Applied Soft Computing 125 (2022) 109189

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Context-aware reinforcement learning for course recommendation

Yuanguo Lin [a], Fan Lin [a,*], Lvqing Yang [a], Wenhua Zeng [a], Yong Liu [b], Pengcheng Wu [b]

[a] School of Informatics, Xiamen University, Xiamen, China
[b] Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University, Singapore

ARTICLE INFO

Article history:
Received 28 February 2021
Received in revised form 4 April 2022
Accepted 13 June 2022
Available online 24 June 2022

Keywords:
Recommender systems
Reinforcement learning
Policy gradient
Attention mechanism

ABSTRACT

Online course recommendation is an extremely relevant ingredient for the efficiency of e-learning. The current recommendation methods cannot guarantee the effectiveness and accuracy of course recommendation, especially when a user has enrolled in many different courses. Because these methods fail to distinguish the most relevant historical courses, which can contribute to predicting the target course that indeed reflects the user's interests from her sequential learning behaviors. In this paper, we propose a context-aware reinforcement learning method, named Hierarchical and Recurrent Reinforcement Learning (HRRL), to efficiently reconstruct user profiles for course recommendation. The key ingredient of our scheme is the novel interaction between an attention-based recommendation model and a profile reviser with Recurrent Reinforcement Learning (RRL) that exploits temporal context. To this aim, a contextual policy gradient with approximation is proposed for RRL. By employing RRL in hierarchical tasks of revising user profiles, the proposed HRRL model enables reliable convergence in revising policy learning and improves the recommendation accuracy. We demonstrate the effectiveness of our proposed method by experiments on two open online courses datasets. Empirical results show that HRRL significantly outperforms state-of-the-art baselines.

1. Introduction

In recent years, we have witnessed a rapid growth of online courses, with Massive Open Online Courses (MOOCs) becoming the most important platform in e-learning. There have been many MOOCs platforms around the world. For example, the pioneering MOOC platforms (*e.g.*, Coursera and edX) offer millions of students an opportunity to access numerous courses from prestigious universities. Besides, in China, XuetangX becomes one of the largest MOOCs platforms, which provides more than thousands of courses and attracts lots of users to learn these courses [1]. MOOCs not only alleviate the uneven distribution of educational resources, but also provide a convenient opportunity for users' learning activities.

However, with the rapid increase of online courses, MOOCs platforms suffer from information overload. Due to the limitation of knowledge background, users usually fail to find their favorite courses when faced with huge quantities of course information. Currently, the main challenge is how to analyze and drive users' sequential learning behaviors to improve the overall learning efficiency of MOOCs. Course recommendation technology is an effective way to address this challenge [2]. Differing from other recommendation scenarios [3–6], course recommender systems mainly aim to recommend the target courses that users prefer and should learn according to the most relevant courses that they have enrolled in. In other words, the major goal is to improve the recommendation accuracy based on the contributing historical courses.

The problem of course recommendation can be defined as given a sequence of historical courses enrolled by a user before time t, the goal is to recommend the target courses that indeed reflect the user's preference at time $t+1$ [7]. From the user perspective, it is important to construct user profiles for fitting the recommendation model. To this aim, we can adopt neural attentive item similarity (NAIS) model [8] to assess which courses in a user profile are more important for recommending the target course. Alternatively, neural attentive session-based recommendation (NARM) [9] may be employed to model user's sequential behaviors and capture her purpose. However, these attention-based models may not recommend the target course that reflects the user's preference, especially when the user enrolled in many different courses, since the effects of the contributing courses are diluted by other historical courses that are irrelevant to the target course. To address this issue, Hierarchical Reinforcement Learning (HRL) [7] was employed to revise the user profiles. The task of profile reviser is divided into two kinds of sub-tasks, in which a high-level task makes a decision to revise the whole profile while a low-level task removes the irrelevant courses. The HRL model

* Corresponding author.
E-mail addresses: xdlyg@stu.xmu.edu.cn (Y. Lin), iamafan@xmu.edu.cn (F. Lin), lqyang@xmu.edu.cn (L. Yang), whzeng@xmu.edu.cn (W. Zeng), stephenliu@ntu.edu.sg (Y. Liu), pengcheng.wu@ntu.edu.sg (P. Wu).

https://doi.org/10.1016/j.asoc.2022.109189
1568-4946/© 2022 Elsevier B.V. All rights reserved.

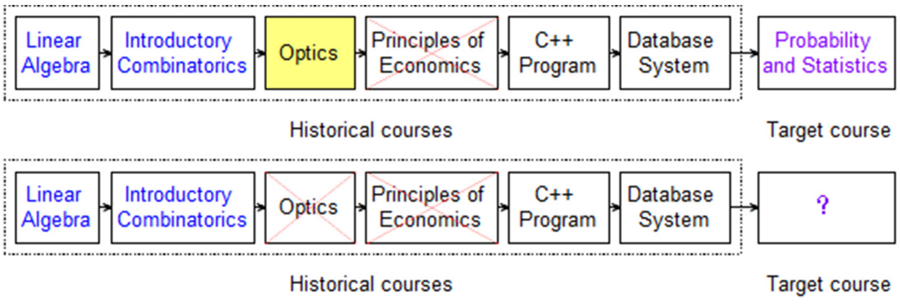


Fig. 1. A example of HRL for course recommendation. The noisy course (*e.g.*, Optics) misleads the model into recommending a target course that a user may not like at next time. We aim at removing more noisy courses to improve the accuracy of recommendations.

jointly trains the profile reviser with a basic recommendation model to improve course recommendation performance.

Although immense progress has been made in the HRL model for course recommendation, there are some details that have not been taken into account. First, the model does not usually remove noisy courses when the high-level task determines not to revise a user profile, even if there might be some irrelevant courses. In other words, high-level task in the model may prevent the proper action and degrade the overall performance by making an inappropriate decision, since the high-level policy is prone to converge to a local optimum. In this case, the whole profile may not be well reconstructed. Second, the model is unlikely to recommend the most relevant courses. The low-level task just randomly removes some noisy courses when there are many irrelevant courses. Since the prediction of the model is unstable, mainly due to the performance of Reinforcement Learning (RL) algorithm (*i.e.*, the randomness of the policy and state transition probability). Thus, it is likely to ignore some more relevant courses when it recommends the target course to a user. As an example illustrated in Fig. 1, due to the noisy course "Optics", the HRL model probably recommends the target course "Probability and Statistics". But intuitively, the user may prefer computer courses, since she has enrolled in the courses "C++ Program" and "Database System" at the last time.

To address the above issues, we propose a Hierarchical and Recurrent Reinforcement Learning (HRRL) model to efficiently reconstruct user profiles for course recommendation. Firstly, a high-level Recurrent Reinforcement Learning (RRL) module is introduced to properly revise user profiles in the high-level task. Afterwards, a low-level RRL module is introduced to iteratively remove the noisy courses and keep the most relevant ones in the low-level task. Finally, we combine the two approaches to enhance the overall performance of the profile reviser. Moreover, a contextual policy gradient with approximation is proposed for RRL. We subtlety synchronize the RRL using temporal context with different episodes of HRL. In this way, the expected update over each episode is in the same direction as the policy gradient, which speeds up the policy convergence per episode in the profile reviser. Therefore, the HRRL model enables reliable convergence in revising policy learning.

In essence, the effectiveness of course recommendation is achieved by a novel interaction between an attention-based recommendation model and the profile reviser. Specifically, the RRL using temporal context appropriately revises user profiles by the policy improvement, which is guided by the contextual policy gradient method with approximation. Thus, the attention-based recommendation model makes accurate recommendations according to the revised profiles, and then provides better policy evaluation for the profile reviser. The profile reviser can assist the attention-based recommendation model, because the RRL using temporal context enhances the representation of user profiles by employing a better policy to take reasonable actions. As a result, by sharing the embeddings of user profiles to jointly train the profile reviser and the recommendation model, the reconstructed user profiles enable the recommendation model to make accurate recommendations.

In summary, we make the following main contributions.

- We propose a context-aware reinforcement learning model (*i.e.*, HRRL) to improve the effectiveness of course recommendation, which is implemented by a novel interaction between the attention-based recommendation model and the profile reviser with RRL using temporal context.
- We propose a contextual policy gradient method with approximation for RRL. In this way, the agent effectively retrieves the contextual rewards of each episode to improve the policy performance. Thus, HRRL efficiently reconstructs user profiles, which can enable the recommendation model to generate accurate recommendation results.
- We conduct empirical experiments on two MOOCs datasets, in which most users enrolled many different courses. Our proposed framework outperforms some competitive baselines by a significant margin across different evaluation metrics.

The rest of this paper is organized as follows. Section 2 briefly reviews related works. Section 3 provides an introduction to policy gradient methods and HRL algorithms. Section 4 introduces our proposed model in details. Experiments and analysis are reported in Section 5. Section 6 concludes our work and put forward the future directions.

2. Related work

2.1. Course recommendation

Recommender systems have obtained great success in many fields, such as e-commerce [5,10], social networks [4,11], location-based service [3,12], and MOOCs [1]. As for course recommendation [13], there are some related methods proposed to handle the issues of different scenarios. For instance, content-based methods [14,15] used Latent Dirichlet Allocation (LDA) to model the topics of courses and employed them to make recommendations for users. Sequence mining-based methods [16,17] were employed to learn the user's course sequence to make recommendations. Based on Collaborative Filtering (CF) approaches, the authors [18] combined a neighborhood-based CF method with matrix factorization to generate personalized course recommendation. And Thanh-Nhan et al. [19] proposed a k-Nearest Neighborhood (kNN) CF method to enhance the matrix factorization method, thereby improving the performance of course recommendation model; From the pairwise learning perspective for course recommendation, a Bayesian Personalized Ranking Network (BPRN) [20] combined item-based CF method to capture pairwise course preferences for users. Moreover, for semi-supervised models, [21] introduced a regression estimator for contextual multi-armed bandits to guarantee the prediction performance. And Parameswaran et al. [13] took into account the course requirements by expressive models, then proposed heuristic techniques to recommend the courses that help satisfy constraints and are desirable for the students.

