Analysis of Large Scale Social Networks

Project assignment

ANALYTICS PROJECT : ANALYSIS OF AMAZON CO-PURCHASING DATA

Mentor: Prof. Bart Thijs

Kin Ho Chan - r0772534
Loris Gallo - r0906521
Hrishikesh Nambiar - r0909789
Milton Ossamu Tanizaka Filho - r0822517

KU LEUVEN

Introduction

With the advancement of e-commerce, the number of online transactions and reviews are rising everyday in an unexpectedly fast manner. The network between each purchase and people's attitude towards a certain group of products is then becoming vital for online recommendation systems. By understanding the relatedness of the purchases, better recommendations can then be made to promote business growth.

Nonetheless, the understanding of such a purchasing network relies on how well the clustering of the vast network of nodes and edges lies within is performed. Among different clustering methods, the Louvain method and its successor, the Leiden method, are the two most popular algorithms at the moment of writing. Both Louvain and Leiden methods are agglomerative methods that move and aggregate the clusters within the communities iteratively to reach an optimal state of clustering with the highest modularity they could obtain. Yet, the difference between these two methods was the step after modularity optimization where Leiden performed an additional step of small community refinement while Louvain did not. The lack of community refinement from Louvain later led to badly connected or disconnected communities while Leiden was able to guarantee the fully connected communities. On top of it, Leiden was found to be faster in computation time than the Louvain. On the other hand, the infomap method, which performed similarly as the Louvain method, has been optimized based on a map equation instead of modularity.

Here we aim to experiment how these three algorithms as well as the infomap method would affect the network analysis and explore which would be the best method. We have selected the Amazon co-purchasing network from March 2003 as study material considering Amazon as one of the biggest e-commerce platforms.

Dataset

In this project, we used the Amazon product co-purchasing network from March 02 2003 for network analysis and community detection experiments, which can be obtained on this website https://snap.stanford.edu/data/amazon0302.html.

The dataset was created by crawling the Amazon website based on the mindset that "customers who bought this item also bought".

Before conducting the analysis, we examined the majority of the products were books (393.561 units) followed by music (103.143 units), video (26.131 units), and dvd (19.828 units) (Figure 1). This makes sense since back in 2003, Amazon was not an "everything store" as it is nowadays.

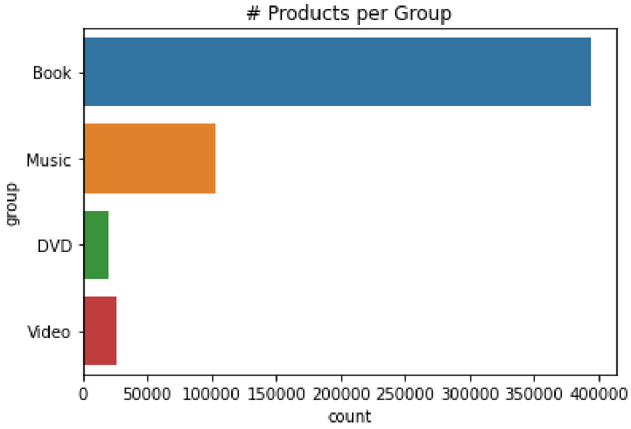

Figure 1. The distribution of products

Apart from the products, there was a large amount of 0 discovered in the rating (Figure 2A). It was directly following from the fact that these products did not receive a review and therefore it was concluded as 0. The graph furthermore shows that the majority of products have an average rating above 4.0. Figure 2B showed the average ratings per product group. DVD was the highest rated product group and Books was the lowest rated out of the four groups.

