Graph Analytics course

Assignments Report

by Viktoriya Bodnar

UniGe 2018

In this report I am going to explain the main steps of each assignment and will provide some clarification of the results obtained from different experiments.

DATASET

First of all, I would like to introduce the dataset that I was working with. The dataset that I used, was manually created after querying the API of TMDB (The Movie DB website <https://www.themoviedb.org/>). Since I was a frequent user of this movie database, during the whole period that I had been using it for movie search, I was adding the movies which I particularly liked to the Favorite Movie List. By scraping the API I extracted this list of 248 movies of different genres and years. The file that I obtained was a txt file with all the movies and some additional information such as title, release date, genres, popularity, average vote and number of votes in total etc. Below you can see the screenshot of the txt file.



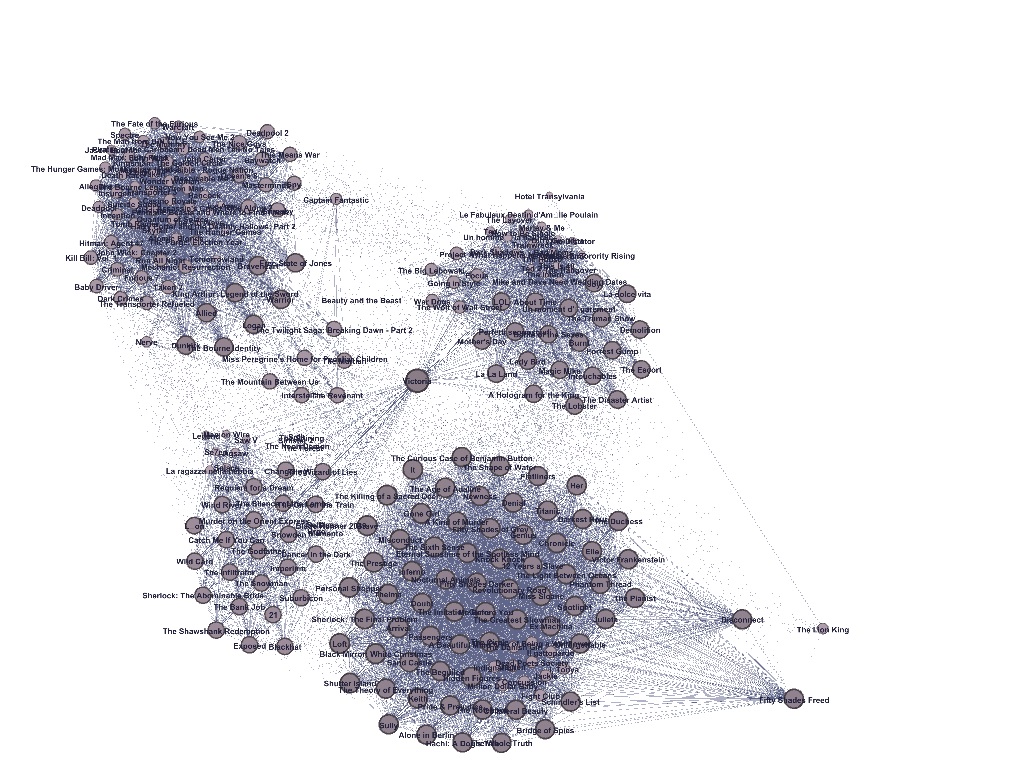
In order to create a graph of my preferred movies, I was considering different possibilities. At the first glance, it seemed easy to create a graph. Initially, I tried to create a connections based on the production companies of different movies. However, I encountered the problem of time consuming computation (more than 30 h) since I had to search each movie by its id within the API dataset and check for its coincidences with the rest of the movies in a production companies list. Therefore, I opted for the simpler approach of creating edges, which was the connection based on the genre of the films. In other words, every time that the two movies share the same genre there is an edge, of cause, by omitting the duplicate links. Finally, the output genreGraph.csv file looked like this. The titles of the movies will represent the nodes and the links are represented by sharing the same movie genre. The resulting graph is undirected.

