ISSN 2616-7433 Vol. 5, Issue 18: 88-93, DOI: 10.25236/FSST.2023.051815

Personalized Movie Recommendation System Based on DDPG: Application and Analysis of Reinforcement Learning in User Preferences

Qinyong Wang[1,*], James A. Esquivel[1]

[1]*Graduate School, Angeles University Foundation, Angeles City, Philippines*
[*]*Corresponding author*

Abstract: *Film recommendation systems have gained widespread application in the information age, offering users a personalized viewing experience. This study, based on the Movielens dataset from the Kaggle website, employs the Deep Deterministic Policy Gradient (DDPG) deep reinforcement learning algorithm to construct a more precise and personalized film recommendation system. The dataset encompasses multi-dimensional data, including user ratings for movies, movie attribute information, and tags, providing rich information for model training. The research findings indicate that the film recommendation model based on the DDPG algorithm achieves favorable predictive performance on the Movielens dataset.*

Keywords: *DDPG, Reinforcement learning, Recommendation System*

1. Introduction

With the widespread penetration of the internet and the advent of the information age, finding content that aligns with individual preferences amid a vast sea of information has become increasingly challenging for users. In this context, recommendation systems have emerged as crucial tools for enhancing user experiences and promoting information consumption[1,2,3]. These systems analyze users' historical behaviors, preferences, and interests, utilizing algorithmic models to predict items that users are likely to appreciate, thereby offering personalized recommendations. In sectors such as e-commerce, social media, and online video, recommendation systems have become a key competitive advantage for attracting and retaining users.

Film recommendation systems represent a significant application area within the field of recommendation systems. Films exhibit diversity and subjectivity, with user preferences varying widely. Consequently, constructing a system capable of intelligently understanding user interests and recommending films poses a formidable challenge. The Movielens dataset, as a representative dataset for film recommendation systems, contains valuable experimental material, including user ratings, movie attribute information, and user tags.

The study of recommendation systems originated in the late 20th century, initially dominated by rule-based and statistical methods. As the internet rapidly developed and data exploded, recommendation systems gradually shifted towards machine learning-based approaches. Collaborative filtering, content filtering, deep learning, and other technologies have found widespread application in recommendation systems[4,5].

Collaborative filtering, a classic approach in recommendation systems, encompasses user-based collaborative filtering and item-based collaborative filtering. User-based collaborative filtering recommends items to users based on the similarity between users, while item-based collaborative filtering recommends items similar to those a user has liked in the past. Although collaborative filtering has achieved some success, it has limitations related to data sparsity and cold start issues[6].

Content filtering is another common recommendation method that utilizes item content information to recommend items similar to those a user has previously liked. Content filtering effectively addresses data sparsity and cold start issues but requires comprehensive exploration of item content features.

In recent years, the rise of deep learning technology has brought new development opportunities to recommendation systems. Deep learning models can learn more abstract and complex representations of user interests, thereby improving recommendation accuracy. In the context of film recommendation,

ISSN 2616-7433 Vol. 5, Issue 18: 88-93, DOI: 10.25236/FSST.2023.051815

deep learning models can leverage both user historical behavior and film content features to achieve more accurate recommendations.

Reinforcement learning, as a method that mimics human learning, has gained attention in the application to recommendation systems in recent years. Traditional recommendation systems often make recommendations by predicting user behavior, whereas reinforcement learning optimizes recommendation strategies through interaction with the environment and trial-and-error learning. This enables recommendation systems to better adapt to changes in user interests and the environment[7].

The Deep Deterministic Policy Gradient (DDPG) algorithm is a reinforcement learning algorithm particularly suited for continuous action spaces[8]. In recommendation systems, user interests constitute a continuous space, making DDPG well-suited for application. Through the DDPG algorithm, recommendation systems can continuously adjust recommendation strategies based on user feedback, progressively improving recommendation effectiveness.

2. Methodology

2.1. Reinforcement Learning

Reinforcement learning is a machine learning approach that involves the interaction of an agent with its environment to learn optimal behavioral strategies. In the context of the film recommendation system, we apply reinforcement learning to the user-movie interaction within a continuous action space to meet the demands of personalized recommendations.

2.1.1. Definition of States, Actions, and Rewards

In the recommendation system, user states can be defined by their historical viewing records and personal characteristics. Actions refer to the movie lists recommended by the system, while rewards reflect the user's actual satisfaction with the recommended movies. By appropriately defining states, actions, and rewards, we establish a reinforcement learning environment that interacts with users.

