Stock Market Evolution - A Network Perspective

Complex and Social Networks Project
Alberto Latino

1. Introduction

The Stock Market is a system that evolves over time and it is influenced by many different factors. Several approaches have been used to predict its trend but it is still a very hard task. This research-oriented project aims at analyzing the evolution of the stock market from a network perspective, researching for interesting trends and interactions of stocks over the analyzed period of time. The S&P 500 index has been analyzed by looking at interactions between stocks and a new clustering over time approach is presented below. Further measures regarding interaction between sectors and the influence of stocks have been measured and commented on as well as other network metrics.

2. Analysis of Data

Time Series of stocks belonging to the S&P 500 index have been analyzed from January **2016** to January **2022.**. The period involves some relevant events that affected the stock market such as the Covid 19 impact that happened in March 2020. Data has been downloaded by using the API of Yahoo Finance: for each stock it was possible to retrieve data of each stock for each trading day. In particular data from Yahoo Finance contained the day, ticker, opening price, closing price, adjusted price. Moreover, further data has been downloaded from the platform Datahub.io in order to retrieve other information such as the company name and its sector for each analyzed ticker.

The plot above shows the time series of the S&P 500 index divided by eleven sectors : **Information Technology, Health Care, Financials, Consumer Discretionary, Communication Services, Industrials, Consumer Staples, Energy, Utilities, Real Estate, and Materials**. Data from Yahoo Finance are transformed by using the log returns, this way prices are normalized. This is the formula that has been used:

$$r_t = log(P_t / P_{t-1}) = log(P_t) - log(P_{t-1})$$

3. Correlation Measures

After having transformed the dataset with the log returns a correlation measure has been computed in order to understand whether some stocks changed their price in correlation with some other stocks. Two main correlation measures have been considered: **Pearson Correlation** and **Distance Correlation.**

- **Pearson Correlation**

$$r = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - m_x)^2 \sum (y - m_y)^2}}$$

Given two stocks, Pearson correlation coefficient is a measure of linear correlation between two sets of data with results' values in a range of [-1,1] where values close to -1 mean that the two stocks are negatively correlated, values close to zero means that the two stocks are uncorrelated and values close to 1 means the two stocks are strongly and positively correlated. The drawback of Pearson correlation is that it only catches linear correlations and for the purpose of the network construction it needs to be normalized in a [0,1] range.

- **Distance Correlation**

The distance correlation of two random variables is obtained by dividing their distance covariance by the product of their distance standard deviations.

$$\mathrm{dCor}(X, Y) = \frac{\mathrm{dCov}^2(X, Y)}{\sqrt{\mathrm{dVar}(X)\ \mathrm{dVar}(Y)}},$$

A better and cutting edge correlation measure, distance correlation, has been used instead. It has the power of catching non-linear correlation by computing values in a [0,1] range. Values close to 0 mean that the variables are uncorrelated and in case of distance correlation it also implies independence. Values close to 1 means strong are strongly correlated.

4. Network Design

Once the correlation matrix of stocks has been computed the network has been constructed by using iGraph package as follows:

- **Nodes:** each node in the network corresponds to a stock of the S&P 500 Index. For the network analysis purpose, each node was enriched with further information such as the **ticker, sector** and a **color.**

- **Edges:** an edge between two nodes is added if the correlation between their associated stocks is above a certain threshold.

It is important to mention that the correlation between stocks is computed over a certain period of time. In this project, the data set has been analyzed in temporal sections and for each section a network has been built by looking at correlation results computed over that time window. Therefore, at the end of the network construction phase a sequence of networks is available to be analyzed, each one representing a sort of snapshot of the market at different moments.

By reading papers and researching the state of art of this kind of analysis, it has been found out that most of the approaches used a fixed threshold. One of the novelties of this project is the implementation of a dynamic threshold that changes over time with evolution of the network.

5. Choice of Threshold

The choice of the threshold that determines whether or not there is an edge between two stocks has been one of the most crucial and important steps of the project. Several approaches have been experimented.

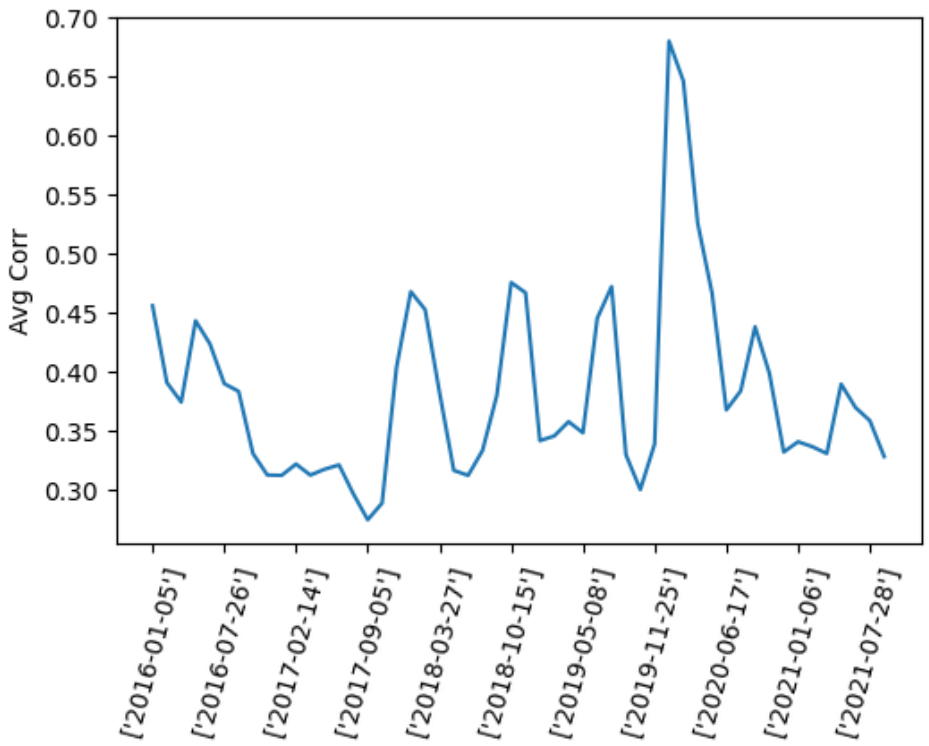
- **Fixed Threshold:** a first approach was inspired by the current state of art of this kind of analysis, that is choosing a fixed threshold of 0.7 as suggested by several papers. In this case the fixed threshold was not able to catch the full dynamics of the market since in periods of low average correlation the network resulted to be sparse and disconnected. Since the goal was to find relationships between stocks relative to a particular period of time, also the threshold has to be adapted to the particular timeframe.

