Homework 08

MO412 - Network Science

Margarita Lacuaña Apaza
RA: 234924

May 2021

Two people are discussing accelerated growth, where $m(t) = t^{\theta}$links are added to the network with each new node. One person stands by the statement in the book, that accelerated growth leads to a scale-free network with exponent

$$\gamma = 3 + \frac{2\theta}{1-\theta}$$

The other person used the rate equation and reached a different formula:

$$\gamma = 1 + \frac{2}{1+\theta}$$

Your task is to test these hypotheses for $\theta = 1/2$.

Person 1 , $\theta = \frac{1}{2}$

$$\gamma = 3 + \frac{2\theta}{1-\theta}$$

$$\gamma = 5$$

Person 2, $\theta = \frac{1}{2}$

$$\gamma = 1 + \frac{2}{1+\theta}$$

$$\gamma = 2.33$$

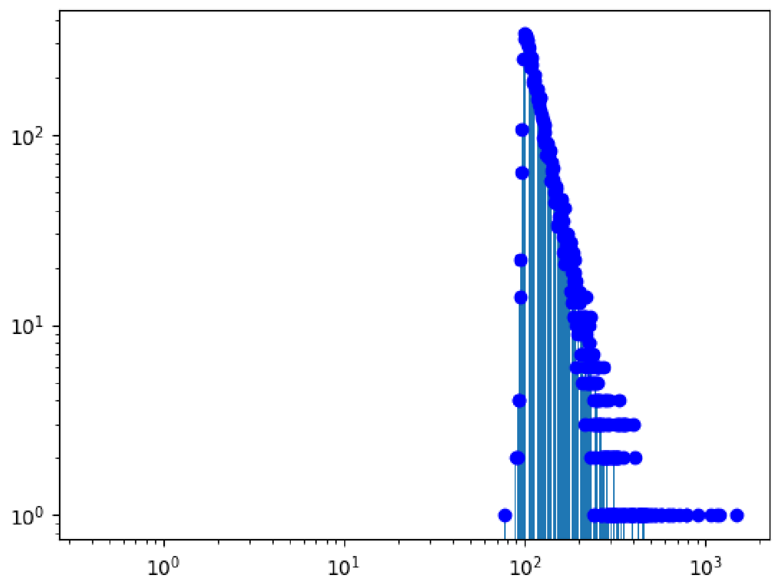
(a) **Start with a network with just one node at time $t = 0$. Then,at each time step $t \geq 1$, add one new node to the network with $m(t) = \lfloor\sqrt{t}\rfloor$ links to previous nodes. The other extreme of each link will be chosen by preferential attachment, that is, with a probability proportional to the degree of each node (not considering the new links). This may produce multiple edges. Let them be. Repeat until you reach $N = 10000$.**

```
def m_t(t):
  return math.floor(math.sqrt(t))

def network(nodes):
  G = nx.Graph()
  G.add_node(0)
  repeated_nodes = [0]
  t=0
  while t < nodes:
    m = m_t(t)
    choose = random.choices(rep,k=m)
    G.add_edges_from([(t,choose[i]) for i in range(m)
        ])
    rep += choose
    rep += [t] * m
    t += 1
  return G

G = network(10000)
```

(b) **Plot the degree distribution of the resulting network in log-log scale.**

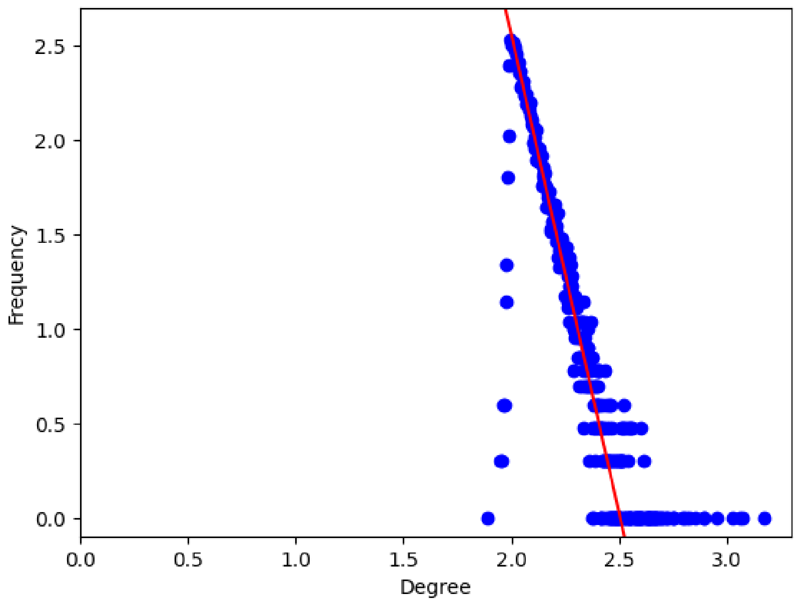



```
degree_freq = nx.degree_histogram(G)
degrees = range(len(degree_freq))

plt.loglog(degrees, degree_freq, 'bo')
plt.bar(degrees, degree_freq)
plt.show()
```

(c) **Fit a straight line to the points as best you can and estimate the γ exponent. Did you get a value close to one of the proposed formulas?**

The value of γ is 5.05921063, it is closer to what the first person said.




```
dee_log = []
freq_log = []

for i in range(len(degrees)):
  if(degrees[i] != 0 and degree_freq[i] != 0):
    dee_log.append(math.log10(degrees[i]))
    freq_log.append(math.log10(degree_freq[i]))

reg = linear_model.LinearRegression()
reg.fit(np.array(dee_log[10:220]).reshape(-1,1),np.
    array(freq_log[10:220]))
print("gamma :",reg.coef_)

rango = np.arange(1,3,0.1)
regression = reg.predict(rango.reshape(-1,1))
```

```
plt.xlim(0,3.3)
plt.ylim(-0.1,2.7)

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

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