Deep Reinforcement Learning Recommendation System based on GRU and Attention Mechanism

Yan-e Hou, Wenbo Gu, Kang Yang and Lanxue Dang

Abstract—**Recommending personalized content from massive data for users is the key function of the recommendation system. The recommendation process of the traditional recommendation systems is often regarded as static, which cannot reflect the changes of user's real-time interest. This paper addressed this problem and presented a recommendation model that leverages the ability of deep learning methods to effectively deal with decision-making problems. In this model, a state generation module containing gate recurrent unit (GRU) and attention network was designed to obtain user's long and short-term preferences as well as history scores. Then, an actor-critic algorithm was employed to imitate the real-time recommendations. We trained the proposed model and evaluated it on four well-known public datasets. It is proved that the proposed model is superior to existing recommendation models.**

Index Terms—**recommendation system, deep reinforcement learning, attention network, GRU, actor-critic.**

I. Introduction

RECOMMENDATION systems (RS) assume an irreplaceable role in current online services to improve the user experience. The effectiveness of RS in information filtering has led to its widespread use in the commercial fields, and there exist some successful applications [1], [2], [3].

Recommendation algorithms can effectively improve the recommendation effectiveness of RS, which are generally regarded as the core of RS. As a general rule, recommendation algorithms can be classified into two distinct types: traditional recommendation algorithms and those based on artificial intelligence methods. Among the traditional recommendation methods, collaborative filtering is the most classical and extensively used method, which is the focus of research in the field of RS [4]. In spite of this, the traditional recommendation algorithms face a very serious data sparsity problem, which can affect the effectiveness of recommendation. Furthermore, providing the long-term accurate item recommendations for users is the main purpose of RS. However, the recommendation process of most current recommendation models usually are taken as static, which does not conform to the actual recommendation. This shortcomings may cause these traditional models not to be well adapted to new situations, and at the same time face serious cold start problems. Therefore, new technologies and methods are seek to solve above issues.

With the rapid development of artificial intelligence technology, combing intelligent technology with recommendation method has become mainstream for the current recommendation systems. Reinforcement learning can be compatible with the process of sequential recommendation because of its step-by-step learning characteristics, which has been applied in some recommendation system [5], [6]. In addition, combing deep learning with reinforcement learning [7] for solving RS problems also has gradually emerged in recent years. These successful experiences provide a new direction to these problems.

This paper presents a deep reinforcement learning-based recommendation model, which is called DRR-GRU. The proposed model was made up of a state acquirement module and an actor-critic algorithm [8]. The former was used to extract features by a gate recurrent unit and an attention model, and imitate the complex and dynamic user interaction process to generate current user's state. Next, the generated status information is taken as the input of the actor-critic algorithm, which is employed to imitate the interaction between users and RS, predict the next step and evaluate the action according to the status information of users. After online and offline model training, we utilize some public datasets to investigate the effectiveness of the proposed model.

The remainder of the paper is organized as follows. Section II gives the related literatures about recommender methods in recent years. The proposed model and its training procedure are presented in Section III. Experimental results and comparison analysis are given in Section IV. Finally, Section V describes the conclusion and future research directions.

II. Related Works

In this review, we focus on the related literatures about traditional methods and those methods based on deep learning respectively.

It is generally agreed that traditional recommendation approaches has three categories: collaborative filtering algorithms, content-based algorithms, and hybrid recommendation algorithms. There are a lot of literatures about them because of their spread applications in many fields [9], [10]. The origin important milestone of collaborative filtering algorithms is the GroupLens system [11]. The collaborative filtering approaches are extensively applied in the traditional recommendation systems [4], [9] owing to its good recommendation performance. However, its sparsity and scalability problems still need to be solved to enhance the recommendation accuracy. Bobadilla et al. [12] introduced genetic algorithms to improve the calculation of similarity

Manuscript received December 17, 2022, revised April 14, 2023.

Yan-e Hou is an associate professor of Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, 475004, China (e-mail: houyane@henu.edu.cn)

Wenbo Gu is a graduate student of College of Computer and Information Engineering, Henan University, Kaifeng, 475004, China (e-mail:guwenbo@henu.edu.cn)

Kang Yang is a graduate student of College of Computer and Information Engineering, Henan University, Kaifeng, 475004, China (e-mail:yangkang@henu.edu.cn)

