exercise_4

October 16, 2018

1 CS-E5740 Complex Networks

1.0.1 Adam Ilyas 725819

Percolation, error & attack tolerance, epidemic models

1.0.2 Percolation theory

"order parameter" P: fraction of nodes in the largest connected component (LCC)

Control parameters f: $\frac{\text{number of active links}}{\text{number of all possible links}}$

```
In [1]: import os

        import random

        import networkx as nx
        import numpy as np
        import matplotlib.pyplot as plt
        import scipy.stats

        from percolation_in_er_networks import *
        from error_and_attack_tolerance import *
```

2 1. Percolation in Erds-Rényi (ER) networks

Erds-Rényi networks are random networks where N nodes are randomly connected such that the probability that a pair of nodes is linked is p.

Sparse ER graph: - average degree $\langle k \rangle$ is some fixed small number - the size of the network N is very large. - N large, p large such that $p(N-1) = \langle k \rangle$ stays constant

Percolation properties of ER Graphs.

percolation threshold which is the value of $\langle k \rangle$ where the giant connected component appears

2.0.1 1a)

Using the idea of branching processes and assumption that large and sparse ER graphs are tree-like,

calculate the expected number of nodes at d steps away, n_d , from a randomly selected node in an ER network as a function of $\langle k \rangle$ and d. Using this result, justify that the giant component appears in large and sparse ER networks when $\langle k \rangle > 1$.

Hints: – Remember that the degree distribution of an ER network is a Poisson distribution when $N \to \infty$ such that $\langle k \rangle$ is constant.

Hence $\langle k \rangle = \frac{\langle k^2 \rangle}{\langle k \rangle} - 1$

$$n_d = \langle q \rangle \cdot n_{d-1}$$
$$= (\frac{\langle k^2 \rangle}{\langle k \rangle} - 1) \cdot n_{d-1}$$
$$= (\frac{\langle k \rangle^2 + \langle k \rangle}{\langle k \rangle} - 1 \cdot n_{d-1}$$
$$= (\langle k \rangle + 1 - 1) \cdot n_{d-1}$$
$$= \langle k \rangle \cdot n_{d-1}$$

$$n_{d-1} = \langle k \rangle \cdot n_{d-2}$$
$$n_{d-2} = \langle k \rangle \cdot n_{d-3}$$
$$\vdots$$

$$n_d = \langle k \rangle^d$$

We recurse d times

where $\langle k \rangle$ is average degree, and d is the number of steps

2.0.2   1b)

