Université catholique de Louvain

LINMA2472 : Algorithms in data science

Homework 1 : Networks

Students : Vincent Cammarano - 5391 21 00
Louis Parys - 7256 17 00
Mattias Van Eetvelt - 1660 18 00

Professor : Jean-Charles Delvenne
Gautier Krings
Estelle Massart
Rémi Delogne
Bastien Massion
Brieuc Pinon

October 2022

1 Introduction

In this first homework we have decided to tackle the last book of the Harry Potter saga : Harry Potter and the Deathly Hallows. All the characters names and the book are provided in the `characters_list.csv` and `hp7.txt` file, respectively.

2 Analysing the communities

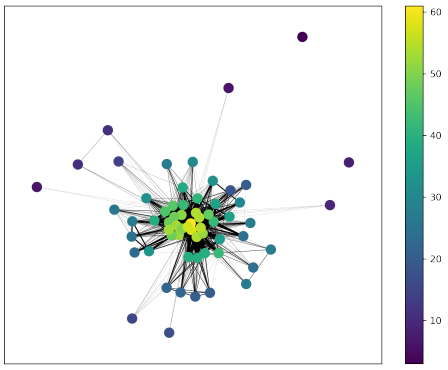
2.1 Network visualisation

Once the data has been prepared, let us build some graphs in order to have a better understanding of the interactions taking place in the book.

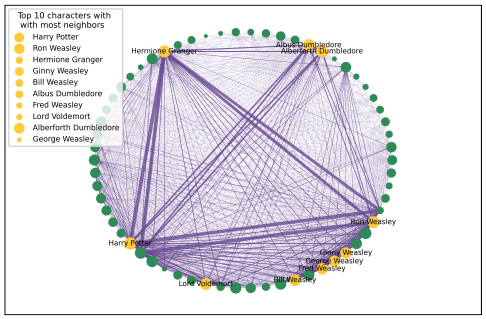
Figure 1a shows that there are a few nodes with a very high degree or a very low one. The majority of the nodes have an average connectivity. As a matter of fact, the average degree of a node is 33,71. Edges width is proportional to weight of said edge. This allows to clearly see the principal characters interactions.

Distinguishing communities is not so trivial. However, it is already possible to note the presence of what looks like to be a hub. On the other hand, some nodes are very poorly connected.

Figure 1b displays interactions between characters in a different way. Harry, Ron and Hermione have the most occurrences with 4100, 2961, and 2648 respectively.



(a) Scatter network with node coloring based on the normalized degree of each node. *NB : Two nodes seem to be unconnected however it is not the case. Width is extremely thin. This is only the consequence of graphical choice.*



(b) Circular network showing connectivity between all characters. Node sizes are proportional to the number of occurrences of said character. The 10 characters with the most occurrences are labeled.

FIGURE 1 – Network visualisation

2.2 Assortativity and degree distribution

The assortativity is computed and has a value of -0.1968. This means there are relationships between nodes of different degree, up to some degree since the assortativity coefficient is close to 0. It means nodes with low degree are connected with nodes of high degree. In this context, it means supporting characters interact with principal ones, which makes perfectly sense.

There is no tendency in the degree distribution. In other words, the distribution is not heterogeneous. However, there are few nodes that share the same degree.

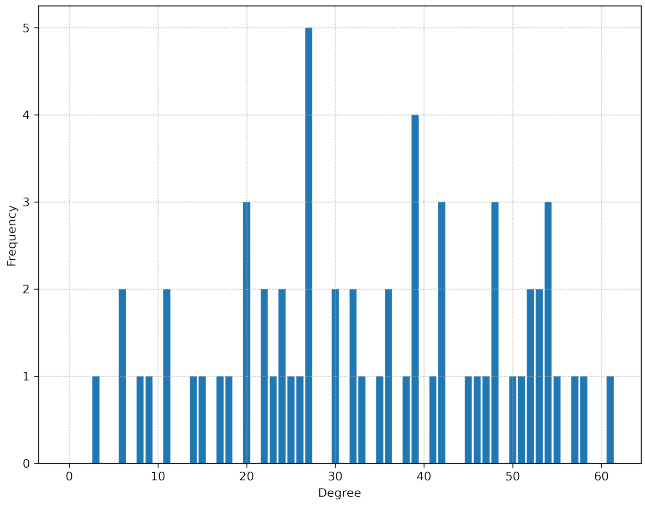


FIGURE 2 – Degree distribution

2.3 Louvain algorithm and spectral clustering on the Laplacian

2.3.1 Louvain algorithm

The Louvain method is a community detection algorithm that has been developed with the aim of maximising the modularity of the graph, in other words merging communities if this improves the strength of cluster present in the network. The following graph is obtained by running the python library with our networkx object. [1]

The Louvain algorithm splits the characters into four communities from the main ones to the supporting ones. Again, this makes sense considering the book that we've chosen.

