Network Analysis Project Report

Stefano Ciapponi, Artificial Intelligence

25/11/2022

1 Abstract

The aim of this project is to apply Network Analysis techniques to analyse **Graphs** extracted from data concerning the **European Economic Area** (EEA) countries' Public Debt evolution over time.
We are also interested in analysing how these graphs' topologies change over time, how correlation impacts the network topology, and how they relate to another graph built from the EEA countries borders, to understand *in which context* the generated network topology becomes closer to it.

The project's iter is the following:

- We'll first explain the technique used to extract networks built upon debt correlation taken from a Time Series Dataset gathered from **Eurostat**.
- Then we'll apply some general network measures on the different graphs to find the most relevant ones.
- Finally, we'll apply some further measures/analyses on the most relevant graphs to see differences in their topology.

The data analysis notebook is available for download at the following github repository link: Notebook.

2 Context

General Field: Economy
Specific Application: Public Debt Quantitative Analysis

3 Problem and Motivation

This Project could make us able to better understand both:

- the dynamics of the evolution of public debt in a timespan which includes the beginning of a global crisis (end of 2019/ beginning of 2020)
- how public debts relate to each other in an economic setting which is governed by the same basic rules (free movement of persons, goods, services, and capital), like the EEA is.

This could also serve as an input for further analysis done by individuals with a stronger economic background, which could tackle the problem from either a macroeconomic perspective or a political science one.

4 Datasets

The main (and only) datasource for this project has been Eurostat (European Statistical Office). The Data, which consists in a publicly available **Time Series** concerning the evolution of the European Economic Area Public Debt over time, has been downloaded from the Directorate-General Database.

It is expressed in **Public Debt/GDP** Format, which is more relevant than looking at just the Public debt itself, since it's more representative of each country economic situation at each timespan.

The Timeseries is sampled at Quarterly timesteps (every three months) and Figure 1 shows a heatmap plot for visualization.

As we can see there's a quick change of hue starting from the *2019-Q4* quarter which represents the start of the Covid Pandemic and a general raise of public debt for all countries taken into consideration.

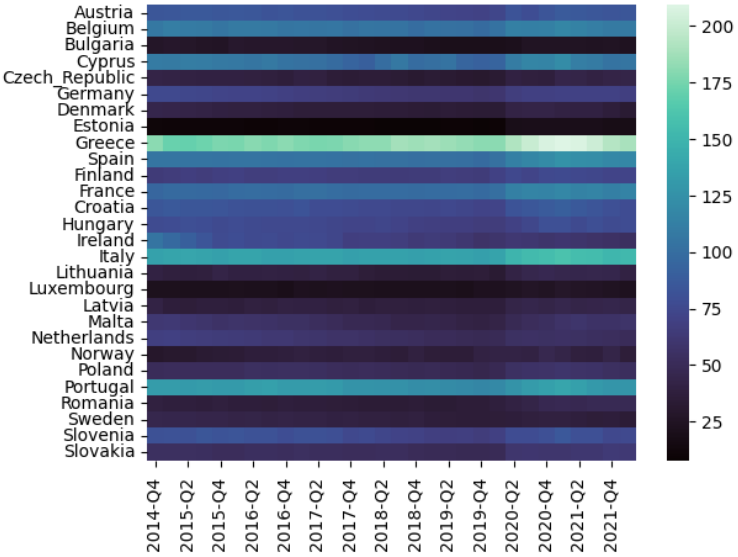

Figure 1: Time Series Heatmap

The Dataset has then been processed (using Python) with techniques inspired by Matesenz Et al. paper, which analyses debt correlation during the Great Recession:

- **First**: The Dataset is tested using a Shapiro-Wilk Test for normality: The *confidence interval* is fixed at 0.05 and 46% of the country timeseries null hypotheses can't be rejected (*p-value* $\leq$ *confidence interval*).

  The distribution is therefore assumed to be normal as a whole, since we are working 30 samples (Central Limit theorem).

  The test output ρ is $\geq 69.4\%$ when the *p-value* $\leq 5\%$, therefore the former will be taken as a threshold for test values, since we know that outputs greater than that one will be taken as statistically significant.