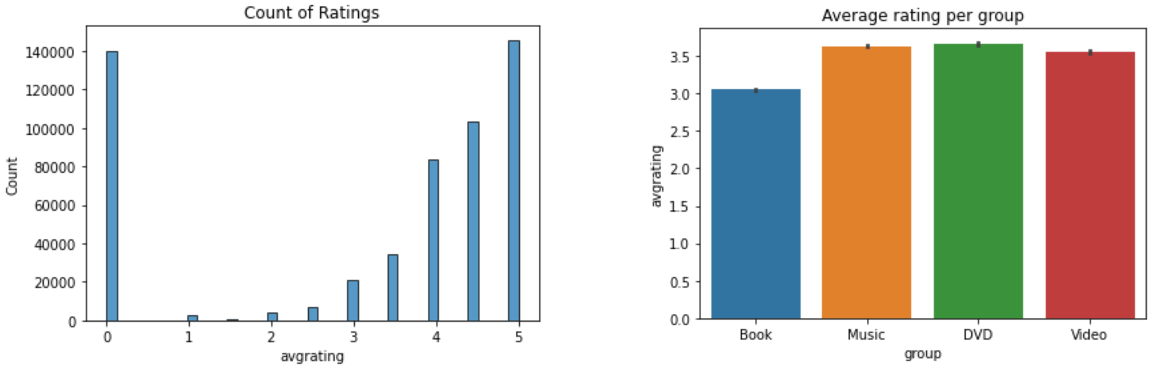

Figure 2. A) The distribution of ratings (left) & B) of average rating per group(right)

The plot below showed the average rating and number of items per category. Most of the categories had average ratings above 3 (Figure 3A), and most of them did not have many items. It was notable that some categories seem to have no rating at all like the first category (Figure 3B), but we kept them for the analysis.

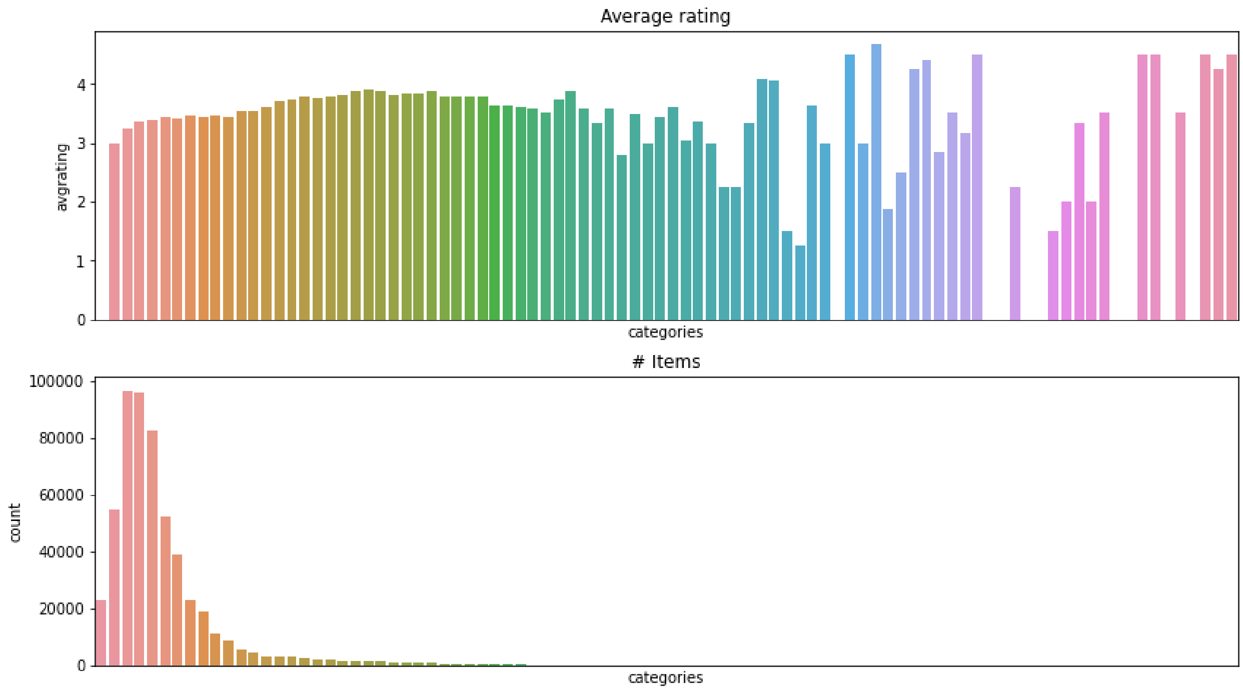

Figure 3. A) The distribution of ratings for each category (left); B) The number of count for each category (right)

Although the dataset was initially intended to be a directed graph, meaning that a product *i* is frequently co-purchased with product *j*, we built as a undirected graph to perform a community detection analysis and identify products that might be similar to each other and might be frequently be suggested for purchase for a recommendation system.

The nodes were defined as each product and the vertices link co-purchased products. We also added a weight vertex to indicate the strength of average ratings of products.

The initial density of the network is $3.59*10^{-5}$, meaning most of the items were bought isolated from other products. The transitivity, or clustering coefficient, measures the probability that two neighbors of a vertex are connected. In our network, the average transitivity was 0.43, which was a measure of the overall probability for the network to have adjacent products bought together.

