A Comparison of Book Recommendation Models

Linka Mitome
l.mitome@student.uva.nl

Bartolomeo Attolico
b.attolico@uva.nl

Naglis Karoblis
n.karoblis@student.uva.nl

Jules Talbourdet
j.talbourdet@student.uva.nl

1 Introduction

A recommendation system is a machine learning algorithm that is used in order to give suggestions and recommendation to an individual based on its interactions with others or with specific products. In our society, they are considered to play a crucial role in shaping humans behaviors. For instance, Amazon uses them to target clients, with products that match their needs and interests, while Netflix, uses them to offer to its users movies that they might like. On Netflix, this is in part done by analyzing users viewing trends, ratings, and preferences to then suggest movies that are in line with what the user likes. In fact, ”because of their capability to predict consumer interests and desires on a highly personalized level, recommender systems are a favorite with content and product providers.” [1]. In a paper released by Netflix, it was stated that that their recommendation system ”in total influences choice for about 80% of hours streamed” [2].

There are three different types of recommendation systems method: content based, collaborative, and hybrid. Collaborative filtering (CF), is done based on the user-user similarities. In CF a user will be recommended an item, that another user with a similar profile, liked. Instead content based filtering (CBF) recommendation are based on the similarity between different contents. By providing a recommendation to a user based on the similarity of an item - the user interacted with - to other items. Thus, in this case the similarity is based on the content of the item, and its similarity to other contents, rather than the similarity of the user profile to another user. However, most of the models used by companies (in production), are hybrid, meaning that they use a combination of both CF and CBF [3]. This is a better approach, as these models are able to take into account, both the similarities between users, and the similarities between contents. See [4], to understand how hybrid recommendation systems can improve the results of both CF and CBF methods.

Recently, Graph Neural Networks (GNNs) have emerged as a successful new recommendation system hybrid method. GNNs, ”are a class of neural networks tailored for handling data organized in graph structures”. The aim of this research is to provide a comprehensive analysis of the standard recommendation systems models that use CBF and CF methods, and compare them a more advanced hybrid method which will leverage GNNs as a structure. In order, to evaluate the models performance these metrics are going to be used: RMSE, Precision@10, Recall@10, F1-score@10, see [5] to gain a better understanding of the different metrics to use to evaluate a recommendation system. The research will be done using a book data set, which is divided into three different sub data sets: users, ratings, and books.

This research differs from others, as it has the objective to provide a straightforward overview of the two CBF and CF methods, while comparing them to the more advanced hybrid method which uses GNNs. This gives a good understanding of the standard recommendation system methods, compare to more advanced architecture, in terms of the metrics highlighted above.

2 Related Works

2.1 Traditional Recommender Systems

Studies in the past 10 years have categorized recommender systems as one of the following main approaches: collaborative-filtering (CF), content-based filtering (CBF), and hybrid (a combination of the two) [6], [7]. CF methods are the most extensively applied approaches, and rely on past information of other users with similar interests to make predictions. Matrix factorizing techniques such as Single Value Decomposition (SVD) are commonly used to decompose the user-item interaction matrix into lower-dimensional matrices [7]. However, CF recommenders notably suffer from 'grey sheep' (when users or items do not fit into traditional clusters) or cold-start (when new users or items are added without any information) problems [8]. CBF is another popular approach, which utilizes attribute information of items to classify items and generate recommendations, using techniques such Naive Bayes Classifiers [7]. CBF mitigates the cold-start problem through the embedding of item information [8]. Hybrid methods combine the previous recommendation methods to reinforce their advantages and/or overcome limitations [7], [8].

2.2 Graph Neural Networks

Historically recommender systems have followed three stages of advancements, namely, shallow models (e.g. matrix factorization methods), neural models, and most recently, graph neural network (GNN). The rapid increase of graph data such as knowledge graphs and social networks in recent years has triggered a surge in studies on GNNs. This wave roots from advancements in convolutional neural networks (CNN) and graph representation learning (GRL). GNNs have been developed to extract structural information and high-level representations by integrating CNN and GRL [6].

In 2019, Fan et al. introduced one of the earliest models on GNNs for Social Recommendation *GraphRec*, which consistently outperformed baseline methods including a variety of matrix factorization and neural techniques, highlighting its superiority over previous methods [9]. Other models such as *LightGCN* have been introduced since then, similarly exhibiting substantial improvements over previous models, establishing its position as the 'state-of-the-art' [6], [10].

