

Movie Recommendation System Based on Collaborative Filtering Network

Ke Tang

North China University of Technology, No. 5 Jinyuanzhuang Road, Shijingshan District, Beijing, China

13701190094@163.com

Abstract. In the era of big data, personalized recommendation systems have become crucial for enhancing user experience and improving information retrieval efficiency, especially in the movie recommendation domain. This study proposes a method based on Collaborative Filtering Network (CFN), utilizing deep learning techniques to construct an efficient and accurate movie recommendation system. We evaluated the model's effectiveness using the MovieLens dataset. The experimental results demonstrate that the proposed CFN model performs well across multiple metrics, including Hit Ratio, providing new insights for research in the recommendation system field.

Keywords: Collaborative Filtering, Movie Recommendation System, Neural Networks, Deep Learning, MovieLens.

1. Introduction

In the era of big data, recommendation systems have become an essential bridge connecting users with products or content, serving as a vital component of internet platforms [1]. As data volumes continue to grow, user choices have become increasingly diverse. Recommendation systems analyze user behavior data to provide personalized suggestions, thereby enhancing user experience and increasing platform retention rates. In the movie industry, users often face the daunting task of choosing from thousands of films, which can lead to decision-making difficulties. Traditional viewing methods typically rely on word-of-mouth or simple searches, which are inadequate in the context of information overload. Recommendation systems can facilitate quick access to films that align with users' interests through precise and personalized suggestions. This tailored experience not only boosts user satisfaction but also enhances platform engagement and conversion rates.

However, traditional recommendation methods, such as user-based and item-based collaborative filtering, face several challenges in a big data environment. Firstly, these methods require significant computational resources, especially as the number of users and items increases, leading to higher computational costs. Secondly, they often struggle with sparse data, making it difficult to provide accurate recommendations when user behavior data is limited. Moreover, traditional approaches frequently fail to capture the complexities of user preferences and item characteristics, resulting in less precise recommendations that fall short of modern users' high expectations for personalized suggestions.

Thus, constructing an efficient Movie Recommendation System (MRS) is crucial in the context of large-scale movie data [2]. As a digital product, the demand for personalized recommendations in films

is continually rising. By analyzing users' viewing histories, rating behaviors, and other data, recommendation systems can effectively identify user preferences and recommend relevant films. The core requirement of modern movie recommendation systems is not only to provide films that match users' tastes but also to filter out irrelevant information, ensuring efficient recommendations within vast datasets. Companies like Amazon and Netflix exemplify how recommendation systems can significantly enhance user experience and movie consumption. Netflix's recommendation algorithm analyzes users' historical viewing data and employs personalized suggestions to increase platform engagement. For instance, Netflix creates complex user models based on watched films, ratings, and viewing durations, enabling refined recommendations. Similarly, China's iQIYI utilizes its recommendation algorithm to help users discover new content, thereby improving user experience and satisfaction. As user demands evolve, the role of recommendation systems extends beyond providing accurate suggestions; it enriches users' viewing experiences by helping them uncover potential interests. Furthermore, the application of recommendation systems is expanding across various domains, from movies to music, books, and e-commerce platforms, fulfilling diverse user needs.

In response to the limitations of traditional methods, this paper proposes a MRS based on Collaborative Filtering Network (CFN). This system leverages deep learning techniques to better capture the intricate relationships between users and items, enhancing the accuracy and personalization of recommendations. The CFN model introduces an innovative approach by integrating user-item embedding with advanced regularization techniques, including L2 regularization and Batch Normalization. These methods enable the model to handle large-scale datasets effectively while improving the robustness and personalization of recommendations. This approach not only surpasses traditional collaborative filtering but also presents a scalable solution adaptable to various domains, providing a new perspective in recommendation system research. This innovative approach not only improves user experience but also offers new insights for the future development of recommendation systems.

2. Previous works

In the development of recommendation systems, traditional methods have maintained a significant role. These methods primarily include user-based collaborative filtering, item-based collaborative filtering, hybrid recommendation systems, and content-based recommendation.

