Lab assignment 2

Social network analysis and modelling

Introduction

In these exercises, you will analyse various social networks and model the diffusion process in a network using the `igraph` package in `RStudio`. The `igraph` package in `R` provides built-in functions to build and measure networks.

The package `igraph` is a library collection for creating and manipulating graphs and analyzing networks. It is written in `C` and also exists as a Python package. `igraph` is capable of handling large networks efficiently.

We follow the principles of *remote group work: A personal, interactive process* as during the first four labs.

In the remainder of this lab assignment, explanations are formatted as plain text, instructions are underlined, questions to answer are labelled and RStudio commands are written written as `code highlighted text`. The number of points available for each question gives an idea of the expected depth of your answers.

To use the required functions during the lab assignment, first install the required packages:

```r
install.packages("igraph")
install.packages("network")
install.packages("intergraph")
install.packages("RColorBrewer")
```

Then, call the libraries of these packages so that you can use all functionalities within the current session.

```r
library(igraph)
library(network)
library(intergraph)
library(RColorBrewer)
```

Exercise 1 Build and analyse a small network from Facebook

In this exercise, we will study a small personal network from Facebook. The dataset comes from Douglas Lukes' UserNetR package, which records the Facebook network of Douglas and the mutual friendships of his friends.

Download the `Facebook_att.csv` and `Facebook_edge.csv` from BlackBoard. Import the `.csv` data into `R` and build a network.

```
nodes <- read.csv("Facebook_att.csv", header = TRUE)
links <- read.csv("Facebook_edge.csv", header = TRUE)
facebook <- graph_from_data_frame(d = links, vertices = nodes$NodeID, directe
d = FALSE)
facebook_net <- asNetwork(facebook)
```

The network is built, but is kind of empty, since, as you might have noticed, we haven't imported the attributes of vertexes. The `igraph` and `network` packages in `R` provide many ways to add the attributes of vertexes. Here, we provide only one of the solutions.

Copy the code below and introduces the attributes of vertexes in the facebook_net.

```
# Load in other attributes.
facebook_net %v% "vertex.names"        <- nodes$vertex.names
facebook_net %v% "sex"                 <- nodes$sex
facebook_net %v% "friend_count"        <- nodes$friend_count
facebook_net %v% "group"               <- nodes$group
facebook_net %v% "mutual_friend_count" <- nodes$mutual_friend_count
facebook_net %v% "na"                  <- nodes$na
facebook_net %v% "relationship_status" <- nodes$relationship_status
```

The summary function provides you an overview of the network that you have built. View the summary of the network via the code below.

```
summary(facebook_net)
```

Use the plot function to have a first scan of the network.

```
plot(facebook_net, vertex.cex = 1.2, main = "Basic plot of Douglas's Facebook
 friends")
```

Do questions **1-4** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

Check out the Summary and Plot, how many friends do Douglas have on Facebook? Is this a directed or undirected graph and why? What is the meaning of the link between nodes in the plot? **[Question 1, 2 points].**

Measure node-level metrics

Douglas seems to have quite some friends on Facebook. First, let's have an investigation on Douglas's friend on individual level (i.e., node level), especially their roles in this social network. Say, we are interested in the network attributes of Friend number 1.

```
degree(facebook, v = 1, mode = "total")
```

```
closeness(facebook, v = 1, normalized = TRUE)

betweenness(facebook, v = 1, directed = FALSE, normalized = TRUE)

eigen_centrality(facebook)$vector[1]

transitivity(facebook, type = "localundirected", vids = 1)
```

Note that when calculating the closeness centrality, you get a warning message, due to the fact that the graph is disconnected. Specifically, this means that there are nodes that cannot be reached from specific other nodes. Therefore, the distance between these disconnected nodes is infinite. `igraph` overcomes this problem by taking the total number of nodes as a maximum distance rather than the actual distance (which is infinite).

