Assignment 1

COMP 599: Network Science

Due on September 20th 2021

an be submitted individually or in groups of up to two.
→ these give quick tips and are not requirements.

1. Choose three datasets in the Barabasi book, compute and report [65%]:
   → make them simple graphs, i.e., remove self-loops, multi-edges, and directions (i.e., make them symmetric).

   (a) node/edge sizes, number of connected components, size of the giant/largest connected component,
   → useful functions: numpy.loadtxt(), scipy.sparse.csc_matrix(), scipy.sparse.csgraph.connected_components()

   (b) degree distribution, and it's power-law fit (also plot the line and report the slope),
   → useful functions: scipy.sparse.csc_matrix(), numpy.polyfit()
   → bin and count to get the distribution and plot in log-log scale.

   (c) shortest paths distribution (also compute/report the average),
   → useful functions: from scipy.sparse.csgraph
   → For larger graphs, consider sampling a fraction of nodes (pairs) to estimate the distribution shape.

   (d) clustering coefficient distribution (compute the average as well),
   → compute if for a (sampled) set of nodes and convert the sequence to distribution similar to (a).
   → same tip (↑) for large graphs also applies here and to (f)-(g).

   (e) eigenvalue spectrum (also compute/report the spectral gap),
   → useful functions: scipy.sparse.linalg.eigs() → compute/plot only 100 first eignevalues, ordered by rank.

   (f) degree correlations (use a scatter plot for d_i vs. d_j, also report the overall correlation),
   → plot degree of source vs. degree of destination, axes would be 0 to the max degree.
   → use a scatter plot: a point is plotted per each edge positioned by the degree values of its incident nodes.
   → alternatively, you can plot this correlation as instructed in Barbasi's book.
   → use counting, binning, or plot edges/points with low intensity to capture regions with high density.

   (g) degree-clustering coefficient relation (plot as scatter d_i vs c_i)
   → plot degree of node vs. its clustering coefficient.
   → to manage points plotted over each other, use the same tip (↑) for i.e., use binning, density plot, or transparency.

2. Report the computational complexity for (a)-(g), as well as the space and time complexity of loading the graphs [5%]

3. Implement the Albert-Barabasi graph model, compute the same (a)-(g) distributions for three synthetic networks generated by this model, with parameters set to create graphs of similar size as the graphs you have chosen in part 1. [30%]
   → same number of nodes, estimate edges added in each iteration based on the total number of edges

4. **[Bonus:]** Tweak the Albert-Barabasi graph model and report the effects on the observed patterns [5%]

Submit the report in pdf and code as separate attachments in the Mycourses portal.