A screenshot of a cell phone

Description generated with very high confidence

The full code of querying the API and obtaining the final output csv file is given inside scraping.py file.

In order to visualize the graph I used Gephi software and the created graph can be seen from different perspectives in the following figures.



In this picture we can see the global situation of the movie graph. In particular, there are several clusters of nodes, some hubs and some “outliers”. The size of the nodes are defined by the degree and the clustering was performed by a mix of different Gephi algorithms for visualization.

A close up of a piece of paper

Description generated with high confidence

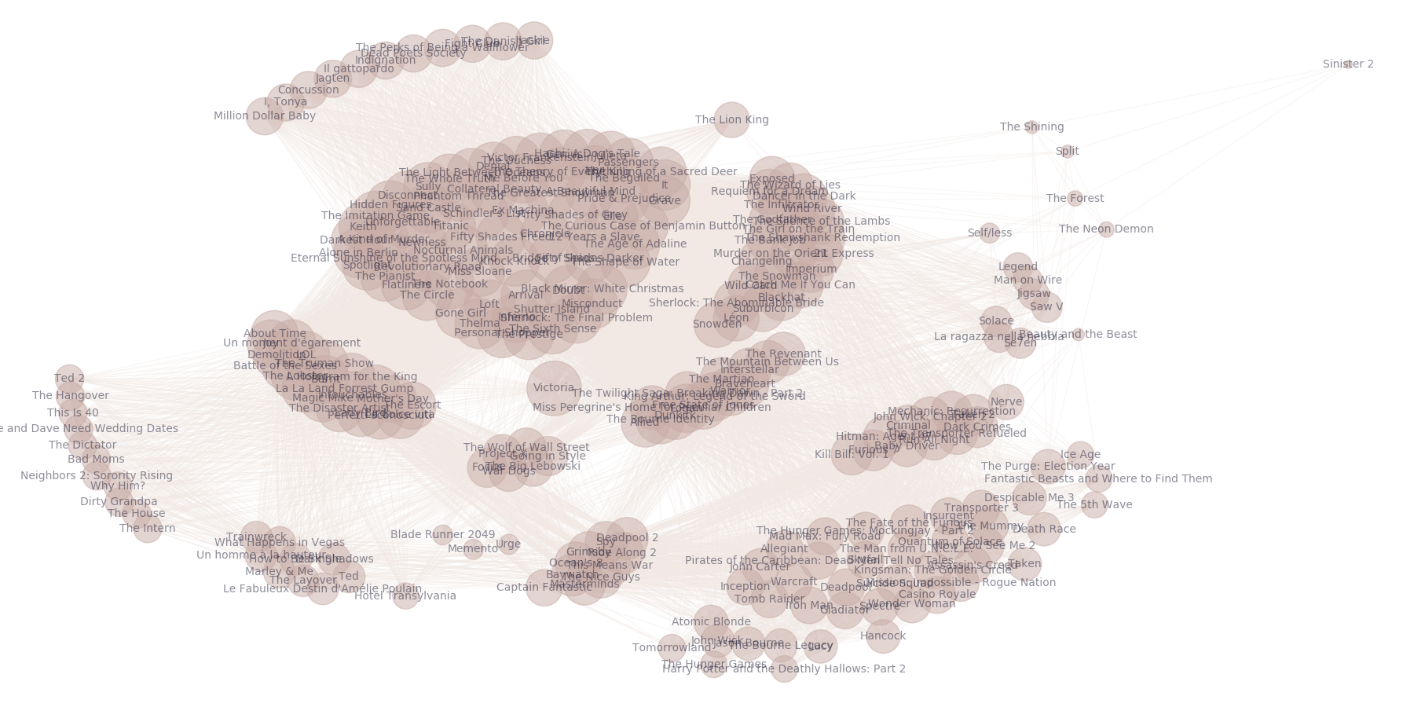
In this figure we have slightly different angle of view, since not only the size but also the color encodes different nodes degrees. For instance, the higher the degree the bigger and the darker the node is. We can easily notice that there is one the biggest node in the center named “Victoria” (by the way, it is a pure coincidence that it is the same as my name), which have the highest number of links. This can be explained by the fact that the movie cannot be defined by only one genre, but it is a multi-genre film. In fact, it is very common to list more genres for the movie in order to make it match the broader amount of queries. In addition, usually there are some genres that go in pairs, like thriller and horror, or drama and romance etc.

