# Assignment 07-08: Network Models

## Performance summary

✓ Assessed

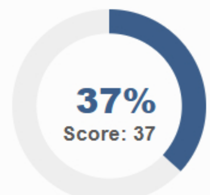
| Success status | Score | Attempts |
|---|---|---|
| Undefined | 37 of 100 points | 1 |

### Results

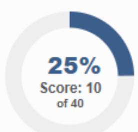
| | |
|---|---|
| Course | **Network Theory and Dynamic Systems (SS 2025)**<br>ID: 4669112833 / 109824322765731 |
| Test | **Assignment 06: Network Models**<br>ID: 4585128403 |

## This are your test results

| | |
|---|---|
| **Duration** | 0h 36m 26s 6/5/2025, 10:24 AM - 6/5/2025, 11:01 AM |
| **Answered** | 13 of 13 questions (100%) |
| **Your score** | 37 of 100 points (37%) |

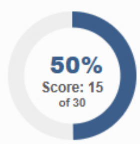



01. Knowledge Tasks (40 points) 5



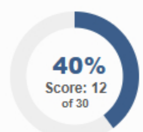

go to section ›

02. Practical Tasks (30 points) 3




go to section ›

03. Programming Tasks (30 points) 5




go to section ›

## 01. Knowledge Tasks (40 points) 10 of 40 points (25%)

### Attractiveness and Fitness Models (10 points)

| | | |
|---|---|---|
| **Status** | Answered | |
| **Your score** | 0 / 10 | 0% |

#### Response

The attractiveness and fitness models are based on the same idea that nodes have an intrinsic appeal, unrelated to their degree. What are the differences between the two models?

```
The fitness and attractiveness models have in common that nodes possess some motivational intrinsic trait which modifies
their potential to receive links, irrespective of their current degree. They differ, however, in how such an intrinsic
quality influences network evolution and topology.

In this model, a node's attractiveness is fixed when it is created, meaning that each node contains fixed attractiveness
parameter A that is in addition to its degree as a proxy for receiving links. This allows the attachment probability to
become proportional to (k + A) where k is the node's degree. This model mitigates the rich-get-richer phenomenon by
rendering even new or low-degree nodes a positive baseline chance of link accrual due to their innate appeal. The outcome
is a degree distribution that is more nuanced than a power law - it is still possible to follow a power law but the
exponent is weaker thanks to smaller values of, A.

A fitness model, on the other hand, gives each node a fitness score - frequently referred to as η - that expresses how
efficiently a node can compete for links over time. The likelihood of a node acquiring a new link is influenced by both
```

the node's degree and its fitness: ηk. Standing in contrast to the attractiveness model, nodes with greater fitness are capable of outcompeting less fit or more connected nodes resulting in competition of a dynamic nature. This may produce "winner take all" states as well as greater variance in the distributions of the degree.

232 words

▸ Solution

### BA Models (10 points)

| | | | |
|---|---|---|---|
| **Status** | Answered | | |
| **Your score** | 0 / 10 | | 0% |

## Response

Give a reason why networks generated by the BA model do not have many triangles.

Due to the dense nature of the network, triangles generated by the Barabási-Albert (BA) model are not particularly abundant. This happens as new links are created. In the BA model, new nodes attach to a fixed number of existing nodes with a probability proportional to their degree (preferential attachment). Connections to high-degree nodes, or hubs, are made, but connections among the neighbors of those nodes are neglected.

Local clustering is achieved through the addition of new nodes. Since new nodes connect to hubs regionally situated far away from the borders, the hubs form with little to no clustering, yielding low local clustering across the entire network. Triangles in a network often form when a new node connects not only to a hub, but also to that hub's neighbors, thereby completing closed loops. A characteristic feature of the BA model is no mechanism that enables such local closure. The outcome is a tree-like, hierarchical architecture in which hubs possess abundant low degree neighbors who are not linked to each other. This structural configuration drastically reduces the network's capability to form triangles.

Consequently, the clustering coefficient in BA networks is comparatively sparse and decreases with growth, which is the opposite of the case in most real-world networks that exhibit a high degree of clustering. More sophisticated models such as the Holme - Kim model were later designed to overtly promote triangle creation to improve this shortcoming.

