Networks Science project relation

Marco Edoardo Santimaria
912404

June 29, 2023

Contents

1 Network Metrics

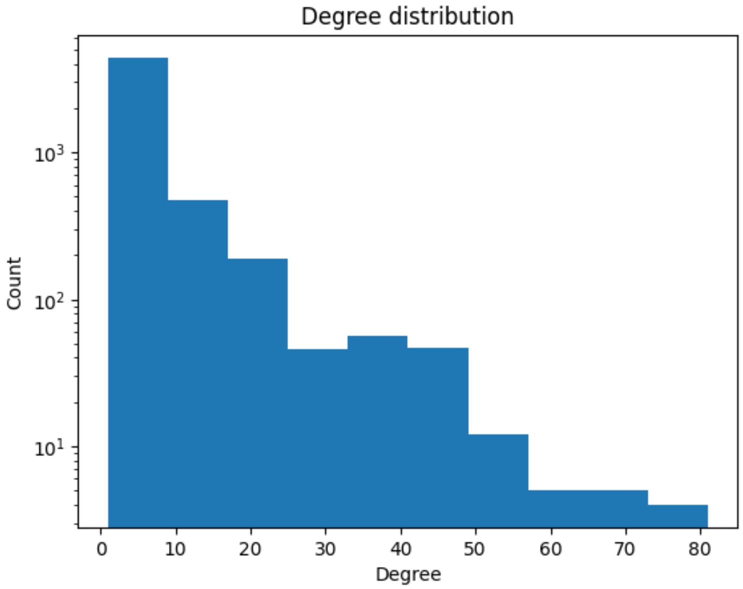
The chosen network is called "General Relativity and Quantum Cosmology collaboration network" and it is available at the following link SNAP Stanford. As reported on the source website, «Arxiv GR-QC (General Relativity and Quantum Cosmology) collaboration network is from the e-print arXiv and covers scientific collaborations between authors papers submitted to General Relativity and Quantum Cosmology category. If an author i co-authored a paper with author j, the graph contains a undirected edge from i to j. If the paper is co-authored by k authors this generates a completely connected (sub)graph on k nodes.».

1.1 Graph size: number of nodes and number of edges

In the graph there are **5242 nodes** and **14496 edges** (verified from snap website).

1.2 Node degree and degree distribution

The degree distribution is plotted as follows:



The degree distribution is not very broad, as the graph did not required the logarithmic scale in the x axis. This would suggest that the network does not have a borad distribution in the degrees. To check that, I computed the heterogenity parameter for my graph wich evaluated to $\kappa = 0.0006$, wich confirms that the degree distribution is not broad, and that my network is not heavy tailed.

1.3 Connected components

In my graph there are 355 connected components. The biggest has a size of 4158 nodes (verified from snap website). Most of the connected components have few nodes (less than 10), but for the 10 biggest, the following table summarizes the average size for connected component size(1)

Since most of the nodes (around 80 %) are in the bisggest comnnected component, the next tasks will be carried on on the biggest connected component.

Component size	$< k >$
4158	6.4589
14	4.1429
12	3.3333
10	2.0
9	3.45
8	3.67
7	3.29

Table 1: This table shows the average degree for component size. This has been calculated by averaging the average degree of the top connected cmomponent grouped by the component size