To make sense of the above values, you not only need to understand what they mean, but also their comparative values to the other nodes in the same network. This can be realized by slightly tweaking the above functions.

```
deg <- degree(facebook, mode = "all")
cls <- closeness(facebook, normalized = TRUE)
btw <- betweenness(facebook, directed = FALSE, normalized = TRUE)
eig <- eigen_centrality(facebook)
lcl <- transitivity(facebook, type = "local")
```

Compare the degree, closeness and betweenness of Vertex 1 to the values of other nodes in the network. How will you evaluate the role of Vertex 1 in this network? **[Question 2, 3 points].**

In the lecture, we introduced a few measures of centrality: degree, betweenness, eigenvector. Try to find top 5 nodes according to a) degree, b) betweenness, c) closeness and d) eigenvector. And develop the scatter plot between different metrics, you can refer to the code below. Discuss with your group and describe to your teacher, 1) how well does the top 5 nodes by different metrics overlap with each other? 2) why we need more than one metric to define centrality? **[Question 3, 5 points]**

```
par(mfrow = c(2,2))
plot(deg, cls, main = "Degree versus closeness",
     xlab = "Degree", ylab = "Closeness")
plot(deg, btw, main = "Degree versus betweenness",
     xlab = "Degree", ylab = "Betweenness")
plot(deg, eig$vector, main = "Degree versus eigenvector",
     xlab = "Degree", ylab = "Eigenvector")
plot(deg, lcl, main = "Degree versus local clustering",
     xlab = "Degree", ylab = "Local clustering")
```

Measure group-level metrics

So far, we have looked exclusively at node level metrics. When we are interested in properties of the network, we could calculate several group-level metrics. For example, the degree distribution gives the relative frequencies of the node-level degrees. That is, it measures how often each degree-value occurs in the data. Additionally, we calculate the density of the network, the diameter, the global clustering and the centralization.

```
deg_dist <- degree_distribution(facebook)
barplot(deg_dist)

edge_density(facebook)

diameter(facebook, directed = FALSE)

transitivity(facebook, type = "global")

centr_degree(facebook)$centralization
```

Discuss within your group on how you understand each of these measures. And describe to your teacher, 1) why diameter should be larger than 1, and other metrics such as edge density and transitivity are smaller than 1? 2) is this a tightly knitted network? **[Question 4, 3 point].**

Detect the component and community

Some of the group-level metrics provides us some estimations of the network structure. Although these quantitative measures are insightful, we might need to know more about the mesoscale structure of this network. For example, we already encountered problems in calculating the closeness centrality in the previous exercise. This is because this network has multiple components. From the basic plot of the network above, you can count the number of components manually. An easier way however, is to use the `components` function in R, that especially comes in handy when you are dealing with a larger network.

Calculate the number of components and their sizes in the network.

```
components(facebook)
```

Do questions **5-10** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

Reflect on what we have discussed about the Facebook network on Slide 42 of the lecture. Do you think this small network of Douglas resonates some general patterns of the entire Facebook network in terms of the components size and number? How can you explain such an observation **[Question 5, 3 point].**

To better understand the components of this network, we can check the node attributes again. Check out the attributes table of variable "nodes", where you can see that the 93 friends of Douglas belong to 8 groups: BookClub, College, Family, GraduateSchool, HighSchool, Music, Spiel, Work. They explain the social context on why they are connected to Douglas.

First thing we want to test here is whether or not that friends in the same components are those of the same group. This can be visualized by simply colouring the vertex according to their group.

```
group <- as.factor(get.vertex.attribute(facebook_net, attrname = 'group'))

pal <- brewer.pal(nlevels(group), "Set1")

plot(facebook_net, vertex.col = pal[group], vertex.cex = 1.5,
     main = "Plot of Facebook Data colored by friend type")
legend(x = "bottomleft", legend = levels(group), col = pal,
       pch = 19, pt.cex = 1.2, bty = "n", title = "Friend type")
```

Based upon the plot you produced, discuss with your group and describe to your teacher the distinctions of the components. Do you find instances of intermingling of Douglas friends (i.e., belong to different groups but end up in the same component)? Do you find any isolated groups here? What can you conclude about the mixing of Douglas' Facebook friends? **[Question 6, 3 point].**

Using the above code as a reference, check out the attributes of other factors (e.g., sex, relationship_status) in terms of people in the same components. Note that you can specify the 'attrname' parameter within the function 'get.vertex.attribute'. Discuss with your group and describe to your teacher whether or not these factors are the keys in determining the formation of components **[Question 7, 3 point].**

Since we now see the groups of Douglas friends, we can also study the characteristics of these subgroups. Recall the fact that the density of this network is relatively low (0.08), but what about the densities of each of the subgroups.