229 words

▸ Solution

### Random Models (5 points)

| | | | |
|---|---|---|---|
| **Status** | Answered | | |
| **Your score** | 5 / 5 | | 100% |

## Response

What is a reason why random networks are not good models of social networks?

- (✓) Random networks have low clustering coefficients
- ( ) Random networks are typically not connected
- ( ) Random networks have small average shortest-path lengths
- ( ) Nodes in random networks have very different degrees

▸ Solution

### The Watts–Strogatz Model (5 points)

| | | | |
|---|---|---|---|
| **Status** | Answered | | |
| **Your score** | 5 / 5 | | 100% |

## Response

The Watts–Strogatz model is useful in capturing which property of real-world social networks not found in Erdős– Rényi random graphs?

- (✓) High clustering coefficient
- ( ) Long average path lengths
- ( ) Low clustering coefficient
- ( ) Short average path lengths

▸ Solution

### Rank Model (10 points)

| | |
|---|---|
| **Status** | Answered |

| | | | |
|---|---|---|---|
| **Your score** | 0 / 10 | | 0% |

### Response

Consider two versions of the rank model, with different ranking criteria. In the first version nodes are ranked by age (the time elapsed since they were added to the network). In the second version nodes are ranked by their degree. Is there a difference between networks generated by the two models, and if so, what is it?

```
There is indeed a remarkable discrepancy between the networks produced by the two versions of the rank model, one of
which ranks nodes by age while the other ranks nodes by degree.

In the age-based rank model, older nodes persistently outperform younger nodes because they tend to be of higher rank due
to simply having been in the network longer. This causes the “first-mover advantage” for the early nodes, which is that
they inevitably get more connections over time, irrespective of their current connectivity. The berserk exponential or
sub-linear preferential attachment networks are borne out from this, which degree heterogeneous networks are exhibiting,
but the distribution of degrees remains dominantly non heavy-tailed as opposed to degree ranking driven models, even when
based on age.

Unlike the previous example, in the degree-based rank model, nodes are ranked as well as connected to based on the number
of their current connections. This creates a self-reinforcing feedback loop because such nodes who have a lot of
connections are further gaining more links. Thus the network is allowed to evolve under preferential attachment,
resulting in a nurture bred scale free distribution of degrees. Degree hubs do emerge in these networks, but are far more
pronounced in non contradictory settings.

Therefore, whichever way you look at it, networks aligned with rich get richer dynamics are generated by both models.
However, with the degree-based model yielding skewed, hub dense networks and the age based distinguishing them with
favouring connectivity over longevity leading to less extreme hierarchies.
```

**253 words**

▸ Solution

< go back to overview

## 02. Practical Tasks (30 points) 15 of 30 points (50%)

#### Tutorial

| | | | |
|---|---|---|---|
| **Status** | Answered | | |
| **Your score** | 0 / 0 | | 0% |

### Response

Go through the Chapter 5 Tutorial on the book's GitHub repository.

**0 word**

▸ Solution

#### Random Network and AVG Node Degree (15 points)

| | | | |
|---|---|---|---|
| **Status** | Answered | | |
| **Your score** | 0 / 15 | | 0% |

### Response

Suppose you are making a random network with 50 nodes, and you want the average node degree to be 10. What approximate value for p would you use in this case?

```
Setting the average node degree to 10 and total number of nodes to 50 means having to create a random network with those
specifications. The 'Erdős–Rényi random network' as well as the 'Erdős–Rényi random graph model' both rely on the average
degree depending linearly on p (probability) as well as the number of connections each node has. Each node's theoretical
degree under both these models is approximately 10 which makes it feasible for the average degree of the graph. Following
these formulas leads to the conclusion that each node has p set to around 0.204% which is similar to what standard
percentage probability suggests.
Around 20% chance connectivity chances between any two nodes would create complete 10 connections across each node in 50
node connections.

Conclusively, an average degree of 10 is achievable using 0.204% probability while also keeping the random network
intact.
```

**142 words**

▸ Solution

#### Link Probabiity (15 points)

| | | | |
|---|---|---|---|
| **Status** | Answered | | |
| **Your score** | 15 / 15 | | 100% |

Response

Suppose you want to construct a random graph with 1000 nodes and about 3000 links. Give a value of the link probability p that could lead to this outcome.