- **Dynamic Threshold - Modularity Maximization:** this approach aims at building a network with different thresholds ranging from 0.6 to 0.95 for each snapshot. For each time window, the threshold was chosen in order to maximize the clustering modularity for that period of time. Modularity maximization was used since one of the objectives of this project was to find out whether stocks behave in groups and modularity measures the strength of division of a network into modules. Indirectly by setting a threshold that maximizes modularity will force to remove edges among weakly correlated pairs of stocks. It has been found out that in most of the time windows, the threshold that maximized modularity was very high and hence favored the sparsity of the network and the creation of many small clusters. Since it is known that in most of the cases modularity favors small clusters against large ones, some penalization terms have been considered: the use of **split penalty** and the use of **locality modularity optimization**. The most difficult part of considering these two penalized measures was that they were computed after clustering was already performed. The correct approach would have been to internally change the optimized function inside of the clustering algorithm, but it required such a long time to do so. A simpler but still effective penalisation has been applied in the end, that is, upper limiting the threshold. This way sparse networks with small clusters, characterized by the choice of a high threshold, were forced to have a smaller threshold hence favoring the creation of bigger clusters.

- **Dynamic Threshold - Mean Correlation:**
  This approach tried to create a threshold based on the particular characteristics of the stock market at that moment. The goal was to understand if in certain time windows the market was in general strongly or weakly correlated. This way, the threshold automatically adapted considering the general behavior of the market. So for each snapshot, the mean correlation was computed and the threshold was constructed afterwards. For example, if the mean correlation for a given snapshot is 0.35 the threshold was setted to a factor K times

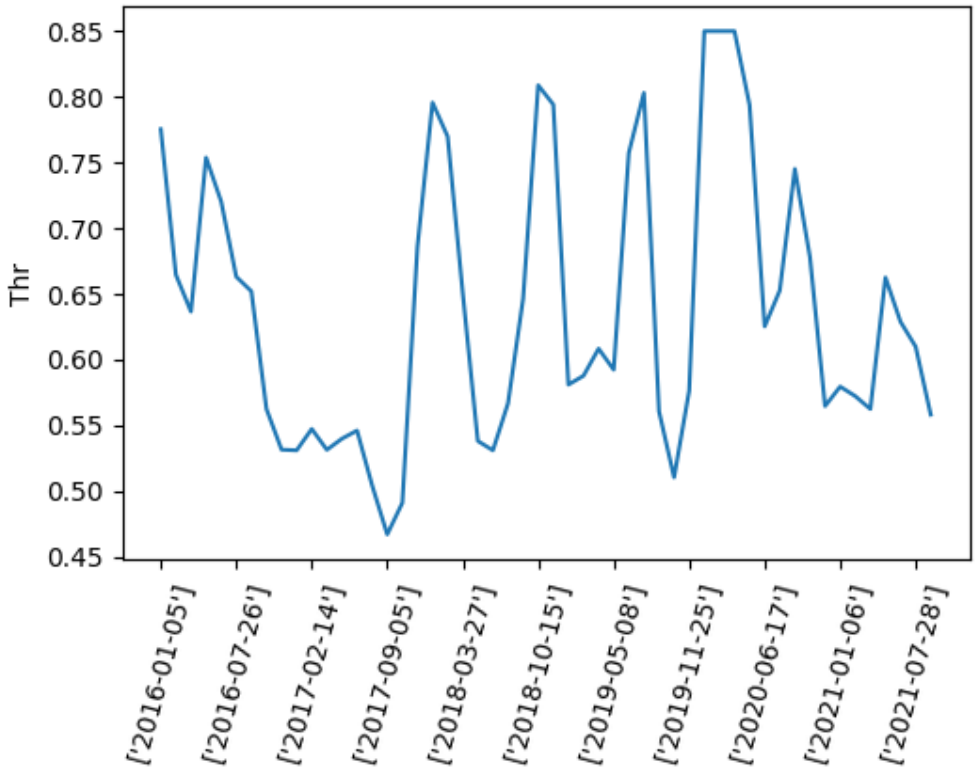
greater. A factor of K=2 would have been too high, and for network in which the mean correlation is around 0.45 an edge would have been added only for correlations above 0.9. At the same time, values of 1.5 returned too dense networks, so an average value of 1.7 has been chosen .This way stocks are connected if their correlation is above the mean correlation of that span of time. In every case, since it has been noticed that higher thresholds led to sparse networks, an upper limit has been set to 0.85 to penalize the creation of many small clusters at the same time. The introduction of this limit has been suggested by the previous experiment of modularity optimization.

Plot of Average Correlation over Time



As it can be noticed, average correlation between stocks of the S&P 500, changes over time. The highest the correlation, the more returns are similar to each other. The big peak corresponds to Covid19 crash of March 2020 when the price of the vast majority of the stocks dramatically decreased.

