Collaborative Filtering for Movie Recommendation

Comparing Classical and Neural Algorithms

Aditya Srivastava
Dept. of Computer Science
University of Colorado
Boulder
aditya.srivastava@colorado.edu

Harsh Gupta
Dept. of Computer Science
University Of Colorado
Boulder
harsh.gupta@colorado.edu

Akimun Jannat Alvina
Dept. of Electrical Engineering
University of Colorado
Denver
akimunjannat.alvina@ucdenver.edu

Niharika Narasimhiah Govinda
Dept. of Applied Mathematics
University of Colorado
Boulder
nina3335@colorado.edu

ABSTRACT

With the rising consumption of digital media, personalized content recommendation has become essential to maintain user engagement. Thus, the development of effective recommendation systems has become paramount, with collaborative filtering methods emerging as one of the key methods for doing so. In this paper, we propose the use of the MovieLens dataset to compare the classical matrix factorization algorithm to a newer neural algorithm and evaluate it on popularly used metrics.

1 Introduction

Viewers today have unlimited access to a wide array of films and shows, bringing in a new era of cinematic choice. While this abundance offers an exciting number of viewing opportunities, it also presents a considerable challenge for viewers in that they must traverse this vast library successfully and find content that suits their own tastes and preferences. Thus, recommendation systems have emerged as a key element in improving user experiences at a time when information and entertainment are becoming increasingly conflated. These systems, sometimes known as “recommender systems”, have developed into a crucial component of a variety of online platforms, from streaming services to e-commerce websites. Movie suggestion has distinguished itself as a compelling and well-accepted use case among the large range of applications for recommender systems and massively impacts all stakeholders, i.e. the viewers consuming content, the media houses producing movies, and the platform itself driving engagement. These points served as our motivation to pursue this project.

Recommendation systems can be generally classified into three types - content-based, collaborative filtering based, and approaches that are hybrids of both.

Content-based systems look at the attributes of objects (such as books, movies, and articles) and past user preferences. They then offer suggestions for products with features the viewer has previously liked. Collaborative filtering recommendation systems leverage the collective choices of users to make personalized predictions. These algorithms discover trends, correlations, and affinities among users by examining user interactions like movie ratings and viewing histories. Hybrid recommendation systems combine the strengths of both content-based filtering and collaborative filtering methods. By leveraging multiple recommendation techniques, they aim to provide more accurate and diverse suggestions.

In this paper we focus on collaborative filtering methods, and explore two algorithms, Matrix Factorization, and Neural Collaborative Filtering.

2 Literature Review

Collaborative filtering (CF) is an RS technique that generates entity preferences from the preferences of the other entities in the group. These entities can be either users (recommend items used by similar users)[1], items (recommend users for the items based on users of similar items).[2, 3] CF can generally be grouped into three categories: latent factor models, memory or neighborhood based approaches, and hybrid models.[4] Latent factor

models, such as Matrix Factorization[5, 6] and SVD[7], project users and items into a shared embedding space and use their latent features to find correlations between the entities and have been very successful. Neighborhood based approaches, such as clustering were used at Amazon[8], to form recommendations by identifying groups of similar users or items based on previous interaction history. Hybrid approaches, such as SVD++[9] combine both approaches, often making the best of the local correlations discovered by the former and the global correlations discovered by the latter.

Deep learning (DL) has seen exponential growth over the past decade, and is currently considered the state-of-the-art in machine learning, with wide ranging applications in computer vision, natural language processing and other domains.[10, 11, 12] DL techniques have also been applied to recommendation systems, with all of the current benchmarks being dominated by methods such as graph representation learning[13] and variational autoencoders[14] and even exploration of LLMs such as chatGPT for recommendation[15].

3 Dataset

3.1 Description

The MovieLens dataset[17] contains records of movie ratings by the users of the MovieLens website and is maintained by the GroupLens organization, which is a research group at the University of Minnesota that maintains and provides access to the different MovieLens datasets. The dataset was chosen due to its popularity as a benchmark for testing new recommendation algorithms and has been extensively peer-reviewed.

The data set contains explicit user-item interactions in the form of ratings (from 1 to 5, inclusive) and contains user demographics (age, occupation, etc.) and movie features (genre, year of release, etc.), in a mix of numerical and categorical attributes.

