Daniel Anzola Delgado
U69227703
Network Science
Assignment 3

The Majority Illusion

Introduction

The purpose of this assignment is to measure the prevalence of the Majority illusion in different networks. To measure the majority illusion eight networks have been selected. These include the US_airport and IMDB_actor data sets. Also, their Forrest-Fire variations and ER (due to higher clustering coefficients). According to the rank in the class a certain range has been chosen by performing a mathematical calculation. For this assignment the range used is 60-90. Within this range a certain number of nodes will be activated which will help to measure the prevalence of the majority illusion.

Methodology

For this assignment the programming language utilized is Python. The graph manipulation library used is NetworkX (v2.2). The code for this assignment consists in two major functions: activate_nodes and measure_MI,

- *activate_nodes:* this function takes as input a file address, a probability p, start and end ranges, and an output file name. Within this function the file that contains the edges is read. Depending on the type of file the functions read_edgelist and read_weighted_edgelist are used. Then, it proceeds to set all node attributes to zero. A dictionary was implemented containing the node and its value, which in this case refers to the attribute of the node. For this assignment the probabilities were given. Thus, there is four possible probabilities p. Within these probabilities the algorithm begins by adjusting the range accordingly to the probability p. Then, it will "activate" the nodes within the range by changing its attribute value to one. From this dictionary we set the attributes to the graph and write the output file. Then the measure_MI function is called.
- *measure_MI:* to measure the majority illusion we first convert the graph to a dictionary. While we iterate through the dictionary, for all the keys (nodes) with value (attribute) zero, we count all the its neighbors and count how many of them have attribute one. If the number of total neighbors divided by two is less than the total number of active nodes then we increase the counter by one. This function provides the total number of nodes with more than 50% of its neighbors with activated nodes. As an extra measurement the assortativity for node attributes is computed.

Important Functions from NetworkX

- *attribute_assortativity_coefficient*: compute assortativity for node attributes. Assortativity measures the similarity of connections in the graph with respect to the given attribute.
- *read_weighted_edgelist:* read a graph as list of edges with numeric weights.
- *read_edgelist:* read a graph from a list of edges.
- *set_node_attributes*: sets node attributes from a given value or dictionary of values.
- *get_node_attributes*: get node attributes from graph.

Experimental Results

Network	P = 0.05	P = 0.1	P = 0.2	P = 0.3
Airport	350	414	449	492
FF-airport	260	297	333	371
FFx10-airport	5648	6070	6684	6940
FF/10-airport	23	28	37	46
IMDB	612	815	1283	2003
FF-IMDB	433	542	782	1170
FF/10-IMDB	239	259	325	385
ER-IMDB	0	0	5	244

Airport	Attribute Assortativity Coefficient
P = 0.05	−0.00510699149358
P = 0.1	0.0017015265193
P = 0.2	0.00705931142324
P = 0.3	0.0175537659396

FF-airport	Attribute Assortativity Coefficient
P = 0.05	−0.00252356484533
P = 0.1	0.00984726943261
P = 0.2	0.000135390230641
P = 0.3	0.0100413159291

FFx10 - airport	Attribute Assortativity Coefficient
P = 0.05	0.000276424804774
P = 0.1	−0.00583779169158
P = 0.2	0.000977706924435
P = 0.3	0.00226851374879

FF/10 - airport	Attribute Assortativity Coefficient
P = 0.05	0.00248205367279
P = 0.1	0.0105755315596
P = 0.2	−0.00727042723248
P = 0.3	−0.051326827598

IMDB	Attribute Assortativity Coefficient
P = 0.05	0.0201124038278
P = 0.1	0.0182106983433
P = 0.2	0.0248515830544
P = 0.3	0.0180704727483

FF-IMDB	Attribute Assortativity Coefficient
P = 0.05	0.000140186117961
P = 0.1	0.000147677975614
P = 0.2	0.000586201724263
P = 0.3	0.000695797079072

FF/10-IMDB	Attribute Assortativity Coefficient
P = 0.05	0.00196084334423
P = 0.1	0.0146151777261
P = 0.2	0.00738651125424
P = 0.3	0.00187770915647

ER-IMDB	Attribute Assortativity Coefficient
P = 0.05	0.00011144829632
P = 0.1	−0.00220511603008
P = 0.2	0.000896007664443
P = 0.3	0.000149088814297

Conclusions and Future Work

From the first table we can observe that the bigger the probability the more non-active nodes have at least 50% of its neighbors with attribute one. This phenomenon is seen in all the networks tested for this assignment. To get more information the attribute assortativity coefficient was measured. This indicates the similarity of connections in the graph with respect to the given attribute. In most networks we can observe the same pattern as in the first table.