# Dynamic User Profiling for Personalized Tourism Recommendations Using Reinforcement Learning Models

Tommaso Ferrario, Elisabetta Fersini, Enza Messina, Gabriele Sormani

**Abstract** The exponential growth of tourism-related information available through digital platforms has created challenges in filtering relevant data to provide personalized recommendations. This paper addresses the dynamic nature of user preferences in tourism by proposing two reinforcement learning (RL) models for dynamic user profiling: a Hierarchical Reinforcement Learning (HRL) model with Matrix Factorization and a Double Deep Q-Network (DDQN) model. These models aim to enhance the recommendation of Points of Interest (POIs) by adapting user profiles based on evolving interests across Topics of Interest (TOIs). Experimental evaluations, using an adapted version of MovieLens 100K dataset, demonstrate that the HRL model excels in updating profiles for users with extensive interaction history, while the DDQN model is more effective for new or less active users. This work highlights the potential of RL models for dynamic user profiling in improving personalized tourism recommendations and offers insights into the suitability of different models based on user interaction patterns.

Tommaso Ferrario
Università degli Studi Milano-Bicocca, Viale Sarca 336, Milano, e-mail: t.ferrario@campus.unimib.it,
Elisabetta Fersini
Università degli Studi Milano-Bicocca, Viale Sarca 336, Milano, e-mail: elisabetta.fersini@unimib.it
Enza Messina
Università degli Studi Milano-Bicocca, Viale Sarca 336, Milano, e-mail: enza.messina@unimib.it
Gabriele Sormani
Università degli Studi Milano-Bicocca, Viale Sarca 336, Milano, e-mail: g.sormani6@unimib.campus.it

## 1 Introduction

In recent years, the amount of tourism-related information available through digital services has increased exponentially, leading to an unprecedented growth in data accessibility [8, 5]. While this copious amount of information serves as a valuable resource, it also introduces challenges, particularly in terms of information filtering. The vast quantity of data available from different sources like online booking platforms, social media and online travel agencies often makes it challenging to distinguish relevant from irrelevant information. A significant portion of the data consists of the feedback that users give to points of interest (POIs). The main objective is to target new POI recommendations to users who have in the past expressed an interest in tourist attractions with similar characteristics. To address this issue, personalized services have been developed, leveraging individual characteristics and preferences contained in users' profiles. Personalized systems facilitate the delivery of user-tailored information and services [2]. User profiles can be constructed using different sources of information, such as demographic data and other dynamically changing attributes, including user choices and feedback. Tourist preferences may shift based on their location, time of year, and other contextual factors. A static profiling system may provide recommendations that fail to align with these evolving interests. This misalignment can lead to a suboptimal user experience and reduce the overall effectiveness of the recommendation system. To capture and take into account these possible changes, profiles need to be updated and adapted accordingly to maintain their accuracy and relevance. For this reason, dynamic user profiling systems have been developed [11].

Dynamic user profiling is grounded in the sequential interaction between users and personalized systems, where the system's actions (recommendations) are followed by user responses (choices and/or feedback). Within this framework, dynamic user profiling can be modeled as a sequential decision-making and reward-optimization problem, for which Reinforcement Learning (RL) presents a promising solution.

Reinforcement Learning is a type of machine learning where an agent learns to make decisions by interacting with an environment. In this context, the RL agent is represented by the user profiler which dynamically updates the user profile that is used by the Recommender System (RS) to suggest POIs. The environment is represented by the user which can reward the RS suggestions either implicitly, by choosing to visit the suggested POI, or explicitly, by leaving feedback. These interactions will be used by the RL agent to update the user profile.

In this paper, we present two RL-based models for dynamic user profiling specifically tailored to enhance touristic recommendation systems.

The structure of the paper is the following. The formulation of the problem is introduced in Section 2. The proposed models are described in Section 3. Section 4 provides an overview of the experimental setup, including dataset construction and evaluation metrics, and Section 5 reports the experimental results along with their discussion. Finally, Section 6 concludes with a discussion of the findings and potential future research directions.

