PROBLEM SET 4 – Weeks 4-6

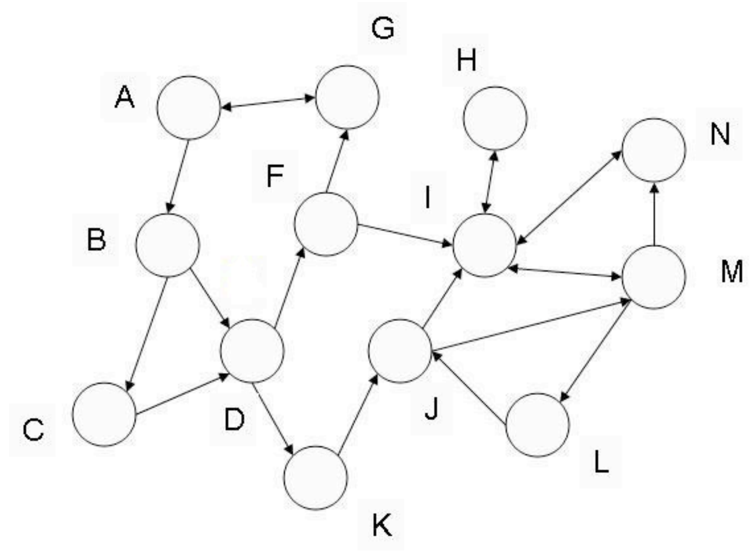
1. A real-world power law network

Download the file gnutella.net, a snapshot of a Gnutella peer-to-peer file-sharing network in the year 2000. You may wish to refer to http://www-personal.umich.edu/~ladamic/courses/si614w06/matlab/index.html and the lab from week 5 to figure out how to complete the tasks.

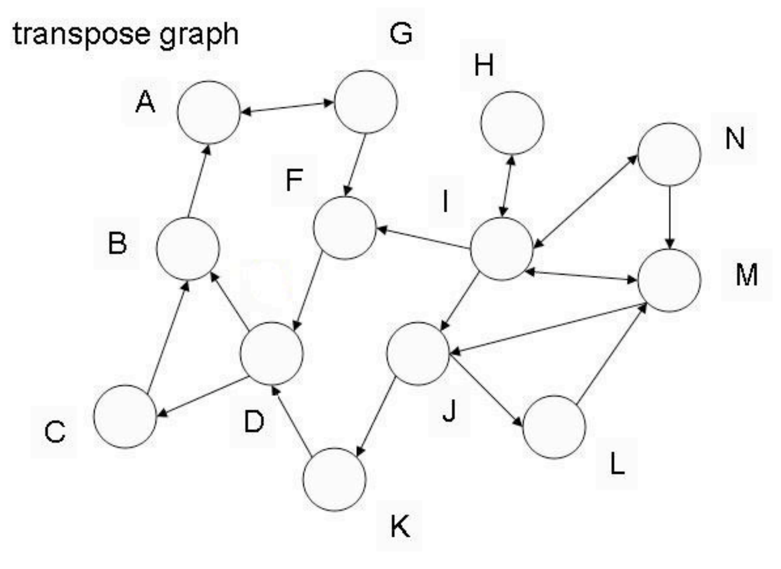
- Compute and plot (using your favorite plotting software, e.g. Matlab, etc.) the degree distribution of the network. Pajek will let you calculate the degree of each individual node (Net > Partitions > Degree > All). Then export the partition as a '.clu' file by clicking on the save icon to the left of the partitions drop-down select menu. Now you can import it into Excel or another program and histogram it. Try plotting both on both linear axes and log-log axes.
- Compute the average shortest path (Net>Paths between two vertices > Distribution of distances > From all vertices and look in the report window).
- Construct an Erdos-Renyi random graph with the same number of vertices and same average degree of each vertex (Net>Random Network>Erdos-Renyi). Check that the random graph you have obtained has the same number of vertices and a similar numbers of edges as the gnutella network. Calculate and export the degrees of all vertices.
- Turn in a plot that shows the two-degree distributions superimposed. In Matlab, you can plot two data sets together as follows: plot(x1,y1,'r-',x2,y2,'b:'). This will plot y1 vs. x1 as a red solid line, and y2 vs. x2 as a blue dotted line.
- Comment on the difference in degree distribution between the gnutella network and the equivalent Erdos-Renyi random graph? How does this explain the differences you observed in average shortest path in a previous lab (or you can recalculate the average shortest paths now).

2. DFS and strongly connected components

Please complete this part by drawing directly on the image of the network.



- Do a depth first traversal of the graph starting with the vertex "A".
- For each vertex, write the discovery and finish time inside the circle (e.g. "1/15" means discovered at time 1, finished at time 15).
- Mark the edges in your depth first forest (or tree) with a "t"
- Mark back edges with "b"s, forward edges with "f"s and cross edges with "c".
- Now do a depth first traversal of the transpose graph below in reverse order of finishing times.



- Write the discovery/finishing times on each vertex as you go.
- As you identify the strongly connected components, draw a boundary around them.

3. Getting started with Guess (visual exploration)

Download the file poliblogs.gdf from cTools. It represents the citation patterns between 40 A list blogs during a couple of months preceding the 2004 presidential election, along with the political leaning of those blogs. Open it in Guess (one way of doing this is by clicking the "Load GDF/GraphML" button after guess starts up). Do the following (submit just the final image and the list of commands you used).

1. Lay the network out using your layout algorithm of choice. Follow up with the center and rescaleLayout() commands, to adjust the position of the network and the size of the vertices.
2. Play with the zoomable interface and figure out how to reposition the nodes.
3. Use the information window to find out what attributes of nodes and edges are specified.
4. Color the conservative blogs red and the liberal ones blue. Color the edges differently depending on the leaning of the from and to nodes.
5. Compute the indegree for all nodes at once (no need to turn in) and resize the nodes according to indegree using the resizeLinear(indegree,minsize,maxsize) command, where you specify minsize and maxsize.
6. Change the width of the edges to reflect the number of citations (given with the 'weight' attribute) using the resizeLinear() command.
7. Make a couple of observations about the blog network and discuss whether modifying the visualization with the above steps helped you make them.
8. Save the commands in a .py file, and turn in along with your exported image (jpg, eps, etc.).

To repeat the process, you would only need to call execfile("yourfilename.py"), or select File>Run Script in the dropdown menu. To save your network with the new colors & positions, you could use the exportGDF("filename.gdf") command. You could also have created a persistent database when starting Guess.