Lanxue Dang is an professor of Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, 475004, China (corresponding author,e-mail: danglx@vip.henu.edu.cn)

between users, significantly enhancing the performance of recommendation model. The description and the faced challenges of collaborative filtering methods was summarized by the literature [9]. Further, the author also provided an outlook on the future development of the filed. Content-based recommendation algorithms do not involve user behavior, which can still make recommendations in a cold start environment. The reference [10] provides an introduction of content-based recommendation algorithms and summarizes their applications in various domains. For complex recommendation environments, any single recommendation approach faces challenges, so hybrid recommendation methods come forth. Tian et al. [13] used collaborative filtering and content-based recommendation algorithms for hybrid recommendation, and the results demonstrated that hybrid recommendation algorithms are more effective than single recommendation algorithms. Further, in order to get higher accuracy of the e-commerce recommendation system, a complex but efficient hybrid recommendation algorithm was proposed in [14], which mixed a content-based recommendation method, an item-based collaborative filtering recommendation approach and a demographic-based recommendation method. The experimental results proved that the hybrid method had better recommendation effects.

The rapid development of deep learning technology makes it an effective tool to tackle many complex problems. More and more researchers has been attracted because of its powerful learning abilities and excellent problem-solving skills. There exist many researches about deep learning methods applied in the field of recommendation systems [15], [16], [17], [18]. The lasted review about existing recommendation systems based on deep learning could be found in [15], which also classified these systems in detail, and gave the future research directions. Fang et al. [18] give a review of temporal recommendation models in detail. To represent user's temporal interests, Zhang et al. [16] introduced an attention mechanism into a sequence-sense recommendation model. After that, Wu et al. [17] adopted the graph neural network(GNN) to get the complex transformation of recommendation items and then created a GNN-based recommendation model. For some real-time recommendation systems, reinforcement learning methods have become the preferred methods [1], [19], [20]. In view of the fact that the recommendation process and reinforcement learning can both be described as a Markov decision process (MDP), Huang et al. [19] designed a recurrent neural network to imitate the interaction process between RS and user. And then the authors proposed a top-N interactive recommender system to increase the accuracy of long-term recommendation. Liu et al. [20] developed a deep reinforcement learning-based recommendation framework, which used actor-critic algorithm to simulate the interaction process.

Review of these research shows that some recommendation algorithms have successfully developed to some extent, but more effective recommendation approaches still need to be developed because of its huge value and application potential in commercial fields. Although deep learning methods has been used in some recommendation systems, it is still worth developing more efficient recommendation model in the era of big data information.

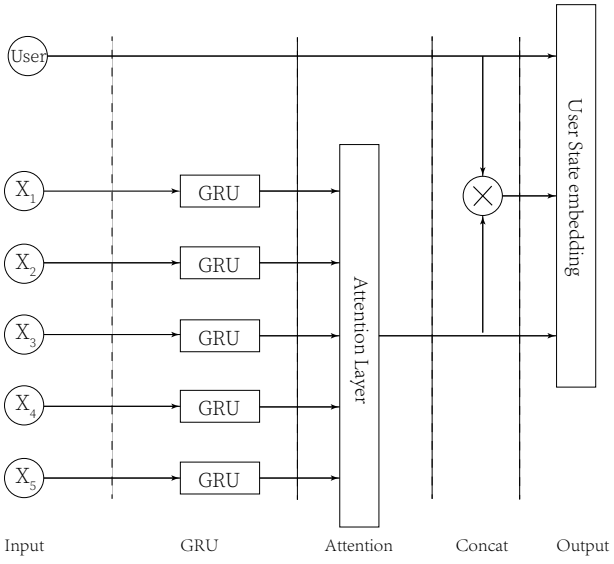


Fig. 1. State Generation Network

III. Model Description and Training

A. Model Design

Our proposed recommendation model named DRR-GRU is made up of two parts: state generation and recommendation generation. The former is based on GRU and attention mechanism which belong to deep learning. The latter is based on an actor-critic algorithm, which is a well-known reinforcement learning method. The state generation module is used to get current state information of users, which is taken as the input of the recommendation generation module. While for recommendation generation module, it firstly generates a Top-N recommendation list based on the user's current state, and then obtains user feedback. At the same time, the recommendation module is also responsible for updating the network parameters. Overall, the designed model can update the network parameters immediately to simulate a real-time recommendation environment for the user.

1) State Generation module: The structure of state generation module is shown in Fig. 1. The module is divided into five layers: the input layer, the GRU layer, the attention remembering layer, the fusion layer and the output layer. The input information in the input layer is the user feature matrix and the feature matrix of the user's first N interaction items at moment t. In the GRU layer, GRU is adopted to get the long-term and short-term interests of the user. The attention remembering layer is used to obtain the important features in the data. Finally, we connect them with the user embedding to obtain the current user state information as the output. Based on the features of GRU, the state generation module is able to take into account the long and short-term interest preferences of users, which can make more accurate item recommendations for users.