Measure the density of each group of Douglas friends using the following code:

```r
sapply(levels(group), function(x) {
  y <- get.inducedSubgraph(facebook_net,
                           which(facebook_net %v% "group" == x))
  paste0("Density for ", x, " friends is ",
         edge_density(asIgraph(y)))
})
```

From this analysis, what do you observe from the density values? Are they similar across different groups? What is the minimum and maximum value you observed here and how do you explain that? **[Question 8, 3 point].**

So far, we analysed the subgroup structure only according to the original friend groups of Douglas. However, is this enough? Is the place where you know this person (e.g., workplace, book club) sufficient to explain the structure of your social network?

To answer this question, search and discuss within your group on the theory of 'community detection'. Describe to your teacher what community detection is, and why it is useful to understand complex networks **[Question 9, 3 point].**

As you might realize after your search, there are plenty of community detection algorithms out there. Larger social media such as Facebook, Twitter, Instagram and WeChat have their own algorithm to identify the communities under their huge user groups. Here in `R`, there are some built-in functions where you can have a trail on the community detection techniques. These functions help us to visualize and explore the structure of a complex network.

Try the following community detection algorithms and save the output for further discussion.

```r
cw <- cluster_walktrap(facebook)
plot(cw, facebook, vertex.label = V(facebook)$group,
     main = "Walktrap")

ceb <- cluster_edge_betweenness(facebook)
plot(ceb, facebook, vertex.label = V(facebook)$group,
     main = "Edge Betweenness")

cfg <- cluster_fast_greedy(facebook)
plot(cfg, facebook, vertex.label = V(facebook)$group,
     main = "Fast Greedy")

clp <- cluster_label_prop(facebook)
plot(clp, facebook, vertex.label = V(facebook)$group,
     main = "Label Prop")
```

```
cle <- cluster_leading_eigen(facebook)
plot(cle, facebook, vertex.label = V(facebook)$group,
     main = "Leading Eigen")
```

Compare the plots that you generate from the different algorithms; do you find them similar and why or why not? **[Question 10, 2 point].**

Further reading

Do questions **11-12** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

Read the following article:

Ana Lucía Schmidt, Fabiana Zollo, Antonio Scala, Cornelia Betsch, Walter Quattrociocchi. Polarization of the vaccination debate on Facebook. *Vaccine* 36, 3606-3612, 2018.

Discuss with your group and describe to your teacher: What is the societal problem that the authors are studying here? How can this problem be addressed using social network analysis? **[Question 11, 3 point].**

Discuss with your group and describe to your teacher: How many communities did the authors detect and how did they do this? Additionally, reflect on which of the communities you expect has the highest density, and which of the communities do you think has the lowest density. **[Question 12, 3 point].**

Exercise 2: Formulate a social network for certain architectures

Exercise 1 illustrates the situation when we have complete information of the network, i.e., we know who is connected to whom. In other cases, however, we might want to work on a synthetic network with special features.

Do questions **13-16** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

Discuss with your group, then describe to your teacher, under which circumstances, we might need to work on a synthetic network. **[Question 13, 2 point].**

Network relationships come in many shapes and sizes, and so there is no single model which encompasses them all. But over time, people do summarize some common paradigm that can be used to build a synthetic network. In the lecture, we mentioned three major architectures for synthetic network, which is Erdos-Renyi Random Graph Model, Small-world Random Graph Model, and Barabasi-Albert (BA) model. All these models have built-in functions in R.