Recently hybrid methods [22,23] have become a prevailing solution to many complex challenges. For example, Zhu et al. [24] developed a hybrid recommendation framework that employs a graph-structured network for the teaching evaluation to represent multi-source heterogeneous data and adopts a Bayesian probabilistic tensor factorization to make course recommendations. Jing et al. [1] integrated a content-based method into the CF approach to address the cold start problem and the issue of sparsity in course recommendation. Besides, the HRL-based recommendation model [7] was the first to revise user profiles for improving the course recommendation performance.

However, when users are interested in many different courses, the existing methods do not distinguish the most relevant historical courses, which contributes to predicting the target courses that reflect the users' preferences. In other words, these methods fail to effectively model users' profiles for course recommendation. To address this issue, in this paper, we propose a context-aware reinforcement learning method to efficiently reconstruct user profiles. Thus, the proposed method is conducive to improving the effectiveness of course recommendation, based on a novel interaction between the recommendation model and the profile reviser.

2.2. RL-based recommendation

In recent years, RL-based recommendation has become an emerging research topic [25]. RL algorithms have been applied to different recommendation scenarios, such as Point-Of-Interest (POI) recommendation [26,27], sequential recommendation [28,29], interactive recommendation [30–32], conversational recommendation [33,34], diversified recommendation [35,36], and explainable recommendation [37,38]. In general, the RL algorithms can be divided into two main categories: value-based methods and policy-based approaches.

The value-based methods select the action with the maximal Q-value as the best action to learn the optimal policy for recommendation [29,39,40]. For example, Zou et al. [31] developed a novel Q-network to optimize short-term and long-term user engagement in recommender systems. The Q-network can capture versatile information of user behaviors. User-specific Deep Q-learning Network (UDQN) [41] was proposed to learn a global policy for the multi-Markov Decision Process (MDP) task in interactive recommender systems. The value-based methods need to evaluate all the action values under a specific state, which causes excessive sampling when there are a large number of actions.

Differing from value-based methods, the policy-based approaches directly optimize the policy in recommender systems [7,34,37,38,42]. For instance, in the deep learning-based conversational recommender system [33], a deep belief tracker analyzed the user's current utterance to extract a user intention, while a deep policy network guided dialogue management by the user's current utterance and user's preferences learned by a recommender module. The policy parameter was optimized by the REINFORCE algorithm [43], which is a common algorithm in the policy-based approaches. The Tree-structured Policy Gradient Recommendation framework [30] also adopted the REINFORCE algorithm to learn the strategy of making recommendation. The policy-based approaches require a lot of samples to assure policy convergence. On the other hand, one common limitation of these approaches is that they do not consider the temporal context. As a result, it is difficult to well evaluate a policy due to the randomness of the policy, whereas the order in which states have their values updated during the sweep has a significant influence on the policy convergence [44]. Therefore, there is a need to present an effective policy-based approach, which achieves robust convergence properties for RL-based recommender systems. To this end, we propose a contextual policy gradient with approximation to enable reliable convergence in revising policy learning. In this way, our RRL method that exploits the temporal context can revise users' profiles effectively.

3. Preliminary

3.1. Policy gradient method

In this paper, we adopt a parameterized policy, which is able to select actions without directly depending on a value function. The parameterized policy learns the related policy parameters based on the gradient of some performance measure $J(\theta)$, which aims to maximize the policy performance. Thus, its update approximates gradient ascent in $J(\theta)$ as follows [44]:

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)}, \tag{1}$$

where θ_t denotes the policy parameter at time t, α is the step size whose parameter equals $\frac{1}{n}$ in processing the nth reward for action a, and $\widehat{\nabla J(\theta_t)}$ refers to a stochastic estimate, whose expectation approximates the gradient of $J(\theta_t)$ according to its argument θ_t. All the methods that follow Eq. (1) are referred to as policy gradient methods.

Policy gradient methods have been widely applied in many scenarios [45–48]. For example,Wang et al. [37] presented a model-agnostic RL model, which combines doubly stochastic policy gradient with an attention-based neural network to generate sentence explanations. Xian et al. [38] introduced a reinforcement knowledge graph reasoning framework that utilizes a policy gradient method and a knowledge graph for explainable recommendations. The policy gradient method offers practical ways of dealing with large action spaces, since it has better convergence properties than other types of RL. Moreover, the continuity of the policy dependence on the parameters makes policy gradient methods approximate gradient ascent [44]. Hence, we use the policy gradient method, which naturally combines RRL using temporal context to guarantee reliable convergence.

3.2. HRL approaches

RL often suffers from the curse of dimensionality [44]. When the dimension of the system state increases, the number of parameters to be trained will increase exponentially, which consumes a lot of computing and storage resources. The HRL approaches decompose a complex problem into several subproblems, and solve the sub-problems one by one via the method of divide and conquer, to finally solve the complex problem. In general, the HRL approaches consist of the following three categories [49].

The *options* formalism [50] extended the notion of action in MDP to include options, which take actions following closed-loop policies over a period of time. The authors also proposed new intra-option approaches to learn about the options from fragments of their executions. The MAXQ framework [51] decomposed the target MDP into hierarchical MDPs, while decomposing the value function of the target MDP into a combination of the value functions of the hierarchical MDPs. The goal of each sub-MDP is to continuously perform actions from an initial state to a new state until the last state terminates. In Hierarchies of Abstract

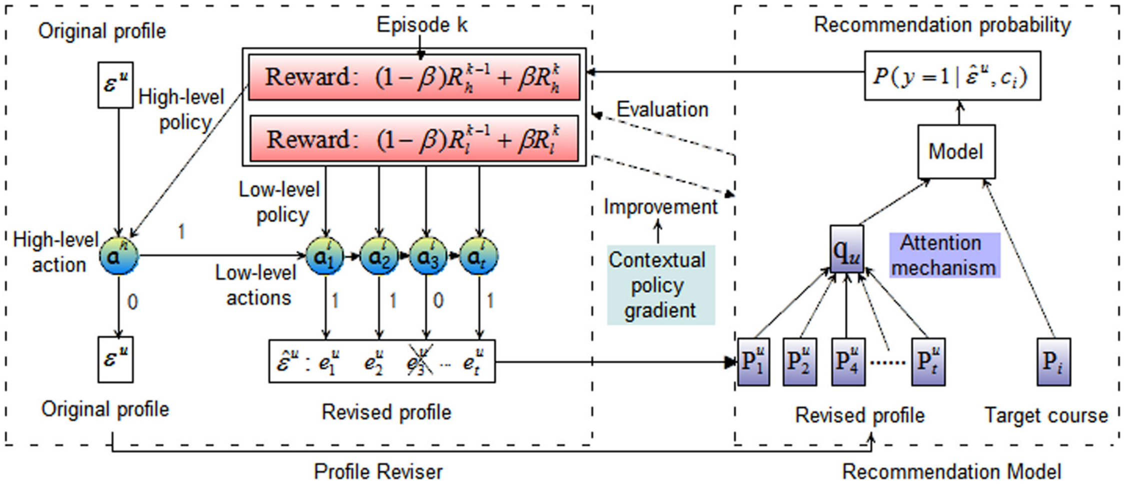


Fig. 2. The overall schematic diagram of our proposed model for course recommendation.

Machines (HAMs) [52], the process of policy learning can be constrained by hierarchies of partially specified machines. Therefore, the search space is reduced by using prior knowledge. The state space of an abstract machine contains four different types: **Action** state executes an action in the environment, **Call** executes the next abstract machine, **Choice** that nondeterministically selects the next state, and **Stop** halts execution of the current abstract machine and returns control to the previous call state [52]. When the abstract machine is in the call state, an agent will urge a high-level machine to call a low-level machine to take an action and then obtains the corresponding reward. Afterwards, the agent selects the next state by a state transition function until the end of the time step or an intersection is reached.

Inspired by the above advanced approaches, we propose a context-aware RL to improve the effectiveness of course recommendation. Our proposed model is novel in that: (1) We are the first to integrate RRL using temporal context into the HRL approach, where a novel policy gradient method is proposed to improve the policy performance; (2) We subtlety synchronize RRL approach with different episodes in the HRL model.

4. Methodologies

In this section, we first present the profile reviser by HRRL. After that, we elaborate separable components of HRRL. Next, we propose a novel interaction between the attention-based recommendation model and the profile reviser with RRL using temporal context. Finally, we describe the process of model training.

Overview. The overall framework of our proposed model is illustrated in Fig. 2. It contains a profile reviser and an attention-based recommendation model. Different from the HRL model [7], we propose a novel interaction between the attention-based recommendation model and the profile reviser with RRL using temporal context. In particular, an RRL using temporal context is proposed for hierarchical tasks, in which an agent chooses a high-level action following a high-level policy, and calls the low-level actions to remove noisy courses following a low-level policy. The key challenge is how to revise user profiles in a stable way. To this aim, we introduce a contextual policy gradient method with approximation. In this way, the RRL approach in the high-level task may make better decisions to revise user profile or not, and the RRL approach in the low-level task may gradually remove the noisy courses (*e.g.*, e_3^u in Fig. 2, etc.). Once the user profile ε^u is revised, the agent gets two different delayed rewards according to the hierarchical tasks from the environment, which can be viewed as the dataset and the pre-trained recommendation model. Then the recommendation model based on the attention mechanism outputs a recommendation probability $P(y = 1 \mid \hat{\varepsilon}^u, c_i)$, which is taken as a policy evaluation for the profile reviser in the next episode. The higher the recommendation probability is, the better policy evaluation can be provided for the profile reviser. Finally, our HRRL recurrently trains the profile reviser, as well as the recommendation model, using RRL with temporal context. The profile reviser and the recommendation model are jointly trained by sharing the embeddings of user profiles. In this way, the reconstructed user profiles enable the recommendation model to generate accurate recommendations.

4.1. Profile reviser by HRRL

To efficiently reconstruct user profiles for course recommendation, we present an RRL using temporal context, which is integrated into the HRL model. In the HRRL-based profile reviser, a high-level RRL module enables the high-level task to makes a decision to revise user profiles more properly, while a low-level RRL module urges the low-level task to iteratively remove the noisy courses that are irrelevant to the target course. In addition, we propose a contextual policy gradient method with approximation for RRL, which guarantees reliable convergence of policy gradient. The details of our HRRL are elaborated in the following subsections.

