Exploring the Actors Collaboration Network

Alessia Musumeci

Abstract

In this project, we study the actor collaboration network or co-stardom network. In particular, we will focus on detecting the most common features in social networks, such as community formation, assortative mixing and high transitivity, and on using different centrality measures to rank the most important actors in the network. We will study in detail its scale-free behaviour and compare our network with a known scale-free model, the Barabàsi-Albert model. Finally, we will study the dynamics of the network when we remove nodes. We will find that the results can be consider consistent with what we know from literature.

1 Introduction

Social network analysis is a branch of network science that studies networks where the vertices, or nodes, are people or a group of people,while the edges between them represent a form of social interaction [7]. Under this category, we can find numerous examples, such as work collaborations, friendships, exchange of goods or information, online social networks (such as *Facebook* or *Twitter*), and so on and so forth. Real networks tend to have a very low density, an average short distance, a power-law degree distribution and high clustering and transitivity. In this work we will study the co-stardom network of actors from all over the world.

In literature [5] it has become popular to rank actors using different centrality measures,for example by computing degree centrality, betweenness centrality and closeness centrality. In this work, for each measure, we will rank the top 10 actors of the network. It has been shown that different centrality measures usually rank the actors in different ways.

Secondly, it is quite natural for very large networks to divide into clusters or communities, and in this case it is quite intuitive to understand why since, for instance, we can expect actors of the same country to form a community inside the worldwide film industry or actors who act in the same genre of movies or in the same time period. Therefore we will study the groups and the communities within the network and their distribution. Moreover, we will investigate whether the nodes have the tendency to link with nodes which are similar to them, a feature known as assortative mixing.

Once we have this information, we can investigate how our collaboration network changes when we perturb it via a removal of nodes. In the end, since we expect a power-law degree distribution, we want to extract the value of the exponential of the distribution function and then we will compare our network with another known model that follows the same type of degree distribution, the Barabàsi-Albert model.

Besides a theoretical interest in studying the properties of real world social networks, these studies are interesting from a economic and (popular) cultural point of view. The film industry last year (2022) generated a business of 77 $ billion and its success revolves around many key figures,including actors. Therefore it might be useful for casting directors to hire,for example, actors that have worked in a specific community of actors, or give a shot to newcomers, that despite their poor experience, have worked with relevant names in the industry. Moreover, it could also be useful for the public, and more specifically movie fans, to look, out of curiosity, at the communities where their favourite actor/actress belongs or these studies can be seen as a complementary instrument for movie recommendations.

2 Methods and Theoretical Aspects

2.1 Construction of the network

A possible network that can be constructed using the available data on film and actors is a bipartite network. A bipartite network is a network where we have two different types of nodes, in our case the actors represent a set of nodes and the films in which they appeared are another set of nodes. We are interested in inferring connection between vertices of just one type. To achieve this goal with our bipartite graph, we create a one-mode projection of the graph, therefore obtaining graphs with only one type of vertexes. In this work we will consider the network where the nodes are the actors and two actors are linked if they share a common actor, obtained from the one-mode projection of the bipartite graph onto the actors alone.

2.2 Centrality measures

Once we have obtained these networks, the first thing that we want to do is to extract some of the measures that tell us something more about the network topology.Here we will study the following centrality measures:

1. *Degree centrality* The degree k_j of a node j is defined as the number of its neighbours, that is, nodes that share a common edge with it. In terms of the adjacency matrix[1]:

$$k_j = \sum_i a_{ij}$$

   It is the simplest and more intuitive measure, however, especially in the context of social networks, it can highlight the individuals who might have a more central role in the industry, more connections and experience, and therefore maybe more successful.

2. *Closeness centrality* The closeness centrality is defined as the mean distance from a vertex to other vertices, where with *distance* d_{ij} we mean the shortest past between two nodes:

$$l_j = \frac{1}{N} \sum_i d_{ij}$$

3. *Betweenness Centrality* The betweenness centrality for a node j is given by the number of shortest path $P_{mn}(j)$ passing through j with respect to the total number of shortest paths passing through the network:

$$BC_j = \frac{\#P_{mn}(j)}{\#P_{mn}}$$

[1] Here and in the following, the summation is referred to the all nodes in the network

