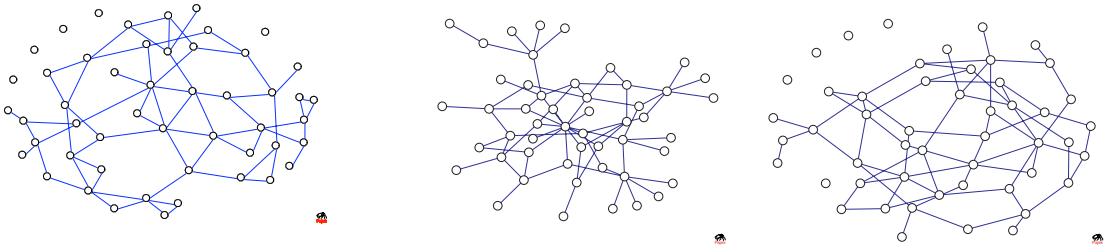
Midterm

SI 508

This is a take-home exam. It is open book, open cTools, open web. But it is not open-human. You are to work on this entirely on your own without the help of others. You can use Pajek, Guess or NetLogo, but I have designed the exam such that you should be able to answer all the questions without them. The exam is worth 20% of your final grade. It is due at 4 pm in class on Thursday the 25th. No late submissions will be accepted without prior extensions granted. If you have questions, email me directly, do not post to cTools. I will post clarifications on cTools, so please check the forum regularly, or at least before you submit your exam. You can turn in the exam (on cTools) or (in person on paper in class). Good luck!

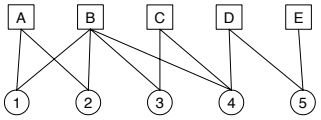
1. (5pts) **Which of the following networks has the highest clustering coefficient? (circle one)**



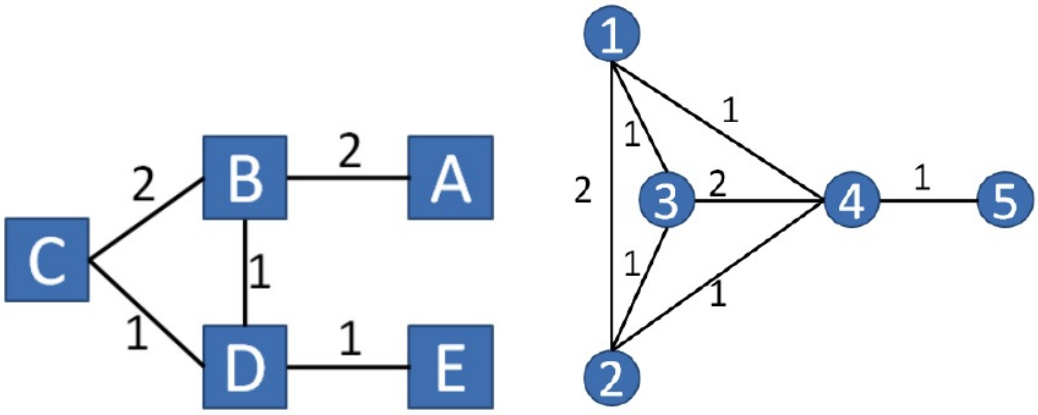
a b c

The correct answer is (a). This is the only network of the three that has a significant occurrence of closed triads. Triads contribute to the clustering coefficient whether it is calculated as a the average proportion of connected neighbor pairs over all the vertices, or as (3 times the number of triangles)/(number of connected triples) in the entire graph.

2. **(15pts) Draw separate one-mode projections of circles and squares with weights marked on the edges**

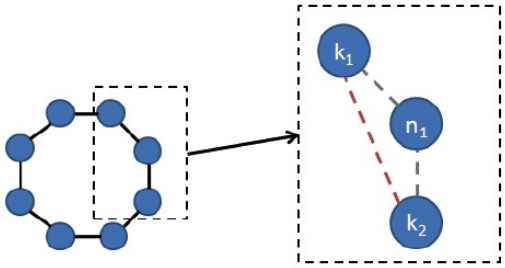


Here are the two one-mode projections:

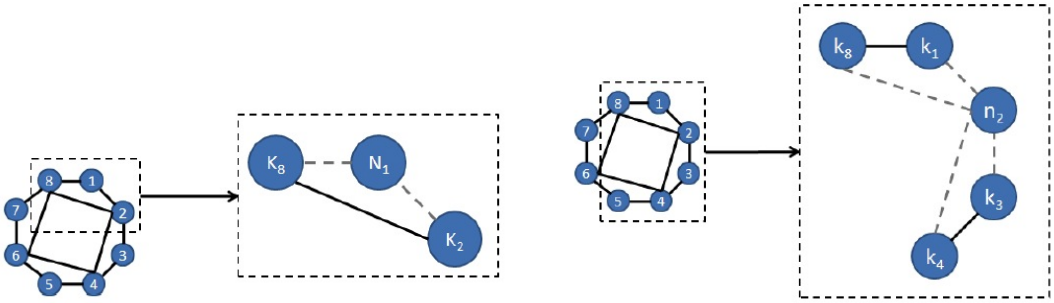


3. **a) (7.5pts) Start out with a ring lattice with each node being connected to *one* neighbor on each side (that's one neighbor to the left, and one neighbor to the right). Sketch the network. What is the average clustering coefficient for this network?**

The average clustering coefficient of this network is 0. There are no triads in this network.



b) (7.5pts) Assume that the number of nodes in the ring lattice is even, and that the nodes are numbered from 1 to n. In addition to the edges specified in (a), connect the closest even nodes to one another (e.g. node 4 connects to nodes 2 and 6, and node 2 connects to node n and node 4). Sketch the network. What is the average clustering coefficient?



odd nodes even nodes

All the odd nodes will have a clustering coefficient of 1 because they only have two neighbors and those two neighbors know one another.

The even nodes have four neighbors, and the two pairs of neighbors on either side know one another. That is 2 edges out of $\binom{4}{2=6}$ possible ones. So their clustering coefficient is $2/6 = 1/3$.

The average clustering coefficient for the whole network is therefore

$$\frac{1}{2}(1 + \frac{1}{3}) = \frac{2}{3} \tag{1}$$

4. **You join a circa 2000 peer to peer filesharing network, where each peer connects to a number of other peers. There is no centralized index that can tell you who has what file, you need to send a query to a node via the network to find out if it has a file you'd like. It goes like this: When a peer would like to download a file, they will send a query asking for that file to their nearest neighbors. Each neighbor will either report back that it has the file and is ready to share it, or will pass the query on to all of its neighbors. This is essentially breadth first search. The query will have a "time-to-live" stamp, that is decremented each time the query is passed on, so that e.g. if the query has a time-to-live of 2, it will only be passed on to the original node's neighbors and those neighbors' neighbors, but not farther than distance 2.**

   a) (7.5 pts) Assume that the files you would like to download are distributed randomly throughout the network and the p2p software protocol only allows you to initially connect to 2 other neighbors upon joining. Also assume (unrealistically) that you have a map of the entire network (just who is connected to whom, not what files they have) and that the nodes you request to connect to will accept your request. You are trying to maximize your chances of finding the files you'd like to download. Consequently, which one of the following three centrality measures would you try to maximize for yourself and why: betweenness, closeness, PageRank?

   Since my messages will only go out to a given radius specified by the time-to-live, I will want to maximize my closeness, so that my average shortest path to nodes in the rest of the network is the smallest and I have the greatest chance of my queries reaching them.

   b) (7.5pts) Now imagine you are a record company that has discovered that the files that are being shared are almost all music files. You have convinced an internet service provider to block some of the nodes' internet access, but they will only do this for 10 of their customers. Suppose (unrealistically) you have a map of the entire network. Which of the following centrality measures would you use to pick the 10 nodes in order to maximally disrupt the network and why: degree, betweenness, closeness, PageRank.

I would want to pick nodes with highest betweenness, since they lie on the shortest paths between other nodes, and without them, nodes will either not be able to reach one another at all, or the number of hops will be increased.

Degree is also a reasonable choice, though it is possible that a high degree node is redundant, and that its removal will not significantly disrupt the network.

Closeness is not a good criterion since a relatively insignificant node can be in the "middle" of the network simply because it attached to another node in the middle of the network.

Pagerank is not particularly relevant here, because the network is undirected, and PageRank will roughly correspond to the degree of a node.

