DEN-DQL: Quick Convergent Deep Q-Learning with Double Exploration Networks for News Recommendation

1st Zhanghan Song

School of Cyber Science and Engineering, Wuhan University
Wuhan, China
songzh@whu.edu.cn

2nd Dian Zhang

School of Cyber Science and Engineering, Wuhan University
Wuhan, China
zhangdian@whu.edu.cn

3rd Xiaochuan Shi

School of Cyber Science and Engineering, Wuhan University
Wuhan, China
shixiaochuan@whu.edu.cn

4th Wei Li

School of Artificial Intelligence and Computer Science
Jiangnan University
Wuxi, China
cs_weili@jiangnan.edu.cn

5th Chao Ma

School of Cyber Science and Engineering, Wuhan University
Wuhan, China
chaoma@whu.edu.cn

6th Libing Wu

School of Cyber Science and Engineering, Wuhan University
Wuhan, China
wu@whu.edu.cn

Abstract—Due to the dynamic characteristics of news and user preferences, personalized recommendation is a challenging problem. Traditional recommendation methods simply focus on current reward, which just recommend items to maximize the number of current clicks. And this may reduce users' interest in similar items. Although the news recommendation framework based on deep reinforcement learning preciously proposed (i.e., DRL, based on deep Q-learning) has the advantages of focusing on future total rewards and dynamic interactive recommendation, it has two issues. First, its exploration method is slow to converge, which may bring new users a bad experience. Second, it is hard to train on off-line data set because the reward is difficult to be determined. In order to address the aforementioned issues, we propose a framework named DEN-DQL for news recommendation based on deep Q-learning with double exploration networks. Also, we develop a new method to calculate rewards and use an off-line data set to simulate the online news clicking environment to train DEN-DQL. Then, the well trained DEN-DQL is tested in the online environment of the same data set, which demonstrates at least 10% improvement of the proposed DEN-DQL.

Index Terms—Reinforcement Learning, Deep Q-Learning, News Recommendation, Double Exploration Networks

I. INTRODUCTION

With the continuous progress of information technology, the rapid development and popularization of the Internet, the number of users on the news media has increased sharply. It leads to the problem of information overload in the system, resulting

in users spending a lot of time on news selection. Therefore, as an effective solution, item accurate recommendation has been the focus of scholars.

The key of recommendation task is an efficient recommendation algorithm. The general step of recommendation algorithm is to analyze some behavior records of users on the Internet, extract representative features, design appropriate methods, and finally recommend appropriate tasks. There are three common processing methods of recommendation algorithm, including content-based recommendation methods [1]–[3], collaborative filtering recommendation methods [4]–[6] and hybrid recommendation methods [7]–[9]. Content-based method has strong stability, which makes it difficult to find users' new hobbies, and may lead users to the information cocoons. Collaborative filtering method has the ability to recommend new information [11], but suffers from data sparsity and cold start [12]. Hybrid method combines the above two methods but can't totally address the aforementioned problems.

As an extension of the above methods, deep learning model has become a new mainstream [13]–[15]. But these models face the following challenges in news recommendation. First, the dynamic changes in news recommendations are complicated to track. Second, traditional models are limited to keep recommending similar items to users, which might lead users to the information cocoons and decrease users' interest in similar topics. [10]

So, deep reinforcement learning framework was introduced to news recommendation. Reinforcement learning model uses

Zhanghan Song and Dian Zhang contribute equally to this work. Xiaochuan Shi is the corresponding author of this paper.

agent to interact with users dynamically, so it can achieve better real-time recommendation. At the same time, the reinforcement learning model uses rewards to judge the quality of the recommended content and selects the next action by calculating the $Q(action, state)$ value of the action space ($Q(action, state)$ represents the sum of the current reward and future reward). In this way, the news recommendation model can take future rewards into account. But, the former framework either use $\epsilon - greedy$ [16] strategy or *UpperConfidenceBound(UCB)* [17] to explore. $\epsilon - greedy$ may recommend totally unrelated content, while UCB get a reward after recommend the same content for several times. State-of-the-art reinforcement learning methods [10] use Dueling Bandit Gradient Descent (DBGD) method [22]–[24] which slowly updates the parameters of the exploration network and is slow to converge. So, it is necessary to take a more effective exploration approach.

