A REVIEW OF RECOMMENDATION SYSTEMS BASED ON REINFORCEMENT LEARNING

Ziyi Guo,   MSc Data Science,   zg2u21@sonton.ac.uk

Department of Electronics and Computer Science, University of Southampton

ABSTRACT

Recommendation System is the advanced information technology in the big era, which is designed to seek valuable services and goods for target users from abundant data and deal with overload issues of information, but remain problems of data sparsity, and cold start as well. As an interactive learning technique combining the actor and the environment modelling, Reinforcement Learning performs better on solving tradition recommendation system problems by dynamically user preferences. This paper reviews and comments on the current state of related technologies and provides future suggestions.

1 INTRODUCTION

With the rapid development of the big data, people are exposed to the abundant growing data. Despite of the great information and knowledge in the data, users are hard to obtain truly valuable content from the complex data in the massive information, and thus facing the problem of *information overload*. As an information filtering technology, recommendation system solves the problem of information overload by providing individual content to users, and is an important technology applied in a great number of fields.[1]

Technically, there are mainly two categories of recommendation technologies, namely, collaborative filtering-based recommendation[2] and content-based recommendation[3].The former depends on the relationship of influence between users, while the latter calculates the matching degree mainly based on the content features. Although the wide application of deep learning technology greatly improves the development of recommendation system by relation mining on user and item features, it performs poorly in the circumstances with data sparsity and cold start problems due to the lack of user reference data[4,5].

Reinforcement Learning achieves great progress in the Human-Computer Interaction fields in recent years. Combined with deep learning, it acquires the ability to process large-scale data, and extract the underlying features, so as to achieve specific goals more accurately. Named as Interactive Recommendation Method, Reinforcement Learning-based recommendation models update recommendation policies by interacting with users in real time and obtaining feedback from users, which is more in line with realistic recommendation scenario compared with traditional static models[6]. Moreover, normally normalized to a Markov Decision Process (MDP), reinforcement learning methods naturally model the user behavior sequence as a basic property[7], which can fully depict the sequence characteristics and capture the dynamic preference of the user. Also, the setting of the exploration mechanism can make the agent explore the state and the action space more, improving the recommendation result diversity in some way. Finally, since such model usually updates the recommendation policy based on an optimization objective of maximizing the cumulative recommendation scores, namely long-term feedback from users, it can improve the long-term user satisfaction[8].

As a focus of research in recent years, this paper starts from the baseline techniques of RL-based recommendation system, emphasizes the DRL applications in such fields, comments on the state-of-art technology and open problems of RL-based recommendation system and suggests future works.

2 RELATED WORK

According to current study and applications of recommendation system based on reinforcement learning, this paper states the overall circumstance of the development of relevant technologies.

Above all, through the analysis of reinforcement learning techniques, it can improve the recommendation system from the aspects as follow.

- **Real-time acquisition of dynamic user preferences.** Recommendation based on reinforcement learning is an interactive recommendation method. Interactive Recommender Systems (IRS) play an important role in personalisation [6]. Traditional methods are static and do not fit the scenario of dynamic interaction with the user in recommendation. In contrast to traditional static recommendation methods, interactive recommendations receive feedback from users after recommending a product to them and then adjust the recommendation strategy in response to the user's feedback at that moment. Therefore, the recommendation strategy is adjusted according to the user's real-time feedback.
- **Capturing the correlation between recommendation items.** As with conversational recommendations, the reinforcement learning approach captures the relationship between items in a sequence by modelling the user's click sequence. Other traditional recommendation methods do not consider sequence relationships and do not capture the dynamic preference changes of users well[9]. A user's interests are dynamic and change over time and with age, which is reflected in the user's click sequence. For example, a user initially likes entertainment news, but over time slowly becomes more inclined to view political news.
- **Exploration mechanism to avoid duplicate recommendations.** Traditional recommendation methods recommend a large number of repeated similar items to users, which greatly reduces their utility[9]. Traditional recommendation methods recommend a large number of similar items by mining the user's historical preferences. The exploration mechanism in the reinforcement learning method can cleverly avoid the problem of recommending a large number of repetitive items to the user, bringing surprise to the user and improving the accuracy of the recommendation[10].
- **Focus on long-term user satisfaction.** Traditional recommendation methods aim to improve users' immediate satisfaction, such as click-through rate, while ignoring long-term satisfaction (e.g., the length of time users continue to use)[9]. Reinforcement learning methods dynamically update policies by maximizing the discounted sum of immediate and future rewards, with the goal of improving long-term user satisfaction[7]. Therefore, reinforcement learning methods can improve long-term satisfaction metrics such as user retention and length of time spent in a single session.

At present, reinforcement learning-based recommendations relies on the extension of standard reinforcement learning models to applications. In general, this paper classifies the methods them into the traditional reinforcement learning recommendations and the recommendations based on deep reinforcement learning. Traditional reinforcement learning can be subdivided into two categories, namely recommendations based on Multi-armed Bandit and recommendations based on Markov Decision Process while recommendations based on deep reinforcement learning can be subdivided into value-based DRL and policy gradient-based DRL.