The superiority of GNN-based recommendation systems stems from their various enhancements over traditional models [6]. Firstly, GNNs are able to unify and leverage multiple forms of data, that traditional recommender systems were not able to fully leverage. For instance, collected data usually entails user profile (age, location, etc.), item features (price, category, etc.), and user-item interactions (ratings, purchase, etc.). While previous models were able to focus on a limited amount of information, GNNs are able to embed all user, item, and interaction data in the form of nodes and edges, thereby increasing their recommendation performance.

Secondly, GNNs are able to capture high-order connections which leverages the collaborative filtering effect to a greater degree, improving their recommendation accuracy. Previous models, trained on user-item interaction data were bound by first-order connectivity, while the GNNs are able to capture higher-order connectivity, and thus, allows for the collaborative filtering effect to be expressed as "multi-hop neighbors" [11].

Thirdly, GNN-based models are able to incorporate and utilize non-target (implicit) behaviors (e.g. click, add-to-cart) to the data. Previous works which only utilize target (explicit) behavior (e.g. purchase) as a supervision signal, suffer from sparse data. Therefore, GNNs are able to significantly improve recommendation performance by incorporating various implicit behaviors, and thus, encoding semi-supervised signals.

3 Data

Our data is composed of three distinct datasets. One concerns users, another one books, and the last one includes ratings linking users to books. Each dataset is linked by a common feature. The ratings dataset shares a feature with both the users and books datasets which allowed us to concatenate the different datasets into a single pandas dataframe. Based on that dataframe, we will be able to easily preprocess the data that needs preprocessing, and apply our baseline models as well as creating our graph.

4 Methodology

4.1 Preprocessing

The dataset did not require a significant amount of preprocessing. The important preprocessing steps that were carried out are as followed: keeping only the users and books that figure in the ratings datasets, and mapping each book's ISBN to a unique integer for compatibility for our graph. We then, for the GNNs, create user nodes, book nodes, and edges weighted based on the rating that a user gave to a book. Subsequently, we added to the users nodes and the books nodes the information we had about them. This allowed to embed information in our graph helping the recommending power of our algorithm.

4.2 Base models

For our base models, we compared two types of algorithm User-Based Collaborative Filtering (CF) and Content-Based Filtering (CBF). In this approach we use a basic SVD on users that have rated the most books and books that have been the most rated. We decided to use these techniques as a baseline as they are very popular and are now a basis in the field. Using the same algorithm for both approach ensures consistency and strong comparability in the results.

4.3 GNN

For our GNNs, we embedded our data into a bipartite heterogeneous graph. The type of data inherently oriented us toward such a graph. We have different types of nodes, users and books, which entails that our graph has to be heterogeneous. Moreover, since each user is connected to at least one book and one book to a user, our graph has to be bipartite. The goal of this approach is to get a graph that has a precise and accurate representation of our data.

4.4 Metrics

- **Root Mean Square Error** measures the average difference between predicted book recommendations and actual books determined either by user-to-user preference (CF) or individual user ratings (CBF).

- **Precision** for top 10 recommendations indicates the proportion of actually relevant books among the top 10 recommended books. It measures how many of the books recommended to the user are actually of interest to them.

- **Recall** for top 10 recommendations measures the ratio of correctly predicted relevant books in the top 10 recommendations to the total number of relevant books that could have been recommended. It indicates how well the system identifies all the relevant books for the user within the top 10 recommendations.

- **F1 Score** for top 10 recommendations provides a balanced evaluation of correctly recommended books based on Precision and Recall metrics.

5 Results and Discussion

To evaluate the performance of book recommendations for both baseline models and GNN, 4 main aforementioned metrics were considered, and are outlined in table 1.

Model	**CBF**	**CF**
RMSE	1.64	3.52
Precision	0.78	0.87
Recall	0.92	0.51
F1 Score	0.73	0.46

Table 1: Metrics for Book Recommendations

Overall, the Content-Based Filtering method performed better in almost every aspect for the following metrics. Since the method is based on individual user's book ratings, the diversity of book data (4 useful columns) may have played a significant role in the metrics' high scores. On the other hand, the Collaborative Filtering approach provided less accurate results, indicating that it struggled with the sparsity of user-to-user interaction data. This may suppose a less effective link in capturing the different preferences of users in a dataset with sometimes limited user data.