Further more, it is hard to simulate online environment in reinforcement learning for recommendation, especially how to define reward. Although we can train our model on real apps, this will affect the user experience and the cost is too high. The previous work either train on off-line dataset, or generate data with GAN [34], [35], or directly apply to commercial apps. News recommendation is associated with time series and according to existing studies [36], [37], time series representation is important for the accuracy. If we train reinforcement model on offline dataset instead of an interactive environment, we can not fully take the advantages of our model. To make reinforcement recommendation model training more convenient and effective, a new method to simulate online environment is necessary.

Therefore, in this paper, we propose a deep Q-learning framework (DQN [21]). It has a quick convergent exploration strategy based on double exploration network. Double exploration network has two sets of update networks. Both sets have a exploration network and share a target network, and apply a DBGD method [18]–[20]. In one set, we add a small disturbance to the parameters of target network as the parameter input of another network, so that we can obtain the close neighborhood of the current recommender. In the other set, we use a bigger disturbance to obtain the far neighborhood of the current recommender. We use $\epsilon - greedy$ method to decide which set will be adopted. And the number of ϵ will be dynamic based on the results. This exploration method can avoid recommending unrelated items and converge faster. Then we propose judging the click probability by the title similarities and calculating the reward by the product of click-label and similarity. In this way we can judge whether our recommend list is good or not, so that we can simulate online environment.

Our contribution can be summarized as below:

- We propose a reinforcement learning framework to do news recommendation. This framework applies a DQL structure and can take care of both immediate and future reward. And our framework can be generalized to many other recommendation problems.

- We apply a more effective exploration method which avoid recommending unrelated items and convergences faster with the combination of $\epsilon - greedy$ and DBGD.
- We propose a new way to calculate reward and a new method to simulate online environment.

The rest of the paper is organized as follows. In Section II we present the problem formulation followed by the methodology introduced in Section III. Then, the experimental results are shown in Section IV. And related studies is discussed in Section V. Conclusions are summarized in Section VI.

II. PROBLEM FORMULATION

We have a list of users and a list of news. Each user clicks news and generates a set of clicked records. In the clicked records, if the user clicked the news, the click label is set to 1, otherwise 0. And in the recommendation problem, we assume that user's interests are constantly changing and recommend the similar news will get little clicks in the long term. Also, we assume that if one user clicks news A, the higher the semantic similarity between news A and B, the more probability that the same user will click news B, and if news A is not clicked, the more probability that the user will not click news B. Then we should put forward a model. The model can fully learn changes in user's interests and recommend news to maximize the number of user's clicks in the long term.

Our problem can be formulated as follows.

Given

- A set of users $U = \{u_i\} (1 \leq i \leq I)$.
- A set of news $N = \{n_j\} (1 \leq j \leq J)$.
- A set of clicked records $L_i = \{l_{ij}\}$. If user u_i clicked the news n_j , then $l_{ij} = 1$. Otherwise $l_{ij} = 0$.

Assume

- Users' interests are constantly changing.
- The higher the semantic similarity between titles of two news, the closer probability that the same user will click both news.

Objective

- Design a model M for recommending news to maximize the number of all users' clicks.

III. METHODOLOGY

Personalized news recommendation has a wide application prospect. The traditional methods including content-based recommendation methods [1]–[3], collaborative filtering recommendation methods [4]–[6] and hybrid recommendation methods [7]–[9] have limitations. Content-based methods have is difficult to find users' new hobbies, and easy to lead to the information cocoons room problems. Collaborative filtering methods can recommend new item [11], while suffer from data sparsity and cold start [12]. Hybrid methods integrate

the above two types of methods but can't totally address the aforementioned problems.

