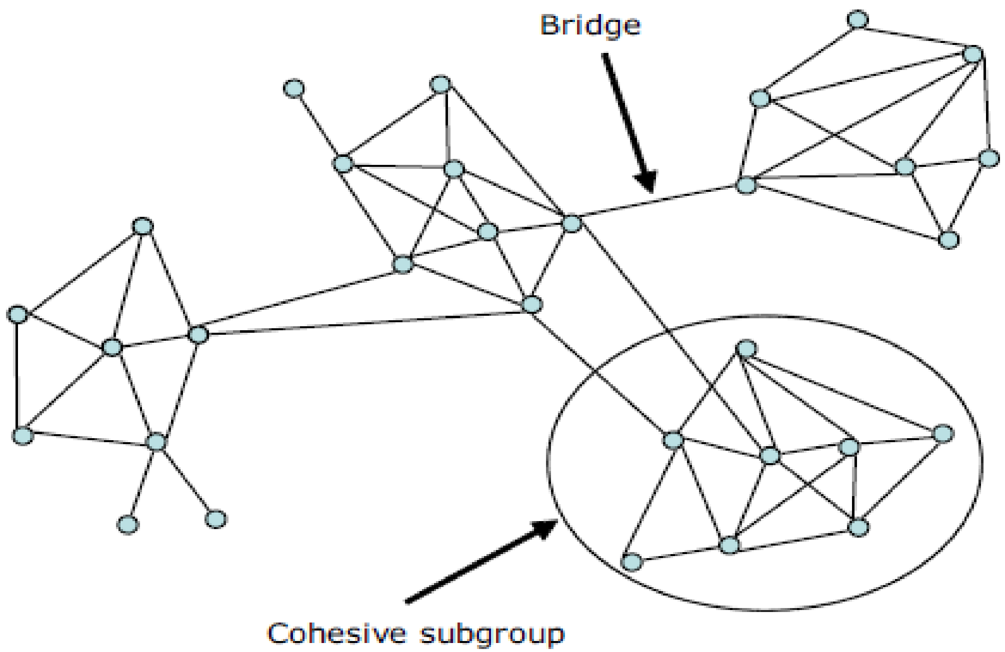
Macro Structure of Social Networks

Co-authorship Network

- Social network emerges as dense cluster or social groups, sparsely connected to each other



Erdo’s number – measure of no. of steps in co-authorship network from Erdo’s to the given researcher

Visualization

- Network visualizations (based on topographic or physical principles) helps to understand group structure of social networks and to identify hub
- Difficult to capture visualization as expected, due to multidimensional scaling of graph
- Dense graph with fewer dimension will degenerate into a meaningless “spaghetti bowl”
- Apart from visualization, it is necessary to find subgroups based on disjoint and overlapping set of nodes

Various definitions of Subgroups

- Subgroups based on densely connected to their members.
- For example, a *clique* in a graph is maximal complete subgraph of three or more nodes
- As complete subgraphs are very rare, the definition of a clique is typically relaxed by allowing some missing connections
- For example, a *k-plex is a maximal subgraph, in which each* node is adjacent to no fewer than $g_s - k$ *nodes in the subgraph*
- The larger we set the parameter *k, the larger the* k-plexes that we will find.

Contd…

- lambda-set analysis method is based on the definition of edge connectivity.

- *the edge connectivity of two* vertices *vi and vj (λ(i, j)) is the minimum number of lines that need to be removed from a* graph to leave no path between the two vertices

- A lambda-set is then defined as - any pair of nodes from the set has a larger edge connectivity than any pair of nodes where one node is from within the set and the other node is from outside the set

- Hence if l(a,b) represents the edge-connectivity of two vertices a and b from a graph G(V,E) then a subset S is a lambda set if it is the maximal set with the property that for all a,b,c € S and d € V-S then l(a,b) > l(c,d)

- Unlike k-plexes, lambda-sets has property that they are not overlapping.