GRU is a model of tackle sequential data, and it is also a kind of recurrent neural network. GRU is a variant of Long Short Term Memory(LSTM), which has a simpler structure than LSTM and works well. GRU can capture the correlation in long-time series well and apply it to the model in this

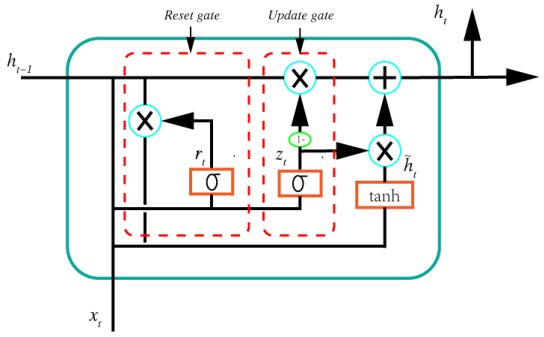


Fig. 2. Illustration of the GRU cell

paper, so that the model can consider both long and short-term preferences. Fig. 2 gives the structure of GRU model.

The GRU mainly includes two parts, namely reset gate and update gate. Assume that the input information at moment — t is x_t. The update gate determines which state information from the previous moment is discarded. When the previous moment state information h_{t-1} and x_t are passed into the current state h_t, the larger the value of the update gate, the more state information is brought at the previous moment. The value of update gate is calculated by the equation (1).

$$z_t = \sigma(W_z + U_z h_{t-1} + b_z) \tag{1}$$

where σ is a sigmoid function. x_t is the input state at moment t, and b_z denotes the bias of update gate. W_z and U_z represent the weight of the update gate.

The reset gate controls how much state information h_{t-1} at the previous time is written to current state $\tilde{h_t}$. When the output of reset gate is smaller, it means that the written state information at previous state is little. The specific calculation formula of the reset gate is shown in equation (2).

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$

where r_t is the output of reset gate at moment t. W_r and U_r represent the weight of the reset gate, and b_r denotes the bias of reset gate.

The output h_t of GRU is affected by z_t, $\tilde{h_t}$ and h_{t-1}. The calculation process is defined in equation(3) and equation(4).

$$\tilde{h_t} = tanh(W * [r_t * h_{t-1}], x_t) \tag{3}$$

$$h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h_t} \tag{4}$$

where $*$ denotes multiplication by element.

Therefore, after the calculation of GRU layer, we can get a state set $H = \{h_1, h_2, , h_t\}$. Similarly, after the attention layer, we can obtain the significant feature set $A = Attention(H)$ of H. Equation (5) represents the state generation module.

$$s_t = [u, u \otimes A, A] \tag{5}$$

where u indicates the users characteristic matrix, and $\otimes$ is the product of elements. The dimension of user matrix u, item A and s_t are k, k, and $3k$ respectively.

2) Actor-Critic Network: This paper uses the Actor-Critic algorithm as the reinforcement learning component. The Actor-Critic algorithm is a reinforcement learning algorithm, which has the characteristics of both value function and policy network. Actor is in charge of the generation of actions and interaction with users according to the status. Critic takes charge of the evaluation of actions.

The input of the Actor network is the state information s_t generated by the state generation module, and the output is the user's action information a_t at time t. The user's action information a_t is matrix multiplied with the item feature matrix to obtain the predicted ranking of the items, and the top N items are taken as the Top-N recommendation list. For the Critic network, the recommendation actions of the Actor are evaluated and the network parameters are updated. The overall network model of the proposed DRR-GRU in this paper is shown in Fig. 3.

B. Model Training

Inspired by [1] and [3], the model training experiment into include online mode and offline mode two types. For offline mode, project recommendation is implemented by the traditional pre-trained models, so the recommendation process is static. The online model uses the features of reinforcement learning algorithm to imitate the real recommendation environment and gradually recommend for users, while updating the network parameters every several steps to better adapt to the changes of users' interests.

1) Offline Mode: Like the traditional static methods, the offline mode consists of training and testing. We set the ratio of training set and test set is 7:3. Firstly, we randomly initialize the user and projection feature matrices using PMF [21]. Secondly, the input data of the state generation module are the user feature matrix U and the project feature matrix I, where U selects the first N interaction projects of the user. The state generation module outputs the users current state information s_t, and then uses the Actor-Critic algorithm to simulate the recommendation process of project and user. The user feedback is taken as a basis for evaluating the quality of recommendations. The evaluation of recommendation is based on r_t, s_t and action a_t. Additionally, the parameters of the network are also updated. For each pair of state-action (s_t, a_t), the calculation of Q-value function used in this paper is defined in equation (6).

$$Q_\omega(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r_t + \gamma Q_\omega(s_{t+1}, a_{t+1}) \mid s_t, a_t] \tag{6}$$

As can be seen from the previous text, the Q-value function evaluates the quality of recommendations and is the output of the critic network. Then the actor network parameters are updated by equation (7), and the update operation is performed through the sampling strategy gradient.