Deep learning models have a good performance in news recommendation [13]–[15]. But these models just maximize the current rewards which just try to recommend similar news. In fact, news recommendation is a long-term task, and only maximizing current rewards may just recommend similar news to users and make them boring. So, we should take future reward into account. Based on deep learning, deep reinforcement learning framework was introduced to news recommendation. Reinforcement learning model selects the next action by calculating the $Q(action, state)$ value of the action space. In this way, the news recommendation model can take future rewards into account. And deep reinforcement learning models utilize neural network to calculate the value of Q which can adapt to a large amount of news as action space. However, most of the existing frameworks either use $\epsilon - greedy$ [16] strategy or $UpperConfidenceBound(UCB)$ [17] to explore the environment. $\epsilon - greedy$ may recommend unrelated content, while UCB gets a reward after recommending the same item for multiple times. State-of-the-art reinforcement learning methods [10] use Dueling Bandit Gradient Descent (DBGD) method [22]–[24] to update the weights of the exploration network which is slow to converge.

So we propose DEN-DQL and a new method to simulate online environment to train our model.

In this section, we will first introduce the overall architecture in Section III-A. Then, we will illustrate the simulation of online environment in Section III-B. After that, the deep Q-learning model is covered in Section III-C. Finally, the exploration module is introduced in Section III-D.

A. Overall Architecture

As shown in Figure 1, our model is composed of double explore networks and target network. We use a set of clicked news to train our explore networks. We will use $\epsilon - greedy$ to select one explore network to choose action and update target network based on its feedback. Then, we use target network to recommend a list of news, and use a list of test news (including clicked news and non-clicked news) to calculate the rewards of our recommended list.

1) *DEN-DQL*: DEN-DQL agent will use mini batch to train its networks, so that it can calculate the approximate $Q(state, action)$. Each exploration network set has explore network $\tilde{Q}$ and shares target network Q . If $\tilde{Q}$ gives better recommendation result, the target network will be updated towards $\tilde{Q}$. Otherwise, Q will be kept unchanged. This update can happen after every recommendation impression happens. The two sets have different methods to obtain parameters of $\tilde{Q}$. The two networks both work, but we choose one of the set to recommend by $\epsilon - greedy$ when train our model. And after a certain number of mini batch, agent will compare the two set and update the value of ϵ . Finally, we use target network to recommend a list of news to test.

2) *feedback*: We calculate the similarity of recommend list and user's next behavior, and then calculate the reward. The

Fig. 1. Overall Framework of DEN-DQL.

total reward of our recommend list will be the feedback to agent to update its parameters.

B. Simulation of Online Environment

In order to train our model, we propose a new method to simulate online recommendation environment. The key problem to simulate online environment for recommendation is determining the reward. Here, we use the multiplication with similarity of titles and clicked label as reward. First, the title of the news is the main focus of users, followed by abstracts. So, we use titles to calculate the similarity of recommended news and user's clicked news. Second, we set clicked label to 1 if the news is clicked, and to -1 if not.

When training, all train news are clicked. And the similarity of clicked news and chosen news is the reward. When test, the test news are consist of clicked and non-clicked news. And for each news in recommended list, we multiply the max similarity and clicked label as reward.

To calculate similarity, we need first switch title to embedding [31]. After that, we use the embeddings of two titles to calculate their cosine similarity. Shown as Equation 1, A and B are word embeddings of two titles

$$Sim(A, B) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

C. Deep Q-learning Model

Considering the dynamic feature of news recommendation and the need to estimate future reward, we apply Deep Q-Network (DQN) to model the probability that one user may click on news. Under the setting of reinforcement learning, the probability for a user to click on a piece of news (and future recommended news) and user ratings are the rewards that our agent can get. Therefore, we can model the total reward calculated in Equation 2.

$$Q(s, a) = r_{current} + \gamma r_{future} \quad (2)$$

State s is represented by the news that user viewed, action a is represented by news to recommend, $r_{current}$ represents the rewards(i.e., click/no click and user rating) of our recommend news, r_{future} is the agent's projection of future rewards, and γ is a loss factor to mitigate the impact of r_{future} .

The algorithm of DQL learning is shown as Algorithm 1.