4.1.1. HRL for profile-revising

Inspired by HAMs [52], the task of profile-revising can be formulated as two-level Markov decision processes (MDPs), in which a high-level task in the hierarchy revises the whole profile of a user, and a low-level task in the hierarchy removes the irrelevant courses. Each kind of task is cast as a 5-tuple MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \beta \rangle$, in which $\mathcal{S}$ denotes a set of states, and $\mathcal{A}$ denotes a set of actions. $\mathcal{P}$ represents a transition function mapping $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$ into probabilities in [0,1]. $\mathcal{R}$ represents the reward function. $\beta \in (0, 1)$ is a discount coefficient of the reward in current episode. The state, action, reward and policy for the hierarchical tasks are defined as follows [7].

State. The state feature s^h in the high-level task is defined as the average cosine similarity between the embedding vectors of each historical course in ε^u and the target course, as well as the average element-wise product between them. For the low-level task, the state feature s^l is defined as the cosine similarity between the embedding vectors of the current historical course e_t^u and the target course c_i, the element-wise product between them, as well as the average of the two previous features over all the reserved historical courses.

Action. The high-level action $a^h \in \{0, 1\}$ decides whether to revise the user profile ε^u or not. The low-level action $a^l \in \{0, 1\}$ decides whether to remove the historical course $e_t^u \in \varepsilon^u$ or not.

Reward. If the high-level task calls the low-level task to remove noisy courses, the agent will receive a delayed reward $R_h(s, a)$ once the last low-level action is conducted. Here $R_h(s, a)$ denotes the difference between the *log* likelihood after and before the user profile is revised. The high-level reward can be defined as below:

$$R_h(s_t^h, a_t^h) = \log P(\hat{\varepsilon}^u, c_i) - \log P(\varepsilon^u, c_i), \text{if } t = t_u, \tag{2}$$

where s_t^h denotes the high-level state at time t, a_t^h denotes the high-level action at time t, $\hat{\varepsilon}^u$ is the revised profile, ε^u is the original profile, c_i denotes a target course, $P(\hat{\varepsilon}^u, c_i)$ is a probability of recommending c_i to the user u after the user profile is revised, $P(\varepsilon^u, c_i)$ is a probability of recommending c_i to the user u before the user profile is revised, and t_u denotes the number of historical courses before the user profile is revised. The high-level reward $R_h(s_t^h, a_t^h) = 0$ if $t \neq t_u$.

The reward of the low-level task $R_l(s, a)$ contains two different types of rewards. The first is the same delayed reward defined as the difference between the likelihood after and before the user profile is revised. The second is an internal reward defined as the difference of the average cosine similarity between the target course and each historical course after and before the user profile is revised. Thus, we define the following low-level reward.

$$R_l(s_t^l, a_t^l) = \big(\log P(\hat{\varepsilon}^u, c_i) - \log P(\varepsilon^u, c_i)\big) + \Big(\sum_{t=1}^{\hat{t}_u} (e_t^{uT} c_i)/\hat{t}_u - \sum_{t=1}^{t_u} (e_t^{uT} c_i)/t_u\Big), \text{if} t = t_u, \tag{3}$$

where s_t^l denotes the low-level state at time t, a_t^l denotes the low-level action at time t, $e_t^{uT} c_i$ represents the cosine similarity between a historical course e_t^u and the target course c_i, and $\hat{t}_u$ is the number of historical courses after the user profile is revised. The low-level reward $R_l(s_t^h, a_t^h) = 0$ if $t \neq t_u$.

Policy. The high-level policy function is defined as follows:

$$\pi_\theta(s_t^h, a_t^h) = softmax\Big(a_t^h \sigma(W_2^h x_t^h) + (1 - a_t^h)\big(1 - \sigma(W_2^h x_t^h)\big)\Big),$$
$$x_t^h = ReLU(W_1^h s_t^h + b^h), \tag{4}$$

where $W_1^h \in \mathbb{R}^{d_1 * d_2}$, $W_2^h \in \mathbb{R}^{d_1}$ and the bias vector $b^h \in \mathbb{R}^{d_1}$ are the parameters to learned with d_1 as the hidden layer size and d_2 as the number of state features. x_t^h is the learned hidden features of the state at time t in the high-level task. σ is a non-linear activation function that transforms the state into a probability, and the Rectified Linear Unit (*ReLU*) is the activation function for the hidden layer. Thus, the high-level policy's parameter is defined as $\theta^h = \{W_1^h, W_2^h, b^h\}$. Similarly, the low-level policy's parameter can be defined as $\theta^l = \{W_1^l, W_2^l, b^l\}$.

4.1.2. RRL using temporal context

We emphasize that in Monte Carlo methods, when an agent in RL gets both current and previous rewards from the environment in different episodes, it is likely to choose a better policy and to take more reasonable actions compared to the situation in which it is getting only the current reward. This is mainly due to the fact that it can retrieve the context information. Following this idea, we propose an RRL method that uses temporal context. Formally, the total reward in each episode can be defined by

$$R^k(\tau) = \frac{1}{m} \sum_{m=1}^{K} \varpi * R^{k-m}(\tau) + \beta * R^k(\tau), k \geq m \geq 1, \tag{5}$$

Algorithm 1 Contextual Policy Gradient Method with Approximation for RRL

```
Input: a derivable policy parameterization $\pi_\theta(s, a)$, a derivable
    state-value parameterization $\hat{v}(s, \mathbf{w})$;
Initialize: parameter $\theta = \theta^0$, $\mathbf{w} = \mathbf{w}^0$, $R^0(s, a) = 0$;
 1: for episode k = 1 to K do
 2:   Generate an episode $S_0, A_0, R_1, \cdots, S_{T-1}, A_{T-1}, R_T$, following
      $\pi_\theta(\cdot, \cdot)$;
 3:   for each step of the episode $t = 0, 1, \cdots, T-1$ do
 4:     $R^k(S_t, A_t) = \sum_{i=t+1}^{T} \gamma^{i-t-1} R_i$;
 5:     $R^k(S_t, A_t) = R^k(S_t, A_t) - \hat{v}^k(S_t, \mathbf{w})$;
 6:     $G_t^k = (1-\beta) R^{k-1}(s, a) + \beta R^k(S_t, A_t)$;
 7:     $\mathbf{w} = \mathbf{w} + \alpha^{\mathbf{w}} G_t^k \nabla \hat{v}^k(S_t, \mathbf{w})$;
 8:     $\theta = \theta + \alpha^\theta G_t^k \nabla \log \pi_\theta(S_t, A_t)$;
 9:   end for
10:   $R^{k-1}(s, a) = R^k(S_{T-1}, A_{T-1})$;
11: end for
```

where τ denotes the sequence of transited states and actions, K represents the number of episodes, ϖ denotes the weight of the previous reward, $R^{k-m}(\tau)$ is the previous reward of each sequence τ in the $(k-m)$th episode (which is equal to zero in the first episode), $R^k(\tau)$ refers to the reward for each sequence τ in the kth episode, and $\beta \in (0, 1)$ represents a discount coefficient of the reward in current episode.

To match with the policy gradient method with approximation which will be introduced in the next subsection, here we only add the previous reward before current episode to the total reward in current episode (*i.e.*, $(1-\beta)R^{k-1}(\tau) + \beta R^k(\tau)$), where the previous reward is obtained according to the recommendation results before the current session.

4.1.3. Contextual policy gradient with approximation

As mentioned before, the RRL approach may obtain the context from an environment. It should be emphasized that the policy gradient method can further improve its performance by RRL using temporal context, since it continues with the optimal policy from the previous episode. Thus, extending the policy gradient theorem [53], we propose the following contextual policy gradient method with approximation for RRL.

Corollary 1. *For any MDP in Monte Carlo methods, given $R^{k-1}(s, a)$, and $\hat{v}^k(S_t, \mathbf{w})$ that is an approximation to $R^k(S_t, A_t)$, for all $s, S_t \in \mathcal{S}$, and $a, A_t \in \mathcal{A}$,*

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a \nabla_\theta \pi_\theta(s, a)\big((1-\beta)R^{k-1}(s, a) + \beta R^k(S_t, A_t)\big),$$
$$R^k(S_t, A_t) = \sum_{i=t+1}^{T} \gamma^{i-t-1} R_i - \hat{v}^k(S_t, \mathbf{w}), \tag{6}$$

where the symbol $\propto$ means "proportional to", $\mu(s)$ denotes on-policy distribution under the policy π, $\mathbf{w}$ denotes the vector of feature weights, which can be defined as the vector of connection weights in all the layers in artificial neural network, $\gamma \in [0, 1]$ is the discount factor of the reward, R_i denotes the immediate reward at time $t+1$, $R^k(S_t, A_t)$ refers to the delayed reward in the kth episode, and $R^{k-1}(s, a)$ is the reward in the $(k-1)$th episode. This novel method is well defined only for the episodic case, as all updates are made after current episode is completed.

The pseudocode (Algorithm 1) may be described explicitly following *Corollary* 1. In each episode, the reward of the previous episode is added to the overall returns of a state at time t (corresponding to line 6 in algorithm 1). Then the policy gradient is updated along with each process of gradient ascent (corresponding to line 7 and 8 in algorithm 1 respectively). After the current episode terminates, the reward is set as the previous reward in the next episode. In this way, the policy is iteratively improved in consecutive episodes.

Algorithm 1 has reliable convergence, because the update for θ is proportional to the return G_t in each episode, and it is stable since $R^{k-1}(s,a)$ can guarantee the continuity of the policy between adjacent episodes. In this case, the expected update over each episode can be in the same direction as the policy gradient, and thereby speeds up the policy convergence per episode.

4.1.4. Objective function

To enhance the overall performance of course recommendation, we propose the context-aware reinforcement learning model (*i.e.*, HRRL), in which the high-level RRL module makes better decision to revise user profiles in the high-level task, and the low-level RRL module iteratively removes the noisy courses in the low-level task.