- **Then**, since we suppose the data distribution to be Gaussian, the dataset is used to compute Pearson Correlation Coefficient for samples for each pair of countries. The test is applied to sliding windows 4 quarters wide (1 year) and outputs 24 $N \times N$ correlation matrices, where N is the number of countries.

  The matrix is then *Thresholded* and values which are $\geq \rho$ are set to 1, while the others to 0 (we are only considering positive correlation because it makes so that networks are less interconnected and makes the computation less heavy).

  A plotted example can be seen in Figure 2.

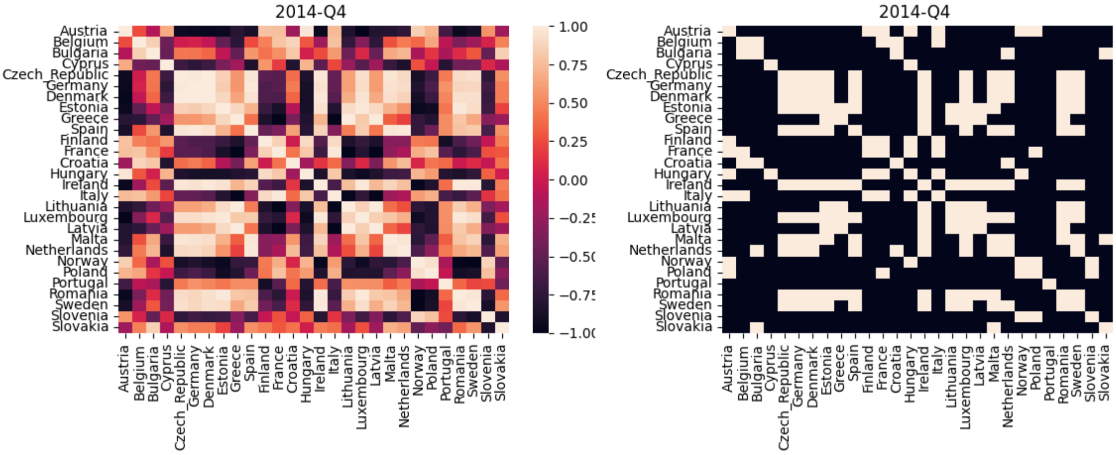

Figure 2: Correlation Matrix and its Thresholding

- **Finally**, the thresholded matrices are then used as adjacency matrices for unidirected graphs (computed using the NetworkX library).

  Example in Figure 3.

