# Research on Personalized Movie Recommendation System Based on Collaborative Filtering

Zijie Sheng

Institute of Statistics and Applied Mathematics, Anhui University of Finance and Economics, Bengbu, Anhui, 233030, China

20221790@aufe.edu.cn

**Abstract.** It is becoming increasingly apparent that the wealth of Internet movie resources can, on occasion, be overwhelming. This has brought to the fore the importance of personalized movie recommendation systems. This paper explores such systems based on collaborative filtering. The MovieLens100k dataset is selected for analysis, and its source, scale, characteristics, user ratings, movies, and user information it contains are described in detail. We then constructed a collaborative filtering recommendation model and analyzed and compared the performance of different algorithms. The paper goes on to propose a hybrid collaborative filtering approach with the aim of achieving a comprehensive system with good scalability and maintainability. Finally, we hope that the paper will be of interest to readers and that it may even encourage further discussion and research in this area, for example in the integration of deep learning and big data processing, with a view to improving the accuracy and intelligence of movie recommendations and enhancing the user viewing experience.

**Keywords:** Collaborative Filtering; Matrix Factorization; Movie Recommendation System; Personalized Recommendation.

## 1. Introduction

Recommendation systems, which are widely used in e-commerce, journalism, social media, and the film and television industries, have emerged as a crucial technology for reducing information overload and improving user experience in the age of information explosion. Movie recommendation systems are particularly important. Online video platforms have a vast number of films, making it difficult for users to choose. Precise and personalized recommendations can save time in selecting films and meet users' viewing needs. The MovieLens100k dataset is a classic material for recommendation system research, containing 20 million user movie rating records and rich metadata, providing a solid data foundation for algorithm verification [1].

Analyzing the complex interaction between users and movies, spotting user preference trends, and correctly suggesting films that users haven't seen but are probably interested in are the main goals of the recommendation system. The method predicts unknown ratings and filters for suggestions of high-quality movies by using past data, including user ratings, browsing history, and favorites.

Collaborative filtering is a classic recommendation technology that makes recommendations based on the preferences of similar users or items. It is divided into user-user collaborative filtering (finding similar users' preferences for recommendation) and item-item collaborative filtering (recommending based on item similarity). Matrix factorization decomposes the high-dimensional sparse user-item rating matrix into low-dimensional dense matrices, extracting latent features to assist in rating prediction, reducing computational load and mining hidden information. Singular Value Decomposition (SVD) and Alternating Least Squares (ALS) are common algorithms [1].

## 2. Related Work

The commonly used algorithms are shown in Fig. 1 and Table 1.

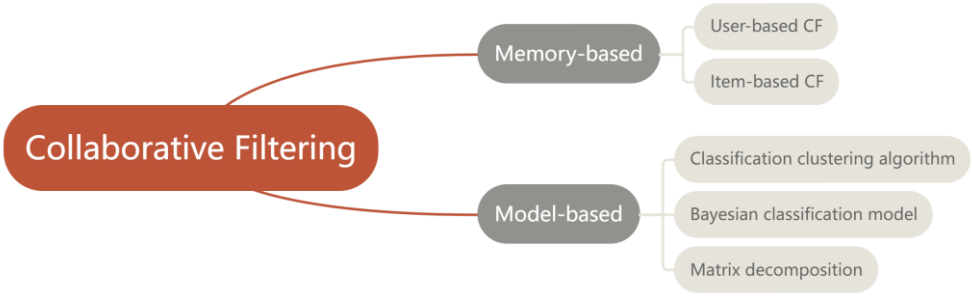


**Fig. 1** Algorithm Classification Diagram [2].

**Table 1.** Comparison of Common Techniques [2]

| | Techniques | Advantages | Disadvantages |
|---|---|---|---|
| Memory-based | User-based CF | Similarities that directly reflect user preferences | Processing a large number of users requires a lot of computing resources and difficult to make accurate recommendations for new users |
| | Item-based CF | Item attributes are relatively stable and more suitable for large-scale data sets, high computational efficiency | Subtle differences in user preferences are easily overlooked |
| Matrix decomposition | SVD | Extract key features for dimensionality reduction | Poor effect on sparse matrix and high computational complexity |
| | LSM | Unearth potential features and interpret user preferences and item characteristics | Limited when the data is sparse and relies on hyperparameter adjustment |
| | ALS | Parallel computation is efficient and suitable for large-scale sparse matrix | Sensitive to hyperparameters and requires a large amount of calculation |

## 3. Datasets Description

The MovieLens100k dataset is constructed based on users' ratings of movies, covering users' basic information, detailed movie information, and interaction data between users and movies. Movie titles may be misleading, with the majority of movies released between 1990 and 2000. The genre column shows imbalance and sparsity, with only 5% of movies having more than four genre tags. The age distribution of users is close to a Gaussian distribution, which is conducive to statistical analysis. The rating distribution is highly skewed, with most movies being rated fewer times and a few movies being rated frequently (Fig. 2).