The hierarchical tasks simultaneously use the contextual policy gradient method with approximation (*i.e.*, Algorithm 1) to learn the policy. The objective function of the policy is to maximize the expectation of accumulated rewards as follows.

$$\theta^* = \arg\max_{\theta} \sum_{\tau} P_\theta(\tau) R(\tau), \tag{7}$$

where θ denotes either the high-level policy's parameter θ^h or the low-level policy's parameter θ^l, τ is a sampled sequence of actions and transited states, which can be $\{s^h, a^h\}$ for the high-level task and $\{s_1^l, a_1^l, \dots, s_t^l, a_t^l, \dots\}$ for the low-level task, $P_\theta(\tau)$ is the corresponding probability of states transition, $R(\tau)$ denotes the reward for the sampled sequence τ [7].

We adopt Algorithm 1 to sample N action–state trajectories. Thus, the gradient for the high-level policy function is computed by Algorithm 1 as follows.

$$\nabla_\theta = \frac{1}{n} \sum_{n=1}^{N} \sum_{t=1}^{t_u} \nabla_\theta \log \pi_\theta(s^n, a^n) \big((1-\beta) R_h^{k-1}(s,a) + \beta R_h^k(s_t^n, a_t^n) \big), \tag{8}$$

where t_u denotes the number of historical courses, (s_t^n, a_t^n) is an action–state pair in sampled sequence τ^n, $R_h^{k-1}(s,a)$ is the reward of each sampled sequence τ^n before current episode, and $R_h^k(s_t^n, a_t^n)$ refers to the delayed reward of each sampled sequence τ^n in current episode in the high-level task.

After the high-level task calls the low-level task to remove the noisy courses, the high-level task receives a delayed reward $R_h(s_t^n, a_t^n)$, and the low-level task receives another delayed reward $R_l(s_t^n, a_t^n)$. The low-level task is conducted by RRL using temporal context. Accordingly, the gradient for the low-level policy function is computed by Algorithm 1 as follows.

$$\nabla_\theta = \frac{1}{n} \sum_{n=1}^{N} \sum_{t=1}^{t_u} \nabla_\theta \log \pi_\theta(s_t^n, a_t^n) \big((1-\beta) R_l^{k-1}(s,a) + \beta R_l^k(s_t^n, a_t^n) \big), \tag{9}$$

where $R_l^{k-1}(s,a)$ is the reward of each sampled sequence τ^n before current episode, and $R_l^k(a_t^n, s_t^n)$ is the reward of each sampled sequence τ^n in current episode in the low-level task. For simplicity, we omit the superscript h of both s and a in Eqs. (8) and (11), and also omit the superscript l of both s and a in Eqs. (10) and (9).

4.2. Separable components of HRRL

We adopt the term HLRRL to denote the high-level RRL module combined with the HRL model. And the term LLRRL denotes the low-level RRL module combined with the HRL model. As mentioned above, the accuracy of course recommendation can be improved when the combination of these two approaches is applied in HRRL. In this section, we elaborate the two approaches. We will conduct an ablation study to compare them with HRL and HRRL in Section 5.

4.2.1. HLRRL

As mentioned previously, a main issue of the HRL model is that the high-level task cannot properly determine whether to revise the user profiles, since the high-level policy converges to a local optimum due to the properties of policy gradient methods. In this case, the whole profile may not be well reconstructed. Indeed HLRRL is intended to help the high-level task to take better decisions about revising user profiles via RRL using temporal context, which employs the contextual policy gradient method with approximation to learn the policy. Accordingly, the gradient for the high-level policy function is computed by Eq. (8), which is also adopted in the HRRL model.

To compare the performance with HRL fairly, we also use REINFORCE algorithm [43] to compute the gradient for the low-level policy function:

$$\nabla_\theta = \frac{1}{n} \sum_{n=1}^{N} \sum_{t=1}^{t_u} \nabla_\theta \log \pi_\theta(s_t^n, a_t^n) R_l(s_t^n, a_t^n), \tag{10}$$

where $R_l(s_t^n, a_t^n)$ denotes the delayed reward of each sampled sequence τ^n in the low-level task.

4.2.2. LLRRL

Another critical challenge in the HRL model is how to remove the noisy courses in an appropriate way. Especially when there are many irrelevant courses with few contributions in recommending the target course, the low-level task in the HRL model just randomly removes some noisy courses, mainly due to the randomness of the policy. To address this challenge, we introduce LLRRL to help the low-level task to remove the noisy courses iteratively, which improves the revising policy by RRL using temporal context and thereby keeps the most relevant courses.

To verify the effectiveness of RRL using temporal context for the low-level task, we also adopt REINFORCE algorithm to compute the gradient for the high-level policy function, which is defined the same as that of the HRL model as follows [7].

$$\nabla_\theta = \frac{1}{n} \sum_{n=1}^{N} \sum_{t=1}^{t_u} \nabla_\theta \log \pi_\theta(s^n, a^n) R_h(s_t^n, a_t^n), \tag{11}$$

where $R_h(s_t^n, a_t^n)$ denotes the delayed reward of each sampled sequence τ^n in the high-level task.

In the same way, the gradient for the low-level policy function is computed by Eq. (9), which is also adopted in the HRRL model.

4.3. Attention-based recommendation model

To fairly compare the recommendation performance with the HRL model, we also adopt the NAIS model [8] employed by the HRL model as the basic recommendation model, in which the embedding vector of user profile $\mathbf{q}_u$ can be aggregated by

$$\mathbf{q}_u = \sum_{t=1}^{t_u} a_{it}^u \mathbf{p}_t^u, \; a_{it}^u = f(\mathbf{p}_t^u, \mathbf{p}_i), \tag{12}$$

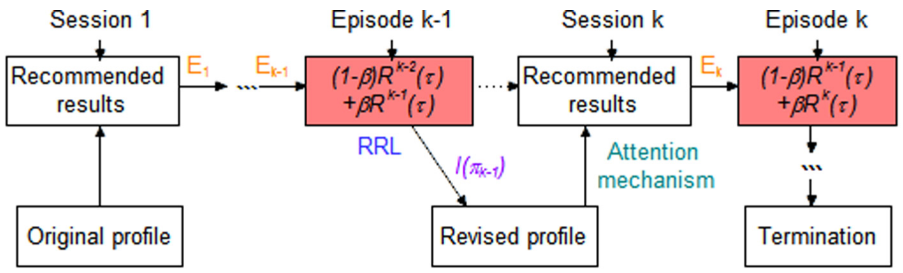


Fig. 3. The novel interaction between the attention-based recommendation model and the profile reviser with RRL using temporal context. E denotes the policy evaluation, and $I(\pi_{k-1})$ denotes the policy improvement in the $(k-1)$th episode.

where $\mathbf{p}_t^u$ is the embedding vector of the historical course e_t^u, $\mathbf{p}_i$ is the embedding vector of target course c_i, and a_{it}^u denotes the attention weight of the course e_t^u when the target course c_i is recommended. The attention function $f(\mathbf{p}_t^u, \mathbf{p}_i)$ is implemented by a multilayer perceptron (MLP) as follows.

$$f(\mathbf{p}_t^u, \mathbf{p}_i) = \mathbf{h}^\top \mathrm{ReLU}\big(\mathbf{W}^{att}(\mathbf{p}_t^u \odot \mathbf{p}_i) + \mathbf{b}^{att}\big), \tag{13}$$

where $\mathbf{h}^\top$ is the vector that projects a hidden layer into an attention weight, $\mathbf{W}^{att} \in \mathbb{R}^{D_h * D_e}$ is the weight matrix, and $\mathbf{b}^{att} \in \mathbb{R}^{D_h}$ is the bias vector, which are the parameters of MLP to be learned with D_h as the hidden layer size and D_e as the course embedding size. Here *ReLU* is served as the activation function for the hidden layer.

In this way, the attention-based recommendation model can generate a recommendation probability according to $\mathbf{q}_u$ and $\mathbf{p}_i$. Formally, the recommendation probability can be defined as follows.

$$\rho_{u,i} = P(y = 1 \mid \hat{\varepsilon}^u, c_i) = \sigma(\mathbf{p}_i^T \mathbf{q}_u), \tag{14}$$

where $y = 1$ indicates that the target course c_i is recommended to a user u, σ denotes the activation function for transforming an input into a recommending probability, and $\rho_{u,i}$ represents the probability of recommending c_i to u [7].

4.4. Interaction method

After introducing the profile reviser with RRL and the attention-based recommendation model, we can illustrate the details of the novel interaction between them in Fig. 2. The profile reviser leverages RRL using temporal context to reconstruct user profiles by the policy improvement, which is guided by a contextual policy gradient method with approximation for the hierarchical tasks. And the course recommendation model makes accurate recommendations according to the revised profiles, and then provides better policy evaluation (*i.e.*, higher recommendation probability) to the agent in the profile reviser. Following this way of interaction, the profile reviser with RRL and the attention-based recommendation model are trained jointly in consecutive episodes.

More specifically, the novel interaction between the agent in the profile reviser and the attention-based recommendation model is illustrated in Fig. 3. First, the recommendation model based on the attention mechanism outputs recommended results and evaluates the policy for the agent. Next, the agent employs RRL that uses temporal context to remove the noisy courses according to the policy improvement $I(\pi_{k-1})$ in the $(k-1)$th episode. Then, based on revised profiles, the recommendation model generates more accurate recommendation results for users, and provides the policy evaluation E_k for the agent in the kth episode. The process goes on until the last episode terminates.

We argue that the interaction method is effective because the attention-based recommendation model provides better policy evaluation for the agent through accurate recommendation, while RRL using temporal context enhances the representation of user profiles by employing a better policy to take reasonable actions. As a result, by sharing the embeddings of user profiles to jointly train the profile reviser and the recommendation model, the reconstructed user profiles enable the recommendation model to make accurate recommendations.