Plot of Dynamic Threshold over Time



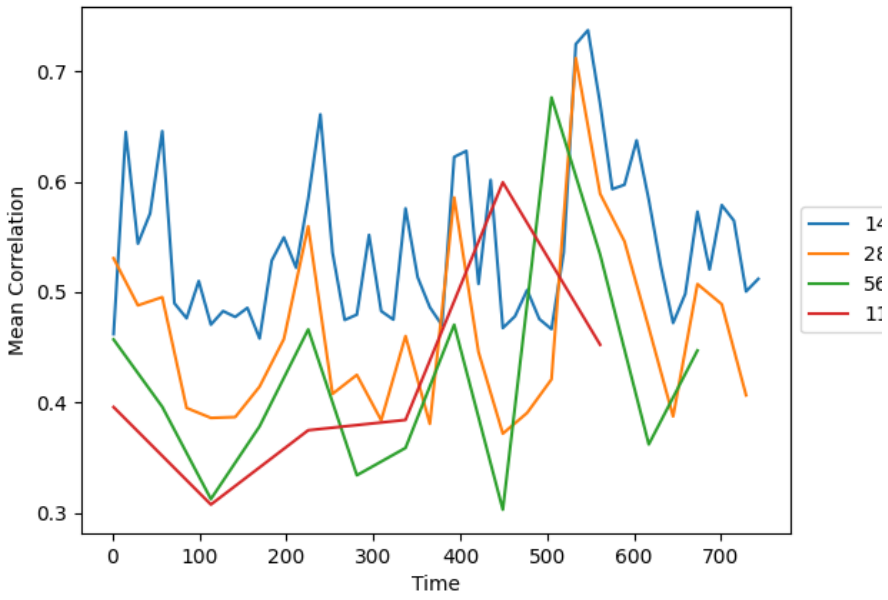
As mentioned above, the threshold dynamically adapts with the current dynamics of the market. During the Covid19 crash, the correlation was quite high and the threshold adapted itself to this event.

6. Choice of Sliding Window and Overlap

Another crucial step of this research has been the choice of the time window's width. The window width corresponds to the number of daily returns included in the computation of the correlation between stocks. Several window widths of 14, 28, 56, 112 days have been tried and the one that better captured the dynamics of the market was chosen. What has been found out is that smaller side windows are very noisy and can be more sensitive to random fluctuations of the market. On the other hand, big size windows were too big to catch

interesting patterns hidden in the data. In terms of Machine Learning, it could be said that small size windows overfit the data while big size ones resulted in underfitting. By visual inspecting the results a window of 56 days, approximately 2 months, has been chosen since it resulted to be a good tradeoff between the two opposite behaviors.

One of the main objectives of this project is to perform temporal clustering and by reading some state of the art research it was found out that one important characteristic of temporal clustering is the so-called **smoothness.** It means that clusters should evolve smoothly over time without having unexpected and abrupt changes. Obviously, smoothness cannot be guaranteed anymore in case of market crashes. In order to achieve this characteristic, consecutive time windows have been overlapped over time. It means that in the computation of correlation, some information is shared among consecutive snapshots. The more information is shared, the more will be the smoothness of temporal clustering. In this case an overlap of 50% of the window length has been chosen, that is 28 days. Blue:14 days. Yellow:28 days ,Green:56 days ,Red128 days.



7. Evolution Data Structure

After that all the above mentioned considerations have been applied, the resulting networks have been gathered all together in order to perform further analysis. In particular, a class “Snapshot” has been created:

```
class Snapshot:
   def __init__(self, start, end, threshold, mean_correlation,
network, clusters):
       self.start = start
       self.end = end
       self.threshold = threshold
       self.mean_correlation = mean_correlation
       self.network = network
       self.clusters = clusters
```

It contains information regarding the span of time the network has been built on and other information such as the chosen threshold, the network and a collection of clusters sorted by their size in decrescent order. As already mentioned, each node contains the information regarding its belonging sector and each cluster has an index. Assigning an index to clusters is useful when clusters are tracked over time.

8. Temporal Clustering

Temporal Clustering is one of the core objectives of this project, several approaches have been implemented to track the evolution of clusters over time. The main key to have evolution tracking of clusters is to connect clusters between consecutive snapshots, so clusters need to be associated with each other over the analyzed span of time. Clusters have been associated with each other by using a similarity measure, in particular the **Jaccard Similarity** has been used.

8.1. Jaccard Similarity

Jaccard Similarity is a statistic used for gauging the similarity and diversity of sample sets. It measures similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

In this particular case, A and B are two clusters of stocks. A Jaccard similarity of one means that the two analyzed clusters are completely overlapping, so all the stocks in A are in B as well, while a value of zero

means the two clusters do not resemble each other in any way.

8.2. Tracking Clusters over Time

The goal of Temporal Clustering is to associate clusters from different snapshots with one another. The main idea is that each cluster **Ci(t)** at time t, is associated with the cluster **Cj(t+1)** that has the highest Jaccard Similarity with **Ci(t).** So, each cluster is associated with the one that resembles it the most. Two main approaches have been tried:

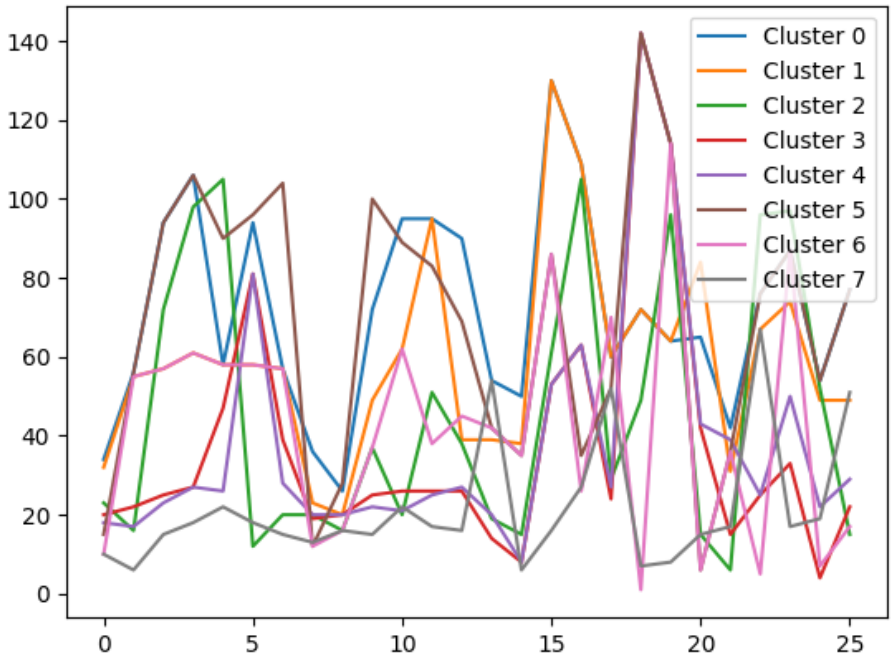
- **Track Clusters of consecutive snapshots**
  Considering the snapshot T and the snapshot T+1, the N biggest clusters of snapshot T were tracked in snapshot T+1. The following function,
  `def track_clusters(subgraphs_t1: list, subgraphs_t2: list)`
  takes as input a set of clusters from T and a set of clusters from T+1 and maps them with each other by returning the indices of clusters in T+1 that resemble the most the clusters in T. This way it was possible to track clusters in consecutive snapshots. The main drawback of this approach is that in case of market crashes, many stocks are clustered together and it means that clusters from the previous snapshot are mapped to the big cluster, called for clarity **C**. At the next iteration, since all the N biggest clusters were mapped to **C,** from this moment on they will be all associated with the cluster that **C** resembles the most. This way the algorithm is not able to catch information regarding clusters that split over time. This drawback could be solved by starting the tracking again after a market crash.

- **Track Clusters from the beginning**

  This approach aims at tracking the N biggest clusters from snapshot T=0 to the end of the span of time. This approach resulted in catching merges of clusters over time and then new splits. The limits of this approach is that the number of clusters that can be tracked is limited and it cannot catch the creation of completely new clusters. Actually, by looking at the results stocks seemed to cluster together mainly by sector and since sectors are fixed in number this approach seemed to be quite effective. A consideration that aims at solving the monitoring of new created clusters is done in the future work section.

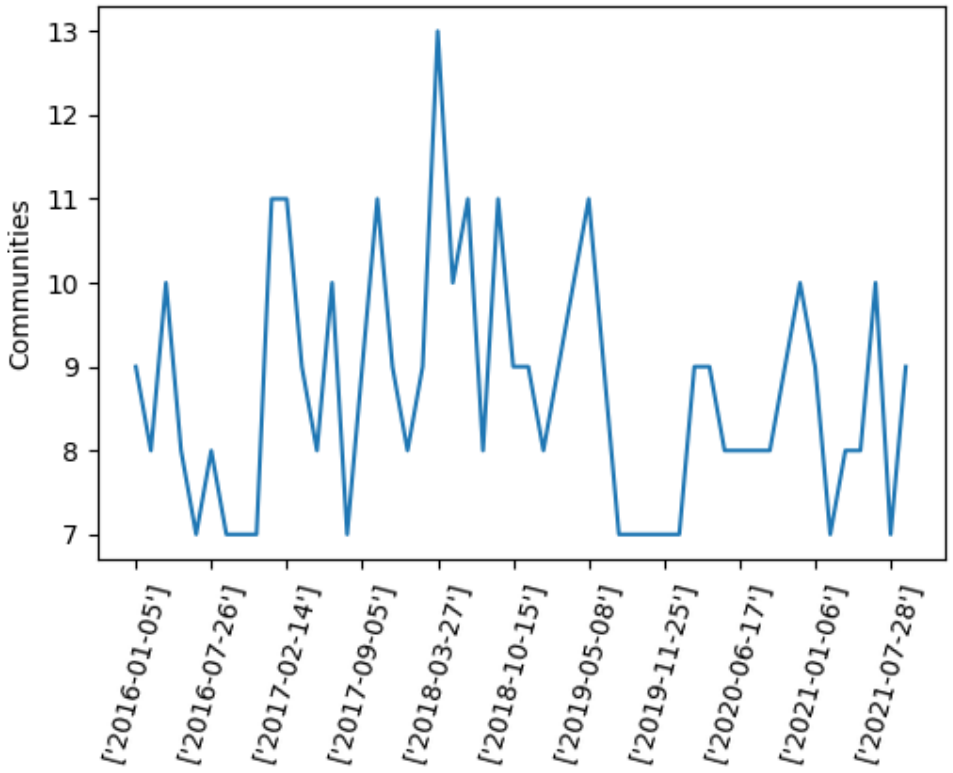
From this plot it can be seen the correspondence of clusters indexes over time. Indices are relative to the network of a specific network, so their values do not give much information. The purpose of this plot is to show that when two clusters' indices converge to the same index, then it means they merged. Following the same reasoning, it can be seen when a cluster splits in multiple clusters.

Plot of Cluster Sizes over time



This plot shows the evolution of the size of tracked clusters over time. Since the tracked clusters are only eight, it can be assumed that it is very unlikely that they have the same size over time. Therefore, whenever the lines go down together, it can be assumed that more clusters merge together. A plot of indexes is not inserted since with this assumption we merge the two informations in this plot and the document is kept shorter.