1.4 Centrality of each node

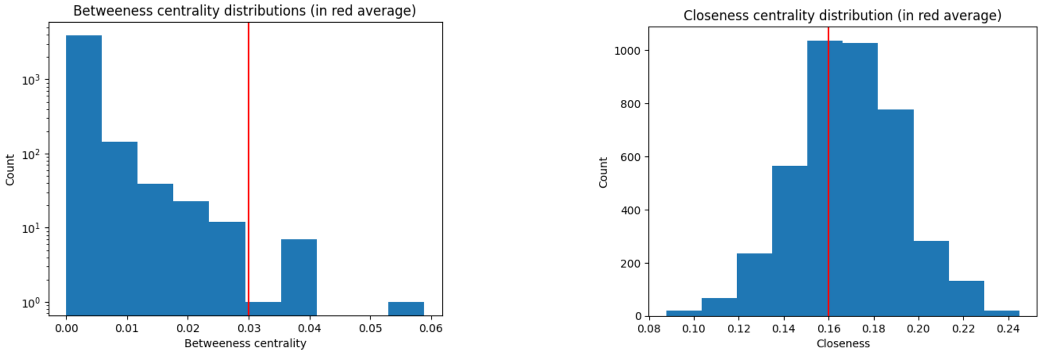
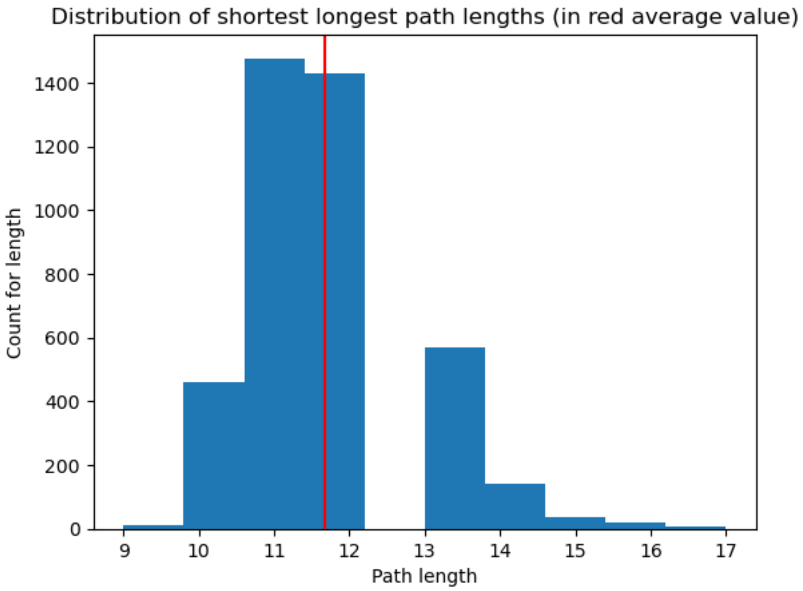


Figure 1: Centrality measures distributions

The betweeness distribution reflects the fact that my network is not heay tailed. Altough it is possible to see the "usual shape" of a heavy tailed distribution, it is not, having a very small heterogenity parameter.
The Closeness centrality instead follows a gaussian distribution. This means that all of the nodes are mostly close to the average distance, which is expected as it is a reflection of the small world propetry that has been verified later in this chapter.

1.5 Longest shortest path and graph diameter

The graph diameter is 17 (verified with data from snap website). The calculation on the shortest longest path had to be done on the biggest connected component, as otherwise we would get an infinte length.



By plotting the distribution of the shortest longest path, and the average path lenght (red vertical line), we can observe that most of the path have a length that is very close to the average path length. We also have that this graph is Small World as $\frac{<l>}{\log |N|} = 0.726$ but it is not ultra small world as $\frac{<l>}{\log \log |N|} = 2.853$

1.6 Clustering coefficient of every node

For this network we have that the average clustring coefficent is 0.529635811052136

1.7 Conclusion

To summarize, I have a small world network that is heavy tailed in the degree distribution, with the biggest component that has most of the nodes, and a high clustering coefficent.

2 Community detection

Community detection has been done on the biggest connected component, as there i had most of my nodes. To detect whether there are communities in my graph, i used several approaches:

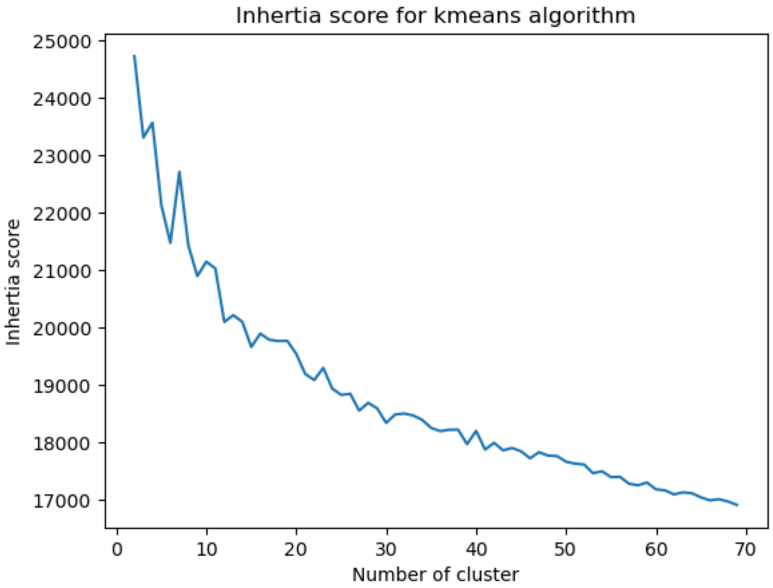
- Louvain algorithm
- Greedy modularity optimization
- Kmeans
- Hierarchical clustering

2.1 Louvain and greedy modularity optimization

Louvain algorithm (which had the same results than the greedy modularity) said that there are around 40 communities, and 70 communities according to the greedy modularity optimization algorithm. This results should not be taken into account, as the same result (for the Louvain algorithm) has been obtained on a random erdosh reiny graph with the same size (which is an expected behaviour). Moreover, there was something strange as the resoult obtained with other algoriths were in a different range of values.

