Personalized Movie Recommendation System Using Collaborative Filtering and Performance Metrics Evaluation

Soma Parvathini, Belwin Julian, Siddharth Hegde
Rutgers University
svp98@scarletmail.rutgers.edu
bj273@scarletmail.rutgers.edu
ssh139@scarletmail.rutgers.edu

ABSTRACT

Recommendation systems play a pivotal role in enhancing user experiences by providing personalized suggestions based on user preferences and behaviors. This project presents the development of a Personalized Movie Recommendation System leveraging collaborative filtering techniques. Using the MovieLens dataset, the system predicts user ratings for unseen movies and generates top-N recommendations.

The project is structured into three key phases: (1) Data preprocessing, where user-movie interactions are split into training and testing datasets; (2) Rating prediction, employing user-based collaborative filtering and cosine similarity to estimate ratings; and (3) Recommendation evaluation, utilizing metrics such as MAE, RMSE, precision, recall, F-measure, and NDCG to assess the system's accuracy and effectiveness.

The results demonstrate the system's ability to provide tailored movie suggestions and highlight the challenges of sparse data and limited user-item overlaps. This work lays the foundation for further enhancements, including hybrid recommendation models and integration of metadata for improved performance and explainability.

KEYWORDS

Collaborative Filtering, Recommender Systems, Rating Prediction, Cosine Similarity, Evaluation Metrics, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), User-User Similarity

1 INTRODUCTION

Recommender systems are tools for personalizing user experiences in areas such as online stores, online streaming, social networks, and in our case, movies. By analyzing user interactions with items, these systems try predicting user preferences and provide recommendations specifically tailored for the user, enhancing engagement and satisfaction. Collaborative filtering is a recommendation technique. It uses the similarities between users or items based on historical interaction data. However, the effectiveness of collaborative filtering can be set back by challenges such as data sparsity, scalability, and the cold-start problem.

This research addresses the problem of predicting user ratings for movies and generating 10 recommendations using a collaborative filtering approach. The method uses a user-user similarity which is computed via cosine similarity to predict missing ratings. To try to reduce data sparsity, the approach normalizes user preferences by subtracting the user's mean rating, ensuring a stronger computation of similarities.

The proposed method is evaluated through a lot of experiments. Prediction accuracy is accounted by using metrics such as mean absolute error (MAE) and root mean squared error (RMSE), while recommendation quality is assessed using precision, recall, F-measure, and normalized discounted cumulative gain (NDCG). Results show us that the model achieves accuracy in rating prediction and provides meaningful recommendations.

2 RELATED WORK

User-based collaborative filtering, was originally introduced by Resnick et al. (1994) [3]. It focuses on identifying users with similar preferences and using their ratings to predict missing values. Item-based approaches, as highlighted in Sarwar et al. [4], analyze item similarities to make recommendations. While both methods have demonstrated effectiveness, they often face challenges such as data sparsity, cold-start problems, and scalability when dealing with large datasets.

Cosine similarity is a metric that is commonly used. It is used for measuring user-user or item-item relationships in collaborative filtering. Research by Breese et al. (1998) [1] found it to be efficient and effective in figuring out user preferences.

When trying to evaluate recommendation systems, there is usually two main metrics. This includes mean absolute error (MAE) and root mean squared error (RMSE). We can use this to figure out prediction accuracy, precision, recall, and normalized discounted cumulative gain (NDCG) for measuring the quality of recommendations. Studies like Herlocker et al. (2004) [2] have discussed these metrics and how they can be applied to different recommendation scenarios.

In this study we build on existing work by using cosine similarity for user-based collaborative filtering, particularly in movies, while incorporating normalization to address sparsity. Additionally, it evaluates recommendation quality in-depth using metrics like MAE, RMSE, precision, recall, and NDCG, providing a broad view of its performance.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

3 PROBLEM FORMALIZATION

The research problem can be mathematically defined as predicting user ratings for items in a user-item interaction matrix and generating personalized recommendations based on these predictions. The formalization is as follows:

3.1 Input

- **User-Item Rating Matrix (R)**: A matrix $R \in \mathbb{R}^{m\times n}$, where m is the number of users, n is the number of items, and $R_{u,i}$ represents the rating given by user u to item i. If a rating is missing, $R_{u,i} = 0$.
- **Training and Testing Splits**: The matrix R is divided into:
  - **Training Matrix ($R_{\mathbf{train}}$)**, containing known ratings used to train the model.
  - **Testing Matrix ($R_{\mathbf{test}}$)**, containing known ratings used for evaluation.

