Unit- 5

1. What is Social network analysis? How it can be modeled?

→ - Social network analysis (SNA) is the process of ~~investig~~ investigating social structure through the use of networks and graph theory.

- SNA is mapping & measuring of relationships & flows between people, groups organizations, computers, URLs and other connected information / knowledge entities

- types of SNA

1) ~~Ego Network Analysis~~

1) Egocentric (personal) Network analysis

:- involves quantification of interactions between an individual (ego) & all other persons (alters) related directly or ~~indirect~~ indirectly to ego.

:- emerged in psychology

2) Sociocentric (whole) Network analysis

:- Involves quantification of interaction among a socially well-defined group of people

:- emerged in Sociology

Social Networks are naturally modeled as graphs called as social graph.

basic components: nodes & edges.

Nodes -

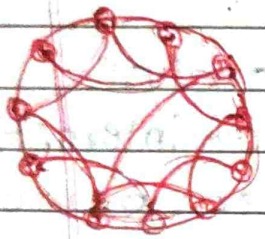
represents entities in the network & can hold self-properties (weight, size ---), & network based properties (degree, No. of neighbours)

Edge -

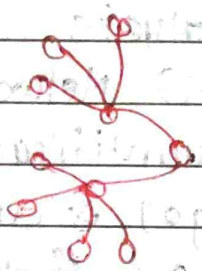
represents the connections bet^{n} nodes, degree is represented by labelling the edges.

often social graphs are undirected.

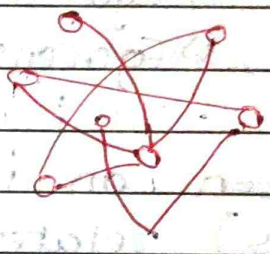
Graphs can be clustered to identify communities



(small-world network)

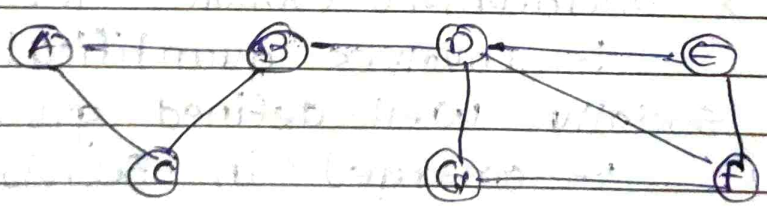


Scale-free graph



Random Network

e.g. of small social network

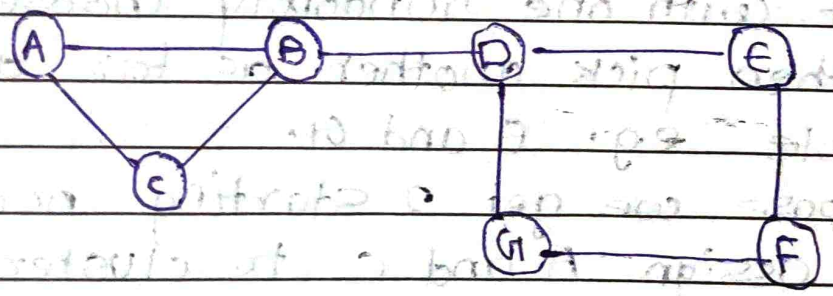


Q2 How Social Network graph can be clustered by applying standard clustering model.

→ there are 2 general approaches to clustering

i) Hierarchical (agglomerative) clustering

- Hierarchical clustering of a Social Network graph starts by combining some two nodes that are connected by an edge.
- Successively, edges that are not between two nodes of the same cluster would be chosen randomly to combine the clusters to which their two nodes belong.
- The choices would be random, because all distances represented by an edge are the same

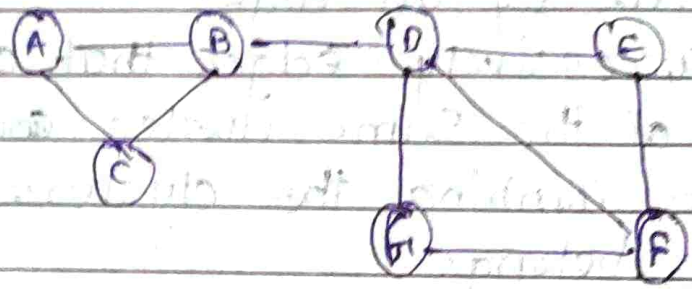


- At highest level it appears that there are two communities {A, B, C} and {D, E, F, G}
- {D, E, F} and {D, F, G} subcommunities of {D, E, F, G} - ever be identified by pure clustering algorithm.