User-based collaborative filtering algorithms recommend items to users by analyzing the similarities between users [3]. This approach uncovers users' latent interests by leveraging the preferences of similar users without relying on specific item attributes. Advantages of this method include its ability to provide personalized recommendations based solely on user interactions. However, it faces significant disadvantages: as the number of users increases, the computational cost for calculating similarities rises dramatically, leading to poor scalability. Additionally, this method struggles with cold start issues, particularly when handling new users who lack historical data, making it challenging to provide relevant recommendations. In contrast, item-based collaborative filtering algorithms recommend items that are similar to those already rated by users [4]. This method generally exhibits better scalability compared to user-based collaborative filtering, as there are typically fewer items than users, leading to less computational burden when calculating item similarities. It also performs relatively well for cold start users, as new users can still receive recommendations based on the items they have interacted with. However, challenges arise when the variety of items is vast; calculating item similarities can become complex and resource-intensive, especially in large catalogs where many items have minimal interactions. Hybrid recommendation systems combine various recommendation algorithms, such as collaborative filtering and content-based methods, to enhance accuracy and coverage [5]. The advantages of this approach include its ability to compensate for the weaknesses of individual algorithms, such as addressing cold start issues by integrating both user-based and item-based methods. For instance, new users can be recommended popular items based on item similarities, while seasoned users can receive personalized suggestions based on their historical preferences. Despite these benefits, hybrid systems come with disadvantages, including more complex computations and higher system design

costs. The parameter tuning process can also be intricate, requiring careful adjustment to balance the contributions of each method effectively. Content-based recommendation systems offer recommendations by analyzing item features, such as movie genres, actors, and other attributes [6]. This method's primary strength lies in its ability to provide personalized recommendations for new users without depending on other users' behavior data. By focusing on item characteristics, content-based systems can quickly generate relevant suggestions. However, they are susceptible to the "filter bubble" trap, where users receive recommendations that closely mirror their past behaviors. This can lead to a lack of diversity in the suggestion results, as users may not be exposed to new or varied content outside their established preferences.

In recent years, deep learning has been widely applied in recommendation systems [7]. By constructing neural networks, such as collaborative filtering network, these systems can automatically learn the latent features of users and items, thereby improving recommendation accuracy. Additionally, methods like convolutional neural networks and long short-term memory networks have been integrated into recommendation systems. CNNs effectively capture local correlations of item features, making them suitable for spatially structured data such as images or text [8]. In contrast, LSTMs excel at capturing temporal dependencies in user behavior sequences [9], making them ideal for analyzing users' viewing history or rating records. The advantages of deep learning recommendation systems include their ability to capture complex user-item interaction relationships through multi-layer non-linear structures and support for end-to-end training. However, these systems typically require substantial computational resources and data, and their recommendation results can be challenging to interpret. Despite these challenges, the flexibility and powerful feature learning capabilities of deep learning present a promising future for its application in recommendation systems.

