Complex Networks: Quiz #12

Due on Jan 27th, 2019

RUOPENG XU
18M38179

Problem 1

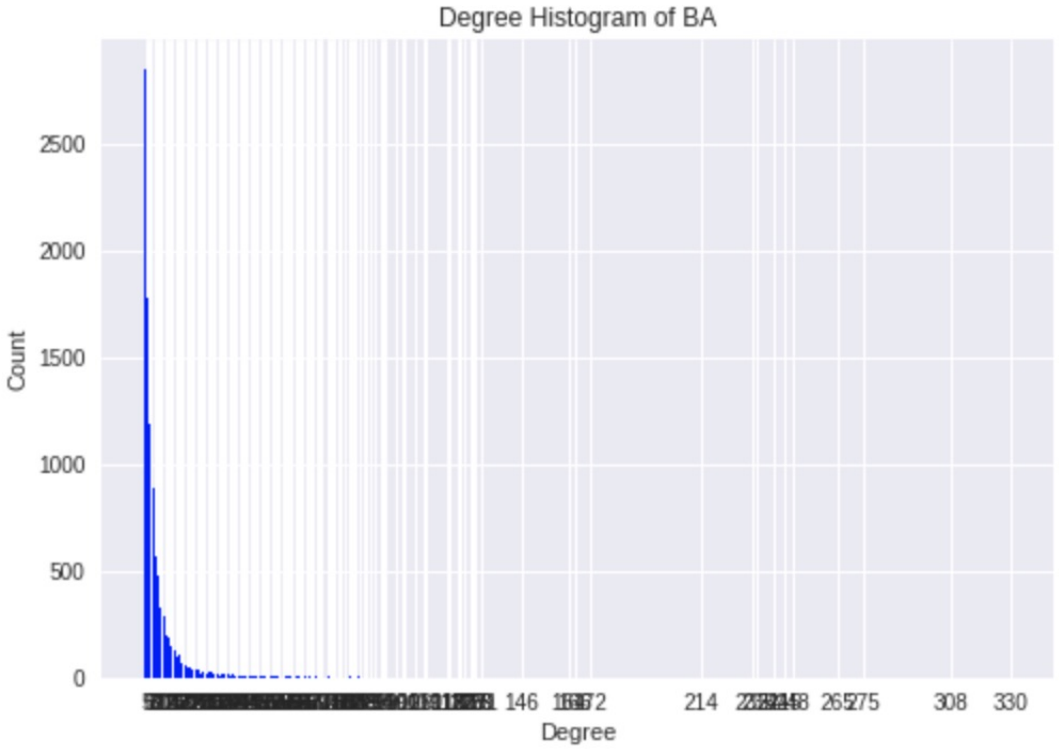
1.Plot the degree distributions of (a) Erdos-Renyi graph and (b) Barabasi-Albert graph of 10,000 nodes and about 50,000 edges.

