# CS7280: Network Science, Fall 2017

Homework-2

October 1, 2017

**Due Date: TUE October 10, 1pm**

## 1 Problem 1 (25 points)

A Cayley tree is a symmetric tree, constructed starting from a central node of degree $k$. Each node at distance $d$ from the central node has degree $k$, until we reach the nodes at distance $P$ that have degree one and are called leaves.

1. Calculate the number of nodes reachable in $t$ steps from the central node.
2. Calculate the degree distribution of the network.
3. Calculate the diameter $d_{max}$.
4. Find an expression for the diameter $d_{max}$ in terms of the number of nodes $N$.
5. Does the network display the small-world property?

## 2 Problem 2 (25 points)

Before you try to solve the following problem, please read Section 5.13 from A.L.Barabasi's textbook to understand the *rate-equation approach.*

After you do so, consider a variation of the preferential attachment model, where at each time step a new node arrives and connects with a directed edge to a node chosen with probability:

$$\Pi(k_i^{in}) = \frac{k_i^{in} + A}{\sum_j (k_j^{in} + A)} \tag{1}$$

Here, $k_i^{in}$ indicates the in-degree of node $i$ and $A$ is the same constant for all nodes. Each new node has m directed edges.

Calculate, using the rate equation approach, the in- and out-degree distribution of the resulting network.

## 3 Problem-3 (50 points)

1 Write a function (ideally in Python – but you can also use any other language) that generates an *undirected growing network based on the K Nearest-Neighbors (KNN)* rule, as follows:

- Start with a clique of $n_0$ nodes that are randomly located in the Cartesian square $S = [0, 1] \times [0, 1]$.
- Grow the network by adding one node in each iteration. The new node must be located at a randomly chosen location within $S$, and it should be connected to its $K$ nearest neighbors.

Use $G = KNN\_model(n_0, K, N)$ as the interface of your function, where $N$ is the number of iterations and $G$ is the final network.

2 Using your $KNN\_model$ function, construct networks with $N = 10000$, $n_0 = 4$ and $K = 4$. Does this model generate scale-free networks? Does it generate small-world networks? Justify your answer with the appropriate graphs.

3 Tweak the previous KNN model such that for every new node, half of its edges are placed based on the nearest-neighbors rule, while the other half are placed based on the random "link-selection" model, described in class and in Section 5.9.2 of Barabasi's book. How does the resulting degree distribution of this model relate to the degree distribution of the original model? (include relevant graphs)

**Hint:** You can find nearest neighbors using the *sklearn* python library.