$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N} \sum_t \nabla_a Q_\omega(s, a) \mid_{s=s_t, a=\nabla_\theta \pi_\theta(s_t)} \nabla_\theta \pi_\theta(s) \mid_{s=s_t} \tag{7}$$

In equation (7), $J(\pi_\theta)$ is the action selection strategy of the actor algorithm. The loss function calculation functions is defined in equations (8) and (9).

$$Loss = \frac{1}{N} \sum_i (y_i - Q_\omega(s_i, a_i))^2 \tag{8}$$

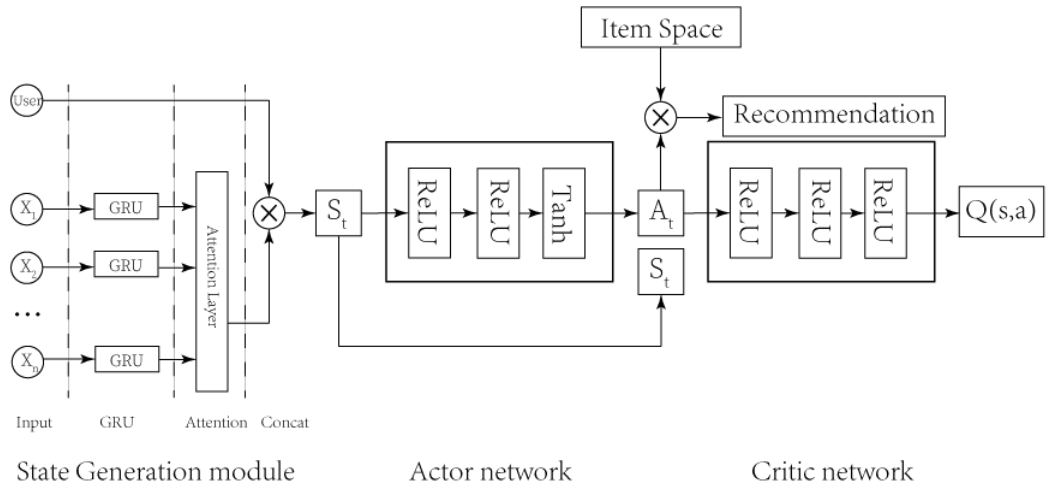


Fig. 3. Overall Network Structure

$$y_i = r_i + \gamma Q_{\omega'}(s_{i+1}, \pi_{\theta'}(s_{i+1})) \tag{9}$$

In equation (8), $Q_\omega(s_i, a_i)$ is the critical network's evaluation of the user's ith recommendation. N is the batch size. In equation (9), r_i indicates the user's rating of the project. The parameters of the actor network and the critic network are θ' and ω' respectively.

2) Online Mode: The online mode is divided into two parts: training and simulation recommendation process, where the training process is also pre trained. In the process of pre trained, the network parameters are trained by the user's previous data information, and the network cannot be updated in real time. Thus, for online mode, the trained network parameters are directly used to simulate the real-time recommendation process, which generates the first recommendation for the user. The online mode records several steps of recommendation records and user feedback, which is defined as (s_t, a_t, s_{t+1}, r_t). We store them in memory at each step, and then these information can be used to update network parameters synchronously after n-step recommendation, which can enable the model to capture the users current interest and change, and make more accurate recommendations.

IV. Experiments Study

In this section, we conducted three experiments to evaluate the effectiveness of the proposed DRR-GRU model. First, we used Book Crossing and Amazon-b two public datasets to verify the DRR-GRU model. Second, we compared the DRR-GRU model with other model based on deep reinforcement learning on MovieLens-1M and Jester datasets. Finally, we tested the influence of different characteristic matrix dimensions.

A. Environment and Parameters settings

All the algorithms proposed in this paper were coded in Python 3.8 and Torch 1.9.1. All experiments were run on a personal computer, whose processor is Intel i7-10875H CPU and RTX 3060 GPU.