2.3 Components, Communities and Clustering

In real-world networks, it is typical to find that our networks is not connected, but our graph is made up by more than one connected component and the path between different components do not exists. In particular, one finds a large components that fills most of the network, usually more than a half and many times over the 90%.
Therefore we will first check that our graph is not connected and then we will investigate its components. We expect the existence of a large component that contains the majority of links and to conduct our analysis using this particular component.
A common feature of social network is *transitivity*. Intuitively, we can explain transitivity in the following way: if x knows y and y knows z, then we have a path xyz of two edges in a network. The path is said to be closed if also x knows z, hence we have triangle in a network. In this context, we introduce the *clustering coefficient*, which can be defined as:

$$\mathcal{C} = \frac{(\#\text{of triangles} \times 6)}{(\#\text{of paths of length 2})}$$

Its value lies between zero and one.
Moreover social networks exhibit an *assortative* behaviour. With assortative mixing, we mean the tendency of an individual to associate with others whom they perceive as being similar to themselves in some way. In the opposite case, we say that the network shows a *disassortative* behaviour. In order to study this feature, we label the nodes using the network topological parameters and, in particular, a common choice is to label the nodes using their degree k. An empirical way to identify this pattern is to plot the degree of each node versus the the average degree connectivity,which is defined as the average nearest neighbor degree of nodes with degree k. We distinguish three cases:

1. If the curve is growing, the network is assortative
2. If the curve is decreasing, the network is disassortative
3. If the curve is flat, we don't recognize any relation

Quantitatively, we can measure the coefficient or assortative coefficient, which quantifies the assortative mixing by degree[6]:

$$r = \frac{\sum_{xy} xy(e_{xy} - a_x b_y)}{\sigma_a \sigma_b}$$

where e_{xy} is the fraction of all edges in the network that join together vertices with values x and y for the degree; a_x and b_y are, respectively, the fraction of edges that start and end at vertices with values x and y; σ_a and σ_b are the standard deviations of the distributions a_x and b_y. Studying assortative mixing by degree is particularly interesting because the fact that the degree, which is a structural property of the network can dictate the position of the other nodes gives rise to a *core/periphery structure* in the network, where the core is made of high-degree nodes and is surrounded by the periphery, which is a less dense structure made of nodes of lower degree. [7].

We define *communities* as a tightly connected set of nodes, with many internal edges and few edges with the remaining portion of the graph. When we have very large graphs, we rely on computer algorithms to find clusters in our network. A network can divide in large groups, in small ones or we can even find a mixture of different sizes. One of the most used methods is the *Louvain algorithm*, proposed by Blondel at al. in [3]. It aims to optimize *modularity*, which is mathematically defined in the following way:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where $2m$ is the sum of all entries in the adjacency matrix, A_{ij} is the adjacency matrix, $\delta(c_i, c_j)$ is 1 when i, j are in the same community and 0 otherwise.

The algorithm is based on two phases that are repeated iteratively.

1. We assign to each node a community. Once we have selected one node, we measure the modularity variation when moving to a neighboring community. We repeat this step until no further improvement of modularity can be achieved.

2. We build a new weighted network whose nodes are the communities that we have found in the first phase and where the weights are given by the sum of the links between the previous communities.

These steps are repeated until no increase of modularity is possible.

2.4 The Barabàsi-Albert model

In real world networks, such as the actor collaboration network, or the World Wide Web and the power grid networks, the probability $P(k)$ that a vertex in the network interacts with the other vertices decays as a power law following $P(k) \sim k^{-\gamma}$. This seems to show that large networks self-organize into a scale-free behaviour, which is not predicted by the random graph theory of Erdòs and Rènyi or the small-world theory by Watts and Strogatz. There are two key aspects that these models do not include. Firstly, the above models start from a fixed number of node, while real networks continually grow thanks to the addition of new nodes (for example, in our case, the actor network expands because of the release of new movies or newcomers). Secondly, the random network model assumes that we randomly choose the interactions between nodes, however in most real networks new nodes prefer to attach themselves to more connected nodes, a process that we call *preferential attachment* (in our case, the more movies an actor has played in, the more likely a casting director will get him/her a role in a new movie.)

The fact that growth and preferential attachment are both features of real world networks led to the birth of the *Barabàsi-Albert* model, also known as the *scale-free* model [2].

The model is defined as follows. We start m_0 nodes and the links between these node are chose arbitrarily, as long as each node has at least one link. Now we include the two key aspects mentioned above.

- *Growth*
  At each time step we add a new mode with $m\,(\leq m_0)$ links that connect the new mode to m nodes already in the network.

- *Preferential Attachment*
  The probability $\Pi(k)$ that a link of the new node connects to node i depends on the degree k_i as:

  $$\Pi(k_i) = \frac{k_i}{\sum_j k_j}$$

  This equation implies that if a new node has a choice between a degree-two node and a degree-four node, it is twice as likely that it connects to the degree-four node.