In all assignments I used python networkx 2.1 library and some additional libraries in order to easily and conveniently perform the calculations, visualizations and simulations.

ASSIGNMENT 2 (assignment2.py)

GRAPH CHARACTERISTICS AND METRICS

Firstly, I imported previously created movie graph and using kamada\_kawai\_layout visualized it. In order to respect more or less the same approach that I used in Gephi, I encoded the node degree into the size of the node. By using the semitransparent colors I partially overcome the problem of overlapping nodes. The resulting visualization is shown in the picture below.



From this figure we can differentiate the nodes clusters and make a guess about the hubs. However, in order to check how actually the graph is structured, which nodes are important and what is the cohesion level of the graph, we need to compute some measurements. Moreover, we need not only compute the metrics, but also analyze them and make some conclusions, always keeping in mind the fact that this graph is only a portion of a real global graph.

Firstly, the number of nodes is 246 with 12192 number of edges (two movies that did not match any other movie were not taken into consideration since it is only 1% of the graph). The following table will show a number of graph metrics that will represent some insight within the movie network.

|  |  |
| --- | --- |
| Average degree | 99.122 |
| Minimum degree | 4 |
| Maximum degree | 167 |
| Density | 0.405 |
| Diameter | 4 |
| Average length of shortest paths | 1.668 |
| Number of connected components | 1 |
| Global clustering coefficient | 0.768 |
| Number of communities | 4 |
| Modularity | 0.315 |

Looking at the average degree of the network, we can see that it is something in between the minimum and the maximum degrees. This can give us an idea that the nodes of the network might have more or less the same degree. However, in order to analyze better the type of the graph I plotted the distribution degree, which is shown in the following figure.

A screenshot of a cell phone

Description generated with very high confidence

In the first chart I present the histogram of the degree distribution and we can see that there is a prominent peak at degree number 137, smaller peaks at 77,116 and 125. In general, I noticed that the nodes with the degrees in the center of the histogram contribute to the majority, while the number of nodes with very small or very high degrees are quite rare. In fact, from the line chart right below the histogram we can follow the “continuous” version of degree distribution. The red line was added after in order make it easier to see the similarity of this degree distribution with the binomial distribution, which is approximated by a Poisson distribution. Taking into consideration this curve shape, we can conclude that this particular movie graph is an instance of a random graph.

The diameter of the graph is 4 that is quite short. This implies the small world phenomena, which means that the average path length depends on the system size but does not change drastically with it. Small world network theory predicts that the average path length changes proportionally to log N, where N is the number of nodes in the network. In fact, by looking at the our average length of shortest paths, we see that two randomly chosen nodes will be just ~ 2 hops apart on average. The probability that two nodes are connected then can be calculated by dividing the average degree 99 by 246-1, which is ~ 40%.

The density of the network is approximately 0.4, that leads us to the conclusion that the graph is not complete and thus can present some vulnerabilities to the attacks.

The global clustering coefficient is 0.768, which means that there are a lot of cliques inside the network. This can be explained by the fact that since the number of genres is relatively small, a lot of different movies will have the same genres. The local clustering confirms the presence of cliques, since a lot of nodes have coefficients equal to 1.

The number of communities that was found within the network is 4. I plotted the graph with 4 different communities detected and colored by different colors to facilitate the discrimination. The graph can be seen in the picture below.

A picture containing text, map, table, indoor

Description generated with very high confidence

The method to find the communities was based on modularity maximization. However, the number is still small 0.315, which can be in part explained by the communities overlapping.

After analyzing the basic metrics of the network I was also interested in some centrality measurements and rankings of the nodes according to different characteristics.

First of all I ranked the nodes due to the degree centrality to find most important nodes (hubs). I reported only 5 top nodes since the total number of nodes is not very big.