## 2 Problem definition

A user profile is a structured representation of a user's characteristics, behaviours, and interests [1, 4]. Since user interests and behaviour may change over time depending on several factors, as for example seasonality, past experiences, new trends, current needs and purpose of the travel, it becomes important to consider this dynamic nature when building user profiles. In this paper, we address this issue by modeling user profiling as a RL problem.

We define a user profile as a set of interest values across a specific set of Topic of Interests (TOIs). TOIs refer to specific areas, themes, or subjects that represent the preferences of tourists and form the basis for their destination choices. These topics may include a wide range of interests, as for example, museum and art galleries, adventure and outdoor activities, wellness, gastronomy, and more. They provide a framework for understanding the diverse motivations behind travel behaviour, enabling stakeholders to recommend specific POIs to design tailored experiences.
Given a predefined set $C = (c_1, \ldots, c_{|C|})$ of TOIs every POI is mapped on the TOI domain, meaning that each POI $i$ is represented by a boolean vector $POI^i = (POI^i_1, \ldots, POI^i_{|C|})$ where element $POI^i_j = 1$ if POI $i$ belongs to TOI $j$ and $POI^i_j = 0$ otherwise. The profile $P(t,u) = (p_1, \ldots, p_{|C|})$ of the user $u \in U$ at time $t$, is also mapped on the TOIs domain. The profile $P(t,u)$ is then used by the Recommender System to suggest the next recommendations of POIs.
The feedback of user $u$ at time $t$ on a POI (or on a set of POIs) are also mapped on the TOIs' domain obtaining $F(t,u) = (f_1, \ldots, f_{|C|})$.
This topic-based modeling approach allows for efficient profile modeling and enables inference on an arbitrary set of POI also facilitating its integration in Recommender Systems.

In this setting, RL can be employed to dynamically build the user profile $P(t,u)$ which represents the state of the system at time $t$. $P(t,u)$ is used by the RS to suggest a list of POI (which represents the action), and returns to the Profiler the feedback vector $F(t,u)$. Feedback is compared with the prediction of the Profiler to obtain a reward that is used to derive the new profile $P(t+1,u)$. See Figure 1.

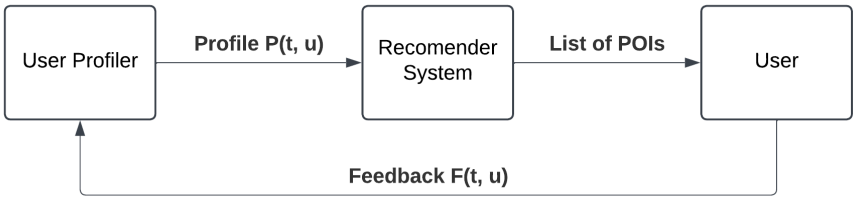



**Fig. 1** Schema of the Reinforcement Learning based dynamic user profiler.

In this paper, we present two different approaches to derive the current state $P(t,u)$ from the past history of interactions, which is then used for making a pre-

diction, denoted as $\hat{P}(t+1,u)$ on user preferences. The first approach, based on Hierarchical RL (HRL)[18], filters the user's past interactions to select the relevant POIs that are then mapped onto a TOI space to obtain the user profile. This approach, referred to as *HRL_based*, relies on having a substantial amount of previous data and it is not suitable for capturing the profile of a new user for which only limited information is available. For this reason, we propose a second approach, based on Deep Q-network (DQN)[16], that defines user profiles directly on the TOI space, using the user feedback within a limited moving time window (limited memory), we refer to this model as *DQN_based*.

To evaluate a profiling model, we need to define for each user $u$ a ground truth profile $G(t,u) = (g_{t,1}, \ldots, g_{t,|C|})$, $t = 0, \ldots, T$. In particular, given the feedback $F(t,u)$ the ground truth profile $G(t,u)$ at time $t$, computed on a time window of size $n$, is defined as:

$$G(t,u) = \left\lceil \frac{\sum_{t=\tau}^{\tau+n} F(t,u)}{n} \right\rfloor \tag{1}$$

This vector represents the average of the user's ratings for each TOI over a defined period $(\tau, \tau+n)$, rounded to the nearest integer.