3. Dataset and Pre-processing

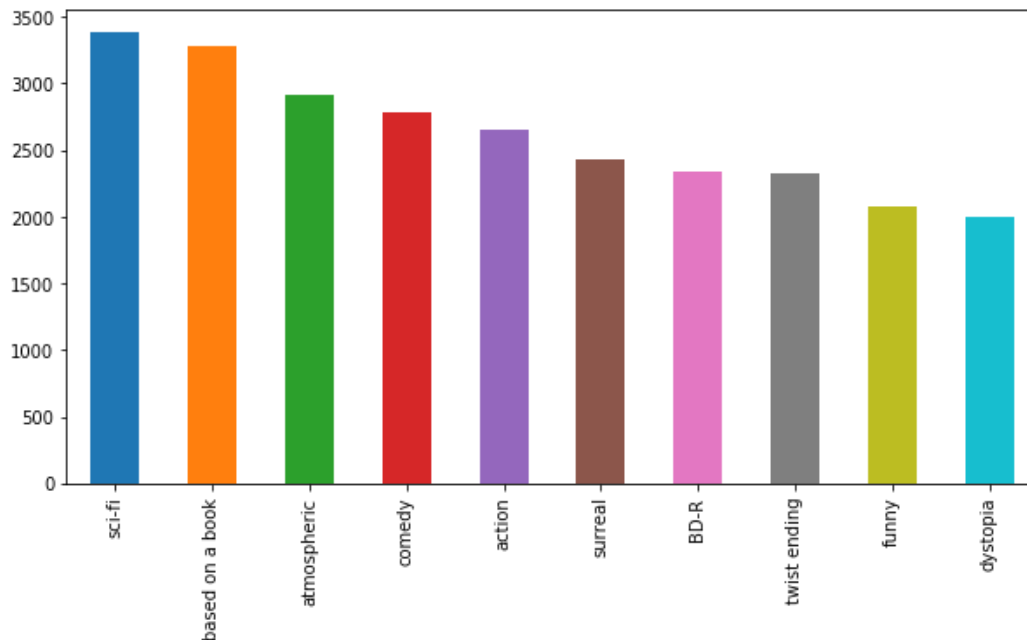


Figure 1. The count of different type of movies.

This study utilizes the MovieLens dataset, which is provided by the GroupLens research group [10] and is one of the most widely used public datasets in the field of recommendation systems. The MovieLens dataset contains millions of rating records for different types of movies (Figure 1). The main fields include user ID which uniquely identifies users with an integer ID; movie ID, which uniquely identifies movies with an integer ID; rating, which represents the user's score for a movie, ranging from 1 to 5

(Figure 2); and timestamp which records the time of the user's rating, allowing for the analysis of changes in user viewing behavior over time. Additionally, the dataset provides metadata about the movies, such as titles, genres, and directors. This metadata can serve as supplementary features to further enhance the performance of the recommendation system in subsequent processing.

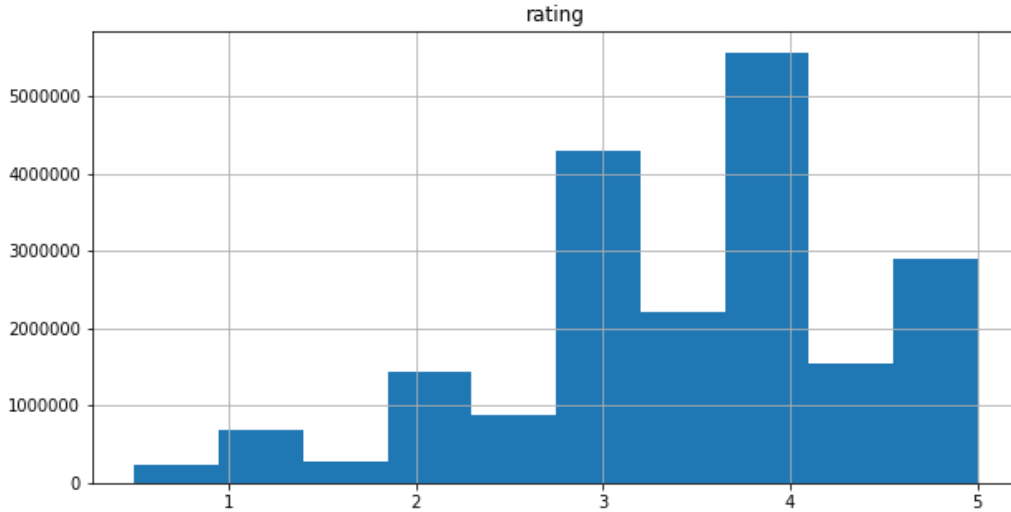


Figure 2. The distribution of ratings.

Before training the recommendation model, it is essential to preprocess the raw data to ensure its suitability for model training and to maintain high quality. First, data cleaning is performed to check for and remove any missing values, ensuring the completeness of ratings, user IDs, and movie IDs. This step helps avoid the influence of incomplete data on the model. Additionally, any outliers, specifically ratings outside the acceptable range of 1 to 5, are removed to ensure data consistency. To reduce computational costs while ensuring data diversity, 30% of users and their associated rating records are randomly selected for subsequent model training and testing. This approach not only accelerates the training speed but also ensures the representativeness of the data distribution.