After t time steps the Barabàsi-Albert model generates a network with $N = t + m_0$ nodes and $m_0 + mt$ links. The network will have a power-law degree distribution with degree exponent $\gamma = 3$. Because of preferential attachment new nodes are more likely to connect to the more connected nodes than to the smaller nodes, therefore the larger nodes will acquire links at the expense of the smaller nodes, eventually becoming hubs. This result is known as the *rich-gets-richer phenomenon.*
Both growth and preferential attachment are needed to have the emergence of the scale-free behaviour.

2.5 Network Perturbation: Error and Attack

When we study a real network, it is usually interesting to study the response of the network to a perturbation, i.e. removal of nodes. In this work we will test the network response to a random (or *error*) and intentional (*attack*) perturbation. Taking inspiration from a work by R. Albert, H. Jeong and A.L. Barabàsi [1], we will study the change of the diameter as a function of the fraction of the removed nodes. The *diameter* d of a network is defined as the average length of the shortest paths between any two nodes in the network. It characterizes the ability of two nodes to communicate with each other therefore, in the context of social networks, it has a rather small value, hence reflecting the small-world feature of these types of networks.

3 Analysis

3.1 Data and Network Construction

To conduct our network analysis we will use the freely available datasets from the Internet Movie Dataset (IMBd). The structure and the type of information are reported in Appendix A. In particular we will use data from *title.basics.tsv* and *title.principles.tsv*, that contain respectively the basic information, such as title,year of release, genre, and so on, about any kind of audio/visual media product (not only movies, but also television series,podcasts,etc.) registered into the dataset. Each product is uniquely identified by an alphanumeric string. In the latter file, in each row, as far it concerns our work, we have the title identifier, a name identifier of a person that worked on the film in any capacity (actor, producer, director...) and the job on the film. To associate the name identifier to the actual name of the person we will be a function that reads this information from the dedicated file *name.basic.tsv*
The first thing that we have to do is to decide which films from the first file will be included in our analysis. The film industry is a business that is older than a century, henceforth we expect our network to have edges and nodes whose number are of order 10^6 or 10^5 respectively. Because of the limiting computer power available and the large number of nodes, performing some network analysis task can be quite cumbersome from a computation cost point of view. Therefore, throughout our analysis, we will specify whenever we select only movies that were produced starting from a given year or produced in a given country. The information about the country of production is contained in the file *title.akas.tsv.* Once we get the list of all the title identifiers, we will filter from the second file only actors/actresses that have worked in that title. We will carry out our analysis using Python programming language, in particular we will use the NetworkX library.
Once we get both the movies and the actors we can construct the bipartite graph. by performing the one-mode projection, we get the network representing the collaboration between the different actors. In this way we will have an edge between two different actors

Actor	k
Brahmanandam	0.002164
Eric Roberts	0.002016
Ron Jeremy	0.001795
Shakti Kapoor	0.001531
Mithun Chakraborty	0.001485
Mohan Joshi	0.001423
Raza Murad	0.001397
Nassar	0.001326
Kiran Kumar	0.001317
Avinash	0.001310

Actor	BC
Eric Roberts	0.058769
Michael Madsen	0.025280
Tom Sizemore	0.017464
Gulshan Grover	0.017293
Ron Jeremy	0.014297
John Savage	0.013228
Anupam Kher	0.012431
Om Puri	0.010341
Franco Nero	0.010294
Udo Kier	0.009418

Actor	CC
Eric Roberts	0.24452
Michael Madsen	0.24045
John Savage	0.23666
Tom Sizemore	0.23362
Harvey Keitel	0.23184
David Carradine	0.23179
Franco Nero	0.23162
Armand Assante	0.23089
Malcolm McDowell	0.22985
Willem Dafoe	0.22971

Table 1: Top 10 actors for centrality measure

if they both starred in the same movie.
At first, we consider the movies that have been produced since 1973 and from any country of production. The result is a network made of 592.160 actors and 426.294 movies, with 1.352.739 edges. Next we perform the one-mode projection onto the actors. After checking out that our network is not connected, hence showing that there are actors in different components that can't be reached, we focus on the connected components of the paths. In particular, we take the largest connected component and consider its associated graph. This new network has 432.920 nodes and 2.103.908 edges. In the following, we will consider it as the object of our analysis, since it contains more than 73% of our nodes and it helps us to obtain reliable results, in addition to reducing the computational cost.