Assuming you know how many nodes are in your network, as well as the probability that any two of them are connected (i.i.d and random), you can generate an E-R random graph using *sample_gnp()* or *sample_gnm()*.

In `sample_gnp`*(n, p, directed = FALSE, loops = FALSE),* the graph has 'n' vertices and for each edge the probability that it is present in the graph is 'p'.

In `sample_gnm`*(n, m, directed = FALSE, loops = FALSE)*, the graph has 'n' vertices and 'm' edges, and the 'm' edges are chosen uniformly randomly from the set of all possible edges. This set includes loop edges as well if the loops parameter is TRUE.

Build an ER-random graph given the number of vertices and probability:

```
ER <- sample_gnp(100, 1/100)

plot(ER, vertex.label= NA, edge.arrow.size=0.02,vertex.size = 0.5, xlab = "ER
 Random Network: G(N,p) model")
```

Plot your network (with n=100, and p=1/100) and compare with those with your group members. Are they identical? Explain why they are/aren't **[Question 14, 1 point].**

Develop three networks with the same number of vertices (n), but different probability (p); Name them as ER1, ER2, and ER3. Develop the plots of ER1, ER2 and ER3, describe how these three graphs look differently as p increase and explain why. **[Question 15, 2 point].**

If p<1, for n great enough, what happens to the clustering coefficient of an ER random graph and why? (You can use the 'transitivity' function to test your guess). Discuss with your group and describe the answer to your teacher. **[Question 16, 2 point].**

Next, let's move on to the small world model (Watts and Strotgatz model). It assumes that you know a certain number of persons (k) and that you are more likely to know your closest neighbors. The algorithm though more complicated than the Erdős-Rényi model's.

We have 3 parameters. The number of the population (N), the number of close neighbors (k) and a rewiring probability p.

Because this model generates some conglomerates of people knowing each other, it is really easy to be linked indirectly (and with a very few number of steps) with anyone in the map. This is why we call this kind of model a small world model. This is, in the three we describe here the closest from the realistic social network of friendship.

The small-world model is built by introducing a rewiring probability to a regular network such as the regular lattice. Run the following code and see how the rewiring probability can change the network:

```
Regular<-watts.strogatz.game(dim=1,size=300,nei=6, p=0)

plot(Regular, layout=layout.circle, vertex.label=NA, vertex.size=5, main= "Network with zero rewiring probability ")

SW1<-watts.strogatz.game(dim=1,size=300,nei=6, p=0.001)

plot(SW1, layout=layout.circle, vertex.label=NA, vertex.size=5, main= "Network with 0.001 rewiring probability ")

SW2<-watts.strogatz.game(dim=1,size=300,nei=6, p=0.01)

plot(SW2, layout=layout.circle, vertex.label=NA, vertex.size=5, main= "Network with 0.01 rewiring probability ")

SW3<-watts.strogatz.game(dim=1,size=300,nei=6, p=0.1)

plot(SW3, layout=layout.circle, vertex.label=NA, vertex.size=5, main= "Network with 0.1 rewiring probability ")
```

Do questions **17-20** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

For rewiring probability p=0.001, develop the networks for n from 20 to 500 and record the diameter of each network; Plot N~log (diameter); what do you find and how will you explain that? **[Question 17, 3 point].**

Check the clustering coefficient and average path length of the Regular, SW1, SW2 and SW3. Describe the trend of clustering coefficient and average path length as p increase. Does any of these graphs show the desirable attributes that you are looking for a small world network? **[Question 18, 3 point].**

You might realize not every value of p can return you a small-world network that you are looking for. Then a question arises as how can you find the range of p.