Most nodes had low degree value, which degrees up to 10 comprise 73% of the nodes (Figure 4A). The kernel density estimate also confirmed that degree was concentrated in lower values (Figure 4B). While this indicated that most nodes were connected to a handful of adjacent nodes, this also revealed that there was rarely an individual product holding most of the connections.

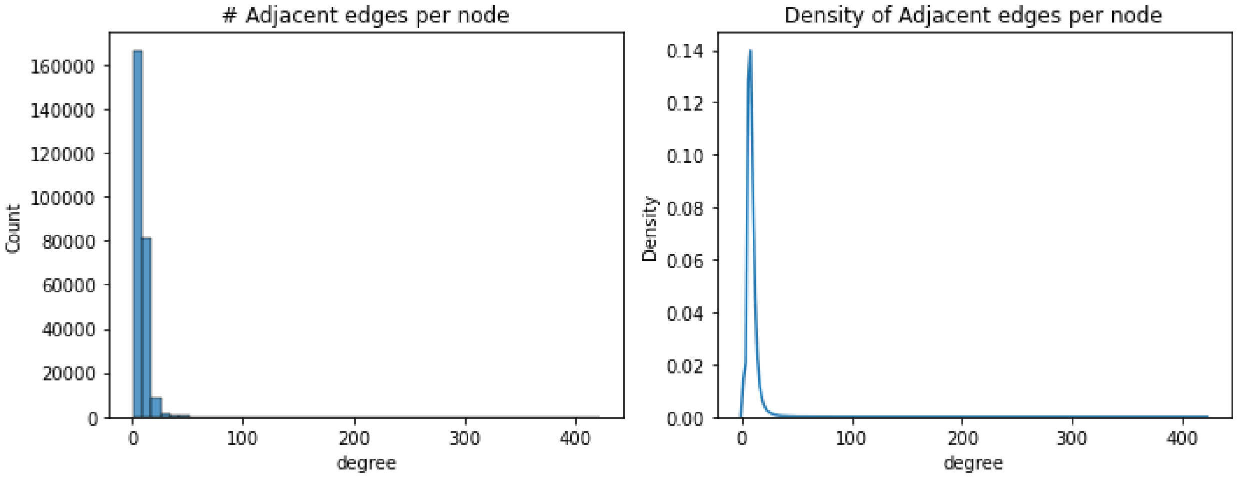

Figure 4. A) The distribution of adjacent edge for each node (left); B) The density of degree for adjacent edges per node (right)

The local transitivity calculates the probability that two products are connected separately for each vertex and the distribution could be found in Figure 5A & 5B. Most of the nodes present were below 0.5 transitivity. But a considerable number of items showed above 0.8, indicating high potential to be purchased along with other items.

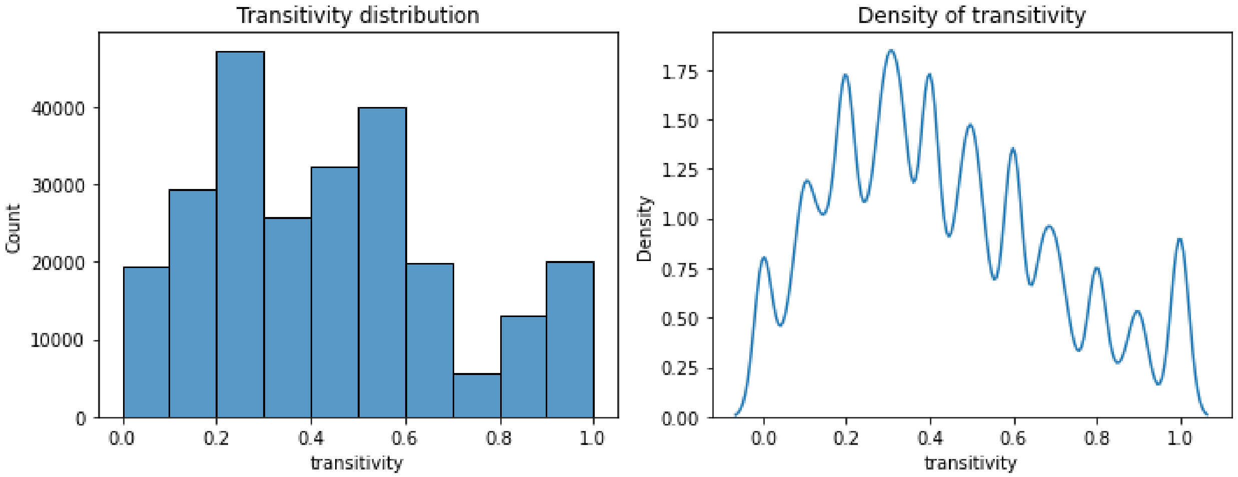

Figure 5. A) The distribution of transitivity for each node (left); B) The density of of transitivity for each node (right)

Since the raw graph presented nodes that were not connected, we considered for the analysis only the subgraph with the most connected components, which comprised 258.958 nodes and 1.206.732 vertices.