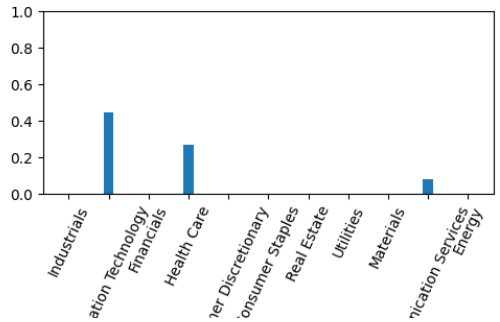
Plot number of Communities over time



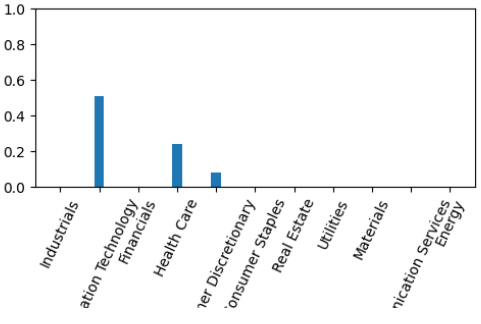
It can be seen that the number of detected communities evolves over time. More communities means that the network is divided in many groups of stocks correlated with each other internally. Few clusters usually indicate a general common behavior of the stocks. It can be noticed as in the Covid19 crash, the number of detected communities dramatically decreased. This is because in that period stocks had mostly the same behavior.

Example of Sectors evolution of a Cluster

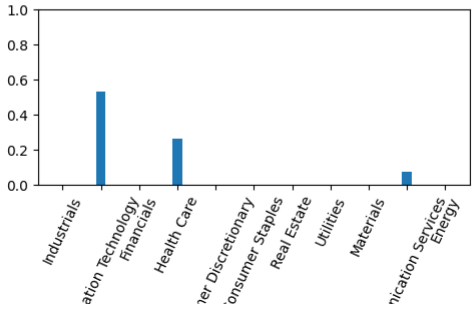
CLUSTER 1- match cluster- 1, TIME = 0



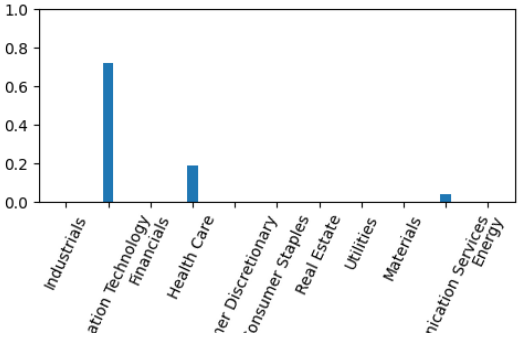
CLUSTER 1- match cluster- 1, TIME = 1



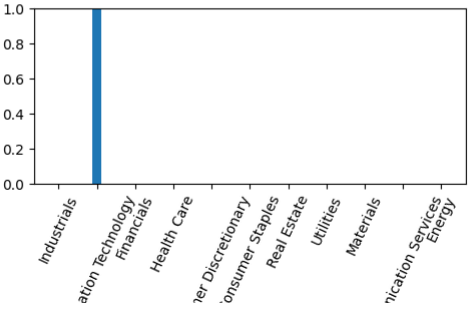
CLUSTER 1- match cluster- 1, TIME = 2



CLUSTER 1- match cluster- 2, TIME = 3



CLUSTER 1- match cluster- 4, TIME = 4

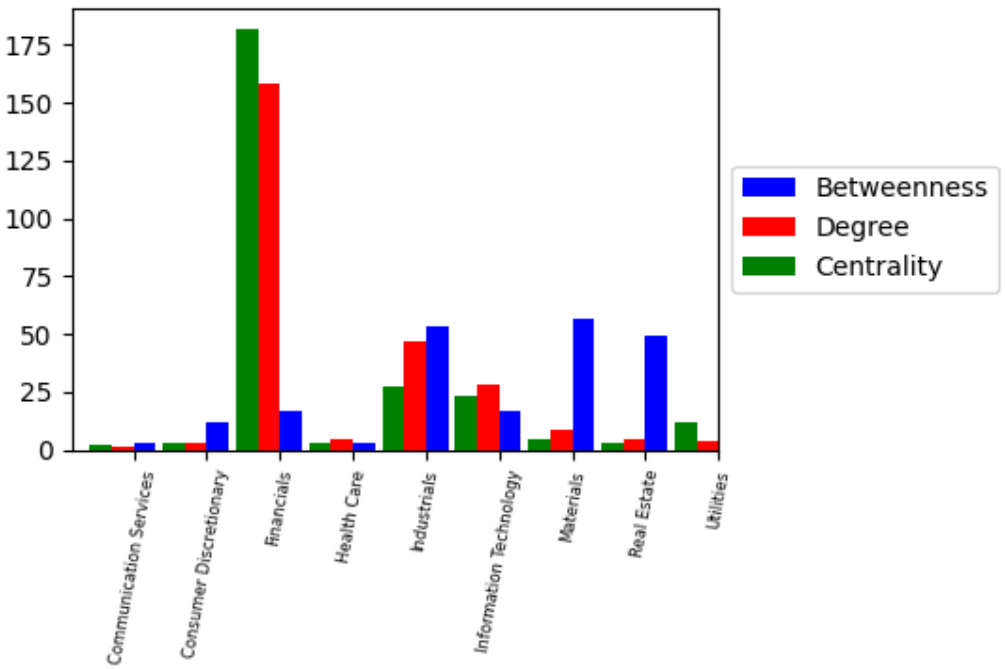


In the plot it is possible to visualize cluster number 3 matching other clusters at different snapshots. The evolution of the sectors in the clusters shows that the IT sector is the predominant one and over time it represents the vast majority of the stocks in the cluster. During the evolution, also Financials and Communication Services appear in the cluster.