Contd…

- edge-betweenness clustering method of Mark Newman takes a different approach

- Rather than density of subgroups, this algorithm targets, the ties that connect them

- Betweenness of an edge is calculated by taking the set of all shortest paths in the graph and looking at what fraction of them contains

  the given edge

- much higher betweenness of edges joining clusters as it serves as shortest path for all nodes between cluster

- By progressively removing the edges with the highest betweenness in the graph ends up with distinct clusters of nodes.

Contd…

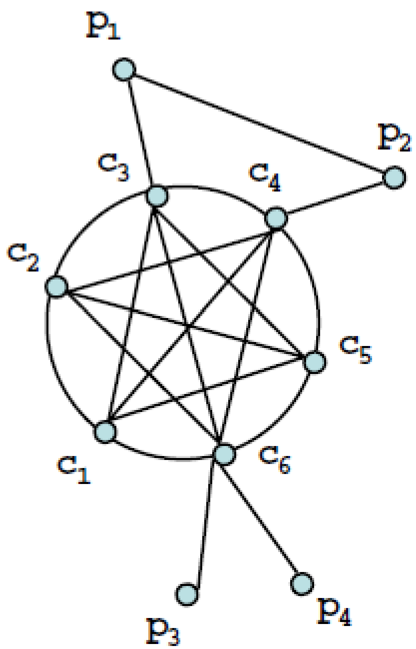
- Clustering a graph into subgroups allows us to visualize the connectivity at a group level

- Core-Periphery structure - nodes divided in two distinct subgroups: nodes in the core are densely connected with each other and the nodes on the periphery, while peripheral nodes are not connected with each other, only nodes in the core

- The matrix form of a core periphery structure is $\begin{pmatrix} 1 & . \\ . & 0 \end{pmatrix}$

- Algorithms for identifying C/P structures *works by dividing the set of nodes in such a way* error the between the actual image and the “perfect” image is minimal

Core-Periphery Structure



c_1	c_2	c_3	c_4	c_5	c_6	p_1	p_2	p_3	p_4
1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	0	1	0	0
1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	1	1
0	0	1	0	0	0	0	1	0	0
0	0	0	1	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0

Contd…

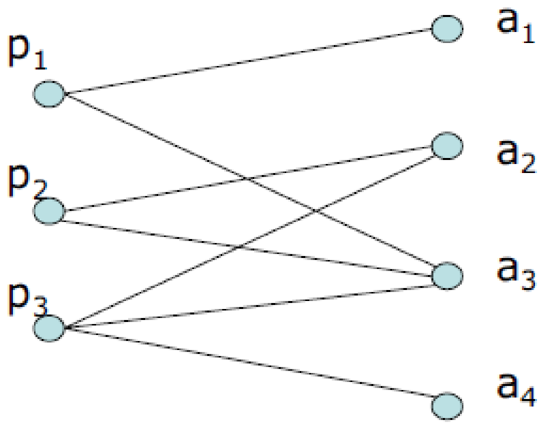
- Affiliation networks contain information about the relationships between two sets of nodes: a set of subjects and a set of affiliations

- An affiliation network can be formally represented as a *bipartite graph*

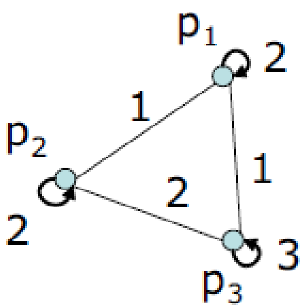
- the set of vertices is divided into *n disjoint sets and there are no edges* between vertices belonging to the same set

- Used for mapping interlocking directorates

Affiliation Network



	p_1	p_2	p_3	a_1	a_2	a_3	a_4
p_1	0	0	0	1	0	1	0
p_2	0	0	0	0	1	1	0
p_3	0	0	0	0	1	1	1
a_1	1	0	0	0	0	0	0
a_2	0	1	1	0	0	0	0
a_3	1	1	1	0	0	0	0
a_4	0	0	1	0	0	0	0



	p_1	p_2	p_3
p_1	2	1	1
p_2	1	2	2
p_3	1	2	3