Top 5 Hubs by Degree centrality-----------

Victoria : 0.682

Inferno : 0.576

Shutter Island : 0.576

Black Mirror: White Christmas : 0.576

The Sixth Sense : 0.576

We can see that our guess about “Victoria” movie importance seems to represent the real situation.

In order to analyze the brokers of the network, I ranked the nodes by betweenness score. In this case we can state that the “Victoria” node is not only important in terms of degree, but also in terms of brokerage, since it is situated between different portions of the graph and can represent the single point of failure. Another important linkage between the clusters is performed by “Captain Fantastic” node.

Top 5 Brokers by Betweenness centrality-------------

Victoria : 0.018

Captain Fantastic : 0.016

Nerve : 0.01

Braveheart : 0.01

Warrior : 0.01

Another centrality measure that I calculated is closeness, that indicates the nodes that are close to the information flow. The 5 top ranked nodes are given below.

Top 5 Central Nodes by Closeness centrality-------------

Victoria : 0.756

It : 0.702

The Killing of a Sacred Deer : 0.702

Grave : 0.702

Inferno : 0.7

Besides the previous metrics, I also applied the PageRank and HITS algorithms in order to see the impact of these techniques on the ranking. As these algorithms were implemented for directional networks, the networkx library methods convert each unidirectional edge into bidirectional to work. This changes the topology of a graph a bit and we need to take it into consideration when analyzing the results.

Top 5 Nodes by PageRank-------------

Victoria : 0.0064

It : 0.0055

The Killing of a Sacred Deer : 0.0055

Grave : 0.0055

The Sixth Sense : 0.0053

The PageRank gave more importance to nodes with important incoming links. As we can see, “Victoria”, “It” and “The Killing of a Sacred Deer” have the highest number of links with nodes that are also very well linked. The distribution of PageRank scores can be seen in the following figure. It highlights that the majority of the nodes are ranked inside the range of 0.0022-0.0055, there exists one node with the lowest score 0.001 and one node (“Victoria”) with the maximum score. There is a peak at ~0.005 which can suggest that there is a cluster of nodes that are linked by the same number of important nodes (probably the same nodes).

A picture containing screenshot

Description generated with very high confidence

In the case of HITS application on undirected graphs, the hubs and the authorities will be scored in the same way, due to the fact that every edge is transformed to bidirectional. Therefore, the HITS scores are higher for the nodes that are both pointing to many authorities and are pointed by many hubs.

Top 5 Nodes by HITS-------------

Shutter Island : 0.00699

Inferno : 0.00699

Black Mirror: White Christmas : 0.00699

The Sixth Sense : 0.00699

Doubt : 0.00699

Since there are more than 5 nodes that got the same highest scores, the order of first 5 depends on how the graph is visited.

To sum up, having analyzed all the metrics I have computed, I can conclude that the node “Victoria” is the most important node from different points of view. Besides, there some more important nodes in terms of ranking and centrality like “It”,” The Killing of a Sacred Deer”, “Grave”, “Inferno” and “The Sixth Sense” (more frequently encountered in the top lists). All of these nodes represent the influencers in the network and must be protected from different attacks in order to keep the network well working.

ASSIGNMENT 3 (assignment3.py)

SOCIAL CONTAGION

In this assignment the purpose is to investigate the social contagion in different networks by implementing the networked coordination game, which aims at maximizing the payoff by making some choices of strategies.

In the first part I was analyzing the social contagion on karate\_club\_graph of networkx library. The generated graph is shown in the picture below.

