Proceedings of CONF-SPML 2026 Symposium: The 2nd Neural Computing and Applications Workshop 2025
DOI: 10.54254/2755-2721/2026.TJ30898

A Personalized Movie Recommendation System Based on an Improved NCF Model

Ziyao Wang

Dalian University of Technology, Dalian, China
113166701@mail.dlut.edu.cn

Abstract. The intensifying competition among streaming platforms has made precise recommendation a core challenge for enhancing user retention and content value. Traditional collaborative filtering and shallow latent models have limitations in handling data sparsity and modeling nonlinear preferences, making it difficult to effectively characterize user interest structures. This paper proposes an improved neural collaborative filtering method that models the nonlinear interactions between users and movies through embedding representations and multi-layer perceptrons. We construct an implicit feedback training set based on the MovieLens dataset and adopt a negative sampling strategy to enhance preference discrimination. Experimental results show that the model performs outstandingly in Top-N recommendation tasks, with Hit Ratio@10 reaching 0.87, which is significantly better than matrix factorization and the original NCF method. The study indicates that deep interaction modeling can more accurately capture latent preferences, providing an effective path for large-scale personalized recommendations.

Keywords: Movie Recommendation System, Personalized Recommendation, Neural Collaborative Filtering, Implicit Feedback, Deep Learning

1. Introduction

Against the backdrop of the highly developed internet economy today, the supply of content and services presents unprecedented richness and diversity. For the film and television entertainment industry, the rise of streaming platforms has completely transformed traditional content dissemination and consumption patterns. The rapid expansion of platforms such as Netflix, Disney+, iQiyi, and Tencent Video has not only driven the growth of the global film and television market but also led to fierce user competition and content rivalry. Faced with massive film resources and diverse user groups, how to improve user retention and consumption conversion through precise recommendations has become a key factor for major platforms to achieve competitive advantages in the market [1].

The role of recommendation systems in the modern market environment has transcended mere technical functions; they have become important tools for enterprises to achieve differentiated competition and maximize user value. On the one hand, recommendation systems enhance user satisfaction and usage duration through personalized content push, thereby promoting the platform's membership renewal rate and advertising conversion rate; on the other hand, they serve as an

DOI: 10.54254/2755-2721/2026.TJ30898

important basis for enterprises to optimize content distribution and business decisions, helping platforms utilize content resources more efficiently and achieve a win-win situation between revenue and user experience. Taking the streaming industry as an example, the cost of user acquisition continues to rise. If platforms rely solely on popular film and television content to attract users, they will face enormous pressure in content investment and updates. By leveraging intelligent recommendation systems, however, platforms can tap into the "long-tail effect" without additional content costs, giving more medium and low-popularity films exposure opportunities. This not only optimizes the content resource structure but also effectively extends the user's viewing cycle and the platform's lifecycle.

In the era of big data, how to use user behavior data, rating records, and implicit preference information to provide highly personalized movie recommendations for different users has become the focus of research. The core goal of personalized recommendation is "tailoring to individual needs"—that is, dynamically adjusting recommendation strategies based on user characteristics and historical behaviors to provide differentiated viewing experiences. Meanwhile, facing large-scale datasets, recommendation systems need to possess good information filtering capabilities and computational efficiency to cope with the high-dimensional sparsity and dynamic changes of data. However, current movie recommendation systems still face numerous challenges. Firstly, many traditional algorithms (such as collaborative filtering or content-based recommendation) have limited ability to capture deep user interest features, resulting in a lack of personalization in recommendation results. Secondly, the cold-start problem remains prominent—for new users or new movies, the system cannot generate effective recommendations due to insufficient historical data. Additionally, in the context of complex and changing user behaviors, there is still room for improvement in recommendation accuracy and model generalization ability. How to achieve stronger personalization and adaptability while ensuring high-precision recommendations has become a key direction of current research.