For the same setting, i.e., size =300, nei=6, what are the range of *p* you will suggest to build a small network and **why**? One solution you can consider is to refer to the Figure 2 in Watts and Strogatz (1998), which explains the properties of small-world network for the family of randomly rewired graphs. Reproduce Figure 2 in the current context (i.e., size=300, nei=6). Discuss with your group and show the answer to your teacher. **[Question 19, 5 point].**

As a follow-up question of Q19, do you find the p values of very large or relatively small? What are its implications? **[Question 20, 3 point].**

The third architecture is to generate scale-free graphs according to the Barabasi-Albert model. This model is computing with a recursive algorithm. Two parameters are needed, the initial number of nodes (n0) and the total number of node (N). At the beginning, every initial node (the n0 first nodes) knows the other ones, then, we create, one by one the other node. At the creation of a new node, this node is linked randomly to an already existing node. The probability that the new node is linked to a certain node is proportional to the number of edges this node already has. In other word, the more links you have, the more likely new nodes will be link to you.

This model is for any network respecting the idea of "rich get richer". The more friends one node has, the more likely the new nodes will be friend with him. This kind of model is relevant for internet network. For example, the more famous is the website, the more likely this website will be known by other websites.

You can easily generate a scale-free network for a given size:

```
g0 <- barabasi.game(100, power = 1, m = NULL, out.dist = NULL, out.seq = NULL
, out.pref = FALSE, zero.appeal = 1, directed = FALSE,algorithm ="psumtree",
start.graph = NULL)

plot(g0, vertex.label= NA, edge.arrow.size=0.02,vertex.size =5, main = "Scale
-free network model, power=1")
```

Do questions **21-23** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

What does the power in the above function mean? How can it govern the structure of the network (e.g., the formulation of hubs)? (Hint: Change the value of power from 0.05, 0.5, 1, 1.5; See how the plot change; if you still fail to see the difference, visualize the vertex size according to the edge number, you can consider the code below.) **[Question 21, 3 point].**

```r
g1 <- barabasi.game(100, power = 0.5, m = NULL, out.dist = NULL, out.seq = NULL, out.pref = FALSE, zero.appeal = 1, directed = FALSE,algorithm ="psumtree", start.graph = NULL)

g1Net<-asNetwork(g1)

VS = 3+ 0.5*degree(g1)

plot(g1, vertex.label= NA, edge.arrow.size=0.02,vertex.size =VS, main = "Scale-free network model, power=0.5")
```

Discuss with your groups, if you are maintaining a network with a power of 0.5 and 1.5, respectively, what will be your plans to build up resilience for random and targeted attack? **[Question 22, 3 point].**

The above theoretical models on network architectures are based upon assumptions on the clustering coefficient, path length and degree distribution. It is interesting to check how true these assumptions are comparing the social networks in the real world. Next, you will investigate the structure of a few real-world networks. We will use four datasets from *Stanford Network Analysis Project (SNAP)*.

The datasets we use are *the "Amazon product co-purchasing network and ground-truth communities"*, *"localation-based social network Brightkite"*, *Collaboration network of General Relativity and Quantum Cosmology*. Check the above links to have an idea what these networks are, what kind of human interactions are involved.

Download the rds data of these network from the BB, import the data to R and build the network. Check out their network attributes. Do you find these real networks show some attributes of the synthetic architectures we studied above (e.g., random ER graph, small-world, and scale-free network)? Show your teachers some numbers, plots and how you interpret the results. **[Question 23, 6 point].**
(Note: you will find that path length is computationally intensive as there are many nodes in the network. Luckily, the website of the data has provided that the diameter of the Amazon network is 44, Brightkite's is 16, and the Collaboration network's is 17. While not the same as the average shortest path, the diameters give some information about the connectedness of the network.)

Exercise 3: Build the social network of this class and simulate the contagion process

This exercise will build a social network of this class then simulate how contagion can spread through the network. We will show you the complete work flow from collecting network data, processing, building contagion model and suggestions based upon model results.

Before this tutorial, we have designed a small survey to collect the network data and parameters that we need to model contagion. You can check out the survey via *https://forms.office.com/Pages/ResponsePage.aspx?id=oFgn10akD06gqkv5WkoQ52b4ptiQO5JFic6iamiXpGJUMzRFTUNCVlhDUFg0VEg4STE2TDFMSzVDMi4u*

The first question is to deal with the data privacy and consensus from interviewee, which data scientists should always bear in mind nowadays. Under General Data Protection Regulation (GDPR), we need to follow a strict protocol in collecting and analyzing any personal data. Standard procedures include but not limited to acknowledgement on how the data will be used, asking for consensus, anonymized the data as soon as possible to eliminate any chance of re-identification. This is also true when you are trying to collect and analyse online data from Facebook, Twitter and others social medias. There is an interesting reading of *"But the data is already public": on the ethics of research in Facebook*. It is based on a controversial case that IDs of the Facebook friendship data in one research were reidentified only 7 days after the dataset was made public.

