Networks and their Structure Assignment

Network Science Topics 2 and 4

Note that the networks in this exercise areundirected.

Recall the Watts-Strogatz network model from Topic 2. Here we define a similar model which we call the **Variable Degree Watts-Strogatz (VDWS) model**. This is how a **VDWS-network** is constructed.

- Let n and m be positive integers. Let p be a real number between 0 and 1. Create a set of n vertices labelled 0 to $n-1$. We think of these as being arranged on a circle so if two adjacent vertices are close to each other on the circle, it is clear what is meant if we say that one is the clockwise neighbour of the other. For example 4 is a clockwise neighbour of 2.
- For each v, let $\ell(v)$ be the local degree of v. Choose the values $\ell(v)$ randomly such that they have a zero-truncated Poisson distribution with parameter m (see below).
- Join each vertex v to vertices $v-\ell(v), v-\ell(v)+1, \ldots, v-1, v+1, \ldots v+\ell(v)-1, v+\ell(v)$ where addition is $\mod n$. Thus each vertex will have degree at least $2\ell(v)$, but it might be more. For example, if 3 has local degree 2, then it will be adjacent to 1, 2, 4 and 5. But if 10 has local degree 7, there will also be an edge between 3 and 10.
- For each vertex v, for each edge from v to a clockwise neighbour w: with probability p, the edge from v to w is deleted and replaced by an edge from v to a vertex x chosen uniformly at random from all the vertices in the network. We call this process *rewiring*. If $x=v$ or x is already a neighbour of v, then we do nothing and the edge from v to w is kept. (Note that in this way, each of the edges created in the previous step should be considered exactly once for rewiring, and that edges that are created by rewiring should not be later rewired themselves.)

You will need to write code to create VDWS-networks. You can use the code for WS-networks (see Topic 4 on Learn Ultra) as a starting point. To create values that are sampled from a zero-truncated Poisson distribution with parameter $m=10$, say, you can use the following code.

```
import numpy as np
local_degrees = np.random.poisson(10, 100)
local_degrees = local_degrees[local_degrees > 0]
```

Note that the number of values created cannot be predicted precisely since the second line samples 100 values from a Poisson distribution and then the third line removes the zeros (although zeros are very unlikely for the value of m we will use).

1. [20 marks] Consider the epidemic model with vaccinations from the Topic 4 lecture notes with states S, I, V, VI and R. Using this model, simulate the spread of disease on a VDWS-network with $n=200000$, $m=25$ and $p=0.01$. Assume that initially 5 randomly chosen vertices are in I and every other node is in S. From $t=50$, at each time step move 400 randomly chosen vertices from S to V (until S is empty). Let $t_I=2$ and consider various values for the other parameters. Initially let $p(\mathrm{I},\mathrm{S})=p(\mathrm{I},\mathrm{V})=p(\mathrm{VI},\mathrm{S})=p(\mathrm{VI},\mathrm{V})=0.01$ (implying that the vaccination is ineffective). Then consider cases where $p(\mathrm{I},\mathrm{S})=0.01$ but the other probabilities are lower, modelling the cases where the vaccination protects against infection, transmission or both. In each case, create plots that show how the number of vertices in each of the five states varies over time. Comment on your findings.
2. [25 marks] Repeat the simulations of the previous question, but with the single change that the 400 vertices moved from S to V at each time step (after $t=50$) can be strategically chosen (rather than being chosen at random). Propose and test three different strategies and comment on their effectiveness.

With the values given, each simulation should require no more than 400 time steps (before the infection dies out). You should be able to run this in a few minutes on your own computer (or any university PC). If you find that with the values I have given, the simulations take too long, then choose alternative values and document this in your submission.