Exercise set #1 (31 pts)

- The deadline for handing in your solutions is Nov 14th 2016 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.

In the following exercises, *network* = *undirected network* unless otherwise mentioned.

1. Basic network properties (6 pts, pen and paper)

Define the following quantities and **calculate** them for the graph $G = (V, E)$ in Figure 1.

a) (1 pt) The adjacency matrix A.
b) (1 pt) The edge density ρ of the graph.
c) (1 pt) The degree k_i of each node $i \in V$ and the degree distribution $P(k)$.
d) (1 pt) The mean degree $\langle k \rangle$ of the graph.
e) (1 pt) The diameter d of the graph.
f) (1 pt) 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 $C_i = 0$.

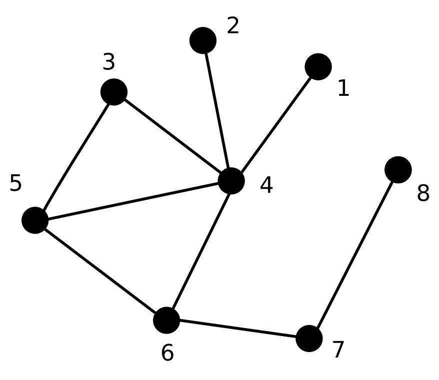

Figure 1: The graph for exercise 1.

2. Computing network properties programmatically (6 pts)

In this exercise, you will get some hands-on experience of NetworkX [1] by calculating some basic network properties. The dataset we use here, the Zachary karate club network [2], 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 edge list file (`karate_club_network_edge_file.edg`) can be found in the course MyCourses page. To get you started, you may use the accompanying Python template (`comp_net_props.py`) but the usage of the **template** is fully **optional**. Then you only need to fill in the required functions. In addition to returning a short report of your results (including the visualizations), return also your commented Python code.

Hint: Check also the NetworkX online tutorial and index:
`https://networkx.github.io/documentation/networkx-1.10/tutorial/index.html`
`https://networkx.github.io/documentation/networkx-1.10/reference/index.html`

a) (1 pts) Load the edge list and **visualize** the network. The split of the club should reflect into the shape of the visualized network.

b) (1 pt) Calculate the edge density of the Karate club network. First, **write your own algorithm** and then **compare** your result to the output of the corresponding NetworkX function.

c) (1 pts) **Calculate** the average clustering coefficient **with your own algorithm and compare** it to the output of the corresponding NetworkX function.

d) (1 pts) **Calculate** the degree distribution $P(k)$ and complementary cumulative degree distribution 1-CDF(k) of the network. **Visualize** the distributions using `matplotlib.pyplot`[1]. **NOTE:** In this course, we use a slightly non-standard definition of 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) (1 pts) **Calculate** the average shortest path length $\langle l \rangle$. Here, you don't need to write your own algorithm. It is sufficient to use the relevant `networkx` function.

f) (1 pts) Using `matplotlib.pyplot` library, **create** a scatter plot of C_i as a function of k_i.

3. Path lengths in simple model networks (8 pts, pen and paper)

In this exercise, we will gain more intuition on what are considered 'short' and 'long' paths in complex networks. The diameter of a network, d, is defined as the length of the longest shortest path between two nodes in the network. By studying the behavior of d when the number of nodes N in the network increases, we can tell how efficiently the network is wired. A commonly known example of a slowly increasing d is the concept of the 'six degrees of separation': in social networks, even in worldwide ones, every two individuals are on average connected through a very short chain of *e.g.* six intermediate contacts [3]. So, when selecting a model for a social network, we should pay attention on the behavior of $d(N)$.

Your task is now to **derive the network diameter d as a function of network size N** for three model networks. These model networks are shown in Fig. 2, but note that you should derive the network diameter for a general N, *i.e.* **do not assume N to have any specific value**.

a) (1 pts) Ring lattice:
You may assume that N is an odd number

[1] Check tutorial at `http://matplotlib.org/users/pyplot_tutorial.html`.

b) (2 pts) Two-dimensional *square* lattice:
Assume that $N = L^2$, where L is a positive integer.

c) (3 pts) Cayley tree:
Cayley tree[2] is a symmetric regular tree, in which each vertex is connected to the same number k of others, until we get out to the leaf nodes. As shown in Fig. 2, the tree is then organized into l layers, where the central node is on the zeroth layer and the leaf-nodes correspond to layer number l. In this exercise, assume that $k = 3$.

Hint: Assume first that your finite Cayley tree has l layers. Given l, calculate then N and d. The formula for the geometric sum may be useful:

$$\sum_{i=0}^{n} r^i = 1 + r + r^2 + \cdots r^n = \frac{r^{n+1} - 1}{r - 1}$$

d) (2 pts) If N is increased, which network's diameter grows fastest? And slowest? A network is said to be 'small-world' if the network diameter grows slower or as slow as $\log N$. Which of these networks fulfill the 'small-world' property?

Hint: Define, how d depends on N if you omit all scalar multipliers and constants and check, for which network d increases fastest and slowest as a function of N.