For the datasets, we first processed the rating data in advance and make its values be uniformly normalized to a number between -1 and 1. Then, we randomly generated user's feature matrices with 200 dimensions and projects by PMF [21] before training. The items that are positively evaluated by users are also deleted from the candidate set. The parameters settings of the model is described in the following. The learning rates of the actor and critic networks are set to 10^{-4} and 10^{-3} respectively. The discount rate γ is 0.9, and N is 64. In addition, the model adopts the Adam optimizer and L2 paradigm regularization.

TABLE I
Precision Comparison on Book-Crossing dataset

Model	Recommended List Length				
	L=5	L=10	L=15	L=20	L=25
HBSADE	0.283	0.253	0.256	0.237	0.232
DRR-Max	0.413	0.356	0.321	0.281	0.264
DRR-GRU	0.414	0.362	0.316	0.278	0.263

TABLE II
Precision Comparison on Amazon-b dataset

Model	Recommended List Length				
	L=5	L=10	L=15	L=20	L=25
HBSADE	0.312	0.297	0.272	0.258	0.258
DRR-Max	0.673	0.548	0.504	0.504	0.518
DRR-GRU	0.684	0.553	0.515	0.511	0.522

B. Experiment on Book Crossing and Amazon-b Datasets

We adopt Precision, Recall and F1 three indexes to evaluated our proposed model. At the same time, we compare it with some existing recommendation models. The comparison models include PMF [21], CMF, CDL [22], DLMR-DAE [23], HBSADE [24] and our previous proposed model DRR-Max [7]. According to the results in [7], the two best models is HBSADE and our previous model DRR-Max. Therefore, we just compared our proposed model with HBSADE [24] and DRR-Max [7].

1) Results of Offline Mode: The experimental results on Book crossing and Amazon-b datasets are shown from Table

TABLE III
RECALL COMPARISON ON BOOK-CROSSING DATASET

Model	Recommended List Length				
	L=5	L=10	L=15	L=20	L=25
HBSADE	0.327	0.349	0.364	0.382	0.371
DRR-Max	0.257	0.388	0.460	0.481	0.522
DRR-GRU	0.264	0.398	0.454	0.491	0.509

TABLE IV
RECALL COMPARISON ON AMAZON-B DATASET

Model	Recommended List Length				
	L=5	L=10	L=15	L=20	L=25
HBSADE	0.341	0.376	0.382	0.398	0.399
DRR-Max	0.510	0.654	0.698	0.701	0.731
DRR-GRU	0.518	0.656	0.698	0.715	0.731

TABLE V
F1 COMPARISON ON BOOK CROSSING DATASET

Model	Recommended List Length				
	L=5	L=10	L=15	L=20	L=25
HBSADE	0.303	0.293	0.300	0.293	0.285
DRR-Max	0.317	0.371	0.378	0.354	0.350
DRR-GRU	0.321	0.379	0.378	0.366	0.347

TABLE VI
F1 COMPARISON ON AMAZON-B DATASET

Model	Recommended List Length				
	L=5	L=10	L=15	L=20	L=25
HBSADE	0.326	0.331	0.318	0.313	0.313
DRR-Max	0.580	0.596	0.585	0.586	0.606
DRR-GRU	0.589	0.601	0.591	0.586	0.612

I to Table VI. In these tables, L represents the length of recommended list. For the precision values and recall values in Table I to Table IV, we find that the DRR-GRU used in this paper has relatively excellent performance. Our DRR-GRU model is superior to the HBSADE model. Compared with our previous proposed DRR-Max [7], the new DRR-GRU model adopt the similar framework, but the DRR-GRU model outperforms the DRR-Max as a whole. When recommended list length L is 5, 10 and 15, the DRR-GRU model has better performance than the DRR-Max on Book Crossing. When L is 20 and 25, the effect of the DRR-GRU model is basically the same as that of the DRR-Max model. For Amazon-b dataset, the DRR-GRU model is always better than DRR-Max model. For the F1 metric, we can find that the DRR-GRU model is superior to HBSADE model on different recommended list lengths. Meanwhile, the DRR-GRU model has better performance than DRR-Max on two well-known datasets expect for the recommended list of 25 on Book-crossing.

2) Results of Online Mode: This section, we compared the online recommended results of DRR-GRU and DRR-Max models. The same reward function defined in [7] was used. The recommended list length L is also set to 5, 10, 15, 20 and 25. The results of both two models are reported in Table VII and Table VIII respectively. As reported in these two tables, the results show that when the recommendation list has different length, the reward value obtained in two datasets is about 0.5. It can be concluded that the recommended effect is relatively stable. At the same time, the experimental results show that DRR-GRU has better recommendation effect.