3 Traditional Reinforcement Learning-based Recommendations

3.1 Multi-aremed Bandit-based Recommendations

The MAB-based approach uses a variety of Bandit algorithms to make recommendations. The starting point of this approach is to balance the exploration-exploitation relationship, not only by recommending products similar to those previously preferred by the user, but also by innovatively exploring other preferences of the user to avoid repetitive recommendations. This will allow the user to not only recommend products similar to their previous favourites, but also innovatively explore their other preferences to avoid repetitive recommendations.

- **Improve the existing Bandit algorithm for recommendations.** Intayoad et al[11] propose a relevance-sensitive contextual Bandit algorithm to solve the online course recommendation problem. In contrast to the novelty of the recommendation list, it takes into account the common repetitive learning behaviour of users, models the Past Student Behavior (PSB) and Current Student State (CSS) in pairs, and builds a correlation matrix to quantify the correlation between candidate actions, in order to maximize the cumulative number of user clicks as an optimization, thus improving the user viscosity of the online learning platform.
- **Introduce a deep neural memory module to reduce human-computer interaction.** Shen et al[12] consider that existing reinforcement learning solutions require a large number of interactions with each user in order to provide high-quality personalised recommendations. To alleviate this limitation, they design a new deep neural memory enhancement mechanism that models and tracks the historical state of each user based on their previous interactions. As a result, user preferences for new items can be quickly understood through a small number of interactions.

3.2 Markov Decision Process-based Recommendations

Markov decision process (MDP)-based reinforcement learning is an early area of research, and for problems with small or discrete state and action spaces, Markov decision modelling can reduce the time complexity of recommendations, and so there is still more research on MDP-based recommendations.

- **Recommendation problems with small state action spaces.** Zhang et al[13] applied the multi-i agent RL method to the dynamic collaborator recommendation problem of scholars, where the similarity between two scholars is calculated based on multiple similarity measures.
- **Reducing the system state action space to improve scalability.** Most recommendation contexts have a large action state space, and when applying Markov decision processes to recommendations alone, the number of learned strategies is limited and scalability is low. De et al[14] used a new belief space sampling algorithm to limit the size of the state space by limiting regret in a multi-intelligence constrained partially observable decision problem, taking into account the impact of recommendations on the available capacity. By exploiting the smooth structure of the problem, this algorithm is more scalable than existing approximate solvers.

4 Deep Reinforcement Learning-based Recommendations

Deep reinforcement learning is a combination of deep learning and reinforcement learning, and thanks to the rapid development of deep learning in recent years, deep reinforcement learning-based recommendations have also become a focus of research.

4.1 Recommendations based on Value-based Deep Reinforcement Learning

Basically, value-based DRL recommendations use deep neural networks to approximate a Q-value function with the optimisation objective of maximising the total reward, and continuously update the neural network parameters through gradient descent to find the optimal policy (Fig 1).

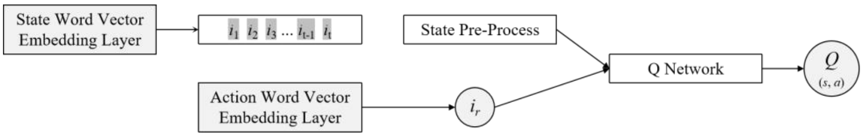

State Word Vector Embedding Layer
i_1 i_2 i_3 ... i_{t-1} i_t
State Pre-Process
Q Network
Q (s, a)
Action Word Vector Embedding Layer
i_r

Fig 1. Value-based DRL Recommendation Framework Baseline

- **Dueling-DQN for dynamic user recommendations.** In some recommendation contexts, users' action choices are only related to the current state, so based on the NatureDQN model, Dueling-DQN[15] divides the Q-network into two parts, calculating the state value function $V(S)$ and the state-dependent action dominance function $A(s,a)$, respectively, to explore the impact of pure state changes on users' decisions. The DRN proposed by Zheng et al[9] uses a Dueling-Double-DQN network structure to capture the dynamics of users' news preferences as news changes. The state value function is used to extract rewards that are determined by state alone. They use user features and contextual features to represent the current state, and news features and news user interaction features to represent a current action. These features are used in the model to output a predicted Q-value for the current state to take this action (Fig 2).

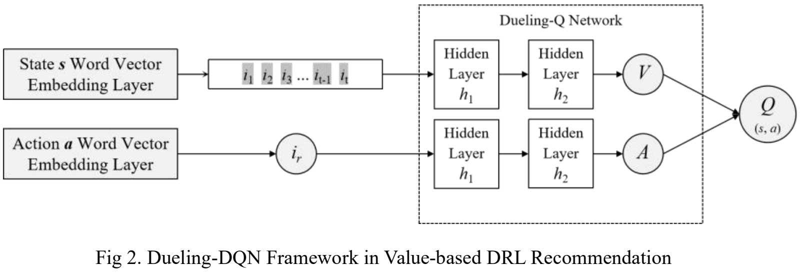
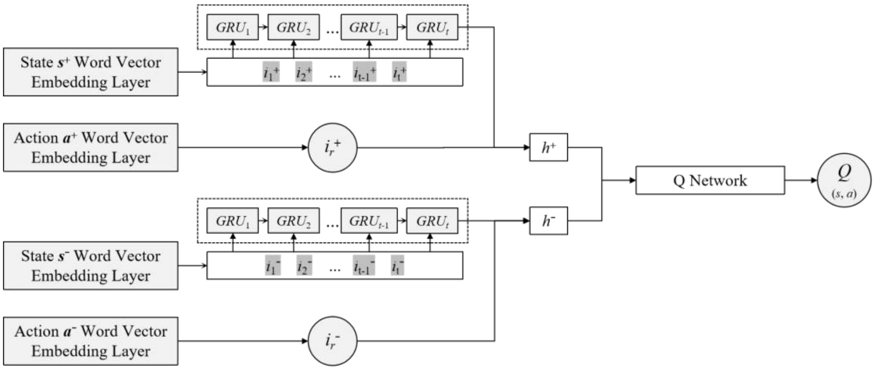