3.2 Output

- **Predicted Rating Matrix ($\hat{R}$)**: A matrix $\hat{R} \in \mathbb{R}^{m\times n}$, where $\hat{R}_{u,i}$ represents the predicted rating for user u on item i.
- **Top 10 Recommendations ($\mathcal{L}_u$)**: For each user u, generate a ranked list $\mathcal{L}_u$ of 10 items with the highest predicted ratings, excluding items already rated in R_{train}.

3.3 Objective

The goal is to minimize the error between predicted ratings ($\hat{R}_{u,i}$) and actual ratings ($R_{\text{test},u,i}$), and to maximize the relevance of the recomendations we give. These objectives are can be evaluated like this:

- **Prediction Accuracy**: Evaluate the difference between predicted and actual ratings using:

$$\text{MAE}$$

$$\text{RMSE}$$

  where $\mathcal{K}$ is the set of user-item pairs in the test set.
- **Recommendation Quality**: Assess the relevance and ranking of recommendations using:

$$\text{Precision@N} = \frac{\text{Relevant Items Retrieved}}{N}$$

$$\text{Recall@N} = \frac{\text{Relevant Items Retrieved}}{\text{Total Relevant Items for the User}}$$

3.4 Constraints

- **Sparsity**: The model must effectively try to infer missing values.
- **Exclusion of Known Ratings**: Recommendations must exclude items already rated by users in R_{train} to ensure it is relevant.

3.5 Optimization Goal

The model trys to:

(1) Minimize prediction error by getting low MAE and RMSE.
(2) Maximize recommendation relevance and ranking quality by optimizing Precision, Recall, and NDCG.

By creating our problem like this, we establish a clear methodology for designing the solution.

4 THE PROPOSED MODEL

This research proposes a collaborative filtering-based model to address the problem of predicting user ratings for items and generating personalized recommendations. The method uses user-user similarity to estimate missing ratings. This makes sure of relevance and personalization in the generated recommendations.

The model begins by making a user-item rating matrix, where each entry represents a user's rating for an "item", in this case a movie. To account for the differences in individual rating behaviors, user ratings are normalized by subtracting the user's mean rating. This normalization helps lower possible biases and ensures a more fair comparison when calculating similarities between users.

User-user similarities are computed using cosine similarity, a measure that computes the similarity between users based on their normalized rating patterns. The similarity values are then used to calculate a weighted sum of ratings from similar users, predicting ratings for items that a user has not rated. This approach makes sure that users are more influenced by those with similar preferences. This will improve the accuracy of the predictions.

Once the predicted ratings are generated, the system produces 10 recommendations for each user. These recommendations are made by ranking items based on their predicted ratings, excluding items the user has already rated in the data that was used to train the model. The resulting list prioritizes items that are most likely to match the user's preferences.

Since this model is so simple, it makes it efficient while maintaining a strong focus on accuracy and personalization. By addressing key challenges such as data sparsity through normalization and similarity-based weighting, our model provides a solution for building effective recommendation systems. In the next section, we will evaluate this experiment, will highlight its performance and discuss its strengths and limitations.

5 EXPERIMENTS

This section presents our experiment and the results of the model. We will analyze and focus on its ability to predict user ratings and generate recommendations that are considered relevant. The experiments were conducted using a publicly available dataset, which was preprocessed to create a user-item matrix. The rows represented a unique user, columns represented items, and matrix entries were the corresponding ratings. Missing entries in the matrix represented unobserved ratings. The dataset was split into an 80% training set and a 20% testing set to evaluate the model's performance.

