ASDSI - Social Network Analysis 2162-F23: Exercise 6 - Graph Theory

2023-03-13 - **Lecture 11**

1.1 Adjacency matrix and edge list

Re-read the concepts of the adjacency matrix and edge list (slides and reading materials). Then inspect manually the Game of Thrones coappearances network file `got_edges.csv` (an undirected network). You will find that it contains two columns. Load then this file in your Jupyter Notebooks either using python lists (one list for each column) or the package `pandas`. You have now a network in edge list format in your programming environment. Then answer the questions below, we suggest some Python or networkX functions, but feel free to use any other method. Please note that you will not need a network analysis module (e.g.NetworkX) until the second class.
Note: In the same folder, you will find the files "got_nodes.csv" and "got_gprops.csv" containing metadata about the network. They are useful for understanding what node numbers mean, but are not needed for the computations.

`Functions: open(), readlines(), strip(), split(), append()`

1.1.1

How many nodes and links are present in the network? What is the size (in MB) of the edge list?

`Functions: <objname>.__sizeof__(), max(), len()`

1.1.2

Convert the edge list into an adjacency matrix in `pandas`, keeping in mind that this is an **undirected** network. What is the size of this matrix? Have you noted any difference in size or in the number of entries to store the same information?

`Functions: pd.DataFrame(?, columns=?, index=?), df.at[?,?], df.shape`

1.1.3

Which representation, edge list or adjacency matrix, do you think should be used for storage? Which one to use for calculation? Why?

1.1.4

Can you spot an isolated node (i.e. a node with no links) in the adjacency matrix of a graph? How? And in the edge list of a graph? How?

`Functions: df.sum(), range(), 'for loop', len()`

1.1.5

(Optional) Load the edge list using the file `got_edges.csv`. Can you think of an alternative storage of the network that is not an edge list or matrix? For instance, if you need to know all the neighbours of a node, what would be a good data representation?

```
Functions: dict(), append(), 'for loop'
```

1.1.6

(Challenging) Construct an edge list from non-network data, e.g. text. For this task, use the first Chapter of Alice in wonderland `DowntheRabbitHole.txt` and build the edge list according to the following procedure: Nodes are defined by words, and they are linked if they appear between 2 punctuation marks, e.g.: [. , ; ! ? etc.].

Save the edge list as a CSV file, it can used later in future exercises.

1.2 Network formation: Group discussion

Discuss possible networks that you are interested in (Not present among the provided datasets). To guide the discussion, think about what are the nodes, the links and the processes behind the relationship between nodes (link formation).

Write a short paragraph, describing this discussed network. Make sure to define clearly the nodes and the links. Justify why you choose to define nodes and links this way, and explain why you would be interested in studying this network.

1.3 Matrix multiplication

Calculate **manually** the following matrix multiplication. Feel free to do the multiplication on a piece of paper and incorporate a photo in the jupyter notebook.

$$\begin{bmatrix} 3a & 5b & 1 \\ 1 & 0 & 3c \\ a & b & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \end{bmatrix} = ?$$

2.1 Network identity card

Reread the concepts of the network identity card discussed in lecture 12. In this exercise, you will build the table discussed in the lecture slides. For that, you will use the python package `NetworkX` which includes several measures and functions for processing network data.

Details on how to install and search for functions in NetworkX can be found in the file `Installing NetworkX.pdf` and a tutorial on how to start with the module can be found HERE.

2.1.1

For each network in the list below, consider its edge list and load it through the networkX module using the command `nx.read_edgelist()`. Then compute the measures specified in the identity card in lecture 12 and make a table. You will repeat this process for another two networks.

(a) Game of Thrones coappearances used in the exercise of lecture 11 (exercise 1.1).

(b) An **undirected** network of your choice from `https://networks.skewed.de`

```
Functions: nx.read_edgelist(), nx.info(), nx.Graph(), nx.density(),
nx.diameter(), nx.average_shortest_path_length(), nx.clustering(), G.degree()
```

2.1.2

Which network among a and b is denser? Which one has a lower shortest path? Is it expected and why/why not?

2.1.3

(optional) Draw the two networks using different layouts algorithms available in NetworkX. See HERE for examples. How changing the layout change the interpretation of the network?

```
Functions: nx.draw(), plt.subplots(), kamada_kawai_layout(), circular_layout()
```

2.2 Clustering coefficient

Manually draw a connected network with five nodes, i.e., a network where there is a path connecting any two given nodes. Label the nodes and manually calculate the clustering coefficient of this network using the lectures' definition.

2.3 Centrality

Find the most central node in the `Game of Thrones` network according to the centrality measures presented in class: degree, betweenes and eigenvector centrality.

(a) Do all measures give the same node? Why, or why not?

(b) What about the five most central nodes, do you obtain the same rank with different measures?

```
Functions: nx.degree_centrality(), nx.eigenvector_centrality(),
nx.betweenness_centrality()
```