But, it is likely to chose to combine B and D even though they surely belong in different clusters.

Sol^n :- Run heirarchical clustering Several times and pick the run that gives most Coherent clusters.

* Point - assignment approach to clustering Social networks



- Suppose we try a k-means approach pick k=2
- pick two starting nodes at random
- start with one randomly choosen node and then pick another as far away as possible eg. E and G.
- Suppose we get 2 starting nodes B, F
  - assign A and C to cluster of B
  - E and G to the cluster of F
  - But D is close to B as it is to F
- Deffered decision about D, untill all are assigned
- Shortest avg distance to all the nodes of the cluster, then D should be assigned to cluster of F

But the fact that all edges are at the same distance will introduce a number of random factors that will lead to some nodes being assigned to the wrong cluster.

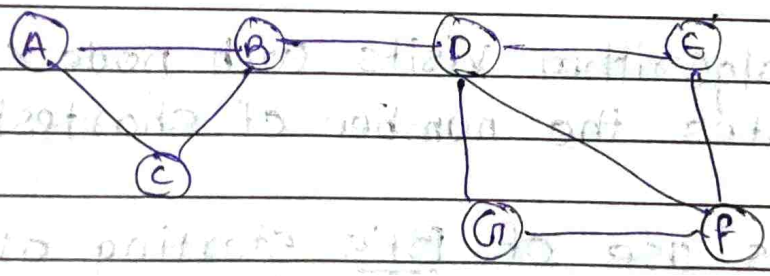
Q3 What is betweenness? Explain Girvan-newman algorithm to calculate.

⟹

The betweenness measures the extent to which a node lies on the shortest path connecting any two nodes in the network.

- This can be interpreted as the extent to which information passes through this node.

- A node with high betweeness possibly connects communities with each other.