3.2 Comparing the different centrality measures

In this paragraph we investigate what are the most important nodes in the network using the different centrality measures that we've presented in Section 2.2.
If we look at the size of our network it is immediately apparent that computing the betweenness centrality and the closeness centrality requires a lot of time and computational power since it has a complexity of $\mathcal{O}(nm)$ where n is the number of nodes and m is the number of edges. We can overcome this difficulty by taking a sample of nodes to estimate the betweenness centrality of all nodes. Again for the computation of closeness centrality, we will restrict the calculation to the top 1000 nodes, sorted by their betweenness centrality [5].
For each measure, we compute the top 10 most important people of the network. The results are presented in Table 1.

We see that, as expected, that we get different lists for centrality measures, with some actors appearing only once. We could ask ourselves: what is the best ranking list? It is known, and we can also see it on a small scale in our results, that closeness centrality suffers from the fact that the values are closely spaced between each other, while other measures, such as degree and betweenness centrality, don't have this kind of problem because of a wider dynamic range. It is particularly evident if we look at the betweenness centrality results, where we have a ratio of 6.24 between the first and the last actor, while for the closeness centrality case this ratio is just around 1.

3.3 Clustering, Community and Assortativity

Clustering coefficient We have said that one of the features of real networks is high transitivity and that transitivity can be quantified using the clustering coefficient.
We have obtained the distribution of the clustering coefficient for the nodes, which is presented in Figure 1, and the average clustering coefficient for the graph, which has a value of $C = 0.76$.

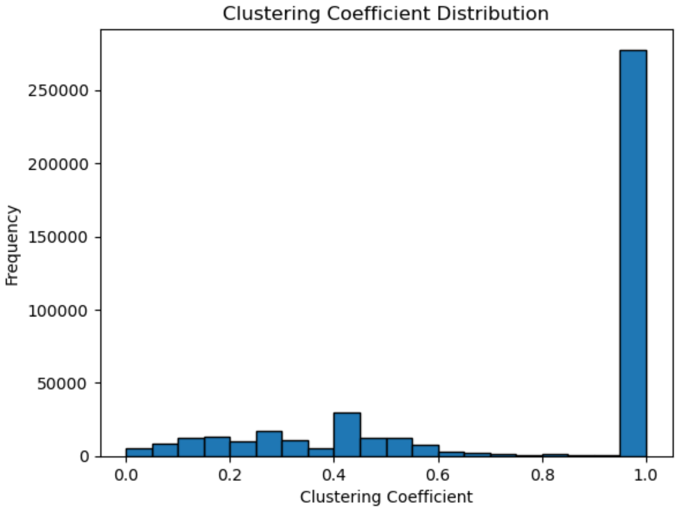

Figure 1: Clustering Coefficient

By looking at the distribution, we immediately see that the majority of the nodes has a high clustering coefficient, near to 1.

Community Detection Next, we apply the algorithm for community detection. We find 1279 communities whose distribution is presented in Figure 2. We see that the majority of the communities have a small size, we have very few units of communities with more than 10000 nodes and, in particular we have a very large community of around 70000 people.

