Exercise set #7 (18 pts)

- The deadline for handing in your solutions is November 12th 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.

1. Weight–topology correlations in social networks (12 pts)

In this exercise, we will do some weighted network analysis using a social network data set describing private messaging in a Facebook-like web-page[1]. In the network, each node corresponds to a user of the website and link weights describe the total number of messages exchanged between users.

In the file `OClinks_w_undir.edg`, the three entries of each row describe one link:
`(node_i node_j w_ij)`,
where the last entry `w_ij` is the weight of the link between nodes `node_i` and `node_j`.

You can use the accompanying Python template (`weight_topology_correlations.py`) to get started. `scipy.stats.binned_statistic` function is especially useful throughout this exercise.

a) (3 pts) Before performing more sophisticated analysis, it is always good to get some idea on how the network is like. To this end, plot the complementary cumulative distribution (1-CDF) for node degree k, node strength s and link weight w.

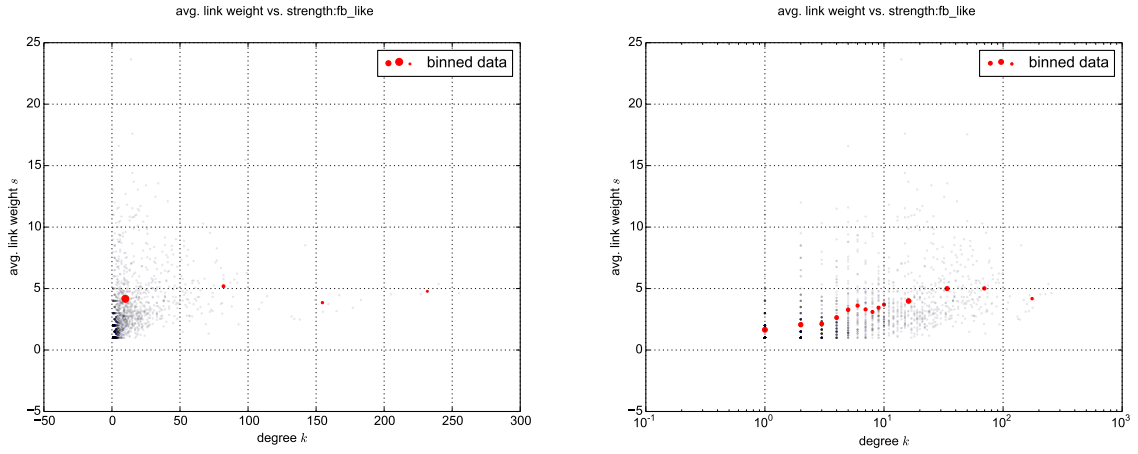
   - **Show** all three distributions **in one plot** using loglog-scale.
   - Briefly **describe** the distributions: are they Gaussian, power laws or something else?
   - Based on the plots, roughly **estimate** the 90th percentiles of the degree, strength, and weight distributions.

   Hints:

   - For reading in the network, use `net = nx.read_weighted_edgelist`
   - See the binning tutorial for help on computing the 1-CDFs.
   - For getting node strengths, use `strengths = nx.degree(net, weight="weight")`

b) (2 pts) Next, we will study how the average link weight per node $\langle w \rangle = \frac{s}{k}$ behaves as a function of the node degree k. **Compute** s, k, and $\langle w \rangle = \frac{s}{k}$ for each node. **Make a scatter plot** of $\langle w \rangle$ as a function of k. Create two versions of the scatter plot: one with linear and one with logarithmic x-axes.

[1] Data originally from `http://toreopsahl.com/datasets/`



(a) (b)

Figure 1: Example of $\langle w \rangle$ as a function of k using another Facebook-like social network data. 1a: linear axes. 1b: logarithmic axes.

c) (2 pts) The large variance of the data can make the scatter plots a bit messy. To make the relationship between $\langle w \rangle$ and k more visible, **create bin-averaged versions** of the plots, *i.e.* divide nodes into bins based on their degree and calculate the average $\langle w \rangle$ in each bin. Now, you should be able to spot a trend in the data.

