**Social Network Analysis Group Project: Community Detection**

**Ekapope VIRIYAKOVITHYA, Priya VARADARAJAN, Harshit PALIWAL, Yen Chun LIU, Deborah KEWON**

**Introduction**

The goal of community detection in social network is to identify highly connected clusters of individuals by finding the nodes that can be easily grouped. Community detection is very important because it helps us have an in-depth understanding of false links and spreading processes in various environments, including but not limited to, epidemics and rumor spread. In this paper, we will use walktrap, modularity optimization and edge betweenness algorithms to find possible divisions of a network. As for evaluation metrics, we will use Variation of Information (VI), rand index and adjusted rand index.

**Data Source**

The data, we are going to use, is the domestic supply of industries in 2012 (US) retrieved from the US Department of Commerce. It contains 405 observations for every individual industry. In order to analyze the communities extensively, the data will be divided into three different grouping schemes:

* 405 Individual Commodities
* 70 Subsectors
* 23 Sectors

**Methodological Approach**

Our goal is to get the number of communities, information of sectors per community and modularity score in the structure. We will also be analyzing the communities based on every sector and how they are divided based on different algorithms. As mentioned earlier, the communities can be detected using the following algorithms

**WalkTrap:** It detects communities through a series of short random walks. The idea is that the vertices encountered on any given random walk are more likely to be within a community.

**Modularity Optimization (Louvain Modularity) :** It detects small communities by optimizing modularity on all nodes. In other words, it joins the pair of communities that most increases modularity until no such pair exists. It is one of the fastest modularity-based algorithms that performs well with large graphs.

**Edge-betweenness:** It detects communities by iteratively including and removing nodes in order to maximize the modularity.

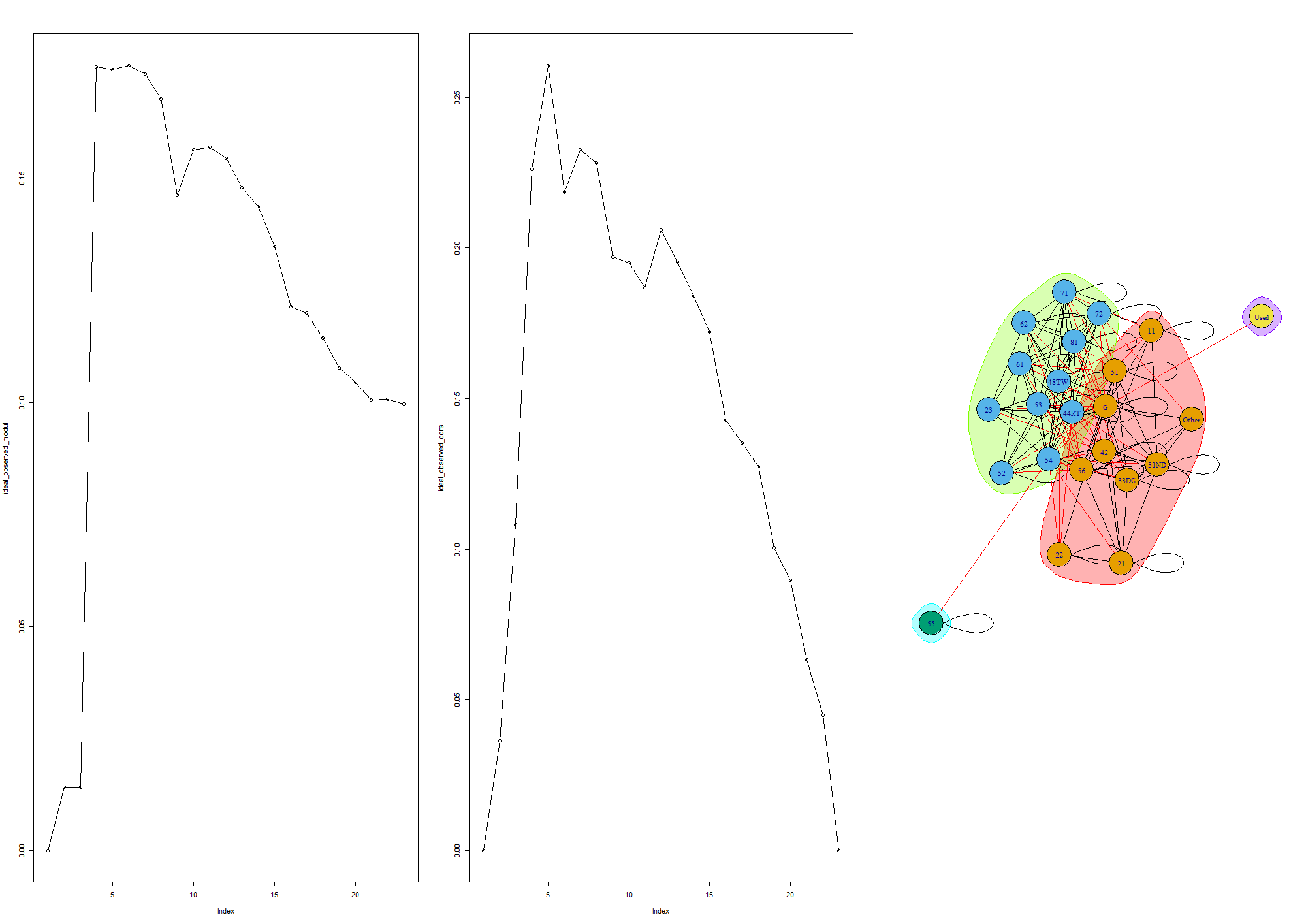
**Results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | # communities | 405 industries | 70 sub-sectors | 23 sectors |
| Walktrap | modularity | 12 | 5 | 4 |
| Modularity | modularity | 5 | 3 | 5 |
| Edge Betweenness | modularity | 20 | 29 | 14 |