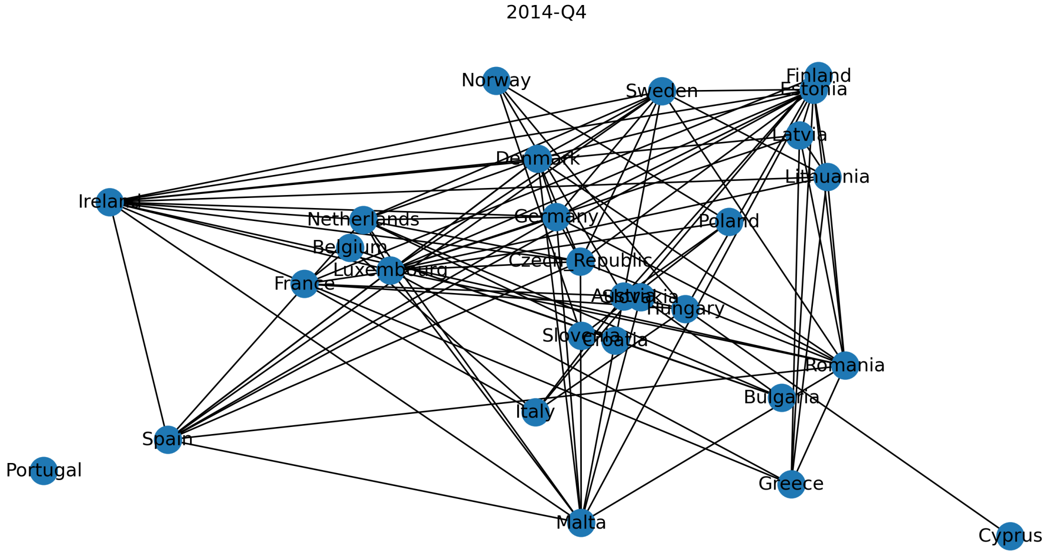

Figure 3: Graph computed starting from the Thresholded matrix in Figure 2

After Computing all 24 graphs NetworkX was used to compute general Measures on all of them to select some of the most relevant ones.
An extensive explanation of the measures applied and the reasons why they were applied is available in the Measures section.
Moreover, a graph based on both *land and sea borders* of the countries taken into examination has been computed querying Wikidata SPARQL API. This will be taken as the input to the computation of some measures.

5 Validity and Reliability

5.1 Data Validity and Reliability

The Data has been collected by a Directorate-General Agency (Eurostat).
We were interested in analysing the evolution of public debt of different countries and the data only consists in Public Debt/GDP values.
We personally didn't take part of the data selection process, but, given it's provenance, it should be considered **reliable** and **valid**.

5.2 Topology Construction/Graph Analysis Validity and Reliability

The graph topology construction process should be considered **reliable** since it's taken from a published paper which has been reviewed by a research committee.
It's **validity** depends on the statistical tests performed: since we are working in a data analysis setting, the more data is gathered the more a phenomena can be approximated and the topology can be considered valid.
From an Analysis point of view: the validity and reliability of the analysis process should take into account that some approximate methods have been used. This will be further explained in the Measures and Results sections.

6 Measures

We decided to apply some Global Measures on all 24 constructed graphs:

- **Number of Edges**: Counting the total number of edges in each graph is a direct way to check how much countries' public debts are correlated (this is due to the graph construction, since the presence of an edge between two countries means that their debts are correlated). As expected the graph *2019-Q4*, which uses data from the start of 2020 during the computation is the one with the highest amount of edges, since all public debts raise in a similar fashion in that timespan.

- **Connected Components**: Checking the number of connected components in each graph is relevant to understand how graphs are divided into groups in a certain timespan. In this case, most connected components are actually outliers from a general bigger graph, which considers most of the graph nodes.

- **Graph Edit Distance w.r.t. Border graph**: This metric is used to answer our question: *How is the Countries' Border graph related to the correlational ones?*

To answer this question a, approximated variant of GED has been applied to all graphs (Hussdorf) using the package gmatch4py. That's because the GED algorithm is notoriously a quite complex computational problem. The results plot can be visualized in Figure 4. The x axis represents the time evolution of the graph, while the y relates to the GED seen as a distance value (normalized between 0 and 1).

The Correlational graph found to be closer to the Borders one is **2019-Q4**.

- **Average Network Clustering Coefficient**: This measure is the average of the node Clustering Coefficient for each network. Locally it quantifies how much the neighbours of a node are close to being a *clique*. Computing an average on all the nodes shows how interconnected the network is on a deeper perspective compared to the mere number of edges. In the correlational debt context: the higher it gets, the more cliques of correlation are shown, meaning that a clique of countries all correlate in a similar way.

Panther Top-K similarity: Panther returns the top k nodes similar to an input node in a Network. Discovering the top-k most similar countries in a correlation graph could be useful to find those who have similar contexts inside the network topology. We initially thought about showing the top-3 similar nodes to Norway (which often appears as an isolated component), but the NetworkX Panther Implementation seemed to have some problems, so we couldn't manage to apply it to every single computed graph, but only on a subset of them. The results are available in the analysis Notebook, but were not considered as an input for further analysis, since we couldn't apply it to all the correlations timespan.