Algorithm 1 DQL learning

```

1: Initialize state action space  $ap$  and update frequency  $uf$ ;
2: Initialize evaluation network  $EN$  and target network  $TN$ ;
3: Initialize replay memory  $RM$ ;
4: for each state  $s$  do
5:   choose action  $a$  via explore method;
6:   take action  $a$  and get next state  $s'$  and reward  $r$ ;
7:   send transition( $s, a, r, s'$ ) to  $RM$ ;
8:    $s \leftarrow s'$ ;
9: end for
10: get a batch of transitions form  $RM$ ;
11: for each transition in batch do
12:   if  $lt = uf$  then  $lt \leftarrow 0$ ;  $TN$  get  $EN$  weights.
13:   end if
14:   calculate  $Q(\tilde{s}, a)$  via  $EN$ ;
15:   calculate next Q value  $Q(s')$  via  $TN$ 
16:   reward of  $(s, a)$ ,  $Q(s, a) \leftarrow Q(\tilde{s}, a) + \gamma * Q(s')$ ;
17:   use  $Q(\tilde{s}, a)$  and  $Q(s, a)$  to update the weights of  $EN$ ;
18: end for

```

D. Exploration Module

The common strategies to do exploration in reinforcement learning are $\epsilon - greedy$ and UCB. $\epsilon - greedy$ randomly recommend new items with a probability of ϵ , while UCB will select new items for many times which may have larger variance. Obviously, these traditional methods may bring a bad user experience at first. The state-of-the-art [10] introduced DBGD, which only have a set of exploration networks and converges slowly. So, we try to combine DBGD and $\epsilon - greedy$ method and propose double exploration networks.

As shown in Figure 2, the agent generates a recommend list L using the network Q and another list $\tilde{L}$ using another network $\tilde{Q}$ from one set of exploration network(the target network of two sets is the same one). Then agent compares the similarity of the two lists and user's clicked list and calculates the reward as feedback. Agent will use these feedback to update its target network. If network $\tilde{Q}$ performs better, set $w = w'$, otherwise do nothing. After fixed recommendation turns, compare the performance of each set and update ϵ .

Through this kind of exploration, the agent can do more effective exploration and converges faster.

IV. EVALUATION

A. Experimental settings

We conduct experiment on a sampled offline dataset MIND [32]. The MIND dataset for news recommendation was collected from anonymized behavior logs of Microsoft News website and was available on the Kaggle. The data randomly

Fig. 2. Detail Design of Exploration Module.

TABLE I
PARAMETER SETTING

Parameter	Setting
epsilon for DQL	0.9
epsilon for DEN-DQL	0.4
epoch	25
Future reward discount γ	0.9,0.6,0.3
Target network update frequency	4
Explore network update frequency	1

sampled 1 million users who had at least 5 news clicks during 6 weeks from October 12 to November 22, 2019. And to make the dataset more suitable for training our model, we only use the users with a lot of behavioral data. The dataset is consist of behavior and news. Behavior include user's news click history and user's click behavior on a list of news(1 for click and 0 for non-click). News include the detailed information of news articles(e.g.,category, title, abstract and so on). The dimension of behavior is 50,000 and the dimension of news is 51,280.

In our experiment, we regard all news as action space where our model select actions. Also, we select a set of data from behavior which has a long list of history to train. And we regard click behavior list as test data to test our recommended list.

The parameters settings are shown in Table IV-A. We have down a set of experiments with three values of γ to show the advantages of our model.

B. Evaluation measure

We train 25 epochs for each history and recommend a list in each epoch after training. We calculate the reward of each recommended list and compare the growth rate of reward. We can judge the convergence of the model by the growth rate and recommended effect through the total reward.

The measure to calculate the reward of recommended list briefly mentioned, and the specific process can be summarized

as Algorithm 2.

Algorithm 2 Reward of recommended list

```

1: Initialize total reward  $tr = 0$ ;
2: for each news  $rn$  in recommended news list do
3:   Initialize  $k = 0$ ;
4:   for each news  $rn_i$  in test news list do
5:     calculate the similarity of  $rn$  and  $tn$ ,  $sim(rn, tn_i)$ ;
6:     if  $sim(rn, tn_i) > sim(rn, tn_{i-1})$  then
7:       index  $k \leftarrow i$ ;
8:     end if
9:   end for
10:   $maxsim = sim(rn, tn_k)$ ;
11:   $flag$  is the click flag of news  $tn_k$ ;
12:  if  $flag = 0$  then  $flag \leftarrow -1$ ;
13:  end if
14:   $tr \leftarrow tr + maxsim * flag$ ;
15: end for

```

To quantitatively demonstrate the convergence rate of different model. We draw the results into curves(cross coordinate is epoch and longitudinal coordinate is the reward of each epoch) and compare the area of the result curves. If two growth curves start and end at close points, the one with bigger area converges faster.