Answer 1

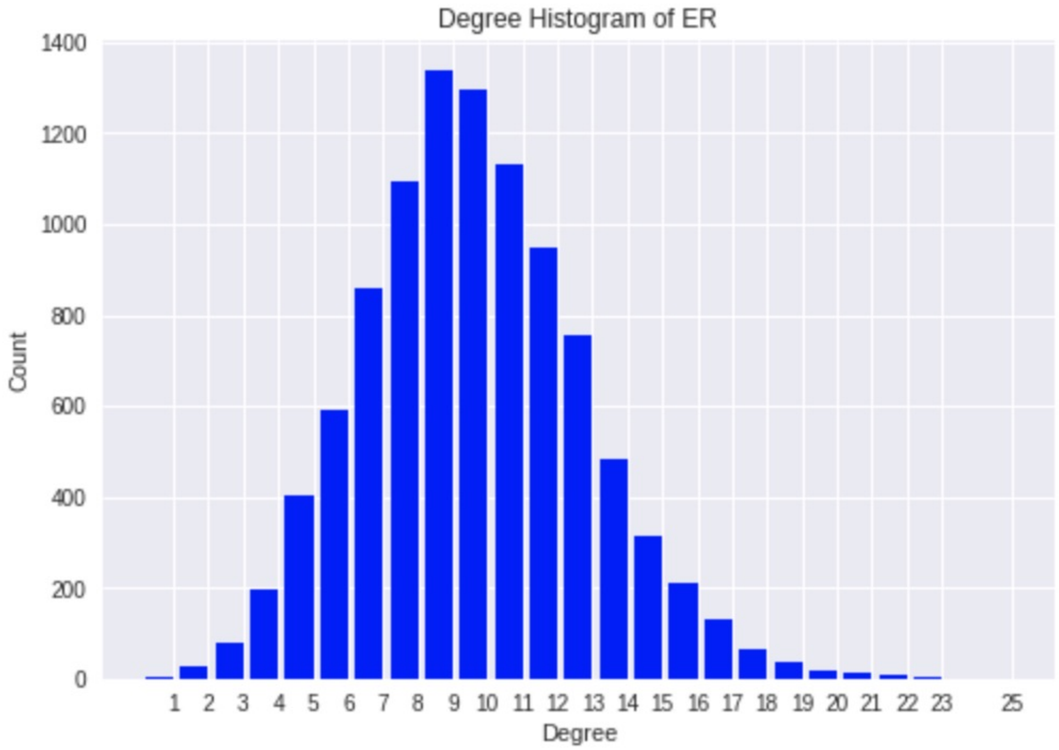
```
import collections
import matplotlib.pyplot as plt
import networkx as nx

ba = nx.barabasi_albert_graph(10000, 5) #only this line is changed
print(nx.info(ba))

degree_sequence = sorted([d for n, d in ba.degree()], reverse=True)  # degree sequence
# print("Degree sequence", degree_sequence)
degreeCount = collections.Counter(degree_sequence)
deg, cnt = zip(*degreeCount.items())

fig, ax = plt.subplots()
plt.bar(deg, cnt, width=0.80, color='b')

plt.title("Degree Histogram of BA")
plt.ylabel("Count")
plt.xlabel("Degree")
ax.set_xticks([d + 0.4 for d in deg])
ax.set_xticklabels(deg)

plt.show()



er = nx.erdos_renyi_graph(10000, 0.001)#only this line is changed
print(nx.info(er))

degree_sequence = sorted([d for n, d in er.degree()], reverse=True)  # degree sequence
# print("Degree sequence", degree_sequence)
degreeCount = collections.Counter(degree_sequence)
deg, cnt = zip(*degreeCount.items())

fig, ax = plt.subplots()
plt.bar(deg, cnt, width=0.80, color='b')

plt.title("Degree Histogram of ER")
plt.ylabel("Count")
plt.xlabel("Degree")
ax.set_xticks([d + 0.4 for d in deg])
ax.set_xticklabels(deg)


plt.show()
```

The result is:

```
Name:
Type: Graph
Number of nodes: 10000
Number of edges: 49975
Average degree:   9.9950
```



```
Name:
Type: Graph
Number of nodes: 10000
Number of edges: 49931
Average degree:   9.9862
```



Problem 2

2. Show the following metrics of the above both graphs.
- Number of nodes
- Number of edges
- Average degree
- Number of connected components
- Number of triangles
- Transitivity (clustering coefficient)
- Maximum degree
- Minimum degree

Answer 2

```
import networkx as nx
#import numpy as np
import matplotlib.pyplot as plt
from networkx.utils.random_sequence import powerlaw_sequence

er = nx.erdos_renyi_graph(10000, 0.001)
print("Erdos-Renyi graph")
print(nx.info(er))
print("connected components:", nx.number_connected_components(er))
sum1 = 0
for i in range(0,10000):
  sum1 = sum1 + nx.triangles(er, i)
print("numer of triangles:",int(sum1/3))
print("transitivity:",nx.transitivity(er))
er_degree_sequence = sorted([d for n, d in er.degree()], reverse=True)
print("max degree:",max(er_degree_sequence))
print("min degree:",min(er_degree_sequence))

print()
ba = nx.barabasi_albert_graph(10000, 5)
print("Barabasi-Albert graph")
print(nx.info(ba))
print("connected components:", nx.number_connected_components(ba))
sum2 = 0
for i in range(0,10000):
  sum2 = sum2 + nx.triangles(ba, i)
print("numer of triangles:",int(sum2/3))
print("transitivity:",nx.transitivity(ba))
ba_degree_sequence = sorted([d for n, d in ba.degree()], reverse=True)
print("max degree:",max(ba_degree_sequence))
print("min degree:",min(ba_degree_sequence))
```

The result is :

```
Erdos-Renyi graph
Name:
Type: Graph
Number of nodes: 10000
Number of edges: 50170
Average degree:  10.0340
connected components: 1
numer of triangles: 167
transitivity: 0.0009979761601223865
max degree: 23
```

```
min degree: 1

Barabasi-Albert graph
Name:
Type: Graph
Number of nodes: 10000
Number of edges: 49975
Average degree:   9.9950
connected components: 1
numer of triangles: 2451
transitivity: 0.005371160091688837
max degree: 439
min degree: 5
```