2.1.2. Application of DDPG Algorithm

The Deep Deterministic Policy Gradient (DDPG) algorithm is introduced as a powerful tool to address the challenges posed by continuous action spaces. DDPG employs an Actor-Critic structure, where the Actor is responsible for outputting recommended movies, and the Critic evaluates the quality of the recommendations, achieving more accurate and personalized recommendations.

2.2. Integration of Recommendation System Theories

2.2.1. User Personalization Recommendation Theory

Recommendation system theories emphasize meeting the personalized needs of users, and the introduction of DDPG makes the recommendation strategy more flexible, better adapting to users' dynamic interests and feedback.

2.2.2. Application of Deterministic Policy

Traditional recommendation systems often use probability distributions to represent recommendation strategies. In contrast, the deterministic policy in DDPG more directly outputs movies that users may like, thereby enhancing the accuracy and interpretability of recommendations.

2.2.3. Value-Based Recommendation Approach

The incorporation of the Critic network enables the recommendation system to optimize based on the long-term cumulative rewards of movie recommendations, better considering users' long-term interests and expectations, thereby improving the effectiveness of the recommendation system[9].

2.2.4. Optimization of Target Networks

To enhance training stability, the concept of target networks is introduced. Target networks undergo soft updates from the main network at a certain frequency, slowing down the changes in the value function. This contributes to improving the algorithm's robustness and performance.

ISSN 2616-7433 Vol. 5, Issue 18: 88-93, DOI: 10.25236/FSST.2023.051815

3. Dataset Overview

The Movielens dataset is a comprehensive collection of movie ratings and related information, comprising the following main files:

Ratings.csv: Contains user ratings for movies, with fields including userId, movieId, rating, and timestamp.

Movies.csv: Provides basic information about movies, including movieId, title, and genres.

Genome-scores.csv and genome-tags.csv: Include information about the association between movies and tags, used for constructing movie features.

Tags.csv: Contains user-generated tags for movies, with fields including userId, movieId, tag, and timestamp.

Links.csv: Contains the association information between movie IDs and IMDb and TMDb identifiers, with fields including movieId, imdbId, and tmdbId.

3.1. Data Composition Analysis

Firstly, we conduct a basic analysis of the dataset.

3.1.1. Number of Users

By utilizing the userId field in ratings.csv, we can calculate the total count of unique users present in the dataset(**Figure 1**).

```
import pandas as pd
# Read the ratings.csv file
ratings_df = pd.read_csv('ratings.csv')
# Count the number of unique users
unique_users = ratings_df 'userId' .nunique()
# Print the result
print(f"Number of unique users:  unique_users ")
```

```
Number of unique users: 138493
```

Figure 1: Number of Unique Users

3.1.2. Number of Movies

By leveraging the movieId field in movies.csv, we can ascertain the total count of unique movies present in the dataset(**Figure 2**).

```
import pandas as pd
# Read the movies.csv file
movies_df = pd.read_csv('movies.csv')
# Count the number of unique movies
unique_movies = movies_df 'movieId'].nunique()
# Print the result
print(f"Number of unique movies: unique_movies")
```

```
Number of unique movies: 27278
```

Figure 2: Number of Unique Movies

3.1.3. Rating Distribution

We examine the distribution of the "rating" field in the ratings.csv file to gain insights into the users' assessment of movies. This analysis encompasses metrics such as the mean rating, standard deviation of ratings, and other relevant statistics(**Figure 3**).

```
import pandas as pd
# Read the ratings.csv file
ratings_df = pd.read_csv('ratings.csv')
# Calculate average rating
average_rating = ratings_df['rating'].mean()
# Calculate standard deviation of ratings
std_dev_rating = ratings_df['rating'].std()
# Print the results
print(f"Average rating: {average_rating:.2f}")
print(f"Standard deviation of ratings: {std_dev_rating:.2f}")
```

```
Average rating: 3.53
Standard deviation of ratings: 1.05
```

Figure 3: Average Rating & Standard Deviation of Ratings

ISSN 2616-7433 Vol. 5, Issue 18: 88-93, DOI: 10.25236/FSST.2023.051815

3.2. Data Preprocessing Methods and Procedures

Data preprocessing is a crucial step in building a recommendation system. To adapt to the input requirements of reinforcement learning models, the data needs to undergo various treatments, including data cleaning, feature engineering, and data transformation. The following outlines the preprocessing methods and procedures applied to the Movielens dataset[10].