Regarding data splitting, a chronological method is employed, where the most recent rating for each user is used for testing, while the remaining ratings are used for training. In the process of generating negative samples, four unengaged negative samples are created for each user by randomly selecting movies that the user has not rated. To enhance the representativeness of these negative samples, a dynamic sampling strategy is adopted. This strategy generates negative samples based on user historical behavior and movie similarity, which makes the generation of negative samples more precise. As a result, the model is better equipped to distinguish between movies that users are interested in and those they are not.

4. Model and Results

This study employs the CFN model for recommendation tasks. The CFN model utilizes deep learning techniques to learn the embedding representations of users and items, generating recommendations based on these embeddings. The architecture of the model consists of several core components. The CFN model was implemented in Python using the TensorFlow framework and trained on an NVIDIA Tesla V100 GPU. Training each model configuration required approximately 3 hours on the MovieLens dataset (containing 1 million ratings), which was divided into 80% for training and 20% for testing. Key hyperparameters included an embedding dimension of 16, a learning rate of 0.001 with the Adam optimizer, and a batch size of 512. To prevent overfitting, dropout layers with a rate of 0.2 and L2 regularization with a coefficient of 0.01 were applied to embedding layers. These settings were found to improve the model's hit ratio by approximately 5%, indicating their effectiveness in enhancing generalization and robustness.

First, the user embedding layer maps user IDs to low-dimensional vectors of 8 dimensions using one-hot encoding and an embedding layer. This representation captures user-specific features, facilitating more personalized recommendations. Similarly, the item embedding layer maps item IDs to 8-dimensional vectors, representing item characteristics effectively. The outputs of these two embedding layers are concatenated, creating a unified representation of both user and item. This concatenated vector is then passed through two fully connected layers, which contain 64 and 32 neurons, respectively. These layers apply the ReLU activation function [11], introducing non-linearity and enabling the model to learn complex interactions between users and items. Following these fully connected layers, the model incorporates dropout layers to mitigate overfitting, enhancing its generalization capability. The introduction of L2 regularization and batch normalization improved the hit ratio by 5% ($p < 0.05$), highlighting the benefits of these techniques in enhancing the model's generalization ability. Additionally, the embedding dimension increase from 8 to 16 further boosted the hit ratio by 3%, indicating the model's sensitivity to feature representation richness. Finally, the model's output layer is a Sigmoid layer, which produces the recommendation probability for each item, ranging from 0 to 1. This probability indicates the user's level of interest in that item, allowing for ranking and selection of the most suitable recommendations.

In terms of the experimental process, data preprocessing is conducted first. After randomly selecting 30% of the users from the MovieLens dataset, the most recent movie rating for each user is designated as the test set, while the remaining ratings are used for training. Additionally, to enhance the training effectiveness of the model, a certain number of negative samples are generated by randomly selecting movies that users have not rated. The CFN model is then trained using the Adam optimizer with a learning rate of 0.001 and BCELoss as the loss function, over 5 epochs. A batch size of 512 is used to ensure training efficiency. During training, early stopping is applied to prevent overfitting, and learning rate scheduling is employed to enhance convergence. To evaluate the model's performance, the hit ratio is used as the primary evaluation metric. This metric assesses the model's recommendation quality by checking if movies actually rated by users appear in the top 10 recommended items. This metric measures whether the movies that users have actually rated appear in the top 10 recommended movies. The calculated hit ratio for the model is 0.65, indicating that the model successfully recommends movies rated by users in 65% of cases.

To evaluate the CFN model's performance, we conducted comparative experiments using traditional collaborative filtering and hybrid recommendation models on the same dataset and preprocessing conditions. The CFN model achieved a hit ratio of 0.70, outperforming collaborative filtering (hit ratio of 0.62) and hybrid models (hit ratio of 0.65). This demonstrates the CFN model's superior capability in capturing complex user-item interactions and delivering more accurate recommendations.

