CS-E5740 Complex Networks, Course project

Alex Herrero Pons, Student number: 918697

December 21, 2020

Task 1: Basic implementation

Implement the SI model using the temporal air traffic data in `events_US_air_traffic_GMT.txt`. Use the provided visualization module to check that your implementation works reasonably. Assume first that $p = 1$, *i.e.*, the disease is always transmitted.

a) If Allentown (node-id=0) is infected at the beginning of the data set, at which time does Anchorage (ANC, node-id=41) become infected?

Solution.

Anchorage (ANC. node-id=41) is infected at time 1229290800.

Task 2: Effect of infection probability p on spreading speed

Run the SI model 10 times with each of the infection probabilities [0.01, 0.05, 0.1, 0.5, 1.0]. Again, let Allentown (node-id=0) be the initially infected node. Record all infection times of the nodes, and answer the following questions:

a) Plot the averaged prevalence $\rho(t)$ of the disease (fraction of infected nodes) as a function of time for each of the infection probabilities. Plot the 5 curves in one graph. You should be able to spot stepwise, nearly periodic plateaus in the curves.

Solution. See Figure 1.

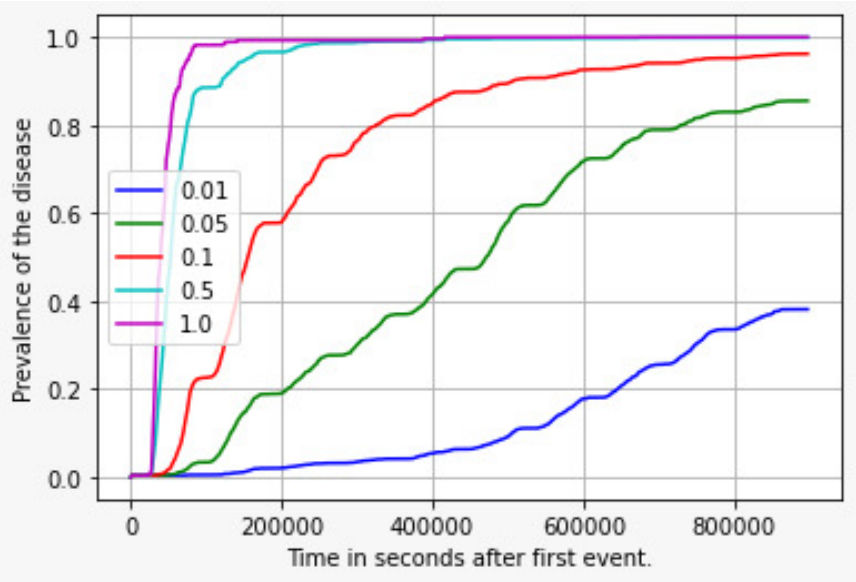

Figure 1: Averaged prevalence $\rho(t)$ of the disease as a function of time for each of the infection probabilities

b) For which infection probabilities does the whole network become fully infected? What are the periodic "steps" in the curves due to?

Solution.

The network becomes fully infected when $p = 0.5$ and when $p = 1.0$.

Probably the periodic steps in the curves are due to the infection of an airport with many Susceptible neighbours.

Task 3: Effect of seed node selection on spreading speed

Next, we will investigate how the selection of the initially infected seed node affects the spreading speed.

a) Use nodes with node-ids [0, 4, 41, 100, 200] (ABE, ATL, ACN, HSV, DBQ) as seeds and $p = 0.1$, and run the simulation 10 times for each seed node. Then, plot the average prevalence of the disease separately for each seed node as a function of time (recycling your code for Task 2).

Solution. See Figure 2.