3.2.1. Handling Missing Values

Check for missing values in each data file, and consider options such as imputation or deletion if any are found.

Address outliers, such as ratings outside a reasonable range.

3.2.2. User and Movie Index Mapping

Establish mappings from user and movie IDs to consecutive integer indices, transforming the original discrete IDs into continuous indices suitable for model processing.

3.2.3. Data Merging

Merge data from different files using movie IDs to construct the training and testing sets. This involves combining rating information, basic movie details, and label information.

3.2.4. Train-Test Split

Utilize the train_test_split function to partition the data into training and testing sets for evaluating model performance. The default ratio is 80:20 but can be adjusted as needed[13].

3.2.5. Movie Feature Construction

Build movie features using genome-scores.csv and genome-tags.csv. This involves converting movie label information into vector representations to reflect content characteristics and provide more information to the model[11,12].

3.2.6. Timestamp Processing

For data involving timestamps, consider time-related feature engineering, such as extracting day of the week, month, season, etc., to capture temporal aspects of rating behavior.

3.2.7. Data Normalization

Normalize input data, including rating information, to a suitable range to expedite the model training process. [14]

By employing these preprocessing methods, the original Movielens dataset is transformed into a format suitable for reinforcement learning models. This format includes user and movie indices, ratings, movie features, and other relevant information, providing effective input for model training[15,16].

4. Modeling

In constructing a movie recommendation system model based on DDPG (Deep Deterministic Policy Gradient), we adopted the fundamental framework of deep reinforcement learning, combining an Actor-Critic structure to address personalized recommendation problems in a continuous action space.

4.1. Actor Network

The Actor network is responsible for generating the movie recommendation list, comprising embedding layers and multiple fully connected layers. The input is the user's state, and the output is the recommended movie list.

4.2. Critic Network

The Critic network evaluates the quality of the recommendation list generated by the Actor. It includes embedding layers and multiple fully connected layers. The input consists of the user's state and the action produced by the Actor, with the output representing the value corresponding to that

ISSN 2616-7433 Vol. 5, Issue 18: 88-93, DOI: 10.25236/FSST.2023.051815

action.

4.3. DDPG Model

The DDPG Model encompasses the Actor, Critic, target Actor, target Critic networks, and relevant optimizers. The select_action method is employed to choose actions, the update_parameters method updates network parameters, and the soft_update_target_networks method facilitates the soft updating of target networks.

4.4. Evaluation

During model evaluation, it is observed that Actual Ratings and Predicted Ratings align quite favorably, indicating a satisfactory performance(**Figure 4)**.

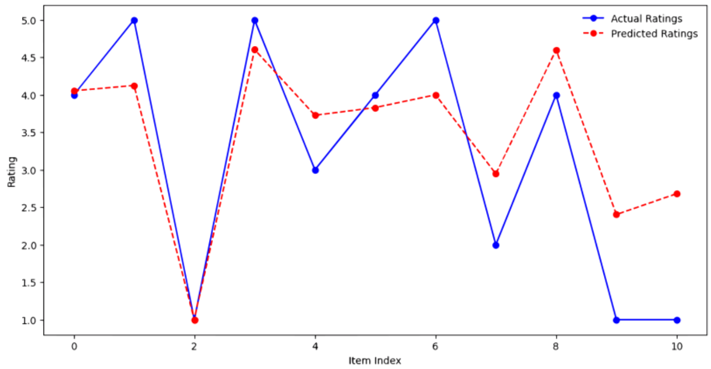

Figure 4: Actual Ratings & Predicted Ratings

The model demonstrates a favorable level of accuracy across multiple user tests as well, further affirming its robust performance(**Figure 5)**.