9. Influent Nodes

An interesting analysis was to discover what are the nodes that mostly lead the market and to find out the “leader” stocks in several clusters. This could be useful for several trading purposes. Influent nodes can be defined from different points of view, for example by considering nodes with the highest degree, betweenness and eigenvector centrality. Here, it has been analyzed what are the favored nodes for each one of the proposed metrics.

9.1. Comparing metrics of Importance



The plot shows that Financial stocks are considered influential by eigenvector centrality and degree, it seems they have a relatively low betweenness. This could be justified because financial stocks perfectly cluster together in big clusters, but they are isolated in their own community. In fact, usually there is a very low presence of other sectors

in financial clusters. The fact that Financial stocks are the most influential has an underlying meaning: they are structurally dependent on what happens in other sectors, since they strongly rely on investment activities

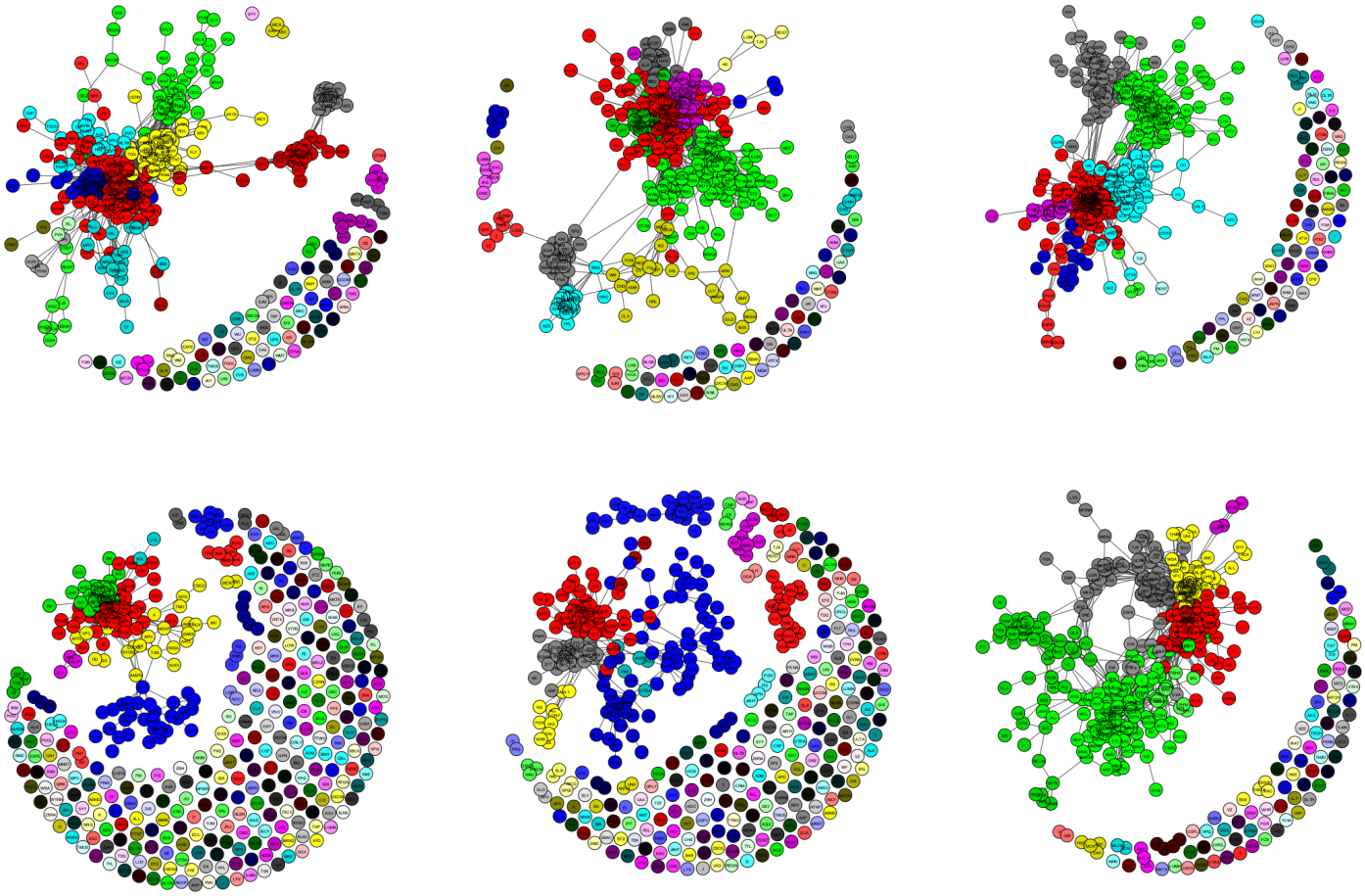
10. Results

In this section are reported other general results obtained during the experiment. Some results regarding the temporal clustering are reported below as well as other evolving metrics of the network.