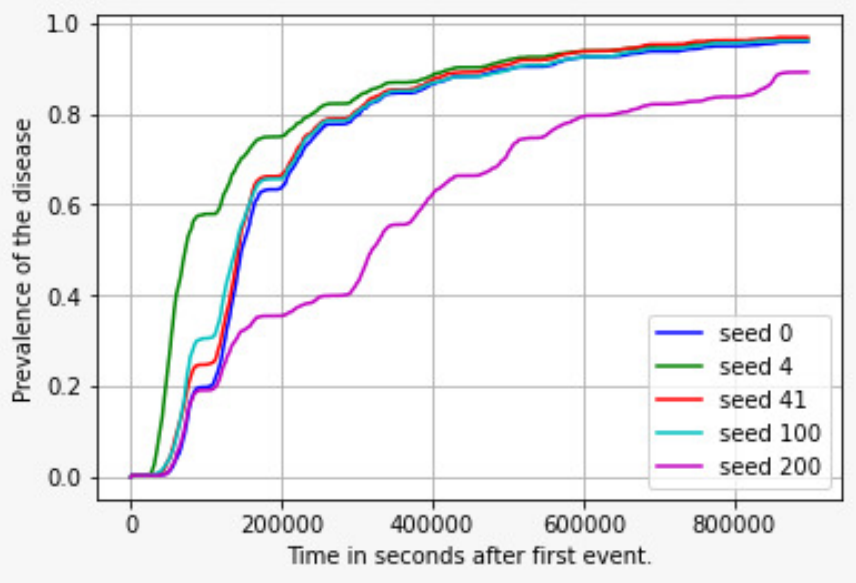

Figure 2: Averaged prevalence $\rho(t)$ of the disease as a function of time for each of the seed nodes

b) The differences in spreading speed between seeds should be mostly visible in the beginning of the epidemic. Explain, why.

Solution.

Because the spreading speed will be faster if the first nodes are more connected. And as more nodes get infected, all executions look more alike because they will have more infected nodes in common.

c) In the next tasks, we will, amongst other things, inspect the vulnerability of a node for becoming infected with respect to various network centrality measures. Why will it be important to average the results over different seed nodes?

Solution.

Because, as we could observe in this task, the prevalence of the disease can change remarkably depending on the seed node.

Task 4: Where to hide?

Now, consider that you'd like to be as safe from the epidemic as possible. How should you select your refuge? To answer this question, run your SI model 50 times with $p = 0.5$ using different random nodes as seeds and record the median infection times for each node.

In this task, you can use the pre-built static network (`aggregated_US_air_traffic_network_undir.edg`) to compute the various centrality measures using the ready-made NetworkX functions.

a) Run the 50 simulations, and create scatter plots showing the median infection time of each node as a function of the following nodal network measures:

i) *unweighted* clustering coefficient c

ii) degree k

iii) strength s

iv) *unweighted* betweenness centrality bc

Solution. See Figure 3.

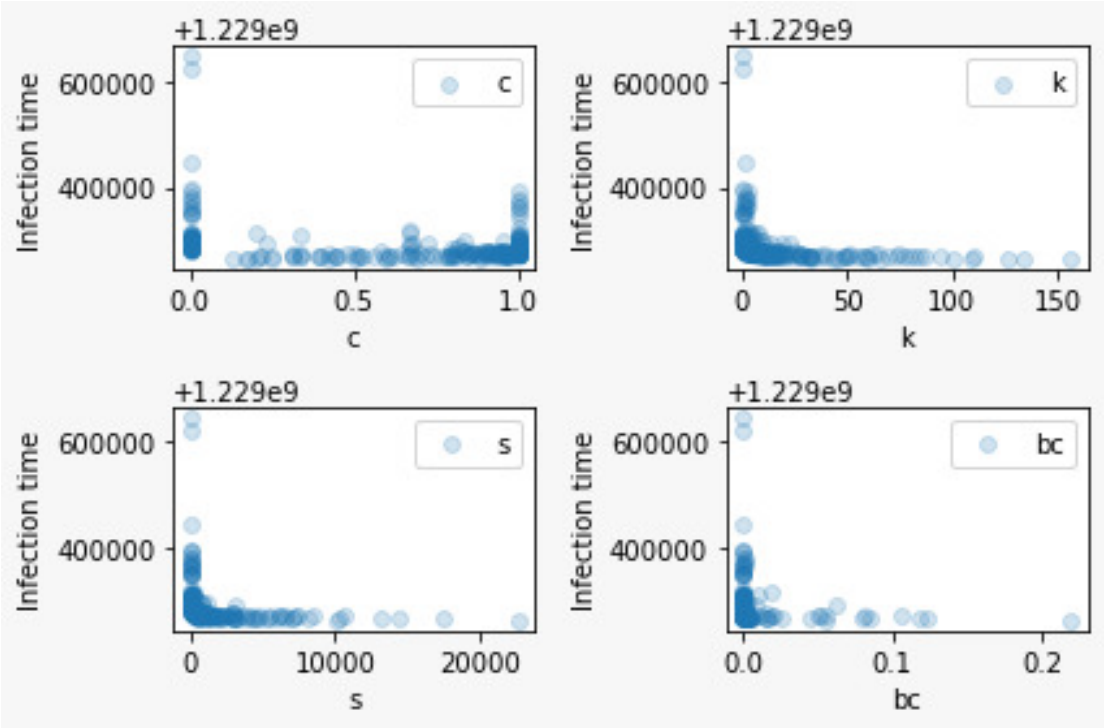

Figure 3: Median infection time of each node as a function of the different nodal network measures

b) Use the Spearman rank-correlation coefficient for finding out which of the measures is the best predictor for the infection times[1].