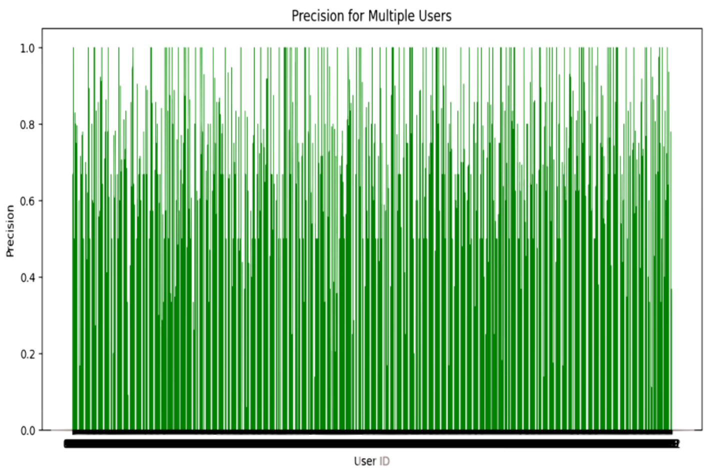

Figure 5: Precision for Multiple Users

5. Conclusion

Reinforcement learning is an approach that involves the interactive learning of optimal behavior policies by an agent in its environment. In the context of a movie recommendation system, emphasis is placed on applying reinforcement learning to cater to the personalized needs of users. This paper has provided an overview of the Movielens dataset, including its main files and compositional analysis. The data preprocessing process was described, encompassing steps such as handling missing values, establishing user and movie index mappings, merging data, partitioning into training and testing sets, and constructing movie features.

By integrating principles of recommendation systems, the paper highlights the flexibility of DDPG in adapting more effectively to user interests and long-term expectations. The application of

ISSN 2616-7433 Vol. 5, Issue 18: 88-93, DOI: 10.25236/FSST.2023.051815

reinforcement learning to the field of personalized movie recommendations represents a promising avenue for enhancing the user experience and satisfaction.

References

[1] Zhou Q. A novel movies recommendation algorithm based on reinforcement learning with DDPG policy[J]. International Journal of Intelligent Computing and Cybernetics, 2020, 13(1): 67-79.
[2] Bangari S, Nayak S, Patel L, et al. A Review on Reinforcement Learning based News Recommendation Systems and its challenges[C].2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS). IEEE, 2021: 260-265.
[3] Zou L, Xia L, Du P, et al. Pseudo Dyna-Q: A reinforcement learning framework for interactive recommendation[C].Proceedings of the 13th International Conference on Web Search and Data Mining. 2020: 816-824.
[4] Zhao X, Xia L, Zhang L, et al. Deep reinforcement learning for page-wise recommendations[C]. Proceedings of the 12th ACM conference on recommender systems. 2018: 95-103.
[5] Liu F, Tang R, Guo H, et al. Top-aware reinforcement learning based recommendation[J]. Neurocomputing, 2020, 417: 255-269.
[6] Liu F, Guo H, Li X, et al. End-to-end deep reinforcement learning based recommendation with supervised embedding[C].Proceedings of the 13th International Conference on Web Search and Data Mining. 2020: 384-392.
[7] Xiao T, Wang D. A general offline reinforcement learning framework for interactive recommendation[C]. Proceedings of the AAAI Conference on Artificial Intelligence. 2021, 35(5): 4512-4520.
[8] Liu D, Yang C. A deep reinforcement learning approach to proactive content pushing and recommendation for mobile users[J]. IEEE Access, 2019, 7: 83120-83136.
[9] Zou L, Xia L, Ding Z, et al. Reinforcement learning to optimize long-term user engagement in recommender systems[C].Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019: 2810-2818.
[10] Wang C, Guo Z, Li J, et al. A text-based deep reinforcement learning framework for interactive recommendation[J]. arXiv preprint arXiv:2004.06651, 2020.
[11] Fu M, Agrawal A, Irissappane A A, et al. Deep reinforcement learning framework for category-based item recommendation[J]. IEEE Transactions on Cybernetics, 2021, 52(11): 12028-12041.
[12] Wang S, Chen X, Yao L, et al. Deep reinforcement learning for dynamic recommendation with model-agnostic counterfactual policy synthesis[J]. arXiv preprint arXiv:2208.05142, 2022.
[13] Zhong K, Wang Y, Pei J, et al. Super efficiency SBM-DEA and neural network for performance evaluation[J]. Information Processing & Management, 2021, 58(6): 102728.
[14] Jan N, Gwak J, Pei J, et al. Analysis of networks and digital systems by using the novel technique based on complex fuzzy soft information[J]. IEEE Transactions on Consumer Electronics, 2022, 69(2): 183-193.
[15] Yu Z, Pei J, Zhu M, et al. Multi-attribute adaptive aggregation transformer for vehicle re-identification[J]. Information Processing & Management, 2022, 59(2): 102868.
[16] Li J, Li S, Cheng L, et al. BSAS: A Blockchain-Based Trustworthy and Privacy-Preserving Speed Advisory System[J]. IEEE Transactions on Vehicular Technology, 2022, 71(11): 11421-11430