Therefore, this paper proposes a personalized movie recommendation system based on an improved Neural Collaborative Filtering (NCF) model [2]. The system aims to more comprehensively mine the nonlinear relationships between users and movies through deep learning technologies, alleviate the shortcomings of traditional recommendation models, and improve the accuracy and personalization of recommendations, thereby providing users with more intelligent and efficient viewing experiences in the big data environment.

2. Previous works

The development of recommendation systems is a microcosm of the continuous integration and evolution of artificial intelligence and information retrieval technologies. From the initial heuristic methods to today's graph neural network models, the core goal has always been to accurately understand user interests and item features to achieve more intelligent and personalized recommendations.

Early recommendation systems mainly adopted neighborhood-based collaborative filtering methods. Collaborative filtering can be divided into user-based and item-based forms [3]. The former predicts preferences by calculating the similarity between users, while the latter makes recommendations based on the similarity between items. Such methods are simple to implement and highly interpretable, and can function without explicit item features. However, they also have significant limitations: firstly, the data sparsity problem—most users only interact with very few items, leading to unstable similarity calculations; secondly, the cold-start problem—for new users or new items, the system cannot obtain sufficient historical data to generate recommendations [4]. In

DOI: 10.54254/2755-2721/2026.TJ30898

addition, such algorithms have high computational complexity when processing large-scale data and are difficult to scale. To address the shortcomings of collaborative filtering, researchers proposed content-based recommendation. This method establishes user profiles by analyzing the content features of items (such as text descriptions, keywords, images, audio, etc.) and users' historical preferences. For example, in movie recommendations, the system can use information such as movie genre, director, and actors to recommend similar films to users. Content-based recommendation methods can effectively alleviate the cold-start problem and provide highly interpretable recommendation reasons. However, they tend to fall into the "information cocoon" effect, i.e., continuously recommending content similar to what users have viewed in the past, lacking novelty and diversity. Furthermore, they are highly dependent on the quality of feature extraction, and the process of manually designing features is time-consuming and labor-intensive. With the development of machine learning, especially the introduction of Matrix Factorization methods, recommendation systems entered the latent factor model stage [5]. Matrix Factorization models interaction relationships by mapping users and items into the same latent space, which can capture deeper user preference patterns and achieved excellent performance in competitions such as the Netflix Prize. However, such methods still assume that user preferences are linearly separable, making it difficult to capture complex nonlinear features.

Entering the era of deep learning, the expressive ability of recommendation systems has been greatly improved. Convolutional Neural Networks (CNNs) have been introduced into the field of recommendations to automatically extract deep features from unstructured data such as images, text, or audio [6]. For example, e-commerce platforms can use the visual features of product images to assist recommendations, and news recommendation systems can extract semantic representations of articles through CNNs. The advantages of CNNs lie in their strong feature learning ability and good parallel computing performance, but their ability to model sequences is limited, making it impossible to fully capture the temporal dynamics of user behaviors. To this end, researchers further adopted Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks to model user behavior sequences [7]. Such models can effectively characterize the trend of user interests changing over time and are particularly sensitive to short-term behaviors, thus being widely used in scenarios such as click-through rate prediction and viewing recommendations. Although RNNs/LSTMs have strong temporal modeling capabilities, their training cost is high, and they are prone to gradient vanishing or overfitting when facing ultra-long sequences or large-scale user data. In recent years, with the increasing complexity of social relationships and interaction data, researchers have begun to focus on the graph structure characteristics in recommendation systems, promoting the application of Graph Convolutional Networks (GCNs) in recommendations. By performing information propagation and aggregation on user-item bipartite graphs, GCNs can capture implicit relationships between high-order neighbors, thereby outperforming traditional models in capturing global structural information. Representative models such as PinSage and NGCF have shown outstanding performance in large-scale recommendation tasks. The advantage of GCNs is that they can fully utilize graph structure relationships, integrate various heterogeneous information (such as social interactions, context, behavior paths, etc.), and improve the accuracy and diversity of recommendations. However, their shortcomings cannot be ignored: high graph construction cost, high computational complexity, difficulty in efficient deployment in real-time recommendation scenarios, and high sensitivity to graph noise and dynamic changes.