**Communities and correlation plot for each scheme**

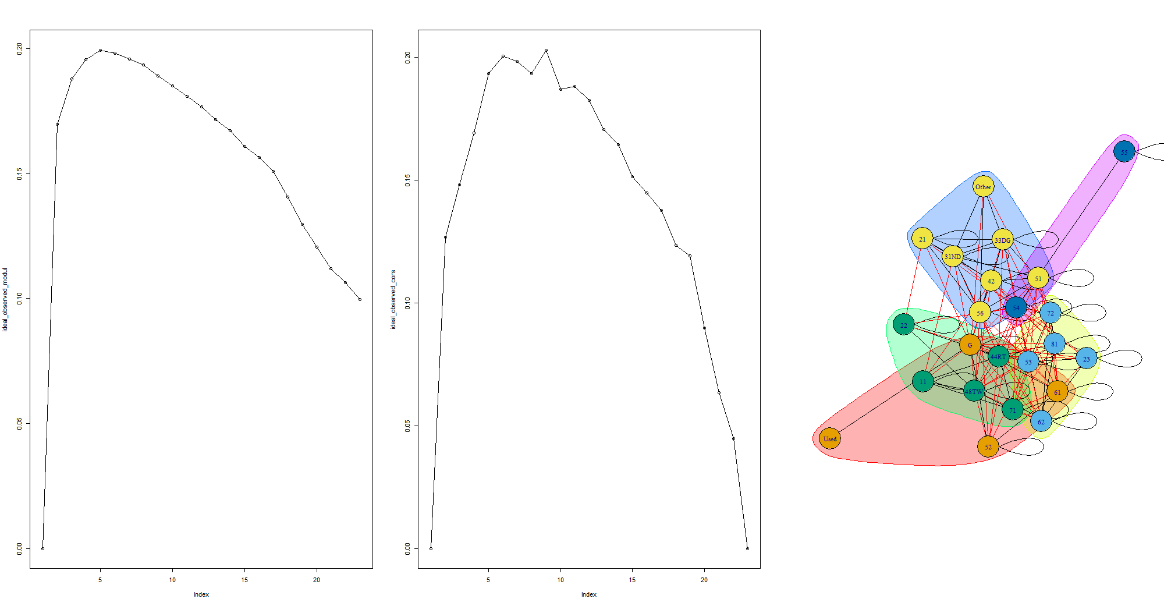
**23 Sectors**

* **Walktrap**



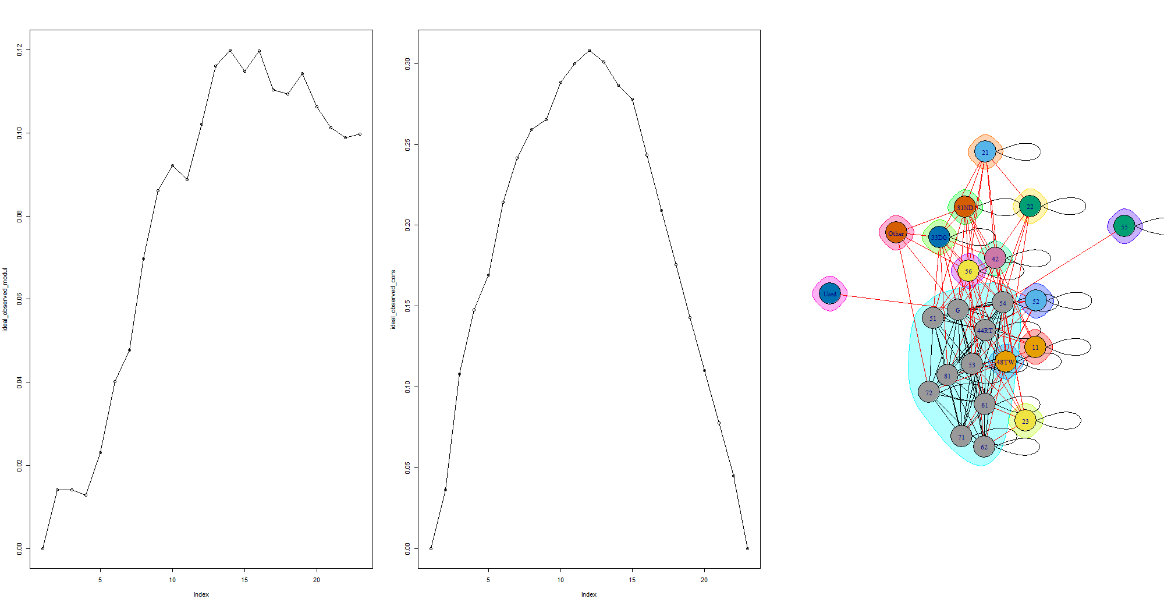
For Walktrap algorithm, the peak in the modularity plot, which is the optimum number of communities is equal to 4. The plot on the right shows us that there are 4 communities detected.

* **Greedy modularity optimization**



In the modularity plot, the peak for modularity optimization algorithm is at 5. The right plot shows us that there are 5 communities detected.