TABLE VII
REWARDS OF TWO MODELS FOR BOOK CROSSING DATASET

Model	Recommended List Length				
	L=5	L=10	L=15	L=20	L=25
DRR-Max	0.5045	0.5029	0.5023	0.5017	0.5015
DRR-GRU	0.5077	0.5080	0.5078	0.5081	0.5076

TABLE VIII
REWARDS OF TWO MODELS FOR AMAZON-B DATASET

Model	Recommended List Length				
	L=5	L=10	L=15	L=20	L=25
DRR-Max	0.4983	0.4990	0.4999	0.5010	0.5013
DRR-GRU	0.5106	0.5100	0.5099	0.5104	0.5103

C. Experiment on MovieLens-1M and Jester Datasets

This section, we compare the DRR-GRU model with other two reinforcement learning recommendation algorithms: DR-R [20] and DRR-att [25]. We carried out offline and online experiments on MovieLens-1M and Jester public datasets, where top-N recommendation items are generated, and the recommended list length is set to 20.

For online mode, we still use reward defined in [7] as the evaluation metric. For offline mode, the evaluate metrics include Precision, Normalized Discounted Cumulative Gain (NDCG), and Mean Average Precision (MAP) three metrics. The metrics NDGG is calculated in equation (10).

$$NDCG = \frac{DCG}{IDCG} \tag{10}$$

where DCG and IDCG represent discounted cumulative gain and maximum discounted cumulative gain respectively. The DCG metric is calculated by equation (11).

$$DCG = \sum_{i=1}^{N} \frac{rel_i}{\log(i+1)} \tag{11}$$

In equation (11), DCG indicates the sum of the score of each recommended item divided by its position in the recommended list, where rel_i is the rating of the recommended item. IDCG is the ideal maximum DCG value, which sort the recommendation list by the score descending, and then recalculated the DCG value of the new list.

MAP is calculated by the equation (12), where AP is the average of all the scores of recommended items for Precision. U represents the user list, and $|U|$ is the length of user list.

$$MAP = \frac{\sum_{u \in U} AP_u}{|U|} \tag{12}$$

Table IX and X gives the results on MovieLen-1M and Jester datasets. As shown in Table IX and X, the DRR-GRU can obtained better recommendation effect on two datasets.

TABLE IX
RESULTS ON MOVIELEN-1M DATASET

Algorithm	Precision	NDCG	MAP	Reward
DRR	0.6227	0.8912	0.7238	0.6746
DRR-att	0.6315	0.8946	0.7516	0.6824
DRR-GRU	0.6630	0.9139	0.7925	0.7385

TABLE X
RESULTS ON JESTER DATASET

Algorithm	Precision	NDCG	MAP	Reward
DRR	0.6075	0.8834	0.6850	0.3315
DRR-att	0.6124	0.8889	0.6937	0.3437
DRR-GRU	0.6875	0.9672	0.7466	0.3814

Among three algorithms, DRR has the lowest recommendation accuracy. Take the DRR as the baseline method, the DRR-GRU algorithm has better improvement effect than DRR-att. Overall, the DRR-GRU algorithm is very effective.

D. Performance Analysis of different characteristic matrix dimensions

This section, we select different characteristic matrix dimensions to test the influence to the model. We assume that the dimensions are 50, 100, 150, 200 and 250, and the recommended list lengths are set to 10 and 20 respectively. Taking the first group datasets as an example, the accuracy and recall results of different recommendation list lengths are shown in Fig. 4 and Fig. 5.

It can be seen from Fig.4 that when the recommended list length is 10 and 20, precision achieves the best effect when the feature matrix dimension is 200 and 150 respectively. At the same time, recall gets the best recommendation effect when the feature matrix dimension is 200, which is slightly higher than the feature matrix dimension is 150. For Amazon-b dataset, as shown in Fig.5, precision and recall can both obtain the best results on different recommended lengths of 10 and 20, when the dimension of feature matrix is 200. On the whole, when the feature matrix dimension is 200, the system achieves the best recommendation effect.

V. CONCLUSION

Providing accurate online real-time recommendation service for the users has been attracted the attention of researchers. Most pre-trained recommendation model could not be updated immediately by the real-time changing interests of users. Thus, we proposed a deep reinforcement learning-based recommendation model denoted as DRR-GRU to acquire the changes of user real-time interest by an actor-critic reinforcement algorithm. The proposed model extracted users historical interaction items and user characteristics by a state generation module combining GRU and attention mechanism. In the process of state generation, the users long-term and short-term preferences were also taken into account. The deep network parameters were updated by the users feedback information.