The second and third questions are on network data, which we ask names of people that you know/ talk frequently in this class. These are quite traditional ways to collect network data, but not necessarily good and feasible ones. This is because such questions ask for too much personal details such as names. And it will become cumbersome when one has too many friends to name. Indeed, we received a few responses saying that "too many"! (Looks like some of your classmates are very well connected!) . Other alternatives include asking only the size of network (e.g., how many friends you have), or providing a map of house numbers for neighbors to point out whom they know.

Do questions **24-26** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

The fourth and fifth questions are designed to collect the parameters that we will need to build 1) an independent cascade (IC) model and 2) a threshold model. Can you see which question is for the IC model and which one is for the threshold model? **[Question 24, 1 point].**

Download the file “Class network survey.xlsx” from BB, check out the response of Q4 and Q5. Among the three types of behaviors, which one is the least contagious? Which one is the most contagious? And why? **[Question 25, 2 point].**

In social network science, we sometimes see the emergence of behavior as a product of network structure. To the context of this small survey, it would mean that your likelihood to share or your threshold to be influenced by others is relevant to your position in this network. For example, someone at the central position at this network might be more open-minded to accept new changes. If you want to draw robust conclusion, you should first build the network, analyse the node-level attributes in terms of different centralities and see if there is a correlation between the node-level attributes and their response. For simple illustration here, you can just check the degree of this network, i.e., the number of friends and close friends and their response to Q4 and Q5 in “Class network survey.xlsx”. Do you think network size can explain the differences in their likelihood to share and their thresholds to adopt? And why their answers are dependent/ independent on network size here? **[Question 26, 2 point].**

Another thing that normally you can test is the phenomena of “birds of the same feather flock together” in human’s social network. That is, for people who are mutual friends in the network, their will have similar likelihood to share and similar threshold to adopt. But given the small sample size (even smaller than 24 since not every friend of the respondents have filled in the survey), we are not testing this phenomenon here.

In total, we have 24 responses by 3 March 2021, out of the 60 students of this class. Response rate is 40%. This response rate is indeed quite high. And thanks to you who have filled in the survey! In other real-world campaign, you should expect a much lower response rate.

We then build a synthetic network of 100 people respecting the distribution of degree, likelihood to share, and threshold to adopt that we drew from the 24 samples. Please download the network data from BB, build the network and import the attributes of nodes.

```
nodes <- readRDS("class_nei.rds")

vertices <- readRDS("Attributes.rds")

g <- nodes %>%

  rename("ID" = matrix..) %>%

  pivot_longer(cols = starts_with("V")) %>%

  filter(!is.na(value)) %>%

  dplyr::select(ID, value) %>%
  graph_from_data_frame(vertices = vertices)
```

Imagine you are a group of data scientists that are interested in robustness of this network. You want to study the optimal percolation problem, e.g., to find a minimal set of nodes which if removed would break down the network into many disconnected pieces. In the context of influence maximization problem, your objective function f(s) will be the size of largest component in the network after the removal of s (s is a set of nodes).