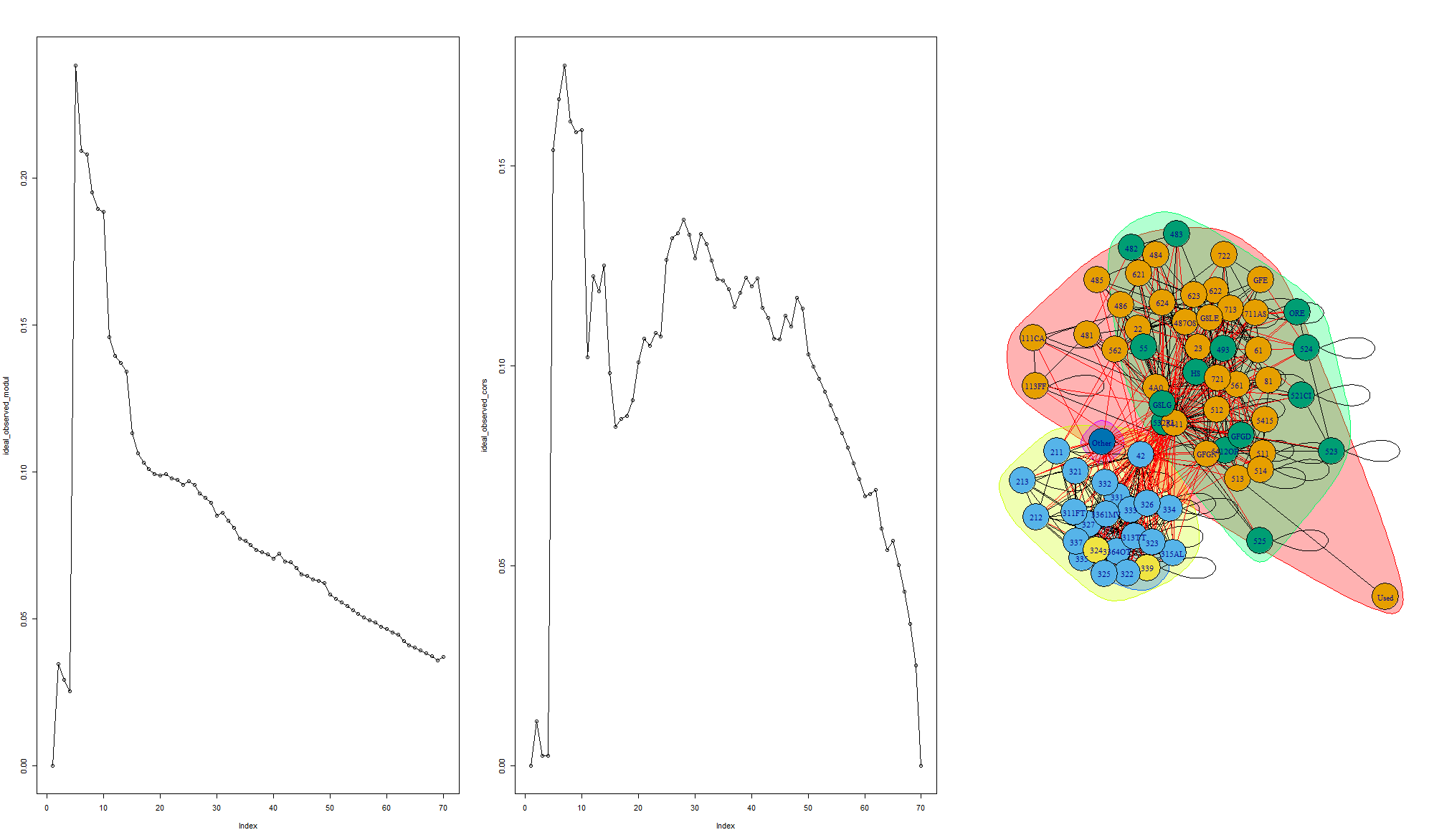
* **Edge-betweenness optimization**



In the above modularity plot for Edge Betweenness algorithm, the peak is at 14. The right graph shows us that there are 14 communities detected.