A close up of a device

Description generated with high confidence

Initially I used 10% of infected nodes and the payoff matrix to be the same as in the slides.

|  |  |  |
| --- | --- | --- |
|  | A | B |
| A | 3,3 | 0,0 |
| B | 0,0 | 2,2 |

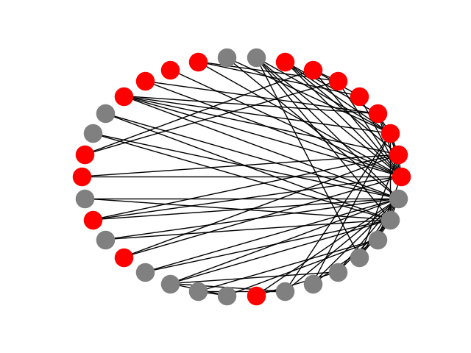
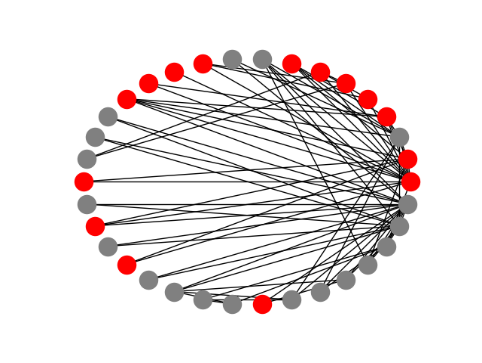
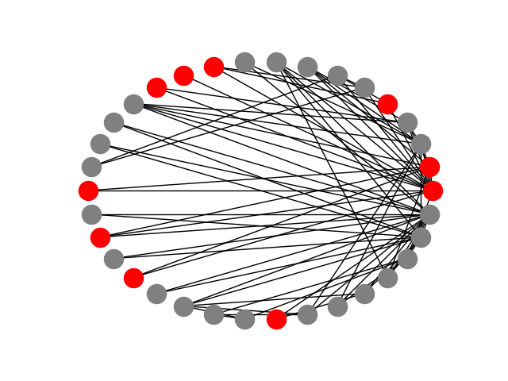
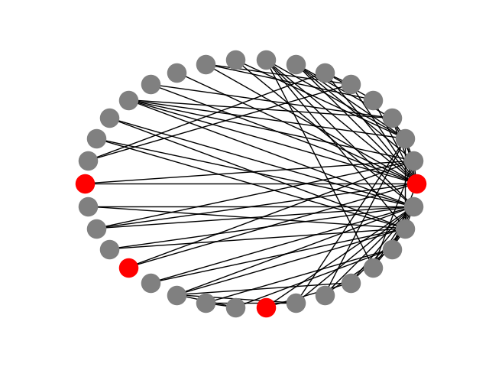
According to this matrix, the nodes are going to switch from B to A if the threshold of neighbors that are using A is at least 2/(3+2) = 0.4. However, I have seen that with this configuration it was very difficult to obtain the spreading effect or complete cascade of the nodes. Thus, in order to see the diffusion of A inside the network I slightly modified the values of matrix and increased the percentage of initially “infected” nodes. So the resulting parameters that I chose based on the empirical experiments are: the percentage of initially infected nodes is set to 12% and the payoff matrix for two strategies A and B is defined as follows:

|  |  |  |
| --- | --- | --- |
|  | A | B |
| A | 6,6 | 0,0 |
| B | 0,0 | 3,3 |

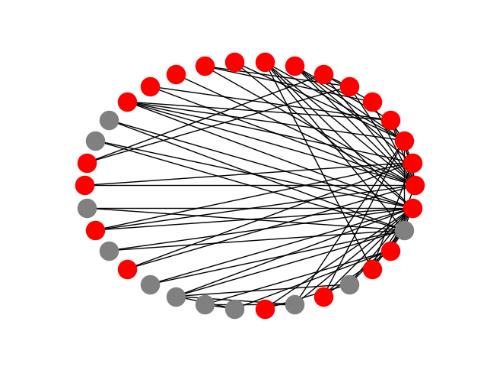
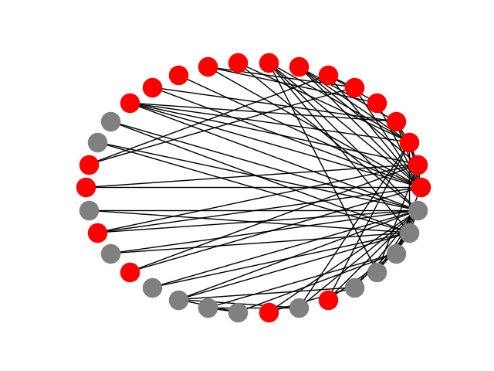
In this case the threshold to be considered at every decision point is 3/(3+6) = 0.33, which is lower and thus, makes it easier to achieve a complete “infection” of the network.