Hints:

- For the linear scale, use bins with constant width. For the logarithmic scale, use logarithmic bins. If in trouble, see the binning tutorial for help.
- Use the number of bins you find reasonable. Typically, it is better to use too many than too few bins.
- An example of how the scatter and bin-averaged plots may look like is shown in Fig. 1. Note that the results will vary according to the number of bins.

d) (2 pts) Based on the plots created in b), **answer** the following questions questions:

- Which of the two approaches (linear or logarithmic x-axes) suits better for presenting $\langle w \rangle$ as a function of k? Why?
- In social networks, $\langle w \rangle$ typically decreases as a function of the degree due to time constraints required for taking care of social contacts. Are your results in accordance with this observation? If not, how would you explain this?

Hints:

- Check your results from a). Is there an equal number of nodes with each degree and strength? Nonequal distribution of observations may obscure the results.
- You are dealing with real data that may be noisy. So, interpretation of results may be confusing at first - do not worry!

e) (3 pts) Lets consider a link between nodes i and j. For this link, *link neighborhood overlap* O_{ij} is defined as the fraction of common neighbors of i and j out of all their neighbors:

$$O_{ij} = \frac{n_{ij}}{(k_i - 1) + (k_j - 1) - n_{ij}}. \quad (1)$$

According to the Granovetter hypothesis, link neighborhood overlap is an increasing function of link weight in social networks. Next, your task is now to find out whether this is the case also for the present data set.

To this end, **calculate the link neighborhood overlap** for each link. **Create a scatter plot** showing the overlaps as a function of link weight. As in c), **produce** also **a bin-averaged version** of the plot. Use a binning strategy that is most suitable for this case. In the end, you should be able to spot a subtle trend in the data. Based on your plot, **answer** the following questions:

- Is this trend in accordance with the Granovetter hypothesis? If not, how would you explain your findings?

2. Network thresholding and spanning trees: the case of US air traffic (6 pts)

In this exercise, we will get familiar with different approaches to thresholding networks, and also learn how they can be used for efficiently visualizing networks. Now, you are given a network describing the US Air Traffic between 14th and 23rd December 2008[2]. In the network, each node corresponds to an airport and link weights describe the number of flights between the airports during the time period.

The data and some code for visualizing the network is provided at the course web-page. The network is given in the file `aggregated_US_air_traffic_network_undir.edg`, and `us_airport_id_info.csv` contains information about names and locations of the airports. The file `air_traffic_network_base.py` contains a function for visualizing the air transport network, and an example how to use it. You can extend your own work to the same file, or import the file as a Python module. In this exercise, you may also freely use all available `networkx` functions.

a) (1 pt) When facing a new network, it is always good to first get some idea, how the network is like. Thus, **compute** and list the following basic network properties:

- Number of network nodes N, number of links L, and density D
- Network diameter d
- Average clustering coefficient C

  Hint: For the clustering coefficient, consider the undirected and unweighted version of the network, where two airports are linked if there is a flight between them in either direction.

b) (1 pt) **Visualize** the full network with all links on top of the map of USA. The resulting figure is somewhat messy due to the large number of visible links.

c) (2 pts) In order to reduce the number of plotted links, **compute** both the *maximal* **and** *minimal spanning tree* (MST) of the network and **visualize** them. Then, **answer** following questions:

- If the connections of Hawai'i are considered, how would you explain the differences between the minimal and maximal spanning trees?

[2]Data from `http://www.rita.dot.gov/bts/`

- If you would like to understand the overall organization of the air traffic in US, would you use the minimal or maximal spanning tree? Why?

d) (2 pts) **Threshold and visualize** the network by taking only the strongest M links into account, where M is the number of links in the MST. Then, **answer** following questions.

- How many links does the thresholded network share with the maximal spanning tree?
- Given this number and the visualizations, does simple thresholding yield a similar network as the maximum spanning tree?

Hint: For computing minimum spanning trees, use `nx.minimum_spanning_tree`. Note that you can obtain the maximal spanning tree by computing the minimal spanning tree with negated weights.

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: `nourlyet`.

References