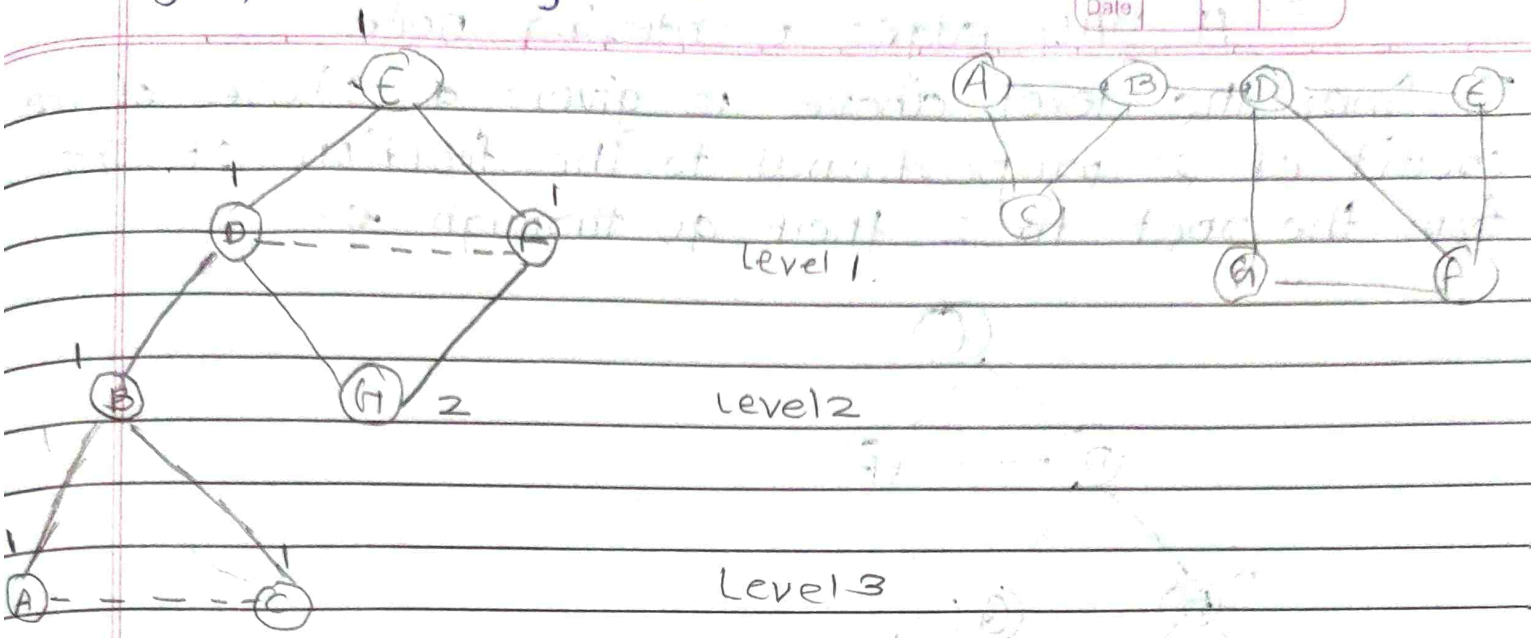
Define the betweenness of an edge (a,b) to be the number of pairs of nodes x and y such that the edge (a,b) lies on the shortest path between x and y. edge (a,b) is ~~created~~ credited with fraction of those shortest path that include the edge (a,b).

- (B, D) has the highest betweenness
- Edge is on every shortest path between any of A, B and C to any of D E, F and G (3×4 = 12)
- Betweenness of edge (D, f) = 4 (on four SP from A, B, C, and F to f).

In order to exploit betweenness of edges, need to calculate number of Shortest Paths (SPs) going through each edge.
Can use Girvan-Newman Algorithm
it relies on the iterative elimination of edges that have the highest number of shortest paths between nodes passing through them. By removing edges from the graph one-by-one, the network breaks down into smaller pieces, so called communities.

- GN algorithm visits each node x once and computes the number of shortest paths from x.
- Makes use of BFS starting at the node x
- Level of each node in BFS is length of SP from x to that node
- Edges between levels are called DAG edges.
- Each DAG edges will be part of at least one SP.

Step - 1 : breadth - first presentation of the graph starting at E



Step 2 - label each node by the number of SPs that reach it from root

Step 3 -

- calculate for each edge e sum over all nodes Y of the fraction of SPs from root X to Y that go through e.

- Calculation involves computing this sum for both nodes edges from the bottom.

- each Nodes and Edges are given Credit.

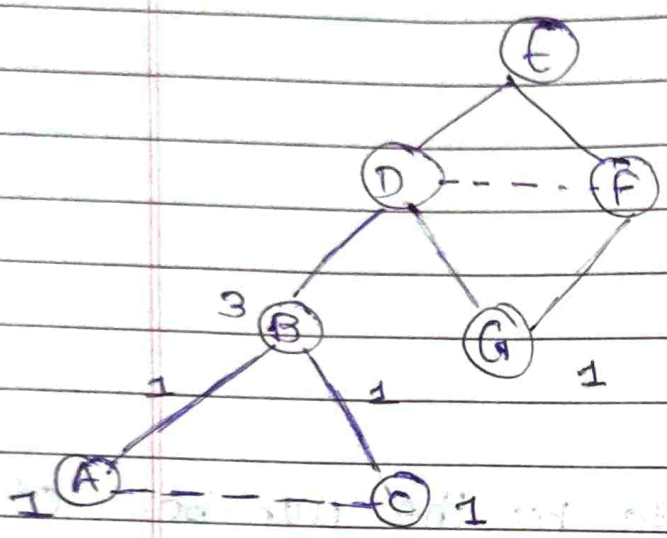
- The rules for the calculation of credit are as follows:

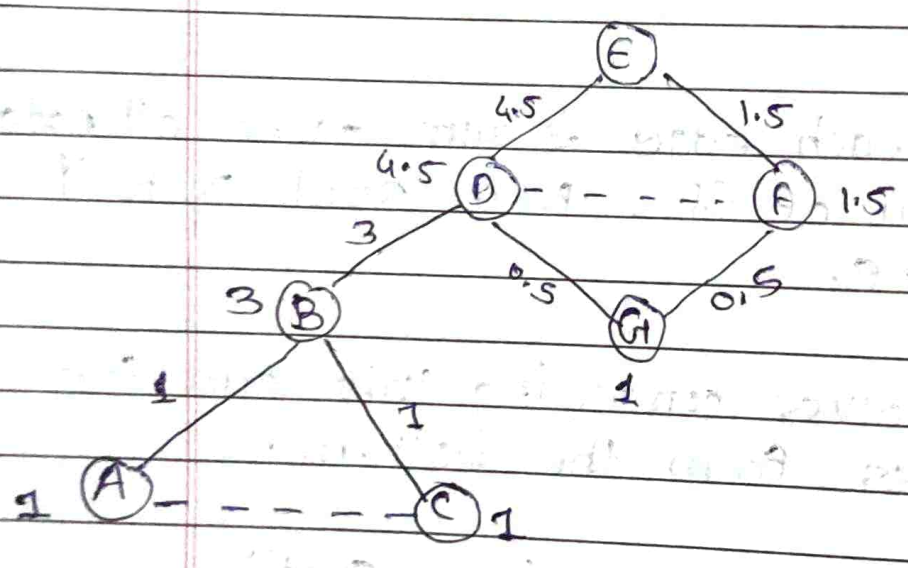
1. Each leaf in the DAG gets a credit of 1.

2. Each node that is not a leaf gets a credit equal to 1 plus the sum of credits of the DAG edges from that node to the

level below.

3. A DAG edges e entering node z from the level above is given a share of the credit of z proportional to the fraction of SPs from the root to z that go through e.





to complete the betweeness calculation,
- repeat this calculation for every node as the root
- Sum the contributions
- finally, we must divide by 2, get the true betweeness.

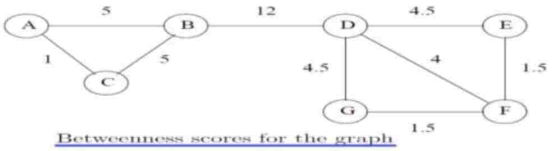
4. What is Betweenness?

-> Q.3)

How Betweenness can be used to Find Communities in Social Network Graph.

->

- Betweenness scores for the edges of a graph behave something like a distance measure.
- Not exactly a distance measure, because it is not defined for pairs of nodes that are unconnected by an edge, and might not satisfy triangle inequality.
- Idea is expressed as a process of edge removal.
- Remove edges with the highest betweenness.



Betweenness scores for the graph

Remove edges with betweenness four or more

All the edges with betweenness 4 or more have been removed

- B is a “traitor” to the community {A,B,C} D can be seen as a “traitor” to the group {D,E, F,G}

- If we apply the above method to a graph of n nodes and e edges, it takes O(ne) running time.
- BFS from a single node takes O(e) time, there are n of the computations.
- If the graph is large – and even a million nodes will high running time.
- Can pick a subset of nodes at random and use these as the roots of breadth-first searches Can get an approximation to the betweenness of each edge.

