INSTITUTO SUPERIOR TÉCNICO | UNIVERSIDADE DE LISBOA
NETWORK SCIENCE 2021/22
Problem Set 2

1. Erdős–Rényi (ER) random networks. Consider an ER network with $N = 3\text{x}10^3$ nodes, connected to each other with probability $p = 10^{-3}$. Estimate, analytically, the following quantities:

a) What is the expected number of links, <E>?
b) In which regime is the network?
c) Calculate the probability p_c so that the network is at the critical point.
d) Estimate the distance between two randomly chosen nodes <L>.
e) Calculate the distance between two randomly chosen nodes <L> for the network at the critical point (i.e., if $<k>=1$).
f) Generate an instance of this network using your favorite language, *gephi*, *mathematica*, *matlab*, or any other way, and plot its degree distribution.

2. Propose an algorithm capable of creating a graph with an arbitrary degree distribution.
Suggestion: Check the so-called "configuration model" nicely discussed in
https://dl.dropboxusercontent.com/s/ofq0c8wiaoh29bk/Lab02-Configuration_model.pdf?dl=0

3. (Small-world networks) Propose an algorithm able of creating a graph with Poisson degree distribution (as in the case of a ER random network), yet with a large cluster coefficient.

4. (Watts and Strogratz model) Compare the degree distribution obtained from Watts-Strogratz model with p=1 and a degree distribution obtained from the ER random network model of the same average degree.

5. (Scale-free networks) Consider a network of collaborations among Hollywood actors, which portrays a power-law degree distribution. This network has 702 388 nodes, 29 397 908 edges, average degree <k>=83.71, and a power-law exponent of 2.12. Estimate the expected maximum degree k_{max} for this undirected network.

6. Choose the topic of your 1st project and let us know if you need any help.

7. (Plotting degree distributions) Let us play with some tools and data. Download an Internet snapshot from github (or any other dataset):
https://gephi.org/datasets/internet_routers-22july06.gml.zip

a) Plot the degree distribution this network in a log-log scale[1].
b) If you find a straight line (i.e., a power-law) try to estimate the exponent the power-law through a linear fit of the cumulative degree distribution. For additional reading on how to plot power-laws see here: https://www.dropbox.com/s/7y00i2l47it5s7c/Lab02-Plotting-powerlaws.pdf?dl=0

8. (Advanced methods for the estimation of power-law exponents) Likely, in 7-b) you faced several difficulties when you tried to fit a power law. The most important difficulty is the fact that the scaling is rarely valid for the full range of the degree distribution. Rather one observes small- and high- degree cutoffs which are not always obvious to find. In the following URL you will find a brief overview on the most used method of estimating the exponent of a degree distribution taking into account this problem:

https://www.dropbox.com/s/d3pmk9yypzwu1lj/Lab02-estimating_exponents.pdf?dl=0

For details (and code) on this procedure, see here:
http://tuvalu.santafe.edu/~aaronc/powerlaws/

[1] You may open the gml file with gephi, yet you will need to export the list of degrees to your favorite software to perform the following steps.

As an example, let us use R to fit a power law to a degree distribution following this method, and plot the result. For simplicity, let's resort to the input file enron.outdegrees (available here: https://dl.dropboxusercontent.com/s/e1kr9ri2btfwg9v/enron.outdegrees?dl=0) and enron.outdegree (available here: https://dl.dropboxusercontent.com/s/79n6w3p3cyat5lx/enron.outdegree?dl=0) created with webgraph in exercise 6 of our problem set 1.

```
[aplf@darkstar lab02]$ R
> install.packages("igraph")
> library('igraph')
> degs <- read.table("enron.outdegrees")
> deg_pmf <- read.table("enron.outdegree")
> degs_pl_fit <- power.law.fit(degs$V1)
> degs_pl_fit > plot(deg_pmf$V1, log="xy", xlab="degree", ylab="#vertices")
> plot(rev(cumsum(rev(deg_pmf$V1))), log="xy", xlab="degree", ylab="#vertices")
> plot(sort(degs$V1, decreasing = TRUE), 1:length(degs$V1), log="xy", xlab="degree",
ylab="rank")
```

Solutions

1.

a) $\langle E \rangle = p\dfrac{N(N-1)}{2} = 4498{,}5$

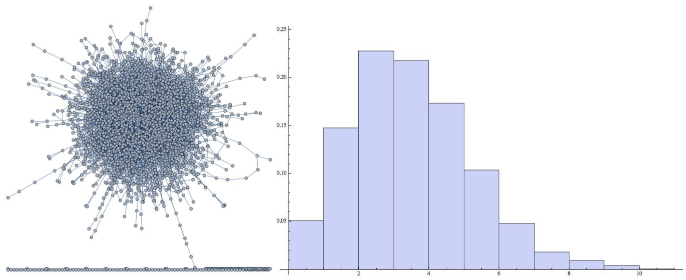
b) $\begin{array}{l}\langle k \rangle \sim p(N-1) \sim 3.0 \\ \langle k \rangle > 1.0 \quad and \quad \langle k \rangle < \ln N\end{array}$, thus it's supercritical!

c) $\langle k \rangle = p(N-1) = 1 \Rightarrow p \sim 3\times 10^{-3}$

d) $<L>\sim \dfrac{\ln N}{\ln\langle k \rangle} = \dfrac{\ln 3000}{\ln 3} = 7{,}28$

e) $<L>\sim \dfrac{\ln N}{\ln\langle k \rangle} = \dfrac{\ln 3000}{\ln 1} = \infty$

f)



3. Hypothesis 1. Choose an intermediate p value in the WS model. Hypothesis 2. If you wish to have exactly the same degree distribution you would have with the ER model or with the WS model with p=1, you may start by creating one of these networks (which offer low CC). Then, repeatedly swap the edges of randomly chosen pairs of links, a procedure which keeps the degree distribution untouched. In each swap, evaluate if the clustering coefficient increases, accepting the change when it does. This method can be used for any starting network and degree distribution.

4. Compare the minimum degree of both networks. In the WS model the minimum degree is truncated at $<k>/2$. All the rest looks alike.

5. If we have a power law, our degree distribution is given by $P_k = \left[(\gamma-1)k_{\min}^{\gamma-1}\right]k^{-\gamma}$. Thus, k_{max} will be such that

$\int_{k_{\max}}^{\infty} P_k \, dk < \dfrac{1}{N}$, which yields $k_{\max} = k_{\min} N^{\frac{1}{\gamma-1}}$. Assuming a minimum degree of $k_{\min} = 1$ we get

$k_{\max} = (702388)^{\frac{1}{2.12-1}} = 166019$.