Solution. See Table 1.

[1]We use Spearman rank-correlation coefficient instead of the linear Pearson's coefficient, as we can not assume the dependency between the average infection time and different nodal network measures to be linear.

Measure	Spearman rank-correlation
c	-0.11770845531769591
k	-0.804752400178131
s	-0.8809395054684258
bc	-0.6247292260844652

Table 1: Spearman rank-correlation coefficient for each of the nodal network measures.

c) Based on your results, answer the following questions:

- Which measure(s) would you use to pick the place to hide, i.e. which measure best predicts node infection time? Why?

  Solution.

  As we can observe in Table 1 the Spearman rank-correlation gives the highest absolute values for the strength (s) and the degree (k) measures. That result makes sense since a node with high strength and degree (has a lot of neighbours and they are highly connected between them) has high probabilities to get infected if one of the neighbours is infected. Therefore, to be safe we should choose a node with the lowest possible strength and degree.

- Why does betweenness centrality behave differently than degree and strength?

  Solution.

  As we observe also in Table 1 the *unweighted* betweenness centrality (bc) it's a worse measure than the degree (k) and strength (s) ones. The measure works good for nodes with high betweenness centrality but it can't make good predictions for nodes with low betweenness centrality.

- Why is clustering coefficient a poor predictor?

  Solution.

  As seen in Table 1 the *unweighted* clustering coefficient (c) is the worst measure. This is because the clustering coefficient says how well connected are the neighbours of the studied node, but actually, it doesn't say anything about the connection of that node so it's useless for predicting the infection times.

Task 5: Shutting down airports

Now take the role of a government official considering shutting down airports to prevent the disease from spreading to the whole country. In our simulations, the shutting down of airports corresponds to immunization: an airport that has been shut down can not become infected at any point of the simulation.

One immunization strategy suggested for use in social networks is to pick a random node from the network and immunize a random neighbour of this node. Your task is now to

compare this strategy against five other immunization strategies: the immunization of random nodes and the immunization of nodes that possess the largest values of the four measures of centrality/importance that we used in task 4. In this exercise, use $p = 0.5$ and average your results over 20 runs of the model for each immunization strategy (120 simulations in total).

To reduce the variance due to the selection of seed nodes, use same seed nodes for investigating all immunization strategies. To this end, first select your immunized nodes, and then select 20 random seed nodes such that none of them belongs to the group of immunized nodes in any of the 6 different strategies.

a) Adapt your code to enable immunization of nodes, and plot the prevalence of the disease as a function of time for the 6 different immunization strategies (social net., random node, and 4 nodal network measures), always immunizing 10 nodes.

Solution. See Figure 4.

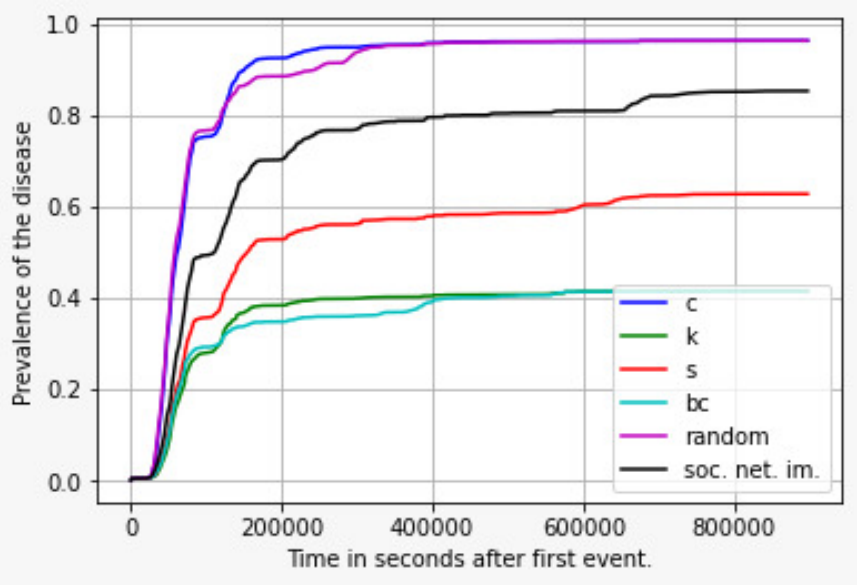

Figure 4: Averaged prevalence $\rho(t)$ of the disease as a function of time for the 6 different immunization strategies

b) Based on your results, answer the following questions:

- Which of the immunization strategies performs the best, and why?