In the following section, we describe in detail the proposed profiling models.

## 3 Dynamic user profiling

### *3.1 Filtering users' past interactions*

This approach applies RL to estimate for each user $u$ a relevance score associated to POIs in order to generate a profile $P(t,u)$ that adapts to possible changes in the user's preferences. The interactions between the user and the system are represented by the feedback $F(t,u)$ derived from the rates that the user has expressed on POIs explicitly or implicitly. In this model, a RL agent predicts $\hat{P}(t+1,u)$ by filtering the subset of past interactions (feedback) to be included in the current user profile.

The objective of the agent is to minimize the error between $\hat{P}(t+1,u)$ and the ground truth $G(u,t+1)$, measured as the *Root Mean Squared Error* (RMSE), given by:

$$RMSE\left(\hat{P}(t+1), G(t+1)\right) = \sqrt{\frac{\sum_{i=1}^{|U|}\left(\hat{P}(t+1,i) - G(t+1,i)\right)^2}{|U|}} \tag{2}$$

The agent is based on a Hierarchical Reinforcement Learning (HRL) [18] architecture, that performs two tasks: a high-level task and a low-level task. These two tasks pursue different objectives that are expressed through the actions of the agent. Specifically:

- The high-level action decides whether or not to filter user past interactions. If no filtering is applied, all past user interactions are considered for profiling. The high-level action is expressed as a binary value $a_{HL} = \{0, 1\}$.
- If $a_{HL} = 1$, then a low-level action must select the subset of feedbacks (interactions) to be considered to build the user profile $\hat{P}_{filtered}(t+1, u)$.

Each POI $i$ is represented through embeddings $E(POI^i)$. Once the relevant feedbacks of a user $u$ are selected, a latent vector for the user $E(P(t,u))$ is obtained by aggregating the selected POIs embeddings as:

$$E(P(t,u)) = \sum_{k=0}^{t} a^u_{i,k} E(POI^i_k) \tag{3}$$

where $E(POI^i_k)$ is the embedding of $POI^i$ which received a feedback at time $k$ and $a^u_{i,k}$ is an attention coefficient. This method of computing the user profile for a POI corresponds to the NAIS (Neural Attentive Item Similarity) [9] model.

The reward $R$ is defined as:

$$R = \begin{cases} RMSE\left(\hat{P}(k,u), G(k,u)\right) - RMSE\left(\hat{P}_{filtered}(k,u), G(k,u)\right) & \text{if } k = t \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $t$ is the time step of the last user feedback and $\hat{P}_{filtered}(k,u)$ and $\hat{P}(k,u)$ represents the user profile obtained by filtering the past interactions or the complete history, respectively. Note that if filtering the user's feedback leads to a lower error compared to the unfiltered past, the reward is positive, while it is negative in the opposite case.

The profile $P(t,u)$ obtained may contain null values. This is because since the profile is defined in the TOI domain, if none of the selected POIs belongs to a given TOI $j$, then the value of the corresponding element $c_j$ can be zero. In this case, a Matrix Factorization model is used to fill the missing values.

In order to train the high-level and low-level RL models jointly with the NAIS model, we use the REINFORCE algorithm [17].

The network architecture of high-level and low-level models consists of two fully connected layers, of 8 neurons each. The input layer sizes are 34 and 18 respectively. Activation functions used are ReLU for the intermediate layers and sigmoid for the output layer. The implementation of the NAIS recommender system used to compute the state and estimate $E(P(t,u))$ is an attention network that estimates the attention coefficient $a^u_{i,k}$. This network has two layers composed of 16 neurons, with ReLU activation functions except for the output layer that uses a Softmax.

## *3.2 Estimating users' interests values*

The model presented in the previous section requires a considerable sequence of feedback to effectively build a user profile. In tourist RS this is not always the case.