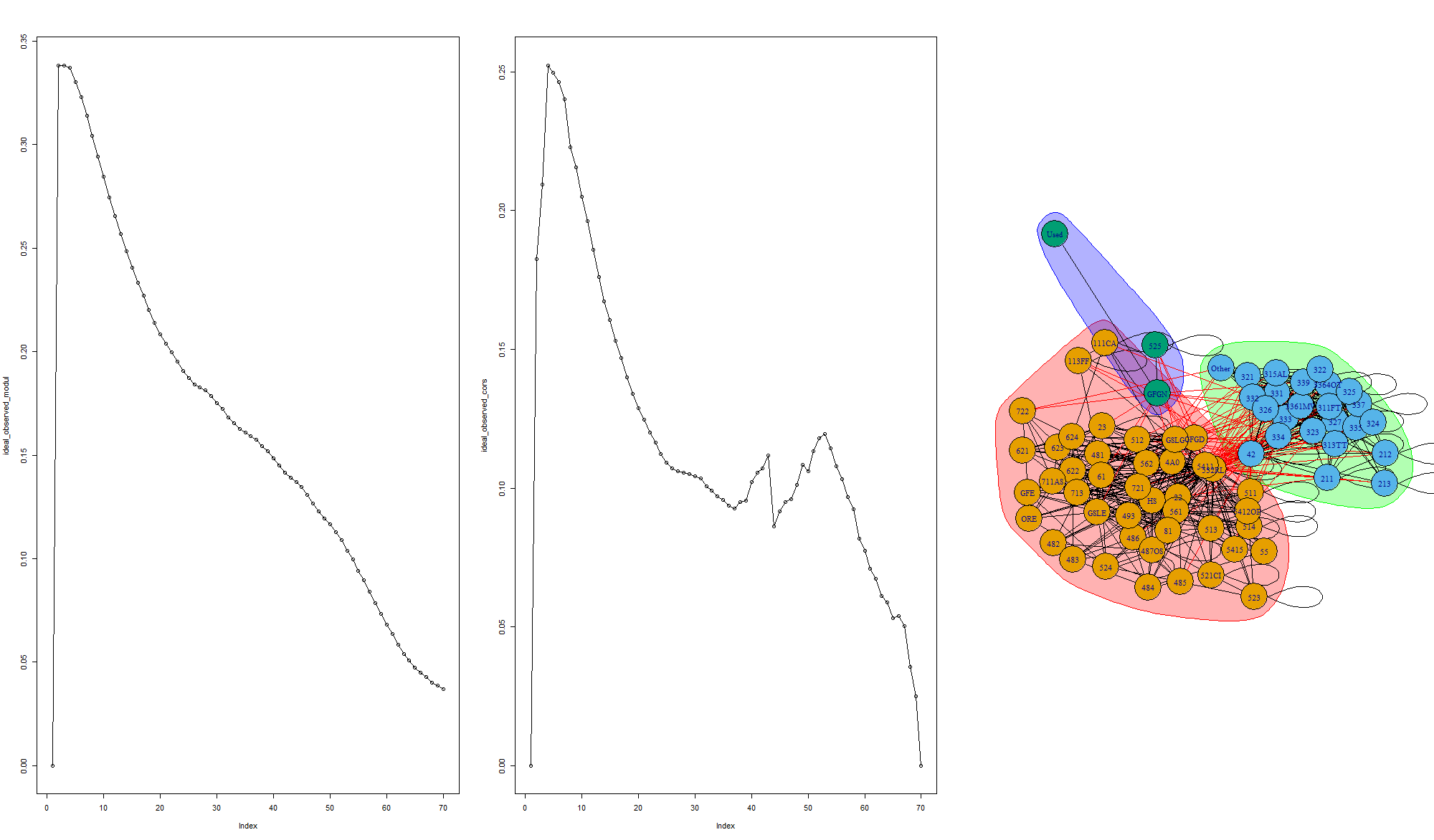
**70 subsectors**

* **Walktrap**



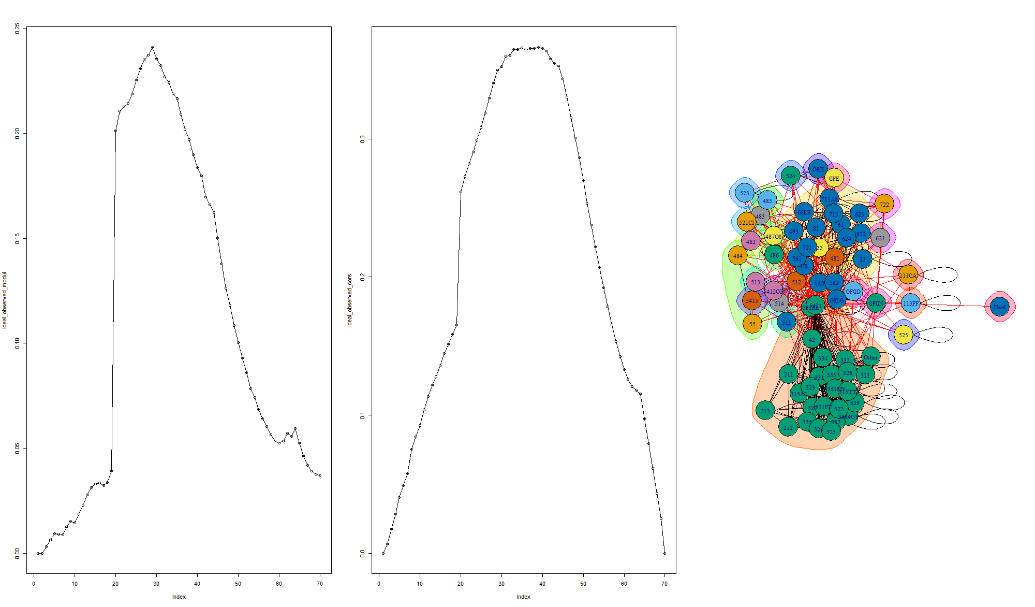
The above plot with modularity for walktrap optimization algorithm shows us that there are 5 communities at the peak for the different sub sectors.

* **Modularity optimization**



The above plot with modularity for Modularity optimization algorithm shows us that there are 3 communities at the peak for the different sub sectors.