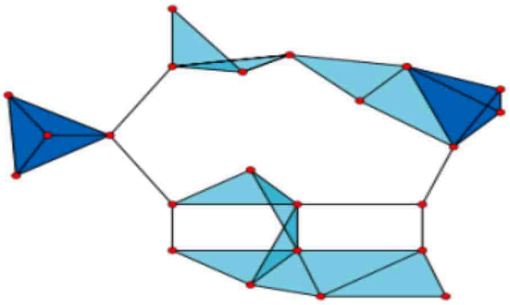
- Limitations of Betweenness to Find Communities.
(Searched for communities by partitioning all individuals in Social network)
- It is not possible to place an individual in two different communities.
- Need a technique for discovering communities directly by looking for subsets of the nodes that have a relatively large number of edges among them.

5. How to discover Communities in Social-Network Graph directly?

->

Clique (graph theory)

In the mathematical area of graph theory, a clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete

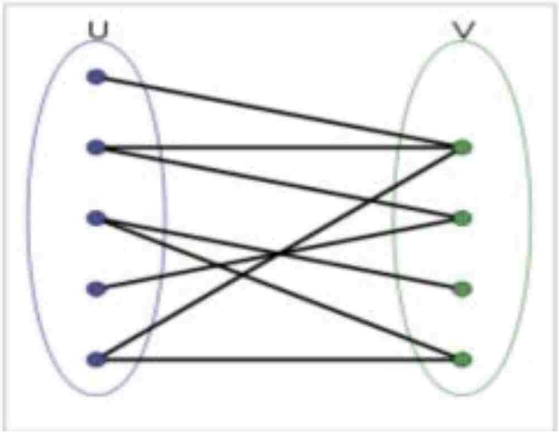


Finding Cliques-
- Find sets of nodes with many edges by finding a large clique However, that task is not easy.
- Finding maximal cliques NP-complete, but hardest of the NP Even approximating the maximal clique is hard.
- Cliques may be relatively small – result in identifying small size community.

Bipartite graphs (graph theory) -

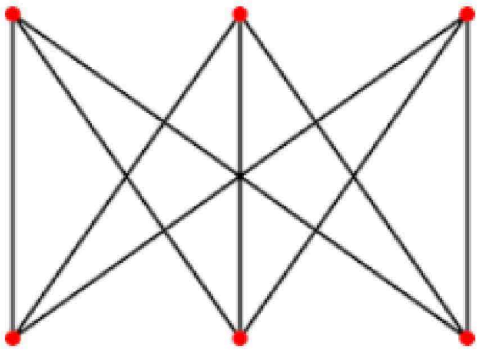
In the mathematical field of graph theory, a bipartite graph (or bigraph) is a graph whose vertices can be divided into two disjoint and independent sets U and V such that every edge connects a vertex in U to one in V.

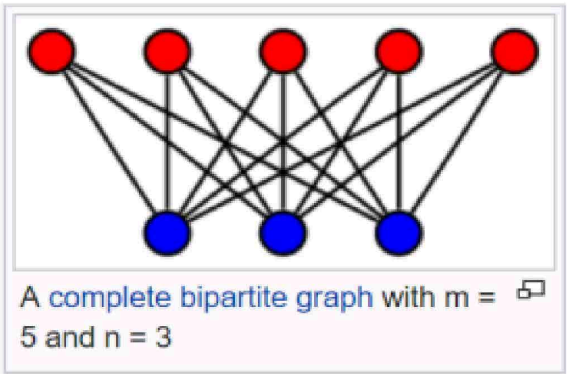
Vertex sets U and V are called the parts of the graph.



Complete Bipartite graphs (graph theory) -

In the mathematical field of graph theory, a complete bipartite graph or biclique is a special kind of bipartite graph where every vertex of the first set is connected to every vertex of the second set.



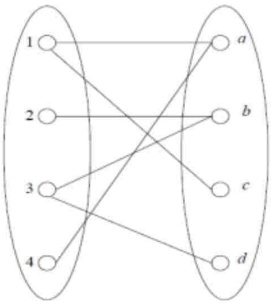


A complete bipartite graph with m = 5 and n = 3

- A complete bipartite graph consists of s nodes on one side and t nodes on the other side, with all st possible edges between the nodes of one side and the other present.
- We denote this graph by Ks,t.
- Can regard a complete bipartite subgraph as the nucleus of a community.