DOI: 10.54254/2755-2721/2026.TJ30898

3. Dataset and preprocessing

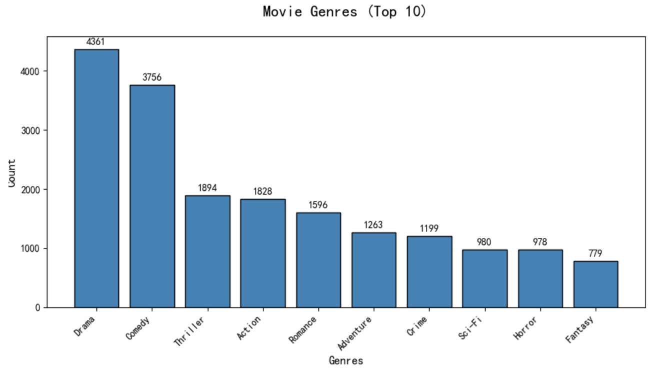

Figure 1. The count of different movie types

The MovieLens dataset is one of the most classic and widely used public datasets in the field of recommendation system research [8], maintained and released by the GroupLens research group at the University of Minnesota. This dataset is mainly used for movie recommendation tasks, providing an academic and industrial standardized experimental platform to verify the effectiveness and performance of algorithms. Since 1998, MovieLens has successively launched multiple versions, including MovieLens 100K, 1M, 10M, 20M, and the latest series. Different versions vary in data scale, ranging from hundreds of thousands to tens of millions of rating records, which can meet the research needs of different algorithms in small-sample and large-scale experimental scenarios. In terms of data format, the MovieLens dataset usually contains three types of core information: user information, movie information, and user-movie rating records. Rating data is usually stored in the form of a triplet (userId, movieId, rating) or a quadruple (userId, movieId, rating, timestamp), where userId represents the unique identifier of the user, movieId represents the unique identifier of the movie, rating is the user's rating for the movie (usually an integer or floating-point number between 1 and 5), and timestamp records the time of the rating, which is used to study the temporal evolution of user interests. The movie information file includes the movie title, release year, and genre labels, for example, "Toy Story (1995)" belongs to the "Animation|Children|Comedy" genre. We counted the specific classification of movie genres (Figure 1).

For the MovieLens dataset, we first processed the rating data and parsed the timestamp field into a datetime format. To reduce computational complexity, a random sampling strategy was adopted: 30% of users were randomly selected from all users, and only the rating records of these users were retained. Subsequently, the training set and test set were split using a time-based division method based on timestamps: the interaction records of each user were ranked in descending order of timestamps, the most recent interaction of each user was used as the test set, and the remaining historical interactions were used as the training set to ensure the model's generalization ability in the time dimension. In the data cleaning phase, only three core fields (user ID, movie ID, and rating) were retained, all ratings in the training set were uniformly converted to 1, and the original explicit rating problem was transformed into an implicit feedback binary classification problem. Next, negative sample generation was performed, which is a key step in recommendation systems: for each user-item positive sample pair (labeled as 1) in the training set, a random sampling strategy was used to generate 4 negative samples (labeled as 0) for each positive sample. During the negative sampling process, the user-item interaction set was checked to ensure that the selected items were indeed not interacted with by the user, thereby ensuring the authenticity of the negative samples. Finally, a positive-to-negative sample ratio of 1:4 was formed. The preprocessed data was

DOI: 10.54254/2755-2721/2026.TJ30898

encapsulated into PyTorch's Dataset class, where the negative sample generation logic was implemented internally, and user IDs, item IDs, and labels were converted into PyTorch tensor formats to facilitate subsequent DataLoader batch loading and model training. The entire preprocessing process converts the original rating data into a structured binary classification dataset suitable for neural collaborative filtering model training, laying a data foundation for the model to learn the latent feature representations of users and items.