Do questions **27-30** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

Before any removal of nodes, this class network is a connected graph with only one component. The size of largest component is therefore 100. Find a single node (n=1) that after removal, will lead to the greatest decrease in the size of the largest component. Show your teacher the ID of node and describe your observations. **[Question 27, 3 point].**

To answer the above question, you might search explicitly to remove the 100 nodes one by one from the network. But can you apply such explicit search if you try to find out a set of nodes (i.e., n>1) that will lead to the greatest decrease? **[Question 28, 2 point].**

In the lecture, we mentioned two board categories of approximation algorithms. One is by heuristics such as degree, closeness, betweenness and eigenvector. The other is by greedy algorithm. Using 1) degree heuristics 2) betweenness heuristics and 3) greedy algorithms, find out a set of 5 nodes that should be removed to produce the greatest decline in component size. Compare the solutions provided by different algorithms. Do you find significant differences between the efficiencies of these algorithms and why? **[Question 29, 5 point].**

Try to find out the percentage of nodes that you should take out from the network that, by doing so, the size of the largest component will decrease dramatically. You should compare the percentages you find according to the greedy algorithms, degree heuristic and betweenness heuristics. Describe the answer to your teacher with the supported plots. What's the potential implication if you consider vaccination strategy for this small community? **[Question 30, 5 point].**
(Hints: you should develop a plot where the x-axis is the percentage of nodes that you take out (q), and y-axis is the size of the largest component after you take out q (G(q)).)

Do questions **31-33** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

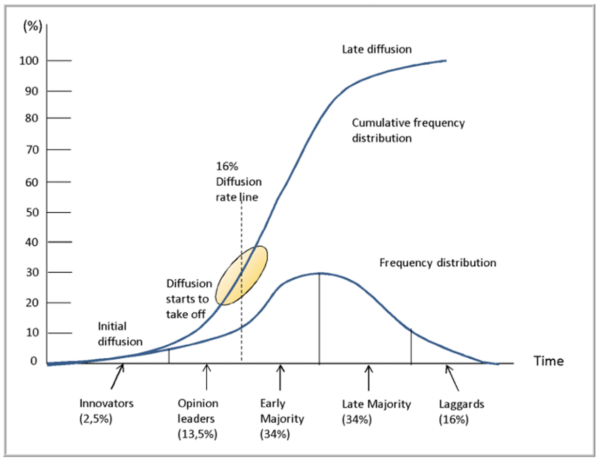
Now let's move on to the functional influence maximization problem. That is, to investigate how to promote 1) a Youtube video 2) a vegetarian recipe and 3) an academic paper in this class. For example, you own a Youtube Channel and want to promote it in this class. You don't have the time and energy to show your most proud video to every student of this class. Therefore, you want to find out a few students that you should talk to and show them the video, which we call this process as *seeding*. And the students that you chose are *seeds*. Afterwards, if they are impressed, you expect that they will share it with their peers and maximize the popularity of your video in this class. Intuitively, these 'seeds' should be the most popular ones in this class. That is the ones that have many friends (those said that you've got too many friends to name in this class!) But is this intuition always true?

Let's first test using an independent cascade (IC) model. According to your response, we assigned the probabilities to share 1) a Youtube video 2) a vegetarian recipe and 3) an academic paper in the node attributes. Build an IC model according to these probabilities. In this model, everyone has two states: S=0, unadopted, and S=1, adopted. Everyone starts from S=0. By seeding, you change the initial states of some nodes in the network from 0 to 1. These seeds pass the signal to their neighbors according to the probabilities we assigned them, which you will find in the node attributes. If you seed node 1, for example, it has a 45% probability to share a Youtube video. A random number will be generated between 0 and 1, and compare with 45%. If the random number is smaller than 45%, node 1 will pass the signals to its neighbours. For every seed and activated nodes, they only send out signal to their neighbours at the timestep right after their own activation. At other time steps, they remained activated but cannot send out signal again.

If you can seed only one person, who will you choose? Will you choose the same person to promote these three different things? **[Question 31, 5 point].**

You now have a little more time or budget to seed 3 people. Using greedy algorithms and degree heuristics to 1) find out the seeds for Youtube, vegetarian recipe and paper, respectively. 2) the differences of the performance by degree and greedy algorithms. 3) Check out the network attributes (e.g., centrality measures) and probabilities of the seeds provided by the greedy algorithms. What do you find? **[Question 32, 5 point].**

Recall the diffusion curve we mentioned in Lecture 3, slide 27. For an ideal case, the diffusion curve is S-shaped, which you can find the 'social tipping' point at the waist of the S. Produce the diffusion curves (x-axis as time, y-axis as the percentage of people being activiated) based on the 3 seeds that you selected according to the greedy algorithms, for Youtube, vegetarian recipe and paper, respectively. Do you find any standardized S-shape curve? And explain how you interpret their shapes. **[Question 33, 3 point].**



(A S-shape diffusion curve)

Do questions **34-36** (below) and show your teacher all of the results together. For questions where you generate a plot, table or code, you should copy this in your answer document (maybe using a screenshot).