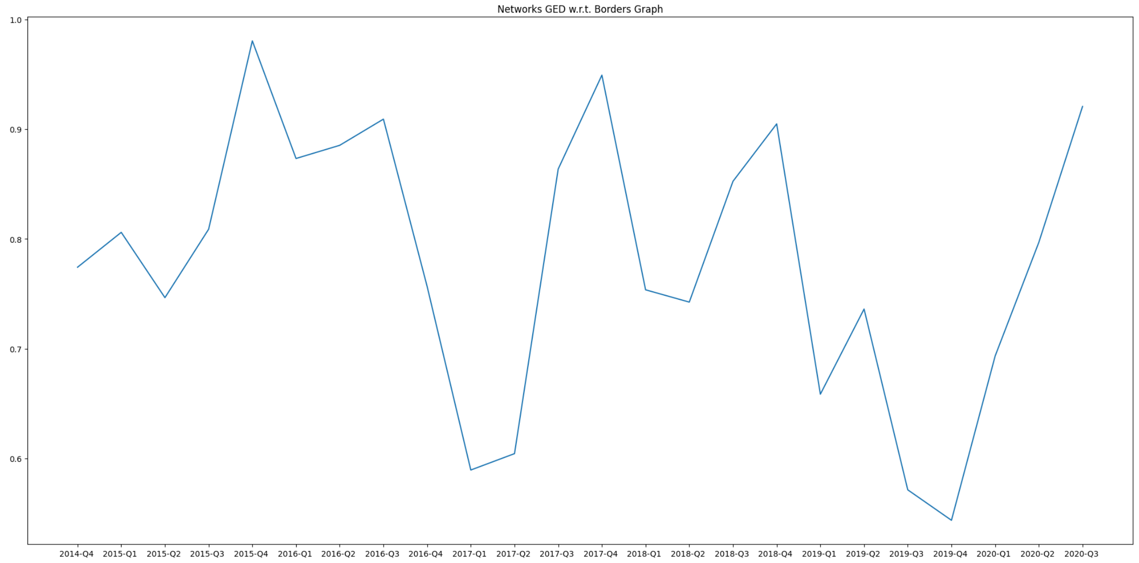

Figure 4: Graph Edit Distance w.r.t. the Border Graph computed on all 24 graphs

All the measure plots can be visualized in the python notebook available in the Github Repository.

After analysing all graphs from a Global point of view, 4 relevant ones have been selected for further Analysis:

- *2015-Q2*: Lowest Clustering Coefficient
- *2016-Q3*: Lowest amount of Edges among 1-Connected Component Graphs.

- *2017-Q2*: High number of connected components (4), low distance w.r.t. Border Graph, Average Clustering Coefficient = 0.8

- *2019-Q4*: Maximum number of Edges and lowest Distance from the Border Graph.

The Local Measures applied to these salient Graphs are the following:

- **Betweenness Centrality**: Nodes with an Higher Betweenness Centrality are contained in the most amount of shortest paths between other nodes. In our context it means that countries with an higher BC are part of more "chains of correlations", meaning that they can be taken as an indicator for the whole correlation network.

- **Closeness Centrality**: It shows how much a node is central in the network (closer to other nodes). Countries whose public debt is directly correlated to others are among the ones with the highest value as an output of this measure.

- **Degree Centrality**: The number of edges connected to each node. It's a useful metric to make us understand, in a quantitative way, how many other countries' public debts a country public debt is related to.

Although the computed graphs are undirected (the correlation matrix used as an adjacency matrix is symmetric), we decided to apply some **Triadic Census Analysis** Techniques to better understand the differences between these selected graphs, seeing countries as sets of three actors relating to each other.
Since the graphs are unidirected, the amount of triad configurations is no longer 16, but it narrows down to 4.

7 Results

In figure 5 we can see the plot of the betweenness centrality for each node of the salient timespans networks.