Algorithm 2 Context-aware Reinforcement Learning Model (*i.e.*, HRRL) for Course Recommendation

```
Input: training data $\varepsilon^u$, pre-train recommendation model parameterized by $\Phi^0 = \{h^\top, W^{att}, b^{att}\}$, and pre-train profile reviser parameterized by $\theta^h_{pre} = \{W_1^h, W_2^h, b^h\}$ and $\theta^l_{pre} = \{W_1^l, W_2^l, b^l\}$;
Initialize: $\theta^h = \theta^h_{pre}$, $\theta^l = \theta^l_{pre}$, $\Phi = \Phi^0$, $R_h^0(s,a)$=0, $R_l^0(s,a)$=0;
1:  for episode $k = 1$ to $K$ do
2:    for each $\varepsilon^u := (e_1^u, ..., e_{t_u}^u)$ and target course $c_i$ do
3:      Sample a high-level action $a^h$ with $\theta^h$ in the high-level task;
4:      if $P(y = 1 \mid \hat{\varepsilon}^u, c_i) > 0.5$ then
5:        $R_h^k(s_t^h, a_t^h) = 0$
6:      else
7:        Sample a sequence of states and actions with $\theta^l$ in the low-level task;
8:        Calculate $R_h^{k-1}(s,a)$, $R_l^{k-1}(s,a)$, $R_h^k(s_t^n, a_t^n)$ and $R_l^k(s_t^n, a_t^n)$;
9:        Calculate the gradients by Eq. (8) and (9) according to Algorithm 1;
10:     end if
11:   end for
12:   Update $\theta^h$ and $\theta^l$ by the gradients;
13:   Update $\Phi$ in the recommendation model;
14:   Output the recommendation probability $P(y = 1 \mid \hat{\varepsilon}^u, c_i)$ by Eq. (14);
15: end for
```

4.5. Training procedure

The training procedure of HRRL is similar to that of the HRL model [7]. The primary difference is that we introduce the novel interaction between the attention-based recommendation model and the profile reviser, which employs the RRL using temporal context in hierarchical tasks. In the first stage, we pre-train the recommendation model (*i.e.*, $\Phi^0 = \{h^\top, W^{att}, b^{att}\}$) with the original dataset. In the second stage, we pre-train the profile reviser (*i.e.*, $\theta^h_{pre} = \{W_1^h, W_2^h, b^h\}$ and $\theta^l_{pre} = \{W_1^l, W_2^l, b^l\}$) by Algorithm 2 to recurrently revise the user profile, and combine this with the recommendation model. Finally, the profile reviser and the recommendation model are trained jointly by Algorithm 2. If the recommendation probability $P(y = 1 \mid \hat{\varepsilon}^u, c_i)$ is larger than a given threshold (*e.g.*, 0.5), indicating that the recommendation

Table 1
Statistic information of the datasets.

Dataset	#Course	#User	#Interaction
MOOCCourse	1302	82,535	458,453
MOOCCube	706	55,203	354,541

model provides an ideal policy evaluation for the profile reviser, the high-level task will decide not to revise the whole profile. As a result, the high-level reward $R_h^k(s_t^h, a_t^h) = 0$ (corresponding to line 5 in Algorithm 2). Otherwise, the high-level task selects a revising action for calling the low-level task to iteratively remove some noise courses. Then, the two tasks obtain the delayed rewards and calculate the gradients for the hierarchical policy functions (corresponding to line 7–9 in Algorithm 2). After that, the profile reviser is updated and provides reconstructed user profiles to assist the update of the recommendation model (corresponding to line 12–13 in Algorithm 2). As shown in Algorithm 2, we carefully synchronize the RRL using temporal context with the different episodes of HRL, and the gradients of θ are calculated by Eqs. (8) and (9) according to Algorithm 1 in HRRL (corresponding to line 9 in Algorithm 2). Overall, the time complexity is $O(K(MN\bar{t}_u))$, where K represents the number of episodes, M denotes the number of instances, N denotes the Monte-Carlo sampling time, and $\bar{t}_u$ represents the average number of historical courses [7]. For HLRRL, the gradients of θ are computed by Eqs. (8) and (10) instead of Eqs. (8) and (9). For LLRRL, the gradients of θ are computed by Eqs. (11) and (9) instead of Eqs. (8) and (9).

5. Experiments and analysis

In this section, we firstly describe the MOOCs datasets and the experimental setup, and then discuss the experimental results.

5.1. Data description

We conduct experiments on two real-world datasets, *i.e.*, MOOCCourse[1] and MOOCCube, which come from the scenario of XuetangX[2] at different stages, in which most users have enrolled in many different courses. More precisely, on the MOOCCourse dataset, there are 1302 courses and 82,535 users. Each user enrolled in at least 3 courses. On the MOOCCube dataset, there are 706 courses and 55,203 users. Each user enrolled in at least 4 courses. Detailed statistical information about both datasets is shown in Table 1. In order to provide a fair comparison, we adopt the same approach of data preprocessing used in the HRL model, according to temporal information of the enrollment data. Specifically, the enrolled behaviors in the training set precede those in the test set, and each instance in the training set or the test set is a sequence of historical courses paired with a target course. During the training process, all courses except the last course in the sequence are treated as historical courses, and the last course in each sequence is regarded as the target course. During the test process, each historical course in the test set is regarded as the target course, and the corresponding courses of the same user in the training set are treated as historical courses [7].

[1] http://moocdata.cn/data/course-recommendation.
[2] http://www.xuetangx.com.

5.2. Experimental setup

5.2.1. Compared methods

To assess its performance, we compare the HRRL model with some competitive baselines, which are given below:

- **MLP** [54]: applies a MLP on a pair of user and item embeddings to make recommendations.
- **FISM** [55]: an item-based collaborative filtering algorithm that learns course similarity matrix as the product of two low dimensional latent factor matrices.
- **NeuMF** [54]: combines the Matrix Factorization method with a MLP to model user-course latent information.
- **NARM** [9]: a modified gated recurrent unit model that estimates an attention coefficient based on user's behaviors and main purpose.
- **NAIS** [8]: an attention network that distinguishes the weights of different historical courses for a prediction.

To assess the effectiveness of our proposed interaction method, as well as the significance of RRL using temporal context in hierarchical tasks, we study the recommendation performances of HRL, and the two variants of HRRL (*i.e.*, HLRRL and LLRRL).

- **HRL** [7]: combines the recommendation model with a profile reviser based on HRL that ignores the novel interaction method.
- **HLRRL**: the simplified version of HRRL that adopts RRL using temporal context only in the high-level task, but ignores it in the low-level task.
- **LLRRL**: the simplified version of HRRL that adopts RRL using temporal context only in the low-level task, but ignores it in the high-level task.

5.2.2. Evaluation metrics

We compare the above-mentioned methods in terms of the following common metrics:

- HR@N (*i.e.*, Hit Ratio of top N items): a recall-based metric that measures the percentage of the data sets which are recommended as top N items to users.
- NDCG@N (*i.e.*, Normalized Discounted Cumulative Gain of top N items): a precision-based metric that normalizes the predicted scores of recommendation lists for users using position factors.

5.2.3. Parameter settings

For the attention-based recommendation model, both course embedding size D_e and the hidden layer size D_h are set to 16, the size of the mini-batch is set to 256, and the learning rate is set to 0.02. For the profile reviser, Monte Carlo sampling time N is set to 4, the learning rate is set to 0.001/0.0005 at the pre-training and joint-training stage, respectively. In the policy function, the hidden layer size d_1 in hierarchical tasks is set to 8, and the discount coefficient β of the reward in current episode is set to 0.5. Without special mention in the following subsections, we report the performance of HRRL with the default settings: (1) the hidden layer size in the attention-based recommendation model is 16; (2) the embedding size is 16; (3) N is empirically set to 10.

5.3. Performance comparison

As shown in Table 2, all our proposed models outperform state-of-the-art baselines in recommendation performance. In particular, our HRRL improves HR from 1.52% to 12.40%, and NDCG from 0.75% to 6.80% on MOOCCourse. Moreover, it improves HR from 2.37% to 15.49%, and NDCG from 2.51% to 8.97%

Table 2
Recommendation performance evaluated by HR and NDCG (%). The best results are highlighted in bold.

Method	MOOCCourse				MOOCCube			
	HR@5	HR@10	NDCG@5	NDCG@10	HR@5	HR@10	NDCG@5	NDCG@10
MLP	52.53	66.74	40.61	44.90	53.25	67.08	40.31	44.81
FISM	53.12	65.89	40.63	45.13	53.22	65.90	40.68	45.21
NeuMF	54.20	67.25	42.06	46.05	54.11	68.14	40.95	45.72
NARM	54.23	69.37	42.54	47.24	54.67	69.15	41.83	47.09
NAIS	56.05	68.98	43.58	47.69	56.18	68.73	43.49	47.50
HRL	59.84	75.00	44.50	50.95	62.03	76.70	45.64	51.27
HLRRL	60.63	76.95	44.99	51.12	63.71	80.93	47.46	52.52
LLRRL	61.02	77.30	45.75	51.31	63.82	81.15	47.92	53.47
HRRL	**61.36**	**78.29**	**45.82**	**51.70**	**64.40**	**82.57**	**48.39**	**53.78**

Table 3
The running time per episode (second).

Method	MOOCCourse			MOOCCube		
	Test time	Training time	Total time	Test time	Training time	Total time
HRL	18.50	97.80	116.30	11.80	58.90	70.70
HLRRL	17.80	97.30	115.10	11.20	58.50	69.70
LLRRL	18.10	96.90	115.00	11.40	58.10	69.50
HRRL	17.60	96.70	114.30	11.10	57.80	68.90

on MOOCCube. These results demonstrate that HRRL can improve the accuracy of course recommendation, especially when users are interested in many different courses. Overall, these results provide empirical evidence about the effectiveness of our proposed models, which are promising in view of further applications.

Among all the considered baselines, MLP, FISM and NeuMF show the worst performance, as they cannot distinguish the effects of the contributing courses when they recommend the target course. And NARM and NAIS perform worse than all RL-based models, since they fail to distinguish the useful and useless courses effectively. All RL-based models outperform the other baselines on both datasets. The main reason is that RL-based models tune course recommendation by the profile reviser, which is suitable to take into account users' preferences. In particular, they perform better on MOOCCube than on MOOCCourse in terms of both HR and NDCG, which suggests that RL-based models are suitable to recommend the target courses to users who are interested in many courses, since most users in MOOCCube enrolled in more courses than users in MOOCCourse. In this case, there are more states available to the agent to take reasonable actions following a revising policy. This allows the agent to better remove the noisy courses, and keep more contributing courses to recommend a more accurate target course to a given user.