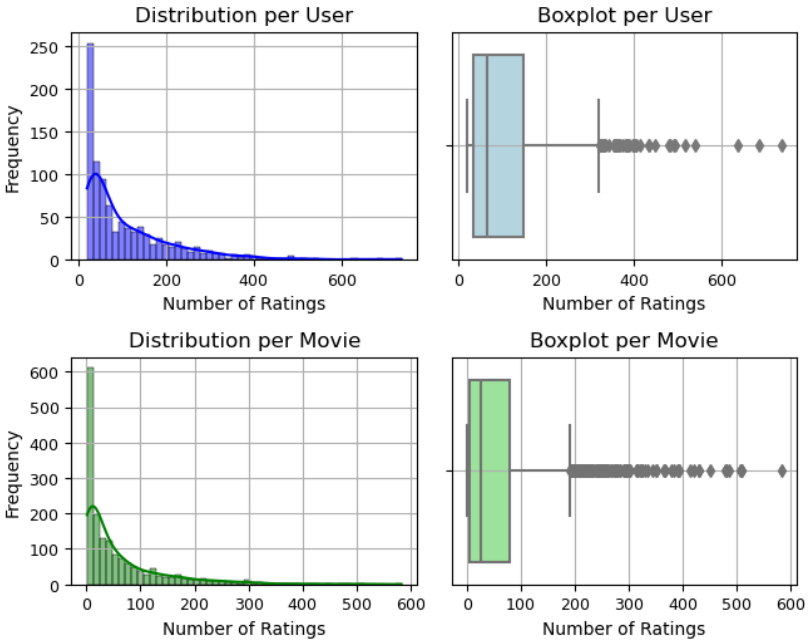


**Fig. 2** Distribution of Ratings per User and Movie (Photo/Picture credit: Original).

### 3.1. Collaborative Filtering

#### 3.1.1 User-User Collaborative Filtering

The cosine similarity metric was used to determine how similar users u and v were to one another.

$$\text{sim}(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} * r_{vi})}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} * \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}} \quad (1)$$

Where Iuv is the set of movies co-rated by users u and v and rui represents the rating of movie i by user u.

After identifying similar users, the predicted rating for an unrated movie j by user u is calculated as:

$$\widehat{r_{uj}} = \bar{r}_u + \frac{\sum_{v \in N(u)} sim(u,v) * (r_{vj} - \bar{r}_v)}{\sum_{v \in N(u)} sim(u,v)} \quad (2)$$

Where N(u) denotes the set of neighboring users, and $\bar{r}_u$ and $\bar{r}_v$ are the average ratings of users u and v respectively [3].

#### 3.1.2 Item-Item Collaborative Filtering

Items i and j are determined to have cosine similarity by

$$sim(i, j) = \frac{\sum_{u \in U_{ij}} (r_{ui} * r_{uj})}{\sqrt{\sum_{u \in U_{ij}} r_{ui}^2} * \sqrt{\sum_{u \in U_{ij}} r_{uj}^2}} \quad (3)$$

In this study, the set of users who have rated both items i and j are denoted by $U_{ij}$.

The predicted rating for an unrated movie j by user u is given by:

$$\widehat{r_{uj}} = \frac{\sum_{i \in N(j)} sim(i,j) * r_{ui}}{\sum_{i \in N(j)} sim(i,j)} \quad (4)$$

Where the term N(j) is used to refer to the set of similar items to movie j.

The algorithm in question constructs a user-item rating matrix, the purpose of which is to store the ratings. It is possible to calculate similarity in real-time or through offline updates [3].

# 4. Matrix Factorization

## 4.1. Overview of Matrix Factorization

The goal is to decompose the user-item rating matrix Rm×n into low-dimensional user matrix Pm×k and item matrix Qn×k (where k is the dimensionality of the latent factors), such that $R \approx PQ^T$ [4].

The user's estimated rating for a film $i$ is: $\widehat{r_{ui}} = P_u^T q_i$ where $p_u$ and $q_i$ are the corresponding vectors for user u and movie $i$, respectively. This approach helps in uncovering latent features that reveal preferences and attributes [4, 5].