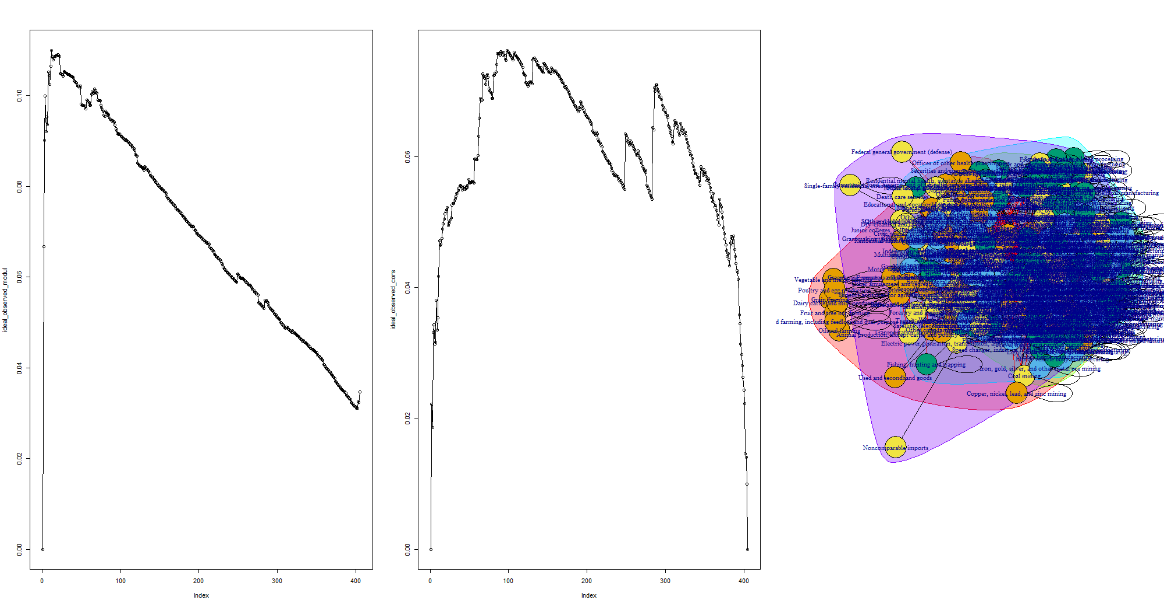
* **Edge-betweenness**



The above plot of modularity for Edge-betweenness algorithm shows us that there are 29 communities at the peak for the different sub sectors.

