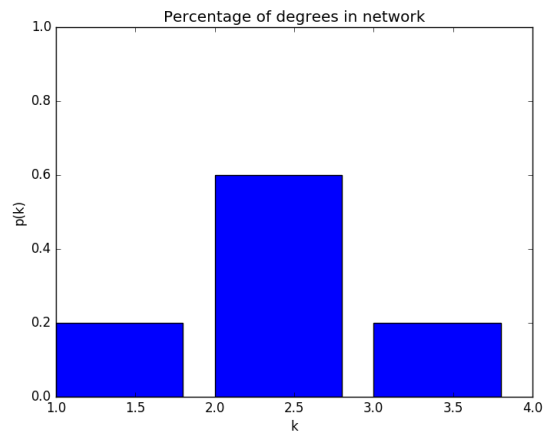


Graphs

Steven Raaijmakers

March 2017

1. (a) Degree histogram:



- (b) Adjacency matrix:

```
0 1 0 0 0
1 0 1 0 0
0 1 0 1 1
0 0 1 0 1
0 0 1 1 0
```

- (c)
- Betweenness centrality: 1: 0, 2: 3, 3: 4, 4: 0, 5: 0. In the picture you can see 1 doesn't lie on any shortest path between two nodes. 2 is on the shortest path from (1,3), (1, 4), (1,5), so gets a bc of 3. 3 is on the shortest path from (1, 4), (1,5), (2, 4), (2, 5), which are four different paths, so node 3 gets a bc of 4. Node 4 and node 5 aren't on any shortest path, thus their bc will be 0.
 - Degree centrality: 1: 0.25, 2: 0.5, 3: 0.75, 4: 0.5, 5: 0.5. The degree centrality of a node is the fraction of the nodes it is connected to (no loops). In the graph there are 5 nodes (no loops) so for one node, it can be max connected to 4 other nodes. Node 1 is connected to 1 of 4, so has a degree centrality of $1/4 = 0.25$, 2 is connected to $2/4 = 0.5$, 3 is connected to $3/4 = 0.75$, 4 is connected to $2/4 = 0.5$ and node 5 is also connected to $2/4 = 0.5$

- (d) By multiplying the adjacency matrix (AM) with a vector of the same height filled with one's:

$$\begin{array}{ccccccc} 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 & 1 \times 1 = 3 & & \\ 0 & 0 & 1 & 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 1 & 0 & 1 & 2 \end{array}$$

Because multiplying the AM with a vector of only one's will return the sum in each row, which are the links a node has aka degree.

- (e) For example, when you take the edge (1,2) you can reach 2 from 1, but you can't reach 1 anymore. Now if you do choose (2, 1), you can not reach the other part of the graph.
- (f) 1: 0.0, 2: 0.0, 3: 0.33, 4: 1.0, 5: 1.0
- (g) The lcc is 1 for every vertex when there are two edges added. One between (1,3), making the lcc of 1: 1 and 2: 1. One between (2, 4) making the lcc of 3: 3/4, so one last edge has to be added for (1, 4).

2. (a)

$$P(deg(v) = k) = \binom{n-1}{k} p^k (1-p)^{(n-1)-k} \quad (1)$$

There are $\binom{n-1}{k}$ ways to choose d vertices from among n total, with a p^k probability that there's an edge between them, so $\binom{n-1}{k} p^k$ is the probability that a node has edges to k nodes. However, there must be no edges to the rest of the n - k nodes, which occurs with probability $(1-p)^{n-1-k}$.

(b)

- (c) The second sumation is the weighted average of all clustering coefficients for a certain degree. The first sumation is the weighted average for all the possible degree's over the second sumamtion.

(d)

$$(e) \langle k \rangle = 2L/N = \frac{N \cdot p(N-1)}{N} = p(N-1)$$

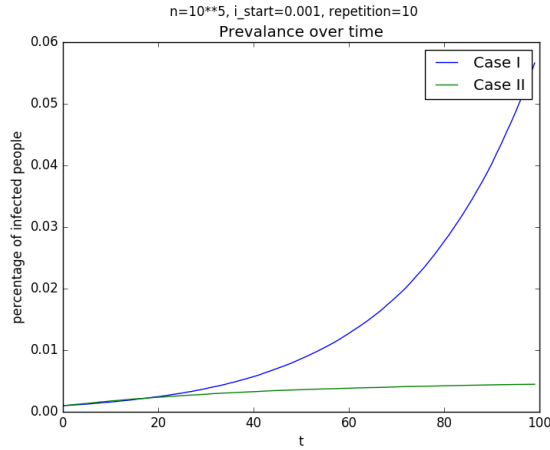
Note: All the plots will have a legend which refers to 2 cases (both start with 0.1% infected people):

