Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/104/20241166

NCF-based Movie Recommendation System

Zehang Li

Harbin Institute of Technology, No. 92, West Dazhi Street, Nangang District, Harbin City, China

1468288815@qq.com

Abstract. This paper discusses the design and evaluation of a Neural Collaborative Filtering (NCF) model for movie recommendations using the MovieLens dataset. It addresses the limitations of traditional recommendation systems, such as content-based filtering and collaborative filtering, which struggle with data sparsity and the cold start problem. By incorporating deep learning, the NCF model enhances the accuracy and personalization of recommendations by learning the latent features of users and items and capturing complex interactions.

Keywords: Neural collaborative filtering, Word embedding, Deep learning, Movie recommendation systems.

1. Introduction

In today's society, an increasing device connecting to the internet produced vast amounts of real-time data. This requires more advanced techniques for data collection and processing. The rapid accumulation of data not only changes our lifestyles but also has a profound impact on business models and marketing strategies. In this context, utilizing big data to analyze user behavior and preferences for personalized product or service recommendations has become crucial. Recommendation systems, especially those providing content recommendations based on user historical data, serve as important bridges connecting users and products, significantly enhancing user purchase intentions and engagement. Movie recommendation systems, as a specific application of recommendation technologies, are becoming increasingly important. Movies, as a form of internet entertainment, have a broad audience. With hundreds of thousands of movies and diverse audience tastes, it is particularly important to recommend films that match viewers' interests. This requires personalized movie recommendations based on user interests and behaviors, filtering out redundant and irrelevant information to ensure accuracy. Furthermore, the development of movie recommendation systems also includes recommending related peripheral products based on user viewing history, such as movie soundtracks, merchandise, or related books. This multidimensional recommendation significantly enhances the user experience and creates more business opportunities for platforms.

However, with the dramatic increase in available data, traditional recommendation methods like Content-based Filtering and Collaborative Filtering have shown limitations in handling large datasets. These traditional methods, despite their early successes, often struggle with the high dimensionality and sparsity of data, especially when the user or item data is extensive. Traditional CF also faces issues like the cold start problem and scalability issues. These methods perform poorly when there is insufficient

DOI: 10.54254/2755-2721/104/20241166

interaction data for new users or items, making it difficult to provide accurate recommendations. Moreover, these algorithms often fail to capture nonlinear and complex patterns in user preferences, thus limiting the overall performance of recommendation systems.

To overcome these challenges, deep learning techniques have been introduced into the realm of recommendation systems, marking a significant advancement. Deep learning can learn hidden features and nonlinear relationships in large data sets through complex network structures, enhancing the depth of understanding of user preferences. This introduction has allowed recommendation systems to predict user behavior more accurately and provide more personalized recommendations. Neural Collaborative Filtering (NCF) is a technique that combines traditional CF with deep neural networks. NCF not only leverages the powerful feature extraction capabilities of deep learning but also learns the latent features of users and items through multi-layer neural networks. Using structures like embedding layers and nonlinear activation functions, NCF can better simulate complex interactions between users and items. Compared to traditional methods, NCF significantly improves the ability to handle large-scale, high-dimensional sparse data, achieving higher accuracy in personalized recommendations.

This paper constructs a lightweight NCF model based on word embeddings. This model employs a simple encoding method to spatially transform user and movie information and uses neural networks to learn the preference relationships between users and movies. We conducted extensive tests on the MovieLens dataset, and the results show that the NCF model can effectively predict users' movie preferences.

2. Previous Works

In the field of recommendation systems, traditional recommendation algorithms such as content-based recommendations, collaborative filtering, and hybrid methods have dominated.

Content-based recommendations are one of the most direct methods, analyzing the intrinsic features of items—such as text descriptions, categories, and tags—to identify user interests [1]. For example, if a user likes science fiction movies, the system analyzes the characteristics of the science fiction movies the user previously enjoyed, such as themes, plotlines, or directors, and then recommends other science fiction movies with similar characteristics. This method is relatively simple to implement because it does not rely on data from other users, allowing it to make effective recommendations even with limited user data. However, its main drawback is that it may confine users within a narrow range of recommendations, creating a so-called "filter bubble" that makes it difficult for users to encounter new or different types of content.