In order to simulate the diffusion effect I used a python library ndlib, which provides some already pre-defined diffusion models. I applied the ThresholdModel which is based on the same logic as the networked coordination game. The only parameters that I passed to the model were the percentage of infected nodes and the threshold of neighbors to check at each iteration while making the choice if to spread further or not. For the real simulation I used previously defined model and a number of iterations to do in order to converge to complete cascade. This number was chosen empirically and was set to 10. The visualization was performed at each iteration and then stopped when the complete cascade was reached. The gif was created using a web based tool (can be found in /pics), but it cannot be shown in a Word document, so I will report the sequence of images here for the visualization purposes.

INITIAL STATE Iteration 1 Iteration 2 Iteration 3



Iteration 4 Iteration 5 Iteration 6 Iteration 7

A picture containing kitchenware

Description generated with high confidenceA picture containing kitchenware

Description generated with high confidence

From the following sequence of images we can see that the full coverage occur after 7 iterations. Of cause, it depends on which nodes are initially infected, since this is random choice the results can be a little bit different each time we simulate the social contagion. In addition, there exist some combinations of randomly infected nodes that will not lead to the cascade. In general, the spread is quite uniform, in the sense that the infection is diffused gradually.

In the second part of the assignment I was exploring the diffusion effect on my own movie graph. Since the movie graph is bigger and the topology is different than the previous one, the configuration that worked before is not valid anymore. Thus, in order to show the impact of spreading within the network the following parameters were chosen: the nodes that use strategy A are 25% and the payoff matrix is:

|  |  |  |
| --- | --- | --- |
|  | A | B |
| A | 6,5 6,5 | 0,0 |
| B | 0,0 | 3,3 |

The values were found by trying different combinations. In this case the threshold to consider is 3/(6,5+3) = 0.31. However, sometimes even this “relaxed” configuration does not lead to convergence.

The same model was used as in the previous experiment and number of iterations was increased to 15, but since the diameter of the graph is small, in the most of the cases it still cascades completely in less than 10 iterations. The same visualization approach was implemented for the visualization. Below you can see the sequence of images that highlight the diffusion process across the movie network. (The gif is inside /pics folder)

INITIAL STATE Iteration 1

A picture containing sky, red

Description generated with very high confidenceA picture containing red, sky, indoor

Description generated with high confidence

Iteration 2 Iteration 3

A picture containing red, sky

Description generated with very high confidenceA picture containing red, sky, indoor

Description generated with high confidence

Iteration 4 Iteration 5

A picture containing red

Description generated with very high confidenceA picture containing red

Description generated with very high confidence

The full cascade was reached after 5 iterations, which ,as I already mentioned, can be explained by the fact that the graph has small world property and the diameter is quite small. The spreading is done from one group of nodes to another (cluster by cluster), resulting in the coverage of the whole network.

ASSIGNMENT 4 (assignment4.py)

NETWORK ROBUSTNESS

In this last assignment I will explore the robustness of different networks by simulating random failures and target attacks.

In the first part I generated a gnp\_random\_graph from networkx library with 100 number of nodes and p value equals to 0.1. Besides that, I also generated a powerlaw\_cluster\_graph with following parameters: 100 of nodes, 5 random links to be added to each node and the percentage of clique after a random linkage to be 0.4 (last parameter ensures higher average clustering). In order to compare the behavior of different graphs I implemented different types of attacks: random failure, hub attack (degree based), highly ranked nodes attack (PageRank scores), attack on nodes with low clustering coefficients and attack on brokers (high betweenness). I have tried to remove the nodes with high clustering coefficients but it was not effective at all because these nodes are the neighbors of cliques that does not necessarily mean to be a part them. Thus, I targeted the low clustering nodes, that can also reflect a certain degree of centrality measurement.

