Course project (52 pts)

- The deadline for handing in your solutions is 21 December 2018 23:55.
- Return your solutions (one `.pdf` file and one `.zip` file containing Python code) in MyCourses (Assignments tab). Additionally, submit your pdf file also to the Turnitin plagiarism checker in MyCourses.
- Check also the course practicalities page in MyCourses for more details on writing your report.

- Note that you need to get at least 25 points from the project in order to pass the course!
- There is no code template similar to the normal exercises in this course.
- Some of the tasks are marked as bonus tasks. These are more open ended tasks which might be more difficult and time consuming. They can have multiple valid solutions and allows for some creativity. It is not expected that all students complete the bonus task. Further, we recommend that you complete the normal tasks before attempting the bonus tasks.
- Discussing project solutions with other students is allowed when properly indicated in the report. Each student should nevertheless return his/her own report.

Introduction

In the project work, your task is to implement an susceptible-infected (SI) disease spreading model and run it against time-stamped air transport data. Especially, we will investigate how static network properties can be used to understand disease spreading.

Model specifications

In our implementation of the SI-model, there is initially only one infected airport and all other airports are susceptible. A susceptible node may become infected, when an airplane originating from an infected node arrives to the airport. (Note that, for the airplane to carry sick passengers, the source airport needs to be infected at the departure time of the flight.) If this condition is met, the destination airport becomes infected with probability $p \in [0, 1]$ reflecting the infectivity of the disease. Nodes that have become infected remain infected indefinitely.

Data description

The data that are used in the project is available at the course MyCourses page in one `.zip` file. The flight data that you will use to perform your simulation is located in file `events_US_air_traffic_GMT.txt`, where each row contains following fields:

```
1st column -> Source [0-278]
```

```
2nd column -> Destination [0-278]
3rd column -> Start Time (GMT) [seconds after Unix epoch time]
4th column -> End Time (GMT)
5th column -> Duration [Same as (EndTime-StartTime)]
```

The aggregated weighted network `aggregated_US_air_traffic_network_undir.edg` is constructed based on the event data, so that weight of each link corresponds to the number of flights between the nodes. Additionally, you can find the information about the airports in file `US_airport_id_info.csv`.

Provided code

To help you in verifying that your code actually works, we provide a Python module `si_animator.py`. In the module, you can find a function called `visualize_si`, which animates the air transport using `matplotlib`. This animation is not needed to complete the project but it looks nice.

For the animations to work, the background image `US_air_bg.png`, the airport id-info file `US_airport_id_info.csv` and the events file `events_US_air_traffic_GMT.txt` need to be located in the directory where you are running Python.

See the documentation within the module for further usage info.

General hints on implementing the model:

- For implementing the SI model, it is practical to keep track of the infection times of all nodes. Initially one node is infected (with an infection time equal to the departure time of the first flight). The infection times of the rest of the nodes can be initialized *e.g.* to `float('inf')`.
- When looping over time, loop over the flights in ascending order of the departure time of the events. This way you always know, whether the current departure of the flight has been infected or not.
- A node is infected (and can infect other nodes) only if the current simulation time (*i.e.,* the departure time of the next flight) is larger than the infection time of the node.
- If a flight mediates the disease (with probability p), the infection time of the target node becomes only updated if the new infection time is smaller than the target nodes existing infection time.
- For reading in the csv data, there are many options, such as `numpy.genfromtxt`, the `read_csv` function from the `pandas` package, or the built-in `csv` library in the Python standard library.
  For example, with `numpy.genfromtxt` use something like
  `event_data = np.genfromtxt('events_US_air_traffic_GMT.txt', names=True, dtype=int)`
  `sorted_data = event_data.sort(order=['StartTime'])` for loading and sorting the data.
- You may freely use any functions from `networkx`, `matplotlib`, `numpy` and `scipy` that you find useful.

Tasks:

Task 1: Basic implementation (5 pts)

Implement the SI model using the temporal air traffic data. 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) (5 pts) 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?
Hint: The time point should fall within the range `1229290000-1229291000`.

Task 2: Effect of infection probability p on spreading speed (5 pts)

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) (4 pts) 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.
Hints:
   - For creating the time axis, divide the whole time spam, from the first departure to the last arrival, into equal-sized steps using *e.g.* `np.linspace`.
   - For each step, you should be able to calculate, how many nodes *on average* were infected before that step. To obtain the prevalence, normalize by the number of nodes.
   - **Note:** The prevalence at time t should be averaged over the 10 iterations, *i.e.* there is no need to average the infection times of individual nodes in this task!.

b) (1 pt) For which infection probabilities does the whole network become fully infected? What are the stepwise, nearly periodic "steps" in the curves due to?

Task 3: Effect of seed node selection on spreading speed (5 pts)

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

a) (3 pts) 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.

b) (1 pt) You should be able to see differences in these spreading speed. Are these differences visible in the beginning of the epidemic or only later on? Why?

c) (1 pts) 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 is it important to average the results over different seed nodes?