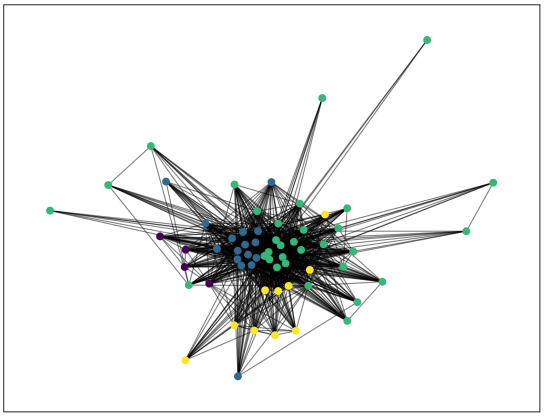


FIGURE 3 – Community detection using the Louvain method

1. the graph has been created thanks to the louvain algorithm python documentation : `https://python-louvain.readthedocs.io/en/latest/`

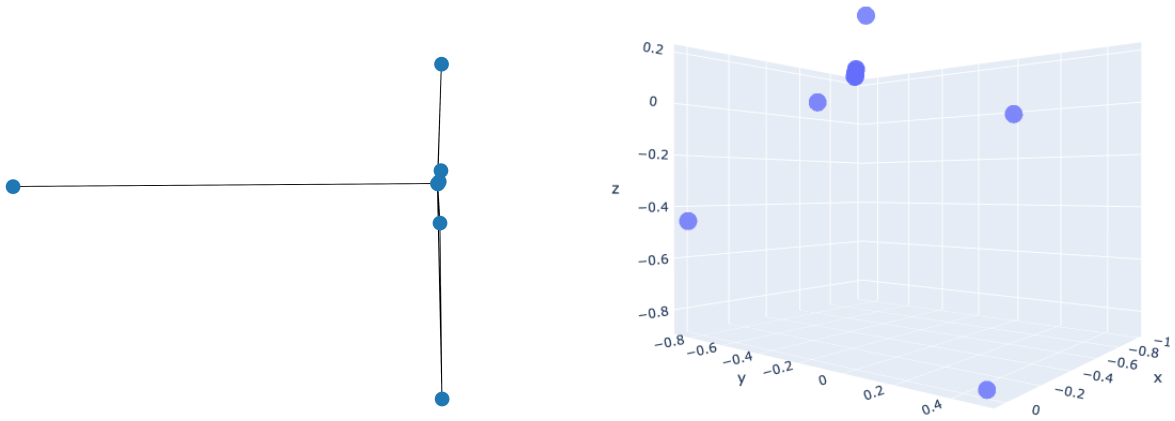
2.3.2 Spectral clustering on the Laplacian

The second part of this section is to reiterate the experiment with the help of the spectral clustering on the Laplacian. The representation of the graph with a Laplacian matrix is :

$$L = D - A$$

Where D is the diagonal matrix and A is the adjacency matrix.

In order to find cuts in the graph and detect communities, we need to compute the eigenvector and eigenvalues. Then, the number of cuts are computed based on the numbers of k eigenvalues with zero value and their eigenvector with constant values. In order to use this method, we applied the *spectral-layout* technique available in the networkx python package. The result is shown below.



(a) Laplacian clustering

(b) 3D Laplacian clustering

FIGURE 4 – Laplacian clustering visualisation

As displayed on the figure 4a, the spectral clustering on the Laplacian method doesn't give a lot of information. This can be explained by the fact that there are a lot of points concentrated in one cluster. Therefore, communities are hard to detect. This is resolved by plotting the Laplacian clustering in 3D. It is now possible to observe the point that was hidden behind the biggest cluster. This corresponds to the dark blue point on figure 4b. It is now safe to say that 6 clusters, corresponding to 6 communities can be observed.