Next we move on to threshold model. According to the survey, we assigned thresholds for one to check out 1) a Youtube video 2) a vegetarian recipe and 3) an academic paper. Build a threshold model according the assigned thresholds by nodes. In this model, everyone has two states: S=0, unadopted, and S=1, adopted. Everyone starts from S=0. By seeding, you change the initial states of some nodes in the network from 0 to 1. At every time step, each node will look at the states of their neighbours and decide whether or not they will adopt. For example, if node 1 is connected with node 3, 5 and 7, and the threshold of node 1 is 50%. At time 0, the states of node 3, 5 and 7 are all 0; node 1 remains unactivated. At time 1, the state of node 3 changes from 0 to 1. Since 1/3<50%, node 1 remains unactivated. At time 2, the states of node 5 also changes from 0 to 1. Since 2/3>50%, node 1 will be activated.

If you can seed only one person, who will you choose? Will you choose the same person to promote these three different things? **[Question 34, 5 point].** (Note that the thresholds we assigned here are independent from the probability in the IC model, that is partly why you might find the seeds are very different from two models.)

Again, you now have a little more time or budget to seed 3 people. Using greedy algorithms and degree heuristics to 1) find out the seeds for Youtube, vegetarian recipe and paper, respectively. 2) the differences of the performance by degree and greedy algorithms. 3) Check out the network attributes (e.g., centrality measures) and threshold of the seeds provided by the greedy algorithms. Do you find some common properties of the seeds? What difficulty will you foresee to implement the theoretical solutions suggested by the greedy algorithm in promoting the vegetarian receipe and the paper? **[Question 35, 5 point].**

Image you are now making a budget plan to show the cost-effectiveness of different seed sizes. You need to investigate the return (i.e., increase of activation size) to input (i.e., change of seed size). Try to produce a plot to show cost-effectiveness of increasing seed size. How will you interpret the shape of the curve? And if you target is to achieve 90% adoption rate, at least how many people you should seed? Demonstrate for the cases for vegetarian recipe **[Question 36, 3 point].**

BOUNS QUESTIONS:
Build your own network formation model. Understanding how networks form and evolve is an essential component in network science. In plain words, network formation model manifests how a new edge is added to an existing network. We can think of the formation of a new edge (i, j) as *i* "choosing" to connect with *j*, where the set of alternatives available to *i* is the set of all other nodes. The key question behind is easy to state: why did *i* choose *j*?

The BA model, for example, argues that *i* chose *j* simply because *j* has a lot of friends (Simply follow the tide when you don't know where to go ☺). Using your intuition or experiences, can you imagine other factors in play? Can you build a network formation model based upon your own mechanism and model the growth of a network from 1 to 500?

Show your teacher 1) the mechanism behind your network formation model 2) how you represent it in the model 3) a few plots on how the network grows over time 4) degree distribution of this network. **[Question 37, 5 point].**
(Hints: If you are running out of idea, you can consider a twist of BA model. Using citation network as an example, researcher found that people not only like to cite the most cited ones but also have a favor towards the newly published ones. In other words, ages of the node (when this node is added to the existing network) is negatively correlated with the likelihood of a new citation. Age can be a new factor that you should consider to explain the growth of social network.)

BOUNS QUESTIONS:
In Q30, you studied how the largest component of this network was break down by the nodes suggested by greedy algorithm and the degree and betweenness heuristics. Can you propose an even more efficient algorithem (i.e., break down the largest component with even less percentage of nodes taken out)? **[Question 38, 5 point].**