Basic network properties
Data Workshop 2017
AIMS-South Africa
Dr Franck KM

1. Basic network properties

Define the following quantities and calculate them for the graph $G = (V, E)$.

(a) The adjacency matrix A.

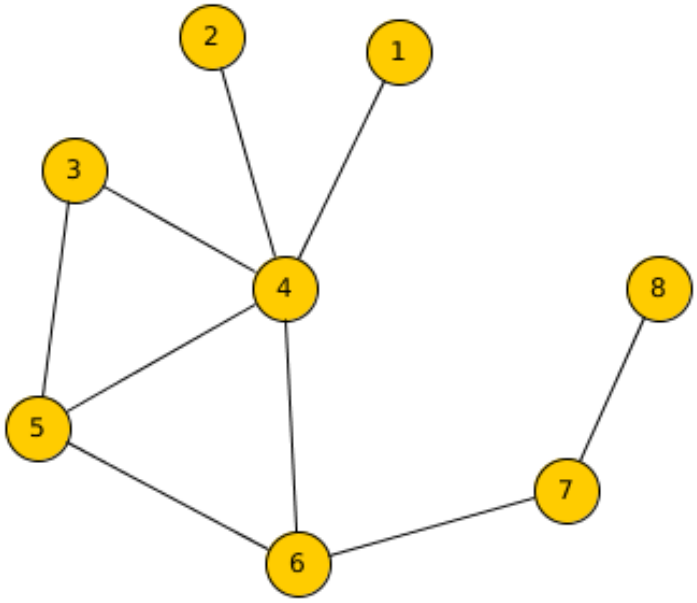
(b) The density D of the graph.

(c) The degree k_i of each node $i \in V$ and the degree distribution $P(k)$. Hint: When a node is picked by random, its probability to have a specified degree k is distributed as $P(k)$.

(d) The mean degree k of the graph.

(e) The diameter d of the graph.

(f) The clustering coefficient C_i for each node $i \in V$ that has degree $k_i > 1$, and the average clustering coefficient (averaged over all nodes). For nodes with $k_i = 0, 1$, we define $C_i = 0$.



2. Computing network properties with Python-NetworkX

In this exercise, you will get some hands-on experience of NetworkX by calculating some basic network properties. The dataset we use here is the Zachary karate club

network, is a famous example of a social network, where the club eventually splits into two fractions because of a dispute between two leaders. The dataset is zachary.net. To get you started, Check also the NetworkX online tutorial and index:

(a) Load the edge list and visualize the network. The split of the club should reflect into the shape of the visualized network. (you can also visualize by using Pajet or Gephi).

(b) Calculate the edge density of the Karate club network. Write first your own algorithm and then compare your result to the output of the corresponding NetworkX function.

(c) Calculate the average clustering coefficient with your own algorithm and compare it to the output of the corresponding NetworkX function.

(d) Calculate the degree distribution $P(k)$ and complementary cumulative degree dis- tribution $1 - CDF(k)$ of the network. Visualize the distributions using matplotlib.pyplot.
NOTE: $1 - CDF(k) : 1 - CDF(k)$ is defined as the probability that a randomly picked node has a degree larger than or equal to k.

(e) Calculate the average shortest path length l . Here, you don't need to write your own algorithm. It is sufficient to use the relevant networkx function.
Hint: Search the networkx documentation!

(f) Using matplotlib.pyplot library, make a scatter plot of C_i as a function of k_i.