Before and right after each type of attack some metrics were calculated in order to see the impact of particular attack: number of nodes, number of edges, average degree, density, number of connected components, number of nodes in the giant component, diameter and the average length of shortest paths in the giant component. Furthermore, I calculated the diameter of the giant component at some threshold points f to see how different attacks influence the connectivity and cohesion of the graphs. The percentage of nodes to remove from the networks initially is set to 5% and increases each time by 10% till 95% is reached. The line plots shown below reflect the network fragmentation according to different attacks.

A picture containing screenshot

Description generated with very high confidence

A close up of a map

Description generated with high confidence

As we can see, the plots confirm the concept that different types of networks behave differently due to random failures or target attacks. In fact, it is easy to notice that the critical points of breaking the graph into pieces is smaller in power law graph than in random network. However, following the gray line we can state, that the power law graph is more robust to random failures, than random graph. Moreover, having analyzed the different charts, it is quite evident that the betweenness attack and hub attack are ones of the most powerful since it turns off the most influential nodes and brokers that connect different parts of the networks. The attacks on PageRanked nodes are more or less have the same effects on interconnectedness. The clustering attack seems to perform better on the power law graph, while it has the same trend as random failure on the random network. These considerations are also confirmed by the metrics (printed on the screen)

Following screenshots show the initial and after attacks measures having removed 25% of nodes of the random graph. We can immediately notice that either random or clustering attacks do not break the graph into more components, while targeting brokers or highly ranked nodes do. According to this, the diameter of the giant after brokers and top nodes removal increased. Despite the fact that the hub attack does not split the network, it still increases the diameter. Furthermore, the attacks that impact the most the average length of geodesic paths are attacks based on high PageRank score and high betweenness and high degree centrality.

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a cell phone

Description generated with high confidence

A screenshot of a cell phone

Description generated with high confidence

In the figures below, the measurements of the power law graph are shown before and after each attack with the same percentage of removed nodes as in the previous experiment (25%). Here it is seen that the hubs targeting attack and PageRanked attack are better in terms of spitting the graph into more components (from 1 to 13/15). These two techniques with the broker attack are the best when considering the diameter increase (from 3 to 17/16). When analyzing the average degree after the attacks, we can see that the most reduction was achieved after the hub and broker attack.

A screenshot of a cell phone

Description generated with high confidence

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a cell phone

Description generated with high confidence

Having analyzed the numbers and the plots, we can confirm that after the different attacks the diameter of the giant component grows and then after a critical point of removed nodes it dramatically decreases. That is the threshold where the network is split into more disconnected components, and no giant component exists.

In the second part of this assignment I investigated different attacks on my “realistic” movie graph in order to see if there are some similarities with the previous graphs behavior. Firstly, the graph highlights our hypothesis that targeting brokers is very efficient technique to disconnect the graph. Indeed, the green line which represents this type of attack increases faster and goes down earlier than others. This means that it has a significant impact on the diameter of the giant component. During the experiment, I have seen that the others attacks breaks the graph into small components only after 85% of nodes are removed, which suggests that this graph is quite robust to the different kinds of attacks. As previously noticed, the hub and PageRank attacks have more or less the same trends. The attack based on low clustering is not so effective as previously mentioned, since it does not impact critically on the diameter of the giant component or the number of components. However, in this particular case it behaves better than the random failure.

A picture containing screenshot, text

Description generated with high confidence

Having analyzed also the numbers before and after attacks, we can conclude that the broker removal influences drastically the average degree reduction and consequently, increases the diameter of the giant component. The attacks on hubs and highly score nodes by PageRank have the same effect on the network cohesion on average. And in this particular case, low clustering attack performs very effectively, in comparison to the other techniques, in terms of cut of the average length of geodesic paths ( from 1.67 to 2.13)

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidence

A screenshot of a cell phone

Description generated with very high confidence

In conclusion, in order to impact in a critical way on the network, we should know the type of the network. Moreover, based on that knowledge and some computed graph metrics we can decide which technique need to be used to produce more effective and more efficient results. The same approach can be used in the opposite direction in order to detect more vulnerable nodes, group of nodes and protect the network from random failures and target attacks.