GT CS 7280: Network Science

Assignment 2: Degree Distributions and Exploring Network Topology

Spring 2025

Overview

The objective of this assignment is to understand which how network features, such as the degree distribution, effect the overall structure and efficiency of the network: By the end of this assignment you should be able to:

- Determine the assortativity of a network
- Measure and interpret the degree distribution of a network
- Understand the differences between random networks and power-law networks
- Test for the small world property
- Understand the difference between transitivity and the average clustering coefficient

Submission

Please submit your Jupyter Notebook **GTUSERNAME-A2.ipynb** with **requirements.txt** so that we may be able to replicate your Python dependencies to run your code as needed.

With Anaconda, you can do this by running:
conda list -e > requirements.txt

Ensure all graphs and plots are properly labeled with unit labels and titles for x & y axes. Producing readable, interpretable graphics is part of the grade as it indicates understanding of the content – **there may be point deductions if plots are not properly labeled.**

Getting Started

For this assignment we will be using a python package called powerlaw to perform our analysis. This package is already included in the assignment zip file so the only thing you need to do to access it is include an `from powerlaw import *` statement in your notebook. Once you have the package imported, you are going to want to look through the library's documentation in the `\A2\powerlaw\manuscript\powerlaw.pdf` file. It contains useful information for how and why to create certain plots and statistical tests.

The data set you will be performing your analysis on is **"blog.txt"**. It represents a directed network that contains hyperlinks between blogs. A node represents a blog and an edge represents a hyperlink between two blogs. The format of each line in the text file consists of two numbers separated by space. The two numbers represent the two nodes and the line represents an edge from the first node to the second node (i.e., it is a directed and unweighted network).

Part 1 - Assortativity and The Friendship Paradox [15 points]

Part 1 focuses on analysis concepts covered in lesson 3. Although "**blog.txt**" is directed, **we have converted this to an undirected network for you in order to do the exercises in this section.** We will be using the network in its directed form for part 2. You may use scipy functions to complete the following exercises.

1. [6 points] First calculate the Pearson correlation coefficient for the degrees of adjacent nodes. Based on the results of this test, **is the network assortative, disassortative, or neutral?** Justify your answer using the p value from the Pearson-r.
2. [9 points] Next, visualize the friendship paradox by plotting the average neighbor degree (averaged across all nodes of degree k) as a function of the node degree k in a scatter plot using log-log scale. Compute the Pearson correlation coefficient between the node degree and average neighbor degree as computed above. Based on the results of this test, **is the network assortative, disassortative, or neutral?** Justify your answer and use a one sample t-test on the Pearson correlation coefficient to evaluate whether your answer is statistically significant.

Part 2 - Power-Law Distributions [20 points]

The first piece of analysis we want to perform on this network is plotting the degree distribution. This graph is directed, which means that we will need to perform a **separate analysis for the in-degree distribution and the out-degree distribution.**

1. [12 points] Plot the degree distribution in the four possible ways shown in Figure 4.22 of your textbook (separately for out-degree and in-degree) and please ignore all the nodes with degrees 0 in the log-scale plot. Use 20 for the number of bins in the histogram.
2. [8 points] The first step in fitting power-law distributions is determining which portion of the data to fit to. (See pages 5-6 of the documentation paper.) Because of the presence of "low-degree saturation" and "structural limit" in real networks, the power-law distribution is usually fit excluding values that are smaller or larger than a certain threshold.

Use the “**Fit**” function from “**powerlaw**” to estimate the exponent of the power-law degree distribution and the minimum-x value for the power-law fit and set **discrete = True** (The function treats the data as continuous by default) within the “**Fit**” function.

Do the estimation twice: once without setting the xmax threshold value, and once setting xmax to remove the maximum outlier value (we recommend you set **xmax = 200** for the out-degree distribution and **xmax = 300** for the in-degree distribution respectively). We recommend you ignore nodes with zero degree (otherwise the function “Fit” will “complain” giving you several warnings).

Part 3 - The Small-World Property [40 points]

Next, we want to compare the structure of the **blog.txt** network with random G(n,p) networks. We will then use the results of this analysis to determine if it satisfies the small-world property.

Clustering Coefficient and Transitivity [20 points]

1. [2 points] Find the **largest strongly connected component** of the directed network, and convert it to an undirected network. This undirected network will be our empirical network and we will label it as $G_{0.}$

2. [4 points] Next, calculate the **clustering coefficient** of each node in your empirical G_0. Plot the C-CDF of the clustering coefficients. Create a random G(n,p) network and perform the same calculation. Plot the C-CDF for the random with the empirical values using linear-linear scale.