Many users may be new or have had just a few interactions with the system. In this case, filtering the information can introduce a lot of noise and is not recommendable. For this reason, we present another RL-based model that can be used also when only a short history is available. To infer $\hat{P}(t+1,u)$, we integrate all $F(t,u)$ of user $u$ belonging to a given interval of time.

The RL algorithm chosen to implement this scenario is *Double Deep Q-Networks* (DDQN). We opt for this approach instead of the more traditional *Deep Q-Networks* (DQN) because one of its main characteristics is mitigating the overestimation of Q values, afflicting DQN, thus learning optimal local policies more easily [21, 16]. Due to the high computational cost of estimating the optimal policy for each individual user, a global policy is typically calculated for all users [12].

To reduce the instability in training, the DDQN uses two neural networks: Q-network and Target network, which are combined to estimate the Value Function. The target network is used to estimate the target value of the Action-Value Function. The parameters of this network are updated only every $K$ step by copying the parameters from the Q-Network. This makes the oscillations during training less frequent [14]. The Q-values considered to choose an action are computed as the average of both networks [7]. An $\varepsilon$-greedy policy, commonly adopted in literature [12], was implemented to promote exploration. This policy specifies that at each step $t$ the agent chooses, for each TOI in the profile, a value that can be randomly chosen with probability $\varepsilon \in [0,1]$ or the best value predicted using $P(t-1,u)$, with probability $(1-\varepsilon)$. Typically $\varepsilon$ has a high initial value which is gradually reduced.

The next state, i.e. the profile $P(t,u)$, is formed by using a Recurrent Neural Network (RNN) model and is given as input to the Target network and the Q-Network to update the profile. The profile $P(t,u)$ comprises the POIs the user has interacted with in a given time window, chronologically ordered. These POI sequences are fed into Gated Recurrent Units (GRU) [19, 10, 20, 3] to better capture the sequential nature of user behaviour. In particular, RNN units correlate the input of the current step with the state of the previous step, to capture the sequence of the most recent user's interests.

The aim of the RNN is to provide the latent representation of the most recent user's feedback $F(t,u)$, $t = (t-n),\ldots,t$. In this work, the size $n$ of the time window was set to 10 (in the case a user has not 10 previous interactions, the missing interactions and ratings are filled with zeros). Feedbacks provided to the RNN are composed of the one-hot encoding of the POI-TOI association and of the POI's rating given by the user. The DQN model also employs a next state for calculating the loss function. In this case, the next state is defined in the same way as the current state, with the only difference that the input of the RNN also considers the interactions $F(t,u)$ at the current time $t$.

The reward (R) function compares the inferred value and the ground truth value, and is defined as:

$$R = -|G(t,u) - \hat{P}(t,u)| \tag{5}$$

The absolute value is used to determine the distance of the inferred value from the correct one and is negated to express a penalty. Obviously, the highest possible value for a reward is 0. The greater the error, the more negative the reward becomes.

To construct a user profile, it's sufficient to infer a value for each TOI $c \in C$.

The structure of both the Q-network and the Target-network consists of two layers, with 256 and 128 neurons, respectively. The activation function used in both layers is ReLU. These networks receive the embedded state generated by a RNN, formed by a vector of 20 elements concatenated with a vector of $|C|$ elements. The RNN is composed of two layers: the first is a GRU layer with 128 units, while the second is a fully connected layer, also with 128 units. The input of the RNN consists of a one-hot vector representing the TOI and the rating given by the user. The learning rate was set to 0.001.