0.006

▸ Solution

< go back to overview

## 03. Programming Tasks (30 points) 12 of 30 points (40%)

The socfb-Northwestern25 network in the book's GitHub repository is a snapshot of Northwestern University's Facebook network. The nodes are anonymous users and the links are friend relationships. Load this network into a NetworkX graph; be sure to use the proper graph class for an undirected, unweighted network. Once you measure the number of nodes and links, use **nx.gnm_random_graph()** to create a separate random network with the same number of nodes and links as the Facebook graph. Use this random network to answer the following questions.

### Q1. Degree (6 points)

| | | |
|---|---|---|
| **Status** | Answered | |
| **Your score** | 0 / 6 | 0% |

Response

What is the 95th percentile for degree in the random network (i.e.the value such that 95% of nodes have this degree or less)? Write code to compute it.

```
import networkx as nx
import numpy as np

G = nx.read_edgelist(r"socfb-Northwestern25.edgelist", nodetype=int, create_using=nx.Graph())

G_random = nx.gnm_random_graph(G.number_of_nodes(), G.number_of_edges())

degrees = [d for n, d in G_random.degree()]
percentile_95 = np.percentile(degrees, 95)

print(f"95th percentile degree in the random network: {percentile_95}")
```

**42 words**

▸ Solution

### Q2. Random Network Properties (6 points)

| | | |
|---|---|---|
| **Status** | Answered | |
| **Your score** | 6 / 6 | 100% |

Response

We are dealing with a random network, so some properties are going to differ somewhat each time one is generated. True or false:

| Unanswered | Right | Wrong | |
|---|---|---|---|
| ☐ | ☑ | ☐ | Given fixed parameters N and L, all random networks created with gnm_random_graph() will have the same mean degree. |

▸ Solution

### Q3. Distribution Shape (6 points)

| | | |
|---|---|---|
| **Status** | Answered | |
| **Your score** | 6 / 6 | 100% |

Response

Which of the following shapes best describes the degree distribution in this random network?

- ○ Left-tailed: most node degrees are relatively large compared to the range of degrees
- ○ Uniform: node degrees are evenly distributed between the minimum and maximum
- ○ Right-tailed: most node degrees are relatively small compared to the range of degrees
- ⊙ Normal: most node degrees are near the mean, dropping off rapidly in both directions

▸ Solution

### Q4. Average Shortest Path Length (6 points)

| | |
|---|---|
| **Status** | Answered |

| | | | |
|---|---|---|---|
| **Your score** | 0 / 6 | | 0% |

### Response

Estimate the average shortest-path length in this random network using a random sample of 1000 pairs of nodes.

```
import networkx as nx
import random

G = nx.read_edgelist(r"D:\Koblenz\Sem1\NTDS\Assignments\Assignment 07-08\socfb-Northwestern25.edgelist", nodetype=int,
create_using=nx.Graph())

if not nx.is_connected(G):
    largest_cc = max(nx.connected_components(G), key=len)
    G = G.subgraph(largest_cc).copy()


nodes = list(G.nodes())
sampled_pairs = random.sample([(u, v) for u in nodes for v in nodes if u < v], 1000)
path_lengths = []

for u, v in sampled_pairs:
    try:
        length = nx.shortest_path_length(G, source=u, target=v)
        path_lengths.append(length)
    except nx.NetworkXNoPath:
        continue

if path_lengths:
    avg_length = sum(path_lengths) / len(path_lengths)
    print(f"Estimated average shortest path length: {avg_length:.4f}")
else:
    print("No valid paths found in the sample.")
```

**91 words**

▸ Solution

### Q5. Average Clustering Coefficient (6 points)

| | | | |
|---|---|---|---|
| **Status** | Answered | | |
| **Your score** | 0 / 6 | | 0% |

### Response

What is the average clustering coefficient of this random network?

```
import networkx as nx
import numpy as np

G = nx.read_edgelist(r"socfb-Northwestern25.edgelist", nodetype=int, create_using=nx.Graph())

G_random = nx.gnm_random_graph(G.number_of_nodes(), G.number_of_edges())

avg_clustering = nx.average_clustering(G_random)
print(f"Average clustering coefficient of the random network: {avg_clustering:.4f}")
```

**37 words**

▸ Solution

‹ go back to overview

## Test execution

### Information

Availability: Expired at 6/9/2025, 11:59 PM

Max. attempts: Unlimited

Results of this test are visible to administrators and tutors of this course.

Start test

▸ Change log

^ Go to top