The presented model was trained by offline and online modes and then was evaluated on two groups of real public datasets. According to the experimental results, the DRR-GRU model is superior to existing models. For the Book Crossing and Amzaon-b datasets, the DRR-GRU model outperform the existing models on three evaluation metrics. The further experiment on MovieLens-1M and Jester datasets, the DRR-GRU model can obtain better recommendation effect than other similar models. Additionally, we designed different characteristic matrix dimensions and analyzed the performance influence of them. Form the results, we can find that when the feature matrix dimension is 200, the model can achieve the better recommendation effect.

In future, we will focus on two areas. One is to extract more multi-dimension information between recommendation items and user. The other is to obtain the richer feature information to have more accurate recommendation effect.

REFERENCES

[1] Zheng G J, Zhang F Z, Zheng Z H, Xiang Y, Yuan J, Xie X and Li Z H, "DRN: A deep reinforcement learning framework for news recommendation," in *Proceedings of the 2018 World Wide Web Conference*, pp. 167-176, 2018.

[2] Zhao X Y, Xia L, Zhang L, Ding Z Y, Yin D W and Tang J L,"Deep reinforcement learning for page-wise recommendations," in *Proceedings of the 12th ACM Conference on Recommender Systems*, pp.95-103, 2018.

[3] Zhao X Y, Zhang L, Ding Z Y, Xia L, Tang J L and Yin D W, "Recommendations with negative feedback via pairwise deep reinforcement learning," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1040-1048, 2018.

[4] Linden G, Smith B and York J, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80, 2003.

[5] Chen X S, Li S, Li H, Jiang S H, Qi Y and Song L, "Generative adversarial user model for reinforcement learning based recommendation system," in *International Conference on Machine Learning*, pp. 1052-1061, 2019.

[6] Afsar M, Crump T and Far B, "Reinforcement learning based recommender systems: A survey," *ACM Computing Surveys*, vol. 55, no. 7, pp. 1-38, 2022.

[7] Hou Y E, Gu W B, Dong W C and Dang L X, "A Deep Reinforcement Learning Real-Time Recommendation Model Based on Long and Short-Term Preference," *International Journal of Computational Intelligence Systems*, vol. 16, no.1, 2023. DOI:10.1007/s44196-022-00179-1

[8] Peters J and Schaal S, "Natural actor-critic," *Neurocomputing*, vol. 71, no. 7-9, pp. 1180-1190, 2006.

[9] Schafer J B, Dan F, Herlocker J and Sen S, "Collaborative Filtering Recommender Systems," *The adaptive web: methods and strategies of web personalization*, Berlin, Germany, Spring-Verlag, pp. 291-324, 2007.

[10] Pazzani M J and Billsus D, "Content-based recommendation systems," *The adaptive web: methods and strategies of web personalization*, Berlin, Germany, Spring-Verlag, pp. 325-341, 2007.

[11] Resnick P, Iacovou N, Suchak M, Bergstrom P and Riedl J, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer Supported Cooperative Work*, pp. 175-186, 1994.

[12] Bobadilla J, Ortega F, Hernando A and Alcalá J, "Improving collaborative filtering recommender system results and performance using genetic algorithms," *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1310-1316, 2011.

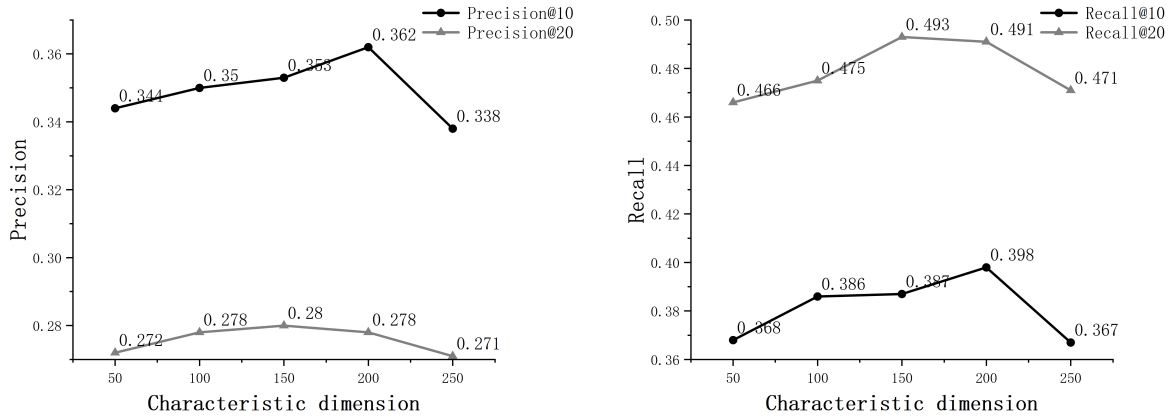
[13] Tian Y H, Zheng B, Wang Y F, Zhang Y and Wu Q, "College library personalized recommendation system based on hybrid recommendation algorithm," *Procedia CIRP*, vol. 83, pp. 490-494, 2019.