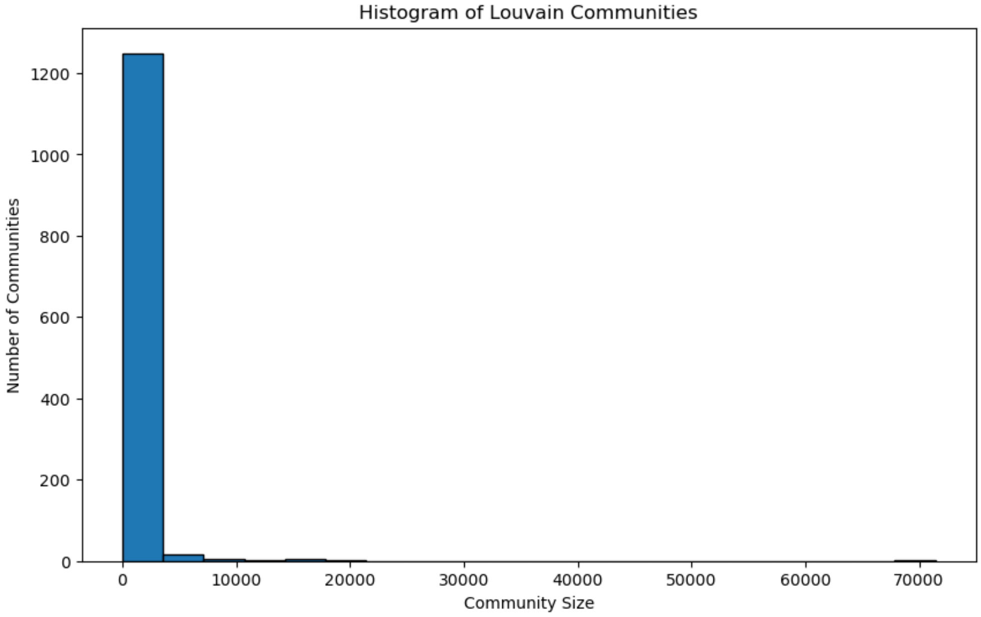

Figure 2: Louvain Communities

Assortativity To study assortativity, we first use a visual approach, by plotting the degree and the average degree connectivity (as defined in the Theoretical section 2.3). The result is presented in Figure 3.

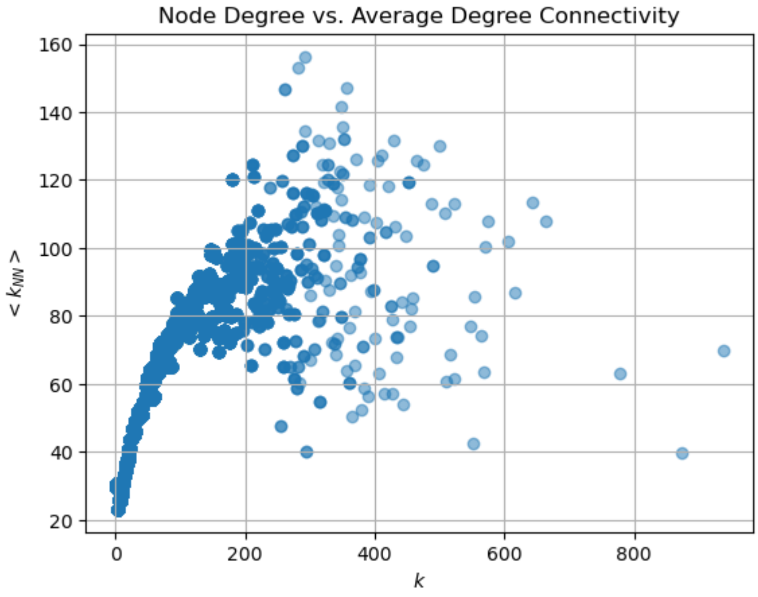

Figure 3: On assortativity

We can easily identify the growing curve that we expect in case of assortative mixing, but we notice an interesting scattered distribution of points in the plot for nodes with higher degree.
To have a quantitative answer, we compute the assortative coefficient, whose value is $r = 0.2467$, hence indicating that our network exhibits a weak assortative mixing pattern.

3.4 Analysis of the degree distribution and comparison with the Barabàsi-Albert model

One of the defining characteristic of a network is its *degree distribution*. We plot it for our degree distribution (Figure 4) using a logarithmic scale on both axis and we can see that the distribution follows a power law, except for a tail of few nodes with lowest degree.

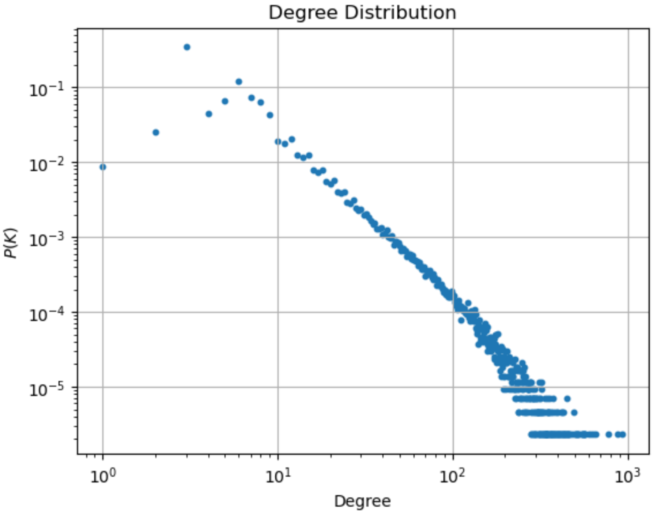

Figure 4: Degree distribution

By fitting the distribution, we find that $P(k) \sim k^{-\gamma_{actor}}$ where $\gamma_{actor} = 2.404 \pm 0.005$.