Dueling-Q Network
State ***s*** Word Vector Embedding Layer
i_1 i_2 i_3 ... i_{t-1} i_t
Hidden Layer h_1
Hidden Layer h_2
V
Q (s, a)
Action ***a*** Word Vector Embedding Layer
i_r
Hidden Layer h_1
Hidden Layer h_2
A

Fig 2. Dueling-DQN Framework in Value-based DRL Recommendation

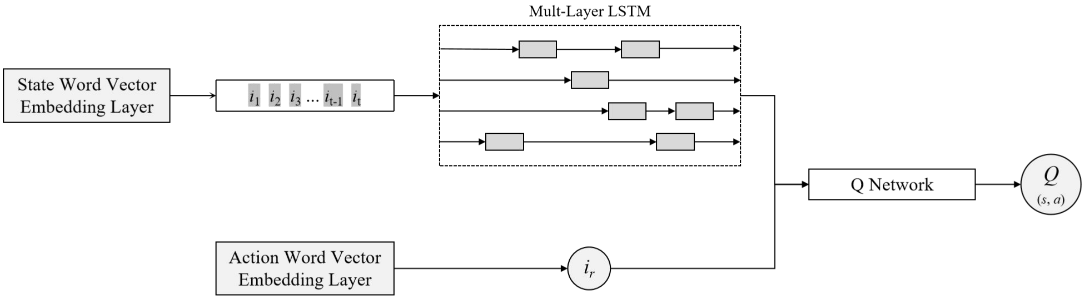
- **DQN considering positive and negative feedback.** The reinforcement learning-based recommendation is about learning the user's feedback to get the optimal policy choice, which includes positive and negative feedback, such as the user ignores or skips the product, indicating the user is not interested in the product. However, previous algorithms have not taken into account negative feedback, and the state remains unchanged when the user skips the item, which can lead to the system continuing to recommend similar items. Therefore, the DEERS model proposed by Zhao et al[8] considers the inclusion of negative feedback as part of the reward in order to avoid recommending products that the user does not like. Both positive and negative feedback update the corresponding state, but the number of negative feedbacks is significantly higher than the positive feedbacks, so it is challenging to combine them for learning. With this challenge in mind, the DEERS recommendation framework is proposed (Fig 3).



GRU_1
GRU_2
...
GRU_{t-1}
GRU_t
State s^+ Word Vector Embedding Layer
i_1^+ i_2^+ ... i_{t-1}^+ i_t^+
Action a^+ Word Vector Embedding Layer
i_r^+
h^+
h^-
Q Network
Q (s, a)
GRU_1
GRU_2
...
GRU_{t-1}
GRU_t
State s^- Word Vector Embedding Layer
i_1^- i_2^- ... i_{t-1}^- i_t^-
Action a^- Word Vector Embedding Layer
i_r^-

Fig 3. DEERS Model Recommendation Framework

➢ **DQN optimizing short and long-term user satisfaction.** Existing recommendation systems generally aim to increase the click-through rate (CTR) of products, without considering the impact of the recommendation results on the long-term satisfaction of users. Therefore, studies have taken into account the synergistic optimization of short and long-term user satisfaction when making recommendations. Zou et al[16] propose a DQN-based FeedRec model, which innovatively optimizes the long-term satisfaction of users in terms of the duration of their browsing and user activity. In order to effectively reflect users' delayed feedback, the Q-Network was redesigned into three layers: Raw Behavior Embedding Layer, Hierarchical Behavior Layer, and Q-Value Layer. The Raw Behavior Embedding Layer is used to input raw user behaviour information to extract the user's state for later optimization. User behaviour includes ignoring, clicking and buying, each of which represents a different meaning. In order to accurately capture the different behaviors, a layered LSTM is applied in the model (Fig 4).