## 4 Compuational Experiments

In the following section, we outline the experiments performed to evaluate the previously presented models. Since no dataset on touristic user profiling systems was available, we adapted the benchmark dataset MovieLens 100K [6], a dataset commonly used to benchmark recommendation systems. Movie names in MovieLens are used for POIs as an alternative to place names, and genres are used for TOI as an alternative to categories of interest. We considered as feedback the ratings available in the gold standard. To show that our models are capable to better adapt to changes in user interests we compared them with state-of-the-art models in recommendation systems, including Top popularity, User KNN, EASE [15], MultiVAE [13] and Matrix Factorization (MF). The code is available on Github (https://github.com/MIND-Lab/Tourist-Profiler)

### *4.1 Performance metrics*

The quality of the profiles inferred by the models was assessed by measuring their deviation from the ground truth profiles. This deviation was quantified using the root mean squared error (RMSE), as defined in Equation 2. To provide a comprehensive view of the results of our analysis, we defined three distinct performance metrics. These metrics differ in how they account for elements that are equal to zero in the profiles, indicating the absence of information:

- **Total RMSE**: This metric measures the difference between the values of the inferred profile $P(t,u)$, for $t = 0,\ldots,T$, and the nonzero values of the ground truth profile $G(t,u)$, as defined in Equation 1, over the same period. Its purpose is to capture the overall discrepancy between the inferred profile and the ground truth.

- **Explicit RMSE**: This error measures the difference between the values of the inferred profile $P(t,u)$, for $t = 0,\dots,T$, and the nonzero values of the ground truth profile $G(t,u)$ that were also nonzero in $G(t-1,u)$. It assesses the model's ability to track changes in the user's interests regarding topics for which feedback has previously been provided.
- **Implicit RMSE**: This error measures the difference between the values of the inferred profile $P(t,u)$, for $t = 0,\dots,T$, and the nonzero values of the ground truth profile $G(t,u)$ that were zero in the previous time step $G(t-1,u)$. It evaluates the model's ability to anticipate user preferences for new topics.

The Explicit RMSE evaluates the model's ability to capture changes in the user's interest for familiar topics, while the Implicit RMSE assesses the ability to predict user preferences for new or unseen topics.

## *4.2 Numerical Results*

To test our models, we performed several experiments using different numbers of TOIs, in particular $C = 6, 12$ and 18. This allowed us to understand the ability of the models to predict more detailed profiles. The categories were selected by prioritizing the most frequently occurring categories in the dataset. Tables 1, 2, and 3 display the results for $C = 6$, 12, and 18, respectively, with the best results highlighted in bold. Specifically, these tables display the RMSE values obtained from the models, reported as mean values along with their corresponding standard deviations.

**Table 1** Results obtained by considering only the 6 most populated categories of the dataset.

| Model | Total RMSE | Explicit RMSE | Implicit RMSE |
|---|---|---|---|
| Top popularity | $0.7728 \pm 0.5082$ | $0.7726 \pm 0.5083$ | $1.0000 \pm 0.7906$ |
| User KNN | $\mathbf{0.2850 \pm 0.2751}$ | $\mathbf{0.2847 \pm 0.2737}$ | $1.3942 \pm 0.9811$ |
| EASE | $0.4956 \pm 0.2120$ | $0.4956 \pm 0.2118$ | $1.2401 \pm 0.7412$ |
| MultiVAE | $0.6700 \pm 0.3305$ | $0.6698 \pm 0.3305$ | $1.0718 \pm 0.5668$ |
| MF | $0.5889 \pm 0.3099$ | $0.5886 \pm 0.3098$ | $1.1129 \pm 0.6755$ |
| HRL NR + MF | $0.5321 \pm 0.3610$ | $0.5317 \pm 0.3611$ | $1.1119 \pm 0.6756$ |
| DQN | $0.5737 \pm 0.3629$ | $0.5735 \pm 0.3631$ | $\mathbf{0.8788 \pm 0.7398}$ |

Experiments with 12 and 18 TOIs demonstrate that the *HRL_based* model with Matrix Factorization outperforms in terms of Explicit RMSE. This indicates that the model is particularly effective at estimating the scores for topics already explored by the user, making *HRL_based* well-suited for detecting changes in users' interests in familiar topics. Practically, these results suggest that *HRL_based* is better suited to update the profiles of users who have engaged extensively with the system.

