

Exercise set #8 (17 pts)

- The deadline for handing in your solutions is November 13th 2023 23:59.
- There are multiple ways for submitting your solutions. Check each question for details.
- Return also one `.zip` file containing all your Python code of the round in MyCourses.
- Check also the course practicalities file in MyCourses for more details on submitting your solutions.

1. Social networks: social signatures and weight-topology correlations (17 pts)

In this exercise, we will do some weighted network analysis using a social network dataset describing private messaging in a Facebook-like webpage¹. 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 `0Clinks_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 code template to get started.

The function `scipy.stats.binned_statistic` is especially useful throughout this exercise.

Submit all your solutions to MyCourses as a single pdf report. Remember to include in your report all the visualizations as well as all the numerical and verbal answers.

- a) (4 pts) Before performing a more sophisticated analysis, let us begin with taking a look at some basic network statistics to get a rough idea of what the network is like. To this end, **plot** the complementary cumulative distribution function (CCDF) for node degree k , node strength s and link weight w .
- Show all three distributions in one plot using log-log scale.
 - The resulting distributions have *heavy tails*, meaning that the right part of the distributions decays slower than exponentially. To see this, **plot** in the same figure the CCDFs of exponential distributions² having the same mean as each empirical distribution.
 - Based visually on the empirical CCDF plots, roughly **estimate** the 90th percentiles of the degree, strength, and weight distributions. **Explain** also how can they be read off from the plots.

Hints:

- See the binning tutorial for help on computing the CCDFs.

¹Data originally from <http://toreopsahl.com/datasets/>

²Check https://en.wikipedia.org/wiki/Exponential_distribution

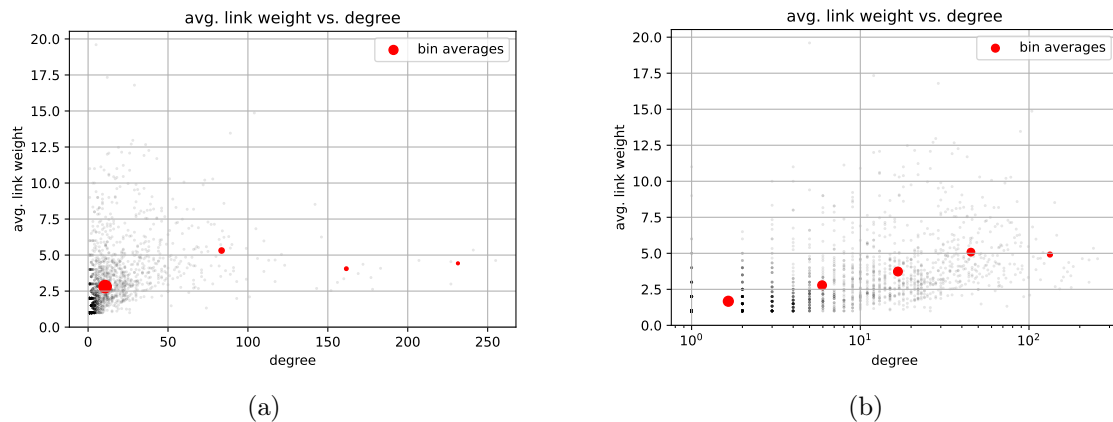


Figure 1: Example of v as a function of k using another Facebook-like social network data. 1a: linear axes. 1b: log-lin axes.

- As a check, calculate the maximum degree k_{\max} and maximum strength s_{\max} . If everything is running okay so far, you should be getting $k_{\max} = 255$ and $s_{\max} = 1546$.
 - The CCDF of an exponential distribution with mean μ is given by $\exp(-x/\mu)$.
 - For further programming tips, see the coding template available in JupyterHub.
- b) (5 pts) Next, we will study how the average link weight per node $v = \frac{s}{k}$ behaves as a function of the node degree k .
- **Compute** s , k , and v for each node.
 - Make **a scatter plot** of all the data points of v as a function of k . Create two versions of the plot: one with linear axes and one with logarithmic horizontal axes.
 - The large variance of the data can make the scatter plots a bit messy. To make the relationship between v 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 v in each bin. Plot the bin-averaged versions on top of the scatter plots.
 - Based on the plots, **which** of the two approaches (linear or logarithmic horizontal axes) suits better for presenting v as a function of k ? **Why?**

Hints:

- For the bin-averaged plots, use bins that are consistent with the scale of the horizontal axis: bins with constant width for the linear scale and logarithmic bins for the logarithmic scale. If in trouble, see the binning tutorial for help.
- An example of how the scatter and bin-averaged plots may look like is shown in Fig. 1. Obviously, the number of bins is too low in these plots. Typically, it is better to use too many than too few bins. A good choice for the number of bins in this case would be 20.
- Check that each degree bin includes roughly the same number of nodes. An unbalanced distribution of observations may obscure the results.

- c) (4 pts) Next, we will study the social signature of egocentric networks. The social signature of a node is a vector that describes the distribution of link weights in the egocentric network of the node. We define the social signature of node i with degree k_i as

$$\mathbf{p}_i = (p_{i,1}, p_{i,2}, \dots, p_{i,k_i}), \quad p_{i,1} \geq p_{i,2} \geq \dots \geq p_{i,k_i}, \quad (1)$$

where $p_{i,j}$ is the weight of j -th link (sorted in the descending order) incident to node i normalized by the sum of the weights of all links incident to i (i.e., strength of node i) :

$$p_{i,j} = \frac{w_{i,j}}{\sum_{j=1}^{k_i} w_{i,j}}. \quad (2)$$

Compute the social signatures of the nodes in the network that have degree $k_i = 25$ and **plot** the signatures $p_{i,j}$ as a function of rank j .

- d) (4 pts) Let's 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:

$$O_{ij} = \frac{n_{ij}}{(k_i - 1) + (k_j - 1) - n_{ij}}, \quad (3)$$

where n_{ij} denotes the number of common neighbors.

According to the Granovetter hypothesis, link neighborhood overlap is an increasing function of link weight in social networks. Your task is to find out whether this is the case also for the present dataset by visualizing it in an appropriate way. Use the binning strategy (linear or logarithmic) that is most suitable for this case:

- **Calculate** the link neighborhood overlap for each link.
- Create a **scatter plot** showing the overlap for every link as a function of link weight.
- As in b), **produce** a bin-averaged version of the plot on top of the scatter plot. In doing this, choose a reasonable number of bins (which, in this case, will be between 10 and 30). **Specify** how many bins you used.
- In the end, you should be able to spot a subtle trend in the data. Based on your plot, **comment** on if the trend is in accordance with the Granovetter hypothesis.

Hints:

- For getting common neighbors of two nodes, you can use `set` datatype and `intersection()` method.
- A naive implementation of Eq. (3) will result in `ZeroDivisionError` for some links. Examine for what kind of links this happens. What would be the appropriate way to define O_{ij} for those links?

Feedback (1 pt)

To earn one bonus point, give feedback on this exercise set and the corresponding lecture latest two days after the exercise round's submission deadline. You can find the feedback form in MyCourses.

Aalto University
CS-E5740 Complex Networks, Autumn 2023
Hiraoka, Kivelä, Korhonen
Exercise set #8

References