[14] Yang F, "A hybrid recommendation algorithm–based intelligent business recommendation system," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 21, no. 6, pp. 1317-1322, 2018.

[15] Zhang S, Yao L N, Sun A X and Tay Y, "Deep learning based recommender system: A survey and new perspectives," *ACM computing surveys (CSUR)*, vol. 52, no. 1, pp. 1-36, 2019.

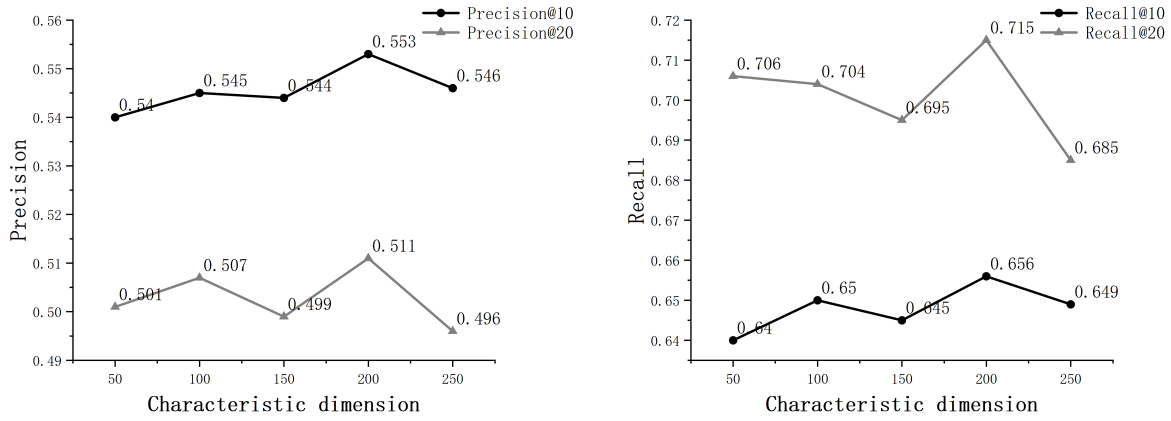
[16] Zhang S, Tay Y, Yao, L N, Sun A X and An J K, "Next item recommendation with self-attentive metric learning," in *Thirty-Third AAAI Conference on Artificial Intelligence*, vol. 9, 2019.

[17] Wu S, Tang Y Y, Zhu Y Q, Wang L, Xie X and Tan T N, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 246-353, 2019.



(a) Precision (b) Recall

Fig. 4. Influence of characteristic matrix dimensions for Book-Crossing dataset



(a) Precision (b) Recall

Fig. 5. Influence of characteristic matrix dimensions for Amazon-b dataset

[18] Fang H, Zhang D N, Shu Y H and Guo G B, "Deep learning for sequential recommendation: Algorithms, Influential Factors, and Evaluations," *ACM Transactions on Information Systems*, vol. 39, no. 1, pp. 1-42, 2020.

[19] Huang L W, Fu M S, Li F, Qu H, Liu Y J and Chen W Y, "A deep reinforcement learning based long-term recommender system," *Knowledge-Based Systems*, vol. 213, 106706, 2021.

[20] Liu F, Tang R M, Li X T, Zhang W N, Ye Y M, Chen H K, Guo H F and Zhang Y Z, "Deep reinforcement learning based recommendation with explicit user-item interactions modeling," *arXiv preprint arXiv:1810.12027*, 2018.

[21] Salakhudinov R and Mnih A, "Probabilistic matrix factorization," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 1257-1264, 2007.

[22] Wang H, Wang N Y, Yeung D Y, "Collaborative deep learning for recommender systems, " in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1235-1244, 2015.

[23] Zhou W, Li J P, Zhang M L, Wang Y Z and Shah F, "Deep Learning Modeling for Top-N Recommendation With Interests Exploring," *IEEE Access*, vol. 6, pp. 51440-51455.

[24] Sivaramakrishnan N, Subramaniyaswamy V, Viloria A, Vijayakumar V and Senthilselvan N, "A deep learning-based hybrid model for recommendation generation and ranking, " *Neural Computing and Applications*, vol. 33, no. 17, pp. 10719-10736, 2021.

[25] Liu F, Guo H F, Li X T, Tang R M, Ye Y M and He X Q, " End-to-end deep reinforcement learning based recommendation with supervised embedding, " in *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 384-392, 2020.