Methodology

There are many different methods to assess clusters of products based on different paradigms. We won't work in an exhaustive manner by trying as many methods as possible. Rather we will try to focus on certain well-known methods like Louvain, Leiden and Infomap. These methods will give us the demanded insights into the different communities and their modularity, transitivity and adjusted rand index. We implemented each algorithm using just the connected products (simple vertex) and a weighted vertices, which considers the product of the average rating of two products.

These methods were used to calculate the classic quantitative metrics such as the Normalized Mutual Information (NMI)[1], which shows between different methods if they produce similar size community distributions. We chose to compute the NMI because of the frequency of usage in community detection and the ability of comparison between partitions where nodes are assigned to a different number of clusters. Another metric used was the Adjusted Rand Index [2]. The ARI will identify the level of similarity between the different methods. Finally Modularity [3] will be one of the most interesting metrics to work with. Modularity measures the difference between the actual fraction of edges within the community and such fraction expected in a randomized graph with the same number of nodes and the same degree sequence. It is widely used as a measurement of strength of the community structures detected by the community detection algorithms.

Furthermore the hub dominance will be calculated in conjunction with the transitivity [4]. This will be important to obtain information about the internal patterns in the communities. Particularly, the hub dominance was often considered as the level of centralization around a node while the transitivity was representing the probability of the adjacent vertices of one vertice to be connected [5]. All methods and experiments were performed within the google colab environment with the use of Python 3.7.13 and iGraph Library.

Results

We compared the insights of each community detection algorithm in terms of modularity, community size distribution, NMI (normalized mutual information), and adjusted rand.

1.1. Modularity

The modularity plot (Figure 6A) showed us that out of all the algorithms the weighted Louvain was the best in detecting communities when relying on the modularity of the algorithms (0.93), followed by Louvain raw (0.91).

Although modularity maximization is a popular technique for detecting communities, it is known that due to resolution limits, it might not be appropriate since small communities might merge together.

1.2. Number of communities

The number of communities varied quite a lot (Figure 6B) . Particularly, the raw Louvain method (i.e. without weighting) identified 198 communities yet the infomap with weighting showed 119.043 communities. Choosing an algorithm with few communities might end up recommending products that are not relevant at all for a certain user, but it is also possible to reach serendipity, i.e., recommending unexpected items that the user might like.

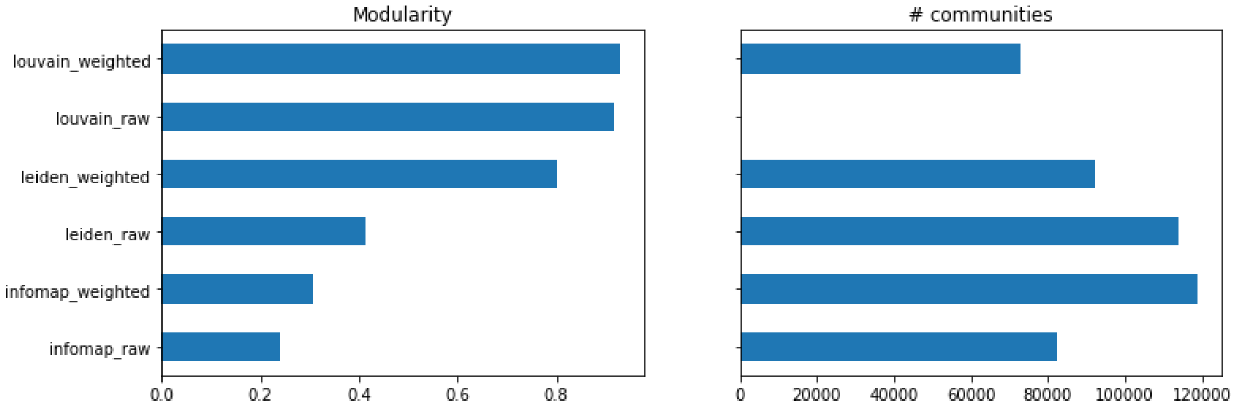

Figure 6. A) The modularity of each method (left); B) The number of communities calculated from each method (right)

We can see in figure 6.b the size of the communities by each method. While leiden_raw gives communities with homogeneous sizes (figure 7) other methods present few communities with many items and the many communities with few products, being the louvain_weighted a more extreme outcome. Louvain_raw seem to present clear defined clusters.

