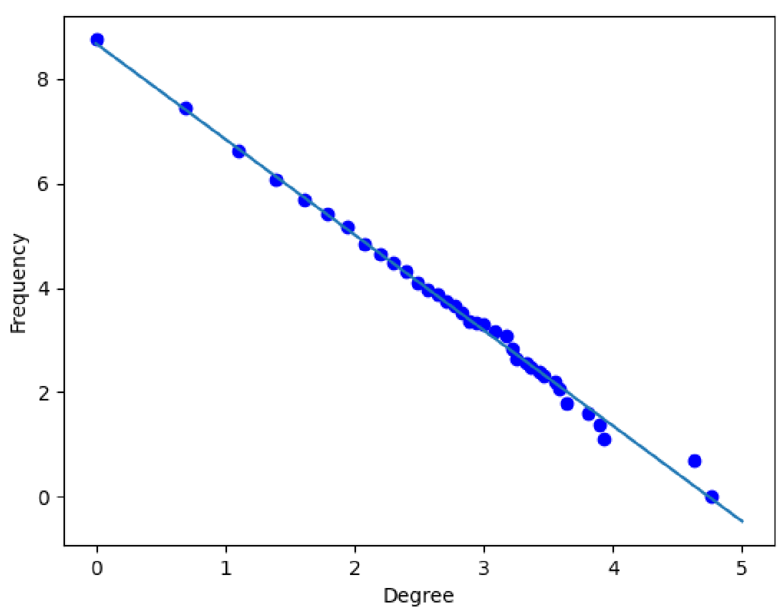
Homework 06

MO412 - Network Science

Margarita Lacuaña Apaza
RA: 234924

May 2021

Start with a network with just one node. Then simulate a Barabási-Albert process where at each time step you add one new node to the network ($m = 1$). Repeat until you reach $N = 10000$.Plot the degree distribution of the resulting network in log-log scale.Did you get a power law? Fit a straight line to the points and compute its slope. This should be a good approximation for the degree exponent in case the distribution follows a power law.



Code

```
import numpy as np
import networkx as nx
import math
from sklearn import linear_model
import matplotlib.pyplot as plt
import random

def probability(degrees):
        sum = 0
        for i in degrees:
                sum += i[1]
        probability = [ i[1]/sum for i in degrees]
        return probability
# ------------------- BarabasiModel ------------------- #

def barbasiModel(m=1,size=10):
        barabasiG = nx.Graph()
        barabasiG.add_node(0)
        barabasiG.add_edges_from([(0, 0)])
        for i in range(1,size):
                degrees = barabasiG.degree()
                probabilities = probability(degrees)
                barabasiG.add_node(i)
                for j in range(len(probabilities)):
                        r = random.uniform(0,1)
                        if(r<probabilities[j]):
                                barabasiG.add_edges_from
                                    ([(i,j)])
                                break
        return barabasiG

G = barbasiModel(size=10000)

degree_freq = nx.degree_histogram(G)
degrees = range(len(degree_freq))

dist_ac = [sum(degree_freq[k:]) for k in degrees]

dee_log = []
freq_log = []

pre_freq = dist_ac[0]
for k in degrees:
        f = dist_ac[k]
```

```
        if f != pre_freq:
            dee_log.append(math.log(k))
            freq_log.append(math.log(f))
            pre_freq = f

reg = linear_model.LinearRegression()
reg.fit(np.array(dee_log).reshape(-1,1),np.array(freq_log
    ))

rango = np.arange(0,6,1)
regression = reg.predict(rango.reshape(-1,1))

plt.plot(dee_log,freq_log,'bo')
plt.plot(rango,regression)

plt.xlabel('Degree')
plt.ylabel('Frequency')
plt.show()
```