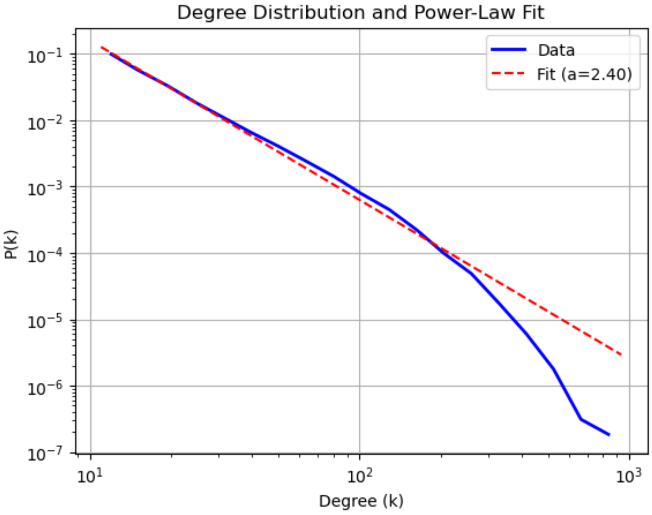

Figure 5: Fit results obtained using the *powerlaw* package in Python

As we have already anticipated, another famous model in literature where we observe this power-law scaling is the Barabàsi-Albert model. We want to investigate if we can draw other similar behaviors between these two networks. To this end we generate a Barabàsi-Albert graph with the same number of edges and links per node. The degree distribution, compared with the one of our network, is presented in Figure 6.

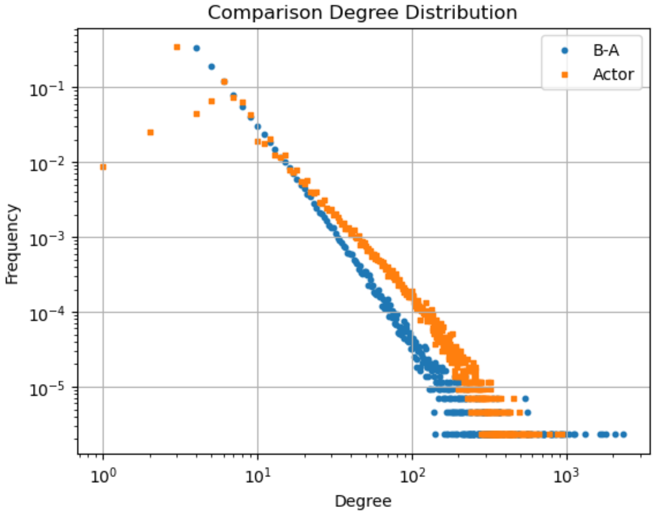

Figure 6: Degree distribution of both the Barabàsi-Albert model and actor collaboration network.

As we did for the actor network, we will first plot the k vs. $\langle k_{NN} \rangle$, reported in Figure 7, and then we compute the assortative coefficient, whose value is $r_{B-A} = -0.01$ and it is really close to zero. Both the plot and the coefficient point to the fact that this network shows no preference regarding the assortative/disassortative mixing.

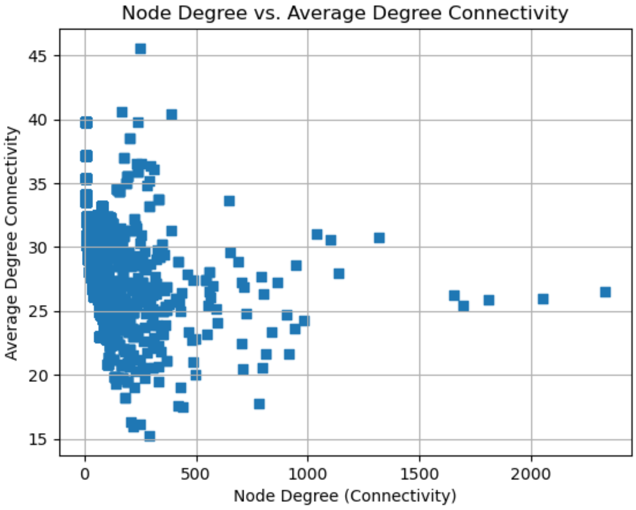

Figure 7: On assoratívity for the Barabàsi-Albert model

Finally, we compute the average clustering coefficient of the graph: $\mathcal{C}_{\mathcal{B}-\mathcal{A}} = 0.0030$, which tells us that the network has low transitivity.
These study show that our Barabàsi-Albert network captures the scale-free behaviour of the real network, however it can't reproduce other aspects, such as the assortative behaviour and the high transitivity.