The dataset comes in various sizes, but we plan on using the one with 100,000 samples in our work, due to its smaller size being more convenient to run tests on.

3.2 Preprocessing

The stock dataset comes as a set of CSV files containing user-item-rating tuples, user-feature tuples and item-feature tuples. We are going to be treating the dataset differently between analysis and modeling, and thus, we follow different preprocessing steps for both. The difference arose from our decision to factor explicit ratings and features in our data analyses, but perform the modeling on implicit (binary) interactions, where if a user has rated a movie, we consider it as the user having watched it, otherwise, the user hasn't watched the movie.

During analysis we kept all features of the dataset, using Min-Max scaling to scale numerical attributes to [0, 1], and One-Hot encoding categorical features.

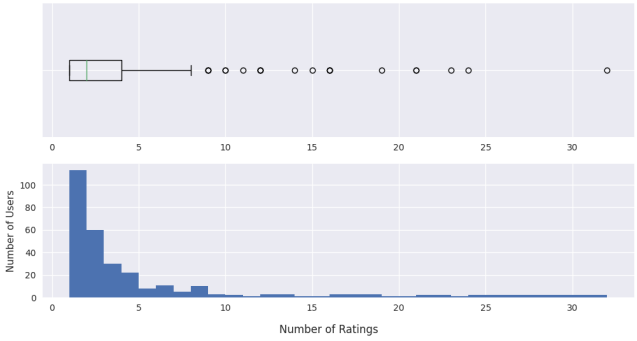
For modeling the data, we first discarded all features but the ratings, since we are not testing content-based filtering algorithms. At this point, every explicit rating in our dataset is an implicit positive sample. To produce the implicit negative samples for each user, we now sample randomly from items that they haven't rated.

For evaluation using the HR@10 and NDCG@10 metrics (refer to section 5 for more details), we also created lists of 100 negative samples, and 1 positive sample that was held out of the training set for that user.

Finally, when creating the training, validation and test splits, we made sure that users who had rated a single item were part of the training set to avoid the cold-start problem[21], i.e. there may exist movies that have too few user ratings to be recommendable, and there may exist users who have rated too few movies to be adequately profiled. Additionally, items that have too few ratings or too many ratings are considered to be outliers and may be dropped during training.

3.3 Analyses

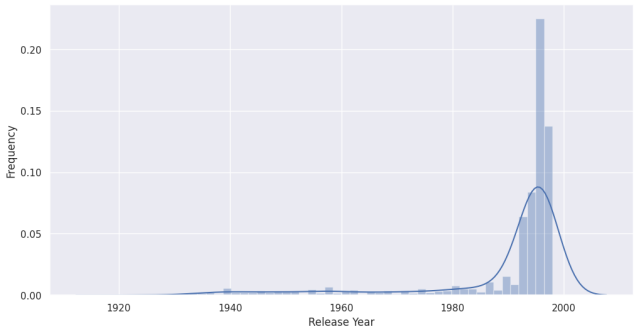
Our goal was to find patterns in the data that we could exploit to get better results and to confirm if our model is performing correctly. The patterns can be used to ascertain that our model's predictions fit the observations gleaned from the data. This is especially helpful in the case of unlabeled samples.

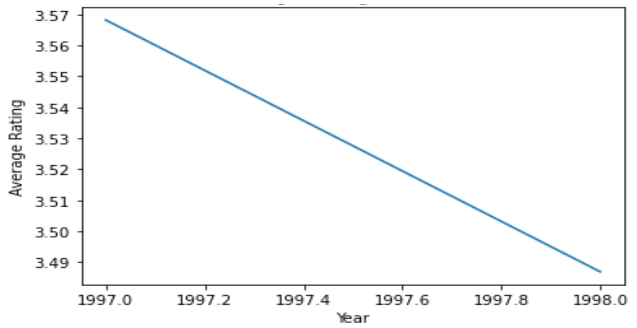


We started by performing a simple frequency analysis over various features in the dataset. In the figure above we can see the distribution of ratings over users. The distribution is left-skewed, indicating that while many users rate only a handful of movies, a smaller subset of users are highly active, contributing ratings for hundreds of films. This suggests a difference in user behavior, with casual raters and a few avid movie watchers who rate extensively.

