Problem Set Two, CR07 ENS de Lyon. Course authorities : Karsai / Pacini

Due date : 20/11/2018, by 11:59 p.m.

Complementary material : N/A

1 The Erdös-Rényi random graph model

Here we calculate some properties of the ER random graph, a staple model in graph theory. There are two closely related variants of the ER model. First, the $G(n, p)$ graph, with n nodes and a constant probability p of an edge existing between any pair of nodes. Second, is the $G(n, m)$ graph, which has n nodes with m edges randomly distributed between node pairs. Both variants have the restriction of being simple graphs, disallowing loops and multiple edges. This means that, as nodes are distinguishable, each graph is drawn from the sets $G_{n,p}$ and $G_{n,m}$ with probability

$$p^m(1-p)^{C_2^n - m}, \tag{1}$$

where C_2^n is the binomial coefficient. We can think of p as a parameter that interpolates between empty graphs, with $p = 0$, and complete graphs, with $p = 1$.

1.1 Generating ER graphs

Write two functions, *er_np* and *er_nm*, that generate Erdös-Rényi graphs, each depending on two input paramaters, n and p or n and m. Although a simple graph is guaranteed in *er_np* as we are concerned with pairwise probabilities, you will have to ensure that your function *er_nm* does not contain multiple or self edges.

1.2 Degree distributions of random graphs

Write a function *compare_edge_count* that takes in the parameters n and p, and outputs the ratio of the number of edges in the corresponding $G_{n,p}$ and $G_{n,m}$ graphs, where $m = \lfloor p \times C_2^n \rfloor$. Show, using plots if necessary, that your graph generation methods obey the theoretical degree distribution

$$P(k) = C_k^{n-1} p^k (1-p)^{n-k-1}, \tag{2}$$

where $P(k)$ is the probability of a node having degree k.

2 The Erdös-Rényi model and connected components

In their 1960 paper, Paul Erdös and Alfréd Rényi provided a deep analysis of the class of graphs $G_{n,p}$, studying the characteristics of the graphs as a function of the parameter p. Their results demonstrated that

- if $np < 1$, a graph in $G_{n,p}$ will almost surely have no connected components of size larger than $\mathcal{O}(\log n)$,
- if $np = 1$, a graph in $G_{n,p}$ will almost surely have a largest component whose size is $\mathcal{O}(n^{2/3})$,
- if $np \to c > 1$, where c is a constant, then a graph in $G_{n,p}$ will almost surely have a unique giant component containing a positive fraction of the vertices. No other component will contain more than $\mathcal{O}(\log n)$ vertices. Furthermore,
- if $np < (1-\epsilon)\log n$, a graph in $G_{n,p}$ will almost surely contain isolated vertices, and
- if $np > (1+\epsilon)\log n$, a graph in $G_{n,p}$ will almost surely be connected.

From the final two points, we observe that $\log n/n$ is considered a sharp threshold with repect to p. Note however that we are normally working in the limit of large n, where $p > \log n/n$ is satisfied anyway.

In this section, the goal is to carry out your own analysis to convincingly illustrate that your random graph methods obey the above properties. Remember that Erdös and Rényi proposed a probabilistic model, so you are not expected you to provide a proof. Rather, design some experiments showing that under the right conditions, the above characteristics are exhibited with high probability.

3 The configuration model of random networks

It is often desirable to generate networks with an arbitrary degree distribution, using a method known as the configuration model (*Networks*, Newman). While the Erdös-Rényi model is useful and ubiquitous, it necessarily corresponds to a binomial, or Poisson degree distribution. More generic network models allow for an arbitrary degree distribution, for instance k-regular, normal, or log-normal distributions, whilst remaining maximally random otherwise. In this section, we develop some basic functions required for network generation under this model, and examine the limitations of the resultant networks.

3.1 Degree sequences obeying a given distribution

Since a normalised degree distribution doesn't contain information about the size of the network, we sometimes favour the degree-sequence description. In a network of size n, the degree sequence is a list

$$\{k_1, k_2, k_3, \ldots, k_n\} \tag{3}$$

containing the degree k_i of node i. Write two functions, *degree_sequence_regular* and *degree_sequence_lognormal* that output degree sequences of length n following these distributions. The input for the k-regular network will be n and k, whereas the input for the lognormal distribution will be n, μ and σ, or the average and standard deviation of the distribution.

3.2 Attaching stubs to form a network

A degree sequence can be a viewed as a list of nodes, each with k_i half-edges or stubs attached. The subsequent list of stubs can be randomly wired to form a network. Write a function called *configure_sequence* that inputs a degree sequence, and outputs the graph obtained by randomly attaching pairs of stubs corresponding to that sequence. Multiple edges and loops are natural consequences of the model definition, and are allowed in your output.

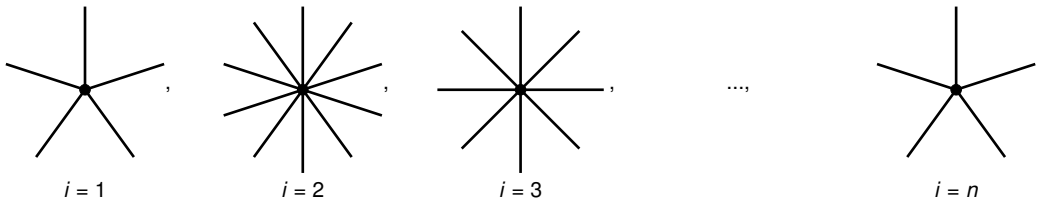

Figure 1: The stub representation of a degree sequence $\{5, 10, 8, \ldots, 5\}$

3.3 Counting double edges and self loops

Write a function *irregular_edge_count*, that inputs a graph and outputs the fraction of edges that are double-edges or loops. Illustrate the relationship between network size n, and the fraction of multiple edges and loops resulting from the random attachment of pairs of stubs. What can you say about the configuration model and the generation of simple graphs?