

LAB 3 · CS249B · NETWORK ANALYSIS: METRICS AND MODELS

DANIEL BILAR, WELLESLEY COLLEGE

Out: Monday, April 7th, 2008; In: Wednesday, April 21st, 2008

1. GOAL

First, we are going to juxtapose real-life networks to model-generated networks and compare their respective metrics. Then, we are going to generate and analyze scale-free networks. Finally, You are going to devise a way of illustrating some claims of Barabasi's *Error and Attack Tolerances in Complex Networks*.

You may want to keep labs 1 and 2 (every lab builds on techniques from the previous ones) and the reference manual for Pajek handy: <http://vlado.fmf.uni-lj.si/pub/networks/pajek/doc/pajekman.pdf>

2. GATHERING METRICS FROM PROTEIN INTERACTION NETWORK

- (1) The dataset to be used is the *yeast protein-protein interaction* dataset. Start Pajek and load the file `YeastL.net` from <http://cs.wellesley.edu/~cs249B/assignments/lab3>
- (2) Fill in the second column ("Full Yeast Network") in Table 1. Look at previous labs for guidance to get the metrics.
- (3) Generate a randomly connected network with the same number of vertices and edges as the yeast network, using *Net* → *Random Network* → *Total No. of Arcs*. Using this network, fill in the fourth column ("Random Network") in Table 1
- (4) What are the differences between the statistics of the two networks? Why do you think these differences may exist?
- (5) The yeast network does not consist of a single connected component. To see how many components do exist, select the Yeast network again and use *Net* → *Components* → *Weak*. Select 1 as the minimum component size. Note that when you do this an entry appears in the Partition box of the Pajek interface, telling you how many components there are. How

TABLE 1. Network Metrics

Metric	Full Yeast Network	Yeast Largest CC	Random Network
Number of vertices			
Number of edges			
Diameter			
Most highly connected node			
Cluster Coefficient			

many are there?

To visualize the components, use *Draw* \rightarrow *DrawPartition*. Caution! It's a large network and you could spend a lot of time redrawing. In the graph window, click on *GraphOnly* and select *Energy* \rightarrow *Kamada - Kwai* \rightarrow *Free*. Each component has its nodes filled in with a different color. The largest component should have yellow nodes.

- (6) Select the largest connected component of the network using *Operations* \rightarrow *Extract from Network* \rightarrow *Partition*, and select clusters just from Partition 1. Fill in column three (Yeast Largest CC) in the table above.
- (7) The yeast network is an undirected network. One of the most interesting characteristics of a network is its degree distribution. Pajek will calculate the degree of each node, but in order to turn this data into a degree distribution, you must save the degree information as a Pajek cluster file (with a `.clu` extension) and import it into Matlab.

Calculate the degree distribution of the yeast network using *Net* \rightarrow *Partitions* \rightarrow *Degree* \rightarrow *All*. Note that when you do this an entry appears in the Partition box of the Pajek interface. Save the partition information by clicking on the save button and save the file under a name like `cs249Blab3.clu` in your home directory. Now we are going to fit the data from the Pajek exercise.

Procedure to load Pajek .clu file. Use the script `hdrload.m` on your saved `cs249Blab3.clu` file as follows: Start Matlab and at the prompt, type

```
> [header,mydata] = hdrload('<home_dir_path >/cs249Blab3.clu'),
```

where `<home_dir_path>` is the path to the folder where the `clu` files resides.

Then, with the data `mydata` compute the power exponent α along the lines you did in lab 1

□

3. PAJEK: GENERATE SCALE-FREE NETWORKS

- (8) Generate a scale-free network using Pajek's *Net* \rightarrow *Random Network* \rightarrow *Scale-Free* command. This produce a Pennock (2002) type scale-free network.

Make the network undirected, with 1000 vertices and an average degree of 4. Select $\alpha_{NOTslope} = 0.3$ (which will then fix beta), and choose a fully connected starting ER network of three vertices ($m_0 = 3$, $p = 0.9999$) to start with.

Note that here $\alpha_{NOTslope}$ is the weight given to the degree of the vertex in the model, not the power law exponent α !

- (9) Draw the network
- (10) Describe briefly a few characteristics you observe about the network.

- (11) Now generate a larger network of 10,000 with the same parameters (no need to draw it out - it will take a looong time)
- (12) Calculate the degree of each vertex (*Net* → *Partitions* → *Degree* → *All*) and save the values to a file by clicking on the save icon next to the partition name (it will be a .clu file).

4. MATLAB: ANALYZE THE DISTRIBUTION

Now we are going to find the most fitting line that goes through the data. In other words, we are going to find the exponent α of the power law $y = kx^\alpha$. We should keep in mind the power law relationship which makes it appear as a straight line on a log log plot:

$$y = kx^\alpha$$

$$\log(y) = \alpha \log(x) + \log(k)$$

which has the same form as the line equation

$$y = mx + c$$

where m is the slope of the line and c is the y -intercept (where the line crosses the y -axis)

- (13) Start Matlab and load the (.clu) file into Matlab.
- (14) Plot the degree frequency distribution as an X-Y scatter plot on a log-log scale.
- (15) Print out the graph and estimate visually the α value of the power law.
- (16) Calculate α (possibly three) with the procedures we used in previous labs. Is your visual inspection accurate? For which α ?
- (17) Go back and generate another scale-free network by changing the parameter $\alpha_{NOTslope}$ to a much lower value: $\alpha_{NOTslope} = 0.1$. Set the network size to 10,000 vertices and repeat the degree distribution fitting procedure above.
- (18) What do you observe about the degree distribution?
- (19) Why is lowering the value of $\alpha_{NOTslope}$ having this effect?

5. PAJEK: DESIGN EXPERIMENT ON ERROR AND ATTACK TOLERANCE

Fig. 2,3 and 4 in the *Error and Attack Tolerances in Complex Networks* paper showed evidence of the behaviour of networks subjected to outages. Your task is to corroborate some of these results with Pajek.

Pick any one figure - certain results are shown in the figure. Design an experiment in Pajek to corroborate these results. Approach this as you would pseudo-coding an algorithm: Take a piece of paper, write down the results that you want to corroborate and lay out the steps you need to be able to show this (generate SF networks with following parameters, etc).

- (20) Describe the results and figure you want to show and write down the pseudocode for the experiment setup.
- (21) Perform the experiment, illustrate your results in an intelligible way and compare them to the paper's results.

6. HANDIN

Please write up your solutions *neatly* (I cannot grade what I cannot read) or typeset them and hand them in in class at the due date. You are allowed to discuss the problems and solution approaches with your classmates; copying is not allowed. Please indicate which problems you did with whom.