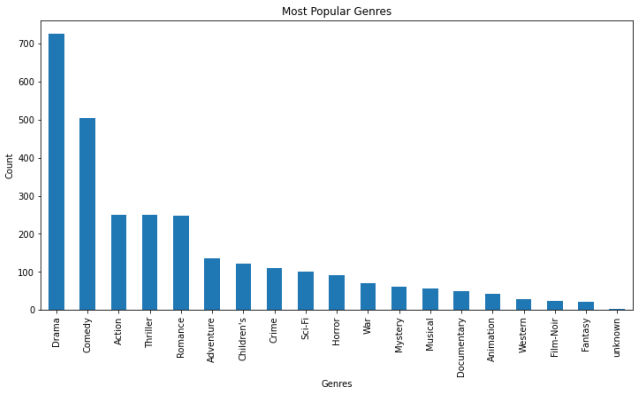
We also found the distribution of ratings over movies, as shown in the figure below. This too is a left-skewed graph, and the fact that many movies have a low number of ratings might suggest that either the dataset has a lot of niche films or that many users tend to rate only the movies they are familiar with, passing over lesser-known films.

Both of the above observations indicate that our model may be likely to suffer from the cold-start problem, since the user-item interaction matrix is extremely sparse, and there are many users and movies with only a few assigned ratings.



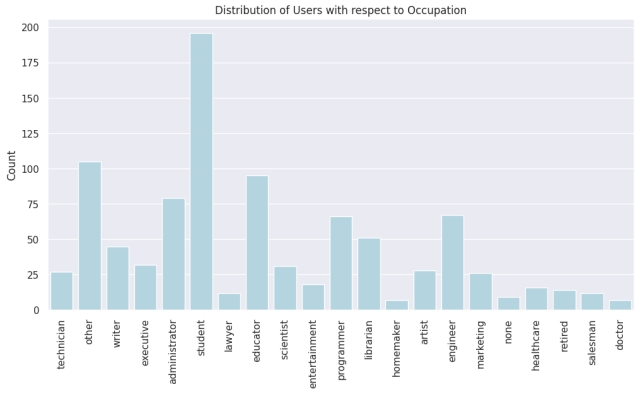


Following this we plotted a correlation matrix between all the features in the dataset. The matrix is too large to fit here and has been provided at the end of the document. An interesting correlation was discovered between the release dates of movies and their ratings. We can see clearly in the graphs above that as the number of movies released each year increases the average score goes down. This seems like an obvious inference considering that there are likely to be only a few good movies among a growing number of productions. Additionally, it is likely that users selectively go back to give high ratings to the few old movies that left a lasting impression.



Going back to frequency analysis, we found *Drama* to be the most popular genre, followed by *Comedy* and *Action*. Surprisingly, based on correlation scores, we can see that

the most frequent age group in our data (students, aged ~20 years), actually correlates negatively with the *Drama* genre of movies, rating it poorly compared to their counterparts.



The graph above reveals the distribution of users across various occupational categories. It is interesting to note that *students* emerged as having the greatest predilection to rate movies compared to other professional spheres. This could be attributed to students' tendency to consume a greater quantity of media and furthermore, a greater likelihood of rating the films they view. It is also noteworthy that occupations such as homemakers and doctors are less common, as indicated by the figure.

From the correlation matrix, we also see some correlation between genres and genders of the users, with women seeming to prefer *Romance* movies over men, and men seeming to prefer *Sci-Fi* over women.

Another notable correlation between features can be found between the genres themselves, where certain genres are likely to co-occur frequently, for example, *Animation* movies are likely to also be genre classified into *Children's* and *Musical*. Similarly, *Mystery* and *Thriller* genres also seem to co-occur.

Finally, we would like to bring attention to some likely specious and perhaps even damaging biases seen in our data. For example, the gender and occupation correlation scores seem to strongly imply that a *male* user is more likely to be a *technician, programmer* or *engineer*, whereas a *female* user is more likely to have the *healthcare* and *homemaker occupations.* We might want to get rid of these demographic labels to avoid subjecting our model to these biases, however it may pose a significant challenge to do so without impacting recommendation quality.

4 Proposed Work

4.1 Method

We plan to compare two algorithms - Matrix Factorization (MF) which is a popular classical method, and Neural Collaborative Filtering (NCF) which is a newer method employing neural networks. Both methods are essentially regression models that rate items on a scale of 0 to 1.