Mult-Layer LSTM
State Word Vector Embedding Layer
i_1 i_2 i_3 ... i_{t-1} i_t
Q Network
Q (s, a)
Action Word Vector Embedding Layer
i_r

Fig 4. FeedRec Model Recommendation Framework

4.2 Recommendations based on Deep Policy Gradient Algorithms

➢ **Optimization for interactive recommendation in large-scale action spaces.** Both DQN and DDPG-based reinforcement learning algorithms need to select an item that maximizes the reward from all items, and the large number of items in a recommendation system results in a high time complexity of this process. Chen et al[17] proposed a tree-based policy gradient model TPGR, in which a balanced hierarchical clustering tree over the items is built to reduce the time complexity of the training and decision stages by decreasing the selection of items to a path from the root of the tree to a particular leaf, and then combined with a policy gradient model to make decisions. In order to train and evaluate the model, they designed an environmental simulator to simulate user behaviour in a standard public dataset, and extensive experiments have shown that the TPGR model works well.

➢ **Model-based deep policy gradient recommendation.** Most of the existing research is model-free, in which the training and learning of strategies requires a large number of frequent interactions between the intelligence and the real environment, which is expensive to learn. Bai et al[18] proposed a model based DRL solution, namely IRecGAN, which models user-intelligence interactions through a generative adversarial network to support offline policy learning. IRecGAN uses a discriminator to assess the quality of the generated data and to scale the generated rewards, thus reducing the bias of the learned user models and strategies.

4.3 Recommendations based on Actor-Critic Deep Reinforcement Learning

Basically, the principle of the recommendation based on Actor-Critic DRL is to use an Actor to train a policy to output an action, a Critic to evaluate the action, and then the Actor to adjust the policy based on the Critic's evaluation (Fig 5).

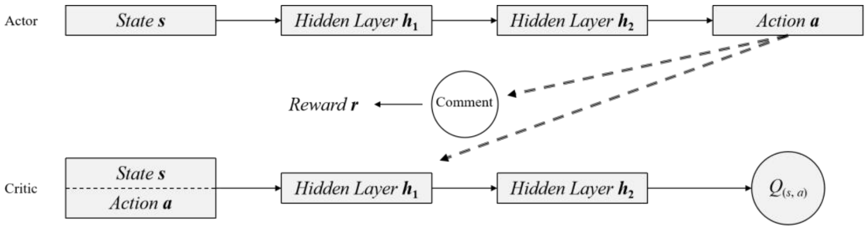


Fig 5. Actor-Critic DRL Recommendation Framework Baseline

➢ **Actor-Critic algorithm-based product list recommendation.** Zhao et al[19] proposed a list-wise recommendation for users of e-commerce platforms, where specifically for items on the list to have complementary relationships, rather than a simple *Top-N* recommendation, which can avoid recommending many similar items. As rewards are difficult to obtain before a recommendation system is rolled out online, an online environment simulator is built to pre-train the parameters. The simulator is based on historical data, which does not contain all states and actions, and the rewards are calculated based on the similarity between unseen state-action pairs p_t and historical state-action pairs m_i. The LIRD model uses the Actor-Critic framework (Fig 6) .

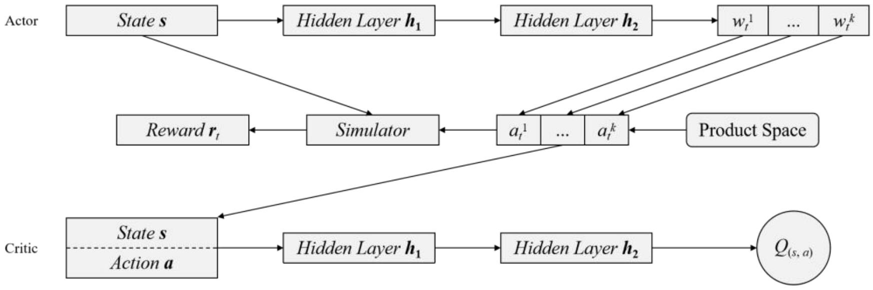


Fig 6. LIRD Model Recommendation Framework

➢ **Actor-Critic algorithm-based 2D shopping list page recommendation.** Zhao et al[20] proposed a page-wise recommendation for users of e-commerce platforms, where DeepPage captures users' visual preferences for the location of products on the 2D page. The Actor network uses an encoder-decoder structure to encode the state space and decode the action space, while Critic uses a DQN for value function approximation (Fig 7).

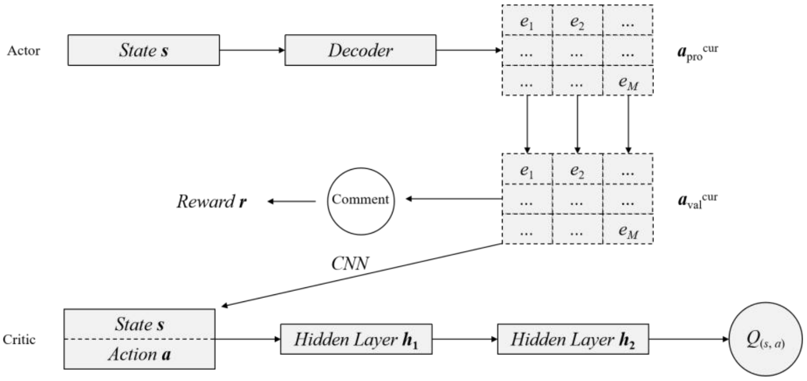


Fig 7. DeePage Model Recommendation Framework

➢ **Actor-Critic reinforcement learning recommendation based on multimodal information.** Simple feedback from users, such as browsing and clicking, often does not fully reflect the user's true preferences. User The textual feedback of users and the visual semantic information about the product can reflect the user's true preferences and visual preferences. However, recommendation systems can easily violate the user's past preference information through textual feedback. The recommendation framework VL-Rec proposed by Yu et al[21] incorporates the visual semantics of products and user review text into the reward function of an intelligent body and effectively merges user preferences over time. Specifically, the Actor uses the minimum Euclidean distance between the visual semantics of the target item, its attributes and the user's preferences as the feature update strategy, and the Critic's reward function and the user's review text ensure that the user's historical preferences are updated sequentially, thus maximizing the desired cumulative future reward.