The results of Table 3 indicate that our proposed models take less running time per episode than the HRL model, on MOOCCube and MOOCCourse, respectively. For example, HRRL takes less than 2.0 s on MOOCCourse and 1.8 s on MOOCCube in total time per episode, compared with the HRL model. These experimental results support the initial judgment that our proposed models can revise user profiles efficiently and achieve reliable convergence. The main reason is that we subtlety synchronize the RRL using temporal context with different episodes of HRL. In this way, the expected update over each episode is in the same direction as the policy gradient, which speeds up the policy convergence per episode in the profile reviser. We believe that the advantage of both time cost and recommendation accuracy can provide a positive application for our HRRL model in online course recommendation scenarios.

5.4. Ablation study

Fig. 4 illustrates the recommendation performance of RL-based models with respect to (w.r.t.) different top N, on the MOOCCourse and MOOCCube datasets. Among the RL-based models, HRL is outperformed by the models using RRL in terms of HR on both datasets. This well demonstrates the effectiveness of our proposed interaction method.

Compared with HRL, HLRRL performs better in terms of all the considered metrics, since it can make better decision to revise user profiles by RRL using temporal context in the high-level task. LLRRL performs much better than HRL in terms of all the considered metrics, since it is able to remove noisy courses gradually by RRL using temporal context in the low-level task. Overall, this results prove the significance of RRL using temporal context in hierarchical tasks.

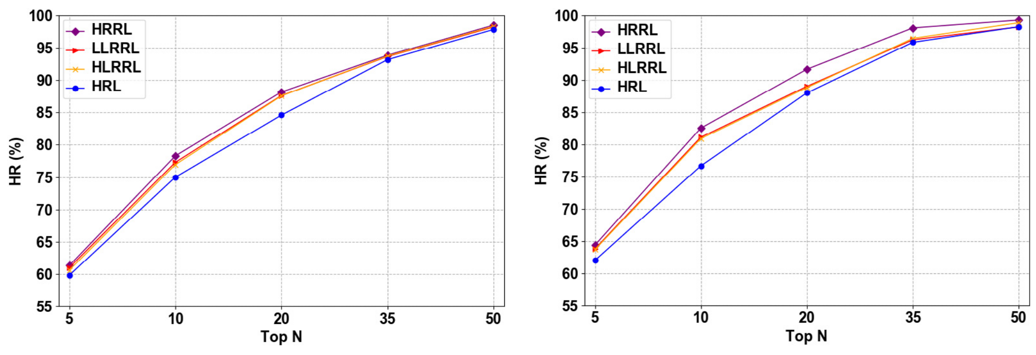
Observed from the above results, HRRL is consistently superior to the other two variants of the method on both datasets. It can be concluded that the combination of HLRRL and LLRRL evidently enhances the overall performance for course recommendation. The reason of this behavior is twofold. On the one hand, HRRL adopts contextual policy gradient method with approximation for both high-level and low-level policy, which reliably improves the policy in hierarchical tasks. On the other hand, HRRL adopts RRL using temporal context to iteratively revise the user profile in the high-level task and iteratively remove the noisy courses in the low-level task, thus achieving more accurate recommendations.

5.5. Influence of hyper-parameters

In this section, we evaluate how the performance of the proposed model varies with different sizes of two hyper-parameters, *i.e.*, the hidden layer size and the embedding size.

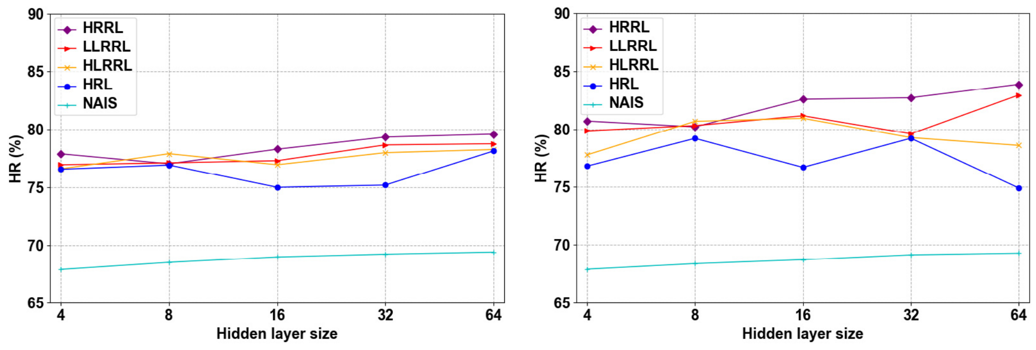
The hidden layer size is an important hyper-parameter of HRRL, which affects the recommendation performance. Fig. 5 illustrates the performance of five competitive models w.r.t. different settings of the hidden layer size. Empirically, the hidden layer size is set to 4, 8, 16, 32, and 64. As shown in Fig. 5(a), our proposed HRRL performs better than other models in terms of HR on MOOCCourse. When the hidden layer size increases, the recommendation performance improves in most cases, with the only exception of the hidden layer size 8, where HRRL performs worse than both HLRRL and LLRRL, suggesting that HRRL may be over-fitting. Fig. 5(b) shows that the HRRL model achieves the best performance on MOOCCube, except the hidden layer size 8, where HLRRL performs better in terms of HR. All our proposed models perform better than NAIS and HRL in all cases, which indicates that our proposed models have strong robustness against the size of the hidden layer.

Moreover, Fig. 6 shows the performances of five competitive models w.r.t. different settings of the embedding size. We



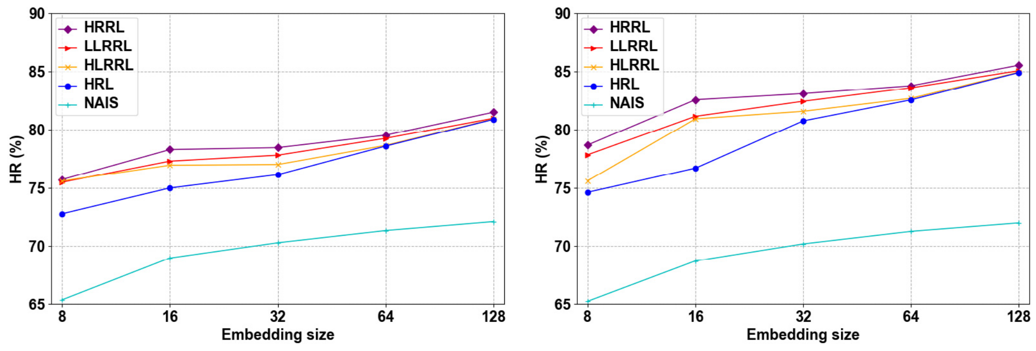
(a) (b)

Fig. 4. Recommendation performance of HRL-based models evaluated by HR (%) w.r.t. different top N. (a) HR on MOOCCourse. (b) HR on MOOCCube.



(a) (b)

Fig. 5. The performance trends of NAIS, HRL, HLRRL, LLRRL and HRRL w.r.t. the hidden layer size 4/8/16/32/64, in terms of HR@10. (a) HR on MOOCCourse. (b) HR on MOOCCube.



(a) (b)

Fig. 6. The performance trends of NAIS, HRL, HLRRL, LLRRL and HRRL w.r.t. the embedding size 8/16/32/64/128, in terms of HR@10. (a) HR on MOOCCourse. (b) HR on MOOCCube.

empirically set the embedding size to 8, 16, 32, 64, and 128, respectively. It is clear that RL-based models significantly outperforms NAIS in terms of HR, and HRRL performs the best on both MOOCCourse and MOOCCube datasets. We suggest that the improvement observed in the recommendation performance stems from the strong representation ability of HRRL.

From these results, we conclude that the recommendation performance of these compared models improves with the embedding size. When the embedding size becomes larger, the attention mechanism expresses a stronger representation power, *i.e.*, the models learn more useful information for the recommendation. Moreover, all RL-based models achieve better performance on MOOCCube than on MOOCCourse according to different embedding sizes. This result demonstrates again that these models are competent to recommend the target courses to users who are interested in many different courses, since most users in MOOCCube enrolled in more courses.

5.6. Qualitative analysis

To understand why the HRRL model is effective, we conduct a qualitative analysis with the case study. Table 4 illustrates two cases of course recommendation generated by HRL and the proposed HRRL. The first case presents that HRRL definitely removes the noisy course “Optics” that is irrelevant to the target course “Advanced Database Systems”, which is highly relevant to the last course “Database System” enrolled by the user. As

Table 4
Some cases of course recommendation by HRRL and HRL.

Methods	The revised profile	The target course
HRRL	Linear Algebra, Introductory Combinatorics, C++ Program, Database System	Advanced Database Systems
HRL	Linear Algebra, Introductory Combinatorics, Optics, C++ Program, Database System	Probability and Statistics
HRRL	Educational Sociology, Introduction to Psychology, Operations Management, Cultural Psychology	Social Psychology
HRL	Educational Sociology, Management Accounting, Economics, Introduction to Psychology, Operations Management, Cultural Psychology	Financial Management

shown in the second case, HRL randomly removes several noisy courses, and probably recommends the target course "Financial Management". In contrary, HRRL removes more noisy courses like "Management Accounting" and "Economics". Therefore, our proposed model probably recommends the target course "Social Psychology" that is in line with the user's interests, since the user has enrolled in more relevant courses like "Educational Sociology", "Introduction to Psychology" and "Cultural Psychology". It can be concluded that our HRRL model not only efficiently removes the courses with few contributions in a prediction, but also is able to recommend the target courses that indeed reflect a user's preference.