10.1. Clustering over Time

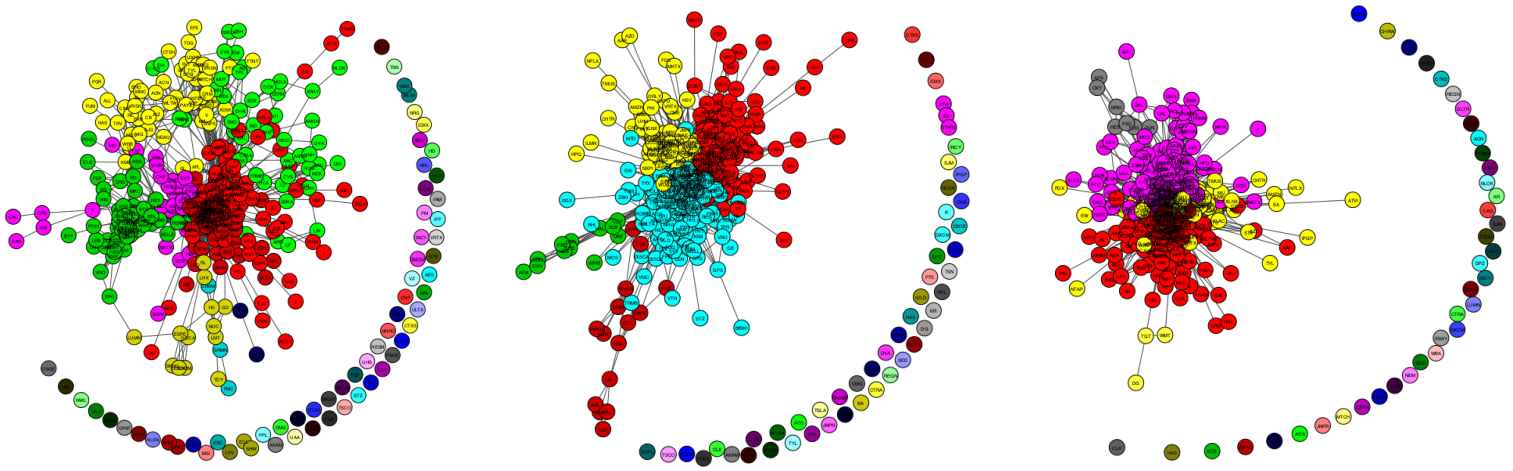
One of the most important results obtained is that stocks tend to cluster together by sector. Percentages of sectors in clusters have been analyzed and it resulted that IT, Financial, Energy, Real Estate and Utilities stocks tend to have a strong correlation with each other. This result can make sense since these sectors are very specific. On the other hand, sectors as Industrials, that includes a way more diverse sub-categories, resulted to be spread among several clusters. By looking at the composition of tracked clusters over time it has been noticed that clusters kept their composition over time. In particular, if a cluster is made mainly by IT stocks, it will keep following this trend over time. It has been noticed that, during particular events that destabilize the market, these sectoral clusters are merged with other clusters related to other sectors. After these ‘traumatic events’ the market stabilizes and starts evolving again with sector-oriented clusters.

Market Snapshots in normal condition



During this phase of the market, there is a consistent number of detected communities, each one mainly clustered by sector. In this phase the market is pretty stable