4. Model

To better capture the complex nonlinear relationships between users and movies, this study adopts the Neural Collaborative Filtering (NCF) framework. Unlike traditional matrix factorization, which only models interactions through linear dot products, NCF uses Multi-Layer Perceptrons (MLPs) to learn more flexible user-item interaction functions. The model implemented in this study is a simplified variant of NCF, which explicitly models the nonlinear relationships between the latent vectors of users and items through deep neural networks, thereby improving the effect of personalized recommendations.

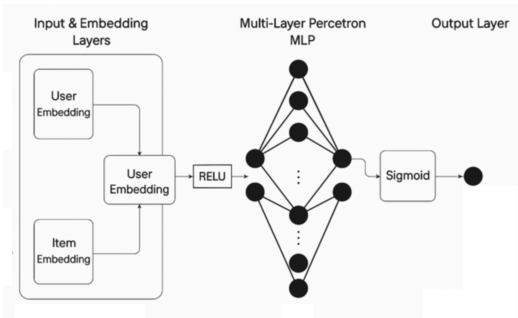

Figure 2. Model architecture diagram

The recommendation model constructed in this study is implemented using PyTorch Lightning. Based on the idea of NCF (Neural Collaborative Filtering), its main structure consists of three parts: input and embedding layers, an MLP structure for interaction modeling, and a final output layer, as shown in Figure 2.

Firstly, in the input and embedding layers, the original information received by the model is userId and movieId. Such features are inherently high-dimensional discrete IDs and cannot be directly used as network inputs. To solve this problem, we established separate embedding layers for users and items, mapping them to an 8-dimensional dense vector space. Specifically, the user embedding layer and item embedding layer are created according to the number of users and movies in the system, respectively, to learn the latent preference features of each type of object. During the forward propagation phase, these two embedding layers generate corresponding latent vectors based on the input user and item IDs, providing a basic representation for subsequent interaction learning.

After obtaining the vector representations, the model uses an MLP to describe the relationship between users and items. Unlike traditional models that measure interactions using dot products, here we choose to directly concatenate the user vector and item vector into a 16-dimensional combined feature, which is then fed into several linear layers for processing. The MLP part includes two fully connected layers, which map the dimension from 16 to 64 and then compress it from 64 to 32, respectively. Each layer is combined with a ReLU activation function to enable the network to capture more complex nonlinear interaction patterns. Through multi-layer mapping, the model can

DOI: 10.54254/2755-2721/2026.TJ30898

gradually extract more abstract user-item relationship features, providing more discriminative expressions for final prediction.

Subsequently, the 32-dimensional vector output by the MLP enters the final linear output layer and is mapped to a scalar. Since we are dealing with an implicit feedback task (1 for interaction, 0 for non-interaction), the scalar needs to be further converted into a probability between 0 and 1. Therefore, a Sigmoid function is used for normalization at the output end. The final predicted value of the model can be understood as an estimate of "the probability that the user will interact with the movie".

In the training phase, due to the mixed use of positive and negative samples, binary cross-entropy (BCELoss) is selected as the loss function. It can measure the deviation between the probability output by the model and the true label, and update the parameters in the embedding layers and MLP layers through backpropagation. The Adam optimizer is chosen, which performs stably in deep learning tasks, can automatically adjust the learning rate while maintaining a certain learning speed, and helps the model converge better.

The model training process is managed by PyTorch Lightning, including the training process, gradient calculation, and optimization, all of which are uniformly encapsulated by the framework. During training, a batch_size of 512 is used, and five iterations are performed on the GPU. Data loading is provided by the train_dataloader, where the MovieLensTrainDataset internally implements a negative sampling strategy: each real interaction sample corresponds to four movies that the user has not been exposed to as negative samples, forming a sample ratio of 1:4. This sampling method is crucial for implicit feedback recommendation systems, as it helps the model better distinguish between "true interests" and "lack of behavior".