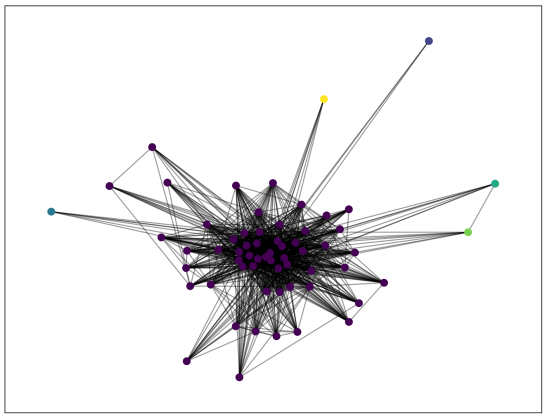
2.3.3 A comparison of the two methods

Using the results obtained with the spectral clustering, it is possible to compare it with the Louvain method. In order to do so, we computed a graph using the K-means function from the scikit-learn package and passing the number of communities to 6.

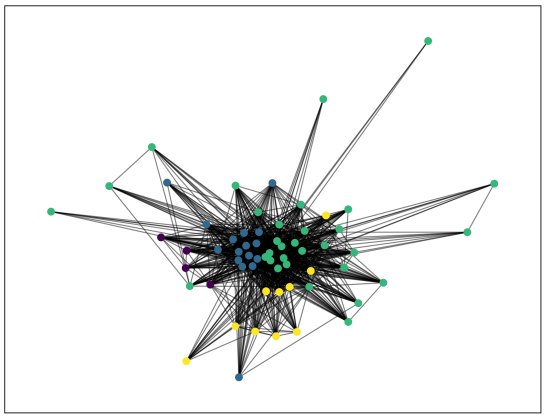
The communities detection comparaison is presented on figure 5. We see that the two methods find different numbers of communities. Indeed, the spectral decomposition of the Laplacian find six non equitable communities. Actually, there are five communities with only one character and one community with the rest of the characters. This representation of the communities taking place in the book is of course not realistic.

On the contrary, the number of communities found by the Louvain algorithm is four, and they are well distributed. This seems to be a good representation of the book if we think about the four "houses" in the Harry Potter book. But we don't have to consider that since our representation is based on the occurrences of personage per page. Therefore, the communities are made on the number of appearances. Thus, the communities regroup the characters by order of importance in the book.

In conclusion, the Louvain algorithm is better to distinguish the communities based on the number of appearances of the characters in the book.



(a) Communities detection with K-means feed by the spectral decomposition of the Laplacian.



(b) Communities detection using the Louvain algorithm

FIGURE 5 – Comparison of communities detection using two different methods

3 Maximasing the influence in the graph

3.1 Influence maximisation problem

In this section, we are going to try to solve the influence maximization problem. To understand the concept, imagine that you want to spread a disease through all the characters of the book as fast as possible. The solution of this method would be a set of characters that maximize the speed of the spread in our network. We have been asked to find a character set as large as 5% of the total amount of characters in our book. We have registered 62 different people so the absolute value of 5% of that would be 3. Thus, our model will return 3 characters. Those can be seen on the graphe below ("Regulus Arcturus Black", "Sirius Black", "Lavender Brown").

For this problem, we will use the "Greedy algorithm". A greedy algorithm is any algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage. In many problems, a greedy strategy does not produce an optimal solution, but a greedy heuristic can locally yield to one that approximate a globally optimal solution in a reasonable amount of time.

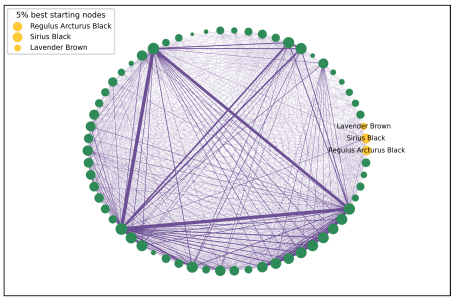


FIGURE 6 – 5% best starting nodes for maximisation problem

It seems odd that the three characters that maximize the spread are ones with very few edges. However, it appears that although they are not linked with many characters, the ones with how they are, possess the most edges. Which may explain our results.

3.2 Independent cascade model

We have, in this part, implemented an "independent cascade model" wich is a stochastic information diffusion model where the information flows over the network through Cascade. Nodes can have two states, active : It means the node already influenced by the information in diffusion. Or inactive : node unaware of the information or not influenced.