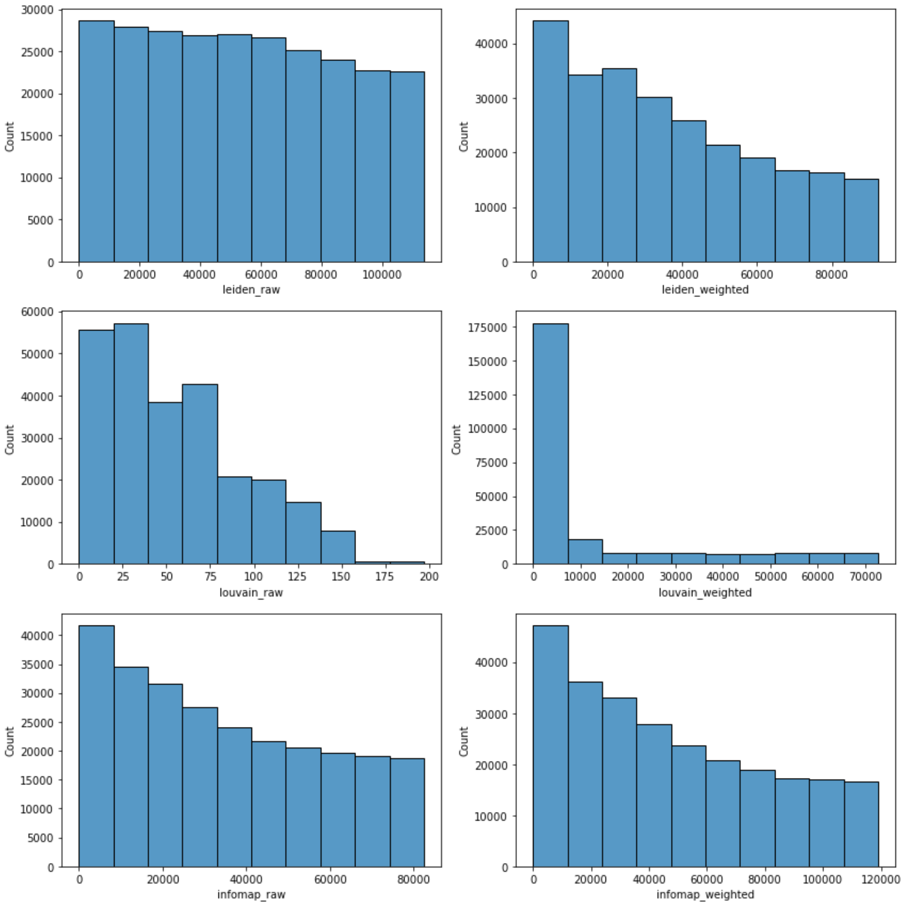

Figure 7 Distribution of community sizes by each community detection algorithm

1.3 NMI and Adjusted Rand

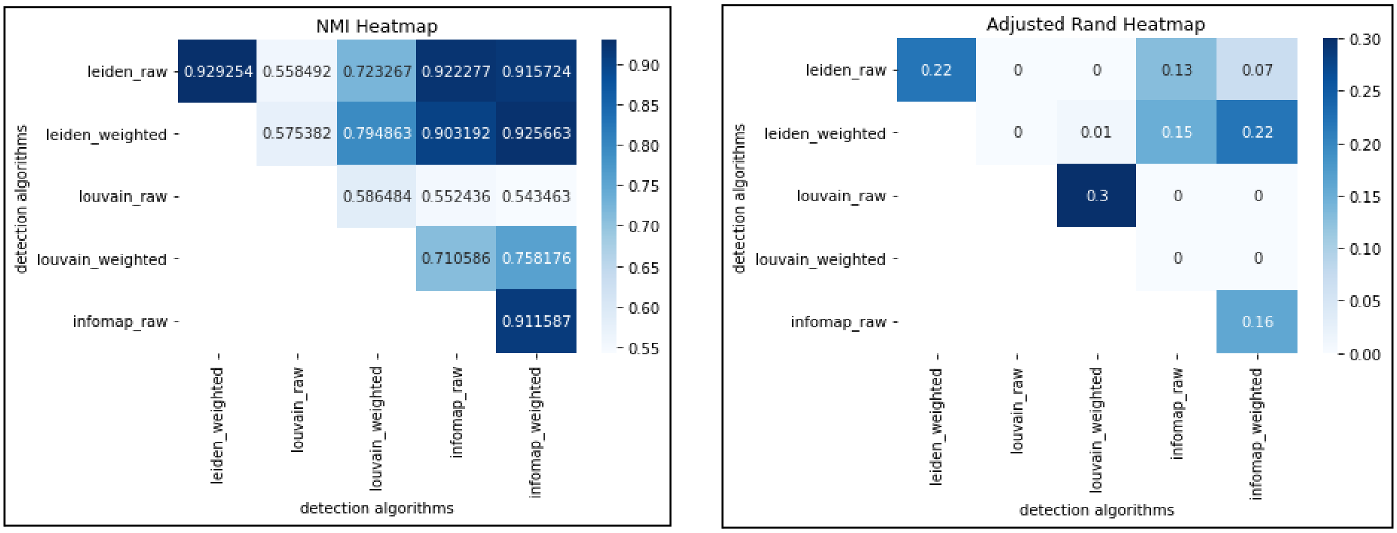

Figure 8 A) The NMI heatmap for the combination of methods; B) the ARI heatmap

The two heatmaps above indicate two different metrics, the Normalized Mutual Information (NMI) (Figure 8A) and the Adjusted Rand Index (ARI) (Figure 8B). The NMI heatmap showed that the different approaches were very similar in results. All scores were positive and between 0,54 and 0,93. The methods [‘leiden_raw’ and ‘leiden_weighted’], [‘leiden_raw’ and ‘infomap_raw’], [‘leiden_raw’ and ‘leiden_weighted’], [‘leiden_weighted’ and ‘infomap_weighted’] and finally [‘infomap_raw’ and ‘infomap_weighted’] all had an NMI score above 0.90. Surprisingly the Louvain raw method was comparatively least correlated to the other algorithms, even to its weighted version.

In terms of the ARI heatmap (Figure 8B), the correlation between the algorithms is very low to non-existent. But in this metric, the Louvain raw method instead showed the highest correlation (0.3) to its weighted version.

1.4 Hub-dominance and transitivity

When searching for patterns in the data it is very useful to take a look at the Hub dominance and Transitivity. After calculating the Hub dominance on the first and biggest cluster of data we got the following results with respect to the transitivity.