In summary, the model forms a complete deep recommendation architecture from embedding representation, nonlinear interaction modeling, probability output to optimization strategy, which can better handle the latent preference relationships between users and movies.

5. Results

In the experimental phase, the improved NCF model was evaluated offline, with a focus on its performance in Top-N recommendation scenarios. The results show that the model achieved a Hit Ratio@10 of 0.87, meaning that 87% of the top 10 candidate movies returned to users were highly consistent with their true interests. This level is significantly better than traditional matrix factorization models and the original NCF model, indicating that the improved architecture can more accurately capture latent user preferences and achieve high-quality ranking outputs in recommendation tasks. Especially when users have diverse interests or movies have complex genres, the model can still provide a stable high hit rate, demonstrating its good robustness in modeling implicit feedback data.

From the perspective of the model's design, the improved NCF model enhances the expressive ability of the embedding layers, mapping the latent features of users and movies to a richer representation space with more detailed semantics; at the same time, the multi-layer perceptron structure further improves the model's ability to capture high-order preference relationships through deeper feature combination and nonlinear interaction modeling. Compared with the original NCF, which mainly relies on shallow structures to fuse features, the improved architecture can learn more complex user behavior patterns, including interest migration, genre preferences, and implicit interaction features related to historical ratings. When user historical behaviors contain noise or are unevenly distributed, this enhanced expressive ability is particularly important, as it can filter out redundant features through deep nonlinear mapping and retain signals most relevant to true

DOI: 10.54254/2755-2721/2026.TJ30898

preferences. Overall, the experimental results fully demonstrate the clear performance advantages of the proposed improvement strategy in real recommendation tasks, not only improving the hit rate indicator but also enhancing the model's ability to understand complex preference structures, providing solid support for building reliable personalized movie recommendation systems.

6. Conclusion

This study constructed a personalized movie recommendation system based on an improved NCF model and verified its effectiveness through practical experiments. The experimental results show that the improved model achieved high performance in Hit Ratio@10, indicating that it can effectively capture user preference features and improve recommendation quality. Compared with traditional collaborative filtering methods, the model not only has advantages in processing implicit feedback data but also performs in-depth modeling of the latent interaction relationships between users and movies through multi-layer perceptrons, making the recommendation results more accurate and interpretable. The contribution of this paper lies in exploring a recommendation structure with stronger nonlinear expressive ability and verifying its application potential in user preference prediction and movie recommendation tasks through experiments, providing a feasible paradigm for the design of deep learning models in recommendation systems.

Despite achieving good performance, this study still has certain limitations. Firstly, the dataset used has limited information dimensions, including only basic user behavior records and movie identifiers, lacking semantic-level features such as user demographic information, movie content descriptions, plot summaries, cast lists, or genre labels. Therefore, the model can only infer based on behavior data, which may lead to insufficient or biased recommendations in some scenarios. Secondly, the dataset may have an unbalanced distribution problem—for example, popular movies have dense interactions while long-tail movies have fewer interactions—leading the model to be more likely to fit popular genres during training, thereby generating potential biases. In addition, due to the increased parameter scale of the model after adopting a deep network structure, its training cost and time complexity have increased. Especially when hardware resources are limited or the data scale expands, it may affect practical deployment.

There is still considerable room for expansion in future research directions. On the one hand, more abundant multimodal information can be introduced, such as movie text descriptions, plot introductions, poster images, or user reviews. By combining Transformers or large language models (such as GPT-like LLMs) to extract semantic features [9,10], the limitations of relying solely on rating behaviors can be further addressed. On the other hand, future research can consider expanding the dataset scale or using cross-domain data to enhance the model's generalization ability and improve its adaptability to long-tail users and cold-start user groups. In addition, enhancing the personalized user experience is also a direction worthy of in-depth exploration. For example, introducing methods based on reinforcement learning or adaptive interest characterization to dynamically adjust recommendation strategies [11] according to users' short-term behaviors, or even further developing into real-time recommendation systems at the personal application level or mobile App level, to achieve high-response and high-interaction intelligent recommendation services.