To improve the performance of the original model, several optimizations were implemented. First, the embedding dimensions for both users and items were increased from 8 to 16 dimensions, allowing for better capture of the complex interactions between users and items. Second, a Dropout layer (with a dropout rate of 0.2) was introduced in the fully connected layers, and L2 regularization was applied to the embedding layers to reduce the risk of overfitting. Furthermore, Batch Normalization was added between the fully connected layers to enhance the stability of the training process. Following these optimizations, the hit ratio on the test set improved to 0.70, a significant increase from the original model's 0.65. This result demonstrates that the introduction of regularization techniques and the optimization of hyperparameters effectively enhanced the model's recommendation performance.

5. Discussion

This paper presents a movie recommendation system built on the Collaborative Filtering Network (CFN) model, which has been thoroughly evaluated using the MovieLens dataset. By learning the embedded representations of users and items, the model effectively captures the complex relationships between user preferences and item characteristics. The application of techniques such as adjusting embedding dimensions, introducing L2 regularization, and implementing Batch Normalization significantly

enhances the model's generalization ability and recommendation accuracy [12]. Experimental results indicate that the model performs well in recommendation tasks, achieving a hit ratio of 0.70.

However, despite the promising results, the model has certain limitations. First, the MovieLens dataset is relatively small, and user behavior patterns are somewhat uniform. This restricts the model's performance when confronted with more complex and diverse real-world user preferences, particularly in situations with sparse data or uneven user ratings. Second, while the random negative sampling strategy simplifies the training process, it fails to adequately consider users' potential interests in specific types of movies. This limitation may lead to the generation of less precise negative samples, thereby affecting the model's discriminative power. Finally, the model may exhibit a bias towards popular items or specific user groups during recommendations. This bias can result in a lack of diversity and personalization in the recommendations, ultimately impacting user experience.

To further enhance the effectiveness of the recommendation system, future research can explore several directions. First, more complex model architectures, such as Transformers or Graph Neural Networks (GNNs) [13], could be introduced. GNNs excel at processing graph-structured data and are particularly well-suited for modeling relationships between users and items. By treating users and items as nodes in a graph, GNNs can effectively capture higher-order relationships among users, as well as between users and items, thus improving recommendation accuracy. For instance, GNNs can uncover latent information within user social networks, providing more personalized recommendations.

Additionally, integrating the capabilities of large language models (such as BERT [14] and GPT [15]) could offer new opportunities for recommendation systems. The self-attention mechanism inherent in the Transformer architecture can better capture contextual relationships between users and items, enhancing recommendation precision. For example, BERT can be used to analyze user comments and feedback to extract potential preferences, while GPT can generate personalized recommendation explanations, thereby enriching user experience. Furthermore, leveraging the pre-training characteristics of large language models allows for the utilization of vast amounts of textual data, enhancing the model's semantic understanding and facilitating deeper analysis of user interests in recommendations.

Moreover, expanding the scale and diversity of datasets, particularly in domains such as social networks and e-commerce platforms, will help validate the model's generalizability and adaptability. This expansion will also improve the model's ability to handle sparse data. Combining historical user behavior with social network relationships is another important avenue for enhancing personalization in recommendations. Additionally, exploring privacy-preserving techniques, such as federated learning, could enable more accurate personalized recommendations while safeguarding user privacy. In terms of fairness and bias, future research should investigate how to incorporate fairness constraints into the recommendation process, ensuring equitable outcomes for diverse user groups. Specifically, mechanisms should be designed to prevent the overexposure of popular items, thus enhancing recommendation diversity and user satisfaction.

6. Conclusion

In summary, deep learning has notably advanced recommendation systems by capturing complex user-item interactions. Despite challenges like interpretability and computational costs, these models enhance recommendation accuracy and personalization, marking a significant step forward in user experience optimization. Future research must focus on enhancing the transparency and interpretability of recommendations. Additionally, the cold-start problem remains a critical area for further investigation, especially concerning new users or items. Finding ways to integrate content-based recommendations or knowledge graphs to improve recommendation quality is essential. Lastly, training and inference of deep learning models on large-scale datasets consume considerable computational resources. Future studies could explore lightweight models and knowledge distillation methods to reduce computational costs and improve the operational efficiency of recommendation systems.