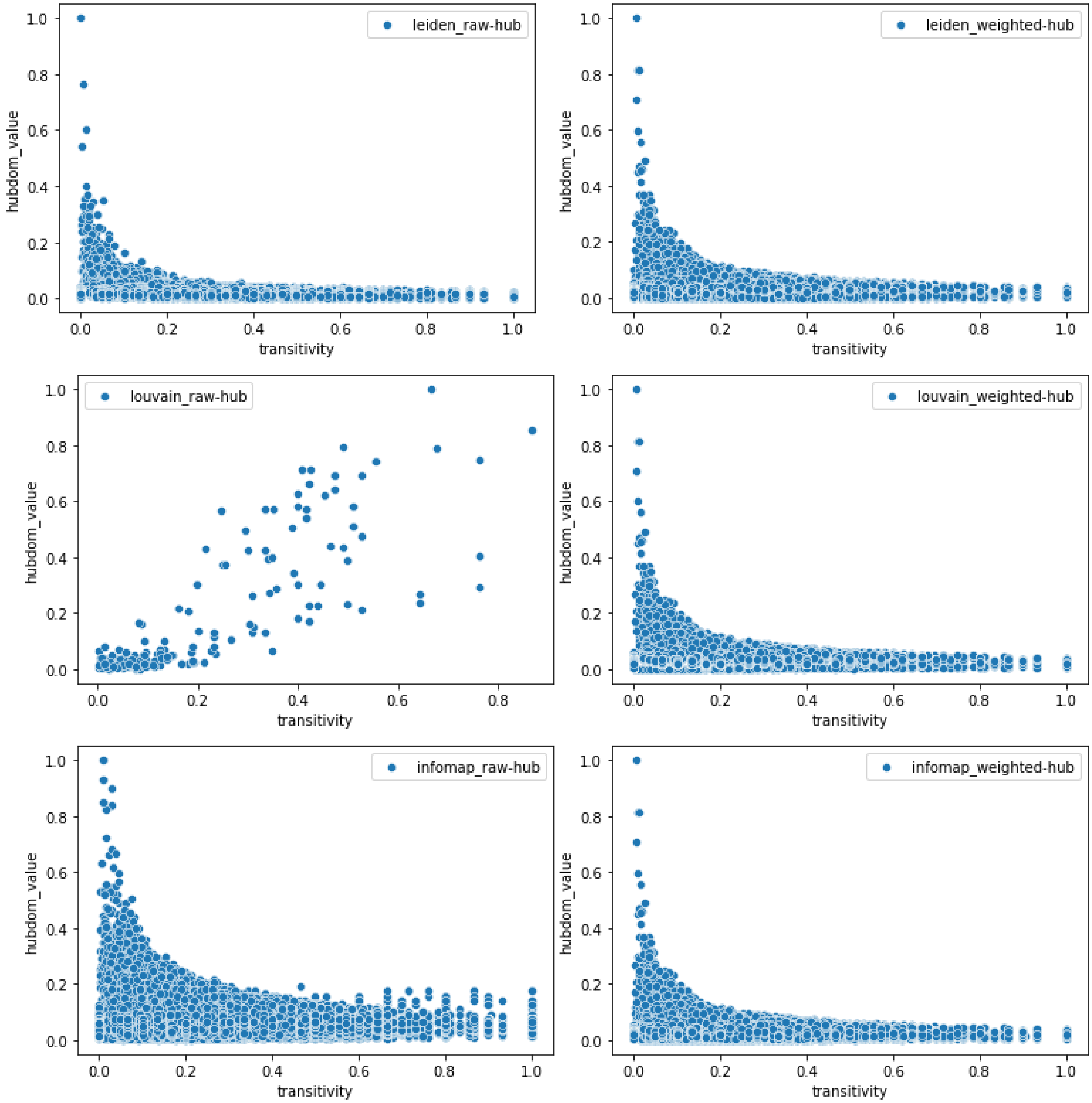

Figure 9. The transitivity and hub dominance for clusters from each method

Among these plots (Figure 9), the raw Louvain method which has by far the least amount of communities is the only one without a decreasing convex function. This result showed that there was a big hub dominance and therefore the different communities in this algorithm looked more like a hub-like structure. The transitivity for the louvain_raw algorithm showed that these hub-like structures are heavily connected. The method with the largest communities shows us that the data is very clustered.

In terms of cluster topology, only the clusters positioned at the top right corner of the raw Louvain method (i.e. with high hub-dominance and large transitivity) indicated a clique-based structure. The other algorithms, as without any cluster with high transitivity and hub dominance, had instead only generated clusters with string-based (i.e. low transitivity and hub dominance), grid based (i.e. high transitivity but low dominance) and star-based (i.e. low transitivity and high hub dominance) structures [5].

With these new insights, the raw Louvain method is the only good candidate. It will offer patterns which are useful for linking the different products with each other.

1.5 Communities of co-purchasing network

In order to better obtain visual insights of the communities, we plotted the network applying the communities of the raw Louvain method. Figure 10 depicts the network, discriminating each color per community, summing up to 198 communities.
We can see that some items that are far from the center of the network seem to be tied to 1 or 2 products, meaning that they may represent niche items or not so popular.