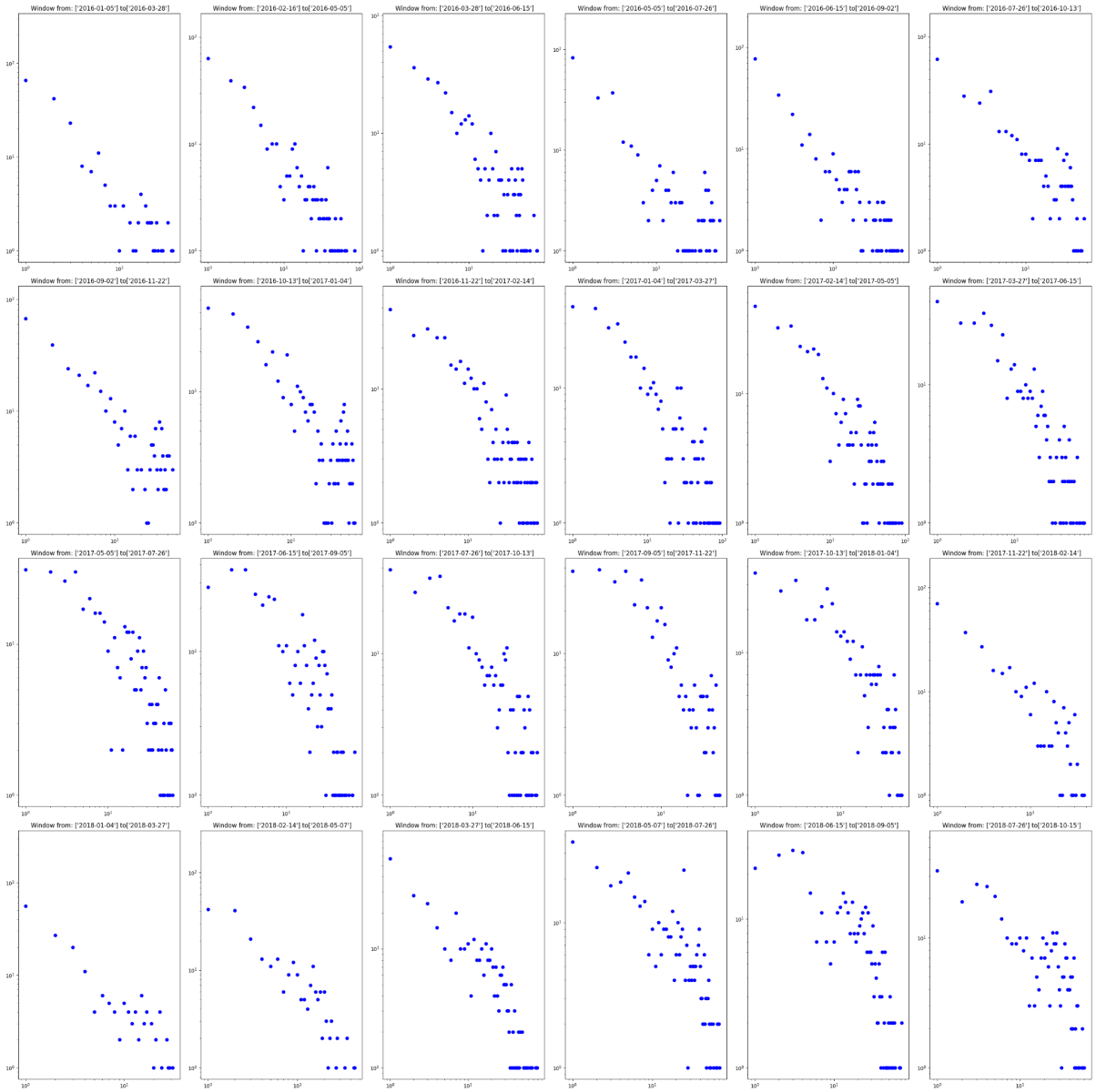
Market Snapshots in market crash



These are consecutive snapshots of the COVID 19 crash in March 2020. It can be seen that the network implodes and is concentrated, creating few strongly correlated clusters. The nodes that were uncorrelated with this event and that were affected less belonged mostly to the Health Care sector.

10.2. Degree Distribution over Time

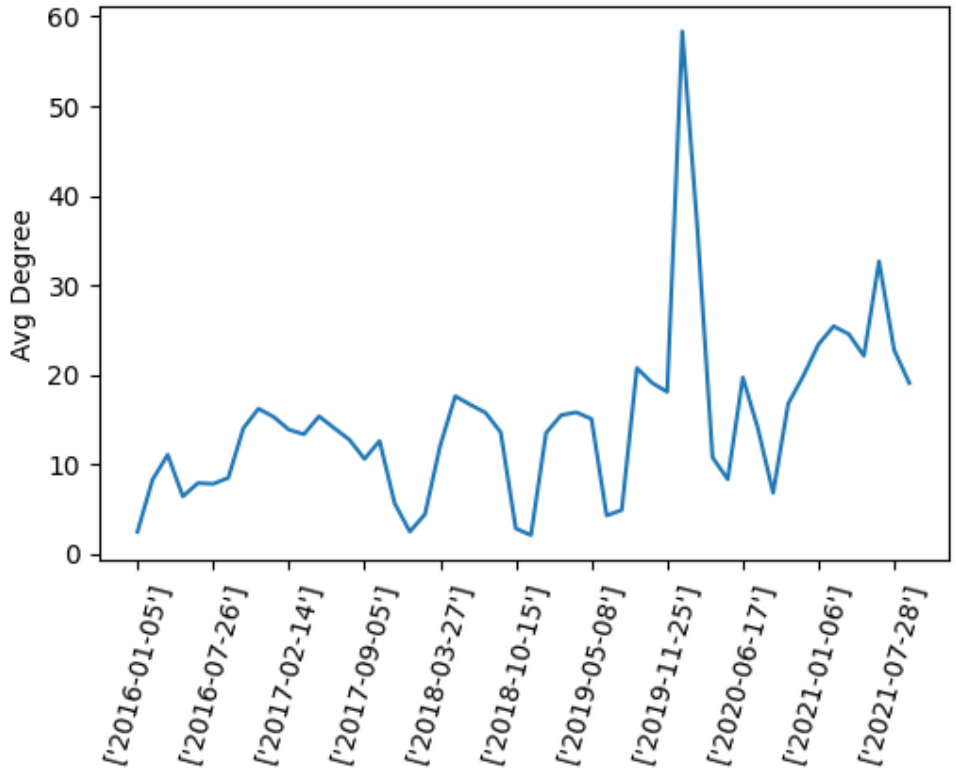
The degree distribution has been plotted for multiple time windows in a log-log plot. In most of the windows the network shows scale free properties and it follows a power-law distribution.



The plot above shows the degree distribution over time of some of the analyzed snapshots. It clearly shows a scale - free behavior.

10.3. Average Degree over Time

For each time window has been computed the average degree of the network and it is evident that it evolves over time since the stock market is a dynamic system. It can be noticed that for highly correlated networks the degree evolved over time. This can be noticed in the peak related to the Covid19 crash.



11. Covid Market Crash Analysis

In the analyzed time series, data regarding the market crash of March 2020 caused by the Covid19 world disease have been considered, in order to understand how the market behaves in these situations. The main consideration that can be done, from the clustering point of view, is that all the stocks tend to cluster together. This makes sense because generally when a market crash happens many stocks go down together. Only a few stocks were not included in this big negative trend: some health-care stocks. This can be justified since pharmaceutical stocks had the potential to make profit by finding a vaccine for the disease. Other characteristics that have been noted is that the mean correlation, the average degree, dramatically increased in this period. After some months, the market stabilized again and its stock started clustering again mostly by sector.

12. Future Work

This research oriented project followed a very experimental approach, exploring the application of networks' theory in the field of Finance. Some approaches worked and some others did not. In this section are mentioned future works in particular regarding the temporal clustering section and the optimization of modularity. The tracking analysis over time can be improved by reasoning at node level: by considering every single node as a single cluster of size 1, it can be monitored whenever it is clustered with other nodes. Then a threshold on the size of the cluster can be set in order to announce the birth of a new cluster whenever this threshold is overcome. This way also newborn clusters can be monitored from the beginning of their existence. Regarding the modularity optimization, an approach would be to internally modify the optimized function, inserting the above mentioned penalization terms. Also, since visual inspection benefits clustering analysis, the usage of visual instruments such as Gephi could be used to better use the results of this project for trading and financial purposes. This could help visualize when specific stocks leave clusters and go to others. It could help to predict cyclic trends over time, and this could contribute to the continuous challenge of predicting such a mysterious system, being the stock market.