6. Conclusion

We put forward a context-aware reinforcement learning model for course recommendation, named Hierarchical and Recurrent Reinforcement Learning (HRRL), which is implemented by a novel interaction between the attention-based recommendation model and the profile reviser with RRL using temporal context. We are the first to integrate RRL using temporal context into the HRL approach, which can reconstruct user profiles efficiently. In this way, hierarchical tasks adopt RRL to iteratively revise user profiles, and thus recommend the most relevant target courses to users. Our scheme is further assisted by a contextual policy gradient method with approximation, which improves the policy performance for RRL. Experimental results demonstrate that our HRRL significantly improve the accuracy of course recommendation. The proposed approach provides a new insight for other recommendation scenarios, such as music and movie recommendations, where users usually listen to lots of music or watch various movies. In this case, HRRL efficiently reconstructs users' profiles to recommend their favorite items.

It should be noted that the contextual policy gradient method with approximation is based on Monte Carlo methods, *i.e.*, the performance measure of the policy is defined in the episodic case. In this way, RRL using temporal context can be naturally integrated into the HRL approach in different episodes. Therefore, a limitation of our HRRL is that it can be utilized only in the episodic case but not the continuing case. Besides, like most existing RL-based recommendation methods, HRRL focuses on the accuracy of course recommendation. However, when some users have diverse interests, such as liking various popular courses from different universities or courses of different majors, it may help to improve user satisfaction if the model simultaneously takes into account the accuracy and diversity of recommendation results. For future work, we will explore the interpretability of the HRRL model, leveraging causal inference approaches to improve the transparency and trustworthiness of course recommendation. Moreover, to achieve multi-objective goals (*i.e.*, the accuracy and diversity) of course recommendation, we will adopt multi-objective evolutionary algorithms [56] to optimize the evaluation measures, which can be taken as combinatorial optimization problems.

CRediT authorship contribution statement

Yuanguo Lin: Conceptualization, Methodology, Software, Writing – original draft. **Fan Lin:** Supervision, Funding acquisition. **Lvqing Yang:** Investigation, Data curation. **Wenhua Zeng:** Resources, Visualization. **Yong Liu:** Validation, Writing – review & editing. **Pengcheng Wu:** Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61977055).

Appendix. Proof of Corollary 1

The Bellman optimality equation for the state-value function $v_\pi^*(s)$ is defined as:

$$\begin{aligned} v_\pi^*(s) &= \max_{a\in\mathcal{A}(s)} R_\pi^*(s,a) \\ &= \max_a \sum_{s',r} p(s',r|s,a)\big[r+\gamma v_\pi^*(s')\big], \end{aligned} \tag{15}$$

where γ denotes the discount factor of the reward, $p(s',r|s,a)$ refers to the probability of transiting state from s to s' by taking action a, for all $s,s',S_t\in\mathcal{S}$, $r\in\mathcal{R}$, and $a,A_t\in\mathcal{A}$,

$$p(s',r|s,a) \doteq Pr\{S_{t+1}=s',R_{t+1}=r \mid S_t=s,A_t=a\}, \tag{16}$$

where S_{t+1} and R_{t+1} denote the state and the reward at time $t+1$ respectively, while S_t and A_t denote the state and the action at time t respectively.

Corollary 1 (*i.e.*, contextual policy gradient method with approximation for RRL) may be proved using the above equations and elementary calculus. To keep the notation simple, here we omit θ in terms of both all gradients and π, as well as $\mathbf{w}$ in terms of $\hat{v}(S_t)$. It should be noted that $\hat{v}(S_t)$ is the expected return of state S_t that does not vary with the next state s', and r is an immediate reward that no longer varies with the next state s' either. Similarly, $R^{k-1}(s,a)$ no longer varies with the next state s' at current episode because it is a generated action-value before current episode. Therefore, all of them are removed when we calculate the derivative of them. Thus,

$$\begin{aligned} \nabla v_\pi^k(s) =& \nabla\Big[\sum_a \pi(a|s)\Big((1-\beta)R^{k-1}(s,a)+\beta\big(R^k(S_t,A_t)-\hat{v}^k(S_t)\big)\Big)\Big] \\ =& \sum_a\Big[\nabla\pi(a|s)\Big((1-\beta)R^{k-1}(s,a)+\beta\big(R^k(S_t,A_t)-\hat{v}^k(S_t)\big)\Big) \\ &+ \pi(a|s)\nabla\Big((1-\beta)R^{k-1}(s,a) \end{aligned}$$

$$
\begin{aligned}
&+ \beta\big(R^k(S_t,A_t) - \hat{v}^k(S_t)\big)\Big)\Big] \\
&= \sum_a \Big[\nabla\pi(a|s)\Big((1-\beta)R^{k-1}(s,a) + \beta\big(R^k(S_t,A_t) - \hat{v}^k(S_t)\big)\Big) \\
&\quad + \pi(a|s)\nabla \sum_{s',r} p(s',r|s,a)\Big((1-\beta) \\
&\quad \times R^{k-1}(s,a) + \beta\big(r + \gamma v_\pi^k(s') - \hat{v}^k(S_t)\big)\Big)\Big] \\
&= \sum_a \Big[\nabla\pi(a|s)\Big((1-\beta)R^{k-1}(s,a) + \beta\big(R^k(S_t,A_t) - \hat{v}^k(S_t)\big)\Big) \\
&\quad + \pi(a|s)\sum_{s'} \beta\gamma p(s'|s,a)\nabla v_\pi^k(s')\Big] \\
&= \sum_a \Big[\nabla\pi(a|s)\Big((1-\beta)R^{k-1}(s,a) + \beta\big(R^k(S_t,A_t) - \hat{v}^k(S_t)\big)\Big) \\
&\quad + \pi(a|s)\sum_{s'} \beta\gamma p(s'|s,a)\sum_{a'}\Big[\nabla\pi(a'|s') \\
&\quad \times \Big((1-\beta)R^{k-1}(s',a') + \beta\big(R^k(S'_t,A'_t) - \hat{v}^k(S'_t)\big)\Big) \\
&\quad + \pi(a'|s')\sum_{s''}\beta\gamma p(s''|s',a')\nabla v_\pi(s'')\Big]\Big] \\
&\doteq \sum_{x\in\mathcal{S}}\sum_{m=0}^{\infty}(\beta\gamma)^m Pr(s\to x,m,\pi)\sum_a \nabla\pi(a|x) \\
&\quad \times \Big((1-\beta)R^{k-1}(s,a) + \beta\big(R^k(x,a) - \hat{v}^k(x)\big)\Big)
\end{aligned}
$$

where $Pr(s \to x, m, \pi)$ is the state transition probability from s to x in m steps under the policy π. For any $\beta \in [0,1]$ and $\gamma \in [0,1]$, we set $\zeta = \beta\gamma \in [0,1]$, where ζ can be treated as a discount factor. The sum of $\zeta^m Pr(s \to x, m, \pi)$ equals the average time $\eta(s)$ (*i.e.*, the number of time steps) spent in state s in a single episode, where ζ should be treated as a form of soft termination when $\zeta < 1$ [44]. After repeated unrolling the gradient of the action-value function, assuming that each episode starts in an initial state s_0, we can unroll the gradient of $J(\theta)$ as follows

$$
\begin{aligned}
\nabla J(\theta) &= \nabla v_\pi^k(s_0) \\
&\doteq \sum_s\Big(\sum_{m=0}^{\infty}\zeta^m Pr(s_0\to s,m,\pi)\Big)\sum_a\nabla\pi(a|s) \\
&\quad \times \Big((1-\beta)R^{k-1}(s,a) + \beta\big(R^k(s,a) - \hat{v}^k(s)\big)\Big) \\
&= \sum_s \eta(s)\sum_a \nabla\pi(a|s) \\
&\quad \times \Big((1-\beta)R^{k-1}(s,a) + \beta\big(R^k(s,a) - \hat{v}^k(s)\big)\Big) \\
&= \sum_{s'}\eta(s')\sum_s \frac{\eta(s)}{\sum_{s'}\eta(s')}\sum_a\nabla\pi(a|s) \\
&\quad \times \Big((1-\beta)R^{k-1}(s,a) + \beta\big(R^k(s,a) - \hat{v}^k(s)\big)\Big) \\
&= \sum_{s'}\eta(s')\sum_s \mu(s)\sum_a\nabla\pi(a|s) \\
&\quad \times \Big((1-\beta)R^{k-1}(s,a) + \beta\big(R^k(s,a) - \hat{v}^k(s)\big)\Big) \\
&\propto \sum_s \mu(s)\sum_a\nabla\pi(a|s) \\
&\quad \times \Big((1-\beta)R^{k-1}(s,a) + \beta\big(R^k(s,a) - \hat{v}^k(s)\big)\Big)
\end{aligned}
$$

where the distribution $\mu(s)$ is defined as the normalized fraction of time spent in each state $s \in \mathcal{S}$ [44]:

$$\mu(s) = \frac{\eta(s)}{\sum_{s'}\eta(s')}. \quad \square \tag{17}$$

References

[1] Xia Jing, Jie Tang, Guess you like: course recommendation in moocs, in: Proceedings of the International Conference on Web Intelligence, 2017, pp. 783–789.

[2] Asmaa Elbadrawy, George Karypis, Domain-aware grade prediction and top-n course recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 183–190.

[3] Yong Liu, Wei Wei, Aixin Sun, Chunyan Miao, Exploiting geographical neighborhood characteristics for location recommendation, in: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, 2014, pp. 739–748.

[4] Yujia Liu, Changyong Liang, Francisco Chiclana, Jian Wu, A knowledge coverage-based trust propagation for recommendation mechanism in social network group decision making, Appl. Soft Comput. 101 (2021) 107005.

[5] Chenyi Lei, Yong Liu, Lingzi Zhang, Guoxin Wang, Haihong Tang, Houqiang Li, Chunyan Miao, SEMI: A sequential multi-modal information transfer network for E-commerce micro-video recommendations, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 3161–3171.

[6] Ming Chen, Xiuze Zhou, DeepRank: Learning to rank with neural networks for recommendation, Knowl.-Based Syst. 209 (2020) 106478.

[7] Jing Zhang, Bowen Hao, Bo Chen, Cuiping Li, Hong Chen, Jimeng Sun, Hierarchical reinforcement learning for course recommendation in MOOCs, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 435–442.

