Exploring the power of the clustering coefficient as a feature of popularity in a co-purchase network of books

Paul Cariou*, Michele Sprocatti†, Riccardo Zuech‡

Learning from Networks Report, January 2025

1 Motivation

This project explores the potential of the local clustering coefficient as a feature of popularity for books and genres.
The clustering coefficient gives a strong metric of correlation between books bought together, so it can be used to determine a possible order of the elements, obtained by analysing how often a book appears in a triangle.
Given the values of the local clustering coefficient, we investigated the possibility of approximating the salesrank order for the books. We also compare the order for the different genres that can be obtained using the salesrank and the clustering coefficient of the different books within the genre.
Then we try to classify each book into 4 different categories based on a new joint definition of popularity that we came up with, with the objective of capturing the nature of the books.

2 Data

Our dataset[1] available at this *link* contains information obtained by crawling the Amazon website and contains product metadata and review information about 548552 different products (Books, music CDs, DVDs and videos). The data was gathered in summer 2006.

For each product, we have important information available:

- Salesrank: Provides us with the 'popularity' of products according to Amazon as a rank number. In case of an unavailable rank, a value of -1 is used.

*id: 2133045, email: paulrenewenyi.cariou@studenti.unipd.it
†id: 2121719, email: michele.sprocatti@studenti.unipd.it
‡id: 2128872, email: riccardo.zuech@studenti.unipd.it

- Categories: Provides us with the tree of categories for each product (subgenres for the books in particular).
- Similar: Reports the ASINs of other products that are usually bought together with the one considered.

Our analysis only covers products in the category "Book", leaving us with a total of 393561 nodes. From our analysis, we also find that there are a total of 11629 genre-like entries with a maximum depth of 7 in the hierarchy.

3 Experiments

3.1 Setup

In order to run our experiments, we used Google Colab with default configuration[1]. Additionally, we used the following libraries:

- NetworkX: management of the graph and utilities.
- Scipy: nDCG score.
- Statistics module: quantiles and other statistics.
- TreeLib: management of the tree of genres.

3.2 Methodology

The fundamental elements of our analysis are the clustering coefficient and salesrank values of the books in the network. Given them, we need to define the notion of popularity as bounds on the set of values assumed by the clustering coefficient and salesrank. In addition, we need to check whether the clustering coefficient is a true feature of popularity. Then, we also need to find a way to translate the notion of popularity of books into popularity of genres. Finally, we want to give a joint definition of popularity for books in order to see their distribution under different genres.

In the following paragraphs, we go through the process we used to find all those elements.

Computing the local clustering coefficient: We apply the exact local clustering coefficient algorithm to the co-purchase network.

Defining the notion of popularity according to clustering coefficient and salesrank: We use quantiles of three steps on the values of cc and salesrank to define the bounds on "unpopular", "average" and "popular". In this phase, we prune invalid entries, i.e. entries of negative salesrank or zero local clustering coefficient, since they hold no informative value. The 3-steps quantile values obtained are:

[1] Intel(R) Xeon(R) CPU @ 2.20GHz, 13 GB of RAM

- Clustering coefficient: $[0.\bar{3}, 0.\bar{6}]$
- Salesrank: $[285111, 690611]$

Compute the popularity of genres at different sub-genres levels: To tackle this challenge, we build a tree-like data structure where each node is a genre and its children the sub-genres; given a popular book, starting from the leafs of the tree (most specific sub-genres of the book) travel upwards toward the root updating an internal counter (payload) in the nodes.

Compare the rankings of popularity given by the two measures: We decided to use the nDCG (normalised discounted cumulative gain) score, which gives us a value between 0 and 1 measuring the quality of the ranking obtained against the ground truth, i.e. the ranking of popularity produced by cc against the one produced by salesrank. We opted to use nDCG in particular for its usefulness in evaluating information retrieval systems, which given the specific context of our analysis (Amazon co-purchase network) is particularly fitting.

Joint definition of Popularity In table 1 the joint definition[2] of popularity we came up with can be found; the idea is to capture the nature of a book given by the combination of the two metrics.

	High cc	**Low cc**
High salesrank	Popular	Standalone best seller
Low salesrank	Central but doesn't sell well	Unpopular

Table 1: Table of popularity

3.3 Results:

Now that we have introduced the building blocks of our analysis, we briefly show its results.