On the other hand, collaborative filtering techniques are based on community opinions and mainly use user similarity for recommendations [2]. It is mainly divided into two types: user-based collaborative filtering and item-based collaborative filtering. User-based collaborative filtering relies on finding groups of users with similar behavior patterns and recommending popular items within these groups to the current user. Item-based collaborative filtering, however, analyzes all users' feedback on items to find similar items, then recommends these items to users. The main advantage of this method is that it can use a large amount of user data to generate a wider range of recommendations, but it also has significant drawbacks, such as the cold start problem, which occurs when there is insufficient evaluation data for new users or items, making it ineffective. Moreover, as the system scales, processing a large number of user and item data requires substantial computational resources, increasing the cost of implementing collaborative filtering.

To address the drawbacks of these methods, hybrid recommendation systems have been proposed to integrate the advantages of content-based recommendations and collaborative filtering. By combining multiple recommendation methods, hybrid systems aim to provide more comprehensive recommendations, reduce the impact of "filter bubbles," and solve the cold start problem for new users or items. Implementation can be as simple as a strategy like weighted averaging or as complex as model fusion techniques, including machine learning models, to optimize recommendation results. This method has shown great potential in improving recommendation accuracy and coverage, but designing and maintaining an effective hybrid recommendation system requires high-level technical integration

DOI: 10.54254/2755-2721/104/20241166

and algorithm adjustment, potentially involving training multiple models and complex data management strategies.

As machine learning techniques are widely used in various application areas, recommendation systems have also begun to utilize these advanced algorithms to enhance the accuracy and efficiency of recommendations. Machine learning methods can handle common issues in traditional recommendation systems, such as data sparsity and the curse of dimensionality, and learn deep patterns and relationships from complex data. Support Vector Machines (SVMs) are a powerful supervised learning model commonly used for classification problems. In recommendation systems, SVMs can be used to distinguish items that users may like or dislike [3]. By constructing a hyperplane that maximizes the margin between two classes of data in high-dimensional space, SVMs can effectively perform binary classification. Moreover, SVMs can handle nonlinear classification problems by using kernel tricks to map data to higher-dimensional spaces, finding the optimal decision boundary. Random Forest is an ensemble learning method that improves prediction accuracy and stability by building multiple decision trees. Each decision tree is trained on a randomly drawn sample from the original dataset and selects a subset of features at random. In recommendation systems, Random Forest can be used to predict users' ratings or preferences for items [4]. By aggregating the predictions of multiple decision trees, it can reduce the risk of overfitting and enhance the accuracy of recommendations. Ensemble learning is a technique that improves the decision-making ability of a single model by combining multiple models. In recommendation systems, ensemble learning can integrate different recommendation algorithms and machine learning models, such as combining collaborative filtering, content-based recommendations, and model-based methods. This method can effectively leverage the advantages of various single techniques to provide more comprehensive and accurate recommendations.

Moreover, deep learning has shown its unique advantages in handling large-scale and high-dimensional data in recommendation systems. Deep learning models, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have been applied to parse complex user behavior sequences and evaluation patterns. For example, CNNs can be used to analyze users' visual preferences, such as learning patterns from users' favorite movie posters, while RNNs are suitable for processing time-series data, predicting dynamic changes in user preferences. NCF is a typical representative of combining deep learning with traditional collaborative filtering techniques [5]. It learns the latent feature representations of users and items through neural networks and uses network layers to simulate interactions between users and items. The key to NCF is that it not only learns explicit rating data but also learns from implicit user feedback (such as clicks or browsing), capturing complex nonlinear relationships and thus improving the accuracy and personalization of recommendations. Compared to traditional collaborative filtering, NCF can more effectively handle data sparsity and provide richer recommendations.

3. Dataset and Preprocessing

The dataset used in this project comes from the MovieLens website created by the GroupLens research team at the University of Minnesota's Department of Computer Science and Engineering. MovieLens provides a rich dataset of movie ratings, which are subdivided into multiple sub-datasets based on factors such as creation time and data volume. Each sub-dataset has a different format and size and is widely used in research in recommendation systems, data mining, and machine learning fields. These datasets not only include users' ratings and viewing records of movies but also contain metadata information about the movies, such as genre, ID, and timestamps.