Verify your analytical calculations for n_d using numerical simulations. Calculate the n_d value for $d \in 0...15$, $\langle k \rangle \in \{0.5, 1, 2\}$, and starting from enough randomly selected nodes to get a good estimate for the expected value. Try out two network sizes: $N = 10^4$ and $N = 10^5$ to see how the size affects the calculations.

```
In [2]: a_dir = "./assets"
        if not os.path.isdir(a_dir):
            os.mkdir(a_dir)

        #Solution for b)-c):
        fig = ER_breadth_first_search(0.5, 10**4, 10000)
        fig.savefig('./assets/er_breadthfirst_05_10k.pdf')

        fig = ER_breadth_first_search(1, 10**4, 10000)
        fig.savefig('./assets/er_breadthfirst_1_10k.pdf')

        fig = ER_breadth_first_search(2, 10**4, 100, show_netsize=True, max_depth=15)
        fig.savefig('./assets/er_breadthfirst_2_10k.pdf')

        fig = ER_breadth_first_search(0.5, 10**5, 10000)
```
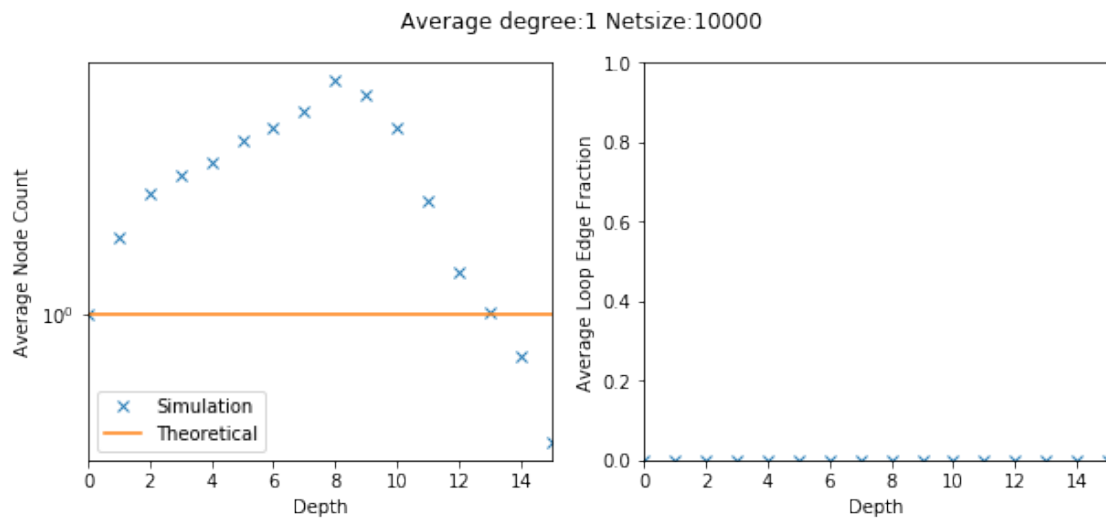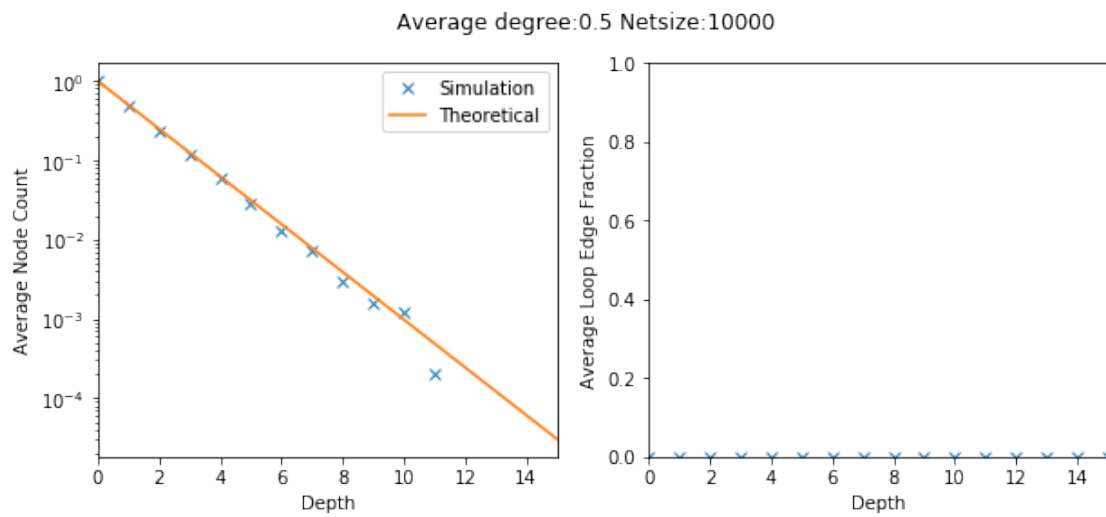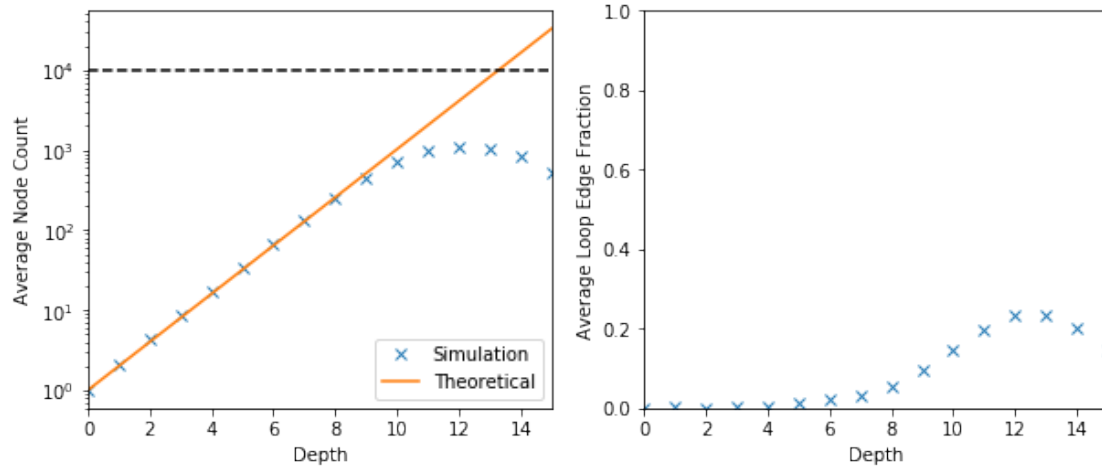
```
fig.savefig('./assets/er_breadthfirst_05_100k.pdf')

fig = ER_breadth_first_search(1, 10**5, 10000)
fig.savefig('./assets/er_breadthfirst_1_100k.pdf')

fig = ER_breadth_first_search(2, 10**5, 100, show_netsize=True, max_depth=15)
fig.savefig('./assets/er_breadthfirst_2_100k.pdf')
```
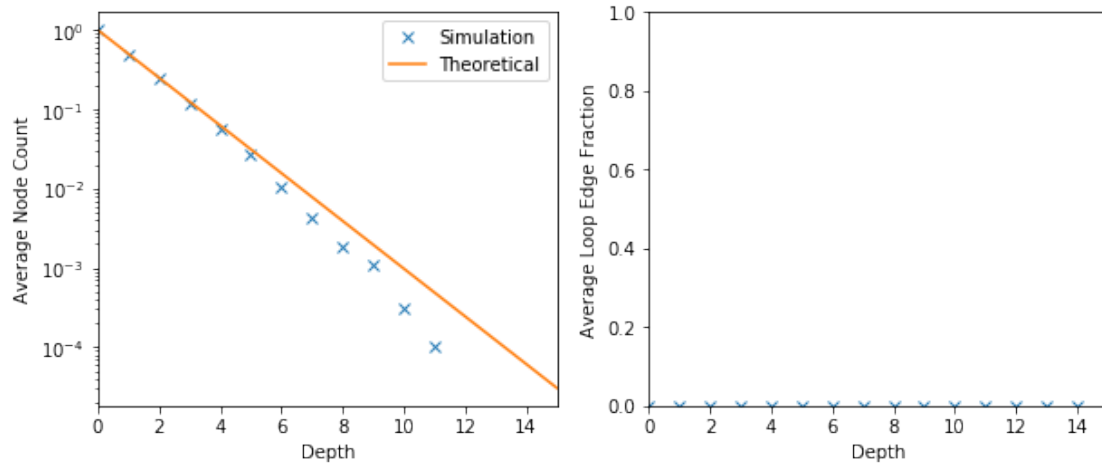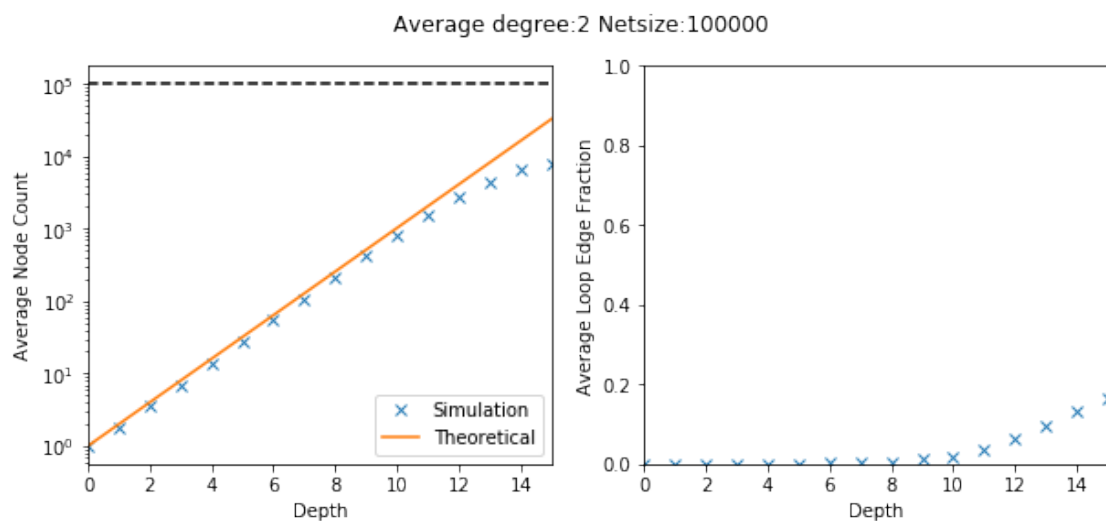
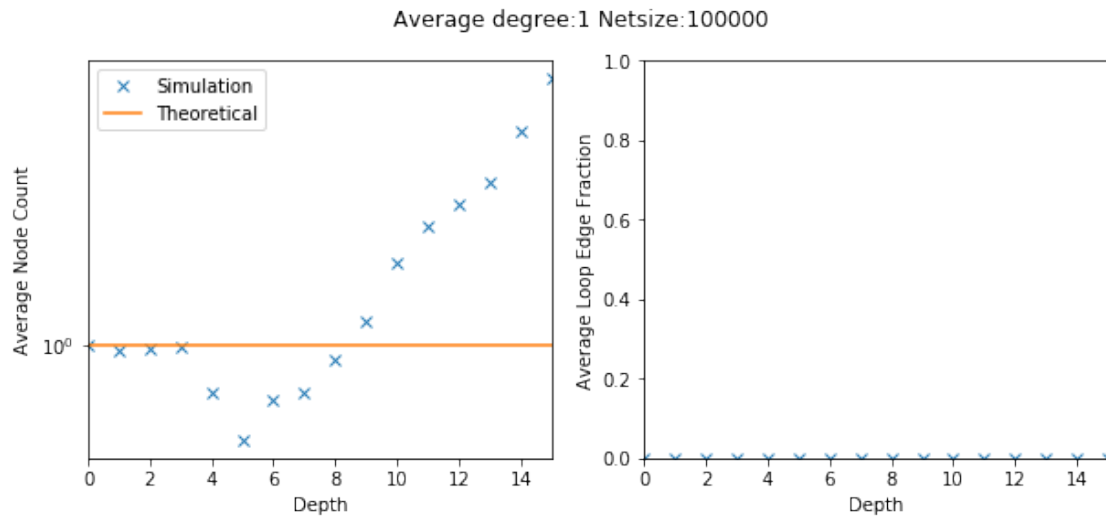## Average degree:2 Netsize:10000