  Solution.

  As we can see in Figure 4 the strategy that performs best are the Degree k and the Betweenness centrality bc.

  For the Degree k this happens because immunizing the nodes with highest degree we ensure that they won't infect any of their neighbours.

With the Betweenness centrality the good performance is explained because a lot of the shortest paths of the network are now disabled, therefore the spreading speed will be reduced.

- Why does betweenness centrality perform better as an immunization strategy than as a predictor for a safe hiding place?

  Solution.

  Because for the immunization strategy we are taking only the 10 nodes with highest betweenness centrality and with the predictor for a safe place we are using all the nodes. The 10 nodes used in the immunization strategy are in fact a good predictor for safe places to hide, but the problem are the nodes with lower value of betweenness centrality that can't be used as a predictor.

c) The pick-a-neighbour immunization strategy probably worked better than the random node immunization[2]. Let us try to understand why.

- First, if the degree distribution of the network is $P(k)$, what is the probability of picking a random node of degree k?

  Solution.

  The probability would be $P(k)$.

- What is the expected outcome if you then pick a random neighbour of the random node?

  Solution.

  The expected outcome is to obtain a node with higher degree than the first one selected. This is explained with the friendship paradox that says that on average a person has less friends than their average friend. Furthermore for a node with degree k the average neighbor has a degree of $\langle k_{nn} \rangle = \frac{\langle k^2 \rangle}{\langle k \rangle}$.

- Consequently, which of the strategies is expected to be more effective and why?

  Solution.

  As shown in the previous questions the pick-a-neighbour immunization is a more effective way.

d) Although the social network immunization strategy outperforms the random immunization, it is not necessarily as effective as some other immunization strategies (and there is random variation). Nevertheless, **explain** shortly, why it still makes sense to use this strategy in the context of social networks?

Solution.

It is used because it's very easy to implement and it is very fast (constant) even for large networks since it doesn't need to go through all the nodes to identify max values of a measure.

[2]However, due to the randomness of the selection process there is a small probability that this was not the case for you.

Task 6: Disease transmitting links

So far we have only analyzed the importance of network nodes—next, we will discuss the role of links. We will do this by recording the number of times that each link transmits the disease to another node. So adapt your code for recording the (undirected) static links which are used to transmit the disease. You can do this e.g. by storing for each node from which other node it obtained the infection.

Run 20 simulations using random nodes as seeds and $p = 0.5$. For each simulation, record which links are used to infect yet uninfected airports (either by first infection-carrying flights arriving to susceptible airports or by infecting flights arriving before the already recorded infection time).

a) Run the simulations, and compute the fraction of times that each link is used for infecting the disease (f_{ij}). Then use the provided function `plot_network_USA` which can be found in `si_animator.py` to visualize the network on top of the US map (see the example code given in the function). Adjust the width of the links according to the fractions f_{ij} to better see the overall structure. Compare your visualization with the maximal spanning tree of the network.

Solution. See Figures 5 and 6.

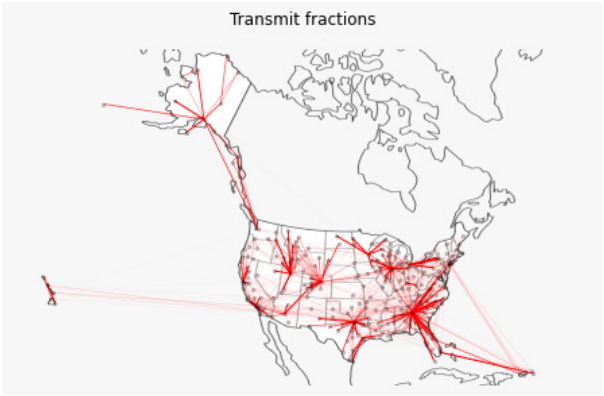

Figure 5: Fraction of times that each link is used for infecting the disease (f_{ij}).