To predict ratings, user-user similarities were computed using cosine similarity after normalizing user ratings to account for individual biases. Predicted ratings were then derived by aggregating the weighted contributions of similar users. The accuracy of the predictions can be determined by comparing the predicted ratings to the actual ratings in the test dataset. This is the reason we have used a 80-20 split with our data. We have used Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as metrics to evaluate our results. The model achieved an MAE of 0.714 and an RMSE of

0.921. This indicates that it effectively minimized prediction errors while maintaining computational simplicity.

To see how well the recommendations worked, we created a list of the top 10 items for each user based on the predicted ratings. We made sure to exclude any items the user had already rated in the training data to keep things consistent and fair. The recommendations were assessed using Precision, Recall, F-measure, and Normalized Discounted Cumulative Gain (NDCG). The results showed a Precision of 0.1885, meaning that approximately 18.85% of the recommended items were relevant. Recall was measured at 0.0901, meaning that 9.01% of relevant items were successfully retrieved. The F-measure, which balances Precision and Recall, was 0.1219. Furthermore, the NDCG score of 0.2240 highlighted that the model prioritized relevant items effectively in its rankings, with higher-ranked recommendations aligning well with user preferences.

The results demonstrate the strengths and limitations of the proposed model. On the positive side, the model achieves competitive accuracy in rating predictions and effectively ranks relevant recommendations, as reflected by the NDCG score. However, the relatively low Recall suggests that the model misses a significant portion of relevant items, potentially limiting the diversity of recommendations. Similarly, the moderate Precision indicates room for improvement in filtering out irrelevant recommendations.

In summary, the experimental results confirm the proposed model's effectiveness in rating prediction and recommendation ranking while identifying areas for refinement. These insights provide a foundation for future work aimed at enhancing recommendation quality and addressing the challenges of sparsity and diversity in collaborative filtering systems.

6 CONCLUSIONS AND FUTURE WORK

This research implemented a collaborative filtering-based recommendation system that predicts user ratings and generates personalized item suggestions. Using cosine similarity and normalization to make sure that there are no user biases, the model achieved promising results in minimizing prediction errors, as shown by low MAE and RMSE values. Additionally, the NDCG score highlighted the model's strength in ranking relevant items effectively. However, the relatively low Recall and moderate Precision reveal areas where the model could be improved, particularly in identifying and retrieving a higher proportion of relevant items while reducing irrelevant recommendations.

To address these limitations, future work could explore different approaches. These approaches could combine collaborative filtering with content-based techniques. Incorporating additional item features, such as genres, descriptions, or other kinds of data we can obtain from the dataset, could enhance recommendation diversity and relevance. Alternative similarity measures, such as Pearson correlation or adjusted cosine similarity, may better capture relationships between users, while dimensionality reduction techniques like Singular Value Decomposition (SVD) could give us new factors that improve both prediction accuracy and recommendation quality.

Integrating user feedback into the recommendation process could enable the system to adapt to changing preferences over time. Reinforcement learning or online learning methods could provide a more personalized experience, improving recommendations in real-time. These enhancements, along with further testing on larger and more diverse datasets, would help create more strong, accurate, and user-centric recommendation systems.

In conclusion, this study demonstrated the effectiveness of a collaborative filtering-based approach for personalized recommendations, highlighting its ability to minimize prediction errors and prioritize relevant items in rankings.

ACKNOWLEDGEMENT

I would like to express my gratitude to everyone who contributed to the success of this project. Special thanks go to our professors and recitation teachers for their discussions and insights, which helped direct us in this project. Additionally, I would like to acknowledge the support and resources provided by peers in this class. They help helped us overcome challenges and made sure we completed this project to the best of our ability. Thank you to everyone who played a role in making this work possible.

REFERENCES

[1] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2013. Collaborative Filtering for Implicit Feedback Datasets. *arXiv preprint arXiv:1301.7363* (2013). https://arxiv.org/pdf/1301.7363

[2] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. In *Proceedings of the 10th International Conference on World Wide Web (WWW '03)*. ACM, 306–314. https://doi.org/10.1145/963770.963772

[3] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. 1994. *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*. Technical Report WP-165. MIT Center for Coordination Science. http://ccs.mit.edu/papers/ccswp165.html

[4] 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)*. ACM, Hong Kong, China, 285–295. https://doi.org/10.1145/371920.372071