3.5 Network Perturbation

For this study we will restrict ourselves to construct the network of the films produced in Great Britain in the last 20 years. The reason is two-fold: on the one hand we ensure

to study the tolerance of the network without worrying about the effects of ageing of the network (actors that started their career in the first decades of the network and are probably dead at the beginning of the 21st century) and, on the other hand, our goal is to study the changes in the diameter of the network, which is computationally expensive, depending on the number of nodes and edges. In the end, following the same procedure as above, we will obtain a network with 12717 nodes and 33636 edges.
To study the error tolerance, we remove a random sample of 300 nodes for our network, while for the attack tolerance we remove a sample of 50 nodes, which are chosen by taking the top nodes ranked by their degree centrality measure. The results are presented in Figure 8.

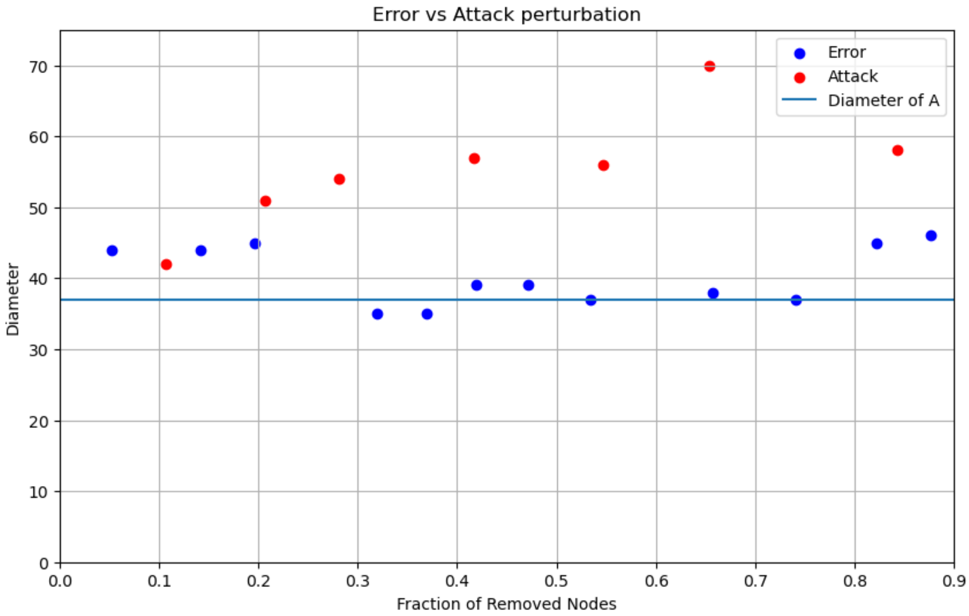

Figure 8: Error and attack tolerance

We immediately see that the network shows a good tolerance to errors, while we have a distinguishable increase in the diameter when we perform an attack,although it is not as sharp as expected from the comparison with other scale-free networks presented in [1]. This might be due by our initial choices in the network construction(country, time period), maybe enlightening some underlining dynamic in the UK movie industry between the nodes in the core/periphery and that can be better understood by linking it to the studies on the core/periphery interaction in the network of film creators (which has been studies in some cases, see for example [4]). However we can't exclude that this results might be flawed because of a non-complete categorization of the original country of production the movies registered in the dataset. Despite these observations, we still think that these results can be consider compatible with the analogous studies on scale-free networks, whose robustness to these perturbations is deeply rooted in the non-homogeneity of the degree distribution. Therefore here we see in action the crucial importance of the power-law degree distribution. It is quite understandable that if we remove the actors that have worked with more people, it is more difficult to reach out to other actors. When we are left with a smaller number of actors, it is also more difficult to find actors that connect nodes that are far apart. Instead, when we remove nodes at random, it is unlikely to remove actors that have high connectivity, and hence, that have a more central role in the topology of the network.

4 Conclusions

In this work we have seen that the actor collaboration network shares many of the common features of real social networks, such as high transitivity and the scale-free connectivity distribution, and we have shed a light on its topological structure by studying the most important nodes in the network. In addition, we have also shown the differences and the similarities with the well studied Barabàsi-Albert model, highlighting the fact that, despite having a similar power-law connectivity distribution, they differ in other features, such as the assortative mixing and transitivity. We have also studied the dynamics on a network made of actors who have worked in UK movie productions via a removal of nodes and highlighted the possible reasons behind the peculiar evolution. Overall, this project allowed us to study and gain a greater insight into the properties of real social networks, category in which our networks belongs.