- We can also use complete bipartite subgraphs for community finding in ordinary graphs.
- Divide the nodes into two equal groups at random.
- If a community exists, then we would expect about half its nodes to fall into each group, and half its edges would go between groups.
- Chance of identifying a large complete bipartite subgraph in the community.
- To this nucleus we can add nodes from either of the two groups, if they have edges to many of the nodes already identified as belonging to the community.

Finding Complete Bipartite Subgraphs Example



The bipartite graph

- Left side is nodes {1, 2, 3, 4} - items and right side is {a, b, c, d} – baskets
- basket *a* consists of "items" 1 and 4;
- a = {1, 4}, b = {2, 3}, c = {1} and d = {3}.
- If s = 2 and t = 1, must find item-sets of size 1 that appear in at least two baskets.
- {1} is one such itemset, and {3} is another.

In this tiny example there are no item-sets for larger, more interesting values of s and t, such as s = t = 2.

6. How Social-Network Graph can be partitioned to identify Communities?

7. How to find overlapping communities in Social Network Graph?

->

- Concentrated on clustering a social graph to find communities.
- In practice communities are rarely disjoint.
- We explain a method for taking a social graph and fitting a model to it that best explains how it could have been generated by a mechanism.

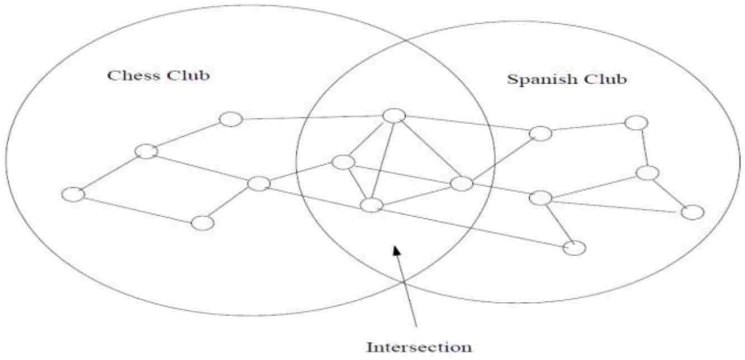
- This assumes that probability that two individuals are connected by an edge (are “friends”) increases as they become members of more communities in common.
- An important tool in this analysis is “maximum-likelihood estimation” (MLE).

In statistics, Maximum Likelihood Estimation (MLE) is a method of estimating the parameters of a statistical model, given observations.
- The method obtains the parameter estimates by finding the parameter values that maximize the likelihood function.
- The estimates are called maximum likelihood estimates, which is also abbreviated as MLE.

In (MLE) Assumption about the generative process (the model) that creates instances of some artifact, for example, “friends graphs”. Model has parameters that determine probability of generating any particular instance of the artifact. This probability is called the likelihood of those parameter values. We assume that value of parameters that gives largest value of likelihood is correct model for observed artifact.

For eg.
- Suppose that each edge is present with probability p and not present with probability 1−p.
- The only parameter we can adjust is p. Each value of p there is a small but nonzero probability that the graph generated will be exactly the one we see.
- Following the MLE principle, we shall declare that true value of p is the one for which the probability of generating the observed graph is the highest.

➤ Nodes are people and there is an edge between two nodes if the people are "friends."



The overlap of two communities is denser than the nonoverlapping parts of these communities

There are 15 nodes and 23 edges. 105 pairs of 15 nodes probability (likelihood) of generating exactly the graph is given by the function p^(23)*(1−p)^(82)

8. Explain Affiliation-Graph Model to find overlapping communities in SocialNetwork Graph.

->

We use the affiliation-graph model, to generate social graphs from communities.

- Once we see how parameters of model influence likelihood of seeing a given graph, we can address how one would solve for values of the parameters that give maximum likelihood. The mechanism, called community-affiliation graphs.