Another key finding is that the *DQN_based* model consistently achieves the best performance in terms of Implicit RMSE across all experiments. This highlights the

**Table 2** Results obtained by considering only the 12 most populated categories of the dataset.

| Model | Total RMSE | Explicit RMSE | Implicit RMSE |
|---|---|---|---|
| Top popularity | $0.8810 \pm 0.4686$ | $0.8844 \pm 0.4706$ | $1.0410 \pm 0.9108$ |
| User KNN | $0.7966 \pm 0.3037$ | $0.7975 \pm 0.3039$ | $1.3045 \pm 0.9157$ |
| EASE | $0.6675 \pm 0.2022$ | $0.6695 \pm 0.2032$ | $1.1501 \pm 0.7911$ |
| MultiVAE | $0.7662 \pm 0.3589$ | $0.7692 \pm 0.3615$ | $1.0503 \pm 0.7001$ |
| MF | $0.6789 \pm 0.2934$ | $0.6815 \pm 0.2967$ | $1.0172 \pm 0.7599$ |
| HRL NR + MF | $\mathbf{0.6569 \pm 0.3113}$ | $\mathbf{0.6593 \pm 0.3154}$ | $1.0161 \pm 0.7905$ |
| DQN | $0.7410 \pm 0.3334$ | $0.7442 \pm 0.3364$ | $\mathbf{0.9344 \pm 0.8436}$ |

**Table 3** Results obtained by considering only the 18 most populated categories of the dataset.

| Model | Total RMSE | Explicit RMSE | Implicit RMSE |
|---|---|---|---|
| Top popularity | $0.9336 \pm 0.4479$ | $0.9430 \pm 0.4515$ | $1.0049 \pm 0.8971$ |
| User KNN | $1.2672 \pm 0.2813$ | $1.2662 \pm 0.2804$ | $1.7153 \pm 1.1979$ |
| EASE | $0.9981 \pm 0.2428$ | $0.9980 \pm 0.2412$ | $1.6019 \pm 1.0477$ |
| MultiVAE | $0.8175 \pm 0.3441$ | $0.8264 \pm 0.3490$ | $1.0076 \pm 0.6995$ |
| MF | $0.7166 \pm 0.2764$ | $0.7249 \pm 0.2839$ | $0.9728 \pm 0.7196$ |
| HRL NR + MF | $\mathbf{0.6963 \pm 0.3031}$ | $\mathbf{0.7043 \pm 0.3124}$ | $0.9530 \pm 0.7481$ |
| DQN | $0.7946 \pm 0.3024$ | $0.8035 \pm 0.3081$ | $\mathbf{0.9137 \pm 0.7753}$ |

model's strength in predicting the value of new topics for users based on the sequence of their previous interactions. In practical terms, this makes the *DQN_based* more suitable for new user profiles or those with limited interaction history with the recommender system.

We also observed that among the benchmark models, those using distance as a comparison metric, namely UserKNN and EASE, tend to outperform RL-based methods when working with only 6 TOIs. Given the construction of the dataset used in these experiments, it is reasonable to assume that changes in user preferences are more evident when the user profile is more detailed.

## 5 Conclusions

In this work, we presented and compared RL-based approaches for dynamic user profiling. Our focus on dynamic profiling aims to capture the evolving nature of tourism data and to better understand users' shifting interests. We demonstrated that, in most cases, our RL models outperform traditional models used in state-of-the-art profiling systems. Throughout the study, we inferred user profiles composed of a set of Topics of Interest (TOIs), which were then utilized as input for a RS for suggesting locations and attractions (POIs). The results indicate that both proposed models are promising for dynamic profiling, but tailored to different scenarios. Specifically, the DQN model is better suited for profiling new users or those with limited inter-

actions, while the HRL model excels at maintaining and updating the profiles of highly active users.

**Acknowledgements** This work was supported by the research project RASTA: Realtà Aumentata e Story-Telling Automatizzato per la valorizzazione di Beni Culturali ed Itinerari; funded by the Italian Ministry of University and Research (MUR) under the PON Project ARS01_00540.

