

## 1 Erdős-Rényi-Graphs

In the  $G(n, p)$  Erdős-Rényi-model, a graph with  $n$  edges is constructed by connecting nodes randomly. Each edge is included in the graph with probability  $p$  independent from every other edge.

Check the statement that almost every graph in  $G(n, 2 \ln(n)/n)$  is connected. Meaning that as  $n$  tends to infinity, the probability of a graph on  $n$  vertices with edge probability  $2 \ln(n)/n$  being connected tends to one.

You can proceed as follows:

- a) Up to a moderately high number of  $n$  ( $n = 100$  should be enough, why not more nodes?) create a sufficiently high number of Erdős-Rényi-graphs (around 1000 should be ok).
- b) Investigate how many connected components each of these graphs has and average over all graphs for a given  $n$ .
- c) Plot the average number of connected components over  $n$ . Does the behaviour of the number of connected components support the above statement? How long does it take to create an Erdős-Rényi-Graph of size  $n$ ?

## 2 Random Walks on Graphs

- a) Create a power law cluster graph with a number of nodes between  $n = 50$  and  $n = 100$ . Save the graph. Make sure that for the following you always use the same graph. Why is this important?
- b) What values for  $m$  and  $p$  yield a graph containing a number of clusters below 10 which are sufficiently clearly distinguishable? Visualize the graph in a way the clusters and node labels are clearly recognizable.
- c) Select two nodes in the graph as starting point for two random walkers. Let the walkers walk across the graph (at each step they choose randomly between the neighbors of their current node) until they meet.
- d) How does the time until the walkers meet depend on whether their starting point is in the same cluster or not? How many trials would you need to make a statistically significant statement?
- e) Can you think of a systematic way to identify clusters and serialize the above test? Implement it!

## 3 Zombie Apocalypse

HINT: for the following you can either use an object oriented or functional approach to make your life easier. It is also possible to handle zombiism via node attributes in the graph although it might be a bit less intuitive.

- a) Create a two-dimensional grid graph with around  $100 \times 100$  nodes. Every node holds a healthy human which has regular contact with its neighbors.
- b) Start the apocalypse by infecting nodes either

- i. at one, two or  $n$  random locations
  - ii. at the border of the grid
- c) At each step the infected humans (zombies) have a probability  $p$  of biting each of their neighbors and turning them into zombies. Implement a method to visualize healthy and infected humans in the graph and to track the spreading of the infection.
- d) Investigate how long it takes for the whole population to get infected and how this depends on the infection probability  $p$  and the number and position of initial zombies. Visualize how the virus spreads throughout the population.
- e\*) At each step, one random zombie is intelligent enough to board an airplane. This Zombie tries to find and infect a human at a location no less than 5 edges away from its initial position. How does aviation influence the apocalypse?
- f\*\*) Someone has developed a cure for zombiism. Once infected it takes a zombie  $m$  turns to become human again. Can you find a balance between  $m$  and  $p$  such that the cure keeps the virus in check and at every time there are only a constant number of persons infected?  
What if the zombies don't cure themselves but there is a third species of humans - the doctors - moving through the grid at random curing all zombies around them? How many doctors do we need to prevent the apocalypse?
- g) What do you think would be the most efficient way to prevent the spreading of the infection throughout the whole population?

## 4 The Slime Mold

A slime mold is an unicellular organism. It starts out from a single spot (we call it the "core") and grows out tendrils to investigate its neighborhood. Whenever a tendril finds a food source, it attaches to the source and funnels nutrients to the core until the source is depleted. Our goal is to simulate the growing of a slime mold and program our mold in a way that it survives for the longest possible time.

- a) Create a network starting from one node that grows tendrils randomly in all directions creating a web with a maximal radius  $r = 10$ . Grow the network sequentially, at each timestep each edge can only grow a maximum distance of  $r/10$  in length. Visualize the network.
- b) Create the surroundings of the mold by putting food sources randomly in the region between  $r/2$  and  $r$  away from the core. Assign a random number of food units  $u \in [1, 10]$  to each food source. Implement a functionality that keeps track of the value of  $u$  in each food source. Control the rate  $k$  of new food units per timestep so that  $k \approx 3$ .
- c) Implement a function that makes a tendril recognize the source when it is closer to it than  $r/10$ . Each tendril can transport a maximum of  $c = 2$  food units to the core each timestep.

- d) The core needs one food unit each timestep. To support its tendrils, the core needs an additional unit of food equal to the total length of the network divided by  $r$  (round to the nearest integer).
- e) Implement a functionality that lets edges decay (vanish) if they have reached the perimeter and not found any new food or if their food source is depleted.
- f) Let the core start with an initial supply of  $u_i = 10$ . The mold dies if it has no more food left at the end of a timestep. Try to answer some of the following questions:
  - i. What is the likelihood of your mold surviving the first 10 timesteps? The first 100?
  - ii. Does it get easier or more difficult for the mold to survive with increasing time?
  - iii. What influence do the parameters  $r$ ,  $u_i$ ,  $k$  and  $c$  have on the survival of the mold?
  - iv. What are weaknesses of the model? Can you think of improvements to the model to make it more realistic?