Appropriate data preprocessing is crucial before using these datasets for model training and testing. This process first involves converting the timestamp fields into an easily processed datetime format. Next, data cleaning is performed, including ensuring the uniqueness of user IDs and randomly selecting 30% of the user data from the dataset for in-depth analysis. To effectively use the most recent user rating data, we sort the ratings based on timestamps, assigning the latest ratings to the test set and the rest as the training set. In the processing of the training set, we mark each user's rating as 1 and generate an interaction dataset between each user and movie. Using the zip function to pair user IDs and movie IDs,

DOI: 10.54254/2755-2721/104/20241166

we remove any potential duplicates. Additionally, to generate negative samples needed for the model, we save the movies each user has already interacted with as positive samples and randomly select movies the user has not interacted with as negative samples, maintaining an approximate 4:1 ratio of positive to negative samples. When selecting negative samples, we take special care to ensure these samples do not include movies the user has already rated, ensuring these negative samples are unknown to the user. By converting the final prepared user IDs, movie IDs, and corresponding label data into PyTorch tensors, we lay the foundation for the subsequent model training, enabling it to be more effective and ultimately enhancing the performance and accuracy of the recommendation system.

4. Model and results

We designed a new evaluation model that combines deep learning technology with the embedding structures of users and items to more effectively predict potential user preferences. The core of this method lies in processing complex input data through deep learning, showing significant advantages over traditional handcrafted feature-based recommendation algorithms.

The implementation of the model is based on the PyTorch framework. In the construction of the model, the initialization function first sets up embedding layers, converting user and item IDs into 8-dimensional embedding vectors. Then, the model processes these embedding vectors further through fully connected layers. The first fully connected layer (fc1) receives the concatenated vector of user and item embedding vectors, which has 16 dimensions (8 dimensions each for users and items), and outputs a 64-dimensional vector. The second fully connected layer (fc2) then converts this 64-dimensional input into a 32-dimensional vector. Finally, the output layer takes this 32-dimensional vector as input and outputs a value between 0 and 1 through a sigmoid function, representing the predicted rating or preference level. During the forward propagation process, the model first obtains the vector representations of users and items through the embedding layers, then concatenates these two vectors using the torch.cat function at the last dimension. After processing through two fully connected layers, where the ReLU activation function is used to introduce nonlinearity and enhance the model's expressive capability, the final output is the predicted user preference for the item.

Table 1. Model Architecture.

Layer	Name	Type	Params
0	user_embedding	Embedding	1 M
1	item_embedding	Embedding	1 M
2	fc1	Linear	1 K
3	fc2	Linear	2 K
4	output	Linear	33

When training the model, it receives a batch of user IDs, movie IDs, and corresponding labels, calculates the predicted values based on these inputs, and uses the binary cross-entropy loss function to evaluate its performance. During the process of optimizing model parameters, the model improves prediction accuracy through multiple iterations, with the training cycle set for a maximum of five times. To enhance training efficiency, we use GPU acceleration during the training process.

Finally, in the evaluation stage of the recommendation system, we generate a set of uninteracted items for each test user, randomly select 99 items, and include these items along with the target item in the test set. By inputting the user ID and item ID into the model, the model outputs the predicted ratings for these test items. By sorting these ratings, we extract the top 10 items with the highest scores, evaluate whether the target item appears among these recommended items, and calculate the Hit Ratio based on this to measure the effectiveness of the recommendation system. This method, which comprehensively utilizes deep learning and traditional collaborative filtering techniques, not only improves the accuracy of recommendations but also greatly enhances the recommendation system's understanding and prediction capabilities regarding user preferences.

DOI: 10.54254/2755-2721/104/20241166

During the testing phase of the experiment, our model underwent extensive evaluation on the user-movie rating dataset. The model's performance is measured by calculating the Hit Ratio, which reached 0.86 in this experiment. The Hit Ratio is defined as the proportion of movies in the recommended movie list that the user is actually interested in (based on the user's true preferences for movies in the test set). This result strongly demonstrates the recommendation system's effectiveness in accurately capturing the complex relationships between user preferences and movie characteristics, thereby providing high-quality recommendations.

5. Discussion and Conclusion

This study delves into the application of deep learning in the field of recommendation systems by constructing a Neural Collaborative Filtering (NCF) model, providing important references and foundations for similar research tasks. The study's results show that deep learning technology can effectively capture the complex relationships between user preferences and item characteristics. This model not only supports movie recommendations but can also be extended to e-commerce, music recommendations, and other fields, contributing to the development of personalized recommendation systems.

