

[INFO-F409] Learning Dynamics

Second assignment

BUI QUANG PHUONG Quang Linh
Université libre de Bruxelles - ULB ID : 000427796
MA1 Computer Sciences

December 2018

Contents

1	Part I: Complex networks	2
1.1	1 - Erdos-Renye network	2
1.1.1	Q1 statement	2
1.1.2	Q2 statement	2
1.2	2 - Barabasi-Albert network	3
1.2.1	Q3 statement	3
1.2.2	Q4 statement	4
1.2.3	Q5 statement	4
1.2.4	Q6 statement	4
1.2.5	Q7 statement	4
1.2.6	Q8 statement	4
1.2.7	Q9 statement	4
1.2.8	Q10 statement	4

1 Part I: Complex networks

1.1 1 - Erdos-Renye network

1.1.1 Q1 statement

Generate Erdos-Renye network (Random networks) [1,2]. Generate the network from scratch and present/describe the part of the code you used to generate the network in your document. For a size of the network of $N=10000$, calculate a K so that each node has in average degree 4.

Pseudo-code

Algorithm 1 Generation of Erdos-Renye network

```

1: procedure ERDOSRENYE( $n, K$ )
2:   Initialize the network edges list
3:   for each 'future' edges  $\equiv 0 \rightarrow K$  do:
4:     Pick two random nodes in the network
5:     Define newEdge variable taking the 2 picked nodes
6:     if newEdge not already exists then:
7:       Add newEdge in the list of edges
   return list of network's edges

```

Algorithm description

The Erdos-Renye network is a random network generated. Before all, we have to initialize a list of edges which will be returned and will form the random network. This edges list is composed by some tuples of 2 elements (x, y) representing the edges where x, y are the two nodes that the edge is connecting. Now, we have to keep creating edges while the number edges K passed in parameters isn't reached. The random part is managed by the step of picking nodes by pair (x, y) in the network which is totally random, all nodes can be picked at this moment. Of course, before accepting the edge between the two nodes chosen, a verification that this edge doesn't already exist, which means that the edge (x, y) **and** (y, x) isn't already created, is done.

K for average degree of 4

To calculate the value of K for an average degree of 4, first we know that an edge generates a link between two nodes and then increases the degree of two nodes. Moreover, we have 10000 nodes, thus we can calculate the value of K for an average degree of 4 using the following formula : $\frac{2K}{10000} = 4$ that gives $K = 20000$.

1.1.2 Q2 statement

Plot the degree distribution of the generated network. Calculate the mean and standard deviation and plot the normal distribution with these same parameters

Degree distribution

The degree distribution of the network generated by the Erdos-Renye algorithm with $K = 20000$ is given by the Figure 1.

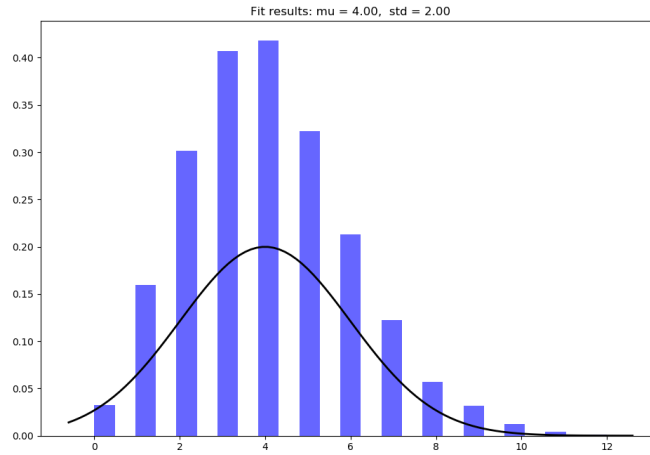


Figure 1 – Distribution for Erdos-Renye generated network

Mean and standard deviation

The value of the mean in the distribution of Figure 1 is

$$\mu = 3.998$$

and for the standard deviation is

$$\sigma = 2.012$$

1.2 2 - Barabasi-Albert network

1.2.1 Q3 statement

Generate a Barabasi-Albert network (Scale Free network) [1,3]. Generate the network from scratch and present/describe the part of the code you used to generate the network in your document.

Pseudo-code

Algorithm 2 Generation of Barabasi-Albert network

- 1: **procedure** BARABASIALBERT(*Graph*, *N*)
 - 2: Initialize a fully connected 4 nodes network
 - 3: **for each** 'future' node until $N \equiv 4 \rightarrow N$ **do**:
 - 4: Add a node *x* to the network
 - 5: **for each** current network's nodes **do**:
 - 6: Compute the probability of the added node *x* being linked to those nodes
 - 7: Append it into a list *probasLinking*
 - 8: Pick 4 nodes with their associated probabilities just computed before in the network's nodes
 - 9: **for** the 4 picked nodes **do**:
 - 10: Link the added node *x* to them
 - 11: Reinitialize *probasLinking* and increment by 8 (4 new edges = 8 nodes with their degree increasing by 1) the total degree number for the computation of the next probabilities
-

1.2.2 Q4 statement

Plot the degree distribution of the generated network using a linear scale on both axes. Plot in the same figure an exponential distribution which looks similar and reports on the parameters of that distribution.

Degree distribution result (linear scale)

Exponential distribution

1.2.3 Q5 statement

Plot the same distribution on log-log scale. Fit the distribution using Least Square fit. You can use existing functions for fitting and plot the fit next to the data. What are the parameters of the fit? How does it fit? Why? Write a paragraph about why we should not use Least Square fit to fit power laws.

Distribution on log-log scale

Fitting using Least Square Fit

1.2.4 Q6 statement

Plot cumulative distribution and fit it with Least Square Fit, report the obtained parameters and plot of the fitted function.

Plotting result and parameters

1.2.5 Q7 statement

Now fit your distribution using maximum likelihood method. You can use any of the packages which has the method developed.

1.2.6 Q8 statement

Report the parameters of the fit and plot them next to distribution.

1.2.7 Q9 statement

Compare the power law fit with the exponential fit (using the same package). Report the log likelihood ratio R and the p -value. What do these numbers mean?

1.2.8 Q10 statement

What is the mathematical formula for scale free distribution you generated? Calculate the mean and the standard deviation of function? What would be the mean and standard deviation if the exponent would be 2.5?