# Homework 01 - Network Science

Gabriel Luciano Gomes (265673)

October 1, 2020

## Exercise 1

In exercise 1, was asked to find a value of p satisfing the following conditions:

- N = 10000 - Have 10000 nodes
- k = 4 - Average degree close to 4

The value o p found was $p = 0.000398$, but any value close to $p = 0.0004$ satisfies the condition of k. To build the Graph above, the following code was used:

```
import random
from networkx import nx

def questionOne():
    N = 10000 # number of nodes
    p = 0.000398 #probability of linking

    # building the Graph
    G = nx.Graph()
    for i in range(N):
        G.add_node(i)

    for i in range(N):
        for j in range(i+1,N):
            if random.random() < p:
                G.add_edge(i, j)
```

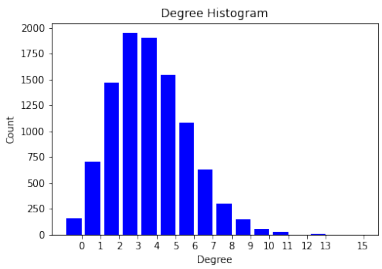


Figure 1: Degree distribution in normal scale

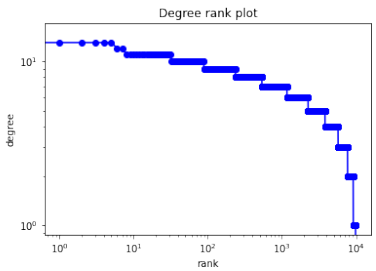


Figure 2: Degree distribution in log-log scale

## Exercise 2

For this exercise, the following condition was given:

For every $j > i$, add a link $(i, j)$ for node $i < j$ with a custom probability

$$p = \frac{k_{ij} + \epsilon}{\sum_{m=1}^{j-1}(k_{mj} + \epsilon)} \times q$$

where $k_{ij}$ is the degree of i at the beginning of iteration $j$, $\epsilon = 0.00001$, and $q = 4/3$.

To build the Graph above, the following code was used:

```
import random
from networkx import nx

def questionTwo():
    N = 10000 # number of nodes
    e = 0.00001 # epsilon error

    # building the Graph
    G = nx.Graph()
    for i in range(N):
        G.add_node(i)

    for j in range(N):
        summatory = 0
        for m in range(1, j):
            summatory += G.degree[m] + e

        for i in range(1, j):
            if (j > 1):
                customProbability = valueOfP(G.degree[i], summatory)
                if (random.random() < customProbability):
                    G.add_edge(i, j)

def valueOfP(kij, summatory):
    """ Calculates the custom probabilty (p) for exercise #2.
    -Parameters:
    kij: Degree of node i at moment j
    summatory: sum of degress at moment j
    """
    e = 0.00001 #epsilon error
    q = 4/3 #rational coefficient

    p = ((kij + e)/summatory) * q
    return p
```

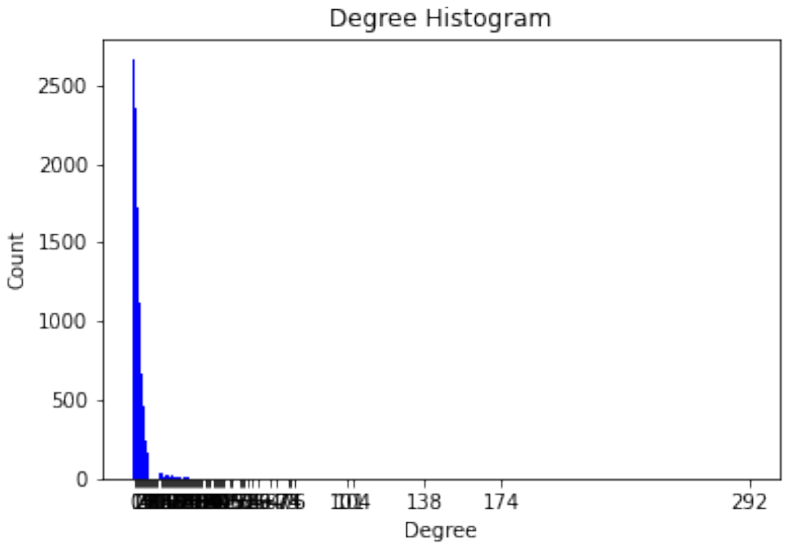


Figure 3: Degree distribution in normal scale

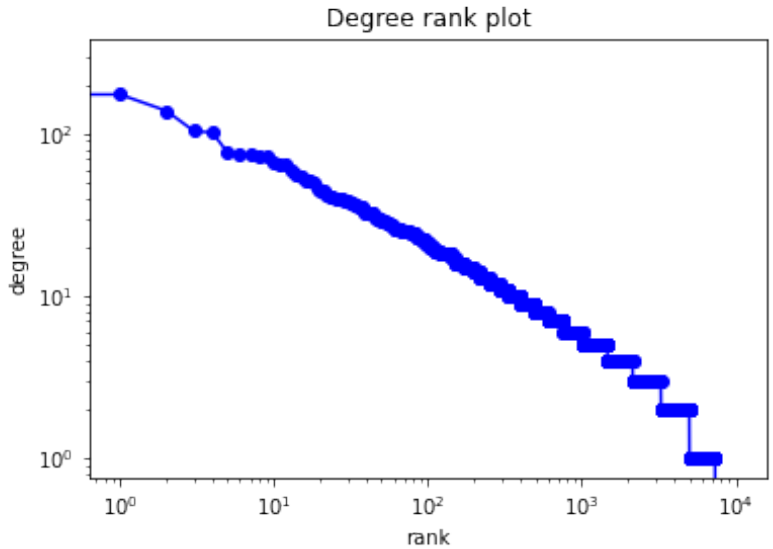


Figure 4: Degree distribution in log-log scale