Average degree:0.5 Netsize:100000

Average degree:1 Netsize:100000

Average degree:2 Netsize:100000

2.0.3 1c)

Explore the range at which the assumption of tree-likeness of the network is valid.

This can be done, for example, by calculating the number of edges that nodes at depth d have that go back to some earlier level in addition to the single edge that connects each node to the level $d1$, and

reporting the average fraction of such edges to all edges that go from depth d to earlier levels/depths.

In a perfect tree this fraction is exactly 0. Comment on the results, and their effect on our results to exercise b). What are the other things that make your analytical calculation of n_d to differ from your simulation results?

5

Ans:

For average_degree = 2, for both $N = 10^4$ and 10^5, we notice from the graph above that we will expect some loops as we go into more depths.

When loops occur, the average node count will not increase as fast as we will encounter the same nodes that we counted before and not count them

Thus, the simulated average node count will not increase as fast as in theory

2.0.4 1d)

Calculate the component sizes of simulated ER networks,

and use this data to (loosely) verify that the percolation threshold of ER networks is at the average degree of $\langle k \rangle = 1$.

That is, for $\langle k \rangle < 1$ the largest connected component is small (size being measured as number of participating nodes), and for $\langle k \rangle > 1$ it quickly reaches the network size. Do this by generating ER networks of size $N = 10^4$ with different average degrees: $\langle k \rangle = [0.00, 0.05, \cdots, 2.45, 2.50]$. For each of the ER networks, compute the size of the largest component and plot it against $\langle k \rangle$

```
In [3]: ER_percolation(10**4, 2.5, 0.05)
```

Out[3]:

6

Number of nodes = 10000
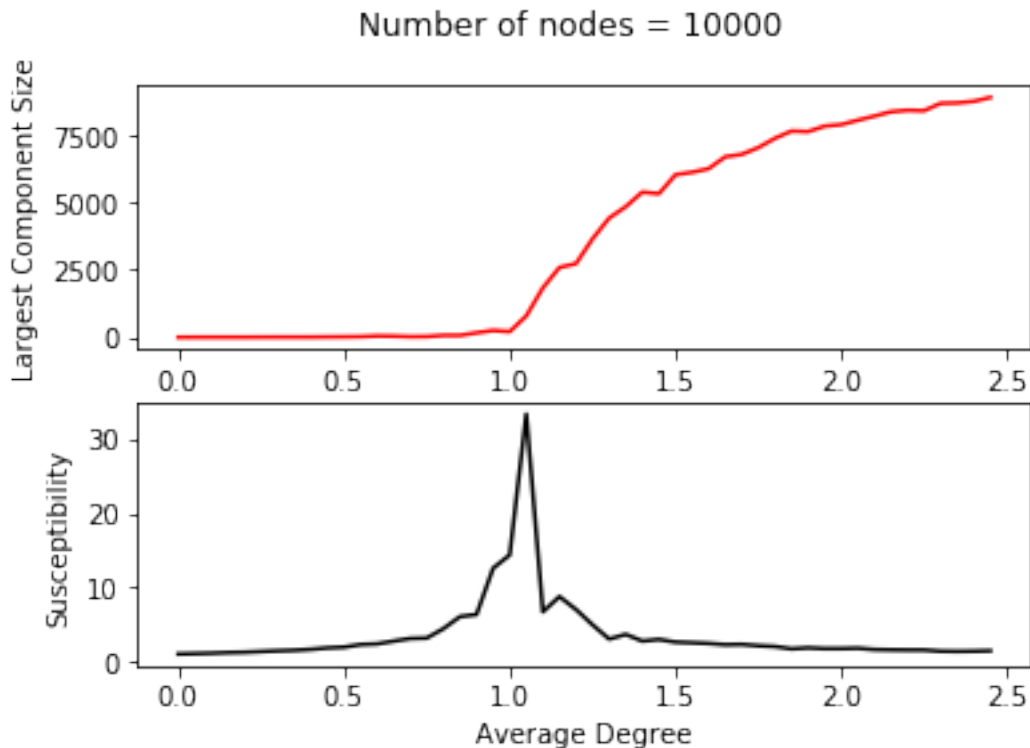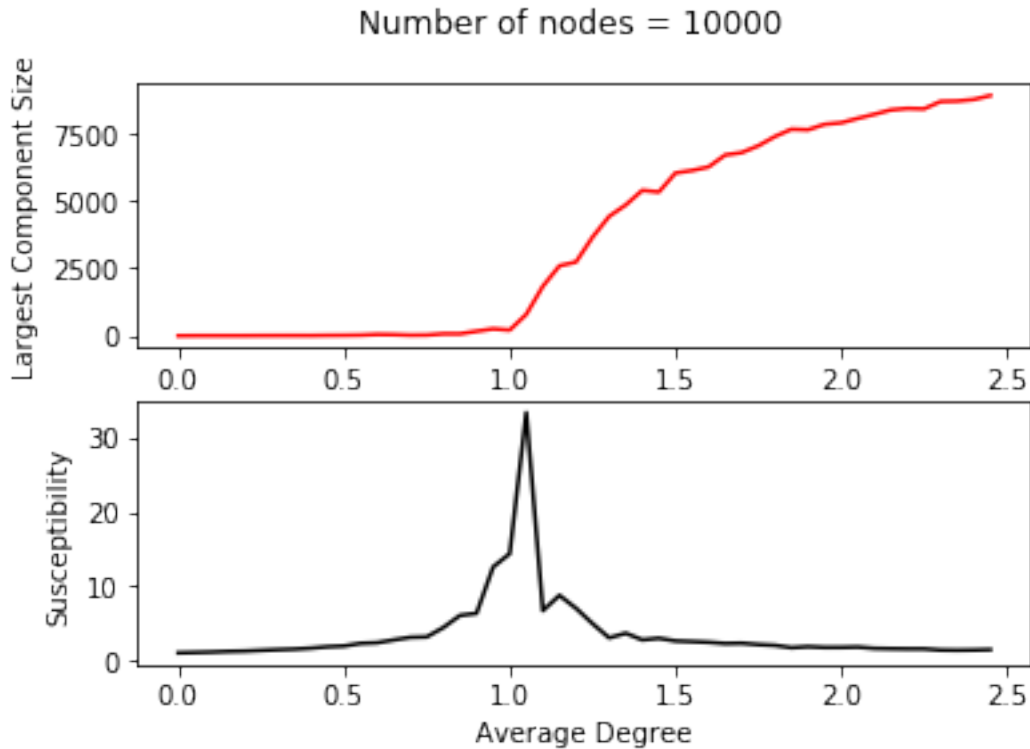
2.0.5 1e)

Another, a more elegant, way to find out when the percolation transition happens is:

to try to find the point at which the possibility for the largest component size growth is the largest when the control parameter (here $\langle k \rangle$ or p) is changed very little.

Think about the situation where $\langle k \rangle$ is changed so slightly that a single link is added between the largest component and a randomly selected node that is not in the largest component.

The expected change in the largest component size in this situation is some times called susceptibility, and it should get very large values at the percolation transition point. The susceptibility depends on the size distribution of all the other components, and it can be calculated with the following formula:

$$x = \frac{\sum_s s^2 C(s) - s_{max}^2}{\sum_s s C(s) - s_{max}^2}$$

where $C(s)$ is the number of components with s nodes. Calculate the susceptibility x for each network generated in exercise d), and again plot x as a function of $\langle k \rangle$. Explain the shape of the curve, and its implications.

Susceptibility is large at the point when the percolation transition happens

Where making a minor change to the average degree k lead to big changes in the size of largest component (LCC) in the network

Observation

We see in the curve when $k \approx 1$, we see a sudden spike in both graph (by definition, both are dependant to another). As such, percolation transition happens when average degree $k \approx 1$

7