4.2.1 Matrix Factorization (MF)

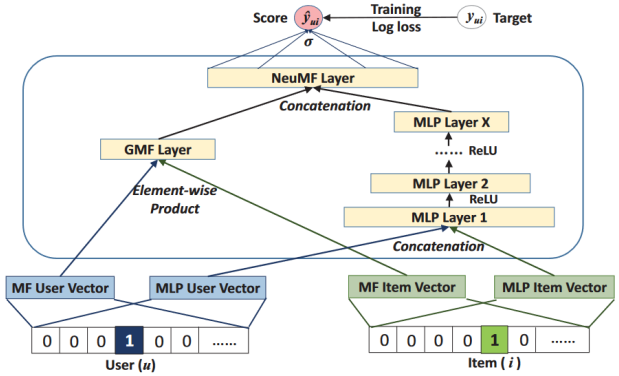
Let vector u_i represent user i, and vector e_j represent item j. The rating r_{ij} given by the user to the item can now be calculated as the inner product of the two vectors, as shown below;

$$r_{ij} = u_i^T \cdot e_i$$

This difference between this rating and the true interaction value (1 if watched, and 0 if unwatched) provided in the data, gives the pointwise loss, and can be backpropagated to the vectors using Stochastic Gradient Descent (SGD) to optimize them.

4.2.2 Neural Collaborative Filtering (NCF)

The CNF method (He et al., 2017)[18] extends MF, by adding a multi-layer perceptron (MLP), and passes a combined output of the MF and MLP subsystems through the final Neural MF (NeuMF) layer to produce the rating. This can be seen below.



This method also uses the same pointwise loss function as used in MF, and is also optimized with SGD.

5 Evaluation

For evaluation of the two methods, we use two commonly used leave-one-out metrics - Normalized Discounted Cumulative Gain (NDCG)[19] and Hit Ratio (HR)[20]. We randomly hold out one watched item, along with N unwatched items, and ask the model to rank them by rating. Intuitively, HR@K measures if the positive item ranked within the top K rated items, whereas NDCG@K also penalizes the lower rank of the positive item within those K items (the equations for the same can be found in the citations).

NDCG is calculated by dividing the discounted cumulative gain (DCG) of the ranked list by the DCG of the ideal ranked list, which is the list with the relevant items ranked in the most optimal order. NDCG ranges from 0 to 1, with higher values indicating better performance.

Hit Ratio can be calculated as the fraction of users for which the system ranks the positive item within the top K, out of all the users for which the predictions were made.

We plan on validating our results by using train, evaluation, and test splits of the, and finally comparing them to prior research, since both models have been well documented in the same. Since the dataset is well established as a recommendation systems benchmark, we don't expect any issues with our sample size. To maintain consistency with prior research utilizing the same dataset, we plan on keeping the outliers during testing, although we might remove the outliers during training to see if that improves our performance.

5 Milestones

We are working towards the following milestones:

1. **Data Preprocessing and Analysis:** We will perform an in-depth exploration of the dataset, calculate basic statistics, visualize data distributions, and identify potential outliers or anomalies. This will help us gain insights into user behavior and movie characteristics. The next step would involve data preprocessing to convert the data to ensure its suitability for building recommendation models. We expect to finish this milestone by mid-October. **[Completed]**

2. **Model Implementation and Testing:** We will implement the recommendation system models - Matrix Factorization and Neural Collaborative Filtering using suitable libraries or frameworks. The next step would involve testing the model where we will split the dataset into training and testing sets to evaluate the initial model performance. Next, we would assess the recommendation accuracy by utilizing common evaluation metrics like NDGC and HR. Finally, we will fine-tune the models' hyperparameters to optimize their performance. We intend to finish this milestone by mid-November. **[In Progress]**

3. **Improving Model Performance:** We will try to modify our model to see if we can improve performance. We will also experiment with some non-collaborative filtering algorithms, and try to utilize more of the content provided in the dataset. We intend to finish this milestone by the end of November.

4. **Graph Neural Network:** As part of our feedback we were suggested to have an additional stretch goal, for which we will be implementing the current state-of-the-art Graph-based Hybrid Recommendation System [13], which will utilize both implicit and explicit features. **[Stretch Goal]**

6 Challenges

In our work till now, we've faced two big challenges;

1. **Dealing with bias:** As explained in section 3.3, our data shows a number of misleading biases. There are a few approaches we can employ to get rid of these biases, for example through stratified sampling or completely dropping certain features, however it is likely that any method we employ will have a negative impact on the recommendation performance.