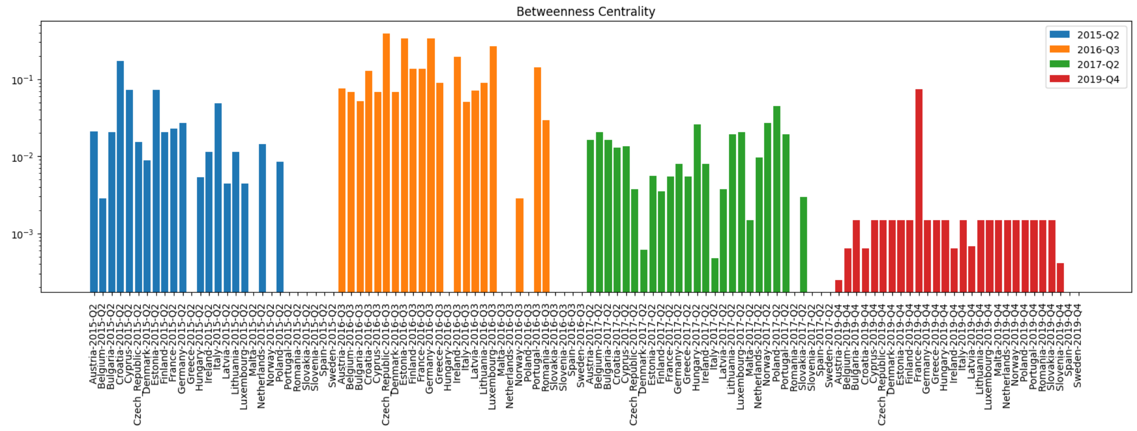

Figure 5: Betweenness Centrality Plot

Nodes of the *2016-Q3* network have, on average, an higher BC, that's probably due to the fact that the network is the one with the fewest amount of edges among the 1-component ones. This makes the "correlation chains" spread all through the network.

On the other hand every node in the *2019-Q4* network, except Finland, has a low BC, because of the extremely high number of correlation edges.

Figure 6 shows the Closeness Centrality plot. As a rule of thumb we can say that nodes become closer, the more edges the network contains. This makes Closeness inversely proportional to Betweenness, which makes sense since it's somehow a complementary measure.

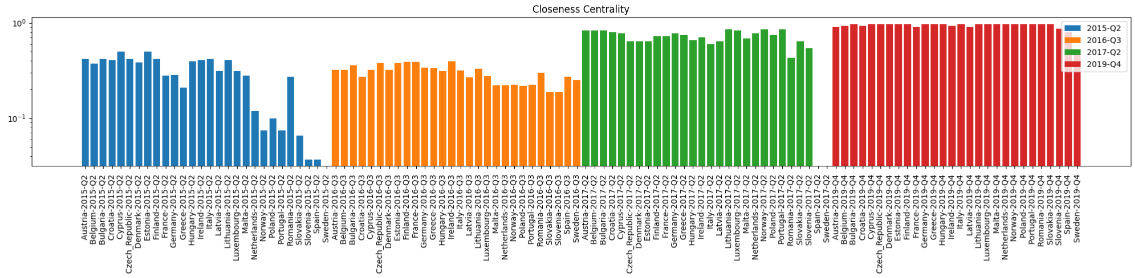

Figure 6: Closeness Centrality Plot

Figure 7 (Degree Centrality Plot) appears to be really similar to the Closeness Centrality one. That's due to the number of edges growing from the leftmost to the rightmost analyzed network.

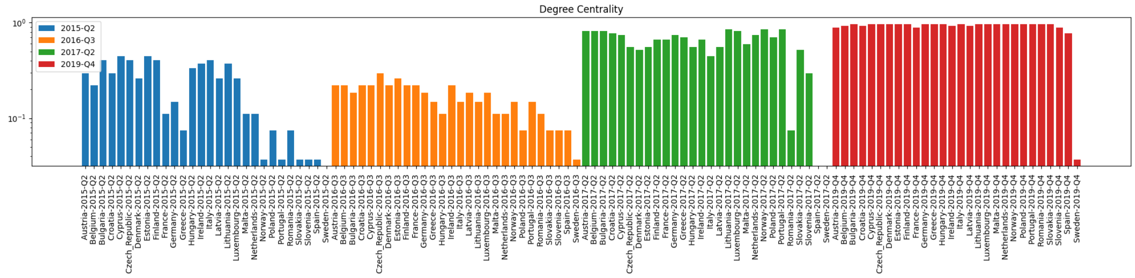

Figure 7: Degree Centrality Plot

The Triadic Census Analysis (Figure 8) shows some interesting results:

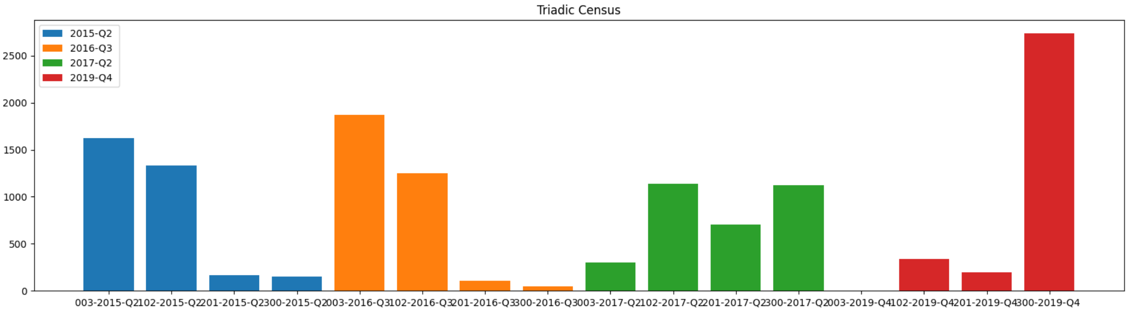

Figure 8: Triadic Census Plot

- the *2016-Q3* which shows a more "chain-like" structure in the Betweenness Centrality plot is mostly composed by **003** and **012** triads, which represent respectively unconnected triangles and triangles in which only two vertices are connected.
- Vice versa, the *2019-Q4* triads are mostly composed by **300** triangles, which represent a fully connected triad.

8 Conclusion

After the whole measure analysis process, we were able to infer a few considerations about the tackled problem:

- Correlational Networks which show a "chain-like" structure are mostly composed by **003** and **012** triads (not fully connected).

  These networks generally have a low number of nodes and a low clustering coefficient.

- On the Other hand Networks with an high amount of correlational edges show a fully connected (**300**) triad structure.

There's apparently no direct correlation between the number of edges and the similarity to/distance from the border graph, although, on a qualitative point of view, we can see that a steep increase in the number of edges implies a sudden drop in the distance of w.r.t. the border graph (as seen in Figure 9).

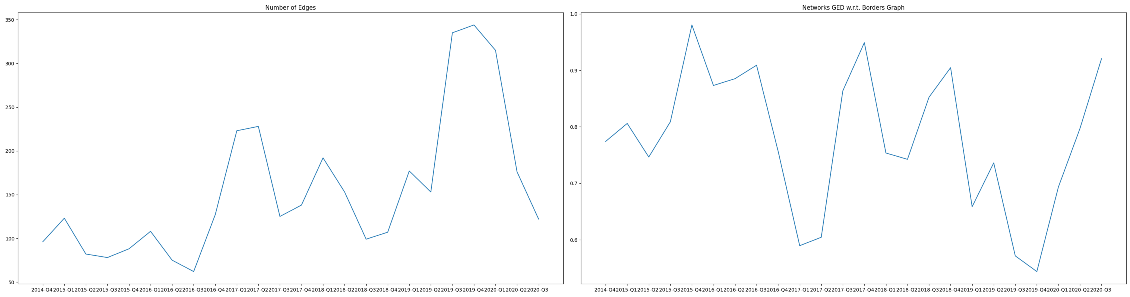

Figure 9: Number of edges and Distance w.r.t. Border Graph

This is not linearly linked to the magnitude of the increment, but somehow related.
The timespan in which the distance narrows isn't also the same as when the increment happens, but around a certain timespan after that.

9 Critique

Personally we don't think our study totally answers the question we asked ourselves at the beginning, but it certainly puts some interesting insights on the table, which could be used as an input for further analysis.
For instance it could be interesting to build a model where the relationship between the raise of correlational nodes and the similarity to the border graph is expressed in a formal way.
To analyse the phenomenon at a deeper level, PD/GDP should be sampled at an higher frequency, but the dataset offered from Eurostat was only available at quarter resolution.
It could also be interesting to examine the problem from an economical/geo-political perspective: since our knowledge on these topics is quite limited we could only consider the most relevant event (start of the COVID pandemic) during our study.
Some students or researchers with a stronger economical background could give some other insights on the graph correlational behaviour over time.
Lastly, we'd like to elaborate more on the node similarity topic, unfortunately the NetworkX Panther implementation made our initial plans unfeasible.