UPPSALA UNIVERSITET
DEPARTMENT OF MATHEMATICS
Denis Gaidashev

Modeling Complex Systems (1MA256)
Spring 2024

LAB 4

The deadline for this assignment is May 12nd.

Please submit in Studium or by email. All code should be included, feel free to submit videos illustrating your results where appropriate, via Studium or uploaded elsewhere such as vimeo, youtube or Jupyter Notebook. You may work in groups of size 1-7, and only one group member needs to submit the assignment. State clearly the members of the group.

Networks and random graphs

Task 1, 3 points:

1) Pick any data set (with a number of vertices not too high) in here.
2) Write a code that reads this data set, and draws the data set as a graph.
3) Try to identify different communities, ex, visually. If you indeed choose to identify them visually, then choose small data sets from the database.

   Alternatively, if you feel ambitious, you can use the so called Leiden algorithm, whose implementation in python is in the igraph package, functions cluster or community_leiden.
4) After having empirically identified the communities, compute the modularity of this network. Please, write the code for the modularity computation yourself, do not use the igraph functions.

Task 2, 4 points: Write a code that generates a random Erdös-Rényi graph $G(n,p)$.

1) Verify that for the probability $p < \frac{(1-\epsilon)\ln n}{n}$ the graph $G(n,p)$ is almost surely disconnected: generate many such graphs with some large n, and count how many of them have an isolated component.
2) Verify that for $p > \frac{(1+\epsilon)\ln n}{n}$ the graph $G(n,p)$ is almost surely connected: : generate many such graphs with some large n, and count how many of them have no isolated component

Verification of disconnectedness or connectedness can be automatized. The most inefficient algorithm is as follows.

Start with a node i. Start taking powers of the adjacency matrix, up to m - the number of edges in the graph. Check if a_{ij} is non-zero in any of these powers - then there is a path from i to j. If for some j these elements are always 0, then there is no path from i to j, and the graph is disconnected. If i is connected to all j's, then move to the next vertex.

Or read about more efficient algorithms here under Computational aspects.

Task 3, 4 points: Write code that computes the global clustering coefficient C of $G(n,p)$ (see wiki for the theory):

$$C = \frac{\sum_{i,j,k} A_{ij}A_{jk}A_{ki}}{\sum_i n_i(n_i-1)} \tag{1}$$

where $n_i = \sum_j A_{ij}$. Notice that the numerator here is the trace of A^3.

Fix some very large n. Plot expected C of $G(n,p)$ as a function of p. (That is, now that n has been fixed, pick a p, generate many $G(n,p)$'s for that p, and compute the expected value of C.

Repeat this for some discretization of $[0, 1] \ni p$. In this way, you obtain the plot of the expected value of C as a function of p.)

Note: There is a theoretical result that says that the expected global clustering coefficient for the Erdös-Rényi model is $C = p + O(n - 1/2)$. Compare with you graph!