3.3.1 Strength of cc as a popularity metric

One of our objectives is to check whether the local clustering coefficient can be used as a popularity metric for both books and genres.

To test this, we first compute the nDCG score of the ranking of books produced by the clustering coefficient against the ground truth, that is, the relative salesrank values. In this case, we obtain an nDCG score of 0.8857, a high value that suggests a strong overlap between the ranking by cc and by salesrank.

[2]The joint definition has slightly changed from the two project proposals.

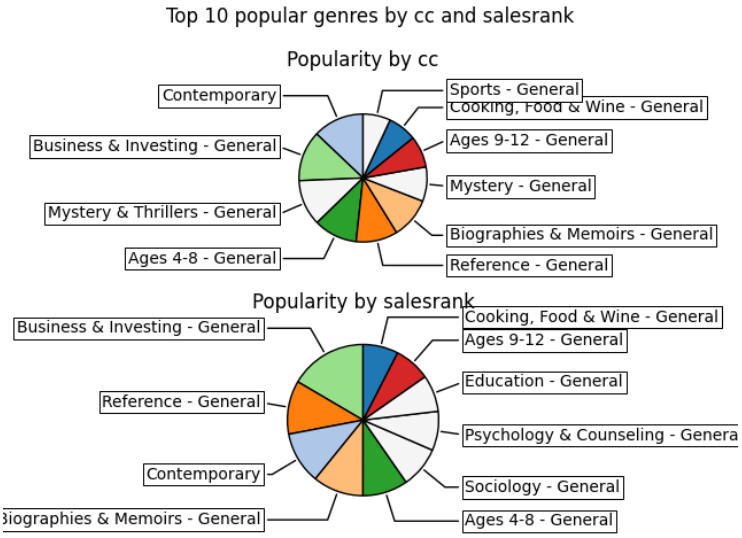

Figure 1: Comparison of top 10 popular genres by both local clustering coefficient popularity (top) and salesrank popularity (bottom).

For genres, we instead use the tree of genres to build a ranking of the subgenres in the leaves by counting the number of popular books according to cc under that subgenre. Then comparing the ranking produced with the ground truth, that is, the number of books popular according to salesrank, we obtain a really high nDCG score of 0.9779.

Given these results, it can be seen that our idea gives a very precise ordering for genres but also a very good ordering for the books, highlighting how the clustering coefficient can serve as a popularity metric almost as strong as the salesrank for retrieving the most popular elements.

3.3.2 Popularity of genres

Given the previous results, we now want to graphically show the rankings according to both the cc and salesrank.

Figures 1 and 2 show our results, and again we see a significant overlap in the ranking between the two popularity metrics at the top. In the repository, the code to produce the same plots for the top genres under a different specific ancestor or depth level can be found.

3.3.3 Distribution of joint popularity for different genres

Now comes into play our joint definition of popularity; here we use again the tree of genres, and for each book update an internal counter from the leaves it falls into (its most specific subgenres) up to the root, for the joint category the

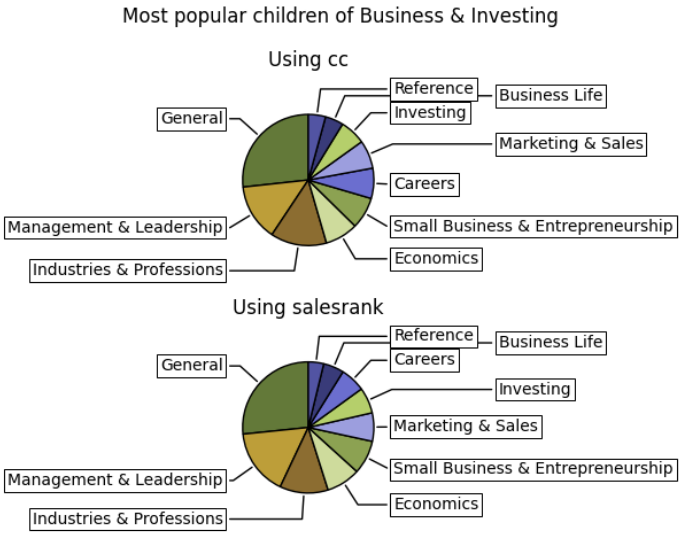

Figure 2: Comparison of top 10 popular subgenres under Business by both local clustering coefficient popularity (top) and salesrank popularity (bottom).

book belongs to.