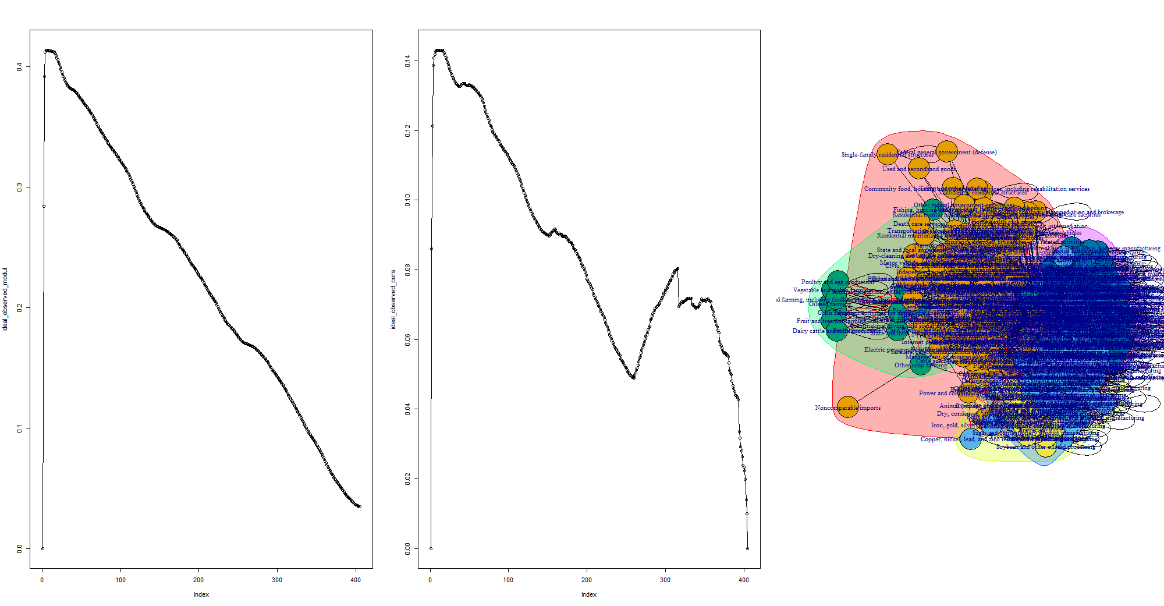
**405 industries**

* **Walktrap**



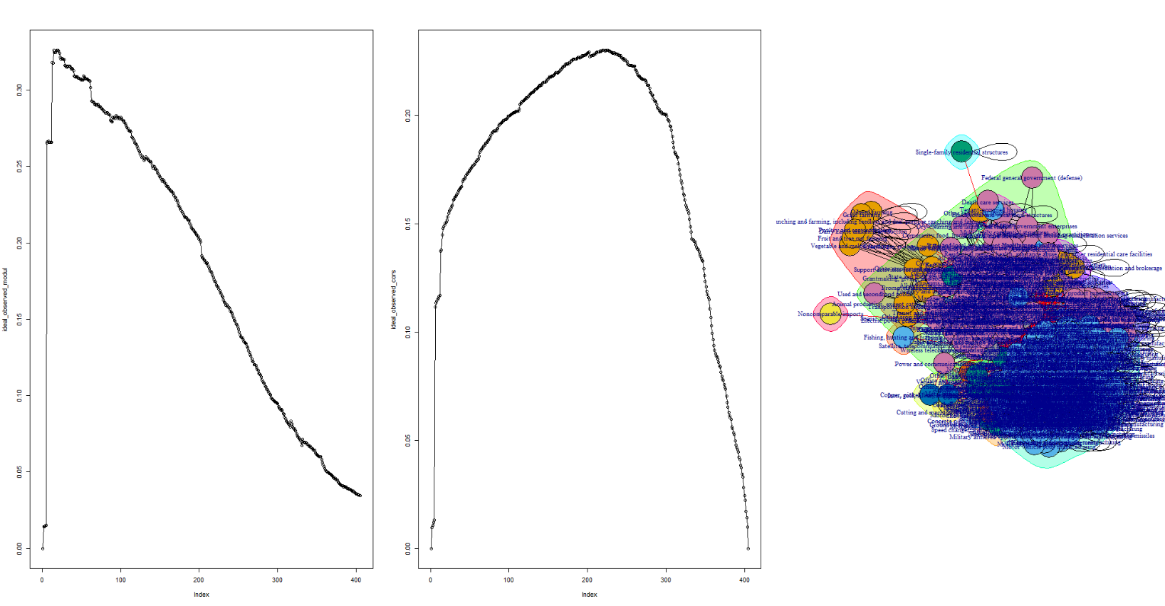
Walk trap has around 12 communities at the peak of modularity with different commodities grouped together.

