Project Proposal: Analysis of Community Detection Algorithms on Artificial Network

Vincent Wilmet
vincent.wilmet@student-cs.fr

Suer Lin
suer.lin@student-cs.fr

Lu Wang
lu.wang@student-cs.fr

Asrorbek Orzikulov
asrorbek.orzikulov@student-cs.fr

1 Introduction

1.1 Motivation

Detecting communities has grown into a fundamental, and highly relevant problem in network science with multiple applications.it can help a brand understand the different groups of opinion toward its products, target certain groups of people or identify influencers,it can also help an e-commerce website build a recommendation system based on co-purchasing.Predicting co-purchased products based on the user's previous order history can help online shopping sites recommend suitable products to users. Predicting the success of co-purchased products can help sites increase revenue [4].

1.2 Problem definition

In this project,we will do experimental evaluation of algorithms and models on co-purchasing graph data from the amazon product co-purchasing dataset. The data was collected by crawling Amazon website and contains product metadata and review information about 548,552 different products (Books, music CDs, DVDs and VHS video tapes).[1]

1.3 Related work

The first step of the research is data analysis, which includes single item analysis (sales rank analysis, item rating analysis and time of first review analysis) and pair analysis (Pearson similarity based on reviews, category similarity and title similarity. The second step is data preprocessing). The third step includes the model training and evaluation. Last step is to do comparison between different algorithms and result analysis.[3]

2 Methodology

2.1 Data set

We decided to use the ogbn-products dataset from the Stanford Open Graph Benchmark (OGB) [2].

The ogbn-products dataset is an undirected and unweighted graph, representing an Amazon product co-purchasing network. Nodes represent products sold in Amazon, and edges between two products indicate that the products are purchased together. Our task is to process node features and target categories. Specifically, node features are generated by extracting bag-of-words features from the product descriptions followed perhaps by a Principal Component Analysis to reduce the dimension to 100.

2.2 Task

The task is to predict the category of a product in a multi-class classification setup, where the 47 top-level categories are used for target labels.

One of the ways we are considering to provide novel insight to this benchmark is by implementing a more challenging and realistic dataset. Will aim to do this by intentional splitting. Instead of randomly assigning 90% of the nodes for training and 10% of the nodes for testing (without use of a validation set), we use the sales ranking (popularity) to split nodes into training/validation/test sets. Specifically, we sort the products according to their sales ranking and use the top 10% for training, next top 2% for validation, and the rest for testing. This is a more challenging splitting procedure that closely matches the real-world application where labels are first assigned to important nodes in the network and ML models are subsequently used to make predictions on less important ones.

2.3 Architecture

We are not yet sure which model works best, but we will explore some of the methods explored in class, and explored in our literature review like MLP, Node2Vec, GCN, SIGN, SAGN or perhaps even a combination of multiple algorithms.

3 Evaluation

For evaluation, we will be using the following 4 metrics, denoting the predicted scores as $\hat{\mathbf{y}}$ and the true labels as $\mathbf{y}$. The first 2 metrics measure the performance for the most frequent 5 labels, but they will be poor performance indicators if a model learns to always predict the top labels. Therefore, we will also use the last two metrics that encourage the model to predict infrequent labels more accurately than frequent labels.

1. Precision for the top-5 labels:

$$P@5 = \frac{1}{5} \sum_{l \in top_5(\hat{\mathbf{y}})} \mathbf{y}_l$$

2. Normalized Discounted Cumulative Gain for the top-5 labels:

$$nDCG@5 = \frac{\sum_{l \in top_5(\hat{\mathbf{y}})} \frac{\mathbf{y}_l}{\log(l+1)}}{\sum_{l=1}^{k} \frac{1}{\log(l+1)}}$$

3. Propensity-scored precision for the top-5 labels:

$$P@5 = \frac{1}{5} \sum_{l \in top_5(\hat{\mathbf{y}})} \frac{\mathbf{y}_l}{p_l}$$

4. Propensity-scored Normalized Discounted Cumulative Gain for the top-5 labels:

$$PSnDCG@5 = \frac{\sum_{l \in top_5(\hat{\mathbf{y}})} \frac{\mathbf{y}_l}{p_l \log(l+1)}}{\sum_{l=1}^{k} \frac{1}{\log(l+1)}}$$

where $top_5(\hat{\mathbf{y}})$ is the first 5 indices when $\hat{\mathbf{y}}$ is sorted in descending order and p_l is the marginal probability of a relevant label l being observed.

References

[1] M. Girvan and M. Newman. "Community structure in social and biological networks." In: *Proceedings of the National Academy of Sciences 99, 7821–7826* (2002).

[2] W. Hu et al. "Open Graph Benchmark: Datasets for Machine Learning on Graphs". In: *arXiv preprint arXiv:2005.00687* (2020).

[3] K. Pearson. "Notes on regression and inheritance in the case of two parents". In: *Proceedings of the Royal Society of London. 58: 240–242* (1895).

[4] Z. Yang, R. Algesheimer, and C. Tessone. "A Comparative Analysis of Community Detection Algorithms on Artificial Networks". In: *Sci Rep 6, 30750* (2016).