C. Results and Analysis

We compared our model with "DQL + $\epsilon - greedy$ " to show the advantages of the final result. DQL with $\epsilon - greedy$ is a simple model and gets general performance. It is limited to $\epsilon - greedy$ which may recommend random items and lead to low reward. However $\epsilon - greedy$ can help it jump out of local optimum.

We also compared to "DRN"(i.e., DQL + DBGD) to show advantages of quick convergence. As the state-of-the-art, DRN performs better than DQL. It uses not only click labels but also user feature and news feature to recommend news. However, DRN utilizes minor updates to train its network which is slow to converge.

Previous work has demonstrated the advantage of reinforcement learning model for news recommendation [10]. Here we only show our improvement on the basis.

For all compared algorithms, the recommendation list is generated by selecting the items with top-k estimated $Q(state, action)$ values.

The final results are shown as Figure 3. DQL with $\epsilon - greedy$ converges faster at the beginning of the 25 epochs due to its simple exploration method, but it falls into local optimum and takes several epochs to jump out. However, even if it grows again, it still gets the lowest reward at last. Although DRN and DEN-DQL converge slower, they can fetch higher reward at last.

DEN-DQL converges faster than DRN while they gets similar reward. In addition, the convergence rate of DEN-DQL is uneven, and it sometimes shows a jumping growth rate. Instead, the growth rate of DRN has a small fluctuation.

The above phenomenon is due to the different methods of the two model to update networks. DRN adopts minor update. It has two networks, one is the target network and the other one slightly updates every time. If the other one performs better, the target network will be updated to the other on. Otherwise, do nothing. Different from DRN, DEN-DQL has two sets of networks, one set updates slightly and another set updates significantly. And DEN-DQL utilizes $\epsilon - greedy$ to choose one set to update and updates the value of ϵ based on the performance of two sets.

We calculate the area of different model curves of the three values of γ . Shown as Figure 4. The areas of the three models are growing and the area of DEN-DQL is always the biggest one. That means DEN-DQL converges faster than DRN and performs better than DQL with $\epsilon - greedy$. The average total reward(i.e., the area) is shown as Table IV-C. And we calculate that DEN-DQL increases by 10% on the speed of convergence compared with DRN.

TABLE II
TOTAL RESULTS OF EACH MODEL

Model	Average total reward(area)
epsilon DQL	64.81
DRN	74.41
DEN-DQL	81.77

V. RELATED WORK

With the continuous progress of information technology, the rapid development and popularization of the Internet, the popularity of software-defined infrastructure all over the world [33], the number of users on the news media has increased sharply and people are able to acquire information conveniently from the cyber space such as Web. The recommend methods can be roughly divided into two categories – traditional methods and deep learning methods. Traditional methods include content-based recommendation methods [1]–[3], collaborative filtering recommendation methods [4]–[6] and hybrid recommendation methods [7]–[9]. Deep learning method is more complex and deep reinforcement method is based on it.

Content-based method has strong stability, which makes it difficult to find users' new hobbies, and may lead users to the information cocoons. Collaborative filtering method has the ability to recommend new information [11], but it has the problem of data sparsity and cold start [12]. Hybrid method combines the above two methods but still has limitations in accurate recommendation.

As an extension of the above methods, deep reinforcement learning model has come out. But reinforcement learning models need to be trained on online environment. And to improve its performance at first, it needs a huge amount of data to train offline. However, it still needs a lot of time to converge when launch online. What's more, it is hard to simulate online environment in reinforcement learning for recommendation, especially how to define reward. Although we can train our

Fig. 3. Rewards for Each Epoch of Three Epsilon

Fig. 4. Areas for Each Epoch of Three Epsilon

model on real apps, this will affect the user experience and the cost is too high. The previous work either train on offline dataset, or generate data with GAN, or directly apply to commercial apps. And this is inconvenient for model research and experiment.