We omit the plot of the entire set of leaves of the tree of genres for obvious reasons; however, in Figure 3 we can see how the joint categories are distributed in the genre of Romance (chosen arbitrarily; note that this result can easily be extended to any genre in the tree by adapting the code).

This type of analysis is quite interesting, since it allows us to visually see the proportion of "agreement" between clustering coefficient popularity and salesrank popularity. More specifically, by looking at the plots, one can appreciate how the combined popular and unpopular joint categories cover a significant area of the plot, signaling again an interesting overlap between cc popularity and salesrank popularity. Additionally, another common trend that can be observed is the really small area covered by the Central category under all the most specific subcategories.

Finally, in Table 2 we report the overall distribution of the books in the joint categories we defined; consider that all invalid entries fall into the relative low cc/salesrank measure.

The notable result here is that the Popular joint category, which indicates an agreement between cc and salesrank for "high popularity" of the books, is quite contained; if we combine this observation with the high nDCG score that we have seen previously, this indicates that the cc metric is really good in returning the most popular books in the correct order, but falls off near the bottom of the ranking. Another interesting result is that there is a low percentage of disagreement between high cc and salesrank, meaning that there is a small chance of false positives when using cc as a popularity metric.

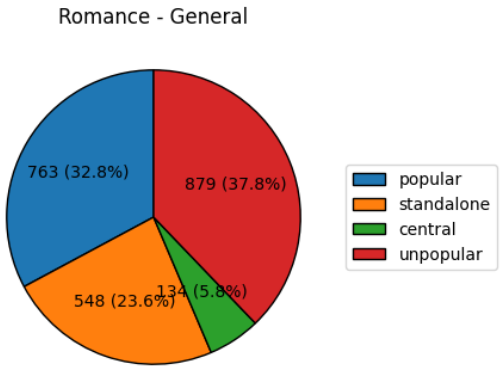

Figure 3: Distribution of books under the joint definition for the Romance genre.

	High cc		**Low cc**	
High salesrank	Popular	$\approx 12.3\%$	Standalone	$\approx 31.9\%$
Low salesrank	Central	$\approx 4.8\%$	Unpopular	$\approx 51\%$

Table 2: Table of popularity with distribution percentages.

4 Conclusions

After analysing the results in Section 3.3, we can suggest that the local clustering coefficient is a good measure to approximate the order of the most relevant books in this particular graph, but it is also a stronger measure for analysing the popularity of subgenres in our tree and approximating the ground-truth order obtained by considering the salesrank of the different books.

Our joint definition allows us to see the distribution of the books under their joint categories for each node, i.e. genre, in the tree. The results obtained show that the popular and unpopular categories together cover a large area of the pie, giving the same strength signals as the nDCG score results, since the two measures agree very often. However, the distribution of the books in the joint categories tells us that the cc is a really strong metric in retrieving the most popular books and genres (as the nDCG indicates) but falls off towards the bottom part of the ordering. This result tells us that the cc can be used extremely well in the context of retrieving the most popular books and genres, a particularly interesting result given the setting of the dataset, but it cannot be used in the context of a total ordering.

References

[1] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

5 Contribution and Additional Details

5.1 Working Fraction

- Paul Cariou: 15 %
- Michele Sprocatti: 42.5 %
- Riccardo Zuech: 42.5 %

5.2 Detailed Contribution

5.2.1 Paul Cariou

Paul came up with the joint definition, and then he has also corrected it. He wrote the very first version of the final report. He participated equally to the other members during the process of coming up with the idea and writing of the first proposal.

5.2.2 Michele Sprocatti

Michele wrote the code to parse the dataset to build first the dictionary and then the book graph, and also wrote the code to generate the tree of subcategories. He wrote the motivation section (1) and the conclusions section (4), and collaborated in general in the writing of the final report. He participated equally to the other members during the process of coming up with the idea and writing the first proposal. Michele also collaborated on writing the mid-term report.

5.2.3 Riccardo Zuech

Riccardo came up with the idea of using the nDCG score to compare the two orders, wrote the experimental section (3) and collaborated in general in the writing of the final report. He wrote the code for the analysis. He participated equally to the other members during the process of coming up with the idea and writing the first proposal. Riccardo also collaborated on writing the mid-term report.

5.2.4 Link to the Repository

LfN_AmazonBookAnalysis[3]

[3]The dataset is not there because it is too large to be uploaded. We have provided a link in the README.