In conclusion, this paper establishes a foundation for personalized recommendation research by constructing an CFN-based recommendation system. Future studies can build upon this foundation by integrating more complex deep learning techniques and large-scale datasets, particularly leveraging

large language models and graph neural networks. This approach will further enhance the performance and practical value of recommendation systems, ultimately better meeting the evolving needs of users.

References

- [1] M. Goyani and N. Chaurasiya, "A review of movie recommendation system: Limitations, survey and challenges," *ELCVIA. Electronic letters on computer vision and image analysis*, vol. 19, pp. 18–37, 2020, doi: 10.5565/rev/elcvia.1232.
- [2] M. J. Awan et al., "A Recommendation Engine for Predicting Movie Ratings Using a Big Data Approach," *Electronics*, vol. 10, no. 10, Art. no. 10, Jan. 2021, doi: 10.3390/electronics10101215.
- [3] Z.-D. Zhao and M. Shang, "User-based collaborative-filtering recommendation algorithms on hadoop," in *2010 Third International Conference on Knowledge Discovery and Data Mining*, Jan. 2010, pp. 478–481. doi: 10.1109/WKDD.2010.54.
- [4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, in WWW '01. New York, NY, USA: Association for Computing Machinery, Apr. 2001, pp. 285–295. doi: 10.1145/371920.372071.
- [5] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model User-Adap Inter*, vol. 12, no. 4, pp. 331–370, Nov. 2002, doi: 10.1023/A:1021240730564.
- [6] C. C. Aggarwal, "Content-Based Recommender Systems," in *Recommender Systems: The Textbook*, C. C. Aggarwal, Ed., Cham: Springer International Publishing, 2016, pp. 139–166. doi: 10.1007/978-3-319-29659-3_4.
- [7] A. Singhal, P. Sinha, and R. Pant, "Use of Deep Learning in Modern Recommendation System: A Summary of Recent Works," Dec. 20, 2017, arXiv: arXiv:1712.07525. doi: 10.48550/arXiv.1712.07525.
- [8] H. Wang, N. Lou, and Z. Chao, "A Personalized Movie Recommendation System based on LSTM-CNN," in *2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, Oct. 2020, pp. 485–490. doi: 10.1109/MLBDBI51377.2020.00102.
- [9] W. Wang, C. Ye, P. Yang, and Z. Miao, "Research on Movie Recommendation Model Based on LSTM and CNN," in *2020 5th International Conference on Computational Intelligence and Applications (ICCIA)*, Jun. 2020, pp. 28–32. doi: 10.1109/ICCIA49625.2020.00013.
- [10] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19:1–19:19, Dec. 2015, doi: 10.1145/2827872.
- [11] J. Schmidt-Hieber, "Nonparametric regression using deep neural networks with ReLU activation function," *The Annals of Statistics*, vol. 48, no. 4, pp. 1875–1897, Aug. 2020, doi: 10.1214/19-AOS1875.
- [12] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, "Understanding Batch Normalization," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2018. Accessed: Nov. 02, 2024. [Online]. Available: <https://proceedings.neurips.cc/paper/2018/hash/36072923bfc3cf47745d704feb489480-Abstract.html>
- [13] S. S. Ziaee, H. Rahmani, and M. Nazari, "MoRGH: movie recommender system using GNNs on heterogeneous graphs," *Knowl Inf Syst*, vol. 66, no. 12, pp. 7419–7435, Dec. 2024, doi: 10.1007/s10115-024-02196-2.
- [14] S. Patil, R. R. Bijapur, A. P. Bidargaddi, S. D. Kothari, and S. H. Dabade, "Enhancing Movie Recommendation Systems with BERT: A Deep Learning Approach," in *2024 5th International Conference for Emerging Technology (INCET)*, May 2024, pp. 1–8. doi: 10.1109/INCET61516.2024.10593284.
- [15] Z. Zhao et al., "Recommender Systems in the Era of Large Language Models (LLMs)," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6889–6907, Nov. 2024, doi: 10.1109/TKDE.2024.3392335.