Our work can can be developed in many other directions. For example we could study the changes in the structure of the network over different time periods or explore in more detail the formation of communities (according to country or genres) or identification of communities using different algorithms, and study the assortative mixing by other scalar characteristics, such as race,gender,age. Moreover, we could try to make predictions on the success of movies, using the ratings from the IMDb dataset and additional data from box offices, award ceremonies and similar, based, for example, on the interaction between the core and the periphery (like for example proposed in [4] for the network of movie creators). In conclusion, this report can be a good starting point for many other studies, that could be useful or interesting to both the public and insiders of the movie industry, in addition to its scientific value in the area of network theory.

References

[1] Réka Albert, Hawoong Jeong, and Albert-László Barabási. "Error and attack tolerance of complex networks". In: *Nature* 406.6794 (July 2000), pp. 378–382. DOI: `10.1038/35019019`. URL: `https://doi.org/10.1038%2F35019019`.

[2] Albert-Làszlò Barabàsi and Rèka Albert. "Emergence of Scaling in Random Networks". In: *Science* 286.5439 (Oct. 1999), pp. 509–512. DOI: `10.1126/science.286.5439.509`. URL: `https://doi.org/10.1126%2Fscience.286.5439.509`.

[3] Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (Oct. 2008), P10008. DOI: `10.1088/1742-5468/2008/10/p10008`. URL: `https://doi.org/10.1088%2F1742-5468%2F2008%2F10%2Fp10008`.

[4] Lengyel B Juhász S Tóth G. "rokering the core and the periphery: Creative success and collaboration networks in the film industry." In: *PloS* (2020).

[5] Rhyd Lewis. *Who is the Centre of the Movie Universe? Using Python and NetworkX to Analyse the Social Network of Movie Stars.* 2020. arXiv: `2002.11103 [cs.SI]`.

[6] M. E. J. Newman. "Mixing patterns in networks". In: *Physical Review E* 67.2 (Feb. 2003). DOI: `10.1103/physreve.67.026126`. URL: `https://doi.org/10.1103%2Fphysreve.67.026126`.

[7] Mark Newman. *Networks: An Introduction*. USA: Oxford University Press, Inc., 2010. ISBN: 0199206651.

A The IMBd Dataset

The Internet Movie Dataset (IMBd) is an online dataset of information related to any type of entertainment product (films, television series, home video, podcast, video games, short movies and so on). The users of the site also play an active role by suggesting new material or edits or by rating the product on a scale from one to ten.

IMDb datasets, which are daily updated, are available publicly to customers for personal and non-commercial use.

Each dataset is contained in a gzipped, tab-separated-values (TSV) formatted file in the UTF-8 character set. The first line in each file contains headers that describe what is in each column. A '$/N$' is used to denote that a particular field is missing or null for that title/name. The available datasets are as follows:

title.akas.tsv.gz

- titleId (string) - a tconst, an alphanumeric unique identifier of the title
- ordering (integer) – a number to uniquely identify rows for a given titleId
- title (string) – the localized title
- region (string) - the region for this version of the title
- language (string) - the language of the title
- types (array) - Enumerated set of attributes for this alternative title. One or more of the following: "alternative", "dvd", "festival", "tv", "video", "working", "original", "imdbDisplay".
- attributes (array) - Additional terms to describe this alternative title, not enumerated
- isOriginalTitle (boolean) – 0: not original title; 1: original title

title.basics.tsv.gz

- tconst (string) - alphanumeric unique identifier of the title
- titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)
- primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release
- originalTitle (string) - original title, in the original language
- isAdult (boolean) - 0: non-adult title; 1: adult title
- startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year
- endYear (YYYY) – TV Series end year. '$/N$' for all other title types
- runtimeMinutes – primary runtime of the title, in minutes
- genres (string array) – includes up to three genres associated with the title

title.principals.tsv.gz

- tconst (string) - alphanumeric unique identifier of the title
- ordering (integer) – a number to uniquely identify rows for a given titleId
- nconst (string) - alphanumeric unique identifier of the name/person
- category (string) - the category of job that person was in
- job (string) - the specific job title if applicable, else ’/N’
- characters (string) - the name of the character played if applicable, else ’/N’

name.basics.tsv.gz

- nconst (string) - alphanumeric unique identifier of the name/person
- primaryName (string)– name by which the person is most often credited
- birthYear – in YYYY format
- deathYear – in YYYY format if applicable, else ’/N’
- primaryProfession (array of strings)– the top-3 professions of the person
- knownForTitles (array of tconsts) – titles the person is known for