* **Greedy modularity optimization**



Greedy modularity has around 5 communities at the peak of modularity with different commodities grouped together.

* **Edge-betweenness optimization**



Edge betweenness has around 20 different communities at the peak for the different industries.

**Evaluation**

Model evaluation is a crucial step in order to determine which model gives better results. There are various evaluation metrics, which are used in order to determine the performance.

For benchmarking the models, a dataset in which we already know the communities are used to check how is the model working against the true values. To evaluate the model, following metrics are used -

* Confusion Matrix
* F-score

After benchmarking the model and finding the best suited model, the model is applied on a similar dataset. This gives optimum results for finding new communities in a similar network.

On the other hand, to evaluate the models which don’t have any labeled communities available, following measures are used -

* RAND
* Variation of information
* Adjusted RAND
* **RAND**

The Rand index/Rand measures a measure of the similarity between two data clustering. A form of the Rand index may be defined that is adjusted for the chance grouping of elements, this is the adjusted Rand index. From a mathematical standpoint, Rand index is related to the accuracy, but is applicable even when class labels are not used. The Rand index has a value between 0 and 1, with 0 indicating that the two data clustering do not agree on any pair of points and 1 indicating that the data clustering is exactly the same.

* **Adjusted RAND**

The expected value of the Rand Index of two random partitions does not take a constant value (e.g. zero). Thus, Adjusted RAND assumes a generalized hypergeometric distribution as null hypothesis: the two clusters are drawn randomly with a fixed number of clusters and a fixed number of elements in each cluster. Then the adjusted Rand Index is the (normalized) difference of the Rand Index and its expected value under the null hypothesis.

* **Variation of information**

Variation of information or *shared information distance* is a measure of the distance between the two clusters. It is derived from the *mutual information*, but it is a true metric, *i.e.* it is symmetric and satisfies the triangle inequality.

After applying Walktrap, Modularity Optimization and Edge-betweenness algorithms, the performance of these models was compared to each other using the RAND, Adjusted RAND and VI method. These performance indices show how similar these models are compared to each other.

Below are the results for different model performances -

**23 Sectors scheme**

* Walktrap and Modularity Optimization

|  |  |
| --- | --- |
| **Method** | **Performance score** |
| RAND | 0.189 |
| Adjusted RAND | 6.850e-17 |
| Variation of Information | 1.542 |

* Modularity Optimization and Edge-Betweenness

|  |  |
| --- | --- |
| Method | Performance score |
| RAND | 0.695 |
| Adjusted RAND | -0.014 |
| Variation of Information | 1.871 |

* Walktrap and Edge Betweenness

|  |  |
| --- | --- |
| Method | Performance score |
| RAND | 0.177 |
| Adjusted RAND | -6.752078e-17 |
| Variation of Information | 2.134 |

**70 Sub-sectors scheme**

* Walktrap and Modularity Optimization

|  |  |
| --- | --- |
| Method | Performance score |
| RAND | 0.785 |
| Adjusted RAND | 0.557 |
| Variation of Information | 0.718 |

* Modularity Optimization and Edge-Betweenness

|  |  |
| --- | --- |
| Method | Performance score |
| RAND | 0.655 |
| Adjusted RAND | 0.301 |
| Variation of Information | 1.784 |

* Walktrap and Edge Betweenness

|  |  |
| --- | --- |
| Method | Performance score |
| RAND | 0.726 |
| Adjusted RAND | 0.304 |
| Variation of Information | 1.995 |

