In this session:

- We are going to compute several metrics on two network models

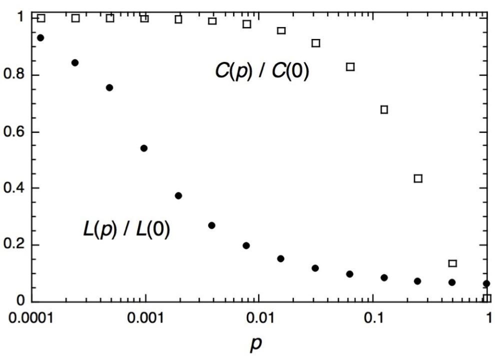
1 Network models

Erdős-Rényi model (ER model). The ER model takes two parameters: n, the number of vertices in the resulting network, and p, the probability of having an edge between any two pairs of nodes. A graph following this model is generated by connecting pairs of vertices with probability p, independently for each pair of vertices. Erdős-Rényi graphs have $\binom{n}{2}p$ edges in expectation.

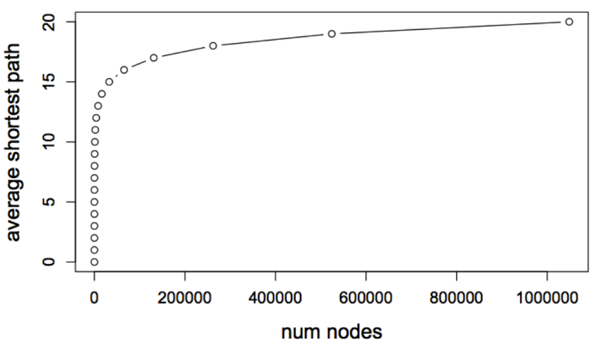
Watts-Strogatz model (WS model). The WS model takes two parameters as well: n, the number of vertices in the resulting network, and p, the probability of rewiring the edges in the initial network. A graph following this model is generated by initially laying all nodes out in a circle, and connecting each node to its four closest nodes. After that, we randomly reconnect each edge with probability p.

2 Your task

Your task is to reproduce these graphs introduced in our first lecture on networks. Feel free to use network analysis software tools such as `networkx` or `igraph`.



(a)



(b)

That is, your task is to: (a) plot the clustering coefficient and the average shortest-path as a function of the parameter p of the WS model, and (b) plot the average shortest-path length as a function of the network size of the ER model.

Notice that in order to include both values — average shortest path and clustering coefficient — in the same figure in plot (a), the clustering coefficient and the average shortest-path values are normalized to be within the range $[0, 1]$. This is achieved by dividing the values by the value obtained at the left-most point, that is, when $p = 0$.

In case of plot (b), you will have to experiment with appropriate values of p which may depend on the parameter n. You will notice that for large values of n your code may take too long, compute values for n that are reasonable for you. Also, make sure that you chose values for p that result (with high probability) in connected graphs. To achieve this, you can use a result from [1] stating (in the following, think of ϵ as a small positive real number):

- If $p < \frac{(1-\epsilon)\ln n}{n}$ then a graph in $G(n, p)$ will almost surely contain isolated vertices, and thus be disconnected
- If $p > \frac{(1+\epsilon)\ln n}{n}$ then a graph in $G(n, p)$ will almost surely be connected

3 Deliverables

To deliver: **You must deliver a brief report (1 or 2 pages) describing your results and the main difficulties/choices you had while implementing this lab session's work. You also have to hand in the source code of your implementations.**

Rules: 1. No plagiarism; don't discuss your work with others, except your teammate; if in doubt about what is allowed, ask us. 2. If you feel you are spending much more time than the rest of the group, ask us for help. Questions can be asked either in person or by email, and you'll never be penalized by asking questions, no matter how stupid they look in retrospect.

Procedure: Pack all these files into a single one (zip, gz, tar...), and deliver it via the Racó - "Practiques via web".

Deadline: Work must be delivered within 2 weeks from the lab. Late deliveries risk being penalized or not accepted at all. If you anticipate problems with the deadline, tell me as soon as possible.

References

[1] Paul Erdős and A Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hungar. Acad. Sci*, 5:17–61, 1960.