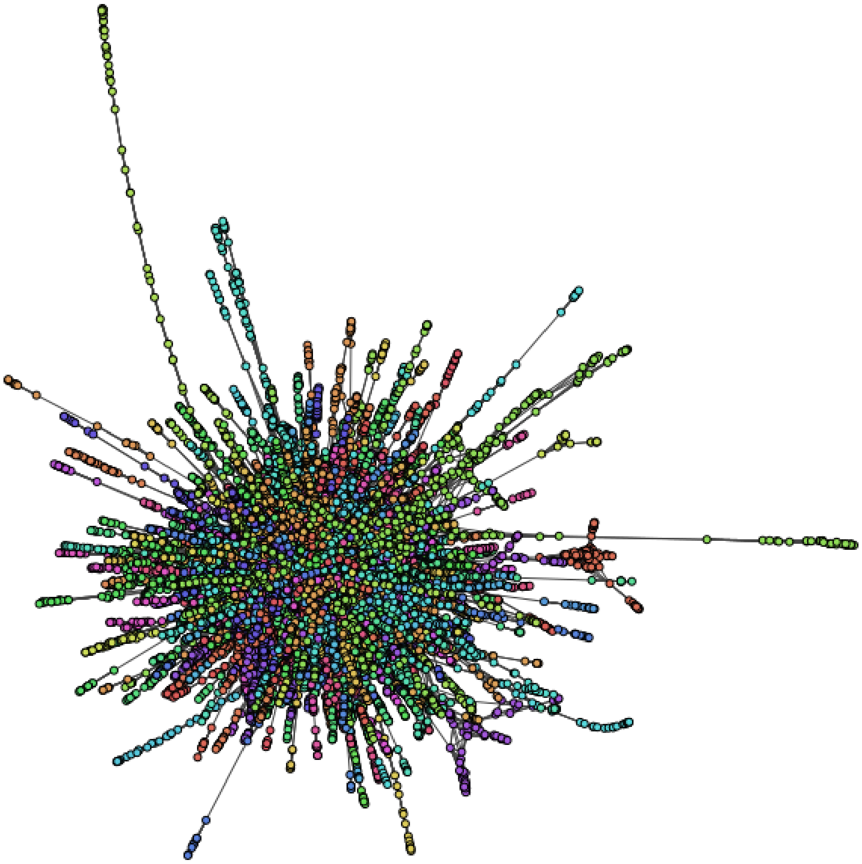

Figure 10. Co-purchasing network discriminated by each community using Louvain Raw method

1.6. Clustering

Clustering with three algorithms: Louvain, Leiden and Infomap algorithms, were implemented to cluster on the largest connected component considering weighted vertices and not considering it.

Validity: Weighted Louvain was able to successfully cluster the data. Louvain detected a very low number of communities but had high modularity. But the weighted Louvain had more communities detected and also the highest modularity. Leiden and infomap also showed great community detection.

Reliability: The communities detected were very low in the non- weighted Louvain model compared to the other models due to the lower resolution limit. Leiden solved the resolution limit problem and was able to detect more communities. Leiden has connected communities and was faster in execution. Infomap is a network clustering algorithm based on the Map Equation. Map equation solves the problem of modularity based algorithms like resolution limit and degeneracy.

Scalability: Infomap algorithm due to its sequential algorithm with random walks is time consuming and is not scalable for large datasets. Louvain and Leiden performed really well in our dataset whereas infomap faced time issues.

Conclusion

Different methodologies exist to define and detect communities in a social network. After having performed three algorithms and having obtained many varying results it still remains hard to explicitly choose one algorithm that works best in providing useful insights.

When we look at the hub dominance, only one method seems applicable. Here the raw Louvain method is the way to go. When we take into consideration the heatmap regarding the correlation between the modularities, the raw Louvain method is indeed the one that separates from the rest and enforces the possibility of being the only ‘good’ method. But when we look at the number of communities it detects in comparison to the other methods it shows a very heavy generalization. This constatation could bring doubt in selecting the raw Louvain method as the right method.

There is not an unbiased ‘way-to-go’ with regards to selecting the right community detection method. As explained throughout the analysis some might be better then others depending on what criteria is most important for the end-user.

Acknowledgment

We hereby take this opportunity to thank Prof. Bart Thijs for his guidance and the opportunity to work on this project.

Bibliography

1. Chakraborty, T., Dalmia, A., Mukherjee, A., Ganguly, N.: Metrics for community analysis: A survey. ACM Comput. Surv. 50(4), 1–37 (2017). DOI 10.1145/3091106

2. Santos, J. M., & Embrechts, M. (2009, September). On the use of the adjusted rand index as a metric for evaluating supervised classification. In *International conference on artificial neural networks* (pp. 175-184). Springer, Berlin, Heidelberg.

3. Chen, M., Kuzmin, K., & Szymanski, B. K. (2014). Community detection via maximization of modularity and its variants. *IEEE Transactions on Computational Social Systems*, *1*(1), 46-65.

4. Orman, G. K., Labatut, V., & Cherifi, H. (2011, June). Qualitative comparison of community detection algorithms. In *International conference on digital information and communication technology and its applications* (pp. 265-279). Springer, Berlin, Heidelberg.

5. Bothorel, C., Brisson, L., & Lyubareva, I. (2021). How to Choose Community Detection Methods in Complex Networks. *Series Computational Social Science*, pp-16.