In particular, CB has a significantly lower RMSE (1.64 vs. 3.52), indicating a more accurate prediction of book recommendations. However, CF does achieve higher precision (0.87 vs. 0.78), suggesting its top recommendations are more relevant. However, CB excels in recall (0.92 vs. 0.51) and F1 Score (0.73 vs. 0.46) indicating that an individual's book rating 'history' is more suited to capturing relevant books and providing more consistently relevant top 10 recommendations.

It is worth outlining that throughout the process of evaluating baseline models, inconsistencies of metric results were encountered. In particular, especially a high difference between the two methods' metrics. This has been solved by choosing a more optimal threshold.

6 Limitations and Future works

The main limitation of this research is that we were not able to implement the GNN model, as it proved more challenging than we anticipated to build the GNN architecture. We were able to implement the graph structure, through the use of the Deep Graph Library (DGL). As previously outlined, we built a bipartite heterogeneous graph, which has two sets of nodes: books and users, and they are linked with an edge: book-rating. However, since our graph was not one-to-one, as not every user read every book, it proved challenging to build an architecture, which had to handle different sized tensors. Another limitation of this research is that the dataset that was used was very limited in the information it contained. For instance, data on the books summary, genre and users' age, would have possibly helped in the improvement of the different models (CF and CBF).

As future research, it would be interesting to repeat the initial objective of this study, which was to compare classical recommendation system methods such as CF and CBF, to an hybrid method with GNNs as a model. It is important to consider well which library to use when building the GNN. In our research DGL was used, and it proved to have less user friendly documentation than PyTorch Geometric (PyG). Another recommendation would be to use a more varied data set, which we believe would make the results more of the different models, more generalizable to a real-world scenario.

7 Conclusion

Throughout the process of testing different baseline models (CF and CBF) for book recommendations, the content-based method was found more accurate. That is, the approach of positively predicting other books based on individual user's previous ratings was observed as less-error-like. This was exemplified by the majority of CB's different metrics outperforming the method of CF. As the initial hypothesis for the best performance of the hybrid (GNN) model, the encountered architectural obstacles did not allow to completely prove the set up idea. Nevertheless, this does not negate the possible better performance of the GNN-based recommender system in regard to the aforementioned baseline models, thus, further research needs to be conducted.

References

[1] NVIDIA. "What is a recommendation system?" [Accessed June 1, 2024]. (2023).

[2] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, Dec. 2016, ISSN: 2158-656X. DOI: `10.1145/2843948`. [Online]. Available: `https://doi.org/10.1145/2843948`.

[3] B. Rocca, "Introduction to recommender systems," *Towards Data Science*, 2019, [Accessed June 2, 2024]. [Online]. Available: `https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada`.

[4] K.-Y. Jung, D.-H. Park, and J.-H. Lee, "Hybrid collaborative filtering and content-based filtering for improved recommender system," in *Computational Science - ICCS 2004*, M. Bubak, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 295–302, ISBN: 978-3-540-24685-5.

[5] P. Aher, "Evaluation metrics for recommendation systems: An overview," *Towards Data Science*, 2023, Accessed: 2024-06-02. [Online]. Available: `https://towardsdatascience.com/evaluation-metrics-for-recommendation-systems-an-overview-71290690ecba`.

[6] C. Gao, Y. Zheng, N. Li, *et al.*, "Graph neural networks for recommender systems: Challenges, methods, and directions," *CoRR*, vol. abs/2109.12843, 2021. arXiv: `2109.12843`. [Online]. Available: `https://arxiv.org/abs/2109.12843`.

[7] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intelligent Data Analysis*, vol. 21, no. 6, pp. 1487–1524, Nov. 2017, ISSN: 1571-4128. DOI: `10.3233/ida-163209`. [Online]. Available: `http://dx.doi.org/10.3233/IDA-163209`.

[8] Y. Li, K. Liu, R. Satapathy, S. Wang, and E. Cambria, *Recent developments in rec-*

ommender systems: A survey, 2023. arXiv: 2306.12680 [cs.IR].

[9] W. Fan, Y. Ma, Q. Li, *et al.*, *Graph neural networks for social recommendation*, 2019. arXiv: 1902.07243 [cs.IR].

[10] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, *Lightgcn: Simplifying and powering graph convolution network for recommendation*, 2020. arXiv: 2002.02126 [cs.IR].

[11] Y. Gu, H. Sun, Y. Wang, and H. Shi, "Improve robustness of graph neural networks: Multi-hop neighbors meet homophily-based truncation defense," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 1–8. DOI: 10.1109/IJCNN54540.2023.10191493.