2.2 Kmeans

According to Kmeans (performed on a matrix build with the eigenvectors of the adjacency matrix), we have around 2 communities, using the inerthia score as a score of how good a community assignement is. This was again extremely strange to me as the result was not in the same (or around) order of magnitude of the Louvain and greedy modularity optimization algorithm.
To obtain this result, Kmeans (from the scikit library) was run 69 times (from 2 to 70 communities) and from each assignement, the inhertia score has been calculated, and then plotted. The highest value of the score, the better the assignement is.



2.3 Hierarchical clustering

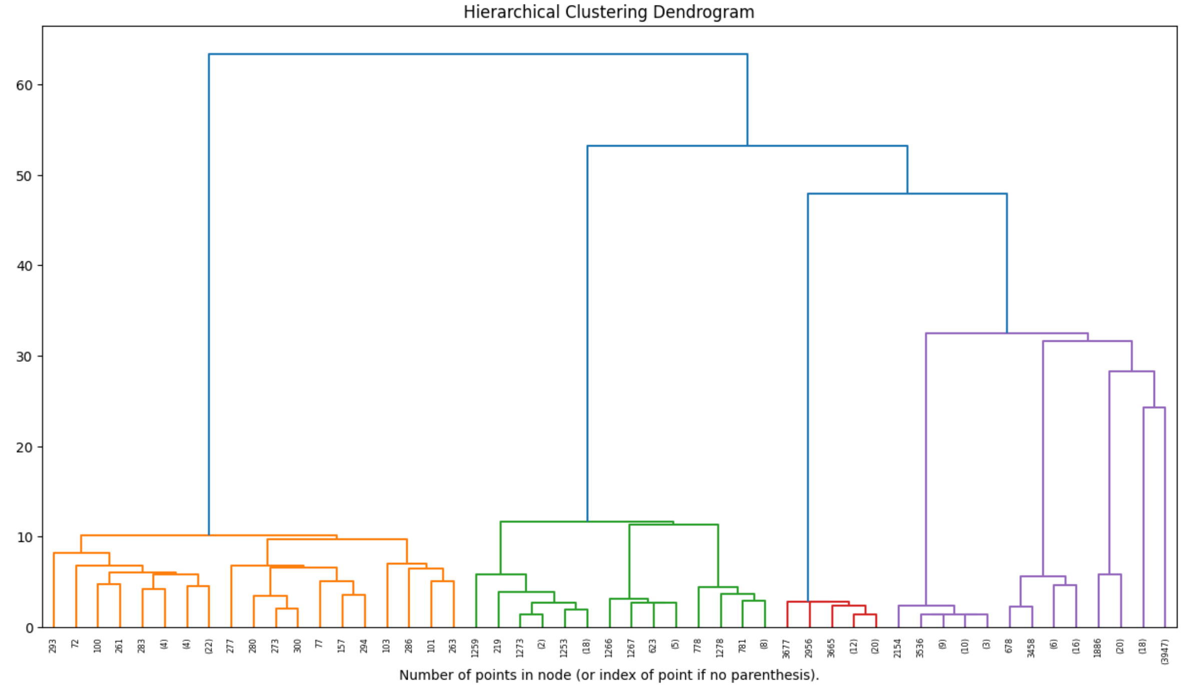
At this point I ran the Hierarchical clustering to try to see whether there is a hierarchical structure on the graph. By plotting the dendogram, it is possible that there is at leas one massive community of around 4k nodes (the oone in the bottom right). There are also a number of other, smaller, communities. By chosing to cut the dendogram just above the point in wich the green subcommunity are merged all toghether, we obtain 8 communities that have a hierarchical structure.