3. [2 points] Use the Kolmogorov-Smirnov test to compare the two distributions you found in 3.2. You may use the scipy library for this. **Are the two distributions different?**

4. [6 points] Plot the **average clustering coefficient** on a scatter plot with log-log scale as a function of the node degree, for both G_0 and for the random network. **Describe the relationship you observe for each network.**

5. [6 points] Create 100 G(n,p) random networks with the same number of nodes and the same number of expected edges as G_0.Calculate the **transitivity coefficient** of the networks. Compare this value with the transitivity coefficient of G_0. using a plot. We recommend you use a “box and whiskers” plot, so that you can show both the median and the first and third quartiles.

CPL/ASPL and Diameter [15 points]

6. [10 points] Using the networks you created in part 3.5, calculate the **average shortest path length** and the **diameter** and create two plots comparing them with that of the empirical network. We recommend box and whiskers plots here as well.

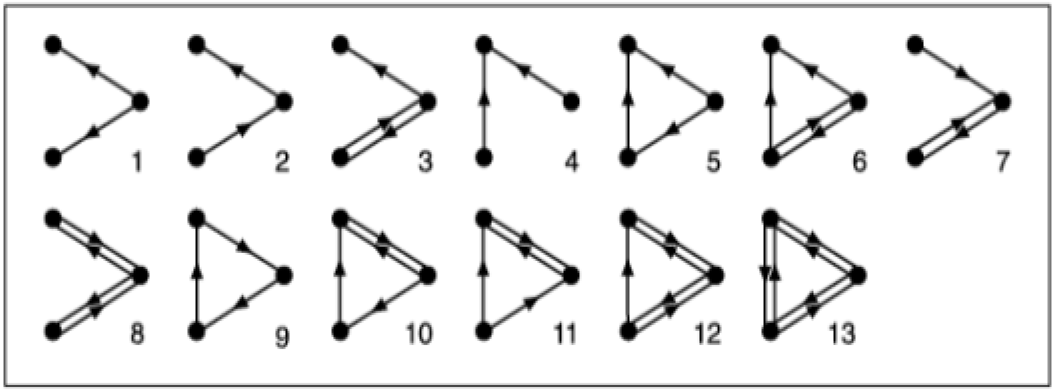
7. [5 points] For the average shortest path length and diameter, use the **one-sample t-test** to examine if each metric of the empirical graph is significantly different from the random networks at a 95% significance level. For each metric answer: **Are the values statistically different? Are they within the same order of magnitude** (e.g. are the values within a 10-fold factor of each other)?

Small World Property [5 points]

8. Based on your findings in 3.1-3.7, can you conclude that G_0 is a small-world network? **Explain which metrics you used and how they support your conclusion.**

Part 4 - Directed Network Motifs [20 points]

Recall that there are 13 different types of directed weakly-connected “node triplets”.



1. [10 points] First, count the number of type-5 (feed-forward loop : A->B->C with the additional edge A->C) and type-9 (directed cycle :A->B->C->A) in the **largest strongly connected component** (call it G1) of the original network. **You are not permitted to use a library function for this.**

2. [10 points] Second, using the “**directed_configuration_model**” function, generate 10 random networks that have the same number of nodes, edges, in-degree distribution, and out-degree distribution with G1. Remove any multi-edges between nodes by converting the random networks from “multi-directed graphs” into (single-edge) directed graphs. Also, remove any self-loop edges. Use these 10 random networks to examine statistically (using a t-test) which of the previous two triplet types are more (or less) common in G1 compared to chance. Visualize your results using a box and whiskers plot.

Hint: see the following links for additional help. Subquadratic Triad Census Algorithm Transitivity and Triads

Part 5 - Transitivity and the Average Clustering Coefficient [5 points]

Note that the Transitivity and the Average Clustering Coefficient are two different metrics. They may often be close but there are also some extreme cases in which the two metrics give very different answers.

To see that, consider a network in which two nodes A and B are connected to each other as well as to every other node. There are no other links. The total number of nodes is n. What would be the transitivity and average clustering coefficient in this case (you can simplify by assuming that n is quite large)? Points will only be awarded for a mathematical derivation, however you may use code to verify your result.