**405 Industries scheme**

* Walktrap and Modularity Optimization

|  |  |
| --- | --- |
| Method | Performance score |
| RAND | 0.622 |
| Adjusted RAND | 0.056 |
| Variation of Information | 2.414 |

* Modularity Optimization and Edge-Betweenness

|  |  |
| --- | --- |
| Method | Performance score |
| RAND | 0.758 |
| Adjusted RAND | 0.517 |
| Variation of Information | 0.998 |

* Walktrap and Edge Betweenness

|  |  |
| --- | --- |
| Method | Performance score |
| RAND | 0.602 |
| Adjusted RAND | 0.094 |
| Variation of Information | 2.253 |

Based on the above results, much of inference cannot be made because we do not have the base model for comparison, however we can interpret that Modularity optimization gives us better results on comparison with any algorithm for all the evaluation metrics.

**Business Interpretation**

In order to further interpret our analysis on commodity detection for the supply chain data from business perspective, we have looked further in the communities that the algorithm provides for all the different types of data (23, 70 and 405 commodities).

The tables below show the detail grouping by each algorithm. For each group, the industries have been provided with similar commodities. Let us take the example of 23 sectors scheme and explain further, we can see ‘Finance and insurance’ and ‘Real estate and Rental and Leasing’ have been group together in group 2, this means that these two sectors have provided similar commodities. In the market of rental and leasing, they do have relationships with insurance industries. They tend to collaborate with cross-selling marketing strategy.

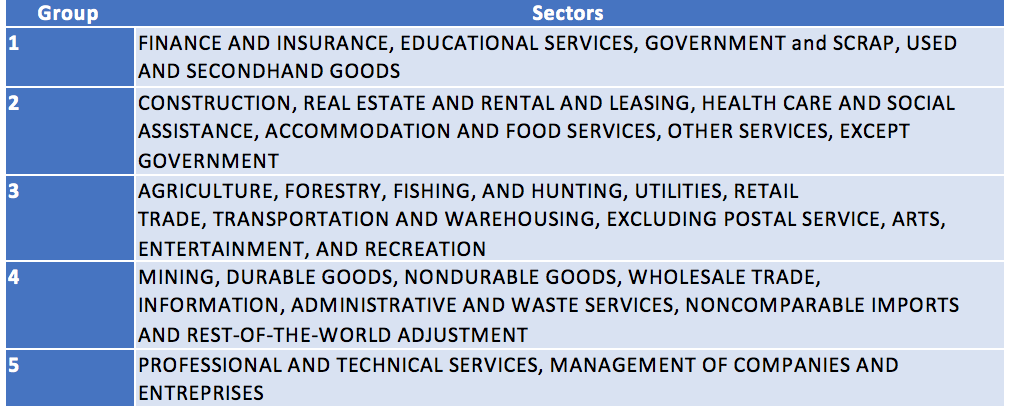
For the ones, which are unable to be explained, we may need to go deeper to businesses or seeking for more external information. The use of this information from the marketing side can provide cross-selling within two or multiple industries. By looking at the overall sectors, it could be understood all possible potential markets or collaborated partners, and even M&A (merge and acquisition) plan.

**23 sectors scheme**

**Walktrap**



**Greedy Modularity**



**Edge-betweenness optimization**

**70 Sectors**

**Greedy modularity optimization**



**Walktrap**



**Edge-betweenness optimization**



**Conclusion**

Comparing the results of the three different algorithms, we can say that the modularity optimization algorithm seems very consistent on comparison across the all three different types of data grouping schemes (405 industries, 70 Subsectors and 23 Sectors), while the other two algorithms give different results on different sets of data. However, based on the discussed business interpretation results, it comes to the conclusion that modularity optimization is chosen for the dataset in this project.

**References**

<https://sna.stanford.edu/lab.php?l=3>

<https://www-complexnetworks.lip6.fr/~latapy/Publis/communities.pdf>

<http://jgaa.info/accepted/2006/PonsLatapy2006.10.2.pdf>

<https://www.comp.nus.edu.sg/~leonghw/Courses/CS3230R/Talks/W10-Duy-CS3230R.pdf>

<https://stackoverflow.com/questions/52717240/cluster-walktrap-returns-three-communities-but-when-plotting-they-are-all-on-to>

<https://stackoverflow.com/questions/24682166/detecting-network-communities-using-walktrap-using-a-large-number-of-steps>

<https://www.bea.gov/industry/input-output-accounts-data>

<https://neo4j.com/docs/graph-algorithms/current/algorithms/louvain/>