## References

1. Eke, C., Norman, A., Shuib, L. & Nweke, H. A Survey of User Profiling: State-of-the-Art, Challenges, and Solutions. *IEEE Access*, **7**, 144907-144924 (2019).
2. Farid, M., Elgohary, R., Moawad, I. & Roushdy, M. User profiling approaches, modeling, and personalization. *Proceedings Of The 11th International Conference On Informatics & Systems (INFOS 2018)*, (2018).
3. Fu, M., Agrawal, A., Irissappane, A., Zhang, J., Huang, L. & Qu, H. Deep Reinforcement Learning Framework for Category-Based Item Recommendation. *IEEE Transactions On Cybernetics*. **52**, 12028-12041 (2022).
4. Gatziolis, K., Tselikas, N. & Moscholios, I. Adaptive User Profiling in E-Commerce and Administration of Public Services. *Future Internet*. **14** (2022).
5. Godoy, D. & Amandi, A. User profiling in personal information agents: A survey. *The Knowledge Engineering Review*. **20**, 329-361 (2005).
6. Harper, F. & Konstan, J. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*. **5** (2015).
7. Hasselt, H. Double Q-learning. *Advances In Neural Information Processing Systems*, **23** (2010).
8. Hawalah, A. & Fasli, M. Dynamic user profiles for web personalisation. *Expert Systems With Applications*. **42**, 2547-2569 (2015).
9. He, X., He, Z., Song, J., Liu, Z., Jiang, Y. & Chua, T. NAIS: Neural attentive item similarity model for recommendation. *IEEE Transactions On Knowledge And Data Engineering*. **30**, 2354-2366 (2018).
10. Huang, L., Fu, M., Li, F., Qu, H., Liu, Y. & Chen, W. A deep reinforcement learning based long-term recommender system. *Knowledge-Based Systems*. **213** pp. 106706 (2021).
11. Koren, Y. Collaborative filtering with temporal dynamics. *Proceedings Of The 15th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*. **53**, 447-456 (2009).
12. Lei, Y. & Li, W. Interactive Recommendation with User-Specific Deep Reinforcement Learning. *ACM Transactions on Knowledge Discovery from Data*, **13** (2019).
13. Liang, D., Krishnan, R., Hoffman, M. & Jebara, T. Variational Autoencoders for Collaborative Filtering. *Proceedings Of The 2018 World Wide Web Conference*, 689-698 (2018).
14. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G. & Others Human-level control through deep reinforcement learning. *Nature*. **518**, 529-533 (2015).
15. Steck, H. Embarrassingly Shallow Autoencoders for Sparse Data. In *The World Wide Web Conference Proceedings*, Association for Computing Machinery, 3251–3257 (2019).
16. Wang, D., Wang, P., Fu, Y., Liu, K., Xiong, H. & Hughes, C. Reinforced Imitative Graph Learning for Mobile User Profiling. *IEEE Transactions On Knowledge And Data Engineering*. **35**, 12944-12957 (2023).
17. Williams, R. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*. **8**, 229-256 (1992).

18. Zhang, J., Hao, B., Chen, B., Li, C., Chen, H. & Sun, J. Hierarchical reinforcement learning for course recommendation in MOOCs. *Proceedings Of The AAAI Conference On Artificial Intelligence*. **33**, 435-442 (2019).
19. Zhao, X., Xia, L., Zhang, L., Ding, Z., Yin, D. & Tang, J. Deep reinforcement learning for page-wise recommendations. *Proceedings Of The 12th ACM Conference On Recommender Systems*, 95-103 (2018).
20. Zhao, X., Zhang, L., Ding, Z., Xia, L., Tang, J. & Yin, D. Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning. *Proceedings Of The 24th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining*, 1040-1048 (2018).
21. Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N., Xie, X. & Li, Z. DRN: A Deep Reinforcement Learning Framework for News Recommendation. *Proceedings of The 2018 World Wide Web Conference*, 167-176 (2018).