## 4.2. Singular Value Decomposition (SVD)

The matrix R is decomposed as $R = U \sum V^T$, where U and V are orthogonal matrices, and Σ is a diagonal matrix. The first k singular values and corresponding vectors are selected for approximate decomposition: $R_k = U_k \sum_k V_k^T$ .The rating matrix is then reconstructed to predict ratings. Considering how sparse real-world data is, preprocessing and algorithm adjustments are required, such as using FunkSVD to improve upon traditional SVD [5].

## 4.3. Alternating Least Squares (ALS)

Matrices P and Q are randomly initialized, and while P is fixed, the algorithm minimizes the loss function: $\sum_{(u,i)\in\Omega}(r_{ui} - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2)$, where Ω is the set of known values, and $\lambda$ is the regularization term.

The least squares method is then used to update Q, and the process alternates between fixing Q and updating P until convergence. Distributed frameworks such as Spark can be used for parallel computation to accelerate large matrix factorization [6].

## 4.4. Model Fusion

The loss function that is selected is the Mean Squared Error (MSE):

$$MSE = \frac{1}{|\Omega|}\sum_{(u,i)\in\Omega}(r_{ui} - \widehat{r_{ui}})^2 \tag{5}$$

The elements of P and Q are fitted using optimization methods like gradient descent and stochastic gradient descent, and overfitting is avoided by regularization. Through cross-validation, the hyperparameters are adjusted [7].

### 4.4.1 Weighted Fusion

For collaborative filtering and matrix factorization prediction ratings, weights $\alpha$ and $1-\alpha$ are assigned, respectively, to fuse the ratings:

$$\hat{r}_{ui}^{fused} = \alpha\hat{r}_{ui}^{CF} + (1-\alpha)\hat{r}_{ui}^{MF} \tag{6}$$

The optimal $\alpha$ is determined experimentally, balancing the strengths of both models [8].

### 4.4.2 Cascade method

Collaborative filtering is used to pre-filter candidate movies, and matrix factorization is used for a second evaluation to refine the recommendations. This fusion of methods at different stages helps to reduce the computational load and improve the accuracy of the recommendations, making the approach suitable for large datasets [9].

# 5. Experimental Design and Analysis

The dataset is randomly divided into training and test sets in an 80:20 ratio, with multiple experiments conducted to ensure randomness.

### 5.1. Accuracy Metrics

Root Mean Square Error (RMSE): $RMSE = \sqrt{\frac{1}{|\Omega_{test}|}}\sum_{(u,i)\in\Omega_{test}}(r_{ui} - \hat{r}_{ui})^2$,the RMSE is used to assess the overall prediction error, with a lower value indicating a higher level of accuracy [10].

Mean Absolute Error (MAE): $\frac{1}{|\Omega_{test}|}\sum_{(u,i)\in\Omega_{test}}|r_{ui} - \hat{r}_{ui}|$, The MAE provides an intuitive measure of the average absolute predicted error [10, 11].

### 5.2. Ranking Metrics

Precision: $Precission = \frac{|\text{Recommended and liked by users}|}{|\text{Recommended by users}|}$. The precision metric quantifies the percentage of suggested products that align with the user's choices.

Recall: $Recall = \frac{|\text{Recommended and liked by users}|}{|\text{liked by users}|}$, it assesses the extent to which the recommended items cover the user's interests.

F1-Score: $F_1 = \frac{2*Precission*Recall}{Precission+Recall}$. The F1-score balances precision and recall to provide a harmonic mean of these two metrics.

Normalized Discounted Cumulative Gain (NDCG): It is calculated based on the ranking of the recommendation list and the user's actual ratings, with higher scores for items recommended higher up the list. $NDCG@N = \frac{DCG@N}{IDCG@N}$, where IDCG is for Ideal Discounted Cumulative Gain, the highest DCG that may be achieved, and DCG stands for Discounted Cumulative Gain [12].

## 6. Results and Discussion

### 6.1. Presentation of Results

**Table 1.** Performance comparison results

| | MSE | RMSE | MAE | Precision | Recall | F1 | NDCG |
|---|---|---|---|---|---|---|---|
| User-User CF | 0.91 | 0.95 | 0.73 | 0.25 | 0.2 | 0.222 | 0.45 |
| Item-Item CF | 0.89 | 0.94 | 0.71 | 0.28 | 0.25 | 0.263 | 0.5 |
| SVD | 0.65 | 0.806 | 0.54 | 0.3 | 0.28 | 0.288 | 0.55 |
| ALS | 0.6 | 0.775 | 0.49 | 0.32 | 0.3 | 0.307 | 0.6 |
| Model Fusion | 0.55 | 0.742 | 0.45 | 0.35 | 0.32 | 0.333 | 0.65 |

### 6.2. Analysis and Discussion

The performance of item-item collaborative filtering (CF) typically outperforms user-user CF in terms of both accuracy and efficiency, especially when applied to large datasets (Table 2). Matrix factorization methods show lower RMSE and MAE values compared to collaborative filtering, indicating better accuracy by revealing latent features. However, these methods are computationally intensive and require significant training time. Hybrid models generally perform better across multiple metrics than single-model approaches, demonstrating the rationale for combining different techniques [13].