5 State-of-the-Art Deep Reinforcement Learning-based Recommendations

From the standpoint of this paper, instead of focusing on models themselves, the frontiers of reinforcement learning develop from different perspectives, including hierarchical and multi-intelligent reinforcement learning and the applications in interpretable recommendations and social recommendations.

5.1 Recommendations based on Hierarchical Reinforcement Learning

Hierarchical reinforcement learning methods learn hierarchical strategies by decomposing the final goal into multiple layers of sub-tasks and combining the strategies to form an effective global strategy[7]. In complex recommendation tasks, it is inefficient to optimize policies directly based on the final goal, so a hierarchical approach to decompose complex tasks can effectively improve recommendation efficiency (Fig 8).

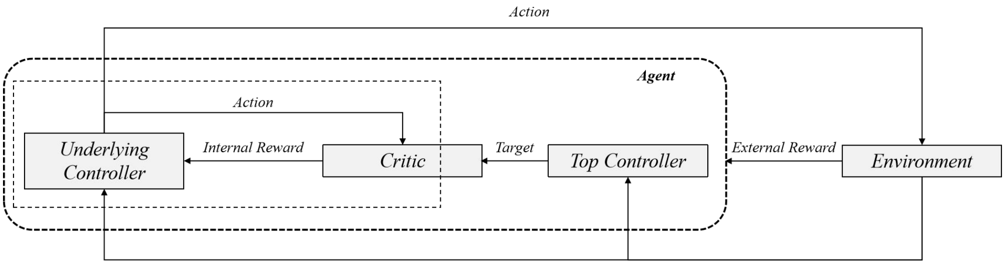


Fig 8. Hierarchical deep Q network Framework[7]

Currently, most recommendation tasks aim to increase the click-through rate of recommended items, but rarely consider the conversion rate of recommended items. In fact, the conversion rate is a better indicator of the real preference of users, but most of the current models do not perform well on the conversion rate because the conversion rate data is extremely sparse and not easy to learn. When feedback is sparse, the intelligence cannot learn effectively because the minimal feedback makes it inadequate in exploring some important state spaces.

A hierarchical DRL algorithm based on spatio-temporal abstraction and intrinsic motivation can reduce the learning complexity by decomposing the overall goal into abstract subgoals, and it can maintain efficient exploration in DRL tasks with sparse feedback problems. Xie et al[22] proposed a hierarchical model for integrated recommendation tasks, in which the high-level intelligence is responsible for recommending items based on the user's fine-grained preferences in a particular channel (e.g., video, graphic). The higher-level intelligence is responsible for recommending items under a specific channel based on the user's fine-grained preferences, while the lower-level intelligence selects channels based on the user's coarse-grained preferences. These hierarchical target settings correspond to a hierarchical reward function, which reduces the learning difficulty of the model by increasing the reward signal.

5.2 Recommendations based on Multi-intelligent Reinforcement Learning

The multi-intelligent reinforcement learning approach is based on the collaborative optimisation of multiple intelligences through policy learning, and is suitable for recommendation tasks that require competitive cooperation, where single-intelligent reinforcement learning approaches are difficult to adapt. For example, in an e-commerce platform, a complete user session is presented with a list of recommendations on the home page, product details page or shopping cart page, which are generated by independent ranking strategies in different scenarios. According to the Cournot Duopoly Model of game theory, the independent optimisation of the ranking strategy may briefly increase the revenue of a given scenario, but may ultimately decrease the total expected revenue of the e-commerce platform. Therefore, it should be discussed how to perform collaborative optimisation to maximise the expected total revenue.

In an e-commerce recommendation task, a user logs in to the e-commerce platform and is first taken to the home page, which contains several recommended products, and is then given the choice of clicking on the product details page or proceeding to the home page, where a new recommended product is available if the user goes to the product details page. Zhao et al[23] proposed the DeepChain model by combining the two recommendation scenarios of home page and product detail page on an e-commerce platform into one recommendation goal, i.e., the behaviour of the customer in one scenario needs to be taken into account in the other scenario. They propose a whole-chain-based recommendation, where recommendations for different scenarios throughout the user's established session take into account the user's historical behaviour in that session. DeepChain is a mode-based reinforced learning approach that does not require a lot of user-intelligence interaction compared to model-free algorithms (Fig 9).