Then we've used this model with three different characters starting sets, each chosen by a different method. The greedy method (used in section 3.1), a highest degree method (which are the nodes with the highest number of edges) and a random method (which chooses three random nodes in the hall set). We can then compare which method was the most efficient.

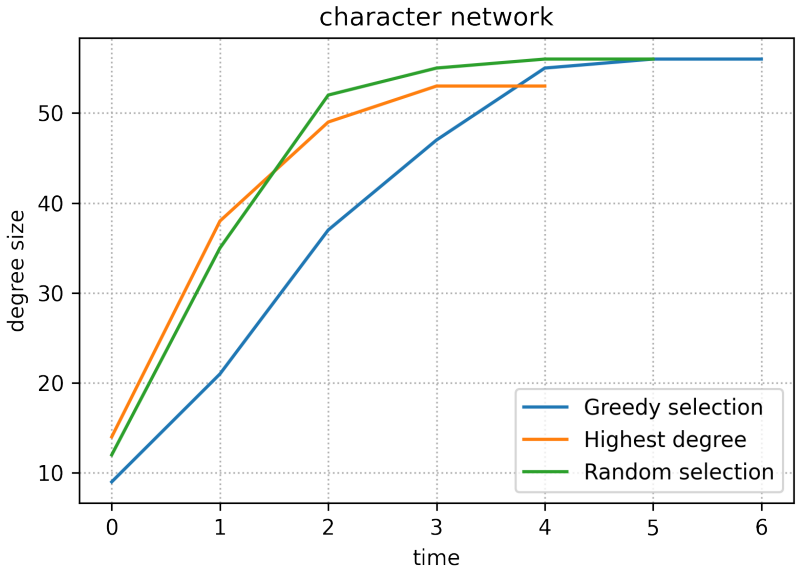


FIGURE 7 – Independent cascade with different algorithms

As expected, the greedy method is the one that performs the "worst" in the short term. It only looks for the optimal local solutions making it very slow but still very efficient as it has spread the disease the furthest. On the contrary, the highest degree method has a very fast spreading rate in the short term because it uses characters that have many edges. Although, it doesn't spread as good as the greedy method.

3.3 Barabasi-Albert network

We generated a Barabasi-Albert graph with the networkx implementation. We set the number of nodes to 62 and the average degree to 34, which is the average degree that we observed on our graph. This parameter allows us to specify how much the Barabasi-Albert graph should be connected.

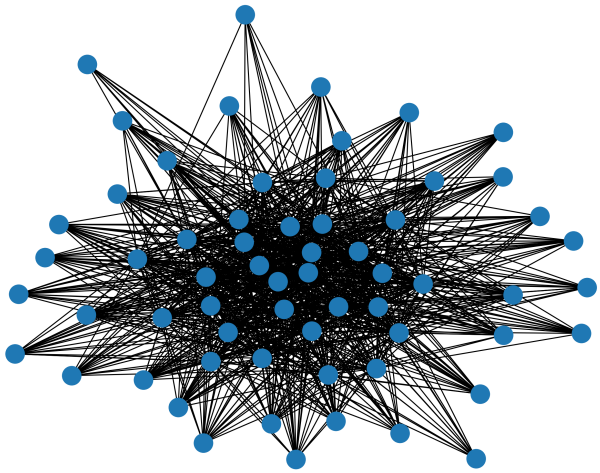
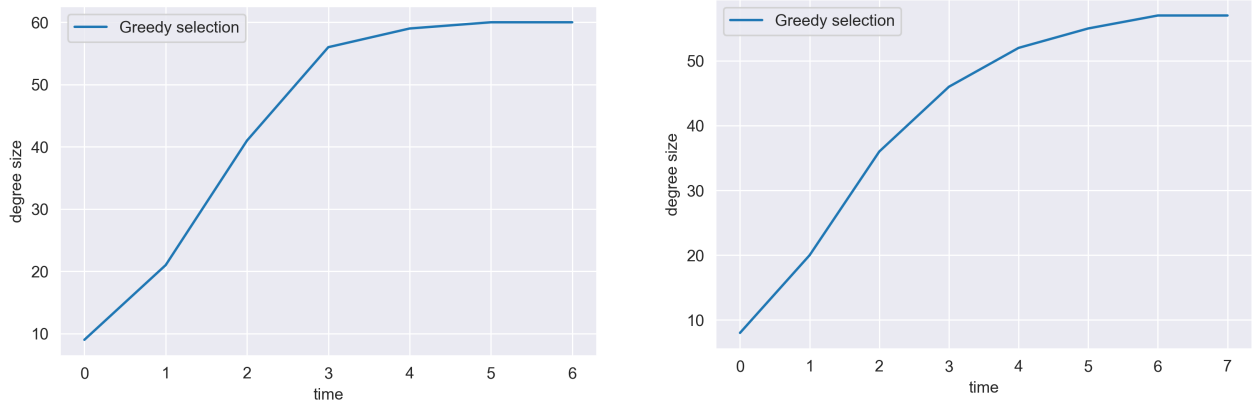