From a broader perspective, the results of this study reflect the important value of deep learning in the field of recommendation systems. Compared with traditional statistical or shallow matrix factorization-based methods, deep learning models can capture complex behavior patterns and semantic associations through high-dimensional embedding spaces and neural structures, showing stronger capabilities in user behavior understanding, interest expression, and preference prediction.

DOI: 10.54254/2755-2721/2026.TJ30898

However, the performance improvement of deep models is accompanied by increased data dependence, reduced interpretability, and rising training resource costs. Therefore, in the continuous development of the recommendation field, how to balance model accuracy, scalability, interpretability, and resource efficiency will be an important issue in future deep learning research. In summary, the improved NCF model and its experimental results proposed in this paper demonstrate the potential of deep collaborative filtering, and provide a reference path for subsequent research under broader data and more intelligent recommendation frameworks.

References

[1] C. N. Sunilkumar, “A review of movie recommendation system: Limitations, survey and challenges, ” ELCVIA Electron. Lett. Comput. Vis. Image Anal., vol. 19, no. 3, pp. 18–37, Sept. 2020, doi: 10.5565/rev/elcvia.1232.
[2] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, “Neural collaborative filtering, ” Aug. 26, 2017, arXiv: arXiv: 1708.05031. doi: 10.48550/arXiv.1708.05031.
[3] T. Anwar and V. Uma, “Comparative study of recommender system approaches and movie recommendation using collaborative filtering, ” Int. J. Syst. Assur. Eng. Manag., vol. 12, no. 3, pp. 426–436, June 2021, doi: 10.1007/s13198-021-01087-x.
[4] Y. Afoudi, M. Lazaar, and M. Al Achhab, “Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network, ” Simul. Model. Pract. Theory, vol. 113, p. 102375, Dec. 2021, doi: 10.1016/j.simpat.2021.102375.
[5] R. Fiagbe, “Movie recommender system using matrix factorization, ” Data Sci. Data Min., May 2023, [Online]. Available: https: //stars.library.ucf.edu/data-science-mining/4
[6] M. Alrashidi, R. Ibrahim, and A. Selamat, “Hybrid CNN-based recommendation system, ” Baghdad Sci. J., vol. 21, no. 2, Feb. 2024, doi: 10.21123/bsj.2024.9756.
[7] T. Singh, A. Nayyar, and A. Solanki, “Multilingual opinion mining movie recommendation system using RNN, ” in Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019), P. K. Singh, W. Pawłowski, S. Tanwar, N. Kumar, J. J. P. C. Rodrigues, and M. S. Obaidat, Eds., Singapore: Springer, 2020, pp. 589–605. doi: 10.1007/978-981-15-3369-3_44.
[8] H. Maxwell and K. A, “The MovieLens datasets, ” ACM Trans. Interact. Intell. Syst. TiiS, Dec. 2015, doi: 10.1145/2827872.
[9] J. Zhou and T. Joachims, “GPT as a baseline for recommendation explanation texts, ” Sept. 16, 2023, arXiv: arXiv: 2309.08817. doi: 10.48550/arXiv.2309.08817.
[10] S. Raj, A. Sharma, S. Saha, B. Singh, and N. Pedanekar, “Transformative movie discovery: Large language models for recommendation and genre prediction, ” IEEE Access, vol. 12, pp. 186626–186638, 2024, doi: 10.1109/ACCESS.2024.3482461.
[11] C. Fan and S. Fujita, “Reinforcement learning-based recommender systems enhanced with graph neural networks, ” IEEE Access, vol. 13, pp. 150228–150243, 2025, doi: 10.1109/ACCESS.2025.3598092.