Advanced Tools for Complex Network Analysis
CNET 5052, Spring 2026

Prof. Brennan Klein and Dr. Milo Trujillo

Assignment 1, due by Friday, January 30, 2026 (a.o.e.)

Please submit a .pdf of your answers via Canvas, including a url to your .ipynb notebook in your fork of our class repository. Please name your Canvas submission as Lastname.pdf, (*e.g.* Klein.pdf). If you need help with LaTeX or GitHub, please visit us during office hours or post questions to the Slack channel.

0. *Question 0 (0 points).*

Create an `assignments/` folder in your fork of our class GitHub repository. Include your code in a (well-documented, commented, clear) `assignment01.ipynb` file. Scores for the code-based questions below will be based on your written responses *and* code.

1. *Watts & Strogatz, 1998 (15 points).*

The "Watts-Strogatz" model[1] is a simple random network model that produces graphs that have 1) the small world property and 2) high clustering. Reproduce the canonical figure[2] from the original 1998 article, and comment on the key insights from the figure.

2. *Visualization (15 points).*

Download the `netscience` dataset at this URL: `https://tinyurl.com/bd6cwsd4`. As described in the `.txt` file, this is a coauthorship network of scientists working on network theory and experiments, compiled by Mark Newman in 2006. Visualize the largest component of the network (Gephi may be the easiest choice of software, but you are free to use your favorite), considering the following:

- Degree or other centrality measures
- Edge properties
- Layout

Make good use of color, size, and layout to create a clear, informative visualization. Describe your approach and comment on your observations (about 5 sentences).

---

[1] Watts, D. & Strogatz, S. Collective dynamics of 'small-world' networks. *Nature* **393**, 440–442 (1998). `https://doi.org/10.1038/30918`.

[2] `https://www.nature.com/articles/30918/figures/2`

3. *Network Generation and Growth (30 points).*

Part I: Read Chapter 5 of *Network Science*, by Albert-László Barabási[3] (hard copy of textbook available in my office) on the origin of the Barabási-Albert network model.

(a) Write code that generates a Barabási-Albert network with $N = 10^4$ and $m = 4$. Start with a fully connected 4-node network as initial condition. (Note: Various network libraries including `NetworkX` in Python already have implementations of the Barabási-Albert model, but you will need to implement it yourself in order to do parts the following sub-questions.)

i. Plot the degree distribution at intermediate steps in the network's growth. Plot these degree distributions when the network has 10^2, 10^3, and 10^4 nodes.

ii. Compare the distributions at these intermediate steps by plotting them in a single figure and fitting each to a power-law with degree exponent γ (include the γ values for each distribution somewhere in the plot). Do the distributions "converge"? Use the `PLfit` Python package to fit degree exponents.

iii. Measure and plot the average local clustering coefficient as a function of N.

iv. Following **Figure 5.6a** in Chapter 5, measure the degree dynamics of one of the initial nodes and of the nodes added to the network at times $t = 100$, $t = 1,000$ and $t = 5,000$.

Part II: In Section 5.9 of the textbook, we are introduced to multiple ways of generating networks with heavy-tail (scale-free) degree distributions: the Link-Selection Model, the Copying Model, and Optimization Model.

(b) Implement the Optimization Model.

(c) Using your implementation of the Optimization Model, reproduce Figure 5.16b (degree distributions when $\delta = 0.1, \delta = 10$, and $\delta = 1,000$) and Figure 5.16c (visualizations of the networks at each δ value; Note: your visualizations do not need to look exactly like the networks from the textbook, but they should capture similar information).

(d) Implement the Link-Selection Model and the Copying Model.

(e) Create a figure with four subplots—inspired by the middle panel of Figure 5.16d—that plot the preferential attachment measure $\Pi(k)$ of networks sampled from each of the following network generating models: 1) the Barabási-Albert model (with $N = 10,000$, $m = 1$), 2) the Copying Model, 3) the Link-Selection Model, 4) the Optimization Model (with $\delta = 10$). Note the caption of Figure 5.16: "We used the method described in Section 5.6 to measure the preferential attachment function. Starting from a network with $N = 10,000$ nodes we added a new node and measured the degree of the node that it connected to. We repeated this procedure $10,000$ times, obtaining $\Pi(k)$."

(f) Describe what you observe in the figure created above.

[3] `https://networksciencebook.com/`

4. *Graph Distance Measures (25 points).*

Create a graph distance measure, or implement one that you found in the literature.

(a) Describe the graph *descriptor*, $\phi(G)$, you use in your distance measure. Justify why this is a good descriptor to use.

(b) Describe the *distance* measure you use to quantify dissimilarity between pairs of graphs, G_1 and G_2.

(c) Implement your graph distance in two functions: the first function should compute the $\phi(G)$ for an input graph, G, and the second function should call the first in order to output the final distance value. For example, if your graph distance was the *Degree Divergence*, your descriptor $\phi(G)$ would be a function that takes a graph as input and outputs the degree distribution of G. Your distance function would then take two graphs as input, create a $\phi(G_1)$ and $\phi(G_2)$, and calculate the distance $d(\phi(G_1), \phi(G2))$. You may not use the Deg. Div. as your graph distance.

(d) For 1,000 iterations, sample pairs of Erdős-Rényi graphs with $N = 500$ nodes and $p = 0.016$. Calculate and store the graph distance between each pair. Plot the distribution of distance values between the ER networks.

(e) For 1,000 iterations, sample pairs of Barabási-Albert graphs with $N = 500$ nodes and $m = 4$. Calculate and store the graph distance between each pair. Plot the distribution of distance values between the BA networks.

(f) For 1,000 iterations, using the same parameterizations above, sample one ER and one BA graph, and compute and store the distance between them. Neatly visualize all three distributions on the same plot, with appropriate legends and color. What do you notice?

5. *Sparsification & Robustness (15 points).*

A classic intuition in network science is that heavy-tailed networks are robust to random failures but fragile to targeted attacks because of hubs. For each ensemble below, generate graphs with $N = 5,000$ nodes and approximately the same expected mean degree $\langle k \rangle \approx 10$:

- **Erdős-Rényi**: $G(N, p)$ with p chosen such that $\langle k \rangle \approx 10$.
- **Barabási-Albert**: preferential attachment with m chosen so $\langle k \rangle \approx 10$.
- **Random regular**: a random d-regular graph with $d = 10$.

For each ensemble, generate at least 10 independent graph realizations (using different random seeds) and average results across realizations.

(a) For each graph, simulate the following node removal from the network (i.e., delete the node and all of its incident edges):

(i) *Randomly:* remove a fraction f of nodes uniformly at random, or

(ii) *Targeted:* remove a fraction f of nodes in descending order of degree.

For each of these simulations, use $f \in \{0.00, 0.02, 0.04, \ldots, 0.40\}$.

(b) After each removal, compute

$$S(f) = \frac{\text{size of the largest connected component after removal}}{N}.$$

Plot $S(f)$ versus f for each ensemble, with two curves per ensemble (random failure vs. targeted attack). Include uncertainty across realizations (e.g. shaded ± 1 standard deviation or error bars at selected values of f).

(c) In a few sentences, discuss your findings and observations.