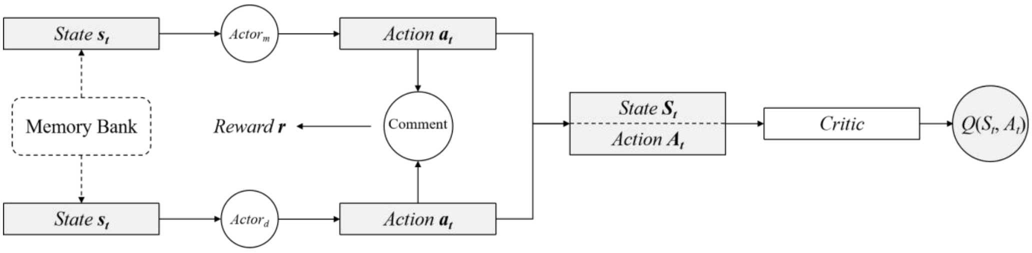


Fig 9. DeepChain Framework[23]

5.3 Interpretable Recommendations based on Reinforcement Learning

As recommendation systems become more and more influential in people's daily lives, it is increasingly important to make users understand and trust the results of the system's recommendations. Explainable recommendations explain the reasons for recommending items to the user, making the user trust the results.

Many recommendation mechanisms are complex and difficult to explain, so there is a need for post-hoc explanation of the recommendation results, i.e. separating the recommendation model from the explanation model, and using a separate model to explain the recommendation results. Xian et al[24] propose an interpretable recommendation method based on knowledge graphs and reinforcement learning reasoning. The knowledge graph contains rich information about users and items, which can provide intuitive and powerful information to support the interpretative problem of recommendation. However, it is difficult to enumerate all paths between user-item node pairs in the knowledge graph for similarity calculation. They therefore apply reinforcement learning methods to interpretative recommendations by training an intelligent body to search for paths. Using the knowledge graph as an environment, the intelligence learns a strategy to navigate from the user to potential items of interest during the training phase. If the correct item is reached, the intelligence receives a higher reward from the environment. Thus, after the strategy training has converged, the intelligence can directly traverse the correct recommended items without having to enumerate all the paths between the user-item pairs, which provide the explanation for the item recommendations.

5.4 Social Recommendations based on Reinforcement Learning

Users are influenced by the preferences of their friends when choosing products[25]. Therefore, when recommending products to users, it is important to consider not only their own personal preferences but also the influence of people around them on their decisions. Lei et al[25] proposed the social attention DQN model SADQN, which takes into account the preferences of close friends in the user's social network while applying DQN recommendations. The Q-value consists of two components, Qp, a function representing personal preferences, and Qs, a function representing social preferences. The social preferences are calculated from a layer of social attention.

Another thinking attempts to eliminate social influence, namely DRL-based end-to-end recommendations. Basic RL-based recommendations consist of *embedding*, *state representation* and *policy* while the *embedding* is obtained by pre-training and is fixed in the subsequent state representation and policy learning. The relationship between users and items is dynamic, and the fixed embedding vector obtained by pretraining cannot represent users or items well. Therefore, Liu et al[26] proposed an end-to-end RL-based recommendation method, EDRR, which allows the embedding part to be trained in conjunction with the state and policy parts. To avoid the instability of the embedding part during training, they introduced a supervised learning signal and proposed three ways to introduce supervised learning. Experiments show that the EDRR-v3 framework achieves end-to-end recommendations in both the value function-based DRL and the policy gradient-based DRL approaches, and performs most consistently.

6 Synthesis and Analysis

According to current research illustrated above, it is indicated that the reinforcement learning method-based recommendation optimization is largely depending on the state of the art of RL model development and related to the actual requirement of recommendations to some extent; while the core elements of RL are designed in parts corresponding to the environment given by a target recommendation system, specifically from the perspectives of the model element definition, the model training design and the *reward* function design.

- Firstly on model element definition, **news recommendation** model considering contextual features represented by Zheng et al[9] encodes news features, user features, news-user interaction features, and contextual features into DRN model, containing information up to 2065 dimensions in each, in which the user features and contextual features represent the current state while the news and interaction features represent an action. **Product list recommendation** model represented by Zhao et al[19] in a LIRD architecture defined action as the recommended list while the state transfer is defined as eliminating the most recently clicked items. **Long-term user satisfaction** based model represented by Zou et al[16] namely FeedRec defines the state as user attributes initially and updates according to the user's heterogeneous behavior with a hierarchical LSTM in the Q-network.

Based on deep learning methods, such as RNN, LSTM, CNN and etc, to extract low-dimensional sequence relationships, the learning effect of the model affected by the precise definition and input processing has always been a problem.

- Secondly on the model training design, **dual network structure** represented by Zhao et al[19] in the LIRD framework of an Actor-Critic structure uses the DDPG algorithm in training and a simulator is pre-trained offline to simulate user behavior to reduce exploration cost. **Environmental simulator-assisted training** represented by Zou et al[16] designs the S-network to simulate user feedback in a real environment assisting off-policy policy learning to avoid blind trial-and-error of the online model and user satisfaction reduction. **User model-assisted training** represented by Zhao et al[23] in the DeepChain framework uses an Actor-Critic structure and the same DDPG for training while it develops a probability network based on supervised learning to predict user behavior and state transitions, namely model-based approximation.