5. **Imagine a similar P2P network. Assume the nodes really are sharing just music files. Also assume, contrary to the scenario in the previous question, that nodes can't be strategic about who they connect to initially, because they don't know the structure of the network. They connect to two nodes at random. Nodes are joining the network one by one and the network is growing. For simplicity, assume that nodes do not leave the network. Also assume that a node will have a favorite genre, e.g. jazz or country, and 80% of the files it is sharing will be in that 1 genre, as will 80% of the files it requests. The rest of its files and requests will be randomly chosen in different genres. We'll assume, unrealistically, that all files are equally popular and are each stored on roughly the same number of nodes. Nodes occasionally query for files, and when they do, they will form a new connection with the node that shared the file with them, and drop one of their previous connections. Assume that the queries are distributed as above (a node queries its nearest neighbors, who query their nearest neighbors, up to some distance).**

   a) (5pts) What do you expect to happen to the clustering of the network over time? Why?

   I expect the clustering to increase over time. There will be more "within-community" links, since nodes with the same favorite genre will exchange files more often (and hence form new connections to one another). Also, since nodes are "discovering" other nodes through their existing connections, they will also form triads in this way.

   b) (5pts) What is the average degree of a node in the network (approximation is OK)?

   The average degree is simply 4, since each new node arrives with 2 edges.

   d) (10pts) Describe how you would visualize the network to figure out as many interesting things as possible about the network by just looking at the visualization.

   I would size the nodes by degree to differentiate those nodes with high vs. low degree centrality. I would color them by their favorite genre, to see whether there are communities forming. I would change the width of the edges to reflect the number of files exchanged between two nodes in total, and perhaps color the edges according to age, to see where the recent links are forming.

   I would use a spring layout algorithm that could further reveal community structure.

   e) (5pts) Do you expect this network to have small world properties? Why or why not?

   I do expect the network to have small world properties. I expect it to have clustering due to the discovery of other nodes through nodes one is already connected to, as well as a tendency of nodes with the same genre preference to link to one another. At the same time, because the remaining 20% of the file requests are outside of the node's favorite genre, I would expect some "random" edges to form.

   f) (5pts) How large do you expect the largest weakly connected component to be? Give an explanation.

I expect it to encompass approximately the whole of the graph, since the average degree of a node is 4, a value above the percolation threshold of a random graph. It may occur that some nodes get disconnected entirely, if all of their neighbors switch away from them at once and no new connections are formed in the meantime. Given the current setup, such a node would not re-join the network.

6. **(20pts) (completely made up scenario) The IRB has approved your study of an illegal game (it's illegal because if people play too long, their heads explode). You tried to recruit subjects to interview through a web survey, but you got no responses: you will need to do face-to-face interviews instead. You know only one person X, who plays the game, and you plan to find others by asking X whom she plays with, and then asking them whom they play with etc. Your time is limited and you can interview only 25 people. You would like to interview as diverse of a set of people as possible (where diverse means "not alike"). The IRB has imposed some restrictions: You cannot ask any of the interviewees anything about their friends besides their names & email addresses. You can only find out more about a person by interviewing them directly, and if they decline to be interviewed, you remove their name and contact info from your data. Given your understanding of social network structure (including how it relates to people's attributes) and graph traversal, outline (in pseudocode or very structured English sentences) and justify a strategy for gathering your diverse sample of 25 willing to be interviewed.**

We learned about homophily in class – friends tend to be like their friends, so if we sample a single group of friends, our sample will not be very diverse. Therefore we want to do our traversal such that we go as far away from previously explored regions of the graph. We'll do this by keeping track of neighbors of nodes we've already added to our list

This is roughly the algorithm in pseudocode. We keep a list, C, of nodes who were nominated by someone we already interviewed. We avoid those nodes, in order to get as far away as possible. This is a depth-first search with the restriction that it tries to avoid its previous trail as much as possible.

```
mark all nodes white
C = {startingnode}
DFS(startingnode)

DFS(node) {
  new_players = {}
  color node gray
  stop if have interviewed 25 individuals

  for each name the node gives you that is white {
    if name not in C, add name to new_players
    add name to C if it is not already there
  }
  for each name in new_players {
    DFS(name)
  }
  color node black
}
```