A side note: I choose to not consider non connected components as communities. This is mainly for two reasons:

- The first is the fact that most of thoose non connected components are just a few nodes.
- The second reason is due more to the interpretation of the graph: Having just a link between two nodes, means that only a paper has ever been published between the two authors. Having a connected component with just a couple of nodes, means that it is most likely that the have published just a paper in astrophisic, and in my opinion, this is not enough to consider ths a community.

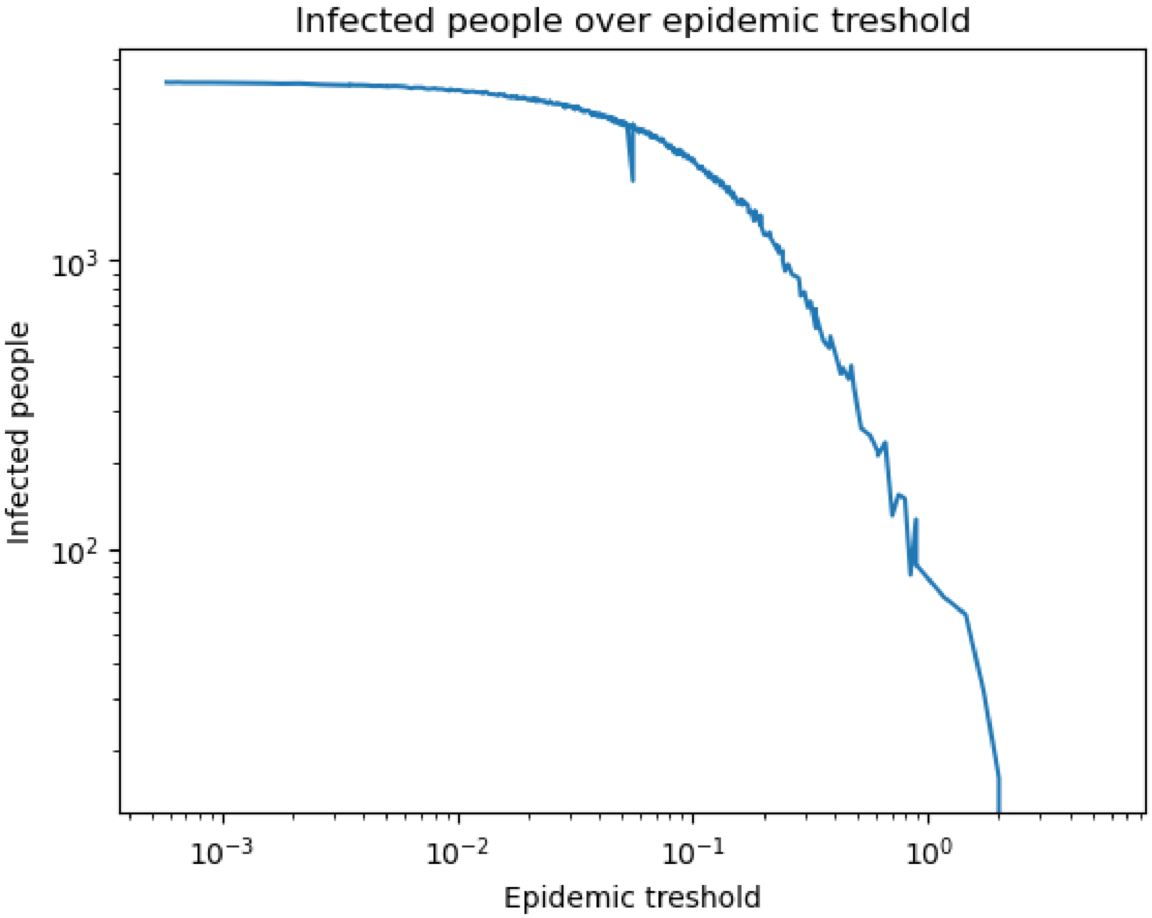
Hence, community detection has been carried out only on the biggest connected component, completely ignoring the non connected components.

In conclusion, I have discovered that my graph has around 10 communities thet have a hierarchical structure.



3 Epidemic simulation

The choosen epidemic model was the SIS model. The reason for choosing the sis model is quite simple: first of all, whenever we talk about interactions of papers publishing, it does not make much sense to assume that once two people have collaborated, they do not collaborate anymore (SIR model). The second reason is that for the SIS model I managed to find examples of the python code, wich made it way more simple to write the custom code for the simulation. The community assignement used, is obtained from the louvain algorithm (which is not the best, but is the fastest to compute, and having had to run the simulation multiple times, i choose to have an approximation of my communities rather than an exact community assignement). After validating the code, by running the epidemic simulation manually several times, I started to plot the number of infected people, in function of the epidemic treshold. As can be seen on the graph, the higher the value for the epidemic treshold, the lower the number of infected people. This is totally expected as the higher the epidemic treshold, the harder it is for an epidemic to spread (hence the number of infected people remains small.