  The DRL models need data in independent and uniform distributions to train neural networks but Markovian and correlated data to train reinforcement learning, which can be quite problematic causing instability in model convergence and make it hard for training design.

- Thirdly on the *reward* function design, the *reward* function **considering negative feedback** represented by Zhao et al[8] is proposed as the DEERS framework, which splits the positive and negative feedback as the input to the DQN and updates state correspondingly, recommending user favorite and avoiding offense. The *reward* functions **considering long-term satisfaction** represented by Zheng et al[9] and Zou et al[16] respectively use either DQN to capture dynamic news properties and user activity in addition to the *reward* or FeedRec Q-network containing LSTM to store a new Delayed Feedback Matrix together with Timely Feedback Matrix and acquire better user preference inferences.

  As a major element of the MDP, the definition of the *reward* function plays a crucial role in the effectiveness of the algorithm. Unlike the static environment of AlphaGo training, the environment of a recommendation system is constantly dynamic and could cause great issues in the *reward* definition and lead to errors in *reward* estimation. Moreover, the varying behavior criteria of different users make them hard to fit the same function and it is difficult to realize individual definition for each, which leads to *reward* error as well.

Despite of a great number of issues to apply reinforcement learning to modern recommendations systems, researchers managed to handle the implementation in different scenarios and achieve considerable results. For instance, the reinforcement-based recommendations contributes a lot to the E-commerce Sector to improve user satisfaction[3,8,16,19,20] and the News sector to improve recommendation timeliness[9,12]. Works also improve the development of Medical, Music and Tourism[27,28,29]. Reinforcement learning-based recommendation is constantly improving with the expansion of social demand.

7 CONCLUSION

According to the review of this paper, it can be concluded that the emergence of big data, artificial intelligence and deep learning technologies in recent years has enhanced human-computer interaction and provided important source data and technical support for the application of reinforcement learning based recommendation systems. The future research directions suggested with the review are as follows. In the future, the research and application of reinforcement learning in recommendation systems will be mainly in the following areas.

- **Design Large-scale Action Space Architecture** for RL recommendations.

  The recommendation environment is huge compared to other environments with simple movements, and finding the products that users like from a large amount of data requires multiple searches by an intelligent body, which makes model training difficult to converge and poor in stability[17]. It is also impossible for the recommendation system to obtain the user trajectory in real time and update the policy timely, which causes update biased using historical user trajectory. Therefore, it is important to explore new reinforcement learning models for recommendation systems with large action spaces. In addition, deep reinforcement learning methods based on model-free learning often require a large number of online interactions to train recommendation policies through online user feedback, which can be costly in terms of interaction and affects the user experience. In contrast, model-based approaches require no large interaction[23], but research in this area is still in its infancy and more in-depth research is needed.

- **Construct Simulators of Recommendation Environment** to reduce training costs.

  Due to the requirements of realistic interactive environment of reinforcement learning methods, consideration are obliged to take into account on the large action space, the large amount of data, the high requirements for computer hardware, and the high user costs for online training of deep reinforcement learning[16]. Training an immature strategy online may result in recommending many uninteresting products to users, leading to higher churn rates. Using an environmental simulator for offline training would be much less costly and researchers have already used their own recommendation environment simulators to support training evaluation[16]. Despite of the current works, this direction still deserves more attention.

- **Design Suitable Representation of *Action* and *Reward*** with deep learning techniques.

  In reinforcement learning, the researcher needs to define each element of the MDP, namely the *action*, the *state*, and the *reward*. As one of the input features, the construction of state is crucial for model training. Most of the current research defines state using recurrent neural networks or capturing sequential relations such as attentional models or memory networks. The definition of the reward function is also an important part of model training as a direction for policy updating, which is often designed manually and cannot reflect the different user satisfaction levels (e.g., 5 for purchase, 3 for add to cart, 0 for ignore, etc.), capturing inaccurate user preferences. Therefore, the use of deep learning techniques to obtain deeper representations of *state*, *reward* and other features is an area of interest for future research.

- **Establish Reasonable Interactive Recommendation Evaluation Mechanism.**

  At present, most of the evaluation metrics used to evaluate reinforcement learning recommendation models are common metrics, such as CTR, accuracy rate, recall rate, NDCG, MAP, etc. [30], which do not accurately reflect user satisfaction with the whole interaction process and recommendation results for interactive recommendation systems. Some researchers use several new evaluation criteria on recommendation performance[16]: the average number of user clicks per session, the average browsing depth of each session, and the average return time of the same user. These evaluation metrics take into account not only the short-term satisfaction of the users during the interaction, but also their long-term satisfaction. For example, the users' satisfaction is greatly reflected by their behavior like adding to cart and purchasing in the case of E-commerce Recommendation Systems while is reflected by the length of time listening to a song instead of clicking experience in the case of Music Recommendation Systems. Therefore, in the context of dynamic recommendation environments, there could be more focus on the construction of new evaluation metrics reflecting user satisfaction.

Reference

[1] MARZ N, WARREN J .Big Data: Principles and best practices of scalable realtime data systems [M]. USA: Manning, 2015: 44-49.