Therefore, in this paper, we propose a deep Q-learning framework (DQN [21]) and a new method to simulate online environment. The model has a quick convergent exploration strategy based on double exploration network. Double explo-

ration network has two sets of update networks. Both sets have a exploration network and share a target network, and apply a DBGD method [18]–[20]. In one set, we add a small disturbance to the parameters of target network as the parameter input of another network, so that we can obtain the close neighborhood of the current recommender. In the other set, we use a bigger disturbance to obtain the far neighborhood of the current recommender. We use ϵ -greedy method to decide which set will be adopted. And the number of ϵ will

be dynamic based on the results. When simulating the online environment, we propose judging the click probability by the title similarities and calculating the reward by the product of click-label and similarity.

A. news recommendation algorithms

The research of recommendation algorithm was first studied by the GroupLens research group of the University of Minnesota in the United States. They want to make a movie recommendation system called Movielens, so as to realize the personalized recommendation of movies for users. With the development of recommendation algorithm, three common methods were introduced – content-based method, collaborative filtering recommendation method and hybrid method. Content-based method [1]–[3] mainly extracts the user’s own unique characteristics and project information by analyzing the content information of the project that the user has selected or evaluated, and then recommends them according to the matching situation of the two. It does not need the user’s own opinions on a certain project as a reference. Instead, collaborative filtering recommendation method finds out the groups that may have the same characteristics as the target users from a large amount of information, obtains the items that these users are interested in, and then generates recommendations according to the priority. And, hybrid method is the combination of the two. Different from these traditional methods, our deep reinforcement learning method use networks to model user’s behavior and use user’s feedback(include current reward and future rewards) to update networks.

B. Reinforcement learning for news recommendation

Reinforcement learning for recommendation can be modeled as Contextual Multi-Armed Bandit(MAB) problem [25]–[27]. But these models do not perform well in accurate recommendation. Also, some literature try to formulate it as Markov Decision Process(MDP) [28]–[30], which both consider immediate reward and future reward at the same time. But the MDP model restricted by sparse matrix. The state-of-art solves this issue but needs a lot of time and data to converge. Based on the literature, we propose a MDP framework and use a more brave method to explore so that the framework can be more widely adopted.

C. News similarity

Text similarity refers to the similarity between two articles. The basic method of calculating text similarity is to transform the text into embedding matrix [31], and then calculate the ‘distance’ between embeddings. There are many ways to calculate this ‘distance’ mathematically, such as Cosine similarity, N-gram similarity, Jaccard similarity and so on. In our model, we choose cosine similarity.

VI. CONCLUSION

In this paper, a DQL-based reinforcement learning framework is proposed for news recommendation and a new method is applied to simulate online environment for reinforcement

learning. Compared to previous frameworks, our framework can effectively recommend news based on current and future reward and can converge stably and quickly. In addition, we apply an exploration strategy into our framework to reduce the unstable items for recommendation and to explore more effectively. Experiments have demonstrated that our method can improve the convergence speed when training model without harming the accuracy.

For the future work, it will be more meaningful to perfect our simulation of online environment, because simulation will be improved considering many more elements(e.g., in different kinds of news, the relevance of news with similar headlines are different) under online environment.

ACKNOWLEDGMENT

This work is supported by Wuhan University’s teaching research project “The innovation of contents construction and organization format of AI laboratory courses” under the category YB-10.

REFERENCES

- [1] Wouter IJntema, Frank Goossen, Flavius Frasinca, and Frederik Hogenboom. 2010. Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops*. ACM, 16.
- [2] Michal Kompan and Mária Bieliková. 2010. Content-Based News Recommendation.. In *EC-Web*, Vol. 61. Springer, 61–72.
- [3] Owen Phelan, Kevin McCarthy, Mike Bennett, and Barry Smyth. 2011. Terms of a feather: Content-based news recommendation and discovery using twitter. *Advances in Information Retrieval* (2011), 448–459.
- [4] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 271–280.
- [5] Benjamin Marlin and Richard S Zemel. 2004. The multiple multiplicative factor model for collaborative filtering. In *Proceedings of the twenty-first international conference on Machine learning*. ACM, 73.
- [6] Steffen Rendle. 2010. Factorization machines. In *Data Mining (ICDM)*, 2010 IEEE 10th International Conference on. IEEE, 995–1000.
- [7] Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. 2012. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 153–162.
- [8] Lei Li, Dingding Wang, Tao Li, Daniel Knox, and Balaji Padmanabhan. 2011. SCENE: a scalable two-stage personalized news recommendation system. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 125–134.
- [9] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. ACM, 31–40.
- [10] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. DRN: A Deep Reinforcement Learning Framework for News Recommendation.
- [11] Kumar R, Raghavan P, Rajagopalan S, et al. Recommendation systems: A probabilistic analysis. *J Computer & System Sciences*. 2001.
- [12] Gediminas Adomavicius, Alexander Tuzhilin. Expert-Driven Validation of Rule-Based User Models in Personalization Applications[J]. *Data Mining and Knowledge Discovery*. 2001.
- [13] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ipsir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.
- [14] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Dynamic Attention Deep Model for Article Recommendation by Learning Human Editors’ Demonstration. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2051–2059.