Having estrablished that the hishest the epidemic treshold, the harder it is for an epidemic to spread, I proceeded to simulate what happen if i were to remove the HUBS from my network. I removed the nodes that have a degree higher than 16 and after the pruning of the hubs, the network has 3793 nodes. The result of the simulations can be summarized by the following plots 2

What can be observed from the results is that even with no hubs, the epidemic still spreads, but manages to infect less people than with hubs. This is expected as by removing the hubs (the more central nodes in the network), we make it harder for informations (hence the epidemic) to spread in the network. This is the reason why hubs are called superspreader!

At this point, I proceeded to see what would happen if I were to start the epidemics with K infected nodes in a single community VS starting wit a single infected node in K communities. The results are

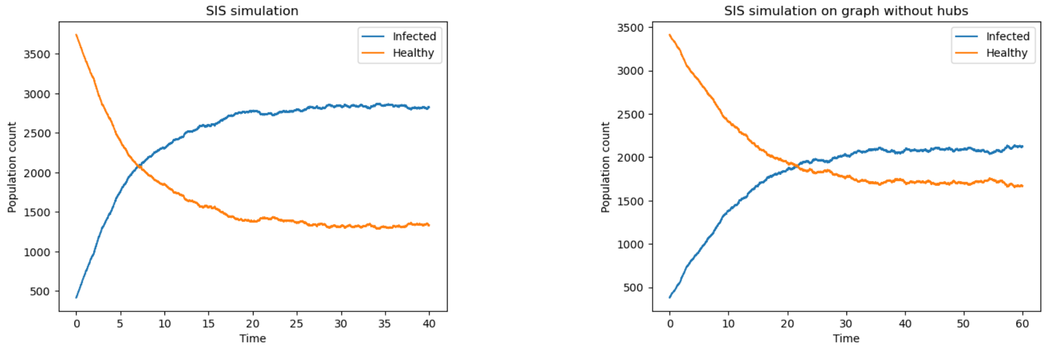


Figure 2: Comparison of epidemic evolution with and without hubs

shown below:

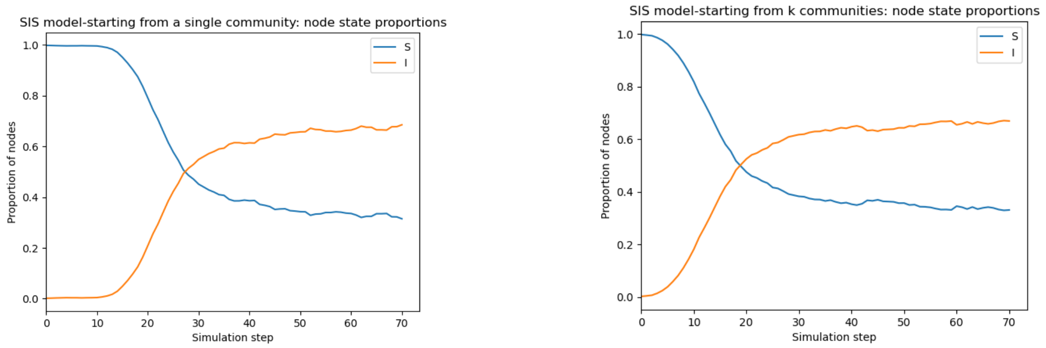


Figure 3: Epidemic evolution starting from 1 community with k nodes infected vs starting from k communities with 1 node infected

The result shows that the epidemics at the end will be mostly equal, but what changes is the time at wich the number of infected people surpass the number of sane people. We can see that the configuration in which a single node is infected in K communities reches this treshold faster than in the K infected individuals in 1 community configuration. This is expected for the following reason: if we suppose that in a single step a infected node infects a hub (which is not an absurd thing to suppose), we would get that at the second step, we would have at least K infected hubs distributed in K communities, making the spreading way more faster that a configuration in wich we have all infected nodes in a single community (moreover the spreading between communities is way harder than spreaing inside a community due to the fact that in epidemic spreading, community structure is a barrier to the spreading of the epidemic).