2. **Preprocessing:** Training, testing and evaluating recommendation models is a complex task, since the objective function for the model, i.e. pointwise loss, may not make for as interpretable an evaluation metric as NDCG or HR. Unfortunately the two of them require considerably different preprocessing to be carried out, while also tracking outliers to mitigate as much of the cold-start problem as possible.

3. **The cold-start problem:** While we haven't started modeling our data just yet, from our data analyses

in section 3.3, we can see that our user-item interaction matrix is sparse and there are like a lot of users and items with very few ratings, making them more difficult to generate recommendations for. This will be an ongoing problem to solve as we proceed with modeling.

REFERENCES

[1] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. 1997. GroupLens: applying collaborative filtering to Usenet news. Commun. ACM 40, 3 (March 1997), 77–87. https://doi.org/10.1145/245108.245126

[2] Xue, Feng, et al. "Deep item-based collaborative filtering for top-n recommendation." *ACM Transactions on Information Systems (TOIS)* 37.3 (2019): 1-25.

[3] Gao, Min, Zhongfu Wu, and Feng Jiang. "Userrank for item-based collaborative filtering recommendation." *Information Processing Letters* 111.9 (2011): 440-446.

[4] Roy, D., Dutta, M. A systematic review and research perspective on recommender systems. *J Big Data* **9**, 59 (2022). https://doi.org/10.1186/s40537-022-00592-5

[5] Billsus, Daniel, and Michael J. Pazzani. "Learning collaborative information filters." *Icml*. Vol. 98. 1998.

[6] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. 2008. Matrix factorization and neighbor based algorithms for the netflix prize problem. In Proceedings of the 2008 ACM conference on Recommender systems (RecSys '08). Association for Computing Machinery, New York, NY, USA, 267–274. https://doi.org/10.1145/1454008.1454049

[7] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In Proceedings of the 10th international conference on World Wide Web (WWW '01). Association for Computing Machinery, New York, NY, USA, 285–295. https://doi.org/10.1145/371920.372071

[8] G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," in *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80, Jan.-Feb. 2003, doi: 10.1109/MIC.2003.1167344.

[9] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In SIGKDD.

[10] Chai, Junyi, et al. "Deep learning in computer vision: A critical review of emerging techniques and application scenarios." *Machine Learning with Applications* 6 (2021): 100134.

[11] Lopez, Marc Moreno, and Jugal Kalita. "Deep Learning applied to NLP." *arXiv preprint arXiv:1703.03091* (2017).

[12] Wang, Fei, Lawrence Peter Casalino, and Dhruv Khullar. "Deep learning in medicine—promise, progress, and challenges." *JAMA internal medicine* 179.3 (2019): 293-294.

[13] Zamanzadeh Darban, Zahra, and Mohammad Hadi Valipour. "GHRS: Graph-Based Hybrid Recommendation System with Application to Movie Recommendation." Expert Systems with Applications, vol. 200, Aug. 2022, p. 116850. Crossref, https://doi.org/10.1016/j.eswa.2022.116850.

[14] Kim, Daeryong, and Bongwon Suh. "Enhancing VAEs for Collaborative Filtering." Proceedings of the 13th ACM Conference on Recommender Systems, Sept. 2019. Crossref, https://doi.org/10.1145/3298689.3347015.

[15] Gao, Yunfan, et al. "Chat-rec: Towards interactive and explainable llms-augmented recommender system." *arXiv preprint arXiv:2303.14524* (2023).

[16] Jia Rongfei, Jin Maozhong, & Liu Chao. (2010). A new clustering method for collaborative filtering. 2010 International Conference on Networking and Information Technology. doi:10.1109/icnit.2010.5508465

[17] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872

[18] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. https://doi.org/10.1145/3038912.3052569

[19] https://towardsdatascience.com/demystifying-ndcg-bee3be58cfe0

[20] https://towardsdatascience.com/ranking-evaluation-metrics-for-recommender-systems-263d0a66ef54

[21] https://en.wikipedia.org/wiki/Cold_start_(recommender_systems)