- Case I: $N = 10^5, i = 0.01, \langle k \rangle = 5.0$
- Case II: $N = 10^5, i = 0.1, \langle k \rangle = 0.8$

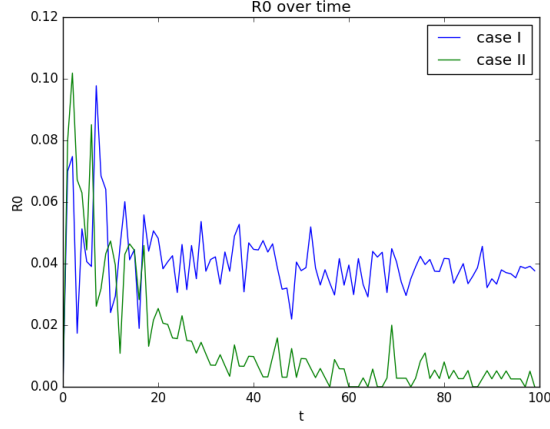
Epidemics

1. see *e1.py*

- (a) The probability of not getting infected while having r infected neighbor, equals $(1 - i)^r$. Therefore the probability of getting infected equals $1 - (1 - i)^r$
- (b) i_start is the infected part of N at the start of the experiment.



- (c) The second case initially grows faster because $i = 0.1$ which is 10 times higher than $i = 0.01$. In the second case, a node has a higher probability of spreading a disease, even though $\langle k \rangle$ is lower.
- (d) The probability that a node with degree k will infect one of its neighbors $1 - (1 - i)^k$
- (e) Case I: $1 - (1 - 0.01)^5 = 0.05$
Case II: $1 - (1 - 0.1)^{0.8} = 0.08$
Yes it does. If you look closely to the figure you can see case II is above case I in the early phase which is due to the infection probability being slightly higher, $0.08 > 0.05$
- (f) Each node has k outgoing edges to healthy neighbors. Each neighbor has a probability of i to get infected by this neighbor, therefore a node with degree k will infect approximately $k \cdot i$ neighbors.
Case I: $5 \cdot 0.01 = 0.05$
Case II: $0.8 \cdot 0.1 = 0.08$
This are the percentages of people getting sick from infected people. Since there are $0.001 \cdot 10^5 = 100$ people infected at the start we can calculate how many people are infected after 1 timestep:
Case I: $100 \cdot (1 + 0.05) = 105$
Case II: $100 \cdot (1 + 0.08) = 108$
- (g) $0.05/0.08 = 0.625$, Case II is $1/0.625 = 1.6$ times faster Case I
- (h) Yes it does. Taking a sample point of both lines at the same t and finding the ratio between them results in the same ratio which was previously found.

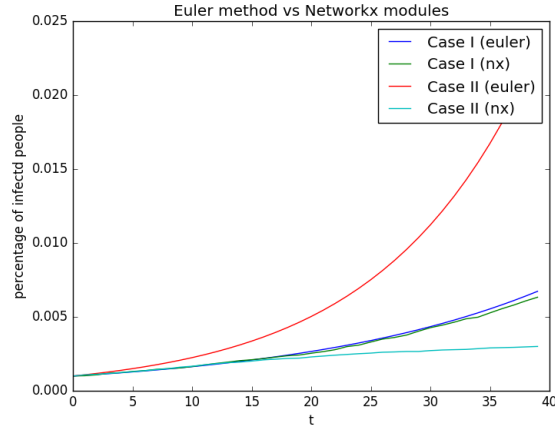


- (i) Eventually everyone will get sick so no one will infect people who weren't infected yet. Since more people will get infected quicker, there will be less people to infect after a certain point, so the curve will drop faster
 - (j) In the first case the outbreak size equals the amount of people there are. This means that eventually everyone gets infected which is due to $\langle k \rangle = 5$. Because of having a large k this means there is a giant component and thus everyone is reachable. In case II the $\langle k \rangle = 0.8$ and thus doesn't have a giant component. The final outbreak size will be much lower because not every node will be reachable
 - (k) Case I: $O(n)$ because with N people eventually N people will die.
Case II: $O(c)$
2. (a) The given ODEs describes the people who get sick, and the people who remain healthy for T . So instead of i (which was the chance of infecting your neighbor when you're sick) b is the chance of getting sick when being healthy, therefore $i \neq b$.
- (b)