[2] KOREN Y, BELL R ,VOLINSKY C. Matrix factorization techniques for recommender systems [J]. 30-37.

[3] BOBADILLA J, ORTEGA F, HERNANDO A, et al. Recommender systems survey [J]. Knowledge Based Systems, 2013, 46: 109-132.

[4] HUANG L W, JIANG B T, LV S Y, et al. Survey on deep learning based recommender systems[J]. 1619-1647.

[5] BATMAZ Z, YUREKLI A, BILGE A, et al. A review on deep learning for recommender systems:challenges and remedies[J]. Artificial Intelligence Review, 2019, 52(1). 1-37.

[6] ZHAO X X, ZHANG W N, WANG J. Interactive Collaborative Filtering [C] // Proceedings of the 22nd ACM International Con- ference on Information & Knowledge Management. ACM Press, 2013. 1411-1420.

[7] LIU Q, ZHAI J W, ZHANG Z Z, et al. A Survey on Deep Reinforcement Learning[J]. Chinese Journal of Computers, 2018, 41(1). 1-27.

[8] ZHAO X Y, ZHANG L, DING Z Y, et al. Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning [C] // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018: 1040-1048.

[9] ZHENG G J, ZHANG F Z, ZHENG Z H, et al. DRN: A Deep Reinforcement Learning Framework for News Recommendation [C] // Proceedings of 2018 World Wide Web Conference. 167-176.

[10] SHANI G, GUNAWARDANA A. Evaluating recommendation systems[M] // Recommender Systems Handbook. Boston: Springer, 2011: 257-297.

[11] INTAYOAD W, KAMYOD C, TEMDEE P. Reinforcement Learning Based on Contextual Bandits for Personalized Online Learning Recommendation Systems[J]. Wireless Personal Communications, 2020 (115): 2917-2932.

[12] SHEN Y L, DENG Y, RAY A, et al. Interactive recommendation via deep neural memory augmented contextual bandits[C] // Proceedings of the 12th ACM Conference on Recommender Systems. 2018: 122-130..

[13] ZHANG Y, ZHANG C W, LIU X Z. Dynamic Scholarly Collaborator Recommendation via Competitive Multi Agent Reinforcement Learning [C] // Proceedings of the Eleventh ACM Conference on Recommender Systems. 331-335.

[14] DE N F, THEOCHAROUS G, VLASSIS N, et al. Capacity-aware Sequential Recommendations [C] // Proceedings of 17th International Conference on Autonomous Agents and Multiagent Systems. 2018: 416-424.

[15] WANG Z Y, SCHAUL T, HESSEL M, et al. Dueling Network Architectures for Deep Reinforcement Learning [C] // International Conference on Machine Learning. 2016: 1995-2003.

[16] ZOU L X, XIA L, DING Z Y, 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.

[17] CHEN H K, DAI X Y, CAI H, et al. Large-scale interactive recommendation with tree-structured policy gradient[C] // Proceedings of the AAAI Conference on Artificial Intelligence. 2019: 3312-3320.

[18] BAI X Y, GUAN J, WANG H N. A model based reinforcement learning with adversarial training for online recommendation[C] // Proceedings of the 33rd Conference on Neural Information Processing Systems. 1-12.

[19] ZHAO X Y, XIA L, ZHANG L, et al. Deep Reinforcement Learning for List-wise Recommendations [J]. arXiv: 1801.00209, 2017.

[20] ZHAO X Y, 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.

[21] YU T, SHEN Y L, ZHANG R Y, et al. Vision-Language Recommendation via Attribute Augmented Multimodal Reinforcement Learning [C] // Proceedings of 27th ACM International Conference on Multimedia. 2019: 39-47.

[22] XIE R B, ZHANG S L, WANG R, et al. Hierarchical Reinforcement Learning for Integrated Recommendation[C] // Proceedings of the 35th AAAI Conference on Artificial Intelligence. 2021: 1-8.

[23] ZHAO X Y, XIA L, ZHANG L, et al. Model Based Reinforcement Learning for Whole-Chain Recommendations [J]. arXiv: 1902.

[24] XIAN Y K, FU Z H, MUTHUKRISHNAN S. Reinforcement knowledge graph reasoning for explainable recommendation [C] // Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019. 285-294.

[25] LEI Y, WANG Z T, LI W J. Social attentive deep q-network for recommendation[C] // Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019: 1189-1192.

[26] LIU F, GUO H F, LI X T, 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.

[27] WANG L, ZHANG W, HE X F. Supervised Reinforcement Learning with Recurrent Neural Network for Dynamic Treatment Recommendation [C] // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2447-2456.

[28] Liebman E, Saar T M, Stone P. DJ-MC: A Reinforcement-Learning Agent for Music Playlist Recommendation [C] // Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems.2015: 591-599.

[29] MASSIMO D, ELAHI M, RICCI F. Learning User Preferences by Observing User-Items Interactions in an IoT Augmented Space [C] // Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization. 2017:35-40.

[30] ZHU Y X, LV L Y. Evaluation Metrics for Recommender Systems[J]. Journal of University of Electronic Science and Technology of China, 2012, 41(2): 163-176.