Despite the achievements of this study in applying deep learning to recommendation systems, there are still several limitations that may affect the model's generalizability and practicality. First, from the perspective of the dataset, although the MovieLens dataset used in this study is widely used in academic research, its relatively small number of users and items limits the model's ability to capture complex behavior patterns of users in the real world. Additionally, this dataset may not fully represent the diversity and dynamic changes of various user groups, which could lead to less precise or biased recommendations in practical applications. Secondly, there is a bias issue in the embedding learning process, especially for users and items with frequent interactions. This bias could cause the model to overly rely on existing frequent interaction data, neglecting those minority but potentially promising new users or items. This not only exacerbates the so-called cold start problem, where the model struggles to make effective recommendations for new users or items, but may also result in recommendations that are too concentrated on popular items, lacking personalization and diversity. In terms of computational complexity, deep learning models, particularly Neural Collaborative Filtering models, generally require substantial computational resources, which could pose a challenge for practical applications, especially in resource-limited environments. As the complexity of the model increases, the required computation time and costs also significantly grow, which may limit the widespread deployment of such models. Additionally, the current model's capability in handling unstructured data still needs improvement. Although deep learning excels in fields like image recognition and natural language processing, it still faces challenges in parsing complex user behaviors and subtle preference differences. This includes how to effectively integrate data from various sources, address data sparsity issues, and enhance the model's sensitivity to emerging trends.

Looking to the future, the application prospects of deep learning in recommendation systems are undoubtedly broad. With deeper research into algorithms and continuous enhancements in computing capabilities, deep learning models are expected to handle larger datasets and parse more complex data structures. In subsequent research, we plan to adopt more advanced neural network architectures, such as Graph Neural Networks, self-attention mechanisms, and transformers [6-7]. These technologies have strong modeling capabilities and can effectively improve the system's accuracy in capturing complex relationships between users and items. Additionally, as large models (such as large-scale pre-trained models based on Transformers) demonstrate superior performance in multiple fields, we also plan to explore the application of similar large model architectures in recommendation systems. These models, by pre-training on massive datasets, can greatly improve the recommendation system's depth of understanding and accuracy in user behavior. We will also strive to collect and integrate more diverse data sources, such as user behavior data from social media, e-commerce platforms, and other online interaction platforms. The diversity and richness of these data are key to improving the model's generalizability. At the same time, to achieve more precise personalized recommendations for users,

DOI: 10.54254/2755-2721/104/20241166

further development and optimization of active learning methods are also in our research plans. Through active learning, recommendation systems can learn users' latest preferences in real-time and quickly adapt to these changes, thereby providing more personalized services. Additionally, we are considering integrating reinforcement learning techniques to optimize long-term user satisfaction and the sustainable development of the system. Reinforcement learning can help the model consider long-term returns in a series of decisions, optimizing the sequence strategy of recommendations.

Overall, this study not only provides valuable experimental results and insights for the field of recommendation systems but also offers a clear research and development roadmap for future researchers. As technology progresses and data resources become more abundant, the research and application of recommendation systems will become more precise and intelligent, greatly promoting the development of personalized services, enhancing user satisfaction, and increasing the commercial value of systems.

References

[1] 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.

[2] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative Filtering Recommender Systems," HCI, vol. 4, no. 2, pp. 81–173, May 2011, doi: 10.1561/1100000009.

[3] S.-H. Min and I. Han, "Recommender Systems Using Support Vector Machines," in Web Engineering, D. Lowe and M. Gaedke, Eds., Berlin, Heidelberg: Springer, 2005, pp. 387–393. doi: 10.1007/11531371_50.

[4] H.-R. Zhang and F. Min, "Three-way recommender systems based on random forests," Knowledge-Based Systems, vol. 91, pp. 275–286, Jan. 2016, doi: 10.1016/j.knosys.2015.06.019.

[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural Collaborative Filtering," in Proceedings of the 26th International Conference on World Wide Web, in WWW '17. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, Apr. 2017, pp. 173–182. doi: 10.1145/3038912.3052569.

[6] C. Sun et al., "Attention-based graph neural networks: a survey," Artif Intell Rev, vol. 56, no. 2, pp. 2263–2310, Nov. 2023, doi: 10.1007/s10462-023-10577-2.

[7] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," AI Open, vol. 3, pp. 111–132, Jan. 2022, doi: 10.1016/j.aiopen.2022.10.001.