$$\begin{aligned}
 (1 - (1 - b)^I) \cdot s &= (1 - (1 - i)^{\frac{\langle k \rangle}{n} \cdot I}) \cdot s \\
 1 - (1 - b)^I &= 1 - (1 - i)^{\frac{\langle k \rangle}{n} \cdot I} \\
 (1 - b)^I &= (1 - i)^{\frac{\langle k \rangle}{n} \cdot I} \\
 1 - b &= \sqrt[I]{(1 - i)^{\frac{\langle k \rangle}{n} \cdot I}} \\
 b &= 1 - \sqrt[I]{(1 - i)^{\frac{\langle k \rangle}{n} \cdot I}} \\
 &= 1 - (1 - i)^{\frac{\langle k \rangle}{n}}
 \end{aligned} \tag{2}$$

- (c) Case I: $B = 1 - (1 - 0.01)^{\frac{5.03}{10^5}} = 5 \cdot 10^{-7}$
Case II: $B = 1 - (1 - 0.1)^{\frac{0.8}{10^5}} = 8.43 \cdot 10^{-7}$

3. See *e3.py*



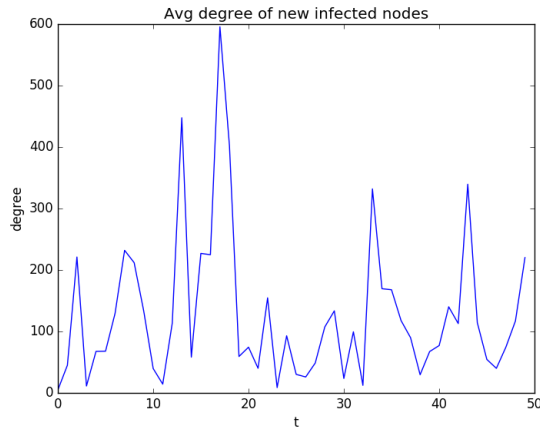
(a)

(b) The euler method for case I seems to approach the curve which was made by using the networkx modules. In case II however the lines seem to be very different, which can be due to the fact that the euler method does not take into account the network dynamics, such as the $\langle k \rangle$ being < 1 and therefore doesn't have a giant component.

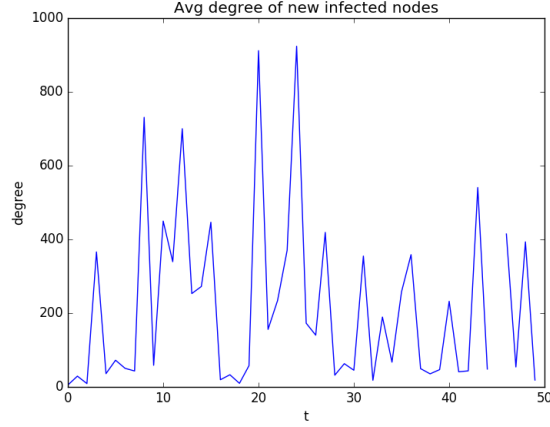
(c) Simplifying the formula results in $S - C^i \cdot s$ which should show us that both the lines have an exponential growth. However in the graph only the case I seems to be exponential, and case II seems to be linear. This weird behaviour can be justified by the absence of the giant component for case II.

4. see *e4.py*

(a) For 5 repetitions for both curves:



- (b) At the scale free case there are hubs (nodes with a very high degree), and once they get infected the disease will spread very quickly due to the fact that they have many neighbors. In the erdos-renyi there are no hubs and thus the infection will need more timesteps in comparison to the sf network.



- (c)
- (d) You can see the average degree at some timesteps is very high. At this timestep probably a hub was infected, which pushes up the average degree. Having hubs in a network is typical for scale free networks since new joining nodes are most likely to connect to nodes with a higher degree. Because of this some node's degree will keep growing, and once such a "popular" node gets infected, the disease starts spreading even quicker.