Task 4: Where to hide? (7 pts)

Now, consider you want 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 of each node. Note that the median infection time is not well defined for nodes that become infected in less than 25 runs. You may leave those nodes out from your analyses.

a) (4 pts) 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) k-shell
   ii) *unweighted* clustering coefficient c
   iii) degree k
   iv) strength s
   v) *unweighted* betweenness centrality
   vi) closeness centrality

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

   Hint: `scipy.stats.spearmanr`

c) (2 pts) Discuss your results for each network centrality metric. Especially, explain the ranking of the network measures as measured by the median infection time.

Task 5: Shutting down airports (10 pts)

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 one of this focal node's neighbors. Your task is now to compare this strategy against seven other immunization strategies: the immunization of random nodes and the immunization of nodes that possess the largest values of the six nodal network measures 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 (160 simulations in total).

To reduce the variance due to the selection of seed nodes, use same seed nodes when investigating each immunization strategy: 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 8 different strategies.

a) (5 pts) Adapt your code to enable immunization of nodes, and plot the prevalence of the disease as a function of time for the 8 different immunization strategies (social net., random node, and 6 nodal network measures) when 10 nodes are always immunized.

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

b) (2 pts) Discuss the ranking of the immunization strategies. In particular, compare your immunization results with the results you obtained in the previous task (Task 4). Are there some measures that are bad at predicting the infection time but important with regards to immunization? Or vice versa? Why?

c) (2 pts) The network immunization strategy suggested for use in social networks should have worked better than the random node immunization[2]. Let us next explain this mathematically by investigating the probability of picking high-degree nodes at random and by the social net immunization strategy:

- First, what is the probability of picking a random node with degree k (in any given network)?
- Let's assume that degrees of neighboring nodes are independent in the air transport network. Now, what is the probability to pick a node with degree k by the social net immunization strategy?

  Hint: How many options there are to pick a neighbor of a k-degree node when picking at random?
- Which of the strategies, random or social net immunization, has higher probability of picking high-degree nodes? How would you thus explain the difference between the two strategies?

How would you thus explain the difference between the two strategies?

d) (1 pt) Although the social network immunization strategy outperforms the random immunization, it is not as effective as some other immunization strategies. **Explain**, why it still makes sense to use this strategy in the context of social networks?

Task 6: Disease transmitting links (8 pts)

So far we have only analyzed the importance of network nodes, but next we will discuss the role of links. We do this by recording the number of times each link transmits the disease to another node. So adapt your code to recording the (undirected) links which are used to transmit the disease. This is best done by storing for each node where 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.

a) (4 pts) Run the simulations, and compute the fraction of times 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. 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.

b) (1 pt) What do you notice? How would you explain your finding?

c) (2 pts) Create scatter plots showing f_{ij} as a function of the following link properties:

i) link weight w_{ij}

ii) *link neighborhood overlap* O_{ij}

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

iii) *unweighted* link betweenness centrality eb_{ij}

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

d) (1 pt) Explain the performance of the three link properties for predicting f_{ij}.

BONUS task (5 pts)

None of the above measures was extremely good for predicting f_{ij}. As a bonus task, you can come up with a measure of your own, or find out an appropriate measure from the literature that you think is better for predicting f_{ij} than the three measures listed above. Motivate selection of method, and perform similar investigations as with the other link properties. Especially evaluate the Spearman correlation between the measure you developed and f_{ij}.

Note:
Remember that your measure should be based only on the information contained in the weighted network *i.e.* do not use geographical or time-stamped air traffic data for devising your measure.

Grading guideline for the bonus:
Good effort, 1 pt; a solution that is better than the three other measures min. 2 pts; for more sophisticated/elaborated/principled attempts and efforts: max. 5 pts. Especially novel, well motivated approaches will be valued (even if the end results would not be outstanding). If you end up trying multiple measures, reporting more than one measure can gain you more points.

Task 7: Discussion (2 pts)

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.

Hint: Check *e.g.* Ref. [2] for ideas.

BONUS task: (max. 5 pts)

You're now given free hands to extend the analysis of the SI model in any way you wish. You may use the SI model and compare its results to some more detailed network analyses and/or extend it to be more realistic in any way you please. For the latter, Refs. [2] and [3] can be helpful. If you decide to extend your model, provide simulation results that illustrate the new behavior of your model (especially show how your improved model differs from the simple SI model). Remember to thoroughly discuss your findings.

Feedback (1 pt)

To earn one bonus point, give feedback on this exercise set and the corresponding lecture latest two day after the report's submission deadline.
Link to the feedback form: `https://goo.gl/forms/cQJfRNw2F2LdCK173`.

References

[1] [Online]. Available: http://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

[2] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *arXiv preprint arXiv:1408.2701*, 2014.

[3] A. A. Alemi, M. Bierbaum, C. R. Myers, and J. P. Sethna, "You Can Run, You Can Hide: The Epidemiology and Statistical Mechanics of Zombies," p. 12, 2015.