FIGURE 8 – Barabasi-Alabert network

As shown above, the graph generated looks a lot like the graph on figure 1a. The nodes are a bit more spread, but we observe a concentration in the center and less connected nodes on the periphery.

Let's compare the two graphs to see how much they differ when we apply the greedy algorithm on them.



(a) Greedy algorithm on the characters network of the book (i.e. the graph presented on figure 1a)

(b) Greedy algorithm on the Barabasi-Albert generated network

FIGURE 9 – Infection rate through two different graphs

Both figures are substantially the same. This is a result we could have expected since the Barabasi-Albert generated has similar parameters as our characters network graph. Even with a low probability to infect other nodes (p=0.1), all the nodes are infected very rapidly. This is explained by the fact that our graph is highly connected. The average degree of a node is about 34 (and this is why we chose a low probability of infection) for a total number of nodes of 62. Thus, the infection spreads rapidly for both graphs.

4 Annexes

4.1 Detailed view of the 3D Laplacian clustering

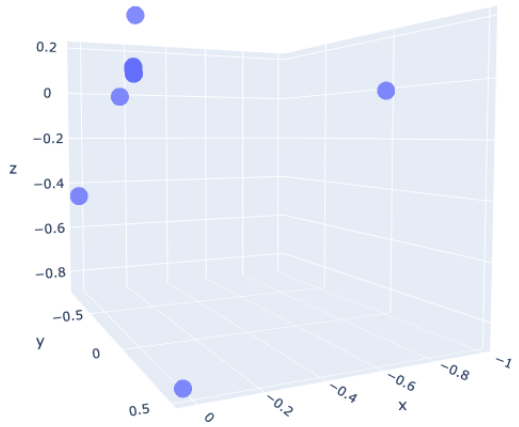


FIGURE 10 – View 1

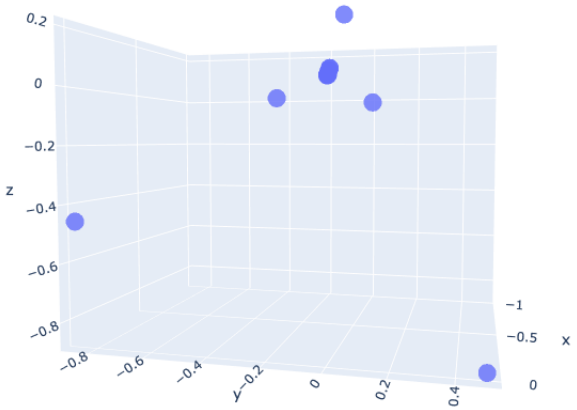


FIGURE 11 – View 2

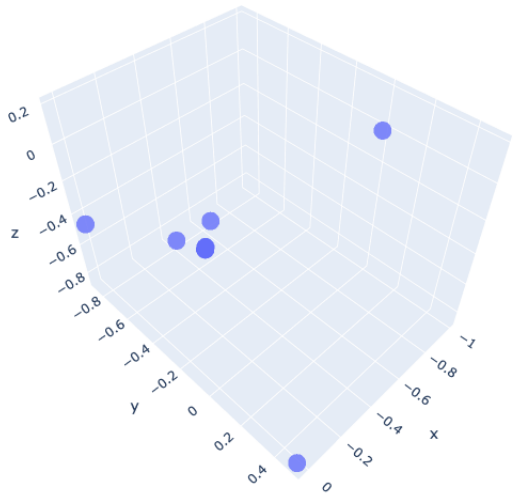


FIGURE 12 – View 3