[8] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, Tat-Seng Chua, NAIS: Neural attentive item similarity model for recommendation, IEEE Trans. Knowl. Data Eng. 30 (12) (2018) 2354–2366.

[9] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, Jun Ma, Neural attentive session-based recommendation, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1419–1428.

[10] Qibin Chen, Junyang Lin, Yichang Zhang, Hongxia Yang, Jingren Zhou, Jie Tang, Towards knowledge-based personalized product description generation in E-commerce, in: Proc. ACM SIGKDD Conf., 2019, pp. 3040–3050.

[11] Yong Liu, Peilin Zhao, Xin Liu, Min Wu, Lixin Duan, Xiao-Li Li, Learning user dependencies for recommendation, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 2379–2385.

[12] Peng Han, Zhongxiao Li, Yong Liu, Peilin Zhao, Jing Li, Hao Wang, Shuo Shang, Contextualized point-of-interest recommendation, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020.

[13] Aditya Parameswaran, Petros Venetis, Hector Garcia-Molina, Recommendation systems with complex constraints: A course recommendation perspective, ACM Trans. Inf. Syst. 29 (4) (2011) 1–33.

[14] Rel Guzman Apaza, Elizabeth Vera Cervantes, Laura Cruz Quispe, José Ochoa Luna, Online courses recommendation based on LDA, in: Proceedings of the 1st Symposium on Information Management and Big Data, 2014, pp. 42–48.

[15] Rel Guzman Apaza, Elizabeth Vera Cervantes, Laura Cruz Quispe, Jose Ochoa Luna, Course content analysis: An initiative step toward learning object recommendation systems for MOOC learners, in: Proceedings of the 9th international conference on educational data mining, 2016, pp. 347–352.

[16] Hao Zhang, Tao Huang, Zhihan Lv, SanYa Liu, Zhili Zhou, MCRS: A course recommendation system for MOOCs, Multimedia Tools Appl. 77 (6) (2018) 7051–7069.

[17] Jie Xu, Tianwei Xing, Mihaela van der Schaar, Personalized course sequence recommendations, IEEE Trans. Signal Proces. 64 (20) (2016) 5340–5352.

[18] Asmaa Elbadrawy, George Karypis, Domain-aware grade prediction and top-n course recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 183–190.

[19] Huynh-Ly Thanh-Nhan, Huu-Hoa Nguyen, Nguyen Thai-Nghe, Methods for building course recommendation systems, in: Proceedings of the Eighth International Conference on Knowledge and Systems Engineering, 2016, pp. 163–168.

[20] Xiao Li, Xiang Li, Jintao Tang, Ting Wang, Yang Zhang, Hongyi Chen, Improving deep item-based collaborative filtering with Bayesian personalized ranking for MOOC course recommendation, in: Proceedings of International Conference on Knowledge Science, Engineering and Management, 2020, pp. 247–258.

[21] William Hoiles, Mihaela Van Der Schaar, Bounded off-policy evaluation with missing data for course recommendation and curriculum design, in: Proceedings of the 33nd International Conference on Machine Learning, 2016, pp. 1596–1604.

[22] Peichann Chang, Chenghui Lin, Menghui Chen, A hybrid course recommendation system by integrating collaborative filtering and artificial immune systems, Algorithms 9 (3) (2016) 47.

[23] Shanshan Wan, Zhendong Niu, A hybrid E-learning recommendation approach based on learners' influence propagation, IEEE Trans. Knowl. Data Eng. 32 (5) (2020) 827–840.

[24] Yifan Zhu, Hao Lu, Ping Qiu, Kaize Shi, James Chambua, Zhendong Niu, Heterogeneous teaching evaluation network based offline course recommendation with graph learning and tensor factorization, Neurocomputing 415 (2020) 84–95.

[25] Yuanguo Lin, Yong Liu, Fan Lin, Pengcheng Wu, Wenhua Zeng, Chunyan Miao, A survey on reinforcement learning for recommender systems, 2021, arXiv preprint arXiv:2109.10665.

[26] Massimo David, Francesco Ricci, Harnessing a generalised user behaviour model for next-POI recommendation, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 402–406.

[27] Fan Zhou, Ruiyang Yin, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, Jin Wu, Adversarial point-of-interest recommendation, in: Proceedings of the 28th International Conference on World Wide Web, 2019, pp. 3462–3468.

[28] Pengfei Wang, Yu Fan, Long Xia, Wayne Xin Zhao, Shaozhang Niu, Jimmy Huang, KERL: A knowledge-guided reinforcement learning model for sequential recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 209–218.

[29] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, Dawei Yin, Recommendations with negative feedback via pairwise deep reinforcement learning, in: Proc. ACM SIGKDD Conf., 2018, pp. 1040–1048.

[30] Haokun Chen, Xinyi Dai, Han Cai, Weinan Zhang, Xuejian Wang, Ruiming Tang, Yuzhou Zhang, Yong Yu, Large-scale interactive recommendation with tree-structured policy gradient, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 3312–3320.

[31] Lixin Zou, Long Xia, Zhuoye Ding, Jiaxing Song, Weidong Liu, Dawei Yin, Reinforcement learning to optimize long-term user engagement in recommender systems, in: Proc. ACM SIGKDD Conf., 2019, pp. 2810–2818.

[32] Ruiyi Zhang, Tong Yu, Yilin Shen, Hongxia Jin, Changyou Chen, Text-based interactive recommendation via constraint-augmented reinforcement learning, in: Advances in Neural Information Processing Systems, 2019, pp. 15214–15224.

[33] Yueming Sun, Yi Zhang, Conversational recommender system, in: Proceedings of the 41rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2018, pp. 235–244.

[34] Claudio Greco, Alessandro Suglia, Pierpaolo Basile, Giovanni Semeraro, Converse-et-impera: Exploiting deep learning and hierarchical reinforcement learning for conversational recommender systems, in: Proceedings of the 16th International Conference on Italian Association for Artificial Intelligence, 2017, pp. 372–386.

[35] Yong Liu, Yingtai Xiao, Qiong Wu, Chunyan Miao, Juyong Zhang, Binqiang Zhao, Haihong Tang, Diversified interactive recommendation with implicit feedback, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 4932–4939.

[36] Qinxu Ding, Yong Liu, Chunyan Miao, Fei Cheng, Haihong Tang, A hybrid bandit framework for diversified recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, 2021, pp. 4036–4044.

[37] Xiting Wang, Yiru Chen, Jie Yang, Le Wu, Zhengtao Wu, Xing Xie, A reinforcement learning framework for explainable recommendation, in: Proceedings of IEEE International Conference on Data Mining, 2018, pp. 587–596.

[38] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, Yongfeng Zhang, Reinforcement knowledge graph reasoning for explainable recommendation, in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 285–294.

[39] Rong Gao, Haifeng Xia, Jing Li, Donghua Liu, Shuai Chen, Gang Chun, DRCGR: Deep reinforcement learning framework incorporating CNN and GAN-based for interactive recommendation, in: 2019 IEEE International Conference on Data Mining, 2019, pp. 1048–1053.

[40] Lixin Zou, Long Xia, Pan Du, Zhuo Zhang, Ting Bai, Weidong Liu, Jian-Yun Nie, Dawei Yin, Pseudo dyna-Q: A reinforcement learning framework for interactive recommendation, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 816–824.

[41] Yu Lei, Wenjie Li, Interactive recommendation with user-specific deep reinforcement learning, ACM Trans. Knowl. Discov. Data 13 (6) (2019) 61.

[42] Liwei Huang, Mingsheng Fu, Fan Li, Hong Qu, Yangjun Liu, Wenyu Chen, A deep reinforcement learning based long-term recommender system, Knowl.-Based Syst. 213 (2021) 106706.

[43] Ronald J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (3) (1992) 229–256.

[44] Richard S. Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, second ed.., MIT, Massachusetts Ave, MA, 2017.

[45] Shixiang Gu, Timothy P. Lillicrap, Zoubin Ghahramani, Richard E. Turner, Bernhard Scholkopf, Sergey Levine, Interpolated policy gradient: Merging on-policy and off-policy gradient estimation for deep reinforcement learning, in: Advances in Neural Information Processing Systems, 2017, pp. 3846–3855.

[46] Shihui Li, Yi Wu, Xinyue Cui, Honghua Dong, Fei Fang, Stuart Russell, Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 4213–4220.

[47] Seyed Sajad Mousavi, Michael Schukat, Enda Howley, Traffic light control using deep policy-gradient and value-function-based reinforcement learning, IET Intell. Transp. Syst. 11 (7) (2017) 417–423.

[48] Haokun Chen, Xinyi Dai, Weinan Zhang, Han Cai, Xuejian Wang, Ruiming Tang, Yuzhou Zhang, Yong Yu, Large-scale interactive recommendation with tree-structured policy gradient, in: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 3312–3320.

[49] Andrew G. Barto, Sridhar Mahadevan, Recent advances in hierarchical reinforcement learning, Discrete Event Dyn. Syst. 13 (1) (2003) 41–77.

[50] Richard S. Sutton, Doina Precup, Satinder Singh, Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning, Artificial Intelligence 112 (1) (1999) 181–211.

[51] Thomas G. Dietterich, Hierarchical reinforcement learning with the MAXQ value function decomposition, J. Artificial Intelligence Res. 13 (1) (2000) 227–303.

[52] Ronald Parr, Stuart J. Russell, Reinforcement learning with hierarchies of machines, in: Advances in Neural Information Processing Systems, 1997, pp. 1043–1049.

[53] Richard S Sutton, David A. McAllester, Satinder P. Singh, Yishay Mansour, Policy gradient methods for reinforcement learning with function approximation, in: Advances in Neural Information Processing Systems, 2000, pp. 1057–1063.

[54] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, Tat-Seng Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.

[55] Santosh Kabbur, Xia Ning, George Karypis, FISM: factored item similarity models for top-N recommender systems, in: Proc. ACM SIGKDD Conf., 2013, pp. 659–667.

[56] Christian Lücken, Benjamín Barán, Carlos Brizuela, A survey on multi-objective evolutionary algorithms for many-objective problems, Comput. Optim. Appl. 58 (3) (2014) 707–756.