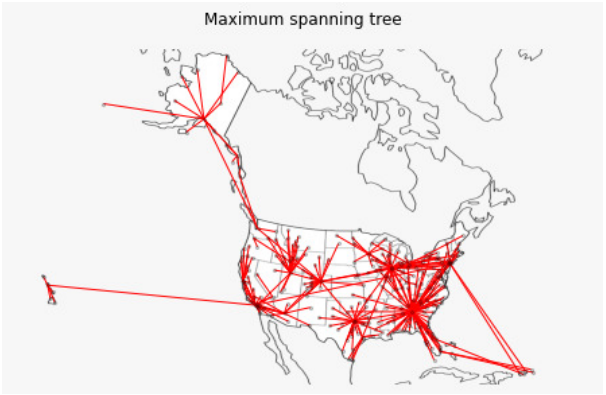

Figure 6: Maximal spanning tree of the network.

b) Explain why your visualization is similar to the maximal spanning tree.

Solution. Because the maximum spanning tree shows the links with more weight and our visualization shows (for each edge), each unit of weight with a probability p. This way the links that have more weight are more likely to appear wider in our visualization.

c) Create scatter plots showing f_{ij} as a function of the following link properties:

i) link weight w_{ij}

ii) *unweighted* link betweenness centrality eb_{ij} (`edge_betweenness_centrality` in networkx)

Compute also the Spearman correlation coefficients between f_{ij} and the two link-wise measures.

Solution. See Figure 7.

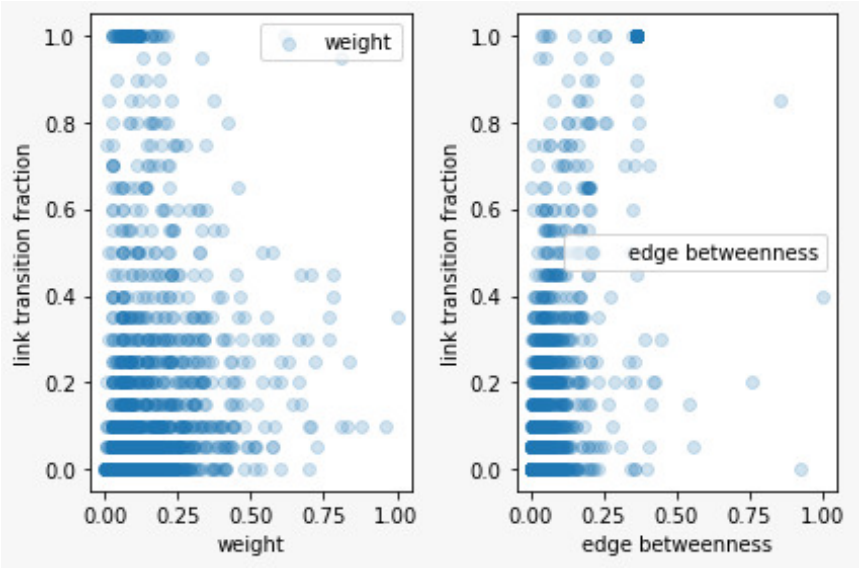

Figure 7: Scatter plots showing f_{ij} as a function of link weight w_{ij} and *unweighted* link betweenness centrality eb_{ij}

Measure	Spearman rank-correlation
w_{ij}	0.39452473303657554
eb_{ij}	0.5054855796468862

Table 2: Spearman rank-correlation coefficient for each of the link-wise measures.

d) Explain the performance of the two link properties for predicting f_{ij}.

Solution.

As we can see in Table 2 both link properties obtained a quite bad score for the Spearman rank-correlation.

In the case of the link weight this is because, even though a link with high weight will be more likely to transfer the infection, the links with low weight does not assure low probability of transmission.

For betweenness centrality we obtained a bit better correlation, but still it's not very good. This happens because links with high betweenness centrality tend to be

centrally located so that they have higher probability of transmission. However, this is not a rule and it is not always applied so it's not a good method of prediction.

Task 7: Discussion

Even though extremely simplistic, our SI model could readily give some insights on the spreading of epidemics. Nevertheless, the model is far from an accurate real-world estimate for epidemic spreading.

Discuss the deficiencies of the current epidemic model by listing at least four (4) ways how it could be improved to be more realistic.
Solution.

1. One way of making more accurate the model would be taking into account also other ways of transportation such as buses, trains, etc.

2. Another way could be taking into account how crowded is the flight and making the probability of transmission proportional to density of people.

3. Also we could take into account the percentage of infected people per city and also making the probability of infection proportional to it.

4. Finally (but there could be much more improvements) we could take in consideration the restrictions each city is having so that if a city has a lot of restrictions the probability of infection will be lower.