1. There is a given number of communities, and there is a given number of individuals (nodes of the graph).

2.Each community can have any set of individuals as members. That is, the memberships in the communities are parameters of the model.

3.Each community C has a probability pC associated with it, probability that two members of community C are connected by an edge because they are both members of C. These probabilities are also parameters of the model.

4.If a pair of nodes is in two or more communities, then there

is an edge between them if any of communities of which both are members, justifies that edge according to rule (3).

We must compute the likelihood that a given graph with the proper number of nodes is generated by this mechanism.
- The key observation is how the edge probabilities are computed, given an assignment of individuals to communities and values of the pC's.
- Consider an edge (u, v) between nodes u and v.
- Suppose u and v are members of communities C and D, but not any other communities.
- Then the probability that there is no edge between u and v is the product of the probabilities that there is no edge due to community C and no edge due to community D.
- That is, with probability (1 − pC)(1 − pD) there is no edge (u, v) in the graph, and of course probability that there is such an edge is 1 minus that.

- More generally, if u and v are members of a nonempty set of communities M and not any others, then p^{uv}, probability of an edge between u and v is given by:

$$p_{uv} = 1 - \prod_{C \text{ in } M} (1 - p_C)$$

- As an important special case, if u and v are not in any communities together, then we take p_{uv} to be ϵ, some very tiny number.
- If we know which nodes are in which communities, then we can compute likelihood of given graph.
- Let M_{uv} be set of communities to which both u and v are assigned.
- Then likelihood of E being exactly the set of edges in observed graph is

$$\prod_{(u,v) \text{ in } E} p_{uv} \prod_{(u,v) \text{ not in } E} (1 - p_{uv})$$

9. Why triangles in Social-Network Graph are counted? Explain algorithm for finding triangles in Social Network Graph.

->

it estimating or getting an exact count of triangles in a very large graph,

Why Count Triangles?

- Consider Graph with n nodes and m edges
- Can calculate this number of triangles without difficulty.
- There are $\binom{n}{3}$ sets of three nodes, or approximately $\frac{n^3}{6}$
- Probability of an edge between any two given nodes being added is $m/\binom{n}{2}$ or approximately $2m/n^2$.
- The probability that any set of three nodes has edges between each pair, is approximately $(2m/n^2)^3 = 8m^3/n^6$.
- Expected number of triangles is $(8m^3/n^6)(n^3/6) = \frac{4}{3}(m/n)^3$

Why Count Triangles?

- If a graph is a social network with **n participants** and **m pairs of "friends,"** then number of triangles to be much greater than value for a random graph.
- The reason is that if A and B are friends, and A is also a friend of C, there should be a much greater chance than average that B and C are also friends.
- Counting the number of triangles helps us to measure **extent to which a graph looks like a social network.**
- The age of a community is related to the density of triangles.
- New community number of triangles is relatively small.

An Algorithm for Finding Triangles

- Consider a graph of n nodes and m ≥ n edges and nodes are integers 1, 2, . . , n.
- Call a node a heavy hitter if its degree is at least $\sqrt{m}$.
- A heavy-hitter triangle is a triangle whose all three nodes are heavy hitters.
- Note that the number of heavy-hitter nodes is no more than 2 $\sqrt{m}$
- Since each edge contributes to degree of only two nodes, there would then have to be more than m edges.

Counting Triangles Cont...

An Algorithm for Finding Triangles

Assuming graph is represented by its edges, pre-process graph as follows:

1. Compute the degree of each node. The total time required is O(m)
2. Create an index on edges, with the pair of nodes at its ends as the key. A hash table suffices. It can be constructed in O(m) time.
3. Create another index of edges, this one with key equal to a single node.

We shall order the nodes as follows

- First, order nodes by degree.
- If v and u have the same degree, recall that both v and u are integers, so order them numerically.
- That is, we say v < u if and only if either
  1. The degree of v is less than the degree of u, or
  2. The degrees of u and v are the same, and v < u.

Time required to find heavy-hitter and other triangles are O(m^(3/2))