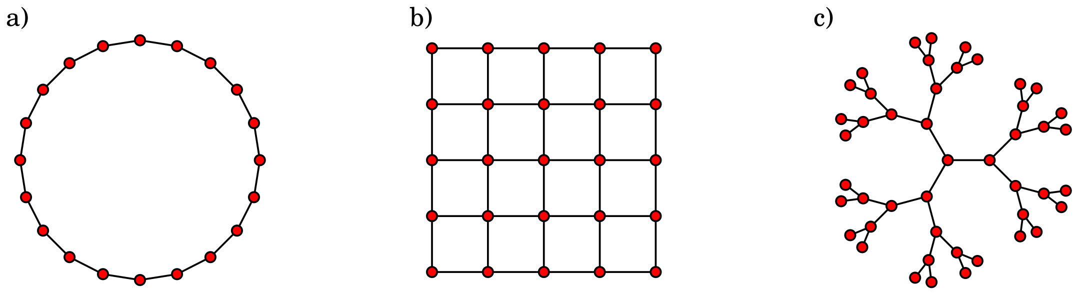

Figure 2: Examples of the model networks. **a)** A simple ring lattice with $N = 20$. **b)** A simple 2D-lattice with $k = 4$, $L = 5$, ($N = 25$). **c)** A Cayley tree with $k = 3$, and $l = 4$ ($N = 46$)

4. Counting number of walks using the adjacency matrix (5 pts, pen and paper)

A lot of network properties can be computed from its adjacency matrix and its spectrum. In this exercise, we investigate the relationship between the powers of the adjacency matrix and number of walks between pairs of nodes.

a) (1 pt) **Draw** the *induced subgraph* G^* that is induced by vertices $V^* = \{1 \ldots 4\}$ of network visualized in Figure 1. **Calculate by hand** the number of walks of length two between

[2] Also known as Bethe lattice, see `http://en.wikipedia.org/wiki/Bethe_lattice`

all node pairs (i, j), $i, j \in \{1, ..., 4\}$ in G^*. The length of a walk is defined as the number of links travelled to get from i to j; a link can be travelled in both directions and the walk can visit a node multiple times. Then, compute the matrix A^2 (you may do this also using a computer), where A is the adjacency matrix of the network G^*. **Compare your results**; what do you notice?

b) (1 pt) Compute the number of walks of length three from node 3 to node 4 in G^*. Then, starting from matrices A^2 and A, **compute by hand** the value of $A^3_{3,4}$ showing also the **intermediate steps**.

c) (3 pts) Now, let's consider a general network with adjacency matrix A. **Show** that the element $(A^m)_{ij}$, $m \in \mathbb{N}$ corresponds to the number of walks of length m between nodes i and j.

Hint: Make use of mathematical induction: Show first that the statement holds for $m = 1$, by analyzing the elements of the matrix A^1. Next, assume that the statement holds for a general m and prove that it holds also for $m + 1$. To do that, consider the element $a^{(m+1)}_{i,j}$ assuming that $a^{(m)}_{i,j}$ gives the number of walks of length m.

5. Bipartite networks (3 pts, pen and paper)

Consider the bipartite network of actors and movies shown in Figure 3.

a) (1 pt) **Construct** the two unipartite projections of the network – the network of actors and the network of movies. In the former, actors are linked if they have acted together in at least one movie, and in the latter, movies are linked if there is at least one common actor.

b) (2 pts) Show that, in general, it is not possible to uniquely reconstruct a bipartite network from its two unipartite projections. **Prove this by providing a counterexample**: Take the same 4 actors and 4 movies, and design a different bipartite network that has exactly the same unipartite projections. That is, connect the actors and movies in some way that results in the same projections as in a).

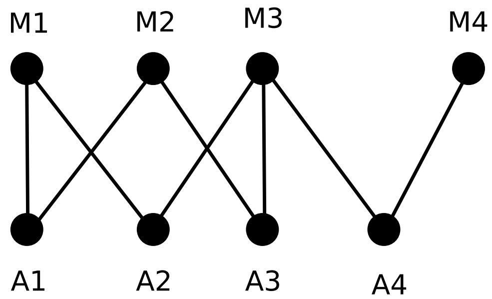

Figure 3: A bipartite network of actors (A) and movies (M).

6. Ensemble averages by enumeration (3 pts, pen and paper)

Graph *ensembles* are distributions of graphs, where each graph G_i has a certain probability π_i. The *ensemble average* of a quantity X is defined as $\langle X \rangle = \sum_i \pi_i X(G_i)$. Let us define following quantities: $k(G)$ is the average degree of the graph G, $C(G)$ is the average clustering coefficient for graph G (assuming that the clustering coefficient gets value 0 for nodes of degree 0 and 1), and $d^*(G)$ is the diameter of the connected component of the graph G that has the largest diameter.

a) (2 pt) **Calculate**, using pen and paper, $\langle k \rangle$, $\langle C \rangle$, and $\langle d^* \rangle$ for $G(N = 3, p = 1/3)$.

b) (1 pt) **Calculate**, using pen and paper, the formulas for $\langle k \rangle$, $\langle C \rangle$, and $\langle d^* \rangle$ for $G(N = 3, p)$ (that is, for $N = 3$ and all values of p). Remember to simplify the formulas you get as results.

Note that there are in general $2^{N(N-1)/2}$ possible networks with N nodes! That is, for $N = 3, 4, 5, 6, \ldots$ there are $8, 64, 1024, 32768, \ldots$ possible networks, and for $N = 100$ the number of possible networks is in the order of 10^{1490}. The naive method used in this exercise for calculating ensemble averages directly from the definition quickly becomes unpractical when the number of nodes increases. In next the exercise set you will use a more practical method for estimating ensemble averages.

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/Lz9dw8iJ2iUYZo0E2`.

References

[1] NetworkX, `https://networkx.github.io/`

[2] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research 33, 452-473 (1977)

[3] J. Travers, S. Milgram, An experimental study of the small world problem, Sociometry 32, 425-443 (1969).