- [15] Atsuyoshi Nakamura. 2015. A ucb-like strategy of collaborative filtering. In *Asian Conference on Machine Learning*. 315–329.
- [16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fiedjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.
- [17] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextualbandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.
- [18] Artem Grotov and Maarten de Rijke. 2016. Online learning to rank for information retrieval: SIGIR 2016 tutorial. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 1215– 1218.
- [19] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. 2013. Reusing historical interaction data for faster online learning to rank for IR. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 183–192.
- [20] Yisong Yue and Thorsten Joachims. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 1201–1208.
- [21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.
- [22] Artem Grotov and Maarten de Rijke. 2016. Online learning to rank for information retrieval: SIGIR 2016 tutorial. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 1215–1218.
- [23] Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. 2013. Reusing historical interaction data for faster online learning to rank for IR. In *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 183–192.
- [24] Yisong Yue and Thorsten Joachims. 2009. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 1201–1208.
- [25] Djallel Bouneffouf, Amel Bouzeghoub, and Alda Gançarski. 2012. A contextualbandit algorithm for mobile context-aware recommender system. In *Neural Information Processing*. Springer, 324–331.
- [26] Olivier Chapelle and Lihong Li. 2011. An empirical evaluation of thompson sampling. In *Advances in neural information processing systems*. 2249–2257.
- [27] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextualbandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. ACM, 661–670.
- [28] Zhongqi Lu and Qiang Yang. 2016. Partially Observable Markov Decision Process for Recommender Systems. *arXiv preprint arXiv:1608.07793* (2016).
- [29] Tariq Mahmood and Francesco Ricci. 2007. Learning and adaptivity in interactive recommender systems. In *Proceedings of the ninth international conference on Electronic commerce*. ACM, 75–84.
- [30] Pornthep Rojanavasu, Phaitoon Srinil, and Ouen Pinnern. 2005. New recommendation system using reinforcement learning. *Special Issue of the Intl. J. Computer, the Internet and Management* 13, SP 3 (2005).
- [31] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations* 2013.
- [32] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, Ming Zhou. MIND: A Large-scale Dataset for News Recommendation.
- [33] G. Huang, C. Luo, K. Wu, Y. Ma, Y. Zhang and X. Liu, "Software-Defined Infrastructure for Decentralized Data Lifecycle Governance: Principled Design and Open Challenges," 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 1674-1683, doi: 10.1109/ICDCS.2019.00166.
- [34] Wei Li, Li Fan, Zhenyu Wang, Chao Ma, and Xiaohui Cui. Tackling mode collapse in multi-generator gans with orthogonal vectors. *Pattern Recognition*, 110:107646, 2021.
- [35] Wei Li, Linchuan Xu, Zhixuan Liang, Senzhang Wang, Jiannong Cao, Chao Ma, and Xiaohui Cui. Sketch-then-edit generative adversarial network. *Knowledge-Based Systems*, 203:106102, 2020.
- [36] Ma, C.; Shi, X.; Li, W.; Zhu, W. Edge4TSC: Binary Distribution Tree-Enabled Time Series Classification in Edge Environment. *Sensors* 2020, 20, 1908. <https://doi.org/10.3390/s20071908>
- [37] C. Ma, X. Shi, W. Zhu, W. Li, X. Cui and H. Gui, "An Approach to Time Series Classification Using Binary Distribution Tree," 2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN), 2019, pp. 399-404, doi: 10.1109/MSN48538.2019.00082.