Challenges remain, such as the sparsity of data in collaborative filtering, which leads to cold start problems for users, and the sensitivity of matrix factorization to hyperparameters, which can affect generalization. These issues require further research for effective solutions [14].

## 7. Conclusion

In this study, the MovieLens100k dataset was used to successfully design a movie recommendation system based on collaborative filtering and matrix factorization. The underlying algorithmic principles, optimization strategies and model fusion techniques are discussed in detail.

Experimental validation confirms the effectiveness of the proposed methods. Collaborative filtering, being simple and intuitive, effectively captures user similarities, while matrix factorization delves deeper into latent features. The fusion of these approaches significantly improves the overall performance of the recommendation system.

Convolutional Neural Networks (CNNs) will be used in future studies to investigate the expansion of deep learning methods for the analysis of textual descriptions and movie images. The integration of Graph Neural Networks (GNNs) will be considered to model user social interactions and film relationships. In addition, context-aware recommendations will be explored, taking into account factors such as viewing time, device and user emotion.

Optimization of cold-start problems will be a priority, using multimodal data to quickly build user profiles for new users. To increase recommendation systems' intelligence and personalization, the creation of dynamic suggestion systems that can monitor and adjust to changing user preferences in real time will also be investigated.

## References

[1] Qin, Z., Zhang, M. Towards a Personalized Movie Recommendation System: A Deep Learning Approach. In: 2021 2nd International Conference on Artificial Intelligence and Information Systems (ICAIIS 2021). Association for Computing Machinery, New York, NY, USA, Article 216, 1–5, 2021.

[2] H K, S., Iyer, K.P., Himaja, K.R., Pokharel, R. Personalized Movie Recommendation System. International Journal of Information Technology, Research and Applications, 2023.

[3] Saifudin, I., Widiyaningtyas, T. Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets. IEEE Access, 2024, 12, 19827-19847.

[4] Raj, K., Das, A.A., Guha, A., Sharma, P., S, M.K. Movie Recommendation System. International Journal of Computer Sciences and Engineering, 2019.

[5] Zhang, J., Wang, Y., Yuan, Z., Jin, Q. Personalized real-time movie recommendation system: Practical prototype and evaluation. Tsinghua Science and Technology, 2020, 25(2), 180-191.

[6] Li, L., Huang, H., Li, Q., Man, J. Personalized movie recommendations based on deep representation learning. PeerJ. Computer Science, 2023, 9, e1448.

[7] Zhao, B. A Personalized Movie Recommendation System Based on Knowledge Graph. In: 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), Chongqing, China, 2023, 588-592.

[8] Salmani, S., Kulkarni, S. Hybrid Movie Recommendation System Using Machine Learning. In: 2021 International Conference on Communication Information and Computing Technology (ICCICT), Mumbai, India, 2021, 1-10.

[9] Paranjape, V., Sharma, S. Enhancing User Experience: Advanced Techniques in Movie Recommender Systems. International Journal of Innovative Research in Computer and Communication Engineering, 2023, 12(04), 4466-4470.

[10] Aramuthakannan, S., Devi, M.R., Lokesh, S., Manimegalai, R. Movie recommendation system via fuzzy decision making based dual deep neural networks. J. Intell. Fuzzy Syst., 2023, 44(3), 5481-5494.

[11] Park, J., You, I., Shin, S., Jeong, U. Material Approaches to Stretchable Strain Sensors. ChemPhysChem, 2015, 16, 1155-1163.

[12] Mandal, D., Aasim, M., Drig, P., Biswas, R., Ramteke, R., Kaur, M. Beyond the Hype: Building a Personalized Movie Experience with Content-Based Recommendation. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 2024.

[13] Siet, S., Peng, S., Ilkhomjon, S., Kang, M., Park, D.-S. Enhancing Sequence Movie Recommendation System Using Deep Learning and KMeans. Applied Sciences, 2024, 14(6), 2505.

[14] Feng, X., Hu, J., Zhu, X. Machine Learning Based Personalized Movie Research and Implementation of Recommendation System. 2022, 74-78.