rmNoto Sans

Comparative analysis of structural, learned, and hybrid feature representations for graph clustering

Learning From Networks - Project Proposal

Leonardo Gusson, Luca Rao, Chiara Frizzarin

14 novembre 2025

1 Motivation

This project aims at solving a question: is it possible to use a co-purchasing network to find out the actual categories of products considered? And which kind of features representation works better: structural, learned or hybrid? The analysis is based on the "Amazon product co-purchasing network metadata" which was collected back in 2006 by crawling Amazon website.

From this dataset a directed graph $G = (V, A)$ is going to be built:

- **Nodes**: Each node $v \in V$ represents a unique product in the Amazon dataset ($|V| = 548,552$);
- **Arcs**: An edge $(u, v) \in A$ exists if product v is often co-purchased after u ($|A| = 1,788,725$).

Each node comes with a set of information following this format:

- **Id**: Product id (number 0, ..., 548551)
- **ASIN**: Amazon Standard Identification Number
- **title**: Name/title of the product
- **group**: Product group (Book, DVD, Video or Music)
- **salesrank**: Amazon Salesrank
- **similar**: number $n \in [0, 5]$ of co-purchased products followed by a list of their ASINs (people who buy X also buy Y) (e.g. 2 B0001500VS B000002WA3)
- **categories**: Location in product category hierarchy to which the product belongs (separated by —, category id in [])
- **reviews**: Product review information: time, user id, rating, total number of votes on the review, total number of helpfulness votes (how many people found the review to be helpful)

All this information and the dataset can be found at Stanford Large Network Dataset Collection

2 Method

The objective of this project is to compare the effectiveness and performance of three different features sets to cluster a product co-purchasing graph, knowing that the number of clusters (i.e. product categories) is four.

2.1 Structural + semantic features

We are going to combine the following structural centrality scores in a feature vector:

- **PageRank** $p(v)$ allows us to calculate the popularity of each product, in terms of important products co-purchased with other important products.
- **Closeness Centrality** $c(v)$ measures the importance of a product in the graph in terms of "trendsetters", i.e. it measures the closeness of a node to the others.
- **Betweenness Centrality** $b(v)$ represents crucial information for discovering key products that introduce customers to new categories.
- **Clustering Coefficient** $cc(v)$ identifies how a product's neighbors are interconnected, indicating whether it belongs to specialized kits of products or connects different groups.

Then we compute a score $rw(v)$ analyzing the rating of the reviews, and a score $sr(v)$ based on the Amazon "salesrank", thus obtaining for each node v :

$$\vec{F}_{st}(v) = [p(v), c(v), b(v), cc(v), rw(v), sr(v)]$$

2.2 Node embeddings

As a second, parallel approach, we will move beyond single-score metrics and try to cluster the graph only using topological features through the use of **embeddings**:

$$\vec{E}(v) = [e_1, e_2, \ldots, e_d]$$

where the dimension d of the vector will be defined during the test phase (probably 64, 128 or 256).

2.3 Hybrid Approach

The two approaches described above are powerful but capture fundamentally different types of information. Each has "blind spots" that the other can cover: indeed, centrality scores are interpretable and capture global roles while graph embeddings are excellent at capturing local context and semantic similarity. A hybrid approach can leverages the strengths of both.
To tackle this we will construct a multi-dimensional feature vector, $\vec{F}(v)$, for each product v. This vector will serve as the input for downstream machine learning models. All features will be normalized (e.g. using Min-Max scaling or Z-score standardization) to bring them into a common range.
The vector for a product v is defined as a concatenation of its feature sets:

$$\vec{F}(v) = [\vec{E}(v) \frown \vec{F}_{st}(v)]$$

3 Intended experiments

3.1 Implementation

Given the graph size, we will implement approximated versions of pagerank, closeness centrality, betweenness centrality and clustering coefficient ourselves. On the other hand, the learned features vector $\vec{E}$ will be computed using the *Node2Vec* algorithm, whose dimension d will be defined during the implementation phase.
To identify the optimal node representation for clustering, we will conduct three independent experiments using K-means with $k = 4$. In the first approach we will tests the efficacy of classic, explicit graph metrics by feeding the standardized (e.g. using the Z-score) vector $\vec{F}_{st}$ into K-means. In the second approach we will evaluate the learned representation by first applying a dimensionality reduction technique (e.g. PCA or UMAP) due to the high dimensionality of embedding vectors, and finally we will use the reduced vector as K-means (or one of its approximate versions) input.
The same pipeline will be applied to the hybrid feature vector.

3.2 Machines used

We have access to the following machine, we will use the fastest one:

- Macbook Air M2 (8Gb RAM)
- Laptop Intel Core Ultra 9 (32Gb RAM)
- Intel Core i7-6600U (8 Gb RAM)

3.3 Experiments

To evaluate the three distinct approaches, we will perform a quantitative comparison based on two criteria: clustering quality and computational efficiency. For quality, we will measure how well the resulting k-means clusters align with the four ground truth categories using evaluation metrics such as the ARI (Adjusted Rand Index) or the NMI (Normalized Mutual Information). For efficiency, we will measure the total execution time required for each full pipeline.

4 Additional details

In this first part of the project we collaborated often in-person and came up with a proposal through out an equally-participated brainstorming. We first looked at the suggested large networks and tried to think what we could have focused on and which techniques we could have applied.
After that we consulted Gemini for better understanding if our ideas were enough challenging, but still doable (and also for proof-reading).
The formalization and the writing of the project proposal were done in presence, similarly to the other phases, and the work split is not so rigorous because again we discussed and tried to write together. Anyway it can be stated that:

- **Chiara:** Developed the project motivation (Section **??**) and led the dataset research and selection.
- **Luca:** Authored the core methodology (Section **??**).
- **Leonardo:** Designed the experimental setup and validation plan (Section **??**).